Secure Short-Cut Routing for Mobile IP

Trevor Blackwell, Kee Chan, Koling Chang, Thomas Charuhas, James Gwertzman, Brad Karp, H. T. Kung, W. David Li, Dong Lin, Robert Morris, Robert Polansky, Diane Tang, Cliff Young, John Zao

Division of Applied Sciences, Harvard University, Cambridge, MA 02138

Abstract

This paper describes the architecture and implementation of a mobile IP system. It allows mobile hosts to roam between cells implemented with 2-Mbps radio base stations, while maintaining Internet connectivity. The system is being developed as part of a course on wireless networks at Harvard and has been operational since March 1994.

The architecture scales well, both geographically and in the number of mobile hosts supported. It supports secure short-cut routing to mobile hosts using the existing Internet routing system without change. The implementation demonstrates a robust, low complexity realization of the architecture, and provides trade-off opportunities between efficiency and cost.

Measured performance of the mobile system is generally excellent. The system can handle a high rate of location updates, and routes packets almost as efficiently for mobile hosts as the Internet does for stationary hosts. We observe reasonable TCP behavior during hand-offs.

1. Introduction

Portable computers, while quite sophisticated in many ways, are hampered by the lack of support for mobility in current network protocols. The most immediate problem, the physical-layer link between computer and network, can be solved with radio. We can build on a long history of work in this area, such as Aloha [Ab 70], and more recently commercial radio hardware such as Altair [BuOdTaWh 91] and WaveLAN [Tu 88]. These radio systems provide limited geographical coverage. The cellular telephone system [Ma 79] solves this problem by tiling the world with radio base stations connected by a wired network. Our overall goal is to adapt this idea to computer networks.

The system we describe is the product of a graduate and undergraduate course on wireless networks taught at Harvard University in the 1993-94 academic year. The design was completed in the fall of 1993. The system has been operational since March of 1994. Our experimental environment includes IBM-compatible PCs running UNIX and 2-Mbps WaveLAN spread-spectrum radio interfaces.

The next section describes goals of our system. Section 3 compares our system with other similar work. Section 4 presents the basic architecture of our system, Section 5 explains enhancement for short-cut routing, and Section 6 gives more detail about the architecture. Section 7 analyzes the security and scalability of the system. Section 8 discusses our experimental implementation and Section 9 summarizes measured performance of the system. Section 10 suggests areas for future work. The final section gives some concluding remarks.

2. System Goals

Our primary goal is that our system be transparent to users as they roam from cell to cell. A move to another office, building, or city should not affect how a user can use network services. The user should not be required to take any special action because of such a move. All the user's existing network connections should be preserved, and there should be no difference in the way new connections are created.

Performance should approach that delivered by non-mobile protocols over the same hardware. In particular, short-cut routing should be supported. A mobile IP system should not compromise the security of communication between existing wired hosts at all, and should provide the maximum practical security for mobile hosts. Packet redirection mechanisms provided for the mobile system should not be manipulable by users to deliberately cause misdelivery of packets.

We also aim at some practical goals less visible to the user. Our system should not limit the number of active mobile hosts. No administrative domain should need to know about mobile hosts from other domains, and mobile hosts should be able to roam to other domains just as they roam within a domain. We do not require changes in IP routers or non-mobile hosts, although changes to the latter are supported to increase efficiency.

Some economic and social concerns are outside the scope of our work. We assume that different organizations are willing to provide base station service to each others' mobile hosts.

3. Background and Previous Work

A number of mobile IP systems have been implemented or proposed. All share a notion of mobile hosts (MHs), each of which keeps a constant IP address regardless of location (see Figure 1). All share the idea of radio-equipped base stations (called Foreign Agents, or FAs), which serve as temporary points of attachment to the Internet for roaming MHs. All use existing Internet routing protocols to direct packets addressed to an MH to a stationary computer (a Home Agent, or HA) capable of forwarding them to the FA to which the MH is currently attached. The fundamental differences among these systems lie in these areas:

(1) How does an HA know where an MH is?

(2) How can ordinary hosts send directly to an MH's current FA, avoiding the wasteful trip through the HA?

(3) How do the mechanisms in (1) and (2) react to MH movement?

Security, scalability, and compatibility drive the choices in these three areas. A mobile IP system should not be easily tricked into redirecting packets to malicious eavesdroppers. The MH location database should not become a bottleneck as the number of MHs grows, and thus must be distributed, perhaps at the cost of some complexity to ensure consistency. Finally, mobile hosts should be able to talk to hosts that know nothing about mobility. We call hosts that send packets to an MH Correspondent Hosts (CHs). They may be ordinary and send packets to an MHs on a dog-leg route through its HA, or enhanced to use short-cut routes direct to an MH's FA.

Below we compare some other mobile IP systems to our work. We have adopted the terminology of the IETF Mobile IP Working Group [MoIP 93], though these names (MH, FA, HA, and CH) are not universally used, nor do they correspond exactly to entities in all the systems we mention. A comprehensive comparison of several of the systems is available elsewhere [MySk 93].

3.1. Columbia's System

The central theme of the Columbia's system is the notion of a single virtual subnet to which all MHs belong [IoDuMaDe 92] [IoMa 93]. Each MH uses a radio to talk to the nearest Mobile Support Router (MSR), each of which has both a radio and a wired Internet connection. Each MSR tells the IP routing system that it has an interface onto the virtual subnet, so that normal IP routers will send packets for an MH to the nearest MSR.

