Routing in Mobile Ad Hoc Networks

Christian Brechenmacher, Jonas Andre*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: brechenm@in.tum.de, andre@in.tum.de

Abstract—**This paper presents the different MANET routing types, introduces some specific routing algorithms, and draws a comparison between them. The paper focuses on the two main types of MANET routing, Table-Driven and On-Demand. It presents the Table-Driven DSDR protocol, and the On-Demand DSR protocol. The comparison will draw a spotlight on differences of performance in the categories overhead, availability and scalability. While Table-Driven protocols outperform On-Demand protocols on availability, as the MANET grows in size, the performance of Table-Driven protocols suffers as their rather large overhead worsens their scalability.**

Index Terms—**manet, routing, protocols, proactive, reactive, dsdv, dsr, ad hoc network, algorithm design and analysis**

1. Introduction

Mobile Ad Hoc Networks (*MANET*) consist of nodes that do not require the intervention of an access point. They are envisioned to be dynamic, multi-hop topologies that are composed of wireless links. Supporting this type of mobility requires routing protocols, as it ensures efficient use of the scarce capacity that wireless bandwidth has to offer. *MANET*s are incredibly useful, especially in situations when no other infrastructure is provided. As a result, there is a plethora of *MANET* routing algorithms.

*MANET*s are flexible alternatives to generic Wifi networks, not only in situations where critical infrastructure goes down, like in war zones, disaster areas or other emergencies; they are also very useful in planes, connecting cars and in other decentralized applications. The basic idea of the Ad Hoc Network in general is to enable transactions between different devices without the need for a router or similar base stations. Another difference to fixed networks is that the connection built by an Ad Hoc Network serves only one purpose and could only be temporary. Information in an Ad Hoc Network may be transferred over multiple nodes, if necessary, which is then called a multi-hop network. Ad Hoc Networks can be used locally, as a connection between several devices communicating wirelessly, or it may permit exogenous traffic to transit through one or several of the nodes [1].

2. Characteristics of *MANET*

Considering this, it becomes clear that *MANET* routing poses some unique challenges. The salient characteristics of *MANET* are [2]:

- Dynamic topologies: The nodes are free to move, which changes the network architecture unpredictably. The links between the nodes may consist of both unidirectional and bidirectional links.
- Bandwidth-constrained, variable capacity links: Wireless links have a lower capacity than hardwired ones. A lower thoughput because of multiple accesses, fading, noise, and interference conditions, may lead to congestion.
- Energy-constrained operation: As the network and its nodes are mobile, the energy resources it drains from may be exhaustible.
- Limited physical security: Because of *MANET*'s wireless connection, the danger of eavesdropping, spoofing or DOS-attacks is increased. However, its decentralized nature proves as a benefit in terms of robustness against single points of failure.

These characteristics provide a set of diverse prerequisites and performance concerns for routing designs, extending beyond those of the static, highspeed topology of wired networks. Routing is the strategy that oversees the way the nodes decide to forward packets between them. For most protocols, the manner in which they are sent boils down to the *Shortest Path Problem*. Each node should therefore maintain a preferred neighbour for each destination packets can be sent.

Due to the lack of one centralised router, in Ad Hoc Networks, nodes have to become aware of their topology themselves. New nodes therefore announce themselves and listen for announcements from their neighbours, hereby attaining knowledge about nodes nearby [2].

3. Related Work

As MANET is an extensive research field, there are many similar works available. Geetha Jayakumar and G. Gopinath [3] thoroughly describe the taxonomy of Ad Hoc Networks, before giving an outline of different Ad Hoc Network routing protocols. They then compare the benefits of the different routing types. Shima Mohseni et al. [4] also give a short overview over different Ad Hoc Network routing protocols, before discussing the discriminating factors between them.

Yuxia et al. [5] ran some simulations to determine how different protocols perform as the network's size increases, the results of which will also be used in this paper.

doi: 10.2313/NET-2020-11-1_03

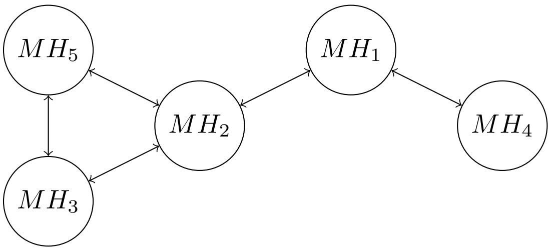

Figure 1: A bidirectional Ad Hoc Network

4. Table-Driven and On-Demand Routing

There are two main approaches to routing in *MANET*: *Table-Driven* (proactive) and *On-Demand* (reactive).

