Distributed Routing

CSE 561, Winter 2021

Ratul Mahajan

What we read

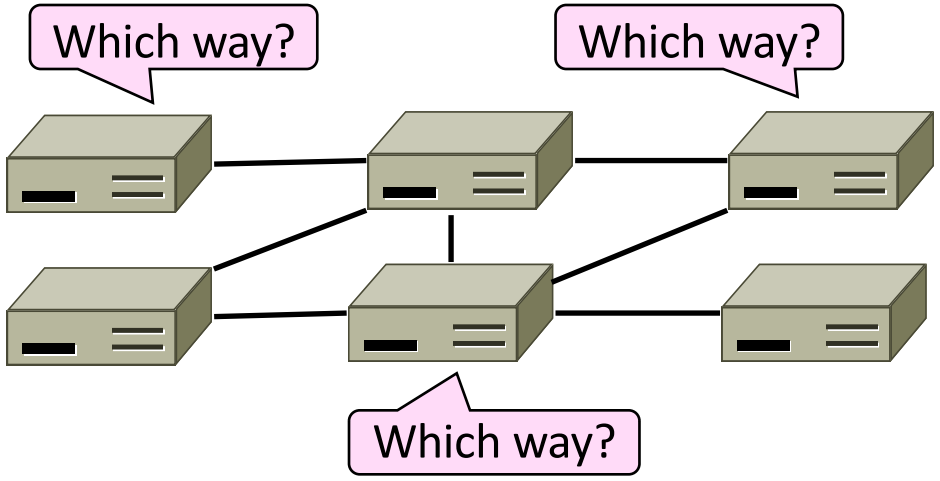
Three ways to achieve distributed routing

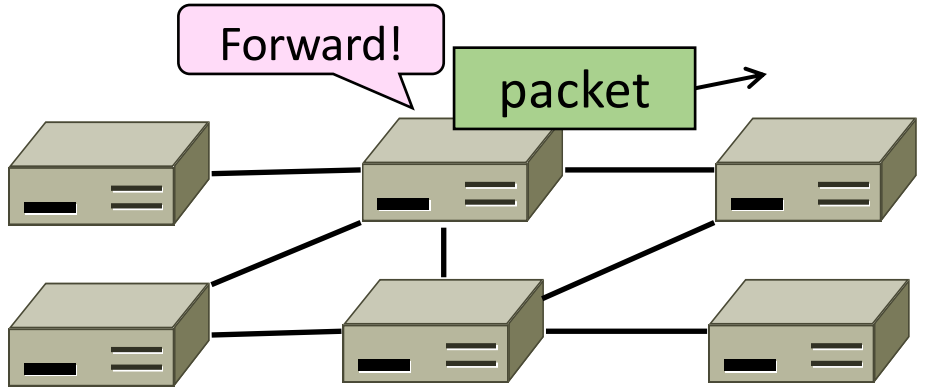
- Distance vector
- Link state
- Path vector, policy-based (BGP)

Routing versus forwarding

Routing: deciding in which direction to send traffic

Forwarding: sending a packet on its way





Centralized versus distributed routing

Centralized

- Collect all information in one place
- Compute good paths
- Tell routers about those paths

Distributed

- Routers exchange information
- Compute good paths

More flexibility in types of paths

- Can handle dynamics better because of global view

More fault tolerant

- Remember nuclear attacks?

Rules of fully distributed routing

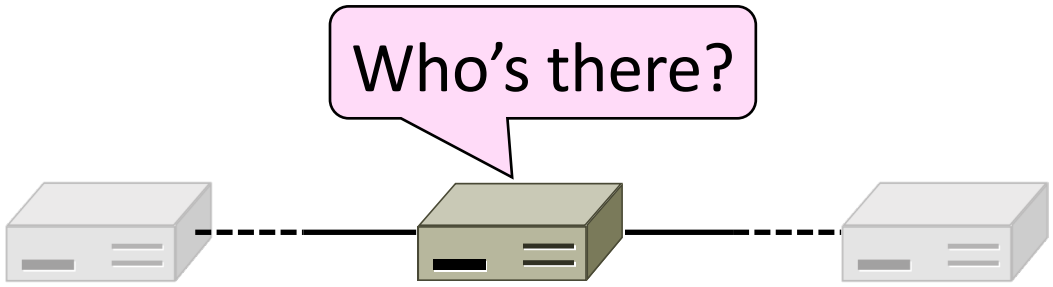
All nodes are alike; no controller

Nodes learn by exchanging messages with neighbors

Nodes operate concurrently

There may be node/link/message failures

Different routing protocols differ in what information is exchanged



Paths computed by different protocols

”Best” or “shortest” paths

- Global notion of goodness
- Distance vector and link state

Policy-based paths

- Nodes have personal preferences
- BGP

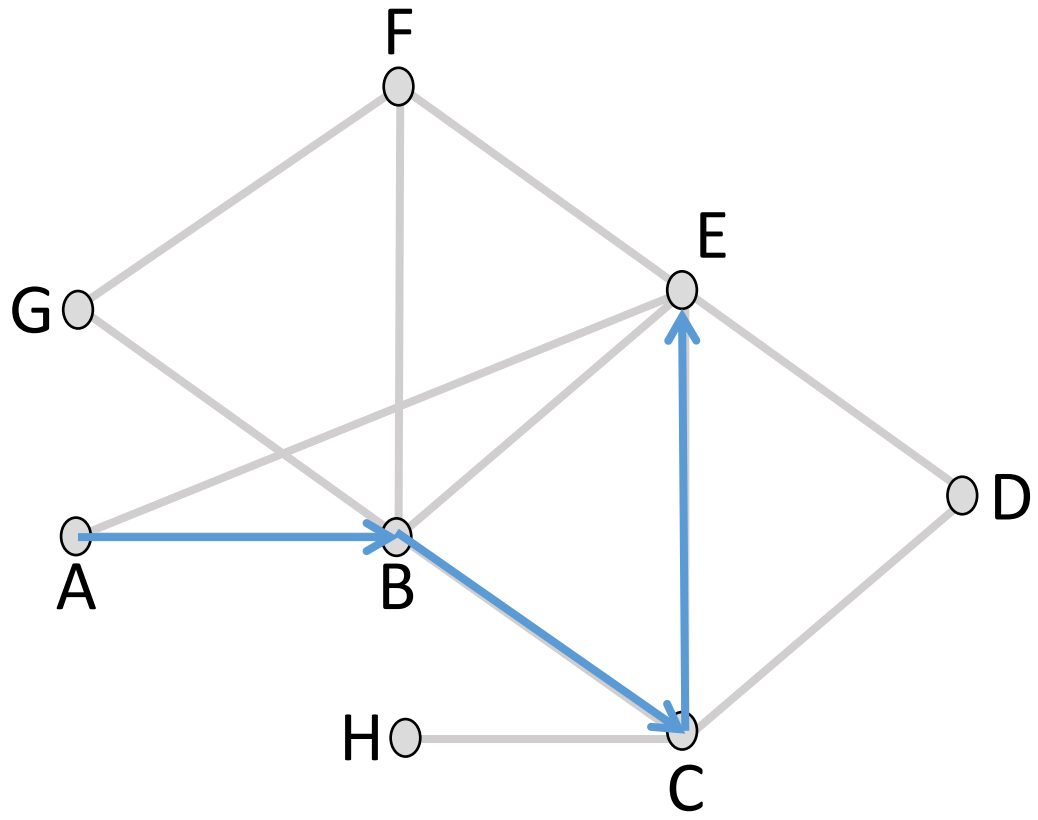
What are “Best” paths anyhow?

Many possibilities:

- Latency: avoid circuitous paths
- Bandwidth: avoid slow links
- Money: avoid expensive links
- Hops: reduce switching

But only consider topology

- Ignore workload, e.g., hotspots



Least cost or shortest Paths

1. Assign each link a *cost* that captures the factors
2. Best path between a pair of nodes is the path with the the least total cost

There may be multiple best paths

Distance Vector Routing

Distance Vector Algorithm

Each node maintains a vector of (distance, next hop) to all destinations

1. Initialize vector with 0 (zero) to self, ∞ (infinity) to others
2. Periodically send vector to neighbors
3. Update vector for each destination by selecting the shortest distance heard, after adding cost of neighbor link
4. Use the best neighbor for forwarding

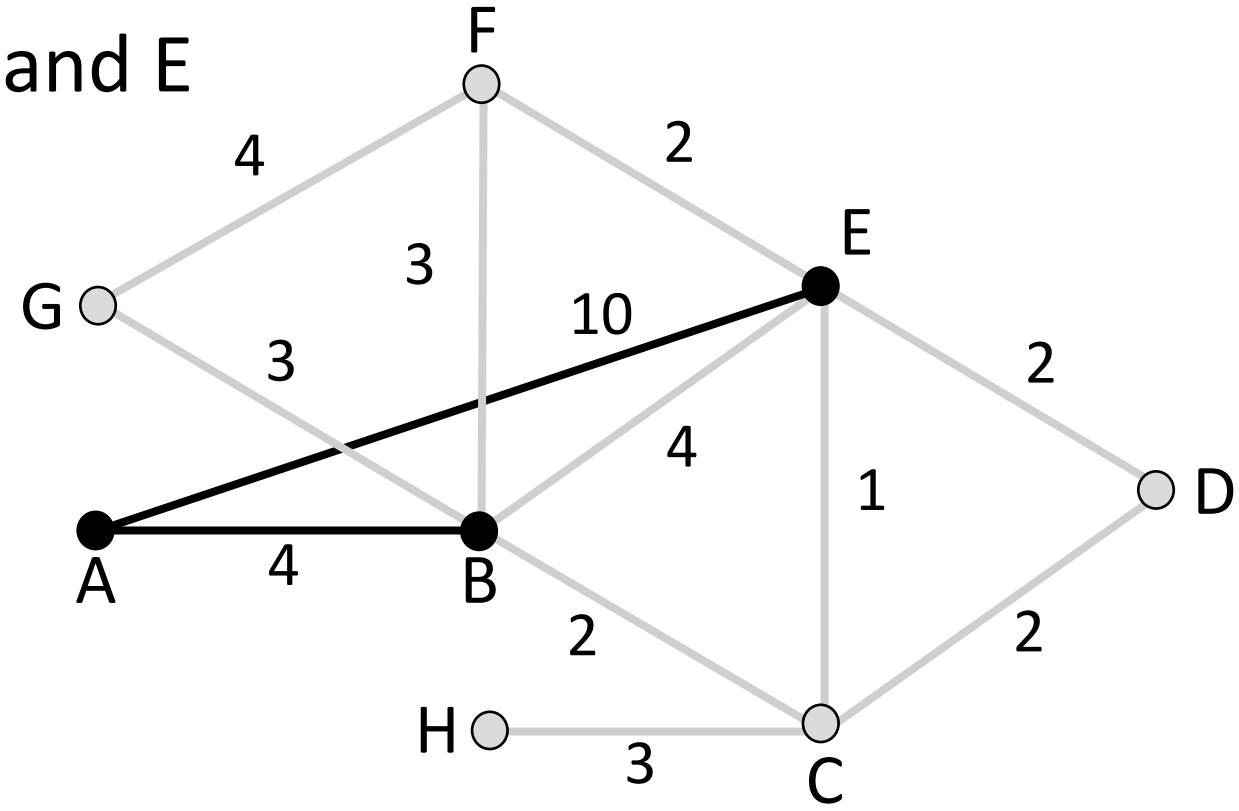
Distance Vector (2)

Consider from the point of view of node A

- Can only talk to nodes B and E

Initial vector →

To	Cost
A	0
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞



Distance Vector (3)

First exchange with B, E; learn best 1-hop routes

To	B says	E says
A	∞	∞
B	0	∞
C	∞	∞
D	∞	∞
E	∞	0
F	∞	∞
G	∞	∞
H	∞	∞

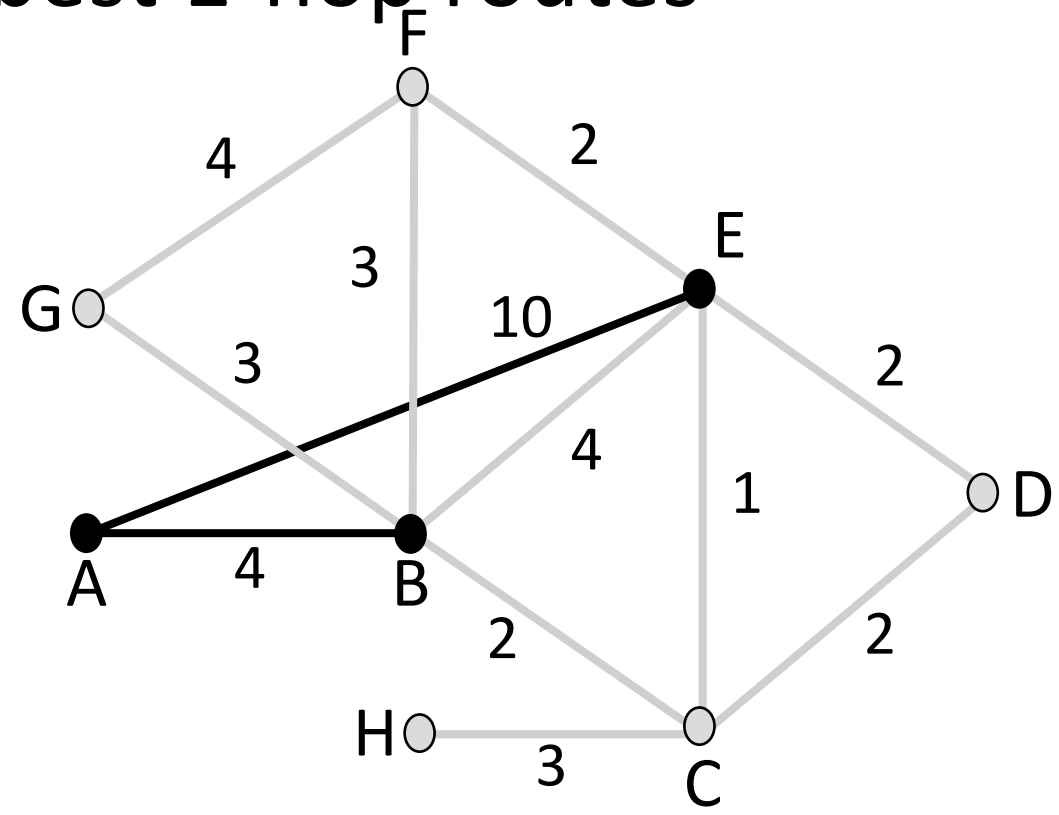
→

B +4	E +10
∞	∞
4	∞
∞	∞
∞	∞
∞	10
∞	∞
∞	∞
∞	∞

→

A's Cost	A's Next
0	--
4	B
∞	--
∞	--
10	E
∞	--
∞	--
∞	--

Learned better route



Distance Vector (4)

