Mobile-IP, Ad-Hoc Networking, and Nomadicity

Charles E. Perkins
Communication Systems
IBM T.J. Watson Research Center
Hawthorne, NY 10532
charliep@watson.ibm.com

Abstract

Continuing advances in wireless communications and feature integration into laptop computers are creating the need to allow nomadic users to make network connections to the Internet and with each other with no loss of functionality. This requires new advances in routing protocols and application interfaces. We use mobile-IP as the basis for providing mobility for nomadic users, and extend it to facilitate additional services for nomadic users both at the network layer and above. As an example of how mobile-IP can be useful with other protocols at the network layer, we show how mobile-IP can naturally extend the usefulness of ad-hoc networking. For higher level protocols, we show that, by offering a callback service, mobile-IP can enable applications to be responsive to changing network conditions. As an example of this technique, we show how service location can benefit from interaction with mobile-IP. We also mention some other nomadic services which will similarly benefit from a callback service from mobile-IP.

1 Introduction

Advances in wireless communications devices have made it possible to have graphics-based laptop devices fully operational with high-speed local area networks even while remaining completely tetherless. However, the necessary Internet Protocol networking software needed to make this physical connection transparent to network applications is only now becoming available. As the basic mobile networking software is widely deployed, other issues will come to the forefront. Problems which are not yet receiving much attention will become very irritating for people who rightly expect to continue their present-day computing practices, and who also know that their mobile computer is receiving all the data which is addressed to it from anywhere in the world. These nomadic computer users [5] will soon be a dominant force within the Internet community.

Unfortunately, the ability to deliver packets to a mobile computer (provided by mobile-IP [9]) does not solve the whole problem of nomadic computing. In fact, at each layer of the network protocol stack, new problems arise from historical assumptions about the traditionally stable nature of the network connections established by the end systems.

This paper discusses some of the other problems that arise after mobile-IP has provided the basic solution for routing of datagrams to mobile computers. Solutions of these new problems are shown to be possible which depend on extensions to the software on the client which implements mobile-IP. There are at least two ways the mobile-IP can interact with other modules implementing features important for nomadic users. One, mobile-IP can interact with other protocols which operate at the network layer to further extend their usefulness. As an example, we describe our work which allows a community of an ad-hoc network to make full use of any routes into the Internet.

Another way in which mobile-IP can support nomadic-aware applications is to offer them a callback service which is activated whenever the mobile computer changes its point of attachment to the internet. There are many possible applications which could derive great benefit from such a callback service.

In the following sections, we give a quick overview of mobile-IP. We then describe one technique for establishing an ad-hoc network, and show how mobile-IP can greatly extend the usefulness of the ad-hoc network.

The rest of the paper deals with showing the benefits of a callback service such as we have implemented with mobile-IP. The particular application we have been most concerned with is the Resource Discovery Protocol service [8], and a quick description of that protocol is given. It will be seen that such higher-level protocols and applications naturally have need for the callback service.

2 Overview of mobile-IP

This section contains a short overview of Mobile-IP so that the paper will be self-contained.

Mobile-IP enables mobile computers to move about the Internet but remain addressable via their *home network*. Each mobile computer has an IP address (a *home address*) on its home network. Datagrams arriving for the mobile computer at its home network are subsequently repackaged for delivery to the mobile computer at its *care-of address*.

The mobile computer informs its home agent about its current care-of address, using mobile-IP registration protocols. When the home agent receives the mobile computer's care-of address, it binds that address with the mobile computer's home address, forming a *binding* that has an associated lifetime of validity. This process is called *registration*, and is transacted between the mobile computer and the home agent each time the mobile computer changes its point of attachment and receives a new care-of address.

Often, the care-of address is advertised by an entity known as a *foreign agent*, which is located nearby the mobile computer and which relays the registration messages back and forth between the mobile node and the home agent. Other times, the mobile computer itself acquires a care-of address by other means (notably, via the Dynamic Host Configuration Protocol (DHCP) [4, 1]) and assign that care-of address to one of its own interfaces. Call this latter configuration a *co-located care-of address*. Note that a mobile computer can easily switch between the two modes of operation depending upon the way care-of addresses are provided at its various points of attachment.

