Mobile networking in the Internet

Charles E. Perkins

Sun Microsystems, Inc., 15 Network Circle, Menlo Park, CA 94025, USA

Computers capable of attaching to the Internet from many places are likely to grow in popularity until they dominate the population of the Internet. Consequently, protocol research has shifted into high gear to develop appropriate network protocols for supporting mobility. This introductory article attempts to outline some of the many promising and interesting research directions. The papers in this special issue indicate the diversity of viewpoints within the research community, and it is part of the purpose of this introduction to frame their place within the overall research area.

1. Introduction

This issue of Mobile Networking and Applications presents research papers probing the effects of mobility on the Internet. As one might expect, given the diverse nature of protocols employed by Internet addressable devices, there are a wide range of effects. In fact, there are so many different aspects to Internet mobility that no single journal issue or book could possibly describe all of them. Thus, we will have to be content with presenting a representative selection of articles that, in their diversity, give a good hint at the larger picture. At the same time, these articles provide new directions and lead the way towards solving the interesting and new problems raised by mobility.

It is the purpose of this introductory article to briefly mention a larger cross-section of the fresh ideas and proposals for solutions of the problems raised by mobile networking, than could be represented by articles for publication in this journal issue. Thus, this paper will touch on current topics in many areas of networking. From cryptography to routing, from billing to expanded techniques for automatic configuration, mobility changes the way we think about computing, and invalidates some of the design assumptions upon which current network protocols and products have been built.

The impetus for all this change is the burgeoning market for mobile and portable computers and computing devices [62]. Besides the growing number of laptop computers, there are numerous other devices gaining popularity, so-called personal digital assistants (PDAs) that can handle messaging, calendars, personal information management, reminders, and address book and telephone directory functions. The role for such devices seems certain to grow as more computing power and communications capability can be included.

Wireless communications has been another growth area affecting the system design of mobile computers [15]. From the beginnings of the Internet, protocol designers have been fascinated by the attractions of wireless communications [30], but the lack of bandwidth and the expense of the equipment has prevented any widespread deployment. As increased bandwidth becomes available and more information resources become available by way of the Internet, the push for inclusion of wireless capabilities in laptop computers will become unstoppable. Adding travel computers to automobiles will provide new opportunities for making productive use of the Internet, as well as enabling new applications for increased road safety. If the appropriate wireless signposts are added to the automotive transportation infrastructure, wireless Internet computing could possibly help realize the age-old dream of automatic piloting on long car trips. Reports of road closures and traffic congestion, or even food preferences, could be automatically taken into account, when the automatic pilot is planning the best routes.

Before those dreams come true, however, a lot of work has to be done. Developing the new network protocols is the theme of this special issue, and this article intends to provide an overview of the variety of network protocols and associated technologies at all levels that must be considered when providing solutions for mobile computer users.

2. Overview

In this paper, we organize the description of mobile networking generally according to a classical layered model of network functions [14]. Each layer, from physical to application, is affected in various ways in the new operating environments encountered by mobile computer users. Although the reduced size and weight of mobile computers has some effect on their system architecture, these effects are not dominant because of the terrific advances in system miniaturization, display technologies, and communications. There are many mobile computers envisioned that will not have hard disks, and many without keyboards, but these more restricted devices are not principal drivers for the mobile networking techniques explored in this special issue, or within this paper. Conversely, many of the techniques and protocols developed for more general purpose mobile computers can be adapted as needed for the special or restricted case. A good model, therefore, for the kinds of mobile computers under consideration is a laptop computer with sufficient disk storage and any of a variety of network interfaces. The variation in the capabilities of the communication devices is one of the main differentiators between mobile computers.

Besides minimal weight and size, there are other hardware implications when designing for mobile computing. Clearly, battery powered operation is highly desirable, and improvements in battery life continue to extend the feasibility of tetherless computing. On the other hand, the proliferation of mobile laptop computers is driving the creation of friendlier computing environments, to attract the professionals who are among the people most likely to own and operate them. Advances in operating system design for intermittently powered I/O devices are being made in order to further reduce power demands and extend battery life.

Wireless and mobility are not the same, but they are features which are quite synergistic. It is possible to have wireless computers that do not move, just as it is possible to move wired computers from place to place. Clearly, however, the possibility for wireless data communications creates an irresistible urge to find ways to support mobility and network access at the same time. As a rule, wireless dominates the design space at the lower levels in the context of mobile computing, because at the lower levels the differences between physical media are most visible. At the network layer and above, mobility dominates. These design parameters require variability in essential protocol elements in ways not envisioned by the designers of existing network protocols.

As mobile computers become smaller and cheaper, it becomes more feasible to use them as commodity devices without any personality, much as one might treat a pad of paper. In this scenario, it becomes important to temporarily allow the notepad computer to operate on behalf of the user, and to have all the authorization proper for that user. This can be easily done by allowing the mobile computer to acquire authorization rights and capabilities from information encoded on a smart card owned by the user. It's easier and more convenient to carry around a smart card in one's wallet or purse, as long as a suitable computer is available when needed that can acquire the rights and privileges of the cardholder.

3. Physical layer considerations

At the physical layer, the main objective is to detect the signals between the two endpoints of a communications link. While physical layer considerations are among the most interesting, they do not form the focus of this article or journal issue. Many different media and channel coding schemes have been proposed, for instance:

- Directional infrared;
- Diffuse infrared;
- Analog cellular telephone;
- TDMA;
- CDMA;
- Short range radio.

There are a number of variations for each of the above channel types.

For the purposes of higher-level protocols, each channel encoding scheme can just as well be considered as a new physical medium. Operations within the lowest protocol layers serve the function of manipulation of various interface registers to set up the physical layer encoding and channelization.

The new wireless media becoming available are among the primary drivers for the interest in mobile computing. Thus, it is appropriate to understand the nature of wireless communications, and the contrast between wireless and wired media.

For *wired* media, there is typically:

- Well defined broadcast range;
- Low bit error rate;
- High bandwidth;
- Symmetric connectivity.

For *wireless* media, there is typically:

- Point-to-point communication only, or vague and poorly controllable boundaries for broadcast range;
- Variable (time and distance dependent) bit error rate;
- Low to medium bandwidth;
- Possibly asymmetric connectivity.

These characteristics make protocol design for wireless communications systems challenging. For instance, one result of the way wireless broadcast works (when it is available at all) is that eavesdropping is more difficult to detect and prevent.

4. Link layer considerations

A great deal of attention has been paid to methods for establishing links between mobile computers and base stations or access points. One typical method is the creation of telephone links; the popularity of this method rests largely on the widespread availability of the physical media which can be used. Cellular telephones using various technologies can provide good coverage within the United States, parts of Asia, and Europe, although no single technology so far provides sufficient breadth of coverage.

The following operations are among those sometimes included at the link layer:

- Handoffs;
- Compression;
- Encryption;
- Elimination of the *hidden terminal* problem;

- Retransmission of garbled data;
- Power control;
- Neighbor discovery;
- Address resolution;
- Adaptive error correction.

The next subsections describe these and their importance for wireless communications systems.

4.1. Handoffs

Central to the concept of seamless mobility is the process of establishing links at each new connection point. Whenever this process requires the transfer of state information from the old connection point (e.g., base station) to the new one, a *handoff* has to occur. There are numerous methods for performing handoffs, as numerous as the kinds of state information that has been designed for mobile nodes, as well as the kinds of network entities that maintain the state information. Often, authentication has to be performed to ascertain the identity of the mobile node.

4.2. Compression

Compression is often desirable because it reduces bandwidth requirements, and that can be very important for many low-speed wireless media. However, use of compression at the link layer is problematic in some circumstances, because the best compression is almost always achievable at higher level protocol levels, especially the application layer – for instance, using techniques described in Web*Express*, described in this issue. Compared to the link layer, the application is much more likely to be able to aggregate larger data objects into an efficient coding scheme, because the link layer only has access to the bit stream, not to the sequence of data objects being transmitted.

Unfortunately, attempting compression at two different protocol levels is typically less efficient than performing it at only level, and can have the effect of increasing the amount of data to be transmitted. Consequently, whenever higher-level protocols use encryption, the link layer should be inhibited from attempting any further compression. The dichotomy between the need for use of compression for naive applications, and the need to inhibit compression at the link layer for more intelligent applications, indicates that any lower-level compression features must be controllable by higher-level protocols. This is only one of several situations where link layer operations must be made visible (perhaps on a packet-by-packet basis) to higher level protocols. As a result, more sophisticated control strategies are often needed for use by higher level protocols.

4.3. Security

Whereas the constrained bandwidth of wireless technologies suggests the use of compression, it is the open propagation of wireless signals throughout the range of the transmitter that suggests applying security techniques to the wireless signal before transmission. A number of encrypting link layer devices and products have been introduced, especially for use in military applications. Link-layer security introduces further requirements for control of features by applications for reasons entirely different than were important for controlling the use of compression. Among the link layer parameters that may need to be specified or controlled are:

- Whether security features are to be used at all;
- The key to be used for encryption;
- The encryption algorithm (and mode);
- Whether the data must be encrypted for privacy, or merely authenticated.

