Networked Systems
ETH Zürich — seit 2015

Communication Networks

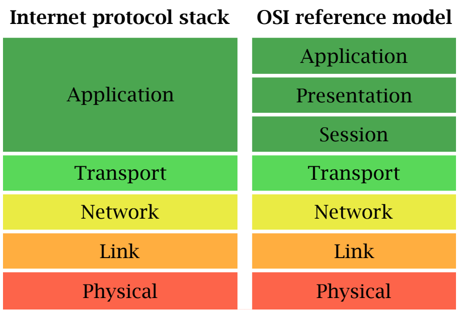
Prof. Laurent Vanbever

Solution: Exercise 1 – Introduction & Routing Concepts

Internet Structure

1.1 Layer Model

Communication over the Internet can be decomposed into independent layers. In the lecture, we have discussed the Internet protocol stack which contains 5 layers. Another often used model is the OSI (Open Systems Interconnection) model with 7 layers. Find the best matching layer for the following operations/devices. You can use the already known 5-layer model. **Hint**: Some of the operations could be implemented in different layers.



Internet communication layers: Internet protocol stack and the OSI reference model.

a) Bit-to-bit transmission between two devices.

Solution: Physical layer

b) Encryption of a message.

Solution: Application layer

c) A switch in a network.

Solution: Link layer

d) Routing path search.

Solution: Network layer

e) Adding a sequence number to each packet.

Solution: Transport layer (e.g. TCP protocol), but also the application layer could provide this functionality.

f) A router in a network.

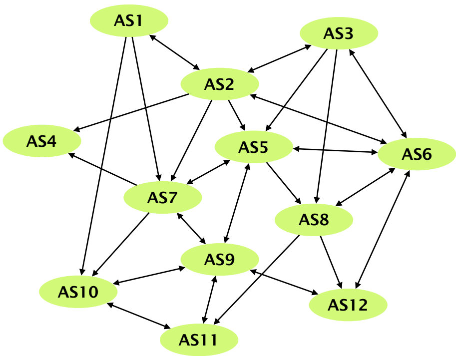
Solution: Network layer

g) A middlebox in a network performing deep packet inspection (DPI) to find malware in Web traffic.

Solution: To analyze the payload of packets the middlebox is operating in the application layer. Most likely, it will also use information from other layers, e.g. IP addresses from the network layer.

1.2 Internet Organization

The network on the left consists of multiple autonomous systems (AS). Single-headed arrows point from providers to their customers. Double-headed arrows represent peer connections.



A network of multiple autonomous systems (AS).

a) For each AS, identify if it is a Tier-1, Tier-2 or Tier-3 ISP network or an IXP.

Solution: Tier-1 ISPs: AS1, AS2, AS3. Tier-2 ISPs: AS5, AS7, AS8. Tier-3 ISPs: AS4, AS10, AS11, AS12. IXPs: AS6, AS9.

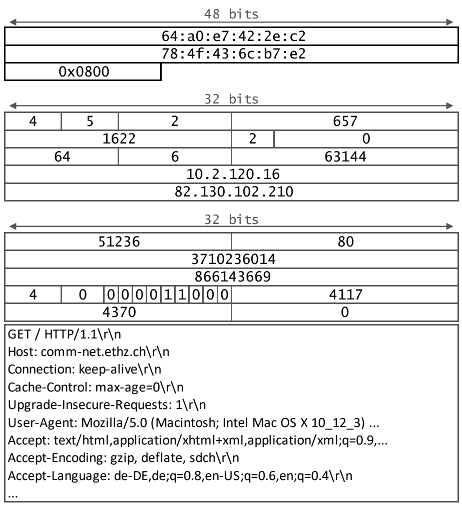
b) AS7 has two different providers (AS1 and AS2). How is this type of interconnection called? What are the advantages of multiple different providers for AS7? Can you see any disadvantages?

Solution: Multihoming.

Advantages: Still connected if one of the provider fails. For destinations (IP prefixes) which can be reached over both providers, AS7 can choose the better one based on cost, trust, ...

Disadvantages: network configuration is slightly more difficult. The total cost could be higher compared to an ISP with only one provider.

