Authenticated Ad Hoc Routing at the Link Layer for Mobile Systems

James Binkley
jrb@cs.px.edu
Technical Report 96-3
Computer Science Department
Portland State University
Portland, Oregon 97207

Abstract

We have built an experimental link-level authenticated ad hoc routing protocol and integrated it with the PSU version of Mobile-IP. The routing protocol is intended to address survivability and link security issues. In our modified version of Mobile-IP, we authenticate ICMP router discovery packets, which are sent by both Mobile-IP agents and mobile nodes. The router discovery packets allow systems to exchange authenticated MAC and IP addresses. Topology problems caused by tying IP subnet schemes to routing on radio links are solved. The security problems associated with ARP spoofing are also reduced. This link-level protocol might be integrated with Mobile-IP on links where increased security is needed, and is complementary to IP Security (IPSEC) mechanisms. It is not intended to function as a higher-level (mobile node as router) ad hoc routing protocol, but might be integrated with such a protocol.

1 Introduction

New networks are being created based on nodes communicating at the link layer via wireless interfaces such as infrared, cellular radio, and wireless lan radio. Lan radio devices such as Lucent's WaveLAN[1], are capable of broadcast and multicast similar to ethernet. Networks based on ethernet-like radio broadcast bring certain topological and security challenges with them not previously encountered with wired technologies.

Mobile-IP [Per96] specifies how end nodes may move across IP subnets and retain an IP address, thus dealing with the problem of how a Mobile host can move and retain

[1] They have a web page at http://www.wavelan.com

a fixed IP identity. Mobile-IP is a network-layer routing protocol. It focuses on how packets from Home (via the *Home Agent* or HA as a routing proxy) may be forwarded to a *mobile node* or MN when it has wandered to foreign subnets. One might claim that the Mobile-IP protocol is focused on the network-layer only. There is some focus on link-layer concerns but for the most part Mobile-IP tries to deal with how a mobile Node may wander from its subnet and yet retain IP unicast connectivity.

A group of mobile systems, say laptops carried by people from different organizations (with different IP subnets), might show up for a face to face meeting in a hotel room, bar, business presentation, or lecture hall. Such a collection of systems should be able to talk to each other sans network administration and without requiring a Mobile-IP agent or normal IP router to be available. We may call such a situation an *ad hoc network*. Depending upon the size of the link-layer wireless technology, the network might or might not need to span multiple hops.

In this paper [2], we present an experimental ad hoc routing protocol that is focused both on routing survivability on a multicast radio link and on certain fundamental routing security issues. The protocol is integrated with Mobile-IP and is intended to be used on links deemed to be fundamentally less secure than ethernet; for example, lan radio links. However the protocol could be used on ethernet too if administrators felt the the ethernet link was subject to physical compromise. We address certain problems inherent in both IP subnetting and radio link topologies by adopting the postulate *"I beacon, therefore I am"*. In this system, we have enhanced the ICMP router discovery [Dee91] packets used in Mobile-IP so that all nodes including both agents and mobile nodes beacon. Thus any system can directly hear what other systems are reachable. This simple mechanism solves problems inherent in IP subnets and allows ad hoc mobile nodes from different subnets to communicate directly without agents or routers.

The protocol also deals with certain link-layer routing security problems. We have done away with ARP [Plu82] and our beacons authenticate the MAC, IP address combination. Thus we have made ARP spoofing more difficult. It has long been observed in network security circles that ARP may be easily spoofed since one node may overwrite another node's (MAC address, IP address) pairing in an ARP cache, replacing the spoofed system's MAC address with the spoofer's MAC address. Mobile nodes running our version of Mobile-IP can use either our beaconing protocol, or they may choose to still use ARP, depending on the link they are attached to (and the type of network adapter they are using). We provide a user interface that does not make a dynamic decision to switch link-layer routing protocols, but allows the user to learn the protocol in use at a link. The user can then decide the security policy he/she wishes to employ, and communicate this policy to the underlying system software.

Section 2 of our paper presents a discussion of background issues and problems. Section 3 presents our architectural solution. Section 4 presents a security analysis of the solution. In Sections 5 and 6 we present some operational details including a

[2]This work was supported by the U.S. Department of Defense Advanced Research Projects Agency and the U.S. Air Force Rome Laboratory under contract F30602-95-1-0046.

short discussion of relevant aspects of our X user interface and some measurements of the cost of authentication. In section 7 we present final conclusions, criticism, and discussion of possible future work.

2 Background Issues

Why do we feel we need to replace ARP with an authenticated link layer addressing system? The reasons include lack of security, problems inherent in the IP subnet architecture, and our desire to more closely integrate ICMP router discovery packets with mobility in general and Mobile-IP in particular.

