A Mobile Host Protocol Supporting Route Optimisation and Authentication

Andrew Myles
Department of Electronics
Macquarie University 2109
Sydney, NSW, Australia
andrewm@mpce.mq.edu.au

David B. Johnson
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891, USA
dbj@cs.cmu.edu

Charles Perkins
T. J. Watson Research Center
IBM Corporation
30 Saw Mill River Road
Hawthorne, NY 10532, USA
perk@watson.ibm.com

Abstract

Host mobility is becoming an important issue due to the recent proliferation of notebook and palmtop computers, the development of wireless network interfaces, and the growth in global internetworking. This paper describes the design and implementation of a mobile host protocol, called the Internet Mobile Host Protocol (IMHP), that is compatible with the TCP/IP protocol suite, and allows a mobile host to move around the Internet without changing its identity. In particular, IMHP provides host mobility over both the local and wide area, while remaining transparent to the user and to other hosts communicating with the mobile host.

IMHP features route optimisation and integrated authentication of all management packets. Route optimisation allows a node to cache the location of a mobile host and to send future packets directly to that mobile host. By authenticating all management packets, IMHP guards against possible attacks on packet routing to mobile hosts, including the interception or redirection of arbitrary packets within the network. A simple new authentication mechanism is introduced that preserves the level of security found in the Internet today, while accommodating the transition to stronger authentication based on public key cryptography or shared keys that may either be manually administered or provided by a future Internet key management protocol.

1. Introduction

In recent years, computers have become increasingly compact and very powerful. At the same time, connectivity to the global network is becoming widespread, and with the recent introductions of commercial wireless network interfaces, users now have a real opportunity for continuous network connectivity wherever they may happen to be working.

Unfortunately, existing internetwork protocols do not easily accommodate mobile hosts. Host movement today, even if only between local subnets, involves slow, manual, error prone host and network reconfiguration procedures that a typical user does not have the skills or desire to carry out. Moreover, even when this movement process is successfully performed, the mobile host loses its former identity in terms of its host network address, and any network applications on the mobile host and on other hosts communicating with it must usually be restarted.

A number of *mobile host protocol* proposals that are compatible with the TCP/IP protocol suite have been proposed [1-9, 16, 19-20]. These proposals each retain the home IP address of a mobile host for use in identifying it at the network level, but also in some way associate a second IP address with the mobile host to indicate the mobile host's current location. Each of the proposals has a number of advantages and disadvantages, some of which are discussed in [10, 11]. Proposals have also been made for supporting host mobility in an OSI environment [12].

Many of these mobile host protocols provide some form of route optimisation that allows other nodes (hosts or routers) to learn the current location of a mobile host, either from management packets or from an IP option attached to data packets. Nodes learning the location of a mobile host in this way can cache this location and then send later packets for the mobile host directly to that location. However, none of these proposals (except [6, 7]) provide a mechanism for the node learning a mobile host's current location to authenticate it. The result is that a malicious host anywhere in the Internet could easily send forged management or data packets in order to intercept or redirect packets destined to a mobile host. In today's Internet, only hosts connected to the normal packet routing path can cause similar disruption [18].

The only previous mobile host protocol proposal that provides for extensions to guard against this type of attack [6, 7] compromises its wide area operation to maintain its

authentication mechanisms. The protocol does not support route optimisation in the wide area, but rather normally forces all packets addressed to a mobile host connected away from its home network to pass through the mobile host's home network before being forwarded to the mobile host at its current location. This non-optimal routing results in reduced *performance transparency* to the user as a result of increased network overhead for packets sent to the mobile host.

This paper describes the design and implementation of a new mobile host protocol, called the Internet Mobile Host Protocol (IMHP), that features both route optimisation and integrated authentication of all management packets. IMHP operates equally well in both the local and the wide area, and provides *performance transparency* and *operational transparency* to the user. IMHP introduces an optional new mechanism for providing simple authentication of management packets that preserves the level of security found in today's Internet [18], and is designed to accommodate stronger authentication based on public key cryptography or on shared keys that may either be manually administered or provided by a future Internet key management protocol. IMHP thus provides effective authentication today, while providing a good migration path to stronger authentication when available. A more detailed specification of IMHP is found in [15].

Section 2 of this paper describes the necessary IMHP infrastructure. Section 3 discusses the IMHP authentication procedures and their operation in the protocol. In Section 4, the rules used by each node when forwarding packets are presented, and in Section 5, the means by which nodes learn and cache the location of mobile hosts are described. Section 6 discusses additional features of the protocol, and Section 7 details a number of examples of the operation of the protocol. Section 8 describes an implementation of IMHP, and Section 9 presents conclusions.

2. IMHP Infrastructure

The IMHP architecture includes four functional entities: *mobile hosts*, *local agents*, *cache agents*, and *home agents*. This section defines each entity and describes its basic operation. Although defined separately, the functionality of several of these entities may be combined within a single node.

Mobile Host

A *mobile host* is a normal host with additional software that allows it to move through the network in a manner transparent to the user and to software above the network routing layer within the host. A mobile host is assigned a constant, unique *home address* that belongs to a *home network* in the same way as any other host. *Correspondent hosts* (either mobile or stationary) use the home address of a mobile host in sending packets to the mobile host regardless of the mobil host's current location.

Local Agent

When a mobile host connects to the network, it must be able to determine that it has moved to a new network and must identify a *local agent* connected to the new local network with which to register. The first function may be performed with network data link layer support, if available, or may use an *advertisement and solicitation protocol* that is defined by IMHP. Registration is performed using a *registration protocol* that is defined by IMHP.

Each local agent maintains a *visitor list* identifying all mobile hosts currently registered with this local agent. A local agent uses the visitor list to forward packets it receives addressed to these mobile hosts to the local network to which the mobile host is connected. A local agent times out the visitor list entry for a mobile host after a lifetime period negotiated with the mobile host during the registration process; once a visitor list entry times out, it is deleted by the local agent. In order to maintain uninterrupted service from its current local agent, a mobile host must re-register with its local agent within this lifetime period.