Second exchange; learn best 2-hop routes

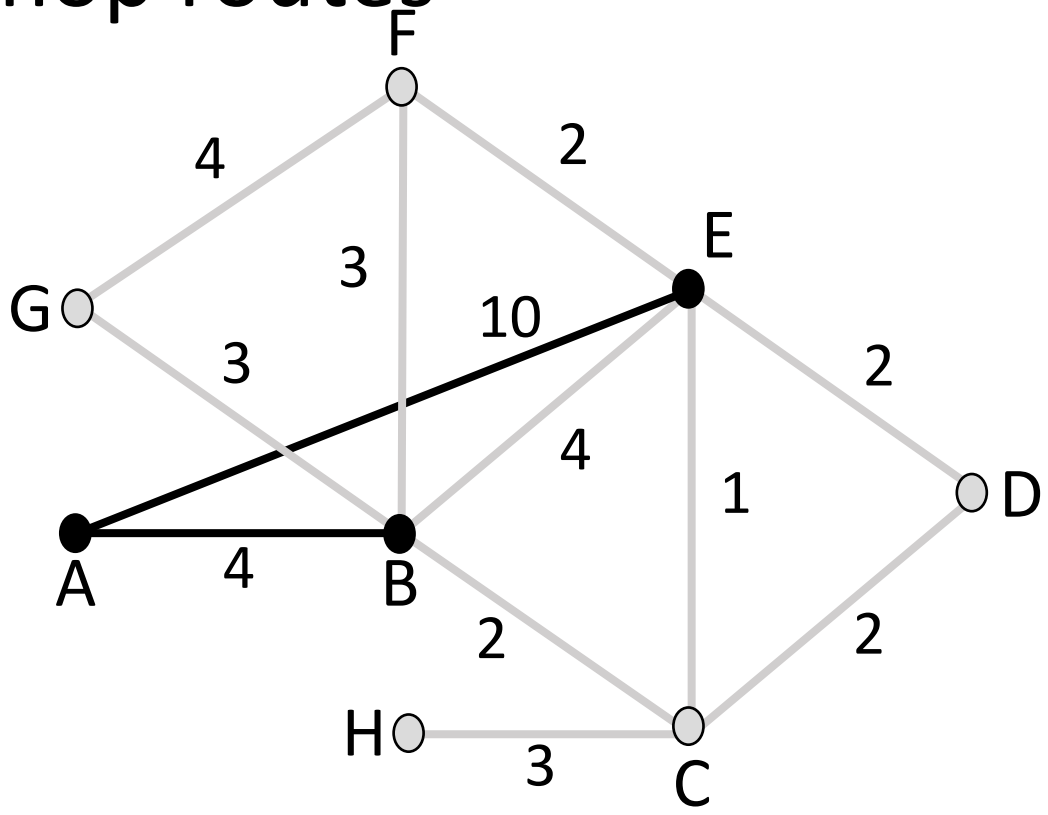
To	B says	E says
A	4	10
B	0	4
C	2	1
D	∞	2
E	4	0
F	3	2
G	3	∞
H	∞	∞

→

B +4	E +10
8	20
4	14
6	11
∞	12
8	10
7	12
7	∞
∞	∞

→

A's Cost	A's Next
0	--
4	B
6	B
12	E
8	B
7	B
7	B
∞	--



Distance Vector (4)

Third exchange; learn best 3-hop routes

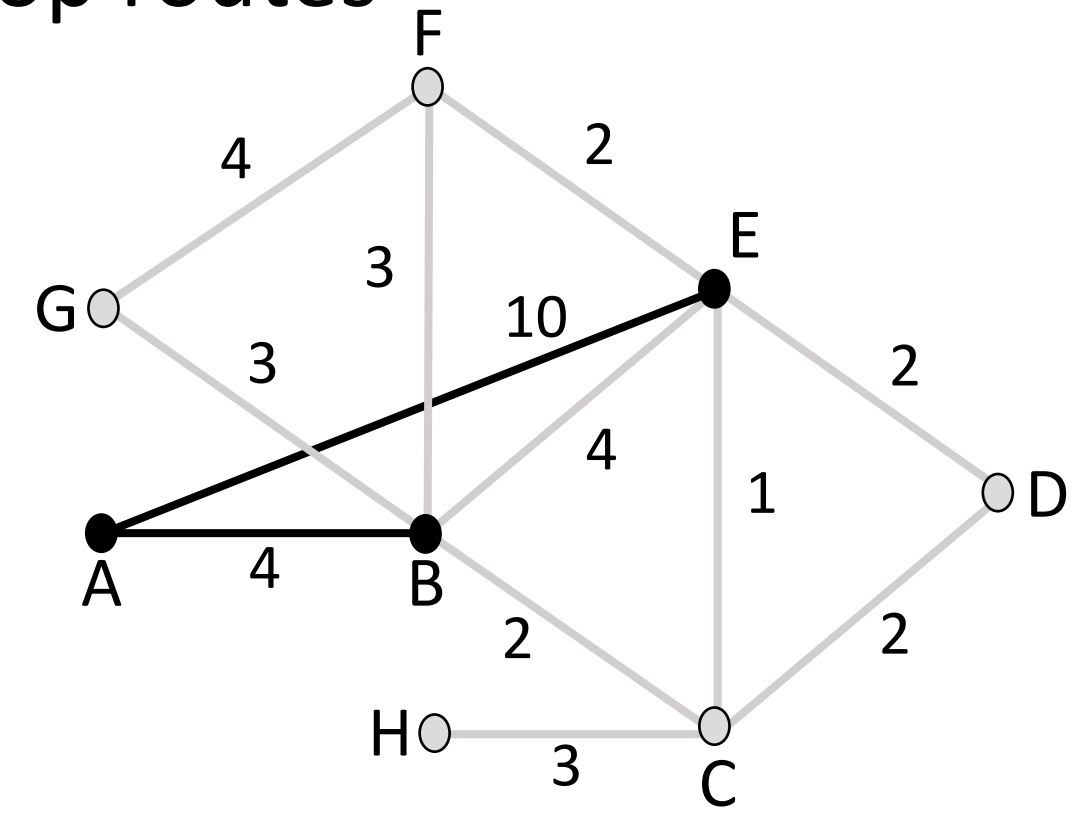
To	B says	E says
A	4	8
B	0	3
C	2	1
D	4	2
E	3	0
F	3	2
G	3	6
H	5	4

→

B +4	E +10
8	18
4	13
6	11
8	12
7	10
7	12
7	16
9	14

→

A's Cost	A's Next
0	--
4	B
6	B
8	B
7	B
7	B
7	B
9	B



Distance Vector (5)

Subsequent exchanges; converged

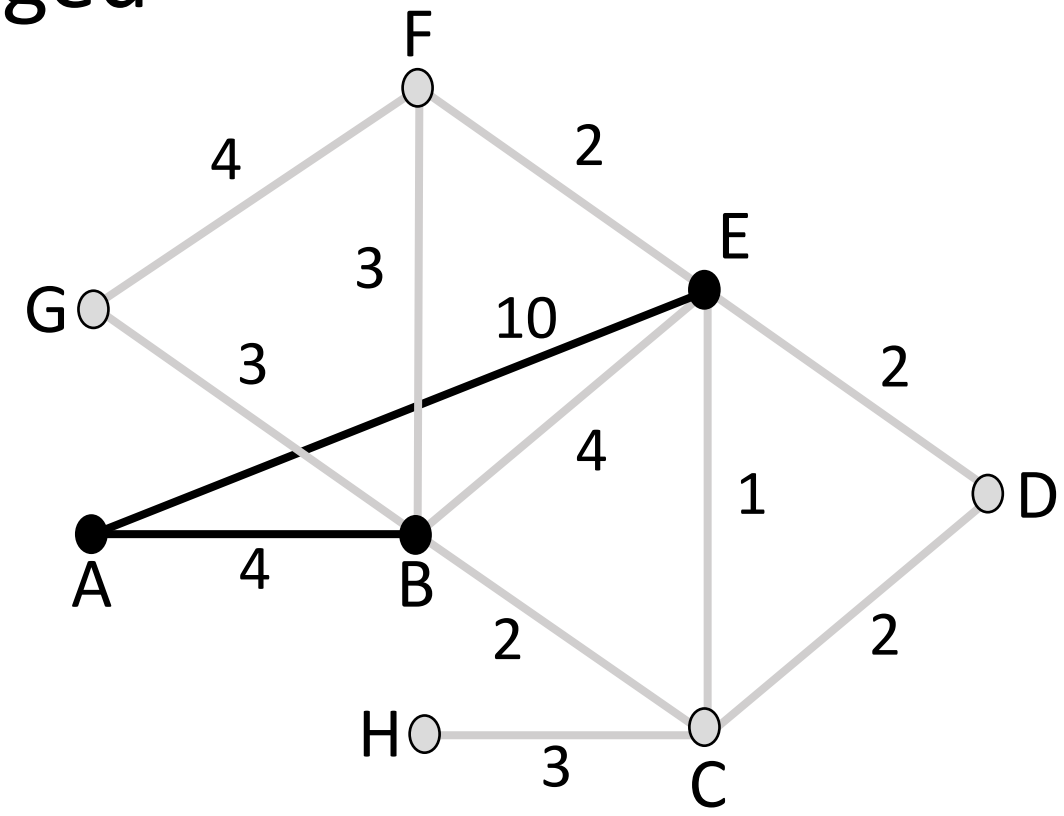
To	B says	E says
A	4	7
B	0	3
C	2	1
D	4	2
E	3	0
F	3	2
G	3	6
H	5	4

→

B +4	E +10
8	17
4	13
6	11
8	12
7	10
7	12
7	16
9	14

→

A's Cost	A's Next
0	--
4	B
6	B
8	B
8	B
7	B
7	B
9	B



Distance Vector Dynamics

Adding routes:

- News travels one hop per exchange

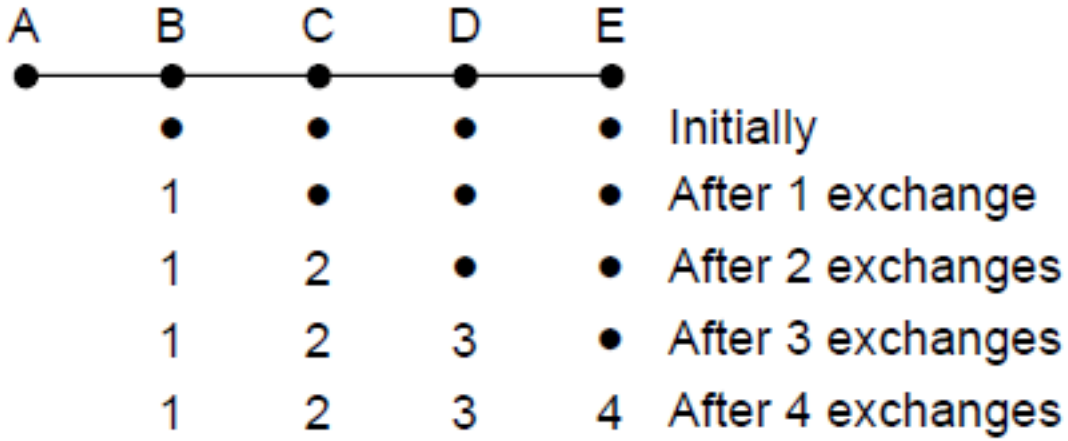
Removing routes:

- When a node fails, no more exchanges, other nodes forget

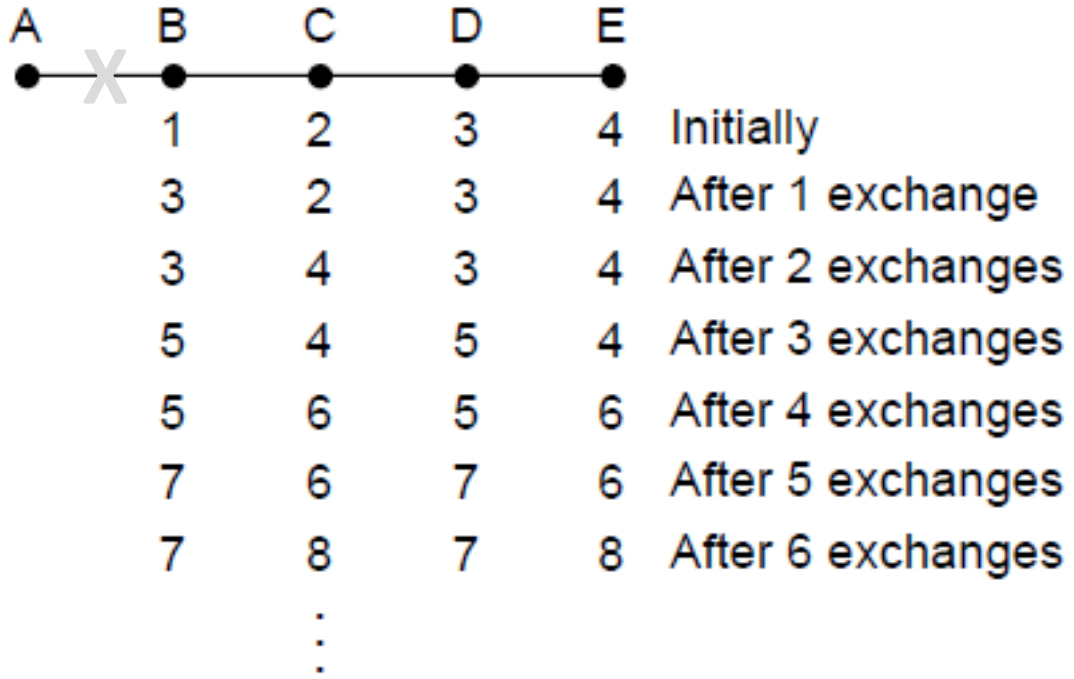
Problem?

