Link-state vs. distance-vector

- Link state: each entity first obtains complete view of the network, then computes the least-cost paths
- Distance vector: each entity obtains incrementally new information about the network at every round

differently: …

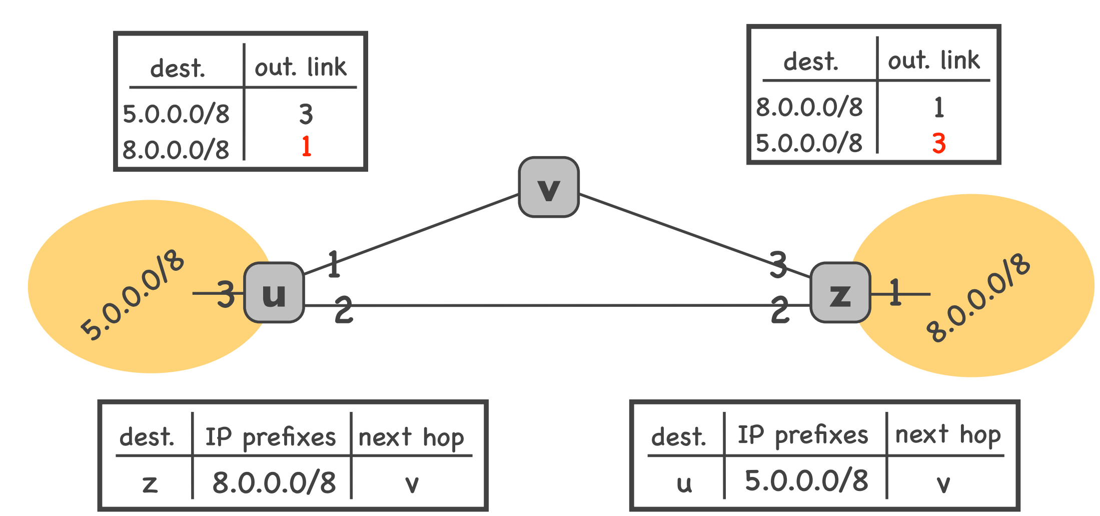
Which one is better and in what scenario? Which one do you think finishes faster? Which one requires more resources?

Link-state vs. distance-vector

- Link-state converges faster
  - *each router starts with full picture of the network*
- Distance-vector uses less bandwidth
  - *each router only talks to its neighbors*

Link-state converges faster: We can always reduce its computation time by running it on a faster computer. In contrast, in distance-vector, a new piece of information may take multiple rounds to propagate from router x to router y (as many rounds as there are routers between x and y).

But distance-vector requires fewer messages: In each round, each router exchanges a message with each neighbor (whatever the size of the network). In contrast, in link-state, each router essentially floods the entire network with information about its adjacent links.



Now let us go back to this picture that we introduced in the beginning of the lecture.

In this example, router u knows how to route packets to IP prefix 5.0.0.0/8, because this is the IP prefix of the local IP subnet, but it does not know how to route packets to IP prefix 8.0.0.0/8. How does routing help router u complete its forwarding table?

Router u, together with routers v and z participate in a routing protocol, which could be Dijkstra or Bellman-Ford or any other routing protocol. Through this routing protocol, router u learns that the best next hop to router z is router v. However, through this protocol, router u also learns that router z “owns” IP prefix 8.0.0.0/8, so, any packet whose destination IP address matches 8.0.0.0/8 should be routed to router z. Combining these two pieces of information, router u maps IP prefix 8.0.0.0/8 to output link 1, and it can now complete its forwarding table.

Similarly, router z, learns, through the same routing protocol, that the best next hop to router u is router v. And it also learns that router u owns IP prefix 5.0.0.0/8. Combining these two pieces of information, router z maps IP prefix 5.0.0.0/8 to output link 3, and it can now complete its forwarding table.

