TDTS04 Lab 4

Distance vector routing

hugho678 & viles222

(i) Distance vector routing is a distributed algorithm where each router maintains a table of the shortest distance (most cost-effective) to every other router in the network
The routers routing table updates periodically by sending and receiving routing tables from its neighboring routers, this continues until every router has a consistent and stable routing table. The routing table is updated when a shorter path between routers is calculated.

(ii) By running the program and comparing the output with the expected result we could determine if the algorithm was running properly and make changes accordingly, this was done with and then without poison reverse implemented. The testing suggested that the algorithm was working as it should.

(iii) The poisoned reverse technique can still fail if an update is lost or delayed (Balakrishnan & Verghese, 2012). In such cases, parts of the network may continue to assume that a specific route is valid, potentially leading to routing loops and triggering the count-to-infinity problem. This issue is particularly evident in networks where multiple nodes form a cycle. If a node loses its direct connection to a destination, it will advertise an infinite cost to its neighbors. However if one of those neighbors had previously used the affected node as part of its route to the destination, it may still advertise an outdated cost back, creating a loop. While poisoned reverse prevents incorrect updates in some directions, it does not eliminate them entirely, allowing the count-to-infinity problem to persist in some cases.

(iv) To solve routing problems like loops and the count-to-infinity issue we can use path-vector routing and link-state routing.
Path-vector routing works by instead of just telling neighbors how far a destination is, it shares the entire route a packet will take. If a router sees its own name in the route, it knows there is a loop and won't use that path. (Balakrishnan & Verghese, 2012).
In link-state routing every router keeps a complete map of the network instead of just relying on its neighbors. This helps routers find the best path on their own, avoiding bad routes. It's smarter but requires more memory and processing power to work properly.
(Kurose & Ross, 2016).
Both methods improve network routing, but they have trade-offs—path-vector routing prevents loops without using too much memory, while link-state routing is more accurate but needs more resources.

References

Balakrishnan, H., & Verghese, G. (2012). *CHAPTER 18 Network Routing - II Routing Around Failures* [lecture notes]. Massachusetts Institute of Technology. [https://ocw.mit.edu/courses/6-02-introduction-to-eecs-ii-digital-communication-systems-fall-2012/resources/mit6_02f12_chap18/] [Retrieved: 31/3-2025]

Kurose, J. F., & Ross, K. W. (2016). *Computer networking: A top-down approach* (7th ed.). Pearson.[https://www.ucg.ac.me/skladiste/blog_44233/objava_64433/fajlovi/Computer%20Networking%20_%20A%20Top%20Down%20Approach,%207th,%20converted.pdf]