1.3 Internet Communication



Packet consisting of data in four layers (link, network, transport and application layer).

The figure on the left shows a (simplified version of a) packet that was recorded at your machine's network interface using `Wireshark`[a].

a) What kind of activity does this packet belong to?

Solution: Web browsing. The easiest way to see this is by looking at the payload of the application layer (the lowermost box in the figure). There, one can see that the packet contains a `HTTP-GET` request to `comm-net.ethz.ch`.

b) From the lecture, you know that a packet is composed of data in different layers. The fields in the left-hand figure are already grouped to four blocks representing the link, network, transport and application layer. Do you know which protocol is used in each of the four layers? *Hint: You may want to have a look at the list of protocols on Wikipedia*[b].

Solution:

- Link layer: Ethernet (indicated by the address-format `00:00:00:00:00:00`). The EtherType tells the parser which protocol is used for the next layer. In this case, the `0x800` stands for IPv4.
- Network layer: IP version 4 (indicated by the address-format `000.000.000.000`). The protocol field identifies the next protocol type: here, the 6 stands for TCP.
- Transport layer: TCP (identifiable by comparing with the format of the TCP-header[c]). The destination port tells the networking stack to which application the payload should be delivered. Port 80 is the standard port for HTTP.
- Application layer: HTTP (HTTP-Request in plain text)

c) After being sent out at your machine's network interface, the packet will traverse a switch which will forward it based on the destination address. On which layer does the switch operate and which is the destination address it will look at?

Solution: Switches operate at the link layer. The destination address in this layer is `64:a0:e7:42:2e:c2`.

d) After leaving your local network, the packet will traverse a router. On which layer does the router operate and which is the destination address it will look at?

Solution: Routers operate at the network layer. The destination address in this layer is `82.130.102.210`.

[a] `https://www.wireshark.org/`
[b] `https://en.wikipedia.org/wiki/Internet_protocol_suite`
[c] `https://en.wikipedia.org/wiki/Transmission_Control_Protocol`

1.4 [On your own] Decipher an Internet Path using Traceroute

When two end-systems (hosts or servers) communicate with each other over the Internet, their communication traverses multiple routers.

The `traceroute` (or `tracert` on Windows) utility lists the routers that are on the path from one end-system to another. E.g., if you type `traceroute <target>` in the command line, where `<target>` is an IP address or a domain name, that will display a list of router IP addresses and/or domain names and the round-trip-times (RTTs) that were measured between your computer and each router on the path to `<target>`.

There also exist traceroute servers (e.g., see `www.traceroute.org`) that allow you to traceroute from them to any other computer in the world.

We have already used `http://traceroute.sdv.fr` to trace the routers on the path from a server in France to `ethz.ch` (cf. left-hand figure):

```
1 wblindix.sdv.fr 212.95.66.126 0.102 ms
2 border-gateway2.sdv.fr 212.95.69.227 0.85 ms
3 ipv4.de-cix.fra.de.as559.switch.ch 80.81.196.147 9.221 ms
4 swiBS1-100GE-0-0-0-0.switch.ch 130.59.37.34 9.437 ms
5 swiPS1-100GE-0-0-1-3.switch.ch 130.59.37.190 10.535 ms
6 swiPS2-100GE-0-0-1-4.switch.ch 130.59.37.58 10.476 ms
7 swiZH3-100GE-0-0-0-2.switch.ch 130.59.36.170 11.268 ms
8 swiZH1-B1.switch.ch 130.59.37.65 11.117 ms
9 swiEZ2-B2.switch.ch 130.59.38.109 13.437 ms
10 swiEZ3-B1.switch.ch 130.59.36.126 11.066 ms
11 rou-gw-lee-tengig-to-switch.ethz.ch 192.33.92.1 10.915 ms
12 rou-fw-rz-rz-gw.ethz.ch 192.33.92.169 10.916 ms
13 cms-publish.ethz.ch 129.132.19.216 10.545 ms
```

Traceroute output from a server in France to ethz.ch

(a) How many routers are there between the server in France and `ethz.ch`?

Solution: There are 12 routers (line 13 corresponds to the target ethz.ch server).

