Flexible Connectivity Management for Mobile Hosts

by
Xinhua Zhao and Mary G. Baker
{zhao, mgbaker}@cs.stanford.edu

Technical Report: CSL-TR-97-735
September 1997

Computer System Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305-9040

Abstract

Powerful light-weight portable computers, the availability of wireless networks, and the popularity of the Internet are driving the need for better networking support for mobile hosts. Users should be able to connect their portable computers to the Internet at any time and in any place, but the dynamic nature of such connectivity requires more flexible network management than has typically been available for stationary workstations.

This report proposes techniques to address a unique feature of connectivity management on mobile hosts: its multiplicity, i.e. the need to support multiple packet delivery methods simultaneously and to support the use of multiple network devices for both availability and efficiency reasons.

We have developed a set of techniques in the context of mobile IP for flexible, automatic network connectivity management for mobile hosts. We augment the routing layer of the network protocol stack with a Mobile Policy Table (MPT) to support multiple packet delivery mechanisms for different simultaneous flows based on the nature of the traffic. We also devise a set of mechanisms, including a backwards-compatible extension to the routing table, to facilitate the use of multiple network devices. We include performance results showing some of the potential benefits such increased flexibility provides for mobile hosts.

Key Words and Phrases: Mobile host, TCP/IP networking, Mobile-IP, Mobile Policy Table (MPT)

Table of Contents

List of Figures

List of Tables

1. Introduction

Lighter-weight portable computers, the spread of wireless networks and services, and the popularity of the Internet combine to make mobile computing an attractive goal. With these technologies, users should be able to connect to the Internet at any time and in any place, to read email, query databases, retrieve information from the web, or entertain themselves.

However, we find in practice that managing the network connectivity of a mobile host can be a complex task. Through day-to-day experience using the MosquitoNet [1] mobile network, we have found two key areas that require improved management. The first is support of multiple packet delivery mechanisms, and the second is support for managing multiple network interfaces. Due to the diversity of environments a mobile host may visit, there may not be a single packet delivery mechanism (for instance, mobile IP [13] or regular IP) that is the most suitable for all its traffic. As a mobile host moves and corresponds with different hosts, the best choice of packet delivery methods may change. Furthermore, mobile users often have multiple network devices, e.g. both Ethernet and packet radio, and use one or more of them at any time, depending on what networks are available in a particular location. Managing these changing network attachments also poses a challenge. This dynamic character of mobile network connectivity presents a more difficult support problem than is usually found in a static workstation environment.

We have developed a set of techniques in the context of mobile IP for flexible, automatic network connectivity management for mobile hosts. First, we augment the routing layer of the network protocol stack with a Mobile Policy Table (MPT) that allows a mobile host to choose different packet delivery mechanisms for different simultaneous flows based on the nature of the traffic. By avoiding a single method of delivery, the mobile host pays for the extra cost of mobility support or security perimeter traversal only when it is truly needed. Such flexibility can provide significant performance benefits. In our test environment, we can cut round-trip latency in half by choosing regular IP delivery over mobile IP whenever possible. Second, we use a set of mechanisms, including a backwards-compatible extension to the routing table, to manage the use of multiple network devices. The connectivity management mechanisms automatically determine which devices are currently connected to available networks and help configure them appropriately.

In the next section of this report, we briefly describe the relevant aspects of mobile IP and our implementation. In Section 3 we illustrate some of the different packet delivery methods enabled by our techniques. In Section 4 we describe how the flexible packet delivery methods are supported in the Linux operating system through the use of the Mobile Policy Table. In Section 5 we address the need for supporting multiple interfaces and describe how our system enables automatic configuration as the set of available interfaces changes. In Section 6 we provide some performance

measurements to show the benefits such flexibility provides. Finally, in Section 7 we describe related work and we conclude in Section 8.

2. Mobile IP

The Mobile IP protocol [13] is a mechanism developed by IETF for maintaining transparent network connectivity to mobile hosts. Mobile IP allows a mobile host to be addressed by the IP address it uses in its home network (home IP address) regardless of the network to which it is currently physically attached. Hosts corresponding with the mobile host (correspondent hosts) may continue to use its home IP address and do not need to know where it is actually located. Packets sent to the mobile host's home IP address are intercepted in the home network by a stationary host (the home agent) and are encapsulated in a new IP packet and forwarded (tunneled [14]) to the mobile host's current point of attachment (care-of address). At the destination, the packet is decapsulated, revealing the original packet sent by the correspondent host.

In the reverse direction, the mobile host sends packets directly to the correspondent hosts, but it uses its home IP address as the source address on these packets, regardless of its current location, to make it appear to the correspondent hosts that these packets originated in its home network. The mobile host thus maintains its home identity even when visiting other networks. Since the paths to and from the mobile host form a triangle between the correspondent hosts, the home network and the mobile host, as shown in Figure 2-1, it is called a triangular route.

