International Journal of Advanced Research in Engineering and Applied Sciences

ISSN: 2278-6252

ANALYSIS OF DYNAMIC SOURCE ROUTING IN WIRELESS AD-HOC NETWORKS

Kuldeep Singh*

Mohd. Husain**

Abstract: 'Dynamic Source Routing' (DSR) is a productive routing protocol for wireless ad-hoc mesh networks. DSR is based on source routing protocol which maintain all the routing information at mobile nodes which continuously updated. This protocol is based on reactive approach which restricts the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in other on demand routing approaches.

In DSR, each source determines the route to be used in transmitting its packets to selected destinations. DSR activity can be divided into two phases Route Discovery and Route Maintenance. DSR uses Route discovery cycle for route finding on Demand by mobile host. It supports unidirectional links so asymmetric routes are supported. We have analyzed the operation of DSR using simulation on various movement and communication patterns.

*Ksoft Vision, Lucknow,Lucknow, UP, India

**Jahangirabad Institute of Technology, Barabanki, UP, India

1. INTRODUCTION OF DYNAMIC SOURCE ROUTING

'Dynamic Source Routing' (DSR) is a routing protocol for wireless mesh networks. It is similar to AODV in that it forms a route on- demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device.

Determining source routes requires accumulating the address of each device between the source and destination during route discovery. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to route packets.

To accomplish source routing, the routed packets contain the address of each device the packet will traverse. This may result in high overhead for long paths or large addresses, like IPv6. To avoid using source routing, DSR optionally defines a flow id option that allows packets to be forwarded on a hop-by-hop basis.

This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only two major phases, which are Route Discovery and Route Maintenance. Route Reply would only be generated if the message has reached the intended destination node (route record which is initially contained in Route Request would be inserted into the Route Reply).

To return the Route Reply, the destination node must have a route to the source node. If the route is in the Destination Node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the Route Request message header (this requires that all links are symmetric). In the event of fatal transmission, the Route Maintenance Phase is initiated whereby the Route Error packets are generated at a node. The erroneous hop will be removed from the node's route cache; all routes containing the hop are truncated at that point. Again, the Route Discovery Phase is initiated to determine the most viable route.

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. 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 Ad Hoc Networks Working Group for adoption as an Internet Standard for IP routing in ad hoc networks.

2. ASSUMPTIONS TAKEN

The DSR protocol is designed mainly for mobile ad hoc networks of up to about two hundred nodes and is designed to work well even with very high rates of mobility. 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 ad hoc network SHOULD also be willing to forward packets for other nodes in the network.

The diameter of an ad hoc network is 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. We assume that this diameter will often be small but it may often be greater than 1.

In Ad-hoc networks packets may be lost or corrupted in transmission. We assume that a node receiving a corrupted packet can detect the error, such as through a standard link-layer checksum or Cyclic Redundancy Check (CRC), and discard the packet.

Now 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.

A common feature of many network interfaces, including most current LAN hardware for broadcast media such as wireless, is the ability to operate the network interface in "promiscuous" receive mode. Use of promiscuous mode does increase the software overhead on the CPU, but we believe that wireless network speeds and capacity are more the inherent limiting factors to performance in current and future systems. 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 switch the interface into promiscuous mode.

Use of promiscuous receive mode is entirely optional. Wireless communications between each pair of nodes will in many cases be able to operate bidirectionally, but at times the wireless link between two nodes may be only unidirectional, allowing one node to successfully send packets to the other while no communication is possible in the reverse direction.

A routing protocol such as DSR chooses a next- hop for each packet and provides the IP address of that next-hop. When the packet is transmitted, however, the lower-layer protocol often has a separate, MAC-layer address for the next-hop node. DSR uses the Address Resolution Protocol (ARP) [RFC826] to translate from next-hop IP addresses to next-hop MAC addresses. In addition, a node may add an entry to its ARP cache based on any received packet, when the IP address and MAC address of the transmitting node are available in the packet; for example, the IP address of the transmitting node is present in a Route Request option (in the Address list being accumulated) and any packets containing a source route. Adding entries to the ARP cache in this way avoids the overhead of ARP in most cases.

