

Phien Nguyen

Christian Doud

1. It has to be consistent with how our graphs or topography is set up. There might be a point where data only flows in one direction rather than bi-directional. Some algorithms also find the direction from one point to another exclusively and need to find the point to the previous point using the same algorithm. If we use a directed link from AB but not from BA, then B has to find a minimum to A using the same algorithm that weighs in directed path.
2. Our routing algorithm produce symmetric routes because each node updates each neighbor node of its change in table and only changes based on the changes from the neighbor rather than outside its neighbor list. Changing outside can introduce more divergent as that path can also have the same amount of hops as the same route from the neighbor node.
3. If it advertised as having neighbors but never forwarding packets, then I must account what nodes can read that node. Then I will introduced dijkstra saying that node is the neighbor of these node rather than that node introduce itself. Therefore it still is part of the routing table without broadcasting its neighbor.
4. If the link state packets are lost then it would simply send different packets with different sequence which would be used instead although that packet has been dropped. If the packets are corrupted, then the node would send an error request to resend the packet to the same destination therefore trying to prevent data loss.
5. If it advertise now and then while simply dropping itself, then I must implement the other nodes to account that node as its neighbor while generating a "pseudo" table for that node to guess its neighbor and generate a routing table based on that table.