advanced topics in ad hoc networks

Dr. Michael Gerharz

While the basic routing challenge is solved using the reactive and proactive routing schemes seen in last weeks lecture, severe challenges remain concerning the quality of the communication. As it turns out, the paths selected by basic routing algorithms have major disadvantages.

Characteristics of ad hoc routes

OLSR is a link state routing protocol and as such tries to maintain a (rather) complete picture of the network topology. From the network topology, it computes end-to-end communication paths using a shortest-path-algorithm such as Dijkstra's algorithm. Note that in OLSR version 1 only the multi point relays forward data which means that the view of the network topology is only partial. However, it may be expected that large fractions of the topology are discovered leading to „quite" short routes. In OLSR version 2 every node issues link state updates, thus the complete network topology is discovered (in principle).

OLSR computes shortest paths.

AODV follows a different approach. It is based on flooding the network with a route request message. In order to save bandwidth, duplicates are completely discarded, i.e. each node processes a route request only once. Since the IEEE 802.11 medium uses a random backoff for medium access, it may happen that the first route request to reach a node did not travel on the shortest path.

AODV processes each route request only once.

This has both drawbacks and benefits. The obvious drawback is that end-to-end communication paths may require more than the minimum number of transmissions on the medium. On the other hand, in heavily used networks, one reason for a route request to be delayed is congestion at a node. Thus, if a node on the shortest path is congested, AODV avoids this path and routes the traffic along a less congested path. Thus, AODV prefers paths with low congestion and small end-to-end delay. However, note that in general this path tends to be rather close to the shortest path because each additional hop adds one more transmission competing for the medium using a random backoff procedure.

AODV avoids congested paths.

AODV prefers little congested, paths with small delays which tend to be rather short.

In conclusion, both OLSR and AODV (and also other ad hoc routing protocols) establish rather short if not the shortest path.

Drawbacks of short paths

By definition, a short path requires few hops to route a packet from source to destination. As the spatial distance between source and destination is given, this means that each single transmission, i.e. each single link on the route, bridges a rather large spatial distance.

short paths lead to long-distance links.

However, long-distance links have two major drawbacks which derive from the fact that the communicating nodes on a long-distance link are rather close to the edge of their respective communication range. As two nodes approach the edge of their communciation range, the signal quality degrades. This leads to an increasing number of packet losses and may lead IEEE 802.11 devices to switch to a lower data rate. As a consequence, the time to transmit a packet, may largely increase. As an example, a 1500 byte packet takes 0.3 ms to transmit using a 54MBit/s modulation while it takes 12.5 ms using a 1MBit/s modulation.

long-distance links have poor link quality.

A second disadvantage is due to the mobile nature of the ad hoc nodes. If one or both of two nodes which are close to the edge of their communication range moves, the link is at a high risk of breaking due to this movement.

long-distance links break easily even on small movements.

What happens when a link breaks?

When a link breaks, a route becomes unusable until a new route is discovered. This means, packet losses may occur. Given that it may take a considerable time to even detect a link break in IEEE 802.11 networks, a considerable interruption may result from the link break.

a link failure leads to interruptions and packet losses.

As a result of the link failure, a new route needs to be discovered. In both, reactive or proactive routing this leads to flooding the network. In the proactive case, a link state update reflecting the new network topology needs to be issued. In the reactive case, a route re-discovery needs to be performed. This causes additional overhead in the network. If link breaks occur frequently, as is suggested by long-distance links, these additional flooding procedures may use up a large portion of the network's resources.

a link failure implies a flooding procedure which causes considerable overhead.

Advanced link metrics

Several approaches towards improving the link failure rate exist. A major reduction of the flooding procedures and the interruptions may be achieved when selecting links based on their anticipated availability. To determine this availability is far from trivial and beyond the scope of this lecture.

selecting links based on their anticipated lifetime may reduce link failures.

Likewise, links may be selected based on their link quality. However, this requires information from the link layer which is usually unavailable on IP layer. Several approaches exist to make this information available, however the details are beyond the scope of this lecture.

A third link metric is the „airtime-metric" which estimates the fraction of time that a transmission occupies the wireless medium. To calculate the airtime, probe packets of a given size are exchanged periodically. The time of the transmission is measured and used to calculate the airtime (see below). An advantage of this approach is that it is applicable on the IP layer. However, it also has two major disadvantages. First, the probe packets issue an additional load on the network. Second, the airtime calculated from the probe packets may be unrepresentative of the actual data transfer that follows because it may use different (and varying) packet sizes and the medium condition may have changed in the meantime.

airtime estimates the fraction of time the medium is occupied.

airtime may be difficult to measure and it causes additional overhead.