The use of security features at the link layer has the effect of requiring additional processing, which uses more power and which can significantly degrade transmission speed on high-speed wireless links. Even when the transmitter can handle encryption at high speeds, the receiver must decrypt at the same speed. Fortunately, there are encryption algorithms [52] which allow relatively speedy decryption by the mobile wireless receiver, which may have limited power or processing capabilities.

4.4. Hidden terminals

The overall wireless bandwidth available to all mobile computer users can be improved by increasing the number of cells (where a cell is considered to be the range of coverage of a base station connecting the mobile node to the rest of the network), reusing the frequencies in each cell, and reducing the number of mobile computers per cell. Reducing the number of mobile computers per cell is typically accomplished by making the cells smaller, so that the access points in the cell are within range of fewer wireless computers. The way that frequencies are used in each cell has to be managed carefully so that neighboring cells do not interfere with each other.

Having multiple computers in a cell can give rise to the *hidden terminal* problem [6] illustrated in figure 1, a difficulty encountered in the use of wireless communications. In the figure, two laptop computers with radio links to an access point AP (say, a base station) may try to communicate with the access point simultaneously. Each computer can hear the access point, and cannot directly detect any interference on the wireless medium. Nevertheless, the access point will likely be unable to receive the transmission from either laptop.

A number of solutions to this problem have been proposed and developed, including MACA [26], W/MACA [26], IEEE 802.11 [26], and FAMA [26]. Typically, a sender asks to transmit its data (e.g., by transmitting a RTS, or *request to send*), and then waits until the intended receiver grants permission (e.g., CTS, or *clear to send*). The intended receiver, then, does not issue the CTS while it is receiving data from some other sender. It is better to have a separate channel for transmitting RTS that will not interfere with data packets. In any case, the sender should make

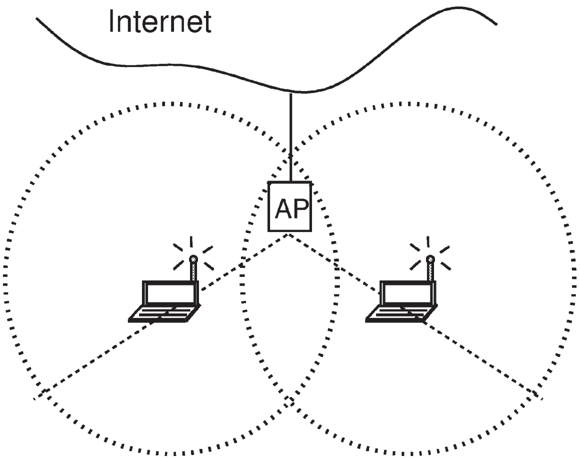


Figure 1. Hidden terminal problem.

sure not to send RTS packets too often, compared with reasonable transmission times for data packets that might be coming to the intended receiver from other transmitters.

4.5. Retransmission

As mentioned above, wireless media often find application in situations where error-free transmission cannot be guaranteed. Problems arise when the wireless stations begin to move apart from each other, introducing fading effects as the received signal power decreases. When signal power becomes about the same as power in the channel from other sources (co-channel interference), data errors occur. Noise can overwhelm the received signal power for other reasons. For instance, a wireless receiver can move through an area which has some obstacle preventing the reception of signal from a transmitter, but soon afterwards might emerge from behind the obstacle. Alternatively, a noise source can traverse the area between the transmitter and the receiver. All of these can disrupt the flow of data between wireless nodes, and cause failures at higher level protocols. In order to combat temporary corruption or loss of signal, and the consequent bit errors detected by wireless receivers, the link layer can be designed to supply acknowledgements for packets received, or to transmit requests for retransmissions when packet losses are detected. If left for correction by higher-level protocols, the delays introduced by timeouts and increased processing requirements cause substantial degradation of the performance of the wireless link, which can even affect the user's satisfaction with interactive applications. Link-layer retransmission can occur on a time scale shorter than what is possible using retransmission schemes in higher level protocols (e.g., TCP), and thus reduced latency for the data stream will be experienced.

Whether or not this is a good idea depends on the additional bandwidth required to transmit the sequence and identification information for each packet at the link layer, as well as the complex interaction between the link layer and retransmissions performed at higher levels. Additional complications arise when the channel is multiplexed for multiple use. For instance, when voice and data are carried on the same channel, TCP can experience retransmission anomalies, as analyzed in detail in the paper by Sudhir Ramakrishna et al. in this issue. Hybrid schemes are possible whereby high-level detection of data loss or corruption can trigger the utilization of retransmission modes by the link layer protocol.

4.6. Neighbor discovery

Central to any link layer operation is the process of *neighbor discovery*, by which a wireless node may determine which other nodes are within range of transmissions made using the particular physical medium and/or channel of interest. Sometimes a particular kind of neighbor is required, such as a base station, and in those cases the neighbor discovery mechanism must take into account marker information included in advertisements from the distinguished neighbor. In other cases, all neighbors might be of interest to the node, and topological connectivity information will be exchanged between the neighboring nodes.

Neighborhood information can be dynamic, changing as the nodes move relative to each other. Thus, neighbor discovery algorithms typically operate periodically, and the period (rate of repetition) defines in some sense the responsivity of the collection of nodes. The period should be chosen so that the neighborhood typically undergoes only incremental change during the time of a single repetition. When the rate of motion is too great to sustain the control traffic needed for neighbor discovery, it probably no longer makes much sense to model the local physical medium connecting two nodes as a link to be established for future communication. Instead, the physical medium becomes essentially a way to relay broadcast messages, and *flooding* is then the only way to get data to a particular destination.

The link layer neighbor discovery algorithms may also maintain network-layer (e.g., IP) address information for later use. When such address information is cached for better performance, node mobility has to be reckoned with since the cached information can become stale [48]. At the link layer, cached information which identifies a neighbor node becomes invalid when the node is no longer in the neighborhood. Additional protocol operations may be needed to cover for the node during the time it is away from the neighborhood, if indeed the node is ever expected to return.

4.7. Power control

As has been noted, additional bandwidth can be made available to mobile nodes if the same wireless medium can be used simultaneously by units which are out of physical range of each other (e.g., by making the cell size smaller). Maximizing the availability of a medium by spatial re-use, then, is an important consideration in wireless system design. The power used to transmit a wireless signal is typically the dominant factor in determining the range for its reception. Consequently, wireless systems should control the amount of power used to transmit their data. In addition to increasing frequency re-use, reducing power may increase battery life. On the other hand, reduced range for signals from mobile nodes increases the probability of loss of signal.

Link layer protocols can control the transmission power used by wireless communications adapters to balance these two needs. If, for instance, a mobile node determines the amount of power needed to contact an essential subset of its neighboring nodes, it can set its power level accordingly. Determining the essential subset is, of course, not always very easy. In some cases, particularly when clustering algorithms are used in ad hoc networking [7,20,21,28], some simplifying assumptions are made so that the process is more manageable.

4.8. Error correction

As error conditions on a wireless link get better or worse, the number of bits employed for error correction could be decreased or increased to enable error-free reception. When the bit-error rate is relatively high, it is better to enable errors to be fixed directly rather than requesting retransmission of packets as discussed above. However, predicting the number of error-correction bits needed to assure error-free reception is not easy. Overestimation wastes a significant fraction of the bandwidth on correction bits that are never used, and underestimation causes more retransmissions to be necessary. Even so, when the error rate is somewhat predictable, this technique often effectively improves the data rate available over wireless links.

5. Network layer considerations

The Internet Protocol (IP) [50] offers a convenient design point for introducing the necessary protocol operations for supporting node mobility. By now, the network layer operations for mobility support are well understood, and are specified in Mobile IP (RFC 2002 [46,48]), a freely available standard. To understand Mobile IP, it is first necessary to understand IP. For the purposes of this paper, and the other papers in this special issue, IP may be considered to offer the following functions:

- Identifying each network;
- Identifying each node on a network;
- Forwarding packets to the correct next hop when they arrive at an intermediate node (router) which is not the final destination;
- Fragmentation and reassembly as needed;
- Triggering mechanisms for resolving IP addresses into lower-level (link layer, or MAC) addresses;
- Generating appropriate control and status information for handling exceptional link conditions.

For the purposes of handling node mobility, the forwarding function is the main thing in IP that needs change. Minor modifications are also needed to the means by which error information is propagated through the network. Lastly, resolving the IP address of a mobile node to a MAC address by way of ARP (the Address Resolution Protocol [49]) presents a delicate design problem because ARP results are usually cached, and the cached information goes stale as soon as the mobile node moves away.