Count to Infinity: Problem

- Good news travels quickly, bad news slowly



Desired convergence



"Count to infinity" scenario

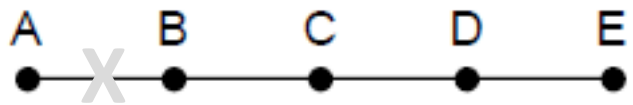
Count to Infinity: Heuristics

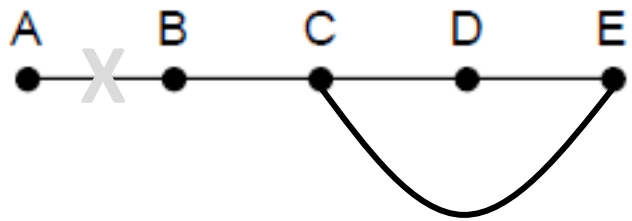
Split horizon

- Don’t send route back to where you learned it from.

Poison reverse

- Send “infinity” when you notice a disconnect





Neither is very effective in practice

Link-State Routing

Link-State Algorithm

1. Nodes flood topology with link state packets
   - Each node learns full topology
2. Each node computes its own forwarding table
   - By running Dijkstra (or equivalent)

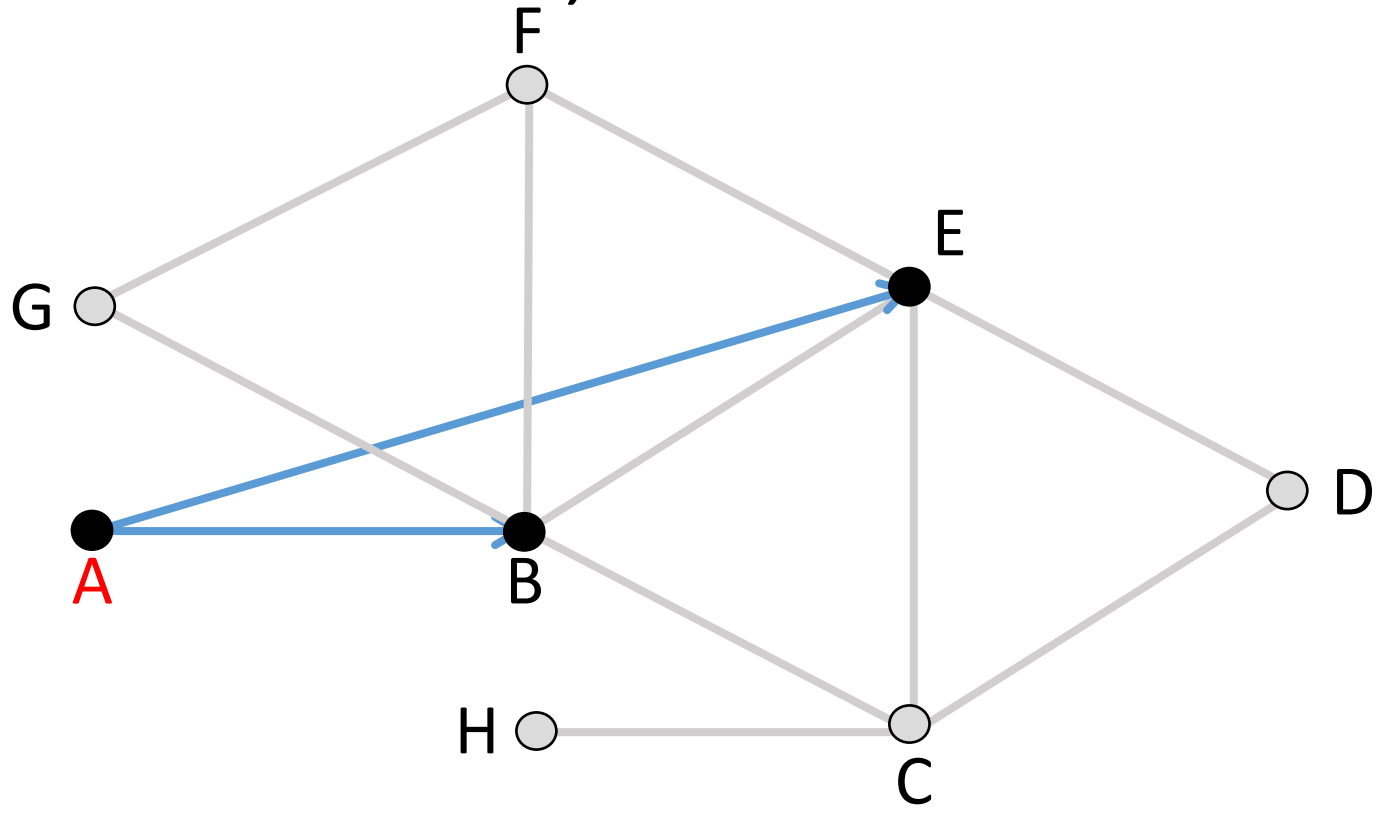
Flooding

Rule used at each node:

- Sends an incoming message on to all other neighbors
- Remember the message so that it is only flood once

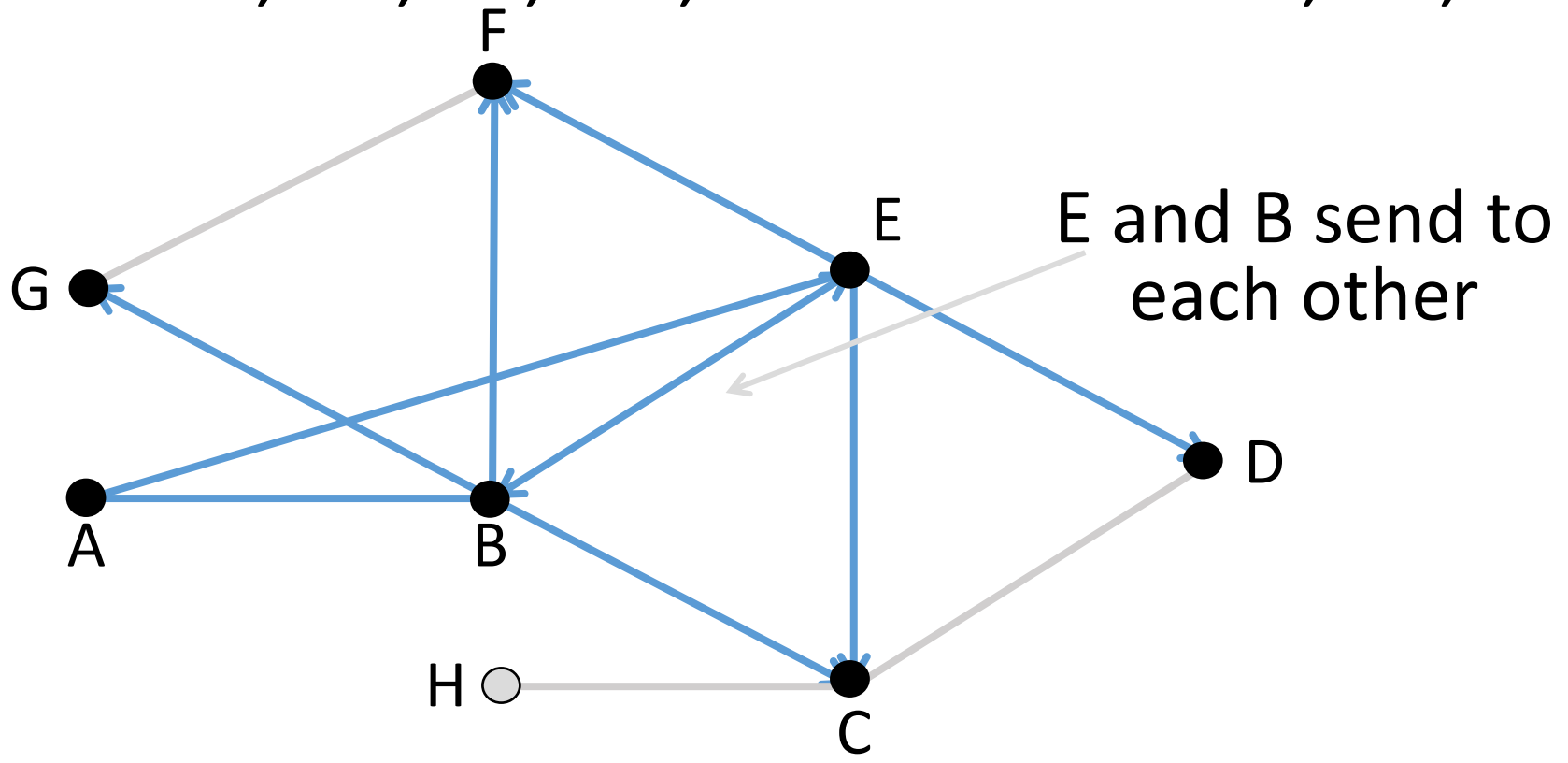
Flooding (2)

Consider a flood from A; first reaches B via AB, E via AE



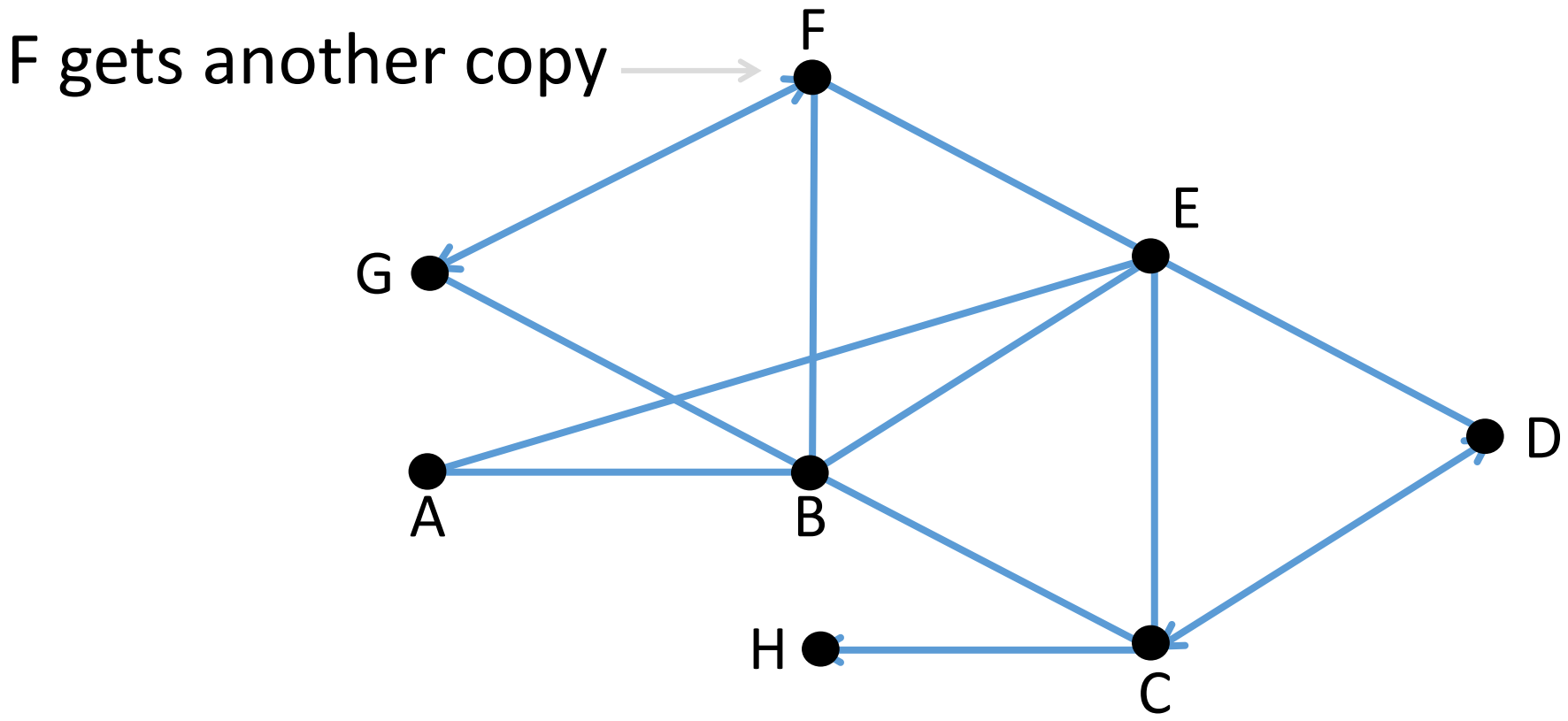
Flooding (3)

