To appear in *IEEE Personal Communications*, 3(1), February 1996.

Protocols for Adaptive Wireless and Mobile Networking

DAVID B. JOHNSON AND DAVID A. MALTZ

Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891

dbj@cs.cmu.edu, dmaltz@cs.cmu.edu

The goal of the *Monarch Project*[1] at Carnegie Mellon University is to develop networking protocols and protocol interfaces to allow truly seamless wireless and mobile host networking. The scope of our efforts includes protocol design, implementation, performance evaluation, and usage-based validation, spanning areas ranging roughly from portions of the ISO Data Link layer (layer 2) through the Presentation layer (layer 6). In this article, we give a status report of our current work in the Monarch Project, placing it in the context of broader efforts by the Internet mobile networking community.

Our work will enable mobile hosts to communicate with each other and with stationary or wired hosts, transparently making the most efficient use of the best network connectivity available to the mobile host at any time. To this end, the networking protocols must support adaptive operation in a number of ways. For example, host mobility means that protocols must be able to adapt packet routing to reach each mobile host in its current location. In addition, different wireless networks, intended for example for local-area, metropolitan-area, and wide-area use, make different tradeoffs in factors such as bandwidth, latency, error rate, and usage cost, providing different levels of quality of network connection with each wireless networking product or service. Network protocols should be able to adapt to optimize use of the best available network connection for each mobile host at any time. Further, in order to allow higher-layer protocols and applications to adapt to these changes in network connection quality, network protocols should be able to provide information to higher layers when such changes take place.

We are experimenting with our protocols in the context of the "Wireless Andrew" infrastructure, currently being built at Carnegie Mellon [10]. The Wireless Andrew infrastructure builds on the current wired network infrastructure on campus, consisting mostly of 10-megabit per second Ethernet equipment. For high-speed wireless access on campus, we are installing an AT&T WaveLAN network covering most of the campus buildings [32]. WaveLAN uses direct-sequence spread spectrum radio in the 900 MHz ISM band to provide a raw data rate of 2 megabits per second. For wireless access off campus or where otherwise out of range of the WaveLAN network, we are using Cellular Digital Packet Data (CDPD) [4]. The CDPD service uses idle voice channels on the existing AMPS cellular telephone network to transmit data packets at a raw data rate of 19.2 kilobits per second.

[1] The Monarch Project is named in reference to the migratory behavior of the monarch butterfly. Each autumn, millions of monarch butterflies migrate from central and eastern United States and Canada to overwintering roosts in central Mexico; with the coming of spring, the monarch population again migrates northward. The name "Monarch" can also be considered as an acronym for "Mobile Networking Architecture."

In the next section of this article, we describe our work in routing packets to mobile hosts in a large internetwork, such as the Internet, and give an overview of our implementation work in this area. Next, we discuss the problem of routing in an *ad hoc* network of wireless mobile hosts, as might be needed in an area without established wireless networking infrastructure; we describe a new protocol we have developed for routing in such a network and summarize the results from a simulation of the protocol. We then describe our recent work in providing support for adaptive operation of higher-layer protocols and applications; we have developed an inexpensive protocol and API for notifying higher layers when the quality of a mobile host's network connection changes as it moves between different locations, possibly including changes in the type of network in use at each location. Finally, we compare our work to related mobile networking research elsewhere and present conclusions.

Mobile Internetwork Routing

Existing internetworking protocols, including IP, NetWare IPX, ISO CLNP, and AppleTalk, do not support host mobility. In order to aggregate the routing information and routing decisions at each level of the internetwork topology, internetworking protocols use *hierarchical* addressing and routing schemes. For example, in the Internet, IP addresses are divided into a separate *network number* and *host number*; routers throughout the Internet need be concerned only with routing a packet to the correct network, and once there, it becomes the responsibility of that network to route the packet to the correct individual host. This routing aggregation becomes increasingly important as the size of the internetwork grows. The Internet, in particular, currently consists of over 6 million individual hosts, and this number has been doubling approximately every year. Indeed, new levels of hierarchy have been added to the Internet addressing scheme with subnetting [21] and CIDR [9], and additional support for further hierarchy is planned in IPv6, the new version of IP currently being designed for the Internet [2].

It is this hierarchy, however, that defeats host mobility. With hierarchal addressing and routing, packets sent to a mobile host can only be routed to the mobile host's home network, regardless of the host's current location possibly away from home. A mobile host could perhaps change its address as it moves from one network to another, but such changes can be difficult and error-prone; changing addresses involves modifications to a number of configuration files on the host and on network servers, and often requires all existing transport-level network connections to be restarted or the host to be rebooted. Instead, a solution is needed for correctly routing packets to any mobile host host in its current location given the host's (constant) home address.

The IETF Mobile IP Protocol

The Internet Engineering Task Force (IETF) is the principal protocol standards development body for the Internet. Over the past few years, the IETF Mobile IP Working Group has been working to develop a standard for routing IP packets to mobile hosts in the Internet, and we have contributed a number of protocol designs to this effort [12, 13, 15, 23]. Working within the IETF provides a direct avenue for transferring the results of our research into the Internet community. In this section, we provide an overview of the basic IETF Mobile IP standard which is currently nearing completion [24].

Figure 1 illustrates the basic architecture of the protocol. In Figure 1, R1, R2, and R3 are routers, each connecting an IP subnet to a simplified Internet backbone. M is a mobile host whose home network is the network connected by R2 but which is currently connected to a wireless network through router R4. Each mobile host must have a *home agent* on its home network, which forwards IP packets to the mobile host while it is away from home. Here, router R2 is serving as the home agent for mobile host M, although any host or router on this home network could serve that role. When visiting any network away from home, each mobile host must also have a *care-of address*. Normally, the care-of address is the address of a *foreign* *agent* within the local foreign subnet that has agreed to provide service for the mobile host; the foreign agent delivers packets forwarded for the mobile host to it on the local network. Here, router R4 is serving as the foreign agent for M. Optionally, if a mobile host can acquire a temporary IP address within the local subnet, such as through DHCP [7], it may instead use this temporary address as its care-of address; packets tunneled to the mobile host are tunneled to this temporary address, while the mobile host continues to use its home address for all other functions. In this case, the mobile host in effect operates as its own foreign agent with this temporary address.

To find a foreign agent with which to register, an *agent discovery* protocol is used. Agent discovery also provides a means for a mobile host to detect when it has moved within range of a different wireless network; it can detect when it has moved to a new foreign network when it receives an advertisement from a new foreign agent, and when it has returned to its home network when it receives an advertisement from its home agent. The agent discovery protocol operates as a compatible extension of the existing *ICMP router discovery* protocol [6].

When moving to a new location, a mobile host must register with its home agent so that the home agent always knows the mobile host's current care-of address. When using the address of a foreign agent as its care-of address, the registration takes place through that foreign agent so that the foreign agent can agree to provide service to the mobile host and knows that the mobile host is using this care-of address. The association between a mobile host's home address and its care-of address is called a *mobility binding*, or simply a *binding*. Each binding has associated with it a *lifetime* period, negotiated during the mobile host's registration, after which the registration is deleted; the mobile host must reregister within this period in order to continue service with this care-of address.

