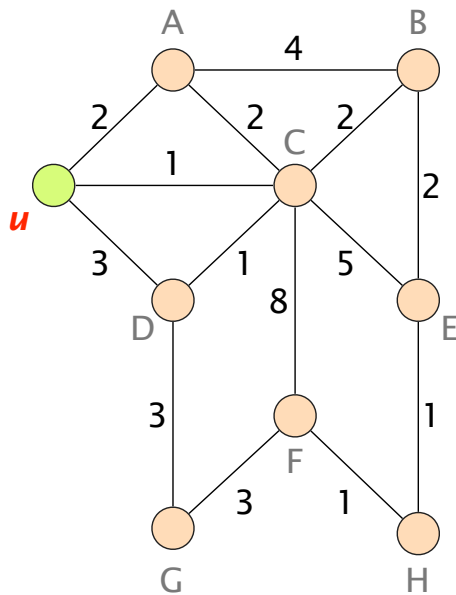# Communication Networks

Prof. Laurent Vanbever

Exercises week 3 – Routing

Dijkstra's Algorithm

Weighted graph representing a network topology.

The figure on the left shows a weighted graph representing a network topology with 9 nodes.

a) Each of the links in the graph has an associated weight. Given that the graph represents a network, what could be the meaning of the link weights?

b) Starting from node u, manually compute Dijkstra's algorithm and list the obtained shortest-paths from u to each of the other nodes. For computing Dijkstra's algorithm, you can use the table below.

c) Based on the shortest-paths from the previous task, derive the forwarding table of node u.

Iteration	Node Set S	$D(\,.\,)$								
		u	A	B	C	D	E	F	G	H
1	u	0	2	∞	1	3	∞	∞	∞	∞
2										
3										
4										
5										
6										
7										
8										
9										

Use this table for computing Dijkstra's algorithm in subtask b.
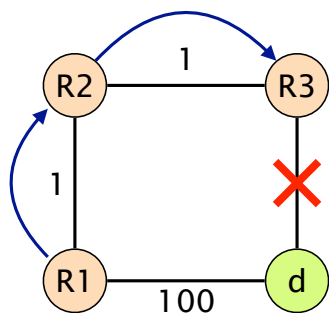
Dijkstra's Algorithm with Link Failure

The routers in the network on the top left use Dijkstra's Algorithm to find the shortest path towards destination d. You can assume that every router knows the entire network graph. In case of a failure, the routers directly affected by it inform the other routers by flooding this information in the network.

The blue arrows indicate how the routers forward the traffic: $R2$, for example, sends packets for destination d to router $R3$.
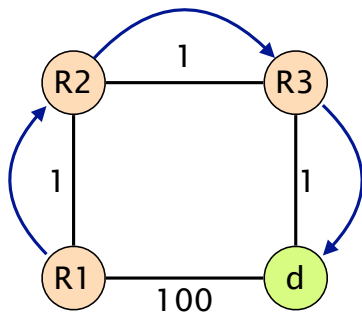
Now the link between router $R3$ and d fails (network at the bottom left) and $R3$ can no longer send packets towards destination d. As $R3$ is directly connected to the failed link, it detects the failure immediately. It starts to flood this information, such that all the routers can update their network view and recompute Dijkstra's algorithm.

Dijkstra's algorithm with a link failure

a) What is the new shortest-path from $R3$ towards destination d?

b) Assume now that the computation of the new shortest-path is *very* fast and finishes before $R3$ starts the flooding of the messages announcing the link failure. $R3$ sends a packet towards d using the new shortest-path. Will the packet reach its destination? Which path will it take?

c) Can you find a sequence of link failure messages and shortest-path computations such that the problem discovered in the previous task is observed between $R1$ and $R2$? The *only* link failure is still between router $R3$ and d.

d) As we have discovered, the order in which the link failure messages are processed and the new shortest-paths are computed is crucial for a correct forwarding behavior in the network. In which order should the router update their forwarding tables, such that the previously observed problems will not occur? Can you find a more general "rule" for a save ordering of forwarding rule updates?