Next B floods BC, BE, BF, BG, and E floods EB, EC, ED, EF



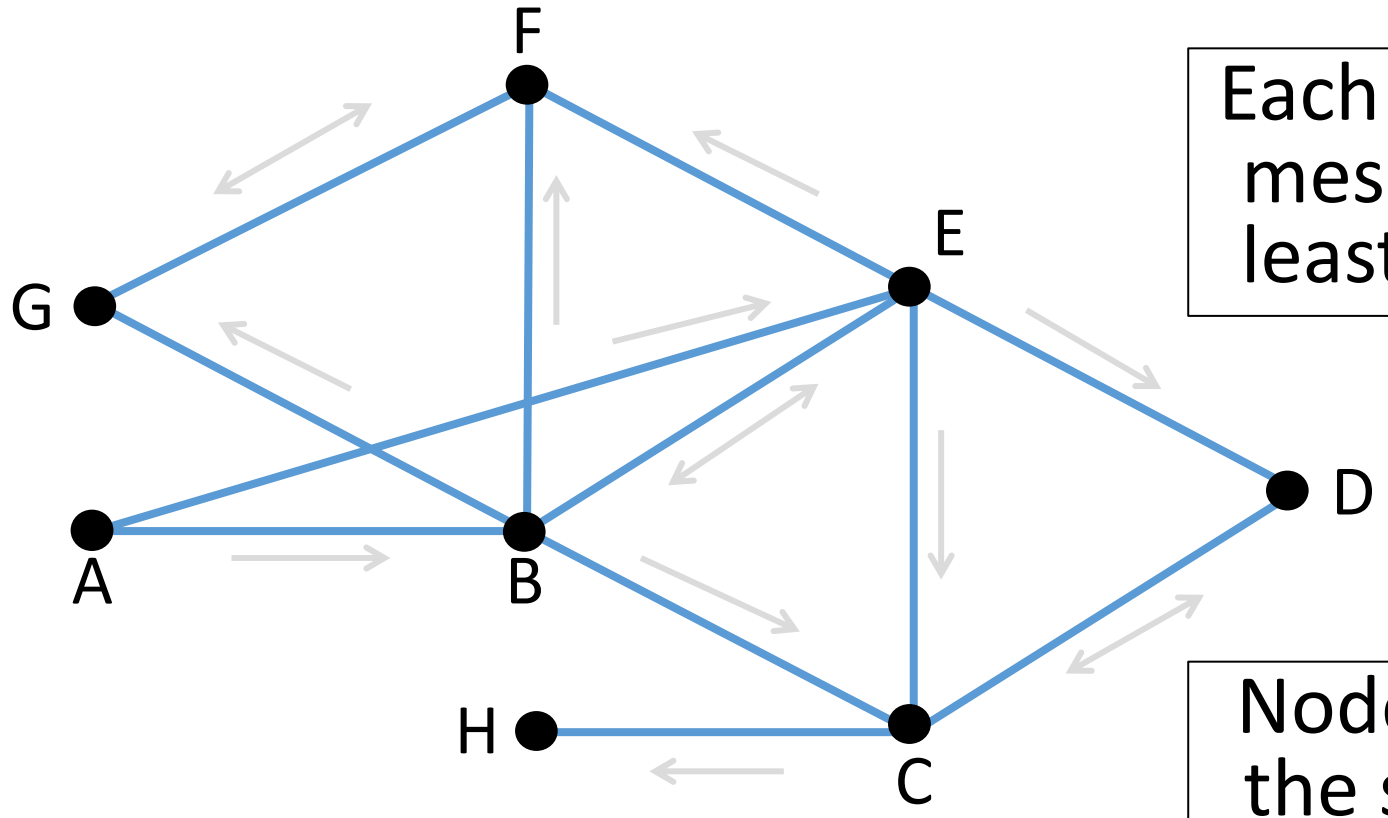
Flooding (4)

C floods CD, CH; D floods DC; F floods FG; G floods GF



Flooding (5)

H has no-one to flood … and we're done



Each link carries the message, and in at least one direction

Nodes may receive the same message multiple times

Dijkstra's Algorithm

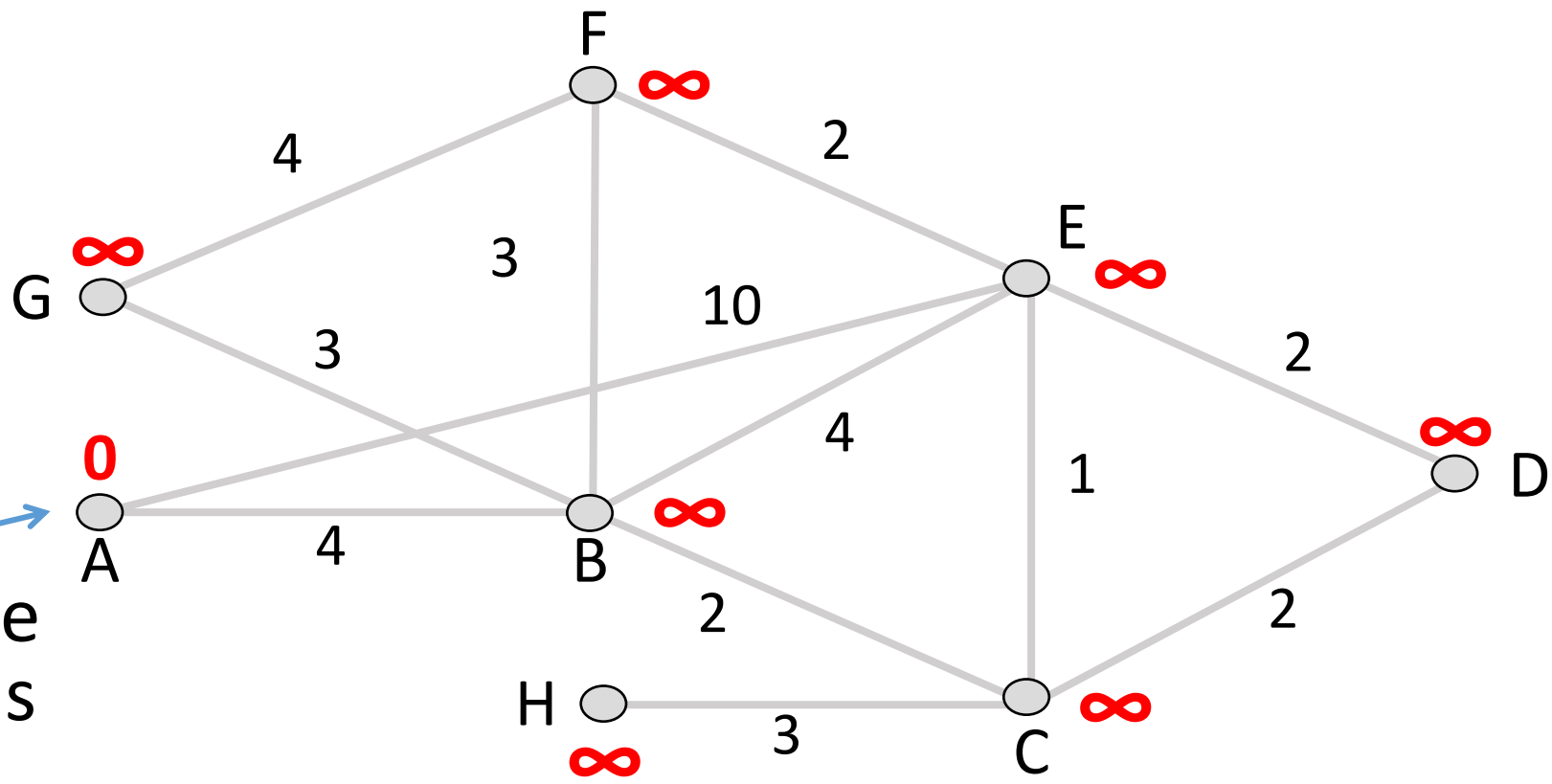
Mark all nodes tentative, set distances from source to 0 (zero) for source, and ∞ (infinity) for all other nodes

While tentative nodes remain:

- Extract N, a node with lowest distance
- Add link to N to the shortest path tree
- Relax the distances of neighbors of N by lowering any better distance estimates

Dijkstra's Algorithm (2)

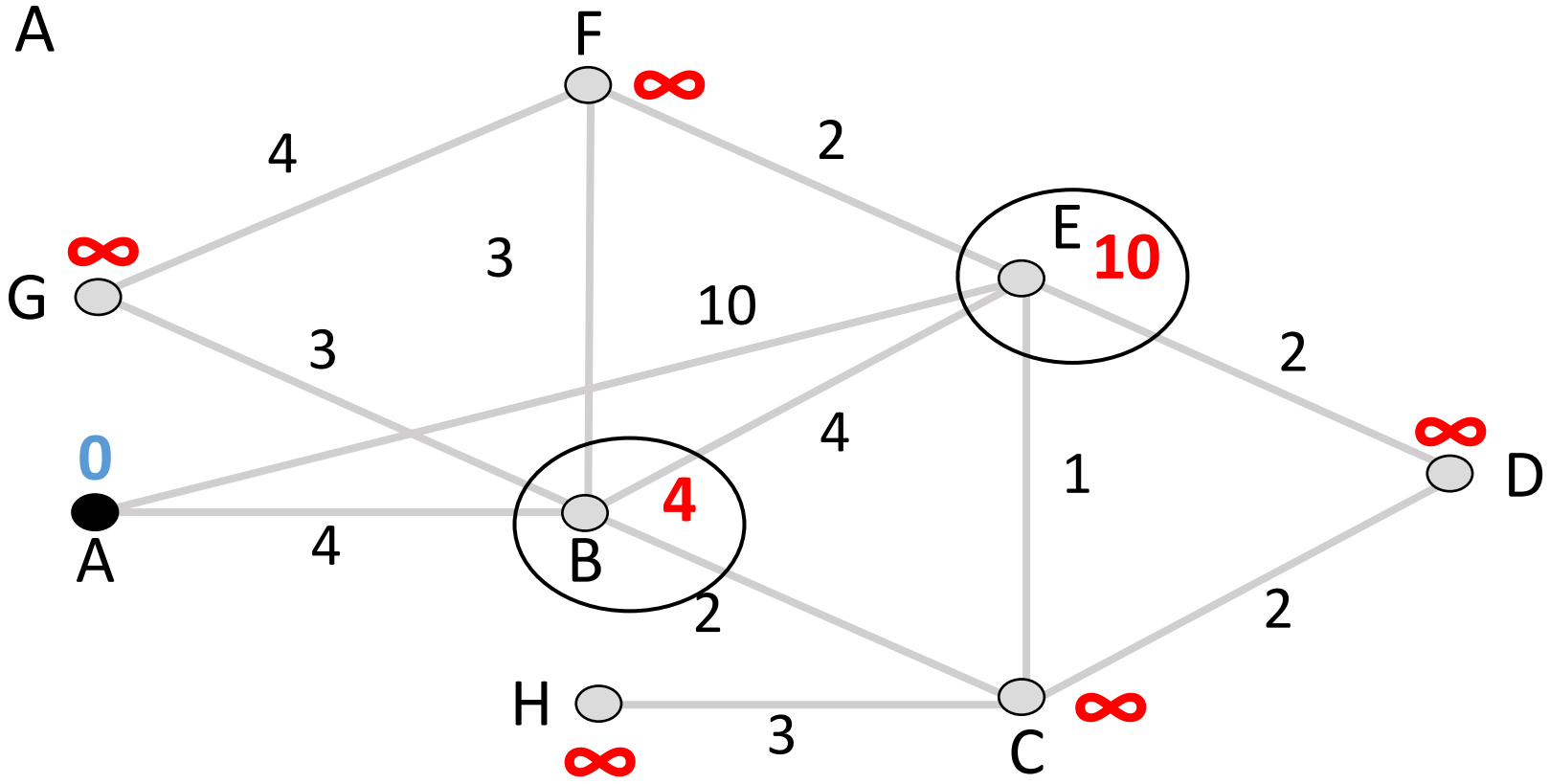
Initialization



We'll compute shortest paths from A

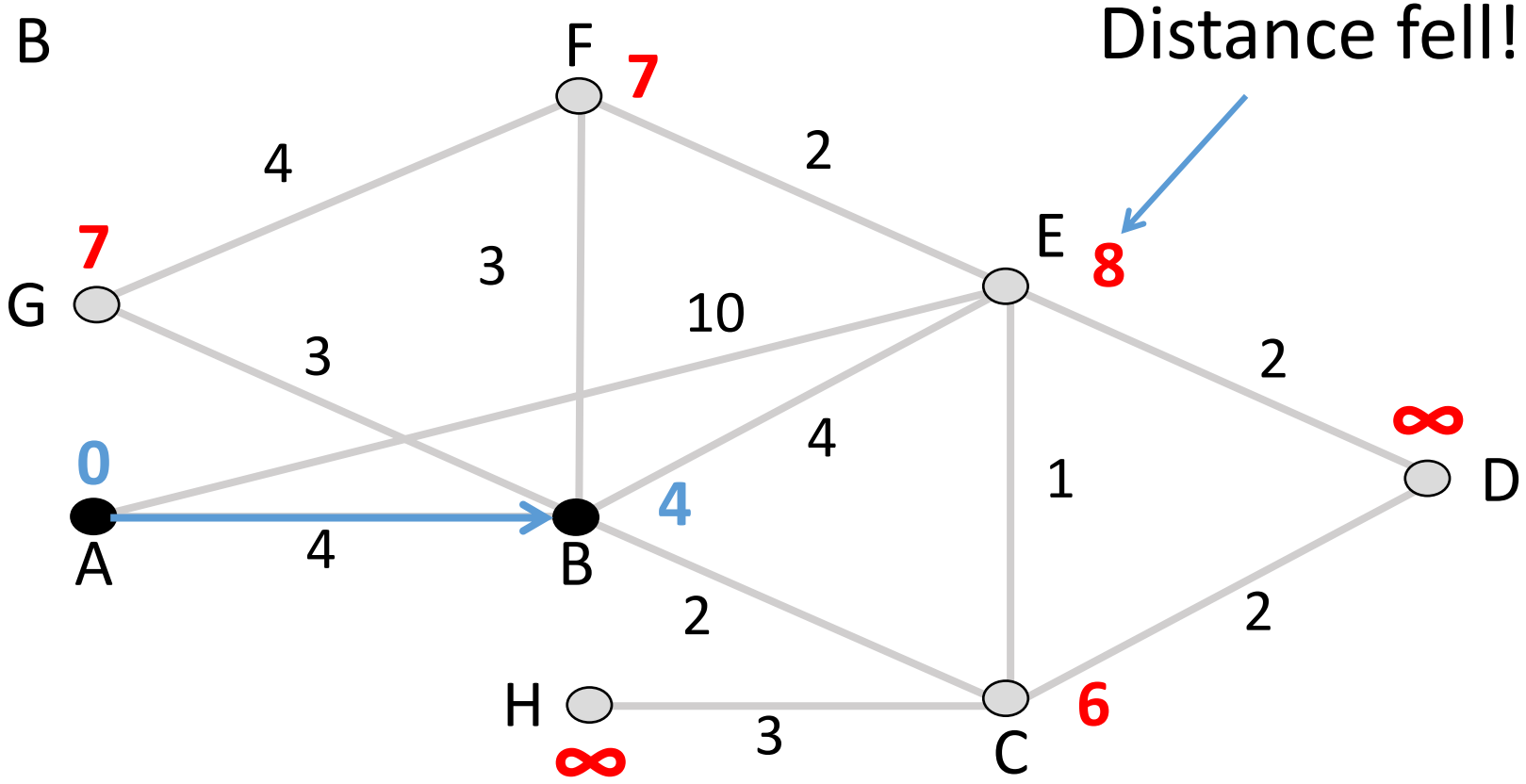
Dijkstra's Algorithm (3)