When sending a packet to a mobile host, a *correspondent host* simply addresses and sends the packet in the same way as any other IP packet, and the packet will thus be routed through the Internet to the mobile host's home network. The correspondent host need not understand the Mobile IP protocol or know that the destination host is mobile. While a mobile host is registered with a care-of address away from home, the mobile host's home agent must intercept any packets on its home network addressed to the mobile host. For each such packet intercepted, the home agent *encapsulates* the packet and *tunnels* it to the mobile host's care-of address.

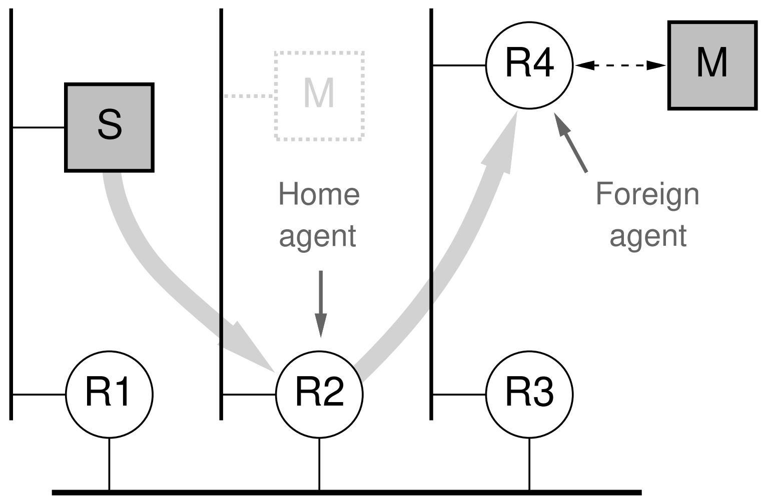

Figure 1 Basic architecture of the IETF Mobile IP protocol

The default encapsulation protocol, known as “IP in IP” encapsulation, is illustrated in Figure 2. With this protocol, a new IP header (shaded) is wrapped around the existing packet. The source address in the new IP header is set to the address of the node tunneling the packet (the home agent), and the destination address is set to the mobile host’s care-of address. The protocol number (such as TCP or UDP) in the new IP header is set to the protocol number for “IP in IP” encapsulation. Once encapsulated, the packet is routed through the Internet in the same way as any IP packet addressed to the foreign agent, and only the home agent and foreign agent need know that tunneling is taking place. When the packet arrives at the foreign agent, the packet is processed by the encapsulation protocol at the foreign agent, as indicated by the protocol number in the IP header. The foreign agent removes the added header and transmits the packet to the mobile host over the local network interface on which the mobile host is registered.

The “IP in IP” encapsulation protocol adds 20 bytes (the size of an IP header) to each packet tunneled to a mobile host away from home. An alternative tunneling protocol, known as “minimal” encapsulation, is also defined within the basic Mobile IP protocol, and adds only 8 or 12 bytes to each packet. This protocol is illustrated in Figure 3. With this protocol, a small tunneling header (shaded) is inserted in the packet *after* the existing IP header, before any existing transport level header such as TCP or UDP. The destination address in the IP header is copied into the tunneling header and is replaced in the IP header by the mobile host’s care-of address. Similarly, the protocol number in the IP header is copied into the tunneling header and is replaced in the IP header by the protocol number indicating minimal encapsulation. Finally, if the original sender of the packet is not the node tunneling the packet (the home agent), the source address in the IP header is copied into the tunneling header and is replaced in the IP header by the tunneling node’s address,

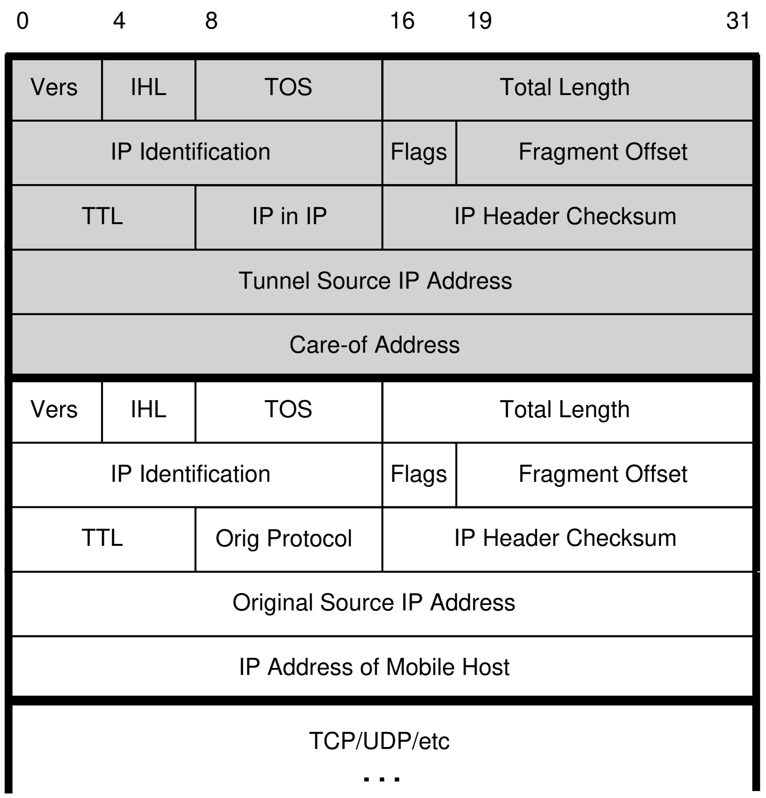

Figure 2 Mobile IP tunneling using “IP in IP” encapsulation

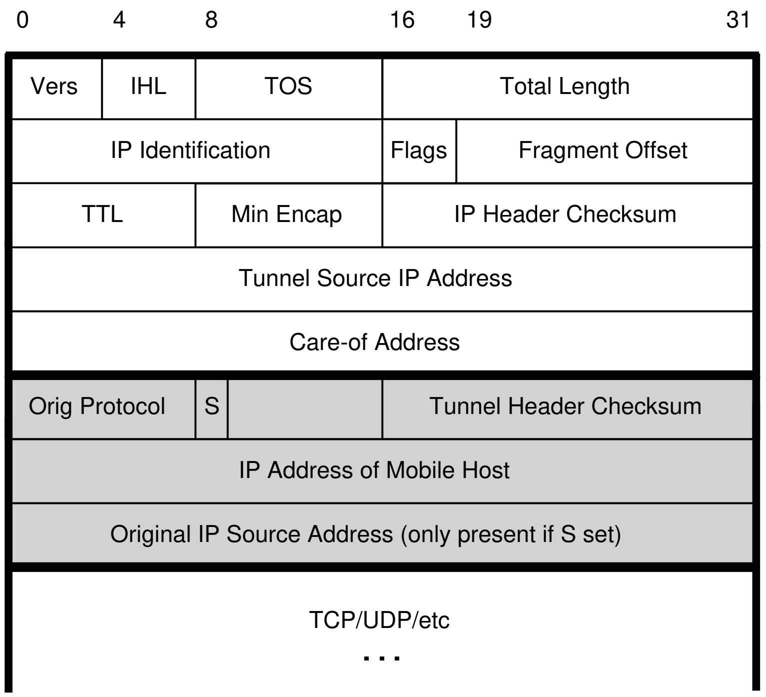

Figure 3 Mobile IP tunneling using “minimal” encapsulation

and a bit is set in the tunneling header to indicate that the copied source address is present. When the packet arrives at the foreign agent, the original IP header is reconstructed and the tunneling header is removed, and the packet is transmitted locally to the mobile host. Although more efficient than “IP in IP” encapsulation, the minimal encapsulation protocol cannot be used with IP packets that have been fragmented [27], since the tunneling header does not provide a means to indicate that the original packet was a fragment.

