M.I.T. LABORATORY FOR COMPUTER SCIENCE

September 24, 1980

Computer Systems Research

Request for Comments No. 199

SOURCE ROUTING FOR CAMPUS-WIDE INTERNET TRANSPORT

by J.H. Saltzer, D.D. Reed, and D.D. Clark

This is preprint of a paper presented at the IFIP Working Group 6.4 Workshop on Local Area Networks in Zurich, August 27-29, 1980. It will appear in the published proceedings of that workshop.

SOURCE ROUTING FOR CAMPUS-WIDE INTERNET TRANSPORT

by

Jerome H. Saltzer
David P. Reed
David D. Clark

Massachusetts Institute of Technology
Laboratory for Computer Science
545 Technology Square
Cambridge, Massachusetts 02139

15 September 1980

Abstract: For the internet addressing layer of a campus-wide local area network, a source routing mechanism may have several advantages over hop-by-hop routing schemes based on universal or hierarchical target addresses. The campus environment requires many subnetworks connected by gateways, and it has relatively loose administration. The primary advantage of source routing in this environment is simplicity of implementation of the gateways that interconnect subnetworks with consequent improvement in cost, maintenance effort, recovery time, ease of trouble location, and overall management effort.

This preprint is of a paper presented at the IFIP Working Group 6.4 Workshop on Local Area Networks in Zurich, August 27-29, 1980. It will appear in the published proceedings of that workshop.

Introduction

This paper proposes that for the internet addressing layer of a campus-wide local area network, the source routing mechanism suggested by Farber and Vittal [1] and discussed by Sunshine [2] may have several advantages over hop-by-hop routing schemes based on universal or hierarchical addresses. The campus environment, which term is applicable to any multi-building site, requires many subnetworks connected by gateways, and it probably has a relatively loose administration. The primary advantage of source routing in this environment is simplicity of implementation of the gateways that interconnect subnetworks with consequent improvement in cost, maintenance effort, recovery time, ease of trouble location, and overall management effort. Secondary advantages of source routing when applied to the campus environment include: 1) a clearer separation of physical addressing from logical target identification mechanisms in protocol design, 2) elimination of stability, oscillation, and packet looping considerations, 3) ability for a source to control precisely a route so as to optimize a particular service goal (e.g., response time, reliability, bandwidth, usage policy, or privacy), 4) deferment to a higher protocol level of the detailed design of the fragmentation/reassembly strategy required to pass through intermediate networks with small maximum packet sizes, and finally, 5) the ability to accomodate both official and unofficial gateways between subnetworks.

This research was supported by the Defense Advanced Research Projects Agency of the United States Government and was monitored by the Office of Naval Research under Contract No. N00014-75-C-0661.

Two disadvantages of source routing are: 1) that the route used will tend to be relatively static and therefore cannot optimize use of communication facilities as well as the potentially more dynamic hop-by-hop route selection system, and 2) route selection must be accomplished somehow, and since this protocol level does not specify the mechanism, some additional mechanism must be designed to provide route selection. The argument made here is that the first disadvantage is not serious in an environment such as a campus, in which the low cost of high bandwidth communication can make optimization less important. The second disadvantage may be less serious than it appears when one considers that a higher-level target identification service is required in any case, which service can also provide route selection service. In fact it may be possible to turn this need into an advantage, since there can be more than one such route selection service, one of which is based on simple global or hierarchical network identifiers, while another, perhaps experimental or research service, provides an elaborate interactive directory search facility or a private route pattern.

This last ability to decouple target identifications and route selection from gateway implementation taken together with the other advantages cited suggests that the fundamental force at work in using source routes is improved modularity of network implementation.

This paper has three parts. The first explores the nature of the campus environment, especially its administrative properties. The second describes how the mechanics of source routes might work using routing services. The final part discusses the advantages that source routing seems to provide when applied to the campus environment. Readers familiar with the idea of source routing will find that the second section can quickly be skimmed; potentially novel observations are confined to the first and third sections.

I. What is a Campus Environment?

"The Campus Environment" is a name used here to identify a particular set of physical properties, geographical extents, data communication requirements, administrative relationships, and needs for flexibility that characterize our own university campus. With only minor exceptions they equally apply to a corporate site, a government complex, or another university. There seem to be seven characteristic properties of this campus environment that provide a basis for design decisions for a data communication network. As will be seen, the properties of this environment are quite different from those of a single building, or of a nation-wide, common-carrier-based network. The seven properties are:

1) limited geographical extent,
2) up to several thousand nodes,
3) forces for both commonality and diversity,
4) multiple protocols,
5) confederated administration,
6) independently administered interconnections, and
7) gateways to other nets.

The following sections explain and discuss each of these properties in turn.

1) Limited geographical extent. The campus environment has a geographical extent beyond a single building, but within a single political and administrative boundary that permits transmission media to be installed without resort to a common carrier.

This first property is essential, so as to allow exploitation of low-cost, high-bandwidth communication technology. With current technology and prices the difference in costs between communicating over privately installed equipment and using common carrier facilities can be a factor between 10 and 100.

2) Up to several thousand nodes. Within this geographical area, a large number of nodes--that is, computers, data sources, and data sinks--require interconnection. Today the number of such nodes may be in the range of ten to one hundred. Looking ahead to the advent of desktop computers, one may be faced with from a few hundred to several thousand nodes by the end of the next decade.