Routing algorithms/protocols

- Each router advertizes which IP prefixes it "owns"
- Routers exchange information about network topology, link costs, and potentially their current routes
- Each router learns how to route to every public IP address in the world

Internet routing challenges

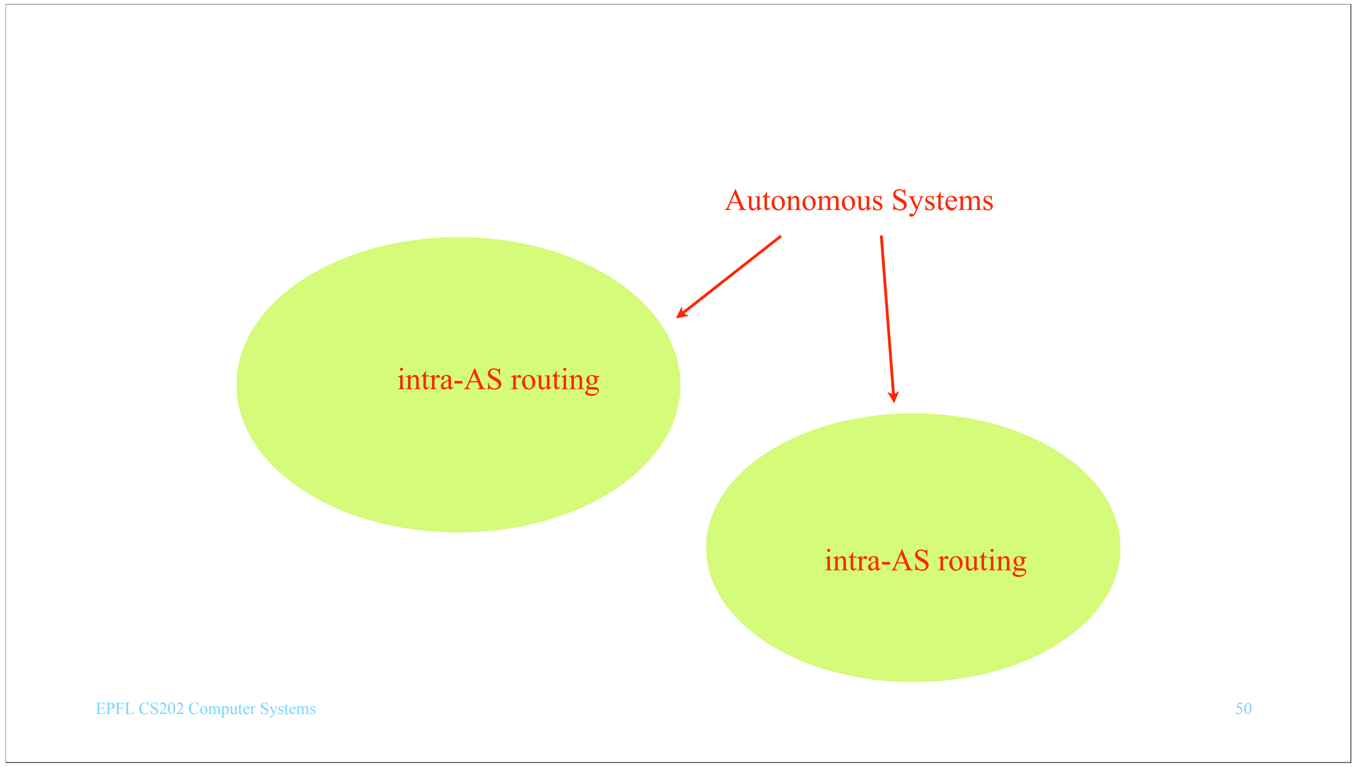
- Scale
  - *link-state would cause flooding*
  - *distance-vector would not converge*
  - *too many IP subnets => too many forwarding-table entries*
- Administrative autonomy
  - *an ISP may not want to do least-cost routing*
  - *may want to hide its link costs from the world*

We will close our discussion on routing with how actual Internet routing works.