Relax around A



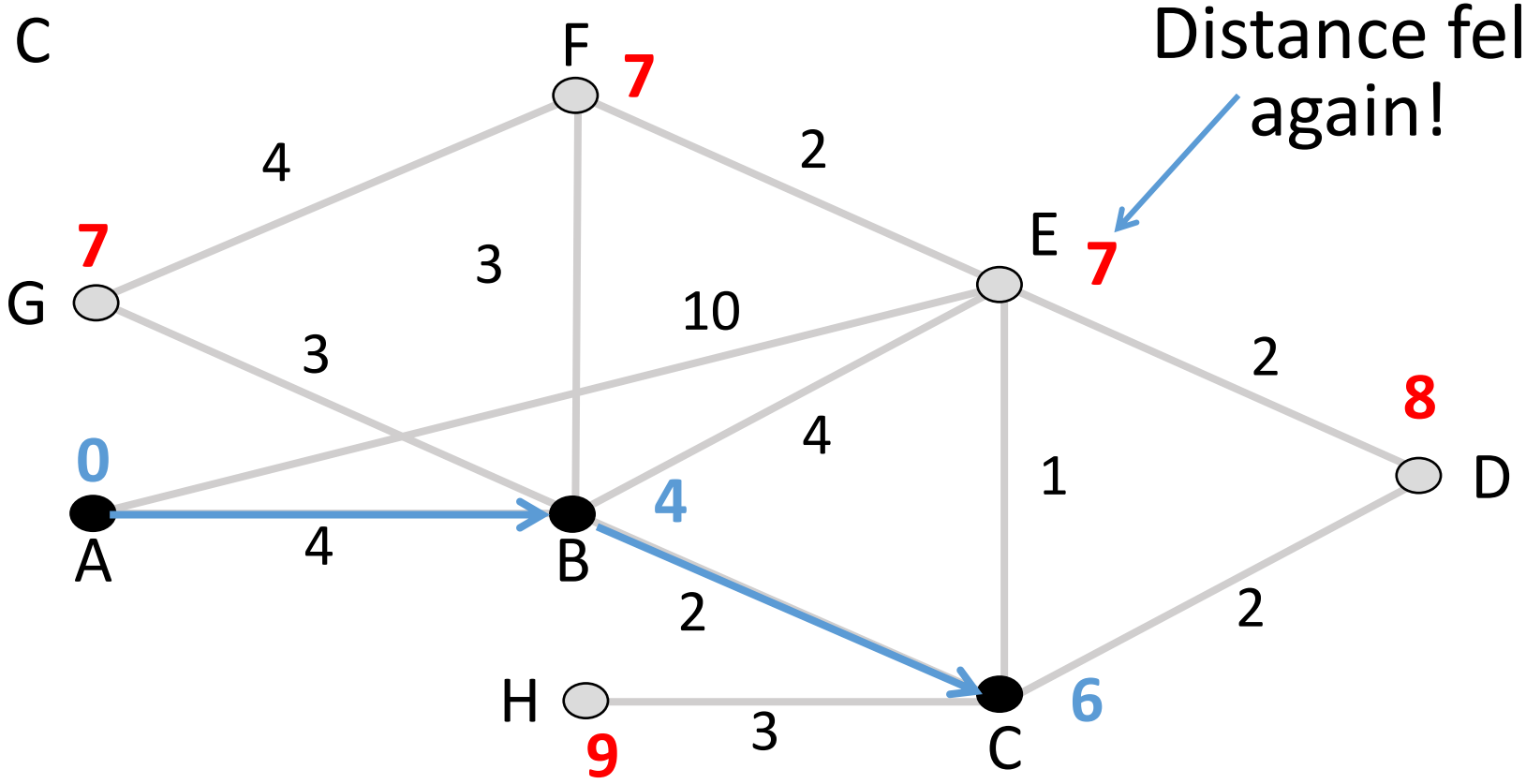
Dijkstra's Algorithm (4)

Relax around B



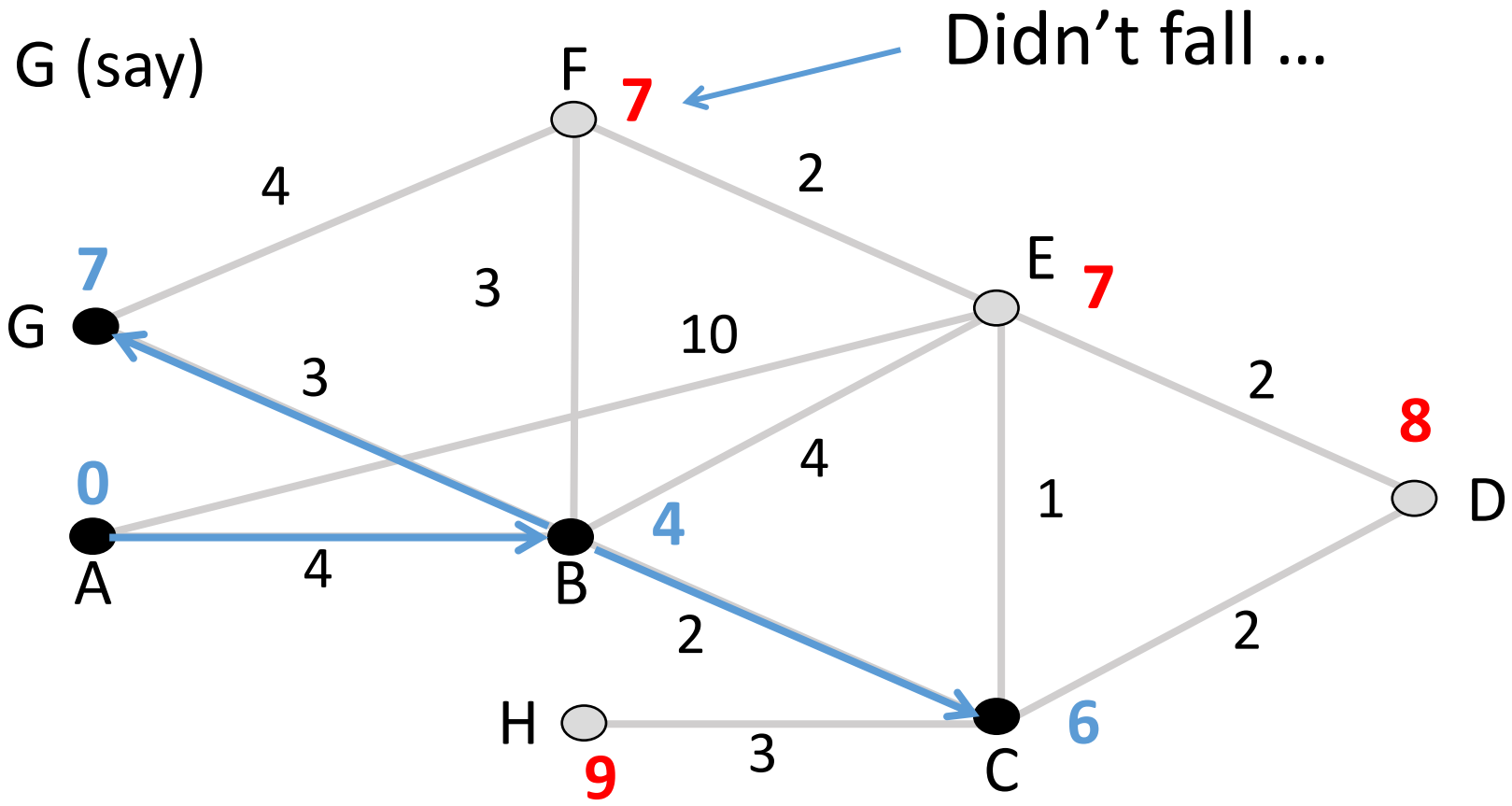
Dijkstra's Algorithm (5)

Relax around C



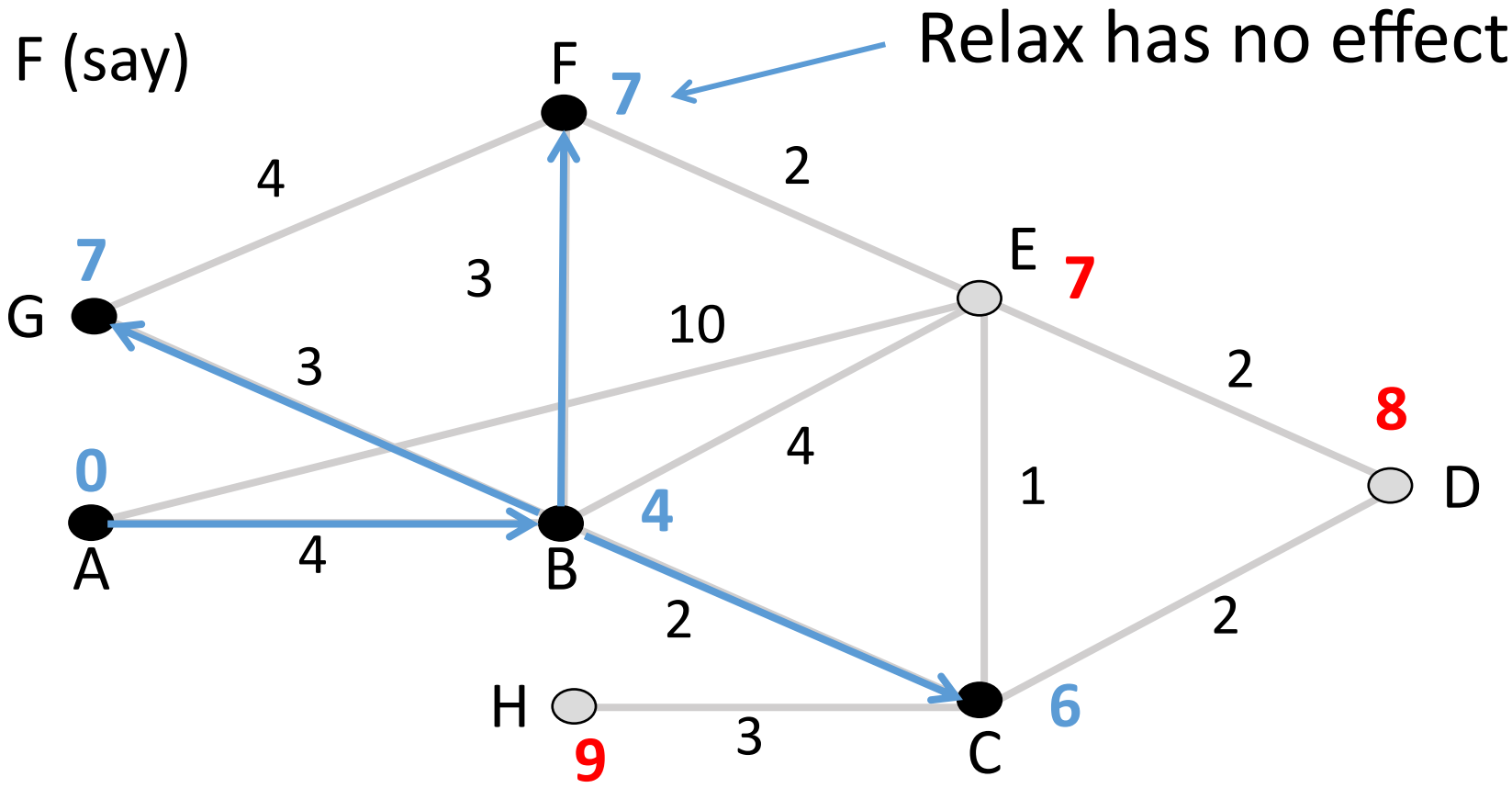
Dijkstra's Algorithm (6)

Relax around G (say)