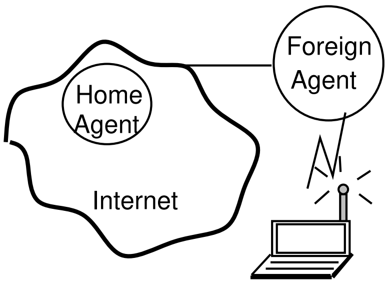

Figure 1: Mobile IP agents

Figure 1 shows a thumbnail sketch of a typical configuration, where the foreign agent has advertised the care-of address used by the mobile computer; the foreign agent and home agent are presumably and typically located on different subnets which have no *a priori* relationship to each other. If the mobile computer had attached via DHCP, there would be no foreign agent, but there would still be (typically) no relationship between the home network and the new point of attachment of the mobile computer.

When a home agent has a valid binding for the mobile node and a datagram for the mobile computer arrives at the home network, the home agent receives the datagram, acting as a proxy agent for the mobile computer on the home network. The home agent subsequently *tunnels* (by encapsulation) the datagram to the mobile computer's care-of address. The *tunnel* is the path between the home agent and the care-of address, and the care-of address is also known as the *tunnel endpoint*. After the datagram arrives at the tunnel endpoint, it is decapsulated and final delivery is made to the mobile computer. When the mobile node has a co-located care-of address, note that the final delivery is accomplished trivially.

Since traffic to the mobile node is controlled by correct operation of the mobile-IP registration protocol, it is of essential importance that no corruption or intentional modifications of registration message data go undetected. If a malicious agent were able to register its own IP address as a false care-of address for the mobile node, the home agent would then route all the datagrams for the mobile node to the malicious agent instead. Clearly, the home agent must be able to ascertain that registration messages were issued authentically by the mobile node itself. This is accomplished by affixing a 128-bit digital signature, computed by using MD5 as a one-way hash function, to the registration messages, and including protection against replay attacks.

3 Ad-Hoc Networking

People will soon become accustomed to taking their computers with them, and with the ever increasing benefits of ubiquitous computer networking. They will naturally expect to establish network communications with any other mobile computer which has a compatible wireless communications device, or which is accessible via any path consisting of existing network links, which themselves may be established over wireless network links.

Any internetwork which contains wireless links among its segments can be expected to have a dynamic and variable topology, as the wireless computers themselves move in and out of range of each other (as illustrated in Figure 2.

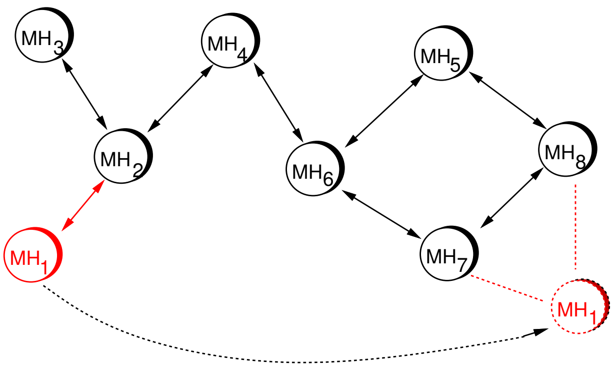

Figure 2: A Mobile Station in an Ad-Hoc Network

Unfortunately, there is not any widely deployed technology which will currently allow mobile computers to operate in such a dynamic topology of interconnection. There are routing protocols which go part way towards allowing such dynamic interconnection between computers, but they are relatively heavyweight, and need certain modifications to take into account the limited broadcast nature of wireless communications. Moreover, the Internet operates under certain assumptions of aggregation into "subnets" which are not really appropriate in the expected domain of operation.

Among the many applications for such *ad-hoc networking*, for instance when there is no Internet infrastructure available, we mention:

- emergency services
- battlefield operations

- on-the-spot conferencing and groupware
- community networking

To enable the style of freewheeling and dynamic interconnection between mobile computers, without any user intervention required, routing algorithms for ad-hoc networking have been developed (for instance, the Destination Sequenced Distance Vector (DSDV) [7] algorithm). These algorithms also enable multi-hop connectivity, so that a group of mobile computers can have complete connectivity even though some members of the group are not within range of each other.

The ad-hoc networking algorithms we use are loop-free Bellman-Ford (distance-vector) algorithms with refinements to eliminate the need for *poisoned-reverse, split-horizon* operation previously used with Bellman-Ford. Nevertheless, our approach does not suffer the problems earlier ascribed to Bellman-Ford algorithms. DSDV avoid problems by use of assigning a sequence number to each route, selected by the destination of the route.