From a security point of view, IP address spoofing at the link layer is too easy. Steve Bellovin pointed out in his classic security paper on TCP/IP [Bel89] that ARP spoofing is dangerous. One system A can pretend to be another system B by responding to ARP broadcast queries for B with a binding of (B's ip address, A's mac address). A system may overwrite another system's ARP cache entry with a gratuitous ARP. This function is sometimes necessary due to a controller board change. Thus ARP spoofing may even be viewed as a feature. However ARP overwrites are not good from a security point of view, especially when coupled with higher level protocols that only authenticate a host in terms of its IP address.

There is probably no reason the ARP protocol cannot be extended to include authentication, even though on the surface the protocol does not support an extension mechanism. However we chose instead to place authenticated MAC addresses into the ICMP data payload in ICMP router discovery packets. Mobile-IP uses such packets and has already extended them in order to announce the existence of agents. We feel our ICMP router discovery extensions (MAC and otherwise) integrate with Mobile-IP.

There is an important problem that is often coupled with ARP, but is not really ARP's fault. That problem is: how are IP subnets to be interpreted at a link where ad hoc networks may occur?

By definition, mobile systems based on Mobile-IP bring with them the semantic that any given link-layer may have more than one IP subnet resident. The notion of "one wire, one IP subnet" has been under stress for awhile partially due to network exhaustion in IP version 4. Any wired segment may have to tolerate greater than one class C subnet, since such a subnet is limited at a maximum to 254 hosts (less if further divided by subnet masks). Mobile-IP makes a necessity out of the notion of many subnets on one segment since a Mobile-IP node retains its IP address as it moves from one subnet point of attachment to another. A subnet with a *Foreign Agent* (a FA is a Mobile-IP entity that allows the mobile nodes access to the network) present may potentially have many mobile nodes present, each with a IP net address different from all the other Mobile systems including that of the interface at the Foreign Agent itself.

In terms of TCP/IP routing, subnetting has a number of functions. We might distinguish a "high-level" function of routing aggregation; i.e., it is common knowledge at this point that unicast IP routing is based on the exchange of a two-tuple consisting of (IP address, net mask). [3] Core IP routers are under stress from too many routes in routing tables and routing aggregation is a grim and important necessity. On the other hand, subnetting has a low-level function that typically manifests itself in the routing implementation of a particular operating system. A subnet mask is AND'ed with a IP destination address to determine if the destination is either on the link or not on the link. An algorithm that has been used in very simple TCP/IP implementations might be stated as follows: [4]

```
if dest IP address AND subnet mask indicate same subnet
    assume same link and ARP for the destination
else
    send to default router
    (use ARP to talk to router)
```

The router itself must share a subnet/subnet mask association with the leaf node. As a consequence, if a link has greater than one subnet resident on it, a node on IP subnet A cannot directly send packets to a node of subnet B, even though they share a physical connection. The router must have one IP address on both subnets and packets must be sent twice on the same wire. Mobile nodes with different subnets and a shared radio link cannot talk directly to each other even if one is laid on top of the other. They would need a router to communicate, of course. This is truly absurd for mobile systems. We feel that the notion of IP subnets as a link-layer routing basis for mobile systems is not useful. Current end-system routing algorithms may not exactly be as stupid as the above algorithm, but the subnetting semantic will probably exist [5]. The current 4.4 BSD routing algorithm employs a mechanism that is not as brutal as the above algorithm, but has the same result. A *clone* or *direct* route [Ste95] is installed for every interface (link) and attempts to talk to same subnet nodes result in a magic cloned link-layer route (LLINFO) that is automatically installed and used. The end result is the same.

There is a further problem inherent in the use of certain kinds of link-layer hardware that needs to be addressed in link-layer communication between all mobile nodes. This problem is tied to the notion that a shared subnet means you can assume reachability between two nodes. One might distinguish two kinds of hardware; e.g., wired broadcast-based media like ethernet, and media like WaveLAN(tm), a 2 megabit spread-spectrum radio LAN technology. We could state that ethernet has an attribute called: *shared address reachability.* Radio (and infrared) devices that are highly likely to be used

[3] Actually we wish this was true, but it will not be until RIP v1 is replaced by RIP v2 or OSPF everywhere.

[4] See [Com91] for an example.

[5] To be fair, it should be noted that the above algorithm only uses ARP for direct delivery on a interface that speaks MAC addresses. We cannot blame the ARP protocol itself as it doesn't care about subnetting. Historically it came first, and nothing prevents it from being employed in a subnet-free way.

with mobile nodes do not have that attribute. In their case, we have to say: *"shared addresses do not mean reachability"*.

In a trivial sense, a mobile node cannot assume that a cousin with a shared IP subnet must be resident simply because of the shared subnet. The cousin might be away on some other subnet, or unreachable in a closet somewhere.

