CSCI 466: Networks

Network Layer – Routing (Control Plane)

Reese Pearsall

Fall 2022

Announcements

NO CLASS on Monday (10/24)

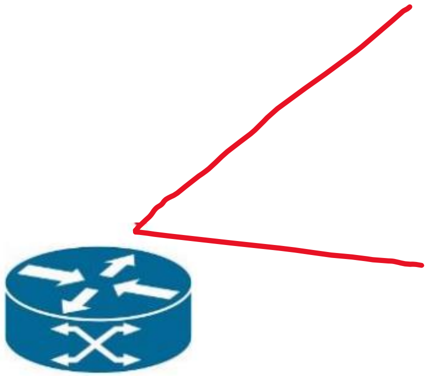
- Email/DM me if you need to chat or have questions

PA2 due **tonight*** at 11:59 PM

→ You can submit it any time this weekend without a late pass and you won't lose points ☺

HW2 Released. Due one week from today (10/28)

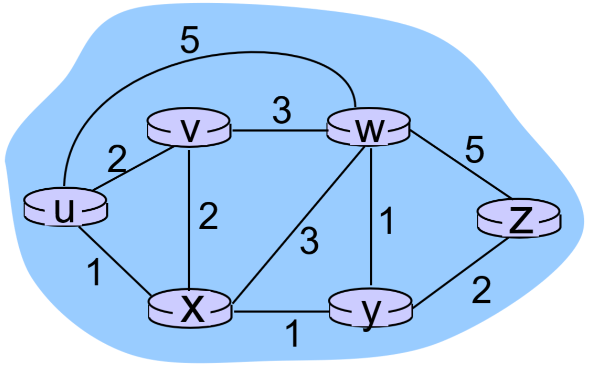
Forwarding refers to moving packets from a **router's input** to appropriate **router output**, and is implemented in the data plane.



Address range	Interface (output link)
128.11.52.0 – 128.11.52.255	1
153.90.2.0 – 153.90.2.255	2
153.90.2.87 – 153.90.2.89	3

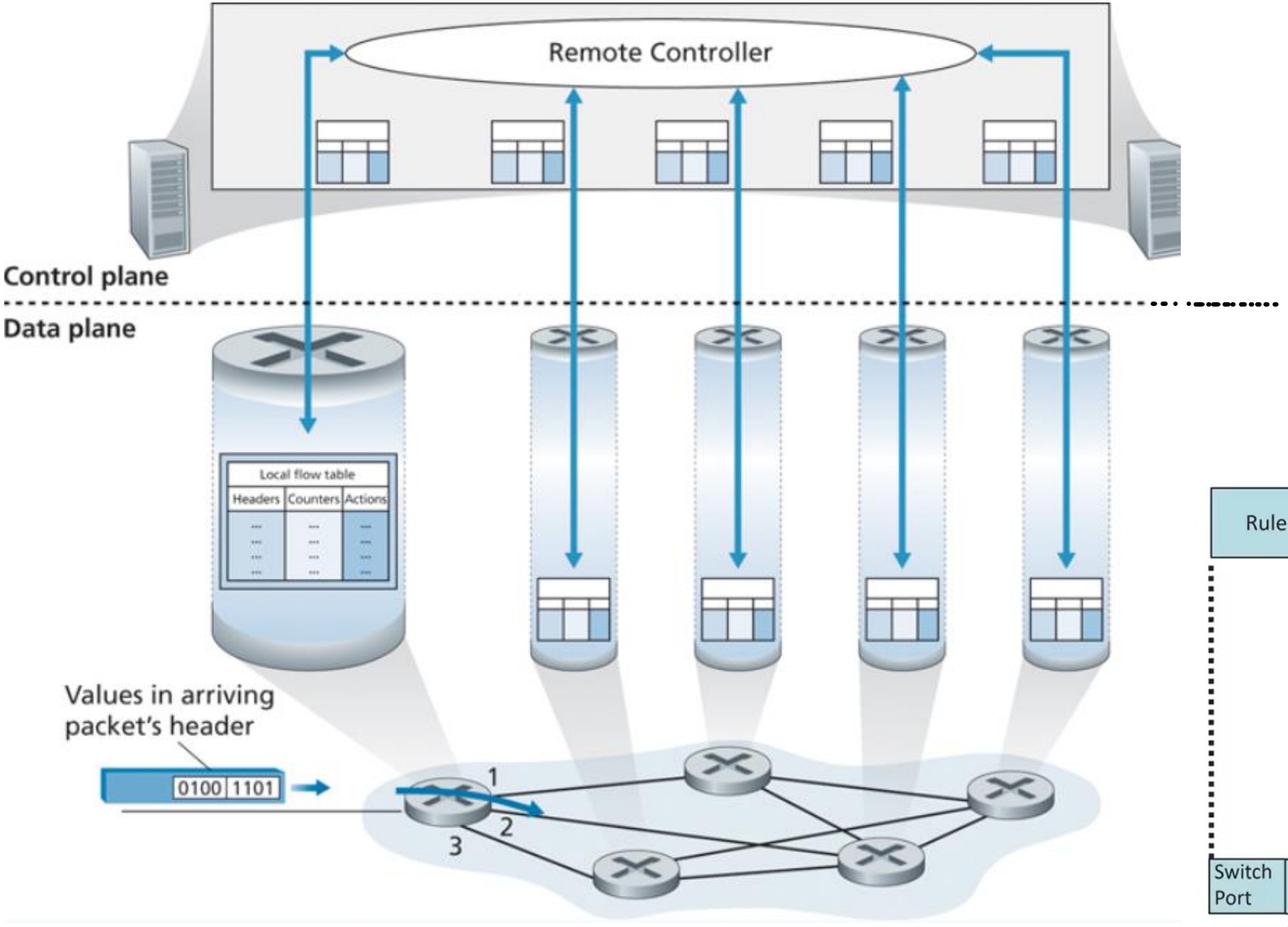
Ideally, this output links are the most optimal path to get to the destination

Routing refers to determining the route taken by packets from **source** to **destination**, and is implemented in the control plane.

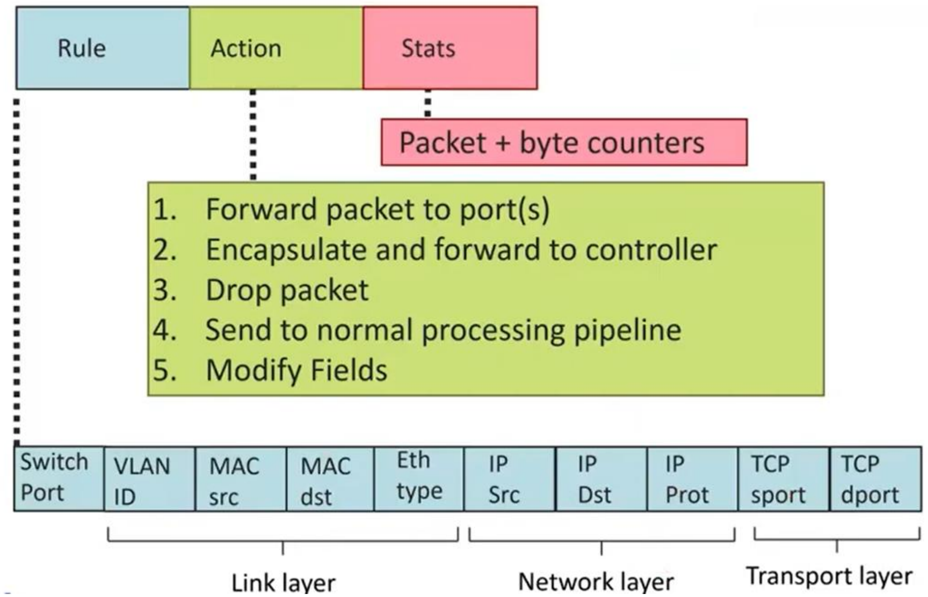


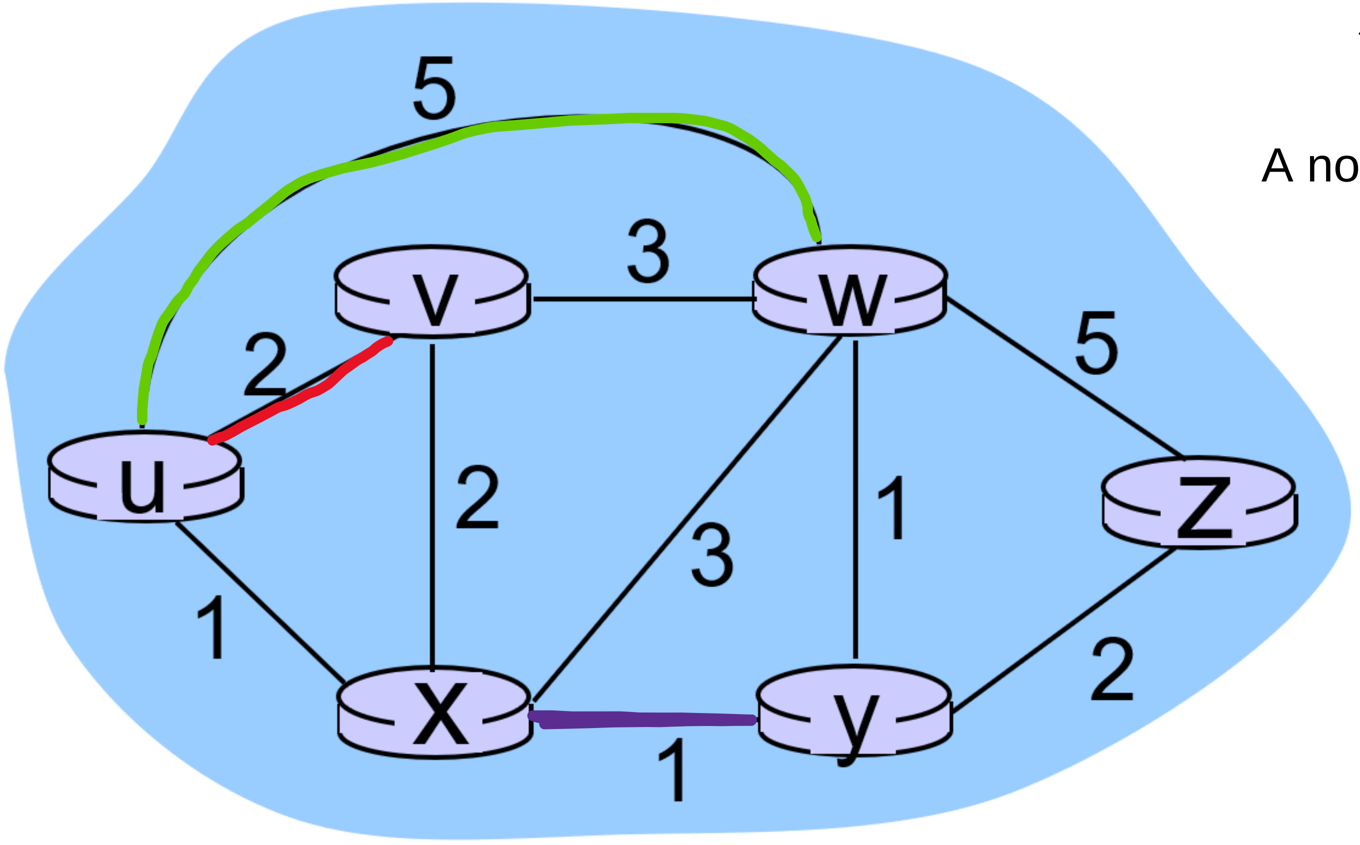
What is the best way to get from **u** to **z**?

Generalized Forwarding and Software Defines Network (SDN)



Control Plane and Data Plane need to work together in order to create efficient routing tables





An example network with 6 hosts/routers

What do the costs represent??

A graph G consists of N nodes

N = { u v x w y z }

A node could be a host, a router, or a smaller network!

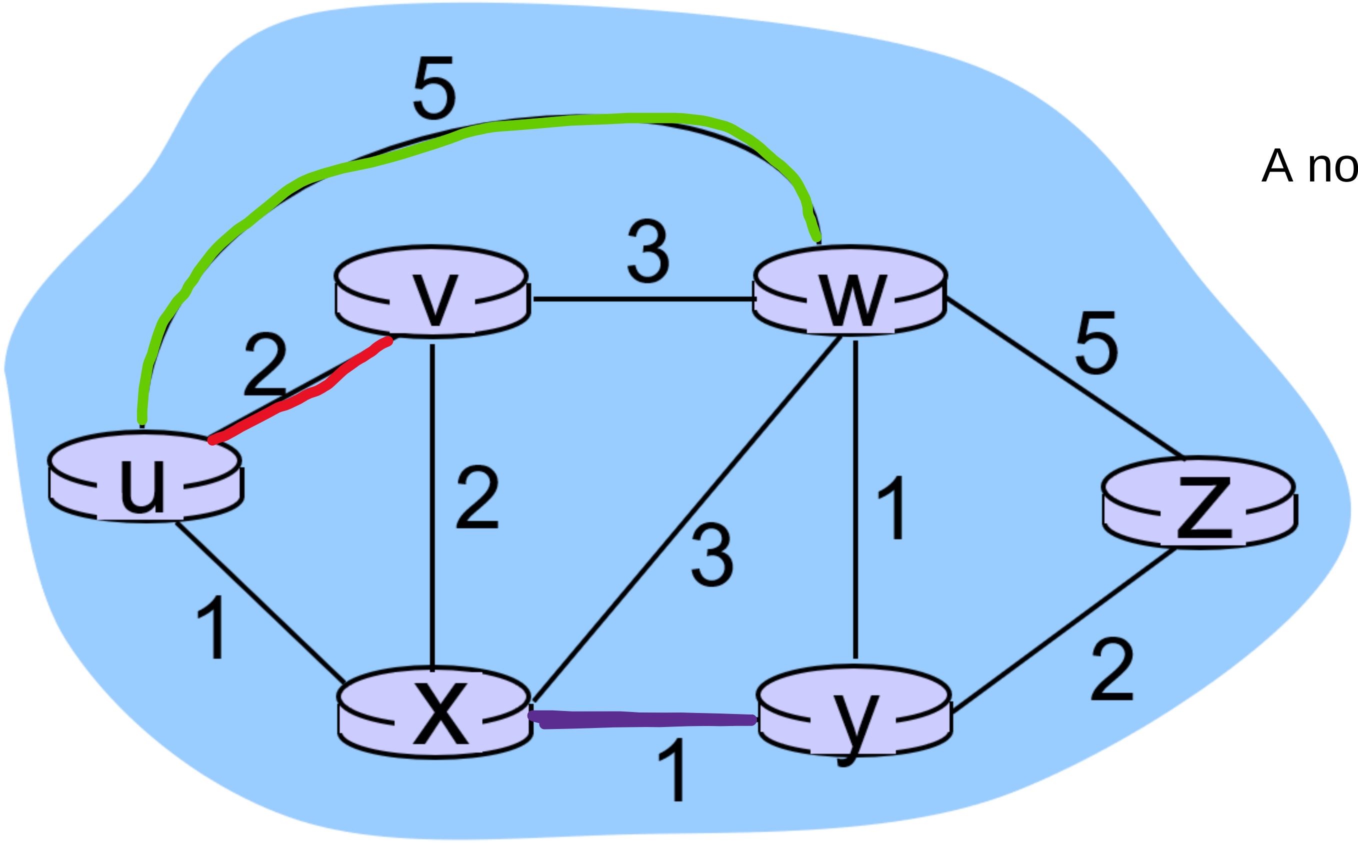
A graph consists of edges E

E_1 (u,v) = 2

E_2 (u,w) = 5

E_6 (x,y) = 1

In most cases, edges will also have a **cost**



An example network with 6 hosts/routers

A graph G consists of N nodes

N = { u v x w y z }

A node could be a host, a router, or a smaller network!