On the link layer, a more detailed calculation of the airtime is possible:

$$c_a = \left[O_{ca} + O_p + \frac{B_t}{r} \right] \frac{1}{1 - e_{pt}}$$

where:

- c_a is the airtime
- O_{ca} is the overhead of accessing the channel (i.e. random backoff and contention)
- O_p is the procol overhead (i.e. headers)
- B_t is the size of the probe packet, r is its data rate, i.e. B_t/r is the time to transmit the probe payload
- e_{pt} is the error probability

Advanced ad hoc routing

Having a sophisticated link quality metric, the question remains how ad hoc routing protocols may utilise these metrics to discover better end-to-end paths.

For proactive routing protocls like OLSR, the answer is straightforward. Each node (or multi point relay in the case of OLSR version 1) includes the link quality metric for each of it's links in the link state updates. This way, every network node learns the link metrics and may calculate best paths using a routing algorithm liks Dijkstra's shortest path algorithm.

proactive protocols include link quality metrics in their link state updates.

For reactive routing, the challenge is much harder.

Advanced reactive routing

As mentioned above, AODV and similar reactive routing protocols process each route request only once. As a consequence, whichever link is visited first, gets selected. Other links that might have higher quality values, are simply ignored. The solution is to not ignore duplicate route requests in order to learn about better paths. Basically, two approaches exist which both have their advantages and drawbacks.

AODV selects the route with the quickest route request.

One solution is to forward a route request more than once. In more detail, if a node has forwarded a route request but afterwards receives a further route request which has travelled along a path with a better quality, it will also forward this route request. This way, every possible path through the netowrk is explored so that the optimal path may be discovered. However, recall that a single flooding procedure may cause severe damage to the network's capacity (broadcast storm). Forwarding multiple route requests does in essence mean to flood the network multiple times for a single route discovery (at least possibly). Therefore, the application of this approach has to be thoroughly checked against the properties of the scenario and is applicable only in small, rather sparse networks.

forwarding a route request multiple times causes huge overhead.

The second solution makes use of the fact that, in effect, the quickest route request determines the selected route.

Delayed routing

A second solution to discover high-quality paths with reactive routing protocols is to make the best path the quickest during a route request procedure. Since the order in which messages are sent on the IEEE 802.11 medium may not be altered, the solution is to delay paths over low quality links in such a way that higher quality routes may surpass.

by delaying „bad" route requests, the best route is made the quickest.

Formally, upon reception of a route request, a node calculates the quality of a path by accumulating the quality of the path carried in the route request and the quality of the last hop. Based on this overall path quality a delay time is calculated. If a better route request arrives before the delay time elapses, the delay is adjusted to a new (and earlier) timeout. If a worse route request arrives, it is silently discarded.

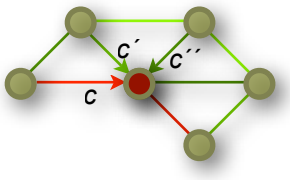
schedule *RREQ* at time *Delay(c)*

IF better *RREQ* arrives before *Delay(c)* has elapsed:

    re-schedule *RREQ* to the earlier time *Delay(c´)*

IF worse *RREQ* arrives at any time:

    silently **discard**



Basically, this algorithm is a distributed implementation of Dijkstra's shortest path algorithm. Dijkstra's algorithm is a greedy algorithm which processes the best currently known route request as the next step. This corresponds to the next timeout which expires. By formally showing the equivalence of these two procedures (beyond the scope of this lecture), the validity of the delayed route request procedure may be shown.

the delaying algorithm is basically a distributed implementation of Dijkstra's algorithm.

While this delayed route discovery procedure doesn't introduce any further overhead in the network, the route setup time is further increased. For many scenarios, this further delay is quite negligible. However, there are scenarios where the additional delay is quite large.

the route setup time is further increased.

IEEE 802.11s – wireless mesh networks

A special kind of ad hoc networks are „wireless mesh networks" which have some specific characteristics that differ from mobile ad hoc networks. The term is not generally agreed upon, but it usually means wireless ad hoc networks with static (or mostly static) nodes. The wireless mesh network usually forms an access network that enables other mobile nodes to access the Internet. Wireless mesh networks are used for public service or for commercial access to IEEE 802.11 hotspots, e.g. to form a city-wide public hotspot. Quality-of-Service and security demands are rather high.

wireless mesh networks are static ad hoc networks.

The reason to use wireless mesh networks for this type of application is that only few connections to the Internet are required while most of the access points are connected using ad hoc routing protocols. This type of application implies a tree hierarchy in the network.

IEEE 802.11s in an amendment to the IEEE 802.11 standard which defines ad hoc protocols for these types of network.

Routing in IEEE 802.11s

Routing in IEEE 802.11s follows a very open architecture. Every manufacturer may implement its own routing protocol. However, there is one mandatory routing protocol, the „hybrid wireless mesh protocol" (HWMP). HWMP is designed to use vendor specific routing metrics, but the airtime metric (see above) is mandatory.

Basically, HWMP is AODV with some proactive enhancements. The basic route discovery procedure follows AODV's route request procedure. However, each route request carries the routing metric (airtime by default). To discover the best path, route request are forwarded multiple times when a better route request arrives later. This is justified by a targeted network size of 25-30 nodes, i.e. a rather small size.

basically, HWMP is AODV with proactive enhancements.

