CS 43: Computer Networks

Routing Algorithms & Routing on the Internet

November 24, 2025

SWARTHMORE COLLEGE

Adapted from Slides by: Kurose & Ross, D. Choffnes, K. Webb, J. Rexford

The Network Layer!

Application: the application (e.g., the Web, Email)

Transport: end-to-end connections, reliability

Network: routing

Link (data-link): framing, error detection

Physical: 1’s and 0’s/bits across a medium (copper, the air, fiber)

Routing Algorithm Classes

Link State (Global)

- Routers maintain cost of each link in the network.
- Connectivity/cost changes flooded to all routers.
- Converges quickly (less inconsistency, looping, etc.).
- Limited network sizes.

Distance Vector (Decentralized)

- Routers maintain next hop & cost of each destination.
- Connectivity/cost changes iteratively propagate from neighbor to neighbor.
- Requires multiple rounds to converge.
- Scales to large networks.

Bellman-Ford Equation

let

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

then

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x

Distance Vectors

- Let $D_x(y)$ = least known cost from x to y (by router x)

- Node x:
  - Knows cost to each neighbor v: $c(x,v)$

  - Maintains its neighbors' distance vectors.
    For each neighbor v, x maintains: $\mathbf{D}_v = [D_v(y): y \in N]$

- **As opposed to link state:**
  - **Only keeps state for yourself and direct neighbors**

Distance Vector Algorithm

- Periodically, each node sends its own distance vector to neighbors
- Upon receiving new DV from neighbor, update its local DV using B-F equation:

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

- Under typical conditions, $D_x(y)$ will converge to the least cost $d_x(y)$

Distance Vector Algorithm

Iterative, asynchronous:

Iteration when:

- Local link cost change
- DV update from neighbor
- Periodic timer

Distributed:

- Each node knows only a portion of global link info

each node:

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

↓

recompute estimates

↓

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

Distance Vector Algorithm Example

node x table

from \ *cost to*	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

node y table

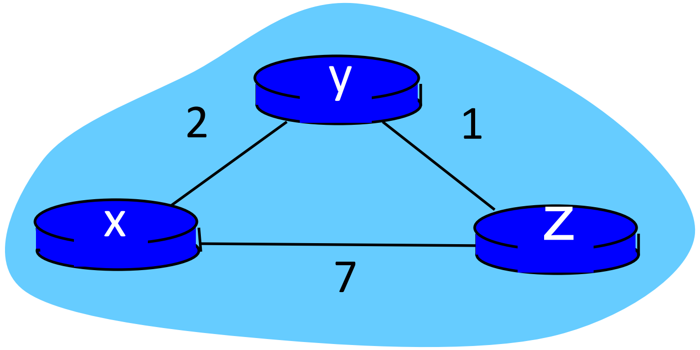
from \ *cost to*	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

from \ *cost to*	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

time

Every node starts with the distance to its neighbors and every other distance is ∞



$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

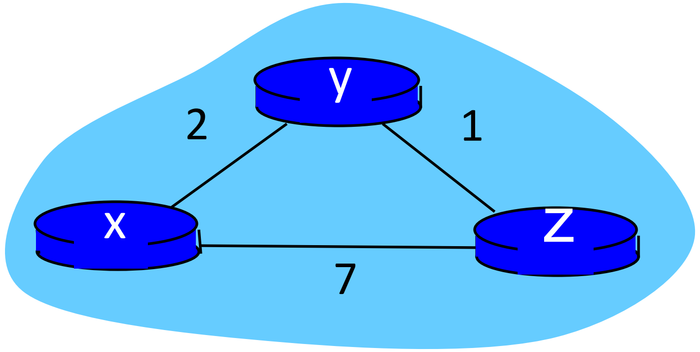
node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

time

Every node sends its distance vector: the smallest distance it is has for every node.

Then each node computes its new minimal distances.



$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\} = \min\{2+0\ ,\ 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\} = \min\{2+1\ ,\ 7+0\} = 3$$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

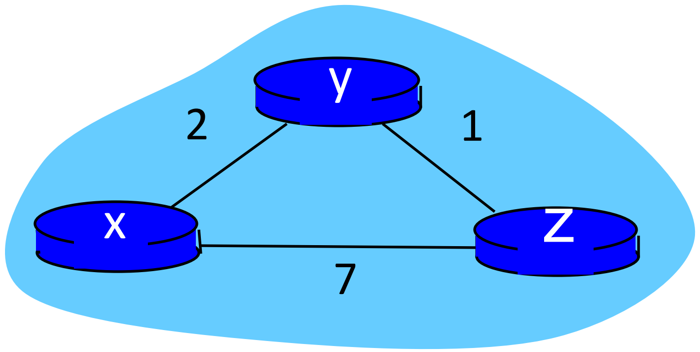
from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

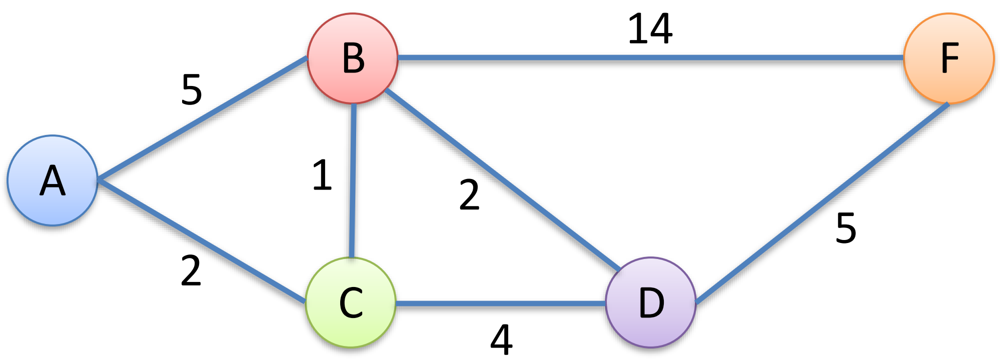
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

time

Each time a node has a new distance vector (found a shorter path) it will send that update to its neighbors.

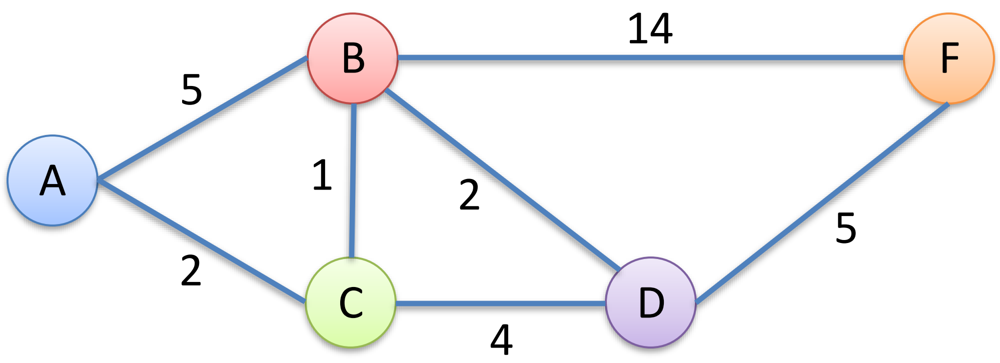


Distance Vector Example



- Same network as Dijkstra's example, without node E.
- What I'll show you next is routing table (of distance vectors) at each router.

Distance Vector – Round 0



Routers populate their forwarding table by taking the row minimum.

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C		1		
D			2	
F				14

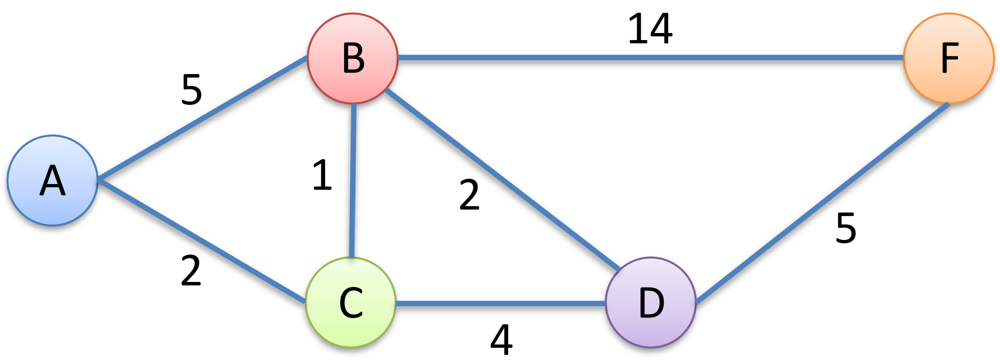
Router C

Via→ ↓ To	A	B	D
A	2		
B		1	
D			4
F			

Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Distance Vector – Round 0



Router exchange their local vectors with direct neighbors.
We'll assume they all exchange at once (synchronous). (Not realistic)

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C		1		
D			2	
F				14

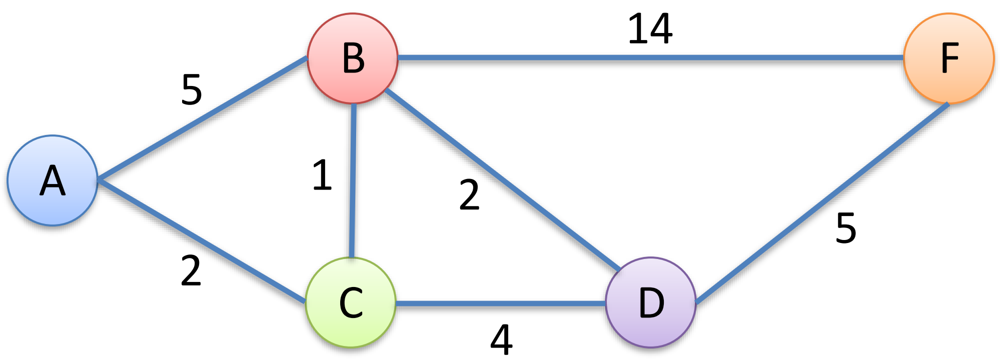
Router C

Via→ ↓ To	A	B	D
A	2		
B		1	
D			4
F			

Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Distance Vector – Round 1



A will send to neighbors (B & C):
I can get to B in 5 and C in 2.

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C	?	1		
D			2	
F				14

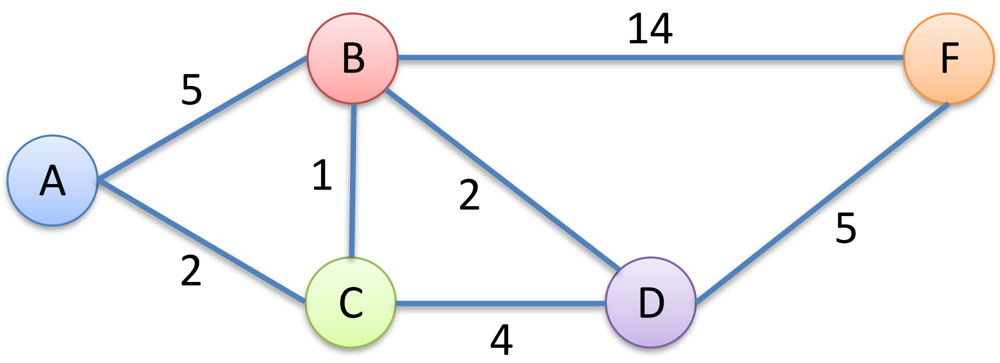
Router C

Via→ ↓ To	A	B	D
A	2		
B	?	1	
D			4
F			

Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Distance Vector – Round 1



A will send to neighbors (B & C):
I can get to B in 5 and C in 2.