(b) How many of these routers are inside the ETHZ network? How many are inside ETHZ's Internet Service Provider (ISP)?

Solution: Routers 11-12 are inside the ETHZ network, as their domain names have the ethz.ch suffix. Routers 3-10 are inside ETHZ's ISP (which is called SWITCH).

Now traceroute from a server in Princeton University (`https://www.net.princeton.edu/traceroute.html`) to `ethz.ch`.

Solution:

```
traceroute to 129.132.19.216 (129.132.19.216), 30 hops max, 40 byte packets
1 core-ns-router (128.112.128.2) 0.930 ms 0.828 ms 0.847 ms
2 rtr-core-west-router.princeton.edu (128.112.12.229) 0.910 ms 0.666 ms 0.741 ms
3 fw-border-hpcrc-router.princeton.edu (128.112.12.14) 1.103 ms 1.129 ms 1.043 ms
4 rtr-border-hpcrc-router.princeton.edu (204.153.48.253) 1.475 ms 1.403 ms 1.251 ms
5 172-96-130.unassigned.userdns.com (172.96.130.49) 3.524 ms 3.058 ms 3.187 ms
6 bundle-ether240.200.core1.newy32aoa.net.internet2.edu (163.253.5.38) 6.312 ms 5.722 ms 4.129 ms
7 198.71.45.237 (198.71.45.237) 77.297 ms 75.866 ms 76.052 ms
8 ae7.mx1.gen.ch.geant.net (62.40.98.238) 83.937 ms 83.921 ms 83.851 ms
9 swice1-100ge-0-3-0-1.switch.ch (62.40.124.22) 85.288 ms 87.816 ms 85.556 ms
10 swiEZ2-400GE-0-0-0-0.switch.ch (130.59.38.82) 90.262 ms 90.852 ms 89.309 ms
11 swiEZ3-B1.switch.ch (130.59.36.126) 88.471 ms 88.290 ms 88.074 ms
12 rou-gw-lee-tengig-to-switch.ethz.ch (192.33.92.1) 88.290 ms 88.272 ms 88.239 ms
13 rou-fw-rz-rz-gw.ethz.ch (192.33.92.169) 88.514 ms 88.316 ms 88.252 ms
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
23 * * *
24 * * *
25 * * *
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

(c) Between which routers does the path cross the Atlantic?

Solution: Between routers 6 and 7, as we can see from the "jump" in the RTTs (from <7 ms to >75 ms).

(d) Why do we see asterisks at certain lines of the traceroute output?

Solution: We see asterisks at line 14 onwards, while based on the previous traceroute, we would have expected to see the ethz.ch server at line 14. One possible explanation is that ethz.ch is configured to not reply to the type of traceroute messages the Princeton server uses (while it still replies to the type of traceroute messages the server in France uses).

(e) Does the network path from the server in France to `ethz.ch` overlap with the path from the Princeton server to `ethz.ch`?

Solution: Yes, they overlap at the last three routers (one inside SWITCH, and two inside ETHZ).

(f) If you were to traceroute again tomorrow, e.g., from the Princeton server to `ethz.ch`, do you think that the traceroute output would necessarily be the same? Why could it change?

Solution: No, the traceroute output may differ because: i) the path has changed (e.g., because a link has failed in the old path), ii) there exist multiple paths between the Princeton server and ethz.ch and the traceroute messages take a different path each time, iii) ethz.ch has moved to a different location, or iv) the traceroute messages are forwarded to another server, which is also “mapped” to the ethz.ch domain name.

Bandwidth and Delay

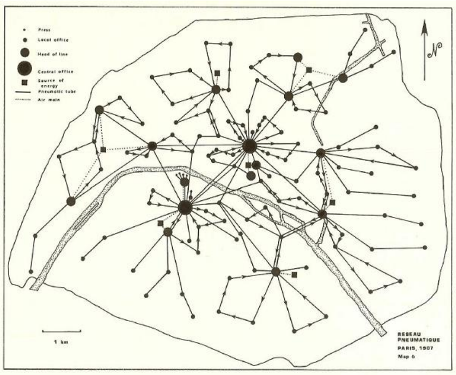
