Proceedings of the Workshop on Mobile Computing, I-SPAN, June 1999. © IEEE.

Supporting Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks

Josh Broch David A. Maltz David B. Johnson

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

`http://www.monarch.cs.cmu.edu/`

Abstract

Much progress has been made toward solving the problem of routing packets inside an ad hoc network, but there are presently no complete proposals for connecting ad hoc networks together to form larger networks, or for integrating them with wired internets. This paper describes a technique that allows a single ad hoc network to span across heterogeneous link layers. Using this technique, we can both integrate ad hoc networks into the hierarchical Internet and support the migration of mobile nodes from the Internet into and out of ad hoc networks via Mobile IP. Taken together, these solutions improve the scalability of flat ad hoc networks by introducing hierarchy, and they enable all nodes participating in the ad hoc network to be reachable from anywhere in the world. We have implemented each of the solutions in a real testbed of 8 nodes using the Dynamic Source Routing (DSR) protocol. Generalizing our solutions, we describe several abstract scenarios and present our ideas for solving them.

1 Introduction

In areas in which there is little or no communication infrastructure, or the existing infrastructure is expensive or inconvenient to use, wireless mobile users may still be able to communicate through the formation of an *ad hoc network*. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. Each node participates in an ad hoc routing protocol that allows it to discover "multi-hop" paths through the network to any other node. The idea of ad hoc networking is sometimes also called *infrastructureless networking* [10], since the mobile nodes in the network dynamically establish routing among themselves to form their own network "on the fly." Some examples of the possible uses of ad hoc networking include students using laptop computers to participate in an interactive lecture, business associates sharing information during a meeting, soldiers relaying information for situational awareness on the battlefield [7, 12], and emergency disaster relief personnel coordinating efforts after a hurricane or earthquake.

In order to deploy ad hoc networks in scenarios similar to those just described, ad hoc network routing protocols will be required to support different types of network interfaces. For example, two teams of disaster relief personnel from different organizations may have different types of network interfaces, but they will still need to communicate effectively and efficiently. Although there are numerous proposals for ad hoc network routing protocols, none of the existing protocols fully address the issues of supporting heterogeneous network interfaces, using heterogeneous interfaces to achieve scalability, and interconnecting with the Internet. In this paper, we describe the initial design of an addressing architecture that solves these problems. We have implemented the architecture in a real ad hoc network testbed [9] using the Dynamic Source Routing protocol (DSR) [3, 5, 1] and Mobile IP [11, 4].

Section 2 of this paper provides an overview of the basic Dynamic Source Routing protocol (DSR). Section 3 details our addressing architecture, while Sections 4, 5, and 6 explain how the addressing architecture can be used to support heterogeneous interfaces, connect an ad hoc network to the Internet, and provide Mobile IP support within an ad hoc network, respectively. Section 7 explains three general problems and our current approach to solving them.

2 Dynamic Source Routing

The Dynamic Source Routing protocol (DSR) [3, 5, 1] works by discovering and using *source routes*. That is, the originator of a packet first learns the complete, ordered sequence of network hops necessary to reach the destination, and each packet sent carries this list of hops in its header. The key advantage of a source routing design is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets that they forward, since the packets themselves already contain all of the routing decisions. This fact, coupled with the *on-demand* nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols [2].

The DSR protocol is composed of two mechanisms: *Route Discovery* and *Route Maintenance*. Route Discovery is the mechanism by which a node **S** wishing to send a packet to a destination **D** obtains a source route to **D**. To perform a Route Discovery, the source node **S** broadcasts a ROUTE REQUEST packet that is flooded through the network in a controlled manner and is answered by a ROUTE REPLY packet from either the destination node or another node that knows a route to the destination. To reduce the cost of Route Discovery, each node maintains a cache of source routes it has learned or overheard, which it aggressively uses to limit the frequency and propagation of ROUTE REQUESTs.

When sending or forwarding a packet to some destination **D**, Route Maintenance is used to detect if the network topology has changed such that the route used by this packet has broken. When a route breaks, the detecting node returns a ROUTE ERROR packet to the original sender **S** of the packet. The sender **S** can then attempt to use any other route to **D** that is already in its route cache, or can invoke Route Discovery again to find a new route.

