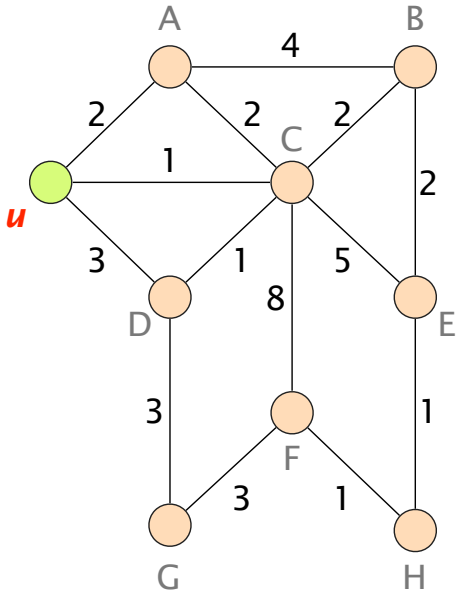
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prof. Laurent Vanbever
Networked Systems Group

Communication Networks

Solution: 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?

Solution: The "cost" of sending traffic via this link (in terms of money, delay, bandwidth, ...)

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.

Solution: The shortest-paths between node u and all other nodes are listed in the following table:

Node	Path	$\sum$(weights)
A	**u** – **A**	2
B	**u** – C – **B**	3
C	**u** – **C**	1
D	**u** – C – **D**	2
E	**u** – C – B – **E**	5
F	**u** – C – B – E – H – **F**	7
G	**u** – C – D – **G**	5
H	**u** – C – B – E – **H**	6

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

Solution: The following table illustrates the forwarding table of node u.

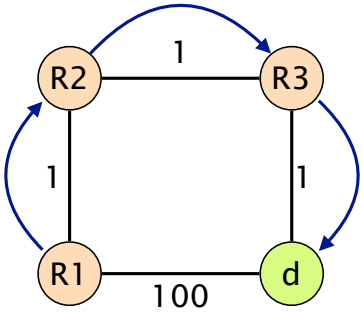
destination	next-hop
A	A
B	C
C	C
D	C
E	C
F	C
G	C
H	C

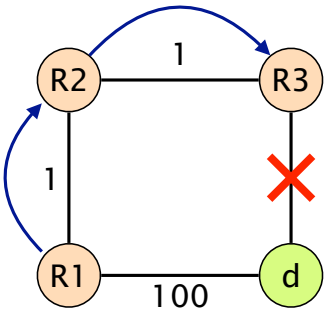
Solution:

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

Use this table for computing Dijkstra's algorithm in subtask b.

Dijkstra's Algorithm with Link Failure





Dijkstra's algorithm with a link failure

For the network on the top left, assume that each router knows the entire network graph. The routers then computed the shortest-path towards the destination node d using Dijkstra's algorithm. The forwarding table of router $R2$, for example, has the following entry: packets towards destination d use router $R3$ as next-hop as indicated by the blue arrow.

Now the link between router $R3$ and d fails (network at the bottom left) and $R3$ can no longer send packets towards destination d. Given that $R3$ is directly connected to the failed link, it will detect the failure immediately. $R3$ will start to flood the network with messages indicating the failed link, such that each router can update its network view. At the same time, $R3$ recomputes Dijkstra's algorithm to find the new shortest-path towards destination d.

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

Solution: $R3, R2, R1, d$

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?

Solution: No, $R2$ does not yet know about the link failure and did not update its shortest-path towards d. It will send the packet back towards $R3$. The packet is therefore stuck in a forwarding loop.

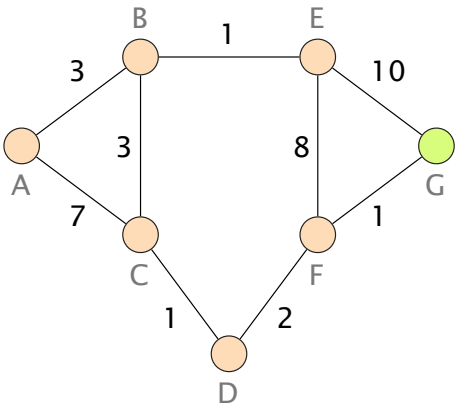
c) Can you find a sequence of link failure messages and shortest-path computations such that the problem discovered in the previous task is observed between $R1$ and $R2$? The *only* link failure is still between router $R3$ and d.