All registrations of a mobile host with its home agent must be authenticated, in order to guard against malicious forged registrations. Without authentication, an attacker could register a false care-of address for a mobile host, causing its home agent to arbitrarily redirect future packets destined to the mobile host. Registration authentication must verify that the registration request legitimately originated with the mobile host, that the request has not been altered in transit to the home agent, and that an old registration request is not being replayed (perhaps long after the mobile host was at that care-of address).

The protocol currently uses authentication based on the MD5 *secure one-way hash* function [28]. A “keyed MD5” algorithm is used, based on a secret key shared between a mobile host and its home agent, such that the authentication value can only be correctly computed by a node knowing the secret key. Administration of the shared secret key should be fairly simple, since both the mobile host and its home agent are owned by the same organization (both are assigned IP addresses in the home network owned by that organization). Manual configuration of the shared key may be performed, for example, any time the mobile host is at home, while other administration of these hosts is being performed. Replay protection currently may use either *nonces* or *timestamps*.

Route Optimization Extensions

In the basic IETF Mobile IP protocol, while a mobile host is away from its home network, *all* packets for the mobile host must follow the path shown in Figure 1. Each packet is routed through the Internet to the mobile host's home network and must then be tunneled by the mobile host's home agent to the mobile host's current location. This indirect routing through the home agent in general causes unnecessary overhead on the home network and on the portion of the Internet leading to and from the home network, and causes unnecessary latency in the delivery of each packet to the mobile host.

We have developed a compatible set of extensions to the basic IETF Mobile IP protocol to address this problem, and these extensions are now being standardized alongside the basic Mobile IP protocol within the IETF [17]. These extensions, known as "Route Optimization," allow other hosts or routers sending packets to a mobile host, to dynamically learn and cache the mobile host's current location; the sending node can then tunnel its own packets directly to the mobile host, bypassing the trip to and from the home agent. This feature has been present in all of our designs submitted to the Mobile IP Working Group [12, 13, 15, 23], and we view it as essential for the efficiency and scalability of the protocol.

In the Route Optimization extensions, when a mobile host's home agent intercepts and tunnels a packet to a mobile host away from home, the home agent also returns a *binding update* message to the original sender of the packet, as shown in Figure 4 (a). This allows the sender to cache the current binding of the mobile host and to use the care-of address in the binding in tunneling its own packets to the mobile host in

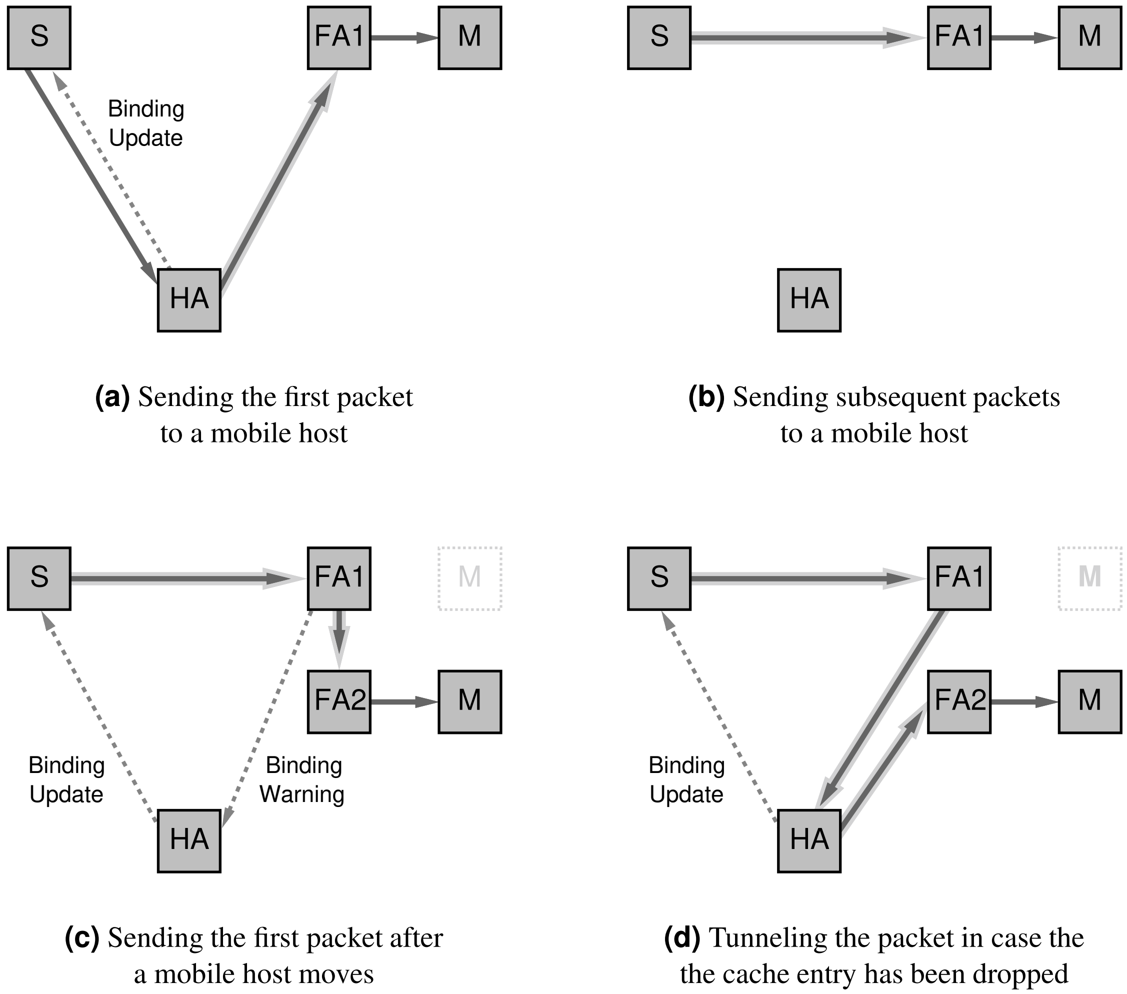

Figure 4 Operation of the Route Optimization protocol extensions

the future, as shown in Figure 4 (b). One challenge that must be addressed in the design of this mechanism, though, is that of *cache consistency*, as when a mobile host moves to a new location, all cached copies of its binding at correspondent hosts become out-of-date.

With Route Optimization, when a mobile host moves from one foreign agent to another, it may notify its previous foreign agent of its new care-of address by sending it a *binding update* message. This allows the previous foreign agent to cache the new binding of the mobile host, forming a "forwarding pointer" to its new location. If a correspondent host later tunnels a packet for the mobile host using an out-of-date cache entry, the previous foreign agent will receive the packet and will re-tunnel it to the new location. The previous foreign agent also sends a *binding warning* message to the mobile host's home agent to request it to send a binding update message to the correspondent host. For example, Figure 4 (c) shows the operation of the protocol after mobile host M has moved from foreign agent FA1 to foreign agent FA2.