IP traditionally makes next-hop decisions based solely on the IP address of the destination; these decisions are not necessarily affected by the mobility of the source of the packet. To support the mobility of the destination node using Mobile IP, IP is modified to tunnel packets to a mobile node at its current point of attachment to the Internet, as part of the forwarding process. By this mechanism, packets arriving at the mobile node's *home agent* are then no longer confined to the network identified by the mobile node's IP address. The new and important additions to IP for handling node mobility all revolve around the *care-of address,* which is the IP address used to identify the mobile node's current point of attachment (not the mobile node itself). The care-of address does not affect the IP address used to identify the mobile node to the rest of the Internet.

Mobile IP can be understood as three interrelated operations involving the care-of address:

- Advertising it at the new point of attachment;
- Registration, or storing it for future use at the mobile node's home agent;
- Use by the home agent, to tunnel data traffic from the home network to the network indicated by the care-of address.

The association between a mobile node's IP address and the care-of address it acquires as it moves about is known as a *binding*; the binding carries along with it information specifying how long the association is allowed to be considered valid.

The papers by Chikarmane et al., and Montenegro and Gupta in this issue provide additional background on Mobile IP. Other good references may be found in [8,40,45, 46,54].

5.1. Reducing registration frequency

One criticism that has been lodged against the base Mobile IP protocol is the need for possibly frequent reregistration as the mobile node moves about from place to place. Such reregistrations can cause dropped packets if the mobile node is far away from its home network and route optimization is not in use by the foreign agents. Moreover, in the situation where, say, thousands of mobile nodes are reregistering upon emergence from a densely traveled major tunnel for automobile traffic, the control traffic from the registration protocol may overwhelm local resources.

For this reason, there has been interest in finding ways to enable local processing for Mobile IP messages by the foreign agents, to reduce network traffic, possibly the number of registration attempts, the registration time, and consequently the time for which pending registration state information has to be maintained. One method is to allow the mobile node to use a multicast address for its care-of address. Then, any foreign agent belonging to the associated multicast group will receive all packets for the mobile node; the designated foreign agent serving the mobile node will actually deliver the decapsulated datagrams to the mobile node. There is some additional control protocol to allow one of the foreign agents to be designated as the one currently serving the mobile node, and to allow new foreign agents to assume the designated function as needed when the mobile node moves. If the foreign agents were organized as an anycast group [38] the packet would only have to be delivered to one of the foreign agents. That foreign agent would then have to forward the packet to to the designated foreign agent, with correspondingly higher requirements for transmitting control information, but greatly reduced storage requirements for most of the foreign agents in the anycast group compared to the case or the multicast group.

Another idea is to arrange the foreign agents into a hierarchy. Then when the mobile node moves, it can restrict its registration messages to stay within the hierarchy as long as it can determine that its new point of attachment is in the same hierarchy as its previous point of attachment. The common ancestor is the nearest foreign agent that can handle the reregistration, and no further ancestors need to be aware of the mobile node's movement. The particular case when an administrative domain has a "gateway" foreign agent with many subordinate foreign agents may initially be a popular design point.

5.2. Route optimization

Aside from the basic operations provided by Mobile IP, extended operations allow for mobile-aware correspondent nodes to send their data directly to the mobile node instead of going through the home agent. This *route optimization* [34,47] is accomplished by sending the mobile node's care-of address to correspondent nodes, in so-called *binding updates*. Therefore, this technique can only work for such nodes that are able to process the protocol messages containing the necessary information; today's product computers cannot. Route optimization messages have almost the same need for security that registration messages do in base Mobile IP, since bogus binding updates sent to correspondent nodes allow the same sort of malicious traffic redirection that bogus registrations sent to a home agent would allow. Privacy considerations dictate that the dissemination of binding updates be controllable by the mobile node, since they carry information describing the mobile node's current location.

5.3. Smooth handoffs

Recent investigations have considered the advisability of buffering at the foreign agent, as part of a process of smooth handoffs. The paper in this issue by Cáceres and Padmanabhan is one of the first published in this area, and shows that substantial speedups can be obtained with minimal buffering strategies. Additional improvements can be obtained by integrating buffers with regionalized registrations. Handing off the buffered packets can be made secure by establishing security relationships using the binding update mechanism specified for use with smooth handoffs in route optimization.

Application of route optimization is also of particular interest. If foreign agents are enabled to maintain binding cache information for a mobile node, then they can improve the robustness of communications with that mobile node even after the mobile node moves away to a new point of attachment. When a foreign agent knows the mobile node's new care-of address, it can forward all packets for the mobile node to that new care-of address. For example, this would help with packets in flight sent to the mobile node during the time it is trying to complete its registration process, which might otherwise be lost. Note that this smooth handoff is even more important when there are correspondent nodes that are maintaining binding cache information for the mobile node acquired by use of route optimization protocol messages.

Smooth handoff is expected to need binding cache information only for some hundreds of milliseconds, the amount of time it takes for mobile nodes to complete a new registration and to update correspondent nodes with new binding cache entries. After this time, the previous foreign agent can drop the binding cache entry for the mobile node. Moreover, establishing the binding cache entry has reduced (but nontrivial) security requirements. Replay attacks would generally be ineffective, since the cached information has such a short lifetime and a foreign agent would not accept a new binding for any mobile node not already in its visitor list.

Providing any security at all for binding updates sent to a foreign agent by a mobile node may be problematic, because the mobile node and the foreign agent are not expected to have any security relationship before the time of the mobile node's registration. There are a number of methods defined by which a mobile node and foreign agent can establish the necessary security relationship. The methods defined attempt to use existing security relationships whenever available, but allow use of Diffie–Hellman key exchange as a last resort. The possibility of a *man-in-the-middle* attack, which frequently plagues Diffie–Hellman exchange protocols, is controlled by using the home agent as a *Key Distribution Center* (KDC) and allowing it to authenticate the extension containing the newly created key for the new security extension between the mobile node and the foreign agent.

5.4. Source routing

Many early approaches to Mobile IP attempted to make use of IP's *loose source route* (LSR) option. This seems an attractive possibility, because packets sent to a mobile node can be delivered directly to the mobile node by a foreign agent if the foreign agent is specified as part of the loose source route. Moreover, if the mobile node sends a packet to a correspondent node and includes the care-of address in the source route, the correspondent node can use the source route to return packets to the mobile node, achieving a cheap form of route optimization. Since IP specifies that higher-level protocols *should* reverse source routes, such source routing approaches accomplish mobile networking without creating any new protocol.

However, the gains offered by source routing approaches are, unfortunately, only illusory. In the first place, as with any such *remote redirection* as indicated by source routes requiring reversal by the receiver, authentication is required, and nodes reversing source routes do not typically perform any such authentication operations. Thus, malicious nodes could impersonate mobile nodes by sending bogus source routes. Because of the opportunity for foul play, most Internet routers do not forward source routed traffic, so that the whole approach is, in practice, unworkable. Moreover, even if the routers were configured to handle source routes, and the end nodes were configured to require authentication before reversing source routes, the performance penalty at the routers proves unacceptable for handling source routes. All of these factors combine to exclude source routing approaches from consideration as a solution for mobile networking in today's Internet.

5.5. Mobile IPv6

IP version 6 (IPv6) [16,23] is a new network layer protocol designed to increase the address space available for nodes within the Internet, and to improve routability for packets using IPv6 addresses. As part of the design process, many deficiencies of the current version of IP (also called IPv4) have been fixed. Support for mobile networking has been laid out as a mandatory requirement for IPv6 [11], and the design for Mobile IP has been modified to take advantage of IPv6's superior capabilities.

All IPv6 nodes are able to autoconfigure an IPv6 address appropriate for their current point of attachment to the Internet [35,60]; moreover there are plenty of IPv6 addresses available, so foreign agents are no longer needed to support mobility. Furthermore, since all IPv6 nodes are required to support authentication and privacy protection at the network layer, binding updates can be supplied in a secure fashion to the correspondent nodes that need them. This means that route optimization fits naturally within the framework offered by IPv6, and does not have to be done as an upgrade to a huge installed base as with IPv4. Since future Internet nodes are expected to be capable of mobility [24], this represents a significant reduction in the network load to be sustained by the IPv6 Internet.

In order to send packets to the mobile node, a routing header (the IPv6 equivalent of source routing) is used by any sender that has the mobile node's care-of address. On the other hand, whenever a packet arrives at the home agent instead of going directly to the mobile node, it can be assumed that the sender does not have the care-of address of the mobile node. In this case, the home agent does not insert a source route to complete the delivery of the packet to the mobile node. Instead, the home agent is required to use encapsulation. Thus, the mobile node can tell whenever it needs to send a binding update to any of its correspondents. Moreover, when the mobile node moves to a new care-of address, it assumes that each of its active correspondent nodes should receive a new binding update. The mobile node can find active correspondent nodes by checking its TCP protocol control blocks; but this only works for TCP traffic.