The system operates as follows. An MH registers with whatever MSR happens to be in radio range, and periodically reconfirms this registration. This particular MSR thus knows where the MH is. When a CH first sends a packet to the MH, the packet is forwarded to the nearest MSR by normal IP routing. If the MH is registered with that MSR, the MSR can deliver the packet to the MH directly. If not, then the MSR must find the MH. It sends a query to all the other MSRs requesting the location of the MH, and forwards the packet to whichever MSR responds. It caches the MH's location to avoid further broadcast queries.

When the MH moves to a new MSR, it informs the previous MSR of its new location. The previous MSR will cache this information and forward any packets for the MH to its new location. If the previous MSR receives a packet forwarded by another MSR, it sends that MSR a redirect specifying the MH's new location. This redirect updates that MSR's cached location for the MH.

The Columbia system's strong points are that it sends packets by efficient routes, even from computers that are not aware of mobile hosts, and that it has no unnecessary points of failure. It does not scale well, because MSRs broadcast to each other. It does have a mode of operation with improved scaling, at the cost of inefficient routing. It has no authentication, and would be vulnerable to malicious location messages.

3.2. Sony's System

Sony's system [TeUe 93] [TeTo 93] allows both CHs and intermediate routers to cache MH locations. Every MH has a permanent Virtual IP (VIP) and a Temporary IP (TIP) address. Using the normal IP routing system, Sony's scheme arranges that a packet addressed to the VIP will end up at the MH's HA, and that a packet addressed to the TIP will end up at the MH's current location.

A mobile host is allocated a TIP each time it moves to a new location; the TIP is an address on a radio LAN at that location. The MH keeps its HA informed of its TIP. When the HA receives packets addressed to the MH's VIP, it forwards them to the MH's TIP.

When the MH sends a packet to a CH, it includes its current TIP in a special IP option [BR 89]. An enhanced CH is able to remember this TIP, and use it instead of the VIP for further communication with the MH. Packets sent to the TIP use a direct route to the MH through the Internet, avoiding the dog-leg route through the HA. Ordinary CHs ignore the option, and continue routing through the HA. When the MH moves and acquires a new TIP, it is not clear how it should notify an enhanced CH. Such a CH might continue sending to the old TIP until the MH sends it a packet containing the new TIP.

The Sony system includes routers which cache MHs' TIPs, and redirect packets sent by ordinary CHs to avoid the dog-leg through the HA. It is not clear how these caches are updated when a MH moves, especially in a network that includes ordinary routers.

The strengths of the Sony system are that it scales well and can provide efficient routing for ordinary CHs. However, its specification seems incomplete, and it provides no authentication for location updates.

3.3. IBM's System

An MH in IBM's system [RePe 92] [BhPe 93] has a permanent IP address. Each MH has an HA, and the HA tells the IP routing system that it is the gateway for its MHs. Thus when a CH sends a packet to the MH, it ends up at the HA, which will forward it to the MH. When an MH moves to a new location, it finds a nearby FA, and sends the FA's address to the MH's HA. The HA tells the MH's previous FA to forget about the MH.

When an MH sends a packet to a CH, it includes an IP Loose Source Route option [Br 89]. This option records the address of the MH's FA. The CH caches the FA address, and sends any further packets for the MH via that FA. If the MH moves, its old FA will forward packets from the CH to the MH's HA. Any reply from the MH will carry the MH's new location, allowing the CH to update its location cache.

If all Internet hosts implemented Loose Source Route correctly, IBM's system would provide efficient routing with no changes to either CHs or routers. Sadly, a dearth of correct Loose Source Route implementations thwarts this elegant system. Few systems actually remember and use the latest source route for TCP, and possibly none do so for UDP; see [MySk 93]. Source routes are not authenticated, so if implemented correctly they could be used to redirect packets arbitrarily.

3.4. Matsushita's System

Matsushita's mobile IP system [WaMa 93] [WaYoOhTa 93] is similar to the IBM scheme except in the way it provides efficient routing from CHs. When an MH moves, it acquires a temporary IP address. The MH then tries to find an FA (called a Packet Forwarding Server or PFS), and registers the FA's address with its HA (which is called the "home" PFS). The HA also receives and forwards packets sent to the MH's home address. When the HA forwards packets to the mobile host, it notifies the sending CH of the MH's current location, so the CH can then send directly to the MH.

When an MH registers a new location with the HA, the HA sends a packet to the old FA to de-register the MH and tell the old FA the mobile host's new location. If a packet for the MH arrives at the old FA, it forwards it to the MH's new location. The old FA will also inform the sending CH of the MH's new location. After a time-out period the old FA discards the new MH location, and returns MH-bound packets to the MH's HA.

The Matsushita system appears similar to our system: both include MHs, FAs, and HAs, registration at home, and support for efficient routing. However, Matsushita's Mobile IP system design does not directly address security issues or failure modes. For example, it is not clear how to perform authentication in this system, and the authors suggest repairing HA crashes by manually querying MHs for their locations. Forwarding from old PFSs to new PFSs complicates their implementation and allows forwarding loops, which must be handled specially.