If, instead, the cache entry at the previous foreign agent no longer exists by this time (for example, because that entry in the cache was replaced with an entry for a different mobile host), the foreign agent instead forwards the packet to the mobile host's home agent by tunneling the packet to the mobile host's own address, as shown in Figure 4 (d). The packet will thus reach the home agent in the same way as any other packet addressed to the mobile host; the home agent will also be able to determine from the tunnel encapsulation header that it was tunneled from this foreign agent, allowing recovery in the case in which the home agent believes that this is the current foreign agent serving the mobile host, but the foreign agent perhaps has crashed and lost its knowledge of the mobile host's registration.

Cache consistency is thus addressed in both cases by dynamically updating any out-of-date cache entry when it is next used. A packet routed based on an out-of-date cache entry will be routed indirectly to the mobile host's new location, and the cache entry will be updated as a side-effect.

A further challenge that must be addressed in design of Route Optimization is that of *authentication*. Unlike the basic IETF Mobile IP protocol, Route Optimization may in general require the ability to authenticate a message to any node in the Internet. In the basic Mobile IP protocol, all control over routing packets to a mobile host rests with the mobile host's home agent, which intercepts and tunnels all packets to the mobile host. Authentication of registration messages with the home agent in this way is reasonably easy, since the home agent and the mobile can share a secret key. However, with Route Optimization, any correspondent host that is to cache a mobile host's binding must be able to authenticate the binding update message in which it learns the mobile host's binding, in order to guard against attacks involving forged binding updates. Authentication is this case is much more difficult, since the correspondent host may in general belong to a different organization than the mobile host and its home agent, and since there is currently no generalized authentication or key management mechanism for the Internet; patent restrictions and export controls on the necessary cryptographic algorithms have slowed development and deployment of such facilities in the Internet.

In the Route Optimization extensions, we are currently using the same style of authentication for binding update messages as is used for registration in the basic IETF Mobile IP protocol. In order for the home agent to send a binding update to a correspondent host, it must share a secret key with the correspondent. Until a key distribution mechanism is defined for the Internet, these keys will be manually configured, and if no shared key exists, the Route Optimization extensions cannot be used with this correspondent. However, we have defined the protocol to minimize the number of pairwise shared secret keys required for operation. By establishing a shared secret key with some home agent, a correspondent host is able to receive authenticated binding updates (and thus to maintain cached bindings) for all mobile hosts served by this home agent. This relationship is fairly natural, since the mobile hosts served by any particular home agent, in general, all belong to a single organization (that also owns the home agent and the home network). If the user of a host often collaborates with any number of people from this organization, manually establishing the shared secret key with this home agent may be worthwhile.

Implementation Status

We have completed an implementation of the mobile internetwork routing protocol under the NetBSD version of the Unix operating system. This implementation contains all features of the basic IETF Mobile IP protocol, and we are currently completing additions to the implementation for Route Optimization and for network connection quality notifications for supporting adaptive higher-layer protocols and applications (described later in this article). Our implementation includes all functions of a mobile host, correspondent host, home agent, and foreign agent, and allows dynamic, transparent switching between the Ethernet, WaveLAN, and CDPD networks of the Wireless Andrew infrastructure. Since NetBSD is based on the 4.4BSD Lite Unix source, we believe our implementation should be able to be ported easily to other versions of Unix derived from one of the Berkeley source distributions, but we have not yet attempted this. We intend to make the source for our implementation freely available once it is completed.

The implementation is divided between a portion in the kernel and a daemon process running on the host. In general, operations which must be performed for each packet, such as encapsulation and decapsulation, are performed in the kernel, whereas higher-level functions and policy decisions are performed within the daemon. For example, the exchange of packets necessary for registration and the management of registration lifetimes is the responsibility of the daemon, which sends messages on a `PF_ROUTE` routing socket to the kernel to manipulate the kernel's routing tables. This structure is similar to the implementation of existing routing daemons for Unix, such as `routed` and `gated` [33].

Routing in Ad Hoc Wireless Networks

At times, no infrastructure such as the Internet may be available for use by a group of wireless mobile hosts, or the use of an available network infrastructure may be undesirable due to reasons such as cost or convenience. Examples of such situations include disaster recovery personnel or military troops in cases in which the normal infrastructure is either unavailable (such as in a remote area) or destroyed (such as after an earthquake); other examples include business associates wishing to share files in an airport terminal, or a class of students needing to interact during a lecture. If each mobile host wishing to communicate is equipped with a wireless local area network interface, the group of mobile hosts may form an *ad hoc network*. An ad hoc network is a temporary network, operating without the aid of any established infrastructure or centralized administration.

In an ad hoc network, some hosts wishing to communicate may be outside of wireless transmission range of each other, but may be able to communicate if other hosts in the network are willing to forward packets for them. For example, Figure 5 depicts a simple ad hoc network of three mobile hosts, in which the transmission range of each host's wireless interface is indicated by a circle around the host. Mobile host A cannot directly send a packet that will reach C, since C is outside A's wireless transmitter range. However, A can send the packet to B, if B is willing to forward the packet to C by retransmitting it.

An ad hoc network in general requires some form of routing protocol in order to dynamically find multihop paths through the network, and in order to adapt to new routes as the mobile hosts in the network move. Furthermore, the protocol must be able to operate correctly in spite of the varying propagation characteristics of each mobile host's wireless transmissions, for example due to changes in sources of interference in the vicinity of each mobile host.

Conventional Routing Protocols

Conventional routing protocols for wired networks use either *distance vector* or *link state* algorithms, and the basic distance vector algorithm has also been used successfully in some wireless ad hoc networks [8, 18, 25]. In distance vector routing, each router broadcasts to each of its neighbor routers its view of the distance

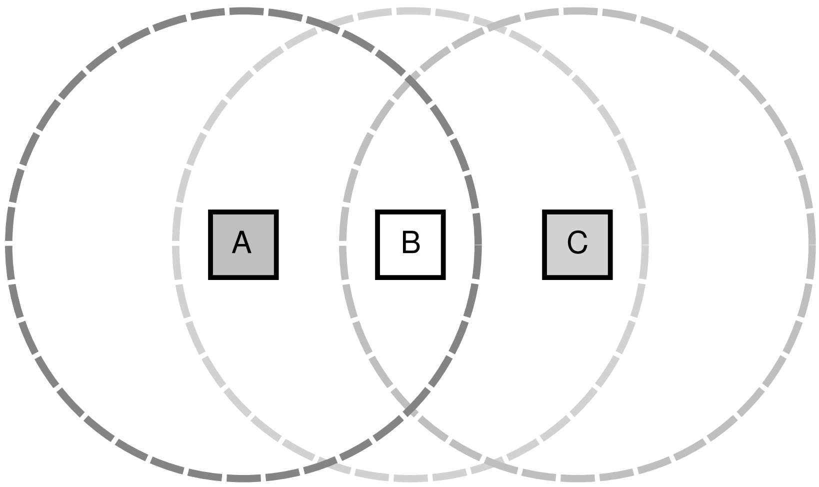

Figure 5 A simple ad hoc network of three wireless mobile hosts

to all hosts, and each router computes the shortest path to each host based on the information advertised by each of its neighbors. For use in ad hoc networking, each mobile host is treated as a router and periodically broadcasts a routing update packet to any neighbor mobile hosts within its transmission range.