Router F

Via→ ↓ To	B	D
A		
B	14	
C		
D		5

Router A

Via→ ↓ To	B	C
B	5	
C		2
D		
F		

Router B

Via→ ↓ To	A	C	D	F
A	5			
C	7	1		
D			2	
F				14

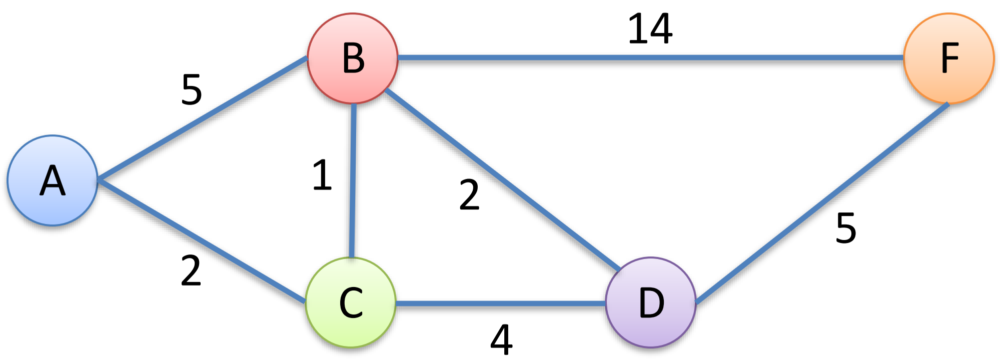
Router C

Via→ ↓ To	A	B	D
A	2		
B	7	1	
D			4
F			

Router D

Via→ ↓ To	B	C	F
A			
B	2		
C		4	
F			5

Distance Vector – Round 1



B will send to neighbors (A, C, D, F):
I can get to A in 5, C in 1, D in 2, and F in 14.

Router F

Via→ ↓ To	B	D
A	19	
B	14	
C	15	
D	16	5

Router A

Via→ ↓ To	B	C
B	5	
C	6	2
D	7	
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5			
C	7	1		
D			2	
F				14

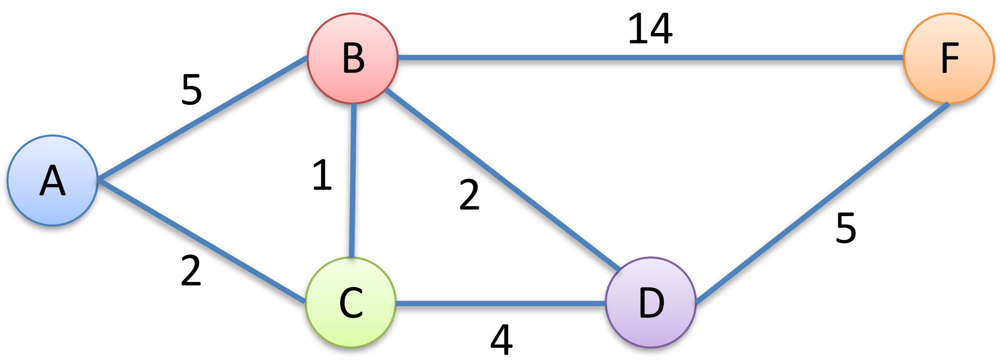
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	
D		3	4
F		15	

Router D

Via→ ↓ To	B	C	F
A	7		
B	2		
C	3	4	
F	16		5

Distance Vector – Round 1



C will send to neighbors (A, B, D):
I can get to A in 2, B in 1, and D in 4.

Router F

Via→ ↓ To	B	D
A	19	
B	14	
C	15	
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1		
D		5	2	
F				14

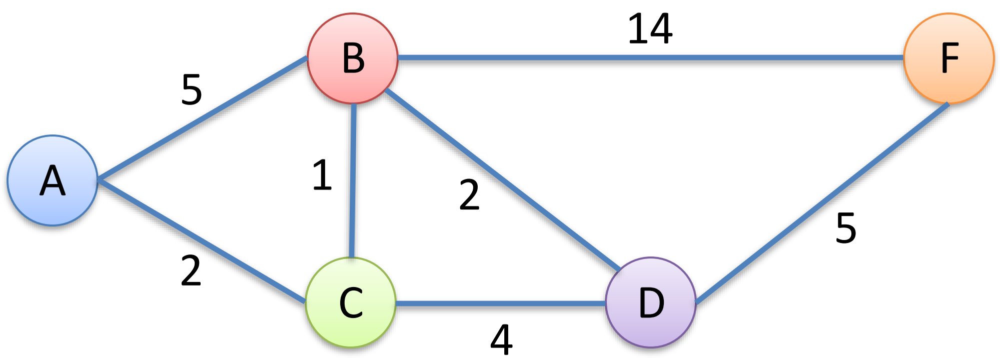
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	
D		3	4
F		15	

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	
C	3	4	
F	16		5

Distance Vector – Round 1



D will send to neighbors (B, C, F):
I can get to B in 2, C in 4, and F in 5.

Router F

Via→ ↓ To	B	D
A	19	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	
F			7	14

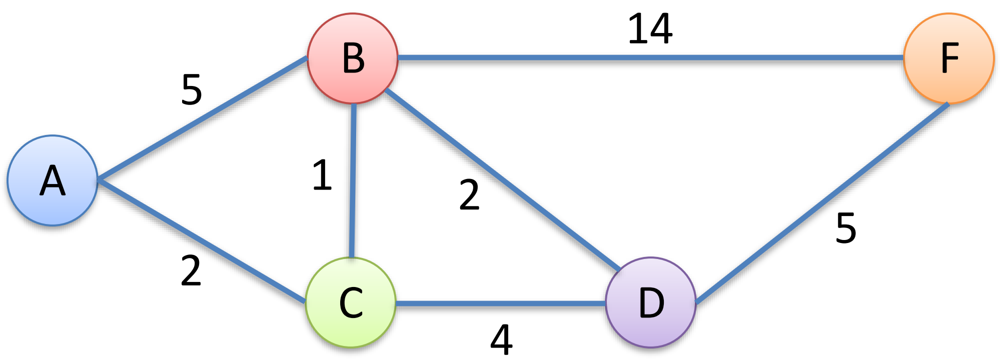
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	6
D		3	4
F		15	9

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	
C	3	4	
F	16		5

Distance Vector – Round 1



F will send to neighbors (B, D):
I can get to B in 14, D in 5.

Router F

Via→ ↓ To	B	D
A	19	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

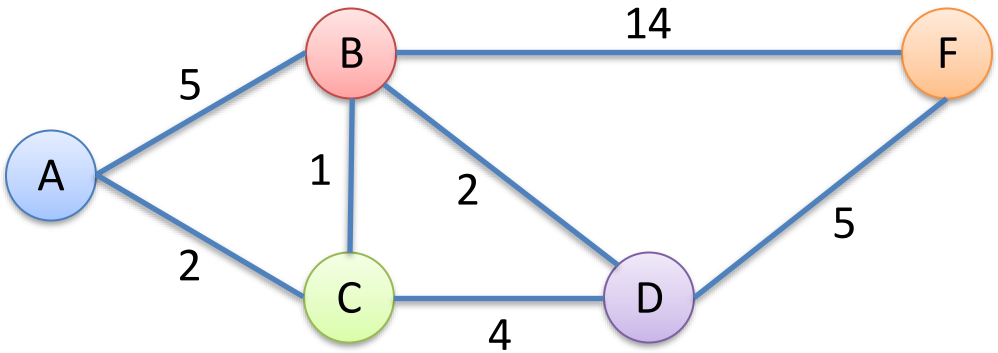
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	6
D		3	4
F		15	9

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	19
C	3	4	
F	16		5

At the end of round 1, how many routers need to update their forwarding tables?



A – 1, B – 2, C – 3, D – 4, E – 5

Router F

Via→ ↓ To	B	D
A	19	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

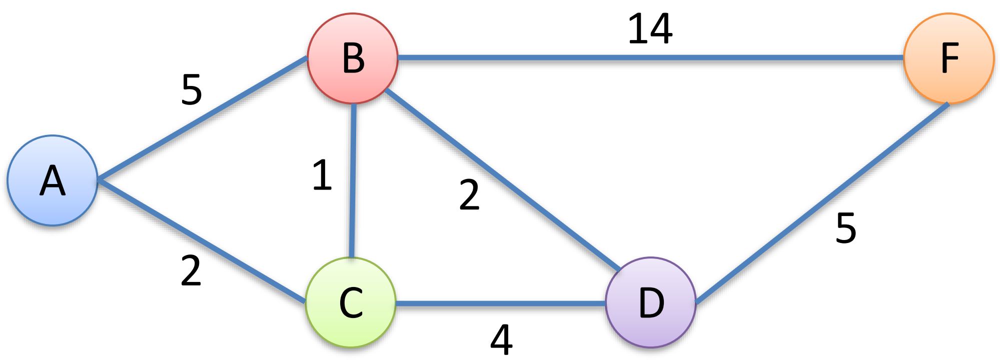
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	6
D		3	4
F		15	9

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	19
C	3	4	
F	16		5

Distance Vector – Round 2



Each router advertises the best cost it has to each destination.
Nothing new to learn from A or F, so we'll skip their announcements.

Router F

Via→ ↓ To	B	D
A	19	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	19	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

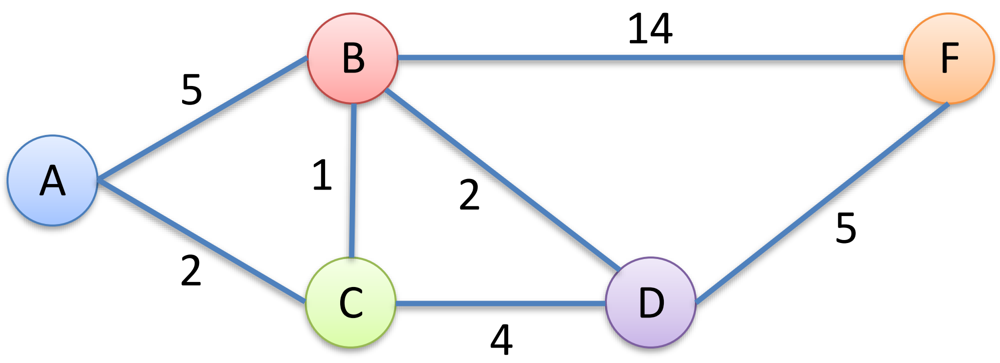
Router C

Via→ ↓ To	A	B	D
A	2	6	
B	7	1	6
D		3	4
F		15	9

Router D

Via→ ↓ To	B	C	F
A	7	6	
B	2	5	19
C	3	4	
F	16		5

Distance Vector – Round 2



B will send to neighbors (A, C, D, F):
I can get to A in 3, C in 1, D in 2, and F in 7.