1.5 [Optional] Network Characterization

Calculate and compare the bandwidth and the delay for different communication methods. (For this task, assume that the delay only consists of propagation delay.)

Solution: General formulas:

$$\text{delay} = \frac{\text{distance from A to B}}{\text{speed}}$$

$$\text{bandwidth} = \frac{\text{total amount of data in bits}}{\text{delay}}$$



The pneumatic tube network of Paris in 1907.

a) Pigeon post: Pigeons can be used as messengers. They are trained to transport messages from one location to another. Assuming you want to send a USB flash drive with 16 GB[a] from Zürich to a friend in Paris (500 km) (only in one direction). Calculate the bandwidth and the delay for one pigeon carrying the USB drive and traveling at an average speed of 80 km/h.

Solution: Delay: $d = \frac{500\,\text{km}}{80\,\text{km/h}} = 6.25\,\text{h} = 22'500\,\text{s}$

Bandwidth: $bw = \frac{1.28\times10^{11}\,\text{bits}}{2.25\times10^{4}\,\text{s}} \approx 5.7\,\text{Mbps}$

b) Pneumatic tube: These systems were introduced in the late 19th century to transport small, urgent items within buildings or even within cities. The capsules travel at an average speed of 8 meters per second. Assuming you send an external hard drive with 2 TB of storage through a tube from ETZ to the main building (distance 400 m), calculate the bandwidth and delay.

Solution: Delay: $d = \frac{400\,\text{m}}{8\,\text{m/s}} = 50\,\text{s}$

Bandwidth: $bw = \frac{1.6\times10^{13}\,\text{bits}}{50\,\text{s}} \approx 320\,\text{Gbps}$

c) AWS Snowmobile: Amazon uses a truck to move data from its customers to their data center. The truck houses a container which can store 100 PB of data.[b] Assuming the truck is transporting data from New York to an AWS data center in San Francisco (distance 4700 km) at an average speed of 100 km per hour, calculate the bandwidth and the delay.

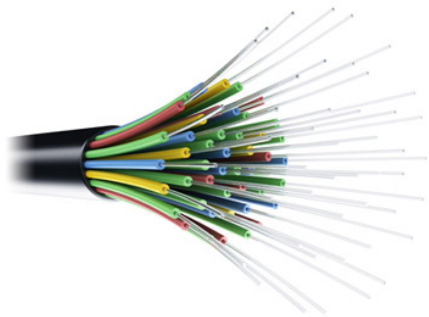
Solution: Delay: $d = \frac{4700\,\text{km}}{100\,\text{km/h}} = 47\,h = 169'200\,\text{s}$

Bandwidth: $bw = \frac{8\times10^{17}\,\text{bits}}{1.692\times10^{5}\,\text{s}} \approx 4.7\,\text{Tbps}$

[a] 1 GB = 10^9 bytes, 1 TB = 10^{12} bytes, 1 PB = 10^{15} bytes

[b] https://techcrunch.com/2016/11/30/amazon-will-truck-your-massive-piles-of-data-to-the-cloud-with-an-18-wheeler/

1.6 Types of Delay



Optical fiber cable.

When accessing a website your data has to travel from your computer through different networks to the server on which the website resides and back.

a) You want to load a website from Sydney (`sydney.edu.au`). How long does it take for a data packet to travel to the server and back (straight-line distance Zürich-Sydney 16′600 km) assuming your data packet travels at the speed of light (3×10^8 m/s)?

Solution: $t = \frac{\text{distance}}{\text{speed}} = \frac{2 \times 1.66 \times 10^7 \text{ m}}{3 \times 10^8 \text{ m/s}} \approx 110.7 \text{ ms}$

b) Measure the time it actually takes for a packet to travel to the server and back by issuing a `ping sydney.edu.au` in a terminal[a]. If you don't have a computer available you can use our measurement:

```
PING sydney.edu.au (20.248.131.216):
time=236.769 ms
```

The calculated and measured times are not even close. Why do we observe such a difference?

Solution: The time calculated in the first task only accounts for the propagation delay (i.e., the time it takes to send the data at the speed of light) assuming a straight-line connection. In the following, we list some points which have been neglected:

- As we have seen in the lecture, there is not only the propagation, but also the transmission, processing and queuing delay.
- The cables usually don't follow the straight-line between the two locations. Hence, the real distance is longer.
- The speed of light in fiber cables is reduced by about 30%.

[a] https://linux.die.net/man/8/ping

Sharing

1.7 Packet vs. Circuit Switching

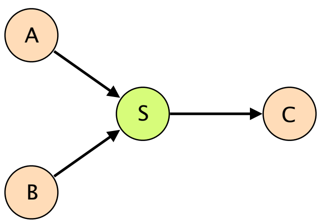
Consider the network on the left. A and B are sending data towards C over a switch S and a shared link. All the links in the network have a bandwidth of 10 Mbps and you can assume that the propagation and switch processing time are negligible. For circuit switching, assume that circuit establishment and teardown each take 50 ms. For packet switching, you can assume that switch S already knows how to reach C.

a) How long does it take if node A is sending a 50 Mbit file to C using packet switching? B sends nothing.

Solution: 5 s

b) How long does it take if node B is sending a 50 Mbit file to C using circuit switching? A sends nothing.

Solution: 5.1 s



Network with a shared link.

Assume now that A and B are using packet switching and are each sending a 50 Mbit file to C at the same time.

c) What will happen if the switch has no buffer?

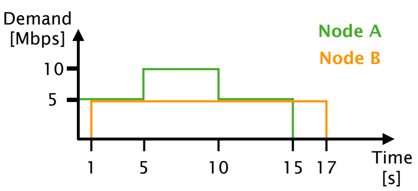
Solution: Some of the packets are dropped.

d) How large has the buffer to be (in Mbit) such that both senders can successfully transmit their files to C when they are simultaneously sending at full speed?

Solution: 50 Mbit

e) For obvious reasons, switch buffers cannot be of unlimited size. How is it possible to successfully send data over the Internet even if some packets are dropped due to full buffers?

Solution: Dropped packets are retransmitted, e.g. using the reliable transport guarantees that the TCP protocol provides (transport layer).



Demand distributions for node A and B.

Finally, assume that A and B have to send data with a demand according to the diagram on the left.

f) How long does it take to send all data if A and B use circuit switching (reserving for the peak demand)?

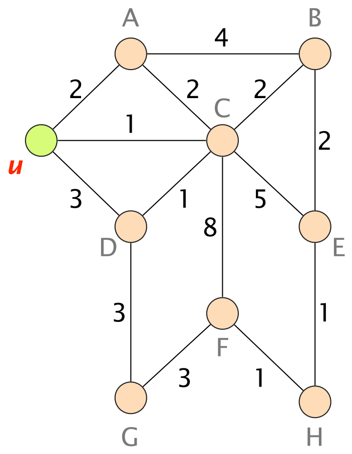
Solution: First, node A reserves 10 Mbps bandwidth. During this time, node B cannot establish its circuit. $0.05 + 15 + 0.05 + 0.05 + 16 + 0.05 = 31.2$ s

g) How long does it take to send all data if A and B use packet switching (you can assume an unlimited buffer size on S)?

Solution: Both nodes start to send packets immediately. From 5 to 10 s, packets are buffered. Assuming the switch always uses the full link bandwidth towards C: $1 + 14 + 2 + 1.5 = 18.5$ s

