A Study on Dynamic Source Routing In Ad Hoc Wireless Networks

Ms.C.Suriya, Mrs.R,Sudha
Asst Prof, Dept of Computer Science, K.S.R College of Arts and Science, Tiruchengode – 637215, India
Asst Prof, Dept of Computer Science, K.S.R College of Arts and Science, Tiruchengode – 637215, India
E-mail:sylvester142@gmail.com

---**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 main mechanisms of "Route Discovery" and "Route Maintenance", which work together to allow nodes to discover and maintain routes to arbitrary destinations in the ad hoc network. 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. The protocol allows multiple routes to any destination and allows each sender to select and control the routes used in routing its packets, for example for use in load balancing or for increased robustness.

Keywords: (Dynamic Source Routing, Hops, packets, MAC layer , Time To Live(TTL),Router**.**)

1.Introduction

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. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. Network nodes co-operate 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.

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 is 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 for subsequent packets to D Route Maintenance for this route is used only when S is actually sending packets to D.

2. Methods

The DSR Protocol as described here is designed mainly for mobile ad hoc networks of up to about two hundred nodes, and is designed to work well with even very high rates of mobility. Other protocol features and enhancements that may allow DSR to scale to larger networks are outside the scope of this document.

We assume in this document 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.

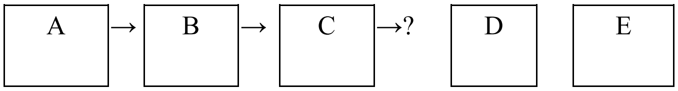
The diameter of a ad hoc network is the minimum number of hops necessary for a packet to reach from any node located at one extreme edge of the ad hoc network to another node located at the opposite extreme. We assume that this diameter will often be small (e.g., perhaps 5 or 10 hops), but may often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

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 made the flooding of every individual data packet the only possible routing protocol.

3. DSR PROTOCOL OVERVIEW

This section provides an overview of the operation of the DSR protocol. The basic version of DSR uses explicit “source routing”, in which each data packet sent carries in its header the complete, ordered list of nodes through which the packet will pass. This use of explicit source routing allows the sender to select and control the routes used for its own packets, supports the use of multiple routes to any destination (for example, for load balancing), and allows a simple guarantee that the routes used are loop-free; bu including this source route in the header of each data packet, other nodes forwarding or overhearing any of these packets can also easily cache this routing information for future use.

When originating or forwarding a packet using a source route, each node transmitting the packet is responsible for confirming that data can flow over the link from that node to the next hop. For example, in the situation shown below, node A has originated a packet for node E using a source route through intermediate nodes B, C, and D:



In this case, node A is responsible for the link from A to B, node B is responsible for the link from B to C, node C is responsible for the link from C to D, node D is responsible for the link from D to E.

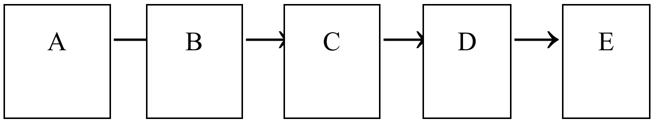
3.3. Additional Route Discovery Features

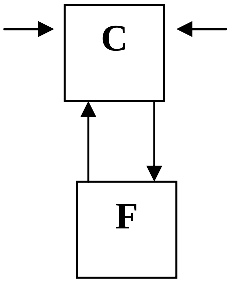
3.3.1. Caching Overhead Routing Information

A node forwarding or otherwise overhearing any packet should add all usable routing information from that packet to its own Rote Cache. The usefulness of routing information in a packet depends on the directionality characteristics of the physical medium, as well as the MAC protocol being used. Specifically, three distinct cases are possible:

Links in the network frequently are capable of operating only unidirectionally (not bidirectionally), and the MAC protocol in use in the network is capable of transmitting unicast packets over unidirectional links.

Links in the network occasionally are capable of operating only unidirectionally (not bidirectionally), but this unidirectional restriction on any link is not persistent, almost all links are physically bidirectional, and the MAC protocol in use in the network is capable of transmitting unicast packets over unidirectional links.





The concatenation of the accumulated route record from the Route Request and the cached route from F's Route Cache would include a duplicate node in passing from C to F and back to C.