Router F

Via→ ↓ To	B	D
A	17	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	12	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

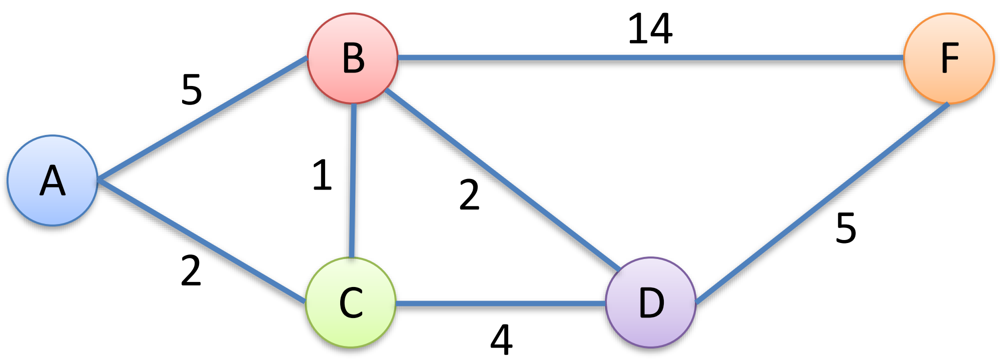
Router C

Via→ ↓ To	A	B	D
A	2	4?	
B	7	1	6
D		3	4
F		8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	
B	2	5	19
C	3	4	
F	9?		5

Distance Vector – Round 2



C will send to neighbors (A, B, D):
I can get to A in 2, B in 1, D in 3, and F in 9.

Router F

Via→ ↓ To	B	D
A	17	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	5
F	12	11

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		4?	2	19
F		10	7	14

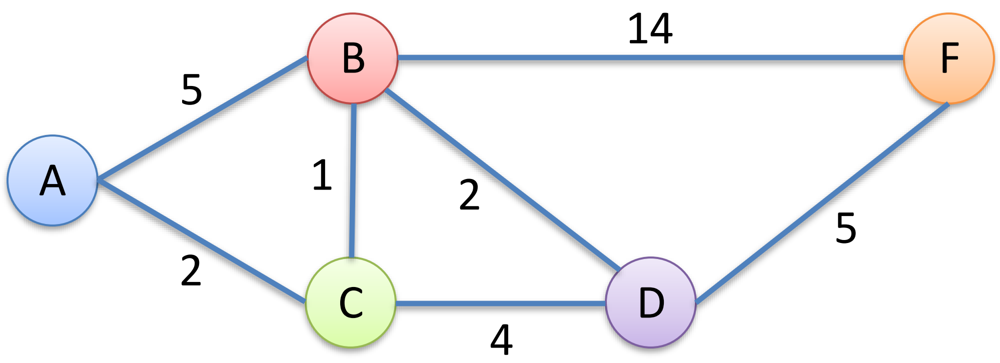
Router C

Via→ ↓ To	A	B	D
A	2	4?	
B	7	1	6
D		3	4
F		8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	
B	2	5	19
C	3	4	
F	9?	13?	5

Distance Vector – Round 2



This process repeats for a while…

Router F

Via→ ↓ To	B	D
A	17	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	5
F	12	11

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		4?	2	19
F		10	7	14

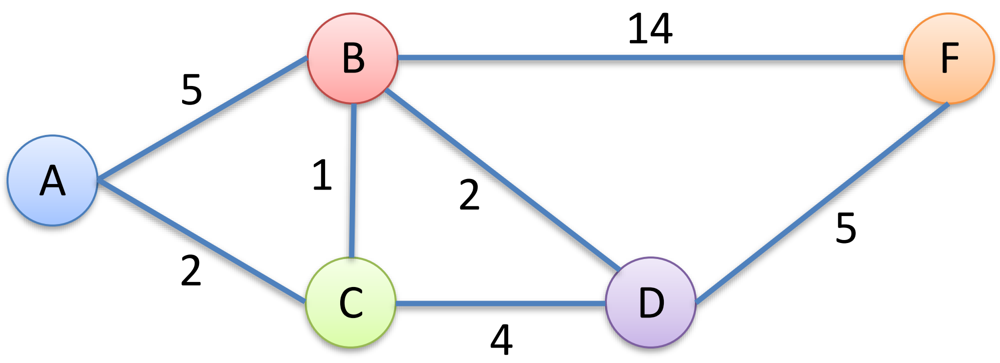
Router C

Via→ ↓ To	A	B	D
A	2	4?	
B	7	1	6
D		3	4
F		8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	
B	2	5	19
C	3	4	
F	9?	13?	5

Distance Vector – Convergence



Eventually, we reach a converged state.

Router F

Via→ ↓ To	B	D
A	17	10
B	14	7
C	15	8
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	5
F	12	10

Router B

Via→ ↓ To	A	C	D	F
A	5	3	7	24
C	7	1	4	22
D	10	4	2	19
F	15	9	7	14

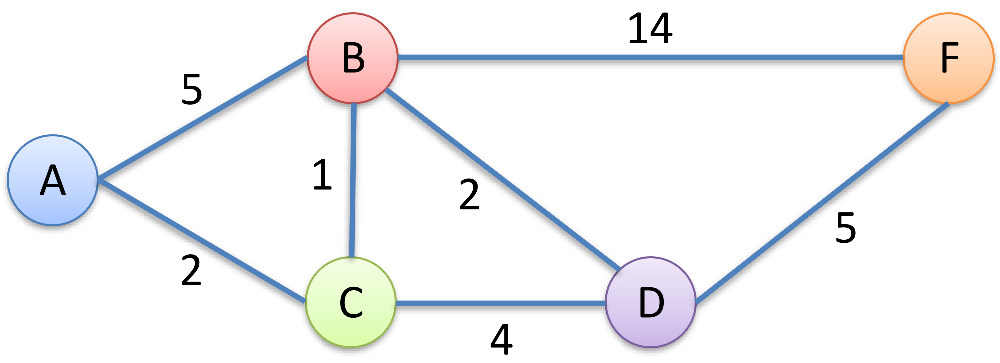
Router C

Via→ ↓ To	A	B	D
A	2	4	9
B	7	1	6
D	7	3	4
F	12	8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	15
B	2	5	12
C	3	4	13
F	9	12	5

Distance Vector – Convergence



Final forwarding tables:

Router F

Via→ ↓ To	B	D
A	17	10
B	14	7
C	15	8
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	5
F	12	10

Router B

Via→ ↓ To	A	C	D	F
A	5	3	7	24
C	7	1	4	22
D	10	4	2	19
F	15	9	7	14

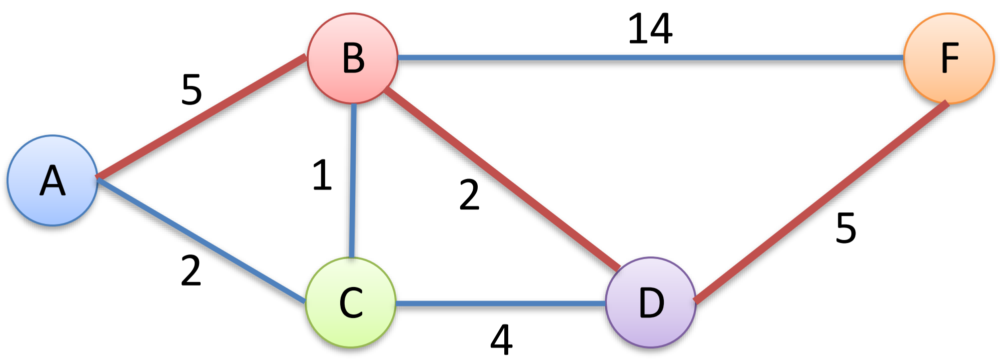
Router C

Via→ ↓ To	A	B	D
A	2	4	9
B	7	1	6
D	7	3	4
F	12	8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	15
B	2	5	12
C	3	4	13
F	9	12	5

Of the links in red below, for how many would a failure cause a loop?



A – 0, B – 1, C – 2, D – 3

Consider the failures independently (not all at the same time).

Router F

Via→ ↓ To	B	D
A	17	10
B	14	7
C	15	8
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	5
F	12	10

Router B

Via→ ↓ To	A	C	D	F
A	5	3	7	24
C	7	1	4	22
D	10	4	2	19
F	15	9	7	14

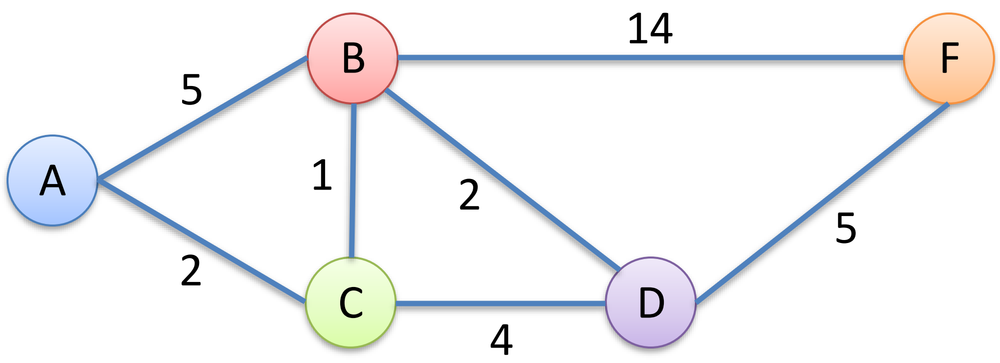
Router C

Via→ ↓ To	A	B	D
A	2	4	9
B	7	1	6
D	7	3	4
F	12	8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	15
B	2	5	12
C	3	4	13
F	9	12	5

Rewind: Distance Vector – Round 2



B will send to neighbors (A, C, D, F):
I can get to A in 3, C in 1, D in 2, and F in 7.

Router F

Via→ ↓ To	B	D
A	17	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	12	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

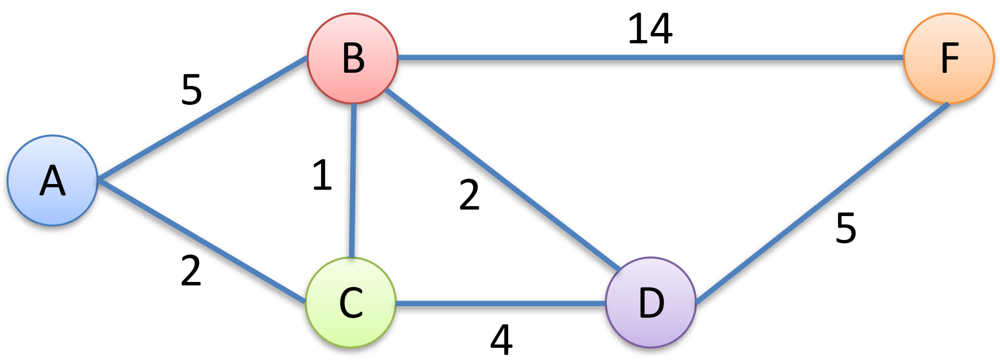
Router C

Via→ ↓ To	A	B	D
A	2	4?	
B	7	1	6
D		3	4
F		8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	
B	2	5	19
C	3	4	
F	9?		5

Rewind: Distance Vector – Round 2



Poisoned reverse: Don't advertise a lower value to a neighbor if you go through that neighbor to get there!

