Communication Networks

Spring 2020

Tobias Bühler

https://comm-net.ethz.ch/

ETH Zürich

March 5 2020

Communication Networks

Exercise 2

Quick Dijkstra repetition

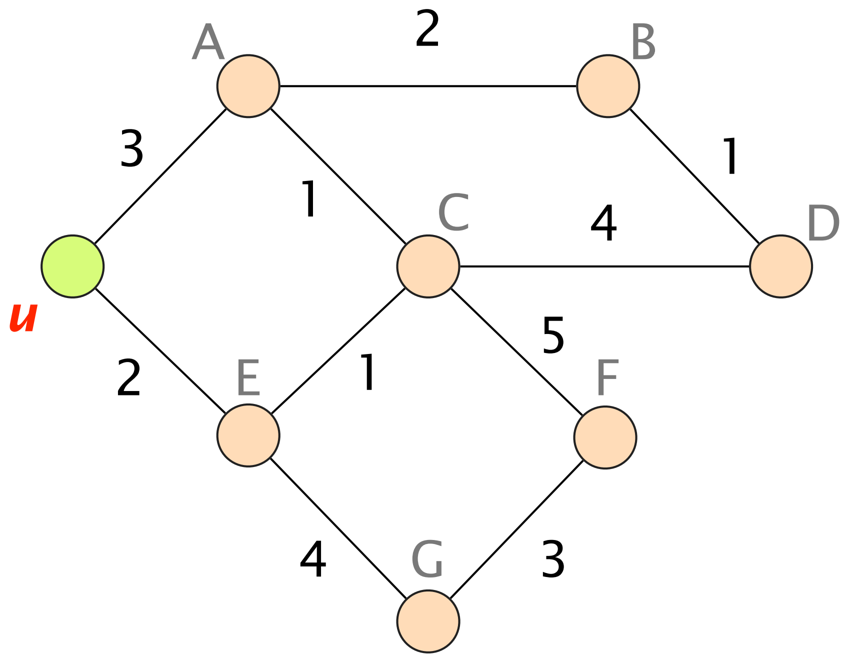
Overview current assignment

Questions?

Time for you to solve the tasks

Solutions will be published next week

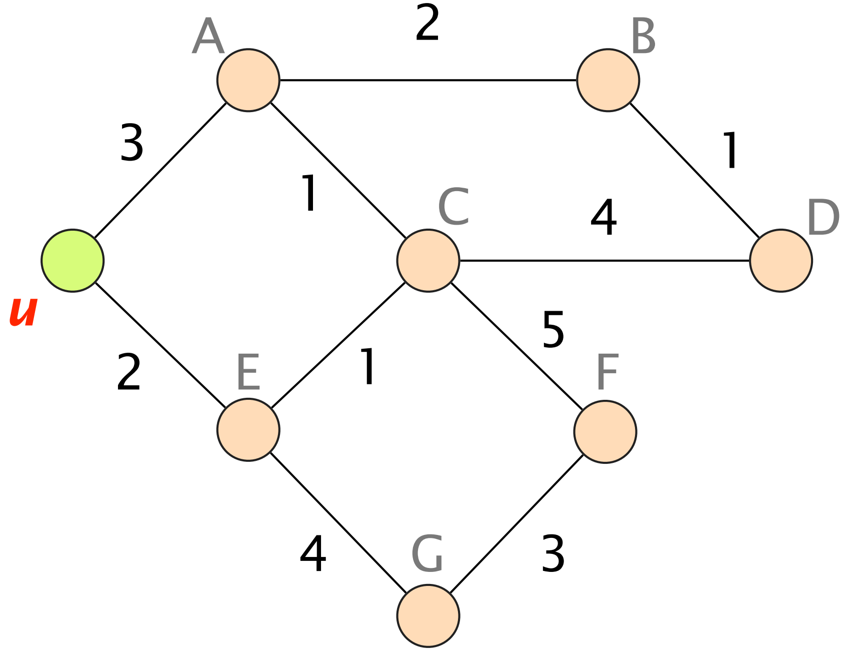
Let’s compute the shortest-paths from *u* using Dijkstra’s algorithm



