# mobile ad hoc networks

Dr. Michael Gerharz

*In 2006, 84 million WiFi-equipped laptops have been sold according to In-Stat. ABI Research forecasts that in 2010, 100 million WiFi equipped mobile phones will be sold. This means, virtually every public place you visit will most likely be crowded with WiFi-equipped devices.*

## What is a mobile ad hoc network?

A mobile ad hoc network is a spontaneous network of mobile and wireless devices to communicate without pre-installed infrastructure. For this purpose, each network node forwards data on behalf of other nodes, i.e. if two nodes want to commuicate although they are outside their mutual communication range, nodes in between forward the data on an end-to-end route. This is called multi-hop communication.

**a mobile ad hoc network is a highly dynamic network of mobile devices.**

Since each network node is potentially mobile, communication links may change frequently. Therefore, mobile ad hoc networks tend to be much more dynamic than wired networks.

**IEEE 802.11 networks are rather bandwidth restricted and use a shared medium.**

Two further challenges arise in comparison to wired networks, especially when looking at IEEE 802.11 networks. First, the wireless communication channel is usually much more bandwidth restricted than a wired channel. Second, the IEEE 802.11 wireless medium is a shared communication medium, the communication channel is blocked not only when transmitting but also when receiving - and even when a close-by node is transmitting to a third node.

## Ad hoc routing

The basic challenge in ad hoc networking (as in any other network) is routing, i.e. to establish end-to-end communication paths. From wired networks, basically two routing approaches are well known. However, both have their weaknesses in a mobile, wireless networking scenario.

**the counting-to-infinity problem is much more severe.**

Distance vector routing suffers from the counting-to-infinity problem. This problem becomes much more severe in highly dynamic networks as links break much more frequently. Given the large amount of time required to recover from the counting-to-infinity situation, the network is likely to be in inconsistent (i.e. looping) states a large fraction of the time.

**dynamic network topologies cause frequent link state updates.**

Link state routing is based on periodically flooding link state updates for every node. This is a huge burden on the network given the small bandwidth, the shared nature of the medium, and the high dynamics which may cause frequent link state updates to be issued.

## Flooding

**flooding may result in broadcast storms that overwhelm the medium's capacity.**

Flooding may lead to a phenomenon called „broadcast storm." A broadcast storm happens when the occupation of the wireless medium by all the transmissions of the flooding nodes overwhelms the medium's capacitiy. This leads to an increasing number of collisions and thus packet losses (and may also affect other active communication). Particularly, this may happen frequently in dense networks.

Nevertheless, flooding is the most important operation in mobile ad hoc networks and is required to explore the topology of the network and to discover end-to-end

communication paths. However, due to the severe restrictions of flooding regarding the capacity of the network, it is of utmost importance to use flooding seldom and in an efficient manner.

**do flooding seldom and efficiently.**

# Reactive routing

A valid question concerning the applicability of traditional routing paradigms in the ad hoc networking context is whether it is at all necessary to maintain routing entries for all possible destinations at any given time. In fact, due to the dynamic nature of the network, it is likely that most of the destinations are not used at all and that most of the routing entries have changed several times before they are needed.

**many routing entries may never be needed.**

This suggests a new routing paradigm, called „reactive routing“ or „on-demand routing“. Routing paths are only discovered when there is a demand for them, i.e. when a node actually requests a communication path to a particular destination. The challenge then is that the nodes may not have information readily available on how to route the packet to the destination. Thus, the route must be discovered prior to the actual data transfer.

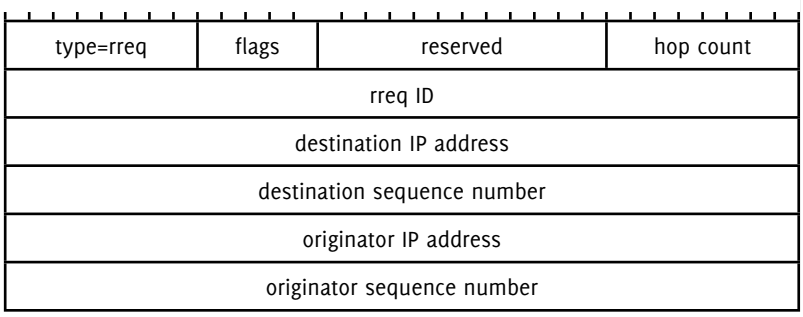
**a route is established only on demand.**