A local agent, during the registration process, provides the mobile host with a *care-of address*, which is generally the local agent's own address, that defines the location of the mobile host. The combination of a mobile host's home address and care-of address is known as a *binding*. If the care-of address and home address elements of a binding are the same, then the mobile host is assumed to be connected to its home network.

Whenever a mobile host registers with a local agent, the mobile host must arrange to reliably notify any previous local agents that might still have a visitor list entry for it that this mobile host has moved. Each previous local agent uses this notification to delete any visitor list entry held for the mobile host, ensuring that the local agent does not continue to forward packets to a local network when the mobile host has moved elsewhere in the network. The notification to a previous local agent must be periodically retransmitted (with

a back-off mechanism) either until it is acknowledged or until the previous local agent would have timed out the visitor list entry it held for the mobile host.

A mobile host will typically use the local agent with which it is currently registered as a default router. However, if the mobile host is connected to a local network, such as its home network, for which it is able to obtain better routing information, it may use any local router.

Cache Agent

A *cache agent* is the functionality within any node that maintains a *location cache* containing the binding of one or more mobile hosts that it has learned through IMHP's *binding management protocol*. When sending any packet, if a cache agent has a binding in its location cache for the destination address of the packet, the cache agent routes the packet directly to that mobile host at its current location by *tunnelling* the packet to the mobile host's care-of address as indicated in the cached binding. Otherwise, the cache agent sends the packet using normal Internet routing, causing the packet to be delivered eventually to the mobile host's home network.

Although in principle, any tunnelling protocol could be used, an IMHP tunnelling protocol has been designed to minimise the processing and space overhead added to each packet tunnelled. To tunnel a packet, a small IMHP tunnelling header is added to the packet between the packet's IP header and any transport-level header in the packet, such as TCP or UDP, as illustrated in Figure 1. The IP header of the packet is modified so that the packet appears to be a normal IP packet addressed from the cache agent to the mobile host's current local agent, and the original values of the modified IP header fields are copied into the new IMHP tunnelling header. The packet then uses only normal IP routing to reach the local agent, which removes the added header and restores the packet's original IP header before delivering the packet to the mobile host. The IMHP tunnelling protocol adds only 8 or 12 bytes of overhead to each packet being tunnelled.

A cache agent times out a location cache entry and deletes it after a lifetime specified by the binding management protocol when the location cache entry is established or according to a local cache use policy. If a cache agent wants to provide continued service for packets addressed to a particular mobile host, it may attempt to reconfirm the mobile host's binding and thus update the corresponding location cache entry before the location cache entry times out. A cache agent also deletes a location cache entry if it receives a new binding for that mobile host indicating that its care-of address is the same as the mobile host's home

address, meaning that it is connected normally to its home network. Such a location cache entry need not be stored, since this is the default routing for packets for that mobile host in the absence of any binding.

Any node that wants to optimise its own communication with mobile hosts should function as a cache agent, allowing it to route packets directly to each correspondent mobile host's current location. Many local agents will also be capable of functioning as cache agents. If such a local agent is notified that a mobile host it previously served has moved, then the local agent may, subject to certain authentication related restrictions discussed later, create a location cache entry for the mobile host indicating the new binding, after deleting the corresponding visitor list entry. This feature ensures fast redirection of packets to the mobile host's current location when a mobile host moves.

Home Agent

Each mobile host must have a *home agent* that is attached to its home network. A home agent maintains a *home list* identifying all mobile hosts that it is configured to serve. The home agent must also serve as a cache agent for at least these mobile hosts. It may serve as a local agent for these or other mobile hosts as well.

When registering with a new local agent, a mobile host must also register with its home agent, so that the home agent always knows the current binding of the mobile hosts it serves. The home agent normally creates a location cache entry for the mobile host, with a binding indicating the mobile host’s current care-of address given in the registration. However, if the mobile host’s home agent is also functioning as a local agent, the mobile host may register directly with its home agent. In this case, if the home agent is a router connected to more than one network, and if the mobile host is registering on a network other than the mobile host’s home network, then the home agent (as a local agent) creates only a visitor list entry for the mobile host. On the other hand, if the mobile host is registering on its home network with its home agent, then no location cache entry or visitor list entry is created; the mobile host is then said to be *at home*.

Any location cache entry or visitor list entry that is created is timed out after a lifetime negotiated by the mobile host and its home agent. The mobile host thus must re-register with its home agent before the lifetime expires if it wants to maintain continued service from its home agent. Typically, the home agent registration lifetime will be greater than the local agent registration lifetime, and so fewer re-registrations are required with the home agent than with the local agent. The home agent assumes that the mobile host is at home if

it does not have a valid binding for the mobile host. This will happen if either its visitor list entry or location cache entry for the mobile host times out before the mobile host re-registers.

If a mobile host is registered away from home, then its home agent must arrange to intercept (for example, through proxy ARP [14]) any packets on the home network that are addressed to the mobile host's home address, including packets that have been forwarded to the home network from elsewhere in the network using normal routing algorithms. If the mobile host is registered directly with its home agent (as a local agent) on a local network other than its home network (the home agent has a visitor list entry for the mobile host), then the home agent delivers each intercepted packet to the mobile host on the local network indicated by the visitor list entry. Otherwise, the home agent tunnels each intercepted packet to the mobile host's current care-of address using the location cache entry it holds (as a cache agent) for the mobile host.

3. Authentication

A limitation of previous mobile host protocol proposals is that they do not provide a method for nodes to authenticate a binding that they receive for a mobile host. Without authentication, providing route optimisation exposes the network to significant security risks. A malicious node could send forged management packets, giving incorrect information on a mobile host's location, and could thus misdirect or intercept packets addressed to that mobile host. The only alternative to this risk among previous mobile host protocols [6, 7] has been to force all packets addressed to a mobile host to be routed through the mobile host's home network. Such an alternative reduces performance transparency and places additional overhead on the network.

