Network Layer Mobility: an Architecture and Survey

Pravin Bhagwat[†] Charles Perkins[†] Satish Tripathi[‡]

[†]IBM, T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

[‡]Computer Science Department
University of Maryland
College Park, MD 20742

1 Introduction

Mobile end-systems frequently change their point of attachment to the network. In such an environment, in order for mobile devices to run without disruption, an inter-networking infrastructure is needed. In addition, a common networking protocol is required which can support network-wide mobility. Mobile devices also need to communicate with the existing pool of information servers and file servers, which means that internetworking solutions for connecting stationary and mobile systems are also required. Unfortunately, the Internet Protocol (IP), which forms the fabric of the current world-wide data communication network, falls short of meeting this demand. The current Internet suite of protocols (TCP/IP) were designed under the assumption that end-systems are stationary. If during an active network session one end of the connection moves, the network session breaks. Naturally, all networking services layered on top of TCP/IP are also disrupted when end-systems become mobile. There are two approaches for solving this problem. One is to completely redesign internetworking protocols with the specific goal of supporting mobile end systems. The other approach is to provide additional services at the network layer in a backward compatible manner which make mobile internetworking possible. The first approach, though an interesting possibility from a research viewpoint, is infeasible since it would require radical changes to the currently deployed networking infrastructure. It is the latter approach that is the focus of our investigation.

To ensure inter-operability with the existing infrastructure, the handling of mobility should be completely transparent to the protocols and applications running on stationary hosts. In other words, from a stationary end-system's perspective, a mobile host should appear like any other stationary host connected to the Internet. This means the same naming and addressing conventions, those originally developed for stationary hosts, must apply to mobile hosts. In addition, any changes in a mobile's network attachment point should be completely hidden from the protocols and applications running on stationary hosts.

In this paper we explore various network layer concepts that play a crucial role in the design of mobile networking systems. We show that mobility is essentially an *address translation* problem and is best resolved at the network layer. We have identified the fundamental services that must be supported at the network layer to carry out the task of address translation. Using these service primitives as building blocks, we describe a network layer architecture which enables smooth integration of mobile end systems within the existing Internet. In the second half of this paper, we present a summary of some of the key Mobile IP proposals and show that each proposal can be viewed as a special case of the proposed architecture. It is worth pointing out that our objective is not to propose a new protocol for supporting mobility, rather it is to highlight various design choices and the tradeoffs involved in the design of large mobile systems.

2 Internet Naming and Addressing

The Internet is a large collection of networks which share the same address space and inter-operate using a common sets of protocols, such as TCP/IP [20, 21]. A fundamental concept of the Internet architecture is that each host[1] has a unique network address, by which it is reachable from other hosts in the network. Data are carried in the form of packets which contain source and destination addresses. To communicate with another host, a source only need to know the address of the destination. Internet routers cooperate to carry packets from a source to a destination node.

Internet routers maintain a view of network topology in the form of routing tables. These tables are consulted when making packet routing decisions. The process of routing involves inspecting the the destination address contained in the packet and, based on the contents of the routing table, determining the next-hop router to which packet should be relayed. Each router along the path from a source to a destination node repeats this process until the packet is finally delivered to the destination host.

If host addresses are treated as *flat identifiers*, routers will be required to maintain routing information on a per-host basis. Obviously, this is not feasible, given the large number of hosts (over 10 million!) that are connected to the Internet. A natural solution is to impose a hierarchy on the address structure. Hierarchical addressing is essential if the routing architecture is to be scalable. The Internet, for example, deploys a multi-level hierarchical addressing scheme [9].

2.1 Internet Addressing

Each host in the Internet is assigned a unique 32-bit internet address (also known as an IP address) which consists of two parts: network-id and host-id. IP addresses are commonly represented using dotted notation where each octet is represented as a decimal number and dots are used as octet separators.

Under the current Internet addressing scheme, routers only need to maintain network topology information at the granularity of individual networks. This means only the network part of the destination address is used in making routing decision. Though hierarchical addressing makes routing simple and manageable, as a natural consequence, it puts certain restriction on the address usage. A hierarchical address can only be used within the domain of its definition. For example, an Internet address is only meaningful so long as the host using it remains connected to that network denoted by the network-id part of the address. When the host moves to a new network, it must be allocated a new address which is derived from the address space of the new network. In order for the Internet routing to work:

A mobile host must be associated with a new address when it moves.

[1] In the Internet jargon, host means an end-system connected to the Internet

2.2 Naming

Hosts are also identified in the network by their *Host Names*. Names are user defined aliases (strings of characters) which are used to denote hosts. An important distinction between names and addresses is that addresses may be protocol specific (e.g., an IP address, CLNP address, IPX address, XNS address), but names are not. Names provide a way for applications to make reference to network entities without having to know anything about the underlying network protocol in use. This is useful, since users find names easier to use and remember than cumbersome network addresses.

Though applications refer to end systems by names, when packets are transported through the Internet, each must contain an IP address of a destination node. This is because Internet routers do not understand names; they can only interpret addresses. A translation mechanism, therefore, is required for mapping host names to addresses. To accommodate a large, rapidly expanding set of names, a decentralized naming mechanism called the Domain Name System (DNS) was deployed in the Internet. DNS stores name to address mappings in a distributed data structure. Finding the address of the host is essentially a directory lookup operation (see Figure 1). When two hosts on the Internet need to communicate with each other, the source node performs a DNS lookup to obtain the destination node's address and then initiates a connection setup procedure. During connection setup, each end of the connection learns about the address of the other end. So long as the connection is active, no additional DNS lookups are performed, since name to address binding is assumed to be static and is not expected to change during a connection lifetime.

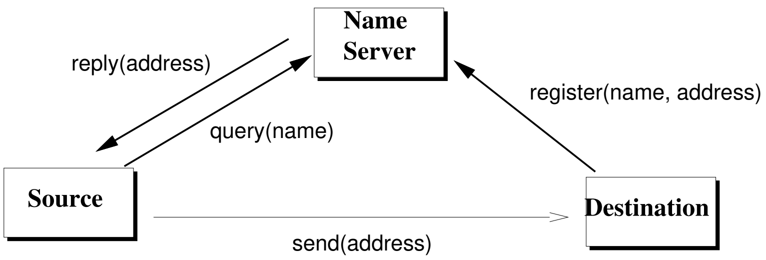

Figure 1: DNS based Name to Address resolution

3 The Mobility Problem

To illustrate why host mobility poses problem at the network later, it is important to emphasize the distinction between the concepts of *name* and *address*. A name is a location independent identifier of a host. An address on the other hand reflects a host's point of attachment to the network. For hosts that remain static throughout their lifetime, both names and addresses can be used interchangeably. For a mobile host, however, an address cannot be used as a unique identifier, since it must change with the location of the host. The name is the only location independent identifier that can be used to refer to mobile hosts.

3.1 Mobility Problem: Directory Service View

In networks where hosts are static, name to address bindings never change. Host mobility makes this binding a function of time. Therefore, network layer mechanisms are required for resolving names into addresses and tracking the location of hosts as they move. The Domain Name System (DNS), which provides name to address translation service in the Internet today, could be enhanced to meet the additional demands. However, this task is made difficult by many hurdles:

- Historically, the DNS had no provision to handle dynamic updates. This is because it was originally designed to provide name lookup service for stationary hosts only.
- The DNS design attempts to optimize the *access* cost, and not the *update* cost. Server replication and client caching provides significant performance gains for access only systems, but results in very poor performance when updates are performed. In a mobile environment, both updates and accesses are likely.
- DNS clients cache DNS records to reduce latency for future accesses and to reduce load on the name servers. There is no call back mechanism generally available from servers to clients in case cache entries become invalid.

A design for a distributed location directory service for mobile hosts was proposed by Awerbuch and Peleg in [2]. They formally proved an important theoretical result which established that a system cannot optimize both *access* and *update* operations[2]. Using the concept of *Regional Directories* (a type of cache) they proposed a distributed directory layout which guarantees that the communication overhead of *access* and *update* operations is within a poly-logarithmic factor of the lower bound.