Designing a routing algorithm for the Internet faces at least two challenges:

One is scale:
- We are talking about the entire Internet -- millions of routers.
- A link-state routing algorithm, like Dijstra's algorithm, would cause flooding, because it requires each router to directly exchange information with every other router in the network.
- A distance-vector algorithm, like Bellman-Ford, would not converge fast enough, because it requires multiple rounds of information exchange between neighbors until every router collects all the information it needs from every other router in the network.

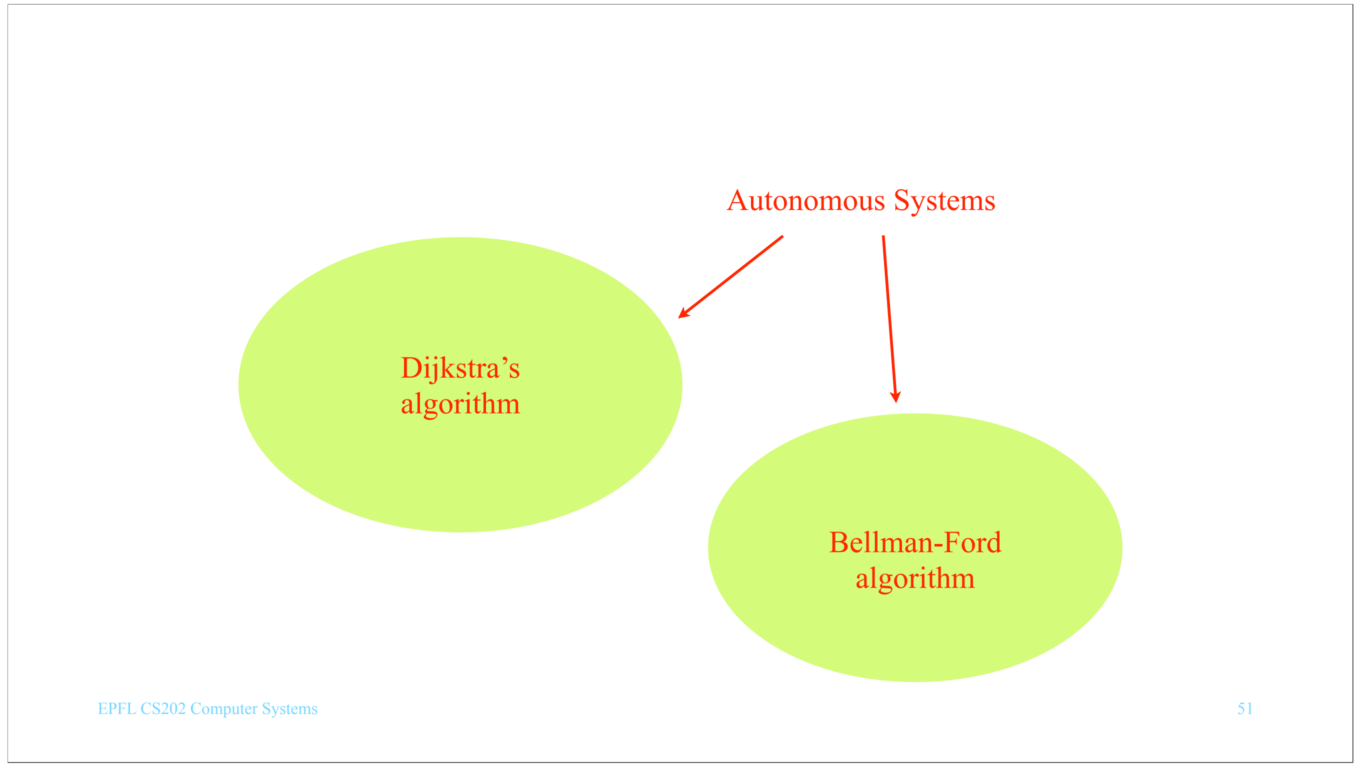
The other challenge is administrative autonomy: In the first lecture, we said that each Internet service provider, or ISP, is typically a separate administrative entity. So, an ISP may not want to apply least-cost path routing inside its own network; or, it may want to hide the characteristics of its internal links from the rest of the world.