Adding authentication features to a mobile host protocol supporting route optimisation is complicated by the need for any node to be able to authenticate a binding received for any mobile host. In general, such authentication requires a key distribution infrastructure which, in the Internet, is particularly difficult to provide since each organisation manages its own nodes, including its own mobile hosts. The required key distribution infrastructure does not generally exist in the Internet, and due to patent and international export restrictions, may not exist throughout the Internet for some time.

IMHP is defined to make use of strong authentication based on such an infrastructure or based on manually configured keys, but also introduces an optional set of simple new

authentication procedures that can be used when no keys are available, yet which preserve the level of security found in today's Internet [18]. In the current Internet, security attacks based on the misuse of ARP [21] or ICMP Redirect messages [22], for example, allow any node connected to one of the physical networks on the normal routing path of a packet to intercept or redirect that packet, but such attacks are not possible for nodes not on this path; nodes elsewhere in the Internet can neither interfere with the packet nor change the routing within the Internet to place themselves on the normal routing path between this source and destination. Based on this assumption, IMHP uses secure cryptographic checksums and a challenge-response mechanism using one-time authenticators to maximise the level of authentication provided when no keys are available. Although these simple authentication procedures are open to some possible attacks, only nodes already in a position to utilise the existing security holes in TCP/IP can utilise these attacks.

This section describes the authentication procedures defined in IMHP and discusses the extension of these procedures to stronger authentication when the necessary infrastructure is available. Note that IMHP makes no attempt to address end to end security or privacy issues, nor does it address the additional privacy issues related to the use of wireless links.

Mobile Host to Home Agent Authentication

When a mobile host in a home agent's home list attempts to register, the home agent must be able to authenticate the binding it receives for the mobile host. Registration with the home agent is a particularly important transaction, because the home agent in the IMHP architecture must always know the current binding of each mobile hosts in its home list. Similarly, a mobile host must be able to authenticate a reply or other management packet it receives from its home agent. This authentication is achieved in IMHP by including an *authenticator* based on a shared secret in all management protocol messages between a mobile host and its home agent (Figure 2 (a)).

This shared secret allows a strong degree of authentication between the mobile host and its home agent. The base level of authentication defined by IMHP in this case involves performing a checksum of the important fields in the registration packet (or reply) and the shared secret, using the MD5 one-way cryptographic hash function [17]. The resulting checksum is sent as the authenticator in registration messages between the mobile host and its home agent. The possibility of replay attacks is minimised by including a monotonically increasing sequence number in registration and reply packets.

Administration of a shared secret between a mobile host and its home agent does not require any network key management infrastructure, since the mobile host and its home agent are both generally owned by the same organisation (they are both assigned home addresses within the same IP network owned by that organisation). The shared secret may be set manually, for example, when the mobile host is at home, at the same time as other configuration of the mobile host and home agent is being performed.

General Authentication Procedures

In general, a node will not share a secret with any particular mobile host or with the mobile host's home agent, and thus will not be able to authenticate IMHP management messages in the same way as a mobile host and the mobile host's home agent can authenticate management messages between themselves.

However, such a node can still obtain an authenticated binding for a mobile host using the simple authentication procedure defined by IMHP, under the assumptions of today's level of Internet security. To obtain an authenticated binding in this way, the node sends a request for the mobile host's binding to the mobile host or to the mobile host's home agent, and includes in the request a random number to be used as an authenticator. If the reply to the binding request contains the same authenticator value, the node may believe the binding contained in the reply (Figure 2 (b)), and may store the binding in its location cache for future use. This random number acts as a one-time disclosing authenticator and is used to guard against a forged reply being sent by some attacker shortly after the request is sent. Only nodes connected to one of the physical networks on the normal routing path taken by the request or reply packet can intercept one of these packets and thus learn the correct authenticator value necessary to forge a reply. Since nodes along these paths must already be implicitly trusted in the current Internet, no new attacks are afforded by this simple authentication mechanism. Other nodes not on the message path are not able to send a successful forged reply because they cannot discover the correct authenticator value.

Any node will normally not know the address of an arbitrary mobile host's home agent, so the IMHP management protocol provides a method by which a packet may be forwarded only to the mobile host's home agent or to the mobile host itself if it is at home. By setting a *route flag* in a management packet addressed to a mobile host, IMHP entities may be instructed to use only normal IP routing in the forwarding of that packet. The management packet thus may not be forwarded using location cache or visitor list entries in intermediate nodes, and the packet will therefore reach the mobile host's home network (and will be

intercepted by the home agent if the mobile host is not at home). If the route flag is set in a packet intercepted by the home agent, the home agent processes the packet on behalf of the mobile host.

If a suitable key management infrastructure is available or a manual key distribution system is used, then a mobile host's binding may be strongly authenticated. The node requesting the binding simply has to confirm that the binding it receives in reply to its request is signed by either a mobile host or its home agent using a cryptographically strong digital signature such as one based on keyed MD5. If the binding is signed by the mobile host's home agent then the node must also be able to confirm the identity of the mobile host's home agent in a strongly authenticated manner, or the home agent must sign the binding on behalf of the mobile host.

Authenticating a Visitor List Entry

A cache agent establishes a location cache entry for a mobile host only when it obtains an authenticated binding for the host. Thus, a location cache entry for a mobile host may always be used to forward packets to that host. In contrast, a local agent may create a visitor list entry for a mobile host as soon as the mobile host registers with it. However, a visitor list entry may not be used to forward packets unless they were tunnelled to the local agent, until after the local agent has authenticated the identity of the registered mobile host. This restriction reduces the possibility of packets being delivered incorrectly by a local agent to a malicious host. An unauthenticated visitor list entry may be used to forward packets tunnelled to the local agent, as the source of the tunnel may be assumed to have an authenticated binding for the mobile host, since otherwise the source would not have tunnelled the packet. This mechanism ensures that existing connections may continue, still using a close to optimal route, as soon as a mobile host registers with a new local agent and notifies its previous local agents.