Routing Concepts

1.8 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, (i) manually compute Dijkstra's algorithm, and then (ii) list the obtained shortest-paths from u to each of the other nodes. For computing Dijkstra's algorithm, you can use the table below. The algorithm follows the one discussed in the lecture. If several nodes could next be added to node set S, select the node that comes first in the alphabet.

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:

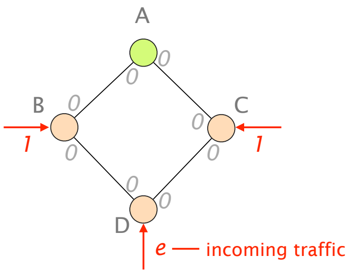
Iteration	Node Set S	$D(\,.\,)$ u	A	B	C	D	E	F	G	H
1	u	0	2	∞	1	3	∞	∞	∞	∞
2	u, C	0	2	3	1	2	6	9	∞	∞
3	u, C, A	0	2	3	1	2	6	9	∞	∞
4	u, C, A, D	0	2	3	1	2	6	9	5	∞
5	u, C, A, D, B	0	2	3	1	2	5	9	5	∞
6	u, C, A, D, B, E	0	2	3	1	2	5	9	5	6
7	u, C, A, D, B, E, G	0	2	3	1	2	5	8	5	6
8	u, C, A, D, B, E, G, H	0	2	3	1	2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

1.9 Changing 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. This begs the question: 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. (This is the only traffic in the network.) The red arrows show the incoming traffic and are labeled with its volume (1 or e). You can assume that $e >> 1$. 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.



Network topology with directional link weights.

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, e.g. 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 specialty 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-step: at every step, consider the load on the link to be fixed to the one of the previous step. With this, compute the forwarding state, and 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

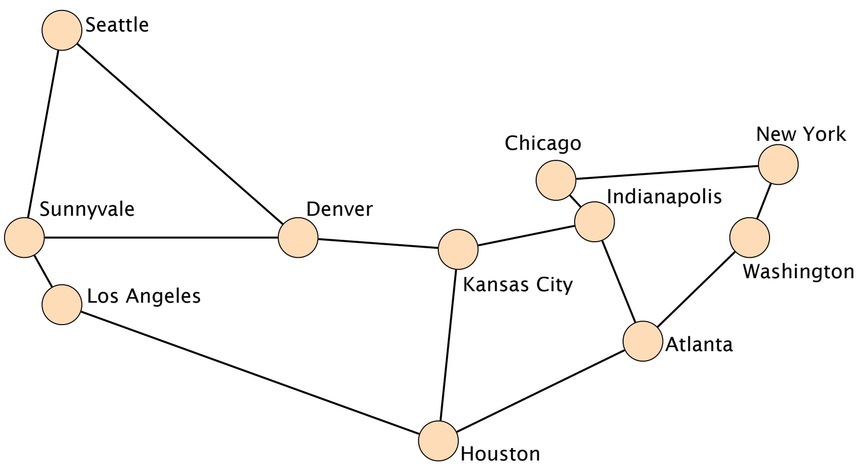
What is the problem with the dynamic weights?

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

1.10 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. For this task, assume that the weights have to be symmetric (i.e., for a given link, the weight is the same in both directions).

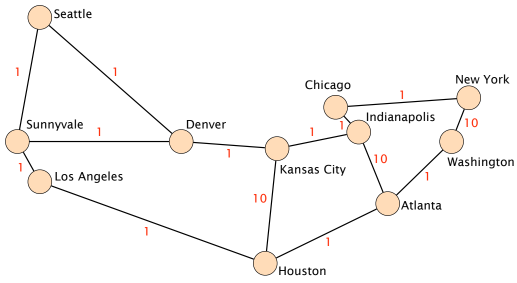
[a] https://en.wikipedia.org/wiki/Abilene_Network



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:

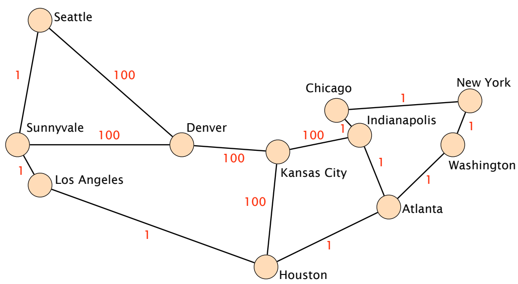


b) Is it possible to configure the link weights such that the packets sent by the router located in **Los Angeles** to the router located in **New York** follow one path while the packets sent by the router located in **New York** to the router located in **Los Angeles** follow a *completely different*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

Solution: Not possible in general.[a] 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 on the direct link. 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:



[a] It would be possible in the following special case: there are at least two paths between New York and Los Angeles with the same minimum total weight and the two routers choose at random different minimum-total-weight paths for the two directions.