This work was supported in part by the National Science Foundation (NSF) under CAREER Award NCR-9502725, by the Air Force Materiel Command (AFMC) under DARPA contract number F19628-96-C-0061, and by the AT&T Foundation under a Special Purpose Grant in Science and Engineering. David Maltz was also supported under an Intel Graduate Fellowship. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of NSF, AFMC, DARPA, the AT&T Foundation, Intel, Carnegie Mellon University, or the U.S. Government.

3 Addressing Architecture

Among the most basic properties of a network is the manner in which the nodes of the network are assigned the addresses by which other nodes will communicate with them. For this discussion, we define a *node* in the ad hoc network to be an entity capable of moving independently from the other nodes in the network. A group of computers that always move together, such as a wired network of components on a vehicle, can be handled by recursively applying the techniques described in this paper.

In the most general case, each node in an ad hoc network will be acting as an independent router. This implies that the addressing scheme inside an ad hoc network should ideally be *flat*, meaning that each address serves only as an identifier and does not convey any information about where one node is topologically located with respect to any other node. For any type of hierarchical addressing scheme inside a single ad hoc network to make sense, nodes would have to be constrained to move together with the other nodes in their branch of the hierarchy, or the hierarchy of addresses would have to be continually updated as nodes move. Such movement constraints would violate the spirit of an ad hoc network as a collection of equal peers opportunistically using each others' services to communicate, and the continual reassignment of addresses could become a very expensive proposition, depending upon the rate of node movement.

Although a single node may have many different physical network interfaces, which in a typical IP network would each have a different IP address, we would like each node in the ad hoc network to have a single identifier by which it is known to all other nodes in the network. This allows each node to be recognized by all other nodes in the ad hoc network as a single entity regardless of which interface they use to communicate with it. We therefore require that each node participating in the ad hoc network select a single IP address from the ones assigned to it and that it use only that address when participating in the DSR protocol. In keeping with the terminology used by Mobile IP, we refer to this address as a node's *home address*.

The selection of a single address is important because if a node were to use multiple addresses when participating in the DSR protocol, two source routes which pass through the same nodes in the same order could contain different sequences of IP addresses. This reduces the ability of Route Discovery to reuse paths to destinations that other nodes may have in their route caches, and greatly increases the work required of Route Maintenance to purge invalid routes from the caches of nodes in the network.

Since each node is known to other nodes by a single IP address, some other notation is required to distinguish between the multiple network interfaces a node might carry. Under our addressing architecture, each node locally assigns a unique *interface index* to each of its network interfaces. In most operating systems, this is already done; for example, the `if_index` field in the `ifnet` structure of BSD Unix-based networking stacks [13] serves this purpose. With the exception of several reserved indices, these index values are local to each node, and the index values chosen by a node have no meaning outside of that node except to represent a unique network interface. This eliminates the need to globally agree on a mapping between interface indices and interface types and allows nodes to encode extra information that is locally significant into the index value.

We define a path through the ad hoc network from a source node $\mathbf{N}_0$ to a destination node $\mathbf{N}_m$ as a source route consisting of a series of hops $\mathbf{N}_0/i_0 \rightarrow \mathbf{N}_1/i_1 \rightarrow \mathbf{N}_2/i_2 \rightarrow \ldots \rightarrow \mathbf{N}_m$. We use $\mathbf{N}_k/i_k$ to indicate that node $\mathbf{N}_k$ must transmit the packet out its interface i_k in order to deliver the packet over the next hop to node $\mathbf{N}_{k+1}$.

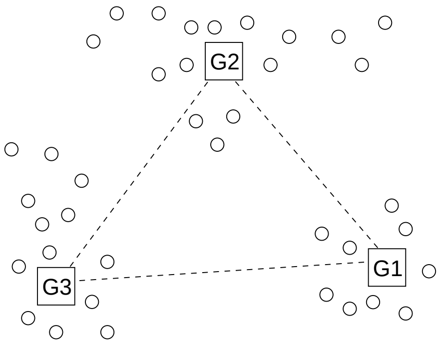

Figure 1 Clouds of nodes communicating via short-range radios and gateway nodes with both short-range and long-range radios. Each cloud may be multiple network hops in diameter.

The nodes in an ad hoc network can have their home addresses assigned using many different mechanisms, subject to the basic requirement that the addresses be unique inside the ad hoc network. If the ad hoc network is guaranteed to never connect to any other internet, then the addresses are only opaque identifiers and can be drawn from any unique numbering space. For example, a node could select the lowest MAC address from its network interfaces cards as an address.