The basic idea of the proactive design is to operate each node as a specialized router, which periodically advertises its view of the interconnection topology with other mobile hosts within the network. The routing information is maintained on every node, even before it is needed.

A node contained in an environment maintained by a reactive routing protocol sends packets over routes that are determined in an On-Demand manner. There are two types of protocols needed for this approach to work [6]:

- *Route Discovery*: a route between two certain nodes is not yet known, meaning it is not yet in the sending node's routing table. A broadcast with a route discovery packet is sent through the network, and once a route has been discovered, a connection is established and data can be transmitted.
- *Route Maintenance*: as one of the characteristics of *MANET* is its dynamic topology, the occurence of link breaks must be taken into account. A node should therefore be able to confirm that a packet has been orderly received by its next-in-line node. If that is not the case, the node can then choose another route that is already known, or can invoke route discovery to find another path.

5. The DSDV Routing Protocol

One proactive protocol is the *Destination-Sequenced Distance-Vector* routing protocol (DSDV) [7]. Its basic idea is to display a shortest path for each node contained in the *MANET* by managing routing tables stored at each station of the network. Each of these tables lists all possible destinations and the number of hops over auxiliary stations required to get there. Each entry in these tables is tagged with a sequence number (see tables 1 and 2), which stems from the route's destination and is, usually, always even. It guarantees the freshness of the information by increasing the sequence number when initializing a new update period. To remain a consistent picture of the network's topology, these updates have to be frequent, and immediate when significant new information arises. As there is no time synchronization between the hosts of the network, the update periods are not coordinated between them, although some regularity can be expected.

The routing information is advertised by broadcasting packets with the network information and is transmitted in *Network Protocol Data Units* (NPDU). They also contain information about the broadcaster and an incremented sequence number in the header of the packet. The extend of the routing information depends on the type of update that is occurring: one type carries all the available information, called the *full dump*. That includes the destination, the metric – which informs about the number of hops needed until the destination is reached – and the sequence number.

The other is called an *incremental* and only contains the information that has changed since the last full dump occured.

TABLE 1: Advertised route table of MH_3 (full dump)

Destination	Metric	Sequence number
MH_1	2	S380_MH_1
MH_2	1	S256_MH_2
MH_3	0	S468_MH_3
MH_4	3	S176_MH_4
MH_5	1	S324_MH_5

Table 1 depicts the advertised full dump of mobile host 3 (MH_3) from figure 1. It is visible that, for instance, MH_1 with its sequence number S380 is two hops away.

A full dump requires, even for small networks, multiple NPDUs, while an incremental regularly only requires one NPDU. Because of its size, full dumps can be scheduled to happen infrequently when the topology of the network is not changing. As change becomes more frequent, full dumps can be scheduled to occur more often to decrease the size of the next incremental.

Routes advertised in a broadcast are updated in the routing tables of the receiver and readvertised in the broadcast of the receiver. Firstly, the receiver integrates the new information in its forwarding information table. That includes the destination, the next hop, the metric, the sequence number, the install number – which informs about the time the station became visible to the current station – flags, and a pointer that points to alternative routes. Table 2 shows such a table from the perspective of MH_3. Picking up on our earlier example, it is now additionally visible that the next hop on the route to MH_1 would be MH_2. We can see that MH_1 was the last host that became visible to MH_3, as it has the highest install number. In this example, all pointers would point to null structures, as there are no routes that compete with each other, or that are in any way likely to be superseded, if they were, for instance, broken links.

If a route's sequence number is more recent then before, it is always used. The number of hops required for one route, received by the broadcast, is increased by one. If the route now requires fewer hops than before and the sequence number is recent, it is updated in the routing table of the receiver. Alternatively to updating the table entries, which effectively discards old information, it is also possible to just create a new entry, and use this one preferably. This, however, results in a bigger overhead, and is therefore not recommendable if the network is stable.