3.5. Mobile IP Working Group's Proposal

This draft proposal [MoIP 93] also differs from the IBM scheme mostly in the way it provides efficient routing from CHs. Ordinary CHs always send packets to an MH via its HA. The HA, however, notices when a CH sends a packet to an MH, and notifies the CH that the MH is mobile. An enhanced CH then asks the HA for the MH's current FA, and sends further packets directly through the FA. To authenticate the HA's reply to the CH, the CH sends a random number to the HA, and the HA must supply the same number along with the MH's location. Only a router along the path between CH and HA could know this number and use

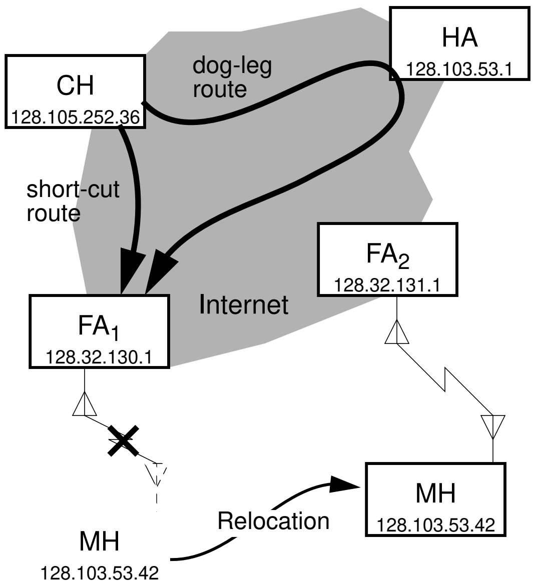

FIGURE 1. Example of mobile IP relocation, showing short-cut and dog-leg routes to the original MH location.

it to forge the HA's reply. But the CH must already trust all of those routers, since it is sending its packets through them.

The Mobile Working Group proposal is similar to our system, though they were developed independently and at roughly the same time. As a draft, it is not always complete and detailed. For instance, it is not clear how a CH determines a trustable address for an MH's HA.

A more recent draft from the Mobile Working Group [Si 94] contains more detail, but omits support for enhanced CHs. It may thus be more secure than the previous draft, and is certainly less efficient. We argue in Section 7 that enhanced CHs need not reduce security below that of the current Internet, and that therefore this omission is not necessary.

4. Basic Architecture

For explanatory purposes we consider the following scenario for our mobile IP system: a mobile host with IP address 128.103.53.42, geographically from Cambridge, Massachusetts, and under the administrative control of Harvard University, is carried to the University of California at Berkeley by its owner, Alice. Alice powers up her mobile host in Berkeley, in a wireless cell with an IP subnet number of 128.32.130. We identify the following four entities involved with providing mobile IP access to Alice's machine:

- **Mobile Host (MH)**: the portable machine with wireless network hardware carried by Alice to Berkeley. It retains the IP address 128.103.53.42 regardless of its location.
- **Home Agent (HA)**: the router at Harvard responsible for routing packets to mobile hosts with IP addresses in subnet 128.103.53. It remembers the locations of all MHs with addresses on that subnet. There is a single HA for each subnet which supports mobile hosts.
- **Foreign Agent (FA)**: the wireless base station at Berkeley that serves as the MH's temporary attachment point to the Internet. The FA has both a radio and a wired Internet connection, and is willing to forward packets between them. An FA may serve more than one MH at the same time.
- **Correspondent Host (CH)**: any host on the Internet, mobile or non-mobile, with which an MH communicates. For our example, the CH in question is in Madison, Wisconsin, with IP address 128.105.252.36.

The entities listed above are the only ones our system modifies. In particular, it uses the existing Internet routing system without any change. This is how our system behaves when the CH is not enhanced to route efficiently to mobile hosts:

- Upon arrival in Berkeley, Alice's mobile host handshakes with a nearby foreign agent. The FA arranges to route packets for the MH out its wireless interface, and the MH starts routing all its packets via the FA. The MH registers its location with its HA at Harvard, after proving its identity to the HA. The HA creates an entry in its routing table to the MH through this FA, and sets a flag indicating that packets for the MH should be encapsulated and forwarded to the FA.
- The CH in Madison sends packets to Alice's MH, at its permanent IP address. The standard Internet routing system routes these packets to the MH's HA, on the MH's home subnet. The HA looks for a route in its routing table to the MH in question, and finds the route through the FA, marked for transport by encapsulation.
- The HA encapsulates the IP packet from the Madison CH in another IP packet, and sends it to the Berkeley FA. When the FA receives this encapsulated packet, it extracts the enclosed packet, and routes it through its wireless interface to the MH.
- Alice's MH receives the packet from the FA.

- If Alice moves out of the range of the FA's radio, and into the range of another FA's radio, her MH registers the new location with its HA. The HA then starts forwarding the MH's packets via the new FA. Some packets from a CH may be forwarded by the HA to the old FA while Alice is in motion; the old FA discards them. Higher level protocols, such as TCP, should re-transmit such packets.

While the above scheme allows normal IP routing for packets from Alice's MH to Madison through the Berkeley FA and the rest of the Internet, it requires packets from the Madison CH to Alice's MH to "dog leg" through Cambridge and then double back cross-country to Berkeley. While inefficient, this routing method offers complete backward compatibility with existing Internet routers and unenhanced CH IP implementations.

The maintenance of location information for MHs by their HAs, the encapsulation of packets by a HA, and decapsulation of packets by an FA all require data structure and code modifications to the IP implementation. See Sections 6 and 8 for these and other details, such as crash recovery.

5. Enhanced Architecture for Short-cut Routing

We now present some IP enhancements made by our system that significantly improve routing efficiency from correspondent hosts to mobile hosts. Note that we maintain the invariant that existing Internet routers (those other than the foreign agent and home agent for a particular CH-MH path) require no software changes. Our goal here is to avoid the dog-leg route CH-HA-FA-MH (in our example scenario, the Madison-Cambridge-Berkeley path) in favor of the more direct CH-FA-MH (Madison to Berkeley) route for all but the first few packets from CH to MH. We modify the above behavior as follows:

- When the CH sends its first packet to the MH via the HA, the HA informs the sending CH that the MH is mobile. A non-enhanced CH ignores this notification message; such a CH continues to use dog-leg routing as outlined previously. An enhanced CH, however, asks the HA to keep it informed of the MH's location.
- The HA remembers all CHs that have subscribed to MH location updates in this way. So long as this subscription is maintained, the HA informs the CH of the MH's current FA each time the MH registers a new location.
- The CH caches the location updates from the MH's HA, installs the appropriate routes in its IP routing table (with the encapsulate flag on), and thereafter encapsulates packets bound for the MH directly to its current FA.