In contrast, when groups of nodes are expected to work together as an ad hoc network and internetwork with other nodes via an IP internet, their home addresses can be assigned from a single IP subnet just as would be done for wired hosts. This does not imply that any hierarchy exists within a single ad hoc network, but rather that a single ad hoc network is a subnet within the hierarchy of some IP internet. We explain how a node can migrate from one ad hoc network to another in Section 6.

Assigning the home addresses from the same legal IP subnet provides several benefits. First, it facilitates connectivity with the Internet (Section 5), since the border routers that connect the ad hoc network to the rest of the Internet can distinguish between IP addresses which are homed inside the ad hoc network and external addresses. Second, the border routers can advertise reachability to the ad hoc subnet on the Internet using the standard Internet routing protocols since each of the nodes in the subnet has a legal routable IP address. Third, as discussed in Section 7, it can be used to artificially limit the size of a single ad hoc network, and thereby increase scalability by breaking a large ad hoc network into several smaller ad hoc networks.

4 Handling Heterogeneous Interfaces

One common architecture for ad hoc networks is depicted in Figure 1 where *clouds* of nodes with one type of wireless network interface are gathered together with *gateway* nodes with two or more types of network interfaces.

Such an architecture is an example of an *overlay network* [6], where the dashed lines between square boxes represent a long-range radio used to connect the clouds of nodes, which in turn use short-range high-speed radios to communicate among themselves. For example, in a military setting a company of soldiers might use short-range radios to communicate among themselves while relaying through a truck-mounted satellite system to communicate with other companies. In an office setting, each room might have a basestation interconnected by a wired network while mobile nodes using short-range infrared transceivers form a multi-hop cloud of nodes in each room.

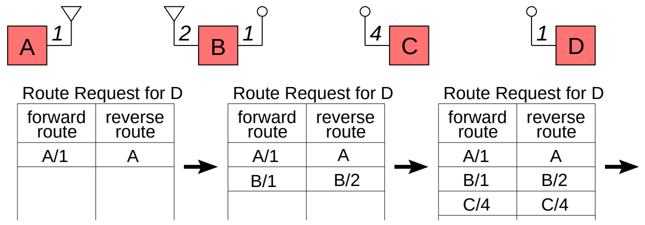

Figure 2 Route Discovery in a ad hoc network with heterogeneous network interfaces.

In the most general case, nothing can be assumed about the home address each node uses or the arrangement of nodes into clouds. Since *each cloud may be several transmission hops across*, all the nodes in the network must participate in the ad hoc network routing protocol in order to communicate even with the other nodes and gateways in their own cloud. Even though there are different network interfaces in use and a hierarchy of clouds is apparent in Figure 1, the routing protocol must treat the entire network as a single flat routing domain since it does not know *a priori* which cloud a given address can be found in.

The addressing architecture detailed in Section 3 gives DSR the ability to treat the overall network as single routing domain since the use of interface indices allows a source route, and thus a Route Discovery, to traverse interface types.

Figure 2 shows an example of an ad hoc network with heterogeneous network interfaces. Node **A** is using one type of network interface (represented by the triangles), node **C** and node **D** are using an entirely different type of physical network interface (represented by the circles), and gateway node **B** is a multi-homed ad hoc network node that can route between the two different types of radio technologies. As described in the previous section, each node independently chooses an interface index for its interfaces, so that while **B** and **D** have both chosen index 1 for their circle interfaces, **C** has chosen index 4.

The example in Figure 2 shows how a ROUTE REQUEST for **D** originated by **A** will propagate across the network. As the REQUEST propagates it will collect both a *forward route* from **A** to **D** and a *reverse route* from **D** to **A** [1]. When **A**'s ROUTE REQUEST is received by **B**, **B** checks if it is already listed on the source route recorded in the packet or has already repropagated a copy of this REQUEST. If neither is true, **B** adds itself to the listed route and repropagates the REQUEST out all its interfaces, including the one it was received on. When **B** transmits the packet out interface i, it lists itself in the forward route as **B**/i. **C** receives the request and repeats this process, so that when the packet is received at **D** it contains both a route from **A** to **D** and a route from **D** to **A**. **D** returns the discovered route, **A**/1$\rightarrow$ **B**/1$\rightarrow$ **C**/4$\rightarrow$ **D**, to **A** in a ROUTE REPLY packet. **D** may return the REPLY to **A** using a cached route, using the reverse route accumulated in the REQUEST, or by doing Route Discovery and piggybacking the REPLY on its request for **A**.