S = {*u*} — set of nodes for which we know the shortest-path

D(*v*) — the smallest distance currently known by *u* to reach *v*

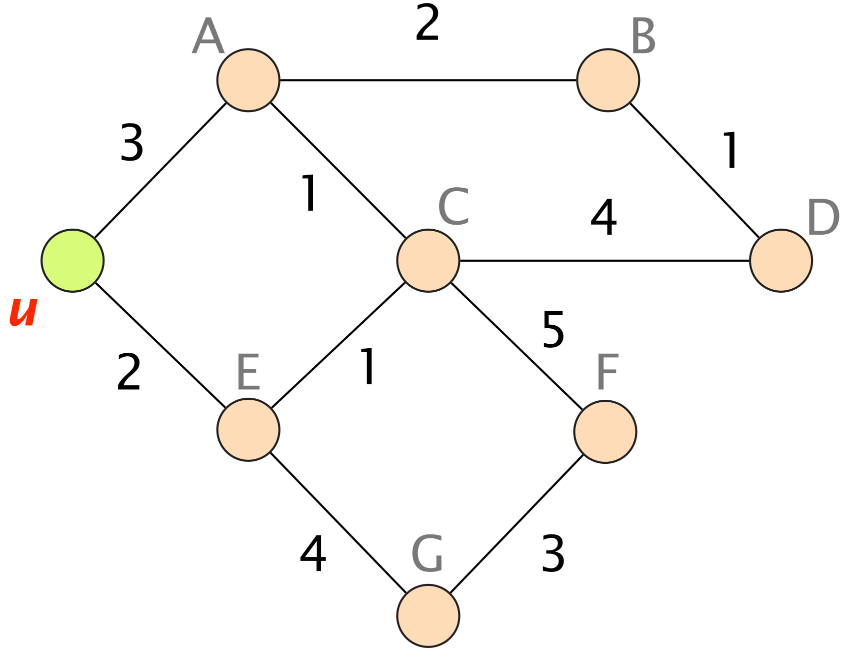
c(*u*,*v*) — the weight of the link connecting *u* and *v*



Initialization

```
S = {u}
for all nodes v:
    if (v is adjacent to u):
        D(v) = c(u,v)
    else:
        D(v) = ∞
```

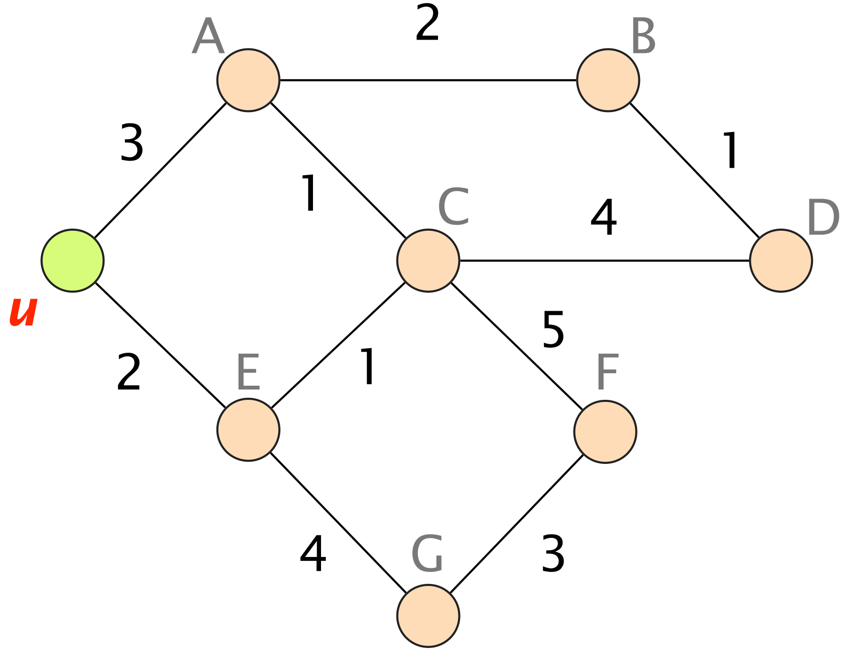
D is initialized based on u’s weight, and S only contains u itself