As far as the Internet is concerned, distributed directory service based solutions do not appear very attractive since they cannot be deployed without changing existing host software. The current size of the Internet makes any such change to host software almost impossible to achieve. Hence, an alternate solution method is required.

3.2 Mobility Problem: Internet View

When host names were originally deployed, it was implicitly assumed that the *name* to *address* binding remained static. Instead of referring to hosts through names, protocols were developed that referred to hosts through their addresses. A standard example is a TCP connection which is identified by a 4-tuple:

$$< \textit{source IP address, source TCP port,} \quad \textit{destination IP address, destination TCP port} >$$

If neither host moves, all components of the connection identifier will remain fixed, and thus a continuous TCP session can be maintained between the two hosts. If either end of the connection moves, we run into the following problem:

- If the mobile host acquires a new IP address, then its associated TCP connection identifier also changes. This causes all TCP connections involving the mobile host to beak.

[2]In their paper they use terms `Find` and `Move` to denote these operations.

- If the mobile host retains its address, then the routing system cannot forward packets to its new locations.

The fundamental problem is that in the Internet architecture, an IP address serves dual purposes. From the transport and application layer perspective, it serves as an *end-point identifier*, and at the network layer, the same IP address is used as a *routing directive*. This problem is not specific to the Internet architecture; in fact all other contemporary connection-less network architectures, such as OSI, IPX, and XNS, suffer from this problem. Since our objective is to ensure that connections do not break when hosts move, we can say that:

In order to retain transport layer sessions, a mobile host's address must be preserved regardless of its point of attachment to the network.

An immediate consequence of this choice is that we can not rely on the existing addressing paradigm for delivering packets to a mobile host's new location. A solution might be to keep per-mobile-host routing information at all routers, but this completely breaks the hierarchical model of routing, causing unbounded growth in the size of routing tables. Thus, the problem of supporting mobile hosts within the Internet is not just keeping track of hosts. A mechanism has to be designed for forwarding packets to mobile hosts without modifying and compromising the scalable nature of the Internet routing mechanism.

4 Network Layer Solution Architecture

In this section we describe a network layer architecture that allows smooth integration of mobile end-systems within the Internet. Our objective is to highlight and analyze the essential aspects of providing mobility extensions in any connectionless network where routers depend on addresses stored in the packet. The specific details involved in designing a mobile-networking system will be discussed later. For ease of exposition, we will first introduce a few definitions.

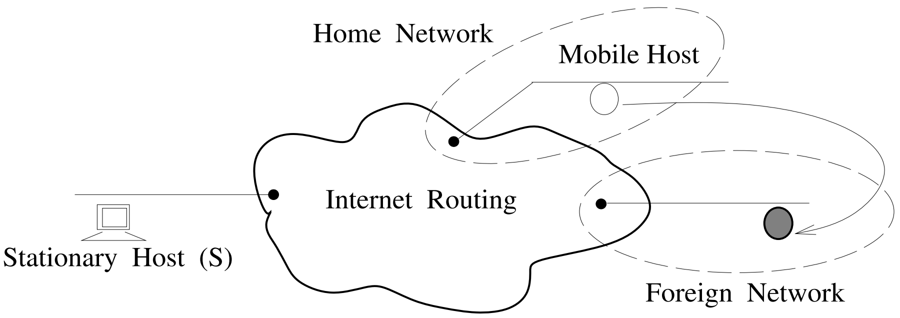

Figure 2: Illustration of Terms

Mobile Host: An internet host is called a *Mobile Host(MH)* if it frequently changes its point of attachment to the network. A change in the attachment point can happen while one or more transport layer sessions involving the MH are in progress. It is assumed that the rate of change of location is slower than the time it takes to for the mobile routing protocols to take into account the mobile host's new location.

Home Address: Like any other internet host, a mobile host is also assigned an IP address which is referred to as its *Home Address (HA)*. A standard 32-bit address is allocated using the same guidelines that apply to stationary hosts. When the DNS is queried with a mobile host's name, it returns the home address of the mobile host.

Home Network: Within each administrative domain, network administrators usually reserve one or more subnetwork(s) for mobile hosts. The home address of a mobile host is allocated from the address space of one of these subnetworks, referred to as the *Home Network* in the subsequent discussion. The terms *home address* and *home network* could also apply to stationary hosts. The only difference is that stationary hosts always remain connected to their home network, while mobile hosts sometimes may not be found at their respective home networks.

Foreign Network: Any connected segment of an Internet, other than the home network of a mobile host, to which the mobile host is allowed to attach is referred to as a *Foreign Network*.

Notice that above definitions are relative to a mobile host. The same network could operate both as a home and as a foreign network, depending on which mobile host is connected to it. So long as a mobile host remains connected to its home network, existing internet routing mechanism are sufficient to route packets up to its current location. It is only when it moves to a different network that additional mechanisms are required. If a mobile host moves within its home network (e.g., detaches from one ethernet point and attaches through another ethernet point), it does not constitute a move from the network layer point of view. A collection of link layer networks, which are interconnected through bridges, is called a layer 2 segment. Existing link layer bridging mechanism are capable of routing packets up to end-systems so long as they remain connected to the same layer 2 segment. Within a layer 2 segment, a packet can be delivered solely on the basis of the destination node's link layer address; the network layer routing is not required.

In the previous section, we made two crucial observations:

1. The home address of a mobile host cannot be used for routing packets to its current location (except when it is attached to its home network).
2. A mobile host's address must be preserved in order to retain all active transport connections involving the mobile host.

These are two conflicting requirements. From the first observation, when a host moves, a new address, reflecting its new point of attachment to the network, must be used for the purpose of routing. The second observation says just the opposite: the original address must be preserved to retain all active network sessions.

4.1 Two Tier Addressing

We introduce the concept of *two-tier addressing* to resolve the problem associated with the dual use of an internet address. Our solution involves associating two internet addresses with each mobile host(see Figure 3). The first component of the address reflects the mobile's point of attachment to the network while the second component denotes its home address. The first address component serves as a *routing directive*. It changes whenever a mobile host moves to a new location. The second component of the address serves as an *end-point identifier*. It remains static throughout the lifetime of a mobile host. The purpose of two-tier addressing is to decouple the dual role of an internet address into two disjoint, well defined functions.

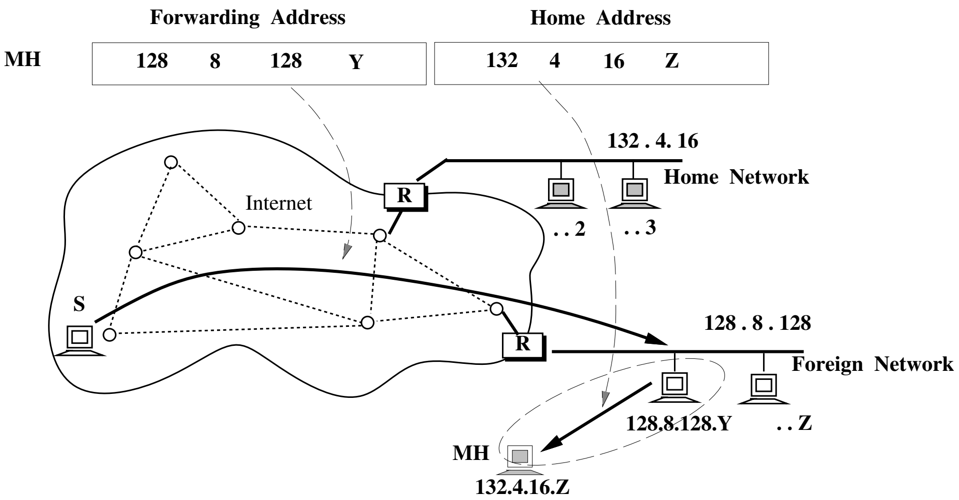

Figure 3: Two Tier addressing for Mobile Hosts

The concept of two-tier addressing is illustrated in Figure 3. Packets that are destined to mobile hosts contain the destination address in the two-tier format. The Internet routing system only looks at the first component of the address and routes those packets to the point where the mobile host is attached. At this point, the first address component is discarded. Only the second address component, the home address of the mobile host, is used in subsequent protocol processing. From an end-host's perspective this means that it notices no difference when it is attached to its home versus when it is located in a foreign network. In other words, the mobile host remains *virtually* connected to its home. Packets which originate from the mobile host and are destined to the stationary host (S) do not require any special handling, since the Internet routing system can deliver those packets based on their destination addresses. If S is also mobile, then the same two-tier addressing mechanism can be used to route packets to its current location.