The packet headers in Figure 3 show how the source route would be used to route a packet from **A** to **D**, with the outlined boxes indicating which hop in the source route is being processed. This example demonstrates the need for a source route to include both the home address *and* interface index of each hop. Otherwise, node **B** would not have the information necessary to determine which of its interfaces should be used when forwarding the packet. Once this

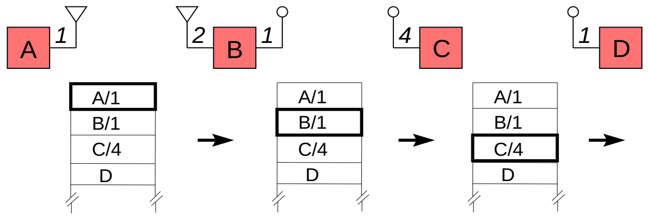

Figure 3 The source route on a packet as it moves through an ad hoc network changing physical interface types from triangle interfaces to circle interfaces. The outlined boxes indicate which entry in the source route is used when transmitting the packet at each stage.

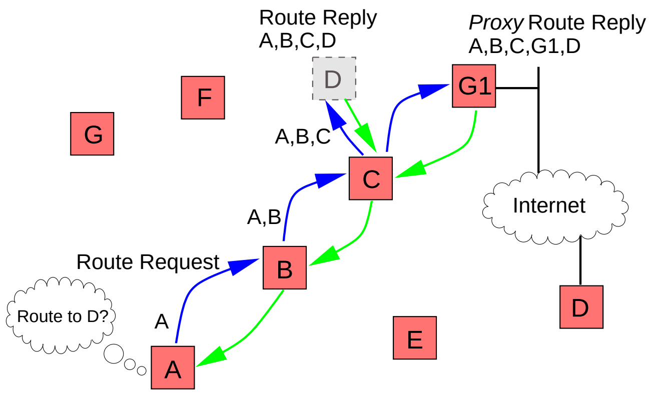

Figure 4 A ROUTE REQUEST for node **D** being answered by **D** and by the gateway node **G1**.

information is present in each packet, packets can be routed seamlessly across heterogeneous network interfaces without any further additions to the system.

Although the examples here have used only one gateway per cloud and only two types of interfaces, there is no limit to the number of gateways in a single cloud, nor to the number of interface types. Since each ROUTE REQUEST packet builds up a source route of the path it has travelled across the network, and since each gateway inserts its unique home address into each ROUTE REQUEST it propagates, DSR Route Discovery across heterogeneous interfaces is guaranteed to be trivially loop-free just as it is across a network with homogeneous interfaces. Additionally, the same optimization that causes each node in a homogeneous network to only repropagate a ROUTE REQUEST once also works in heterogeneous networks, causing ROUTE REQUESTS to flood fill the network in an orderly fashion.

5 Integration with Internet Routing

Another issue that the addressing architecture described in Section 3 solves is the problem of connecting an ad hoc network to the Internet. Since routing within the ad hoc network is flat, and routing within the Internet is hierarchical, it is necessary to provide the illusion to the outside world that the ad hoc network is simply a normal IP subnet. Local delivery within the ad hoc "subnet" is accomplished using the DSR protocol (possibly over many hops) while standard IP routing mechanisms decide which packets should enter and leave the subnet.

Figure 4 depicts how an ad hoc network can be connected to the Internet. Node **G1** is a gateway (border router) between the ad hoc network and the Internet. Routing on **G1**'s interface internal to the ad hoc network is accomplished using DSR, while its interface connected to the Internet is configured to use normal IP routing mechanisms.

[1] Although each node's address is shown twice in each packet in Figure 2, in the actual packet format used, each address appears only once, together with the interface index for the forward route and the reverse route at each node.

In order for a node **A** within the ad hoc network to communicate with a node **D** outside of the ad hoc network, **A** simply initiates Route Discovery (Section 2) for **D**. As the ROUTE REQUEST from **A** targeting **D** propagates, it is eventually received by the gateway node **G1**, which consults its routing table. If **G1** believes **D** is reachable outside the ad hoc network, it sends a *proxy reply* listing itself as the second-to-last node in the route and **D** as the last node in the route. When generating a proxy reply, the reserved *gateway interface index* (253) is used to distinguish this reply from normal ROUTE REPLYs.