Additionally, HWMP includes a proactive mode which proactively maintains a route to a root node (the Internet connection point). This root node periodically issues a „proactive route request" which is a route request to the broadcast address. Each node that receives this message updates its routing entry towards the root. Depending on the setting of the „proactive route reply" bit, the node may (bit=0) or shall (bit=1) send a route reply back to the root.

HWMP's proactive part utilises the tree nature of the network.

If in this scenario, a node needs to establish a route to another node, it may send the data directly to the root node instead of issuing a route request procedure. Three cases may occur. First, the node is located outside of the local network in which case the root node forwards the data. Second, the node is located in the local network, but the root does not know a route in which case it initiates a route request. Third, the root node knows a route in which case it forwards the data to the destination. The destination may then issue a route request for the source node.

Congestion Control

To avoid an exhaustion of the medium's capacity, IEEE 802.11s contains a protocol for congestion control. Each node may monitor the channel activity in its local neighbourhood. When it detects congestion, it may use the following procedure to signal this situation.

congestion control may avoid a capacity exhaustion.

It may issue either a unicast congestion control request to a specific neighbour or a broadcast neighbourhood congestion announcement to all of its neighbours. Either message informs the recipient of the congestion situation and contains a request to limit the transmission data rate. A node receiving such a request shall limit its data rate and inform the issuing node about its future offered load using a unicast congestion control response message.

Obviously, this procedure is targeted towards a network that is in control of a single entity. Since, the frequency bands used by IEEE 802.11 are unlicensed and everybody is free to use those frequencies, hard guarantees cannot be given using this procedure.

Since the frequency band is unlicensed, this may not always work.

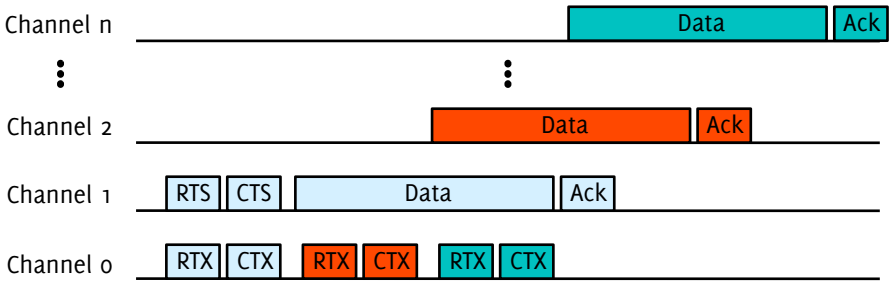
Channel Selection

Each IEEE 802.11 channel uses only a fraction of the overall bandwidth available in the 2.4 GHz or 5GHz unlicensed frequency band. The capacity of a wireless mesh network could be greatly enhanced if it was allowed to used multiple channels in the available frequency bands. The challenge is to agree upon a common channel for each transmission.

using multiple channels, the network capacity may be increased.

For this purpose, IEEE 802.11s introduces the Common Channel Framework. It works similar to the RTS-CTS procedure. In addition to the RTS-CTS parameters it is used to negotiate a channel for the following data exchange. For this purpose, the sender sends an RTX message to propose a channel for the following data transmission which the receiver must acknowledge using a CTX message. It is important that any RTX-CTX exchange is performed on a common control channel and that all nodes obey to the transmission schedule negotiated on this control channel.

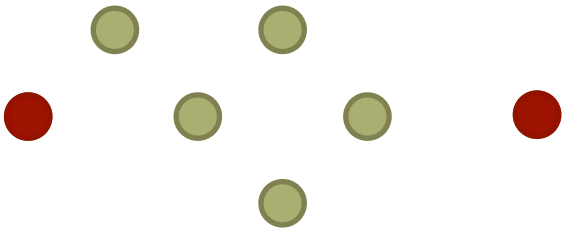
an RTX-CTX-exchange determines the transmission channel.

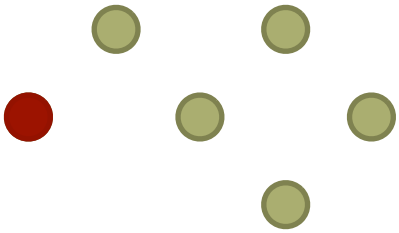


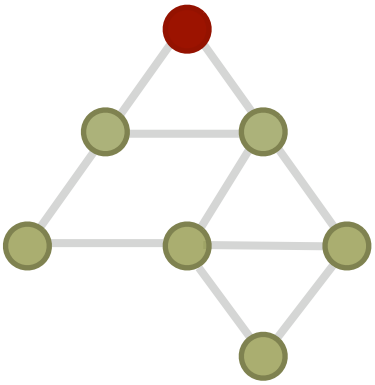
Notes

advanced topics in ad hoc networks

Notes







Notes