A local agent authenticates a visitor list entry by confirming that the mobile host's home agent has a binding indicating that the mobile host is registered with this local agent. This authentication can be done using the mechanisms previously described. When the lifetime indicated with this binding expires, the visitor list entry must be re-authenticated. The visitor list entry may also be authenticated as part of the registration process by combining the same type of authentication mechanism into the registration request and reply messages as is used in the general procedure for obtaining an authenticated binding.

The assumption that a local agent may use an unauthenticated visitor list entry to forward a tunnelled packet introduces a minor security risk. Suppose a cache agent has a location cache entry that indicates that a mobile host is registered with a particular local agent but, in fact, the mobile host is located elsewhere or has disappeared from the network entirely. Also, assume that a malicious host pretends to be the mobile host and registers with the local agent. Any packets the cache agent tunnels to the local agent will be forwarded to the malicious host. Fortunately, this risk is limited in duration, in the worst case, by the lifetime of the location cache entry in the cache agent.

Previous Local Agent Authentication

When a mobile host registers with a new local agent, the mobile host arranges that each of its previous local agents that might still have a visitor list entry for the mobile host are notified of the movement. Each previous local agent can authenticate a received notification using the same general authentication procedure previously described, which generally requires a management packet exchange to be carried out with the mobile host's home agent. If the home network is far away or only accessible by a slow or unreliable link, the authentication delay might reduce the performance transparency to the user.

IMHP thus defines an alternative mechanism that may be used for fast simple authentication of notifications to previous local agents. When a mobile host registers with a local agent, the mobile host provides a random number to the local agent for use as an authenticator, thus essentially establishing a shared secret with this local agent for the duration of this registration. Based on this shared secret, the mobile host later uses a strong authentication function, such as keyed MD5, to authenticate any notification it sends to that local agent indicating that the mobile host has moved. The local agent authenticates the notification and any binding contained in it in the same way as other IMHP management messages using strong authentication (Figure 2 (c)). The establishment of the shared secret is subject to attack from other nodes connected to the same physical network as the mobile host during registration, but such nodes can already exploit similar attacks in the current Internet. Whereas the shared secret is established using the assumptions of simple authentication over the single hop between the mobile host and its local agent, the longer path between the mobile host and its previous local agent is able to use strong authentication.

An authenticated notification is used to delete a visitor list entry that the previous local agent holds for a mobile host. It may also be used to create a location cache entry for the

mobile host if the local agent is also capable of functioning as a cache agent. However, a notification to a previous local agent should only be used to create a location cache entry if the current visitor list entry has been authenticated. A location cache entry created in this way must be marked to time out after a period no greater than timeout on the original visitor list entry to stop a malicious host on the local network that gained access to the temporary shared secret forcing the creation of a long lasting false location cache entry. The potential risk is thus limited to be no worse than the effect of any malicious host using the today's Internet protocols.

As with other IMHP authentication procedures, if a suitable key management infrastructure is available or a manual key distribution system is used, then a mobile host's notification to its previous local agent may be strongly authenticated (using no assumptions of simple authentication). The mobile host receiving such a notification simply has to confirm that the binding it receives is signed by the mobile host using a strong authentication function such as keyed MD5.

Transition to Strong Authentication

It is important that a simple authentication mechanism already in existence when strong authentication mechanisms are introduced does not compromise the security of these new mechanisms. For example, if a particular mobile host wants other nodes to strongly authenticate its binding and yet other nodes are willing to use the less secure simple authentication mechanisms, then there is an opportunity for malicious hosts to compromise the new mobile host's security, since an authentication mechanism is only as strong as its weakest link.

IMHP avoids this problem by defining that a node may only allow a mobile host's binding to be authenticated using the simple authentication mechanisms if the node knows, by whatever means, that the mobile host accepts this arrangement. The default case is thus strong authentication. Unfortunately, this emphasis reduces the convenience of the simple authentication mechanisms, as the means of knowing that a mobile host accepts the simple authentication mechanism will usually be manual. However, it does allow cooperating users the possibility of using route optimisation until a key distribution infrastructure become widely deployed.

The principle of default strong authentication can be relaxed in certain circumstances. For example, if a node acts as a cache agent and the cache agent's location cache is only used to tunnel packets sourced by the node then it is sometimes reasonable for the node to attempt

to use simple authentication methods regardless of the wishes of correspondent mobile hosts. The worst that can happen is that packets are intercepted by a malicious host. However, this is a risk that the node is willing to take. Usually, the correspondent mobile host or its home agent will simply reject attempts to use simple authentication mechanisms and a non-optimum route through the mobile host's home network will be used until the node uses strong authentication mechanisms.

4. Forwarding Rules

IMHP defines some rules for packet forwarding that ensure that, whenever possible, packets are routed directly to a destination mobile host rather than being routed through that mobile host's home network and home agent. Some of these rules apply to all IMHP entities, some apply specifically to home agents, and some apply only to local agents and cache agents.

One special case in the forwarding rules, used to avoid a possible routing loop, occurs for a tunnelled packet in which the destination of the tunnel is the same as the original destination of the packet. Such a tunnelled packet is called a *special tunnel packet*, and is always forwarded to the destination mobile host's home network without redirection. No location cache entries or visitor list entries may be used in routing a special tunnel packet; the packet must be routed using only normal IP routing, and will thus reach the destination mobile host's home network, where it will be intercepted by its home agent if the mobile host is away from home. The tunnelling protocol must be designed so that a special tunnel can still be detected after any IP fragmentation. The use of a special tunnel packet is described below in the specific rules in which one is sent or received.

Basic Rules

The following two basic forwarding rules, which apply to all nodes, are designed to ensure that a node receives and correctly processes any packets addressed to itself:

- If a node receives a tunnelled packet and the destination address of the tunnel belongs to the node, then the node should extract the inner packet carried by the tunnel and continue applying the following rules.
- If a node receives a packet that is not tunnelled (or that it has extracted from a tunnel) and the destination address of the packet belongs to the node, then the packet should be passed to the next protocol layer within the node for further processing.

