L16: Routing & BGP

CS202 - Computer Systems

Today we will discuss routing in general and Internet routing in particular.

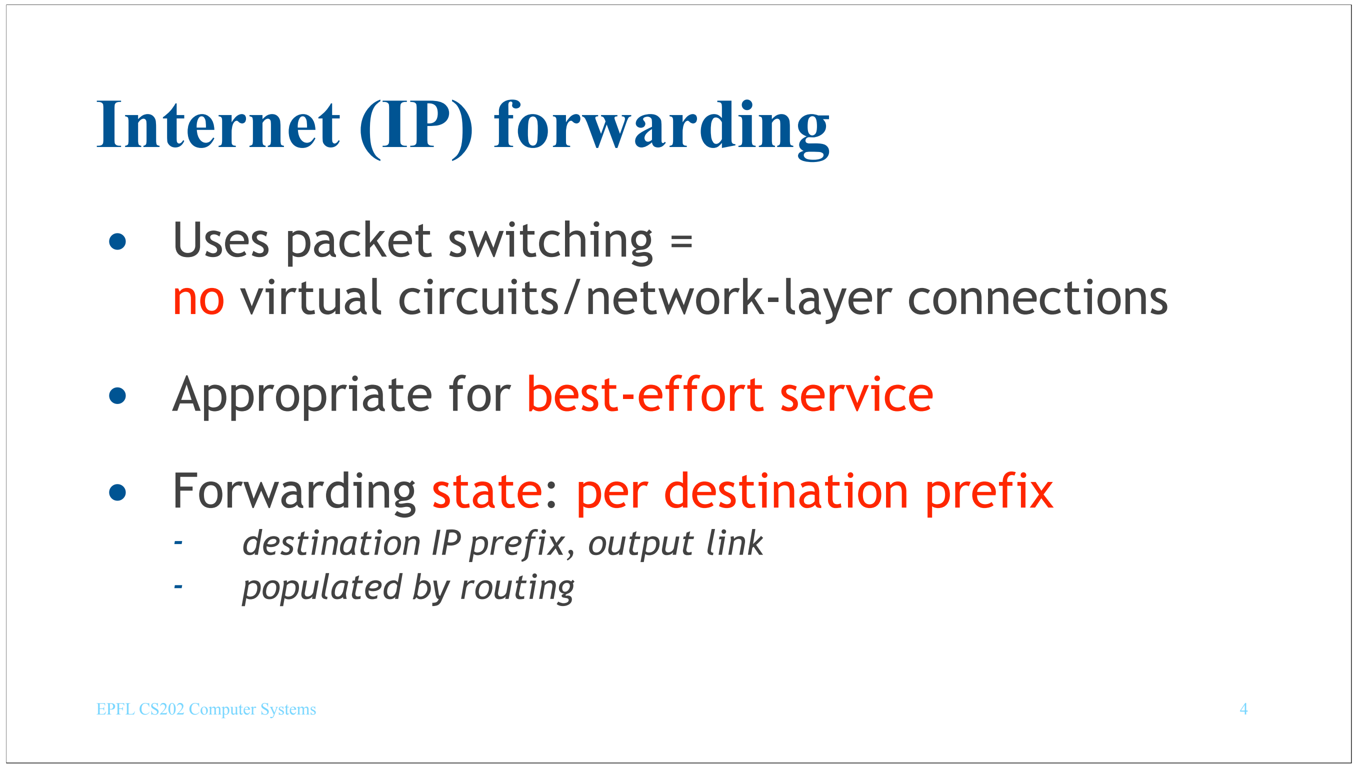
Last time, we said that the network layer performs two basic operations: forwarding and routing.

Forwarding

- Local operation that takes place at a router whenever a new packet arrives; its goal: determine the packet's output link
- How: read field from packet's network-layer header; use the value to search the forwarding table for the correct output link

Routing

- Network-wide operation that populates forwarding tables
- How: routing algorithm run on a logically centralised network controller or the routers themselves