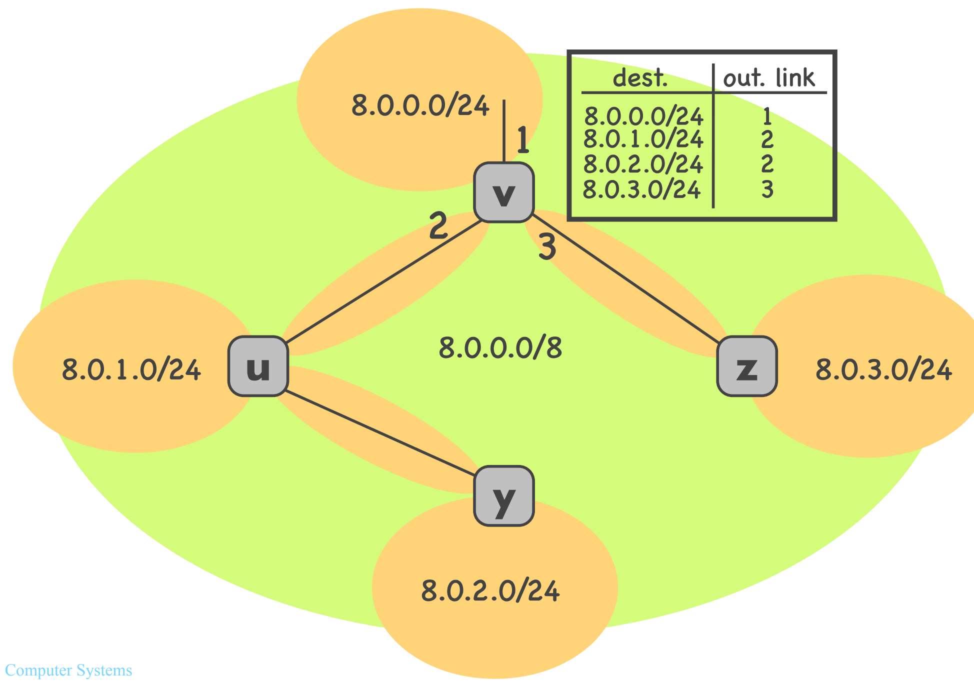
The current Internet architecture addresses these two challenges through *hierarchy*.

In particular, the Internet is divided in separate networks, which are called “Autonomous Systems” or ASes.

Each AS runs its own separate routing algorithm among its local routers. We refer to these algorithms as “intra-AS routing algorithms,” because each algorithm involves only routers from a single AS.



Each AS may run any intra-AS routing algorithm it wants, e.g., Dijkstra or Bellman-Ford.



