Dynamic Source Routing in Ad Hoc Wireless Networks

David B. Johnson
David A. Maltz

Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891

dbj@cs.cmu.edu

Abstract

An *ad hoc* network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. This paper presents a protocol for routing in ad hoc networks that uses *dynamic source routing*. The protocol adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. Based on results from a packet-level simulation of mobile hosts operating in an ad hoc network, the protocol performs well over a variety of environmental conditions such as host density and movement rates. For all but the highest rates of host movement simulated, the overhead of the protocol is quite low, falling to just 1% of total data packets transmitted for moderate movement rates in a network of 24 mobile hosts. In all cases, the difference in length between the routes used and the optimal route lengths is negligible, and in most cases, route lengths are on average within a factor of 1.01 of optimal.

1. Introduction

Mobile hosts and wireless networking hardware are becoming widely available, and extensive work has been done recently in integrating these elements into traditional networks such as the Internet. Oftentimes, however, mobile users will want to communicate in situations in which no fixed wired infrastructure such as this is available, either because it may not be economically practical or physically possible to provide the necessary infrastructure or because the expediency of the situation does not permit its installation. For example, a class of students may need to interact during a lecture, friends or business associates may run into each other in an airport terminal and wish to share files, or a group of emergency rescue workers may need to be quickly deployed after an earthquake or flood. In such situations, a collection of mobile hosts with wireless network interfaces may form a temporary network without the aid of any established infrastructure or centralized administration. This type of wireless network is known as an *ad hoc network*.

A version of this paper will appear as a chapter in the book *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers, 1996.

If only two hosts, located closely together, are involved in the ad hoc network, no real routing protocol or routing decisions are necessary. In many ad hoc networks, though, two hosts that want to communicate may not be within wireless transmission range of each other, but could communicate if other hosts between them also participating in the ad hoc network are willing to forward packets for them. For example, in the network illustrated in Figure 1, mobile host *C* is not within the range of host *A*'s wireless transmitter (indicated by the circle around *A*) and host *A* is not within the range of host *C*'s wireless transmitter. If *A* and *C* wish to exchange packets, they may in this case enlist the services of host *B* to forward packets for them, since *B* is within the overlap between *A*'s range and *C*'s range. Indeed, the routing problem in a real ad hoc network may be more complicated than this example suggests, due to the inherent nonuniform propagation characteristics of wireless transmissions and due to the possibility that any or all of the hosts involved may move at any time.

Routing protocols in conventional wired networks generally use either *distance vector* or *link state* routing algorithms, both of which require periodic routing advertisements to be broadcast by each router. In distance vector routing [9, 17, 26, 27, 29], each router broadcasts to each of its neighbor routers its view of the distance to all hosts, and each router computes the shortest path to each host based on the information advertised by each of its neighbors. In link state routing [10, 16, 18], each router instead broadcasts to all other routers in the network its view of the status of each of its adjacent network links, and each router then computes the shortest distance to each host based on the complete picture of the network formed from the most recent link information from all routers. In addition to its use in wired networks, the basic distance vector algorithm has also been adapted for routing in wireless ad hoc networks, essentially treating each mobile host as a router [11, 19, 25].

This paper describes the design and performance of a routing protocol for ad hoc networks that instead uses *dynamic source routing* of packets between hosts that want to communicate. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which to forward the packet; the sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host. Source routing has been used in a number of contexts for routing in wired networks, using either statically defined or dynamically constructed source routes [4, 5, 12, 20, 22, 28], and has been used with statically configured routes in the Tucson Amateur Packet Radio (TAPR) work for routing in a wireless network [14]. The protocol presented here is explicitly designed for use in the wireless environment of an ad hoc network. There are no periodic router advertisements in the protocol. Instead, when a host needs a route to another host, it dynamically determines one based on cached information and on the results of a *route discovery* protocol.

We believe our dynamic source routing protocol offers a number of potential advantages over conventional routing protocols such as distance vector in an ad hoc network. First, unlike conventional routing protocols, our protocol uses no periodic routing advertisement messages, thereby reducing network bandwidth overhead, particularly during periods when little or no significant host movement is taking place. Battery power is also conserved on the mobile hosts, both by not sending the advertisements and by not needing to receive them (since a host could otherwise reduce its power usage by putting itself into "sleep" or "standby" mode when not busy with other tasks). Distance vector and link state routing, on the other hand, must continue to send advertisements even when nothing changes, so that other mobile hosts will continue to consider those routes or network links as valid. In addition, many of the "links" between routers seen by the routing algorithm may be redundant [11]. Wired networks are usually explicitly configured to have only one (or a small number) of routers connecting any two networks, but there are no explicit links in an ad hoc network, and all communication is by broadcast transmissions. The redundant paths in a wireless environment unnecessarily increase the size of routing updates that must be sent over the network, and increase the CPU overhead required to process each update and to compute new routes.