D(.) =

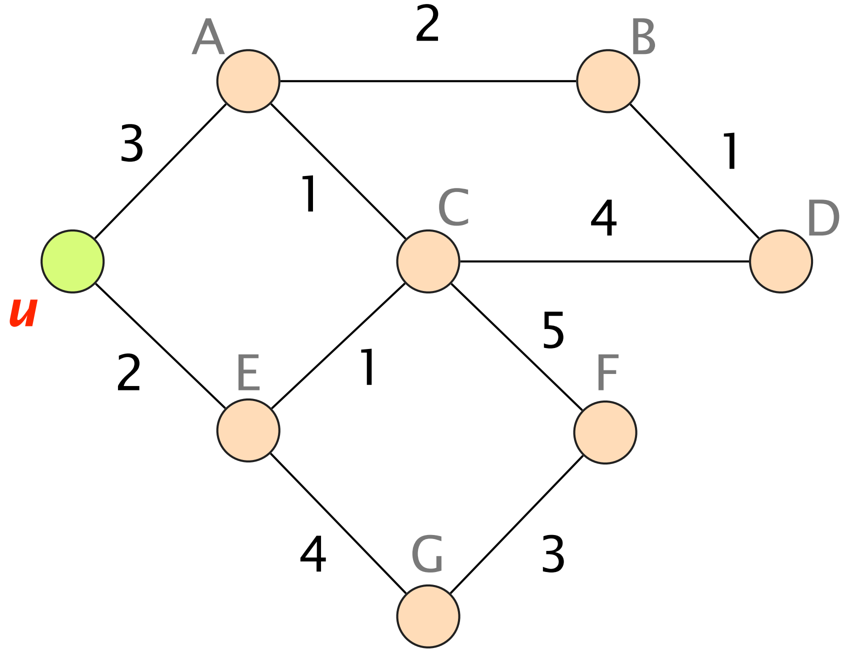
S = {u}

A	3
B	∞
C	∞
D	∞
E	2
F	∞
G	∞



Loop

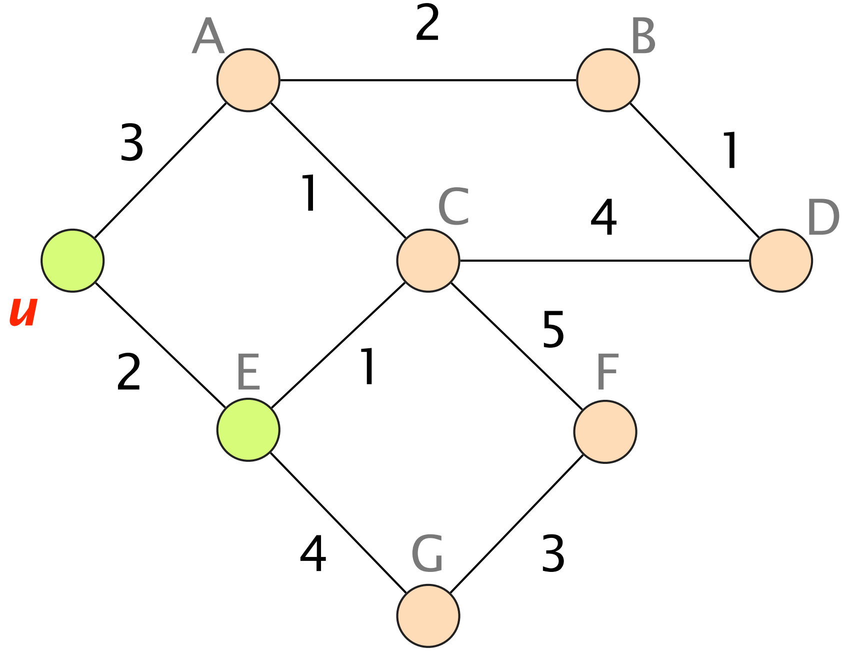
```
while not all nodes in S:
    add w with the smallest D(w) to S
    update D(v) for all adjacent v not in S:
        $D(v) = \min\{D(v), D(w) + c(w,v)\}$
```