However, in an ad hoc network, network bandwidth, battery power, and available CPU processing time on each host are likely to be limited resources. With distance vector routing, a mobile host must continue to send periodic routing updates, occupying network bandwidth and consuming battery power on the host for the transmissions. Furthermore, each of its neighbor mobile hosts must continue to receive these updates and thus cannot easily conserve its own battery power by putting itself into "sleep" or "standby" mode when not busy with other tasks. In addition, many of the "links" between routers seen by the routing algorithm may be redundant, since all communication is by broadcast transmissions. These redundant links unnecessarily increase the CPU overhead required to process routing updates and to compute new routes.

Finally, conventional routing protocols are not designed for the type of dynamic environment that may be present in ad hoc networks. In conventional networks, links between routers occasionally go down or come up, and sometimes the cost of a link may change due to congestion, but routers do not generally move around dynamically as may happen in an ad hoc network. Distance vector algorithms, in particular, converge slowly to new, stable routes after changes in topology, and may create temporary routing loops and "black holes." Furthermore, in some environments and host configurations, distance vector protocols may compute some routes that do not work, since wireless transmissions between two hosts may not necessarily work equally well in both directions, due to differing propagation or interference patterns around the two hosts. Depending on the wireless network MAC protocol in use, even though a host such as A in Figure 5 may receive a routing update from another mobile host such as B, packets A might then transmit to B for forwarding may not be able to reach it.

A Dynamic Source Routing Protocol

We have designed a new routing protocol for ad hoc networks based on a different type of routing. Rather than using either distance vector or link state routing, our new protocol uses *dynamic source routing* of packets between hosts in the ad hoc network [14, 16]. In source routing, the sender of a packet determines the complete sequence of nodes through which to forward the packet, and lists this route in the packet's header; when received by each node along this path, the packet is simply retransmitted to the next "hop" indicated in the path. Source routing has been used in a number of contexts for routing in wired networks, using either statically defined or dynamically constructed source routes, and has been used with statically configured routes for routing in a wireless network [19].

In our dynamic source routing protocol, there are *no* periodic routing messages of any kind. Each mobile host participating in the ad hoc network maintains a *route cache* in which it caches source routes that it has learned. When one host sends a packet to another host, the sender first checks its route cache for a source route to the destination. If a route is found, the sender uses this route to transmit the packet. If no route is found, the sender may attempt to discover one using a *route discovery* protocol. While waiting for the route discovery to complete, the host may continue normal processing and may continue to send and receive packets with other hosts. The host may buffer the original packet in order to transmit it once the route is learned from route discovery, or it may discard the packet, relying on higher-layer protocol software to retransmit the packet if needed.

In route discovery, the sender broadcasts a *route request* packet, which propagates as needed through the ad hoc network either to the intended destination host or to another host that can reply giving a route from the sender to the destination. The reply is returned to the original sender in a *route reply* packet. Each route request packet from a sender contains a unique *request id*. As the request propagates, each host adds its own address to a route being recorded in the packet, before broadcasting the request on to its neighbors (any host within range of its wireless transmission). When receiving a request, if a host has recently seen this request id or if it finds its own address already recorded in the route, it discards that copy of the request and does not propagate that copy further. The protocol makes extensive use of caching of routes and partial routes, and hosts may reply from their cache with routes to other hosts in order to avoid propagating a route discovery packet. Capitalizing on the broadcast nature of wireless transmissions, the protocol also takes advantage of promiscuous receive mode in the network interface to optimize route discovery. For example, mobile hosts can learn routes from arbitrary passing data packets (even those not addressed to the host), and can automatically shorten routes in use when two hosts move close enough together to remove an intermediate hop from the route.

Figure 6 illustrates a sample execution of the basic route discovery protocol, in which mobile host A is attempting to discover a route to host D. Each individual route request message sent is indicated by an arrow from the sending to the receiving mobile host, and the wireless transmission range of each mobile host is indicated by a circle around that host. When a host receives a route request message, it discards the request if it appears to be a duplicate; the non-duplicate request messages in Figure 6 are indicated in bold. As shown, the route request messages propagate outward from the host initiating the discovery, with only non-duplicate messages causing further propagation. Optimizations to the protocol making full use of the route cache also prevent a host receiving a route request from propagating the request if it can complete the request from its route cache.

While a host is using any source route, it monitors the continued correct operation of that route. If the sender, the destination, or any of the other hosts named as hops along a route should fail or be turned off, or should move out of wireless transmission range of the next or previous hop along the route, the route can no longer be used to reach the destination. We call this monitoring of the correct operation of a route in use *route maintenance*. Route maintenance may use both *active* or *passive* acknowledgements. Active acknowledgements may use the hop-by-hop link-level acknowledgements already present in many wireless network MAC protocols, or may rely on a combination of existing transport or application acknowledgements or explicitly requested network-level acknowledgements. Passive acknowledgements [18] provide "free" hop-by-hop acknowledgements by using promiscuous receive mode in a host's network hardware to receive the transmission of the packet to the next hop from the host to which this host sent it on this hop; for example, in Figure 5, host A can generally hear B's transmission of the packet on to C. When route maintenance detects a problem with a route in use, the host detecting the error returns a *route error* packet

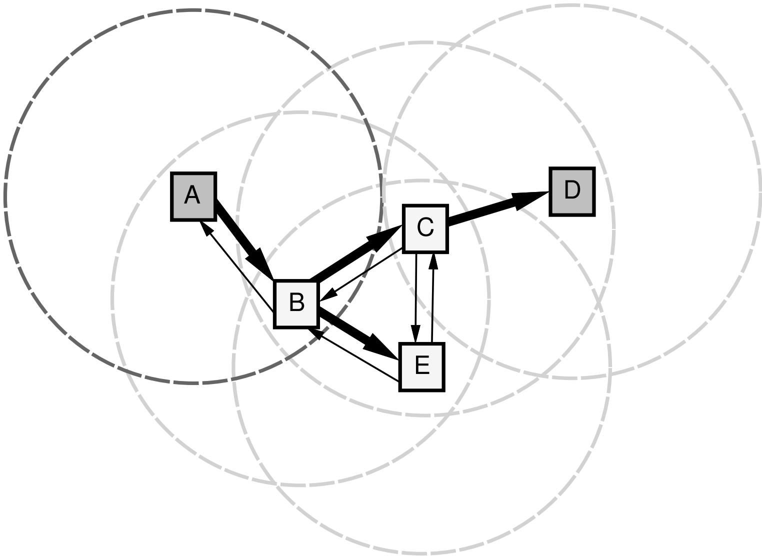

Figure 6 Operation of the route discovery protocol

to the original sender of the failed data packet, which then uses route discovery again to discover a new, correct route to the destination.