When node **A** subsequently originates a data packet for node **D**, the source route on the packet will be **A**/1→ **B**/1→ **C**/1→ **G1**/253→ **D**. When node **G1** receives the packet for **D** it will notice the reserved gateway interface index in the source routing header, remove the source routing header from the packet, and transmit the packet out its interface to the Internet. This packet will have an IP source address of **A** and an IP destination address of **D** and is identical to a packet that **A** would send to node **D** if it were attached to a normal IP subnet instead of a DSR ad hoc network.

If the target node **D** is actually inside the ad hoc network (Figure 4) then node **A** will receive a ROUTE REPLY from both **G1** and **D**. Since the REPLY from **D** will not contain a gateway interface index, **A** can prefer the direct route when sending packets to **D**.

With the mechanism described above, nodes inside the ad hoc network can discover routes that allow them to send packets to nodes outside the network. Allowing packets from the Internet to be routed into the ad hoc network merely requires that the gateway (node **G1**) be configured as a standard IP router for the ad hoc network subnet.

For example, referring to Figure 4, if node **D**, located somewhere in the Internet, were to transmit a packet destined for node **A**, normal IP routing techniques would be applied to get the packet from **D** to **G1**. After examining the packet, **G1** would determine that the packet is destined for a node in its subnet and would attempt to route the packet to **A** using DSR. If **G1** does not have a cached source route for node **A**, it performs a Route Discovery. Supposing that it discovers the source route **G1**/1→ **C**/1→ **B**/1→ **A**, it would then insert the source route **D**/253→ **G1**/1→ **C**/1→ **B**/1→ **A** into **D**'s IP packet and transmit the packet into the ad hoc network.

The technique described in this section to connect a single ad hoc network to the Internet can also be applied to increase the *containment* of Route Discovery [8] in a network of heterogeneous interfaces, even if the network is not connected to any Internet infrastructure. Containment is defined as the fraction of nodes in the ad hoc network that do not overhear a particular ROUTE REQUEST, and this metric correlates directly with scalability.

Figure 5 shows three different ad hoc clouds, a shaded cloud, a white cloud, and a striped cloud, each connected to the other clouds using long-range radios. Suppose node **A** in the shaded cloud is performing Route Discovery for node **B** in the white cloud. Using the technique described in Section 4, this Route Discovery would propagate throughout the entire ad hoc network, bothering nodes in all three clouds. However, if the home addresses are assigned such that each cloud is a distinct IP subnet, the multi-homed gateways (**G1**, **G2**, and **G3**) can be configured not to forward ROUTE REQUEST packets into their cloud if the REQUEST targets an address not belonging to their subnet. In our example, the ROUTE REQUEST would be contained to the three gateways (**G1**, **G2**, and **G3**) and the white and shaded clouds; it would not needlessly be propagated into the striped cloud.

Furthermore, each gateway can *proxy reply* for nodes in their cloud. If **G2** proxy replies for node **B**, this decreases the latency of Route Discovery observed by **A**. When a packet from node **A** to node **B** arrives at **G2**, **G2** will take responsibility for delivering the packet to **B**, performing Route Discovery as necessary. This is extremely advantageous because topological change in the white cloud is then completely hidden from node **A**, meaning that **A** will not need to perform Route Discovery simply because **B** is moving around inside of its cloud.

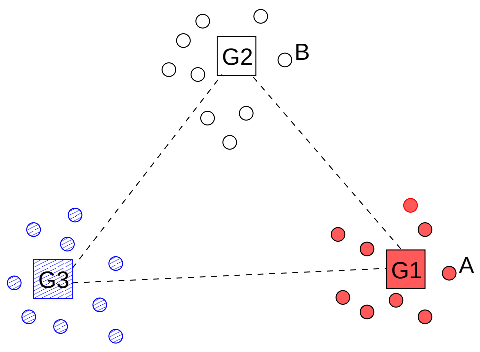

Figure 5 Hierarchical routing in the absence of wired infrastructure.