Internet (IP) forwarding

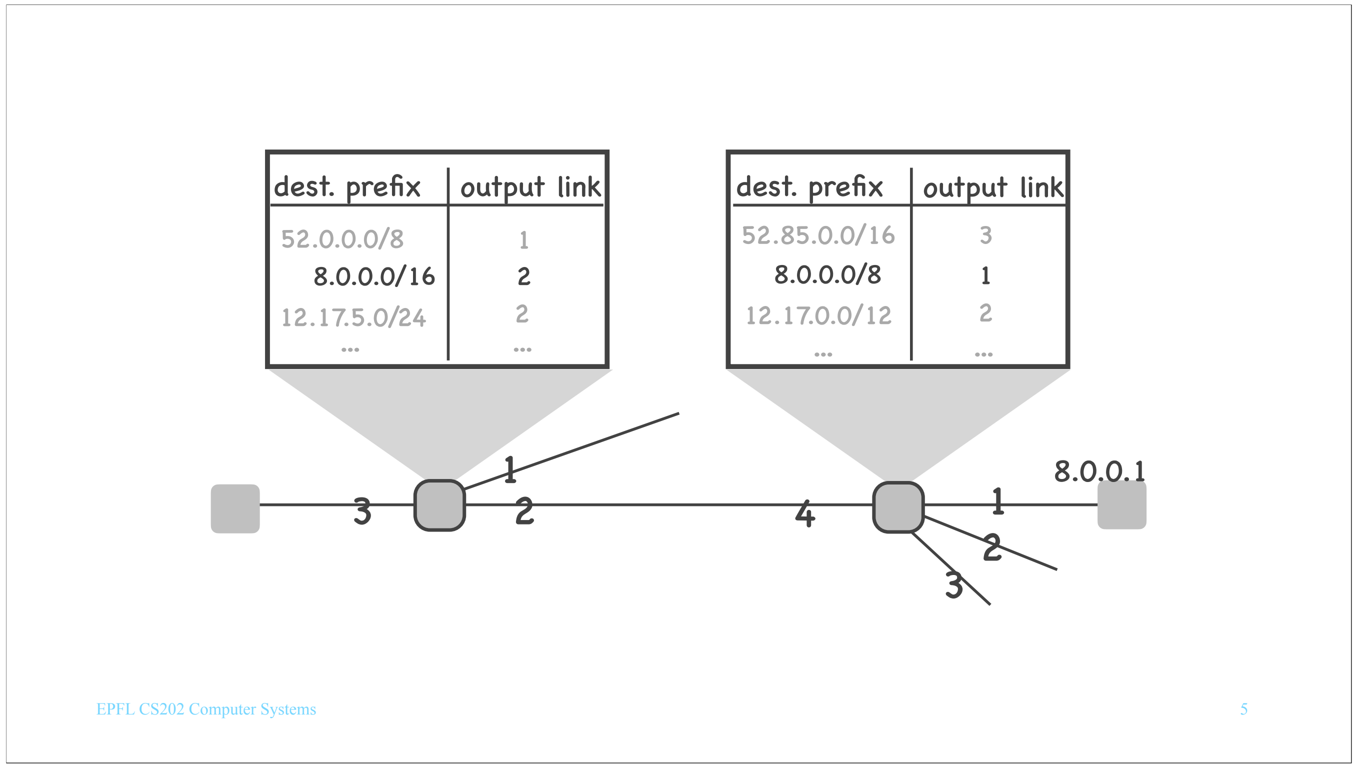
- Uses packet switching =
  no virtual circuits/network-layer connections
- Appropriate for best-effort service
- Forwarding state: per destination prefix
  - *destination IP prefix, output link*
  - *populated by routing*

We said that the Internet network layer performs pure packet switching.
It does not use virtual circuits (which are also called network-layer connections).

This is appropriate for a network layer that offers best-effort service,
meaning it does not guarantee delivery or performance.

We discussed a lot the forwarding state (=information) that each router keeps in its forwarding table.
We said that each entry consists of a destination IP prefix and the corresponding output link.

We didn’t really say how many entries each forwarding table has;
we only said that they may be few or many, depending on how nicely IP addresses can be grouped in ranges…
Today we will talk more concretely about the size of forwarding tables.



dest. prefix	output link
52.0.0.0/8	1
8.0.0.0/16	2
12.17.5.0/24	2
...	...