But there is a more serious flaw lurking in the shared reachability notion. Assume there are four systems, A (MN), B(FA), C(MN), and D(router) arranged from left to

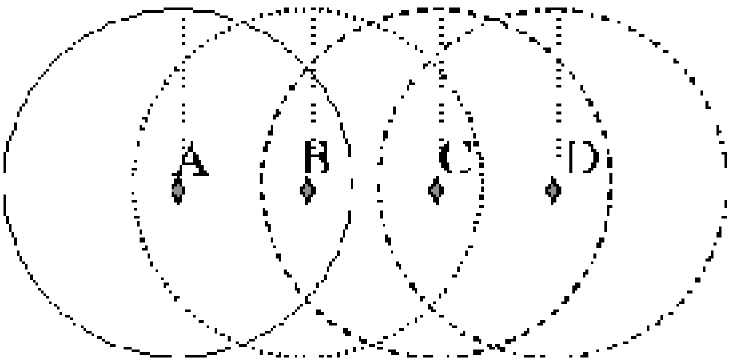

Figure 1: 4 Radio Nodes, A can hear B, but not C, etc.

right. (Please see Figure 1). Assume that each has a radio cell that is roughly circular and that A and B's cells overlap, but the cells for A and C do not overlap. B in turn can hear C (and A), but cannot hear D. D can only hear C. Assume that A, B, C, and D all share a common IP subnet and that A and C are mobile nodes, B is a Mobile-IP Foreign Agent, and D is an agent (or router).

If A tries to talk to C, with current routing implementations, it will ARP for it, without success. If C moved to a Foreign Agent far away, A might still ARP for it. To add insult to injury, the ICMP redirection mechanism could possibly rear its (in this case, ugly) head. Assume that A is using B as its FA, which enables it to talk to some other host X. B and D might be on the same interior ethernet subnet. Hence B might "know" due to IP routing assumptions, that D is a better router "on the same link" in the case of a given destination X. It might then send a redirect to A to tell it that it should use D to get to X. With many current kernel routing implementations, A would simply accept that routing redirect and no longer be able to talk to X. Given the above two problems, we assert that the assumption of shared address reachability due to a shared subnet address is not valid on multicast-based "lan" radio links. It makes sense on ethernet. It makes less sense on radio links and other similar link technologies.

From a survivability point of view, mobile nodes from different IP subnets should be able to talk directly to each other without needing a router. They should be able to flush MAC address state in other systems easily (their MAC addresses might change if a PCMCIA card changes) and they should ignore such things as ICMP redirects. There should certainly be no argument that an ad hoc network is not useful or needed. Our proposed protocol deals with link-layer reachability and addresses a fair number of the problems mentioned above [6].

It is one form of an ad hoc routing protocol, but it is not multi-hop and only applies to same link reachability. Other research is being done in higher level ad hoc routing. For example, Charles Perkins at IBM [PB94] has proposed a vector-distance protocol that would allow participating Mobile Nodes to route each others packets. Dave Johnson [JM96] has proposed a multi-hop protocol based on source-routing. It is possible that we may need a number of ad hoc protocols based on different attributes including link technology and the type of motion expected. Certainly high-speed movement (F16s), might have to fallback on flooding. A set of mobile systems that are moved from one building to another, where they will sit for days after deployment, could use a simple vector-distance based technique where convergence takes a bit of time. If scarce bandwidth is a concern, systems could take advantage of "on-demand" end system source-routing.

Our simple authenticated beaconing proposal works for a number of ad hoc laptops sitting in a big open space using a technology like WaveLAN. One might also argue that it is not incompatible with a multi-hop ad hoc protocol. Our proposal informs systems about direct reachability and it might be subsumed by or run alongside a higher-level routing protocol in the same way that ES-IS [Per92] is subsumed by IS-IS (ISO's routing protocol similar to OSPF [Hui95]).

3 Solution Overview

In this section, we will present an overview of how the ad hoc system operates.

We have enhanced the ICMP Router Discovery packets (beacons) used in Mobile-IP so that they contain MAC and IP addressing information, and authentication similar in style to that currently used in the UDP registration packets in Mobile-IP. There are no other packet exchanges in the protocol. We assume that if you can hear a beacon, you can talk to the sender.

The protocol is fundamentally multicast in outlook, since a beacon may be addressed to either IP limited broadcast (255.255.255.255) or to the multicast group 224.0.0.1. All systems, either Mobile-IP agent (HA/FA) or mobile nodes (MNs) will beacon at a certain periodicity. We are currently experimenting with agents beaconing at the rate of 1/sec. Our MNs beacon at a slower rate, say 5 or 10 second intervals. We assume that an interface is either ARP-capable or adhoc-capable. A mobile agent typically will have an ARP-capable ethernet interface and an ad hoc capable WaveLAN interface. The MNs, of course, only use the ad hoc system. It is not desirable for security reasons to mix ARP and adhoc on the same link.

[6] ICMP redirects are not addressed. They might be ignored on a per interface basis.

