Scalability of routing protocols in wireless ad-hoc networks

Sjoerd Langkemper

February 17, 2006

Abstract

With wireless devices increasing in popularity and ad-hoc wireless networks getting larger, scalable routing protocols are needed. There are already many routing protocols for ad-hoc wireless networks, but very few of them were conceived with scalability in mind. This paper will provide an overview of various routing techniques and consider their scalability. We will show that only the HSLS protocol is scalable without imposing trivial restrictions on the nodes.

1 Introduction

Wireless networks are increasingly popular because of technological improvements. As wireless devices become more portable, wireless networks may be set up in places where it is not suitable to set up an infrastructure. Wireless networks without infrastructure are called ad-hoc networks. Uses include disaster recovery, law enforcement, military and collaborative.

Routing protocols for wired networks do not necessarily work in ad-hoc wireless networks, because of rapidly changing topology, unreliable links and lack of structure.

Many routing protocols for ad-hoc wireless networks exist, where nodes forward traffic for each other. However, few of them have been conceived with a large network in mind. Therefore, we will look into the scalability of ad-hoc wireless networks.

Instead of examining all protocols one-by-one, we will look into protocol classes and explain a few protocols in detail. We will show that only Hazy Sighted Link State protocol is simple to set up and scales well.

In section 2, we will discuss proactive routing protocols, which keep active routing information. In section 3 on-demand protocols are described, which only seek a routing path when needed. Hierarchical routing protocols, which divide the network in clusters, are discussed in section 4. Geographical protocols are reviewed in section 5. Finally, section 6 concludes.

2 Proactive routing protocols

Proactive routing protocols continuously try to maintain up-to-date routing information on every node in the network. This has as advantage that connection times are fast, because routing information is already available when the first packet is sent. A disadvantage of proactive protocols is that they continuously use resources to communicate routing information, even when there is no traffic.

2.1 Link State

In the link-state routing protocol (commonly abbreviated as LS), each forwarding node floods the network with information about its neighbours. Every other router updates its router table with this information and thus has a full overview of the network. The advantage of LS is that every router knows the shortest path to every destination, making forwarding packets easy.

Link-state works very well in static, wired networks. In a large wireless network, In large wireless networks, however, radio links are unreliable, nodes move and devices are shutdown. In such a situation, flooding the network every time the network topology changes is not practical. Providing every router with an up-to-date routing table is impossible, since the topology may change again before the message communicating the old change has reached every router.

2.2 Distributed Bellman-Ford

In distributed Bellman-Ford, on which many routing protocols are based, each node sends its neighbours a routing table periodically. In this routing table is information on the distance to each node in the network.

When a node receives routing tables from its neighbours, it updates its own routing table by using the shortest path to each destination. If node A advertises a distance of 30 to node C, while node B advertises a distance of 20 to node C, packets to node C will be routed through B.

This algorithm suffers from the *count-to-infinity* problem, which means that if a node goes down, it takes very long before this information spreads through the network. As seen in fig. 1, each node updates its information based on old information from other nodes, making the path longer on each update. This property makes it unsuitable for ad-hoc networks, where nodes may fail frequently. [Tan03] [Ace94]

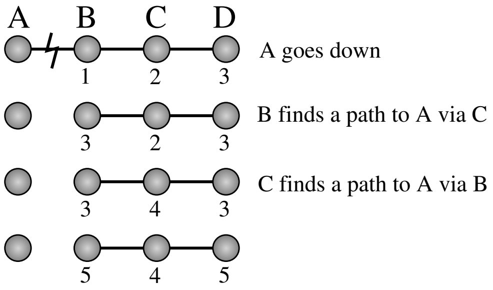

Figure 1: Count-to-infinity problem

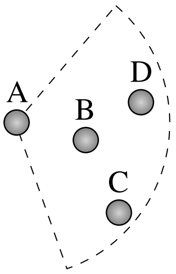

Figure 2: Node B does not have to forward messages

2.3 Optimized Link State Routing

OSLR ([RFC3626], is very similar to link-state, with a few optimizations. Each node selects some nodes in its neighbourhood which are called multipoint relays (MPRs). Only these nodes will forward messages for the node. Update messages are flooded over the entire network just as in link-state, but now only the MPRs relay the messages, decreasing the overal bandwidth. A node chooses its MPR in such a way that it can reach the whole network.