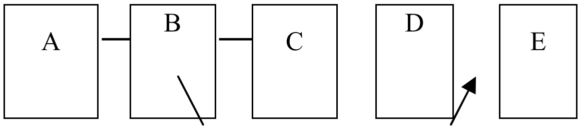
Request; it provides an inexpensive method for determining if the target is currently a neighbor of the or if a neighbor node has a route to the target cached (effectively using the neighbors' Route caches as an extension of the initiator's own Route Cache). If no Route Reply is received after a short timeout, then the node sends a "propagating" Route Request for the target node (i.e., with hop limit as defined by the value of the Discovery-Hop-Limit configuration variable).

3.4. Additional Route Maintenance Features

3.4.1. Packet Salvaging

When an intermediate node forwarding a packet detects through Route Maintenance that the next hop along the route for that packet is broken, if the node has another route to the packets destination in its Route Cache, the node should "salvage" the packet rather than discarding it. To salvage a packet, the node replaces 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. For example, in the situation shown in the example of, if mode C has another route cached to node E, it can salvage the packet by replacing the original route in the packet with this new route from its own Route Cache, rather than discarding the packet.

When salvaging a packet, a count is maintained in the packet of the number of times that it has been salvaged, to prevent a single packet from being salvaged endlessly. Otherwise, since TTL is decremented.



Cache). On the Route Request packet initiating this Route Discovery, node A piggybacks a copy of this Route Error spreads well to other nodes, and guaranteeing that any Route Reply that it receives (including those from other node's Route Caches) in response to this Route Request does not contain a route that assumes the existence of this broken link.

3.5. Optional DSR Flow State Extension

This section describes an optional, compatible extension to the DSR protocol, known as "flow state", that allows the routing of most packets without an explicit source route header in the packet. The DSR flow state extension further reduces the overhead of the protocol yet still preserves the fundamental properties of DSR's operation. Once a sending node has discovered a source route such as through DSR's Route Discovery mechanism, the flow state mechanism allows the sending node to establish hop-by-hop forwarding state within the network, based on this source route, to enable each node along the route to forward the packet to the next hop based on the node's own local knowledge of the flow along which this packet is being routed. Flow state is dynamically initialized the by first packet using a source route and is then able to route subsequent packets along the same flow without use of a source route header in the packet.

The state established at each hop along a flow is "soft state" and thus automatically expires when no longer needed and can be quickly recreated as necessary. Extending DSR's basic operation based on an explicit source route in the header of each packet routed, the flow state extension operates as a form of "implicit source routing" by preserving DSR's basic operation but removing the explicit source route from packets.

3.5.2. Receiving and Forwarding Establishment Packets

Packets intended to establish a flow, contain a DSR options header with a Source Route option, and are forwarded along the indicated route. A node implementing the DSR flow state extension, when receiving and forwarding such a DSR packet, also keeps some state in its own Flow Table to enable it to forward future packets that are sent along this flow with only the flow ID specified. Specifically, if the packet also contains a DSR Flow State header, this packet should cause an entry to be established for this flow in the Flow Table of each node along the packet's route.

The Hop Count field of the DSR Flow State header is also stored in the Flow Table, as is Lifetime option specified in the DSR options header.

4. CONCEPTUAL DATA STRUCTURES

This document describes the operation of the DSR protocol in terms of a number of conceptual data structures. This section describes each of these data structures and provides an overview of its use in the protocol. In an implementation of the protocol, these data structures may be implemented in any manner consistent with the external behavior described in this document. Additional conceptual data structures are required for the optional flow state extensions to DSR.

4.1. Route Cache

Each node implementing DSR must maintain a Route Cache, containing, routing information needed by the node. A node adds information to its Route Cache as it learns of new links between nodes in the ad hoc network; for example, a node may learn of new links when it receives a packet carrying a Route Request, Route Reply, or DSR source route. Likewise, a node removes information from its Route Cache as it learns that existing links in the ad hoc network have broken; for example, a node may learn of a broken link when it receives a packet carrying a Route Error or through the link-layer retransmission mechanism reporting a failure in forwarding a packet to its next-hop destination.

5. Additional Conceptual Data Structures for Flow State Extension

This section defines additional conceptual data structures used by the optional "flow state" extension to DSR. In an implementation of the protocol, these data structures may be implemented in any manner consistent with the external behavior described in this document.

6. DSR OPTIONS HEADER FORMAT

The Dynamic Source Routing protocol makes use of special header carrying control information that can be included in any existing IP packet. This DSR Options header in a packet contains a small fixed-sized, 4-octet portion, followed by a sequence of zero or more DSR options carrying optional information. The end of the sequence of DSR options in the DSR Options header is implied by total length of the DSR Options header.