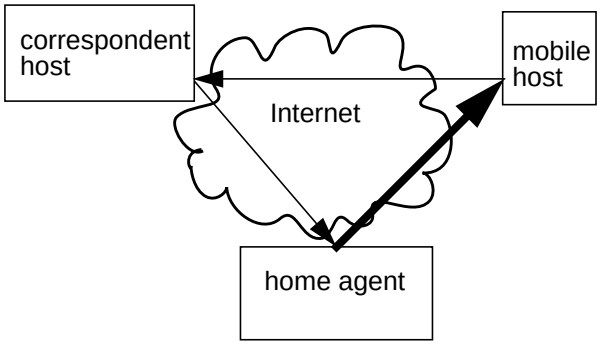

Figure 2-1. Mobile IP protocol

In the basic mobile IP specification, packets from the correspondent host to the mobile host are always sent to the mobile host's home network first, and then forwarded by the home agent to the mobile host's current point of attachment. Packets originating from the mobile host are sent directly to the correspondent host, thus forming a triangular route. These packets use the mobile host's home IP address as their source address to preserve its home identity. The thick line indicates the original packet is encapsulated in another IP packet when forwarded, and is therefore of a larger size.

The mobile IP specification allows for two types of attachment for a mobile host visiting a "foreign" network (a network other than the mobile host's home network). For the first type of attachment, the mobile host can connect to the foreign network through a "foreign agent" by registering the foreign agent's IP address with its home agent. The home agent then tunnels packets to the foreign agent which decapsulates them and sends them to the mobile host via link-level mechanisms. The mobile host in this scenario does not need its own IP address on the foreign network, because it is reachable only through the foreign agent. The mobile host's care-of address is merely the IP address of the foreign agent.

The second type of attachment provides a mobile host with its own "co-located" care-of address in the foreign network. In this scenario, the mobile host receives an IP address to use while it visits the network, via DHCP [5] or some other protocol or policy. It registers this address with its home agent, which then tunnels packets directly to the mobile host at this address. The mobile host's networking software is responsible for decapsulating the packets itself. While this scenario requires more IP addresses, it also allows the mobile host to be more directly responsible for the addressing and routing decisions for the packets it sends out.

Although our mobile IP implementation in the Linux operating system supports either the use of foreign agents or a co-located care-of address, the mechanisms for connectivity management described in this report assume the use of a co-located care-of address, in which mode a mobile host has the maximum flexibility for acting both as a host virtually attached to its home network and as a normal host in the network it is visiting. These mechanisms are largely applicable to mobile hosts running with a foreign agent, but it is easier to implement this flexibility when the mobile host can obtain its own IP address in the foreign network. For instance, a mobile host with its own IP address on the foreign network can choose to send certain traffic via normal IP and other traffic via mobile IP. If the mobile host must operate through a foreign agent, it will need to make complex arrangements with the foreign agent to accomplish this same task.

While mobile IP has laid the groundwork for Internet mobility, there are still many challenges to tackle, as seen from on-going efforts in this area. These efforts include route optimization [9], firewall traversal [7], and "bi-directional tunneling" (or "reverse-tunneling") to allow packets to cross security-conscious boundary routers [12]. This last problem, as described in Section 3, is one of our motivations in making it possible for mobile hosts to choose dynamically between different packet addressing and routing options. We believe that these efforts are evidence that mobile hosts will need to use different techniques for different circumstances, and that connectivity management and configuration support such as that described in this report is needed to make this process automatic for users.

3. Supporting multiple delivery methods

The first goal of this work is to avoid the undue cost of a single packet delivery method on a mobile host. By manipulating the Mobile Policy Table, a mobile host can decide to use different delivery methods for different flows of traffic and can change the behavior dynamically on a per-flow or even per-packet basis.

In this section, we illustrate some of the situations for which we have found such flexibility to be beneficial in practice. While these are the only examples we have so far implemented, the mechanism can be extended to support other delivery methods when other choices become desirable.

3.1. Incurring the cost of transparent mobility support only when necessary

Mobile IP is useful when a mobile host wishes to maintain communication with correspondent hosts while changing its point of attachment to the Internet. Using this protocol, a mobile host may be reached by its home IP address regardless of its current care-of address. Such transparent mobility support is important for long-lived connection-oriented traffic such as a telnet session, or for traffic initiated by correspondent hosts to reach a mobile host no matter where it is currently located.

However, this transparent mobility support does not come without cost. In the absence of route optimization for Mobile IP, packets destined to a mobile host are delivered to its home network and then forwarded to the mobile host's current care-of address in the network it is visiting. If a mobile host is far away from home but relatively close to its correspondent hosts, the path traversed by these packets is significantly longer than the path traveled if the mobile host and the correspondent host can talk to each other directly. The extra path length not only increases latency, but it also generates extra load on the Internet. It even increases load on the home agent, potentially contributing to a communication bottleneck if the home agent is serving many mobile hosts simultaneously.

Fortunately, there are certain types of traffic for which a mobile host may not require Mobile IP support. An example is most web browsing traffic. Web connections are usually short-lived (an exception is the current web push technology which uses long-lived transport connections for now but is heading towards connectionless multicast), so it is unlikely that a mobile host will change addresses in the midst of a connection. Even if it does, the user can simply press reload, and the web transfer will be retried. Also, it is the mobile host that initiates the web transfers most of the time, so it is usually not necessary for the server (the correspondent host) to recognize the mobile host across different connections if the mobile host changes addresses.

3.2. Supporting bi-directional tunnels and triangular routes simultaneously

When communication requires transparent mobility, there still remains a choice of packet delivery methods. An important example is communication that must traverse security-conscious boundary routers.