4 Mobile-IP and Ad-Hoc Networking

It is quite reasonable to expect that any one of the mobile computers that has a connection to the Internet at large should be able to share that connectivity to its ad-hoc partners which themselves have no direct Internet connections otherwise.

For instance, suppose that a crew of emergency services personnel arrange to keep in contact with each other via radio links, using an ad-hoc network. Some of the workers may be able to maintain radio contact with a local base station, but others will have to go out of range in order to perform needed services. It would be very helpful if the outlying stations could still automatically maintain networked contact with the Internet at large, as if they were registered directly with the base station using mobile-IP. A similar situation obviously applies to battlefield situations.

At each mobile station, Internet connectivity is to be managed by mobile-IP; however, ad-hoc networking was not a consideration during the specification of mobile-IP. It turns out that there are several areas of minor conflict.

The first problem we faced was that, according to the mobile-IP specification, a mobile node is assumed to have a direct link to its foreign agent (we do not consider in this paper how to support ad-hoc networking without foreign agents). However, the nodes in the ad-hoc network which are not adjacent to a foreign agent will all need to register with their home agents if they are to use mobile-IP. A (possibly remote) foreign agent offers the only real care-of address which could be useful for the nodes in the ad-hoc network. We extended the mobile-IP model to allow mobile nodes to use care-of addresses even if they were more than one hop away in the ad-hoc network. In our implementation, there are two effects of this:

- default routers are allowed to be more than one hop away, and
- foreign agents are allowed to use an ad-hoc route to a mobile client instead of merely forwarding packets onto a directly connected network link.

The other problem that we had to solve was to find a way to let both mobile-IP and the ad-hoc networking algorithms affect the management of IP's routing table. Resolving the conflict proved to be non-trivial; we eventually implemented a *route manager* with certain precedence rules to arbitrate the simultaneous needs of the two routing protocols.

5 Enabling Nomadicity in Higher Level Protocols

We have enhanced our mobile-IP dæmon with additional features. Important among them is a *callback* feature, which allows application to post requests for the mobile-IP dæmon to inform them when a change occurs in the mobile node's point of attachment to the Internet. We believe that this simple idea will go a long way towards enabling applications to respond to the needs of the nomadic computer user. A similar idea was discussed in [3]; in this paper, it was proposed that the operating system deliver a signal called `SIG_CELLSWITCH` to applications. However, the callback feature is equally flexible, and requires no additional operating system support. In our implementation, applications use the same port number (434) as specified for use with other mobile-IP registration messages, albeit with a different message-ID.

We propose that this is a useful feature, because whenever a mobile node reattaches to the network, a number of important environmental factors may have changed that would be important to applications that might be running on the mobile computer. For instance, the cost of the connection could increase, so that information that was previously cost-effective is no longer wanted. Alternatively, as often discussed (see, for example, [12]), the available bandwidth should be be monitored so that information can be filtered, compressed, or otherwise massaged to obtain acceptable performance across the possibly constrained communications medium.

In our implementation, we include the care-of address (at the mobile node's new point of attachment) as part of the callback notification. In the future, mobile nodes may have multiple heterogeneous points of attachment to the Internet, each with different characteristics. In fact, each interface may have a different apparent rate at which it changes its point of attachment to the Internet. For this reason, and for ease of programming, we felt the care-of address would often be useful for nomadic-aware applications.

It is likely that many nomadic-aware applications will eventually require finer-grained information regarding the nature of the mobile node's attachment to the Internet than would be provided by mobile-IP. Indeed, wireless media can offer variable bandwidth even while the connection remains intact; it is possible to trade bandwidth for reliability and experience overall increases in throughput, because higher reliability means fewer retransmissions. Nevertheless, for a wide class of applications and network interface adapters it is good enough just to discover when the mobile node's care-of address changes. A good example of such a higher-level protocol is the Resource Discovery Protocol [8], as described in the next section.

6 Resource Discovery Protocol

Clearly, once mobile computers change their point of attachment, they may be faced with a new and unknown environment in which locating necessary network resources is currently not easy to do. For instance, a mobile computer will not be able to rely on static configuration to find out how to send a file over the network to a printer, or fax machine. The mobile computer will, instead, need to have access to services by some generic access method, if possible mediated by local directory services.