For Ipv4, the DSR Options header must immediately follow the IP header in the packet. (If a Hop-by-Hop Options extension header, as defined in Ipv6, becomes defined for Ipv4, the DSR Options header must immediately follow the Hop-by-Hop Options extension header, if one is present in the packet, and must otherwise immediately follow the IP header).

<table>
<tr><td>Option Type</td><td>Opt Data Len</td><td>Identification</td></tr>
<tr><td colspan="3">Target Address</td></tr>
<tr><td colspan="3">Address (1)</td></tr>
<tr><td colspan="3">Address (2)</td></tr>
<tr><td colspan="3">……………..</td></tr>
<tr><td colspan="3">Address (n)</td></tr>
</table>

IP fields

Source Address

must be set to the address of the node originating this packet. Intermediate nodes that retransmit the packet to propagate the Route Request must not change this field.

must be set to the IP limited broadcast address (255.255.255.255). Hop Limit (TTL) may be varied from 1 to 255, for example to implement non-propagating Route Requests and Route Request expanding-ring searches.

Route Request fields: Option Type 2. Opt Data Len 8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Opt Data Len fields.

Identification

A unique value generated by the initiator (original sender) of the Route Request. Nodes initiating a Route Request generate a new Identification value for each Route Request, for example based on a sequence number counter of all Route Requests initiated by the node.

This value allows a receiving node to determine whether it has recently seen a copy of this Route Request: if this Identification value is found by this receiving node in its Route Request Table (in the cache of Identification values in the entry there for this initiating node), this receiving node must discard the Route Request. When propagating a Route Request, this field must be copied from the received copy of the Route Request being propagated.

Target Address

The address of the node that is the target of the Route Request. Address [1..n]

Address[i] is the address of the I-th node recorded in the Route Request option. The address given in the Source Address field in the IP header is the address of the initiator of the Route Discovery and must not be listed in the Addresss [i] fields; the address given in Address[1] is thus the address of the first node on the path after the initiator. The number of addresses present in this field is indicated by the Opt Data Len field in the option (n=(Opt Data Len – 6)/4). Each node propagating the Route Request adds its own address to this list, increasing the Opt Data Len value by 4 octets.

The Route Request option must not appear more than once within a DSR Options header

6.3. Route Reply Option

The Route Reply option in a DSR Options header is encoded as follows:

0 1
2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9 0 1

Option Type	Opt Data Len	1	Reserved
Address (1)			
Address (2)			
……………..			
Address (n)			

Source Address

Set to the address of the node sending the Route Reply. In the case of a node sending a reply from its Route Cache or sending a gratuitous Route Reply, this address can differ from the address that was the target of the Route Discovery.

Destination Address must be set to the address of the source node of the route being returned. Copied from the Source Address field of the Route Request generating the Route Reply, or in the case of a gratuitous Route Reply, copied from the Source Address field of the data packet triggering the gratuitous Reply.

6.4. Route Error Option

The Route Error option in a DSR Options header is encoded as follows:

0 1 2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
3 4 5 6 7 8 9 0 1

Option Type	Opt Data Len	Error Type	Reserved	Salvage
Error Source Address				
Error Destination Address				
Type Specification Information				

Option Type

3. Nodes not understanding this option will ignore this option.

Opt Data Len

8-bit unsigned integer, Length of the option, in octets, excluding the Option Type and Opt Data Len fields.

The type of error encountered, Currently, the following type values are defined:

1 = NODE_ UNREACHABLE
2 = FLOW_STATE_not_SUPPORTED
3 = OPTION_not_SUPPORTED

Other values of the Error Type field are reserved for future use.

Reserved

Reserved must be sent as 0 and ignored on reception.

Salvage

A 4-bit unsigned integer, Copied from the Salvage field in the DSR Source Route option of the packet triggering the Route Error.

6.5. Acknowledgement Request Option

The Acknowledgement Request option in a DSR Options header is encoded as follows:

0 1 2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
3 4 5 6 7 8 9 0 1

Option Type	Opt Data Len	Identification

Option Type

160. Nodes not understanding this option will remove the option and return a Route Error.

Opt Data Len

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Opt Data Len fields.

Identification

The Identification field is set to a unique value and is copied into the Identification field of the Acknowledgement option when returned by the node receiving the packet over this hop. An Acknowledgement Request option must not appear more than once within a DSR Options header.

Len fields. For the format of the DSR Source Route option defined here, this field must be set to the value (n * 4) + 2, where n is the number of addresses present in the Address(i) fields.