As a result of IP address spoofing attacks, and in accordance with the IAB [6] and CERT [4] advisories, more routers are filtering on the source address as well (ingress filtering) and will drop a packet whose address is not “topologically correct” (whose originating network cannot be the one identified by the source address). In the presence of such routers, the triangle route as specified in the base Mobile IP protocol will fail. Figure 3-1 illustrates an example of this problem.

As another example, if the boundary router is in the domain visited by the mobile host, it may drop packets that are received from inside but claim to originate from outside; these packets look as if they are “transit traffic,” and not all networks will carry transit traffic.

We can address the above problems by tunneling packets sent by the mobile host through its home agent to its correspondent hosts, in much the same way packets sent to the mobile host are tunneled. This is called “bi-directional tunneling” [12]. Figure 3-2 illustrates the solution.

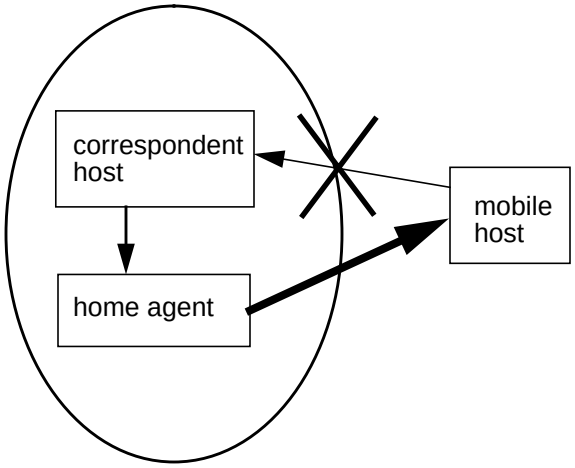

Figure 3-1. Ingress filtering router

This figure illustrates the problem with source IP address filtering when a security-conscious boundary router is in the mobile host’s home domain. When the mobile host sends packets directly to the correspondent host in its home domain with the source IP address of the packets set to the mobile host’s home IP address, these packets will be dropped by the boundary router because they arrive from outside of the institution and yet claim to originate from within.

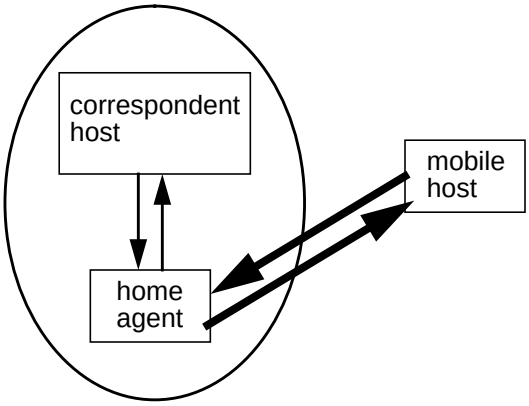

Figure 3-2. Bi-directional tunneling

To address the problem caused by source IP address filtering on security-conscious boundary routers, the mobile host sends packets by tunneling along the reverse path as well. Since the encapsulated packets in the tunnel from the mobile host to its home agent use the mobile host's care-of address as their source IP address (which is topologically correct), these packets will no longer be dropped by the security-conscious boundary routers.

This bi-directional tunneling addresses the problem related with ingress filtering routers, but again, this doesn't come without cost. If a mobile host visits a network far away from home and tries to talk to a correspondent host in a nearby network, packets originating from the mobile host will now have to travel all the way home and then back to the correspondent host, increasing the length of this reverse path.

However, not all the packets need to be sent this way. It is unnecessary to force all traffic through a bi-directional tunnel just because ingress filtering routers would drop traffic sent to specific destinations. Such tunneling may be unnecessary for a large part of the traffic for which the topologically incorrect source IP address in packet headers is not a problem.

Fortunately, we can avoid unnecessary bi-directional tunnels by supporting both the triangle route and the bi-directional tunnel simultaneously through appropriate entries in the MPT. This choice usually depends upon the destination network, and only certain destinations will require the overhead of the bi-directional tunnel. We insert entries matching these destinations into the MPT, and all packets with a matching destination address will be sent encapsulated through the home agent, unless a more specific entry (by port number perhaps) indicates otherwise. However, there are scenarios when the default behavior should be to use bi-directional tunneling. If the host is visiting a network that drops transit traffic, the default entry for the MPT will specify bi-directional tunneling, and only more specialized entries will turn it off for other traffic to destinations within the local domain.

3.3. Joining multicast groups in different ways

The final packet delivery mechanism we have experimented with is to allow a mobile host to choose to join multicast groups either remotely (through its home network), using its home IP address, or locally, using its care-of address in the foreign network. This flexibility is useful, because there are advantages and disadvantages to either choice.

- Joining through the home network: All multicast traffic has to be tunneled bi-directionally between the mobile host and its home agent. The advantages of this choice are that it does not require multicast support on the foreign network, and the mobile host will retain its membership as it moves around. The disadvantages are that the route is less efficient, and the home agent has to tunnel a copy of a multicast datagram to each mobile host that joins the multicast group this way, regardless of what foreign networks they are visiting.
- Joining locally: In this respect the mobile host is no different from any normal host on the same subnet. The advantage is that the delivery of multicast traffic to the mobile host is more efficient. The disadvantages are that it requires the existence of a multicast-capable router in the foreign network, and those mobile hosts actively participating in the multicast session will lose their identity within the group when they move to another network.