## Reactive route discovery

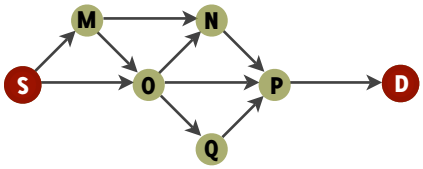
Basically, reactive route discovery means flooding the network with a route request (RREQ) message. Upon reception of this RREQ message, a node checks whether it knows the destination and if not it re-broadcasts the RREQ to its neighbourhood. Additionally, it records the backwards path to the originating node, i.e. the source node of this route discovery process. In detail, the route discovery process depends on the actual protocol implementation. For this lecture, we will concentrate on the Ad hoc On-Demand Distance Vector (AODV) routing protocol which has been published by the IETF as RFC 3561-

**to discover a route, the network is flooded with a route request**

The originating node assembles a route request message according to this message format. The RREQ message contains the IP addresses of both the originating and the destination node as well as sequence numbers for both nodes (which we will ignore for the moment). Furthermore, the message contains a RREQ-ID which is unique in conjunction with the originator IP address. Its purpose is to identify duplicates during the flooding procedure. Additionally, the message contains the hop count distance from the originating node and some flags which we will ignore.




Upon reception of a RREQ, a node updates its routing table. First, it updates the routing entry for the preceding node, i.e. the node which it has received the RREQ from. Second, it updates the routing entry towards the destination. If it doesn't already know a better route, it records the distance according to (hop count +1) and enters the preceding hop as the next hop towards the destination. Then, the node checks whether it knows a path to the destination. If not, it re-broadcasts the RREQ with an updated hop count field. Otherwise, if the node does know a path to the destination or is the destination itself, it assembles a „route reply“ message containing the destination IP address, the originator IP address, the hop count and some further information. This route reply is called an „early reply“ if the node itself is not the destination. Upon reception of a route reply, a node updates its routing entry towards the destination.




**a route reply is sent back in order to establish the route.**

Note that every RREQ is processed only once. Any duplicate reception caused by the flooding procedure will be silently discarded. In particular, this means that AODV will not optimise on the shortest path but on the path with the shortest RREQ propagation time. This path tends to have low congestion and a small end-to-end delay (although this is not guaranteed).

**AODV prefers paths with low congestion and a short delay.**

## Sequence numbers

Due to the dynamic nature of ad hoc networks, routing loops are an imminent danger. A routing loop occurs when a node promotes outdated routing information which cause the routed message to visit a node twice (and thus, infinitely). In AODV, this could happen when a node issues an early reply but has outdated routing information.

To prevent this, each RREQ is annotated with a sequence number which is increased for every route discovery. Only nodes that know a route to the destination which is more recent than the requested route may issue an early reply, i.e. the recorded sequence number for the requested destination must be higher than the sequence number in the RREQ.

**sequence numbers help to avoid routing loops.**

Note that sequence numbers are also a means to solve the counting-to-infinity problem because they enable the differentiation of old and new routing information. This solution was not known until the ad hoc routing protocol DSDV was developped in 1994. However, using sequence numbers for this purpose has also disadvantages which have to do with the order in which nodes update their distance vectors (details left to the reader).

**sequence numbers solve the counting-to-infinity problem.**

## What happens when a link breaks?

When two nodes move outside each others communication range or one of the nodes is switched off, their communication link breaks. A node does usually recognise this by missing acknowledgments after a timeout has occured. In this situation, each affected end-to-end path needs to be re-routed.

For this purpose, the node issues a route error message towards its upstream neighbours which it determines from its routing table. After reception of the route error message, the source node(s) may issue a new route discovery procdure to discover a fresh route.

**route error messages signal a link break and trigger a new route discovery.**

Note that the interruption due to a link break may be quite considerable. First, the link break needs to be discovered which may easily take a few ten or even hundreds of milliseconds until the last link layer re-transmission times out. Second, the route error message must be transmitted towards the source nodes, competing for medium access with other messages. Finally, the new route discovery takes time until the new route is established and ready to use.