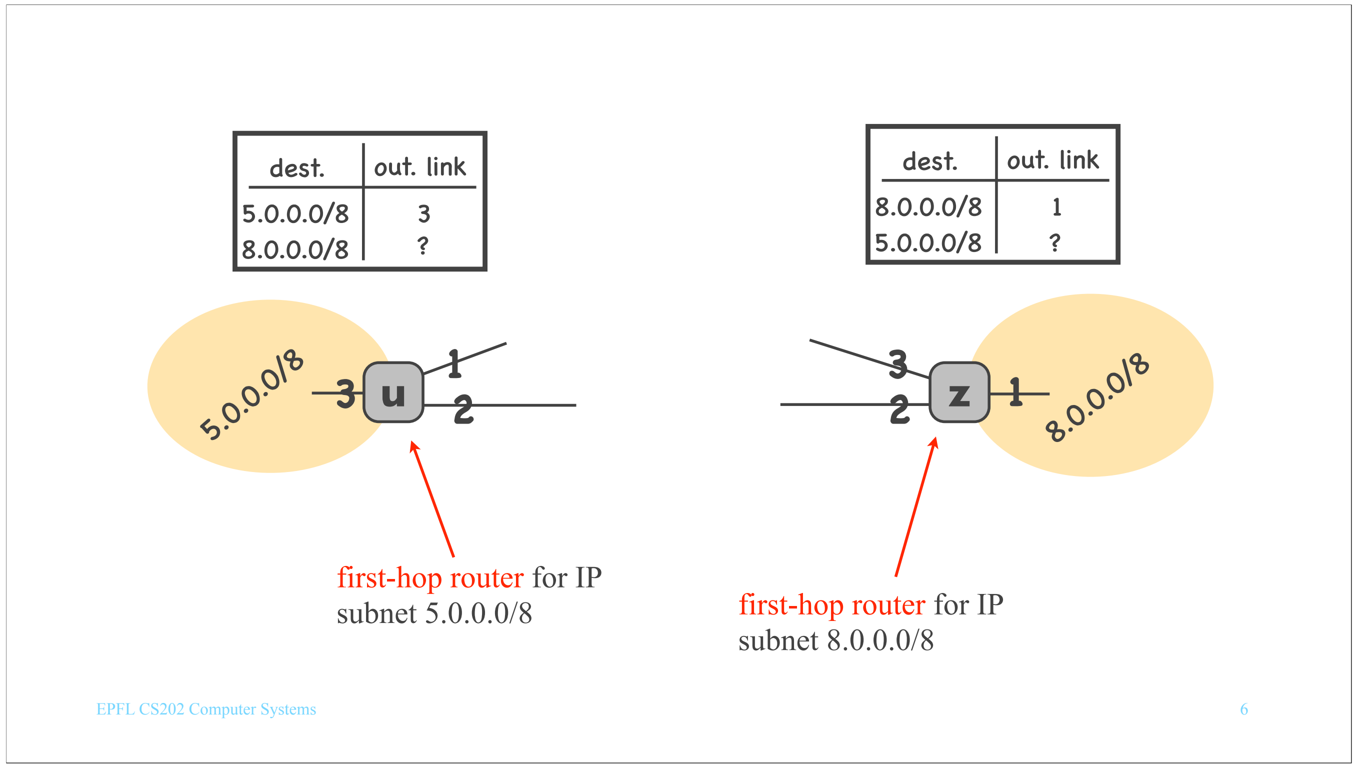
dest. prefix	output link
52.85.0.0/16	3
8.0.0.0/8	1
12.17.0.0/12	2
...	...

Consider a source and a destination end-system, the latter with IP address 8.0.0.1.

Every router on the Internet has, in its forwarding table, an IP prefix that matches 8.0.0.1 and indicates the correct output link for reaching that end-system.

In fact, every router on the Internet has, in its forwarding table, an IP prefix that matches *any* public Internet IP address.

The purpose of routing is to populate the forwarding tables of routers.



dest.	out. link
5.0.0.0/8	3
8.0.0.0/8	?

dest.	out. link
8.0.0.0/8	1
5.0.0.0/8	?

first-hop router for IP subnet 5.0.0.0/8

first-hop router for IP subnet 8.0.0.0/8

Suppose we have two IP subnets:

- the one on the left, with IP prefix 5.0.0.0/8,
- and the one on the right, with IP prefix 8.0.0.0/8.

Suppose each of these IP subnets is attached to a single router: u and z, respectively.

We say that router u is the “first-hop router” for the IP subnet on the left: it is the first router that handles packets coming from this subnet.

Similarly, router z is the “first-hop router” for the IP subnet on the right.

Each router knows how to forward packets addressed to its own subnet.

E.g., router u has, in its forwarding table, an entry, which says that, any packet with destination IP address matching 5.0.0.0/8 must be forwarded through output link 3 (which is connected to the router’s local subnet).

