Authenticated Ad Hoc Routing at the Link Layer for Mobile Systems

Jim Binkley
Department of Computer Science, Portland State University, Portland, Oregon, USA
E-mail: jrb@cs.pdx.edu

William Trost
Trost Computing, Portland, Oregon, USA
E-mail: trost@cloud.rain.com

We have designed an authenticated link-level ad hoc routing protocol and integrated it with the Portland State University implementation of Mobile-IP. The routing protocol addresses link security issues. In our protocol, mobile nodes, as well as agents, broadcast ICMP router discovery packets. The router discovery packets are authenticated and bind the sender's MAC and IP addresses. Problems caused by tying IP subnet schemes to routing on radio links are eliminated. Security problems associated with ARP spoofing are also reduced. This link-level protocol is integrated with Mobile-IP on links where increased security is needed. The protocol replaces ARP, and may be integrated with higher-level multi-hop ad hoc routing protocols.*

Keywords: link layer, ad hoc, ARP spoof, authentication, security, routing, Mobile-IP

1. Introduction

New networks are being created based on nodes communicating via wireless technologies such as infrared, cellular radio, and radio LAN. LAN radio devices, such as Lucent's WaveLAN^{TM}, a 2 Mbps spread-spectrum radio LAN technology, are capable of broadcast and multicast similar to Ethernet.[1]

Networks based on Ethernet-like radio broadcast bring certain topological and security challenges with them not previously encountered with wired technologies. Unlike a wired network, a wireless network is accessible to anyone in the vicinity. Predicting what machines can communicate with each other in a wired network is easy (any machine on a wire can reach any other machine on the same wire), but in a wireless network, reachability depends on the location of the computers and extrinsic sources of interference.

[1] See `http://www.WaveLAN.com/`.

Mobile-IP [12] is a network-layer routing protocol that specifies how hosts may move across IP subnets and retain an IP address. Thus, it permits a *mobile node* to change its point of connectivity while retaining a fixed IP identity. It does this primarily by arranging for packets from home (via the *home agent*, the mobile node's routing proxy) to be forwarded to the mobile node when it moves to foreign subnets.

Mobile-IP does not deal with some important situations. A group of mobile systems with different IP subnets (laptops carried by people from different organizations, for example) might gather together for a meeting in a hotel room, bar, business presentation, or lecture hall. These systems should be able to communicate with each other without network administration or additional infrastructure; that is, they should be able to form an *ad hoc network*. Mobile-IP does not address this situation. Hosts that are mobile are expected to send and receive all traffic via a foreign agent acting as an intermediary for the node. Mobile-IP does not attempt to provide for direct link traffic between end systems.

This paper presents a link-layer ad hoc routing protocol with two major goals. First, the protocol solves various reachability problems associated with traditional IP subnetting and ARP. These improvements aid routing survivability: mobile nodes using this protocol are better able to communicate with other mobile nodes in situations where they would be unable to communicate using Mobile-IP or traditional IP subnetting. Second, the protocol improves link-layer routing security, including problems associated with *ARP spoofing*.

The ad hoc protocol replaces ARP [14]. It is not intended as a multi-hop ad hoc routing protocol like the one described by Johnson and Maltz [8]. In fact, a multi-hop ad hoc routing protocol may be layered on top of our link-layer protocol.

The protocol is based on the axiom, *"I beacon, therefore I am"*. The ICMP router discovery packet [5] ("beacon") used in Mobile-IP has been extended to include the MAC and IP addresses of the sender, and authentication information that can be used to verify the beacon. The protocol requires that *all* nodes, both agents and mobile nodes, beacon. The protocol is similar to the ISO ES-IS protocol [13]. Nodes install link-layer routes for hosts whose beacons they receive.[2]

The protocol also improves link-layer routing security. All systems on a given link use either ARP or the ad hoc protocol, but not both. Since beacons authenticate the binding of the IP address to the MAC address, the Mobile-IP ICMP discovery packets are always authenticated for all agents and mobile nodes. This is not the case with standard Mobile-IP.

People interested in network security have long known that ARP may be easily spoofed. That is, an attacker may send an ARP packet containing its own MAC address and the victim's IP address, thereby usurping the IP-to-MAC address binding of the victim in other party's ARP cache. This causes the attacker to receive packets intended for the victim. The ad hoc protocol makes spoofing this binding impossible, although MAC address spoofing is still a threat.