[Che01] also proposes to make some nodes play a passive role, with a protocol named Span. Although Span has the goal of limiting power usage, OSLR wants to decrease the overhead. As can be seen in fig. 2, node B does not have to relay messages. If it does anyway, it will interfere with other relaying nodes. Selecting active and passive nodes can decrease resource usage in any protocol.

OSLS, however, still floods the network and this incurs a large overhead. Moreover, in sparse networks many nodes will be selected as MPR, making the optimization technique useless.

2.4 Global State Routing

Because of the advantages of the link-state routing protocol, research has been done to make the link-state protocol more scalable. This is typically done by not flooding the network with routing information, but sending only a part or sending it only to some nodes.

[Ger98] describes a protocol (GSR) which only exchanges link-state information with neighbours, instead of flooding it over the network. Each node has a table of all nodes in the network. Periodically, each node sends its neighbours this list so that they can update their network topology information.

2.5 Fisheye State Routing

Fisheye routing is based on the assumption that routes to nodes which are far away do not have to be precise. The routing table is accurate for nodes close by, but approximate for nodes far away. When a packet is sent to a node, the route will become more precise when the packet closes in on the destination.

Fisheye state routing ([Iwa99]) is based on GSR and also only sends packets to neighbours. It works by not sending all available link-state information, but focussing on the network topology in the direct neighbourhood of the node. Changes in the network which occur close by are send at a high frequency. Distant changes are send in a lower frequency. Not knowing how to reach a distant destination exactly is not a problem, since the route becomes more accurate as the packet gets closer to its destination.

Since both GSR and FSR only transmit information to their direct neighbours, it may take a long time before a change in topology has been communicated through the network.

2.6 Hazy Sighted Link State

Hazy sighted link state (HSLS), as proposed in [San03], limits the spreading of topology information in space and time. It assumes that changes in the network close by are more important than changes far away, as in GSR. Instead of only sending updates to its neighbours, HSLS uses the a time-to-live (TTL) field to limit to which nodes the information spreads.

For example, each node within one hop is sent a Link State Update (LSU) packet each second. Nodes two hops away are sent a LSU each two seconds, etc. Using this scheme, a node will receive LSU packets each second from nodes within its range and it will recieve update packets each eight seconds from nodes which are eight hops away. This makes sure that a node has precise information on nodes close by, with the minimal overhead.

The optimal parameters for the timing and TTL can be calculated from variables such as mobility, size of the network and traffic. This makes it possible to optimize the protocol for each network, incurring the lowest overhead possible.

HSLS communicates routing information relatively fast throughout the network, without using flooding, which makes it scalable. Furthermore, being a proactive protocol, it does not have a delay when setting up a connection.

3 On-demand routing protocols

On-demand routing protocols do not try to keep their routing tables up-to-date. Instead, a node tries to find a route only when it wants to send a packet. This reduces the traffic needed for routing, but introduces a delay when the first packet is sent to a host.

3.1 Ad-hoc On-demand Distance Vector protocol

AODV, as proposed in [RFC3561] floods route request packets throughout the network to discover a certain host. Although all hosts in the network cache the information in all routing packets, flooding the network to find a host is a resource-intensive process. When the route to a non-existing host is requested, all nodes in the network get a route request packet.

AODV may be useful in networks where there is little traffic, or where connections are relatively static. In a large network where the topology and the connections between hosts change often, AODV induces too much overhead and a long delay before the connection is established.

3.2 Dynamic Source Routing

DSR ([Joh04]) uses source routing, which means that a complete, ordered route is carried in each packet. This makes it easy to control the route from the source node and guarantees loop-free paths.

A node can discover a path to another node by sending a route discovery packet. The node locally broadcasts this packet. Each neighbour appends its own address to a list in the packet and broadcasts it. When it reaches the destination node, a complete path from the source to the destination is available in the packet. The destination node returns this route discovery packet by sending it over the reverse path.