5.6. Vertical IP

Recent experiments at University of California at Berkeley (UCB) have shown the feasibility of using Mobile IP to assist mobile nodes when switching between heterogeneous physical media. This is important in many applications, for instance when a mobile node moves from a high-speed wireless LAN in an office environment, to a wide-area wireless connection, as with cellular telephones. The main considerations are handling discovery mechanisms in the disparate media, and making policy decisions about when it is best to change from one medium to another. For instance, one would like to maintain a high-speed and cost-free connection to the local wireless LAN as long as possible, until the error rate becomes too high for comfort, and correspondingly to switch back to the wireless LAN as soon as possible upon re-entering the campus or office environment where it is available. Other considerations such as security, proxy availability, route selection, or latency may also come into play. A good example of the work in this area is the paper in this issue by M. Stemm and R. Katz.

5.7. Multicast

Multicast protocols have, in the past, not been designed for the case of mobile nodes. In Mobile IP, a mobile node can pretend to be on its home network and receive tunneled packets, joining multicast groups through the tunnel. It can also attempt to join local multicast groups on the foreign network, but this leads to possibly poor performance in reconstructing the multicast routing tree after each movement, and possibly violates some of the implied semantics of multicast. These design points and many others are explored in the paper by Chikarmane et al. in this issue.

5.8. Tunneling

Mobile IP depends upon tunneling [22,41,42]. But, tunneling also plays a part in other protocol operations of interest to mobile nodes. For instance, access to enterprise computing resources for mobile users often depends upon establishing a tunnel through the firewall protecting the enterprise computing environment from malicious abuse by external Internet attackers. In fact, there seems to be a gradual convergence of efforts in the areas of mobile networking, *virtual private networks* (VPNs), and dial-up access to local or remote points of attachment to the Internet. One relatively new effort in this area is the Tunnel Establishment Protocol (TEP) [12], which takes as its initial design point the fact that Mobile IP is, among other things, a way to establish a tunnel between two points. For Mobile IP, the tunnel endpoints are the home agent and the care-of address, but this can be generalized. In fact, the previous ideas developed for hierarchical foreign agents (see section 5.1) carry over to TEP, and help motivate a way to establish multi-segment tunnels across multi-level security domains.

When Mobile IP was specified, IP-within-IP seemed to be the most suitable candidate for a default tunneling algorithm. Recent developments call for re-examination of that decision. Now, newer tunneling protocols such as L2TP [37] are receiving widespread deployment, and this author believes that they may represent another opportunity for offering the benefits of mobile computing to a new population of mobile users.

5.9. Network address translation

Network address translation is becoming a feature with wide deployment within the Internet. The basic idea is that a collection of nodes can use private IP addresses in a network which is attached to the global Internet by way of a *network address translation* (NAT) [55] unit, which “hides” the other nodes’ IP addresses. As data traverses the NAT unit towards the nodes using the private addresses, the network layer (IP) addresses in the IP header are translated from externally known IP addresses to the privately known addresses of the other nodes.

This technique spells trouble for Mobile IP, because a care-of address on the “inside” of the NAT unit does not make sense to a home agent on the “outside”. Until the NAT boxes can be programmed in detail about how to translate tunnel addresses, and the addresses inside Mobile IP Registration Requests and Replies, it seems unlikely that Mobile IP can work across NAT boundaries. This is not at all trivial to do, considering Mobile IP’s need for authenticating the registration messages; changing any of the internal fields would destroy the authentication data. Moreover, since NAT (typically) depends on port numbers, and IP-within-IP does not have a port number to use, there is a basic design incompatibility. To overcome this problem, the NAT device should probably also be the foreign agent.

6. Transport layer considerations

Supporting mobility at the transport layer usually means modifying the Transmission Control Protocol (TCP) [1]; other commonly available transport control protocols have not been investigated nearly as often. TCP provides for congestion control, reliable delivery, and sequenced reception of datagrams by the destination.

Providing for mobility by modifying TCP cannot be considered as a complete solution for mobile networking. In fact, modifying the User Datagram Protocol (UDP) to support mobility does not make very much sense, because UDP doesn’t keep track of any state relevant to the source or destination nodes. Neither the mobile node’s IP address nor anything else about it is used by UDP to identify the state of the data communication, so nothing can be done by UDP to help improve the forward progress of communications to or from a mobile node. RTP is not as widely deployed as TCP or UDP, and makes up only a tiny percentage of the total data flowing in the Internet, so that there has been much less consideration given to the transmission of data by mobile nodes using RTP.

TCP, however, offers many interesting possibilities. Careful coordination between the mobile node and TCP running at a base station can provide the following benefits:

- Reduced retransmission delays;
- Smooth handoffs;
- Improved throughput.

For data streams to or from a mobile node, which flow through a base station, several investigators have proposed breaking the data stream into two parts which are handled separately; both substreams can be terminated at the base station. Some approaches [3,4,63] suffer from the problem of providing TCP ACKs to correspondent nodes, for packets that are never actually delivered to the mobile node. This violates the well-understood end-to-end semantics of TCP, and requires very careful handling, or perhaps even making modifications to application software.

Going a step further, it is possible to equip TCP at the base station with the power to transfer internal state related to the mobile node to a new base station, whenever the mobile node moves from place to place. Providing for mobility in this way shares some features in common with Mobile IP. In the first place, it is often presumed that only the mobile node or the base station can be modified to provide the mobility support. As the mobile node moves to a new point of attachment to the Internet, it must notify the previous intermediate TCP connection point about its new location, so that all necessary TCP control information can be modified or transferred. This can be considered as a variation on Mobile IP’s registration procedure, and carries with it all the same requirements for authentication (and, in certain applications, privacy).

6.1. Snooping

When a base station is the last hop from the wired Internet to a wireless mobile node, the base station can improve end-to-end performance by retransmitting lost TCP packets only over the wireless link, while still maintaining TCP's end-to-end semantics. These retransmissions are invisible to the remote connection endpoint, and are comparable in effect to retransmissions performed at the link layer 4.5. As the base station delivers packets over the wireless link to the mobile node, it buffers the packets until the mobile node sends the expected TCP acknowledgement. If the acknowledgment does not come (in time substantially shorter than the end-to-end *round-trip-time* (RTT)), or, if the mobile node sends another TCP acknowledgement (a *dupack*), the base station can retransmit the lost packet and avoid end-to-end timeouts and retransmissions. This approach of snooping and buffering offers big performance improvements [5].

Unfortunately, recent security protocols preclude inspection of the relevant packet contents (e.g., TCP sequence numbers) by base stations. It seems unlikely that the mobile computer user would wish to share its privacy keys with every base station (or foreign agent) that it establishes a connection with. There do not appear to be any simple approaches to this problem, so that performance increases available from snoopy base stations will be lost for the duration of encrypted data transmissions.

6.2. Errors vs congestion

TCP, as commonly implemented, offers advanced features for controlling Internet congestion. The primary observation about such control algorithms, is that control traffic has to be minimized or nonexistent after congestion occurs, because there is a high probability that any control packets would be dropped, and besides that they add to the congestion anyway. TCP's *slow start* [56] performs as needed to reduce congestion, by first throttling the data transmission of the connection, and then slowly building back up to an efficient transmission rate.

The problem comes when errors are mistaken as evidence of congestion. Packets which are lost or garbled will effectively not be delivered to TCP, and may trigger the slow-start mechanism. This is bad, because packets with corrupted data should be retransmitted right away, and should not cause such a slowdown in the data rate. Thus, the effect of errors due to wireless media is magnified by slow-start. Poor interactivity and reduced throughput are the likely results. It would be better if TCP could be modified to detect whether a lost packet was the result of congestion or instead was lost because of bit errors [9,10,32].

One theory suggests that packets lost due to congestion tend to be lost in long contiguous sequences, and that packet loss because of bit corruption occurs more randomly, intermingled with error-free packet reception. It remains to be seen whether TCP can be modified to make the determination about causes for errors, and whether the determination can be exact enough to produce improvements in the overall response of TCP packet-loss algorithms in real Internet operational environments.

6.3. Asymmetry

Satellite communications with mobile nodes can provide an important type of wireless connectivity to the Internet. In many cases, the communications path is then asymmetric, for mobile nodes that do not transmit data back to the satellite. The mobile node might use a telephone or other land line to maintain end-to-end connectivity with other Internet nodes, relying on the satellite link only for downloading bulk data (for instance, video information). The data rate available on the satellite downlink is typically far greater than the reverse link from the mobile node back to the Internet. Thus, both the data rate and the routing path are different.

TCP was not constructed to work well with such asymmetric data rates. When the asymmetry is too great, the mobile node cannot supply ACKs back to the source of TCP data fast enough, and the supply of data to the downlink operates at far below capacity. Most solutions to this problem require changes to the Internet node providing data to the mobile node [2]. As with route optimization, solutions in this class will take a long time to deploy, and will probably only happen as satellite communications become important for the general operation of the Internet.