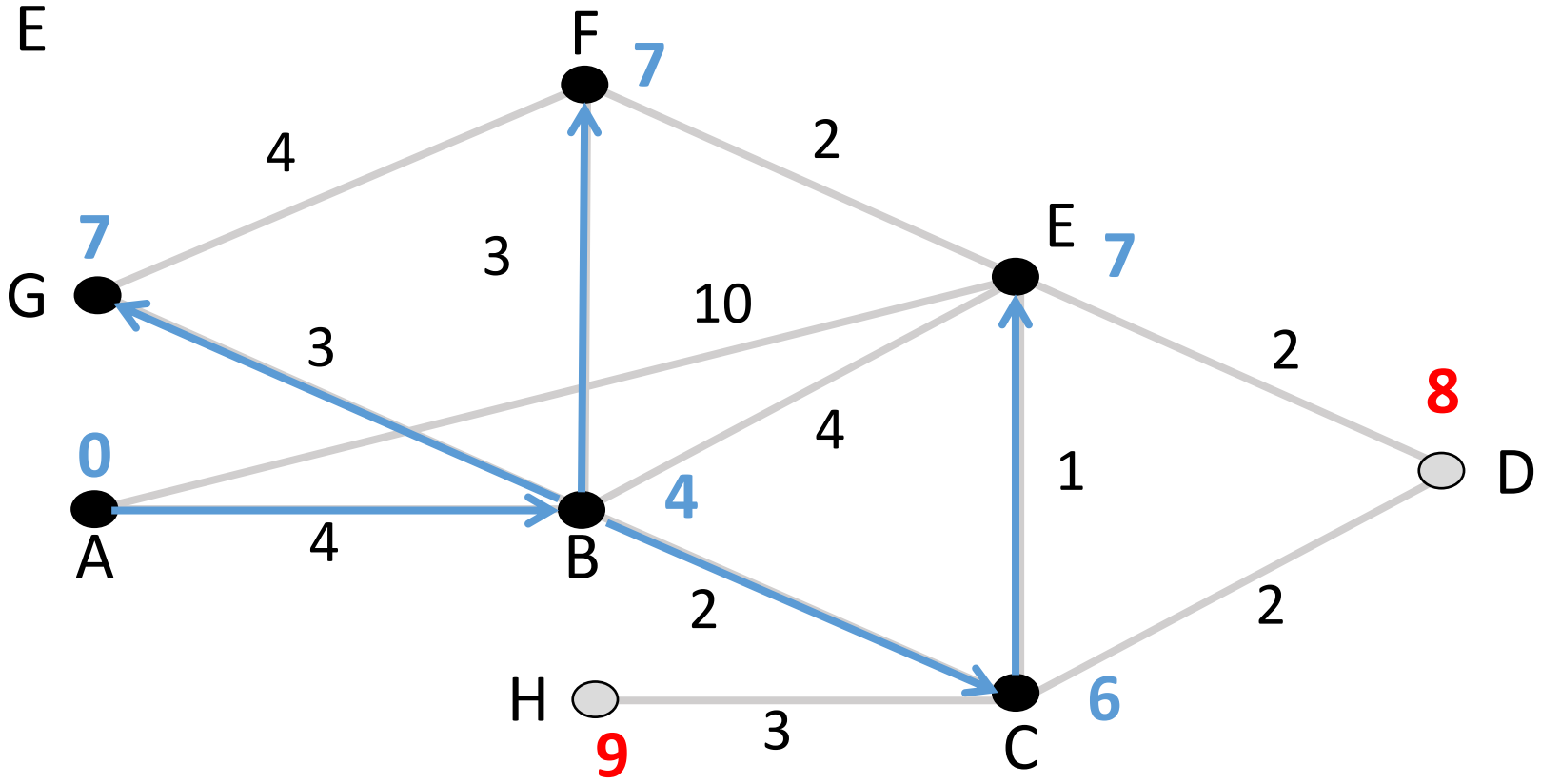
Dijkstra's Algorithm (7)

Relax around F (say)



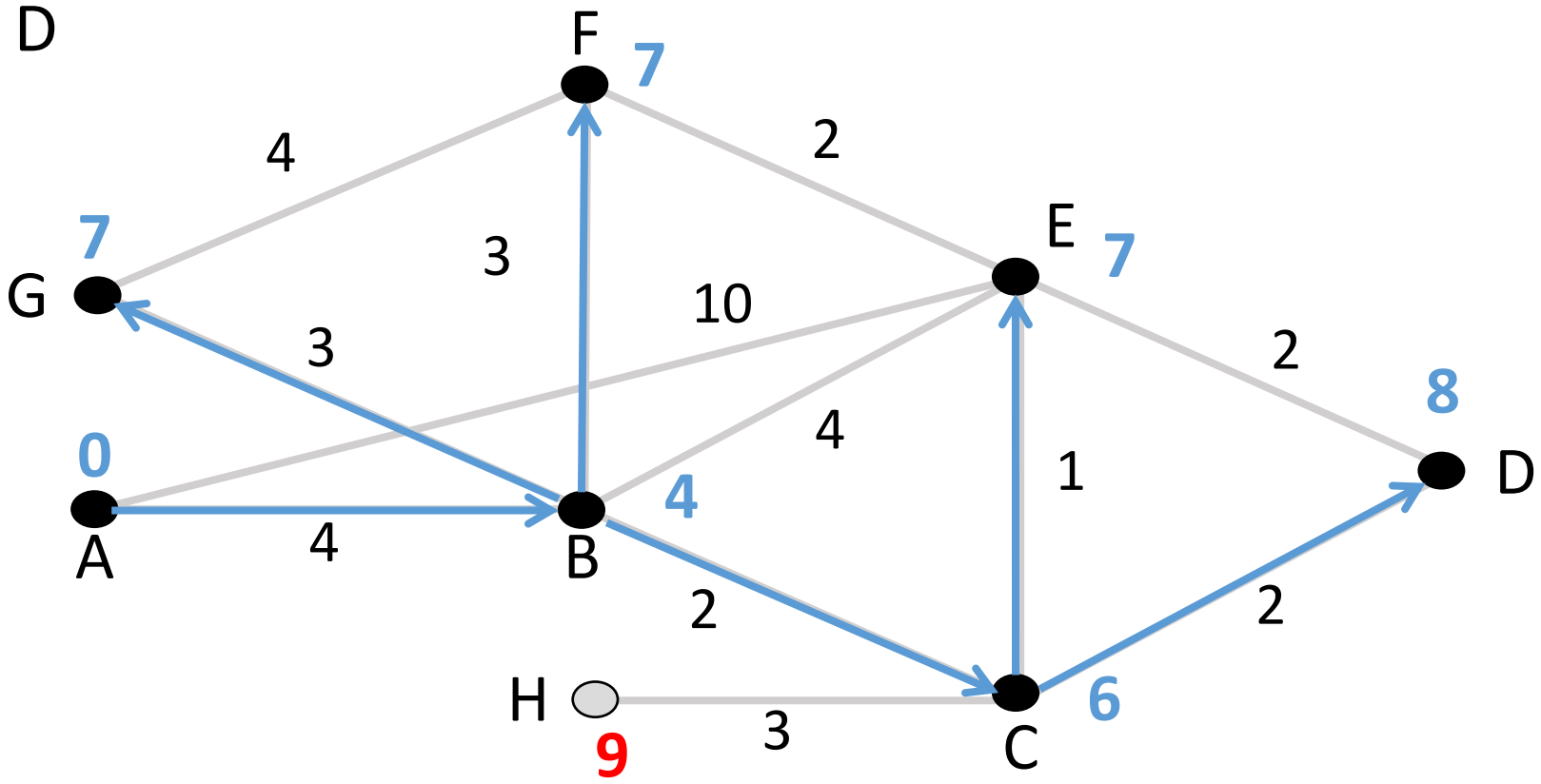
Dijkstra's Algorithm (8)

Relax around E



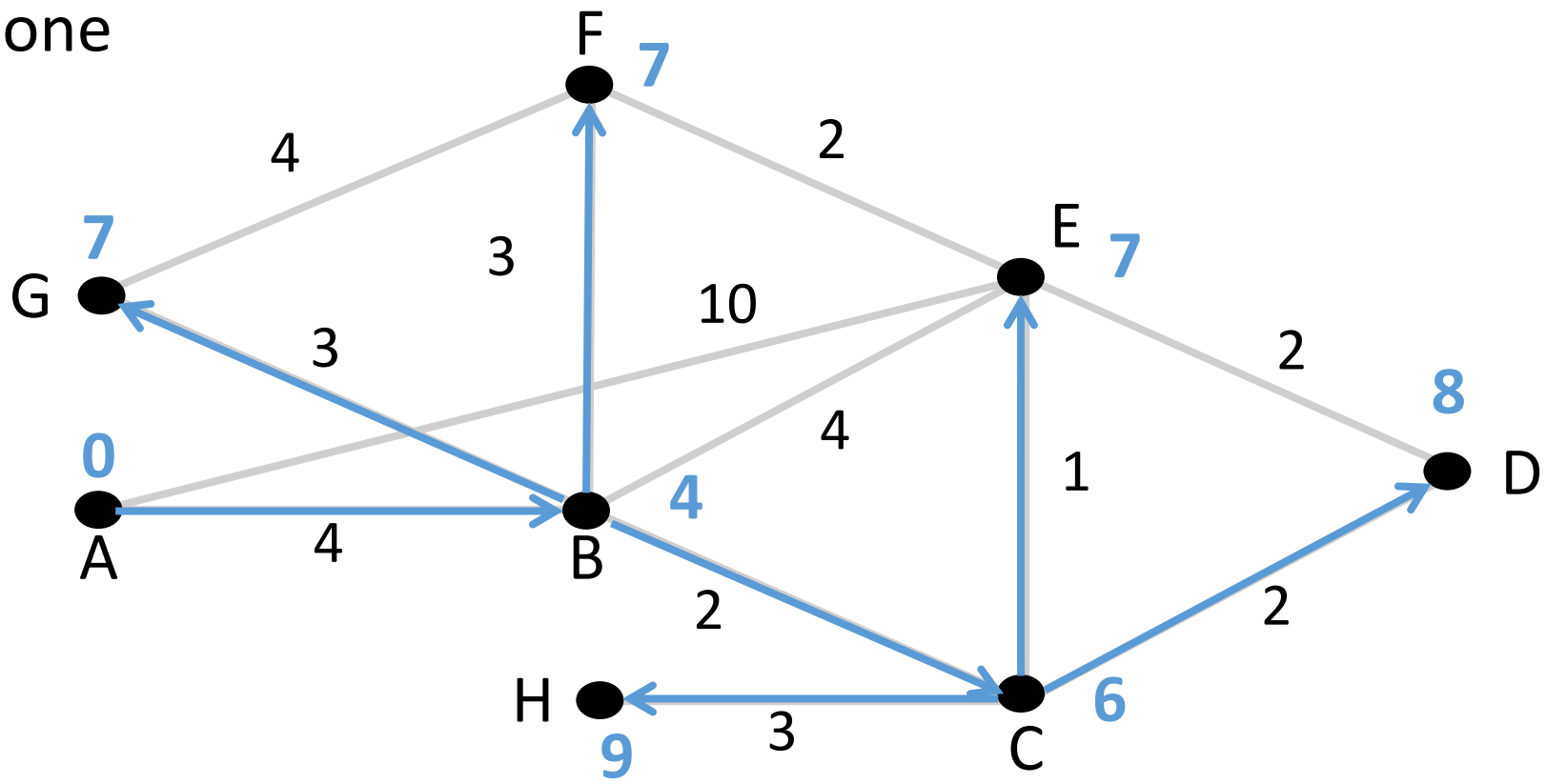
Dijkstra's Algorithm (9)

Relax around D



Dijkstra's Algorithm (10)

Finally, H … done



DV/LS Comparison

Both compute the same paths but differ in other ways

Goal	Distance Vector	Link-State
Convergence	Slow – many exchanges	Fast – flood and compute
Scalability	Excellent – storage/compute	Moderate – storage/compute

Link state is now favored except when resource-limited

Policy-based routing

Policy-based routing

Suppose each node was owned by a different organization

Each organization's interest differ (economic, political, security,..)

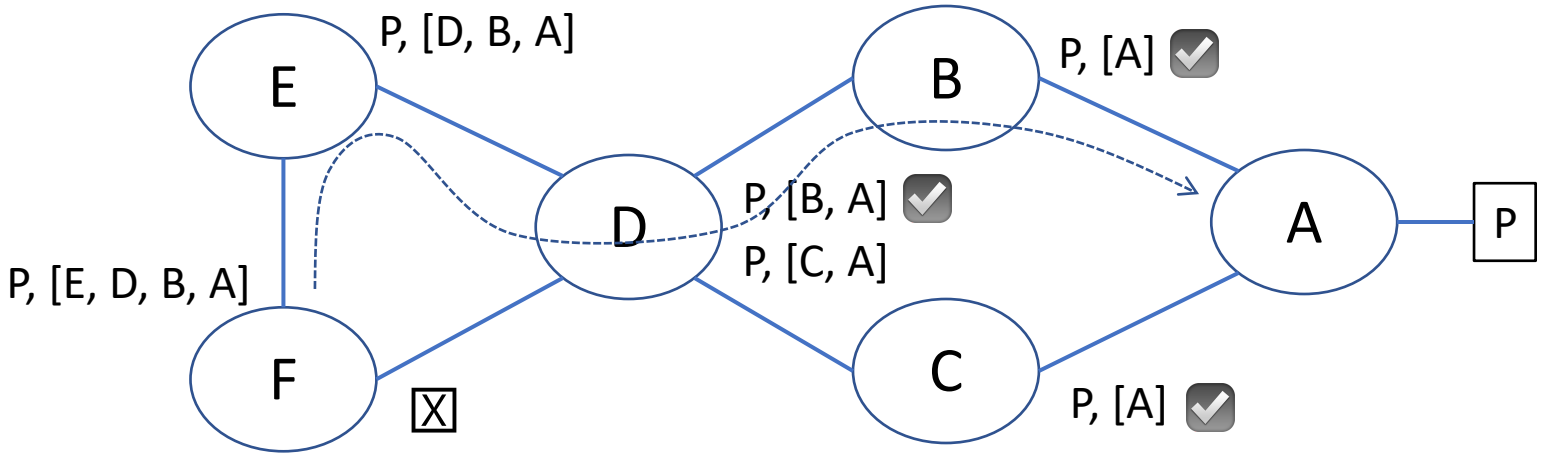
- Who you use to send traffic through
- Who can use you to send traffic

Internet's answer: Path-vector routing

Like distance-vector but

1. Embed full path in routing messages
2. Pick best among those obtained based on local policy
3. Send routing messages only to neighbors you are OK with routing through you

Path vector illustrated



Does not support arbitrary policies

- E does not get Path [D, C, A] even if it is preferred over [D, B, A]
- D may get F's traffic via a different path

No protocol can make everyone happy all the time – policy conflicts

Path vector convergence

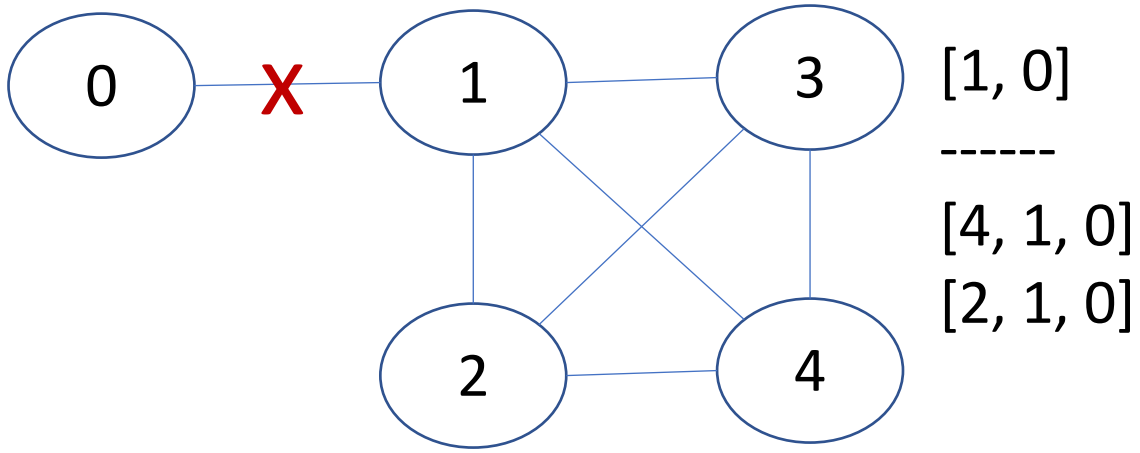
Avoiding loops was part of the motivation behind path vector