Beacons logically consist of the following headers:

(IP, ICMP, Mobile-IP extension, optional MAC,
optional location, optional network authentication,
extremely optional ad hoc authentication)

If a link is ad hoc, beacons sent on it must contain the MAC portion, and at least one authentication suffix, which must come last. Both MAC and IP addresses must come before all authentication headers, else they would not be authenticated. The IP address itself is in the ICMP portion. We will discuss the authentication headers more below.

An optional information string may be provided. Our agents currently state where they are, along the lines of "Bridgeport: PCAT room 128C". Mobile nodes might state their user's email address or might simply not use this facilty at all if privacy is desired. Location information is displayed in our X user interface for laptops and allows a user to easily discern the current Mobile-IP agent's name and location.

One or two authentication extensions must complete a packet. Our authentication mechanism here is analogous to the Mobile-IP authentication mechanisms; i.e, we authenticate via a shared secret based on a MAC algorithm (currently MD5, but there is a SPI (Security Parameter Index) in the authentication extension, and the algorithm in use could of course be changed). We will discuss the authentication header more below, but for now there are several important points that need to be made:

- We are associating a secret key with a link. Key distribution is currently "out of band" (a priori distributed) and as a software problem falls in with the general problem of multicast key distribution.
- The authenticated information logically includes at least the following information:

  ```
  (ip address, MAC address, a few other items)
  ```

  The authentication is done over the ICMP header (where the IP address is), and all higher headers up to and including the SPI in the authentication extension. Thus system "names" (addresses) are authenticated.
- There are at least two possible keys that may be concatenated. Each may be thought to be paired with a logical link.

Any agent (FA or HA) or mobile node that belongs to the same logical link define themselves in terms of a fundamental network authentication extension via one or more shared keys. For now, assume one key for the links at a given site. We call that key the *network authentication key*. Since one can easily argue, that the more a key is used, the less secret it is, we point out that each agent at a site might have its own key. In that case, the MNs would be expected to know a list of such *network authentication* keys. A second ad hoc key may be generated on the spot and given away for a brief period for face to face meetings between end systems from different sites. For example, the key might be generated via a shared passphrase fed to MD5 when visitors come to your site or when your laptop has a face to face rendezvous with some other laptop anywhere. We call this the *ad hoc* key. This key exists primarily so that no user ever gives away the site *network authentication* keys. Depending on the user interface policy setup, beacons are accepted if either key is successfully authenticated. This allows MNs that are at home to speak to the ad hoc systems (the visitors) directly and retain their own current agent connection. It also allows a site manager to teach a FA about the ad hoc key for the duration of a meeting, thus visitors may (or may not) be allowed controlled access to the Internet. Further, a site manager would not have to give a visitor a local network key that was in use by others, thus compromising the link.

In order to maintain compatibility with Mobile-IP, our agents and mobile nodes are capable of acting in two link modes. We call one mode "ARP" and the other "ad hoc". "ARP" mode means whatever Mobile-IP specifies for correct mobile node behavior with ARP. Users will get information based on beacon inspection from our user interface about what kind of links (agents) they are visiting, and can instruct their laptop to move from one mode to another by clicking on a button. This is a high-level security policy decision. A user configured with our ad hoc link system might choose to not speak ARP for security reasons, but we have decided to allow them to make that choice. The system itself will not dynamically switch from one mode to the other. Agents are presumably configured beforehand and can either use (Mobile-IP style) ARP or run with the ad hoc beaconing system.

Our mobile nodes send a beacon out in front of the UDP Mobile-IP registration. As a result, they can beacon more slowly if one feels that mobile node to mobile node interaction is not very likely.

The use of beacons as outlined above solves a number of routing problems. MNs that do not share a common subnet can now communicate directly sans router or agent. If you can hear a system (and you share a key), you can install a link-layer route for the other system. As another example, two systems that cannot hear each other directly but are present at a cooperating Foreign Agent, will simply route through the Foreign Agent because link-layer routes will be present in the FA's routing table. If those systems used ARP in the classic sense and did not share a subnet association, they would be able to talk, but their packets (sans a routing optimization not currently present in IPv4) would have to first go to the Mobile-IP Home Agent before returning to the FA.

Due to the previously mentioned ARP/subnet semantic, we can also cleanly heal certain problems that the Mobile-IP specification cannot (but in fairness, could claim is a link-layer/routing implementation problem). For example, when we use ARP paired with the clone route in our FreeBSD systems, two laptops that share a common subnet cannot talk to each other when they are both at different Foreign Agents. This problem exists because the per link clone route on each box causes them to assume that their peer is local. Since both systems at both FAs exhibit such behavior, even if one system behaved correctly, two-way communication would still fail. With our ad hoc scheme, they won't use the clone route (or ARP) and will know that the peer is not local, as there won't be local link routes in the routing table. As a result, thanks to Mobile-IP, packets will be correctly forwarded to the agents and the systems will be able to communicate.