6. Architecture Details

We divide the architecture into four protocols: hand-off, registration, location update, and routing and encapsulation. Each of these protocols involves software that runs on more than one host; for instance, hand-off involves both FAs and MHs. The interfaces between the protocols modules on any one host are simple.

6.1. Hand-off

Each FA periodically broadcasts a beacon packet on all of its radio interfaces. If an MH is not attached to any FA and hears a beacon, it asks the FA that sent the beacon if it can attach. The FA accepts if it is not overloaded, and sends an acknowledgment. At that point the FA puts a host route for the MH in its IP routing table pointing out the radio interface, and the MH installs a default route pointing to the FA.

The MH monitors the beacons from its current FA; if it does not hear a beacon for a while, it scans for other FAs. The frequency with which FAs broadcast beacons governs how soon an MH notices that it is out of range of its current FA, and therefore how long its service will be interrupted before it acquires a new FA.

The MH periodically tells its FA that it still wants service. If the FA does not hear from the MH for a while, it deletes the route to the MH. If the FA receives an encapsulated packet for an MH for which it has no route, it silently discards the packet.

The FA provides service for an MH without any sort of authentication. This allows an unauthorized MH to send packets into the Internet via the FA, but it does not allow the MH to receive packets unless they are specifically encapsulated and sent via the FA. The only way to arrange for the MH to receive packets addressed to its IP address is by authenticated HA registration.

6.2. Registration

After the MH establishes a connection with an FA, it sends its HA a registration request. This request contains the MH's IP address and its FA's IP address. The HA replies with a randomly chosen challenge number. The MH signs the challenge along with the FA address using MD5 [Ri 92] and a secret key shared with the HA, and sends this signature back to the HA. Upon validating the signature, the HA updates its routing table for this MH and sends back an acknowledgment. If the authentication fails, the HA replies with a denial packet. The MH must periodically re-register with its HA, in case the HA reboots and thus forgets the locations of its MHs.

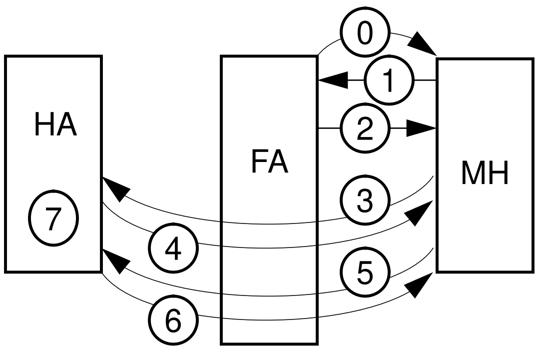

FIGURE 2. MH Hand-off and Registration. The FA periodically broadcasts beacons (0). The MH replies with an attachment request (1); the FA responds with an attachment grant (2). After attaching, the MH sends a registration request (3) to its HA. The HA replies with a challenge packet (4) to the MH. The MH sends a signed reply (5). If the reply is good, the HA sends a registration confirmation packet (6). An HA function call tells the update layer of the new MH location (7); this triggers location updates to subscribed CHs.

The HA chooses a new challenge number for an MH each time the MH registers successfully. The challenge prevents replay attacks. It also functions like a sequence number, to help the MH and HA ignore all the but the latest messages. It is especially useful when the MH changes location frequently. The MH could save one packet exchange with the HA by sending a non-repeating sequence number, rather than waiting for a challenge; we decided it would be too hard to keep the sequence numbers on the MH and HA consistent.

The HA requires stable storage to hold one registration key for each MH it serves. The key management between the HA and his MHs is straightforward as they are assumed to be under the same administrative authority.

6.3. CH Update

As described above in Section 5, the HA directly informs any CHs using dog-leg routing that the destination MH is mobile. The HA can detect when a CH talks to an MH because the Internet routes the CH's packets via the HA. An HA limits the rate at which it notifies any one CH that an MH is mobile, since unenhanced CHs will never stop sending via the HA.

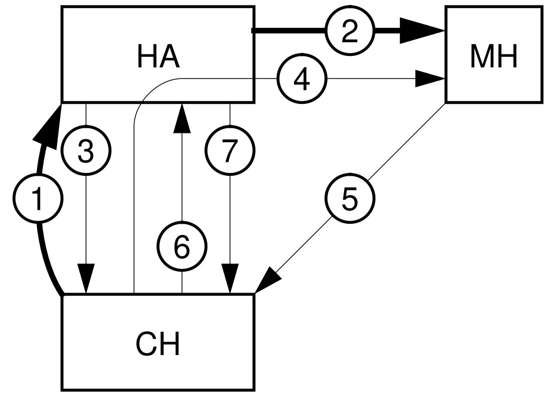

FIGURE 3. CH acquisition of direct route (FA acts only as a bridge to the MH, so it is omitted). (1) normally routed packet intercepted and (2) forwarded to MH triggers a notification message (3) to the CH. The CH asks the mobile host to name its home agent (4); after receiving the reply (5), it sends a subscription request (6) to the HA. The HA replies with a location update (7), which is then installed in the CH routing table.