But path vector protocols have a version of count to infinity problem

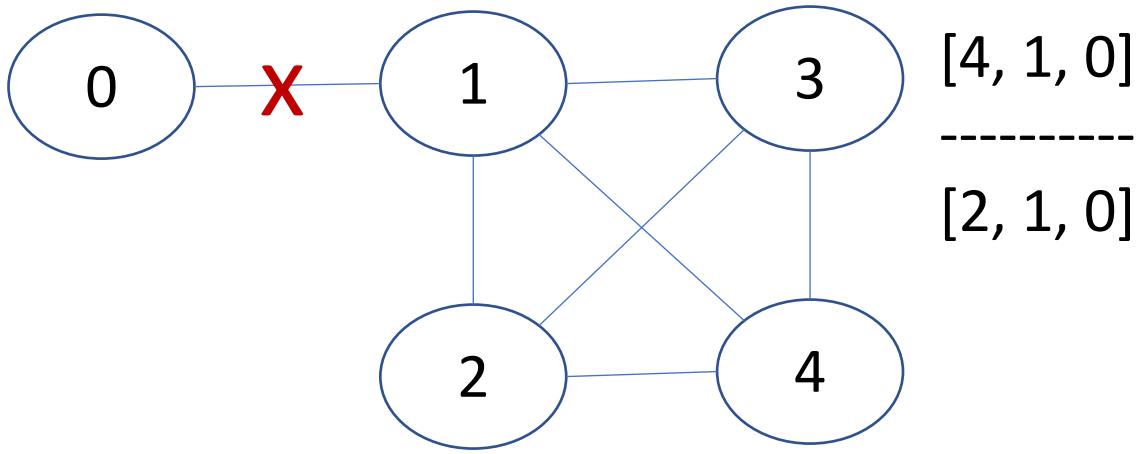
- Explore many non-existent paths

Worse, uncoordinated policies can lead to never converging

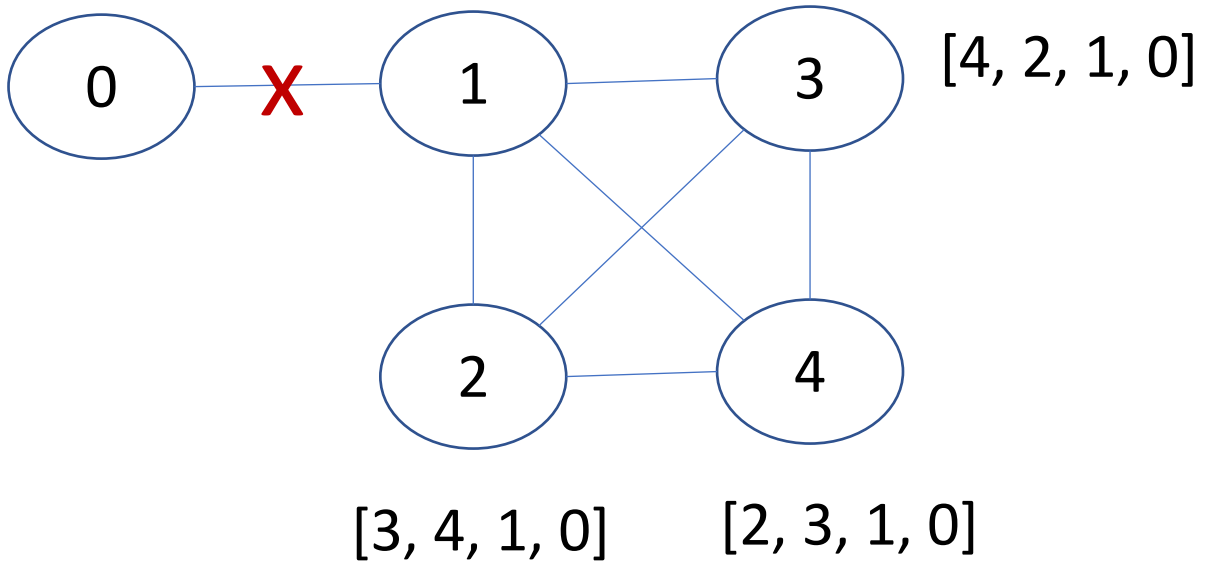
Slow convergence of path vector



Slow convergence of path vector

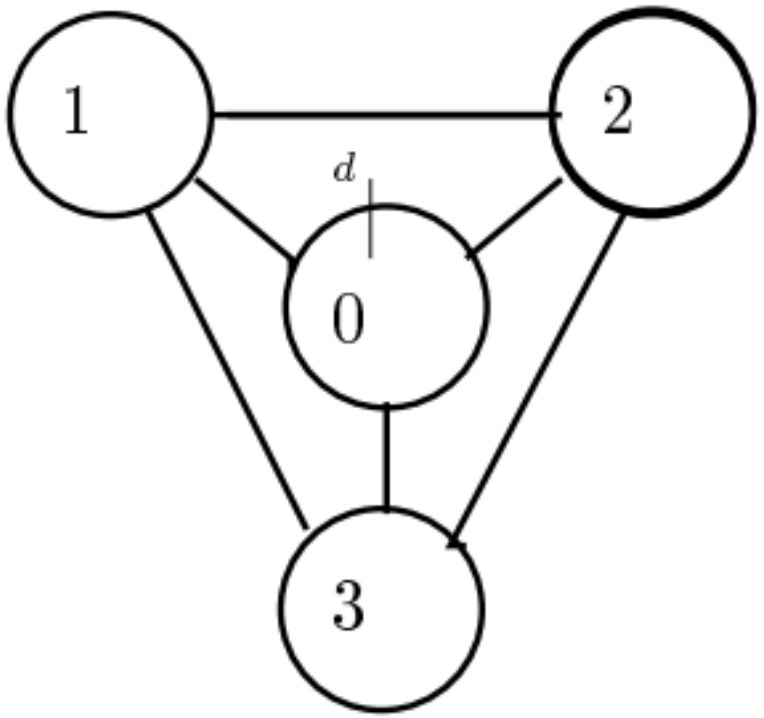


Slow convergence of path vector



Non-convergence of path vector

[2, 0] > [0] > [2, 3, 0]

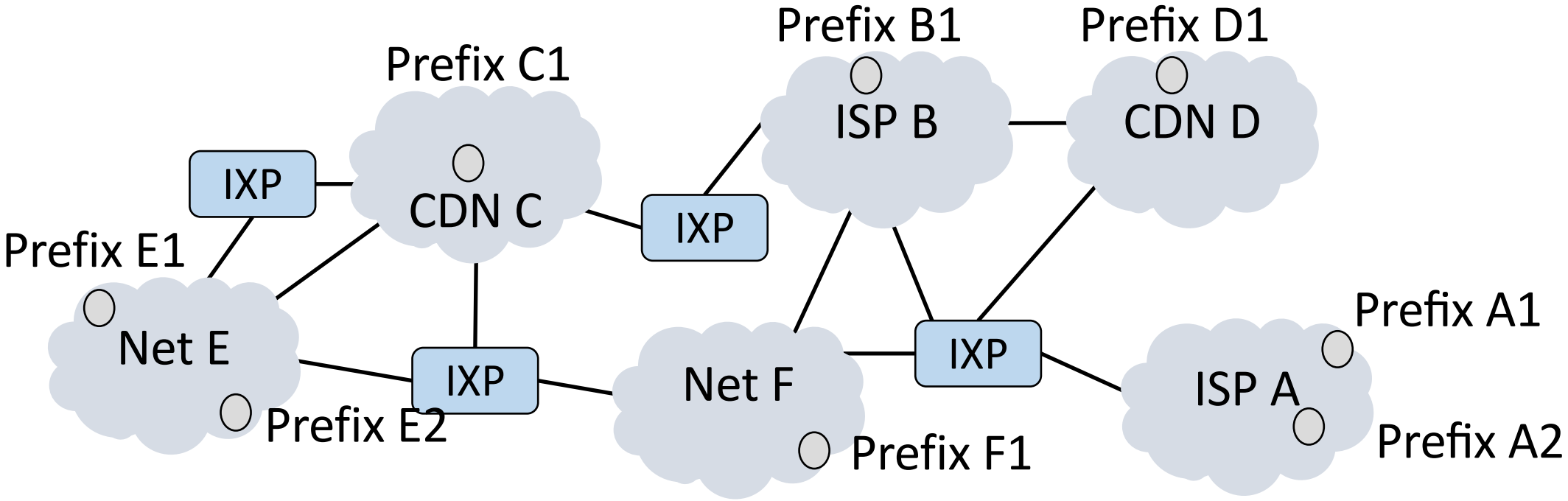


[3, 0] > [0] > [3, 1, 0]

[1, 0] > [0] > [1, 2, 0]

“bad gadget”

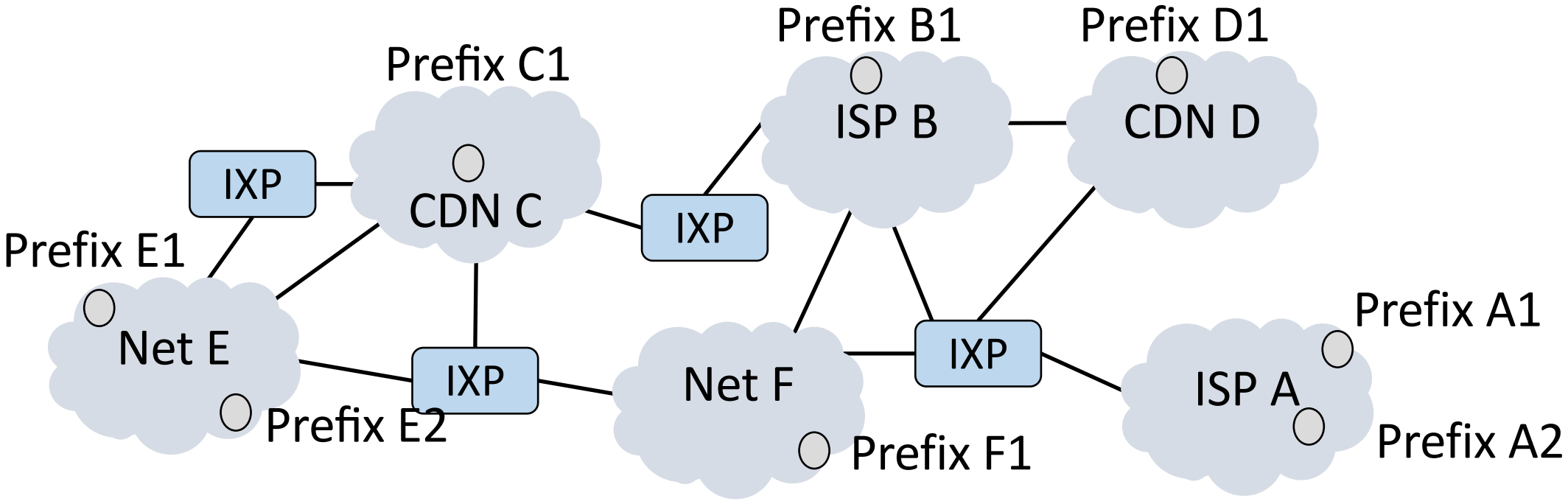
Structure of the Internet



Networks (ISPs, CDNs, etc.) have multiple IP prefixes

Networks are richly interconnected, often using IXPs

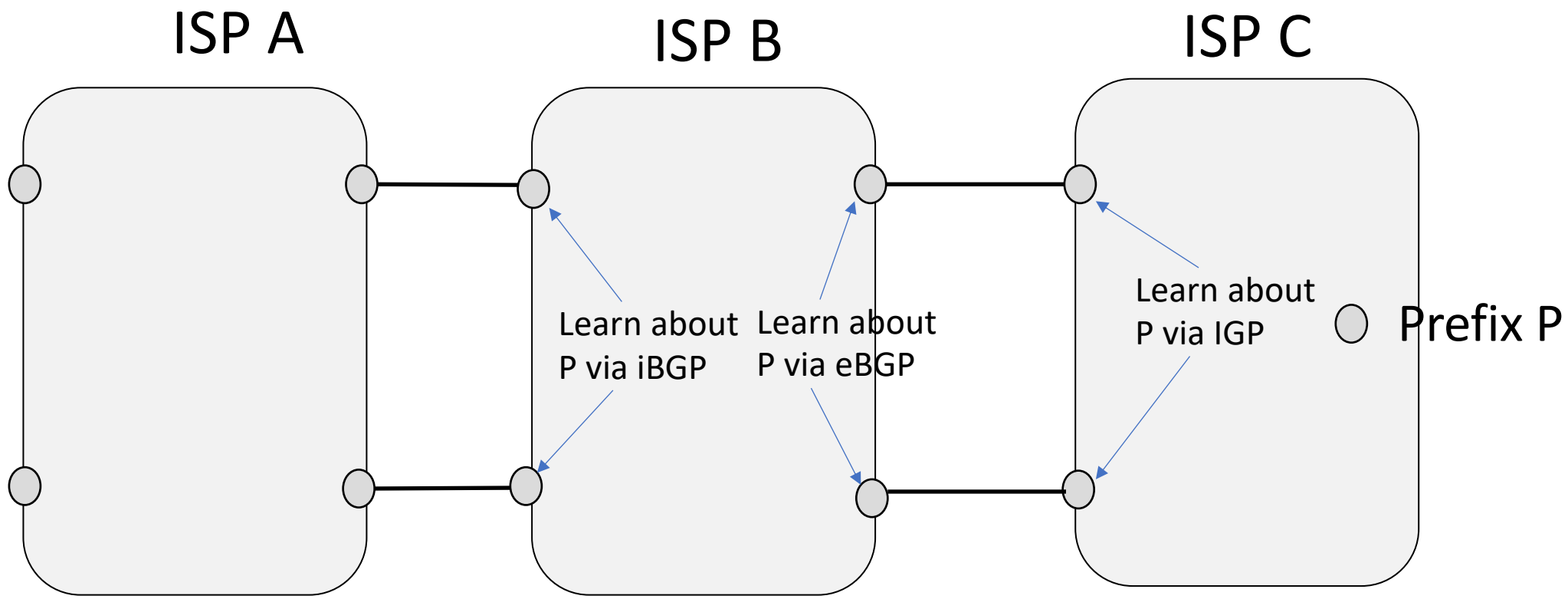
Structure of the Internet



Intra-domain routing within a network (IGP)