The combination of the previous two properties seems to make it inevitable that local interconnection technologies such as the ETHERNET [3], Century Data Bus [4], L.C.S. Ring net [5], HYPERCHANNEL [6], MITRENET [7], or the Cambridge Ring [8], cannot by themselves completely accomplish the required interconnection, since all such technologies that have so far been demonstrated have limitations on distance on the order of a thousand meters and limitations on node count on the order of a hundred nodes. Thus one would expect to use those technologies to attach clusters of nodes into subnetworks, for example all the nodes in a single building, and then install interconnections (gateways) among these subnetworks. For our own (M.I.T.) campus, one might envision by 1990 as many as 100 subnetworks each comprising an average of, say, 50 to 100 nodes, thus linking up to 10,000 stations. Subnetworks and gateways introduce the problem of how to route a message from a source node through a series of subnetworks and gateways, so that it ends up at a desired target node.

3) Forces for both commonality and diversity. Administratively, there exist forces both for commonality and for diversity of network attachment strategies.

The primary force for commonality is a desire to be able easily to set up communications between any pair of nodes on the campus. The primary force for diversity is that the choice of a computer, data source, or data sink typically pre-determines the technology of the network to which it must be attached, because off-the-shelf network hardware for that node may be available in only one technology. Further, some applications may have special requirements for some connections (e.g., high bandwidth) that can be met only with a particular network supplier's equipment, yet still need occasional "ordinary" connections to nodes elsewhere. Thus the emerging diversity of local networks will continue, and probably increase, rather than decrease, with time.

4) Multiple protocols. Although there are several ongoing standardization efforts, the worldwide academic, commercial, and regulatory community has not yet reached anything resembling a consensus on how networks should be organized, how protocols should be layered or how functions should be divided. Arguments range over issues ranging from obscure matters of taste, through fundamental technical disagreements about which requirements should have priority in design, to alternative opinions of the directions that communication technology is moving. Many different and competing standards have been proposed, and one can find in the literature a good technical case against any one of them. One must anticipate that these arguments will be reflected internally in the campus environment, in the form of a diversity of protocols and

standards, and particularly in the requirement that any mutually consenting* set of nodes be able to carry on communication with one another using a protocol that no one else has ever heard of, much less agreed to.

The diversity of protocols arises for much the same reason as the previously-described diversity of network hardware. The foremost consideration in acquiring a computer is usually how well it meets the immediate application requirement. Ability to communicate with other, not-yet-integrated, applications is a low priority consideration. If the best-looking computer comes with a particular supplier's collection of provincial protocols, the purchaser will tend to question them only if it appears that they might hamper the initial application. Otherwise, he will plan to postpone thinking about interconnection until some real requirement appears, and after the equipment and its protocols have been installed.

This protocol diversity suggests strongly that any network interconnection strategy that must be implemented today should have at the lowest possible layer a campus-wide protocol that accomplishes communication between any two nodes while making an absolute minimum number of assumptions about the higher-level nature of the communications that are taking place or the policy of network administration. Some typical assumptions that should be avoided unless an unusual opportunity can be taken are: what level of reliability/delay tradeoff is appropriate; how routing should be optimized; fragmentation/reassembly strategy; flow control requirements; addressing plan; and particular network topology.

---

* Imagery borrowed from a Chaosnet working paper by David Moon.

5) Confederated administration. Because a data communication network is a campus-wide service, there will be no single user or user group with a wide-enough interest to administer the entire network. This means that network administration will either be done by a haphazard confederation of special interest groups or else by a chronically underfunded central service organization modeled on the one whose role is to minimize telephone costs.

In either case, this property places a requirement on the network interconnection technology that it be robust and self-surviving to every extent imaginable. Trouble isolation must be easy to accomplish and easy for individual users to participate in if they are so inclined, because trouble isolation and repair may involve multiple administrations. Simplicity of operation of gateways is important, so that operation can be completely unattended for long stretches of time. Although some central monitoring of network operations can be very helpful in isolating problems a network design approach that requires close monitoring is undesirable.

6) Independently administered interconnections. The topology of subnetwork interconnection will be administered partly with central planning and partly without.

This property arises from two needs: First, a "dependable" set of gateways that one can expect to exhibit predictable and stable properties is an essential backbone to a useful service. A centrally planned and administered set of gateways would provide this dependability. Second, whenever a node finds that for some reason it is attached to two subnetworks, it may find that it is useful in some of its applications to serve also as a gateway between the subnetworks; yet it may not want to take on the official responsibility

of being a publicly available gateway. Another example of a gateway that is not centrally administered may arise if some particular application needs, and has purchased the gateway equipment to support, a path through the network with special properties of delay, reliability, bandwidth, or privacy. The person or organization that has purchased the special gateway equipment may not be prepared or willing to allow public use of it. A related requirement is that a user may wish to avoid use of a sometimes troublesome gateway that is claimed by its owner to be perfectly operating.

7) Gateways to other nets. External, public data networks such as TELENET, the ARPANET, TYMNET, XTEN, SBS, DATAPAC, or A.C.S., may be attached to some nodes, and some of those nodes will serve as gateways between the campus network and the external networks. In some cases, the external network will be used simply as a "long link" in the campus net. In other cases, facilities within the campus net will set up communication paths to services having no other connection with or knowledge of the campus net. Both kinds of cases require careful consideration of the interactions between internal and external network properties.