3. DESCRIPTION OF DSR PROTOCOL

3.1 Properties of the Protocol

Dynamic source routing protocol (DSR) is an on-demand protocol designed to restrict the bandwidth consumed by control packets in ad hoc wireless networks by eliminating the periodic table-update messages required in the table-driven approach. The basic approach of this protocol (and all other on-demand routing protocols) during the route construction phase is to establish a route by flooding RouteRequest packets in the network. The destination node, on receiving a RouteRequest packet, responds by sending a RouteReply packet back to the source, which carries the route traversed by the RouteRequest packet received.

Consider a source node that does not have a route to the destination. When it has data packets to be sent to that destination, it initiates a RouteRequest packet. This RouteRequest is flooded throughout the network. Each node, upon receiving a RouteRequest packet, rebroadcasts the packet to its neighbours if it has not forwarded it already, provided that the node is not the destination node and that the packet's time to live (TTL) counter has not been exceeded. Each RouteRequest carries a sequence number generated by the source

node and the path it has traversed. A node, upon receiving a RouteRequest packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate RouteRequest. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same RouteRequest by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forward a RouteRequest packet during the route construction phase. A destination node, after receiving the first RouteRequest packet, replies to the source node through the reverse path the RouteRequest packet had traversed. Nodes can also learn about the neighbouring routes traversed by data packets if operated in the promiscuous mode (the mode of operation in which a node can receive the packets that are neither broadcast nor addressed to itself). This route cache is also used during the route construction phase.

DSR 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 DSR Route Discovery Process

To perform route discovery, the source node broadcasts a route request packet with a recorded source route listing only itself. Each node that hears the route request forwards the request (if appropriate), adding its own address to the recorded source route in the packet. The route request packet propagates hop-by-hop outward from the source node until either the destination node is found or until another node is found that can supply a route to the target.

Nodes forward route requests if they are not the destination node and they are not already listed as a hop in the route. In addition, each node maintains a cache of recently received route requests and does not propagate any copies of a route request packet after the first.

All source routes learned by a node are kept (memory permitting) in a route cache, which is used to further reduce the cost of route discovery. A node may learn of routes from virtually any packet the node forwards or overhears. When a node wishes to send a packet, it examines its own route cache and performs route discovery only if no suitable source route is found.

Further, when a node receives a route request for which it has a route in its cache, it does not propagate the route request, but instead returns a route reply to the source node. The route reply contains the full concatenation of the recorded route from the source, and the cached route leading to the destination.

Naturally, if a route request packet reaches the destination node, the destination node returns a route reply packet to the source node with the full source to destination path listed.

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.

When any node A in the network 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. When another node receives a ROUTE REQUEST, if it is the target of the Route Discovery, it returns a ROUTE REPLY message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the ROUTE REQUEST; when the initiator receives this ROUTE REPLY, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the ROUTE REQUEST has

recently seen another ROUTE REQUEST message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the ROUTE REQUEST message, it discards the REQUEST. Otherwise, this node appends its own address to the route record in the ROUTE REQUEST message and propagates it by transmitting it as a local broadcast packet (with the same request id).

When initiating a Route Discovery, the sending node saves a copy of the original packet in a local buffer called the Send Buffer. The Send Buffer contains a copy of each packet that cannot be transmitted by this node because it does not yet have a source route to the packet's destination. Each packet in the Send Buffer is stamped with the time that it was placed into the Buffer and is discarded after residing in the Send Buffer for some timeout period; if necessary for preventing the Send Buffer from overflowing, a FIFO or other replacement strategy can also be used to evict packets before they expire.

While a packet remains in the Send Buffer, the node should occasionally initiate a new Route Discovery for the packet's destination address. However, the node must limit the rate at which such new Route Discoveries for the same address are initiated, since it is possible that the destination node is not currently reachable. In particular, due to the limited wireless transmission range and the movement of the nodes in the network, the network may at times become partitioned, meaning that there is currently no sequence of nodes through which a packet could be forwarded to reach the destination. Depending on the movement pattern and the density of nodes in the network, such network partitions may be rare or may be common.