6.4. Resource reservation

Mobile IP uses tunnels as part of the path for packets to be delivered to the mobile node, and that affects the flexibility of paths reserved for multimedia data between Internet nodes [59]. Once the tunnel is established, it is not so easy way for another Internet endpoint to make sure the intermediate points of the tunnel are willing to offer the necessary resources for a new multimedia data stream. This is especially true because the correspondent node may not even be aware that it is communicating with a node that is mobile.

Worse yet, the tunnel from the home network to the mobile node is re-established every time the mobile node moves to a new point of attachment to the Internet. One solution to this problem has the mobile node establishing paths with sufficient resources to a possibly large set of attachment points [58]. When the node arrives at a particular point of attachment, the path to that attachment point becomes active, so that the data can still be delivered effectively. The downside of this approach is that many resources are reserved which may never be used, and the reserved resources remain unused even though they are available for other uses.

This same problem occurs for multicast delivery of data to mobile nodes, although in that case there are already protocol methods in place for pruning the multicast routing tree so that data need not be delivered to a previous point of attachment of the mobile node.

For a connectionless algorithm for reserving bandwidth and making it available to applications needing a well-defined flow availability, see the paper in this issue by Murthy and Garcia-Luna-Aceves, which leads the way to some interesting new research directions.

7. Middleware

For the purposes of this paper, we will define *middleware* to be the software which does not directly handle application protocol needs, but on the other hand fulfills, in a generic way, an intermediate or ancillary role in providing network services or environment to network applications.

Nomadic computer users, by definition, change their locality and thus need to periodically re-establish their link and connectivity to the Internet. Since the parameters of such connectivity typically depend upon the characteristics of the current point of attachment, nomadic users require that their connectivity be parameterized by those relevant characteristics. This introduces many problems that are not very well satisfied by existing solutions for network connectivity.

For instance, Mobile IP can be understood as a protocol to allow parameterization of the IP address of the mobile node's current point of attachment (i.e., to allow for variable care-of addresses). But, Mobile IP is not invoked by application software, and usually is considered to operate at the network layer; thus it is not middleware. DHCP (see section 7.2), on the other hand, could (theoretically) be invoked by applications to obtain application-specific parameters, like server IP addresses that can be used by the client application to initiate a transaction. Thus, DHCP could be considered as middleware.

In this section we list a few potential candidates for middleware functions that are likely to become more important as computers become more mobile.

7.1. Service location

When a nomadic user arrives at a new computing environment, it is likely, and probably typical, that the user will be unaware of basic configuration details about local network services. For instance, there may be a dozen local printers, each with varying capabilities, and each possibly useful at various times to the nomadic user. It would be nice if the user could resolve service needs automatically, dynamically, and based only on the nature of those needs, independent of local naming conventions or network topology.

This ability to dynamically resolve service needs, which is a matter of convenience now, is likely to become a necessity in the service-oriented network of the future. There is likely to be an increased emphasis on accessing data across the network, as the Internet becomes more fully deployed. Consequently, when the network is viewed as a universal (and robust) resource, applications will begin to make use of network resources as a matter of course, much as Web applications now often assume multimedia capabilities which were quite rare and expensive ten years ago. If a typical computer hosts applications which together make use of dozens or hundreds of disparate network resources and services, then typical users are quite unlikely to be willing to reconfigure these applications at each new point of attachment. The number and diversity of network services will make manual configuration obsolete, and the ease and speed of network reattachment offered by wireless communications will make even hard-coded profile-based reconfiguration seem quite awkward.

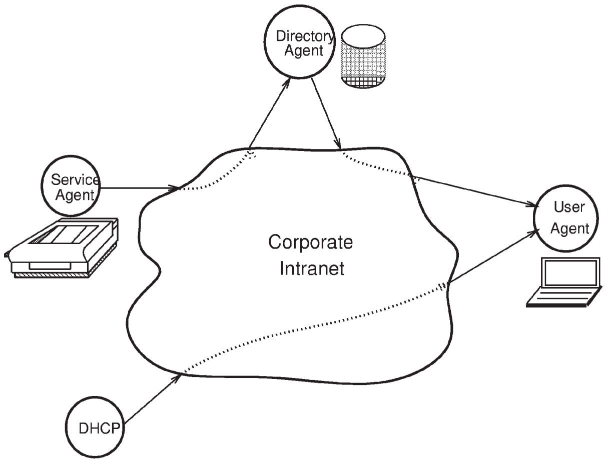


Figure 2. SLP agent model.

Service Location Protocol (SLP) [61] enables simple *service requests* from user agents to be resolved by receiving *service replies* which contain URLs from *service agents*. The user agents act on behalf of the application needing service, and the service agent acts on behalf of the network-attached service. The protocol for user agents and service agents is lightweight and places minimal load on the communications medium, as appropriate for typical nomadic computing platforms.

User agents can obtain the necessary service handles directly from service agents, or alternatively they can query a nearby *Directory Agent* (DA) for the information. These relationships are illustrated in figure 2, where the printer is shown represented by a service agent. In the configuration shown, the User Agent discovers the Service Agent using DHCP [39].

SLP offers other features for convenience and scalability not relevant to this article.

7.2. DHCP and dynamic DNS

The Dynamic Host Configuration Protocol (DHCP)[1] is likely to play a prominent role in the deployment of future mobile computers. DHCP fulfills the basic requirement for allocation of an IP address to a node which needs to begin communications at its new point of attachment. Today, DHCP is not typically employed by mobile computers, but is seeing use with portable computers. When a computer is attached to a LAN, for instance, it can call DHCP to get its IP address, along with a default router, the domain name server for the local network, and various other bits of useful information. This works for mobile computers, too,

[1] R. Droms, Dynamic Host Configuration Protocol, RFC 2131 (March 1997).

but each time the connection is made the mobile computer typically needs to be restarted.

Even if the computer could work without restarting, there are severe difficulties with establishing connections to a mobile computer that relies only on DHCP for its network attachment. For one thing, most communications with the mobile node start with its domain name (often, its Fully Qualified Domain Name (FQDN)). Each new IP address would require updating the IP address resolution for that mobile node's domain name unless, all communications with the mobile node are to be initiated by the mobile node. On the other hand, updating DNS is an operation that can be performed only with very tight security. If a bogus update were ever accepted for the mobile node's domain name resolution, all communications depending on that resolution would be disrupted and possibly hijacked. Such security operations are tricky and are only now becoming standardized [17,18].

Moreover, there remains a problem with DNS caching. Whenever the resolution of a mobile node's domain name is cached at an intermediate name server, that cache will be stale as soon as the mobile node moves to a new point of attachment. Thus, as more and more mobile nodes are deployed, misusing DNS for this purpose will cause a proportionate increase in the already huge amount of traffic taken up for name resolution. Combatting the problem by disallowing DNS resolutions to be cached only adds to an already worrisome problem in today's Internet.

Supposing that the appropriate security measures can be taken, supplying new resolution information for each new point where the mobile node attaches to the Internet does not preserve existing communications. And, as wireless cell sizes decrease (see section 3), this will be viewed as increasingly inconvenient until finally it is just unacceptable.

If, on the other hand, the mobile node uses Mobile IP, and is equipped to use DHCP as a mechanism for obtaining a (co-located) care-of address, it can maintain its existing home address resolution for its FQDN. This allows simplified communication with the mobile node at all times, as well as enabling the node to preserve its ongoing communications at each new point of attachment. In this mode of operation, the mobile node can also make use of the default router configuration delivered to it by DHCP. Since no beacons may be expected from any foreign agent, the mobile node with a co-located care-of address may be designed to substitute pings to the default router instead of detection of agent advertisements. DHCP can also be used to get information about SLP directory agents at the same time that the care-of address and default router information is obtained, as illustrated in figure 2.

7.3. PPP

Just as a mobile node which employs Mobile IP can use an IP address acquired from the local DHCP server as its care-of address, it can also use the IP address allocated to it by establishing a PPP (dialup) connection as a care-of address. In this situation, the mobile node will use the NAS (Network Access Server) as its default router. Furthermore, it is possible that the mobile node, by using recently defined PPP extensions [53], can detect whether or not the NAS is in fact a foreign agent, and thus able to perform decapsulation for the mobile node.

Mobile nodes using PPP now face a stringent requirement for managing authentication (e.g., by using CHAP), so that the guest network can charge for local network resources used. This authentication process stands in the way of real mobile networking, and relegates the PPP users to intermittent connectivity, which can be called *portable* computing. Automatic ways to perform billing will be needed before PPP-based approaches can offer seamless mobility. On the other hand, situations where connectivity is achieved by discrete dial-up operations do not present the user with the occasion to expect continuous connectivity anyway.