It is important to note that the two-tier addressing is only a logical concept. Its realization doesn't necessarily require carrying two addresses in the destination address field of the network layer packets. In fact, doing so would require changes in the existing packet formats, necessitating changes to host and router software. It is desirable to support the two-tier addressing method using the existing mechanisms available in the Internet. The following sections describe how this goal can be achieved.

4.2 Architecture Components

4.2.1 Forwarding Agent (FA)

When away from its home network, a mobile host can attach to the Internet through a foreign network. For the purpose of forwarding datagrams to its new location, an address derived from the address space of the foreign network must be used. Packets destined to the mobile host contain

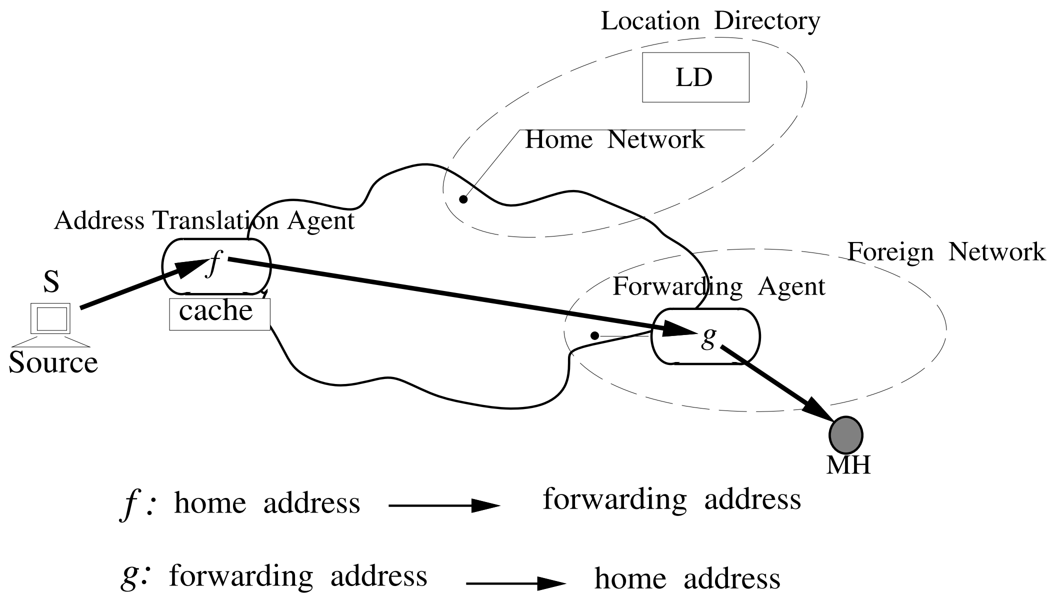

Figure 4: Packet Forwarding Model

the address of a *Forwarding Agent (FA)* in the forwarding address sub-field of the two-tier address. An FA provides an access point through which mobile hosts can attach to the network. It receives packets on behalf of mobile hosts, and forwards them to appropriate mobile hosts after necessary protocol processing.

Conceptually, the processing at the FA involves stripping the forwarding address part of the two-tier address and exposing the home address of the mobile host. Once the packet arrives at the FA, the forwarding address is no longer required in the subsequent protocol processing. When a packet arrives at the FA, it contains the address of the FA in its destination address field. The FA, essentially, maps the contents of the destination address field (the forwarding address) to the home address of the associated mobile host. We use the notation g to denote this mapping function:

$$g(forwarding\ \ address) \rightarrow (home\ \ address)$$

An FA should be able to relay packets to the mobile host on the basis of its home address. This is easy if the FA and the MH are directly connected (normally over a wireless link). Otherwise, the routing protocol operating in the foreign network should advertise host specific routing information within the foreign network to facilitate routing of these packets to mobile hosts. Normally, we would expect a wireless base station to operate as an FA in which case the MH and the FA would be directly connected to each other over a wireless link.

A mechanism is required so that mobile hosts can discover the address of an FA when they connect to a foreign network. Similarly, a mechanism is required so that the FA can determine the identities of all mobile hosts that require its service. The simplest way to achieve this is through a route advertisement and a registration protocol. Forwarding agents periodically advertise their presence in the foreign network. Beaconing, the periodic broadcast of messages over the wireless medium, is the most commonly used method. Mobile hosts can listen to broadcasts, determine the identity (address) of the nearest FA, and initiate a registration sequence.

4.2.2 Location Directory (LD)

The component in the architecture that records the association between the home and the forwarding address of a mobile host is called a Location Directory (LD). The LD contains the most up-to-date mapping between a mobile host and its associated FA. Mobile Hosts are required to send updates to the LD whenever they move to a new location.

Since the number of mobile hosts is expected to be very large, a centralized realization of the LD is deemed infeasible. A policy for distributing LD components should take many factors into consideration, such as the cost of access, ease of locating LD components, and security and ownership of location information. Since the LD will be accessed very frequently, a good distribution method should exploit the locality of access patterns and provide uniform load balancing among all LD components. Given a model for the LD access pattern, the LD distribution can be formulated as an optimization problem[1]. Unfortunately, these mathematical results [1, 4, 3] cannot be directly applied in the Internet. The primary reason is that in the Internet factors such as ease of location, security, and ownership take precedence over any cost optimization considerations.

A feasible distribution scheme in the Internet is the *owner-maintains* rule. According to this scheme, the LD entries for mobile hosts are maintained at their respective home networks. Within each home network, a good place for locating an LD component is at the home network. Advantages of this scheme are:

1. Some agent on each home network is responsible for maintaining, securing, authenticating, and distributing LD information for its mobile hosts. This policy fits well within the Internet philosophy of autonomous operation.

2. No special mechanisms are required to locate the LD components. It is important to point out that in a distributed scheme, in order for a source to send a query to the right LD component, the source is required to know the address of the LD component in advance. Under the *owner-maintains* rule, a source simply sends a query that is addressed to the mobile host. The packet is delivered to the home network by normal internet routing where it is intercepted by the home router and subsequently relayed to the correct LD component.

This is certainly not the only possible distribution scheme. Later in this paper we'll discuss other options while reviewing various Mobile IP proposals.

4.2.3 Address Translation Agent (ATA)

Hosts that need to communicate with a mobile host insert the mobile's home address in the destination address field of all packets they issue. At some point during the routing process this address should be replaced by the address of the FA associated with the mobile host. The entity which performs this operation is called an *Address Translation Agent.* The process of address translation involves querying the LD, obtaining the FA address, and subsequently making use of this address in forwarding packets to the correct location of the mobile host. The address translation function is:

$$f(home\ \ address) \rightarrow (forwarding\ \ address)$$

From a two-tier addressing perspective, an ATA initializes the forwarding address part of the destination address. In an actual implementation this could be achieved by prefixing the original destination address of the packet with the FA's address. This operation can be performed at the source host; however, the only problem is that the function f cannot be computed without making changes to the existing host software of millions of hosts.

For performance reasons, an ATA may decide to cache LD entries which are frequently used in making forwarding decisions. Querying the LD before making each address translation operation could be prohibitively expensive, particularly so when the ATA and the LD are geographically separated. Caching, however, introduces a new requirement in the architecture; that of maintaining consistency between the LD and its cached entries throughout the Internet.

4.3 Location Update Protocol

Keeping the LD up-to-date in the face of frequently changing host location is crucial. Keeping cached LD entries consistent with the master LD is an equally important consideration. Inconsistencies could make mobile hosts inaccessible and even cause the formation of routing loops in some cases. The purpose of *Location Update Protocol(LUP)* is to provide reliable mechanisms for keeping the LD and its cached copies consistent at all times.