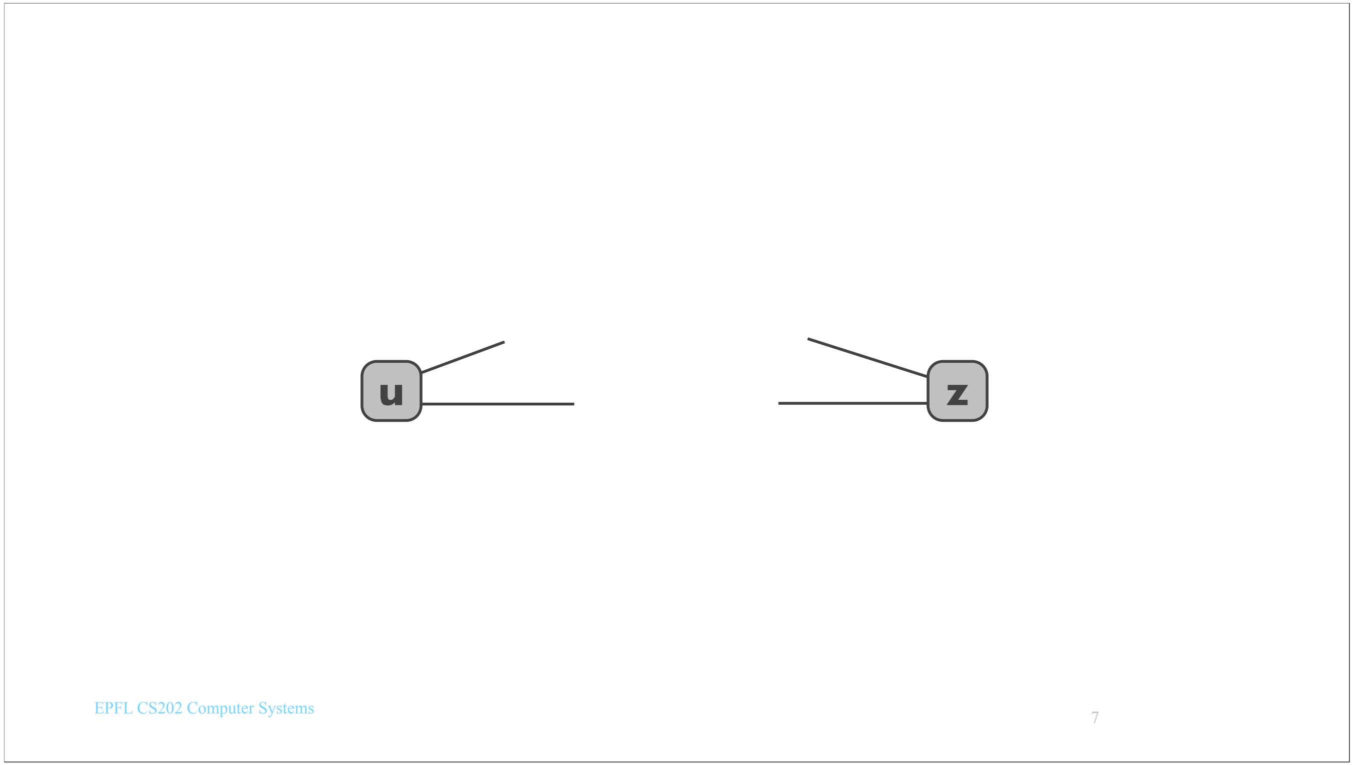
Similarly, router z has, in its forwarding table, an entry, which says that, any packet with destination IP address matching 8.0.0.0/8 must be forwarded through output link 1 (which is connected to the router’s local subnet).

These entries are typically setup when the network administrator configures each router.