We can accommodate either choice independently of whether Mobile IP is used for other traffic and switch between them easily. This is done by adding entries in the MPT to instruct traffic destined to certain multicast addresses to use either conventional IP support or bi-directional tunneling. To join a multicast group locally, we add an entry in the Mobile Policy Table instructing traffic destined to certain multicast addresses to use conventional IP support, meaning Mobile IP is not used for these addresses. To join a multicast group remotely we add an entry in the MPT to use bi-directional tunneling for traffic destined to multicast addresses. The mobile host also needs to notify its home agent, in a registration packet, to forward multicast packets to it.

4. A mechanism for flexible connectivity management

In this section we describe the general-purpose mechanism we use to enable mobile hosts to choose the different delivery methods described in the previous section. The central idea is to introduce a Mobile Policy Table (MPT) in the IP route lookup routine. Whenever a route lookup is done to determine how a packet should be sent, the MPT is consulted together with the normal routing table.

The routing and addressing policy decisions currently supported are

- whether to use transparent mobility support (mobile IP) or use conventional IP;
- whether to use triangular routing or bi-directional tunneling if using mobile IP.

These policies are specified through two types of entries: “per-socket” entries and “generic” entries, with per-socket entries taking precedence. A per-socket entry allows any application to

override the general rules. Without a per-socket entry, traffic is subject only to generic entries in the policy table, which specify the delivery policy for all traffic matching the given characteristics. For generic entries, the MPT lookup determines which policy entry to use based on two traffic characteristics: the destination address and the port number (for TCP and UDP). The destination address is useful, because we often want to treat flows to different destinations differently. The port number is useful as well, because there are many reserved port numbers that indicate the nature of the traffic, such as TCP port 23 for telnet, or port 80 for HTTP traffic. While these are the characteristics currently taken into consideration, we can extend the technique to include other characteristics in the future.

The MPT lookup operation always chooses the most specifically matched entries, i.e. those with more restricted netmask and/or port number specifications, over more general ones. This lookup differs from a routing table lookup in that the port number also has to be taken into consideration. Table 4-1 shows, as an example, the MPT currently used on our mobile hosts when visiting places outside of their home domain.

Destination	**Netmask**	**Port Number**	**Transparent Mobility**	**Bi-directional Tunneling**
a.b.0.0	255.255.0.0	0	Yes	Yes
0.0.0.0	0.0.0.0	80	No	N/A
0.0.0.0	0.0.0.0	0	Yes	No

Table 4-1. A sample mobile policy table

This mobile policy table specifies that all traffic destined back to the mobile host's home domain should use bi-directional tunneling to satisfy the boundary routers at its home institution; all traffic to port 80 (web traffic) should avoid using transparent mobility support; and all the rest of the traffic should by default use mobile IP with a regular triangular route. The second entry applies to all traffic with a destination port number of 80, even for destinations matching the first entry, since port number specification takes precedence.

Figure 4-1 illustrates the use of the MPT and routing table within the Linux kernel. The modification to the kernel is mainly limited to the route lookup function. For backwards compatibility, the normal routing table remains intact. During a route lookup, extra arguments such as the source IP address to use (if it has already been chosen), and other characteristics of the traffic flow (currently the TCP or UDP port number) are used in addition to the destination address of the packet in deciding how the packet should be sent.

The new route lookup function uses the source IP address to determine if the packet is subject to policy decisions in the MPT. If the source IP address has already been set to the IP address associated with one of the physical network interfaces, this indicates that no mobility decision should be

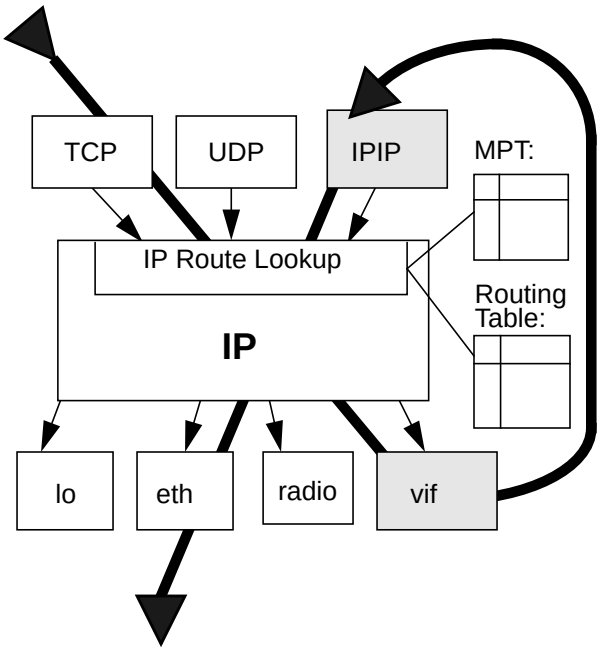

Figure 4-1. Mobile Policy Table on mobile hosts

This figure shows where the Mobile Policy Table (MPT) fits into the link, network, and transport layers of our protocol stack. The MPT resides in the middle (network) layer, consulted by the IP route lookup function in conjunction with the normal routing table to determine how packets should be sent. The bottom (link) layer shows the device interfaces, with vif being a virtual interface that handles encapsulation and tunneling of packets. The top (transport) layer shows TCP and UDP, along with an IP-within-IP processing module. The solid arrows depict the passing of data packets down the layers. The shaded boxes indicate that the IPIP and vif modules are actually implemented as a single module. The bold line shows the path an outgoing encapsulated packet may take.