7.4. Adaptivity

As indicated previously, a mobile-aware application running on a mobile node might profitably be designed to take advantage of information about the link conditions or other information about the mobile node's point of attachment. For a simple example, if the wireless connection offers only a small amount of bandwidth, it would be fruitless for an application to attempt to acquire large volumes of graphic presentation data from the network. Instead, high-bandwidth graphics should in those circumstances be deferred or eliminated entirely from the data stream. This determination should be made dynamically if possible, since transient conditions cause great variability in the available bandwidth to the application. Furthermore, smart wireless adapters are able to trade off bandwidth for error correction. While this tradeoff is likely to improve bandwidth availability overall, it also directly illustrates the point that wireless bandwidth should not be considered a constant over the life of a particular link between a mobile node and the rest of the Internet.

Bandwidth, however, is not the only wireless parameter of possible interest to applications. Cost can also be a big factor, and can determine whether to spool results for later processing or printing. In many circumstances, bandwidth is less important than controllable delay bounds. Furthermore, applications may wish to adapt to changing security factors or make variations in other policy selections.

Well-known methods of adapting to changing conditions make sense when designing mobile-aware applications, depending upon the form in which the information is made available. For instance, if the wireless communication channel parameters of interest are stored in a system data area, the application may be designed to poll the system data at some time granularity. If, on the other hand, the operating system supports event notifications as well as the ability to define events based on parameter values for the wireless communication attributes, a more efficient, event-driven system can be designed. One likely strategy for constructing such events uses the idea of defining high-water and low-water marks for each appropriate parameter, with the necessary amount of hysteresis built in to avoid unnecessary flurries of borderline events. Both system data and event notifications are useful for many applications, so having one should not preclude the other.

7.5. MNCRS

The Mobile Network Computer Reference Specification (MNCRS) is under development by a consortium of companies interested in enhancing the marketability of *network computers*, where a network computer is supposed to be dependent upon external services for its operation. Thus, a network computer (NC) platform is typically associated with a department or enterprise server that is specially configured to serve that kind of computer; moreover, the network computer is considered to belong to a large population of similar computers, each of which using the same server or a similarly configured server. The main differentiation between different instances of a network computer might result from automatic interpretation of a customized user profile, or from specific interactions with the user.

Schematically, MNCRS looks like a Java API on top, and some mandatory, standardized networking protocols on the bottom. This is intended to guarantee device interoperability as well as application portability. The specification applies to four *classes* of devices:

1. Professional assistant (e.g., a laptop);
2. Information access device (e.g., for calendaring);
3. Basic messaging, paging, and telephony devices.

Some parts of the specification may not apply to all device classes.

The MNCRS effort is organized into four working groups:

- Data synchronization;
- Mobile communications;
- Booting and automatic configuration;
- Operation for devices in classes 2 and 3.

Data synchronization, derived from ideas developed at CMU [51], is required to allow mobile wireless nodes to work independently whenever their network connection is down, either intermittently or for a protracted period of time. With reasonable care, the mobile node can cache enough programs and data from a server, so that the operation of the mobile node proceeds independently from the server until connection is re-established. Then, data from the mobile node can be applied as updates to files resident on the server, and files on the server that have changed can be resent to the mobile node. Simultaneous inconsistent updates have to be taken care of; with good engineering design, the problems can be minimized, but the mobile user will sometimes have to decide manually to resolve conflicts.

The mobile communications group has the most interesting job, from the standpoint of the topics covered in this special issue. That group has the charter to specify mobile networking protocols, an API application adaptivity to changing network conditions, tunneling protocols, use of network computers with Network Address Translation (NAT) devices, and a messaging API useful (for instance) by class 3 devices. The network model visible to MNCRS applications should enable direct communications between network computers and servers, as well as indirect communications (transparently or explicitly) through proxy devices. Again, network interoperability between NCs, and between NCs and servers, will rely as much as possible on network protocols standardized or under consideration for standardization within the IETF. Examples of protocols that are specified as part of MNCRS include HTTP, HTTPS, Mobile IP, DHCP, IPSec, Service Location Protocol (SLP), and Secure Sockets Layer (SSL), as well as infrastructure favorites like IP, TCP, UDP, and DNS.

The working group which is considering Booting and Automatic Configuration has a difficult task, given today's infrastructure and the complications of today's Internet. Some of the protocols just mentioned, for instance DHCP and SLP, will go a long way towards enabling autoconfiguration Java offers features to help, for instance the Java Naming and Directory Interface (JNDI), which allows a unified interface to directory services like SLP and the Lightweight Directory Access protocol (LDAP), even though the sorts of directory entries storied in the heterogeneous directories might be handled quite distinctly. Along with autoconfiguration, MNCRS specifies flexible boot sequences, and offers power-saving services by way of Java APIs. These are currently important considerations for all mobile devices.

7.6. Environment management

As wireless mobile nodes become more common, and people are more likely to carry data processing and computing power with them in various forms, the computing nodes themselves may be provided with additional environmental services to support their improved operation. Such environmental services may suggest or indicate the selection of appropriate user profiles. For instance, it would be nice if telephones (cellular or otherwise) did not ring in conference rooms. If user data is being displayed on a clip-area of a common white board, the user may wish to disable output from certain applications for privacy reasons. Privacy concerns may cause the user to run applications in different modes when roaming away from home, even if just for the reason that wireless communications seem more vulnerable to eavesdropping than do wireline communications on building LANs.

In fact, the kinds of environmental considerations that may come into play at a mobile node's new point of attachment to the Internet are probably so varied and difficult to classify that we cannot make a very good analysis of future developments in this area. Perhaps protocols such as ACAP (Application Configuration Access Protocol [36]) will help us organize our directions. On the other hand, other kinds of environmental interactions are less related to user application profiles, and more related to the user's constantly changing need for information. For instance, it doesn't make much sense to route the user's data to a local printer unless the user has some way to physically access that printer. Similarly, making positional determinations by GPS or by local building coordinates are both useful at times, depending upon the execution environment of the application. Moreover, such application adaptivity is not naturally modeled as a need for a particular service or network parameter, so that DHCP or SLP is not likely to be of much help.

Designing mobile-aware applications to make use of such environment management functions will be the subject of much future interest.

7.7. Policy determination

Mobile nodes will eventually be equipped with policy engines to enable applications to make the right decisions along many dimensions. Some examples have been mentioned previously, including choice of graphic resolution, cost management, and avoiding embarrassment. Nodes using co-located addresses may be faced with a choice of which IP address to use, based on high-level considerations [13,64]. If a mobile user is accessing data from a service that does not care about the IP address of the source, the mobile node should more likely use its care-of address to avoid having traffic go back through the home network. Likewise, for TCP sessions which are unlikely to last very long, Mobile IP may offer little advantage. Such sessions should also be identified internally by the care-of address instead of with the mobile node's home address.

Appropriate application designs are difficult to build in a modular fashion with today's tools. What is needed is a better way for various system modules to determine which policies should be put into effect at a certain moment, depending upon environmental considerations, link conditions, time of day, the user's work (or recreational) purposes, and many other factors. While not exclusively arising from user mobility, it is mobility that brings these design needs into sharp focus. Changes in a user's computing context naturally correlate very well with changes in the user's physical location.

7.8. Proxies

Not only will mobile nodes rely on standardized middleware components to simplify their design and operation, they may well also rely on standardized network components to perform specialized services such as voice recognition, protocol translation, or to provide access to specialized hardware. These network components are called *proxies*, and a great deal of recent interest has been focussed on ways to offload functions from the mobile node to suitable proxies [27]. Use of proxies (possibly in conjunction with so-called *intelligent agents*) offers the following advantages to mobile nodes:

- Power savings;
- Reduced storage requirement;
- Access to specialized hardware;
- Reduced mobile platform maintenance;
- Performance (speed) improvements;
- Disconnected operations.

On the other hand, reliance on external entities such as proxies introduces a new requirement for compatibility. Software on the mobile node could be difficult to upgrade unless the proxy software is also upgraded to match. Upgrading a proxy that serves dozens or hundreds of mobile nodes could be quite tricky. Moreover, the proxy represents another required stopover for data *en route* to the mobile node. As the mobile node moves away from its proxy, it will see gradual performance reductions. This performance loss may indicate a need to enable the mobile node to switch proxies; such a switching operation is new and has not been fully investigated. Just switching to a closer replica of a distant database is already tricky [57]. Managing additional dynamic program behavior could prove infeasible.

8. Security

Security is an increasing concern in the design of mobile networking protocols and systems. As seen in the discussion about Mobile IP, authentication is critical to authorizing operations indicating the mobile node's new point of attachment. As another example, we have seen how the link layer can be augmented to supply encryption; the need for encryption is increased because of the frequently untrustworthy nature of the mobile computer's surroundings. Privacy takes on added importance, when the mobile user does not wish to divulge his or her current whereabouts.