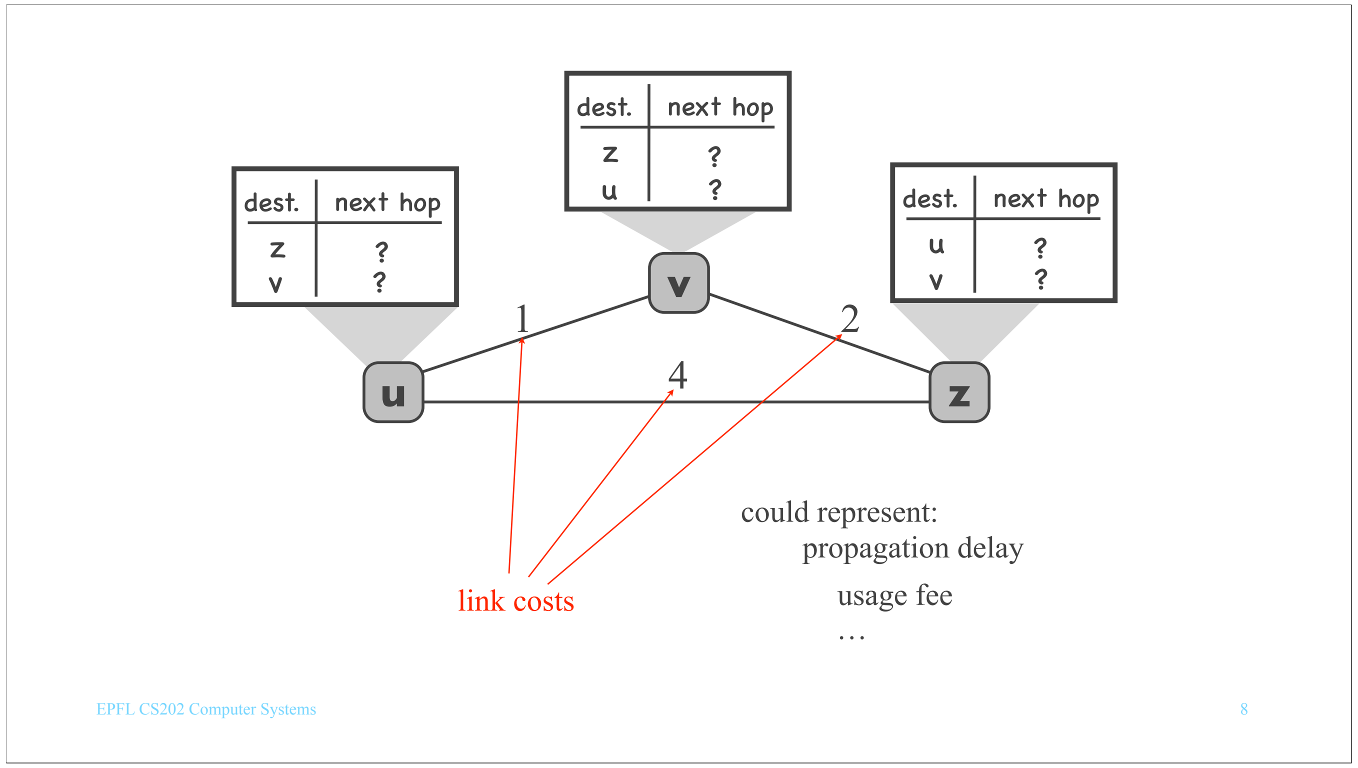
But what about packets addressed to foreign subnets?

E.g., how does router u learn how to forward packets addressed to router z’s subnet and vice versa? Who adds the second entry to each of our example forwarding tables?

A routing protocol.



Let’s forget about IP subnets for a moment.



Suppose routers u and z are part of a 3-router topology.

When these 3 routers participate in a routing protocol,
their goal is to learn the best path to each other.

For example, router u on the left, when it wants to send a packet to router z, how should it do that? Send the packet to z through the direct link that connects them? Or through router v? How about when it wants to send a packet to router v?

The other two routers, v and z, need to answer similar questions.

But what does "best path" mean?

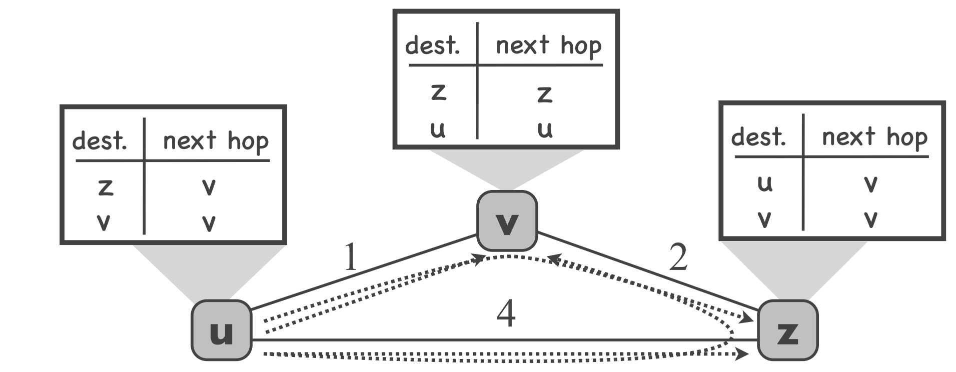
Each link has a cost, which represents the link's "badness." E.g., the cost could be computed based on the link's propagation delay, or how much money it costs to send traffic over that link, or some other cost metric.

(Parenthesis: In all the examples I will show you, a link's cost is the same in both directions. In reality, however, links can have different characteristics, hence different costs, in each direction.)

The cost of a path is equal to the sum of the costs of its links.

The best path from a source router to a destination router is the path with the least cost.

Btw, the data structures I show here are *not* forwarding tables; they are auxiliary data structures that I will use to illustrate various routing protocols.