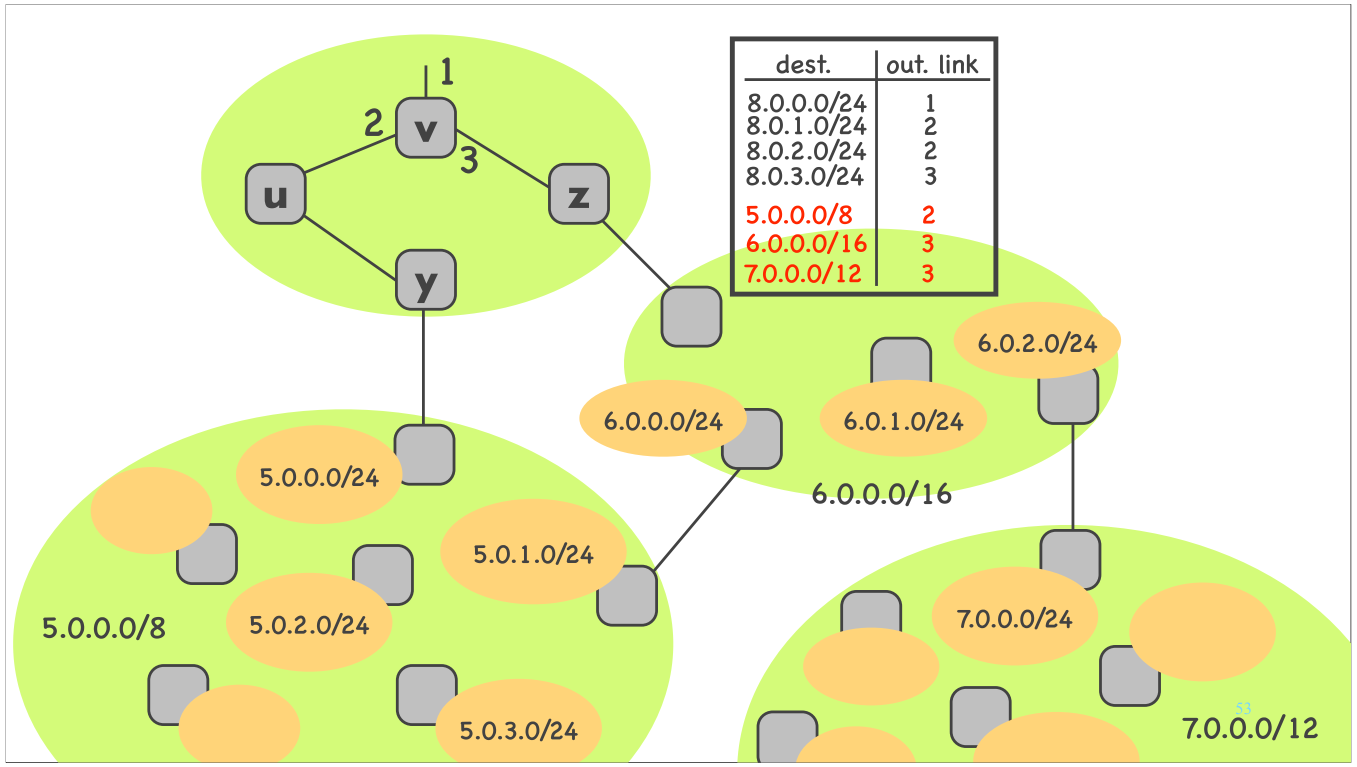
Consider an AS that has 4 routers, each providing connectivity to 1 IP subnet.

Each of the 4 routers must discover the best path to each local router/subnet.

They achieve this by participating in an intra-AS routing protocol.

The slide shows what the forwarding table of router v might look like.

(There are also small subnets that correspond to the links between routers, but I am not showing these, to keep the picture simple.)