6 Integration with Mobile IP

The previous section detailed how the flat addressing scheme of an ad hoc network could be integrated with the hierarchical addressing used in the Internet to facilitate communication between nodes in the ad hoc network and nodes anywhere else in the Internet. Suppose, however, that in addition to an ad hoc network that is connected to the Internet, there is also a mobile node whose home network is *not* the ad hoc network. For some period of time, this mobile node roams through the area where the ad hoc network is located and during that time would like to join the ad hoc network and take advantage of the ad hoc network to access other resources on the Internet.

One specific example of this scenario could be a construction site where each vehicle on the site participates in an ad hoc network. A technician might occasionally travel to the site to service the vehicles. While doing so, this technician would like to join the ad hoc network so that he can use it to access manual pages or other resources at his home office which is connected to the Internet.

The primary mechanism that we use to support visiting mobile nodes is Mobile IP. Suppose that node **MN** in Figure 6 is a mobile node not homed within the ad hoc network and that node **FA** is a gateway between the ad hoc network and the Internet that provides Mobile IP foreign agent services.

The mobile node (**MN**) will typically keep its network interface in promiscuous receive mode and so will know that it has entered a DSR network when it overhears DSR packets like ROUTE REQUESTs, ROUTE REPLYs or data packets with DSR source routes on them. After node **MN** decides to participate in the ad hoc network, it will transmit a Mobile IP AGENT SOLICITATION piggybacked on a ROUTE REQUEST targeting the IP limited broadcast address (255.255.255.255). This allows the SOLICITATION to propagate over multiple hops through the ad hoc network, though gateways will not propagate it between subnets. When **FA** receives the SOLICITATION, it will reply with an AGENT ADVERTISEMENT, allowing **MN** to register itself with this foreign agent and with its home agent as a Mobile IP mobile node visiting the ad hoc network. Once the registration is complete, the mobile node's home agent will use Mobile IP to tunnel packets destined for mobile node **MN** to foreign agent **FA** and **FA** will deliver the packets locally to the mobile node using DSR.

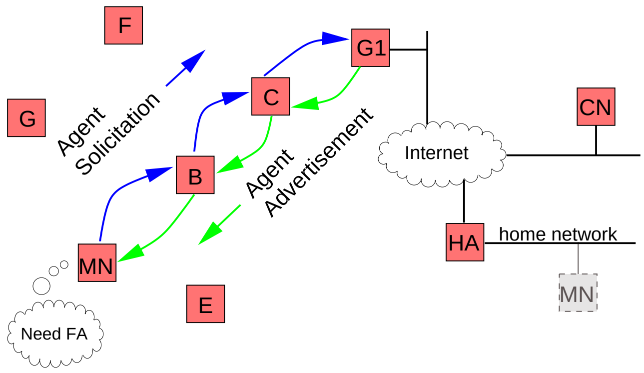

Figure 6 A visiting mobile node registering with a foreign agent (**FA**) in the ad hoc network.

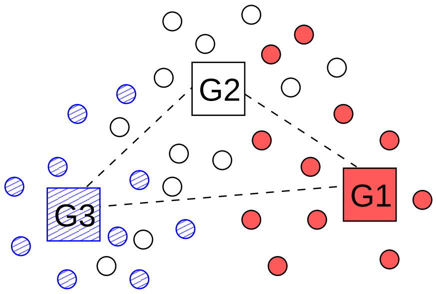

Figure 7 An ad hoc network where the clouds of nodes overlap and are in wireless transmission range of each other.

7 General Problems

The techniques described thus far successfully enable (1) the use of heterogeneous interfaces, (2) the integration of an ad hoc network into the Internet as a subnet, and (3) the movement of mobile nodes into and out of an ad hoc network using Mobile IP. This functionality has been completely implemented and tested in our physical ad hoc network testbed, which has been in operation since December 1998 [9].

These techniques improve the scalability of an ad hoc network in situations where nodes in different ad hoc clouds can only communicate via the gateways. This enables the gateways to contain Route Discoveries because they have enough information about the hierarchy of subnets to proxy reply for their cloud and to determine which ROUTE REQUESTs can safely be excluded from their cloud. This section explores the more general cases in which the ad hoc clouds can directly interact, and presents our current ideas for solving the new problems that arise.