Note that the campus environment has all these properties only if we assume the technological opportunity mentioned in point one: that low-cost hardware and media can provide communication paths in the range from 1 to 10 Mbits/sec. between any two points within the campus. Availability of interconnect media and subnetworks with this bandwidth has been demonstrated in several forms. Gateways that operate with such bandwidths may be harder to construct, and that concern is one of the considerations involved in developing a campus-wide net. Individual nodes that can sustain these data rates for very long are likely to be rare; software often limits the rate at which a node can act as

either a data source or data sink. Instead, the high bandwidth technology is to be exploited in two ways:

1) to provide enough capacity to handle the aggregate demand of many lower-bandwidth sources and sinks of data.

2) non-optimal strategies that are relatively simple to implement or administer can be considered; it is not a requirement that every bit of the available bandwidth be optimally utilized.

The availability of high bandwidth, together with lack of a requirement to use that bandwidth efficiently, is probably the most fundamental technical difference between the "campus-wide network" and the commercial long-haul data communication network, a difference that can lead to significantly different design decisions.

II. How Source Routing Works

1. The basic mechanism. Source routing among a collection of subnetworks is a mechanism that comes into play at a relatively low layer of protocol, sometimes called the "internet" layer.* Figure one illustrates this layer arrangement. The lower layers, which we may collectively call the "local transport" layers, constitute a protocol for delivery of a packet within a local subnetwork such as a single ETHERNET or ring net. Routing within the local transport protocol is usually accomplished by physically broadcasting the packet to all nodes on one subnetwork; any node that recognizes its own local transport address at the front of the packet will receive it.

---

*If one tried to interpret a collection of interconnected subnetworks according to the ISO reference model [9], source routing might appear somewhere in the "transport" layer.

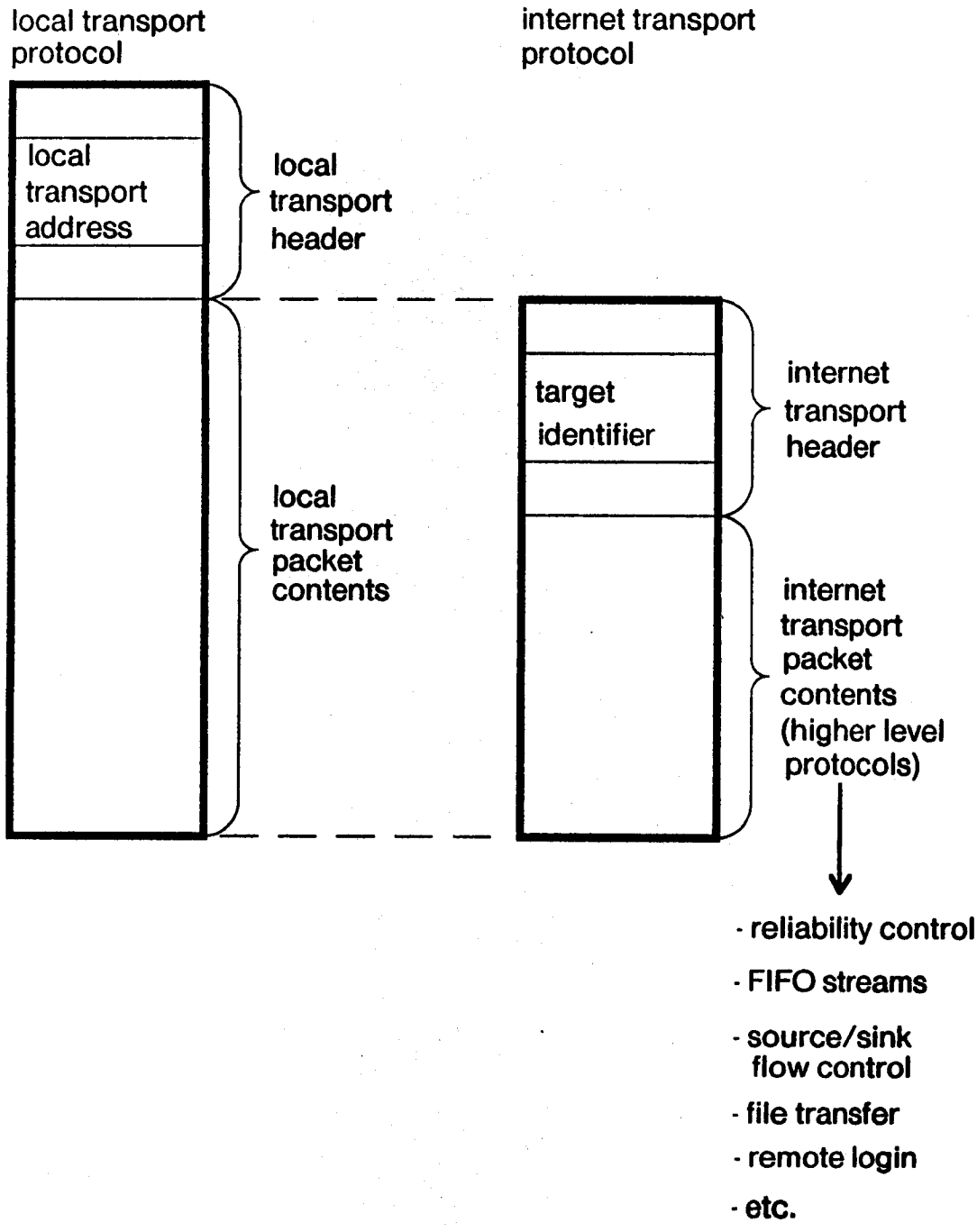

Figure 1 -- Relation between local transport protocol, internet transport protocol, and other communication protocols.