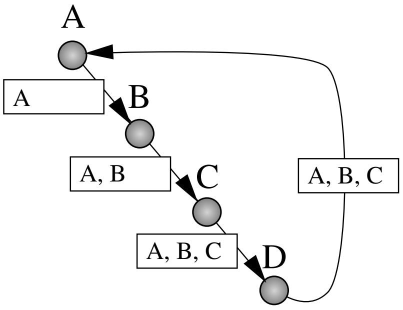

Figure 3: Route discovery in DSR

At this point, a path between sender and reciever is established. DSR proposes that all nodes cache information in route discovery packets, so that a route discovery is not always needed.

When a part of the link goes down, the node before the broken link informs each node which tries to route packets over the broken link. For example, if the link between C and D in fig. 3 goes down, C will send a route error packet to A. Because B overhears this packet, it will also remove the path through C from its cache.

In the [Joh04], the network size is assumed to be no larger than 200 nodes, with a network diameter of 5 to 10. Bigger networks are not feasible, because flooding is used to discover routes. When network topology changes and new connections are made, the network may collapse under the load of the routing packets.

4 Clustering/hierarchical routing protocols

A large network can be clustered so that it contains multiple sections or zones. Traffic between clusters is routed by clusterheads. This has as advantage that the routing protocol does not have to deal with all nodes, just the clusterheads. In large networks, superclusters

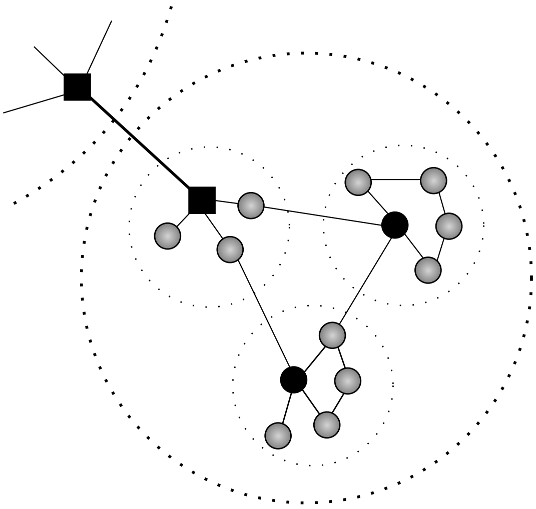

Figure 4: Hierarchical routing

can be made.

Packets travel from cluster to supercluster and down again, as in a tree. Therefore, the routing protocol used is commonly named hierarchical routing.

When clusters are in place, routing is efficient and scalable because routing information is only spread within the cluster, reducing overhead. However, creating clusters with the right properties is a problem of its own.

Selecting a clusterhead within a cluster is usually done by making the nodes with the lowest MAC address a clusterhead. This algorithm does not take into account properties which may affect the quality of the clusterhead, such as speed, uptime and battery power remaining.

Another proposed method for selecting a clusterhead is selecting the node which has the most connections, but this proved to create a bottleneck at the clusterhead.

Because there is not a clear way to cluster nodes or select a clusterhead, these protocols are not ready for use in a large network.

4.1 Hierarchical State Routing

HSR, proposed in [Iwa99], is a typical example of a hierarchical routing protocol. HSR maintains a hierarchical topology, where elected clusterheads at the lowest level become members of the next higher level. On the higher level, superclusters are formed, and so on. Nodes which want to communicate to a node outside of their cluster ask their clusterhead to forward their packet to the next level, until a clusterhead of the other node is in the same cluster. The packet then travels down to the destination node.

Furthermore, HSR proposes to cluster nodes in a logical way instead of in a geological way: members of the same company or in the same battlegroup are clustered together, assuming they will communicate much within the logical cluster.

HSR does not specify how a cluster is to be formed.

4.2 Cluster Based Routing Protocol

CBRP, which is described in [Jia02], uses source routing in a clustered network. Nodes periodically broadcast HELLO messages. When a node enters the network, it waits until it receives a HELLO message from a clusterhead. When it does not receive such a message in a specified time, it becomes a clusterhead itself.