7.1 Overlapping Ad Hoc Clouds

Although nodes in an ad hoc network may often be arranged into clouds containing gateway nodes with multiple interfaces, and these clouds may have been formed with addresses drawn from the same subnet, it may frequently be the case that these clouds overlap spatially. As shown in Figure 7, some nodes from the shaded, white and striped clouds are in range of each other via their short-range radios. In this environment, if a shaded node transmits a ROUTE REQUEST for a white node, the request will directly flood the entire network via the short-range radios. The fact that the multi-homed square nodes have been configured to proxy reply on behalf of their subnet clouds will not allow them to contain the Route Discovery as in Section 5.

The spread of a Route Discovery across the entire network is a concern because the number of overhead packets required by the routing protocol typically increases with the number of nodes in the routing domain. We are just beginning to study the scaling properties of DSR with respect to the number of nodes in the routing domain, though initial simulations show that DSR performs well with 25, 50, and 100 nodes. While we have yet to simulate larger networks, we believe the maximum practical size of a routing domain that DSR can efficiently handle, given the optimizations we have experimented with so far, will be on the order of 500 nodes.

In order to contain Route Discovery, we need a mechanism to restrict a ROUTE REQUEST packet originated by a node in one cloud from being propagated by nodes homed in a different cloud. This will keep the nodes logically separate even though they are physically co-located. We assume that each cloud is a separate IP subnet, and that each node is configured with the netmask of the subnet it is a part of. This allows each node to determine whether or not another node's address is inside its subnet, and hence its cloud. Gateway nodes must be configured with a netmask that identifies not only their own subnet, but also the subnets of their peers, as belonging to their same cloud.

We require that nodes only repropagate a ROUTE REQUEST packet if the REQUEST was last propagated by a member of the same cloud (i.e., the last address in the source route carried by the REQUEST is from the same subnet). This rule results in a REQUEST originated by a shaded node only being repropagated by other shaded nodes, and thereby prevents the Route Discovery from spreading directly between clouds. This filtering rule applies only to forwarding ROUTE REQUESTS and not to forwarding packets, so that if a source route is somehow discovered that crosses directly between clouds, packets may flow along it.

We must provide an exception to the filtering rule, however, to handle cases in which a node legitimately intends to have its ROUTE REQUEST propagated by nodes outside its subnet. A node invokes the exception by setting the "I" bit in the ROUTE REQUEST. A node receiving a packet with the "I" bit set will ignore the filtering rule, add itself to the recorded source route, and clear the "I" bit before repropagating the request. The "I" bit is cleared so that future propagations will obey the filtering rule.

The "I" bit is used to *introduce* the REQUEST to a new cloud, as it permits nodes from the new cloud to repropagate the request once. Because the source route then ends with an address from the new cloud, other nodes in the new cloud will repropagate it. For example, when a gateway needs to forward a ROUTE REQUEST into a cloud to which its home address does not belong, it sets the "I" bit in order to introduce the REQUEST to the nodes in the cloud.

7.2 Wandering Nodes

A different scenario is depicted in Figure 8. In this figure, the white ad hoc network cloud and the shaded cloud do not overlap spatially, but one node from the shaded cloud has wandered into the white cloud, becoming partitioned from the rest of the nodes in its home cloud. This problem is exactly the problem described in Section 6 and is solved using Mobile IP; gateway **G2** acts as a foreign agent and gateway **G1** acts as a home agent for the shaded node that is visiting the white cloud. This allows the shaded node to continue communication just as if it were still connected to its home cloud.

Because the shaded node is completely surrounded by white nodes, it must set the "I" bit on ROUTE REQUEST packets that it originates which contain Mobile IP AGENT SOLICITATIONs for a foreign agent. This results in the SOLICITATIONs spreading through the clouds neighboring the shaded node and finding a nearby foreign agent.

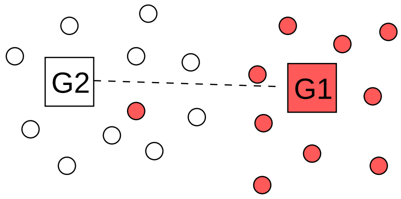

Figure 8 A node from the shaded cloud that is completely inside the white cloud and out of range of any other shaded node.

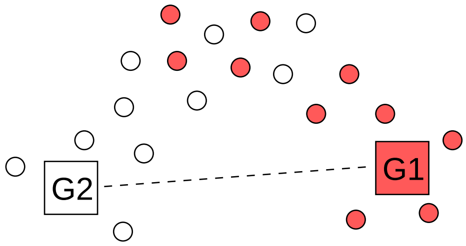

Figure 9 Nodes from the white and shaded clouds cooperating on a joint task away from the square nodes that relay between clouds.