As another security feature, mobile nodes that lack a link authentication key are unable to gain service from agents on the wired infrastructure. They may send packets to agents, but no packets are sent in return. This makes the perimeter defense against unauthorized mobile nodes more secure.

Mobile nodes running Portland State University's Mobile-IP can choose to use our beaconing protocol or the ARP-based address resolution mechanisms described in the Mobile-IP standard on a per-link basis. Our system's user interface allows the user to learn what protocol (ARP or ad hoc) is in use on a link. The user can then decide the security policy he/she wishes to employ and communicate this policy to the underlying system software.

The rest of the paper is organized as follows: Section 2 presents a discussion of background issues and problems to be solved by the protocol. Section 3 presents our protocol details and describes the mobile node user interface. In section 4 we show how our protocol solves the problems presented previously, and in section 5 we present a security analysis of the protocol. Section 6 concludes the paper.

2. Problem Discussion

In this part of the paper, we present a number of problems that justify our protocol. The problems fall in two general categories:

1. Problems based on the assumption that a shared subnet implies link reachability. Assuming that a mobile system can be reached

[2] We assume that the medium is symmetric: If host A can send a packet to host B, then host B can send one to A. ARP also makes this assumption.

simply because it shares the same subnet causes connectivity problems.

2. Wireless links offer opportunities for entrance into a presumably secure wired infrastructure. ARP spoofing may be a hazard, especially at Mobile-IP foreign agents, where it is possible that visitors from different sites (with different subnets) may congregate.

2.1. Subnetting

IP subnetting has a number of functions. At a high level, it makes possible route aggregation. IP routing is based on the distribution of pairs consisting of an IP network address and a netmask [7] as found in routing protocols such as OSPF [11] and RIPv2 [9]. Aggregating routes with netmasks has made the growth rate of routes in core Internet routers tolerable. This aspect of IP subnetting is not the issue here.

On the other hand, subnetting has a low-level function that typically manifests itself in implementations of the IP layer. A subnet mask may be used to determine if the destination node is on a directly-connected link. This information is used to determine whether to attempt to send packets directly to the recipient or to send them via a router.

The following is an algorithm that a very simple TCP/IP implementation might use:[3]

```
dest subnet := dest address AND subnet mask
if dest subnet = my subnet
    assume same link
    ARP for the destination
else
    send to a router
    (use ARP to talk to router)
```

Thus, a shared subnet implies the host can directly reach the destination. This important assumption is evidence of "wired thinking" that is detrimental when considering wireless or mobile systems. Some of the problems this assumption causes are described below.

[3] See Comer [4] for an example.

2.1.1. Adjacent Mobile Nodes

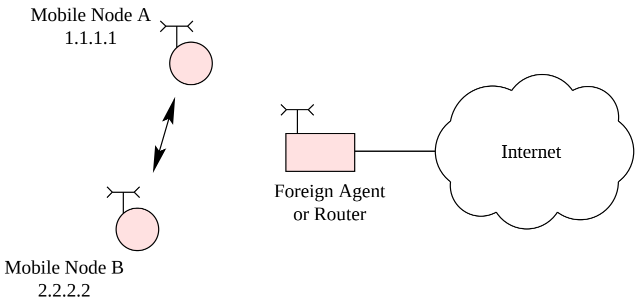

Figure 1. Different subnets, same link

Consider the case depicted in Figure 1. Mobile nodes A and B do not share IP subnets, but they do share the same link. They should be able to communicate directly. With conventional IP, however, they must use a router to communicate. Mobile-IP has the same problem, as the mobile nodes must use a foreign agent (and, in some implementations, a potentially distant home agent) to communicate. This is very wasteful, as the data travels over the medium used by the mobile nodes twice. Worse, if the router fails, A and B are completely cut off from each other. An intermediary should not be required for the nodes to communicate.

