Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Networked Systems
ETH Zürich — seit 2015

Communication Networks

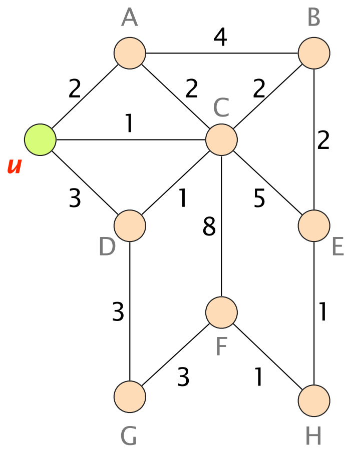
Prof. Laurent Vanbever

Solution: Exercise 2 – Routing and Transport Concepts

Routing Concepts

2.1 Dijkstra's Algorithm

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



Weighted graph representing a network topology.

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.

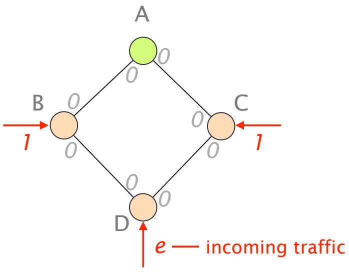
destination	**next-hop**
A	A
B	C
C	C
D	C
E	C
F	C
G	C
H	C

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.

2.2 Changing weights



Network topology with directional link weights.

So far, we have only seen cases in which the link weights in a network were static. However, the Internet itself is not static at all: traffic volumes change constantly; devices connect, move and disconnect. Hence, the following question arises: If the Internet is so dynamic, why should one not use dynamic weights instead?

Consider the figure on the left where B, C, D send traffic to the green destination A. The red arrows show the incoming traffic and its volume (1 or e). Unlike before, the weights on the links are bidirectional. Hence, the weight from A to B can be different to the weight from B to A.

In this network, the traffic is always forwarded along the shortest path according to the link weights. If two paths have the same cost, the path with the (alphabetically) lower next-hop is picked. Initially, for example, when all the weights are 0, B has two paths available to reach the destination: one path via A and another one via D. According to the rule, B picks the path via A.

A speciality of this network is, that the weights are dynamic and always represent the link load. Hence, if there is traffic of volume 1 being forwarded from A to B, the load of the link from A to B and therefore also the weight is 1.

In the following, we ask you to compute the forwarding state. As the link weights are dynamic, the forwarding state changes quite frequently. Therefore, you should approach the task step-by-setp: at every step consider the load on the link to be fixed to the one of the previous step. You can compute the forwarding state and then afterwards, update the load on every link.

Fill in the following table:

Solution:

	Link Load								Next Hop		
	A → B	A → C	B → A	B → D	C → A	C → D	D → B	D → C	B	C	D
0	0	0	0	0	0	0	0	0	A	A	B
1	0	0	1 + e	0	1	0	e	0	D	A	C
2	0	0	0	1	2 + e	0	0	1 + e	A	D	B
3	0	0	2 + e	0	0	1	1 + e	0	D	A	C
4	0	0	0	1	2 + e	0	0	1 + e	A	D	B
5	0	0	2 + e	0	0	1	1 + e	0	D	A	C
6	0	0	0	1	2 + e	0	0	1 + e	A	D	B
7	0	0	2 + e	0	0	1	1 + e	0	D	A	C

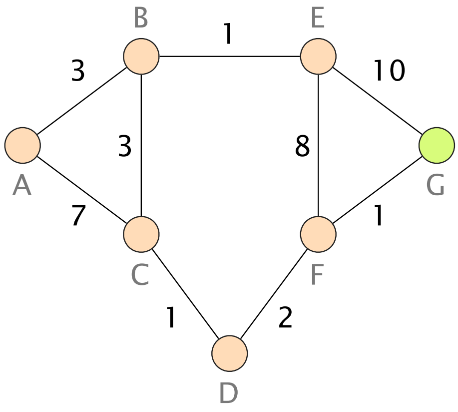
What is the problem with the dynamic weights?

Solution: The dynamic weights lead to oscillations in the network. The forwarding state oscillates between two states: all left and then all right.

2.3 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.

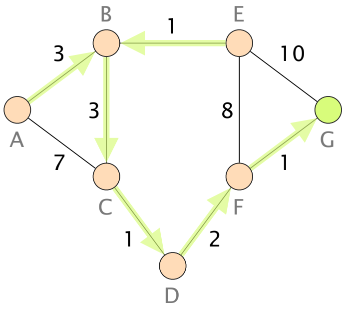
a) Compute the paths from any node in the network to G. Use the provided table to fill in the state of each node at every time step. Stop when a stable state is reached. The first time step is provided as an example.

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.

2.4 Source-and-Destination-Based Routing

As we have seen in the lecture, destination-based routing is the default in the Internet. Hence, based on the destination of a packet, the router decides where to forward an incoming packet next. However, it can also base its decision on other criteria, such as the source of a packet.

a) Is it possible to design a routing scheme that does not rely on the destination of a packet and still produces a valid global forwarding state? Justify.