In a perfect world, the HA could use one message both to inform the CH that a host is mobile and to carry the forwarding address information. Unfortunately, the CH cannot trust the contents of the notification message without creating a redirection security loophole. First, it must determine the correct address of the HA of the MH mentioned in the notification. It does this by sending a query to the MH containing a random number; the MH replies with the random number and its HA's address. The random number assures the CH that the response must have come either from the MH or from some router along the path between CH and MH. The CH must trust all routers along this path, since it sends its data through them.

After the CH has discovered the MH's HA, it sends a subscription request to the HA. The HA replies with the address of the MH's current FA. The subscription request and reply are also protected by a random number.

The relationship between HA and CH takes place under a *subscription* model. The HA remembers all of the CHs that have recently placed subscription requests. If the MH changes location, it notifies all subscribers of the new location. If a *subscription lapse time* (SLT) passes without receiving a subscription request from a particular CH, then the HA assumes that the CH no longer wishes to receive location updates. The CH must periodically resubscribe to the HA's location update service in order to continue to receive updates. CHs determine whether a "conversation" with a particular MH is still active by checking the packet counter in the kernel routing table.

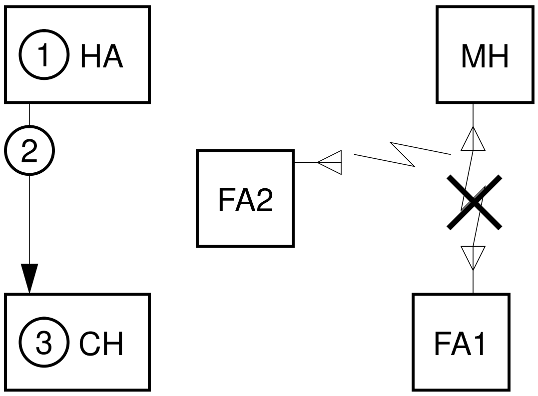

FIGURE 4. Location updates when the MH moves. (1) Function call from registration layer triggers (2) a new location message from the HA to all subscribed CHs. On receipt, a CH installs the new MH location into its route table (3).

If the CH reboots, it will begin using dog-leg routes again. The HA still sends notification messages, even for supposedly subscribed CHs, so the CH will go through the normal location update process.

If the HA reboots, it will forget all current subscribers. CHs periodically re-subscribe to help recover from such reboots. A CH removes the route to an MH if its HA fails to reply after a time-out period. This prevents permanent misdirection by a fake HA which can respond to only a finite number of the CH's subscription requests.

The MH could implement the update protocol instead of the HA. We did not choose this approach because we expect some MH operating systems will not support monitoring of packets from CHs.

6.4. Routing and Encapsulation

Both CHs and HAs need to send packets to MHs by way of FAs. They cannot directly use the regular IP routing system, since it would send packets with an MH's address to its HA. We use a simple encapsulation scheme for this, in which an IP packet for an MH is placed inside another packet, with a special IP protocol number, addressed to an FA. A flag in each routing table entry controls encapsulation. A CH encapsulates only packets that it originates, and that it knows it is sending to an MH. An HA acts as an IP router, receiving packets from CHs that don't know an MH is mobile, encapsulating them, and forwarding them to the MH's FA.

An FA knows when it has received an encapsulated packet by looking at the IP protocol number. It strips off the outer header, and processes the packet inside almost as if it had been received in the normal way. The difference is that the FA discards the encapsulated packet if it is not addressed to an MH currently attached to the FA. This prevents routing loops.

When an MH moves, its old FA could forward packets to its new FA, rather than dropping them. This might eliminate a few lost packets. However, it is unlikely that this would eliminate all loss; for TCP, at least, a few dropped packet are very little better than many consecutive drops. In addition, it would be difficult for the old FA to authenticate the location updates that the MH would have to send it.

7. Analysis

7.1. Security

Although mobile hosts introduce some new security concerns, the fact that radio communication is easy to intercept, disrupt, and forge is not one of them. Much of the current wired Internet uses media with the same problems. Since these issues are not special to mobility, we do not attempt to address them. Systems such as Kerberos [StNeSc 88] and Privacy-Enhanced Mail [Li 89] can solve some of these problems by providing privacy and authentication between applications at either end of the network. Our aim is to maintain the Internet's current level of security for existing applications, and to help prevent denial-of-service attacks on all applications, even those with end-to-end security.

One attack we face involves a fake MH trying to register under another MH's address; the other is a fake HA sending location update messages to a CH by spoofing messages from a real HA. The second problem is particularly serious since CHs do not know which hosts are mobile; thus the same attack could be used to divert traffic from a wired host.

Security on the wired Internet is a function of what hosts are along the path between sender and receiver. More formally, if hosts A and B are communicating then a BE (Bad Element, a network-connected computer under the control of a malicious user) along the path between A and B can read all packets from A to B, forge packets from A to B, and cause packets not to be delivered to B. A host not along the path, however, cannot read packets from A to B.

Although any host can forge a source address in the IP header, that is usually not sufficient to carry on an entire fraudulent conversation; the forger must usually see the replies to his messages to do any real harm. For instance, to send packets as part of a TCP session, the sender must use the right sender sequence number or the receiver will ignore the packets. These sequence numbers are allocated at connection setup time, using a random number generator, so that it is difficult for an attacker to guess a valid sequence number[1]. Much of the security of our system is based on the assumption that attackers cannot see packets between the HA and CH for an indefinitely long period—such attackers would be able to intercept traffic directly without bothering to attack our system.

7.1.1 Security of MH-HA Registration

Our authentication scheme is implemented by two protocols. The MH-HA registration protocol authenticates the MH's identity and location to the HA. The HA-CH update protocol allows a CH to verify that it is receiving location updates from an MH's HA.