If a link to another router is broken, it can be detected through the layer-2 protocol, or can just be inferred if there has been no communication from this host in a while. Its sequence number is then updated and its metric is set to infinity. Sequence numbers indicating a broken link with an infinite metric are always odd numbers. That way they

TABLE 2: All forwarding information of MH_3

Destination	NextHop	Metric	Sequence number	Install	Flags	Stable_data
MH_1	MH_2	2	S380_MH_1	T003_MH_3		Ptr1_MH_1
MH_2	MH_2	1	S256_MH_2	T001_MH_3		Ptr1_MH_2
MH_3	MH_3	0	S468_MH_3	T001_MH_3		Ptr1_MH_3
MH_4	MH_2	3	S176_MH_4	T002_MH_3		Ptr1_MH_4
MH_5	MH_5	1	S324_MH_5	T002_MH_3		Ptr1_MH_5

are superseded by a new sequence number if a connection towards that host can be established again. A broken link, as well as information about a host which was previously unreachable, is important information which needs to be broadcasted immediately.

However, this is not the case for a normal update. In a suboptimal setting, a router can receive a route with an updated sequence number, but with a bad metric, first, before it receives another route with the same sequence number and a better metric. Were it to immediatly broadcast the new route, it would lead to a burst of route broadcasts from that receiver, which resolves in a chain reaction of broadcasts from all the hosts in the network. A solution to this problem is to delay the broadcast of the router, but already use the new information for itself to forward received packages. That concludes that two tables have to be maintained: one for the purpose of routing, the other one for broadcasting. The host also has to maintain a history of the average time it takes from the first route to the best route to arrive. Based on that, a host may be able to predict how long it has to wait before advertising its routes.

6. The DSR Routing Protocol

In the reactive *Dynamic Source Routing* routing protocol (DSR) [6], route information is maintained in a *Route Cache*. If a destination can not be found in there, the Route Discovery protocol is initiated. All possible routes to a target are contained in the Route Cache in order to dynamically replace them if one link breaks, and not having to initiate a Route Discovery immediately. All routes have to be maintained, a task taken over by the Route Maintenance protocol. The route of a specific packet is carried in its header. This provides control over its path, for example ensuring loop prevention.

6.1. Route Discovery

The Route Discovery is initiated by the node willing to send data (initiator). It broadcasts a *Route Request* to all nodes that are in range. Every Route Request contains the following information: the initiator and the target of the Route Request, a unique identifier that is set by the initiator, and a record of the nodes through which this Route Request has already passed.

If the receiver of such a Route Request is the intended target, it generates a *Route Reply* that contains the record of nodes through which the Route Request has passed. Upon receiving the Route Reply, the initiator of the request stores the information in its Route cache and can now send packets to the target.

If the receiver is not the intended target, it compares the identifier and initator of the Route Request with that of Route Requests it has recently seen. If it finds a match, or if the record of nodes already passed contains this node, the Request is discarded silently. The node then looks up its Route Cache for a route to the target. If it finds one, it generates a Route Reply itself, and adds the route from its Route Cache to the record of the Route Request. Here, it has to eliminate possible loops. For example, if the record shows the route A→B→C with destination E and node C has cached the route C→B→D→E, the loop B→C→B is eliminated to generate a Route Reply with the route A→B→D→E. Otherwise, the receiver adds itself to the record, and then broadcasts the Route Request to be processed further.

A Route Request generally has a hop limit that is used to limit the number of nodes allowed to forward the request. This can be used to limit the spread of a request through the network. If a request with a low hop limit does not find its target, the hop limit is increased and the Route Reply is sent again.

The Route Reply can be sent back in two different ways: The target can use its own Route Cache and utilize a route stored from it. If a route to the initiator is not present, the target has to initiate a Route Discovery to the initiator himself. To avoid infinite Route Discoveries, however, the target has to store the Route Reply to the initator in its Route Request. The other way is to simply reverse the sequence of nodes of the Route Request and use this as the source route of the Route Reply.

Routes can also be discovered by transmitting nodes that overhear packets that are sent over an unknown route. However, one has to keep in mind that in a network containing mainly unidirectional links, routes can only be cached from the current route forward. If a network contains mainly bidirectional links, routes should be cached in both directions.

While the Route Discovery is ongoing, the initiating packets must be stored in a buffer. Every node therefore maintains a ‘Send buffer’ that contains every packet that could not yet be transmitted because there is no route available. Each packet is tagged with its arrival time and is discarded after a specific amount of time passes. If the buffer is overflowing nevertheless, the packets will have to be deleted with some replacement strategy, like FIFO. While a packet resides in the Send Buffer, its node should, at a limited rate, initiate new Route Discoveries. The initiation should be limited as the target node could not be available at the moment.