**interruption may be considerable.**

## Drawbacks of reactive routing

While reducing the number of flooding procedures in many scenarios, reactive routing protocols have some drawbacks as well. First of all, the route setup time is large compared to classic routing protocols, which have routing entries readily available. Likewise, the interruptions after link breaks are larger. Furthermore, using more sophisticated routing metrics is far from trivial and requires tradeoffs to be made. Finally, once established, a route is (usually) used until it breaks although it may deteriorate during this time. Classic routing protocols might have learnt better routes in the meantime.

**reactive routing protocols suffer from large setup times**

**using sophisticated routing metrics is not trivial.**

# Proactive routing

In contrast to reactive routing protocols, classic Internet routing protocols maintain routes to all destinations all the time, i.e. proactively. The disadvantage in mobile ad hoc networks is the need to periodically flood the network. In order to maintain the advantage of readily available routing information, the flooding procedure of the link state updates has to be made more efficient.

**proactive ad hoc routing protocols need efficient flooding.**

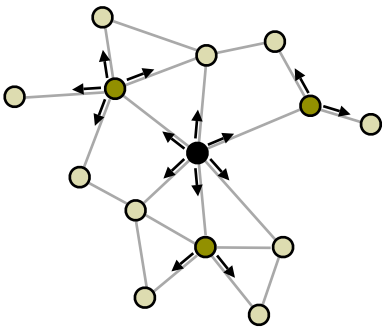
The best known proactive ad hoc routing protocol is the Optimised Link State Routing (OLSR) protocol. As the name suggests, it is a link state protocol with some optimisations regarding the flooding procedure, the most important of which is the concept of multipoint relays.

**OLSR ist the best known proactive routing protocol.**

## Multi point relays

The major problem of flooding in ad hoc networks is the shared wireless transmission medium. Each node is not only blocked when it is transmitting but also when any of its neighbouring nodes is transmitting, regardless of whether this transmission is intended for the first node or not. On the other hand, this means that unlike wired transmissions, each transmission is heard by any node in the transmitting node's communication range. This fact may be utilised to reduce the number of broadcast transmissions during a flooding procedure.

Consider the following network topology. To reach all of the black node's 2-hop neighbours, it is in fact unneccesarry that all of its 1-hop neighbours re-broadcast a link state update. Instead, it suffices that only three nodes re-broadcast the message. These three nodes are called multi point relays in OLSR. In general, a node is called a multi point relay if it belongs to the set of nodes that covers the complete 2-hop neighbourhood of a given node.



To find the minimum set of multi point relays means finding a minimal connected dominating set. Unfortunately, this problem is NP-complete. Therefore, OLSR (usually) discovers an approximate solution using this greedy algorithm:

**repeat**

*sort* neighbours by number of covered 2-hop-neighbours

*select* neighbour with largest coverage

*delete* covered nodes from 2-hop-neighbourhood

**until** all nodes are covered

**a set of multi point relays covers all 2-hop neighbours of a node.**

Note that each node computes its own set of relay nodes independent of other nodes. Thus, multi point relays are not computed network-wise but node-wise.

**each node computes its own set of multi point relays.**

## Neighbourhood discovery

A prerequisite to establishing a set of multi point relays is to gain an understanding of which nodes belong to a node's 2-hop neighbourhood. This challenge is usually solved using „hello" messages. In OLSR, this works as follows:

Each node periodically transmits a hello message. Upon the reception of a hello message, a node knows of the presence of the originating node. In particular, it knows that a unidirectional link is available from the originating node to itself. However, since IEEE 802.11 relies on acknowledgments, links need to be bi-directional. Therefore, each node includes in its hello message not only its own ID but also the IDs of all of its neighbouring nodes as well as the link status (unidirectional, bidirectional, broken). As soon as a node receives a hello message that includes its own ID, it knows that a bi-directional llink to the originiating node has been established. At the same time, it learns about its 2-hop neighbourhood.

**neighbours are discovered by periodical hello messages.**

**to discover bidirectional links, a hello contains all neighbour IDs.**

## Topology discovery

To be able to route messages using OLSR, a node needs to learn the network topology to a degree that allows the establishment of end-to-end paths. In OLSR, the multi point relays are responsible for the dissemination of this topology information.