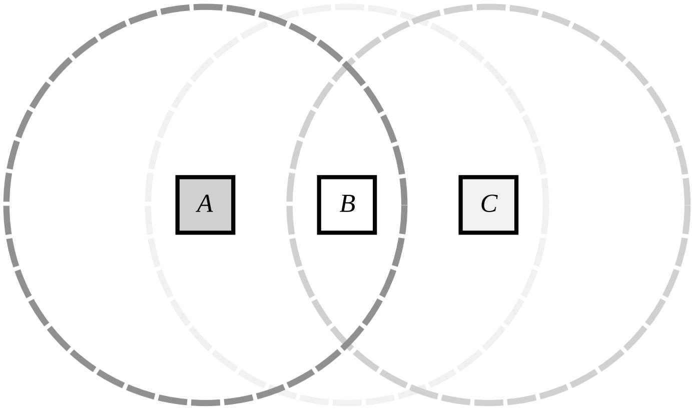

Figure 1 A simple ad hoc network of three wireless mobile hosts

In addition, conventional routing protocols based on link state or distance vector algorithms may compute some routes that do not work. In a wireless environment, network transmission between two hosts does not necessarily work equally well in both directions, due to differing propagation or interference patterns around the two hosts [1, 15]. For example, with distance vector routing, even though a host may receive a routing advertisement from another mobile host, packets it might then transmit back to that host for forwarding may not be able to reach it. Our protocol does not require transmissions between hosts to work bidirectionally, although we do make use of it when afforded, for example, by MAC-level protocols such as MACA [13] or MACAW [2] that ensure it.

Finally, conventional routing protocols are not designed for the type of dynamic topology changes that may be present in ad hoc networks. In conventional networks, links between routers occasionally go down or come up, and sometimes the cost of a link may change due to congestion, but routers do not generally move around dynamically. In an environment with mobile hosts as routers, though, convergence to new, stable routes after such dynamic changes in network topology may be slow, particularly with distance vector algorithms [20]. Our dynamic source routing protocol is able to adapt quickly to changes such as host movement, yet requires no routing protocol overhead during periods in which such changes do not occur.

Section 2 of this paper details our assumptions about the network and the mobile hosts. The basic operation of our dynamic source routing protocol is described in Section 3, and optimizations to this basic operation are described in Section 4. In Section 5, we present a preliminary evaluation of the performance of our protocol, based on a packet-level simulation. In Section 6, we discuss related protocols for wireless network routing and for source routing, and in Section 7, we present conclusions and future work.

2. Assumptions

We assume that all hosts wishing to communicate with other hosts within the ad hoc network are willing to participate fully in the protocols of the network. In particular, each host participating in the network should also be willing to forward packets for other hosts in the network.

We refer to the number of hops necessary for a packet to reach from any host located at one extreme edge of the network to another host located at the opposite extreme, as the *diameter* of the network. For example, the diameter of the ad hoc network depicted in Figure 1 is two. We assume that the diameter of an ad hoc network will be small but may often be greater than one.

Hosts within the ad hoc network may move at any time without notice, but we assume that the speed with which hosts move is moderate with respect to the packet transmission latency and wireless transmission range of the particular underlying network hardware in use. In particular, we assume that hosts do not continuously move so rapidly as to make the flooding of every packet the only possible routing protocol.

We assume that hosts can enable a *promiscuous* receive mode on their wireless network interface hardware, causing the hardware to deliver every received packet to the network driver software without filtering based on destination address. Although we do not require this facility, it is common in current LAN hardware for broadcast media including wireless, and some of our optimizations take advantage of it. Use of promiscuous mode does increase the software overhead on the CPU, but we believe that wireless network speeds are more the inherent limiting factor to performance in current and future systems. We believe that portions of the protocol are also suitable for implementation directly in hardware or within a programmable network interface unit to avoid this overhead on the CPU.

3. Basic Operation

3.1. Overview

To send a packet to another host, the sender constructs a *source route* in the packet's header, giving the address of each host in the network through which the packet should be forwarded in order to reach the destination host. The sender then transmits the packet over its wireless network interface to the first hop identified in the source route. When a host receives a packet, if this host is not the final destination of the packet, it simply transmits the packet to the next hop identified in the source route in the packet's header. Once the packet reaches its final destination, the packet is delivered to the network layer software on that host.