2.1.2. Same Subnet, Different Links

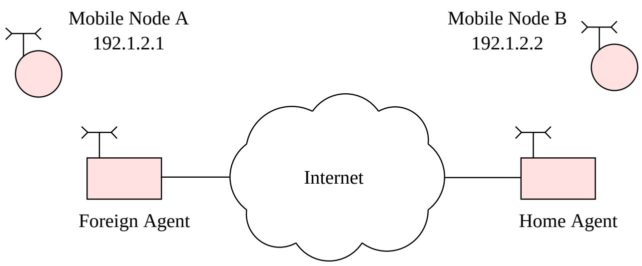

Figure 2. Shared subnet, different links

Figure 2 shows two mobile nodes that possess a common subnet. However, because of the physical separation, they are unable to send packets to each other directly. If they use the shared subnet

ARP algorithm, they cannot communicate, as each assumes the other is local.

In a sense, the shared subnet assumption is an optimization that allows systems to determine whether or not a target is local. This optimization is seriously flawed for communication between mobile nodes on any link type, and causes even more complications on radio links.

2.1.3. Wireless Cells and Subnetting

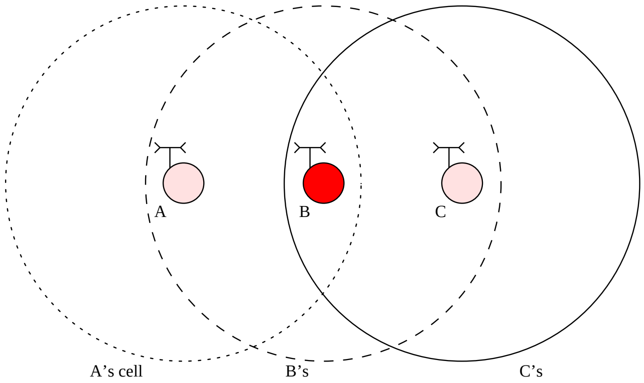

Figure 3. Three radio nodes. A can hear B, but not C, etc.

Figure 3 depicts three radio-based systems, A, B, and C, arranged from left to right. Each has a LAN broadcast radio cell that is roughly circular. A and B are in each other's cells, so they can hear each other. B in turn can hear C. C can hear B but not A, and A cannot hear C. Assume that A, B, and C all share a common IP subnet.

As described in section 2.1.2, the shared IP subnet assumption means that A cannot send packets to C. A will try to use ARP to determine C's MAC address, but A's ARP messages will not reach C. In addition, conventional low-level subnetting not only gives A no clues about whether C is reachable, but B can do nothing to help A. Indeed, since A and C are on the same IP subnet, B has no reason to believe it should help!

Furthermore, with radio cells, assumptions about transitive reachability also come into question. For example, B cannot assume that, because it can reach A and C, then A and C can talk directly. Unfortunately, the transitive reachability assumption is used in current IP systems. ICMP redirection messages are based on this assumption.

Consider what happens if B and C are routers that use ICMP redirection. When A sends a packet thru B that should be routed through C, B may send A an ICMP redirect instructing it to use C. A, unable to reach C, will thereafter be unable to reach the hosts for which C carries traffic.

In summary, the conventional IP subnet model at the link layer does not fit wireless cell reachability needs and must be reconsidered.

2.2. ARP Spoofing

In this section, we consider security problems based on attacks via the ARP protocol.

In his classic paper on TCP/IP security, Steve Bellovin describes how easy it is to spoof IP-to-MAC address bindings using ARP [2]. Assume that an attacker is trying to steal the IP address of some victim on the same net. To make the attack, the attacker uses ARP so that other hosts on the network bind its MAC address to the victim's IP address, instead of having the victim's MAC and IP addresses bound together as the victim would desire. This way, packets destined to the victim instead go to the attacker.

ARP spoofing might be combined with Mobile-IP. Imagine that node A in Figure 3 is the attacker, B is a Mobile-IP foreign agent, and C is the victim. A overwrites B's ARP cache entry for C to steal the Mobile-IP tunnel from the home agent to the foreign agent and receive C's packets. The irony of this attack is that A can evade the authenticated Mobile-IP registration mechanism with a low-level attack. The theft of the IP to MAC address binding has subverted a network-layer tunnel.

Furthermore, the non-transitive nature of radio links means that A can rewrite B's ARP cache without C finding out. With some topologies and/or the help of a confederate, the attacker might even forward the Mobile-IP UDP registration packets returning from the home agent back to the victim to keep the Mobile-IP connection fresh.