The intermediate, internet layer is a protocol for delivery of a packet between any pair of nodes on the campus. One starts a packet on its way by placing the address of a gateway in the local transport address field, and some form of identification of the target node in what may be called the "target identification" field. The local transport medium carries the packet to the gateway, which examines the target identification and determines what local transport address to use to get to the next gateway. In turn, the target identification is again interpreted by successive network gateways to determine which local transport address should be used for the next step of this packet's journey. This series of local transport addresses describes the route that the packet takes on its journey to the target.

There have been suggested several alternatives for the interpretation of target identifiers by gateways ranging, on the one hand, from a pure label of the target to, on the other hand, a description of some route from the source to the target. Three possibilities along this spectrum are:

1) Unstructured unique identifier. Every node on the campus-wide net has as its target identifier a permanent unique identifier. Each gateway has a set of tables or other rules that allow it to determine the appropriate next step in the route to every possible identified node. (Thus this approach is sometimes called "step-by-step" or "hop-by-hop" routing.) In its most general form, the unique identifier provides no routing information whatsoever. The unique identifier may be interpreted either as the identification of the target node or as the identification of the point on the network to which the target node is attached, depending on the network's convention on what happens to the identifier if a node is disconnected and reattached at a different place.

2) Hierarchical identifier. In this alternate form of hop-by-hop routing, the target identifier of each node is a multi-part field. For example, a two-part hierarchical identifier might consist of an identifier of the subnetwork to which the node is attached and a node number (usually the local transport address) of the node on that subnet. For this kind of target identifer, each gateway has a set of tables or rules that allow it to determine the appropriate next step in the route to every possible named subnetwork. Since there are many fewer subnetworks than nodes, these tables should be much smaller than in the case of the unstructured unique identifier. Reduction in table size is the chief attraction of the hierarchical identifier, and the argument can be extended to identifiers of more than two parts, network groups, and still smaller tables. Because the hierarchical identifier contains components that identify parts of the network, this kind of network identifier is almost always thought of as identifying the network attachment point, rather than the node that is attached to it.

3) Source route. The internet transport layer contains, in the place of the target identifier, a variable-length string of local transport addresses, with the property that each gateway merely takes the next local transport address from the string, moves that address to the local transport protocol address field, and sends the packet on its way. With this approach, a gateway needs no knowledge of network topology, so the tables required for hop-by-hop routing vanish. A source route unquestionably identifies a network attachment point, quite independently of what node is attached to that point. Any attempt to make an interpretation that a

source route identifies a node rather than an attachment point would be strained at best.

Note that if the network is arranged as a two-level hierarchy, with a single "supernet" acting as the only communication path among all the remaining subnetworks, then the two-part hierarchical identifier taken together with the local address of the nearest gateway to the supernet is an example of a source route and the gateways can become very simple. However, the hierarchical identifier can be used even if the network topology is not hierarchical, by providing an appropriate routing algorithm in the gateways. In that case, only the final part of the hierarchical identifier might be directly usable as part of the route; even it might actually be interpreted or mapped by the final gateway.

Note also, that it is common for a single node to have several activities underway at once. For example, a time-sharing system may have many logged-in users, several of which are using the network for communication between their terminal and the time-sharing system. The receiving network software in the time-sharing system then finds that it is acting as a kind of gateway, between the campus network on the one hand and the array of activities inside the node on the other. As a result it is commonly proposed that the target identifier not identify a node but rather a particular activity within that node. This proposal usually takes the form of an additional field in a hierarchical identifier, known as a "socket number" or "link". There is a controversy over what level of protocol should recognize this socket number, and how big it should be. For our purpose, it is sufficient to observe that the socket number is actually a kind of route for use by the receiving node.

The mechanics of operation of a source-routing gateway as a packet passes through are quite simple; this simplicity is the chief attraction of source routing. There are several alternative detailed approaches; to permit explicit discussion one implementation will be described here.* This implementation dynamically constructs a reverse route. It works as follows:

1) The internet source route field is structured as shown in figure two, with two one-octet numerical fields and a variable (but constant for the lifetime of the packet) number of octets of route. Each local transport address uses an integral number of octets, typically one or two. The first count is the number of octets in the route. The second count is the position of the next unused octet of the route. The first count remains constant for the lifetime of the packet; the second is updated at each gateway.

2) A gateway receives a packet using the local transport protocol of one network (call it network A) and wants to send it out on a second network (call it network B). For the moment, assume that a gateway interconnects exactly two nets; generalization for a multinet gateway involves a simple conceptual extension described in step 4, below.

3) The gateway parses the source route field using the "start of next local address" count to obtain the next step of the route. (We presume that the gateway is endowed with the knowledge of how many octets of route are required by network B.) It extracts the appropriate octets and places them in the local transport address field for network B. Then it replaces those octets of the internet source route with its own local

* This implementation is only a slight variation of the one proposed by Farber and Vittal.