Members of a cluster are always within reach of exactly one clusterhead. When two clusterhead can connect to each other, one of them becomes a member of the other cluster.

When a member of cluster A receives an HELLO message from a member in cluster B, it reports to its clusterhead that cluster A neighbours cluster B, and that this member can be used as a gateway to cluster B.

When a node wants to know the route to another node, it broadcasts a route request message. Although this message is flooded over the entire network, only the clusterheads will handle route requests. When a member receives a route request, it directly forwards it to its clusterhead. The clusterhead appends its own address to the route request message and forwards it to the neighbouring clusters. When the route request message ends up at the destination, it contains a list of clusterheads and gateways which form a route from the source to the destination, much like in DSR.

CBRP is based on simple flooding. However, because only the clusterheads have to handle messages, overall overhead is reduced. CBRP will particularly perform well in small, highly connected networks: when a cluster has many members, the clusterhead does the work for all these members. In big, sparse networks, flooding the route requests may be a bad approach. Another disadvantage is that that the load is poorly balanced: the clusterhead has to do most of the work.

5 Geographical routing protocols

Some routing protocols make use of geographical information, such as GPS coordinates. Typically, nodes communicate their location through the network, so that other nodes can determine the shortest path. Using geographical information makes it possible to select the truly shortest path. Furthermore, one only needs an (geographical) address to contact a host.

The disadvantage is, obviously, that each node has to know its location. In some environments, a GPS receiver can be used to obtain the location. However, GPS receivers are expensive and consume relatively much power.

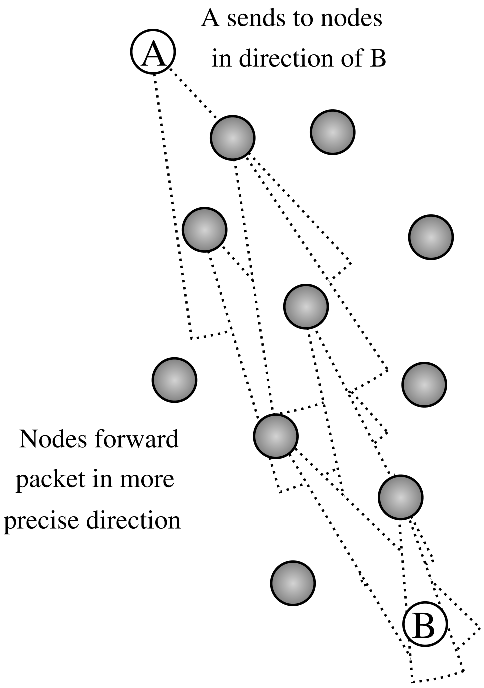

Figure 5: Geographical routing

5.1 DREAM

The DREAM protocol was proposed in [Bas98]. It assumes that each node knows its own location. Each node then communicates its address and location through the network. When a packet is sent, it is sent to the *direction* of the receiving node.

DREAM makes use of what is called the *distance effect*: the greater the distance separating two nodes, the slower they appear to be moving with respect to each other. Neighbours close by are frequently informed of the location of a node; nodes which are farther away only occasionally receive this information. The further a packet travels to its destination, the more detailed the position of the destination becomes.

Furthermore, DREAM takes into account the *mobility* of the nodes. When a node travels fast, it frequently sends its location informa-

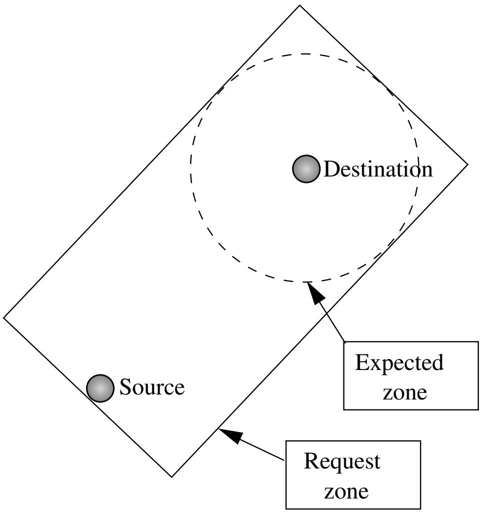

Figure 6: Location Aided Routing

tion to its neighbours.