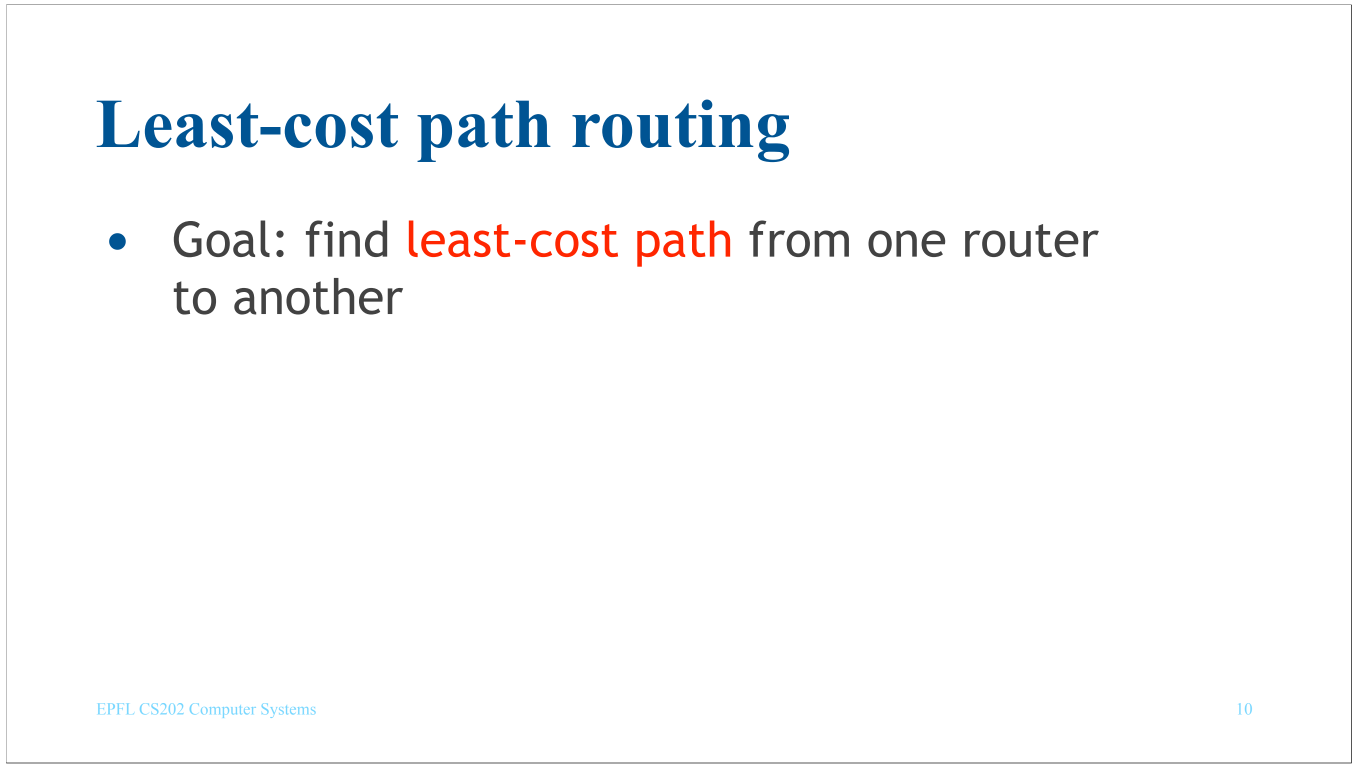
least-cost path from u to z: u v z

least-cost path from u to v: u v

From router u to router z, there are two paths: a direct one, of cost 4, and an indirect one, through router v, of cost 1+2=3. The least-cost path is the indirect one.
Hence, from the point of view of router u, the best next hop for reaching router z is router v.

From router u to router v, there are also two paths: a direct one, of cost 1, and an indirect one, of cost 6. The least-cost path is the direct one.
Hence, from the point of view of router u, the best next hop for reaching router v is v itself.