7.3 Cooperating Ad Hoc Clouds

The filtering rule described in Section 7.1 prevents nodes from re-propagating Route Discoveries initiated by nodes in other clouds, which forces the interaction between clouds to occur at the gateways. This is important as it increases the scalability of the routing protocol. However, the filtering rule does not prevent a node in one cloud from *answering* a ROUTE REQUEST it receives from a node in another cloud. This enables nodes from different clouds who wish to communicate to potentially use the most optimal route available between them, rather than forcing all traffic between them to traverse the gateways.

Figure 9 illustrates a scenario in which several nodes from two different clouds are cooperating on a joint task in a work area far from their gateways. The most optimal route for communication between the nodes is clearly via the short-range radios, and not via the gateways. Since there is significant overlap in the clouds, ROUTE REQUEST packets transmitted by nodes in the white cloud will be overheard by nodes in the shaded cloud. This allows nodes in the white cloud to effectively query the caches of nodes in the shaded cloud in order to find a route to destinations in shaded cloud. Previous work [8] has shown that routes to each node in a single ad hoc network are well distributed among the caches of nodes in the network. Therefore, it is very likely that a ROUTE REQUEST performed by a node in one cloud will be overheard by either the target itself, or by another node which already has a cached route to the target. This will result in a ROUTE REPLY being sent to the requester containing a direct route across the short-range radios.

8 Conclusions

In this paper we have presented a solution for supporting heterogeneous network interfaces in a multi-hop wireless ad hoc network. Extending this technique, we have shown how to connect an ad hoc network to the Internet, and how to use Mobile IP to support nodes visiting the ad hoc network. We have implemented and validated these ideas using a real ad hoc network testbed, which has been in regular use for approximately 5 months [9]. In addition, we discussed how our techniques could be applied to even more general scenarios.

There is much more work to be done in the area of effectively and efficiently using hierarchy within an ad hoc network. For example, complications arise when two ad hoc networks that have been assigned addresses by completely different administrative domains attempt to communicate. The issue of how to provide support to nodes whose home network has been completely destroyed or is unreachable for an extended period of time is also a very important concern. We are currently working to resolve these issues and then simulate and implement the solutions.

References

[1] Josh Broch, David B. Johnson, and David A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft-ietf-manet-dsr-01.txt, December 1998. Work in progress.

[2] Josh Broch, David A. Maltz, David B. Johnson, Yih-chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 85–97, Dallax, TX, October 1998.

[3] David B. Johnson. Routing in Ad Hoc Networks of Mobile Hosts. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pages 158–163, December 1994.

[4] David B. Johnson. Scalable Support for Transparent Mobile Host Internetworking. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 3, pages 103–128. Kluwer Academic Publishers, 1996.

[5] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[6] Randy H. Katz and Eric A. Brewer. The Case for Wireless Overlay Networks. In *Proceedings of the SPIE Multimedia and Networking Conference (MMNC'96)*, San Jose, CA, January 1996.

[7] Barry M. Leiner, Robert J. Ruth, and Ambatipudi R. Sastry. Goals and Challenges of the DARPA GloMo Program. *IEEE Personal Communications*, 3(6):34–43, December 1996.

[8] David A. Maltz, Josh Broch, Jorjeta Jetcheva, and David B. Johnson. The Effects of On-Demand Behavior in Routing Protocols for Ad Hoc Networks. *IEEE Journal on Selected Areas of Communications*, 1999. To appear.

[9] David A. Maltz, Josh Broch, and David B. Johnson. Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed. Technical Report 99-116, School of Computer Science, Carnegie Mellon University, March 1999.

[10] National Science Foundation. Research priorities in wireless and mobile communications and networking: Report of a workshop held March 24–26, 1997, Airlie House, Virginia. Available at http://www.cise.nsf.gov/anir/ww.html.

[11] Charles Perkins, editor. IP Mobility Support. RFC 2002, October 1996.

[12] Neil Siegel, Dave Hall, Clint Walker, and Rene Rubio. The Tactical Internet Graybeard Panel briefings. U.S. Army Digitization Office. Available at http://www.ado.army.mil/Briefings/Tact%20Internet/index.htm, October 1997.

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