The above demonstrates that ARP is not a good foundation for Mobile-IP, especially on radio links where a radio cell might easily extend outside buildings. While Mobile-IP has mechanisms for authentication of its UDP control messages, including replay prevention, ARP can be abused to circumvent these protections.

3. Protocol and Implementation

3.1. Overview of the Protocol

In Mobile-IP, each mobility agent broadcasts ICMP Router Advertisement packets ("beacons") to inform nearby mobile nodes that it is available to carry traffic for the mobile nodes. In the ad hoc protocol, all communicants broadcast beacons, not just Mobile-IP agents. We have augmented the ICMP Router Discovery packets with the MAC and IP addresses of the sender and authentication akin to mechanisms used in Mobile-IP's UDP registration scheme.

It is important to understand that no other packets are exchanged in the protocol. Beacons are transmitted without regard for what other hosts are capable of receiving them, and any beacons a host receives are silently processed, regardless of whether the beacon was authenticated successfully. Any host (mobile node or agent) receiving a beacon is immediately able to send to the host that transmitted the beacon. The resulting protocol is similar to the ISO's ES-IS protocol.[4]

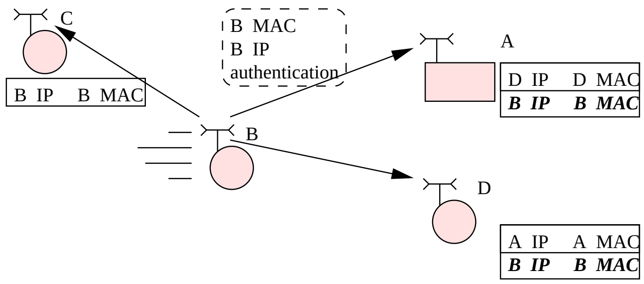

Figure 4. Mobile node B moves into range of A and D and beacons.

[4] See Perlman for a comparison of ARP and ES-IS [13].

Figure 4 depicts the protocol in use. Mobile node B has moved into range of hosts A and D and has transmitted a beacon. When A and D receive the beacon, they check its authenticity and, assuming it is authentic, add the MAC-to-IP address binding contained in the beacon to their tables of known bindings. Node C already has a prior binding for B, so it does not need to add to its bindings.

In our environment, a Mobile-IP agent typically uses ARP on wired interfaces and the ad hoc protocol on wireless interfaces. Mobile nodes only use the ad hoc protocol. While hosts might be able to simultaneously use both ARP and ad hoc modes on the same interface, doing so negates the security benefits of the ad hoc protocol.

Mobile-IP agents and mobile nodes beacon at a fixed rate. Agents normally beacon once per second, while mobile nodes beacon less frequently — typically only once every ten seconds. Mobile nodes also transmit a beacon immediately before sending a Mobile-IP registration request. This beacon ensures that the agent knows the mobile node's MAC address when it sends a registration reply.

Beacons contain the following information:

- IP header
- ICMP header (including the sender's IP address)
- Mobile-IP extensions
- MAC address extension
- network authentication extension
- optional ad hoc authentication extension

3.1.1. Beacon Authentication

Each beacon must contain the MAC extension and at least one authentication extension. The authentication data in the extension is computed over the ICMP header, the initial part of the authentication extension, and all extensions between the two. Thus, the authentication extension(s) must come after the data to be authenticated.

Table 1
Authentication Extension Format

name	description
tag	this is an authentication extension
length	size of extension in bytes
flags	type of authentication (network or ad hoc)
time	time beacon was sent
agent	IP address of mobile node's agent
spi	security parameter index
hash	authentication checksum

Table 1 describes the authentication extension format. The flags field tells the listener whether the authentication suffix is of the "network authentication" or "ad hoc" type (explained in section 3.1.2). The agent IP address may be set when a mobile node decides to use a particular agent. This allows other agents to quickly discard the beacon and avoid the authentication computation. The time portion of the beacon, which includes both seconds and fractional seconds in NTP format [10], makes replay attacks harder. The security parameter index is an opaque identifier shared by the sender and recipients that indicates which algorithm and key to use to authenticate the packet.

Nodes share a secret key they use as input to a MAC algorithm. Our implementation only provides keyed MD5 [15], the MAC algorithm used in Mobile-IP. The authenticated information includes at least the IP and MAC addresses of the sender, thereby authenticating the binding of those addresses.