Modern approaches to authentication and encryption use cryptographic approaches. The algorithmic results are made unforgeable by including secret keys (possibly with some additional unique data, such as a timestamp, to avoid matching any previously authenticated data) along with the data to be authenticated or hidden. Distribution of the secret key is a difficult problem in today's Internet [31].

Other security measures common in today's Internet affect mobile networking. Firewalls, which are installed to protect an enterprise computing environment from external intrusion and/or disruption, make it more difficult for mobile workers to make use of their office computing environment. Border routers that enforce forwarding policies based on the *source address* of packets [19] (as opposed to the traditional reliance only on the *destination address*), make it difficult for mobile nodes to use their home address in foreign domains. This *ingress filtering* can force even further detours in the routing path between a mobile node and its correspondent nodes [13].

Authentication algorithms typically rely on the possession of a secret key to establish the identity of the sender. For instance, in Mobile IP, the originator of the Registration Request *implicitly* claims to be a mobile node under the care of the destination home agent. This claim is verified if the authentication data has been correctly calculated. Instead of such an implicit claim, authentication could also be done using other more explicit claims of identity, such as the node's FQDN. Using some identification other than the IP address of the mobile node might have the beneficial effect of enabling mobile networking across NAT boundaries, for instance.

See the paper in this issue by Gupta and Montenegro for further discussion of some of the issues in this area, along with an effective design for firewall traversal. It is to be hoped that as more experience is gained with secure mobile networking, useful techniques will become widespread enough to offer standardized products to be readily available.

9. Ad hoc networking

Suppose for the moment that the needs for wireless services and connectivity could be supplied to a population of mobile users while they are within range of foreign agents or base stations connected to the Internet. Next, imagine that the same users met together at a conference which did not offer wireless connectivity to the Internet. These users might still need to communicate data files to each other, browse each other's Web pages, transact electronic mail, or use any of the many network applications which have motivated the tremendous growth of the Internet. They would find that their mobile networking software was useless without the needed infrastructure, and might even seriously get in the way.

These users need a way to deliver packets between wireless stations without infrastructure routers. If all the wireless nodes are within range of each other, this is not difficult. Mobility poses no problem, unless two nodes that need to communicate have moved out of range from each other. Otherwise, any necessary routing functions must be performed by the mobile nodes themselves. Intermediate mobile nodes could cooperate to forward data from source to destination.

Ad hoc networking is a name given to the creation of such dynamic and multi-hop networks that are created by the mobile nodes as needed for their communication purposes. The mobile nodes can do this in many ways. Most solutions involve running routing protocols on the mobile nodes. Routing protocols have the advantage that they are inherently multi-hop. Their dynamic behavior requires careful attention, however, because the typical rate of change in an ad hoc network is likely to be substantially greater than that for the topology of the Internet, for which most routing protocols are engineered. There are numerous routing protocols proposed and in use within the Internet today, and each of them could be potentially modified and applied to the creation of ad hoc networks.

The two main kinds of routing algorithms in use today are link state algorithms, which provide each node with a complete representation of the network topology, and distance vector algorithms. Examples of distance vector routing algorithms modified to work in ad hoc networks include DSDV [43] and AODV [44]. Link-state routing algorithms (e.g., OSPF [33]) have also been modified to provide reasonable response for communications between any two mobile nodes, even if they have not been in communication recently.

Instead of running routing protocols, and thus treating the ad hoc network as an *intranet*, mobile nodes can treat it instead as an incompletely connected physical medium. In such an approach, all IP addresses are considered to be part of the same communications medium, but the multihop nature of the medium requires the cooperation of various mobile nodes to keep it together. Viewed in this way, the process of finding a path to a destination can be handled by extending ARP to return the layer-2 address of the next hop towards the destination. The Dynamic Source Routing [25] method enables source routes to be returned to the ARP requestor, and extends the domain of applicability further to handle asymmetric wireless connectivity.

Other ideas which have been applied to the construction of ad hoc networks include formation of hierarchies, tracking signal strengths to select the most robust data path, and maintaining multiple routes for improved handling of preferential service needs or bounded delay paths. The IETF Mobile Ad Hoc Network (`manet`) working group is attempting to establish standards for creation of ad hoc networks, and all of the above techniques are receiving attention.

Ad hoc networking presents interesting challenges for traditional client/server applications. For one example, consider whether DNS might be accomplished in an ad hoc network. First, there is no clearly defined way for the ad hoc nodes to discover which node or nodes are offering domain name resolution. Even without that problem, the non-hierarchical nature of ad hoc addressing does not map well into the standard hierarchical domain conventions of DNS. The trouble is the defining characteristic of the ad hoc network, which is that the IP addresses of the nodes are assumed to be unrelated to each other. If, instead, the IP nodes somehow acquire IP addresses dynamically and perform some sort of aggregation, their relative movement would soon make the initial aggregation ineffective. Besides that, it is difficult anyway to cause the nodes to dynamically select IP addresses which are unique across the ad hoc network. Various client/server applications present other difficulties.

As it turns out, there are nontrivial issues surrounding the simultaneous use of ad hoc networks with Mobile IP. Users would naturally expect that both should be useful together; a foreign agent attached to an ad hoc network should provide Internet connectivity to every node in the ad hoc network. On the other hand, manipulation of the route table by Mobile IP is not completely consistent with the way ad hoc routing protocols may wish to do route table management [29]. Furthermore, the rules for Mobile IP need to be adjusted or else interpreted correctly so that the agent advertisements can be delivered to every mobile node in the ad hoc network.

Use of multicast, RSVP, and other quality of service issues are not yet widely discussed in the context of ad hoc networks. As basic protocols become available these topics will assume additional importance, not least because of the obvious military applications for ad hoc networks.

10. Conclusion

Mobile computing opens the door to a fresh examination of practically every area of network protocol engineering. The areas discussed in this article, and the articles in this special issue, are only a sampling of the kinds of new research results being reported. It is my sincere hope that this special issue will pique the interest of new researchers, and provide a better overall understanding of the problem areas needing more attention and new solutions.

Acknowledgement

I would like to express my gratitude especially to all the authors of this issue, for their patience in handling the requests of the guest editors, and for their exciting contributions to this issue and to the field. Thanks also to the referees for their reviews, without which the editorial decisions could not have been made. Lastly, thanks to Nitin Vaidya and those who have helped with the preparation of this article.

References

[1] Transmission Control Protocol, RFC 793 (September 1981).

[2] M. Allman and D. Glover, Enhancing TCP over satellite channels, draft-ietf-tcpsat-stand-mech-05.txt (August 1998) (work in progress).

[3] A. Bakre and B.R. Badrinath, Implementation and performance evaluation of indirect TCP, IEEE Trans. Computers (March 1997) 260–278.

[4] A. Bakre and B.R. Badrinath, Handoff and systems support for indirect TCP/IP, in: *Proc. 2nd USENIX Symposium on Mobile and Location-Independent Computing*, Ann Arbor, Michigan, USA (April 10–11, 1995) pp. 11–24.

[5] H. Balakrishnan, S. Seshan, E. Amir and R.H. Katz, Improving TCP/IP performance over wireless networks, in: *Proc. 1st ACM Conference on Mobile Computing and Networking (Mobicom)* (November 1995).

[6] D.F. Bantz and F.J. Bauchot, Wireless LAN design alternatives, IEEE Network (March 1994) 43–53.

[7] M. Bergamo, R.R. Hain, K. Kasera, D. Li, R. Ramanathan and M. Steenstrup, System design specification for mobile multimedia wireless networks (MMWN) (draft), Technical report, BBN Systems and Technologies (October 1996).

[8] P. Bhagwat, C. Perkins and S.K. Tripathi, Network layer mobility: an architecture and survey, IEEE Personal Communications Magazine 3(3) (June 1996) 54–64.

[9] S. Biaz and N.H. Vaidya, Distinguishing congestion losses from wireless transmission losses, in: *IEEE 7th Int. Conf. on Computer Communications and Networks* (October 1998).

[10] S. Biaz and N.H. Vaidya, Sender-based heuristics for distinguishing congestion losses and wireless transmission losses. Technical report, Computer Science Department, Texas A&M University, College Station (June 1998) (under preparation).

[11] S. Bradner and A. Mankin, The recommendation for the IP next generation protocol, RFC 1752 (January 1995).

[12] P. Calhoun and C. Perkins, Tunnel establishment protocol (TEP), draft-ietf-mobileip-calhoun-tep-00.txt (December 1997) (work in progress).

[13] S. Cheshire and M. Baker, Internet mobility 4x4, in: *Proc. ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York (August 26–30, 1996). ACM SIGCOMM Computer Communication Review, 26(4) pp. 318–329.

[14] D.E. Comer, *Principles, Protocols, and Architecture. Internetworking with TCP/IP*, Vol. 1, 3rd edn. (Prentice Hall, Englewood Cliffs, N.J., 1995).