Home Agent Rules

A node functioning as a home agent must always also act as a cache agent at least for the mobile hosts in its home list, and may act as a local agent for those or other mobile hosts as well. It must also process special tunnel packets, as well as management packets in which the route flag is set that are addressed to a mobile host in its home list.

These properties help define the following forwarding rules for a home agent when dealing with packets addressed to the mobile hosts in its home list:

- If a home agent receives an IMHP management packet in which the route flag is set, that is addressed to a mobile host in its home list, then the packet should be passed to the next protocol layer within the home agent node for further processing.
- If a home agent receives a special tunnel packet addressed to a mobile host in its home list, then the home agent should extract the inner packet carried by the tunnel and continue applying the following rules.
- If a home agent receives a packet addressed to a mobile host in its home list, and the home agent (in its role as a local agent) has a visitor list entry for the mobile host, then the home agent should use the visitor list entry to forward the packet locally to the mobile host on the network interface indicated by the visitor list entry.
- If a home agent receives a packet addressed to a mobile host in its home list and the home agent (in its role as a cache agent) has a location cache entry for the mobile host, then the home agent should use the location cache entry to tunnel the packet to the care-of address indicated by the binding in the location cache entry, subject to the following restriction.
- To avoid looping caused by any inconsistent bindings held by different nodes, a home agent should never tunnel a packet back to a node that has just tunnelled the packet to the home agent.
- In all other cases, normal IP routing mechanisms should be used to forward the packet.

Local Agent and Cache Agent Rules

A local agent and a cache agent use forwarding rules that are similar to those defined for a home agent. Differences arise because a home agent normally has an authenticated binding for the mobile hosts in its home list, whereas a local agent might not have an entry in its

visitor list, and a cache agent might not have an entry in its location cache. The following forwarding rules are used by local agents and cache agents:

- If a local agent or a cache agent receives a special tunnel packet or a management packet in which the route flag is set, then the local agent or cache agent should forward the packet using normal IP routing mechanisms.
- If a local agent receives a packet tunnelled directly to this local agent, and the local agent has an entry for the packet's destination in its visitor list, then the local agent should use the entry to deliver the packet locally to the mobile host.
- If a local agent receives a packet that is not tunnelled, and the local agent has an authenticated entry for the packet's destination in its visitor list, then the local agent should use the entry to deliver the packet locally to the mobile host.
- If a cache agent receives a packet, and the cache agent has an entry for the packet's destination in its location cache, then the cache agent should use the location cache entry to tunnel the packet to the care-of address identified in that location cache entry.
- If a cache agent or a local agent receives a packet that was tunnelled directly to this node, and the cache agent or local agent is unable to forward the packet using any of the preceding rules, then it should tunnel the packet to the mobile host's home network using a special tunnel. If the mobile host is at home, the packet will be delivered to it there; otherwise, its home agent will intercept the packet and tunnel it to the mobile host's current care-of address. If the cache agent or local agent instead sent the packet to the mobile host's home network (to the mobile host) using normal IP routing mechanisms (plus any location cache entries or visitor list entries encountered along the way), and if the packet happened to then be routed again through this node, then a routing loop would be formed. Use of the special tunnel ensures that any loops can be easily and quickly detected and broken without having to rely only on the IP time-to-live field in the IP header of the packet.
- In all other cases, normal IP routing mechanisms should be used to forward the packet.

5. Binding Management

IMHP uses lazy notifications to inform other nodes that a mobile host’s binding has changed. A node sends a binding notification to another node advising it to obtain a new binding only when it determines that the other node might have an incorrect binding or no binding for this mobile host, and that a new binding might improve packet routing.

However, a mobile host always notifies its home agent when it moves, and a mobile host always arranges to notify any previous local agents that might still have an old binding for the mobile host.

Sending Binding Notifications

If a node functioning as either a home agent, a cache agent, or a local agent receives a packet that it must tunnel to some mobile host, it is likely that the source node of the packet has an incorrect binding or no binding for the destination mobile host. If the packet was not tunnelled to this node, then the sender apparently had no binding, since otherwise it would have tunnelled the packet itself. If, instead, the packet was tunnelled to this node (not as a special tunnel), then the sender of the tunnel apparently has an incorrect binding, since this node needed to re-tunnel the packet to a new local agent.

In either case, this node may send a binding notification to the source node of the packet. For each such packet, it only sends a single binding notification, but if additional packets arrive from the same source node, addressed to the same destination mobile host, this node may return a binding notification in response to each. Thus, later packets from the same source node effectively trigger a retransmission of the binding notification, in case some notifications are lost by the network.

The binding notification does not necessarily contain the new binding for the mobile host. If the notification contains no binding, it serves to notify the receiving node that it should obtain a new authenticated binding. If the notification does contain a binding, the node receiving it must authenticate the binding before updating its location cache. The procedures for obtaining and authenticating a received binding are described in Section 3.

Notification Back-off

It is important that the network not be flooded with binding notifications, especially as many existing correspondent hosts will not implement IMHP at first and thus will not understand the notifications. In other cases, it may take some time for the notification to be processed and an authenticated binding to be acquired. However, it is also important that a node be able to send more than one notification to another cache agent concerning a mobile host in case the original notification is lost.

A node sending binding notifications must limit the frequency with which they are sent to another node regarding a particular mobile host. After some small number of binding

notifications about the same mobile host to the same node, a back-off algorithm should be used to quickly limit the rate of new notifications about the same binding to that node.

6. Other Features

Restrictions on Advertising Bindings

It is sometimes desirable for a mobile host's binding to be kept private and not be advertised to others in the network. For example, the mobile host might not want correspondent hosts to know where it is connected to the network. Alternatively, a home agent may decide that it wants to distribute a binding only to certain nodes. The cost of this privacy in both cases is non-optimal routing.

IMHP allows a mobile host, when it registers with its home agent, to specify that its binding may not be distributed to other nodes, by setting the *private flag* in the registration request packet. The home agent will subsequently always reply to binding requests from nodes other than the local agent currently serving the mobile host, by indicating that the mobile host is at home.