D(.) = S = {u}

A	3	
B	∞	
C	∞	
D	∞	
E	2	smallest D(*w*)
F	∞	
G	∞	

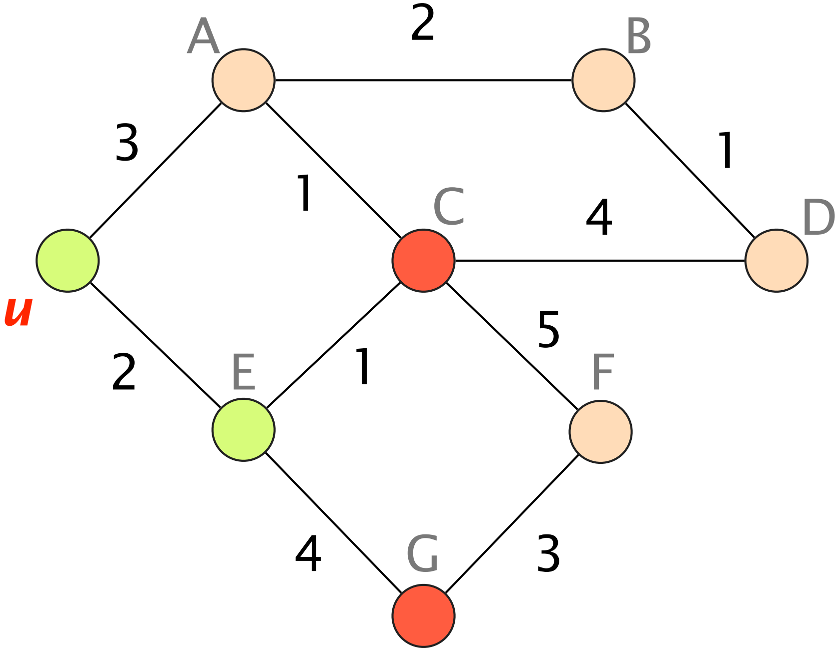
add E to S



D(.) =

S = {u, E}

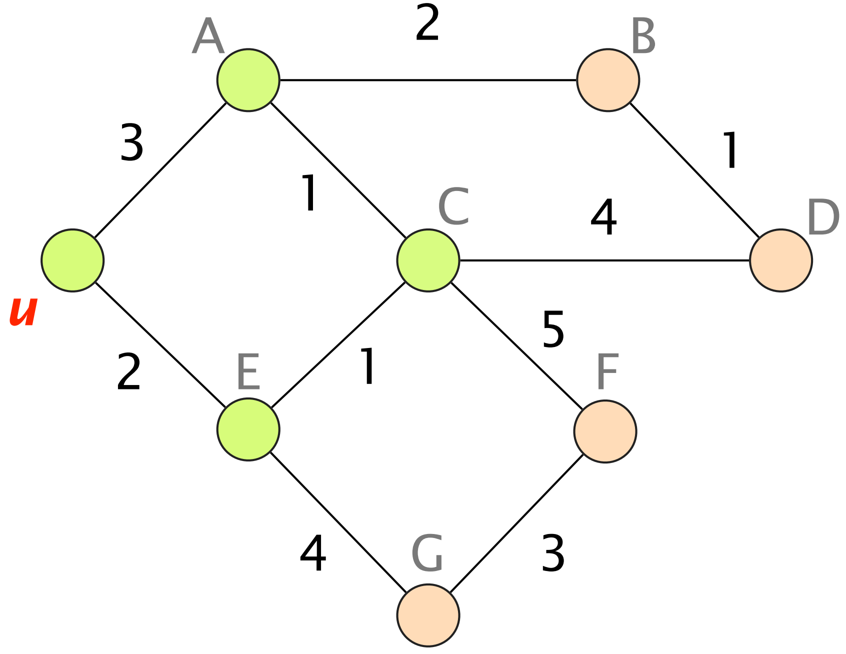
A	3
B	∞
C	∞
D	∞
E	2
F	∞
G	∞



D(.) = S = {u, E}

A	3	
B	∞	
C	3	$D(v) = \min\{\infty, 2 + 1\}$
D	∞	
E	2	
F	∞	
G	6	$D(v) = \min\{\infty, 2 + 4\}$