Each mobile host participating in the ad hoc network maintains a *route cache* in which it caches source routes that it has learned. When one host sends a packet to another host, the sender first checks its route cache for a source route to the destination. If a route is found, the sender uses this route to transmit the packet. If no route is found, the sender may attempt to discover one using the *route discovery* protocol. While waiting for the route discovery to complete, the host may continue normal processing and may send and receive packets with other hosts. The host may buffer the original packet in order to transmit it once the route is learned from route discovery, or it may discard the packet, relying on higher-layer protocol software to retransmit the packet if needed. Each entry in the route cache has associated with it an expiration period, after which the entry is deleted from the cache.

While a host is using any source route, it monitors the continued correct operation of that route. For example, if the sender, the destination, or any of the other hosts named as hops along a route move out of wireless transmission range of the next or previous hop along the route, the route can no longer be used to reach the destination. A route will also no longer work if any of the hosts along the route should fail or be powered off. This monitoring of the correct operation of a route in use we call *route maintenance*. When route maintenance detects a problem with a route in use, route discovery may be used again to discover a new, correct route to the destination.

This section describes the basic operation of route discovery and route maintenance. Optimizations to this basic operation of the protocol are then described in Section 4.

3.2. Route Discovery

Route discovery allows any host in the ad hoc network to dynamically discover a route to any other host in the ad hoc network, whether directly reachable within wireless transmission range or reachable through one or more intermediate network hops through other hosts. A host initiating a route discovery broadcasts a *route request* packet which may be received by those hosts within wireless transmission range of it. The route request packet identifies the host, referred to as the *target* of the route discovery, for which the route is requested. If the route discovery is successful the initiating host receives a *route reply* packet listing a sequence of network hops through which it may reach the target.

In addition to the address of the original initiator of the request and the target of the request, each route request packet contains a *route record*, in which is accumulated a record of the sequence of hops taken by the route request packet as it is propagated through the ad hoc network during this route discovery. Each route request packet also contains a unique *request id*, set by the initiator from a locally-maintained sequence number. In order to detect duplicate route requests received, each host in the ad hoc network maintains a list of the ⟨initiator address, request id⟩ pairs that it has recently received on any route request.

When any host receives a route request packet, it processes the request according to the following steps:

1. If the pair ⟨initiator address, request id⟩ for this route request is found in this host's list of recently seen requests, then discard the route request packet and do not process it further.

2. Otherwise, if this host's address is already listed in the route record in the request, then discard the route request packet and do not process it further.

3. Otherwise, if the target of the request matches this host's own address, then the route record in the packet contains the route by which the request reached this host from the initiator of the route request. Return a copy of this route in a *route reply* packet to the initiator.

4. Otherwise, append this host's own address to the route record in the route request packet, and re-broadcast the request.

The route request thus propagates through the ad hoc network until it reaches the target host, which then replies to the initiator. The original route request packet is received only by those hosts within wireless transmission range of the initiating host, and each of these hosts propagates the request if it is not the target and if the request does not appear to this host to be redundant. Discarding the request because the host's address is already listed in the route record guarantees that no single copy of the request can propagate around a loop. Also discarding the request when the host has recently seen one with the same ⟨initiator address, request id⟩ removes later copies of the request that arrive at this host by a different route.

In order to return the *route reply* packet to the initiator of the route discovery, the target host must have a route to the initiator. If the target has an entry for this destination in its route cache, then it may send the route reply packet using this route in the same way as is used in sending any other packet (Section 3.1). Otherwise, the target may reverse the route in the route record from the route request packet, and use this route to send the route reply packet. This, however, requires the wireless network communication between each of these pairs of hosts to work equally well in both directions, which may not be true in some environments or with some MAC-level protocols. An alternative approach, and the one we have currently adopted, is for this host to piggyback the route reply packet on a route request targeted at the initiator of the route discovery to which it is replying. This use of piggybacking is described in Section 4.2.

3.3. Route Maintenance

Conventional routing protocols integrate route discovery with route maintenance by continuously sending periodic routing updates. If the status of a link or router changes, the periodic updates will eventually reflect the changes to all other routers, presumably resulting in the computation of new routes. However, using route discovery, there are no periodic messages of any kind from any of the mobile hosts. Instead, while a route is in use, the route maintenance procedure monitors the operation of the route and informs the sender of any routing errors.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.