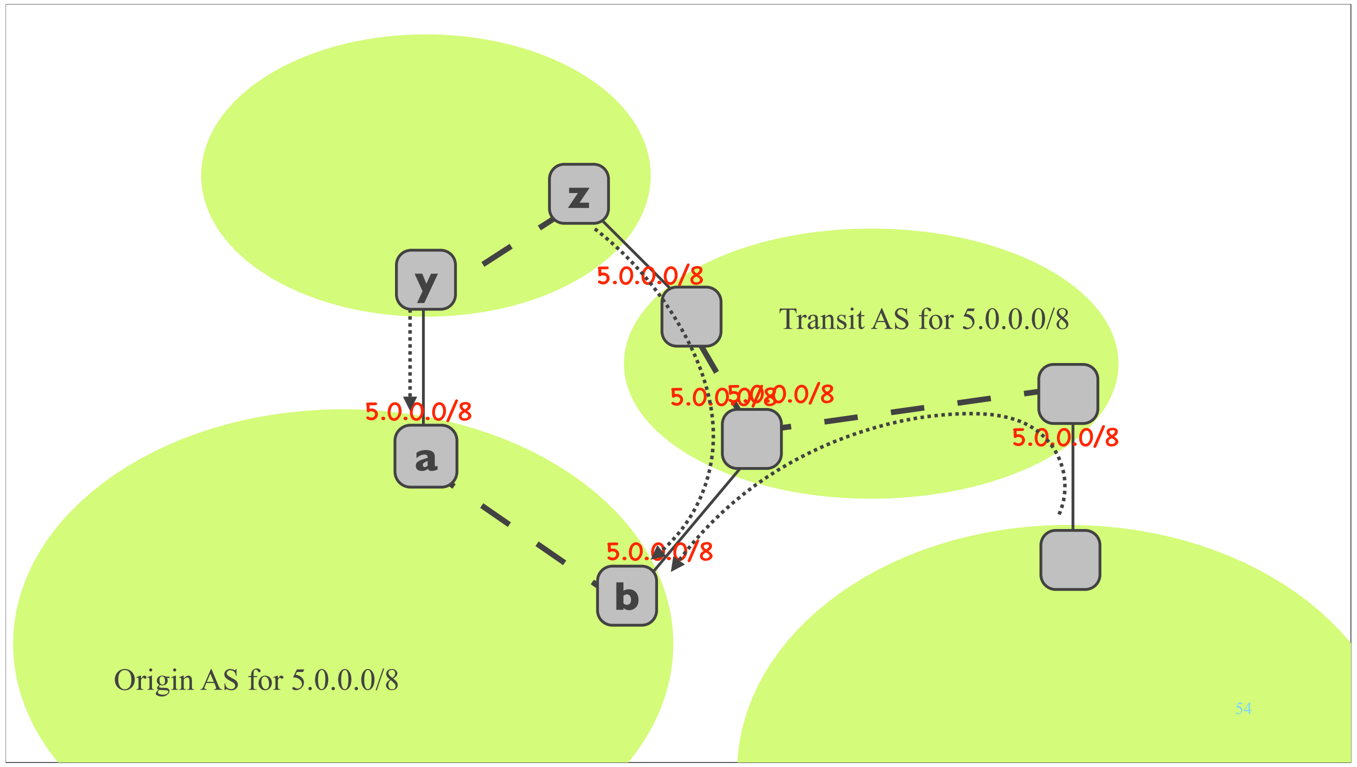
Moreover, each of the 4 routers must discover the best path, not only to each of the local routers but also to each foreign AS.

However, they do not need to discover a path to each individual IP subnet in each foreign AS.

For instance, in this example topology, router v (whose forwarding table is shown on the slide) needs only 1 entry overall for each of the 3 foreign ASes shown in the picture.

This is an important way in which Internet routing scales: A router does not need an entry, in its forwarding table, for every single IP subnet on the entire Internet. It typically merges the many little IP prefixes of each foreign AS into one big IP prefix or a few big IP prefixes.

We said that a router discover routes to the local routers through intra-AS routing.
How does it discover routes to foreign ASes?
That’s done through…



…the inter-domain routing algorithm/protocol.

This is a routing algorithm run by all the *border routers* of all ASes (the routers shown in this picture).

The border routers are the routers that are at the "edge" of each AS and are directly connected to (border) routers of other ASes.

All the border routers participate in an instance of a routing algorithm, called Border Gateway Protocol (BGP), which is a variant of the Bellman-Ford algorithm we saw earlier.

Through BGP, each border router (a) advertizes prefixes from its own AS to the outside world, and (b) it learns the best route to each foreign prefix.

I will illustrate with an example:

Each of the two border routers of the bottom left AS aggregates all the IP prefixes of the local IP subnets into one aggregate IP prefix (5.0.0.0/8).

We say that this AS is the "origin AS" for 5.0.0.0/8.