A home agent may also restrict a particular node from distributing a mobile host's binding, by setting the *private flag* with the binding sent to that node. The node may then not reveal the binding to other nodes.

Mobile Host Popup Mode

Quite often, a mobile host will want to connect to a network where a local agent is not available, at least until IMHP achieves widespread deployment. One possibility is to revert to using a *popup mode* of operation similar to that described in [6, 7]. Effectively, a mobile host becomes its own local agent after manually or automatically (using facilities such as DHCP [13]) obtaining a local address to be a care-of address.

When a mobile host in popup mode moves, and thus acquires a new local care-of address, there is a danger that location cache entries will remain in various cache agents that indicate the old care-of address, which no longer corresponds to this node or perhaps to any node, sine the address was temporarily allocated. Packets tunnelled to this old care-of address are likely to be lost or misdelivered. Thus, it might be desirable, when a mobile host is in popup mode, that its binding is marked as private so that it is not advertised to other nodes.

This would, however, force all packets to the mobile host in popup mode to be routed through the mobile host's home agent.

An alternative, which avoids this non-optimal routing, is for the home agent to reliably notify any cache agent that may have a binding for the mobile host, when the mobile host moves. This type of operation is inefficient in that it forces the home agent to do more work on mobile host movement than is desirable, it requires the home agent to track the identity of all cache agents that may have a mobile host's binding, and has potential reliability problems. However, this approach does allow route optimisation to be used with mobile hosts operating in popup mode. The mobile host may request this service from the home agent during registration.

Home Network Operations

When a mobile host is at home, it is important that its performance be approximately the same as if it were a stationary host. This ensures that the addition of mobile host protocol software to hosts does not degrade home network performance.

In this case, the mobile host no longer needs to periodically re-register with its home agent (which is also its local agent), and the mobile host's routing table should be set for normal operation as for any host in its home network. It is also important that any hosts connected to a mobile host's home network are always able to communicate with the mobile host no matter where the mobile host is connected. When the mobile host is not at home, then the home agent must answer any ARP requests for the mobile host with the home agent's own MAC address. When the mobile host is at home, then it may answer any ARP requests itself.

When a mobile host moves from its home network to some other network, any ARP cache entries held by correspondent hosts on the home network that indicate the mobile host's MAC address must be deleted. There is no guaranteed way to achieve this within the existing ARP protocol [21], but usually it will be sufficient for the home agent to issue a small number of gratuitous ARP replies whenever a mobile host leaves its home network. More than one gratuitous ARP reply must be sent over a period of time, depending on the underlying network, to reduce the possibility of all the gratuitous ARPs being lost. Otherwise, IMHP must depend on ARP timeouts, which are typically too long to maintain performance transparency.

Unfortunately, some existing hosts do not process gratuitous ARP replies correctly. An alternative is for a home agent to always answer ARP requests on behalf of the mobile hosts in its home list even when the mobile hosts are at home. The cost of this alternative is that all communications to a mobile host will be transmitted through its home agent, even on the home network, with a resulting loss of performance transparency. A slight optimisation is to have the home agent or mobile host issue the correct MAC address for a mobile host only to those correspondent hosts that it knows to process gratuitous ARPs correctly.

Special ARP processing may be avoided completely if the home network is a *virtual network*, to which no host may directly connect ; in a virtual network, only the home agent is connected to the network, and all hosts with home addresses in this network are always treated as being mobile hosts that are away from home. However, in this case, routing to a mobile host from a correspondent host always involves a third party (cache agent or home agent) even when connected to the same local network. The solutions proposed by [7] is not applicable as it assumes all hosts on the same subnet as a mobile host are also mobile hosts.

Intermediate Cache Agents

A large part of IMHP's design is motivated by the goal of allowing packets to be routed directly to a mobile host in its current location, rather than forcing all packets for a mobile host to be routed through its home network and home agent. By using route optimisation, IMHP maintains performance transparency to the user. However, IMHP as described so far depends on the correspondent host being able to function as a cache agent. Certainly, during early deployment of IMHP, this will not normally be the case. Intermediate cache agents may be used to allow correspondent hosts that are unaware of IMHP to benefit from its route optimisation features.

Normally a binding notification sent to an unaware correspondent host will be ignored, and eventually the node sending the notification will back-off and only send one very infrequently. However, if an intermediate cache agent snoops on the notification, this cache agent may use the notification as a trigger to acquire an authenticated binding for the mobile host. If normal routing mechanisms route packets from the correspondent host to the mobile host through the intermediate cache agent as a router, then this cache agent may use any location cache entry it may have for the destination mobile host. If a location cache

entry exists, then the cache agent should tunnel the packet to the mobile host's current location.

Depending on the location of the intermediate cache agent relative to the mobile host and the correspondent host, the use of intermediate cache agents can result in optimal or nearly optimal routes even for stationary hosts that do not implement IMHP.

7. Mobility Examples

This section illustrates the operation of IMHP, using the example network configuration shown in Figure 3. This configuration includes two mobile hosts (MH_1 and MH_2) and three local agents (LA_1, LA_2, and LA_3), all with wireless network interfaces. MH_1's home agent is HA_1, and MH_2's home agent is HA_2. In the following examples, the local agents and the mobile hosts are assumed to also be capable of functioning as cache agents.

Basic Operation Example

Suppose MH_1 and MH_2 are within range of the wireless networks controlled by local agents LA_1 and LA_2, respectively, as shown in Figure 4. After discovering LA_1, MH_1 registers with LA_1 and with its home agent, HA_1. Likewise, after discovering LA_2, MH_2 registers with LA_2 and with its home agent, HA_2. (The management packet paths illustrated do not show any tunnelling required to deliver them.)

Now suppose MH_1 wants to send a packet to MH_2. MH_1 first transmits the packet to LA_1, acting as MH_1's default router. Assuming LA_1 does not initially have a binding for MH_2, LA_1 forwards the packet using normal Internet routing mechanisms. The packet is thus forwarded to the MH_2's home network, where it is intercepted by MH_2's home agent, HA_2. As HA_2 should have a location cache entry for MH_2 indicating the care-of address provided by LA_2, HA_2 tunnels the packet to LA_2. LA_2 then uses its visitor list entry for MH_2 to deliver the packet locally to MH_2.