To a large extent, the choice of the LUP depends on the caching policy used. Together, they determine the scalability and routing characteristics of a mobility solution. In systems which do not permit LD caching, ATAs must be co-located with the LD, since issuing an LD query for each packet that an ATA forwards is prohibitively expensive. In such systems, packets addressed to mobile hosts first travel all the way to the home network before any address translation (function f) is performed. Clearly, the paths that packets follow are non-optimal in this case. Caching improves the routing efficiency of a mobile networking system, as packets do not have to travel to home networks before being forwarded toward the FAs associated with the destinations. At the same time, caching makes the system more complex and vulnerable to security attacks. If cache updates are not properly authenticated, it is possible to redirect packets away from a mobile host and cause denial-of-service.

4.4 Packet Forwarding Operation

With the inclusion of address translation agents and forwarding agents, the operation of packet forwarding can be easily illustrated. Figure 4 illustrates how packets from a stationary host (S) are routed to a mobile host (MH). S sends out packets which are addressed to the home address of the MH. These are intercepted by an address translation agent which maps (using function f) the original destination of the packet to the address of the forwarding agent. Once these packets arrive at the forwarding agent, the FA remaps (using function g) the destination to the home address of the mobile host and delivers them to the mobile host. Along the path from the source to the destination, packets twice undergo an *address translation* operation. The end result of this translation process, the function $g o f$, is an identity mapping, which means that the whole process of address translation is completely transparent to hosts located at both ends of the path. They communicate as if they were stationary. The transport layer protocols and the applications running on stationary as well as mobile hosts operate without any modifications whatsoever. This property of the solution architecture is termed as *transport layer transparency.*

The proposed architecture preserves *transport layer transparency* regardless of where and how in the network the LD, ATAs, and FAs are distributed. This flexibility enables us to capture the design choices made in other Mobile IP proposals. In section 5, we'll show that each one of these proposals can be viewed as a special case of the proposed architecture.

4.5 Address Translation Mechanisms

So far we described how various components of the architecture co-operate with each other to perform the necessary address translation operations, but the actual mechanisms were not mentioned. Within the Internet there are two possible ways of doing it: either using *encapsulation* or using *loose source routing*. A brief description of both follows:

4.5.1 Encapsulation

In the encapsulation method a new header is appended at the beginning of the original datagram (see Figure 5). The outer header contains the address of the forwarding agent while the inner header contains the home address of the mobile host. Since the Internet routing system only looks at the outer datagram header, it routes this packet to the forwarding agent. The forwarding agent strips the outer datagram header and delivers the original datagram locally to the mobile host.

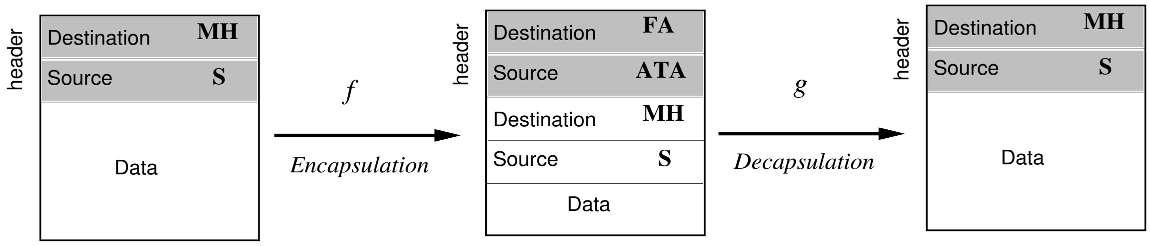

Figure 5: Illustration of Encapsulation and Decapsulation

4.5.2 Loose Source Routing (LSR)

Loose Source Routing is an option that is supported in IP which can also be used to perform address translation operation[3]. Using IP's source routing option, an address translation agent can cause datagrams addressed to a mobile host's home address to be routed via a forwarding agent. Figure 6 illustrates how this is done. An LSR option is used to specify a list of addresses. The Internet routing system routes the datagram containing the LSR option to each address, one by one, in the sequence it appears in the list. The current destination is kept in the destination address field of the datagram header and a pointer points to the address which is to be visited next in the sequence. When the datagram arrives at the current destination, the contents of the destination address field are swapped with the address pointed by the next hop pointer, and, the pointer is advanced to the next address in the list. This process is repeated until the datagram is delivered to the address which occurred last in the original list of addresses included in the LSR option. At this point the next hop pointer in the LST option points past the last address.

As natural consequence of the LSR option processing, the path that a packet follows (the list of addresses visited en-route) is automatically recorded in the packet. The destination can reverse this list and send a reply back to the source along the reverse path. In [17, 19], authors show how this feature is used to design a mobile networking scheme that co-locates the ATA with the source, and the FA with the destination.

[3]Originally it was included in IP not for this purpose, but to help in debugging network problems

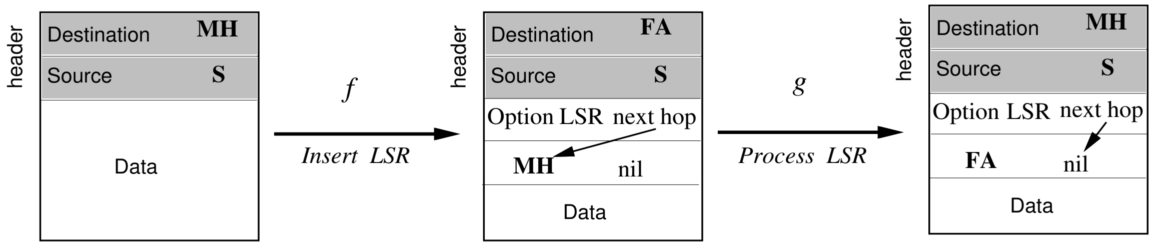

Figure 6: Using Loose Source Routing to perform address translation

In this section we showed how components of the proposed architecture mutually co-operate to overlay a packet forwarding service on top of an existing routing infrastructure. It is important to point out that the ATA and the FA only represent functions that need to be supported, not machines that need to be deployed in the network. In fact, the proposed architecture allows a great deal of flexibility in placement of these functions in the network. This flexibility allows us to experiment with various design alternatives and devise a solution for a specific target environment.

5 Mapping to candidate Mobile IP proposals

Over the past several years, many proposals have been made for supporting host mobility on datagram-based internetworks. A vast majority of these proposals have been designed to be compatible with today's TCP/IP-based Internet. The candidate proposals differ widely in terms of the specific components they propose to add to the Internet, the mechanisms they use for address translation, and the policy they use for managing location updates. In this section, we'll show that all mobile IP proposals can be viewed as a special case of our proposed network architecture.

In our model, the ATA and FA represent the two basic functions that must be supported by any proposal that supports mobility. We'll demonstrate this fact by explaining the operation of each Mobile IP proposal in terms these two functional entities. Basically, all proposals attempt to provide an address translation service through deployment of some additional entities in the network. They only differ in terms of their choice of where they locate these functions, the specific location update protocol they use, and whether they use encapsulation or source routing to effect address translation. Below we present a short summary of related Mobile IP proposals, with a short note following each proposal outlining how its operation can be captured by our proposed solution architecture.