1.11 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 global 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 high as every device needs at most n^2 entries, one for each source-destination pair.

1.12 [Optional] Reverse Dijkstra (Exam Question 2020)

The network engineer at your company just retired and you have to take over. Unfortunately, it is unclear how the current network looks like. All you know is that it consists of 10 nodes (see below). In addition, you know that there is at most one link between two nodes and that each link has a non-negative weight. However, you neither know which links exist nor the weights configured on these links.

a) To figure out the links and the corresponding weights, you look at an output of Dijkstra's algorithm performed from node **U**. The table below shows the entire output of the algorithm. For each iteration, the table indicates the shortest path found so far towards each other node (starting from node **U**). The algorithm follows the one discussed in the lecture. If after one iteration there are multiple nodes with an equally-shortest path, the algorithm continues with the node which comes first in the alphabet.

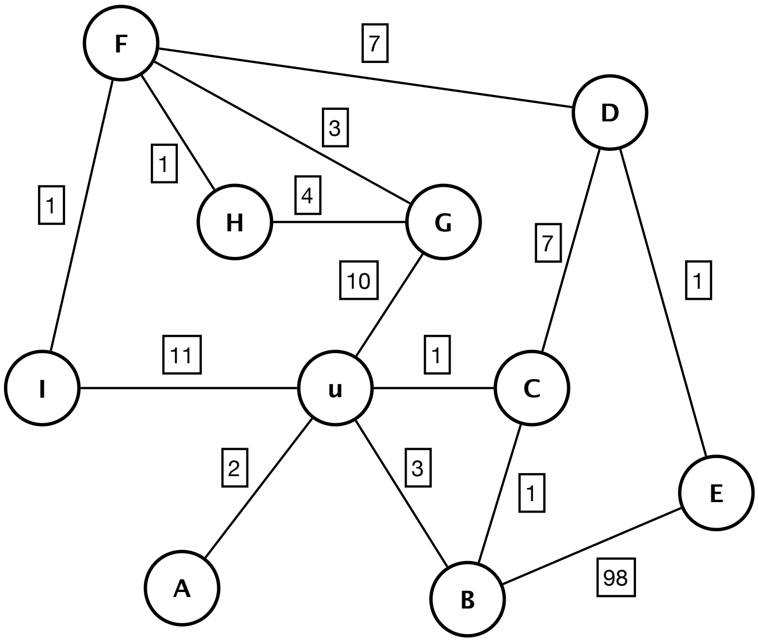
Add all the links with their corresponding weight that you can identify based on the output from Dijkstra's algorithm.

A network consisting of 10 nodes with unknown links and link weights.

#	**U**	A	B	C	D	E	F	G	H	I
1	0	2	3	1	-	-	-	10	-	11
2	0	2	2	1	8	-	-	10	-	11
3	0	2	2	1	8	-	-	10	-	11
4	0	2	2	1	8	100	-	10	-	11
5	0	2	2	1	8	9	15	10	-	11
6	0	2	2	1	8	9	15	10	-	11
7	0	2	2	1	8	9	13	10	14	11
8	0	2	2	1	8	9	12	10	14	11
9	0	2	2	1	8	9	12	10	13	11
10	0	2	2	1	8	9	12	10	13	11

For each iteration (1 to 10) the table shows the shortest path found by Dijkstra's algorithm performed on node **U** towards all other nodes.

Solution: The figure shows the found links and weights.



b) After analyzing the output from Dijkstra's algorithm, you are unsure if you really found all links in the network.

Could there be an additional link starting from node **U** which you could not identify based on the output from Dijkstra? If you think that is possible, give an example (link between node **U** and node ...) and indicate in which range the weight of this link could be. Otherwise, explain why this is not possible.

Solution: Not possible. Given that we perform Dijkstra's algorithm starting from node **U** (and there is only a single link between each pair of nodes) we see all possible links in the first iteration as Dijkstra adds all adjacent nodes during its initialization.

c) Could there be an additional link starting from node C which you could not identify based on the output from Dijkstra? If you think that is possible, give an example (link between node C and node ...) and indicate in which range the weight of this link could be. Otherwise, explain why this is not possible.

Solution: Possible. For example link between C and G with weight greater (or equal) than 9.