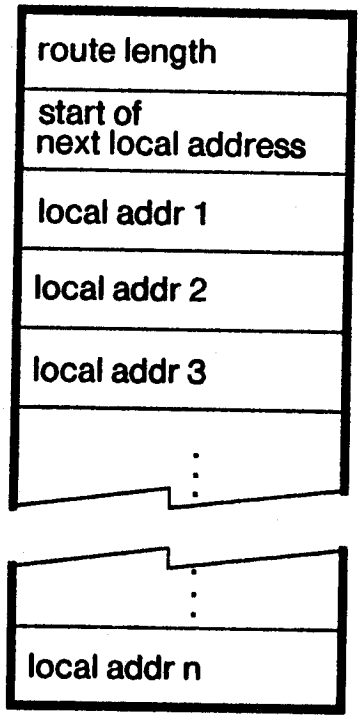

Figure 2 -- Possible implementation of an internet source route.

transport address on network B, thus contributing its part of the reverse route. Finally, it increments the "start of next address" field by the number of octets it extracted from the route, and it invokes the local transport level to send the packet out on network B. (Note that this reverse route construction strategy assumes that all paths are bi-directional and that all local transport addresses on any single network are of the same size. If the source route field is protected by a checksum, that checksum must also be recalculated. Finally, the reverse route comes out upside down, and would have to be turned upright before being used in a return message.)

4) If a gateway interconnects three or more subnetworks, it simply behaves as though it is itself a subnetwork with three or more gateways to other subnetworks. The next octet of route is interpreted as a local address on this hypothetical subnetwork. The reverse route is constructed as usual. (One should actually make even two-net gateways go through this step, so that expansion to three or more nets is easy and so that one can route packets to a gateway and back, for testing, as described later.)

The operation described above is repeated at every gateway, and may also be repeated one or more times inside the target node to dispatch the packet to the correct activity within that node. Similarly, when a packet originates, it may go through one or more route selection steps before it actually is placed on the first subnetwork.*

---

* From a viewpoint of telephone terminology, a source route system is a kind of electronically implemented step-by-step switch, with each subnetwork, multi-network gateway, or multi-activity host acting as a multi-position switch. However, because it is electronically implemented and thus not restricted to ten-position mechanical switches, this step-by-step switch does not have the limitations of the corresponding telephone technology.

2. Where Routes Come From. For source routing to work, the source of a message must somehow know what route to place in the internet header of a packet before it launches the packet into the internet environment. This requirement superficially implies that every source of packets be very knowledgeable, which sounds like a terrible burden to small nodes--every node on the network would have to be able to create or deduce suitable routes. In fact, that implication is unwarranted--all that is really required is that every source of messages know of a place in the network to ask to obtain routes. Once a source has learned of a suitable route to a particular target, it can encache that fact and reuse it as often and as long as it wants--until the route fails to work or there is a reason for it to believe that a better route exists.

Thus route selection would be accomplished by consulting a routing service somewhere in the network. A routing service would be provided by one (or more, for reliability, quick response, or administrative convenience) specialized node whose function is to maintain an internal representation of the topology of network interconnection (along with any useful class-of-service information about various subnetworks and gateways) and also to act as an identity resolver. The desired target must, of course, have some identifier, perhaps the unstructured unique identifier or hierarchical identifier earlier suggested as an alternative internet id. The routing service then implements a map from target identifiers to routes.

There are two independent dimensions along which this routing service may be more or less sophisticated: in its identifier-resolution abilities, and in its route-choosing abilities. To begin with, let us assume a particular fixed, fairly simple identifier resolution scheme--say a hierarchical

identifier--with the understanding that this choice has little or no bearing on routing sophistication. The routing choice mechanism, then, can range from a simple fixed table of routes from all possible sources to all possible targets (perhaps cleverly compressed with knowledge of the actual net topology) to a dynamic mechanism based on frequent exchanges of traffic statistics with gateways and other routing servers throughout the network.

Thus, to get started, a node that wants to originate messages needs to know one route: a route that can be used to send a request to a routing service to obtain other routes. It would be possible, though poor practice, to embed this "route to the nearest routing service" in the software of every node; a more general and flexible approach would be for a newly-arrived node to use either a broadcast or a breath-of-life strategy to discover this one route. In the broadcast strategy, a node broadcasts on its local transport network a request for the "route to the nearest routing service". For this particular broadcast route request, at least one gateway on every subnetwork is prepared to act as a rudimentary routing server. In the breath-of-life strategy each gateway periodically (say once every ten seconds) broadcasts over its local subnetworks a packet containing the route to the nearest routing service. A newly-operating node waits for the next breath-of-life packet before it can request its first route.

Having found a route to a routing service and then to a target node, if that node carries on more than one activity it may be necessary to hold a further negotiation with the target to learn how the target wants the source to identify the particular activity in which it is interested at the target. This negotiation probably takes place by sending a rendezvous packet to the host and receiving in return a packet that contains some extra routing steps

to be appended to the route originally obtained from the routing service. (Note that this protocol step is just the source-routing variation on a negotiation that takes place in every such protocol; it is not an extra step introduced by source routing.)