Router F

Via→ ↓ To	B	D
A	17	
B	14	7
C	15	9
D	16	5

Router A

Via→ ↓ To	B	C
B	5	3
C	6	2
D	7	6
F	12	

Router B

Via→ ↓ To	A	C	D	F
A	5	3		
C	7	1	6	
D		5	2	19
F			7	14

Router C

No!
∞

Via→ ↓ To	A	B	D
A	2	~~4?~~	
B	7	1	6
D		3	4
F		8	9

Router D

Via→ ↓ To	B	C	F
A	5	6	
B	2	5	19
C	3	4	
F	9?		5

Loop-prevention

- Route poisoning helps prevent loops, but doesn’t guarantee loop free.
- Other mechanisms help too
- <u>There will always be a window of vulnerability</u>

Real Protocols

Link State

- Open Shortest Path First (OSPF)
- Intermediate system to intermediate system (IS-IS)

Distance Vector

- Routing Information Protocol (RIP)
- Interior Gateway Routing Protocol (IGRP – Cisco)
- Border Gateway Protocol (BGP) (sort of, we'll look at this next)

Summary

Link State

- + Fast convergence (reacts to events quickly)
- + Small window of inconsistency

- – Large number of messages sent on events
- – Large routing tables as network size grows

Distance Vector

- + Distributed (small tables)
- + No flooding (fewer messages)

- – Slower convergence
- – Larger window of inconsistency

Intra-AS Routing

- Also known as *interior gateway protocols (IGP)*

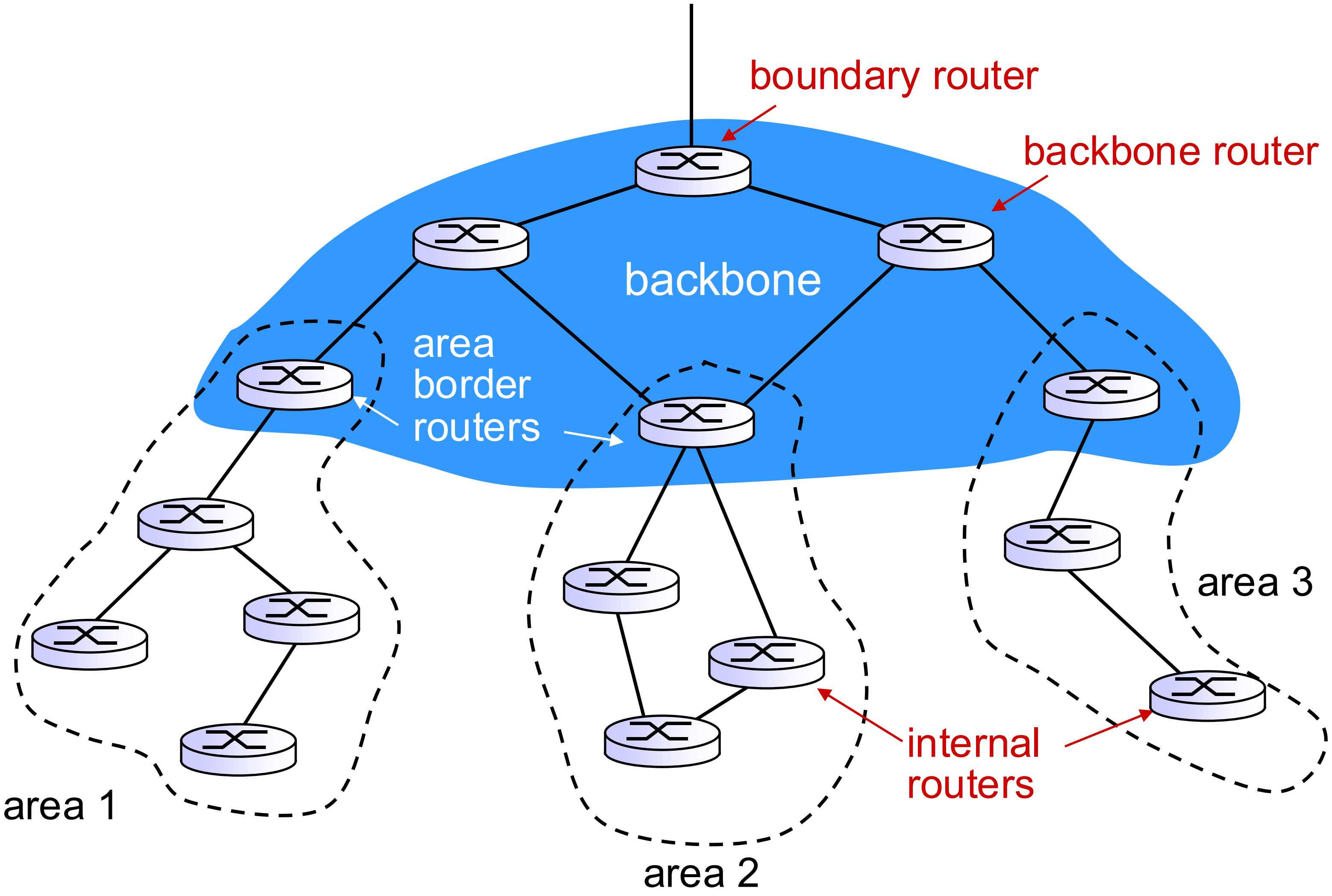
Goal:
Get traffic that is already in an AS to a destination inside that same AS.

OSPF and IS-IS are deployed most commonly today

OSPF (Open Shortest Path First)

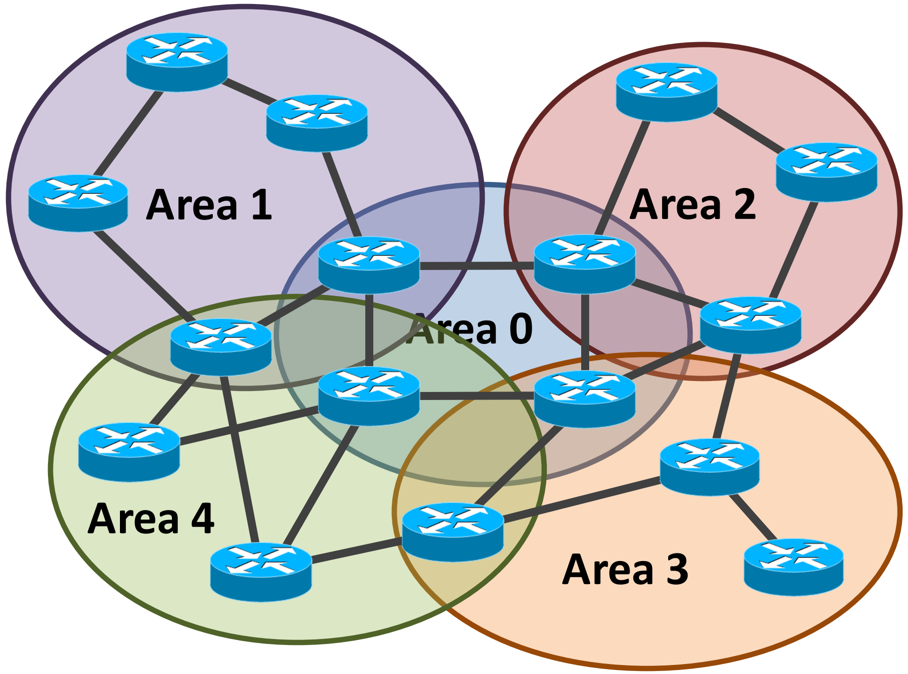
- Link state protocol (reliable flooding of LSAs)
- “Open”: standardized, publicly available implementations
- Multiple equal-cost paths allowed (load balancing)
- Additional features:
  - OSPF messages authenticated (to prevent malicious intrusion)
  - Hierarchical OSPF for large autonomous systems.

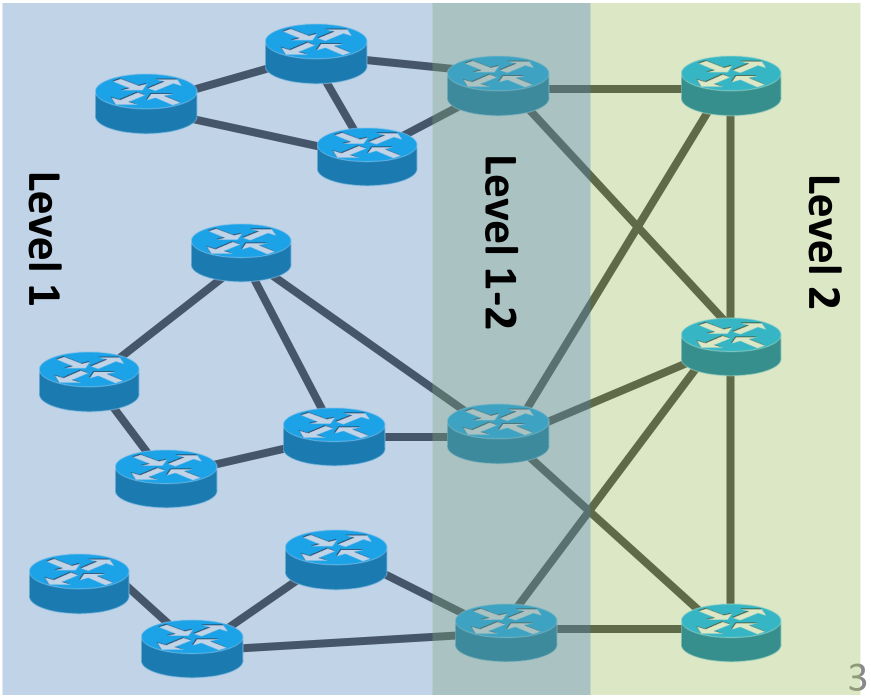
Hierarchical OSPF



Different Organizational Structure

OSPF	IS-IS
• Organized around overlapping areas • Area 0 is the core network	• Organized as a 2-level hierarchy • Level 2 is the backbone





Real Protocols: OSPF vs. IS-IS

- Two different implementations of link-state routing

• OSPF	• IS-IS
• Favored by companies, datacenters • More optional features • Built on top of IPv4 – LSAs are sent via IPv4 – OSPFv3 needed for IPv6	• Favored by ISPs • Less "chatty" – Less network overhead – Supports more devices • Not tied to IP – Works with IPv4 or IPv6

Internet/inter-AS Routing

Goal:

Get traffic from one AS to another.

Why do we need different Intra and Interdomain AS routing ?

A. Scalability
B. Performance
C. A and B
D. More than just A and B

Why do we need different Intra and Interdomain AS routing ?

Policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its net.
- intra-AS: single admin, so no policy decisions needed

Scale:

- hierarchical routing saves table size, reduced update traffic

Performance:

- intra-AS: can focus on performance
- inter-AS: policy may dominate over performance

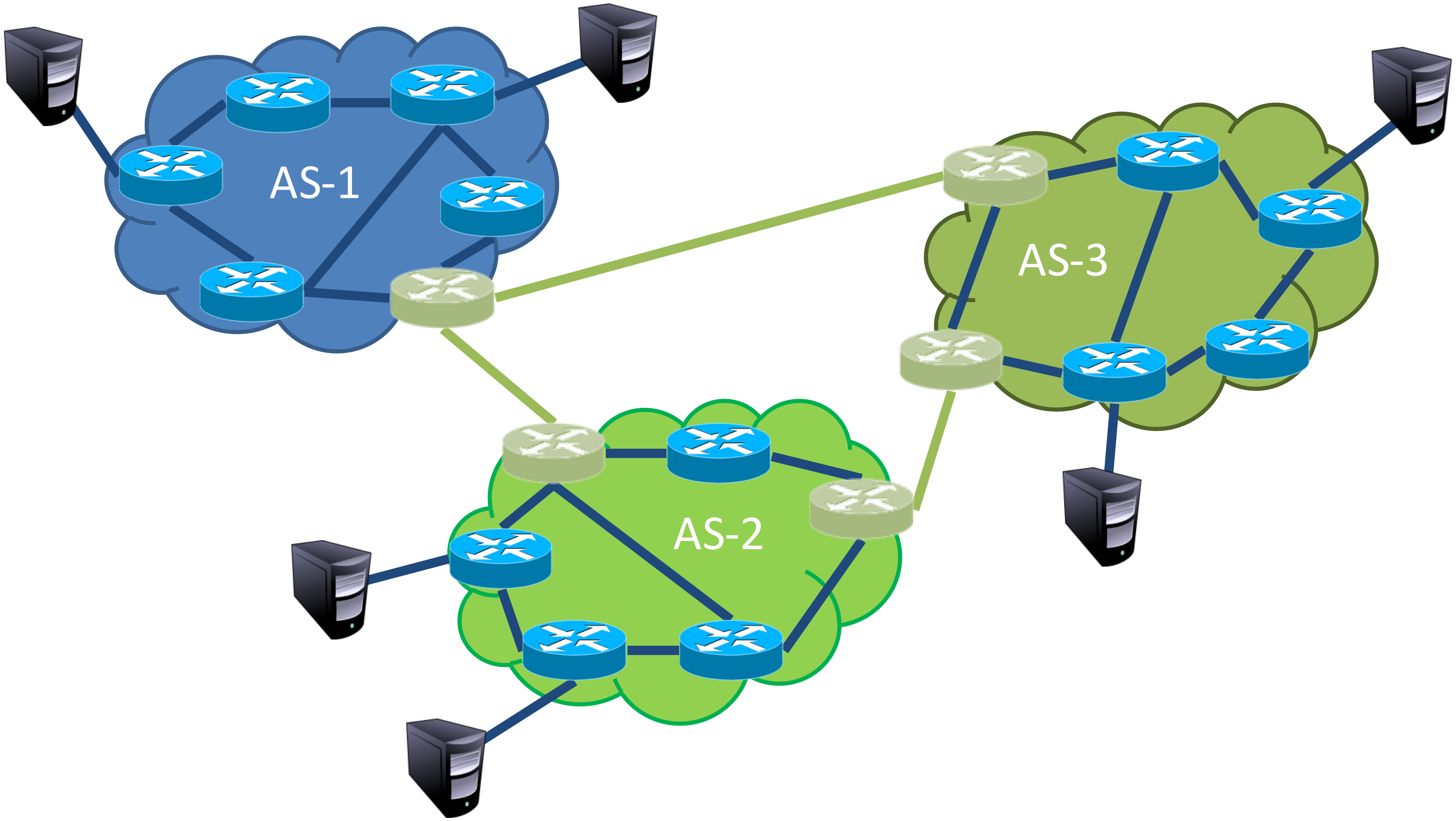
The Inter-domain routing protocol, needs to be an agreed upon protocol across all Autonomous Systems

A. Yes, for interoperability
B. Not necessarily, but reduces overhead
C. No, each AS can have its own inter-domain routing protocol of choice.

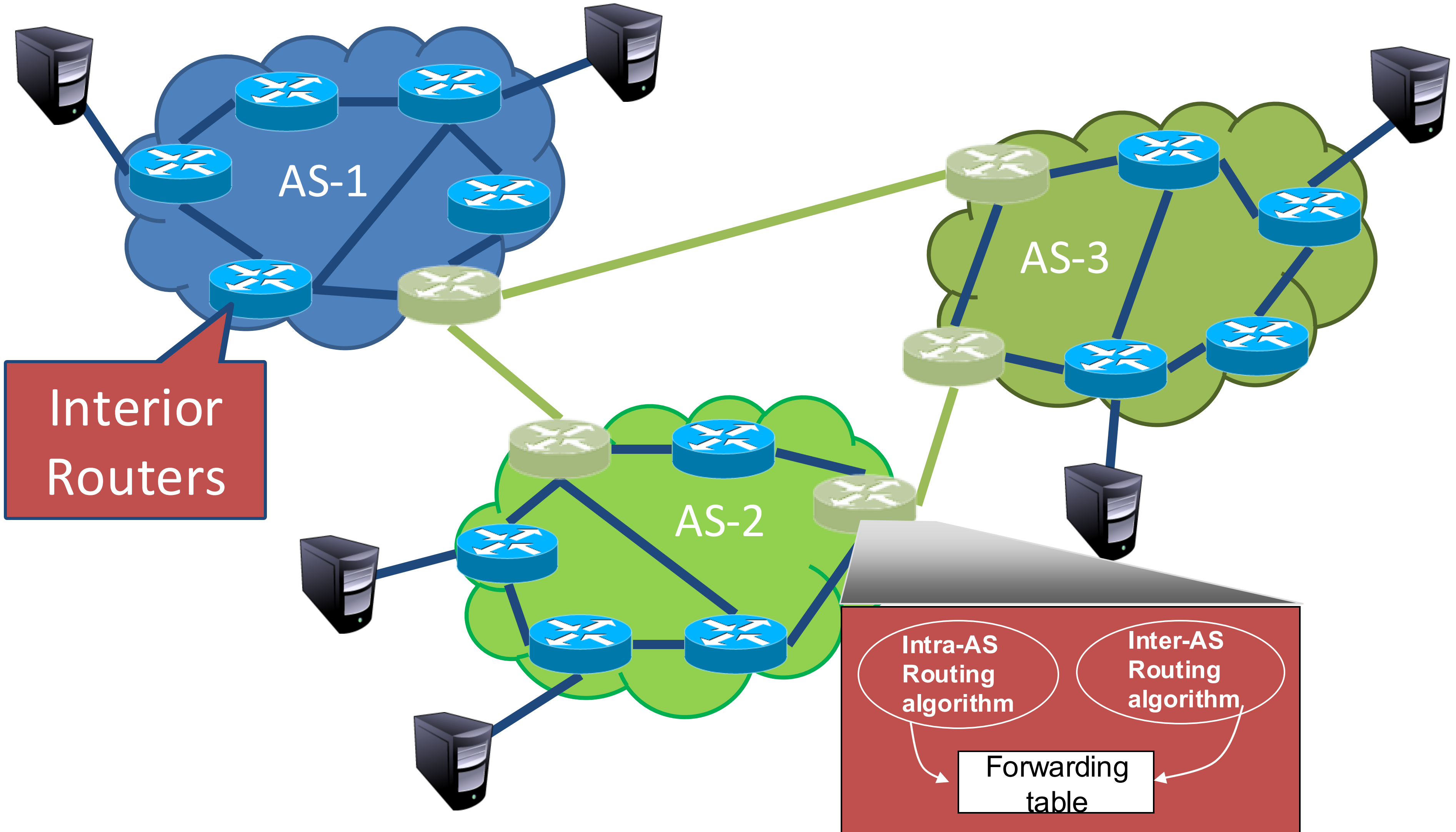
The Inter-domain routing protocol, needs to be an agreed upon protocol across all Autonomous Systems

- Global connectivity is at stake!
  - Thus, all ASs must use the same protocol
  - Contrast with intra-domain routing
- What are the requirements?
  - Scalability
  - Flexibility in choosing routes
- Question: link state or distance vector?
  - Trick question: BGP is a path vector protocol

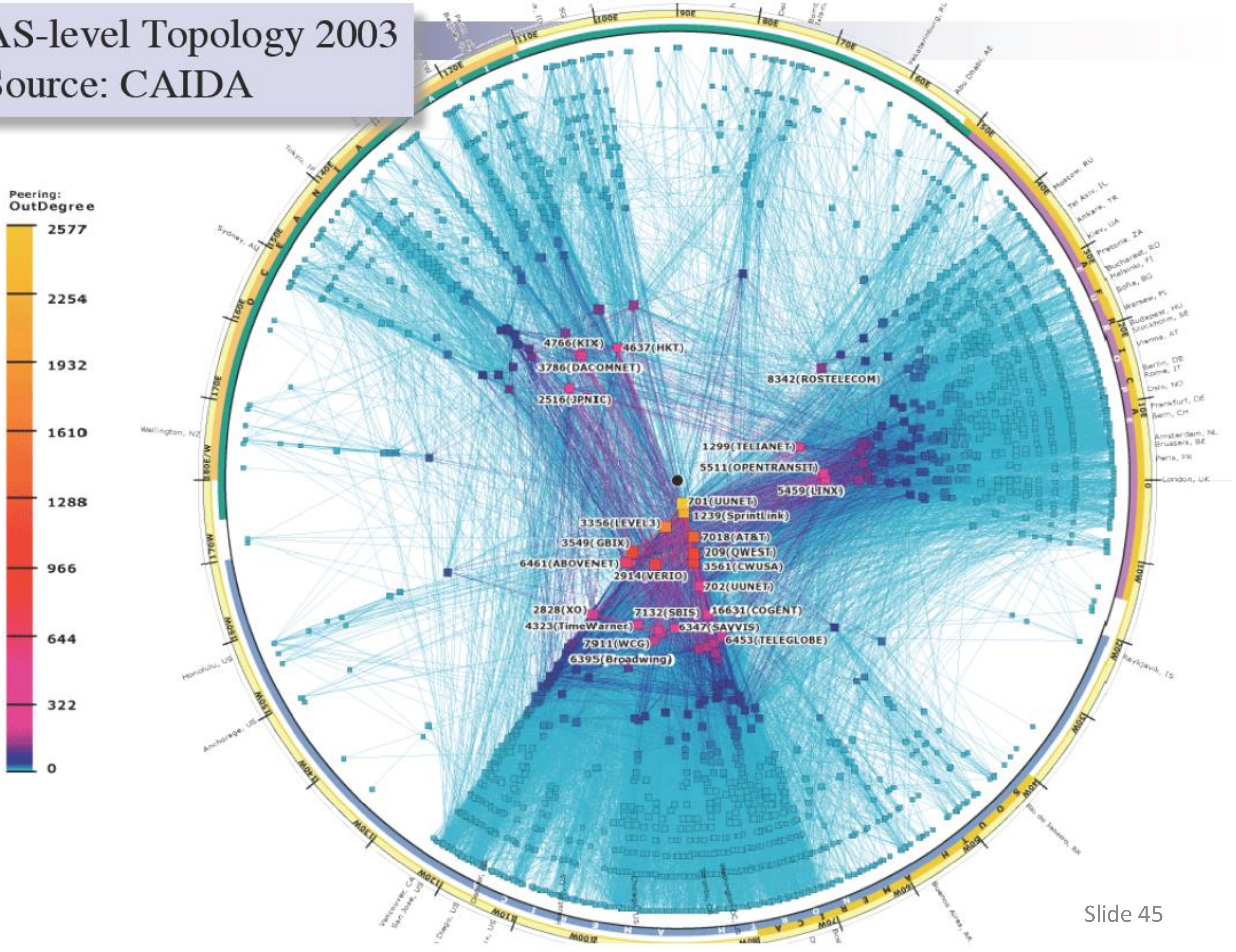
Hierarchical routing: Autonomous Systems