Simulation Results

We have not yet implemented our dynamic source routing protocol for routing in ad hoc wireless networks, but we have performed a detailed simulation study of its behavior and performance using a packet-level simulator [16]. In addition to a number of parameter choices in the protocol, the simulator allows us to vary certain environmental factors such as the number of mobile hosts, the pattern and speed of host movement, and the distribution of the hosts in space. Each host is initially placed at a random position within in the simulation area. During the simulation, a host pauses at its current position for a configurable period, and then chooses a new location and velocity with which to proceed there. Each host continues this behavior, alternately pausing and moving to a new location, for the duration of the simulation, appearing to wander through the simulation area with its restlessness determined by the configured pause time. During the simulation, hosts may originate up to three simultaneous conversations, with each conversation lasting for a randomly chosen number of packets sent at randomly chosen rate. For each transmission, the simulator includes a small probability (5%) of a transmission error due to wireless interference, and the Data Link layer in the simulation retransmits a packet up to 3 times before reporting a transmission failure to the Network layer. We executed 20 runs of the simulator for each of a number of different movement rates and numbers of mobile hosts in the simulated ad hoc network, with each run simulating over one hour of execution (4000 seconds).

Figure 7 shows the average total number of Network-layer transmissions performed relative to optimal, over the 20 runs. Here, the optimal number of transmissions is taken to be the minimum number necessary for each data packet to reach its destination, if perfect routing information were available and if no wireless transmission errors occur; the number of transmissions actually performed includes those necessary for route request, route reply, and route error packets, as well as those needed to forward data packets over the routes determined by the protocol. For all but the highest rates of host movement simulated, the overhead of the protocol is quite low, falling to just 1% of total data packets transmitted for moderate movement rates

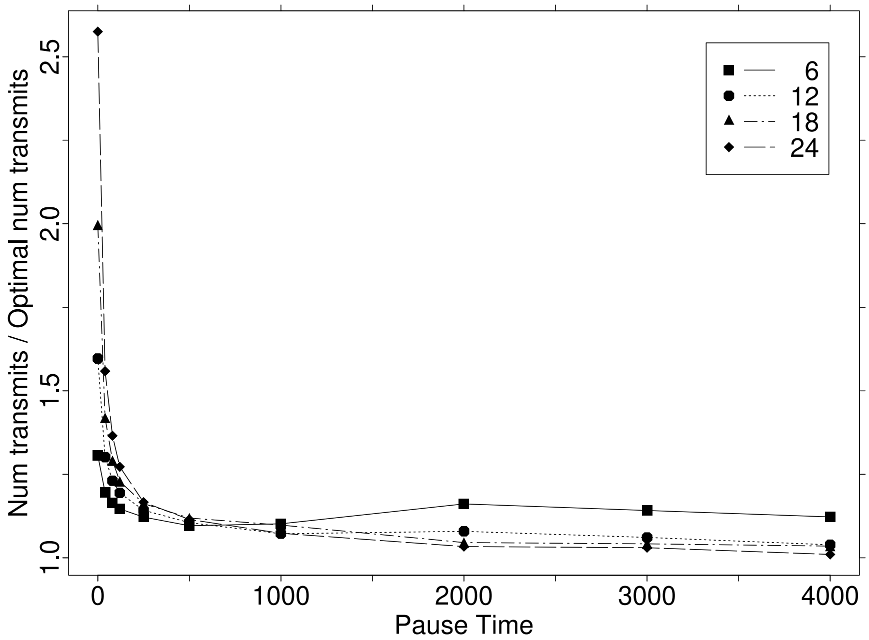

Figure 7 Average total number of transmissions performed relative to optimal

in a network of 24 mobile hosts. Figure 8 shows the length of routes used for forwarding data packets, relative to the optimal route length if perfect routing information were available. (The scale on the vertical axis on this graph differs significantly from that in Figure 7 in order to clearly show the relevant data.) In most cases, the average route lengths are within a factor of 1.01 of optimal, indicating the degree to which the protocol is able to track the mobile hosts as they move about.

Support for Adaptive Higher-Layer Protocols and Applications

Because of the wide variety of wireless networking hardware and services, there may be substantial changes in connection quality when a mobile host moves from one location to another, particularly if the best available network connection in the new location uses a different type of network than the old connection. For example, when moving from a high-speed wireless LAN to a wide-area wireless data service, bandwidth may decrease and latency may increase each by about two orders of magnitude. In addition, some changes in type of network may involve equally significant changes in factors such as transmission error rate and usage cost in bytes or packets transmitted or in connection time charges. In general, such differences are inherent in the wireless media, since each product or service must make certain tradeoffs between these factors in order to make best use of the limited electromagnetic spectrum shared among all users.

If protocols and applications on the mobile host and on correspondent hosts are able to learn of such changes in the mobile host's network connection quality, they may be able to adapt their behavior to the new conditions. For example, reliable transport protocols such as TCP could adjust their congestion control and recovery algorithms [3] and their timeout and retransmission strategies. At the application level, a disconnected file system such as Coda [20] may be able to make better decisions about when or how to reintegrate modified files, avoiding sending modified files back to the file server over a slow or expensive network connection [29]; or a program such as a World Wide Web browser or server may be able to dynamically alter the type or level of compression used in transferring images or video [22].

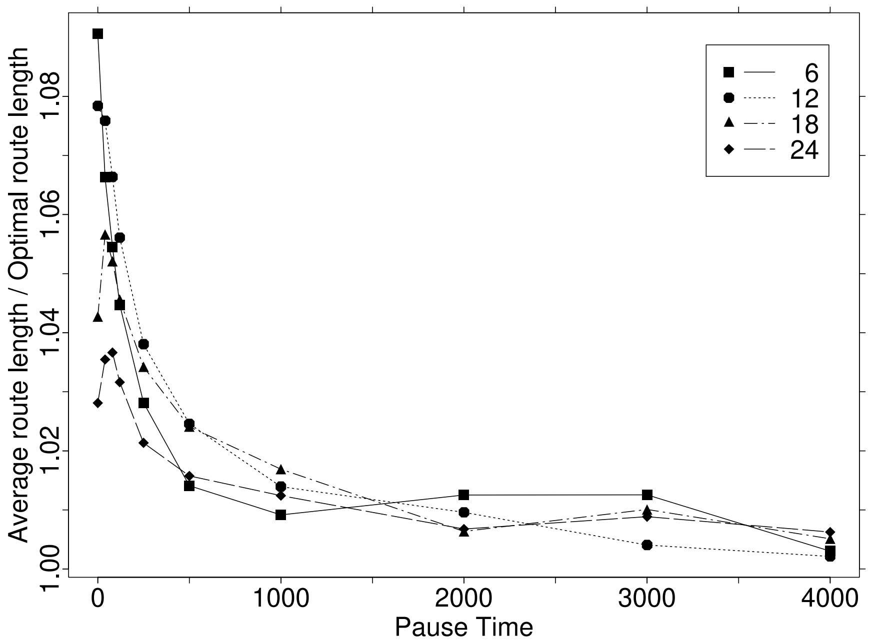

Figure 8 Average route length used relative to optimal

We have designed a preliminary protocol API and set of extensions to the IETF Mobile IP protocol to provide notification to mobile-aware protocols and applications on a mobile host, when the quality of that mobile host's connection changes as it moves from one location to another. In addition, these extensions support the dynamic extension of this notification to other hosts (and thus to the mobile-aware protocols and applications on those hosts) that communicate with the mobile host. The notification includes information on the bandwidth, latency, error rate, and service cost of the mobile host's current network connection. When a mobile host discovers and registers with a new foreign agent, it will obtain from the foreign agent an indication of the properties of the local network on which it is registering. As a part of the new registration on the mobile host, the Mobile IP software will cause an upcall [5] into each other protocol module or application that has registered interest in such changes. We have also extended the Route Optimization mechanism of the Mobile IP protocol to include notification of these connection quality changes along with the binding update message used to update a correspondent host's routing to the mobile host. When received by the correspondent host, these notifications will cause similar upcalls to notify mobile-aware protocols and applications on the correspondent host.