A graph consists of edges E

E_1 (u,v) = 2

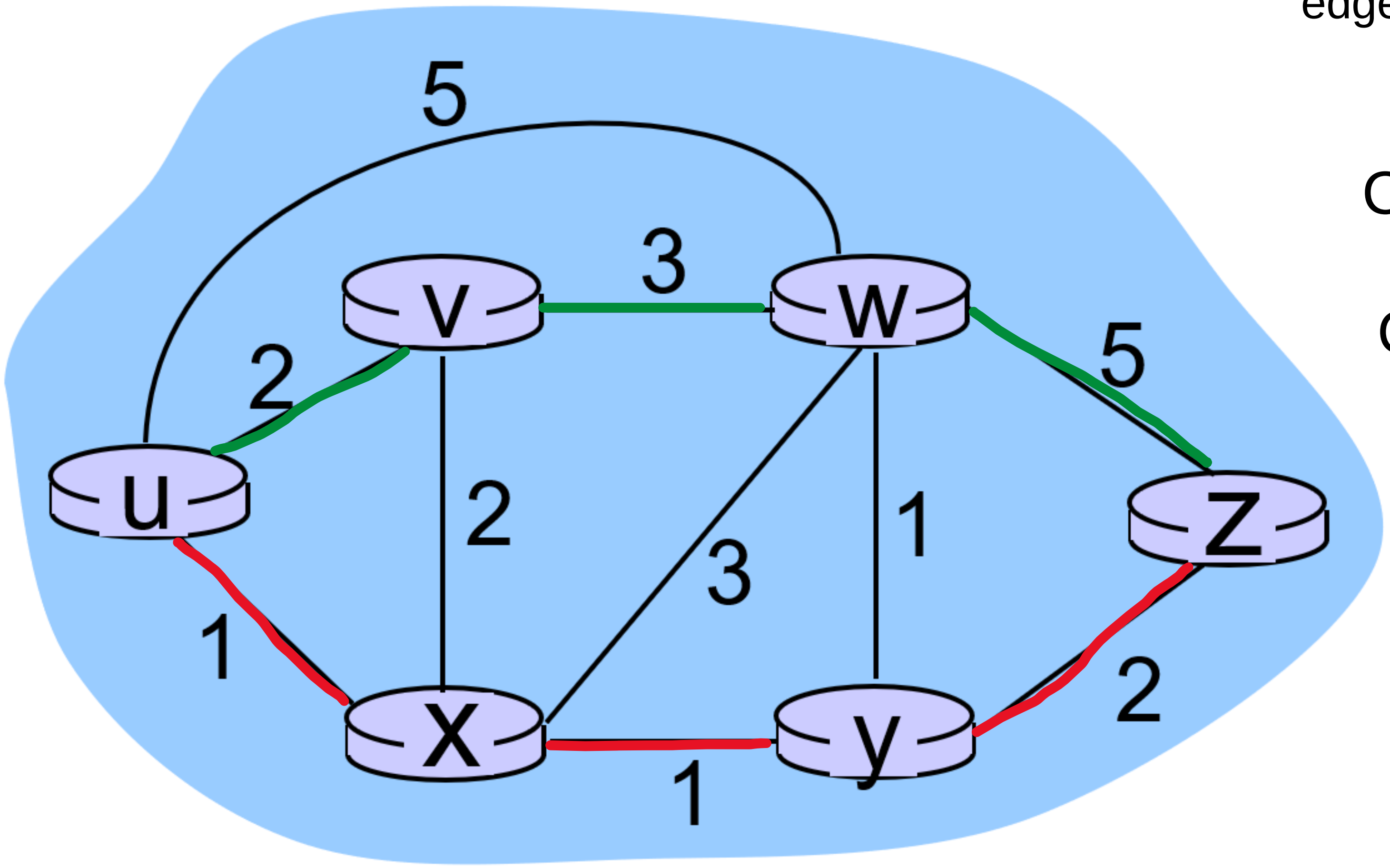
E_2 (u,w) = 5

E_6 (x,y) = 1

In most cases, edges will also have a **cost**

What do the costs represent?? Physical distance, time needed, bandwidth, delay

Network as a Graph



The cost from A to B is the **sum** of the edge weights of the path taken

$C(u,z) = 1 + 1 + 2 = 4$

$C(u,z) = 2 + 3 + 5 = 10$

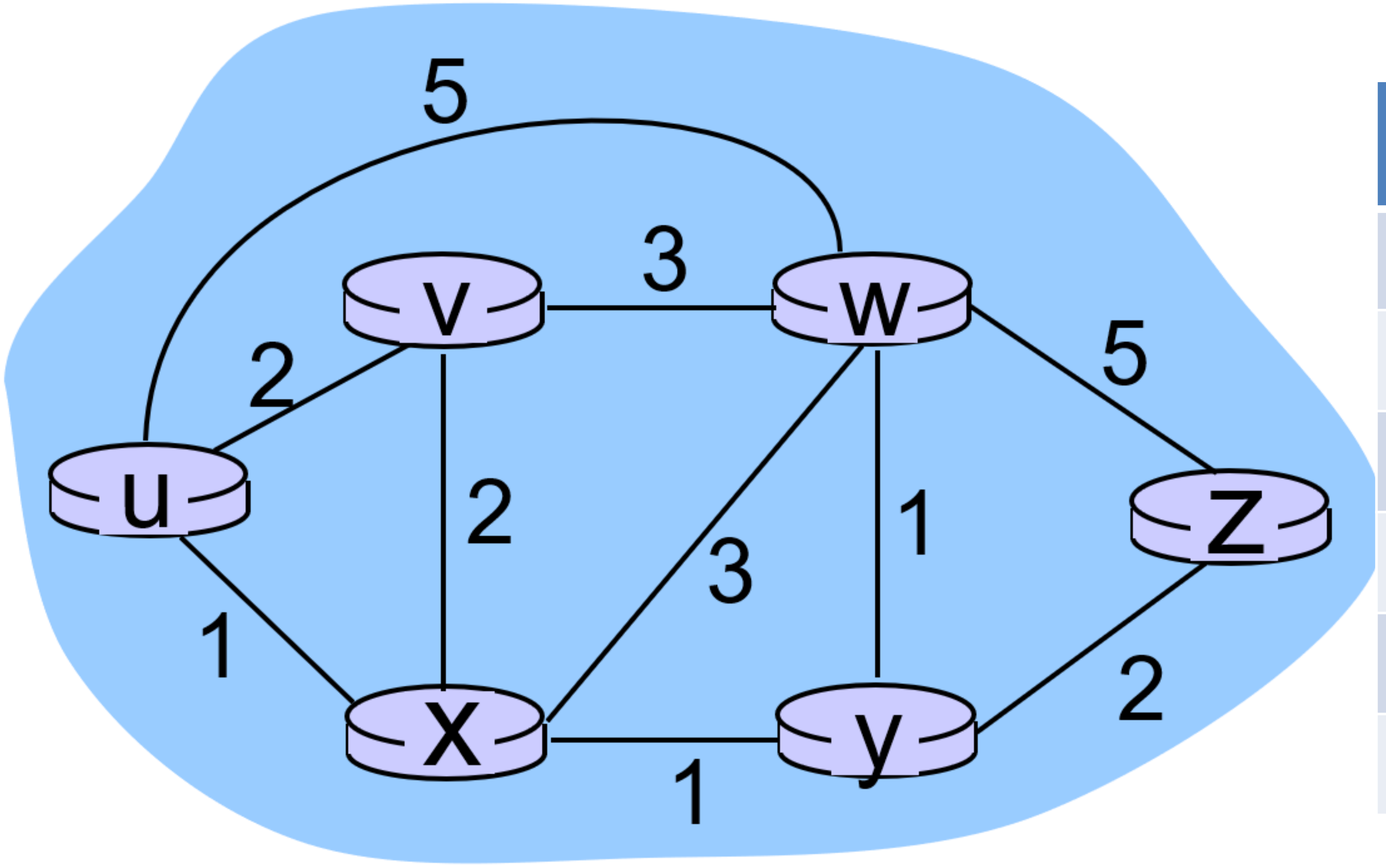
If a packet needs to go from U to Z, we want it to take the most optimal path!

Routing algorithm: algorithm that finds the least cost path

Do you know any? 👀

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution



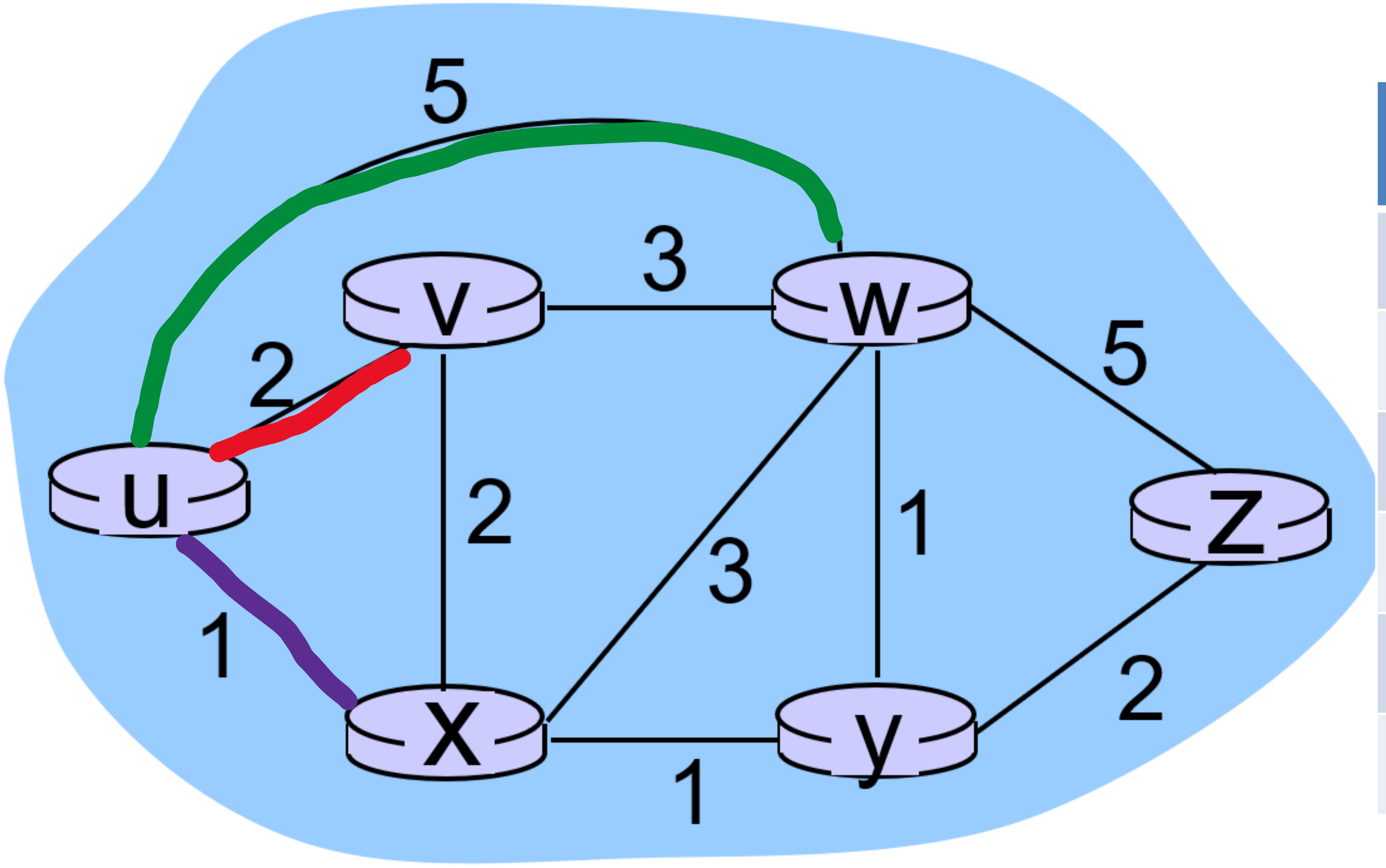
step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1						
2						
3						
4						
5						