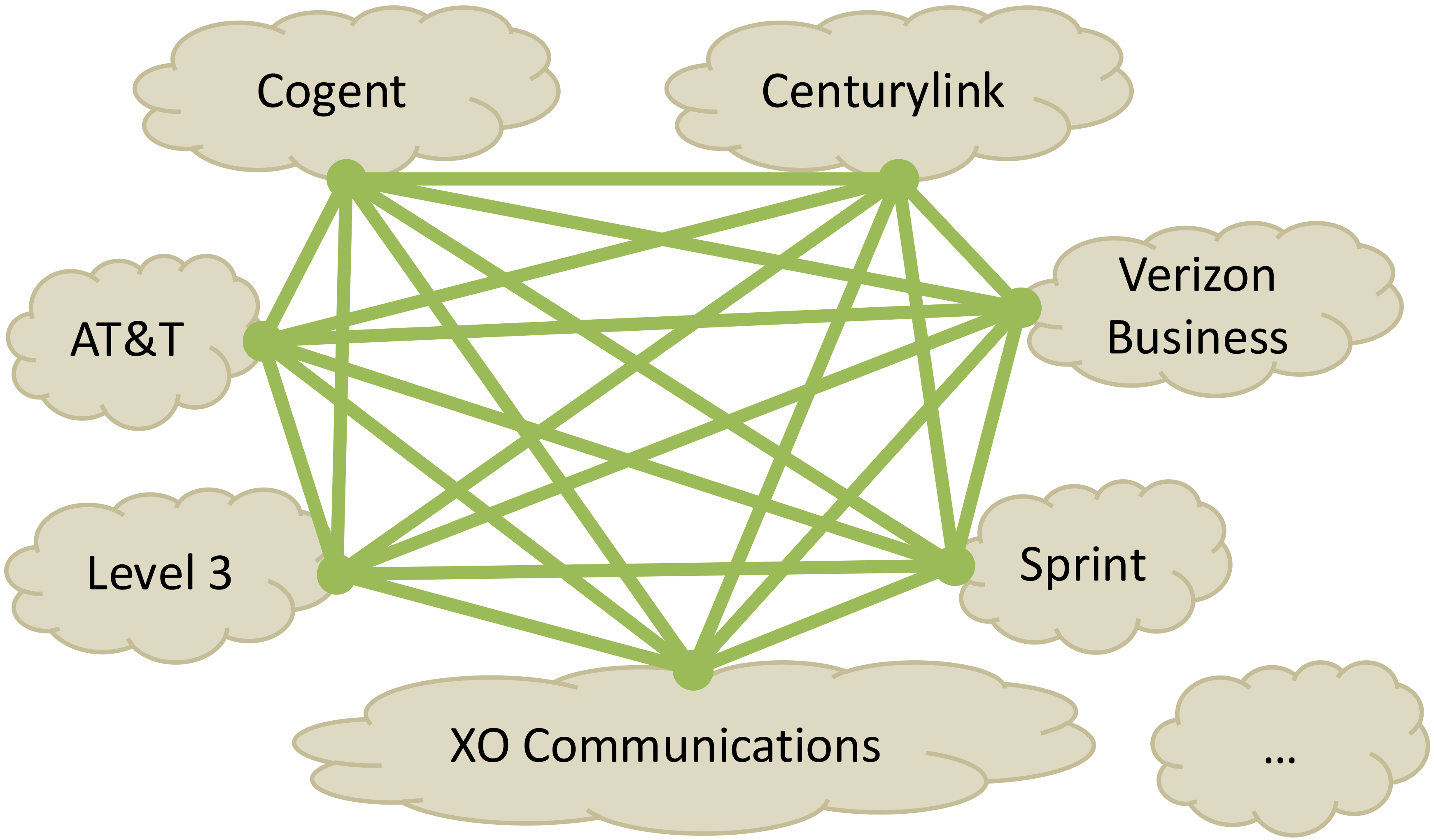
Hierarchical routing: Interconnected ASes



AS-level Topology 2003
Source: CAIDA

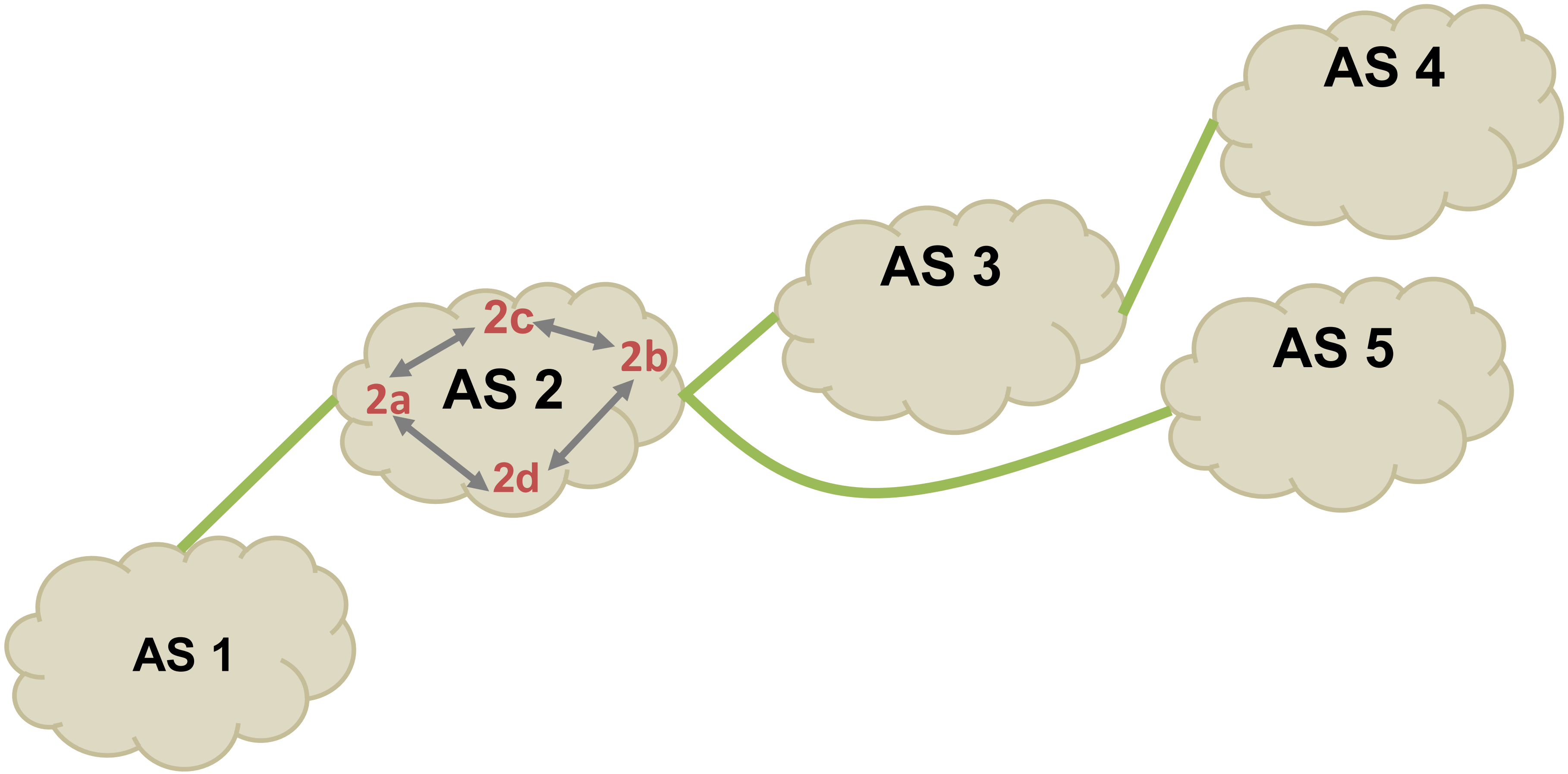


Tier-1 ISP Peering



Path Vector Protocol

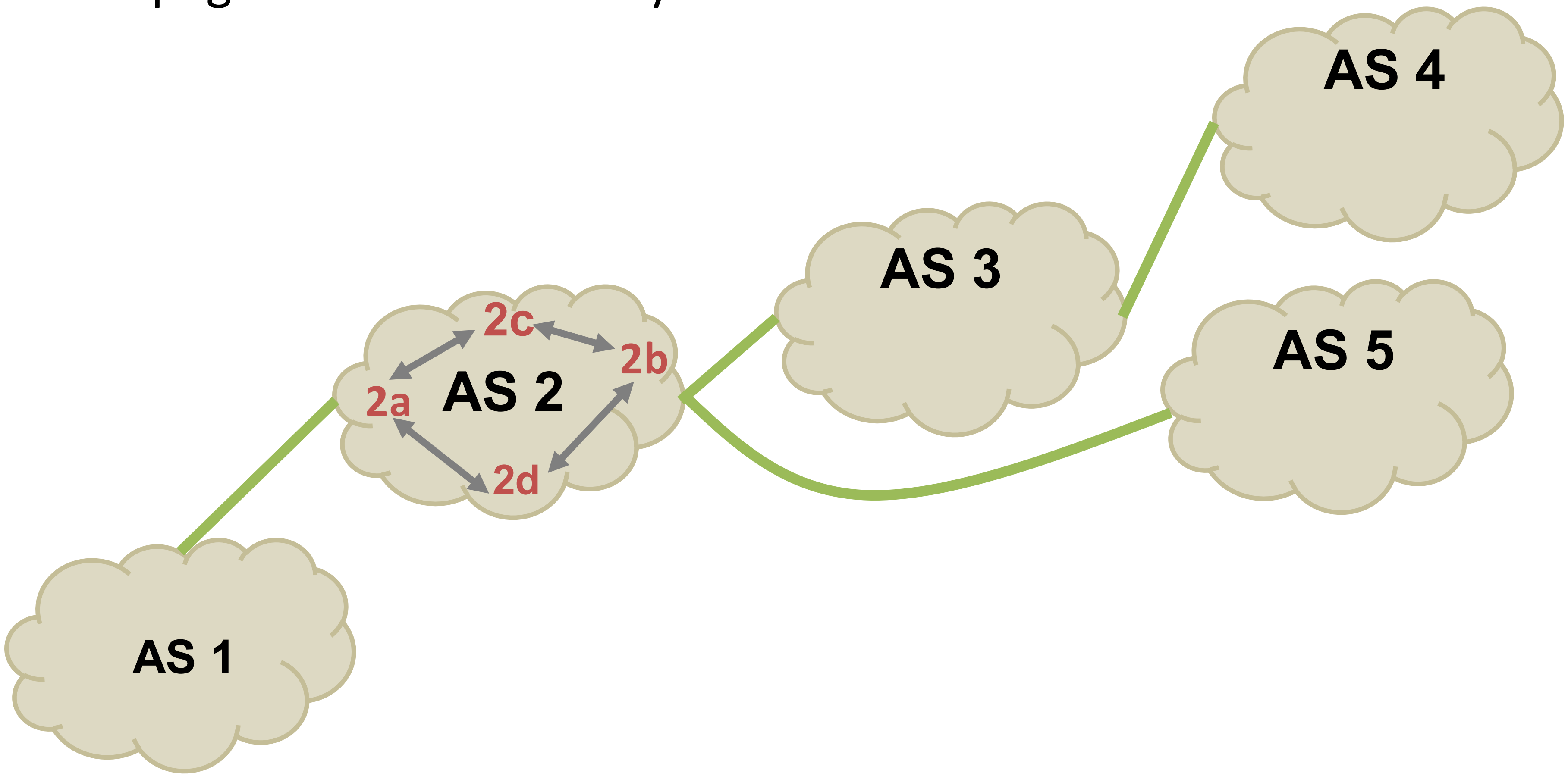
- Key idea: advertise the entire path
  - Distance vector: send *distance metric* per dest d
  - Path vector: send the *entire path* for each dest d