The registration message from the MH to the HA, containing the IP address of the MH's current FA, must be signed by the MH to prevent impersonation of the MH by BEs. To accomplish this, we use an MD5 signature to guarantee the authenticity of registration messages. It is unlikely that a BE could create a false registration message that the HA would accept. Replay attacks are prevented by the use of a randomly chosen challenge, which is different for each registration. Although the true MH can effectively be denied service by interception of the registration message, this is an unavoidable characteristic of any Internet connection.

We chose MD5 over some other available signature algorithms because it does not cause any interoperability problem with foreign hosts due to export restrictions. The MH and its HA must share a key which is added to the message when computing the MD5 hash, but not actually sent over the network. We generate the key and store it on both machines when the MH is first configured.

1. In fact, many Berkeley-derived TCP implementations use an easy-to-predict sequence number generator, but this should be considered broken.

7.1.2 Security of CH Location Update

Packet redirection in order to avoid dog-leg routing creates a potential security hole. A Bad Element who can forge location update messages from the HA to the CH can cause all traffic destined for an MH to be redirected to it or any other destination.

We use *tickets* to enforce the property that although BEs anywhere may be able to forge *packets* from the HA to the CH, a host can only send *valid* location update messages to the CH if it can see packets from the CH to the MH's subnet. This general security strategy is prevalent in the Internet; it is how NFS, TCP, X11's magic cookie system, and DNS achieve their security [Su 88][Ny 92][Mo 87]. No administration is necessary for this security system, and processing overhead is very small.

The CH sends the HA a subscription request asking to receive location updates for an MH, containing a ticket consisting of some randomly generated bytes X_a. Hosts not along the path between the CH and HA will not see X_a. When sending location updates, the HA includes the most recent X_a it received from that CH. The CH will only accept updates accompanied by the X_a it generated. X_a is chosen from a range large enough (2^{128} in our implementation) that a BE is unlikely to guess a valid X_a. Thus, BEs not on the path between the HA and CH cannot fool the CH into redirecting packets for a mobile host.

Recall from Section 6.3 that the CH *periodically* sends subscription requests to the HA and updates its route to the MH based on the HA's reply. Thus, even if a BE on the CH-HA path is able to see the most recent X_a, the BE can only fool the CH for a limited time until the CH's next subscription takes effect. To arrange for permanent misdirection of packets from the CH to a MH, the BE would have to be able to spoof packets on the CH-HA *continuously.* In this case, the BE would have been able to steal ordinary data packets from the CH to the MH in the first place, even without using our mobile IP system.

7.2. Scalability

We may divide the knowledge that entities in our system have about other entities into two categories:

- Static, administrative knowledge: MHs and their HA are presumed to be under control of the same administrative authority. It is assumed in our system that MHs know their home HA and that HAs know the MHs for which they route packets. A

HA and its MHs share keys for authentication purposes. This information is static in nature, and is maintained by system administrator action.

- Dynamic, online knowledge: The entities which exchange location information consist only of the MH, HA, and CH (while the FA passes along location information, it does not produce or consume it). This location information is dynamic in nature, and is automatically maintained by our system through registration and location update messages.

The limited number of parties who require knowledge in both of the above categories is a strong asset of our scheme; no entity in an administrative domain needs administrative knowledge about entities outside its domain, and no entity needs online information about any entity other than those with which it is currently communicating. This fact makes our system fundamentally scalable to a large number of entities.

The location update messages exchanged when an MH moves flow only between entities involved in communication with the MH. As shown in Section 9, the registration procedure of our system is reasonably fast.

Large numbers of MHs can be accommodated by increasing the number of HAs and home subnets; this policy for network expansion is identical to that in practice today for wired networks.

8. Implementation Notes

We have a working implementation of the system described in this paper.

Because our system involves changes to IP, we need an operating system for which we can obtain source code. We use Berkeley Software Design's UNIX (BSDI) for IBM-PC compatible computers, which includes source.

Another reason we use BSDI UNIX is that it supports the Berkeley Packet Filter [McJa 93], which can give a copy of every packet received by the system to a process. The HA software uses this to detect when a new CH starts sending packets to an MH.

Our MD5 implementation comes from the RSAREF library available from RSA Data Security, Inc.

We use WaveLAN radio interfaces [Tu 88]. WaveLAN uses spread spectrum modulation to avoid interference between nearby radios that do not wish to communicate. Radios that do wish to talk must be set to the same "code." If multiple nearby WaveLANs are set to the same code, they can communicate peer-to-peer, though we do not currently use this feature. WaveLAN has a range of a few hundred feet and provides about 2 megabits of bandwidth per second. It uses the same frame and address format as Ethernet, and uses CSMA/CA for medium access control. The WaveLAN interfaces fit in an ISA slot in a PC. We do not currently have a truly portable radio interface using, e.g., PCMCIA, due to the difficulty of obtaining UNIX drivers for them.

8.1. Kernel Changes

Four UNIX kernel changes are needed to support the system. BSDI already allows two hosts with different IP network numbers to talk to each other over the same physical network; this situation arises when an MH talks to an FA. The only problem is with broadcasting beacon packets. The only universally acceptable IP broadcast address has all bits set. However, UNIX cannot determine on which network interface to send such a packet. We added a socket option to specify the routing table entry to be used when sending packets from a socket; in this case we would specify a route pointing to the desired interface.