4 Security Analysis

In this section we will show our authentication header packet format and briefly discuss various link-layer threats and associated defense mechanisms.

The header format is roughly:

size	name	info
byte	tag	this is an authentication header
byte	len	size in bytes
byte	flags	type of authentication, network or ad hoc
byte	pad	zero
long	seconds	NTP time in seconds
long	fractions	NPT time in fractional seconds
long	agent	agent IP address for mobile node
long	spi	Security Parameter Index (type of MAC algorithm used)
byte	hash[16]	MAC checksum (for example, MD5)

The flags field tells the listener whether the authentication suffix is of the "network authentication" or "ad hoc" type. The agent IP address field is set to 0 by agents and to a current IP address for an agent by a Mobile Node. This is an optimization that allows agents to quickly discard beacons that are not for them, before they authenticate the beacon. The 64-bit NTP portion includes both seconds and fractional seconds. We will discuss its function below in reference to replay attacks. The security parameter index specifies which MAC algorithm is used for the authentication checksum and the hash itself contains the checksum.

There are at least four threats that are addressed here:

1. ARP spoofing in the active sense; i.e., a spoofing system steals the IP address of another system by sending fake ARP messages. With traditional ARP spoofing, the spoofer changes its IP address and retains its MAC address.

2. denial of service due to loss of a link-layer route courtesy of an ARP spoof.

3. denial of service due to unauthenticated ICMP beacons sent by a fake agent. This is more dangerous in a radio topology if handoff mechanisms (like our current Mobile-IP handoff mechanism) are based on signal strength.

4. the possibility that a user might due to frustration decide to give away the network authentication key. This is why we want to emphasize and make available a key that can be easily manufactured, destroyed, and given away in short term face to face situations. In such a scenario, laptops from different domains may be present and all may be disconnected from the Internet.

Cases 1 and 4 are probably the most important cases that need to be considered. Case 2 is an outgrowth of case 1. Case 3 is a very real possibility and should not be considered unimportant. For case 3, our beacons are authenticated, and the MAC address is pushed up high enough to be covered by the authentication. We have also introduced some simple replay protection mechanisms that do not make replay impossible, but do make it more difficult. We will come back to case 3.

Without explaining exactly how to do this, (although we fear the mechanism is all too obvious), we have "successfully" performed an ARP spoof at a Foreign Agent. One mobile node (the attacker) overwrote the ARP cache for another system (the victim) at a Foreign Agent. The result was that the attacker, which assumed the IP address of the innocent node, was able to steal both the link layer routing entry and the tunnel between the Home Agent and the Foreign Agent. As a result, without running Mobile-IP, the attacking system was able to use the victim's Mobile-IP setup. The only Mobile-IP attribute that the attacking system needed was the ability to bind the default route to a agent not on its subnet. Of course, at a Home Agent the attacker would not need that ability, but at home, the scenario is exactly equivalent to traditional ARP spoofing. On a radio link, it is further possible that an ARP cache re-write at the agent might not be detectable by the victim, due to the non-transitive nature of the radio link topology we discussed earlier. Possibly, something even worse might be accomplished, as there might be a way for an attacker to feed the Mobile-IP UDP reply packets returning from the HA back to the victim and thus keep the Mobile-IP connection fresh.

We suggest that ARP is not a very good foundation for Mobile-IP, especially on radio links where a radio cell might easily extend outside buildings, even to passing vehicles. Mobile-IP has decent mechanisms for authentication of its UDP control messages including replay prevention. However we hijacked an authenticated MN/HA UDP Mobile-IP connection. Further if there had been authentication between the mobile node and the Foreign Agent, it would not have mattered. Mobile-IP authentication mechanisms only apply currently to its UDP control packets (registration/reply). We might judge the hijacking of such a Mobile-IP connection to be a more serious breach than a traditional ARP spoof, since we highjacked an authenticated tunnel.

One might think that ARP over ethernet is fine since ethernet links are (maybe) more subject to physical control than radio links. Under certain circumstances, this idea may very well be an illusion. There is nothing to prevent our authenticated beacons from being used on ethernet too.

IPSEC[Atk95] (IETF proposed network layer security) can be used to authenticate and encrypt ordinary data traffic, or new applications like *ssh/slogin* might be used for authenticated and encrypted TCP sessions. However neither Mobile-IP or IPSEC address ARP as a protocol. Use of IPSEC could reduce such a highjacking to simple denial of service (we hope). However our protocol could prevent a denial of service attack base on loss of link-layer routing or at least make it more difficult.