Inter-domain (Inter-ISP) Routing

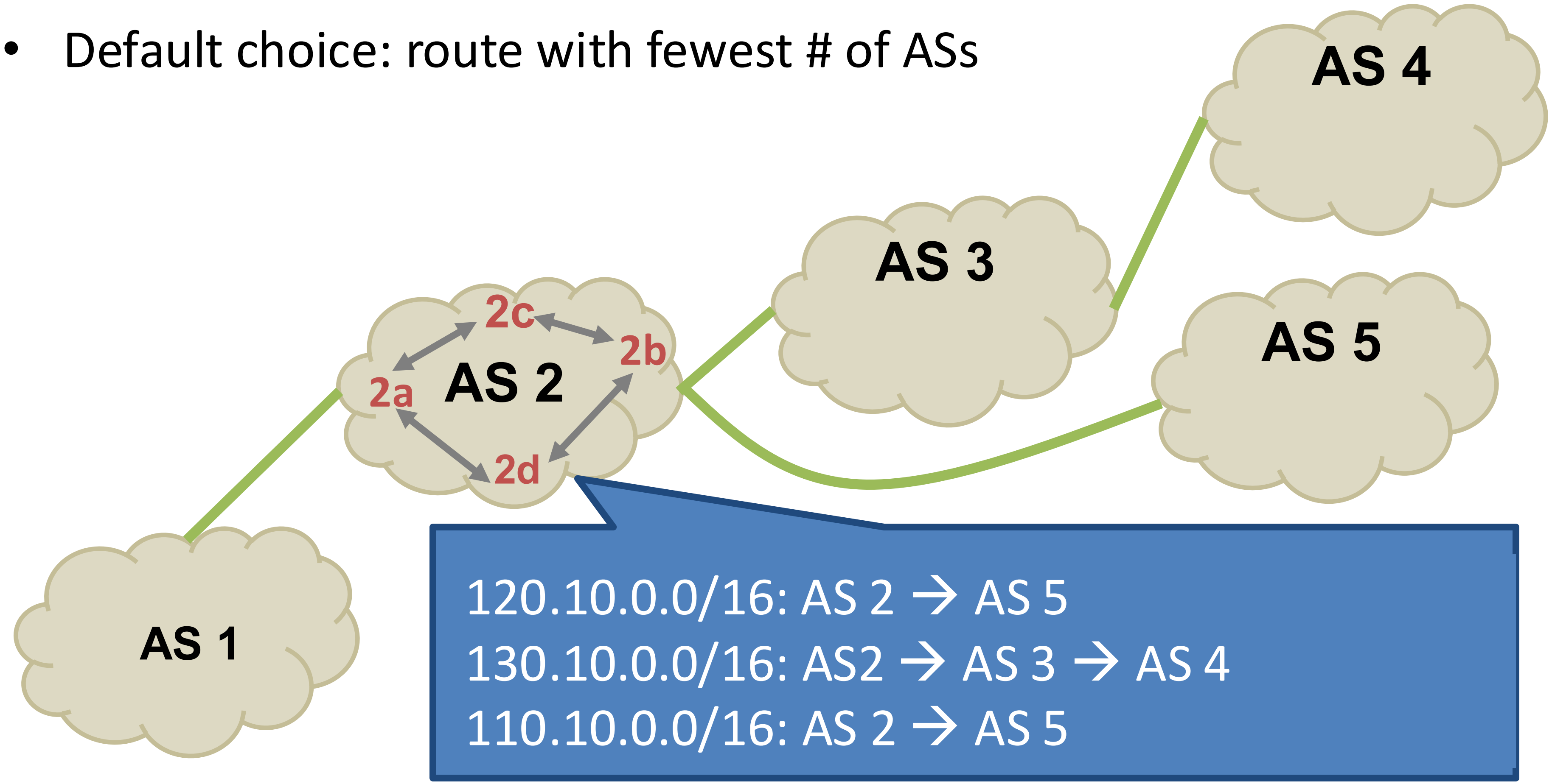
AS2 must:

1. Learn destinations reachable through AS2
2. Propagate this reachability info to all routers in AS2



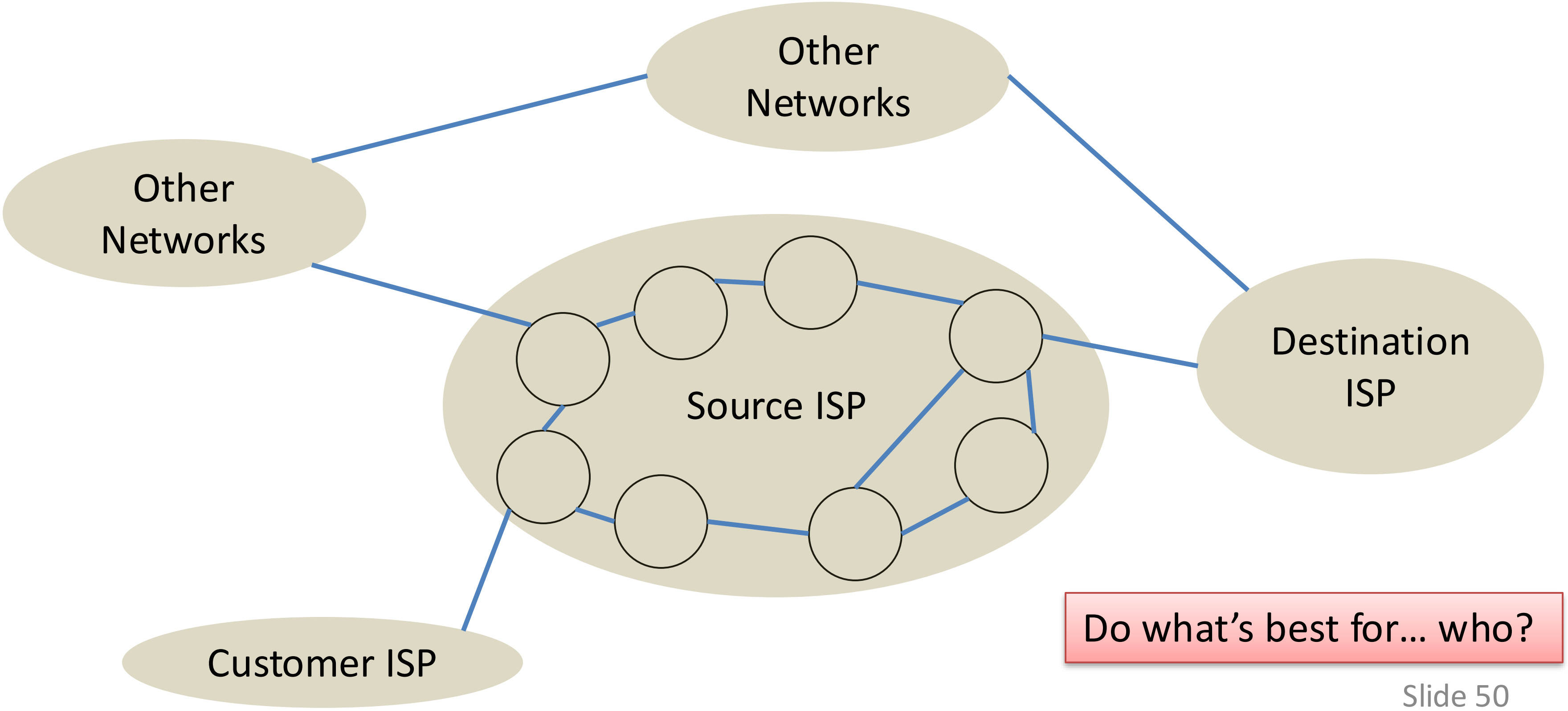
Path Vector Protocol

- AS-path: sequence of ASs a route traverses
  - Like distance vector, plus additional information
- Used for loop detection and to apply policy
- Default choice: route with fewest # of ASs



Routing Policy

- How should the ISP route the customer's traffic to the destination?