Solution: It is not possible. As we learned in the lecture, the globabl forwarding state is valid if it always delivers packets to the correct destination. However, when forwarding packets without looking at their destination, it is not possible to know where to send the packet next such that it reaches eventually the destination.

b) Compare destination-routing that is solely based on the destination and source-and-destination routing that uses both the source and destination. What are the advantages and disadvantages of the two in terms of path diversity and the state required?

Solution:

Destination-Routing The path diversity is low as all the packets follow the same path at some point. The required state is also low as every device needs at most n entries if there are n different addresses in the Internet.

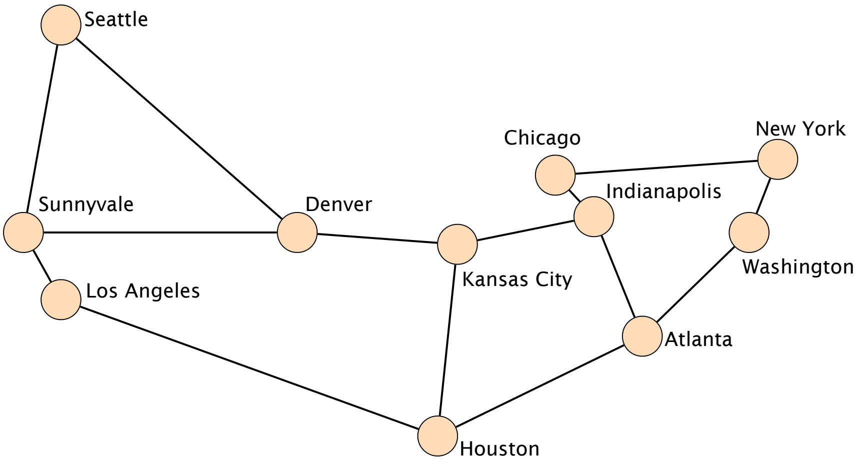
Source-and-Destination-Routing The path diversity is high as packets for one destination coming from one source could take a different path as the packets from all the other sources. The required state is also low as every device needs at most n^2 entries, one for each source-destination pair.

2.5 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.

---

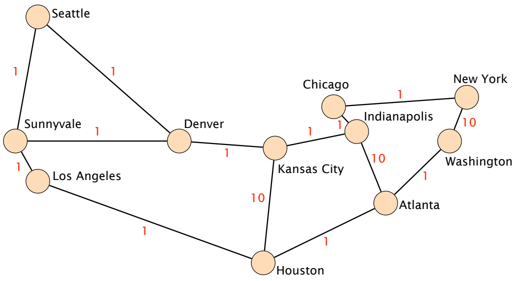
[a] https://en.wikipedia.org/wiki/Abilene_Network



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:

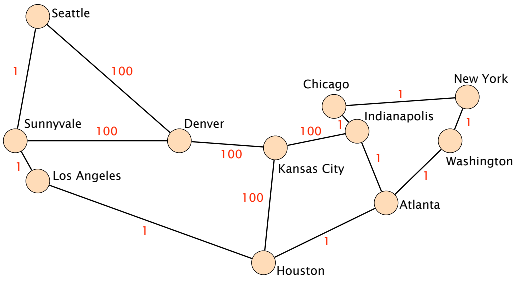


b) Is it possible to configure the link weights such that the packets sent by the router located in **Los Angeles** to the router located in **New York** follow one path while the packets sent by the router located in **New York** to the router located in **Los Angeles** follow a *completely different*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



Reliable Transport Concepts (Part 1)

2.6 Reliable versus Unreliable Transport

In the lecture, you have learned how a reliable transport protocol can be built on top of a best-effort delivery network. However, some applications still use an unreliable transport protocol.

a) What are the characteristics of best-effort and of reliable transport?

Solution:

- Best-effort delivery: There is no guarantee for packets to arrive in the correct order, correctly (bit corruption) or even arrive at all.
- Reliable transport: It provides all the above guarantees by making use of sequence numbers, checksums and acknowledgements.

b) What could be advantages of using an unreliable transport protocol?

Solution:

- Better performance/less overhead since you don't have to wait for ACKs to arrive;
- Lightweight implementation;
- As no connection setup is required (e.g., TCP three-way handshake), you can immediately start sending.

c) What type of applications are suitable to use unreliable transport protocols?

Solution: Applications for which it is more important to have "live" data than to have "complete" data. In voice/video-calls, for example, lost packets lead to lower quality, but delayed packets lead to distorted conversations.

d) As we will later see, the User Datagram Protocol (UDP) only provides unreliable transport. Assume you are forced to use a network which only supports UDP as a transport protocol. You must transmit an important document which eventually should be correctly transmitted. Do you see a way to implement some of the reliable transport mechanisms despite using UDP?

Solution: Yes, the reliable transport mechanisms could be implemented by the application/in the application layer.