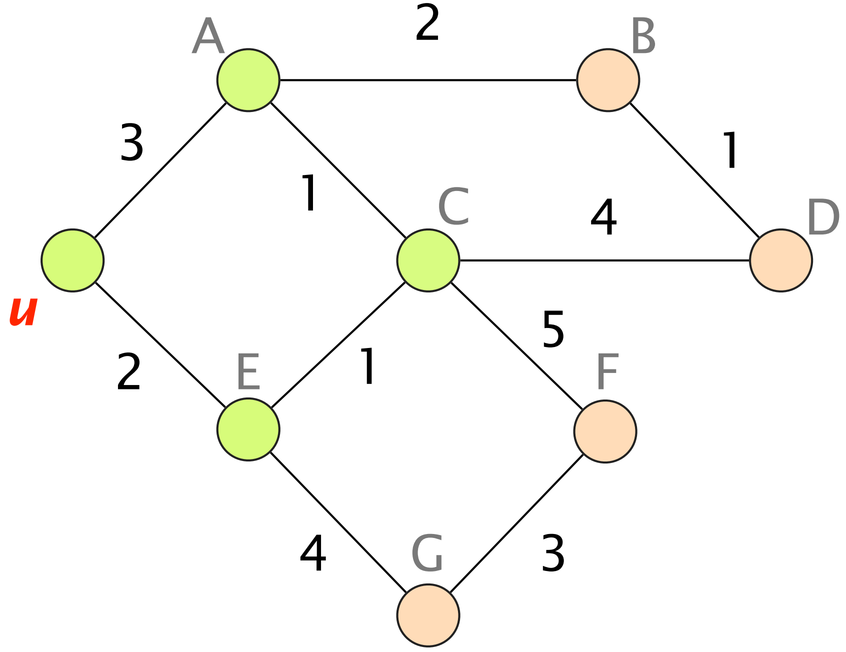
Skipping a few steps…



D(.) =

S = {u, E, A, C}

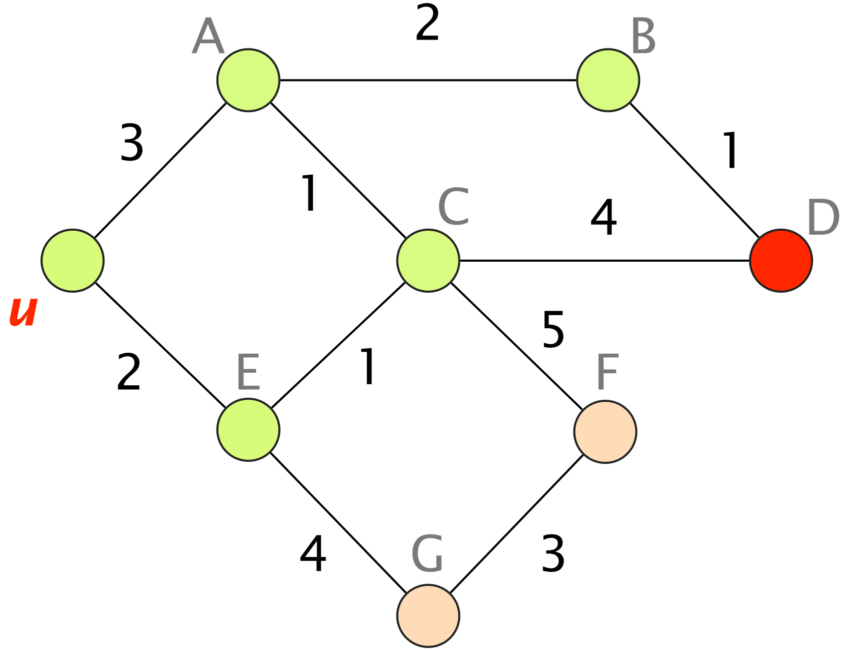
A	3
B	5
C	3
D	7
E	2
F	8
G	6



D(.) =

S = {u, E, A, C}

A	3	
B	5	smallest D(w)
C	3	
D	7	
E	2	
F	8	
G	6	



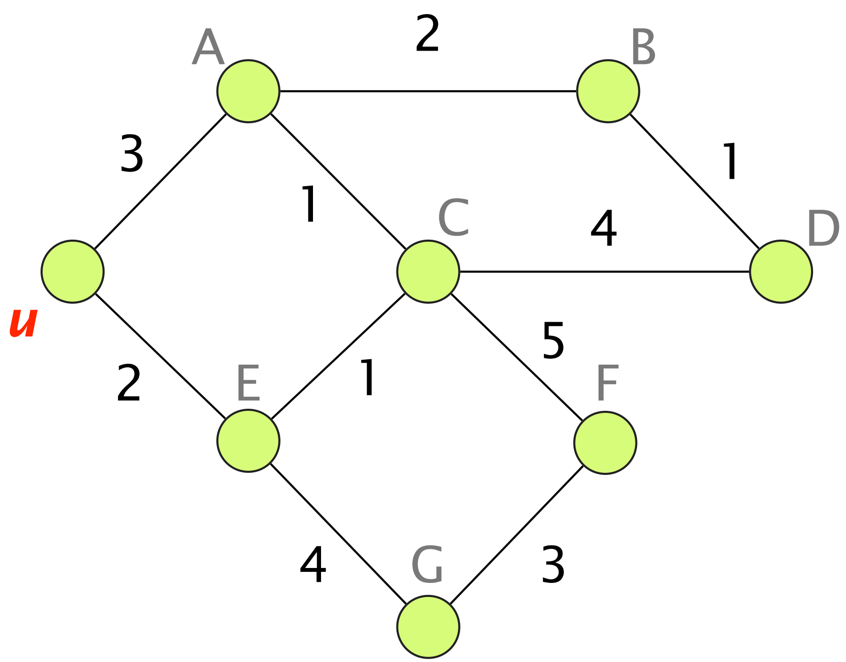
D(.) =

S = {u, E, A, C, B}

A	3
B	5
C	3
D	6
E	2
F	8
G	6

D(v) = min{7, 5 + 1}

Here is the final state