In OLSR version 1, each multi point relay periodically floods through the network a link state update containing all its bidirectional links. Note that only multi point relays will forward this information. A node knows that it is a multi point relay from the hello messages. Each node includes the IDs of all its multi point relays in its hello messages.

**only multi point relays forward link state updates.**

In OLSR version 2, not only the multi point relays but every node periodically broadcasts a link state update. However, still only the multi point relays forward these messages.

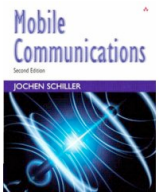
# Concluding Remarks

Mobile ad hoc networks have a highly dynamic topology and use a shared medium that is rather constrained in capacity. Therefore, they require optimised routing protocols to deal with these constraints. Reactive routing protocols such as AODV try to minimise the number of required flooding procedures by discovering routes only on demand. However, this leads to longer route setup times and has drawbacks concerning the routing metrics.

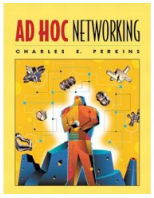
On the other hand, proactive link state routing protocols such as OLSR try to make the flooding procedure more efficient. For this purpose, multi point relays are elected to cover each node's 2-hop neighbourhood. Using multi point relays, the network may be covered with much less transmissions than with traditional flooding. However, still the network is flooded periodically which could be avoided using reactive routing protocols.

The optimal choice depends on several factors. These include, amongst others, the scenario, the actual traffic patterns, and reliability requirements.

# Literature




**Jochen Schiller**
**Mobile Communications**
Addison-Wesley




**Charles E. Perkins**
**Ad hoc Networking**
Addison-Wesley

RFC 3561 C. Perkins, E. Belding-Royer, S. Day – Ad hoc On-Demand Distance Vector (AODV) Routing, July 2003

RFC 3626 T. Clausen, P. Jaquet – Optimized Link State Routing Protocol (OLSR), October 2003

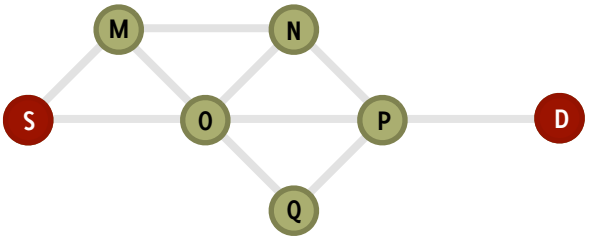
OLSRv2 – T. Clausen, C. Dearlove, P. Jaquet – The Optimized Link State Routing Protocol version 2, Internet Draft, expires Dec. 2008

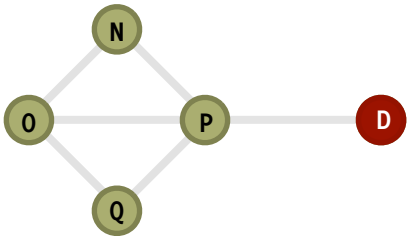
## Notes

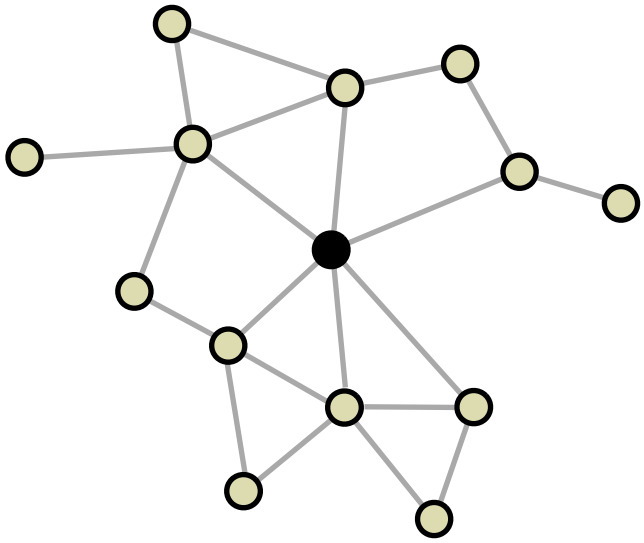
# mobile ad hoc networks



## Notes



## Notes