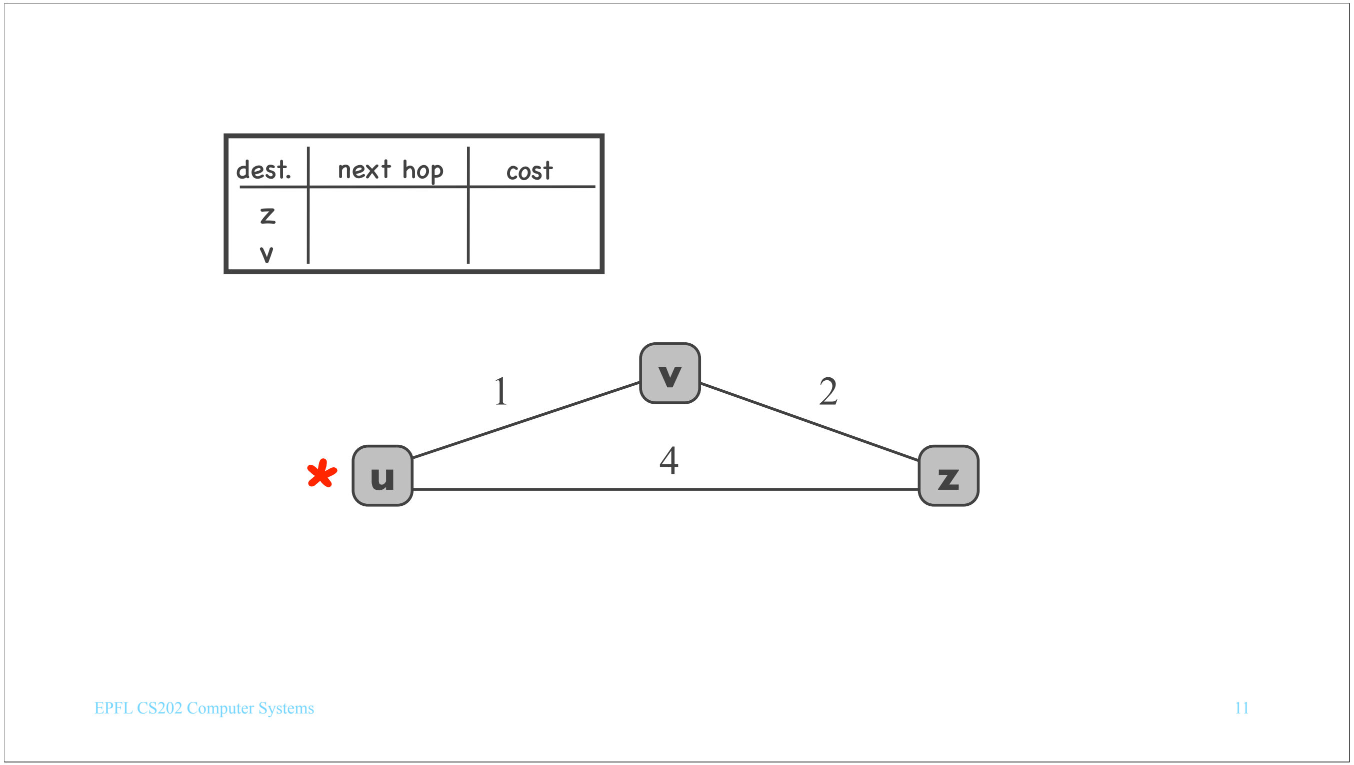
With similar reasoning, we can fill the tables of routers z and v.



Least-cost path routing

- Goal: find least-cost path from one router to another

So, the goal of least-cost path routing is to find the least-cost path (or least-cost route) from one router to another.



dest.	next hop	cost
z		
v		

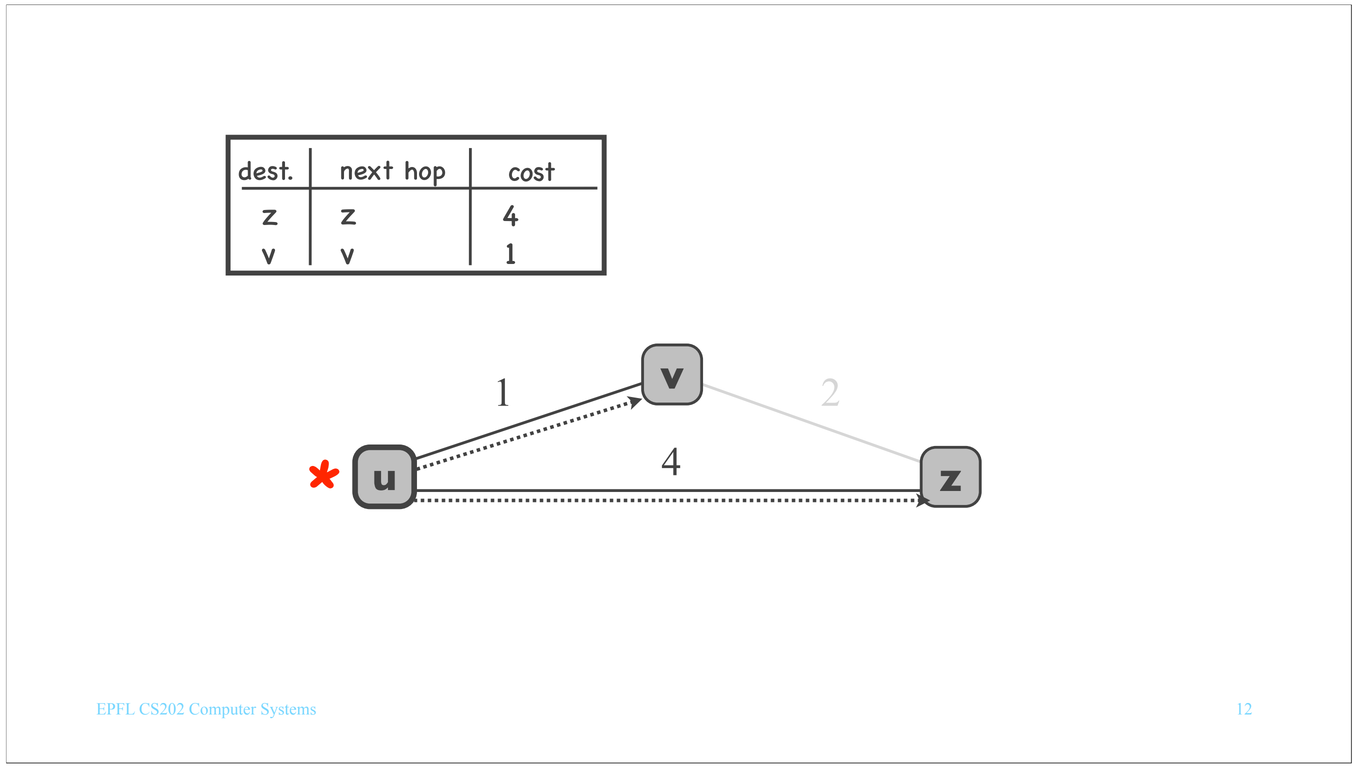
It’s time to dive in and study a representative routing algorithm.