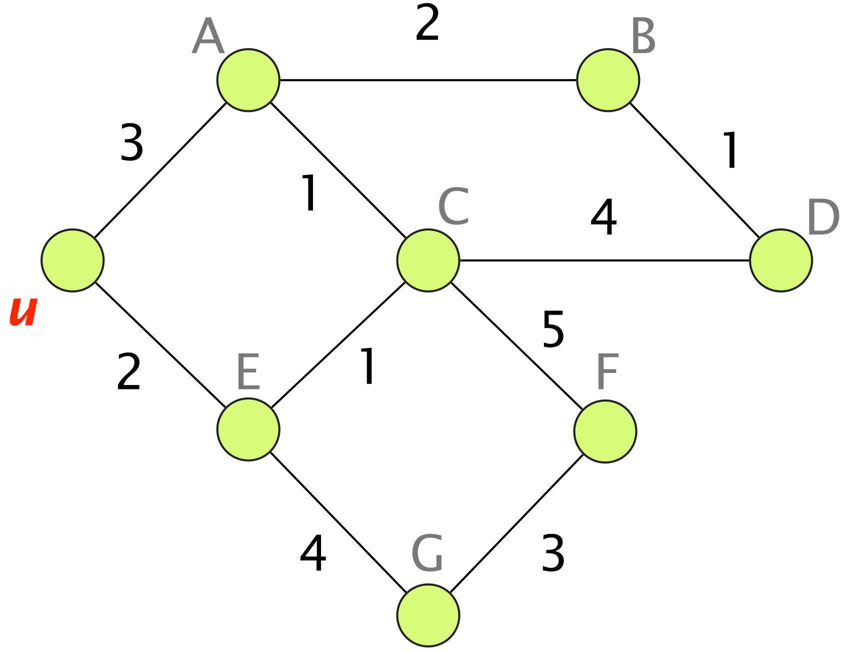
D(.) =

A	3
B	5
C	3
D	6
E	2
F	8
G	6

S = {u, A,
B, C, D, E,
F,G}

From the shortest-paths,

u can directly compute its forwarding table



Forwarding table

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

Communication Networks

Exercise 2

Quick Dijkstra repetition

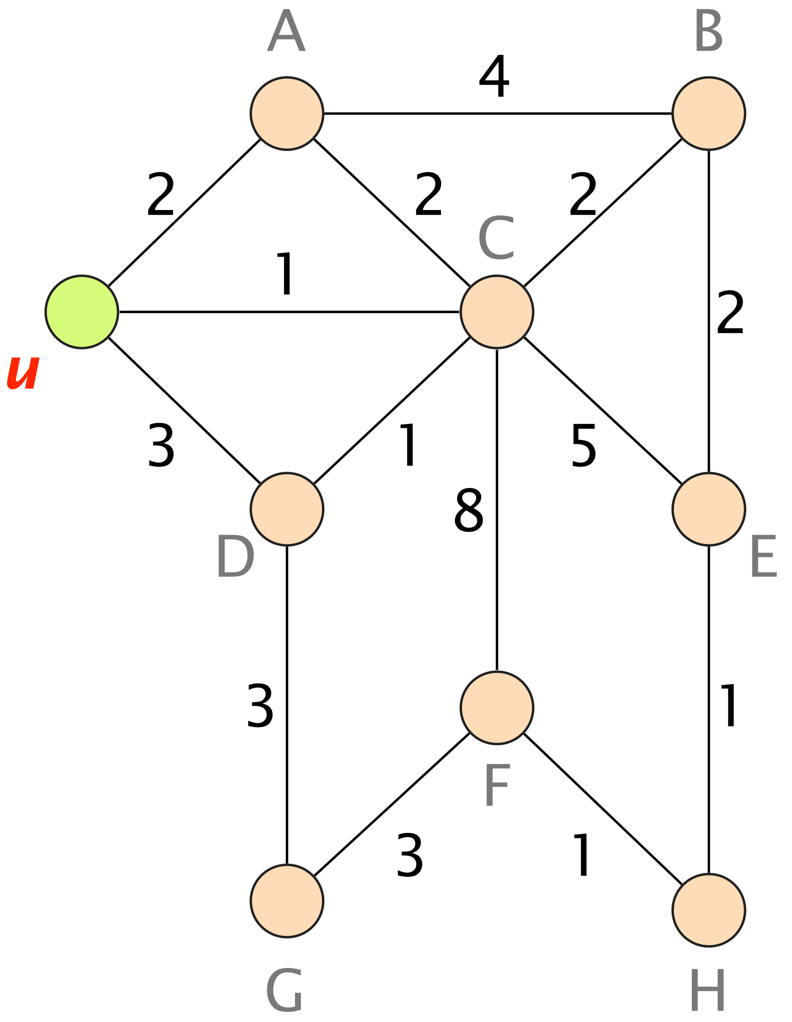
Overview current assignment

Questions?