3.1.2. Types of Keys

Agents and mobile nodes define logical links via the network authentication extensions and their keys. The implementation provides for two keys. One key, the *network authentication key*, is for daily use at a given site. The other, the *ad hoc key*, may be generated on the spot for brief "face-to-face" meetings between systems from different sites. For example, the key might be generated via a shared passphrase when one mobile node has a short encounter with another.

3.2. Compatibility with Standard Mobile-IP

In order to maintain compatibility with Mobile-IP, our agents and mobile nodes can be switched between ad hoc mode and "ARP mode". The latter is the behavior Mobile-IP specifies for mobile nodes.

Switching between ad hoc and ARP modes is a high-level security policy decision. The system does not dynamically switch from one mode to the other. The user interface lets the user determine what protocol is in use on the link. The user may then switch to ARP mode if necessary and forego the security protections provided by the ad hoc protocol.

3.3. User Interface

We have implemented a simple program that lets the user with monitor and control the mobile nodes software, including its security features. The program provides high-level security policy information and status indicators about link security.

The interface lets the user control the link layer mode and what authentication extension types are accepted. The user can independently control the acceptance of beacons authenticated with the network authentication key and the ad hoc key.

The user can generate an ad hoc key by entering a passphrase. This is useful for face-to-face meetings: One person can write a passphrase on a blackboard for everyone to install on their mobile nodes for the duration of the meeting. At the conclusion of the meeting, they disable acceptance of beacons authenticated with the ad hoc key. As explained in section 5.1, this convenient way of forming ad hoc networks helps protect network security.

The interface also provides information about the beacons the mobile node is receiving. Authenticated beacons cause the "valid beacons" indicator to flash. Various other indicators allow the user to tell if authenticated beacons are present or if certain gross errors exist (such as authentication failures due to invalid keys). If no indicators are blinking, the user knows that there are no agents nearby or that there is a problem with the link.

4. Benefits

In section 2, we described various problems that the ARP subnet model causes in a mobile environment. We now explain how the beaconing protocol addresses those problems.

4.1. Subnetting

The ad hoc protocol directly addresses the problem depicted in Figure 1. Mobile nodes that do not share a common subnet can communicate directly without any sort of router. If the mobile node hears beacons from another system (and they share a key), it can install a link-layer route for the other system. This avoids the need for a separate router and related infrastructure that is required by systems using Mobile-IP. Note that A and B do not communicate directly because they share a subnet (they do not). Rather, they can communicate because they share a key and can receive each other's beacons.

The problem in wireless networks where one machine tries to reach another machine on its subnet that is out of range (as described in Figure 2) is also eliminated. Since the destination is out of range, the source never receives beacons from it, so no link-layer route is installed. The node instead uses normal IP routing mechanisms.

The first problem illustrated by Figure 3 may be solved if B is the default router for A and C. B will install link-layer routes for A and C and forward packets between them. However, if B does not forward packets (most mobile nodes do not, for instance), A will not be able to send packets to C. That functionality requires a multi-hop ad hoc routing protocol.

The ICMP redirect problem may be solved by making redirects available only on a per-interface basis. Thus, the transmission and recognition of ICMP redirects should be turned off on wireless links, while still permitted on an Ethernet or other wired link.

4.2. Security Benefits

The ad hoc protocol directly addresses the threat of an attacker stealing a host's packets by making other nodes bind its MAC address to the victim's IP address. Hosts implementing the protocol completely ignore ARP packets on designated links. IP-to-MAC address bindings are created only when an authenticated beacon is received.

The protocol eliminates the threat of a denial-of-service attack caused by an ARP spoof destroying a link-layer route to a host. This is important because an ARP spoof can cause damage even when the data itself is protected at the network, transport, and/or application layers.

The protocol also limits what attacks an unwanted visitor to a wireless network can make. While the attacker is able to send packets to other hosts on the wireless network, and possibly have those packets routed on to other hosts, the attacker is unable to get responses to its packets because none of the ad hoc nodes have a link-layer route for it. The attacker cannot make any attack that requires it to be able to converse with the victim. While this "routing gateway" function is no substitute for a firewall, it may be a useful part of an overall security system.