The use of authenticated MAC and IP addresses changes the playing field for spoofing. An attacking system that hopes to appear as a victim, must now spoof both the MAC address and IP address. Certainly some lan controllers have programmable MAC addresses and it is possible to adapt some other system's MAC address. Note however that since the attacker intends to use the MAC address of the victim, the victim may possibly see returning packets (or all packets) due to the shared unicast MAC address. For example, think again about our radio setup with the three systems, A, B (FA), and C. If A spoofs C, C might not be able to see A's packets. However packets returning from B to A would be seen by C because A and C would have the same unicast MAC address. C therefore has an opportunity that it would not have had with a simple ARP/IP spoof, where the attacker's MAC address is different. If A starts a TCP connection, C might issue a TCP reset against it. In general, C should be "alert" and notice anything peculiar going on at the link. It should note use of its own MAC address elsewhere. It would be useful to have a user interface that informs the user about network stack security events of interest including packets with your own MAC address, TCP resets, UDP ICMP "no such port" reachability errors and the like.

On the other hand, an attacker might simply be a passive listener. This may be less obvious and simpler than going to the trouble of reprogramming a MAC interface address. IPSEC's encryption mechanisms or higher-level application encryption would address such a threat. IPSEC provides network and transport authentication, and encryption and we feel our ad hoc mechanism is complimentary with it. An authenticated ad hoc link-layer routing protocol coupled with IPSEC mechanisms between MN and HA may be said to form the "moat" and "castle wall" of a more secure roaming node.

One question remains and that is what if anything should be done about possible replay attacks? What if an attacker soaks up your link-layer authenticated beacon at one FA and goes to another and plays it back? In this case, the victim would not be present to notice what is going on. Possibly replay attacks are a higher level problem and we should not worry about them here. After all, all you gain is a route in either a mobile node or agent. However, we have ruled replay attacks out at agents due to careful implementation. Our agents will not install the link-layer route (at either HA or FA), until the Mobile-IP UDP registration packet returns successfully. This means that an attacker must also minimally breach at least the MN/HA Mobile-IP security relationship. Thus at agents, we are taking advantage of Mobile-IP replay protection and authentication. The remaining question may be reduced to what, if anything should be done about MN to MN relations? We have experimentally specified a 64-bit NTP timestamp as part of the protocol. We want the mechanism to have sub-second granularity in case fast handoff is an issue. However for now we are limiting our mobile node implementation to simply check the timestamp issued to make sure that it is increasing, and to not compare it with an inner clock. This is a weaker implementation choice, and does not rule out a stronger time check, when and if a mobile node has a distributed source of time it can trust. In effect, the timestamp is simply a nonce, but a nonce that does not have to be saved over a reboot. Also it does make replay of authenticated beacons somewhat harder since a set of beacons must be recorded and played back in order to make the attack.

5 X User Interface - Security Aspects

We have implemented an X-based graphical user interface tool that provides both mobile node status information and control over the underlying system software. It includes several panels that are focused on security. Please refer to Figure 2. It shows

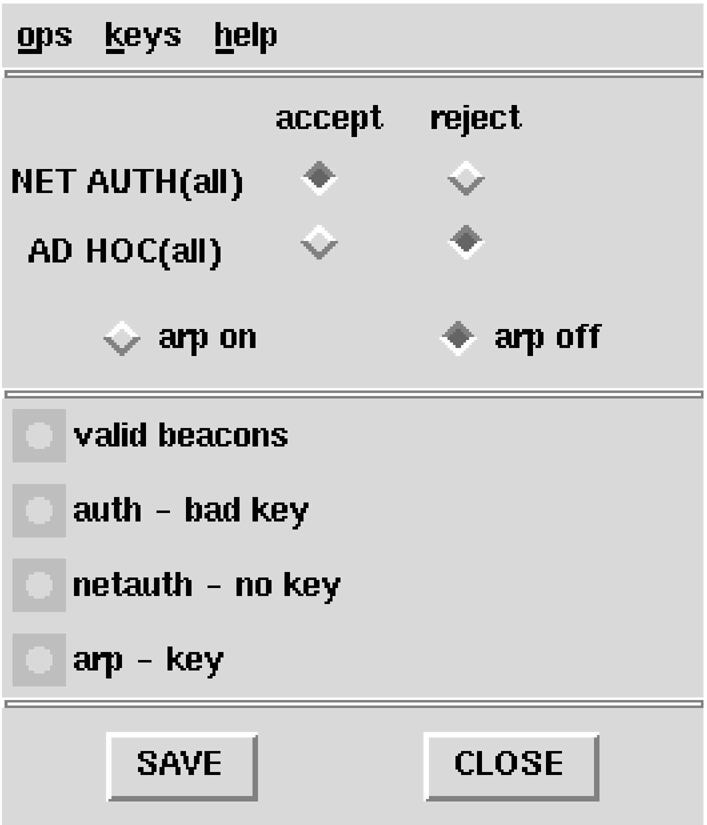

Figure 2: Link Authentication Panel

the panel that provides high-level security policy information and status indicators about current link security. One may choose ARP (in Mobile-IP terms) or ad hoc modes. If a system is in ad hoc mode, agents and mobile nodes are ignored if packets are lack authentication or the authentication fails; i.e., routes will not be installed.