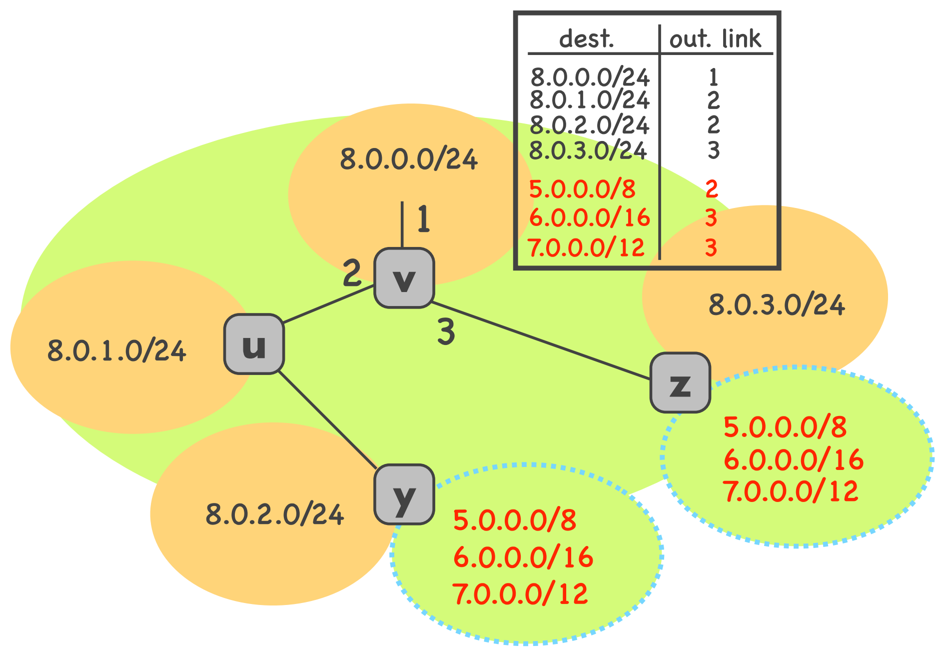
Router a (a border router of this AS) advertises to its neighbor router y that it has a route to 5.0.0.0/8.

At this point, router y has learned a route to 5.0.0.0/8 (it knows where to forward packets with a destination IP address that belongs to this IP prefix).

Moreover, router b (another border router of the bottom left AS) advertises to its neighbor router that it has a route to 5.0.0.0/8.

Suppose the top right AS decides that it will do transit for 5.0.0.0/8, so, it propagates the route to other ASes (the bottom right and top left AS).

In the end, router z also learns a route to 5.0.0.0/8.



From the point of view if router v:

v participates only in intra-domain routing (it's not a border router). So, it learns the best route to each local prefix (from the other local routers) AND it learns the best route to each foreign prefix (from the border routers, which are y and z).

If both y and z have routes to the same prefix, v will choose the least-cost route.

In a way, as local routers act as gateways to local subnets/prefix, border routers act as gateways to foreign ASes/prefixes.

(It is possible for a router to do both. E.g., y is the gateway to 8.0.2.0/24 AND to foreign prefixes.)

Internet routing

- Each router learns one route to each IP subnet in local AS
- Each router learns one or a few routes to each foreign AS
  - *but not one route to each IP subnet of each foreign AS*

Intra-AS routing

- Run by all routers in the same AS
- Goal: propagate routes within local AS
  - *each router advertizes routes to its local IP subnets*
  - *and potentially routes to other ASes that it has learned through BGP*
- OSPF, RIP, …

Inter-AS routing

- Run by all border routers between ASes
- Goal: propagate routes outside local AS
  - *each border router talks to external neighbors (eBGP)*
  - *and to the other border routers of the local AS (iBGP)*
- BGP = Border Gateway Protocol

There is 1 inter-AS routing protocol on the Internet: BGP.

There has to be 1, if we want to ensure that all ASes can communicate with each other, i.e., discover paths to each other.