5. Security Analysis

5.1. Key Protection

Good security is more than just correct implementation of sound protocols. A careless user can easily destroy the security of any system. This is why we emphasize and make available a key that can be easily manufactured, given away in short-term face-to-face situations, and destroyed when no longer needed. Users have little temptation to give away the network authentication key.

5.2. Replay Attacks

So far, our description has not addressed the threat of replay attacks. An attacker might record a link-layer authenticated beacon at one foreign agent and then later retransmit it. In this case, the victim may not be present and hence unaware of the attack.

Our foreign agents do not install the link-layer route until the Mobile-IP UDP registration has been accepted by the mobile node's home agent. This means that an attacker must also breach the Mobile-IP security relationship. Thus, at agents, we take advantage of Mobile-IP replay protection and authentication.

Replay attacks are therefore only a threat to communications between mobile nodes. To address this, beacons contain a timestamp. The 64-bit timestamp permits hand-offs with sub-second granularity, should it ever be necessary.

In our current implementation, mobile nodes do not compare the timestamp with an internal time source, but simply make sure that timestamps increase. The timestamp serves a role similar to that of a nonce, except that the replay identifiers are ordered. To attack this replay protection scheme, an attacker must either record a series of beacons for replay or forward beacons it receives.

A more sophisticated implementation might instead check that the time contained in the beacon is close to the current time. Such an approach cannot work unless both the sender and receiver can securely determine the current time, which may be impossible in some circumstances. In particular, ad hoc networks may be unable to access resources outside the local network.

5.3. MAC Spoofing

The use of authenticated MAC and IP address bindings makes spoofing substantially more difficult, but not impossible. In addition to IP spoofing, an attacker might choose to spoof a MAC address. Some LAN controllers (WaveLAN controllers in particular) can be configured to use an arbitrary MAC address. However, since the attacker is using the victim's MAC address, the victim may see responses to the attacker's transmissions (and possibly the attacker's packets as well) because the two systems share the MAC address. The victim will then have "self-defense" opportunities not available with an ARP spoof.

Refer again to the radio setup shown in Figure 3. Suppose that A spoofs C's MAC and IP addresses. C does not receive A's packets because it is out of range. However, C still receives packets travelling to A via B. C can then respond to those packets. For example, if A starts a TCP connection, C can issue a TCP reset against it [16].

In general, a wireless node should be alert to anything peculiar. It should inform the user about network security events of interest, including packets with its own MAC address, unexpected TCP resets, and ICMP "no such UDP port" errors.

The protocol described here protects routing connectivity, not user data. An attacking host might simply listen to passing traffic. This is easy to do and almost impossible to detect, and works regardless of how hosts learn each other's MAC addresses. Packet confidentiality must be protected using IPSEC or other mechanisms [1]. IPSEC's authentication and encryption complements our ad hoc mechanism. An authenticated ad hoc link-layer routing protocol coupled with IPSEC between a mobile node and a home agent form the "moat" and "castle wall" for a more secure roaming node.

6. Conclusion

This paper presents two key ideas. First, the conventional IP subnet model ARP uses is inadequate for mobile systems at the link layer. It may be replaced with a protocol where beacons are used to determine reachability. This paper describes such a protocol. The protocol is naturally integrated with Mobile-IP as agents already beacon.

Second, the beacons binding MAC and IP addresses are authenticated. This gives us several security opportunities, including the ability to wall off agents and not let "strangers" access network resources. It minimizes the danger of link-layer spoofing attacks. Authenticated beacons also improve the overall security of Mobile-IP.

The protocol makes the subnet of corresponding hosts irrelevant for the purpose of determining link-layer reachability. If two nodes can communicate at the link layer, our protocol permits them to directly exchange TCP/IP packets. This is important in wireless networks, where the ability to communicate is not transitive, and it makes constructing single-hop ad hoc networks as simple as, or simpler than, setting up wired networks.

Below we address two remaining issues: beacon scalability and key management.

6.1. Scalability

Beacon scalability is only an issue if there are a lot of nodes compared to the available bandwidth. Beacons are about 100 bytes long, and get transmitted once every ten seconds. WaveLAN, which typically has a throughput of about 1 megabit per second, should be able to support about 13,000 beaconing nodes. We have observed that WaveLAN performance suffers when more than 4 nodes run simultaneous TCP downloads, so beacon scalability is unlikely to be an issue.