6.2. Route Maintenance

Each node is responsible to ensure that the link to its next hop is working. It can do so by the following means:

- The acknowledgement is an existing part of the MAC layer in use (e.g. link-layer acknowledgement defined in IEEE 802.11 [8]).
- Passive Acknowledgement: the node overhears its next hop send the package itself to another node.
- Explicit request: the node can issue an explicit request to receive a software acknowledgement by the next hop, either directly or via a different route. Upon receiving an acknowledgement from the next hop, the node may not require another for a specific amount of time. The requesting node issues a request a specific number of times, before it assumes the link as broken and sends a *Route Error* message to the sender of the packet. The node receiving a Route Error should update its Route Cache accordingly and use a different route, or initiate a Route Discovery for the target node.

7. Comparison

Table 3 depicts the performance metrics Availability of Routes, Delay of Route Acquisition, Control Overhead, Required Bandwidth, Memory Requirements and Scalability, that will now be inspected further.

TABLE 3: Pro- and reactive routing compared

	AoR	DoRA	CO	RB	MR	S
Proactive	o	+	–	–	–	–
Reactive	o	–	+	+	+	o

+: efficient, –: non-efficient, o: context-dependent

Availability of Routes – The availability of routes is one of the major differences between the proactive and reactive routing: In proactive routing, the routes are always available, whereas in reactive routing they have to be determined when needed.

Delay of Route Acquisition – As all the routes are already known in proactive routing, the delay is usually low, while in reactive protocols, the delay when acquiring a new route might larger.

Control Overhead – As proactive networks maintain their routes proactively, their control overhead is considerably higher than reactive ones.

Required Bandwidth – A consequence of the large control overhead is that the required bandwidth for proactive protocols is higher than for reactive protocols.

Memory Requirements – That depends on the routes kept in the routing cache using a reactive protocol, but it is normally lower than in proactive protocols. Specifically, the DSDV protocol has to maintain two tables of routing information alone, while the DSR protocol only has one. It can be smaller than that, too, like in the reactive AODV [9] protocol, where it is not possible to have two routes to the same destination, also they expire after a certain amount of time and are discarded.

Scalability – Proactive routing protocols are not very suited for large networks, as every node needs to keep an entry for every node in the network in their tables. In larger networks, the overhead of the reactive DSR protocol can also increase significantly, as it keeps the route in the header of each of its packets. Protocols like AODV only have their target in the header of its packets, making the scalability to bigger networks much better for reactive protocols.

Yuxia Bai et. al. [5] have carried out simulations between the DSDV, FSR, AODV and DSR algorithms, that showed that both proactive and reactive protocols are very competitive. However, each protocol has its unique weakness. They looked at throughput, packet delivery ratio, and average end to end delay. Concerning throughput, the DSR algorithm was the worst performer, while, with increasing network size, it became the best performer concerning the packet delivery ratio. The AODV protocol has been proven to be the best choice for throughput and average end to end delay as the network size increases, while the perfomance of the other protocols worsened.

They concluded that, while in a small network the proactive algorithms are competitive, as the network grows larger, the reactive algorithms become the dominant players.

8. Conclusion and Future Work

In this paper, we provided an overview of the different *MANET* routing types and several routing protocols. A comparison between these two types has been drawn, outlining their strengths and weaknesses, features and characteristics.

It has been shown that, while there is no ‘perfect’ protocol for every circumstance, there are different protocols best suited for different requirements. However, it also became clear that reactive routing protocols are better suited for managing bigger *MANET*s.

In the future, *MANET*s will have an increasingly big influence on mobile computing. A possible next step is the integration of *MANET* as a common expansion of wireless networks and fixed network architectures, a move that could see benefits like less centralized routing, resulting in higher speed, more mobility, and cheaper maintenance costs.

References

[1] https://techterms.com/definition/adhocnetwork, [Online; accessed 15-March-2020].

[2] S. Corson and J. Macker, *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, January 1999.

[3] G. Jayakumar and G. Gopinath, “Ad hoc mobile wireless networks routing protocols - a review,” *Journal of Computer Science*, vol. 3, no. 8, pp. 574–582, 2007.

[4] A. P. Shima Mohseni, Rosilah Hassan and R. Razali, “Comparative review study of reactive and proactive routing protocols in MANETs,” 2010.

[5] Y. M. Yuxia Bai and N. Wang, “Performance comparison and evaluation of the proactive and reactive routing protocols for MANETs,” 2017.

[6] D. A. M. David B. Johnson and Y.-C. Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR),” 19th July 2004.

[7] C. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” pp. 234–242, 1994.

[8] IEEE Computer Society LAN MAN Standards Committee, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11,” 1997.

[9] E. B.-R. C. Perkins and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*, July 2003.