We added encapsulation code to the kernel, controlled by a flag in each routing table entry. If the flag is set for the route a packet would use, the packet is encapsulated by adding a new IP header with a special protocol number. The encapsulating packet is addressed to the destination in the gateway field of the routing table entry, and is then routed in the usual way. A host knows it has received an encapsulated packet by the IP protocol number; the host strips off the encapsulating header, and processes the inner packet as if it had been received in the usual way.

This encapsulation mechanism suffices in a CH, and in an HA after an MH has registered. However, before an MH has been authenticated, its HA still needs to send it encapsulated packets. It cannot create a routing table entry for a potentially fake MH because that would divert packets away from the real MH. So we use the per-socket routing option described above during registration.

To prevent packets to un-registered MHs from being forwarded by the HA using the usual IP routing system, we added a special network interface that discards packets. The HA configures that interface with the network number used by the MHs it manages. The host routes installed for each registered MH override use of this interface. We do this in preference to giving the HA a real radio interface for its MHs so that packets for un-registered MHs are not broadcast to anyone listening to the radio. The presence of this special interface also causes the UNIX routing daemons to announce the MHs' network number to the Internet routing system.

UNIX caches a route for every socket, which it keeps using until the route is deleted from the routing table. When a CH first receives an MH's location from an HA and installs a route for the MH, the routes cached by any sockets already connected to the MH are not affected. So while new sockets connected to the MH will use the efficient route, the first socket to send to an MH will continue to send via the HA. We have partially fixed this problem, but the UNIX IP code does not make a clean and complete solution easy.

8.2. Software Structure

Most of the software in our system runs as daemon processes, with a different type of daemon for each of the four entities (MH, FA, HA and CH). The daemons communicate across the network with UDP.

We designed and partitioned the system to make it easy for a group of students to implement as independent modules. This has worked well in most cases. For instance, we require one process to run on the MH, which combines modules for hand-off and HA registration. The interaction between them is limited to a function call made by the hand-off module to tell the registration module the IP address of the MH's current FA. The modules at both ends of each protocol, such as MH/HA registration, were implemented by the same group.

In some cases this modularity works badly. One might want to make a single computer an HA, a CH, and an FA. The three modules cannot just be executed on the same computer. All three modify the routing table, and the modifications may conflict. Worse, the FA adds routes for MHs without any authentication. Usually this is not harmful, since an MH still has to register with its HA to receive any packets. But if the FA is also the MH's HA, the MH will receive packets without registration because of the route added by the FA. We could solve this by tighter integration of the FA and HA modules.

A class of a dozen students implemented this system in a month of programming.

9. Measured Performance

We have measured performance of our mobile IP system in three areas: TCP throughput, TCP delay during hand-off, and registration speed. The computers involved in our experiments were 66 MHz 80486 PCs. The HA, FA, and CH were connected by a single isolated Ethernet segment, with no other traffic.

Table 1 shows TCP throughput over three routes. The short-cut route performs significantly better than the dog-leg route, and approaches the performance observed on the radio link alone.

	TCP Throughput
Dog-leg Route CH->HA->FA->MH	1.1 Mbps
Short-cut Route CH->FA->MH	1.3 Mbps
Route over Radio Link Only MH -> FA or FA -> MH	1.3 Mbps

TABLE 1. TCP throughput comparisons.

Table 2 depicts the impact of hand-off time on TCP delay. Even if an MH moves between FAs with overlapping radio ranges, there will be some amount of time during which packets sent by a CH to the MH will not be delivered. This includes time for the MH to realize it has lost contact with the old FA, for the MH to scan for a new FA and attach to it, for the MH to register with the HA, and for the HA to send a location update to the CH. Some packets will be lost during this time, and must be retransmitted after an additional time-out interval by higher protocol layers such as TCP. Previous work [CaIf 93] has suggested that short hand-off times can result in disproportionately long interruptions in TCP traffic. Our experiments, summarized in Table 2, indicate that the expected interruption in service on a TCP connection is little more than twice the dead time. This is consistent with the fact that TCP doubles its retransmission time-out on each consecutive failed retransmission. Some of TCP's behavior shown in the table is due to its minimum retransmission time-out of one second and timer granularity of half a second.

Hand-off Time (Seconds)	TCP Delay (Seconds)
0.3	1.2
1.0	1.2
1.6	4.4
2.8	4.4
3.4	4.3

TABLE 2. TCP delay as a function of hand-off time.

Table 3 depicts the results of some stress tests measuring the speed of the registration process. A registration takes no more than 20 milliseconds elapsed time. This is equally divided between CPU time and transmission time.

	# Registrations per Second
MH registration to HA without CH subscribing	56
MH registration to HA with one CH subscribing	54

TABLE 3. Registration speed.

10. Future Work

Areas that warrant further investigation include improving the security of location update messages, optimizing hand-off for special cases, and load balancing for FAs in overlapping cells. We briefly explain two of these areas.

As explained earlier, our location update messages from the HA to the CH are vulnerable to spoofing and replay by persistent malicious hosts along the path between the HA and the CH. We could use digital signatures to provide better security for these updates. This would require a key and certificate management, storage, and distribution architecture to guarantee that CHs verify signatures with the correct keys.

Hand-off in our system is not particularly fast, as it requires the mobile hosts to scan channels listening for beacons from foreign agents. More coordination among FAs and MHs might allow them to locate each other faster.

11. Conclusions

The existing IP routing system makes no provision for mobile hosts; it cannot react to rapid changes in network topology, and its global knowledge of topology cannot scale to the size required to track individual hosts. We have presented the architecture and implementation of a solution to this problem. It makes use of IP routing and the Internet infrastructure without modification. It maintains a database of mobile host locations, partitioned in a way that allows scaling. It is backward-compatible with existing hosts, but gives the option of increasing routing efficiency by adding short-cut routing to host IP software. Neither the location database nor the host IP modifications decrease security below the level provided by today's Internet.