Caceres has already suggested that agents need to beacon more often to permit quick mobile node hand-off [3]. Agent beacons represent the lifeline to the local wired infrastructure and/or the Internet. The most important consideration for a mobile node may be its umbilical cord connection to the local agent or router. In many situations, ad hoc connectivity to other mobile nodes does not matter. This is why mobile nodes beacon more slowly than agents.

Sending a beacon immediately before each Mobile-IP registration message suggests a possible optimization. Systems might switch between active beaconing, as described above, and more passive beaconing. With passive beaconing, the ad hoc protocol is only used for communication between the agent and the mobile nodes. This mode would be used when large numbers of mobile nodes congregate but only want access to the Internet.

Sparse beaconing could even be integrated into the Mobile-IP registration packets. The registration packet could include the authenticated addressing information and the mobile node would have no need to beacon.

6.2. Key Management

Our current system assumes all nodes in the campus network share a symmetric key used only for beacon authentication. In addition to problems with scalability, every agent and mobile node at the site has to know the network authentication key. Once everyone knows a secret, it is not much of a secret.

The symmetric keys might be replaced with public key cryptography [6]. Portland State University is experimenting with replacing the Mobile-IP and ad hoc symmetric key authentication with public-key authentication. Public-key signature and verification of beacons and Mobile-IP registration messages is feasible, even though transmitting such a signature requires more link bandwidth. Every node can possess its own key and simply sign its beacons and registrations. The distribution of certificates so that mobile nodes and agents can verify beacons is a higher-level problem we hope to consider in another paper.[5]

6.3. Availability

The source code for the ad hoc routing system is part of Portland State University's Mobile-IP implementation, which may be accessed via `http://www.cs.pdx.edu/research/SMN/`.

[5] Zao describes one method for mobile nodes to obtain certificates [17].

References

[1] R. Atkinson. Security Architecture for the Internet Protocol. Request for Comments, RFC 1825, Internet Engineering Task Force, Aug 1995. Naval Research Labs.

[2] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2), March 1989.

[3] R. Caceres and V.N. Padmanabhan. Fast and Scalable Handoffs for Wireless Internetworks. In *Proceedings of ACM, MobiCom*, 1996.

[4] D. Comer. *Internetworking with TCP/IP*, volume 1, pages 150–151. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[5] S. Deering. ICMP Router Discovery Messages. Request for Comments, RFC 1256, Internet Engineering Task Force, Sep 1991. Xerox PARC.

[6] W. Diffie and M. Hellman. New directions in cryptography. In *Proc. IEEE Trans. on Info. Theory, IT-22, 6*, pages 644–654, 1976.

[7] C. Huitema. *Routing in the Internet*, chapter 5. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[8] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, 1996.

[9] G. Malkin. RIP Version 2 Carrying Additional Information. Request for Comments, RFC 1723, Internet Engineering Task Force, November 1994. Xylogics, Inc.

[10] David Mills. Network Time Protocol (Version 3). Request for Comments, RFC 1305, Internet Engineering Task Force, March 1992. University of Delaware.

[11] J. Moy. OSPF Version 2. Request for Comments, RFC 2178, Internet Engineering Task Force, July 1997. Cascade Communications Corp.

[12] C. Perkins. IP Mobility Support. Request for Comments, RFC 2002, Internet Engineering Task Force, October 1996. IBM.

[13] R. Perlman. *Interconnections: Bridges and Routers*, pages 195–198. Addison-Wesley, Reading, Massachusetts, 1992.

[14] D.C. Plummer. An Ethernet Address Resolution Protocol. Request for Comments, RFC 826, Internet Engineering Task Force, 1982. Symbolics Inc.

[15] R. Rivest. The MD5 Message-Digest Algorithm. Request for Comments, RFC 1321, Internet Engineering Task Force, April 1992.

[16] W.R. Stevens. *TCP/IP Illustrated*, volume I, page 227. Addison-Wesley, New York, 1994.

[17] J. Zao, S. Kent, J. Gahm, G. Troxel, M. Condell, P. Helinek, N. Yuan, and I. Castineyra. A Public-Key Based Secure Mobile-IP. *MobiCom97*, September 1997.