First Hop External (F)

Set to indicate that the first hop indicated by the DSR Source Route option is actually an arbitrary path in a network external to the DSR network; the exact route outside the DSR network is not represented in the DSR Source Route option. Nodes caching this hop in their Route Cache must flag the cached hop with the External flag. Such hops must not be returned in a Route Reply generated from this Route Cache entry, and selection of routes from the Route Cache to route a packet being sent must prefer routes that contain no hops flagged as external.

present in the Address(1..n) field is indicated by the Opt Data Len field in the option (n=(Opt Data Len-2)/4)

When forwarding a packet along a DSR source route using a DSR Source Route option in the packet's DSR Options header, the Destination Address field in the packet's ultimate destination. A node receiving a packet containing a DSR Options header with a DSR Source Route option must examine the indicated source route to determine if it is the intended next-hop node for the packet and determine how to forward the packet.

7.1. General Packet Processing

When originating any packet, a node using DSR routing must perform the following sequence of steps:

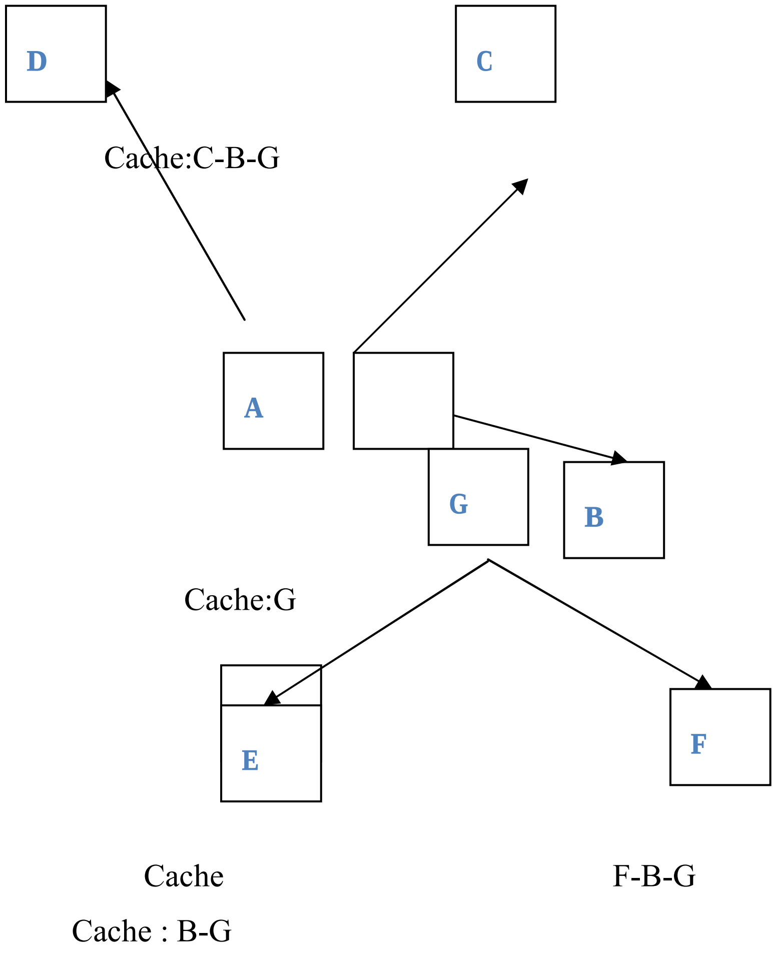
✓ Search the node's Route Cache for a route to the address given in the IP Destination Address field in the packet's header.

✓ If no such router is found in the Route Cache, then perform Route Discovery for the Destination Address.

7.2. Route Discovery Processing

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. The node initiating a Route Discovery is known as the "initiator" of the Route Discovery, and the destination node for which the Route Discovery is initiated is known as the "target" of the Route Discovery.

neighbor may attempt to send a Route Reply, thereby wasting bandwidth and possibly increasing the number of network collisions in the area.



7.3. Route Maintenance Processing

Route Maintenance is the mechanism by which a source node S is able to detect, while using a source route to some destination node 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 that 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 for subsequent packets to D. Route Maintenance for this route is used only when S is actually sending packets to D. Specifically, when forwarding a packet, a node must attempt to confirm the reachability of the next-hop node, unless such confirmation had been received in the last MaintHoldoffTime.

7.4. Multiple Network Interface Support

A node using DSR may have multiple network interfaces that supported hoc network routing. This section describes special packet processing at such nodes.