III. Advantages of source routes in the campus environment

1. Separation of routing from target identification. The main difference between source routing and its alternatives is that the responsibilities both of route choice and of target identification are moved from the internet gateways to some other agent. In turn, this responsibility change allows the internet transport protocol to be defined and the gateways to be implemented without freezing a particular form of network-wide identification of nodes or services. A commitment to a particular form of network-wide identification is made in the design of the identity resolution part of a routing service, and since it doesn't matter to a gateway where a route comes from (the gateway cares only that the next step works,) there can be more than one kind of identity resolution going on at the same time, perhaps implemented by different routing services. In practical situations there might be one centrally administered and widely-used identification method implemented by standard routing services, and in addition some experimental or special-purpose routing services developed for special applications or to experiment, for example, with interactive resolution of catalogued service identities, or protocols that allow sending one packet to more than one target node. These latter ideas, while likely of interest for the future, seem inappropriate to embed now in the internet transport protocol layer on grounds of inexperience. But they can be tried in the environment of a source-routing internet transport strategy without disruption and without change to the

gateways. It is even possible for one routing service to have a different view of the extent of the network from that seen by other routing services. Overlapping virtual networks are thus implementable using multiple source-route services. This feature might be used, for example, to segregate "local" communication paths from "long-distance" paths that involve routes through external, tariffed, networks.*

At the same time, the source route field format places little constraint on the format of the local transport addresses for any particular subnetwork--only that there be an integral number of octets whose number is known by the gateway that moves the packet to that subnetwork. This flexibility means that paths can go almost anywhere: in particular they can traverse "outside" networks no matter what their addressing or internal routing strategy, so long as at the far end of the outside network is a gateway that understands how to continue the packet on its journey.

Separation of the mechanics of routing from the functions implemented by a labeling or addressing system has the advantage of clarifying some frequent protocol design arguments that boil down to how much naming function should be embedded in the lowest protocol layers. For example, it is usually proposed that an extra field, for use within the target node, be carried along as part of the internet address. This field is known as a "link" field in the ARPANET

* Note that this separation of identity resolution from routing applies in both the case where internet identifiers label network attachment points and the case where internet identifiers label nodes or services. In the latter case, one can imagine also an additional layer of binding between attachment point identifiers (e.g., network addresses) and the internet node and service identifiers; this additional layer of binding could be the function of a service similar to the routing service described here. Although this extra modularity might have value in certain situations, one should understand that it is distinct from the modularity here imposed between routes and internet identifiers, whatever their form.

[10], the "channel" in X.25 [11], and the "socket" in ARPA's Internet for TCP [12] and in the internet layer of the Xerox PUP [13]. One argument develops over how big this field should be--just large enough to distinguish among the activities or connections a host carries on at one time, or generously large enough to distinguish among all activities or connections the host will ever carry on. The former choice takes the view that the field in question is merely the last step in a route, the latter choice makes the socket number a unique identifier, which is handling a labeling function for the host, perhaps allowing it to distinguish old connections from current ones. A second argument superficially concerns whether this field can be interpreted in different ways by different higher-level protocols. This argument is really one about whether the protocol level that can most efficiently perform the fan-out mechanics required is the same one that should be interpreting the labeling properties of the field.

The source routing strategy finesses both these arguments in that it allows the design of the packet format at the level of the internet transport layer address to be frozen without forcing a decision about socket number size or position in the protocol layering. As many octets of route as the target host needs to distinguish among its current connections can be included as part of the source route and learned as part of the initial negotiation with the target host using the initially obtained route to its negotiator. A unique identifier for a connection can be returned as part of that negotiation, and it can be included in a connection identifier field of the next higher level of protocol, to insure that packets arriving over a route are part of a current connection.

2. Gateway simplicity and network maintenance. With the source routing scheme just described, a gateway makes no decisions (possibly it should check to insure that the route octet count hasn't been exceeded) and it remembers nothing after the packet goes by. This simplicity of operation and lack of memory means that one can in principle implement such a gateway with a small amount of random logic and a pair of packet buffers interconnecting two local network hardware interfaces. Such an implementation, since it does not involve a stored program, has an exceptionally simple recovery strategy: a hardware reset to a standard starting state will always suffice. In practice, at least a microprocessor would probably be used to collect statistics and respond to trouble diagnosis requests, but the basic principle that recovery is trivial remains intact.

(There is one way in which a source-routing gateway is more complex than its hop-by-hop counterpart. Every packet that arrives may have a different source route size and different next step offset, so a small amount of lookup is needed to perform the forwarding operation. A related consequence is that higher-level protocols find that their headers don't always start in the same position within the packet.)

To create a gateway that can sustain a through transmission rate comparable to that of the subnetworks involved requires careful budgeting of the machine cycles involved. For example, a bandwidth of 10 Mbits/sec. requires being able to pass twelve hundred fifty 1000-octet packets/second, leaving a time budget of only 800 microseconds per packet. If a 0.5 MIPS processor is used for the gateway, there must therefore be fewer than 400 instructions executed for each packet, with the implication that whatever routing scheme is used, it must be extremely simple. The source routing approach makes meeting such a budget a realistic possibility.

Maintenance is directly aided by having such a simple gateway mechanism. With little to do, there is little to go wrong, failures should be relatively rare and diagnosis and repair should be straightforward. Even in the case where a gateway is actually implemented by software in a node attached to two local transport networks, the simplicity of action required of a gateway means that the program required is short, the cycles required are few, and that therefore the program is not only likely to be trouble-free but also it is acceptable to embed it in the innermost part of the supervisor, where it is less likely to fail because of interference by other programs in the same node. Perhaps even more important in the case of a software gateway, the simplicity of the source-routing approach means that the software required can be quick to implement.