HA_2 can determine that MH_1 probably does not have a binding for MH_2 by the fact that it has to tunnel the packet, and so HA_2 notifies MH_1 that it should acquire MH_2's binding. Subsequently, MH_1 transmits a binding request for MH_2 including a random number as an authenticator and the route flag set. HA_2 intercepts the request because MH_2 is in its home list. HA_2 then replies, including the original authenticator, which causes MH_1 to create a location cache entry for MH_2.

Until the cache entry times out, MH_1 tunnels future packets for MH_2 directly to LA_2, avoiding the non-optimal routing through MH_2's home agent. MH_1 may attempt to reconfirm the binding for MH_2 before timeout occurs if it determines that its past use of the location cache entry justifies the overhead.

Movement Example

Now suppose that MH_2 moves away from LA_2 and into range of the wireless network controlled by local agent LA_3. After detecting the movement using the advertisement and solicitation protocol or lower layer network facilities, MH_2 registers with LA_3 and with its home agent, HA_2, as illustrated in Figure 5.

MH_2 also notifies its previous local agent, LA_2 that it has moved, using the authenticator negotiated during its earlier registration with LA_2. After authenticating the notification, LA_2 deletes its visitor list entry and creates a location cache entry for MH_2 (assuming LA_2 had previously authenticated the visitor list entry).

Now suppose that MH_1 wants to send a packet to MH_2. Assuming MH_1's location cache entry has not timed out, MH_1 tunnels the packet to LA_2. When LA_2 receives the packet, LA_2 uses its location cache entry for MH_2 to re-tunnel the packet to LA_3.

LA_2 also sends a binding notification to MH_1, notifying MH_1 that it should acquire a current binding for MH_2. As before, MH_1 sends a request for MH_2's binding and uses the authenticated reply to update its location cache entry for MH_2. Until this location cache entry times out, MH_1 will tunnel any future packets it sends to MH_2 directly to LA_3.

If, instead, LA_2 has timed out its location cache entry for MH_2 before MH_1's tunnelled packet for MH_2 reaches LA_2, then LA_2 uses a special tunnel to tunnel the packet to HA_2. HA_2 subsequently tunnels the packet to LA_3. Subsequent notifications from LA_2 to MH_1 eventually cause MH_1 to acquire a location cache entry for MH_2, so that MH_1 can tunnel packets directly to MH_2's current local agent. These operations are not shown in Figure 5.

Intermediate Cache Agent Example

A stationary host that does not implement IMHP may need to send packets to a mobile host. It is desirable, in this case, that the packets be routed as directly as possible to the mobile host's current location. Suppose some stationary host, SH, that does not implement IMHP and is thus unaware of host mobility, wants to communicate with mobile host MH_2 from Figure 3. When SH receives a notification advising it to acquire a current binding for

MH_2, it will ignore the notification. Packets from SH to MH_2 would thus always be routed non-optimally through MH_2's home agent, HA_2, before being then forwarded to MH_2.

IMHP allows intermediate cache agents to snoop on binding notifications, and for such a cache agent, if it chooses, to obtain an authenticated binding for a mobile host in response to such a notification. Suppose CA is a router on the path from SH to MH_2's home network, such as SH's default router, which is capable of functioning as a cache agent. In the example illustrated in Figure 6, CA snoops on notifications from HA_2 to SH. CA then acquires an authenticated binding for MH_2 and uses the resulting location cache entry to tunnel packets it receives addressed to MH_2 directly to MH_2's local agent, LA_2.

8. Implementation

IMHP has been implemented at Macquarie University using both Ethernet and 1□Mb/s infrared network interfaces on an IBM RT platform running under Mach Unix and on an IBM PS/2 running under AIX. This section briefly outlines the modifications that have been made to the kernel and the user code to implement IMHP as described in this paper.

The main change to the kernel is to add a visitor list and a location cache that are searched before the main routing table. Both the visitor list and the location cache are built in the same format as a normal routing table so that the standard routing table look-up can be used. However each entry contains additional information such as timeouts and authentication information. The normal user level interfaces using *route* and *netstat* have been modified to be able to show and change visitor list and location cache entries.

A home list is implemented by marking as permanent those entries in the visitor list or location cache corresponding to the mobile hosts in the home list. If a local cache entry or a visitor list entry marked as permanent entry times out, rather than deleting it, the entry is converted into a visitor list entry indicating the mobile host is at home.

Other changes to the kernel are listed below. It should be noted that no changes have yet been made to any ARP code, as the home network is a virtual home network in the Macquarie University configuration.

- The IP input routine was modified so that it can recognise management packets with the route flag set and special tunnels addressed to mobile hosts on its home list.

- The IP output routine and IP forwarding routine were modified to use the visitor list and the location cache.
- Code was added to implement tunnelling and de-tunnelling of packets.
- Code was added that determines the source of a packet that may have an incorrect binding for a destination mobile host and to send the source of the packet a binding notification.

The rest of IMHP is implemented as user level code, running as a single daemon process on each node. The daemon implements all processing of the IMHP advertisement and solicitation protocol, the registration protocol, and the binding management protocol.

The implementation of IMHP is operational, although only limited performance testing has been done due to restrictions on the hardware configuration available. Initial evaluation of the IMHP implementation, though, indicates IMHP's goals with respect to performance transparency have been achieved.

9. Conclusion

This paper has described the main features, operations, and implementation of a new mobile host protocol, called the Internet Mobile Host Protocol (IMHP), which achieves transparent mobility featuring route optimisation and integrated authentication of all management packets. IMHP is designed to take advantage of strong authentication mechanisms when the necessary infrastructure becomes available in the Internet and yet provides a simple authentication mechanism that can be used today that preserves the current level of security in the Internet.