[15] D. Cox, Wireless network access for personal communications, IEEE Comm. Magazine 30(12) (December 1992) 96–116.

[16] S. Deering and R. Hinden, Internet protocol, version 6 (IPv6) specification, RFC 1883 (December 1995).

[17] D. Eastlake, Secure domain name system dynamic update, RFC 2137 (April 1997).

[18] D.E. Eastlake and C.W. Kaufman, Domain name system protocol security extensions, draft-ietf-dnssec-secext-09.txt (January 1996) (work in progress).

[19] P. Ferguson and D. Senie, Ingress filtering in the Internet, RFC 2267 (January 1998).

[20] M. Gerla and J.T.-C. Tsai, Multicluster, mobile, multimedia radio network, ACM J. Wireless Networks 1(3) (July 1995).

[21] Z.J. Haas and M.R. Pearlman, The zone routing protocol (ZRP) for ad hoc networks, draft-zone-routing-protocol-00.txt (November 1997) (work in progress).

[22] S. Hanks, T. Li, D. Farinacci and P. Traina, Generic routing encapsulation (GRE), RFC 1701 (October 1994).

[23] R. Hinden and S. Deering, IP version 6 addressing architecture, RFC 1884 (December 1995).

[24] C. Huitema, *IPv6 – The New Internet Protocol* (Prentice Hall PTR, Upper Saddle River, NJ, USA, 1996).

[25] D.B. Johnson and D.A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks* (Kluwer Academic, 1996) pp. 153–181.

[26] P. Karn, MACA: A new channel access method for packet radio, in: *ARRL/CRRL Amateur Radio 9th Computer Conference* (September 1990).

[27] R.H. Katz, Adaptation and mobility in wireless information systems, IEEE Personal Commun. Magazine 1(1) (1994) 6–17.

[28] P. Krishna, N.H. Vaidya, M. Chatterjee and D.K. Pradhan, A cluster-based approach for routing in dynamic networks, ACM Computer Commun. Review (March 1997) 372–378.

[29] H. Lei and C.E. Perkins, Ad hoc networking with mobile IP, in: *Proc. 2nd European Personal Mobile Communications Conference* (October 1997) pp. 197–202.

[30] B.M. Leiner, D.L. Nielson and F.A. Tobagi, in: *Proc. IEEE Special issue on “ Packet Radio Networks”*, Issues in Packet Radio Network Design 75(1) (1987) 6–20.

[31] D. Maughan, M. Schertler, M. Schneider and J. Turner, Internet security association and key management protocol (ISAKMP), draft-ietf-ipsec-isakmp-10.txt (July 1998) (work in progress).

[32] G. Montenegro and S. Dawkins, Wireless networking for the MNCRS, draft-montenegro-mncrs-00.txt (August 1998) (work in progress).

[33] J. Moy, OSPF version 2. Request for comments (draft standard) 2178, Internet Engineering Task Force (July 1997) (Obsoletes RFC1583).

[34] A. Myles, D. Johnson and C. Perkins, A mobile host protocol supporting route optimization and authentication, IEEE J. Selected Areas in Commun. 13(5) (June 1995) 839–849.

[35] T. Narten, E. Nordmark and W. Simpson, Neighbor discovery for IP version 6 (IPv6), RFC 1970 (August 1996).

[36] C. Newman and J.G. Myers, ACAP – application configuration access protocol, RFC 2244 (November 1997).

[37] W. Palter, T. Kolar, G. Pall, M. Littlewood, A. Valencia, K. Hamzeh, W. Verthein, J. Taarud and W.M. Townsley, Layer two tunneling protocol ‘L2TP’, draft-ietf-pppext-l2tp-08.txt (November 1997) (work in progress).

[38] C. Partridge, T. Mendez and W. Milliken, Host anycasting service. Request for Comments (Informational) 1546, Internet Engineering Task Force (November 1993).

[39] C. Perkins, DHCP options for service location protocol, draft-ietf-dhc-slp-04.txt (November 1997) (work in progress).

[40] C.E. Perkins, Mobile IP, Int. J. Commun. Systems 11(1) (March 1998) 3–20.

[41] C. Perkins, IP encapsulation within IP, RFC 2003 (May 1996).

[42] C. Perkins, Minimal encapsulation within IP, RFC 2004 (May 1996).

[43] C. Perkins and P. Bhagwat, Routing over multi-hop wireless network of mobile computers, SIGCOMM’94: Computer Commun. Review 24(4) (October 1994) 234–244.

[44] C.E. Perkins, Ad hoc on demand distance vector (AODV) routing, draft-ietf-manet-aodv-02.txt (November 1998) (work in progress).

[45] C.E. Perkins, *Mobile IP*: *Design Principles and Practice* (Addison–Wesley, Reading, Massachusetts, 1998).

[46] C.E. Perkins, Mobile networking through mobile IP, IEEE Internet Computing Magazine 2(1) (January 1998) 58–69.

[47] C.E. Perkins and D.B. Johnson, Route optimization in mobile-IP, draft-ietf-mobileip-optim-07.txt (November 1997) (work in progress).

[48] C. Perkins, ed., IP mobility support, RFC 2002 (October 1996).

[49] D.C. Plummer, An ethernet address resolution protocol: or converting network protocol addresses to 48bit ethernet addresses for transmission on ethernet hardware, RFC 826 (November 1982).

[50] J.B. Postel, ed., Internet protocol, RFC 791 (September 1981).

[51] M. Satyanarayanan, J.J. Kistler, P. Kumar, M.E. Okasaki, E.H. Siegel and D.C. Steere, Coda: a highly available file system for a distributed workstation environment, IEEE Trans. Computers 39(4) (April 1990) 447–459.

[52] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (Wiley, New York, 1994).

[53] J. Solomon and S. Glass, Mobile-IPv4 configuration option for PPP IPCP, RFC 2290 (February 1998).

[54] J. Solomon, *Mobile IP*: *The Internet Unplugged* (Prentice Hall, Englewood Cliffs, 1998).

[55] P. Srisuresh and K. Egevang, Traditional IP network address translator (traditional NAT), draft-ietf-nat-traditional-00.txt (July 1998) (work in progress).

[56] W. Stevens, TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, RFC 2001 (January 1997).

[57] C. Tait and D. Duchamp, Service interface and replica consistency algorithm for mobile file system clients, in: *1st Int. Conf. Parallel and Distributed Information System* (1991).

[58] A.K. Talukdar, B.R. Badrinath and A. Acharya, On accommodating mobile hosts in an integrated services packet network, in: *Proc. Infocom’97* (April 1997).

[59] A. Terzis, J. Krawczyk, J. Wroclawski and L. Zhang, RSVP operation over IP tunnels, draft-ietf-rsvp-tunnel-01.txt (August 1998) (work in progress).

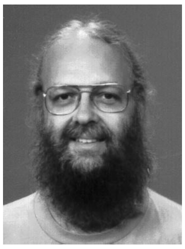
[60] S. Thomson and T. Narten, IPv6 stateless address autoconfiguration, RFC 1971 (August 1996).

[61] J. Veizades, E. Guttman, C. Perkins and S. Kaplan, Service location protocol, RFC 2165 (July 1997).

[62] M. Weiser, Some computer science issues in ubiquitous computing, Commun. ACM 36(7) (July 1993).

[63] R. Yavatkar and N. Bhagwat, Improving end-to-end performance of TCP over mobile internetworks, in: *Workshop on Mobile Computing Systems and Applications* (December 1994).

[64] X. Zhao, C. Castelluccia and M. Baker, Flexible network support for mobility, in: *4th ACM Int’l Conf. Mobile Computing and Networking (Mobicom’98)* (1998).



Charles E. Perkins is a Senior Staff Engineer at Sun Microsystems, developing Service Location Protocol and investigating dynamic configuration protocols for mobile networking. He is an editor for ACM/IEEE Transactions on Networking, for ACM/Baltzer/URSI Wireless Networks in the area of wireless networking, and for ACM/Baltzer Mobile Networks and Applications. He is serving as document editor for the mobile-IP working group of the Internet Engineering Task Force (IETF), and is author or co-author of standards-track documents in the mobileip, svrloc, dhc (Dynamic Host Configuration) and IPng working groups. Charles is also associate editor for Mobile Communications and Computing Review, the official publication of ACM SIGMOBILE. He is serving on the Internet Architecture Board (IAB) of the IETF. Charles has authored a book on Mobile IP, and has published a number of papers in the areas of mobile networking, ad-hoc networking, route optimization for mobile networking, resource discovery, and automatic configuration for mobile computers. Charles has served on various committees for the National Research Council, and is currently the chairperson of the Nomadicity Working Team of the Cross-Industry Working Team (XIWT). Charles holds a B.A. in mathematics and a M.E.E. degree from Rice University, and a M.A. in mathematics from Columbia University. He is a member of ISOC, ACM, IEEE, and the IETF.
E-mail: cperkins@eng.sun.com