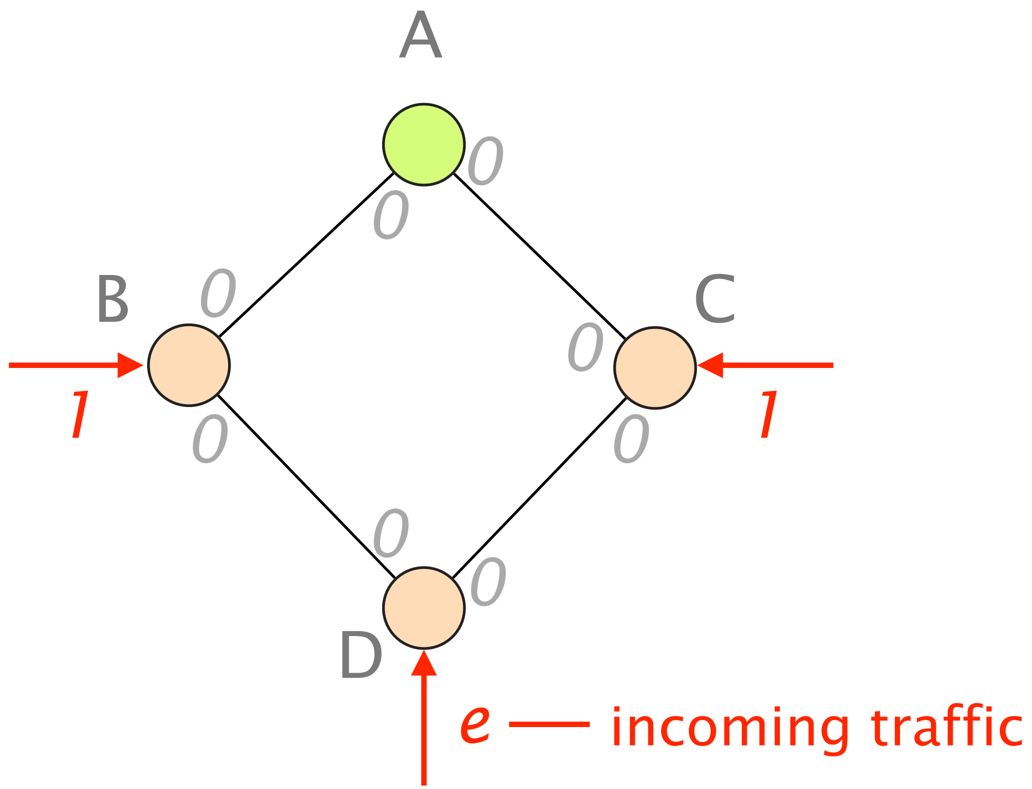
Time for you to solve the tasks

Solutions will be published next week

Task 1: Dijkstra’s algorithm



Task 2: Changing weights

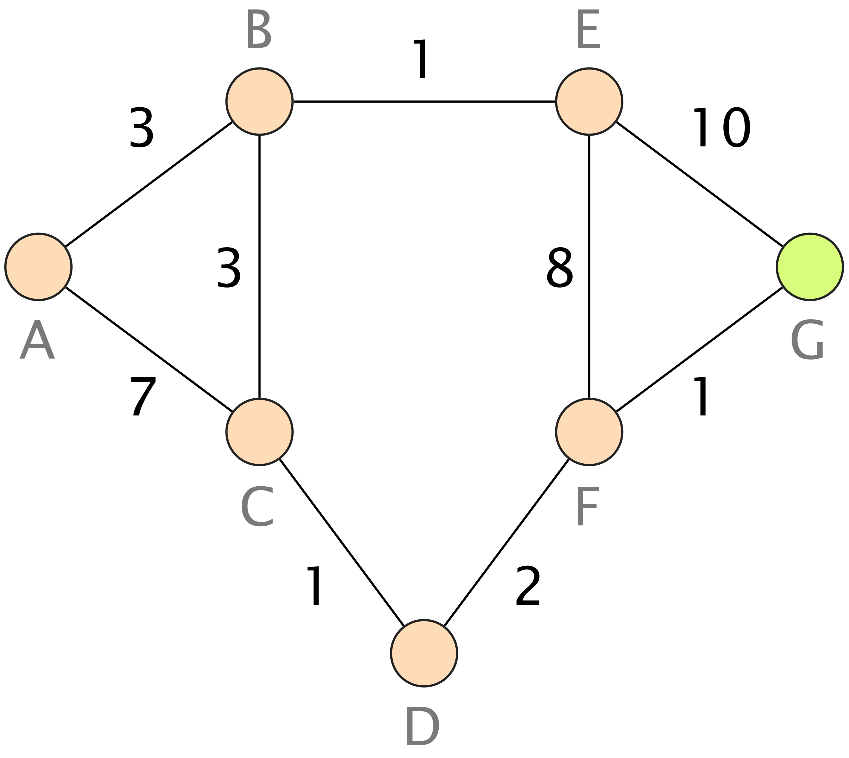


Consider dynamic weights

Important: different weights for both link directions

Next-hop as tie-break value

Task 3: Distance Vector



Now we look at a distance vector approach (Dijkstra: link-state)