5.1 Columbia Scheme

The scheme proposed by Ioannidis at. el. [10, 11] is designed primarily to support mobility within a campus environment. Mobile hosts are allocated addresses from a subnetwork which is reserved for use by wireless hosts. A group of cooperating Mobile Support Routers (MSR), advertise reachability to the wireless subnet. MSRs provide an access point through which mobile hosts can connect to the campus back-bone, and are also responsible for forwarding traffic to and from mobile hosts. Each mobile host, regardless of its location within a campus, is always reachable via one of the MSRs. When a host sends a packet to a mobile host, it first gets delivered to the MSR closest to the source host. This MSR either delivers the packet (if the destination MH lies in its wireless cell),

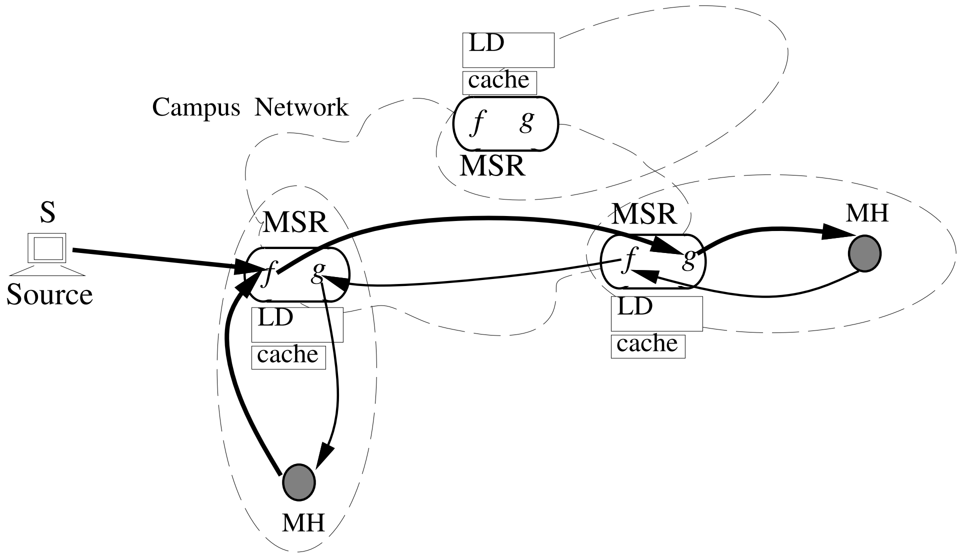

Figure 7: Mapping to Columbia Proposal

or forwards it to the MSR responsible for the destination MH. If an MSR does not know which MSR is currently responsible for a destination, it sends a `WHO_HAS` query to all MSRs in the campus and awaits a reply message from the responsible MSR. When sending a packet to the destination, an MSR encapsulates the packet and delivers it to the target MSR. Upon receiving this packet, the target MSR strips the encapsulation header and relays the original packet to the mobile host.

In the Colombia proposal, an MSR performs both *encapsulation* and *decapsulation* operations, meaning that both functions, f and g, are co-located at the MSR. For packets addressed to MHs in its coverage area, an MSR acts like an FA. For packets addressed to other MHs it acts like an ATA. Each MSR maintains a table of MHs in its wireless cell. These tables together constitute the segment of the LD which is associated with mobile hosts on the campus network. This LD distribution scheme can also be thought of as a distributed realization of the *owner-maintains* rule. Recall that in the *owner-maintains* rule, the segment of the LD was co-located with the home router. An MSR in the Colombia scheme is a distributed realization of the home router. As a result, the table of mobile hosts maintained at an MSR constitutes a distributed segment of the LD that is required to be maintained at the home router.

MSRs acquire LD cache entries on a need-to-know-basis by sending a multicast `WHO_HAS` query to all MSRs in the campus. The response to this query is generated by the MSR which possesses the primary copy (in other words, the MSR which is responsible for the destination MH). The Location Update Protocol uses a *lazy-update* approach. When a mobile host moves, only the primary copy and the previous copy of the LD entry is updated. Cached entries are assumed to be correct by default. When cached entries turn stale, the first packet which is forwarded using the stale entry generates an error message from the old MSR, causing the source MSR to flush its cache and then multicast a `WHO_HAS` message.

Since functions f and g are required to be supported only in new entities (MSRs) that are added to the system, the Columbia proposal can operate without requiring any modifications to the existing host and router software. This proposal presents a good combination of design choices for handling mobility within a campus environment. However, it does not scale well for use with the global Internet.

5.2 Sony Scheme

In Sony's proposal [24, 22, 23], a mobile host is assigned a new temporary address when it is attached to a new network. The router of the home network is notified of this new address through a special control message. Packets addressed to the MH, in addition to carrying its home address, can also carry its temporary address. Packets originating from an MH that is away from its home network always carry both home and temporary addresses in the source address field. Routers that forward these packets can examine the source addresses and cache the mapping (home to temporary) in their Address Mapping Tables (AMT). A source includes both addresses in all outgoing packets if it already has an AMT entry for the target host. Otherwise, packets are forwarded to the home address. If a transit router has an AMT cache entry for the destination, it can intercept the packet and forward it to its correct location. If none of the transit routers have a cache entry, the home router is eventually responsible for forwarding the datagram.

When a host moves to a new location, all AMT cache entries are invalidated through a special `disconnect` control message which is broadcast in the network. Since this message of invalidation is not reliable, there is also a timeout associated with all AMT cache entries, which, on expiration, causes AMT entries to be purged.

This method requires modifications to routers and host software and has problems inter-operating with the existing hosts since it also requires modifications to IP packet formats.

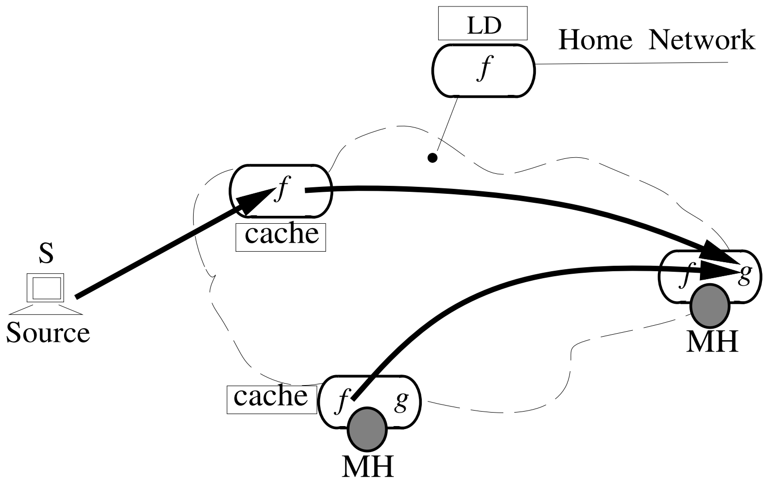

Figure 8: Mapping to Sony Proposal

The Sony proposal co-locates the forwarding agent function, g, with mobile hosts. In other words, it requires each mobile host to act as its own forwarding agent. The advantage is that packets can be directly tunneled to the mobile host, without intervention from a forwarding agent. This is useful, particularly for wired mobile hosts, which may at times connect to foreign networks which have no forwarding agents attached. The approach of co-locating g with the mobile host has a disadvantage. It doubles the address space requirement for mobile hosts, since in addition to a home address, a temporary address is also required for operation. In some environments, this may be a serious problem.

In Sony's proposal, the home router acts as an ATA, and it also maintains the LD for mobile hosts that have been assigned addresses on the home network. To avoid routing each packet via the home router, Sony proposal allows flexibility to co-locate f with internet routers. Since LD cache entries are carried in the source address field of the Virtual IP protocol[4], routers can acquire them just by inspecting the source address of packets they relay. Distributing LD caches across the Internet improves routing performance; however, it makes updates very costly. Sony's proposal, therefore, has a scalability problem. When a host moves to a new location, it is required to send a broadcast in the network to purge all cached LD entries.