In the short term, IMHP can provide a valuable service to the majority of users on the Internet who either do not have the need for strong authentication mechanisms beyond that provided by the Internet today or are unwilling to pay the price of less than optimal routing in the meantime in the absence of a key distribution infrastructure. In the long term, as key management systems become available, IMHP can provide route optimisation for all packets from any correspondent host to any mobile host, regardless of a mobile host's location, in a manner that will satisfy future authentication requirements.

References

[1] Charles Perkins, "Providing continuous network access to mobile hosts using TCP/IP", *Computer Networks and ISDN Systems*, vol. 26, pp. 357-369, 1993.

[2] Charles Perkins, Yakov Rekhter, "Support for mobility with connectionless network layer protocols (transport layer transparency)", *Internet Draft*, Jan. 1993.

[3] Fumio Teraoka, Yasuhiko Yokote, and Mario Tokoro, "A network architecture providing host migration transparency", *Proceedings of the SIGCOMM '91 Conference: Communications Architectures & Protocols*, pp 209-220, Sept. 1991.

[4] Fumio Teraoka, "A study on host mobility in wide area networks", PhD dissertation, Keio University, Japan, Jan. 1993.

[5] Fumio Teraoka, Kim Claffy, and Mario Tokoro, "Design, implementation, and evaluation of Virtual Internet Protocol", *Proceedings of the 12th International Conference on Distributed Computing Systems*, pp. 170-177, June 1992.

[6] John Ioannidis, "Protocols for mobile networking", PhD dissertation, Columbia University, 1993.

[7] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr., "IP-based protocols for mobile internetworking", *Proceedings of the SIGCOMM '91 Conference: Communications Architectures & Protocols*, pp. 235-245, Sept. 1991

[8] David B. Johnson, "Ubiquitous mobile host internetworking", *Proceedings of the Fourth Workshop on Workstation Operating Systems*, pp. 85-90, Oct. 1993.

[9] Charles Perkins, Andrew Myles, “Mobile IP”, *SBT/IEEE International Telecommunications Symposium*, Rio De Janeiro, 22-25 August 1994.

[10] Andrew Myles, David Skellern, "Comparison of mobile host protocols for IP", *Journal of Internetworking Research and Experience*, vol. 4, no. 4, pp 175-194, Dec.□1993.

[11] Andrew Myles, David Skellern, "Comparing four IP based mobile host protocols", *Computer Networks and ISDN Systems*, vol. 26, pp. 349-355, 1993.

[12] Kenneth G. Carlberg, "A routing architecture that supports mobile end systems", *Proceedings of MILCOM'92*, vol. 1, pp. 159-164, Oct. 1992.

[13] R. Droms, "Dynamic Host Configuration Protocol", RFC 1541, Oct. 1993.

[14] J. B. Postel, "Multi-LAN address resolution", RFC 925, Oct. 1984.

[15] David B. Johnson, Andrew Myles, Charles Perkins, "The Internet Mobile Host Protocol (IMHP)", *Internet Draft*, Feb. 1994.

[16] C. Sunshine and J. Postel, "Addressing mobile hosts in the ARPA Internet environment", IEN 135, Mar. 1980.

[17] R. Rivest, "The MD5 message-digest algorithm", RFC 1321, Apr. 1992.

[18] S. M. Bellovin, "Security problems in the TCP/IP protocol suite", *Computer Communication Review*, vol. 19, no. 2, pp. 32- 48, Apr. 1989.

[19] David B. Johnson, "Mobile host internetworking using IP loose source routing", Technical Report CMU-CS-93-128, School of Computer Science, Carnegie Mellon University, Feb. 1993.

[20] David B. Johnson, "Scalable and robust internetwork routing for mobile hosts", *Proceedings of the 14th International Conference on Distributed Computing Systems*, pp. 2-11, June 1994.

[21] David C. Plummer, "An Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit Ethernet addresses for transmission on Ethernet hardware", RFC 826, Nov. 1982.

[22] J. B. Postel, editor, "Internet Control Message Protocol", RFC 792, Sept. 1981.

List of Figures and Captions

Figure 1. Adding the IMHP tunnelling header to a packet

Figure 2. Simple authentication procedures

Figure 2 (a) Home agent/mobile host authentication

Figure 2 (b) General binding authentication

Figure 2 (c) Previous local agent authentication

Figure 3. Example configuration

Figure 4. Basic operation example

Figure 5. Movement example

Figure 6. Stationary host example

Biographies

Andrew Myles received a B.Sc in 1984 and a B.E (Electrical) with First Class Honours and University Medal in 1986 from the University of Sydney. From 1987 to 1989 he worked at Hewlett Packard Laboratories in Bristol, UK before returning to Australia. He is currently completing his Ph.D. at Macquarie University in Sydney with a special interest in MAC and network layer protocols for wireless networks. He is a student member of the IEEE.

David Johnson received the B.A. degree in computer science and mathematical sciences in 1982, and the M.S. and Ph.D. degrees in computer science in 1985 and 1990, respectively, all from Rice University. He is currently an Assistant Professor of Computer Science at Carnegie Mellon University, where he has been since 1992. Prior to joining the faculty at Carnegie Mellon, he was a Research Scientist and Lecturer at Rice University for three years. His research interests include network protocols, distributed systems, and operating systems. Dr. Johnson is a member of the IEEE Computer Society, IEEE Communications Society, ACM, USENIX, Sigma Xi, and the Internet Society.

Charles Perkins received the B.A. degree and the M.E.E. degree from Rice University in 1976 and 1977, and the M.A. degree from Columbia University in 1992. Since 1984, he has worked for IBM on a variety of projects related to networks, multiprocessors, network protocols, and mobile computing, and is listed as the inventor in several patents related to mobile computing. He is a member of USENIX, IEEE, ACM, and the Internet Society.

Conference Presentation Information

Portions of the material in this paper were presented at INET'94, the Annual Conference of the Internet Society, held in conjunction with the 5th Joint European Networking Conference, in Prague, Czech Republic, in June 1994. These ideas have also been presented and discussed within the Mobile IP Working Group of the Internet Engineering Task Force during 1994.