Changes in a mobile host's network connection quality may occur at times not associated with mobility, such as by gradual increases in congestion, but these types of changes are similar to those that occur even in wired networks of stationary hosts. In contrast, when switching to a new type of network, connection quality changes may be dramatic. Even when moving to a new location serviced by the same type of network as in the previous location, the local environment may be significantly different; for example, there may be many more users sharing the network in the new location, or there may be substantially different sources of interference present. By integrating detection and notification of these changes with the mechanism necessary to update the routing to the new location, we are able to perform this detection and notification with little or no cost. We are also currently exploring methods for combining this type of connection quality detection and notification with other approaches including periodic active monitoring of the network.

Related Work

Mobile Internetwork Routing

A general plan for mobile host routing on the Internet was first suggested by Sunshine and Postel in 1980 [30], although the first complete protocol designs did not appear until a decade later [11, 31]. Columbia University's "Mobile*IP" protocol is perhaps the most popular of these early protocols since an implementation of it is available, but the protocol provides only limited support for mobility outside a mobile host's home campus environment. Sony's VIP protocol provides global mobility, but is less compatible with the existing Internet infrastructure. VIP also supports a function similar to Route Optimization, although it included no facilities for authenticating cache updates. The caching support in VIP also is less scalable than in the Route Optimization extensions, since VIP attempts to cache the location of each mobile host at all intermediate routers between the sender and the mobile host's home network, including at backbone routers which could be handling traffic for many different mobile hosts.

The first version of the current form of the Route Optimization extensions appeared in our protocol using IP's loose source routing option [12]. Unlike IBM's protocol developed at the same time that also used IP loose source routing [26], though, our protocol used this IP option only as a tunneling mechanism and used separate control packets similar to the current binding update packets. This difference is important, since many existing implementations of the IP loose source routing option do not work correctly for more than the simple tunneling behavior required by our protocol. In later versions of our work, we developed an encapsulation protocol instead of using the loose source routing option, in order to include additional optimization and robustness functions in the protocol which the existing IP loose source routing option could not do [13, 15]; this encapsulation protocol is present in the current IETF Mobile IP protocol as the "minimal" encapsulation protocol. Use of encapsulation rather than the loose source routing option also avoids the significant performance degradation in the forwarding of packets containing IP options experienced by many IP router implementations.

Working together with Charles Perkins of IBM and Andrew Myles of Macquarie University, we later developed a new protocol containing many features of this protocol and including a simple form authentication that did not require key management or encryption [23]. A similar simple authentication mechanism was also used in recent mobile routing work done at Harvard University [1]. This scheme relies on a general property of routing in the Internet in which hosts or routers not connected to the normal routing path of a packet cannot eavesdrop on or reroute that packet. By including a randomly generated authenticator value in a packet sent to another node, the original sender can authenticate the reply from that node, by requiring that the same random value is returned in the reply. Although this simpler scheme requires no configuration of shared secret keys, it is less secure; this general property of Internet routing security has been severely weakened by increasing attacks in recent years, and any of the links over which such an authentication may take place may be wireless, enhancing the ability of any attacker to eavesdrop on the exchange containing the authenticator value.

Routing in Ad Hoc Wireless Networks

Routing in ad hoc networks was the subject of extensive study in the ARPA Packet Radio project [18]. Although dynamic source routing protocols were considered in this work, the protocols used were based on distance vector routing. The amateur radio community has also worked extensively with routing in wireless networks of (sometimes) mobile hosts [19], and originally used source routing with static, manually constructed routes. Although some had considered the possibility of a more dynamic source routing scheme, the routing functions were instead automated using a distance vector routing protocol known as NET/ROM [8]. The recent DSDV [25] protocol is an improved distance vector protocol for use in ad hoc networks, which uses sequence numbers in routing updates to prevent the formation of routing loops.

The general operation of the route discovery protocol is similar in part to that of the Internet's Address Resolution Protocol (ARP), except that ARP requests do not propagate from a router to its neighbors. The route discovery protocol is also similar to that used for finding source routes in source routing bridges in IEEE 802 LANs. However, in wired networks, a bridge can copy a request from one network interface onto each of its other interfaces and be sure that the request will propagate through the network in an orderly and complete way. In a wireless network, however, a router cannot transmit individually to only some of its neighbors, since all transmissions in a wireless network are broadcast; furthermore, since the hosts in an ad hoc network are mobile, a host cannot generally know the identity of all of its current neighbors.

In general, when hosts move quickly enough and frequently enough, the best strategy that any routing protocol can use is to flood data packets throughout the network in hopes that at least one will reach the mobile host. With distance vector routing, the routing overhead is essentially constant, whether or not hosts are moving. If hosts move more quickly than the routing protocol can converge to new routes, data packets will not be able to reach their intended destinations. With dynamic source routing, instead, there is little or no routing overhead (only route discoveries for hosts for which no route is yet cached) when host movement is very slow or infrequent. When movement rates increase, routing overhead correspondingly increases as new route discoveries are triggered by route maintenance; by performing new route discoveries as needed, though, data packets can continue to be correctly routed even during periods of frequent most movement.

Conclusion

We are currently completing our implementation of the Mobile IP protocol and our extensions to it, and will soon begin implementation of our dynamic source routing protocol for wireless ad hoc networks. We are also considering the interface between these two protocols to allow the interconnection of an ad hoc network with a wide-area network such as the Internet, reachable by some but not all of the ad hoc network hosts. The ad hoc network would essentially form a "cloud" around the foreign agent with which some of the of the ad hoc network hosts are also registered. The ad hoc network routing thus serves to extend the range of service of the foreign agent. We are also expanding our simulator to study other ad hoc network routing protocols, including those based on distance vector and link state protocols, and to study the performance and scalability of the Mobile IP protocol and extensions. We also plan to study a number of additional extensions to the Mobile IP protocol to further improve handoff speed and efficiency when moving to a new location, and to develop a protocol for internetwork routing of multicast packets to and from groups including mobile hosts, supporting efficient routing to each receiver and efficient updating of routing state to balance updating and routing costs for different host movement rates and multicast packet transmission rates.

The protocols described in this article support transparent movement of mobile hosts throughout the Internet, including dynamic switching between different types of network connections to utilize the best available network connection at any time. For example, a user's laptop computer may be connected to an Ethernet while in his or her office, but when disconnected and carried away, can dynamically and transparently switch to a high-speed wireless LAN connection such as through AT&T WaveLAN. When carried off campus or otherwise too far from a building equipped with WaveLAN, the mobile host can again switch transparently to a wide-area data service such as CDPD. When returning again within range of WaveLAN or when reconnecting to the Ethernet, the network connection can again dynamically switch. With each change in location or type of network, the routing of packets to the mobile host is dynamically adapted, and higher-layer protocols and applications on the mobile host and on other hosts communicating with the mobile host are able to adapt their behavior to the new network connection quality. When not connected to the Internet, mobile hosts can dynamically form ad hoc networks, with automatic routing of packets between hosts in the ad hoc network, utilizing other hosts in the ad hoc network to forward packets to the destination if necessary.