If a new Route Discovery was initiated for each packet sent by a node in such a situation, a large number of unproductive ROUTE REQUEST packets would be propagated throughout the subset of the ad hoc network

3.3 DSR Route Maintenance Process

Conventional routing protocols integrate route discovery with route maintenance by continuously sending periodic routing updates. If the status of a link or node changes, the periodic updates will eventually reflect the change to all other nodes, 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 nodes. 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.

If a node along the path of a packet detects an error, the node returns a route error packet to the sender. The route error packet contains the addresses of the nodes at both ends of the hop in error. When a route error packet is received or overheard, the hop in error is removed from any route caches and all routes which contain this hop must be truncated at that point.

There are many methods of returning a route error packet to the sender. The easiest of these, which is only applicable in networks which only use bidirectional links, is to simply reverse the route contained in the packet from the original host. If unidirectional links are used in the network, the DSR protocol presents several alternative methods of returning route error packets to the sender.

Route maintenance can also be performed using end-to-end acknowledgments rather than the hop-by-hop acknowledgments described above. As long as some route exists by which the two end hosts can communicate, route maintenance is possible. In this case, existing transport or application level replies or acknowledgments from the original destination, or explicitly requested network level acknowledgments, may be used to indicate the status of the node's route to the other node.

When originating or forwarding a packet using a source route,each node transmitting the packet is responsible for confirming that the packet has been received by the next hop along the source route; the packet is retransmitted (up to a maximum number of attempts) until this confirmation of receipt is received. When node A in network has originated a packet for E using a source route through intermediate nodes B, C, and D. In this case, node A is responsible for receipt of the packet at B, node B is responsible for receipt at C, node C is responsible for receipt at D, and node D is responsible for receipt finally at the destination E. This confirmation of receipt in many cases may be provided at no cost to DSR, either as an existing standard part of the MAC protocol in use, or by a passive acknowledgement. If neither of these confirmation mechanisms are available, the node transmitting the packet may set a bit in the packet's header to request a DSR-specific software acknowledgement be returned by the next hop; this software acknowledgement will normally be transmitted directly to the sending node, but if the link between these two nodes is uni-directional, this software acknowledgement may travel over a different, multi-hop path. If the packet is

retransmitted by some hop the maximum number of times and no receipt confirmation is received, this node returns a ROUTE ERROR message to the original sender of the packet, identifying the link over which the packet could not be forwarded.

3.4 Some Additional Route Maintenance Properties

3.4.1 Packet Salvaging

After sending a ROUTE ERROR message as part of Route Maintenance a node may attempt to salvage the data packet that caused the ROUTE ERROR rather than discarding it. To attempt to salvage a packet, the node sending a ROUTE ERROR searches its own Route Cache for a route from itself to the destination of the packet causing the ERROR. If such a route is found, the node may salvage the packet after returning the ROUTE ERROR by replacing the original source route on the packet with the route from its Route Cache. The node then forwards the packet to the next node indicated along this source route.

When salvaging a packet in this way, the packet is also marked as having been salvaged, to prevent a single packet being salvaged multiple times. Otherwise, it could be possible for the packet to enter a routing loop, as different nodes repeatedly salvage the packet and replace the source route on the packet with routes to each other.

3.4.2 Increased Spreading of ROUTE ERROR Message

When a source node receives a ROUTE ERROR for a data packet that it originated, this source node propagates this ROUTE ERROR to its neighbours by piggybacking it on its next ROUTE REQUEST. In this way, stale information in the caches of nodes around this source node will not generate ROUTE REPLYs that contain the same invalid link for which this source node received the ROUTE ERROR.

We have also considered, but not simulated, a further improvement to Route Maintenance in which a node, that receives a ROUTE ERROR will forward the ERROR along the same source route that resulted in the ERROR. This will almost guarantee that the ROUTE ERROR reaches the node that generated the ROUTE REPLY containing the broken link, which will prevent that node from contaminating a future Route Discovery with the same broken link.

3.5 Multicast Routing with DSR

DSR does not currently support true multicast routing, but does support an approximation of this that is sufficient in many network contexts. Through an extension of the Route

Discovery mechanism, DSR supports the controlled flooding of a data packet to all nodes in the ad hoc network that are within some specified number of hops of the originator; these nodes may then apply destination address filtering (e.g., in software) to limit the packet to those nodes subscribed to the packet's indicated multicast destination address. While this mechanism does not support pruning of the broadcast tree to conserve network resources, it can be used to distribute information to all nodes in the ad hoc network subscribed to the destination multicast address. This mechanism may also be useful for sending application level packets to all nodes in a limited range around the sender.