Drawing from the experiences of Appletalk users, and incorporating recent technology from the World-Wide Web, we have built such a Resource Discovery Protocol which is suitable for use by mobile computers.

Our Resource Discovery Protocol (RDP) allows services to register with a local Directory Agent, and allows user agents (say, residing on mobile computers) to contact the Directory Agent to obtain Uniform Resource Locators (URLs) [2] which are pointers to the desired services. By taking advantage of URL technology, developed for use with the Web and able to be interpreted by Web browsers, we find, as one might expect, that it is easy to locate the desired resources. For each URL, the Directory Agent stores attributes and keywords (e.g., PostScript, blue_paper) which allow clients to select exactly which service provider meets their current needs.

RDP requires for its operation that clients be able to *name* the services they want, and specify the details about that service. We have borrowed, also from World-Wide Web technology, the concept of a Uniform Resource Name (URN) [11]. RDP, then, allows services to register URLs, and allows clients to retrieve the URLs by submitting requests to the Directory Agents containing the URNs which describe the desired services.

Suppose a mobile station wishes to discovery a local printer using RDP. Figure 3 shows a typical scenario for a wireless mobile station. Briefly, the station sends a URN query for local postscript printer, and use the returned printer URL in the command which sends data to be printed.

- The wireless mobile client first gets access to a nearby Directory Agent by sending DHCP option 11 (for Resource Location Server (RLS)) to a DHCP server (usually via a DHCP relay, and physically by way of a wireless access point).

- The DHCP server returns the IP address of the Directory Agent (muffin.watson.ibm.com), which may be on another subnet.

- The mobile client contacts `muffin` to get the address of the local printer service. The URN (namely, `n2l:///printer/local/postscript`) is delivered to `muffin` (the Directory Agent).

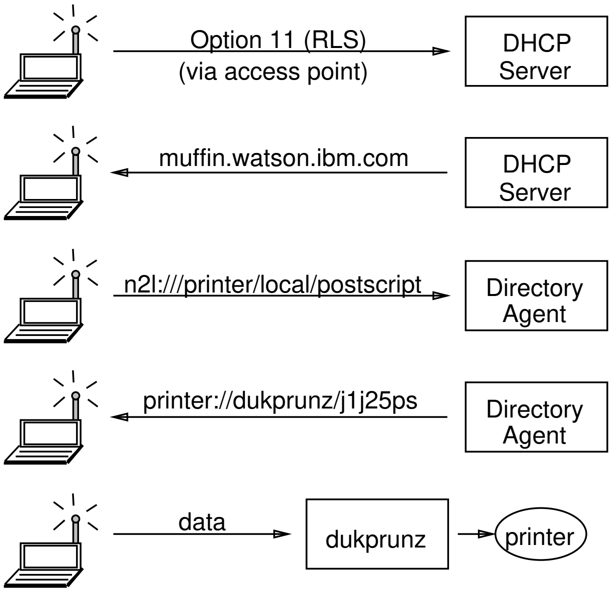

Figure 3: Using RDP to find a Printer

- `muffin` returns `printer://dukprunz/j1j25ps` (the printer URL) to the mobile client.

- Finally, the client is able to access the printer by specifying the correct destination printer address along with the print command.

Applications running on mobile computers often need to discover when their last URLs are no longer valid. With the callback mechanism, the means to do so is straightforward. Each nomadic-aware application that needs to perform resource discovery simply posts a callback request to the mobile-IP dæmon. When the mobile computer moves, each such client is awakened with the notification, and the information about the new care-of address. With that stimulus, the client can then send out new requests to find the local Directory Agent, and subsequently obtain the needed URLs from the Directory Agent. Note that, in this case, the care-of address (from the move notification issued by the mobile-IP dæmon) is not needed in order for the Resource Discovery protocol to work. Also, note that upon each movement by the mobile station, network services may have to be resolved again, even if the Directory Agent remains the same for the new point of attachment.

7 Other High-Level Nomadic Services

Many other nomadic-aware applications will undoubtedly find natural uses for the callback mechanism described above. In fact, any *proxy* services [6] employed by mobile clients may need to be re-established by the client each time it changes its point of attachment. By the same token, however, any nomadic-aware applications that depend on the *fidelity* [10] of the network link will certainly need to reevaluate the important parameters at each new point of attachment.