Which routes a BGP router advertises will depend on…

A. which ISPs have contractual agreements.

B. the shortest path to a subnet/prefix.

C. which subnets are customers of an ISP.

D. More than one of the above. (which?)

Which routes a BGP router advertises will depend on…

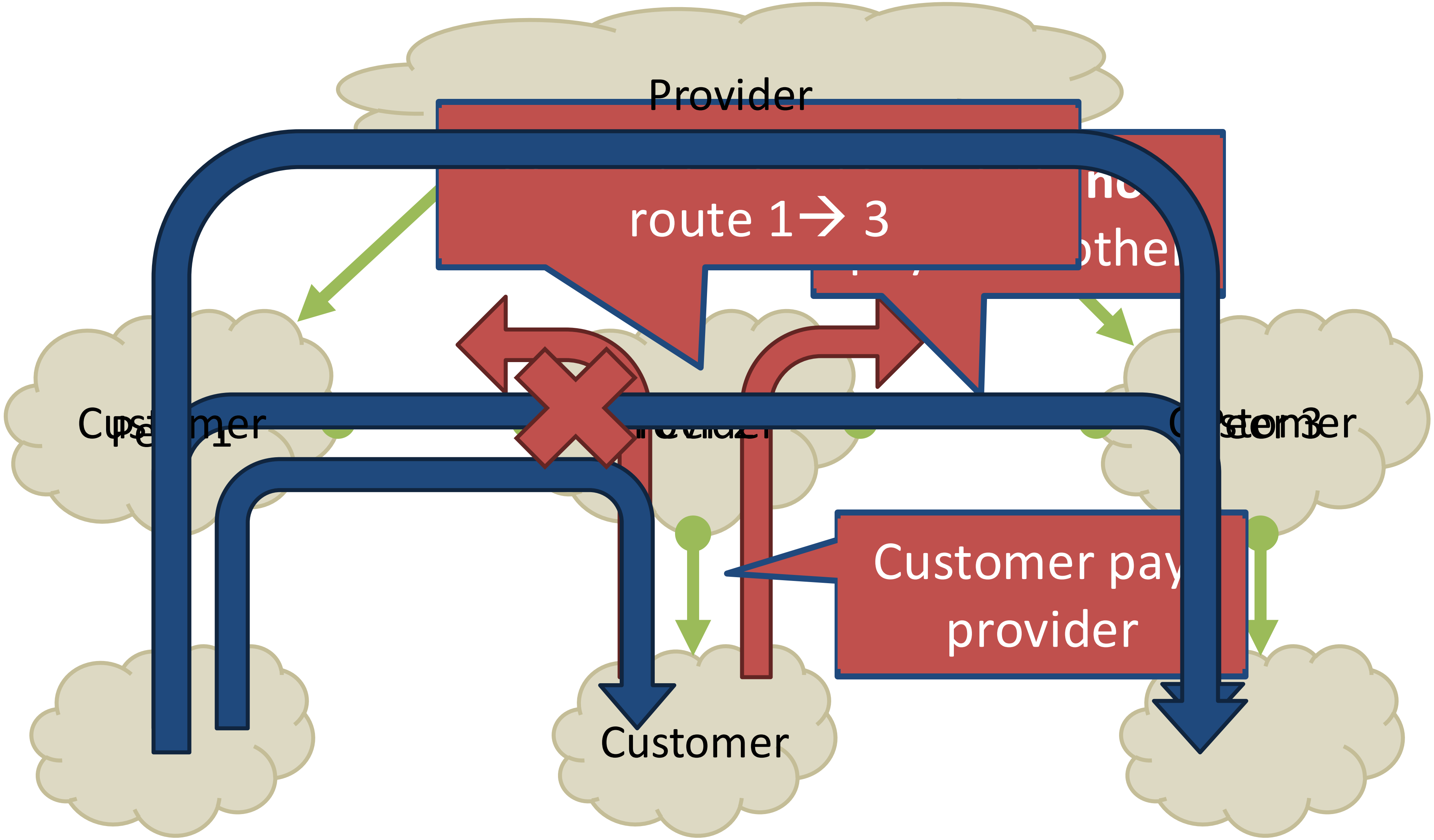
A. which ISPs have contractual agreements.

B. the shortest path to a subnet/prefix.

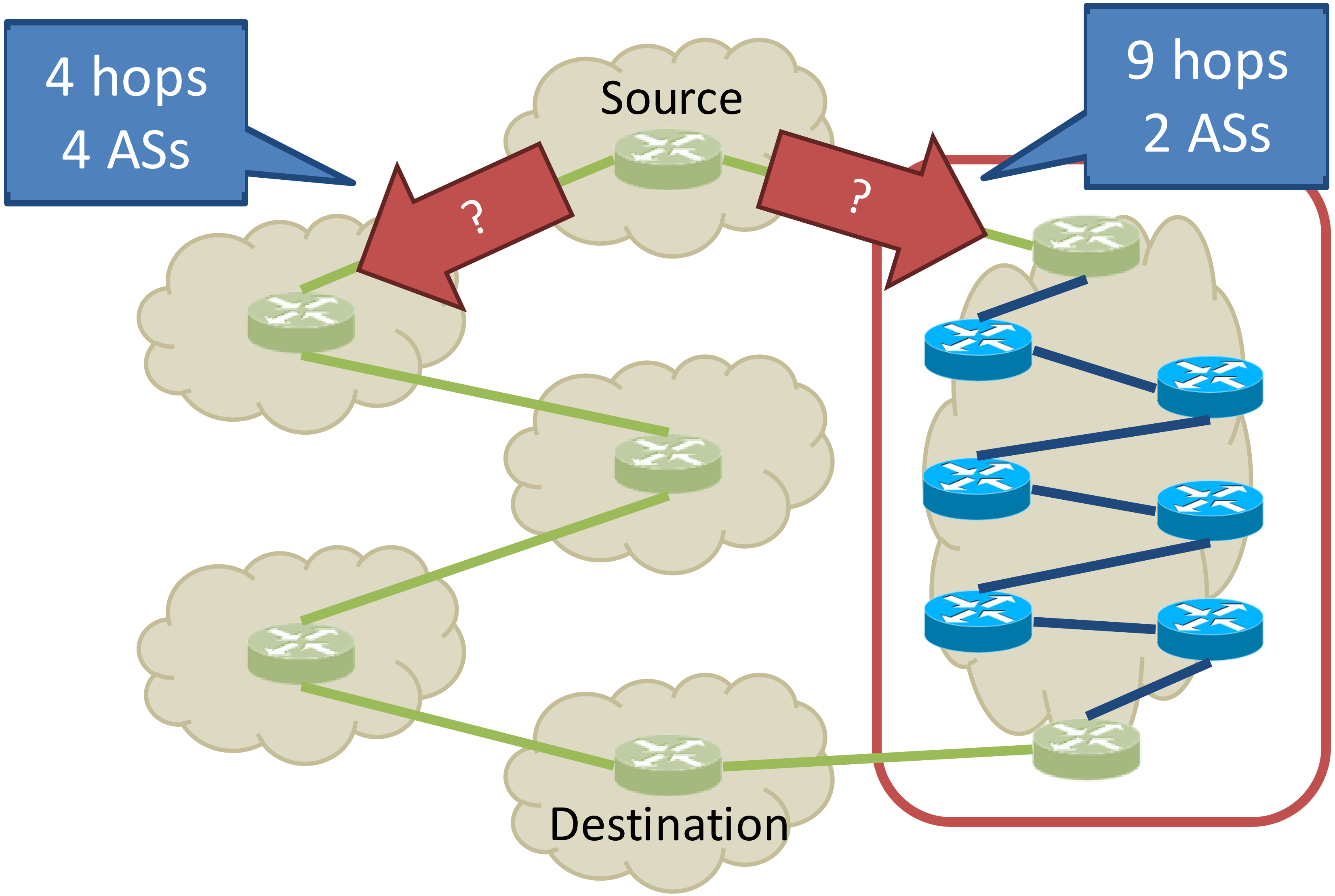
C. which subnets are customers of an ISP.

D. More than one of the above. (which?)

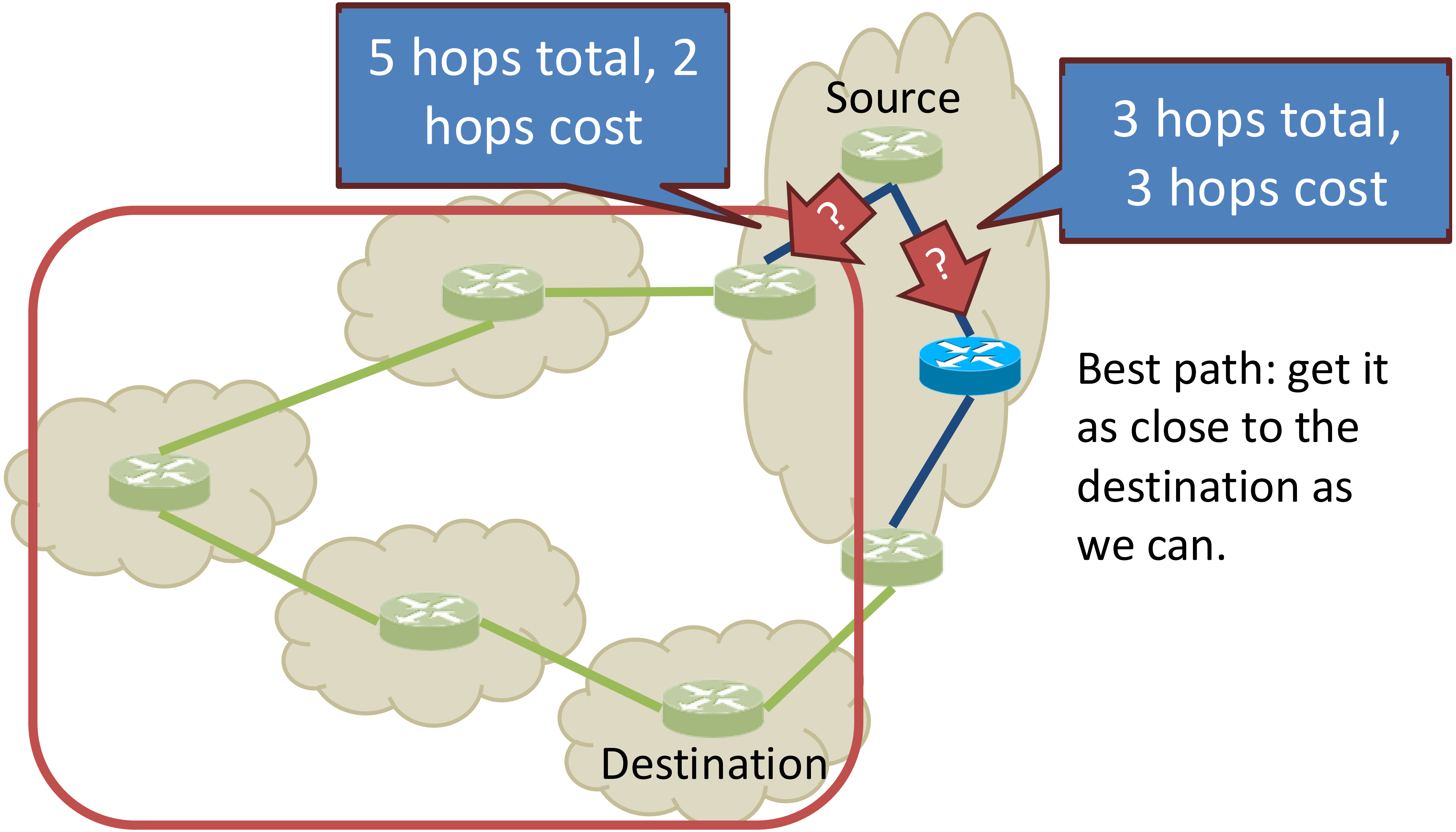
BGP Relationships



Shortest AS Path != Shortest Path



Hot Potato Routing: get rid of packets ASAP!



Best path: get it as close to the destination as we can.

Route Selection Summary

Highest Local Preference	Enforce relationships
Shortest AS Path Lowest Multi-Exit Discriminator Lowest IGP Cost to BGP Egress	Traffic engineering
Lowest Router ID	When all else fails, break ties

Peering/Interconnection Wars

- Peer

- Reduce upstream costs
- Improve end-to-end performance
- May be the only way to connect to parts of the Internet

- Don't Peer

- You would rather have customers
- Peers are often competitors
- Peering agreements require periodic renegotiation

Peering struggles in the ISP world are extremely contentious, agreements are usually confidential

Hierarchical routing: Interconnected ASes