Host mobility and wireless networks require us to rethink design strategies and decisions at every level of the protocol hierarchy. This article has focused on our current work in the Monarch Project at Carnegie Mellon University in developing a set of protocols and protocol interfaces for supporting adaptive wireless and mobile networking support. With the proliferation of mobile computers such as laptops and and personal digital assistants, and with the increasing availability of wireless networking products and services, the need for this support is of great current practical importance. We expect this work to play a key role in building the mobile computing infrastructures of the future.

Acknowledgments

The Route Optimization extensions to the IETF Mobile IP protocol were jointly developed with Charles Perkins of IBM and Andrew Myles of Macquarie University, and we would like to express our thanks to them for their collaboration in our work within the IETF. The IETF Mobile IP protocol itself represents the work of many people within the Mobile IP Working Group; discussions with them have been of value in our research and have helped to shape the ideas presented in this article.

This research was supported in part by the Wireless Initiative of the Information Networking Institute at Carnegie Mellon University, by AEG Transportation Systems, and by the National Science Foundation under CAREER Award NCR-9502725. David Maltz was also supported in part by an IBM Cooperative Fellowship. The CDPD service used in this project is provided under a service grant from Bell Atlantic NYNEX Mobile.

References

[1] Trevor Blackwell, Kee Chan, Koling Chan, Thomas Charuhas, James Gwertzman, Brad Karp, H. T. Kung, W. David Li, Dong Lin, Robert Morris, Robert Polansky, Diane Tang, Cliff Young, and John Zao. Secure short-cut routing for Mobile IP. In *Proceedings of the USENIX Summer 1994 Technical Conference*, June 1994.

[2] Scott Bradner and Allison Mankin. The recommendation for the IP Next Generation protocol. Internet Request For Comments RFC 1752, January 1995.

[3] Ramón Cáceres and Liviu Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal on Selected Areas in Communications*, 13(5):850–857, June 1995.

[4] CDPD Consortium. Cellular Digital Packet Data system specification. Release 1.0, July 1993.

[5] David D. Clark. The structuring of systems using upcalls. In *Proceedings of the Tenth ACM Symposium on Operating Systems Principles*, pages 171–180, December 1985.

[6] Stephen E. Deering. ICMP router discovery messages. Internet Request For Comments RFC 1256, September 1991.

[7] Ralph Droms. Dynamic Host Configuration Protocol. Internet Request For Comments RFC 1541, October 1993.

[8] Daniel M. Frank. Transmission of IP datagrams over NET/ROM networks. In *ARRL Amateur Radio 7th Computer Networking Conference*, pages 65–70, October 1988.

[9] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): an address assignment and aggregation strategy. Internet Request For Comments RFC 1519, September 1993.

[10] Alex Hills and David B. Johnson. Wireless data network infrastructure at Carnegie Mellon University. *IEEE Personal Communications*, 3(1), February 1996.

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

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

[13] David B. Johnson. Ubiquitous mobile host internetworking. In *Proceedings of the Fourth Workshop on Workstation Operating Systems*, pages 85–90, October 1993.

[14] David B. Johnson. Routing in ad hoc networks of mobile hosts. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 158–163, December 1994.

[15] David B. Johnson. Scalable and robust internetwork routing for mobile hosts. In *Proceedings of the 14th International Conference on Distributed Computing Systems*, pages 2–11, June 1994.

[16] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth. Kluwer Academic Publishers, 1995. To appear.

[17] David B. Johnson and Charles Perkins. Route optimization in Mobile IP. Internet Draft, July 1995. Work in progress.

[18] John Jubin and Janet D. Tornow. The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75(1):21–32, January 1987.

[19] Philip R. Karn, Harold E. Price, and Robert J. Diersing. Packet radio in the amateur service. *IEEE Journal on Selected Areas in Communications*, SAC-3(3):431–439, May 1985.

[20] J. J. Kistler and M. Satyanarayanan. Disconnected operation in the Coda file system. *ACM Transactions on Computer Systems*, 10(1):3–25, February 1992.

[21] J. Mogul and J. Postel. Internet standard subnetting procedure. Internet Request For Comments RFC 950, August 1985.

[22] José M. F. Moura, Hirohisa Shiojiri, Jyh-Cherng Lin, and Radu S. Jasinchi. Video over wireless. *IEEE Personal Communications*, 3(1), February 1996.

[23] Andrew Myles, David B. Johnson, and Charles Perkins. A mobile host protocol supporting route optimization and authentication. *IEEE Journal on Selected Areas in Communications*, special issue on Mobile and Wireless Computing Networks, 13(5):839–849, June 1995.

[24] Charles Perkins, editor. IP mobility support. Internet Draft, August 1995. Work in progress.

[25] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, August 1994.

[26] Charles E. Perkins and Pravin Bhagwat. A mobile networking system based on Internet Protocol. *IEEE Personal Communications*, 1(1):32–41, First Quarter 1994.

[27] J. B. Postel, editor. Internet Protocol. Internet Request For Comments RFC 791, September 1981.

[28] Ronald L. Rivest. The MD5 message-digest algorithm. Internet Request For Comments RFC 1321, April 1992.

[29] Mahadev Satyanarayanan. Mobile access to information. *IEEE Personal Communications*, 3(1), February 1996.

[30] C. Sunshine and J. Postel. Addressing mobile hosts in the ARPA Internet environment. Internet Engineering Note IEN 135, March 1980.

[31] Fumio Teraoka, Yasuhiko Yokote, and Mario Tokoro. A network architecture providing host migration transparency. In *Proceedings of the SIGCOMM '91 Conference: Communications Architectures & Protocols*, pages 209–220, September 1991.

[32] Bruce Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Technical Journal*, 72(4):27–37, July/August 1993.

[33] Gary R. Wright and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Addison-Wesley, Reading, Massachusetts, 1995.

Biographies

DAVID B. JOHNSON is an Assistant Professor in the School of Computer Science at Carnegie Mellon University, and also holds a courtesy appointment as an Assistant Professor in the Electrical and Computer Engineering Department. His research interests include network protocols, distributed systems, and operating systems. He has worked actively within the Mobile IP Working Group of the Internet Engineering Task Force (IETF) for the past three years and is one of the principal designers of the current IETF Mobile IP protocol. Prior to joining the faculty at Carnegie Mellon in 1992, he was on the faculty at Rice University for three years as a Research Scientist and Lecturer in the Computer Science Department. He holds a B.A. in computer science and mathematical sciences, an M.S. in computer science, and a Ph.D. in computer science, all from Rice University. He is a member of the IEEE Computer Society, IEEE Communications Society, ACM, USENIX, Sigma Xi, and the Internet Society.

DAVID A. MALTZ is a Ph.D. student in Computer Science at Carnegie Mellon University studying mobile networking and computer supported collaborative work. He received the S.B. and S.M. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 1994. Prior to attending Carnegie Mellon, he interned at the Xerox Palo Alto Research Center and at Lotus Development Corporation. He is a member of ACM and Eta Kappa Nu.