Fisheye State Routing

Fisheye State Routing (FSR) is a proposal for an implicit hierarchical routing protocol targeted to ad hoc networks.[1] The basic principles of FSR are shared with other proactive, link-state routing protocols. In proactive link-state protocols every network node constantly updates a topology map that makes it possible to compute the shortest path (and thus the next hop) to any destination in the network. The originality of FSR is inspired by the "fisheye" technique to reduce the size of information required to represent graphical data: The eye of a fish captures with high detail the pixels near the focal point, while the detail decreases as the distance from the focal point increases.

In routing, the fisheye approach translates into maintaining an updated information set about distance and path quality information for the immediate neighborhood of a node, against a progressively less updated information as the distance increases. Fisheye represents a valid trade-off between the accuracy of the routing function and the overhead due to the generation of control messages by the routing protocol.

FSR was never released to the public as a stand-alone routing protocol, and its specification was never finalized.[2] The base principle was included in the widely used OLSRd daemon (an open source implementation of the OLSR routing protocol[3]).

Protocol Working Principle

FSR is a link-state routing protocol, thus it is made of three tasks:

1. Neighbor Discovery: every node sends an HELLO message every δ seconds to its one-hop neighbours, in order to establish and maintain neighbour relationships.
2. Information Dissemination: every node disseminates Link State Announcements messages (LSA) every Δ seconds (with Δ > δ), that contain neighbour link information, to all other nodes in the network.
3. Route Computation: from the information contained in the LSA messages the node can reconstruct the whole network topology and use Djiksta's algorithm to compute the routes to any node in the network.

The peculiarity of FSR is that LSA messages are generated every Δ seconds using a sequence of distinct Time-To-Live values. Take as an example the sequence 1, 3, 8, 64, the 1-hop neighbours receive the LSA every Δs, so they have the most updated information. 2-hop neighbours receive the LSA with TTL 3, 8, 24. Nodes at a distance from 4 to 8 hops receive only the LSA with TTL 8 and 64. All the others receive only the LSA with TTL 64. As a consequence every node has progressively less updated information on the network topology as the distance increases.

The protocol exploits the fact that when a packet moves from a source to a destination, the nodes encountered on the shortest path have an increasingly precise topology information about the topological position of the destination (as their distance to the destination decreases), so the loss of accuracy in the shortest path computation from the source node is compensated along the path to the destination.

FSR thus decreases the overall quantity of information spread in the network, since LSA are not sent with a fixed maximum TTL.

Drawbacks

One of the typical issues with link-state protocols is that when a node or link breaks, temporary loops can be created. This is due to the fact that HELLO messages are sent with a higher frequency than LSA messages, so if a node fails, its neighbors sense the broken link way before the other nodes. They immediately recompute their routing tables, which can conflict with the routing table of the other nodes, and a loop can be created. This can happen when two nodes have information with a different age and thus they compute their routing tables on two different network topologies. FSR does this by design, it introduces areas in the network with potentially different information sets, so it increases the probability of creating temporary loops.[4]

References

1. http://nrlweb.cs.ucla.edu/publication/download/203/05_75_fisheye-state-routing-in.pdf
2. http://tools.ietf.org/html/draft-ietf-manet-fsr-03
3. https://github.com/OLSR/olsrd/blob/master/unmaintained/README-Link-Quality-Fish-Eye.txt
4. Yasir Faheem, Jean Louis Rougier: Loop avoidance for Fish-Eye OLSR in sparse wireless mesh networks (https://hal.inria.fr/file/index/docid/496803/filename/Loop_Avoidance_for_Fish-Eye_OLSR.pdf)