Other general services are likely to become important, and can benefit from the callback service of mobile-IP. An example would be a generalized user profile service. As a user moves from place to place, it might be desirable to activate different parts of the user's profile. The profile would have stanzas for security classification of information, electronic mail topics which should be considered important depending upon the circumstances, and many other characteristics which we will learn to identify. The correct stanza will be selected depending upon several factors, among them the type of network connectivity currently in use.

For example, a mobile user in a car at lunchtime will usually be more interested in restaurants than a mobile user in an office at mid-morning. In the former situation, a nomadic-aware application for selecting restaurants will want to know whether the mobile user is in a car, the current time, where the car is geographically located, and what kind of food the user prefers. There might be many different nutritional stanzas in the user profile, and selecting the right one will depend on the details of the user's network connectivity. This information will be crucial to the operation of any program filtering out broadcast data (e.g., advertisements) received by the mobile user.

In the near term, however, most profile choices will depend on a few obvious characteristics, for instance:

- whether the user is at home, at work, or travelling,
- whether the network connection is secure or susceptible to loss of privacy,
- link speed,
- time of day.

In each case, choice of profile elements can be assisted by comparing the care-of address of the mobile computer with ranges of subnet addresses associated with various user environments. Thus, nomadic-aware applications utilizing such profile elements to custom-tailor their responses to the user's environment will all benefit from the callback mechanisms when they are available with mobile-IP.

Other modular structures are certainly possible. For instance, a "link-adaptation" service could implement its own call-back services to applications. This server could rely on mobile-IP for information about changes in point of attachment to the Internet, and it could additionally monitor continuously changing network conditions by periodically polling the network interfaces. Network interface drivers could also implement high-water or low-water marks for a variety of parameters related to network connectivity, and these drivers would then issue alarms to applications or adaptation servers when the parameters exceed their specified boundary values.

8 Summary and Conclusions

Mobile-IP has opened the doors for a new computing paradigm called *nomadicity* which supports the convenience of mobile computing with the enormous power of the Internet. We are just now beginning to understand the problems that nomadic, mobile users will face when they attempt to remain equally productive regardless of their point of attachment to the network. In that paper, we have presented only an overview of the kinds of software needed by mobile computer users. Protocols, including mobile-IP, DSDV, and a Resource Discovery Protocol, and a new procedure (a callback procedure) are described which help solve the new problems encountered by mobile, nomadic users.

References

[1] S. Alexander and R. Droms. DHCP Options and BOOTP Vendor Extensions. RFC 1533, October 1993.

[2] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). RFC 1738, December 1994.

[3] P. Bhagwat, C. Perkins, and S. Tripathi. Transparent Resource Discovery for Mobile Computers. In *Proceedings of IEEE Mobile Computing Systems and Applications*, dec 1995.

[4] Ralph Droms. Dynamic Host Configuration Protocol. RFC 1541, October 1993.

[5] L. Kleinrock. Nomadic Computing - An Opportunity. *ACM SIGCOMM, Computer Communications Review*, 25(1):36–40, January 1995.

[6] C. Perkins. Handling Multimedia Data for Mobile Computers. In *Proceedings of the Twentieth IEEE Computer Software and Applications Conference*, August 1996.

[7] Charles Perkins and Pravin Bhagwat. Routing over Multi-hop Wireless Network of Mobile Computers. *SIGCOMM '94 : Computer Communications Review*, 24(4):234–244, October 1994.

[8] Charles Perkins and Harry Harjono. Resource Discovery Protocol for Mobile Computing. In *Proceedings of IFIP '96 14th World Congress*, September 1996.

[9] C. Perkins, Editor. ietf-draft-mobileip-protocol-16.txt - work in progress. IPv4 Mobility Support, April 1996.

[10] M. Satyanarayanan. Mobile Information Access. *IEEE Personal Communications*, 3(1):26–33, February 1996.

[11] K. Shafer, E. Miller, V. Tkac, and S. Weibel. URN Services. draft-shafer-uri-urn-resolution-00.txt - work in progress, June 1995.

[12] B. Zenel and D. Duchamp. Intelligent Communication Filtering for Limited Bandwidth Environments. In *Proceedings Fifth IEEE Workshop on Hot Topics in Operating Systems*, pages 28–34, May 1995.