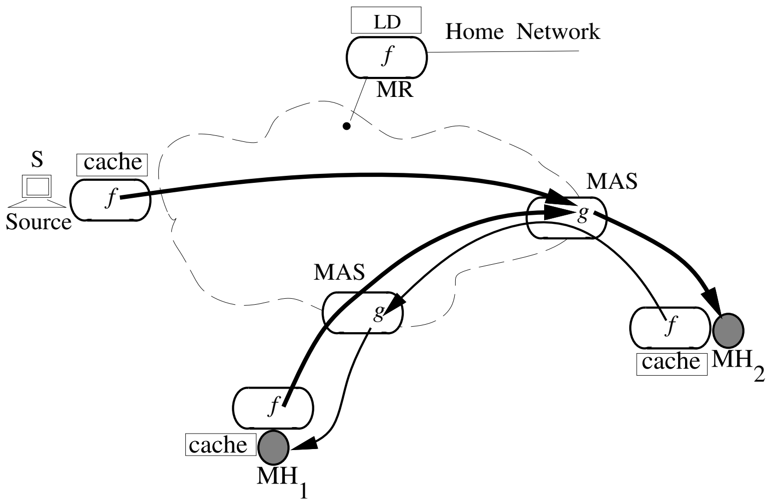

Figure 9: Mapping to LSR Scheme

5.3 LSR Scheme

In contrast with other proposals which are encapsulation based, the LSR proposal [5, 17, 19, 12] is based on the use of an existing IP option called Loose Source Route. The LSR scheme also allows each mobile host to retain its home address regardless of its current location. Associated with each home network is a Mobile Router (MR), which is responsible for advertising reachability to the home network, and for keeping track of the current location of each mobile host that has been assigned an address on that network. In a foreign network, mobile hosts attach to the Internet via wireless base stations known as Mobile Access Stations (MAS). When a mobile host walks into the wireless cell of an MAS, it informs its Mobile Router the internet address of the current MAS. The Mobile Router records this information in its routing table, and also informs the previously recorded MAS that the mobile host has moved out of its wireless cell. The packets sent to the mobile host first arrive at the Mobile Router by the normal routing process. To forward a packet to the a mobile host's current location, the Mobile Router inserts an LSR option in the packet,

[4]The modified IP protocol

specifying the current MAS as a transit router. The inserted LSR option causes this packet to be routed to the mobile host via the MAS. When the mobile host sends a reply to the source, it also inserts the LSR option in all outgoing packets, again specifying the current MAS as a transit router. When the stationary host receives this packet, it will reverse the recorded route, and insert it in all outgoing packets that are sent to the mobile host. Thus, subsequent packets originating from the stationary host will be automatically routed along an optimal path.

Notice that this proposal relies on the end-host's ability to perform route reversal. Unfortunately, a vast majority of hosts in the Internet either do not perform correct route reversal, or, in some cases, even drop LSR packets due to the security risk involved. Another problem is that packets carrying LSR option receive poor service from IP routers. Most router vendors optimize their forwarding loop for the common case of simple IP header. When a packet with options is received, it is bumped into a low priority queue. Due to these limitations, the LSR proposal was not accepted as a candidate for further consideration within the IETF.

In this proposal, the MR acts as an ATA, and is also responsible for maintaining the LD. The MAS acts as an FA for mobile hosts that lie in its wireless cell. The key feature of this proposal is that it enables function f to be co-located with all internet hosts without requiring changes to host software. All internet hosts, when generating replies to packets that are received with the LSR option, are required to do the route reversal[6]. For TCP connections, the route reversal is performed by the protocol processing module, and in case of UDP connections, this responsibility lies with the applications. From our reference architecture view point, the process of route reversal corresponds to the task that an ATA is required to carry out. Thus, this scheme attempts to exploit mechanisms already available within IP protocol, to achieves co-location of ATA with end hosts.

The LSR scheme maintains a distributed version of the LD, yet no special protocol is required for distributing and managing LD cache entries. LD entries are automatically acquired through the incoming LSR option. Recall that packets which arrive at a stationary host already contain the address of the MAS. This, together with the source address of the packet, constitutes an LD cache entry. When a host starts a new session with a mobile host, it has no LD cache entry for the destination. Naturally, the first packet is routed to the destination via the MR. When the ACK for this packet arrives, it contains the LD cache entry in the incoming LSR option. This LD entry is maintained on a per-session basis, and it maintained only as long as the corresponding TCP session is alive. When the session terminates, the corresponding LD entry is purged. If the destination moves during an active session, the LD cache entry becomes inconsistent. However, it gets updated as soon as the next packet from the destination arrives at the source. This constitutes a pure on-demand-cache-update policy which has a good scaling property. Following a host's movement, only those LD cache entries are updated which are in use. Compared with Sony's proposal, which requires a message to be broadcast to the network, significantly fewer messages are exchanged. Naturally, an on-demand-cache-update policy lends a scalable design; both with respect to the size of the network, and the rate of host mobility.

5.4 Mobile IP working-group Proposal

IETF has created a Mobile IP working group to come up with a proposal for near term deployment within the Internet. In this design [16], each mobile host retains its home address regardless of the mobile host's location. When the mobile host visits a foreign network, it is associated with a *care-of-address*, which is an Internet address associated with the mobile host's current point of

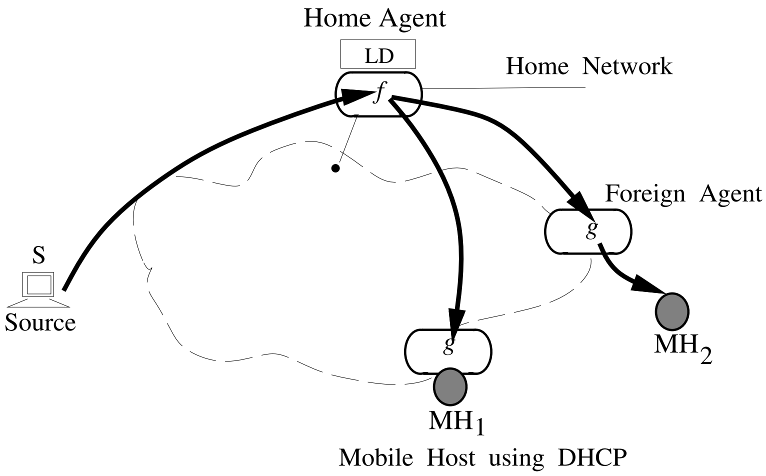

Figure 10: Triangle Routing: Mobile IP Proposal

attachment. The care-of-address either identifies the mobile host directly (if the address is acquired through Dynamic Host Configuration Protocol (DHCP) [8]) or identifies a Foreign Agent that is responsible for providing access to visiting mobile hosts. When away from home, the mobile host registers its care-of-address with a Home Agent; the Home Agent is responsible for intercepting datagrams addressed to the mobile host's home address and tunneling (encapsulating) them to the associated care-of-address.

In this scheme all datagrams addressed to a mobile host are always routed via the Home Agent. However, the packets in the reverse direction, i.e., those originating from the mobile host and addressed to a stationary host, are relayed along the shortest path by the Internet routing system. This gives rise to what is known as the triangle routing problem. Route optimization is possible if the location information is allowed to be cached; however, this proposal does not permit caching of LD entries because of security concerns. Currently, the Internet does not provide any secure mechanism for distributing cache entries. Any entity in the Internet can masquerade as a Home Agent and re-route traffic away from a mobile host just by re-distributing fake cache entries. This proposal, therefore, takes the stand that routing based on cached location information is insecure, and the best possible defense against security attacks is to not use it at all. The cost of this choice is that routing is always non-optimal.

When the mobile host arrives at a foreign network, it can listen for (or solicit) agent advertisements to determine whether a Foreign Agent is available. If so, the registration request to the Home Agent is sent via the Foreign Agent; otherwise, the mobile host must acquire a care-of-address (through DHCP), and then register with the Home Agent.

The IETF Mobile IP proposal reflects a design choice that co-locates f with the Home Agent and g with the Foreign Agent. This proposal also allows g to be co-located with the mobile host. This happens when the mobile host acquires a temporary address via DHCP or PPP. The location update protocol is very simple; the mobile host notifies the Home Agent whenever it moves to a new location. Since the LD entries are never cached, the question of maintaining consistency doesn't even arise.