DREAM is similar to HSLS in that the update frequency is dependant on the distance to a node, only it uses coordinates instead of routing paths. This makes it very well scalable, but introduces the problem of determining the location for each node.

5.2 Location Aided Routing

The Location Aided Routing Protocol (LAR, [Ko00]) only uses geographical information for route discovery. It is based on an on-demand protocol, like AODV. If the sender knows where the destination was at some time and it knows its speed, it can determine in which area the destination is now. This area is called the expected zone. LAR uses this to limit the flooding of route discovery packets. Packets are flooded within an defined area containing both the source and the expected zone. When a node outside this area recieves a packet, it ignores it. When the destination receives the route discovery packet, it returns it with its current location and speed, which can assist in future route discoveries.

When a node enters the network, it has no information about the geographical position of other nodes. LAR will then fall back to the underlying protocol, which floods the route discovery packet. For LAR to be an improvement over flooding, the network has to be stable. LAR will perform well for moving nodes and disappearing nodes, but not when a lot of new nodes are added to the network. Furthermore, connections in the network has to be stable. If nodes connect to many other nodes for a short time, location information will not be available or accurate enough to make use of LAR.

6 Conclusion

Wireless ad-hoc networks are expected to grow larger and a routing protocol which scales accordingly is needed. Proactive protocols are most suited for this, because they have total information on the network topology. Hazy sighted link state is the most efficient in this class of protocols and therefore the most scalable.

Ad-hoc protocols are only scalable when the topology and connections between nodes are fairly static. In other cases, they flood the network and incur a delay on setting up a connection.

Hierarchical protocols are scalable, because they split a large network into smaller clusters. However, constructing clusters with the right properties and selecting a clusterhead in each clusters are problems which are not yet solved.

Geographical routing protocols are probably the most scalable. However, each node has to know its location, which may not be possible in all situations.

Which routing protocol to use depends on many properties: the reliability and bandwidth of the links between nodes, the stability of the network topology, whether there are popular nodes or connections are random. In most cases, however, HSLS is the most scalable routing protocol. In situations where a positioning system is already in place, a geographic routing protocol is probably the best solution.

7 References

References

[Ace94] *An Efficient Routing Protocol for Wireless Networks* – Murthy & Garcia-Luna-Aceves, 1994

[Bas98] *A Distance Routing Effect Algorithm for Mobility (DREAM)* – Basagni, Chlamtac, Woodward & Syrotiuk, 1998

[Ger98] *Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks* – Chen & Gerla, 1998

[Iwa99] *Scalable Routing Strategies for Ad Hoc Wireless Networks* – Iwata, Chiang, Pei, Gerla & Chen, 1999

[Ko00] *Location-Aided Routing (LAR) in mobile ad hoc networks* – Ko & Vaidya, 2000

[Che01] *Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks* – Chen, Jamieson, Balakrishan & Morris, 2001

[Haa02] *The Zone Routing Protocol (ZRP) for Ad Hoc Networks* – Haas, Pearlman & Samar, 2002

[Jia02] *Cluster Based Routing Protocol (CBRP)* – Jiang, Li & Tay, 2002. `http://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01.txt`

[RFC3561] *Ad hoc On-Demand Distance Vector (AODV) Routing* – Perkins, Belding-Royer & Das, 2003. `http://www.ietf.org/rfc/rfc3561.txt`.

[San03] *Hazy Sighted Link State (HSLS) Routing: A Scalable Link State Algorithm* – Santiváñez & Ramanathan, 2003

[Tan03] *Computer Networks* – Tanenbaum, 2003

[RFC3626] *Optimized Link State Routing Protocol (OLSR)* – Clausen & Jacquet, 2003. `http://www.ietf.org/rfc/rfc3626.txt`.

[Joh04] *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)* – Johnson, Maltz & Hu, 2004

[Wiki06] *Wikipedia: Ad Hoc Protocol List* – Anonymous, 2006. `http://en.wikipedia.org/wiki/Ad_hoc_protocol_list`.