DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks

David B. Johnson *David A. Maltz* *Josh Broch*

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

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

Abstract

The *Dynamic Source Routing* protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. The protocol is composed of the two mechanisms of *Route Discovery* and *Route Maintenance*, which work together to allow nodes to discover and maintain *source routes* to arbitrary destinations in the ad hoc network. The use of source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use. All aspects of the protocol operate entirely *on-demand*, allowing the routing packet overhead of DSR to scale *automatically* to only that needed to react to changes in the routes currently in use. We have evaluated the operation of DSR through detailed simulation on a variety of movement and communication patterns, and through implementation and significant experimentation in a physical outdoor ad hoc networking testbed we have constructed in Pittsburgh, and have demonstrated the excellent performance of the protocol. In this chapter, we describe the design of DSR and provide a summary of some of our simulation and testbed implementation results for the protocol.

1 Introduction

The *Dynamic Source Routing* protocol (DSR) [Johnson 1994, Johnson 1996a, Broch 1999a] is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. Network nodes (computers) cooperate to forward packets for each other to allow communication over multiple "hops" between nodes not directly within wireless transmission range of one another. As nodes in the network move about or join or leave the network, and as wireless transmission conditions such as sources of interference change, all routing is automatically determined and maintained by the DSR routing protocol. Since the number or sequence of intermediate hops needed to reach any destination may change at any time, the resulting network topology may be quite rich and rapidly changing.

The DSR protocol allows nodes to dynamically discover a *source route* across multiple network hops to any destination in the ad hoc network. Each data packet sent then carries in its header the complete, ordered list of nodes through which the packet must pass, allowing packet routing to be trivially loop-free and avoiding the need for up-to-date routing information in the intermediate nodes through which the packet is forwarded. By including this source route in the header of each data packet, other nodes forwarding or overhearing any of these packets may also easily cache this routing information for future use.

This work is a part of the Monarch Project at Carnegie Mellon University [Johnson 1996b, Monarch], a long-term research project that is developing networking protocols and protocol interfaces to allow truly seamless wireless and mobile networking. The Monarch Project is named in reference to the migratory behavior of the monarch butterfly, and can also be considered as an acronym for "Mobile Networking Architectures." The scope of our research includes protocol design, implementation, performance evaluation, and usage-based validation, spanning areas ranging roughly from portions of the ISO Data Link layer (layer 2) through the Presentation layer (layer 6).

In designing DSR, we sought to create a routing protocol that had very low overhead yet was able to react quickly to changes in the network, providing highly reactive service to help ensure successful delivery of data packets in spite of node movement or other changes in network conditions. Based on our evaluations of DSR and other protocols to date, through detailed simulation and testbed implementation, we believe this goal has been well met [Johnson 1996a, Broch 1998, Maltz 1999a, Maltz 1999b]. In particular, in our detailed simulation comparison of routing protocols for ad hoc networks [Broch 1998], DSR outperformed the other protocols that we studied, and recent results by Johansson et al. [Johansson 1999] have shown generally similar results. The protocol specification for DSR has also been submitted to the Internet Engineering Task Force (IETF), the principal protocol standards development body for the Internet, and is currently one of the protocols under consideration in the IETF Mobile Ad Hoc Networks (MANET) Working Group for adoption as an Internet Standard for IP routing in ad hoc networks [MANET].

This chapter describes the design of the DSR protocol and provides a summary of some of our current simulation and testbed implementation results for DSR. Section 2 of this chapter discusses our assumptions in the design of DSR. In Section 3, we present the design of the DSR protocol and describe the resulting important properties of this design. In particular, we describe here the design of the two mechanisms that make up the operation of DSR, *Route Discovery* and *Route Maintenance*; we also discuss the use of DSR in supporting heterogeneous networks and interconnecting to the Internet, and describe the current support present in DSR for routing of multicast packets in ad hoc networks. Section 4 then summarizes some of our simulation results for DSR and describes a physical outdoor ad hoc network testbed we have built in Pittsburgh for experimenting with DSR. Finally, we discuss related work in Section 5 and present conclusions in Section 6.

2 Assumptions

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

We refer to the minimum number of hops necessary for a packet to reach from any node located at one extreme edge of the ad hoc network to another node located at the opposite extreme, as the *diameter* of the ad hoc network. We assume that the diameter of an ad hoc network will often be small (e.g., perhaps 5 or 10 hops), but may often be greater than 1.

Packets may be lost or corrupted in transmission on the wireless network. A node receiving a corrupted packet can detect the error and discard the packet.

Nodes within the ad hoc network may move at any time without notice, and may even move continuously, but we assume that the speed with which nodes move is moderate with respect to the packet transmission latency and wireless transmission range of the particular underlying network hardware in use. In particular, DSR can support very rapid rates of arbitrary node mobility, but we assume that nodes do not continuously move so rapidly as to make the flooding of every individual data packet the only possible routing protocol.

We assume that nodes may be able to enable *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 link-layer destination address. Although we do not require this facility, it is, for example, common in current LAN hardware for broadcast media including wireless, and some of our optimizations can take advantage of its availability. 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 also believe that portions of the protocol are suitable for implementation directly within a programmable network interface unit to avoid this overhead on the CPU [Johnson 1996a]. Use of promiscuous mode may also increase the power consumption of the network interface hardware, depending on the design of the receiver hardware, and in such cases, DSR can easily be used without the optimizations that depend on promiscuous receive mode, or can be programmed to only periodically switch the interface into promiscuous mode.