Distance Vector
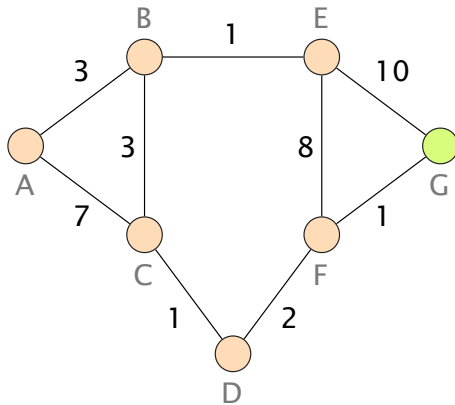
The figure on the left shows a weighted graph representing a network topology with 7 nodes. The nodes in the network use a distance vector algorithm to compute the shortest-paths in a distributed way. It takes one time step for a distance vector message to be sent from one node to another on a link. A node can send the distance vector message on multiple links at the same time.

In case paths have the same weight, the node picks the path traversing the smaller number of links. In case there is still a tie, the node picks the path of the neighbor with the lower identifier (alphabetical order).

Weighted graph representing a network topology.

a) Compute the paths from any node in the network to G. Use the provided table to fill in the state of each node at every time step. Stop when a stable state is reached. The first time step is provided as an example.

#	A	B	C	D	E	F	G
0	∅	∅	∅	∅	∅	∅	0
1	∅	∅	∅	∅	10	1	0
2							
3							
4							
5							
6							

b) Highlight the actual paths taken in the graph.

c) The network operator realizes that there is a potential bottleneck as all traffic is crossing the following links: *C-D*, *D-F*, and *F-G*. She prefers to balance the traffic across the available links in the network. Therefore, she would like to have all traffic from the nodes *A*, *B*, *E* to go across the link *E-G* and the traffic of the remaining nodes to go across *F-G*.

   (i) If she can only change the weight of the link *E-G*, what should she change it to?

   (ii) If she cannot change the weight of the link *E-G*, what should she change instead? Propose a change that requires to change the weights of as few links as possible.

Link Weight Configuration

The Abilene network[a] was a high-performance backbone network in the US. You are the network operator in charge and you have to configure the link weights in the network. Initially, all links have a weight of one and routers will always use the shortest-path available to reach a destination.

[a]https://en.wikipedia.org/wiki/Abilene_Network
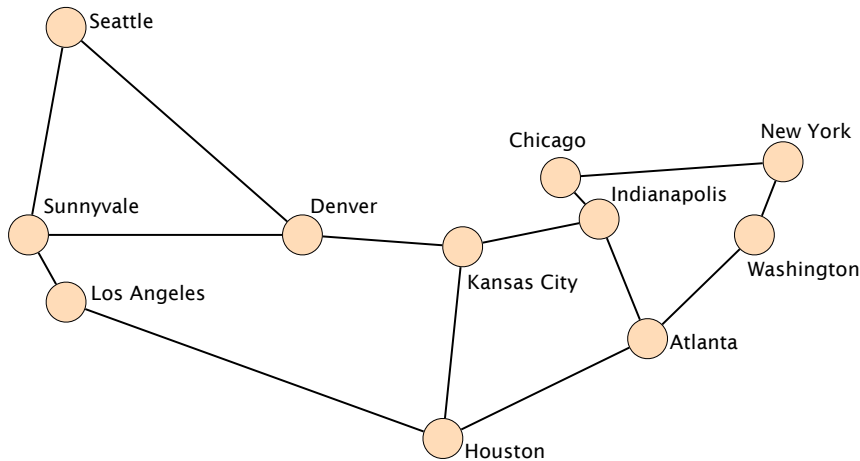
The Abilene network in the US.

a) Is it possible to configure the link weights such that the packets sent by the router located in **Los Angeles** to the routers located in **New York** and to the ones in **Washington** take a different path? Note: the path from Los Angeles to New York and the one from Los Angeles to Washington *should not* have any link in common.

b) Is it possible to configure the link weights such that the packets sent by the router located in **Los Angeles** to the router located in **New York** follow one path while the packets sent by the router located in **New York** to the router located in **Los Angeles** follow a *completely different* path?

c) Assume that the routers located in **Denver** and **Kansas City** need to exchange lots of data. Can you configure the link weights such that the link between these two routers does not carry *any packet* sent by *any other* router in the network