Goal: Find shortest path from u to z

N' is our current optimal path. We must start at u obviously

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution

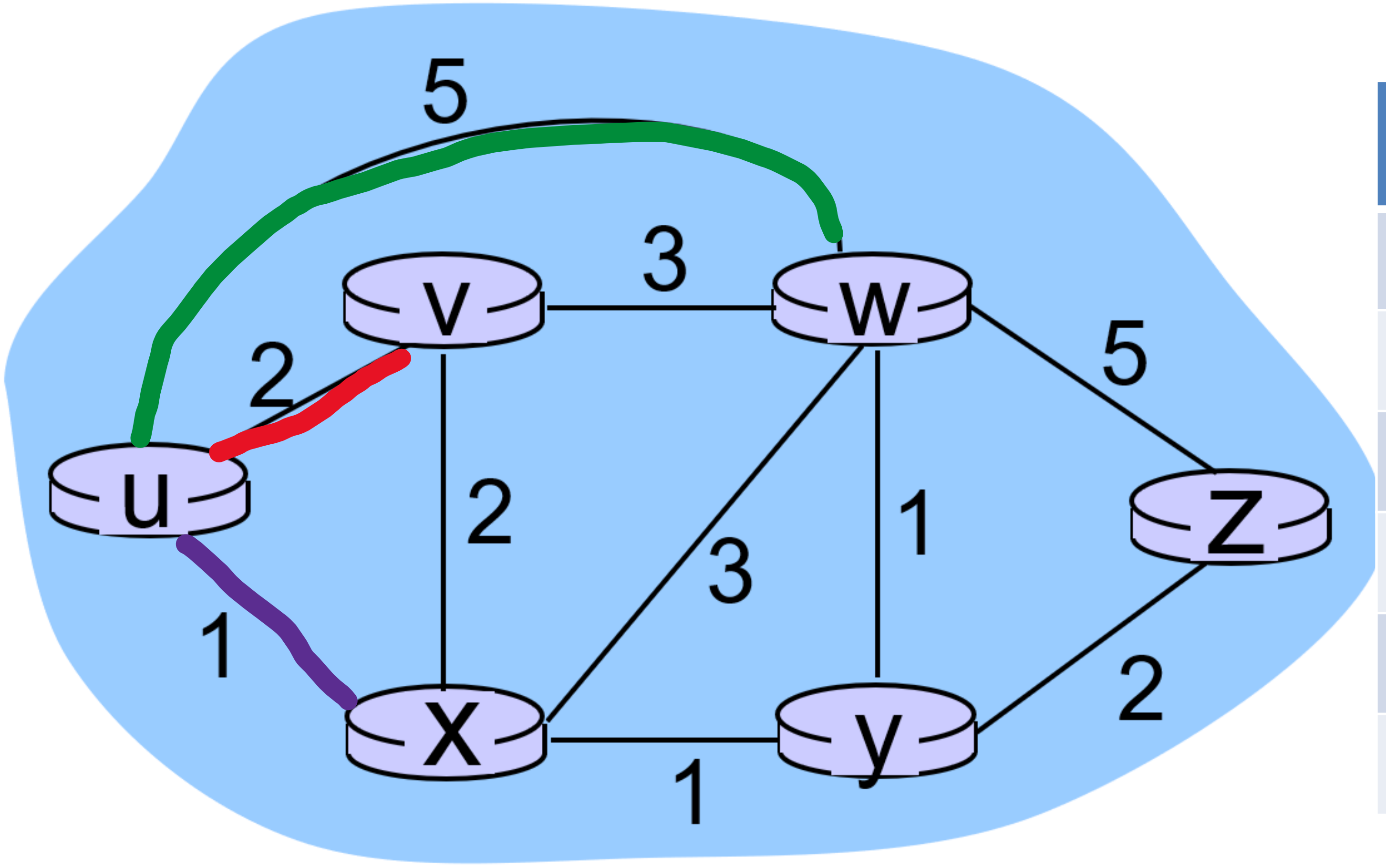


step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1						
2						
3						
4						
5						

Goal: Find shortest path from u to z

N' is our current optimal path. We must start at u obviously

Link-state Algorithm (Dijkstra's algorithm)



Goal: Find shortest path from u to z

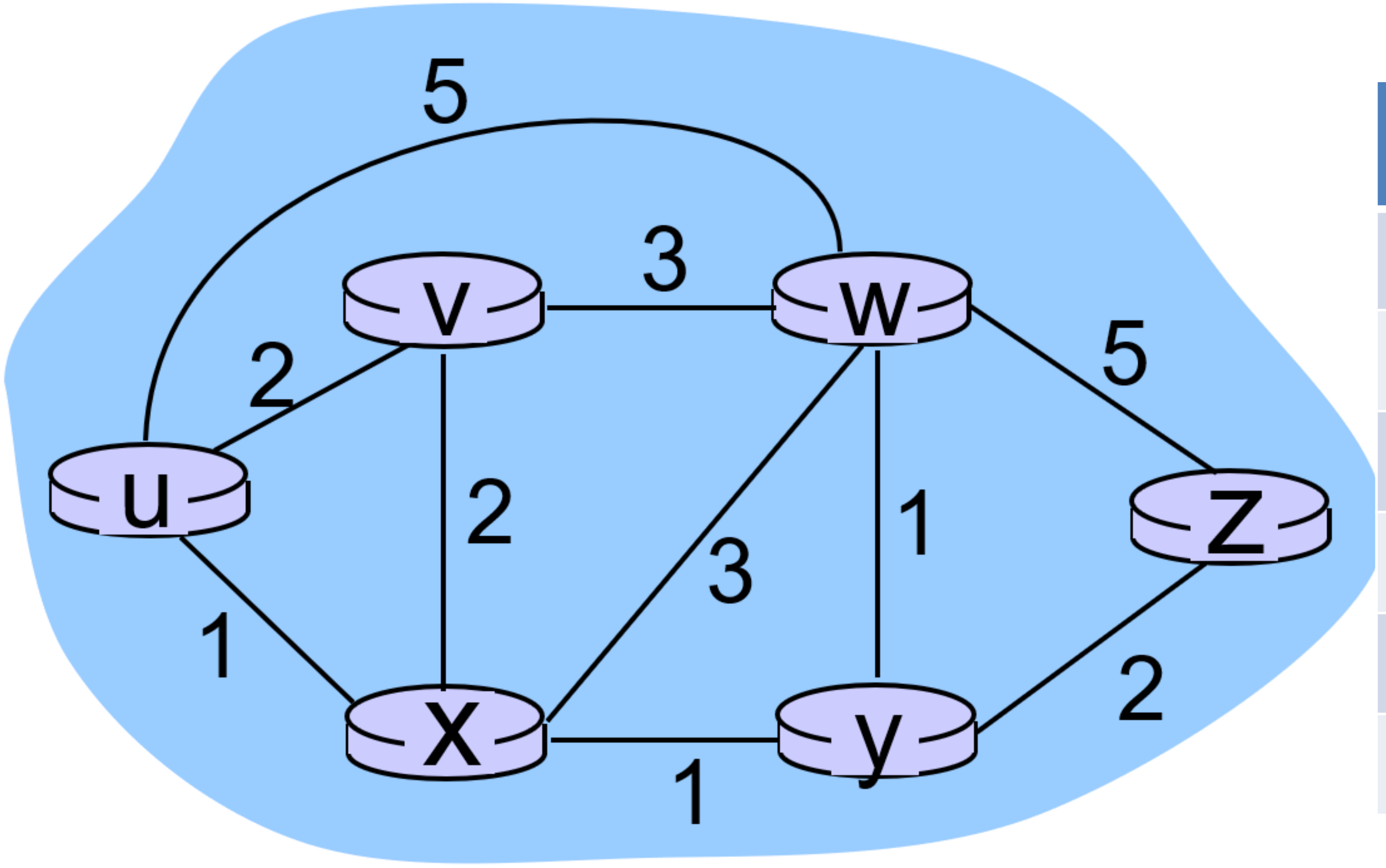
Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution

step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1						
2						
3						
4						
5						

The path with the least cost is to X, so x will get added onto our path!

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution

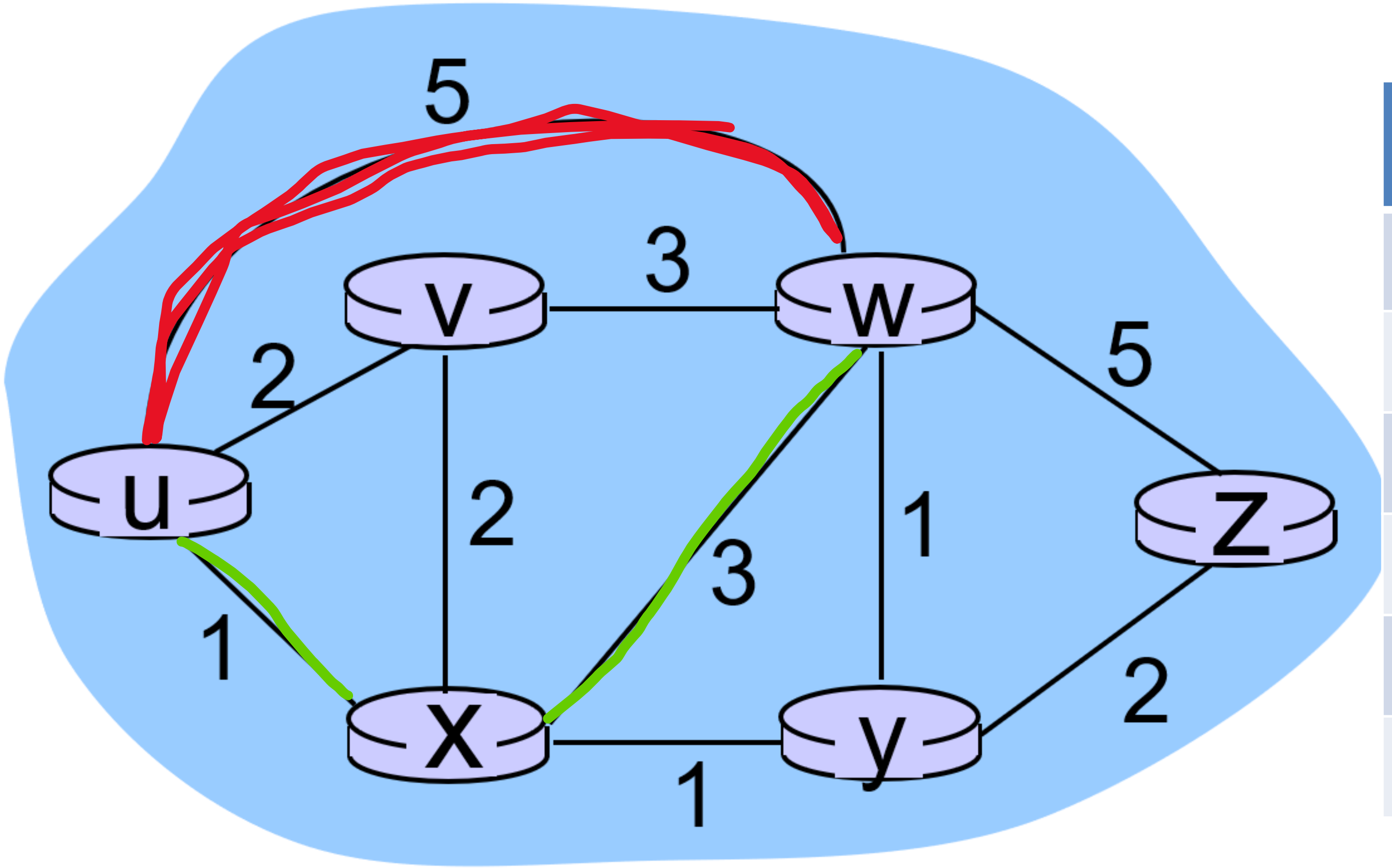


step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1	ux					
2						
3						
4						
5						

Goal: Find shortest path from u to z

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution



step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1	ux	2, u	4, x		2, x	N/A
2						
3						
4						
5						

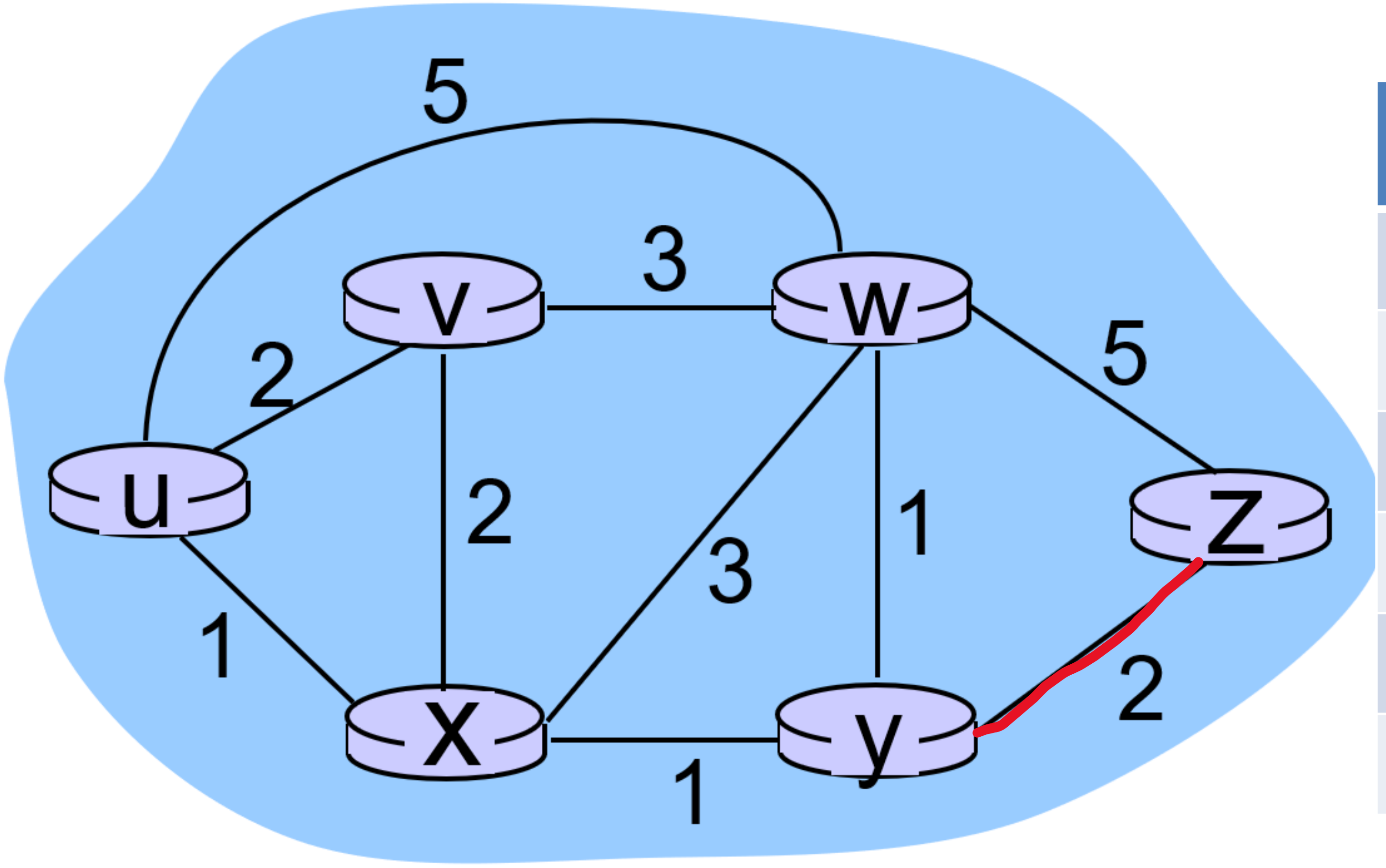
The shortest path from u to v is still the same

The shortest path from u to w is now 4 (travel through x) instead of taking the 4 cost path

Goal: Find shortest path from u to z

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution



step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1	ux	2, u	4, x		2, x	N/A
2	uxy	2, u	3, y			4,y
3						
4						
5						

We also have a more optimal path to W now

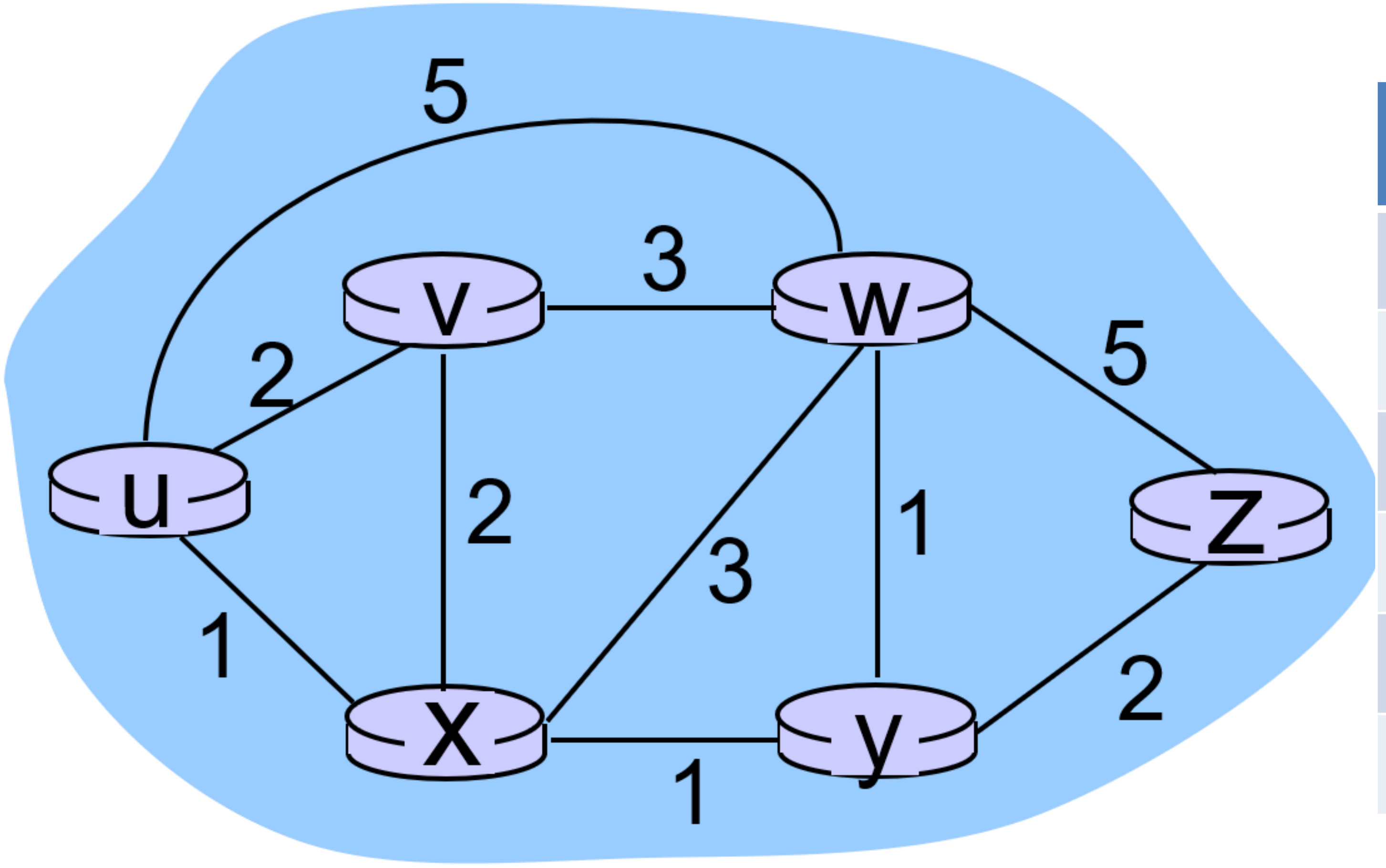
Now that y is on our path, we can now reach z!

Goal: Find shortest path from u to z

Path(u,z) = uxyz
Cost = 1 +1 + 2 = **4**

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution

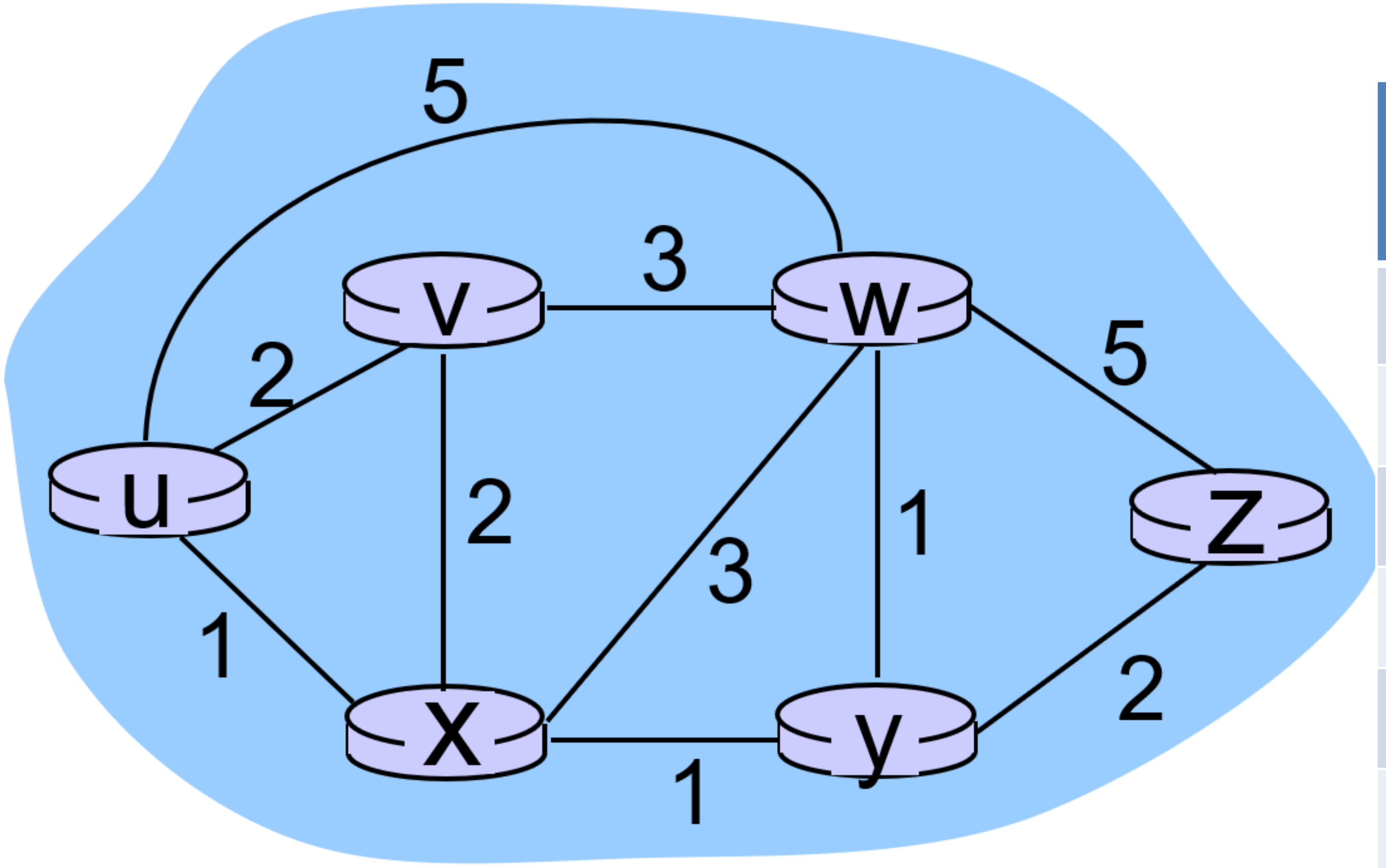


step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1	ux	2, u	4, x		2, x	N/A
2	uxy	2, u	3, y			4,y
3	uxyv		3,y			4,y
4						
5						

NEW Goal: Find shortest path from u to any node

Link-state Algorithm (Dijkstra's algorithm)

Dijkstra's algorithm is a shortest path algorithm that is guaranteed to find a solution



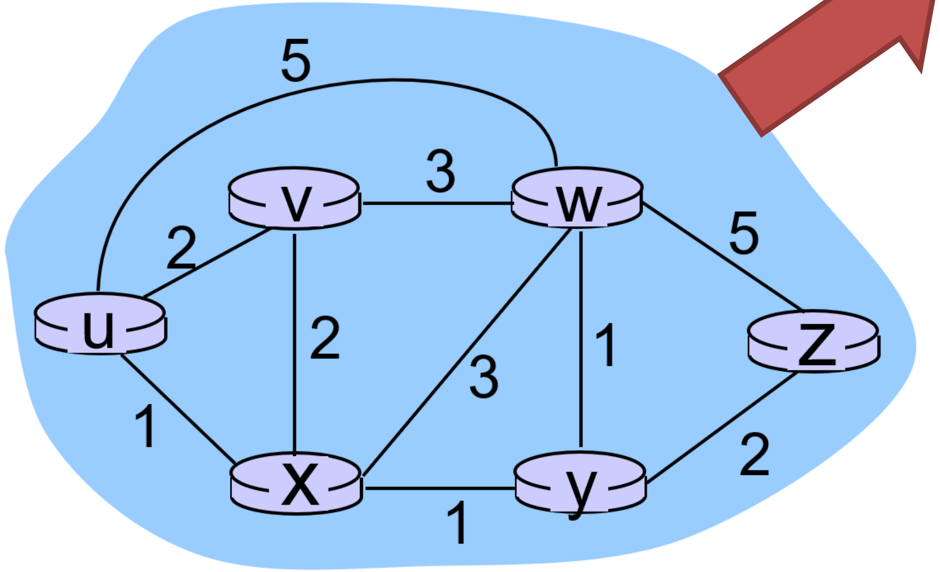
step	N'	D(v), p(v)	D(w), p(w)	D(x), p(x)	D(y), p(y)	D(z), p(z)
0	u	2, u	5, u	1, u	N/A	N/A
1	ux	2, u	4, x		2, x	N/A
2	uxy	2, u	3, y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

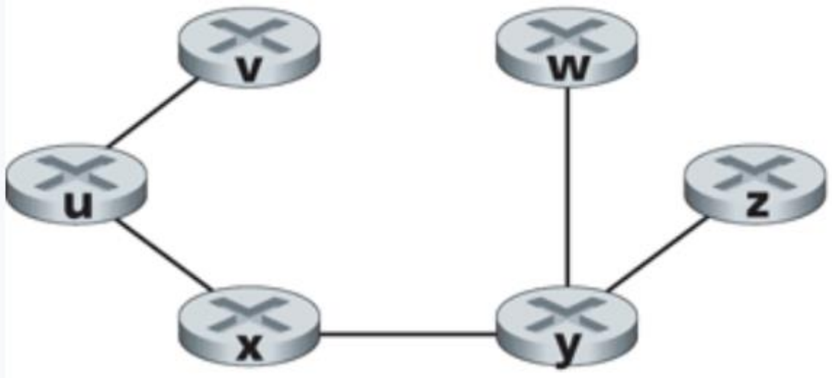
NEW Goal: Find shortest path from u to any node

Link-state Algorithm (Dijkstra's algorithm)

LS complexity = **O(N^2)**

Issues? Concerns?





Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Routing table for node u

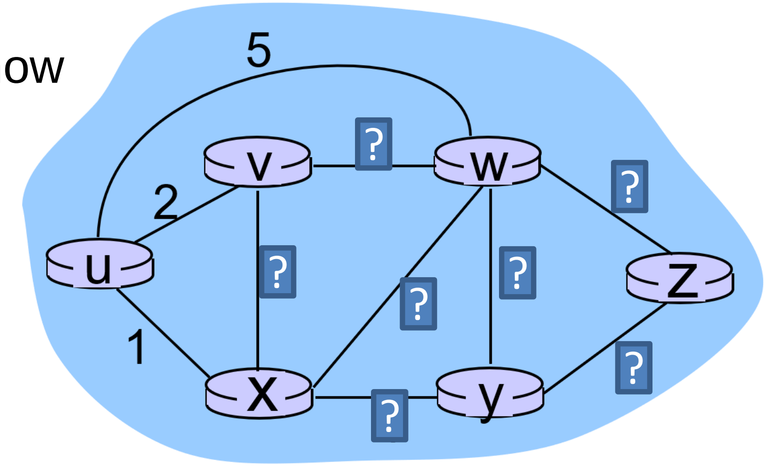
We will then run the same algorithm on each node to create its forwarding table!

Link-state Algorithm (Dijkstra's algorithm)

LS is a **centralized** routing algorithm, which means it has global knowledge about all the edge of the network!

However, it is very likely we will not know information about the network beforehand!

Additionally, path costs can frequently change based on network congestion



Distance Vector Algorithm

- $D_x(y)$ = estimate of least cost from x to y
  x maintains distance vector $\mathbf{D}_x = [D_x(y): y \in N]$
- node x:
  knows cost to each neighbor v: $c(x,v)$
  maintains its neighbors' distance vectors. For each neighbor v, x maintains
  $\mathbf{D}_v = [D_v(y): y \in N]$

DV is **not** a centralized algorithm, which means that it does not require global knowledge about network edge costs

key idea:

from time-to-time, each node sends its own distance vector estimate to neighbors

when x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

Distance vector algorithm

iterative, asynchronous:

each local iteration caused by:

local link cost change

DV update message from neighbor

distributed:

each node notifies neighbors *only* when its DV changes

neighbors then notify their neighbors if necessary

each node:

wait for (change in local link cost or msg from neighbor)

recompute estimates

if DV to any dest has changed, *notify* neighbors

Comparison of LS and DV algorithms

message complexity

LS: with n nodes, E links, O(nE) msgs sent

DV: exchange between neighbors only

- convergence time varies

speed of convergence

- ***LS:*** $O(n^2)$ algorithm requires O(nE) msgs
  - may have oscillations
- ***DV:*** convergence time varies
  - may be routing loops
  - count-to-infinity problem

robustness: what happens if router malfunctions?

LS:

node can advertise incorrect *link* cost

each node computes only its *own* table

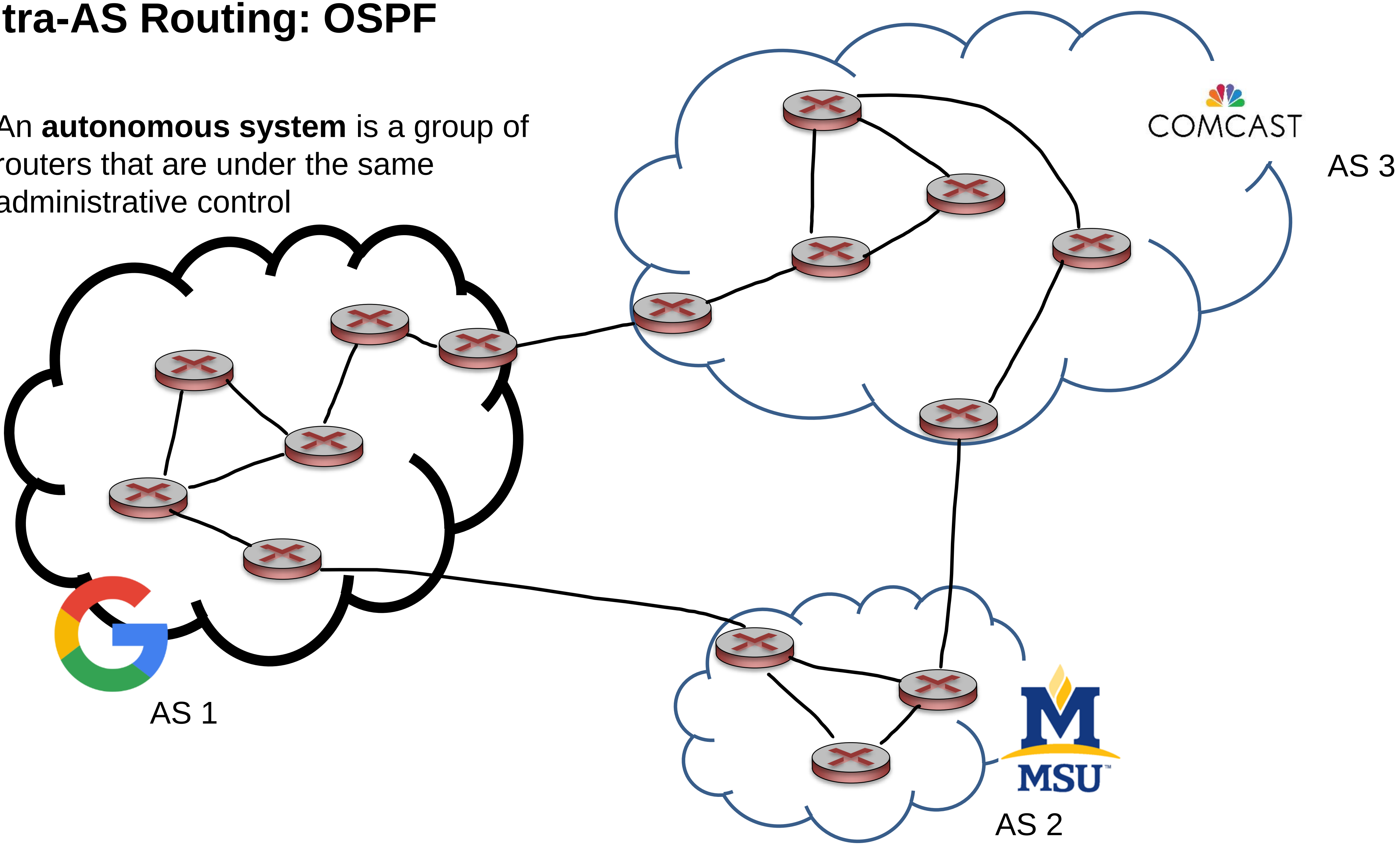
DV:

- DV node can advertise incorrect *path* cost
- each node's table used by others

error propagate thru network

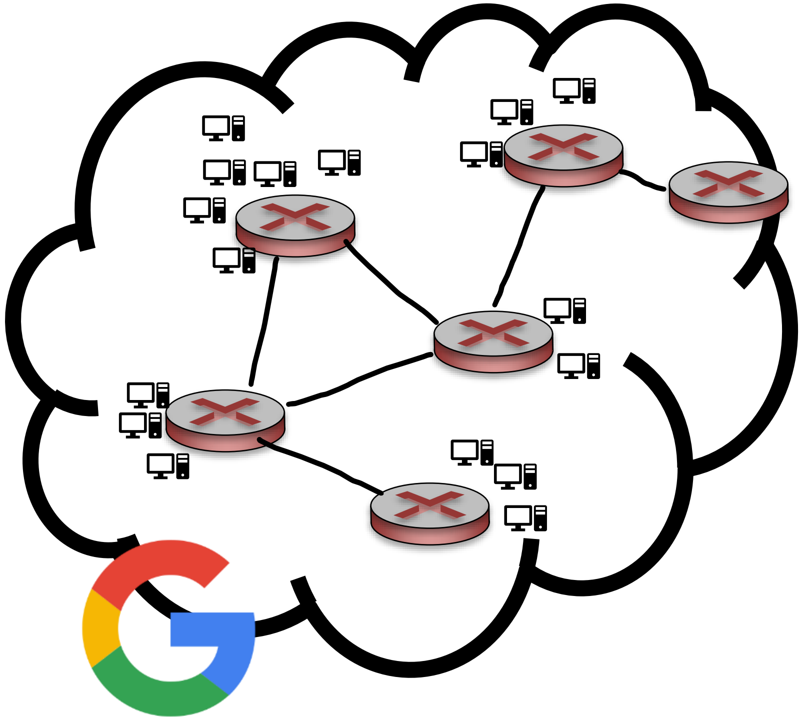
Intra-AS Routing: OSPF

An **autonomous system** is a group of routers that are under the same administrative control



Intra-AS Routing: OSPF

An **autonomous system** is a group of routers that are under the same administrative control



AS 1

Open Shortest Path First

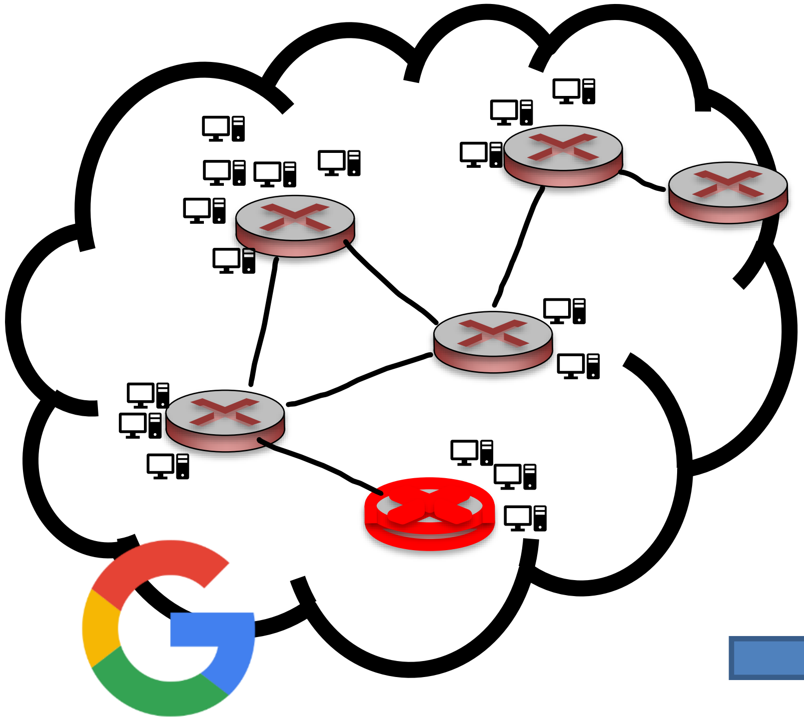
OSPF is a link-state protocol that uses flooding of link-state information and Dijkstra's least-cost algorithm

Intra-AS Routing: OSPF

Open Shortest Path First

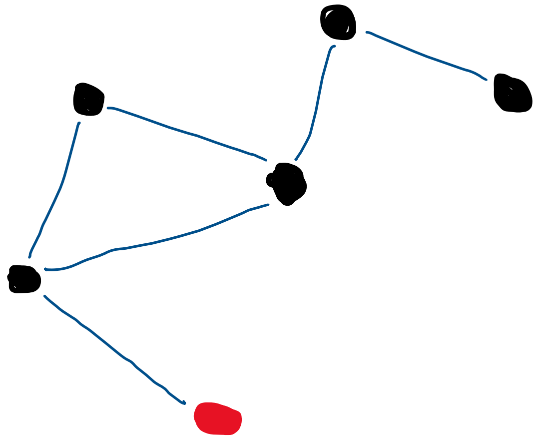
An **autonomous system** is a group of routers that are under the same administrative control

OSPF is a link-state protocol that uses flooding of link-state information and Dijkstra’s least-cost algorithm



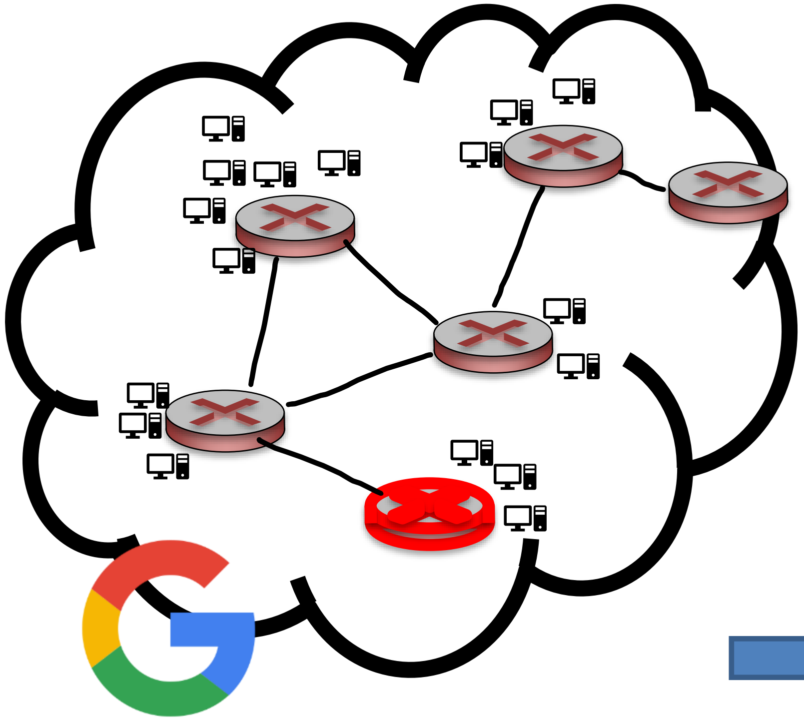
AS 1

1. Each router constructors a topological map of the AS



Intra-AS Routing: OSPF

An **autonomous system** is a group of routers that are under the same administrative control

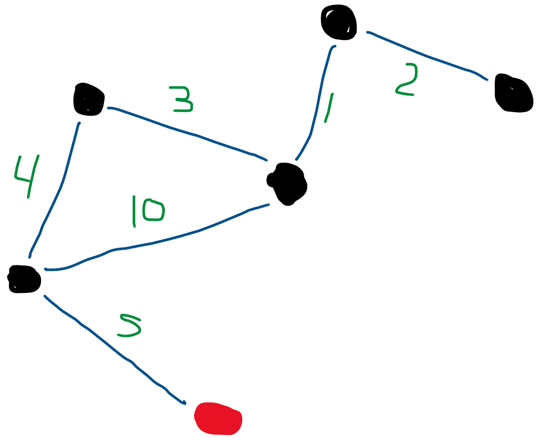


AS 1

Open Shortest Path First

OSPF is a link-state protocol that uses flooding of link-state information and Dijkstra's least-cost algorithm

1. Each router constructors a topological map of the AS
2. Run Dijikstra's to determine shortest path to each subnet



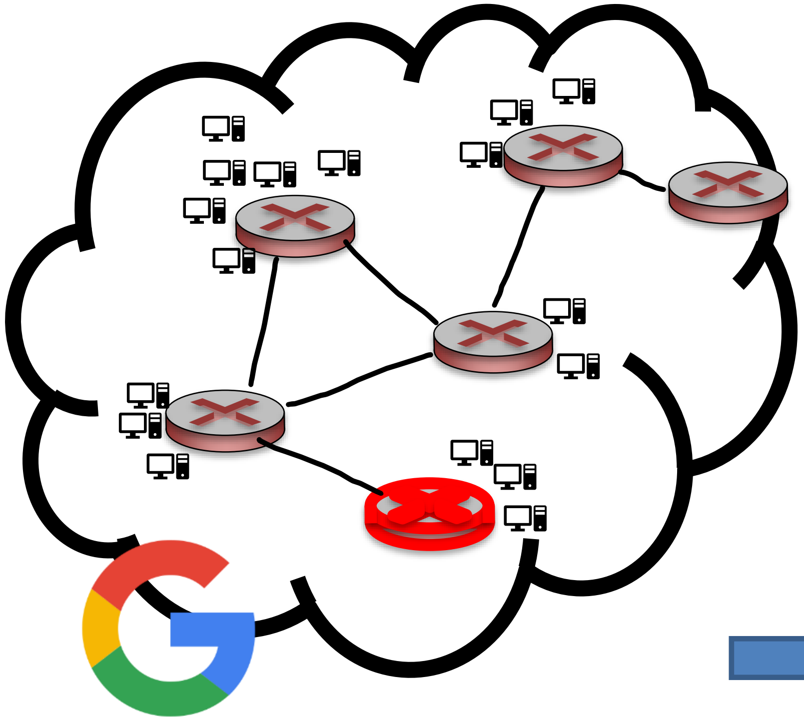
(Edge costs will be set by a network administrator)

(could set all edges to be a cost of 1)

Intra-AS Routing: OSPF

Open Shortest Path First

An **autonomous system** is a group of routers that are under the same administrative control

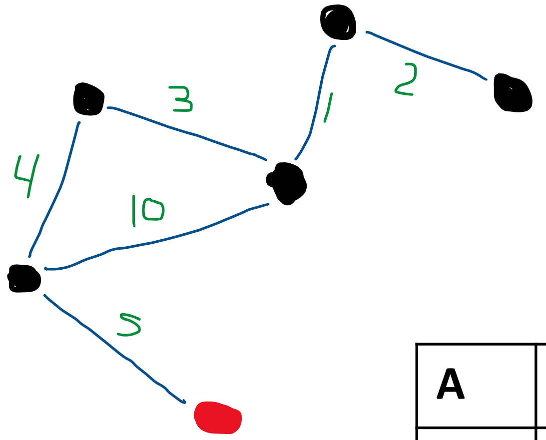


AS 1

OSPF is a link-state protocol that uses flooding of link-state information and Dijkstra’s least-cost algorithm

1. Each router constructors a topological map of the AS

2. Run Dijikstra’s to determine shortest path to each subnet



(Edge costs will be set by a network administrator)

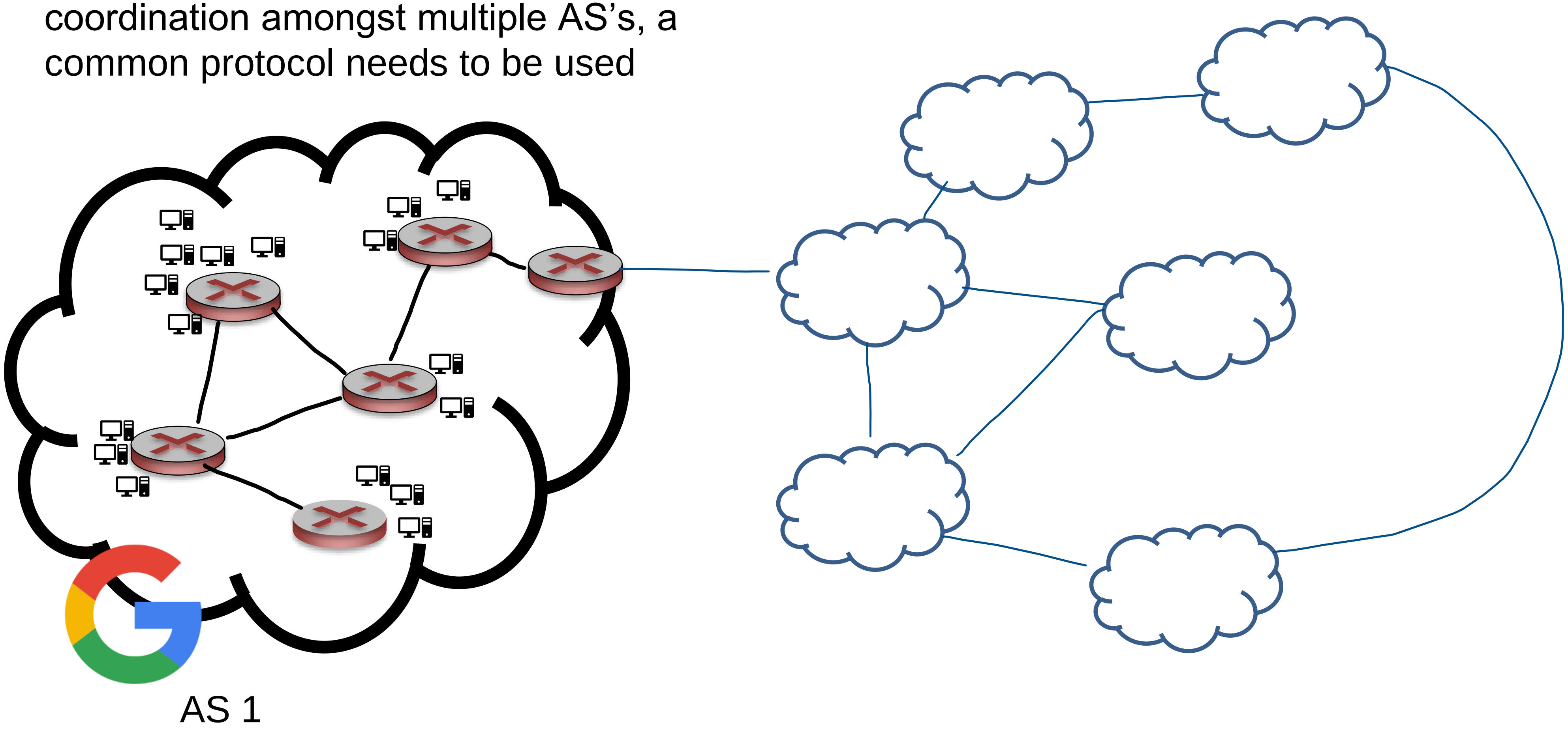
(could set all edges to be a cost of 1)

3. Fill in routing table

A	**1**
B	**2**
C	**3**
…	**…**

Routing Among the ISPs: BGP

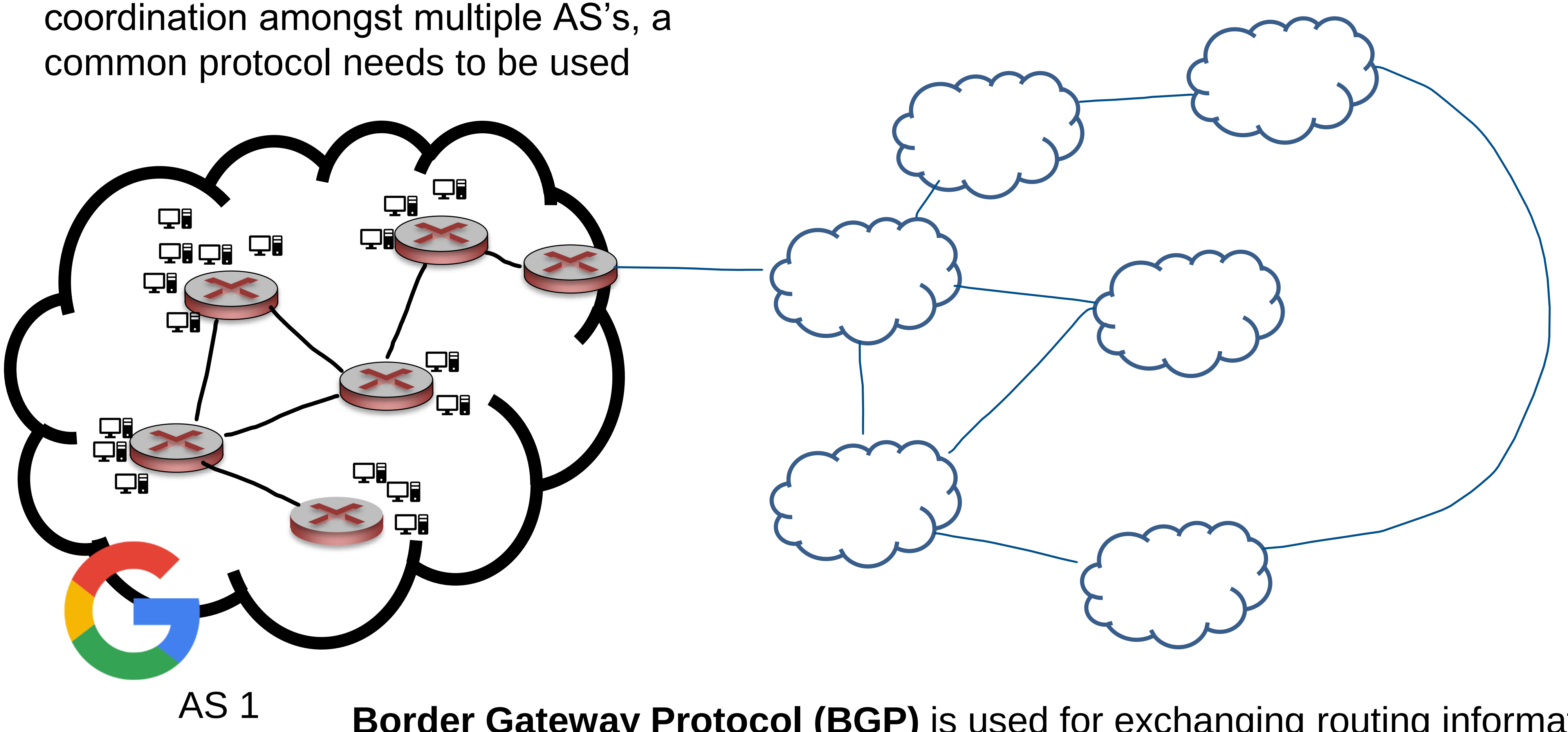
Inter-AS routing protocol involves coordination amongst multiple AS's, a common protocol needs to be used



Border Gateway Protocol (BGP) is used for exchanging routing information between AS

Routing Among the ISPs: BGP

Inter-AS routing protocol involves coordination amongst multiple AS's, a common protocol needs to be used



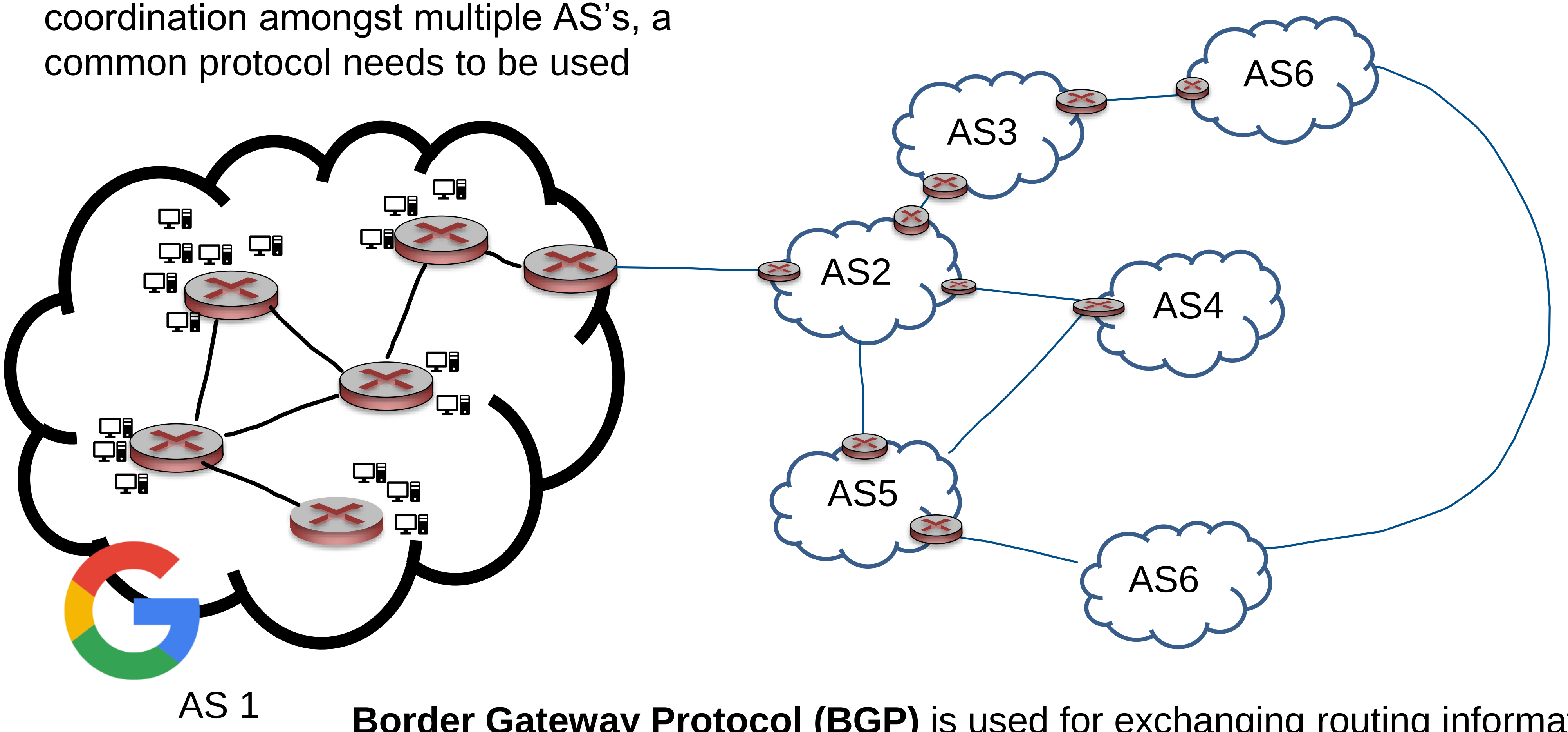
Border Gateway Protocol (BGP) is used for exchanging routing information between AS

BGP allows a router to tell other AS's that it exists and needs to be connected

Routing Among the ISPs: BGP

AS consists of **gateway routers** and **internal routers**

Inter-AS routing protocol involves coordination amongst multiple AS’s, a common protocol needs to be used



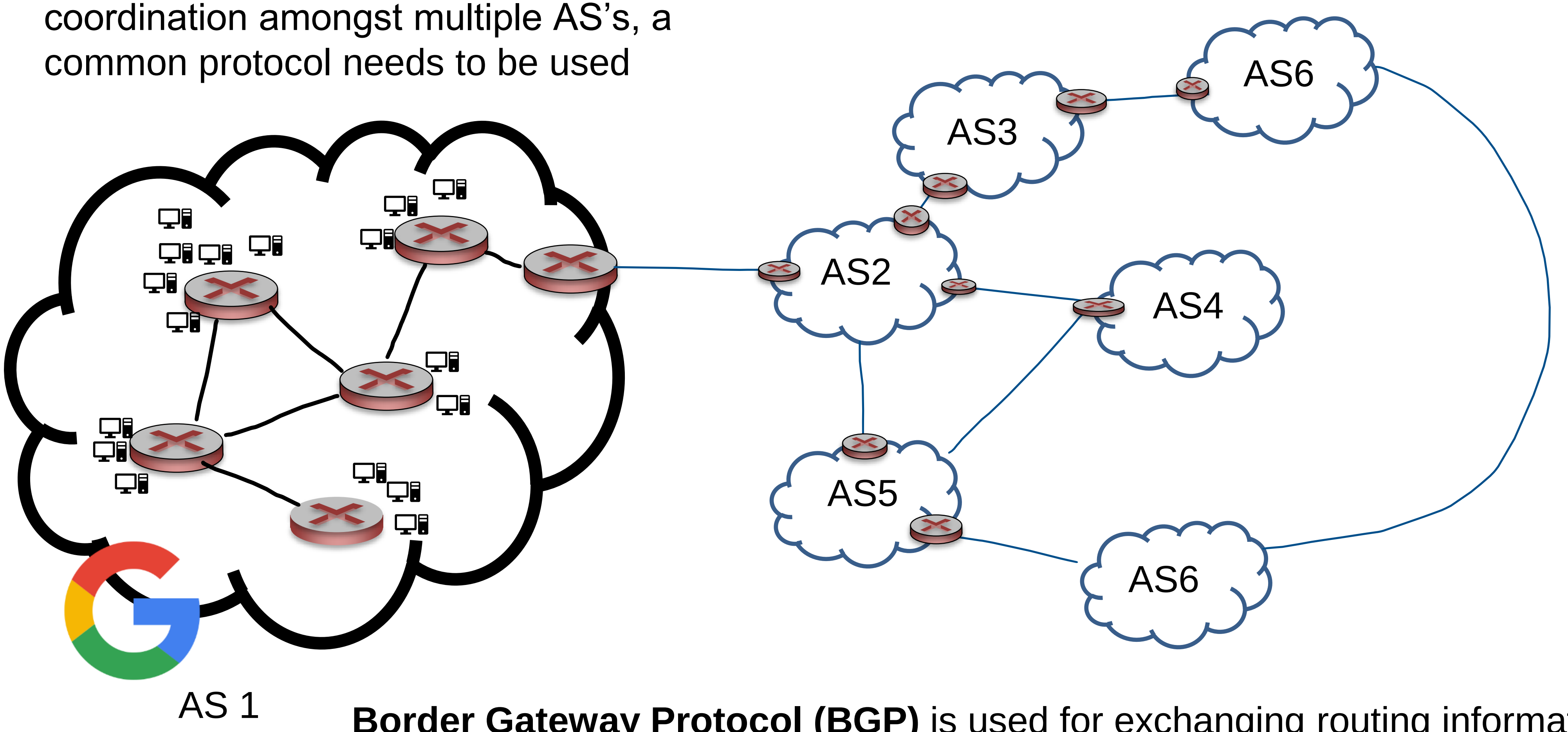
Border Gateway Protocol (BGP) is used for exchanging routing information between AS

BGP allows a router to tell other AS’s that it exists and needs to be connected

Routing Among the ISPs: BGP

AS consists of **gateway routers** and **internal routers**

Inter-AS routing protocol involves coordination amongst multiple AS's, a common protocol needs to be used



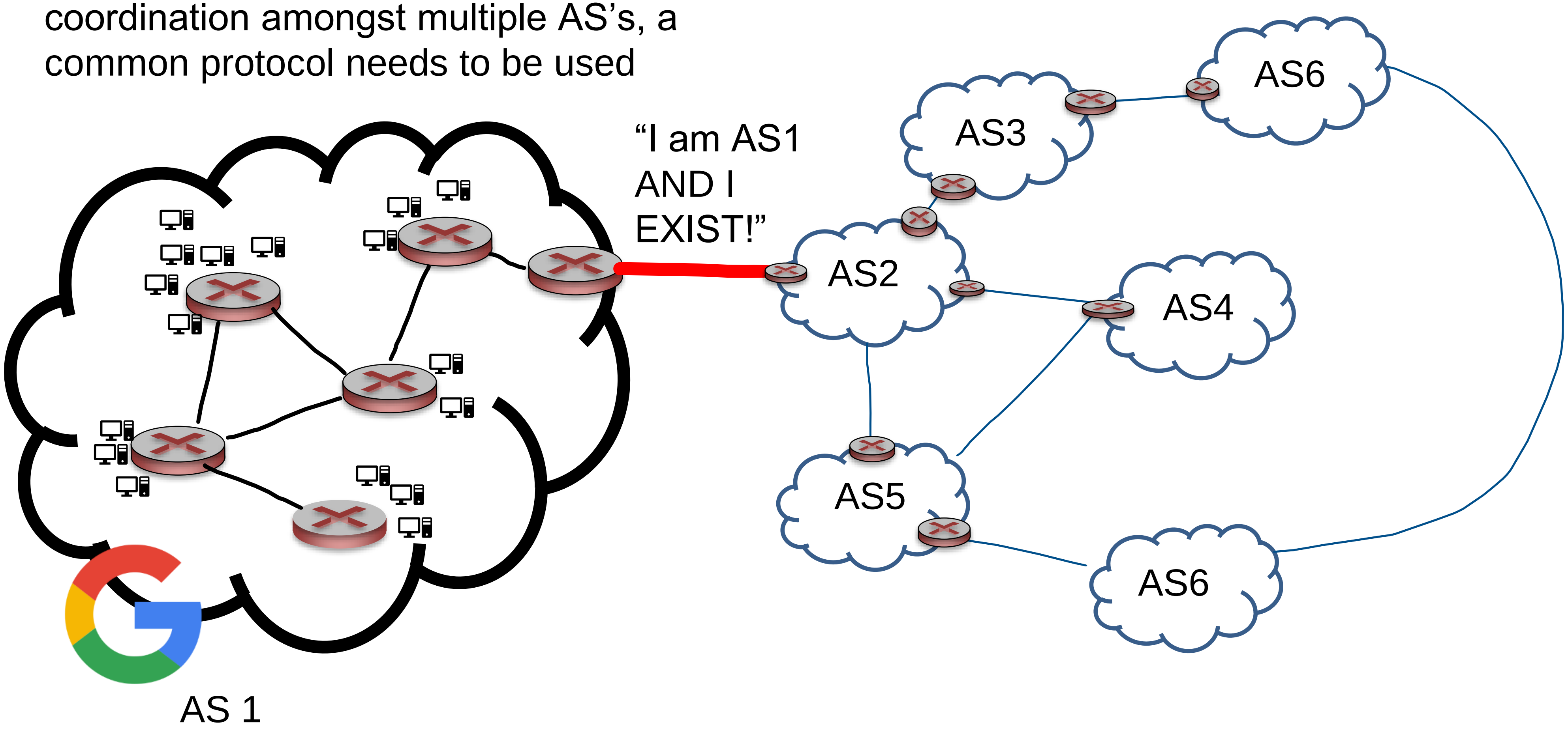
Border Gateway Protocol (BGP) is used for exchanging routing information between AS

BGP allows a router to tell other AS's that it exists and needs to be connected

Routing Among the ISPs: BGP

AS consists of **gateway routers** and **internal routers**

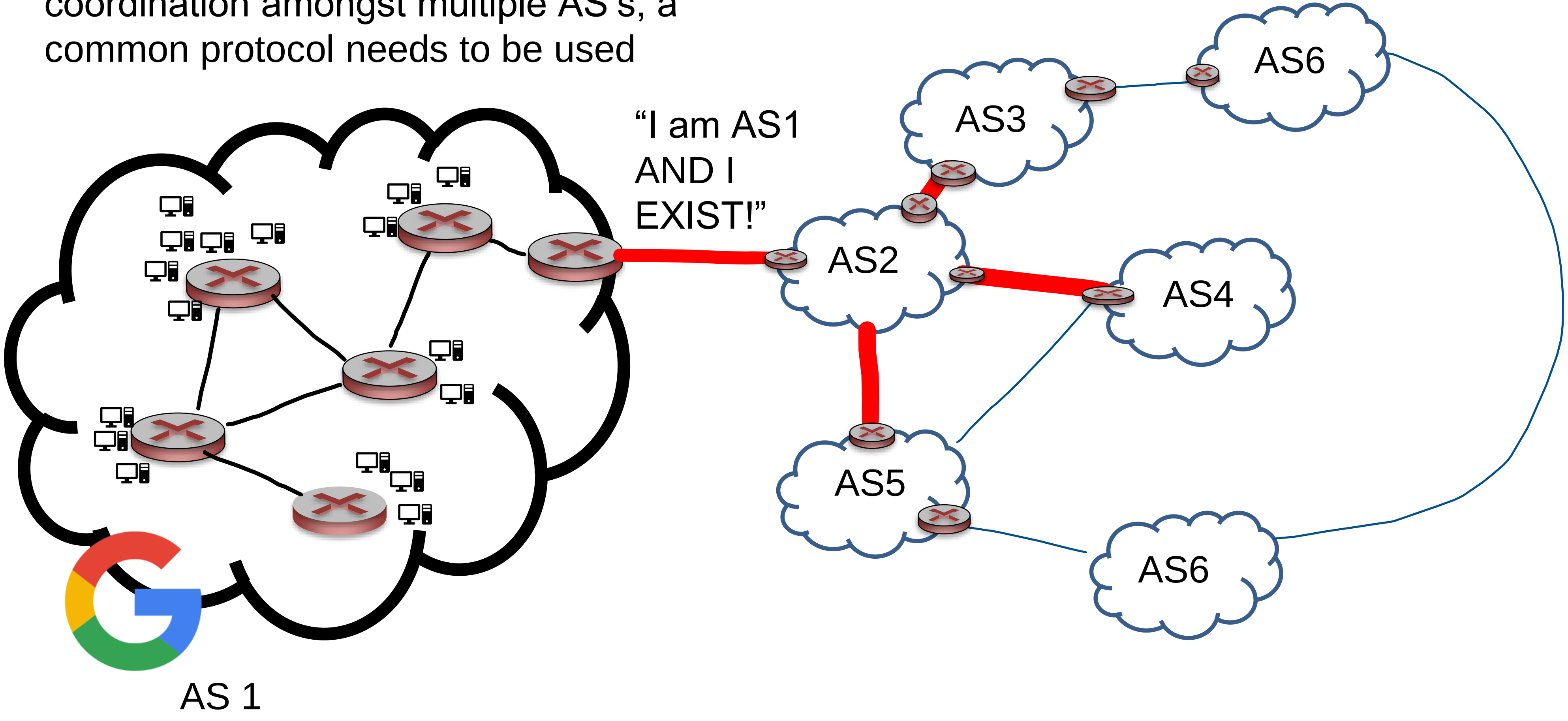
Inter-AS routing protocol involves coordination amongst multiple AS's, a common protocol needs to be used



Routing Among the ISPs: BGP

AS consists of **gateway routers** and **internal routers**

Inter-AS routing protocol involves coordination amongst multiple AS's, a common protocol needs to be used



"A1 EXISTS AND FOUND THROUGH AS2"

Announcements

HW2 due tonight

PA3 due Monday November 14th

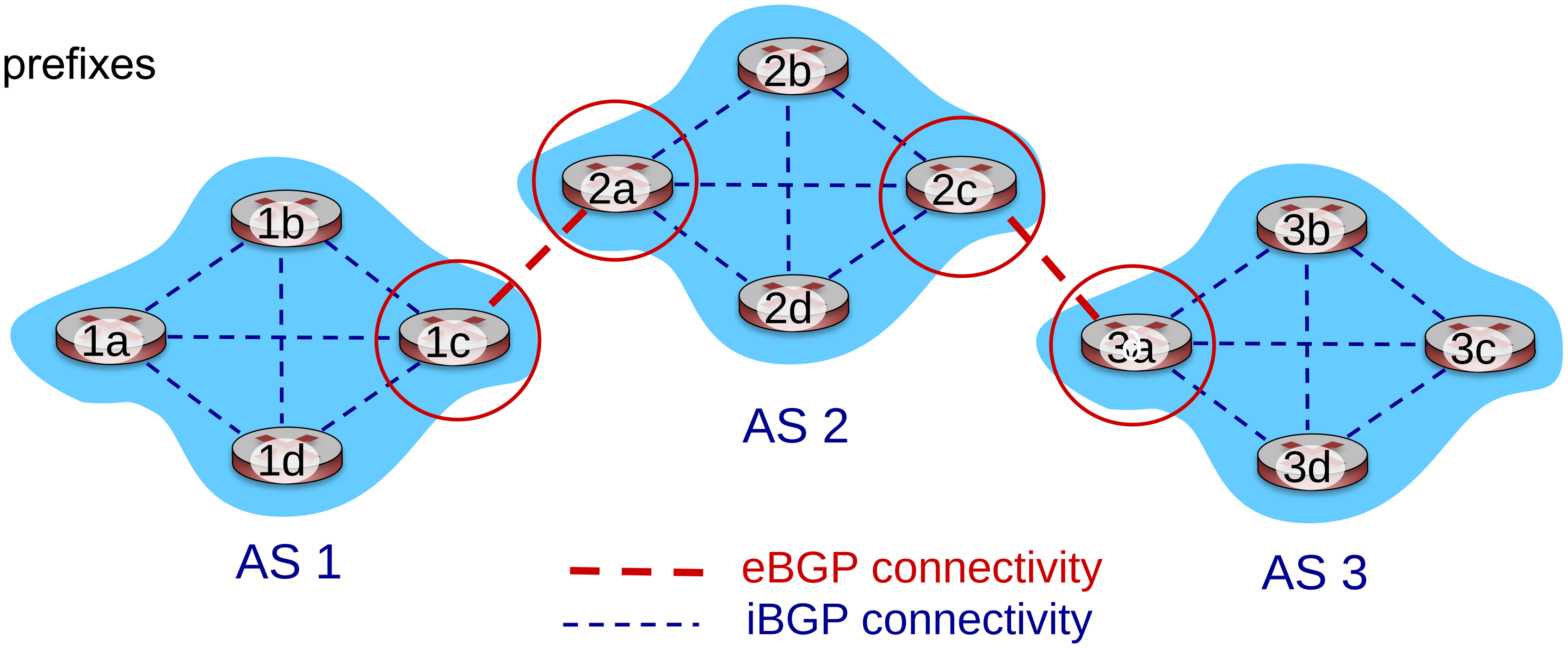
Routing Among the ISPs: BGP

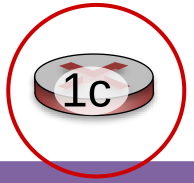
BGP is the routing protocol used for routing amongst different ISPs + AS

Two important functions

→ Obtain prefix reachability information from neighboring ASs

→ Determine the “best” routes to the prefixes





gateway routers run both eBGP and iBGP protools

Routing Among the ISPs: BGP

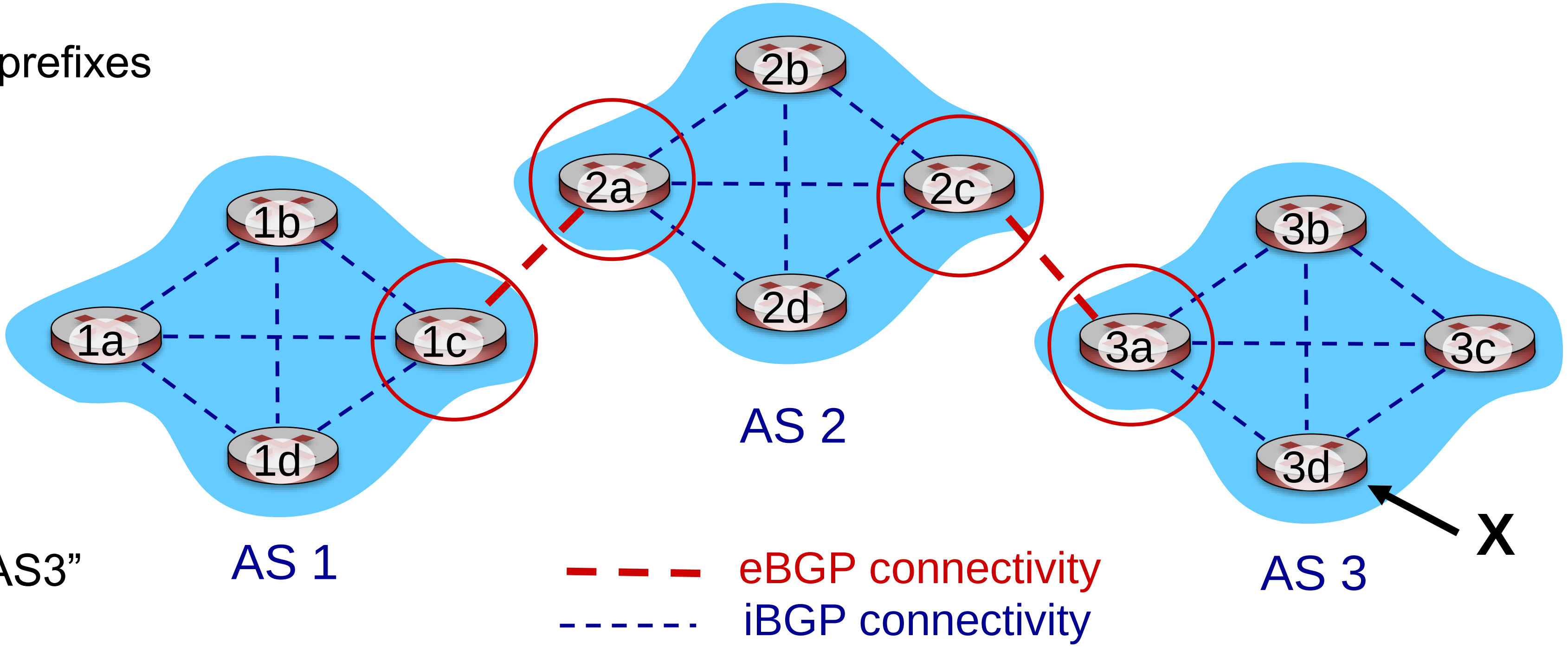
BGP is the routing protocol used for routing amongst different ISPs + AS

Two important functions

→ Obtain prefix reachability information from neighboring ASs

→ Determine the “best” routes to the prefixes

Prefix **X** connect



External BGP (eBGP)

3a → 2c “Hey I have X”

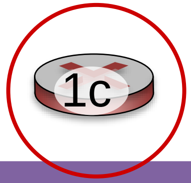
2a → 1c “Hey AS 3 has X and I have AS3”

Internal BGP (iBGP)

2c → 2b

2c → 2d

2c → 2a



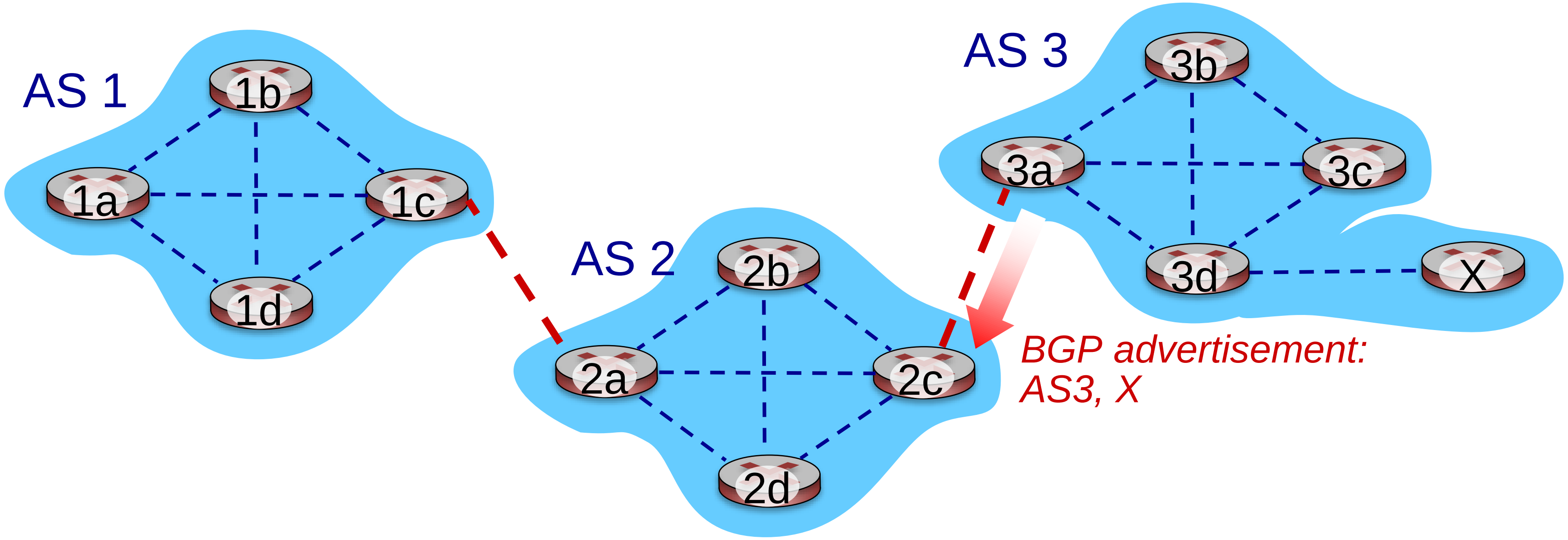
gateway routers run both eBGP and iBGP protools

BGP basics

- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a "path vector" protocol)

when AS3 gateway router 3a advertises path AS3,X to AS2 gateway router 2c:

AS3 *promises* to AS2 it will forward datagrams towards X



BGP basics

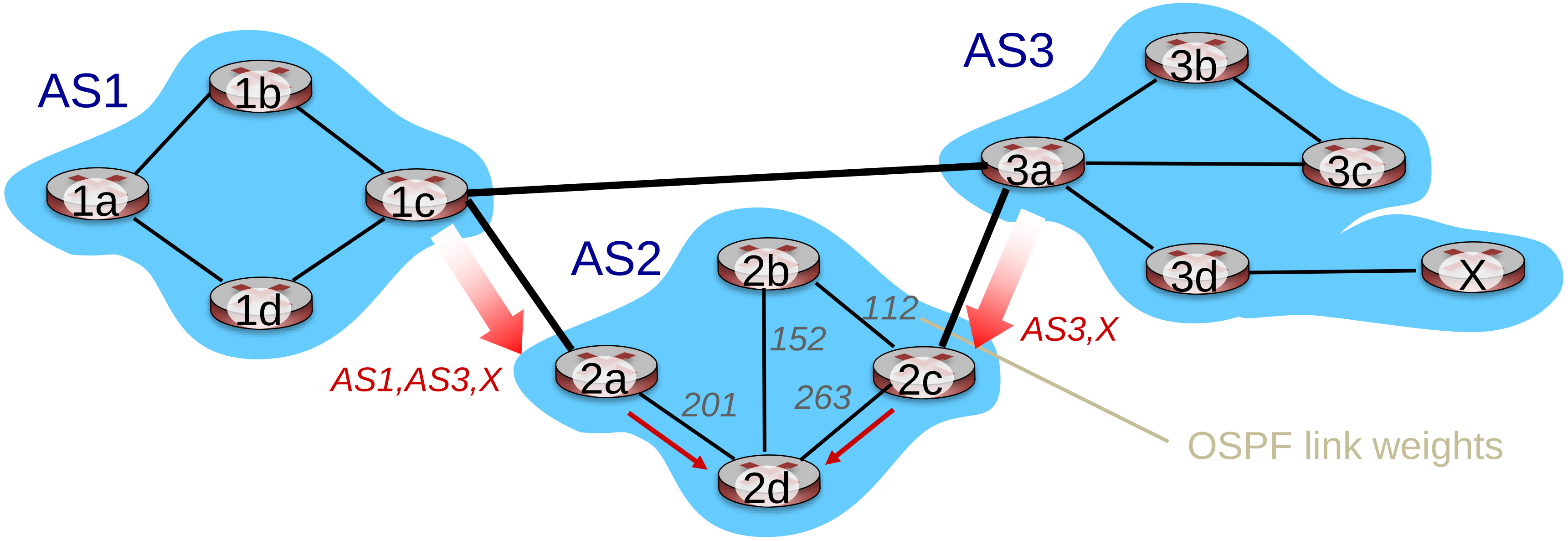
Advertisements with BGP also include “BGP attributes”

Prefix (ip address or subnet) + **attributes** = “**route**”

Two important attributes:

- AS-PATH: list of ASes through which prefix advertisement has passed
- NEXT-HOP: indicates specific internal-AS router to next-hop AS

Hot Potato Routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- *hot potato routing:* choose local gateway that has least intra-domain cost (e.g., 2d chooses 2a, even though more AS hops to *X*): don't worry about inter-domain cost!

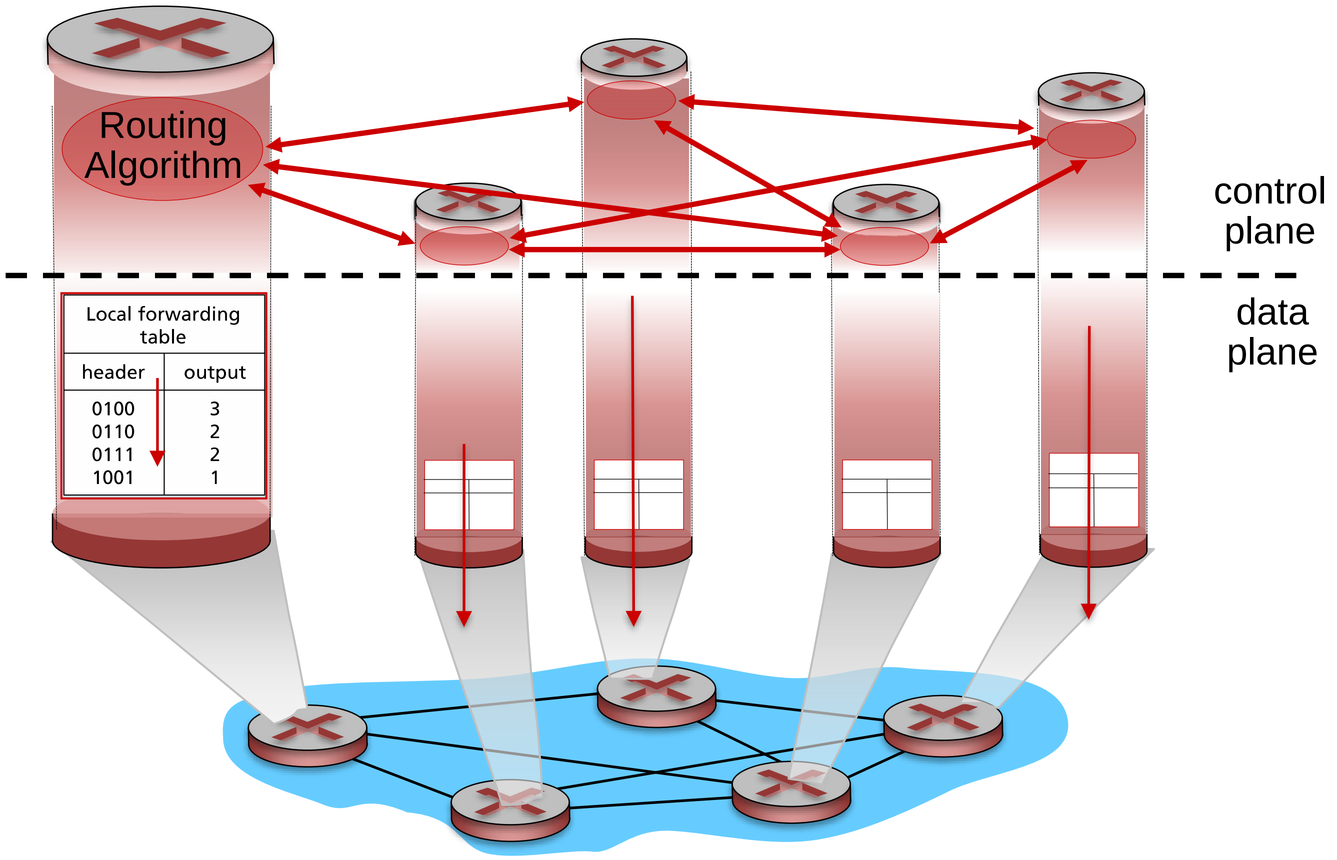
ICMP (Internet Control Message Protocol)

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Control Plane Wrap up

- approaches to network control plane
  - per-router control (traditional)
  - logically centralized control (software defined networking)
- traditional routing algorithms
  - routing: link state, distance vectors
  - implementation in Internet: OSPF, BGP
- Internet Control Message Protocol



Next:
1. Link Layer
2. Security
3. Random Topics in session, presentation, and physical layers