made for the packet. Packets may have their source address set either by the virtual interface (described below), or by applications (such as the mobile host daemon handling registration and deregistration with the home agent) that wish to force a packet through a particular real interface using normal IP. In this case, only the normal routing table is consulted based upon the destination address and the resulting route entry is returned. For the rest of the packets, the MPT needs to be consulted in order to choose among multiple packet delivery methods.

The virtual interface (“vif”) handles packets that need to be encapsulated and tunneled. It provides the illusion that the mobile host is still in its home network. Packets sent through vif are encapsulated and then looped back to the IP layer (as shown by the wide looped line in the figure) for delivery to the home agent. This time, however, the source IP address of the encapsulating packet has already been chosen, so it will now be sent through one of the physical interfaces.

To maintain reasonable processing overhead, policy table entries are cached in a manner similar to routing table entries. If the characteristics of the traffic match a cached entry, the software uses the

cached entry to speed up the process of policy lookup. Otherwise, a new policy table lookup will be carried out. Whenever the mobile policy table is modified, the cached entries are flushed.

We believe this is a general-purpose mechanism, because it can be easily extended to take other traffic characteristics into consideration and to add more policy decisions if it becomes desirable to do so. For instance, if particular correspondent hosts have the ability to decapsulate packets themselves, we could note this information in the MPT for those destinations, and the mobile host could send encapsulated packets directly to these hosts, bypassing the home agent yet still providing the robustness of a bi-directional tunnel. It is also a flexible mechanism, because it provides the mobile host with a set of delivery choices for different traffic types.

5. Supporting multiple interfaces

In this section, we turn to the second goal of this work, i.e. facilitating the use of multiple network devices. To achieve connectivity in any place at any time, mobile hosts will likely require more than one type of network device. For example, our mobile hosts use 10 or 100 Mbit Ethernet when in a suitably equipped office or home, but they use a slower packet radio network elsewhere. This use of multiple devices is one of the network characteristics that makes mobile hosts more complex than the ordinary stationary host. Stationary hosts do not normally require multiple functioning network interfaces unless they are used as routers. On a mobile host, these interfaces represent multiple ways the mobile host may connect to the outside world, and do not necessarily imply that the mobile host is a router.

Managing the configuration of a changing set of active devices and making use of them simultaneously is a challenge. In this section we describe our attempt to manage the following two associated issues.

5.1 Detecting available networks

When multiple network devices are available, managing their configuration and determining which care-of address to use in registering with the home agent becomes an issue. First, the existing system usually uses device insertion or activation events to trigger switching between devices. This does not work well with PCMCIA Ethernet cards that are inserted but not connected to a network, or with radios that are out of range with other radios. Second, the mobile host needs to figure out which network these devices are on.

Automating this process is important. Manual configuration changes should be avoided if possible, since it requires an unrealistic level of expertise from users. Especially in a mobile environment where connectivity is not always guaranteed, it is extremely hard for a user to figure out if the

problem is simply bad configuration or if it is due to communication media failure. We have devised a set of practical and simple mechanisms to address this problem.

First, we do not attempt to use an interface unless link level detection is successful. We have added an *ioctl* call in the device driver that can probe the PCMCIA Ethernet card for the presence of network media to the interface. If the ioctl returns "unconnected," the configuration script will not attempt to use the interface. While our packet radio network is generally available in our whole geographic area, it is appropriate to probe for radio connectivity using the same technique. In this case, the result depends on whether the radio has detected the existence of other radios in its vicinity. This detection of available networks avoids a lot of potential confusion and is important to our second mechanism.

After we have figured out which network devices are available, we need to decide what configuration (including the IP address, netmask, and gateway address) to use for each interface. Our approach is to try to obtain configuration information from DHCP [5] first. If no DHCP server is available on the local subnet, we next refer to a set of configurations specified by the user. The user can specify this information for a set of known locations that the mobile host frequently visits. The mobile host tests the configuration for each location until one works. To test the setup, the host configures a minimal route (the direct route to the new subnet) and attempts to ping the first-hop router on this subnet through this interface. If the gateway is reachable, the host is on the corresponding subnet. Finally, if none of the configurations fits, the mobile host probes for a foreign agent.

In case none of the above works, manual configuration still has to be the last resort. Although we have not done so yet, we plan to prompt the user with a friendly interface for the necessary configuration information.

We believe this sequence of steps provides the user with the most convenient scenario for identifying the current network. We provide the users with a way to specify a set of configurations to try based on the observation that many users have a small set of networks they visit from time to time and they usually have IP addresses assigned to their mobile hosts in the subnets. Whatever authority provides the IP address to the user could supply the other configuration information for that entry in the user's list. This would help shield the user from a need to understand the details of the configuration information. We choose to probe for a DHCP server and preset configurations before seeking foreign agent support, because we favor the use of a co-located care-of address on a mobile hosts for the extra flexibility it makes possible. However, it can be argued whether we should probe for a DHCP server or the preset configurations first. Our choice reflects our hope that more networks will begin to use DHCP and that eventually there will be no need for users to have any preset configuration information.