Solution: $R2$ did receive a link failure message and updated its shortest-path. $R2$ then sends a packet towards d to the next-hop $R1$ before $R1$ can update its shortest-path.

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?

Solution: Good order: $R1, R2, R3$. To prevent forwarding loops, the routers should update their forwarding tables based on their distance to the destination. The router nearest to the destination should update its forwarding table first.

Distance Vector



Weighted graph representing a network topology.

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).

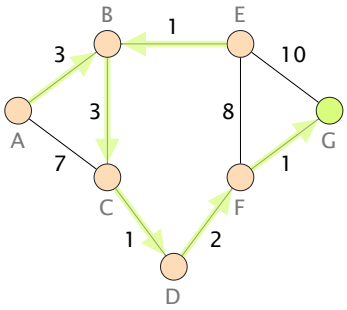
a) Compute the paths from any node in the network to G. Use the provided table to fill in the state of each node at every time step. Stop when a stable state is reached. The first time step is provided as an example.

Solution:

#	A	B	C	D	E	F	G
0	∅	∅	∅	∅	∅	∅	0
1	∅	∅	∅	∅	10	1	0
2	∅	11	∅	3	9	1	0
3	14	10	4	3	9	1	0
4	11	7	4	3	9	1	0
5	10	7	4	3	8	1	0
6							

b) Highlight the actual paths taken in the graph.

Solution:



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?

Solution: 6 or below

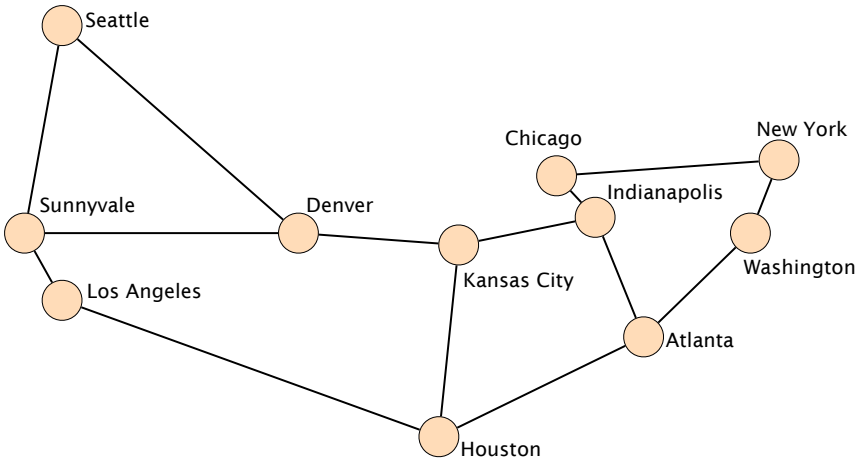
(ii) If she cannot change the weight of the link *E*-*G*, what should she change instead? Propose a change that requires to change the weights of as few links as possible.

Solution: She could set the weight of *F*-*G* to a value in the range from 5 to 10.

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.

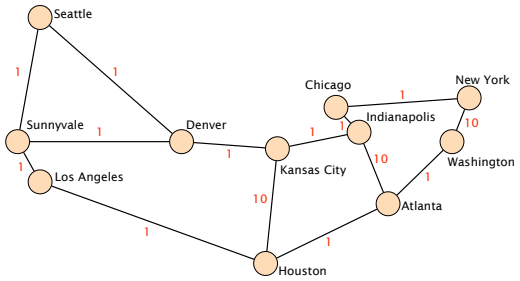
[a] https://en.wikipedia.org/wiki/Abilene_Network



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.

Solution:



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?

Solution: Not possible. We consider only links which have the same weight in both directions. If the two routers would use different paths for the two traffic directions, the two paths would need different total weights. That implies that one path is shorter and one router is not using the shortest-path available. A contradiction to our initial assumption.

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

Solution: