CS2105 Introduction to Computer Networks

Lecture 7

Network Layer: Control Plane

1 October 2018

Learning Outcomes

After this class, you are expected to know:

- the purpose of routing protocols on the Internet
- the differences between inter-domain and intra-domain routing
- the workings of link-state and distance vector routing algorithms
- the principle of Bellman-Ford equation
- how RIP works

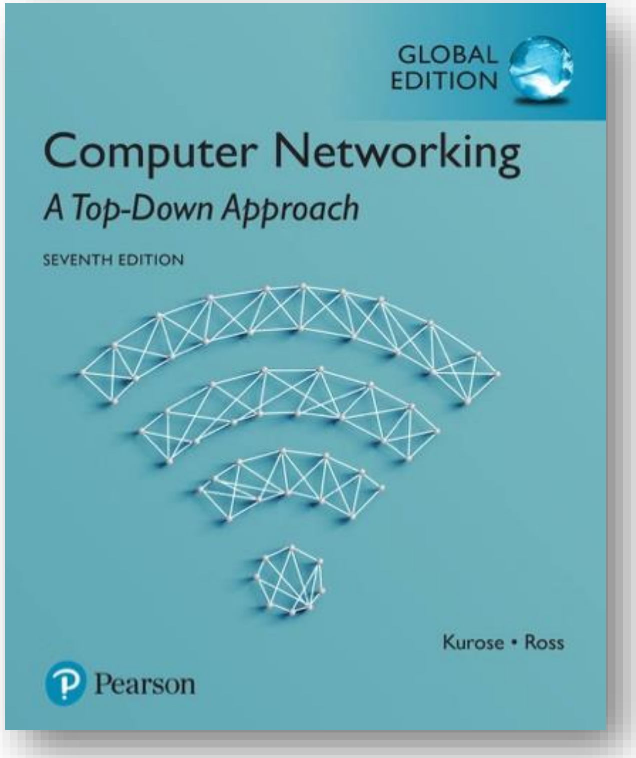
Chapter 5: Roadmap

5.1 Introduction

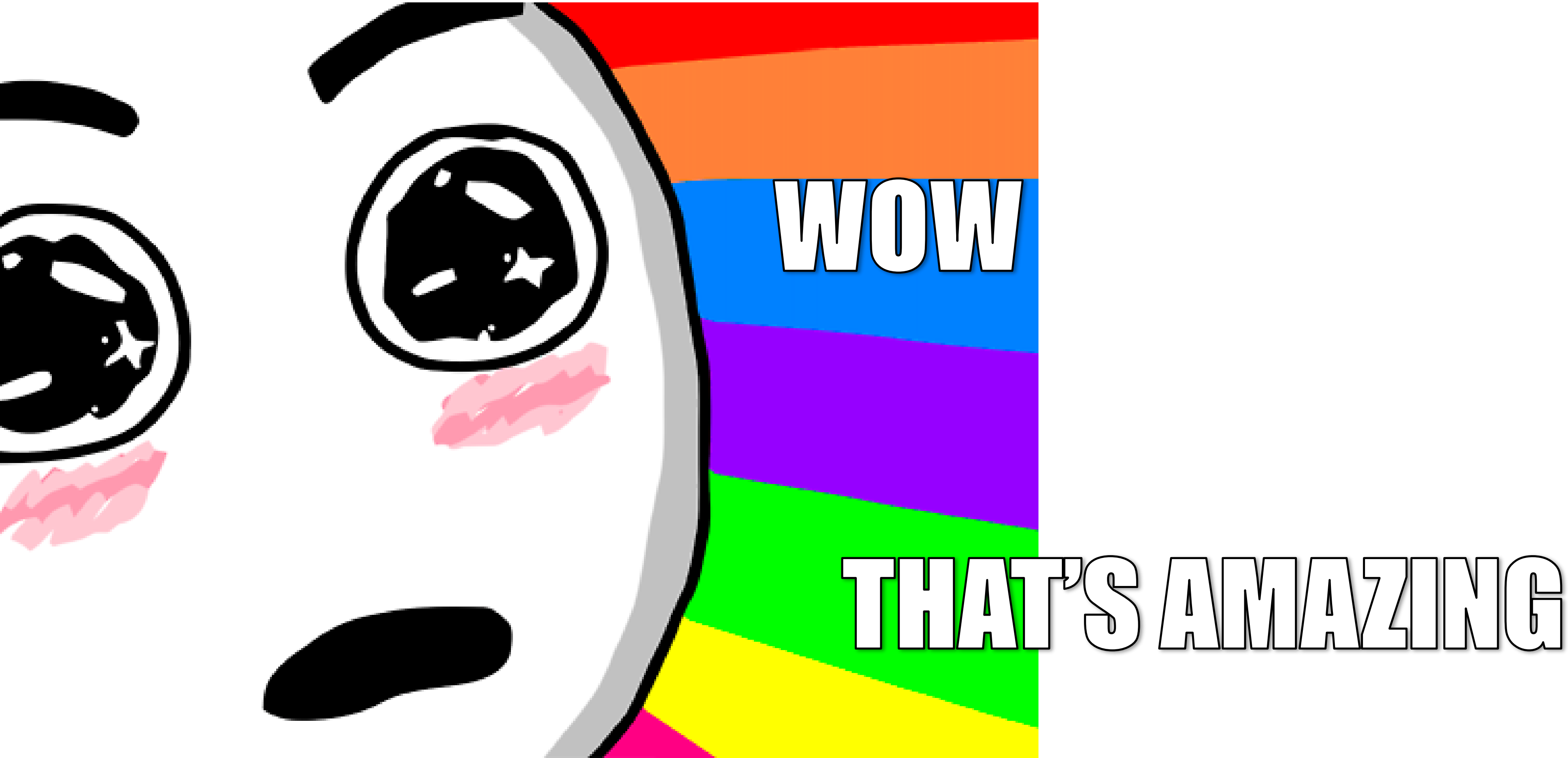
5.2 Routing Algorithms

  5.2.1 Link State (LS) Routing Algorithm

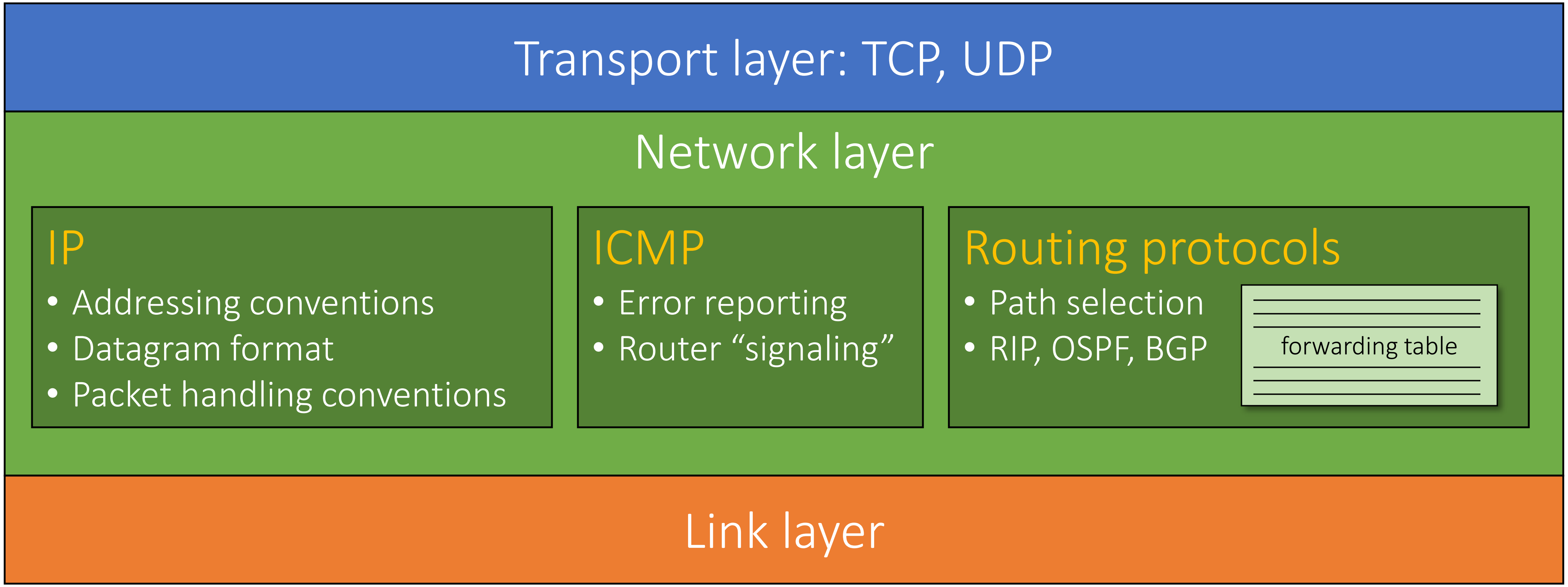
  5.2.2 Distance-Vector (DV) Routing Algorithm



“Millions of routers work in concert to route packets on the Internet –
all based on one simple equation.”



Network Layer Services



Transport layer: TCP, UDP

Network layer

IP

- Addressing conventions
- Datagram format
- Packet handling conventions

ICMP

- Error reporting
- Router “signaling”

Routing protocols

- Path selection
- RIP, OSPF, BGP

forwarding table

Link layer

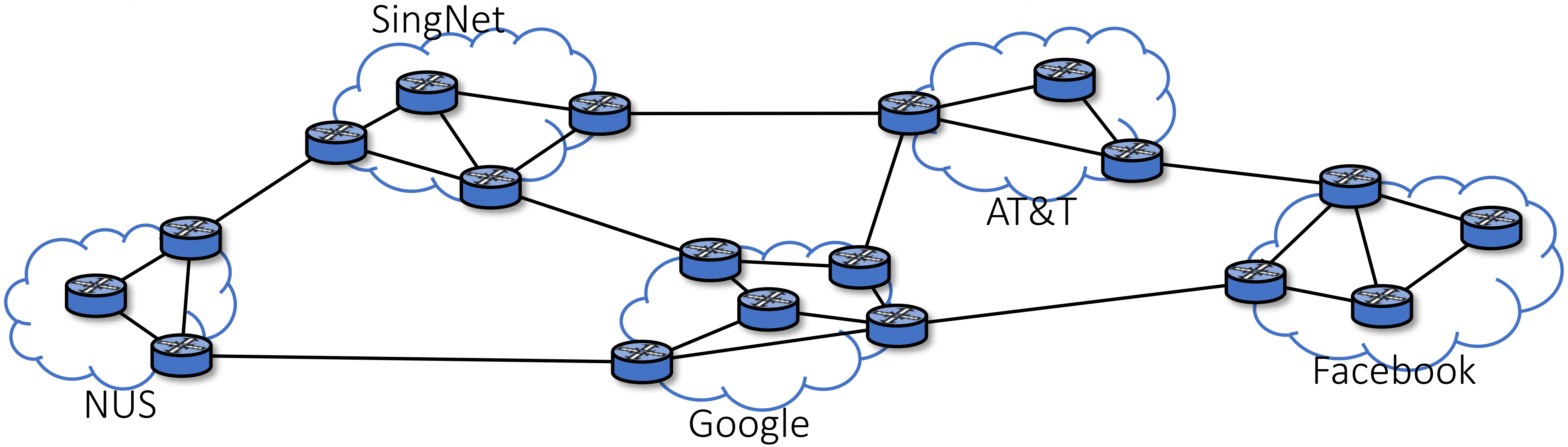
Internet: Network of Networks

The Internet is a "network-of-networks".

- A hierarchy of Autonomous Systems (AS), each owning routers and links.

Due to the size and the decentralized administration of the Internet

- Routing on the Internet is done hierarchically.



Inter-AS routing
vs
Intra-AS routing

Inter-AS routing

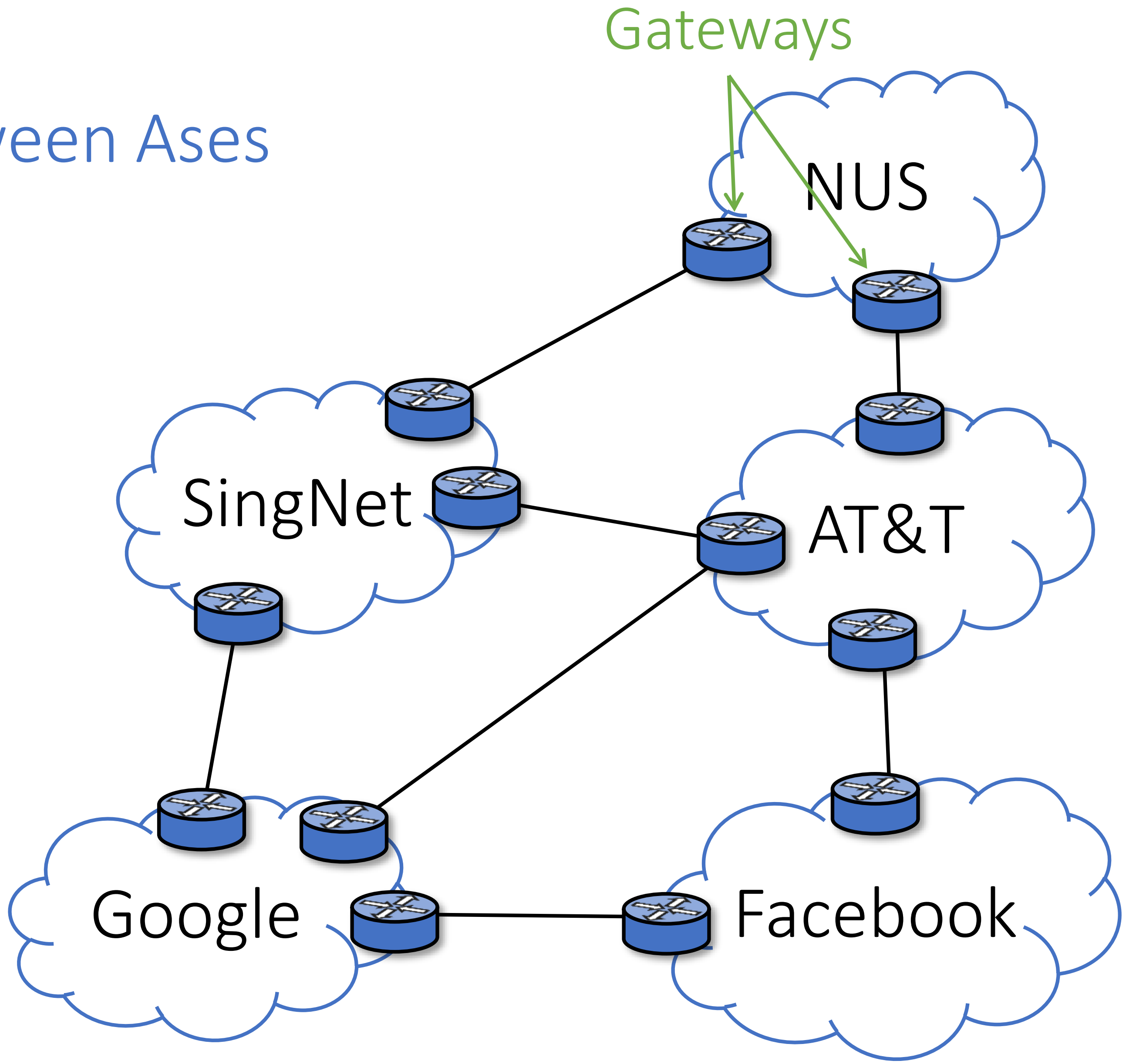
Handles the interfaces between Ases

- De facto protocol: BGP

Routing is based on

- business considerations
- policies

Performance is secondary



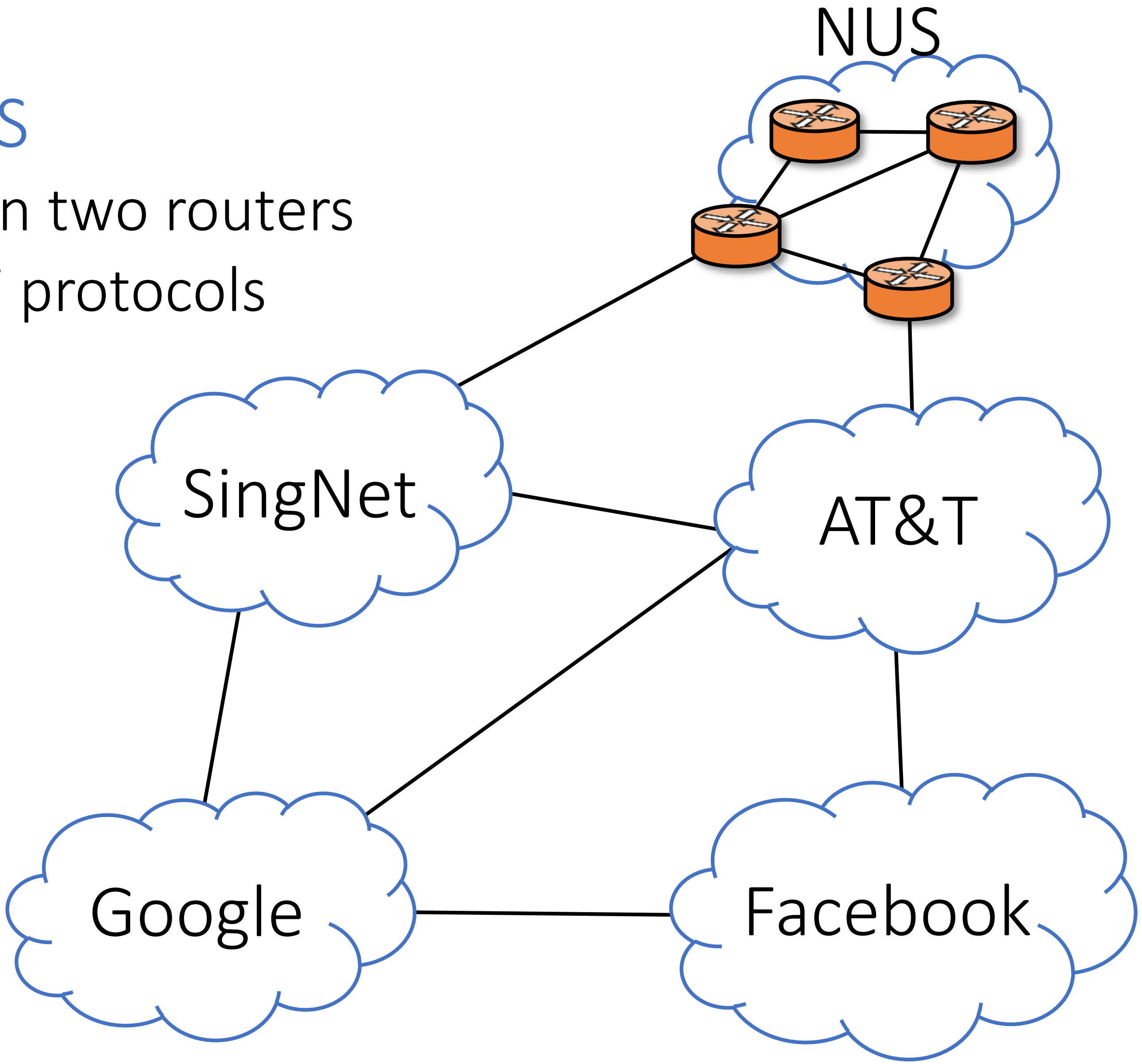
Intra-AS routing

Handles routing within an AS

- Finding a good path between two routers
- Commonly uses RIP or OSPF protocols

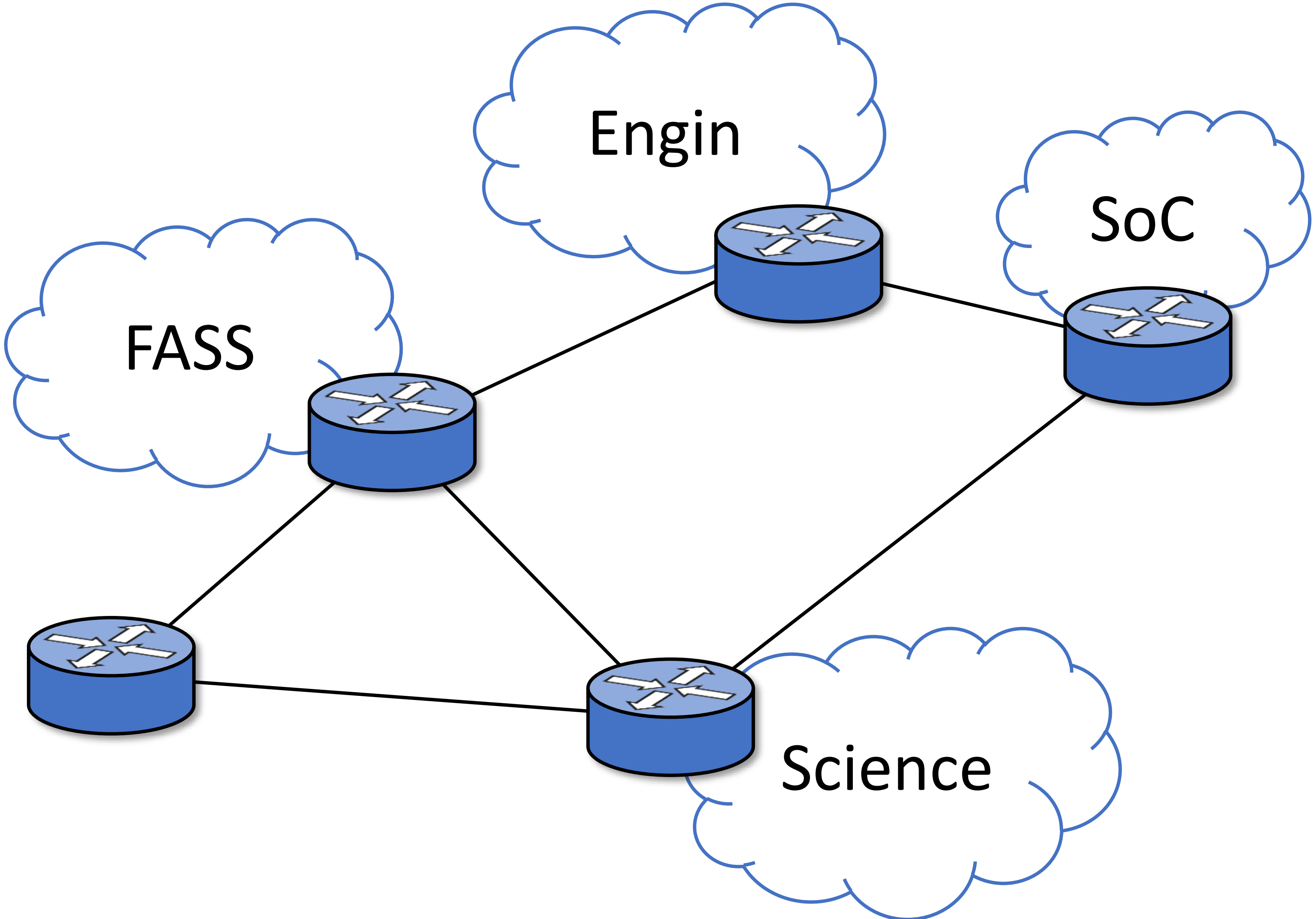
Routing

- handled by single entity
- is performance based



Abstract View of Intra-AS (Intra-domain) routing

Abstract View of Intra-AS Routing

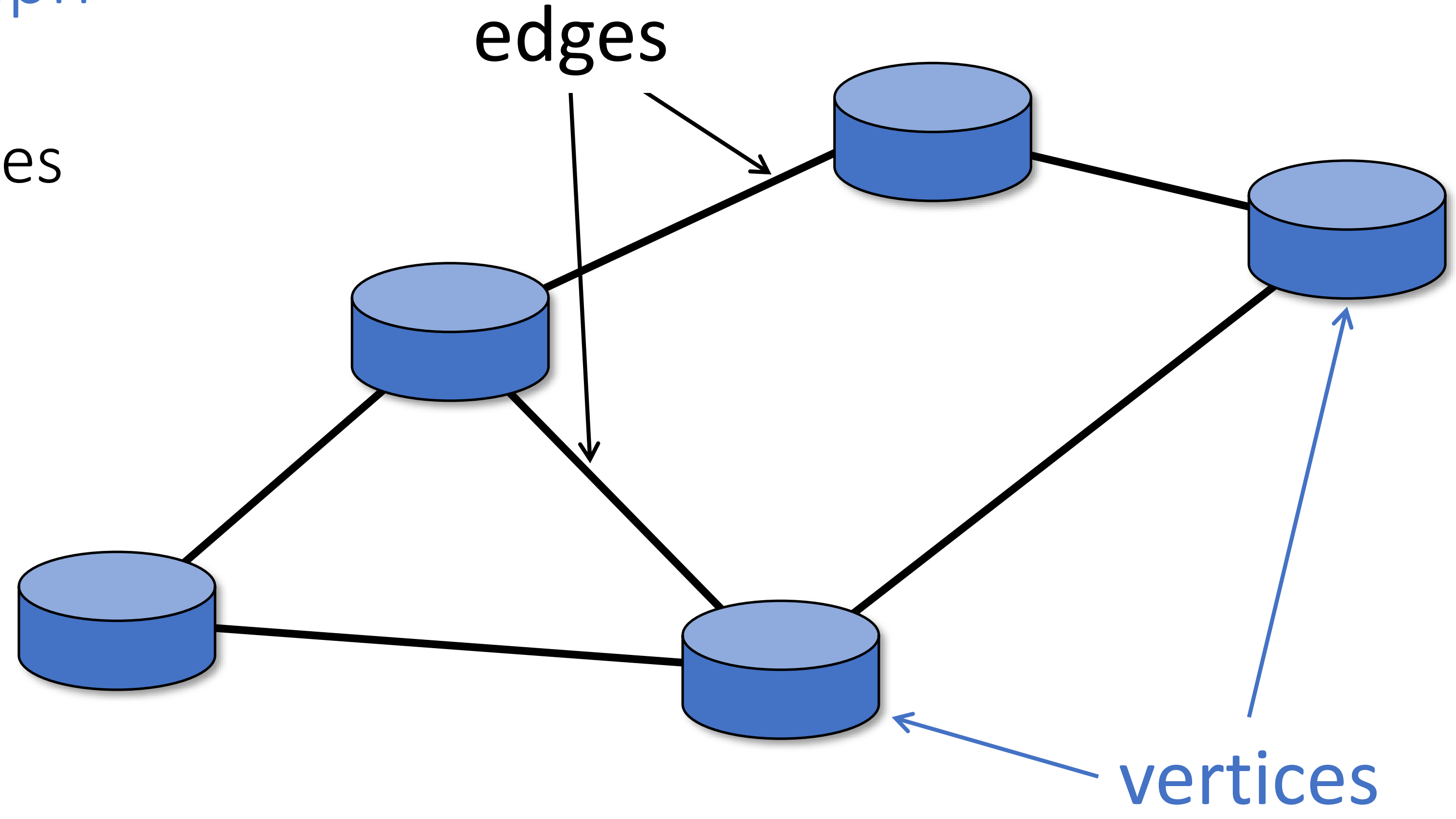


Abstract View of Intra-AS Routing

Abstract away the sub-nets

View network as a graph

- routers are vertices
- physical links are edges



Abstract View of Intra-AS Routing

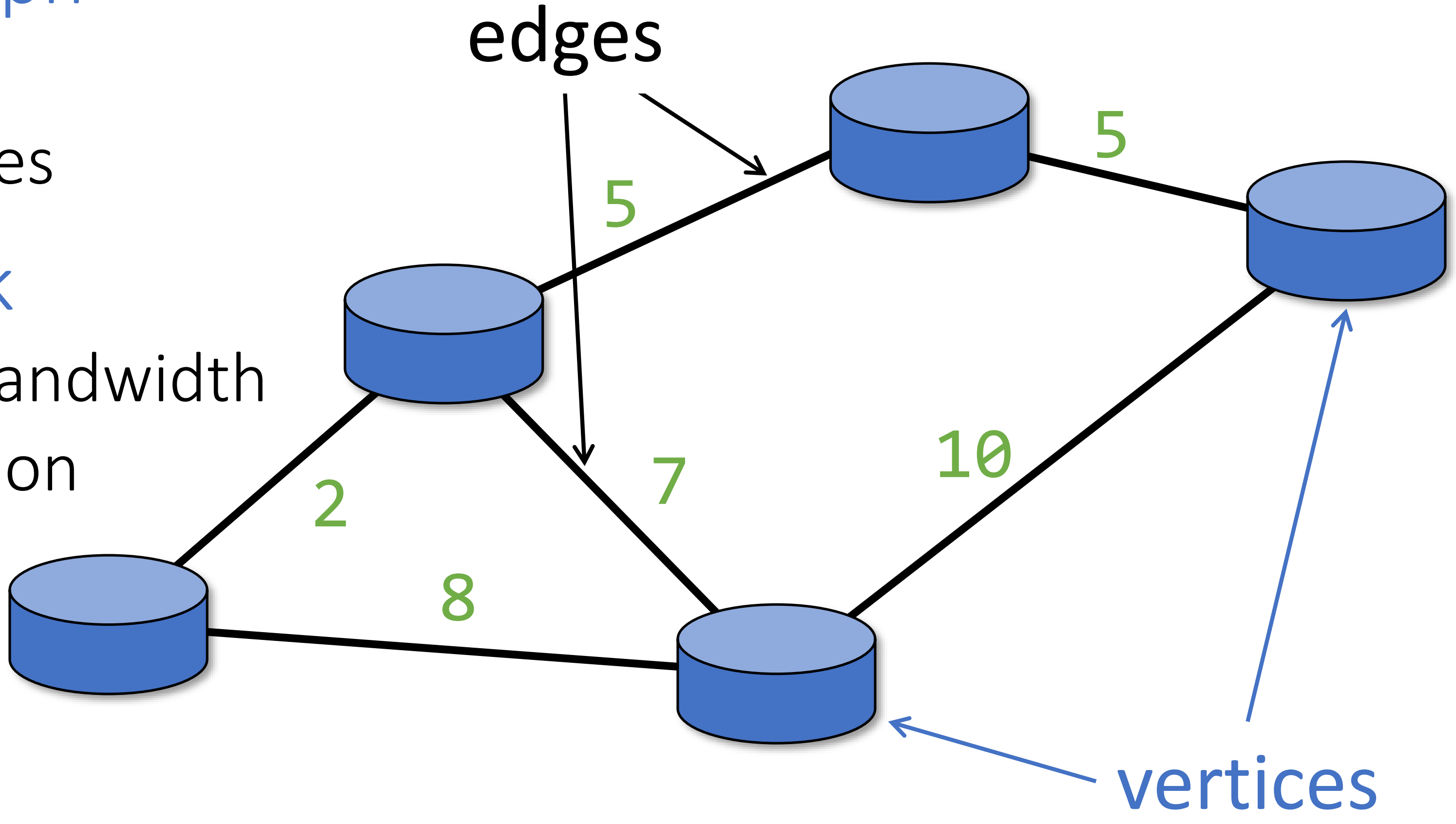
Abstract away the sub-nets

View network as a graph

- routers are vertices
- physical links are edges

Add a cost to each link

- inversely related to bandwidth
- or related to congestion



Routing

Finding the least cost path in the graph

Link-State Algorithms

1. Routers periodically broadcast link cost to each other
    - Every router has the global view of network
2. Compute least cost path locally
    - Use Dijkstra algorithm
    - Not covered in CS2105 ☺

Distance Vector Algorithms

Routers only know

- physically connected neighbours
- and link costs to neighbours

Iterative process

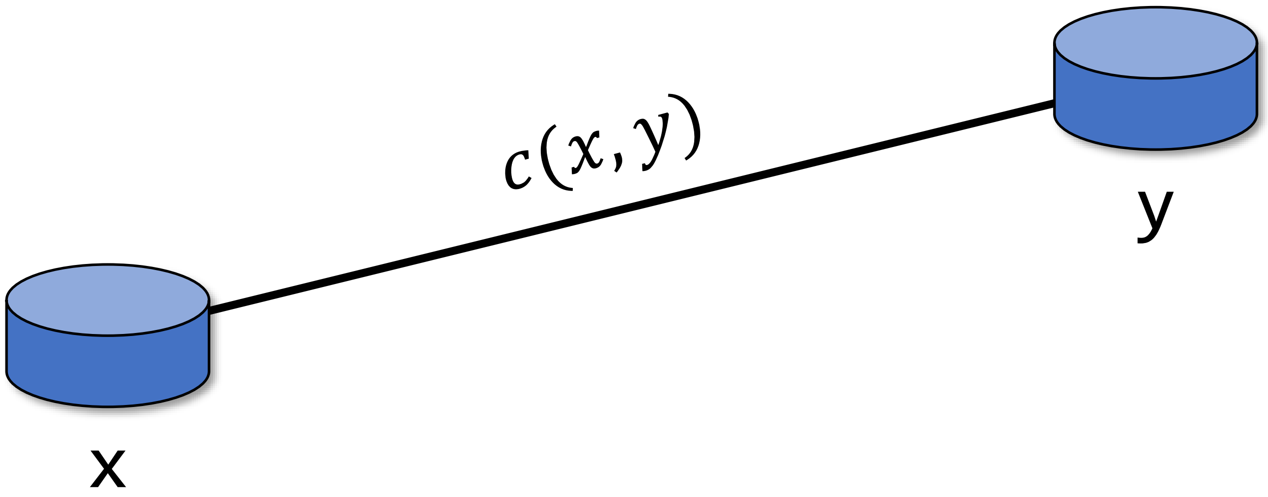
1. Exchange local view with neighbours
2. Update local view based on neighbours’ views
3. Repeat until no further changes

Distance Vector Algorithm

- ✓ Decentralized
- ✓ Self-terminating
- ✓ Iterative
- ✓ Asynchronous

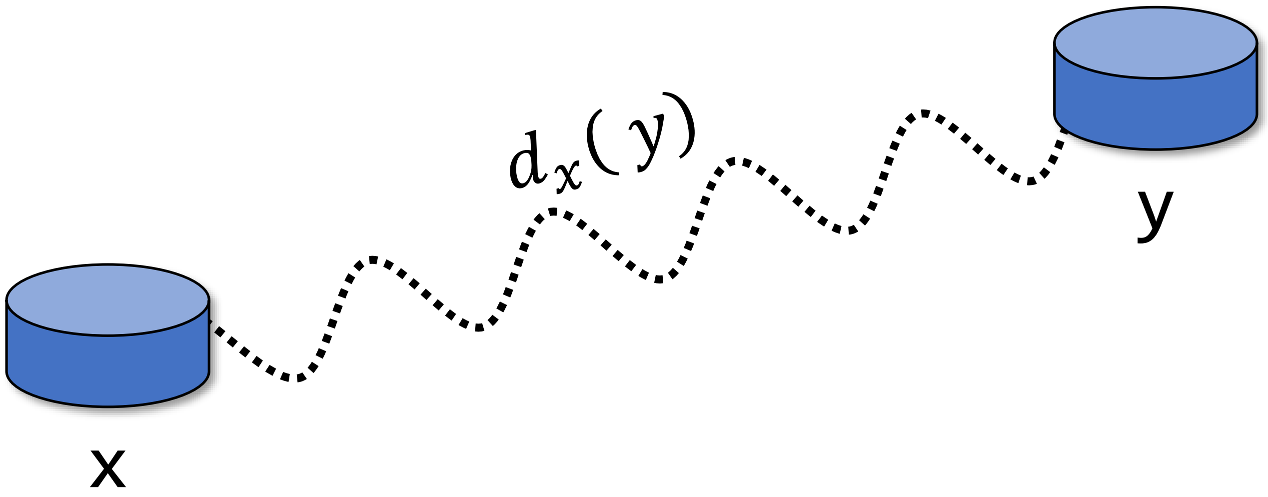
Some Graph Notation

$c(x, y)$: cost of link between x and y



Some Graph Notation

$d_x(y)$: least cost of path from x to y



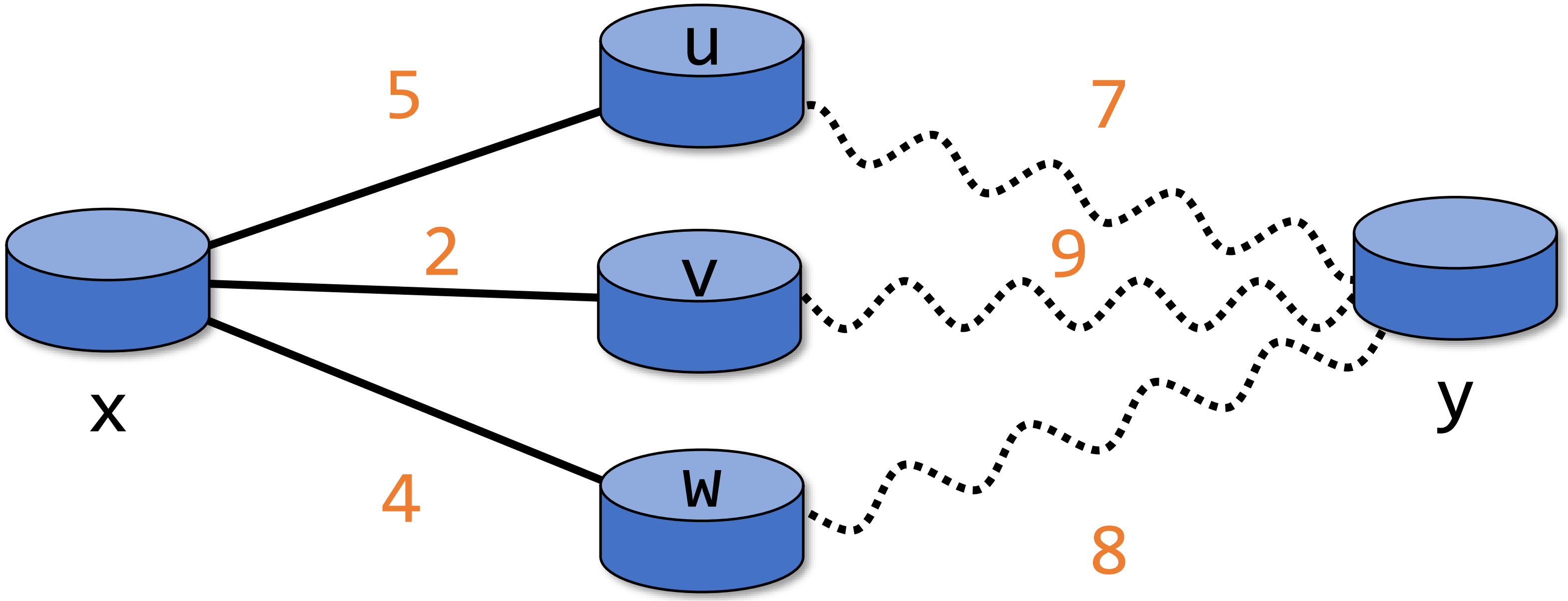
Bellman-Ford Equation

$$d_x(y) = \min_i\{c(x,i) + d_i(y)\}$$

$$d_x(y) = min\begin{Bmatrix} c(x,u) + d_u(y) \\ c(x,v) + d_v(y) \\ c(x,w) + d_w(y) \end{Bmatrix}$$

$$= min\{12, 11, 12\}$$

$$= 11$$



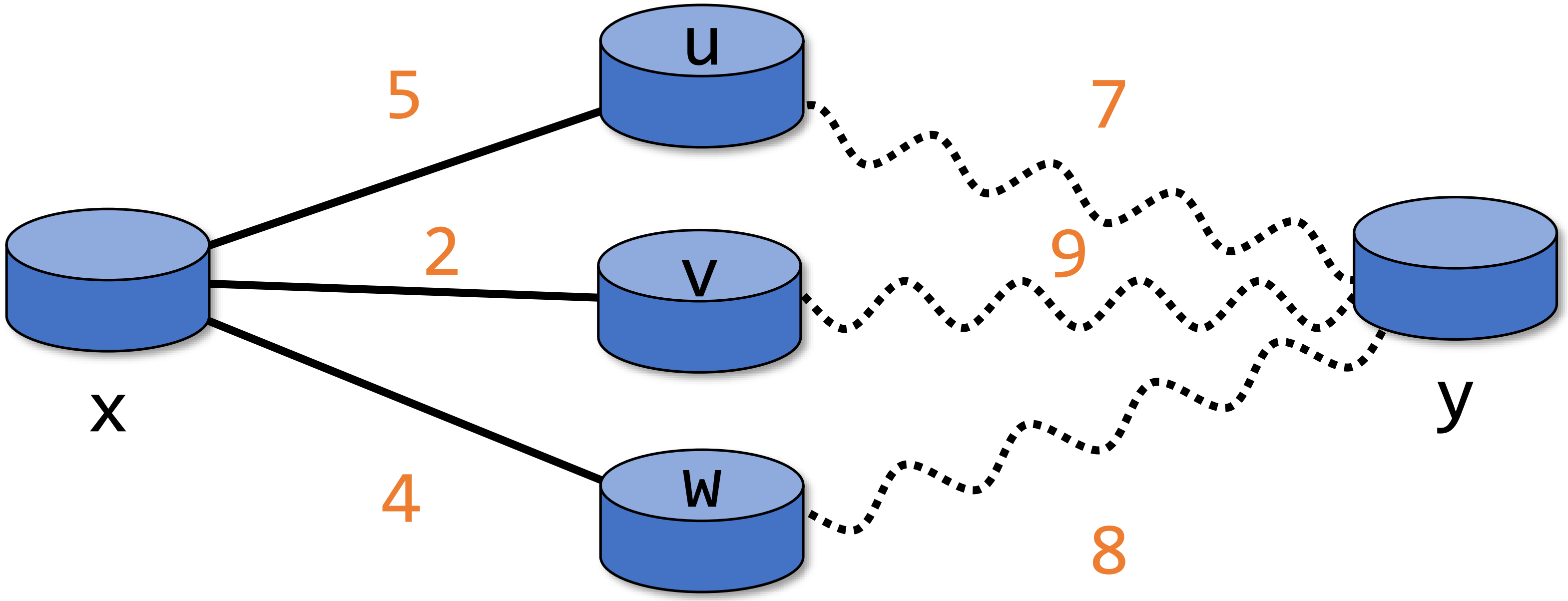
Bellman-Ford Equation

To find least cost path

- x needs to know the cost from each of its direct neighbours to y

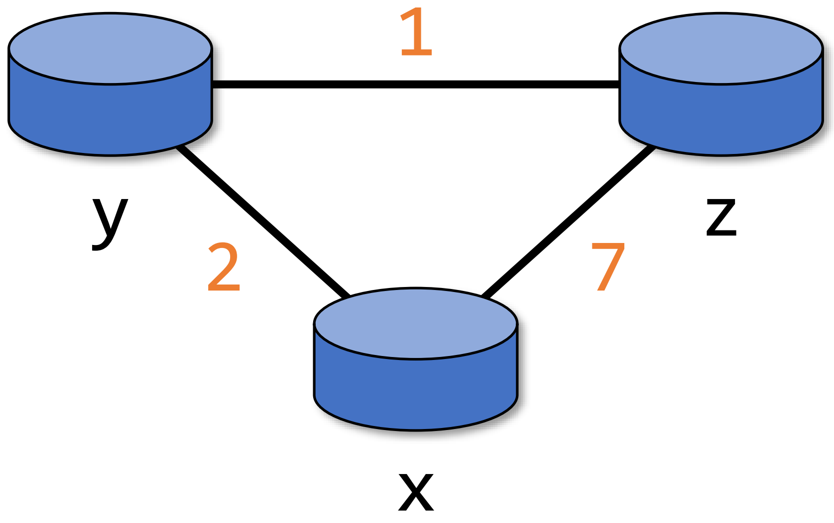
Each neighbor i sends its distance vector (y, k) to x

- informing x that its cost to y is k



Now x knows that to reach y, it should be forward to v and the total cost would be $2 + 9 = 11$

Bellman-Ford Example



cost to

from

	x	y	z
x			
y			
z			

y's view

cost to

from

	x	y	z
x			
y			
z			

x's view

cost to

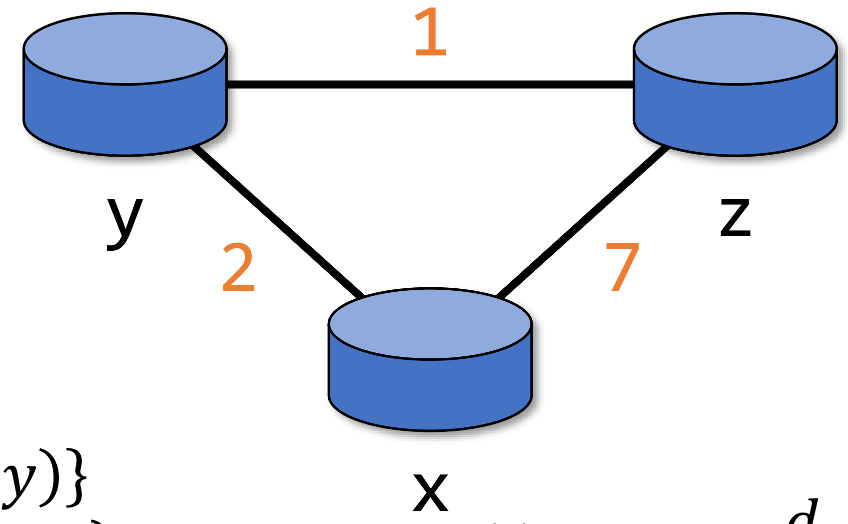
from

	x	y	z
x			
y			
z			

z's view

Bellman-Ford Example

$$\begin{pmatrix} x & 2 \\ y & 0 \\ z & 1 \end{pmatrix}$$



$$\begin{pmatrix} x & 7 \\ y & 1 \\ z & 0 \end{pmatrix}$$

$$\begin{aligned} d_x(\,y) &= \min_i\{c(x,i) + d_i(y)\} \\ &= \min\begin{Bmatrix} c(x,y) + d_y(y) \\ c(x,z) + d_z(y) \end{Bmatrix} \end{aligned}$$

$$d_x(\,z) = \min\begin{Bmatrix} c(x,y) + d_y(z) \\ c(x,z) + d_z(z) \end{Bmatrix}$$

cost to

from

	x	y	z
x	0	2	7
y			
z			

x’s view

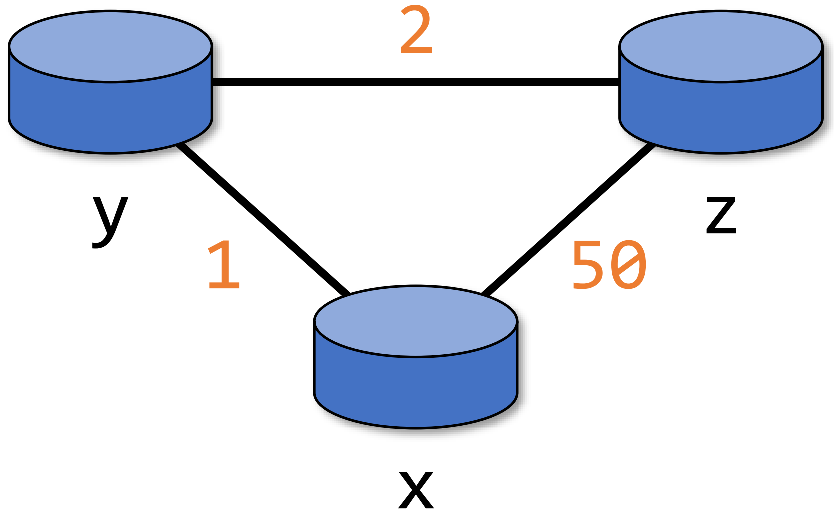
Example: In steady state

cost to

from

	x	y	z
x	0	1	3
y	1	0	2
z	3	1	0

y’s view



cost to

from

	x	y	z
x	0	1	3
y	1	0	2
z	3	2	0

x’s view

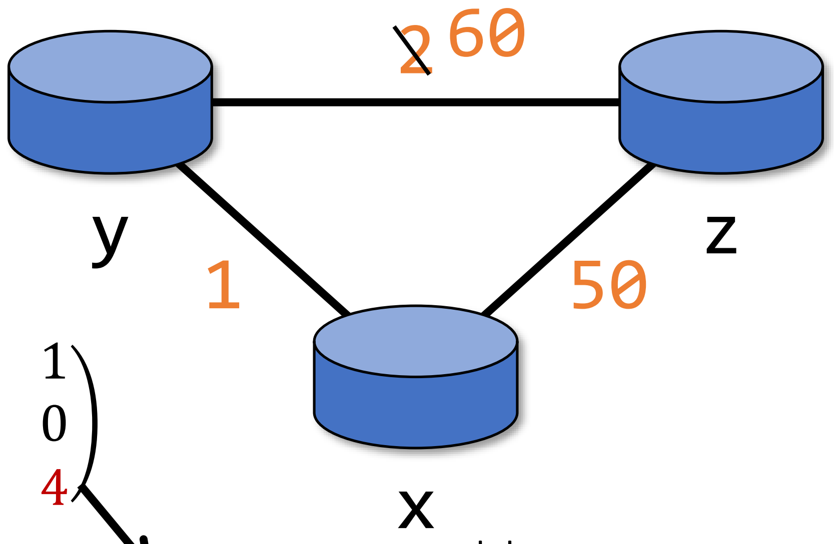
cost to

from

	x	y	z
x	0	1	3
y	1	0	2
z	3	2	0

z’s view

Example: A link change



cost to

from	x	y	z
x	0	1	3
y	1	0	2
z	3	1	0

y's view

$$\begin{pmatrix} x & 1 \\ y & 0 \\ z & 4 \end{pmatrix}$$

$$\begin{pmatrix} x & 0 \\ y & 1 \\ z & 5 \end{pmatrix}$$

cost to

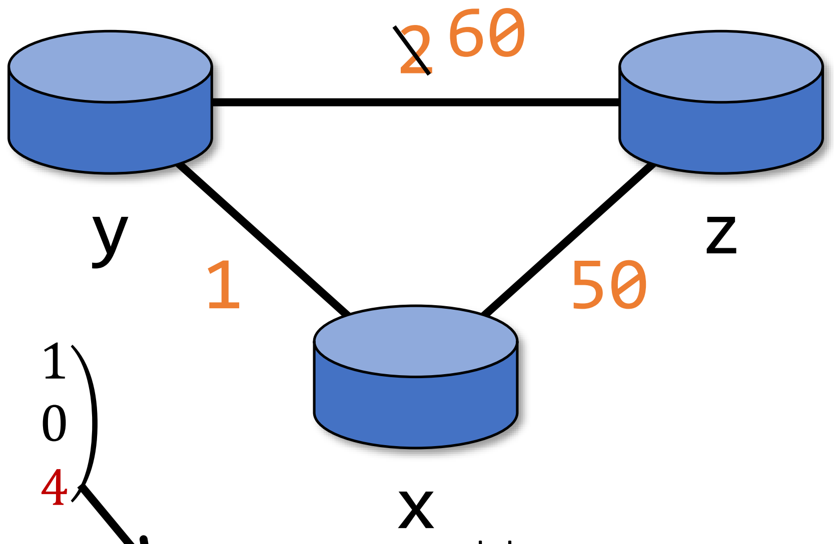
from	x	y	z
x	0	1	5
y	1	0	4
z	3	2	0

x's view

Cost of direct link to z increases

- y thinks shortest route to z in now via x
- sends DV with x
- Now x thinks shortest route to z is via y
- sends DV with y

Count to Infinity



cost to

from	x	y	z
x	0	1	3
y	1	0	4
z	3	1	0

y's view

$$\begin{pmatrix} x & 1 \\ y & 0 \\ z & 4 \end{pmatrix}$$

$$\begin{pmatrix} x & 0 \\ y & 1 \\ z & 5 \end{pmatrix}$$

cost to

from	x	y	z
x	0	1	5
y	1	0	4
z	3	2	0

x's view

- Both x and y will slowly exchange and update DV until fixed-point
- Will take a really long time
- Meanwhile, where will the packets be forwarded?