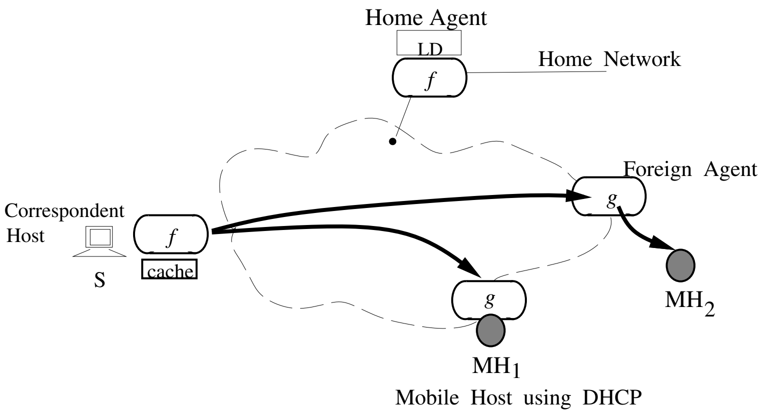

Figure 11: Mobile IP with Route Optimization

5.5 Mobile IP with Route-Optimization

Route optimization [13] is basically a protocol by which Internet hosts can learn the current care-of address for a mobile node – that is, they can create a valid binding (an LD cache entry) for the mobile node, and become address translation agents. Once an Internet host has a valid binding, the host can encapsulate packets and send them directly to the care-of address for the mobile node, just as the mobile node's home agent does in the basic Mobile IP specification. The correspondent host can also optionally use an abbreviated style of encapsulation called *minimal encapsulation* [15], which typically in this case adds 8 bytes to the original IP datagram.

Aside from the difficulty of changing existing Internet hosts to use new techniques to deal with mobility, route optimization faces the additional technical difficulty (and requirement) of enabling the recipient hosts to be sure that the location update information is authentic. The absence of authentication techniques would leave a home agent vulnerable to cooperating with a malicious hosts which wanted to hijack traffic destined for mobile hosts. Similarly, any other host accepting cache updates on the mobile node's location needs to be able to ensure the updates are authentic. Providing a high enough degree of confidence in the authenticity of the location updates has been a driving factor in the design of the route optimization protocol. The goal, then, of route optimization, is to enable the delivery of authentic binding updates (as needed) to arbitrary Internet hosts.

If a correspondent host has no binding for a mobile node, the home agent will receive packets from the correspondent host destined for the mobile node. In this case, the home agent is well placed to send a valid binding to the correspondent host. If the correspondent host has a stale (incorrect) binding for a mobile node, the situation is more difficult. Usually, the binding associates the mobile node to a care-of address offered by a foreign agent which no longer serves the mobile node. In this case, the foreign agent notifies the correspondent host (via a *binding warning* message) to request a new binding update from the home agent. Route optimization assumes that foreign agents typically have no security association with correspondent hosts, and thus cannot send authenticated binding updates directly to them.

If the binding associates the mobile node to a care-of address that is stale or does not exist, then the correspondent host will have to purge its binding in response to an ICMP message indicating that the care-of address is unreachable. The worst case occurs when the care-of address is reachable, but consumes the datagram with no ICMP error and still does not deliver it to the mobile node. In this case, the correspondent node will have to rely on timing out the binding for the mobile node according to its lifetime, or in more fortunate circumstances purging the binding because of error conditions generated by higher level protocols (such as TCP, or the application protocol generating the datagrams). Fortunately the latter case is likely to be rare, only occurring when a correctly functioning foreign agent hangs.

Route optimization proposal improves the basic Mobile IP design by co-locating f with correspondent hosts. The location update protocol operates to maintain valid LD cache entries at those hosts which are likely to send packets to the mobile node. This constitutes an on-demand location update protocol. Binding update messages are authenticated in order to ensure that malicious hosts cannot disrupt traffic between correspondent hosts and mobile nodes using the route optimization protocol. Smooth handoffs between foreign agents are modeled as a special case of the route optimization techniques, and session keys obtained to secure the necessary binding update messages in this common case.

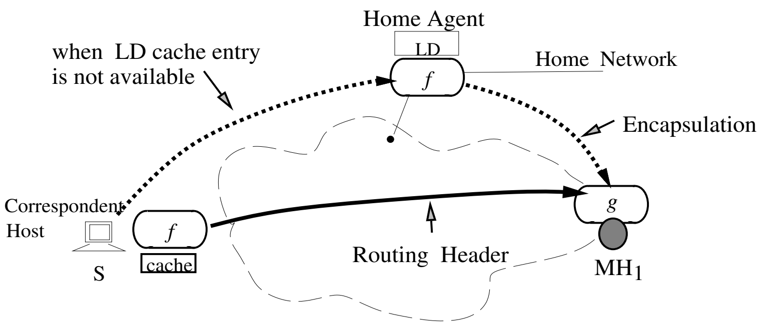

Figure 12: IPv6 Mobility Proposal

5.6 IPv6 Mobility Proposal

A new version of the Internet Protocol (IP) [20] has been designed, and is conventionally known as IPv6 (see [7] for protocol details). IPv6 has a 128-bit address space, and a greatly improved scheme for handling options. Part of the basic set of requirements for IPv6 is that it be designed to handle mobility well. In this section we will briefly describe a proposal which achieves this objective for IPv6. The existing version of IP will sometimes be referred to as IPv4 (IP version 4) to distinguish it from IPv6.

Since there isn't any substantial base of installed IPv6 systems yet, the IPv6 proposal is not constrained by compatibility requirements with existing systems. This is a huge advantage; for example, the main obstacle to designing an efficient protocol for route optimization with the base (IPv4) mobile-IP specification is that most existing Internet hosts cannot be expected to successfully interpret the route optimization messages. If all IPv6 hosts support the mobile-IPv6 protocol described in this section, almost all traffic for mobile nodes will follow optimal routes. Also, since every IPv6 router can be assumed to support mobile-IPv6, every network can be a home network, and any node on that network can roam the Internet.

IPv6 mobility operates by carefully sending binding updates (i.e., LD cache entries) to any host that is likely to need it. The binding update associates a care-of address (just as with IPv4) with the mobile node's home address – but with IPv6 no foreign agents are needed. This is because g can always be co-located with IPv6 hosts. Each mobile node receives packets at the care-of address it obtained via Neighbor Discovery at its current point of attachment; nevertheless its correspondent hosts send datagrams addressed to the mobile node at its home address. However, before the correspondent host actually transmits the datagram, it places the care-of address in a *routing header*, which are analogous to *loose source routes* in IPv4 (see section 5.3). In this way, a mobile node can move from one care-of address to another and still maintain all of its existing connections which are associated with its home address, not any fleeting care-of address. However, as the mobile node moves from place to place, the correspondent host inserts different routing headers in the datagrams before transmitting them.

There is still a home agent (address translation agent) needed for IPv6, in case a correspondent IPv6 host does not have a binding for a mobile node. However, the home agent should not see many packets. Since the home agent tunnels packets to a mobile node by using encapsulation instead of a routing header, the mobile node can easily detect whenever its correspondent hosts do not have a binding for it.

In IPv6, the mobile node is always responsible for delivering binding updates to its correspondent hosts. If a mobile node moves to a new point of attachment, then any correspondent hosts that have recently sent packets to the mobile node should get binding updates. Most correspondent hosts with open TCP connections with the mobile node should get binding updates. And, as just mentioned, any time a mobile node gets a datagram encapsulated by the home agent, the mobile node should certainly send a binding update to the source of that datagram. In IPv6, the mobile node that does the best job of sending out binding updates appropriately will receive the best performance from the Internet, and will place the least load on the Internet. Of course, any such binding update sent to a correspondent host should be authenticated to allow the correspondent host to trust the veracity of the update.

Since there are no foreign agents, we should consider the effects of moving from one point of attachment to another. With each such point of attachment, the mobile node will get a new care-of address; unfortunately there are no foreign agents to help effect smooth handoffs from one point of attachment to the next. However, and especially in the case of wireless communications, there is no reason for the mobile node to halt operation at its previous care-of address. If the mobile node is still within range of the previous point of attachment, it can still receive packets at its previous care-of address. Since each IPv6 node is required to be able to handle multiple IPv6 addresses at each of its network interfaces (i.e, each IPv6 must have multi-homing capability), each IPv6 mobile node must be prepared to handle multiple care-of addresses as needed. With this in mind, we expect that smooth handoffs will be easily processed by the mobile nodes themselves without any help from foreign agents. In the case of wired attachments, of course, smooth handoffs are simultaneously harder to visualize and more difficult to provide for by such multi-homing techniques.