Each node sends a distance vector repeatedly to all neighbors

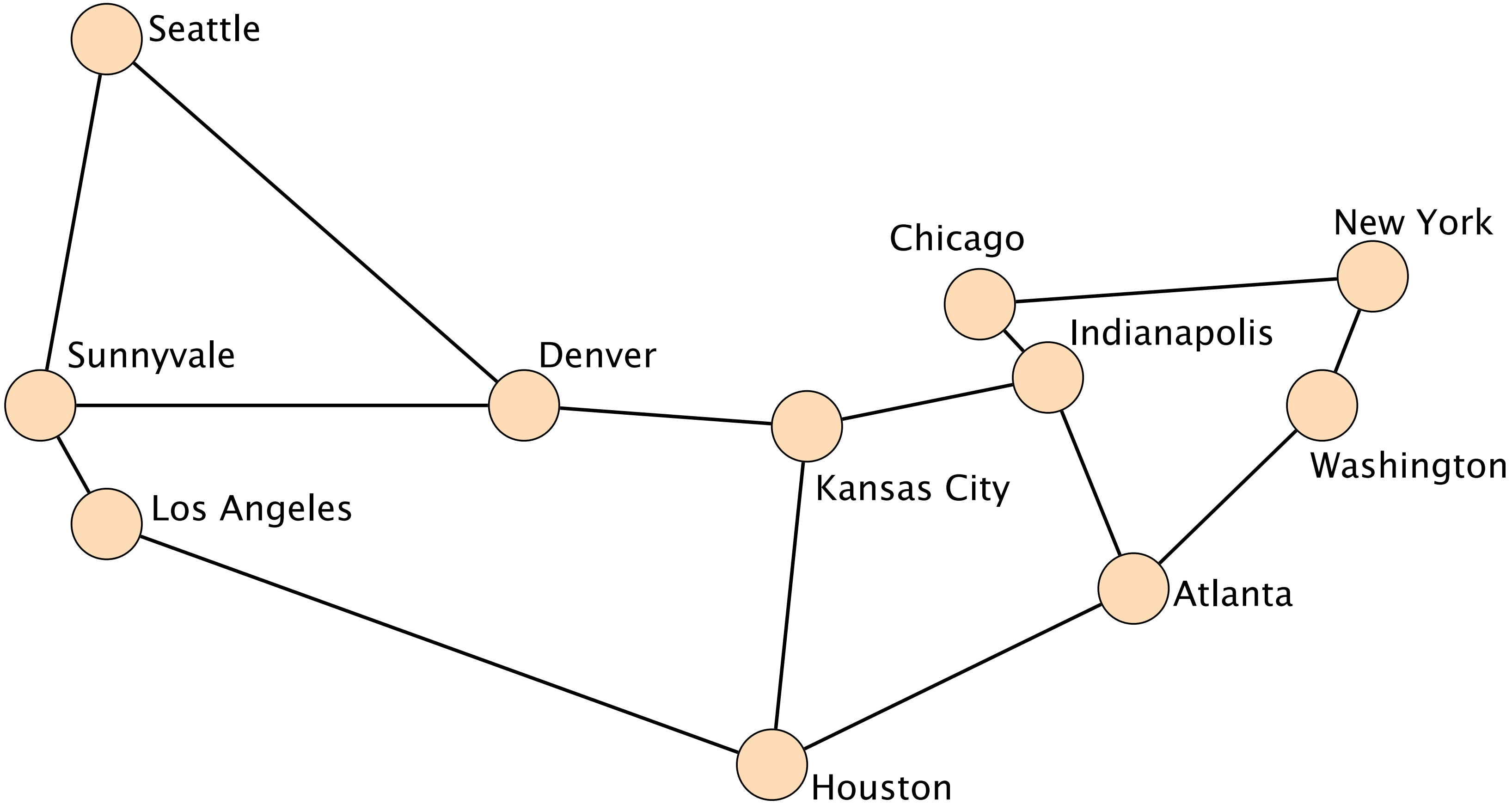
No global network view required

Task 4: Source-and-Destination-Based Routing

Is it possible to route packets based on the source address?

What are advantages/disadvantages?

Task 5: Link Weight Configuration



The Abilene network in the US

Communication Networks

Exercise 2

Quick Dijkstra repetition

Overview current assignment

Questions?

Time for you to solve the tasks

Solutions will be published next week

Group registration is open

Register on https://comm-net.ethz.ch/registration

Enter your nethz usernames

Please build groups of three students

Join the *#group_search* Slack channel if you need more members

“Internet Hackathon”

March 26, 6 - 10 pm

Communication Networks

Exercise 2

Quick Dijkstra repetition

Overview current assignment

Questions?

Time for you to solve the tasks

Solutions will be published next week