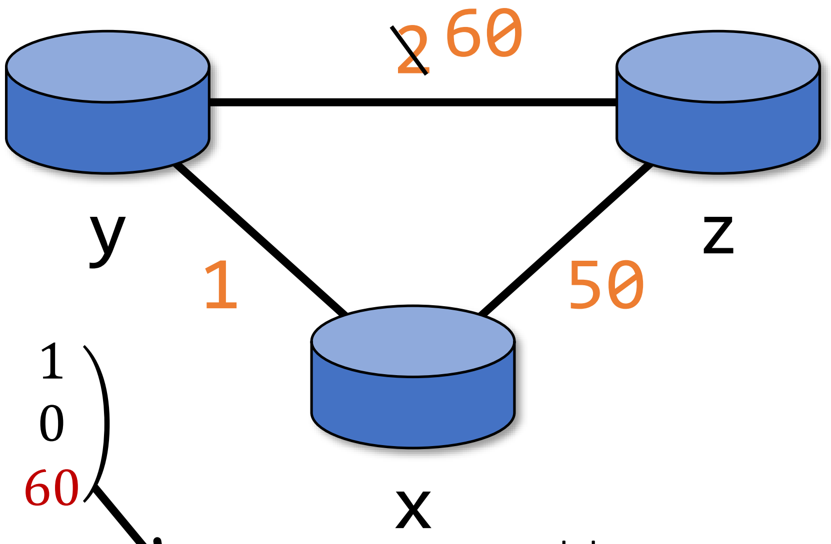
Solution: Poisoned Reverse

cost to

from

	x	y	z
x	0	1	∞
y	1/x	0	60/z
z	3	1	0

y's view



$$\begin{pmatrix} x & 1 \\ y & 0 \\ z & 60 \end{pmatrix}$$

$$\begin{pmatrix} x & 0 \\ y & \infty \\ z & \infty \end{pmatrix}$$

cost to

from

	x	y	z
x	0	1/y	50/z
y	1	0	2
z	3	2	0

x's view

Maintain "via" in routing table

- Set cost to infinity if route is via the router

Now y thinks x has no route to z

- Will not update to route through x

Comparison

	Link State	Distance Vector
Message overhead	✗	✓
Convergence speed	✓	✗
Robustness (error)	✗	✓

Routing Information Protocol (RIP)

Implements the DV algorithm

- Uses hop count as the cost metric (i.e., insensitive to network congestion).

Entries in the routing table are aggregated subnet masks

- routing to destination subnet

Exchange routing table every 30 seconds

- over UDP port 520

“Self-repairing”

- if no update from a neighbour router for 3 minutes
- assume neighbour has failed.

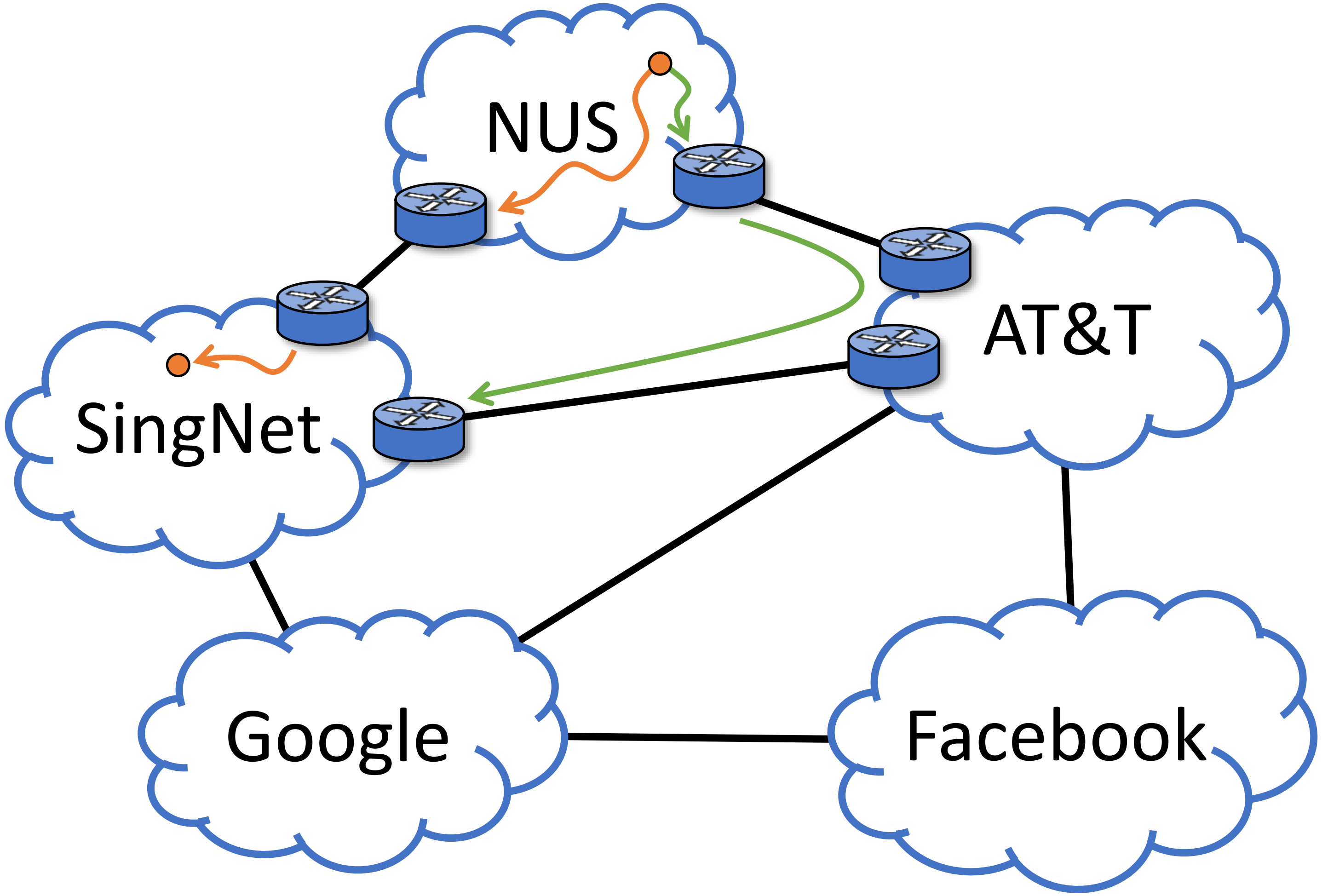
Inter-AS routing
BGP: Border Gateway Protocol

(not covered)

Hot Potato Routing

Route to AS whose gateway has the least cost (nearest)

Hot Potato Routing



Summary

Intra-AS Routing

Link State

- OSPF

Distance Vector

- Bellman-Ford equation
- Count to infinity
- Reverse poisoning
- RIP

Hot Potato Routing

Inter-AS Routing

BGP

- not covered