The IPv6 mobility proposal reflects an ideal design choice which co-locates f and g with all IPv6 hosts. The proposal employs a new address translation mechanism called routing header, which is functionally similar to the LSR option of IPv4. The location update protocol is based on on-demand update policy and mobile hosts are responsible for issuing all location updates. Overall, IPv6 looks very promising as an efficient and natural protocol for supporting mobility. See [18] for details about the protocol.

6 Summary

In this paper, we first identified network layer concepts that play a crucial role in the design of mobile networking systems. We showed that the process of *address translation* is fundamental to any mobility solution at the network layer. Our proposed network architecture employs three basic set of entities: *Address Translation Agent*, *Forwarding Agent*, and *Location Directory*, which co-operate with each other to carry out the operation of address translation. We showed that all candidate proposals for Mobile IP can be visualized as a specific instance of our general architecture. We demonstrated this by showing a one-to-one mapping between the entities in our architecture, and those required by the candidate proposals. Mappings represent set of design choices (i.e., where in the network these entities are located) made in the candidate proposals. Table 2 presents a summary of our observations.

In addition to these design choices, there are several other considerations such as inter-operability, backward-compatibility, security, and authentication, which also play a crucial role in the design of a mobile networking system. Interested readers can refer to articles [23, 25, 11, 14] for an in-depth description of design and implementation issues.

References

[1] V. Anantharam, M. L. Honing, U Madhow, and V. K. Wei. Optimization of a database hierarchy for mobility tracking in a personal communications network. In *Proceedings of Performance 93*, Septembero 1993.

[2] Baruch Awerbuch and David Peled. Online Tracking on mobile users. In *Proceedings of ACM SIGCOMM*, 1991.

[3] Amotz Bar-Noy and Ilan Kessler. Tracking Mobile Users in wireless communication Networks. *IEEE Transactions on Information Theory*, pages 45–65, Jan 1994.

[4] Amotz Bar-Noy, Ilan Kessler, and Moshe Sidi. Mobile Users: To Update or not to Update? In *Proceedings of IEEE INFOCOM*, pages 570–576, Toronto, Canada, June 1994.

[5] Pravin Bhagwat and Charles Perkins. A Mobile Networking System based on Internet Protocol(IP). In *Proceedings of USENIX Symposium on Mobile and Location Independent Computing*, pages 69–82, Cambridge, MA, Aug 1993.

[6] R. Braden. Requirements for Internet Hosts – Communication Layers. RFC 1122, Oct. 1989.

[7] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC1883, Dec 1995. (ftp://ds.internet.nic/rfc).

[8] R. Droms. Dynamic Host Configuration Protocol. RFC1541, Oct. 1993. (ftp://ds.internet.nic/rfc).

[9] Peter S. Ford, Yakov Rekhter, and Hans-Werner Braun. Improving the Routing and Addressing of IP. *IEEE Network*, 7(3):10–15, May 1993.

[10] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr. IP-based Protocols for Mobile Internetworking. In *Proceedings of ACM SIGCOMM*, pages 235–245, 1991.

[11] John Ioannidis and Gerald Q. Maguire Jr. The Design and Implementation of a Mobile Internetworking Architecture. In *Proceedings of Winter USENIX*, pages 491–502, San Diego, CA, Jan 1993.

[12] David B. Johnson. Mobile host internetworking using ip loose source routing. Technical Report CMU-CS-93-128, Carnegie Mellon University, Pittsburgh, PA 15213, February 1993.

[13] David B. Johnson and Charles Perkins. draft-ietf-mobileip-optim-03.txt. Draft RFC - work in progress, Nov. 1995. (ftp://ds.internet.nic/internet-drafts).

[14] Andrew Myles and David Skellern. Comparing Four IP Based Mobile Host Protocols. In *Joint European Networking Conference*, May 1993.

[15] Charles Perkins. draft-ietf-mobileip-minenc.txt. Draft RFC - work in progress, Oct. 1995. (ftp://ds.internet.nic/internet-drafts).

[16] Charles Perkins. draft-ietf-mobileip-protocol-15.txt. Draft RFC - work in progress, Feb 1996. (ftp://ds.internet.nic/internet-drafts).

[17] Charles Perkins and Pravin Bhagwat. A Mobile Networking System based on Internet Protocol. *IEEE Personal Communication Magazine*, 1(1):32–41, Feb 1994.

[18] Charles Perkins and David Johnson. draft-ietf-mobileip-ipv6-00.txt. Draft RFC - work in progress, Jan. 1996. (ftp://ds.internet.nic/internet-drafts).

[19] Charles Perkins and Yakov Rekhter. Short-cut Routing for Mobile Hosts. Internet draft, July 1992.

[20] J. Postel. Internet Protocol. RFC 791, Sep 1981.

[21] J. Postel. Transmission Control Protocol. RFC 793, Sep 1981.

[22] Fumio Teraoka and Mario Tokoro. Host Migration Transparency in IP Networks. *Computer Communication Review*, pages 45–65, Jan 1993.

[23] Fumio Teraoka, Keisuke Uehara, Hideki Sunahara, and Jun Murai. VIP: A Protocol Providing Host Mobility. *Comminications of the ACM*, 37(8), August 1994.

[24] Fumio Teraoka, Yasuhiko Yokote, and Mario Tokoro. A Network Architecture Providing Host Migration Transparency. In *Proceeding of ACM SIGCOMM*, Sept 1991.

[25] Hiromi Wada, Takashi Yozawa, Tatsuya Ohnishi, and Yasunori Tanaka. Mobile Computing Environment Based on Internet Packet Forwarding. In *proceeding of Winter USENIX*, pages 503–517, San Diego, CA, Jan 1993.

Scheme	Address Translation Agent (f)	Forwarding Agent (g)	Location Directory	Location Update Protocol
Columbia	Co-located with MSR	Co-located with MSR	distributed among MSRs	Only primary copy is modified. *Lazy-update* policy is used for updating cache entries
Sony	Co-located with all hosts and routers	Co-located with mobile hosts	LD is maintained at home router. Cache entries are acquired by snooping a packet header	Only primary copy is modified by the explicit `connect` message. Cache entries are modified by broadcasting a `disconnect` message, or are auto-flushed by a timeout mechanism
LSR scheme	Co-located with all hosts and home routers	Co-located with mobile hosts	LD is maintained at home router. Cache entries are acquired through incoming LSR option	Only primary copy is modified. Cache entries automatically get updated when packets with new LSR option arrive. On-demand update policy, no broadcasts.
IETF Mobile IP	Co-located with home routers	Co-located with Foreign Agent, or with mobile host if DHCP is used.	LD is maintained at home router only.	Caching of LD entries is not allowed. When a host moves only the primary copy is modified.
with Route Optimization	Co-located with correspondent hosts		Correspondent hosts can cache LD entries	Home Agent is responsible for sending binding update message.
IPv6 Proposal	Co-located with with all hosts and home routers	Co-located with mobile hosts	LD is maintained at home router. Cache entries are acquired through binding update messages issued by mobile hosts.	Mobile host is responsible for updating the primary copy and all LD cache entries. On-demand update policy, no broadcasts

Table 1: Functional comparison of Mobile IP schemes

Table 2: Comparison of Mobile IP schemes

Property	Columbia		Sony	LSR	IETF Mobile-IP		IPv6
	local	popup				+route-opt	
Optimal Routing	Always	Never	Only if all routers are modified	Provided hosts support correct route reversal	Never	Always	Always
Address Translation Mechanism	Encapsulation		Encapsulation	LSR	Encapsulation		Routing Header
Additional Address Space Required	None	Double	Double	None	None. But required when using DHCP.		Double
Scalability	Good		Poor	Good	Good		Excellent
Compatibility with IP	Total		Requires changes	Only if hosts and routers conform to standards	Total	Requires Changes to Hosts	compatible with IPv6
Security	No Authentication		Insecure	Insecure	Fully Secure		Security via authentication header

Table 1: Property Comparison of MobileIP schemes