Our system turns out to be quite close to the overall direction outlined in a recent draft [MoIP 93] of the IETF Mobile Working Group. It appears that we have one of the first working implementations of the architectural approach being pursued by the Group. Our implementation demonstrates the practicality of the approach, including secure short-cut routing.

Acknowledgments

Digital Equipment Corporation lent us WaveLAN interfaces and provided equipment discounts for the project's PCs. Anders Klemets wrote the original version of the WaveLAN driver while at CMU. NCR has allowed others to use this driver. Motorola gave us Altair radio network units. Bell-Northern Research supported our efforts via unrestricted research grants. Hewlett-Packard and IBM gave us laptops. We thank them for making our work possible.

References

[Ab 70] N. Abramson, "The ALOHA System - Another Alternative for Computer Communications." *Proceedings, Fall Joint Computer Conference,* 1970.

[BhPe 93] Pravin Bhagwat & Charles Perkins, "A Mobile Network System based on Internet Protocol (IP)", USENIX Symposium on Mobile and Location Independent Computing, Aug, 1993, Cambridge, MA, 69-82.

[Br 89] R. Braden, "Requirements for Internet Hosts - Communication Layers", RFC 1122, Oct 1989.

[BuOdTaWh 91] Dale Buchholz, Paul Odlyzko, Mark Taylor, and Richard White, "Wireless In-Building Network Architecture and Protocols," IEEE Network Magazine, Nov. 1991.

[CaIf 93] Ramon Caceres & Liviu Iftode, "Effects of Mobility on Reliable Transport Protocols," Matsushita Information Technology Labs, TR73-93.

[IoDuMaDe 92] John Ioannidis, Daniel Duchamp, Gerald Maguire, & Steve Deering, "Protocols for Supporting Mobile IP Hosts", Internet Draft, June 1992.

[IoMa 93] John Ioannidis & Gerald Maguire, "The Design and Implementation of a Mobile Internetworking Architecture", Proc. of Winter USENIX, Jan 1992, San Diego, CA, p. 491-502.

[Li 89] J. Linn, “Privacy Enhancement for Internet Electronic Mail, Parts I, II, and III,” RFC 1113-1115, January 1989.

[Ma 79] V. H. MacDonald, “The Cellular Concept,” Bell System Technical Journal, vol. 58 no. 1, part 3, Jan. 1979.

[McJa 93] Steven McCanne and Van Jacobson, “The BSD Packet Filter: A New Architecture for User-level Packet Capture,” Winter 1993 USENIX Conference.

[Mo 87] P. V. Mockapetris, “Domain Names: Implementation and Specification,” RFC 1035, 1987.

[MoIP 93] Mobile IP Working Group, “Routing Support for IP Mobile Hosts”, Internet Draft, Dec 1993.

[MySk 93] Andrew Myles & David Skellern, “Comparison of Mobile Host Protocols for IP”, available from authors. (andrewm@mpce.mq.edu.au)

[Ny 92] A. Nye, ed., “The X Window System, Volume 0: X Protocol Reference Manual, Third Edition,” O’Reilly and Associates, Sebastopol, California, 1992.

[RePe 92] Yakov Rekhter & Charles Perkins, “Optimal Routing for Mobile Hosts using IP’s Loose Source Route Option”, Internet Draft, Oct 1992.

[Ri 92] R. L. Rivest, “The MD5 Message-Digest Algorithm,” RFC 1321, April 1992.

[Si 94] W. A. Simpson (ed.), “IP Mobility Support,” IETF Internet Draft, March 1994.

[StNeSc 88] J. Steiner, C. Neuman, and J. I. Schiller, “Kerberos: An Authentication Service for Open Network Systems,” in *Proc. Winter USENIX Conference*, Dallas, 1988.

[Su 88] Sun Microsystems, “NFS: Network File System Protocol Specification,” RFC 1094, March, 1988.

[TeTo 93] Fumio Teraoka & Mario Tokoro, “Host Migration Transparency in IP Networks: The VIP Approach”, Comp. Comm. Rev., ACM, Jan 1993, p. 45-65.

[TeUe 93] Fumio Teraoka & Keisuke Uehara, “The Virtual Network Protocol for Host Mobility”, Internet Draft, July 1993.

[Tu 88] Bruch Tuch, “Development of WaveLAN, an ISM Band Wireless LAN,” AT&T Technical Journal, July/August 1993.

[WaYoOhTa 93] Hiromi Wada, Takashi Yozawa, Tatsuya Ohnishi, & Yasunori Tanaka, “Mobile Computing Environment Based on Internet Packet Forwarding”, Proc. of Winter USENIX, Jan 1993, San Diego, CA, p. 503-517.

[WaMa 93] Hiromi Wada & Brian Marsh, “Packet Forwarding for Mobile Hosts”, Internet Draft, July 1993.

Author Information

James Gwertzman and Diane Tang are undergraduates at Harvard.

Trevor Blackwell, Kee Chan, Koling Chang, Thomas Charuhas, Brad Karp, W. David Li, Dong Lin, Robert Morris, Robert Polansky, Cliff Young and John Zao are graduate students at Harvard.

H. T. Kung is Gordon McKay Professor of Electrical Engineering and Computer Science at Harvard. He is the instructor of the course (CS96) on wireless networks that has developed the system described herein.

The postscript file of the current version of this paper is available via anonymous FTP from virtual.harvard.edu:/pub/cs96/usenix94.ps.