A node with multiple network interfaces must have some policy for determining which Route Request packets are forwarded all Route Requests out all network interfaces.

- ✓ Append the address of the incoming network interface.
- ✓ Append the address of the outgoing network interface.

7.5. IP Fragmentation and Reassembly

When a node using DSR wishes to fragment a packet that contains a DSR header not containing a Route Request option, it must perform the following sequence of steps:

- ✓ Remove the DSR Options header from the packet.
- ✓ Fragment the packet. When determining the size of each fragment to create from the original packet, the fragment size must be reduced by the size of the DSR Options header from the original packet.

7.6. Flow State Processing

A node implementing the optional DSR flow state extension must follow these additional processing steps.

7.6.1. Originating a Packet

When originating any packet has never had a DSR flow state established along it (or the existing flow state has expired):

- ✓ If the route to be used for this packet has never had a DSR flow state established along it (or the existing flow state has expired):
- ✓ Generate a 16-bit Flow ID larger than any unexpired Flow IDs used for this destination. Odd Flow IDs must be chosen for "default" flows; even Flow IDs must be chosen for non-default flows:
- ✓ Add a DSR Options header.

8. CONCLUSION

This paper has presented a protocol for routing packets between wireless mobile hosts in an ad hoc network. Unlike routing protocols using distance vector or link state algorithms, our protocol uses dynamic source routing which adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which host 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. This document does not specifically address security concerns.

References

[1]. David F.Bantz and Frederic J.Bauchot. Wireless LAN Design Alternatives. IEEE Network, 8(2):43-53, March/April 1994.

[2]. 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, August 1994.

[3]. Robert T.Barden, editor, Requirements for Internet Hosts --- Communication Layers. RFC 1122, October 1989.

[4]. Scott Bradner, Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, March 1997.

[5]. Josh Broch, David A.Malz, David B.Johnson, Yith-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, October 1998.

[6]. David D.Clark. The Design Philosophy of the DARPA Internet Protocols. In Proceedings of the ACM SIGCOMM'88 Conference, pages 106-114, August 1988.

[7]. Stephen E.Deering and Robert M.Hinden, Internet Protocol Version 6 (Ipv6) Specification, RFC 2460, December 1998.

[8]. Ralph Droms, Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[9]. The Free BSD Project, Project web page available at http://www.freebsed.org/.

[10]. Yih-Chun Hu and David B.Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. In Proceedings of the

Sixth Annual ACM International Conference on Mobile Computing and Networking, August 2000.

[11]. Yih-Chun Hu and David B.Johnson. Implicit Source Routing in On-Demand Ad Hoc Network Routing. In Proceedings of the Second Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), pages 1-10, October 2001.

[12]. Yih-Chun Hu, Adrian Perrig, and David B.Johnson. Ariadne: A Secure On Demand Routing Protocol for Ad Hoc Networks. In Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002), pages 12-23 September 2002.

[13]. IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, New York 1997.

[14]. Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 195-206, August.

[15]. 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.

[16]. 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.

[17]. David B.Johnson and David A.Maltz. Protocols for Adaptive Wireless and Mobile Networking. IEEE Personal Communications, 3(1):34—42, February 1996.

[18]. John Jubin and Janet D.Tornow. The DARPA Packet Radio Network Protocols. Proceedings of the IEEE, 75(1):21-32, January 1987.

[19]. Phil Karn. MACA—A New Channel Addess method for Packet Radio. In ARRL/CRRL Amateur Radio 9th Computer Networking Conference, pages 134-140, September 1990.

[20]. Gregory S.Lauer. Packet – Radio Routing. In Routing in Communications Networks edited by Martha E.Steenstrup, chapter 11, pages 351—396. Prentice-Hall, Englewood Cliffs, New Jersy, 1995.

[21]. David.A.Malz, Josh Broch, Jorjeta Jetcheva, and David B.Johnson. The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks. IEEE Journal on Selected Areas of Communications, 17(8): 1439-1453, August 1999.

[22]. David A.Maltz. Josh Broch, and David B.Johnson, Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed. Technical Report CMU-C8-99-116, School of Computer science, Carnegic Mellon University, Pittsburgh, Pennylvania, March 1999.

[23]. David A.Maltz. Josh Broch, and David B.Johnson, Quantitative Lessons from a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed. In Proceedings of the IEEE Wireless Communications and Networking Conference, September 2000.

[24]. David A.Maltz, Josh Broch, and David B.Johnson, Lessons From a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed. IEEE Personal Communications, 8 (1) 8-15, February 2001