In the first algorithm we will see:
- first, *all* routers exchange information about their adjacent links;
- once each router has computed the network graph (with link costs), it runs this algorithm to compute the best path to each of the other routers.

So, each router runs its own separate instance of the algorithm. But given that they all provide the same input to the algorithm (the same network graph and link costs), they all reach compatible conclusions.

Let’s now focus on router u and see what happens when router u runs the algorithm.

The algorithm works in steps.



dest.	next hop	cost
z	z	4
v	v	1

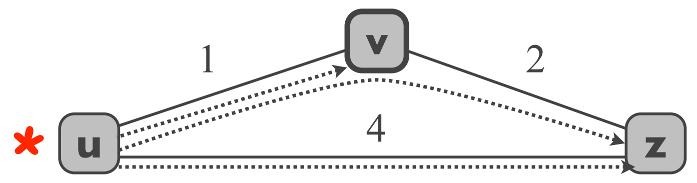
In the first step, the algorithm considers only the links that are directly connected to router u. It forgets about the rest of the network (it's as if the other links do not exist).

Considering *only* these two direct links,
the algorithm fills its table as follows:

- It asks: is there a path from router u to router z?
- The answer is: yes, there exists a direct path, the next-hop is router z itself, and the cost is 4.

- Then it asks: is there a path from router u to router v?
- The answer is: yes, there exists a direct path, the next-hop is router v itself, and the cost is 1.

dest.	next hop	cost
z	~~z~~ v	~~4~~ 3
v	v	1



In the second step, the algorithm chooses a neighbor of router u — in our example, router v, and considers the links that are directly connected to that neighbor as well.

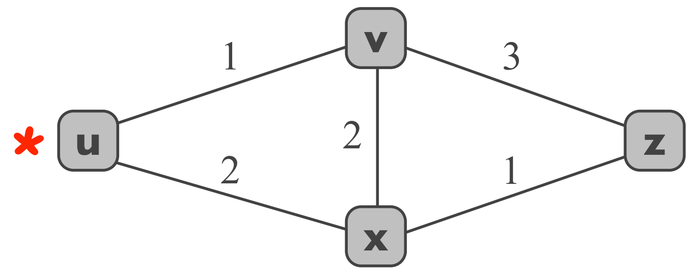
The algorithm asks: can router u improve its existing paths by routing through router v?

Let's consider the path to destination z:
- Can router u improve its path to destination z by routing through v?
- Yes, indeed, it can.
- So, the algorithm removes the old, direct path from u to z and adds a new path through v.
- Now let's update the table: for destination router z, the new next hop is router v, and the new cost is 3.

Now let's consider the path to destination v:
- Can router u improve its path to destination v by routing through router v?
- No, because the path to destination v is already through router v itself, so there is nothing to improve.

At this point, the algorithm is done:
it has computed the least-cost path from router u to every other router in the network.

dest.	next hop	cost
z		
v		
x		



Let’s look at a second example.

Again, we will concentrate on router u.