If one chooses ad hoc security, it is necessary to have 1 or more keys in place and the keys must a priori be in place on other systems. For the ad hoc choice, on a per key type basis, one can choose to totally accept or reject beacons that use those keys. In this mode a system might be configured to accept beacons authenticated with the *net authentication* key and reject any beacons that show up with the *ad hoc authentication* key. Elsewhere in a part of the interface that is not pictured, the user can generate an ad hoc key via a face to face passphrase fed to the MAC algorithm. The passphrase is used to generate a shared key. For example, a professor could write the passphrase on a blackboard and the students might install it. The ad hoc users would then change the ad hoc policy to accept such authenticated beacons for the duration of a meeting, and then change the policy to rejec them when the meeting was over.

Various "dumb lights" exist that attempt to communicate the link status in terms of authentication to the user. The valid light shows the arrival of correct beacons (actually slower than they arrive) and if it isn't blinking, the user knows that there are either no agents nearby or there is a link problem of some sort. The various other lights allow the user to tell if authenticated beacons are present (or not present) or if certain gross errors exist (key failures).

6 Computational Measurements

We measured the speed of the keyed-md5 checksum over various size packets with two computers, a 75 megahertz Pentium laptop (that one may assume is a candidate mobile node) and a 166 megahertz desktop (agent). We wanted to determine how much computation is needed for authentication of beacons and to reflect on computation versus bandwidth issues.

The table below shows the computer speed, the number of bytes authenticated in a packet, the number of authenticated packets per second that could be sent, and the maximum algorithm throughput in kbytes/sec that can be supported for that size of packets. Note that the MD5 key is 16 bytes long. As a yardstick, real bandwidth for WaveLAN is on the order of 1 megabit/second. For example, with ftp (ncftp) transfers we can get about 117 kbytes/second on an unloaded radio link.

$computer$	$size$	$pkts/sec$	$kbytes/sec$
$75mhz$	60	33000	1980
$75mhz$	512	9680	4956
$75mhz$	1024	3683	3771
$166mhz$	60	74074	4444
$166mhz$	512	21549	11033
$166mhz$	1024	8203	8399

From examination of the maximum throughput figure in column 4, one can see that even the smallest packet size on the slower computer well exceeds the radio link speed. Our 60 byte sample is roughly the size of the authenticated part of our beacons. In fact, the computer's speed is more than enough to deal with the authentication task for current 10 megabit ethernet links, and on faster computers may begin to be able to deal with a 100 megabit ethernet link. Of course authentication algorithms that have more rounds or longer-keys will be slower. Still using authentication in routing is not something that should be questioned on the basis of efficiency. Mobile-IP Agents might be getting many beacons from nearby mobile nodes but they should be able to handle the computational load.

One may also observe that over WaveLAN, it would be fairly senseless to initiate a denial of service attack based on feeding authenticated packets with the wrong key to a node. A node can checksum the packets much faster than they can be written out to the link. One might just as well spray the radio link with packets in order to use up all available bandwidth.

7 Criticism and Conclusion

In summary, there are two important ideas in this paper. One is that the MAC, IP address binding should be authenticated in order to mitigate against spoofing attacks. The other is that notions of subnetting at the link-layer need to be reconsidered.

The physical security of radio links is harder to guarantee than that of wired segments like ethernet. Therefore we feel that the MAC address portion should be authenticated along with its IP address binding. As a result, attacks akin to ARP spoofing and certain denial of service attacks will be made more difficult and hopefully not worthwhile. Our protocol is one way this might be done, but ARP itself might be authenticated on ethernet in a manner similar to the way IPSEC authentication works with the IP protocol itself. What is needed is that the MAC address has to be made available at at higher level in a packet. This might be done in Mobile-IP by placing the MAC address in the ICMP beacon as we have done or it might also be placed in the UDP data payload of the Mobile-IP registration packet where it would be authenticated by current Mobile-IP mechanisms.

We feel that mobile systems should use beaconing to overcome certain link-layer routing problems caused by subnetting and by the inherent non-transitive nature of lan radio. It is simply not reasonable to assume that ad hoc mobile nodes from different IP subnets must have a router or agent to communicate. Mobile links made up of technologies like radio have their own problems and the notion that IP subnets can define what is "here" and what is "there" is also not reasonable. We propose a simple reachability mechanism as a solution, but there might be other solutions. Our proposal should not be regarded as an attack on subnetting "in the large"; i.e., it has nothing to do with the use of masks to define routing hierarchy. We simply point out that subnetting "in the small" cannot be used to determine local reachability on a radio link. Other mechanisms like ICMP routing redirects also need to be reconsidered.

However there are two larger issues that we should discuss, key management, and scalability.