3. Route Control. One of the more interesting opportunities that arises when source routing is used is that the node that is the source of a message can, if appropriate, control precisely the route through the internet that outgoing packets follow. This control can be applied to solve several problems, as follows:

a) Trouble location. If trouble develops in a network gateway, it will be noticed first as failure of packets routed through that gateway to arrive at their destination. Starting at any node that notices such a problem, one can route a test packet "out and back", through some set of gateways and back to the originating node. A series of such tests, tracing successive steps in the route that failed, should quickly locate the troublesome gateway. One can also imagine extending this idea to route a message into a target node and back out again, as a check on the operation of the lower levels of that node's operating system. An

interesting aspect of this approach to trouble location is that any user, if sufficiently desperate, can undertake network diagnosis; trouble location is not restricted to a network maintenance center that has some particular address or special hardware.

b) Policy implementation: Some local networks may be paid for by a supporting organization that wants to have a say in their usage policy. (For example, use of the ARPANET is supposed to be restricted to government-sponsored business.) If such a network has gateways to two other networks, it could be used as an intermediate transport link on some packets flowing between those networks. If source routing is used, the node that originates a packet can control whether the packet is routed through the network in question or, alternatively, avoids that network. (Obviously, sophisticated help from routing services is needed to actually implement such a policy, but the opportunity is there.)

c) Class-of-Service implemention. There are a variety of properties that an internet connection can have, and that may be different on different routes: error rate, transport delay, probability of wiretapping, bandwidth. Again, assuming considerable knowledge on the part of a routing service, with source routing one can choose a route that has class-of-service properties that are tailored to the application.

d) FIFO streams. Assuming that all gateways along a given route relay packets in the same order that they are received, if the same source route is used on several packets, those packets will arrive at their target in the same order that they left the source, eliminating any need for the target to restore order in what is intended to be a FIFO stream.

In a hop-by-hop dynamic routing system, FIFO delivery cannot be easily insured, so the source and target must work harder if that is a function they require.

Finally, in an inter-network environment that includes both public and private gateways, the precise route control provided by source routing seems to be a key to effective use; private gateways can be used by their owners while being ignored by everyone else; flaky gateways can be bypassed by wary users no matter what administration is responsible for them.

4. Other observations. There are a variety of other observations that one can make about source routes. These are, in no particular order:

1) Source routing avoids several problems that can accompany more dynamic, highly optimal routing schemes. There is no danger of packets circulating in a loop forever, so techniques such as hop counts are not needed. There is little concern for startup transients, stability, or oscillation in the dynamics of route selection. Extra traffic to exchange traffic statistics among gateways is not involved, and one does not have to worry about the interaction between the reliability of that traffic and the stability of the network. There is no requirement that each gateway maintain a table that has a number of entries proportional to the size of the network.

2) Source routing is "compatible" with hop-by-hop routing in the following curious way: one can start with an existing collection of linked subnetworks that uses hop-by-hop routing, and add a routing service and an independent set of gateways that use source routing. If a packet is first sent to a hop-by-hop gateway, it will continue on its way using

hop-by-hop routing (and never encountering a source-route gateway). If, on the other hand, a packet is filled with a source route and sent to a source route gateway, it will then follow the prescribed route. This observation encourages the experimental use of source routes, or their use ofr a narrow purpose, say, as a way to provide class-of-service control. (We are indebted to Danny Cohen for this observation.)

3) Source routing is consistent with at least two proposed fragmentation/reassembly strategies. Fragmentation can be done by a gateway on entry to a subnetwork that has a small maximum packet size: by using the same route for all fragments of a given packet reassembly can be accomplished either at the gateway leaving that subnetwork or by the target node. Fragmentation can also be done by a fragmentation service, which might be a node whose address appears "in the middle" of a route unbeknownst to the source, target, or intervening gateways. If it receives a packet that it believes is too large to get through some intermediate subnetwork, it can fragment that packet and also reroute the fragments through a reassembly service on the other side of the bottleneck. Finally, one might successfully finesse fragmentation completely by sending big packets over a longer or less desirable route that allows big packets, while sending small ones the short, desirable way.

4) In a manner similar to the fragmentation/reassembly servers just described, one can place other specialized services along a route to act as filters, translators, etc. This idea has not been explored, but it seems to represent an interesting opportunity.

5) Attachment of a single host to several subnetworks (the "multi-homing problem") is simplified. In a complex internet installation, one might expect to find some hosts that have attachments to two or more different subnetworks of the internet, perhaps for added reliability or for assured bandwidth to services found on different subnetworks. If the several attachment points are functionally equivalent, then when another node tries to send a message to such a host, there is a question of to which one of the several attachment points the message should go. A hop-by-hop routing scheme in which gateways interpret internet identifiers would require that either the different attachment points be assigned different internet identifiers (so the originator has the burden of choosing which internet identifier to use) or else a single internet identifier be used for all the attachment points of the target host and the gateways add this topological fact to their storehouse of routing knowledge and make the choice on the fly. With source routing, the burden of choice can move to the routing service, where the topological information is available to choose a route from the originator to the nearest attachment point of the target host. Neither the originator nor the internet gateways need realize that the target has several attachment points.

In this last case, as in some others, one can argue that some of the apparent simplifications or advantages obtained by using source routing are actually only shifts of the underlying problem over to the routing service. This argument is correct, but it underemphasizes two points:

1) Separation of two tangled problem areas, resolving the identity of a target node and routing, into two distinct and largely independent mechanisms simplifies and clarifies design, algorithms, and code. Modularity of network implementation is improved.

2) When one implements routing as a service supplied by a server, it becomes possible to introduce variations on the service by changing just the server, or providing an alternate server. When the function of routing is distributed among the gateways, changes in the service require changing all of the gateways, an undertaking that is more difficult and hazardous. Again, modularity is the key consequence of using source routes.