3.6 DSR in the ISO Network Model

When designing DSR, we had to determine at what layer within the protocol hierarchy to implement ad hoc network routing. We considered two different options: routing at the link layer (ISO layer 2) and routing at the network layer (ISO layer 3). Originally, we opted to route at the link layer for several reasons.

4. DSR SIMULATION SUMMARY

Our simulation environment consists of a set of wireless and mobile networking extensions that we have cre-ated [Broch 1998], based on the publicly- available ns-2 network simulator from the University of California at Berkeley and the VINT Project [Fall 1997]. These extensions provide a detailed model of the physical and link layer behavior of a wireless network and allow arbitrary movement of nodes within the network. At the physical layer, we provide realistic modeling of factors such as free space and ground reflection propagation, transmission power, antenna gain, receiver sensitivity, propagation delay, carrier sense, and capture effect [Rappaport 1996]. At the link layer, we model the complete Distributed Coordination Function (DCF)Media Access Control (MAC) protocol of the IEEE 802.11 wireless LAN protocol standard [IEEE 1997]. These wireless and mobile networking extensions are available from the Carnegie Mellon University Monarch Project web pages [Monarch] and have been widely used by other researchers; a version of them have also now been adopted as a part of the standard VINT release of ns-2.

We have done a number of different simulation studies with this environment, analyzing the behavior and performance of DSR and comparing it to other proposed routing protocols for ad hoc networks [Broch 1998, Maltz 1999a].

In the random waypoint mobility model [Johnson 1996a], each mobile node begins at a

random location and and moves independently during the simulation. Each node remains stationary for a specified period that we call the pause time and then moves in a straight line to some new randomly chosen location at a randomly chosen speed up to some maximum speed. Once reaching that new location, the node again remains stationary for the pause time, and then chooses a new random location to proceed to at some new randomly chosen speed, and the node continues to repeat this behaviour throughout the simulation run. We have found that this model can produce large amounts of relative node movement and network topology change, and thus provides a good movement model with which to stress DSR or other ad hoc network routing protocols.

5. CONCLUSION

The Dynamic Source Routing protocol (DSR) provides excellent performance for routing in multi-hop wireless ad hoc networks. DSR has very low routing overhead and is able to correctly deliver almost all originated data packets, even with continuous, rapid motion of all nodes in the network.

A key reason for this good performance is the fact that 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 routing 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.

6. REFERENCES

1. [Broch 1998] 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 (MobiCom'98), pages 85–97, Dallas, TX, October 1998. ACM.
2. [Bantz 1994] David F. Bantz and Fred´eric´ J. Bauchot. Wireless LAN Design Alternatives. IEEE Network, 8(2):43–53, March/April 1994.

3. [Bharghavan 1994] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A Media Access Protocol for Wireless LAN's. In Proceedings of the ACM SIGCOMM '94 Conference, pages 212–225. ACM, August 1994.
4. [Braden 1989] Robert T. Braden, editor. Requirements for Internet Hosts Communication Layers. RFC 1122, October 1989.
5. [Broch 1999a] 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-03.txt, October 1999. Work in progress. Earlier revisions published June 1999, December 1998, and March 1998.
6. [Broch 1999b] Josh Broch, David A. Maltz, and David B. Johnson. Supporting Hierarchy and Heterogeneous Interfaces in Multi-Hop Wireless Ad Hoc Networks. In Proceedings of The International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'99), Workshop on Mobile Computing, Perth, Western Australia, June 1999. IEEE Computer Society.
7. [Castaneda˜ 1999] Robert Castaneda˜ and Samir R. Das. Query Localization Techniques for On-demand Routing Protocols in Ad Hoc Networks. In Proceedings of the Fifth International Conference on Mobile Computing and Networking (MobiCom'99). ACM, August 1999.
8. [Cheshire 1996] Stuart Cheshire and Mary Baker. Internet Mobility 4x4. In Proceedings of the SIGCOMM '96, pages 318– 329. ACM, August 1996.