5.2 Using multiple interfaces simultaneously

The approach described in the previous section does not continuously probe and monitor the availability of the networks associated with the multiple network devices. With multiple wireless devices such as WaveLan [18], Metricom radio [11], or CDPD [2], this active monitoring becomes more important since it is harder for users to determine if they still have connectivity through a certain device. We must be able to probe beyond immediate hop through a certain interface while still maintaining the overall default route for traffic that does not specify a particular interface. Most existing systems do not support this, since they only have a single default route in use and do not consider device choices in route lookup.

We make use of the metric field in the existing Linux routing table entry to associate a different route for each interface without interfering with how the normal default route is used. Usually, the normal default route has a metric of one, and we set the routes through the other interfaces to have metrics greater than one. A route lookup that does not specify a particular interface will thus find the default route, since the lookup always chooses the matching route with the smallest metric. However, for those sockets specified to use a certain interface, we have modified the route lookup so that only routes for that interface will be considered. Table 5-1 illustrates a sample routing table.

Destination	**Gateway**	**Netmask**	**Flags**	**Metric**	**Iface**
a.b.0.0	0.0.0.0	255.255.0.0	U	0	eth0
c.d.0.0	0.0.0.0	255.255.0.0	U	0	st0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	lo
0.0.0.0	a.b.0.1	0.0.0.0	UG	1	eth0
0.0.0.0	c.d.0.1	0.0.0.0	UG	100	st0

Table 5-1. A sample routing table

This is a routing table with different first-hop routers associated with different network interfaces. eth0 is an Ethernet interface, st0 is a packet radio interface. Traffic forced through the radio interface will use the gateway address c.d.0.1. Normal lookups that do not specify a specific interface will choose a.b.0.1, since it has a lower metric than the route to the radio's gateway. Thus the default route uses the ethernet interface.

This mechanism has several uses. First, it enables a mobile host to probe beyond the local subnet level to hosts such as its home agent through different network interfaces. Second, it can provide a smoother handoff between networks for mobile IP. We can safely register with the home agent the IP address associated with a non-default interface before changing the default route to use that interface and gateway. Finally, we can direct traffic to different interfaces as desired by applications that want to have control over which interface to use.

To make the last option (directing traffic to different interfaces) possible, we provide a “bind-to-device” socket option that applications can call to associate a specific device with a certain socket. This makes it possible to use multiple interfaces for different flows simultaneously.

6. Performance measurements

The performance results in this section are just an example of the benefits a user may possibly see. These results depend on the relative locations of the mobile host, its home network and the correspondent host, as well as on the types of networks packets traverse, and so results in other network configurations may vary significantly.

The setup of our test environment is illustrated in Figure 6-1. The hosts are connected to Ethernet segments on campus. We do not use our wireless network for these experiments, since its higher latency exaggerates the results.

6.1. Benefits of multiple delivery methods

The choice of delivery method affects both latency and throughput. This set of performance data is collected on a mobile host in a real scenario to illustrate the potential benefits enabled by the flexible mechanism.

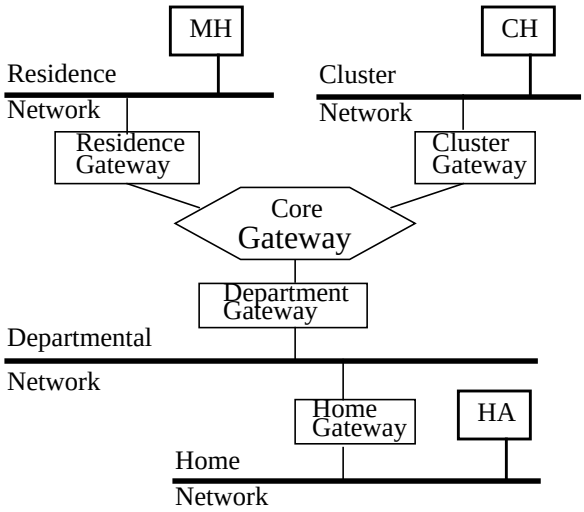

Figure 6-1. Setup of the test environment

The mobile host visits the campus residence network as a foreign network, and accesses computing servers in the cluster network. The mobile host (MH) is a Thinkpad 560, and the home agent (HA) is a 90MHz Pentium, both running Linux. The correspondent host (CH) is a SPARCstation 20 running SunOS.

First, we look at the latency improvement resulting from using the most direct route possible under the circumstances. For these experiments, the mobile host connects to a foreign network in one of the campus residences, which is also on Ethernet, and registers its current care-of address with its home agent. The mobile host sends ping (ICMP echo request) packets to the default gateway of its local subnet and we measure the round-trip latency using the following three delivery methods, switching between them by manipulating the Mobile Policy Table:

- Conventional IP (no transparent mobility support);
- Triangular delivery (the default behavior in the mobile IP specification);
- Bi-directional tunneling (for security-conscious boundary routers).