Conclusions

The premise of our argument is that source routing is particularly well-suited to the campus environment. The argument goes as follows: in the campus environment, one can install high bandwidth lines at low cost, since reliance on common-carrier offerings is not required and physical facilities are under administrative control. This high bandwidth permits using strategies, such as source routing, that may waste some part of the communications capacity by not being optimal. At the same time, source routing may make possible the high-bandwidth gateways required to fully exploit the available transmission bandwidth. The campus administrative environment calls for diversity in protocol, for which source routing caters by providing a lowest campus-wide transport protocol with a minimum amount of predetermined function that might constrain higher level protocol choices. The campus administrative environment also calls for diversity in administration, for which source routing caters by permitting precise control of complete routes for particular messages, and multiple strategies for resolving service identifiers or network addresses, as required. It also permits messages to flow through an internetwork arrangement despite some of its topology not being centrally planned. Source routing allows particularly

easy trouble location and source routing gateways are exceptionally simple, two properties that are important when one assumes a central administration that must be cost-conscious or even under-funded. Finally, the modularity of network implementation that source routing and routing services provide seems especially important in an environment that must cope with evolving technologies and protocols. Thus, from these arguments one can conclude that, at least for the campus-wide internetwork case, source routing is an attractive scheme well worth considering.

We have concentrated on the application of source routing to the campus environment, without attempting to identify parallel situations elsewhere for which source routing might be similarly important. For example, the British Post Office, in its recommended standard end-to-end transport protocol [14], suggests that source routing be used in passing packets through the concatenation of a local network, a public net, and another local net, because of the small likelihood that all of these separately administered networks will have a common station numbering plan.

Finally, the remodularization of network function implied by source routing involves a substantially clever routing service. Although we believe such a routing service to be a straightforward design, that design has not been sketched here; it remains an area of continuing investigation.

Acknowledgements

This paper records a series of intensive discussions with, among others, Kenneth Pogran and Noel Chiappa. It also borrows ideas and terminology from working papers of the ARPA internet project by Danny Cohen, Jon Postel, and John Shoch and from working papers of the M.I.T. Artificial Intelligence

Laboratory Chaosnet project by David Moon. Welcome comments on early drafts were made by Danny Cohen and John Shoch. The basic idea of source routing and the mechanics of source route operation and reverse route construction were suggested by Farber and Vittal in their 1973 paper; the present paper contributes only observations and implications for the special adminstrative environment by the local area or campus network. Some implications of the simplicity of a source routing gateway and the notion of a routing service were suggested by Hopper and Wheeler [15].

References

[1] Farber, D.J., and Vittal, J.J., "Extendability Considerations in the Design of the Distributed Computer System (DCS)," Proc. Nat. Telecomm. Conf., (November, 1973), Atlanta, Georgia, pp. 15E-1 to 15E-6.

[2] Sunshine, Carl A., "Source Routing in Computer Networks," Computer Communication Review 1, 7, (January, 1977) pp. 29-33.

[3] Metcalfe, R.M., and Boggs, D.R., "Ethernet: Distributed Packet Switching for Local Computer Networks," Comm. ACM 19, 7 (July, 1976), pp. 395-404.

[4] Okuda, N., Kunikyo, T., and Kaji, T., "Ring Century Bus-an Experimental High Speed Channel for Computer Communications," Proc. Fourth Int. Conf. on Computer Communications, September, 1978, pp. 161-166.

[5] Clark, D.D., Pogran, K.T., and Reed, D.P., "An Introduction to Local Area Networks," Proc. IEEE 66, 11 (November, 1978), pp. 1497-1517.

[6] Thornton, J.E., Christensen, G.S., and Jones, P.D., "A New Approach to Network Storage Management," Computer Design, November, 1975, pp. 81-85.

[7] Hopkins, G.T., "Multimode Communications on the MITRENET," Local Area Communication Network Symposium, Boston, Mass., May, 1979, pp. 169-177.

[8] Wilkes, M.V., and Wheeler, D.J., "The Cambridge Digital Communication Ring," Local Area Communication Network Symposium, Boston, Mass., May, 1979, pp. 47-61.

[9] International Organization for Standardization, Open Systems Interconnection, "Reference Model of Open Systems Architecture," Association Francaise de Normalisation Tour Europe, Paris, France, November, 1978.

[10] Feinler, E., and Postel, J., Editors. "ARPANET Protocol Handbook," Stanford Research Institute, NIC 7104, January, 1978.

[11] The International Telegraph and Telephone Consultative Committee (CCITT), "Provisional Recommendations X.3, X.25, X.28, X.29 on Packet-Switched Data Transmission Services," International Telecommunication Union, Geneva, 1978.

[12] Postel, J., Editor. "DOD Standard Transmission Control Protocols," Information Sciences Institute, IEN 129, January, 1980.

[13] Boggs, D.R., et al., "PUP: An Internetwork Architecture," IEEE Trans. on Comm. COM-28, 4 (April, 1980), pp. 612-623.

[14] Linington, P.F., editor, "A Network Independent Transport Service," British Post Office PSS User Forum Study Group Three, SG3/CP(20)2, 1980-2-16, February, 1980.

[15] Hopper, A., and Wheeler, D.J., "Binary Routing Networks," IEEE Trans. on Computers C-28, 10 (October, 1979), pp. 699-703.