Our protocol could use a multicast key management protocol on top of it, possibly along the lines of GKMP [HMR94]. We are targeted at lower-level mechanism and there is no reason why a higher-level multicast key management protocol could not be used, if and when, there is such a protocol with general acceptance in the IETF. In the meantime, we will assume "out of band" key distribution. Our current system can be enhanced to one agent, one key. We also have the "ad hoc" key, which can be generated with suitable carefully constructed software, so that in a face to face situations, a short-lived meeting key can be created, used, and thrown away.

Scalability is always an issue with routing protocols. One must ask how much beaconing a link can support and how much bandwidth is lost? Fundamentally it is hard to see how mobile systems can avoid beaconing. A system lacks any way to discern whether a target is local or remote since subnet associations mean nothing. It can only decide to forward a packet to an agent or send it to a host route that has appeared in the routing table due either to link proximity (our protocol), by some higher ad hoc routing protocol, or by human intervention.

The general question is this: if we are sending packets to agents (assume the current agent is bound to the default route), is there some way a system can discern who is local and who is not?

Of course, one might look at the address and a priori know, but that is exactly what is wrong with the subnetting assumption. How do you look at an address and know that someone is local?

In an Internet context, it does not seem reasonable that a system should try local contact before remote contact. Most likely destinations are going to be remote (off-link, think about web browsing). If there was no such thing as a remote system, such an assumption may be reasonable.

If we assume we will use beaconing, we might ask if we can reduce its cost? As an obvious measure, from a policy point of view, one might simply decide to not talk to other mobile nodes and only talk to agents. If this choice is made, our protocol could be collapsed into Mobile-IP by having agents use it and by adding an authenticated MAC address in the Mobile-IP UDP registration message. Agents would put their MAC address into an authenticated ICMP packet (or UDP reply). Mobile nodes could place their MAC address in the UDP registration. In effect, our current implementation behaves along those lines as we push the authenticated beacon out before the UDP registration (or de-registration) message sent by mobile nodes.

As a very simple improvement, our mobile nodes beacon far more slowly than agents. We also have an agent field in the authentication extension so that agents can quickly ignore MN beacons that are irrelevant to them. Mobile nodes aren't currently that mobile. They might move a few times a day on the busiest day and are more likely to simply migrate from office to home a couple of times a day. As a result, slow beaconing is satisfactory.

Other mechanisms to increase scalability could be tried as well. At some future date, it is possible that 100 laptop equipped students might attend the same lecture. It should not be difficult for mobile nodes to discern that they have a lot of peers, and hence scale their own beacons back (presumably exponentially).

The most important consideration here is that mobile nodes really need to maintain a connection to an agent. Caceres [CP96] has already suggested that agent beacons need to be faster for quick mobile node handoff. Agent beacons are more important than mobile node beacons and represent an umbilical cord to the wired infrastructure. Possibly if mobile nodes can't hear agents and can only hear mobile nodes, they might (carefully) ask other mobile nodes which way it is to the agent as well as scale back their own beacons to make sure there is breathing room. A higher level ad hoc routing protocol or a multicast search mechanism could be employed for that purpose.

References

[Atk95] R. Atkinson. Security Architecture for the Internet Protocol. Request for Comments, RFC 1825, Internet Engineering Task Force, Aug 1995. Naval Research Labs.

[Bel89] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2), Mar 1989.

[Com91] D. Comer. *Internetworking with TCP/IP*, volume I, pages 150–151. Prentice-Hall, New York, 1991.

[CP96] R. Caceres and V.N. Padmanabhan. Fast and Scalable Handoffs for Wireless Internetworks. In *Proceedings of ACM, MobiCom*, 1996.

[Dee91] S. Deering. ICMP Router Discovery Messages. Request for Comments, RFC 1256, Internet Engineering Task Force, Sep 1991. Xerox Parc.

[HMR94] H. Harney, C. Muckenhirn, and T. Rivers. Group Key Management (GKMP) Architecture. Internet Draft, Internet Engineering Task Force, Sep 1994. Work in progress.

[Hui95] C. Huitema. *Routing in the Internet*, chapter 5. Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[JM96] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, 1996.

[PB94] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the SIGCOMM '94 Conference on Communications Architectures*, pages 234–244, Aug 1994.

[Per92] R. Perlman. *Interconnections: Bridges and Routers*, pages 195–198. Addison-Wesley, Reading, Massachusetts, 1992.

[Per96] C. Perkins. IP Mobility Support. Request for Comments, RFC 2002, Internet Engineering Task Force, October 1996. IBM.

[Plu82] D.C. Plummer. An Ethernet Address Resolution Protocol. Request for Comments, RFC 826, Internet Engineering Task Force, 1982. Symbolics Inc.

[Ste95] W.R. Stevens. *TCP/IP Illustrated*, volume II, pages 560–561. Addison-Wesley, New York, 1995.