We collect the data by repeating the above test 100 times with an interval of two seconds for each delivery method. The results are shown in Table 6-1 for small packets and Table 6-2 for large packets. For this experimental setup, we reduce the latency by one half for both small and large packets when choosing regular IP over the default mobile IP behavior. Even when mobility support is necessary, we still reduce the latency by about one third for small and large packets when we can choose the triangular route over the safe but costly bi-directional tunnel.

Delivery Method	**Min (ms)**	**Max (ms)**	**Average (ms)**	**Standard Deviation**
Bi-directional Tunneling	7.3	9.3	7.9	0.36
Triangular Route	4.5	6.2	5.0	0.35
No Transparent Mobility Support	2.0	4.0	2.4	0.35

Table 6-1. Latency comparison with small packets

Using 64 bytes of ICMP data (i.e., the default ping packet size), the test for each delivery method is repeated 100 times with an interval of 2 seconds.

Delivery Method	**Min (ms)**	**Max (ms)**	**Average (ms)**	**Standard Deviation**
Bi-directional Tunneling	37.4	42.4	38.4	1.2
Triangular Route	24.6	30.9	25.4	1.0
No Transparent Mobility Support	11.5	14.5	12.4	0.6

Table 6-2. Latency comparison with large packets

Using 1440 bytes of ICMP data (i.e. the whole packet is of the size of the Ethernet MTU), the test for each delivery method is repeated 100 times with an interval of 2 seconds.

We also examine the effect of multiple delivery choices on the throughput of bulk transfers. For this experiment, the mobile host is in a campus residence downloading and uploading a large file (16,576,673 bytes) from and to an FTP server on a campus cluster network. We did the transfer twice before collecting the data to warm up any caches. We repeat each test ten times for the three delivery methods.

We look at both downloading and uploading, since the routes are not always symmetric (as is the case for a triangular route). The results for downloading are shown in Table 6-3. We can potentially speed up downloading by about one third if we can avoid unnecessary transparent mobility support for traffic that does not always need it, such as web browsing. This is in addition to the fact that we also avoid generating unnecessary traffic back and forth to the home network of the mobile host.

	Conventional IP	**Triangular Route**	**Bi-directional Tunneling**
Average (seconds)	31.4	45.8	47.2
Standard Deviation	1.8	3.2	3.6

Table 6-3. Comparison of time to download a large file

The table shows the time to download a 16,576,673-byte file. We repeat each test ten times.

The results for uploading are in Table 6-4. We see only a slight benefit in using regular IP as compared to default mobile IP for uploading, since the bulk traffic from the mobile host is sent directly to the correspondent host anyway. However, when compared with using bi-directional tunneling, our mechanism provides a speedup of about one third for this experimental setup.

	Conventional IP	**Triangular Route**	**Bi-directional Tunneling**
Average (seconds)	36.4	39.8	62.5
Standard Deviation	1.7	1.8	3.9

Table 6-4. Comparison of time to upload a large file

The table shows the time to upload a 16,576,673-byte file. We repeat each test ten times.

7. Related work

Mobile IP is the context for our work. It provides support for a mobile host to be reachable by its home IP address regardless of its current point of attachment in the Internet. In this section we list some of the other projects also working in this context.

There are several projects working on extensions to the basic mobile IP protocol. Bi-directional tunneling (reverse tunneling) [12] addresses the source IP address filtering problem on security-conscious boundary routers by tunneling the packets from the mobile host through its home agent to the correspondent hosts.

Mobile host firewall traversal [7] addresses the need to deal with the intervening firewalls between a mobile host and its home agent. Currently, it is assumed that all the firewalls are in the mobile host's home domain (thus the mobile host knows the order in which they must be traversed) and the mobile host is running in the co-located care-of address mode.

Mobile IP route optimization [9] defines extensions to the operation of the base Mobile IP protocol to allow for optimization of datagram routing from a correspondent host to a mobile host. It provides a means for correspondent hosts that implement these extensions to cache the mappings from a mobile host's home address to its current care-of address and then to tunnel their own datagrams for the mobile host directly to that location. The above projects are indication that Mobile IP will be used in a variety of ways, and the goal of our work is to make this conveniently possible within one framework.

BARWAN [10] aims at building mobile information systems upon heterogeneous wireless overlay networks to support services allowing mobile applications to operate across a wide range of networks. BARWAN's vertical handoff protocol was developed to allow low-latency switching among multiple network devices, but it is dependent on the existence of base stations.

The Monarch project [8] aims to enable mobile hosts to communicate with each other and with stationary or wired hosts, transparently and adaptively making the most efficient use of the best network connectivity available to the mobile host at any time. It has an overall goal similar to ours, although the project currently does not focus on providing multiple packet delivery mechanisms.

Our previous work on Internet mobility [3] enumerates a variety of optimizations for packet delivery on mobile hosts and concludes that different optimizations are appropriate in different circumstances. The best choice depends on three factors: the characteristics the protocol should optimize, the permissiveness of the network over which the packet travels and the level of mobile awareness of the host with which the mobile host corresponds. However, this previous work did not include any implementation. We also mention the Mobile Policy Table in a previous paper [1], but it did not include support for multiple packet delivery methods. This report builds upon our previous work with new implementation techniques (like Mobile Policy Table manipulation) and also multiple interface management techniques (like backward-compatible extension to routing table) now in day-to-day use in our network.