Wireless communication ability between any pair of nodes may at times not work equally well in both directions, due for example to differing antenna or propagation patterns or sources of interference around the two nodes [Bantz 1994, Lauer 1995]. That is, wireless communications between each pair of nodes will in many cases be able to operate *bi-directionally*, but at times the wireless link between two nodes may be only *uni-directional*, allowing one node to successfully send packets to the other while no communication is possible in the reverse direction. Although many routing protocols operate correctly only over bi-directional links, DSR can successfully discover and forward packets over paths that contain uni-directional links. Some MAC protocols, however, such as MACA [Karn 1990], MACAW [Bharghavan 1994], or IEEE 802.11 [IEEE 1997], limit unicast data packet transmission to bi-directional links, due to the required bi-directional exchange of RTS and CTS packets in these protocols and due to the link-level acknowledgement feature in IEEE 802.11; when used on top of MAC protocols such as these, DSR can take advantage of additional optimizations, such as the route reversal optimization described below.

Each node selects a *single* IP address by which it will be known in the ad hoc network. 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 require each node to select one of these and to use only that address when participating in the DSR protocol. This allows each node to be recognized by all other nodes in the ad hoc network as a single entity regardless of which network interface they use to communicate with it. In keeping with the terminology used by Mobile IP [Johnson 1995, Perkins 1996], we refer to the address by which each mobile node is known in the ad hoc network as the node's *home address*, as this address would typically be the address that the node uses while connected to its home network (rather than while away, being a member of the ad hoc network). Each node's home address may be assigned by any mechanism (e.g., static assignment or use of DHCP for dynamic assignment [Droms 1997]), although the method of such assignment is outside the scope of the DSR protocol.

3 DSR Protocol Description

3.1 Overview and Important Properties of the Protocol

The DSR protocol is composed of two mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network:

- *Route Discovery* is the mechanism by which a node **S** wishing to send a packet to a destination node **D** obtains a source route to **D**. Route Discovery is used only when **S** attempts to send a packet to **D** and does not already know a route to **D**.

- *Route Maintenance* is the mechanism by which node **S** is able to detect, while using a source route to **D**, if the network topology has changed such that it can no longer use its route to **D** because a link along the route no longer works. When Route Maintenance indicates a source route is broken, **S** can attempt to use any other route it happens to know to **D**, or can invoke Route Discovery again to find a new route. Route Maintenance is used only when **S** is actually sending packets to **D**.

Route Discovery and Route Maintenance each operate entirely *on demand*. In particular, unlike other protocols, DSR requires *no* periodic packets of *any kind* at *any level* within the network. For example, DSR does not use any periodic routing advertisement, link status sensing, or neighbor detection packets, and does not rely on these functions from any underlying protocols in the network. This entirely on-demand behavior and lack of periodic activity allows the number of overhead packets caused by DSR to scale all the way down to *zero*, when all nodes are approximately stationary with respect to each other and all routes needed for current communication have already been discovered. As nodes begin to move more or as communication patterns change, the routing packet overhead of DSR *automatically* scales to only that needed to track the routes currently in use.

In response to a single Route Discovery (as well as through routing information from other packets overheard), a node may learn and cache multiple routes to any destination. This allows the reaction to routing changes to be much more rapid, since a node with multiple routes to a destination can try another cached route if the one it has been using should fail. This caching of multiple routes also avoids the overhead of needing to perform a new Route Discovery each time a route in use breaks.

The operation of Route Discovery and Route Maintenance in DSR are designed to allow uni-directional links and asymmetric routes to be easily supported. In particular, as noted in Section 2, in wireless networks, it is possible that a link between two nodes may not work equally well in both directions, due to differing antenna or propagation patterns or sources of interference. DSR allows such uni-directional links to be used when necessary, improving overall performance and network connectivity in the system.

DSR also supports internetworking between different types of wireless networks, allowing a source route to be composed of hops over a combination of any types of networks available [Broch 1999b]. For example, some nodes in the ad hoc network may have only short-range radios, while other nodes have both short-range and long-range radios; the combination of these nodes together can be considered by DSR as a single ad hoc network. In addition, the routing of DSR has been integrated into standard Internet routing, where a "gateway" node connected to the Internet also participates in the ad hoc network routing protocols; and has been integrated into Mobile IP routing, where such a gateway node also serves the role of a Mobile IP foreign agent [Johnson 1995, Perkins 1996].

3.2 Basic DSR Route Discovery

When some node **S** originates a new packet destined to some other node **D**, it places in the header of the packet a *source route* giving the sequence of hops that the packet should follow on its way to **D**. Normally, **S** will obtain a suitable source route by searching its *Route Cache* of routes previously learned, but if no route is found in its cache, it will initiate the Route Discovery protocol to dynamically find a new route to **D**. In this case, we call **S** the *initiator* and **D** the *target* of the Route Discovery.

For example, Figure 1 illustrates an example Route Discovery, in which a node **A** is attempting to discover a route to node **E**. To initiate the Route Discovery, **A** transmits a ROUTE REQUEST message as a single local broadcast packet, which is received by (approximately) all nodes currently within wireless transmission range of **A**. Each ROUTE REQUEST message identifies the initiator and target of the Route Discovery, and also contains a unique *request id*, determined by the initiator of the REQUEST. Each ROUTE REQUEST also contains a record listing the address of each intermediate node through which this particular copy of the ROUTE REQUEST message has been forwarded. This route record is initialized to an empty list by the initiator of the Route Discovery.