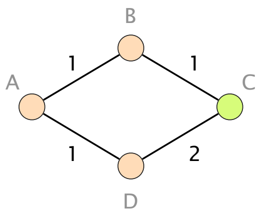
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prof. Laurent Vanbever
Networked Systems Group

Communication Networks

Solution: Exercises week 7 – Internet Routing

Convergence (Exam Style Question)



Loopy or not?

Consider this simple network running OSPF as link-state routing protocol. Each link is associated with a weight that represents the cost of using it to forward packets. Link weights are bi-directional.

Assume that routers A, B and D transit traffic for an IP destination connected to C and that link (B, C) fails. Which nodes among A, B and D could potentially see their packets being stuck in a transient forwarding loop? Which ones would not?

Solution: Nodes A and B could see their packets stuck in a forwarding loop if B updates its forwarding table before A, which is likely to happen as B would be the first to learn about an adjacent link failure. On the other hand, D would not see any loop as it uses its direct link with C to reach any destination connected beyond it.

Assume now that the network administrator wants to take down the link (B, C), *on purpose*, for maintenance reasons. To avoid transient issues, the administrator would like to move away all traffic from the link *before* taking it down and this, without creating any transient loop (if possible). What is the minimum sequence of increased weights setting on link (B, C) that would ensure that *no packet* destined to C is dropped?

Solution: One example of a minimum sequence of weight settings is [1, 3, 5].

Note: The problem highlighted above happens because B shifts traffic to A before A shifts traffic to D, hence creating a forwarding loop. By setting the (B, C) link weight to 3, (only) A shifts from using (A, B, C) to using (A, D, C). Once A has shifted, it is safe to shift B by setting the link weight to 5 (or higher). Once B has shifted has well, the link can be safely torn down.

Link-State vs. Distance-Vector Routing

Comparison

Qualitatively compare link-state and distance-vector routing in the following points:

a) Information sent to neighbors;

Solution: DV: each router sends to its direct neighbors a list (vector) of known networks/destinations and the corresponding distance.

LS: each router sends its own network view (all directly connected routers and link weights) to its neighbors. This information is distributed/flooded to each router in the network until each router knows the entire topology of the network.

b) Convergence time;

Solution: DV: slow. It takes some time until a distance change is distributed and converged through the whole network.

LS: As soon as a network/topology change reaches the router, it can quickly recompute the new shortest-paths.

c) Memory and CPU requirements;

Solution: DV: low. A router only has to update its own distance to a destination based on the received information from the neighbors.

LS: high. Each router has to compute the best paths for each destination using e.g. Dijkstra's algorithm. Each router also has to save the whole network topology.

d) Usability in large networks.

Solution: DV: good. Scales easily to large topologies. Only needs information from its directly connected neighbors. But the convergence time may be very slow.

LS: poor. Each router has to keep track of the whole network topology at all the time. Better usable if the network can be divided into smaller parts/regions.

Exam Question

For the following statements, decide if they are *true* or *false*. Motivate your decision. These questions are directly taken from last year's Communication Networks final exam.

a) Consider a positively weighted graph G. Applying the Bellman-Ford (used by distance-vector protocols) or Dijkstra (used by link-state protocols) algorithm on G would lead to the same forwarding state.

Solution: True. Both solve the shortest-path problem.

b) Link-state protocols (such as OSPF) are guaranteed to compute loop-free forwarding state as long as the link-state databases are consistent on all routers.

Solution: True. However, they can experience transient loops while it isn't the case.

c) Link-state protocols (such as OSPF) require routers to maintain less state than distance-vector protocols (such as RIP).

Solution: False. Link-state protocols require routers to maintain the entire topology in memory (Link-State database). Distance-vector protocols only need to maintain the costs to reach each prefix.

d) Poisoned reverse solves the problem of count-to-infinity.

Solution: False. The problem is still there it is mitigated by having a small infinity value.

e) Consider a positively weighted graph G. Multiplying all link weights by 2 would change the all-pairs shortest paths computed by the Dijkstra algorithm on G.

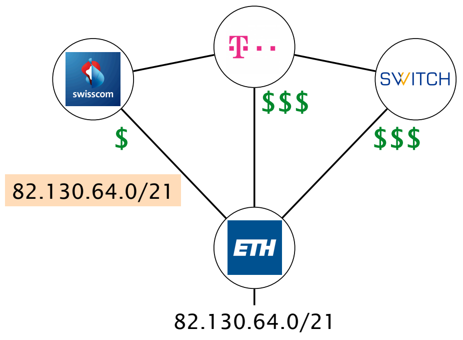
Solution: False. Multiplying by a constant factor keeps the ranking between the paths constant.

f) Consider a positively weighted graph G. Adding 1 to all link weights would change the all-pairs shortest paths computed by the Dijkstra algorithm on G.

Solution: True. Longer paths will see a bigger increase than shorter ones.

Traffic Engineering

Assume that ETH has only one prefix: 82.130.64.0/21. As depicted on the left, the ETH network is connected to three providers (Swisscom, Deutsche Telekom and Switch) and the providers are interconnected with each other. The contract with Swisscom is the cheapest one (indicated by the dollar symbols). For this reason, ETH wants to receive all the incoming traffic over the Swisscom link and therefore announces its prefix only to Swisscom.



ETH is connected to three providers with different costs.

a) Do you think that is a good configuration? What happens if the link between ETH and Swisscom fails?

Solution: Not a good solution. If the link fails, ETH will no longer receive any traffic. ETH is no longer reachable from other networks.

b) To improve the connectivity in case of a link failure between ETH and Swisscom, ETH wants to optimize its announcements. Write down the prefixes which ETH announces to Swisscom, Deutsche Telekom and Switch. During normal operation (no link failure) ETH should still receive all incoming traffic over the Swisscom link.

Solution:

To Swisscom: 82.130.64.0/22 and 82.130.68.0/22 (other splits are also possible)

To Deutsche Telekom: 82.130.64.0/21

To Switch: 82.130.64.0/21

c) After further investigations, ETH decides that only traffic towards 82.130.68.0/23 has to be received over the Swisscom link. All the other traffic can enter over any of the providers. Which prefixes do you have to announce to achieve this traffic distribution?

Solution:

To Swisscom: 82.130.68.0/23 and 82.130.64.0/21

To Deutsche Telekom: 82.130.64.0/21

To Switch: 82.130.64.0/21