8. Conclusions

This report proposes a general-purpose mechanism that enables mobile hosts to make decisions such as which packet delivery method to use among a number of choices (Mobile IP, to avoid transparent mobility support, bi-directional tunneling, triangular route, etc.) based on characteristics of the traffic such as destination address and TCP port number. Our performance measurements show that some traffic, such as HTTP traffic, can benefit from a mobile host's ability to choose the desired degree of mobility support.

We also address the need to use multiple network devices on a mobile host. We provide a mechanism to detect available networks. With minimal backward compatible changes to the routing table lookup function, we are able to make use of multiple networks by directing different traffic flows through different interfaces.

There are many open issues in our system that require further work, but automatic configuration is one of the most necessary. One important area is to determine dynamically which MPT policies to associate with particular destinations. A possibility is to start off by using the most conservative policies and begin probing for more efficient delivery methods from time to time by sending test packets to the destination using a more aggressive policy. For example, we could start off by using bi-directional tunneling to reach a certain correspondent host and then probe by sending triangularly routed packets to the host. If the probe succeeds, the software will update the MPT and it will continue to deliver packets using the more efficient triangular route.

We also plan to support other packet delivery methods in the MPT, such as reverse tunneling directly to the correspondent hosts instead of via the home agent. These policy decisions are dynamically installed into the MPT by probing for whether the correspondent host has the ability to decapsulate packets tunneled to it. The probe for this route optimization should be initiated by the mobile host, and can be accommodated by manipulating the policy entry in our general-purpose connectivity management mechanism.

Another important area of automatic network configuration is to extend the work on multiple device management. We are working on actively monitoring the usability and characteristics (such as bandwidth and latency) of available interfaces, as well as automatically selecting the best interface to register with the home agent.

The source code of our implementation under Linux is available at our web site: http://mosquito-net.stanford.edu/software/mip.html.

9. Acknowledgments

We thank Jim Geist for doing the initial work on detecting available networks in this report. We gratefully thank our group members including Kevin Lai, Petros Maniatis, Elliot Poger, Mema Roussopoulos, and Diane Tang for their constructive comments on earlier versions of the report. Claude Castelluccia from INRIA also read and made insightful comments on the report.

We want to thank Xerox PARC for providing financial support for Xinhua. This work was also supported in part by an NSF Faculty Career Award, a Terman Fellowship, and a grant from the Keio Research Institute at SFC, Keio University and the Information-technology Promotion Agency, Japan.

10. References

1. Mary G. Baker, Xinhua Zhao, Stuart Cheshire and Jonathan Stone, “Supporting Mobility in MosquitoNet.” Proceedings of the 1996 USENIX Technical Conference, January 1996.

2. CDPD93, “Cellular Digital Packet Data System Specification Release 1.0.” July 1993.

3. Stuart Cheshire and Mary Baker, “Internet Mobility 4x4.” *Proceedings of SIGCOMM ‘96*, August 1996.

4. Computer Emergency Response Team (CERT), “IP Spoofing Attacks and Hijacked Terminal Connections.” *CA-95:01*, January 1995.

5. R. Droms, “Dynamic Host Configuration Protocol.” *RFC 2131*, October 1993.

6. P. Ferguson and D. Senie, “Network Ingress Filtering: Defending Against IP Source Address Spoofing.” *Internet Draft (work in progress)*, March 1997.

7. V. Gupta and S. Glass, “Firewall Traversal for Mobile IP: Guidelines for Firewalls and Mobile IP Entities.” *Internet Draft (work in progress)*, March 1997.

8. David B. Johnson and David A. Maltz, “Protocols for Adaptive Wireless and Mobile Networking.” *IEEE Personal Communications, 3(1):34-42*, February 1996.

9. David B. Johnson and Charles Perkins, “Route Optimization in Mobile IP.” *Internet Draft (work in progress)*, November 1995.

10. R. H. Katz, E. A. Brewer, “The Case for Wireless Overlay Networks.” *Proceedings 1996 SPIE Conference on Multimedia and Networking, MMCM ‘96*, January 1996.

11. Metricom, “The Ricochet Wireless Network Overview.” Available HTTP: http://www.metricom./net/ricochet/netoverview.html.

12. G. Montenegro, Editor, “Reverse Tunneling for Mobile IP.” *Internet Draft (work in progress)*, March 1997.

13. C. Perkins, Editor, “IP Mobility Support.” *RFC 2002*, October 1996.

14. C. Perkins, “IP Encapsulation within IP.” *Internet Draft (work in progress)*, October 1996.

15. Charles E. Perkins and David B. Johnson, “Mobility Support in IPv6.” *Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom’96)*, November 1996.

16. Charles E. Perkins, Andrew Myles and David B. Johnson, “The Internet Mobile Host Protocol (IMHP).” *Proceedings of INET’ 94*, June 1994.

17. Charles E. Perkins and Tangirala Jagannadh, “DHCP for Mobile Networking with TCP/IP.” *IEEE ISCC’95*, Alexandria, June 1995.

18. WaveLAN, “WaveLAN Library.” Lucent Technologies. Available HTTP: http://www.wavelan.com/support/library.htm.