Inter-domain routing across networks (EGP)

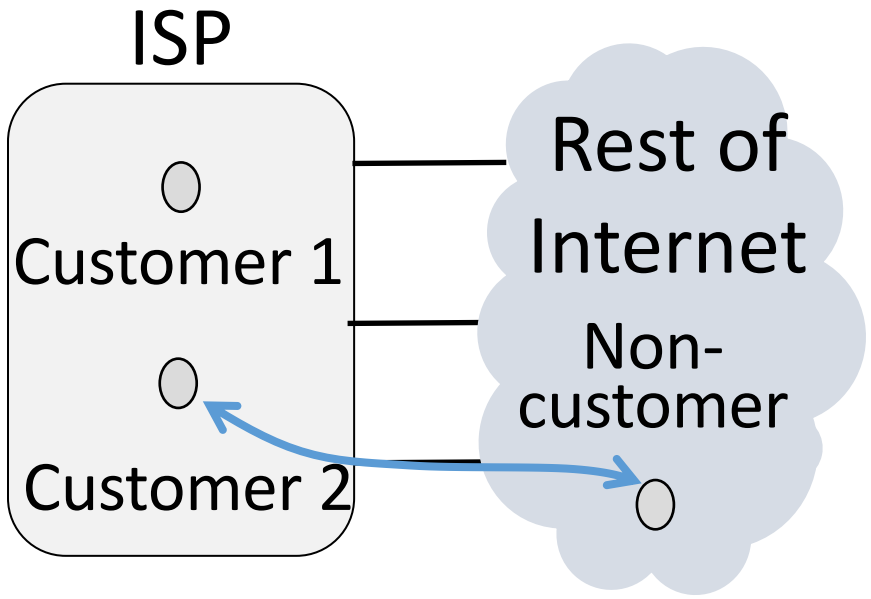
IGP, eBGP, iBGP



Common Routing Policies – Transit/Customer

Customer gets service from its *transit* provider

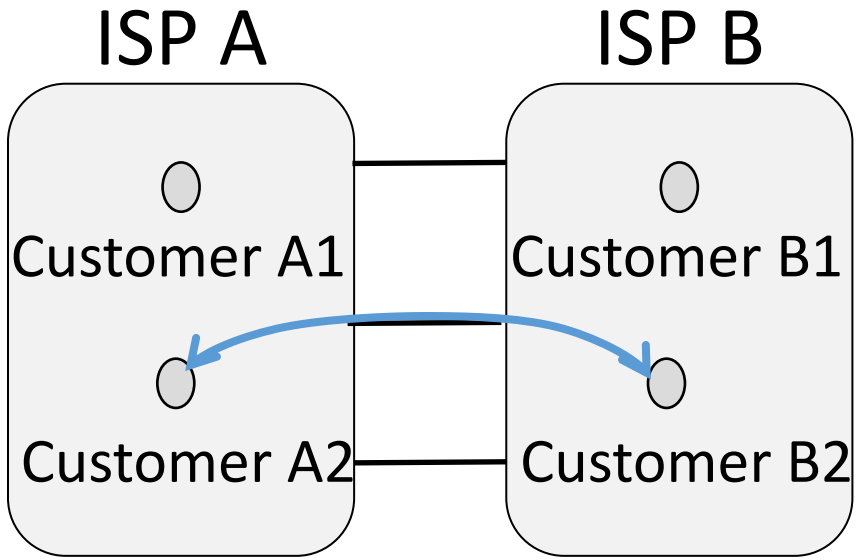
- Provider accepts traffic for customer from the rest of Internet
- Provider sends traffic from customer to the rest of Internet
- Customer pays provider for the service



Common Routing Policies – Peer

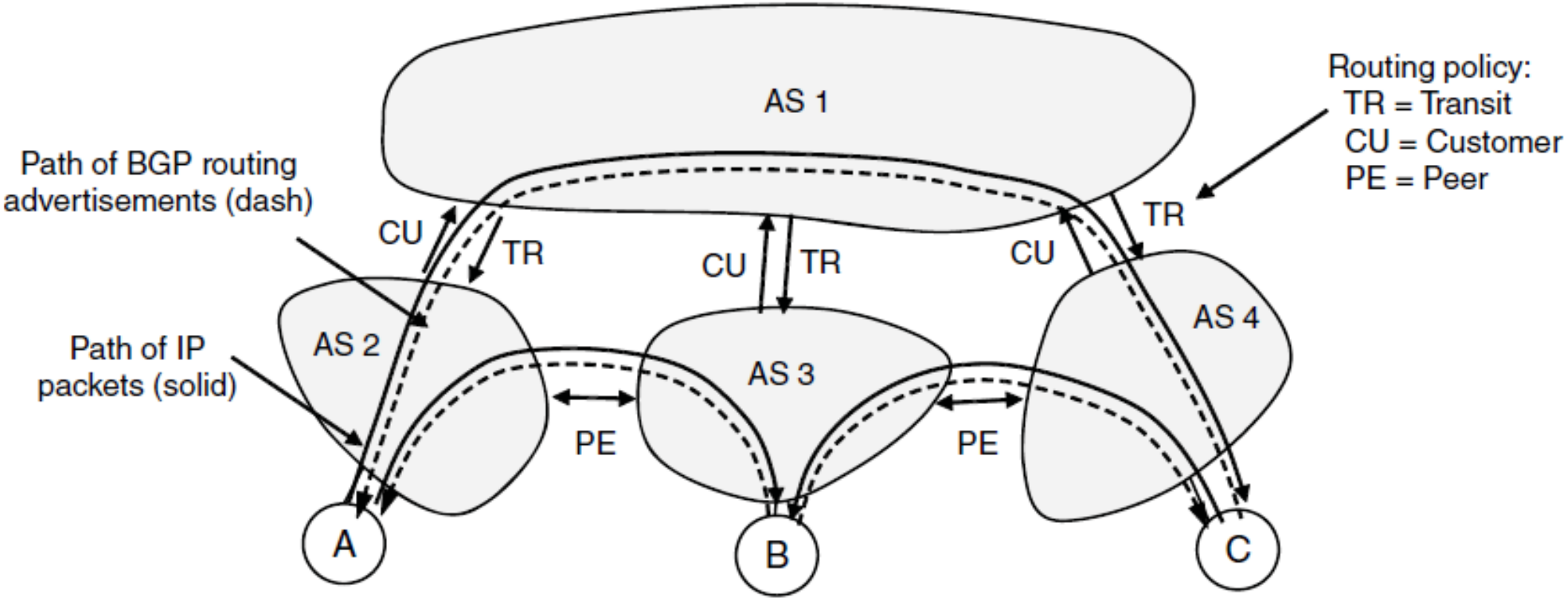
Parties get *PEER* service from each other

- Each peer accepts traffic from the other peer only for their customers
- Peers do not carry traffic to the rest of the Internet for each other
- Peers don’t pay each other



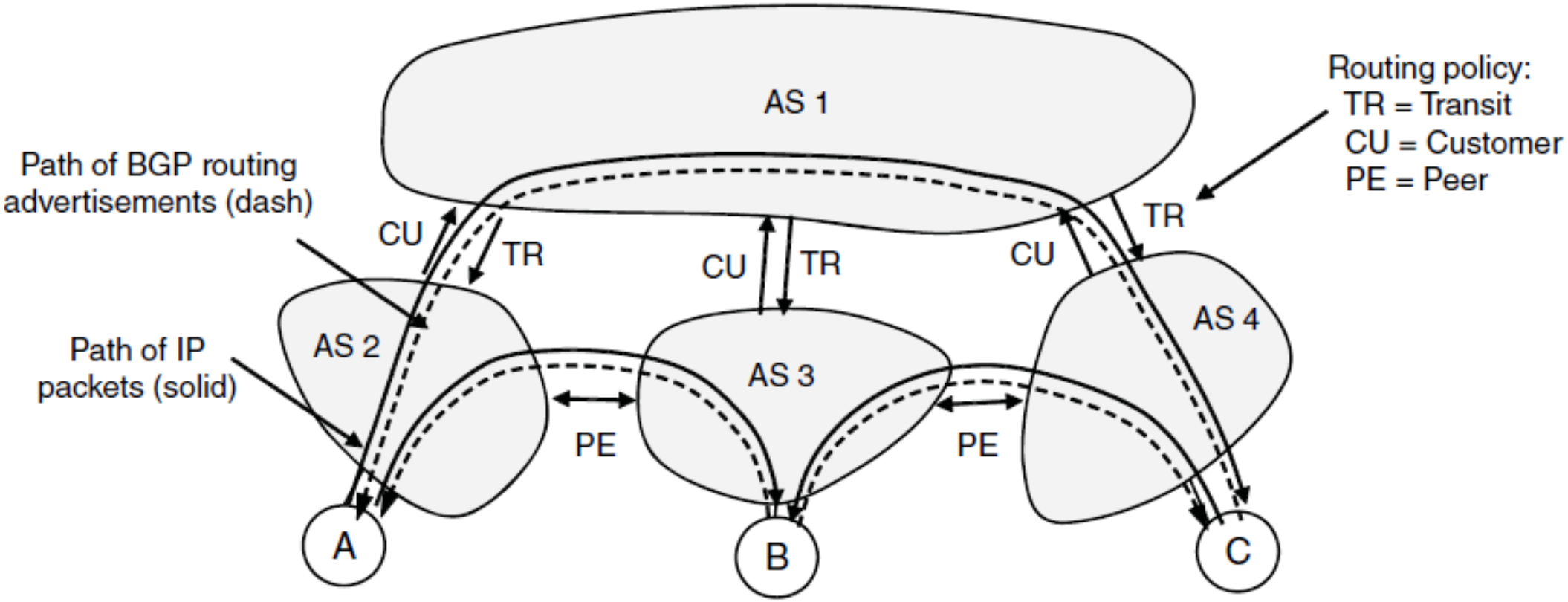
Routing with BGP (1)

TRANSIT: AS1 says B, [AS1, AS3], C, [AS1, AS4] to AS2



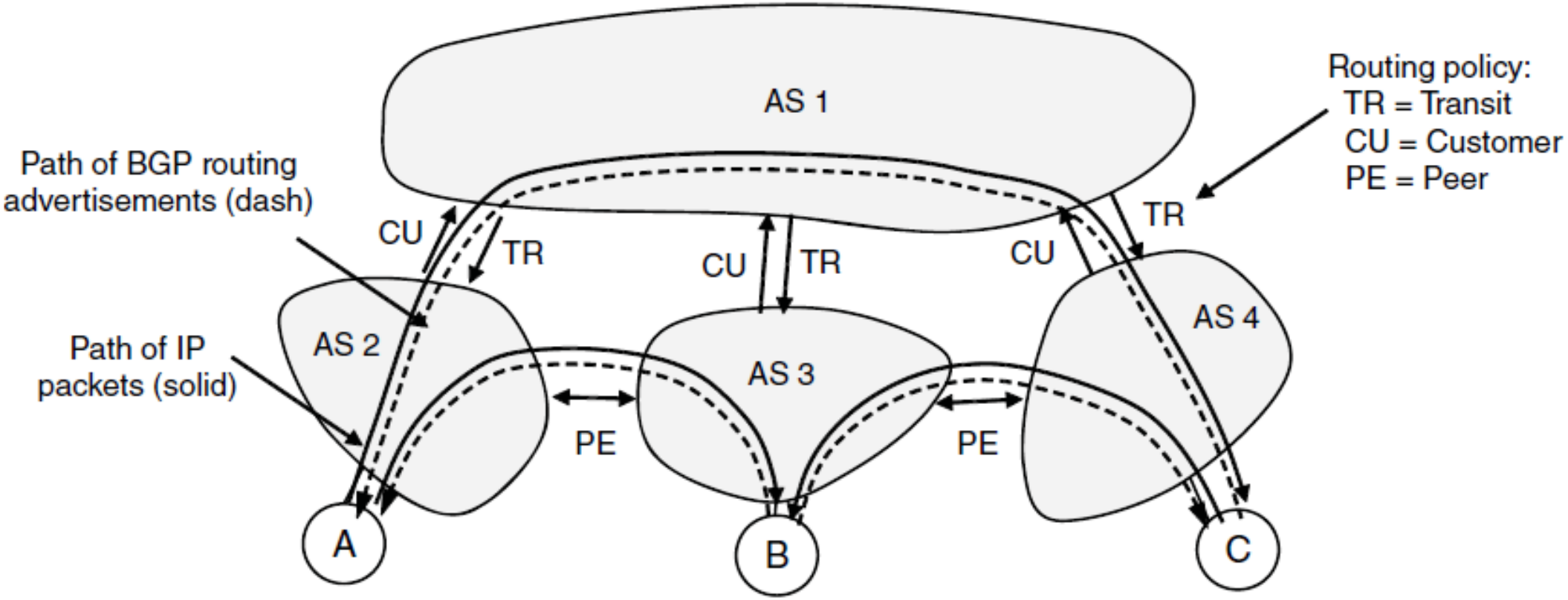
Routing with BGP (2)

CUSTOMER: AS2 says A, [AS2] to AS1



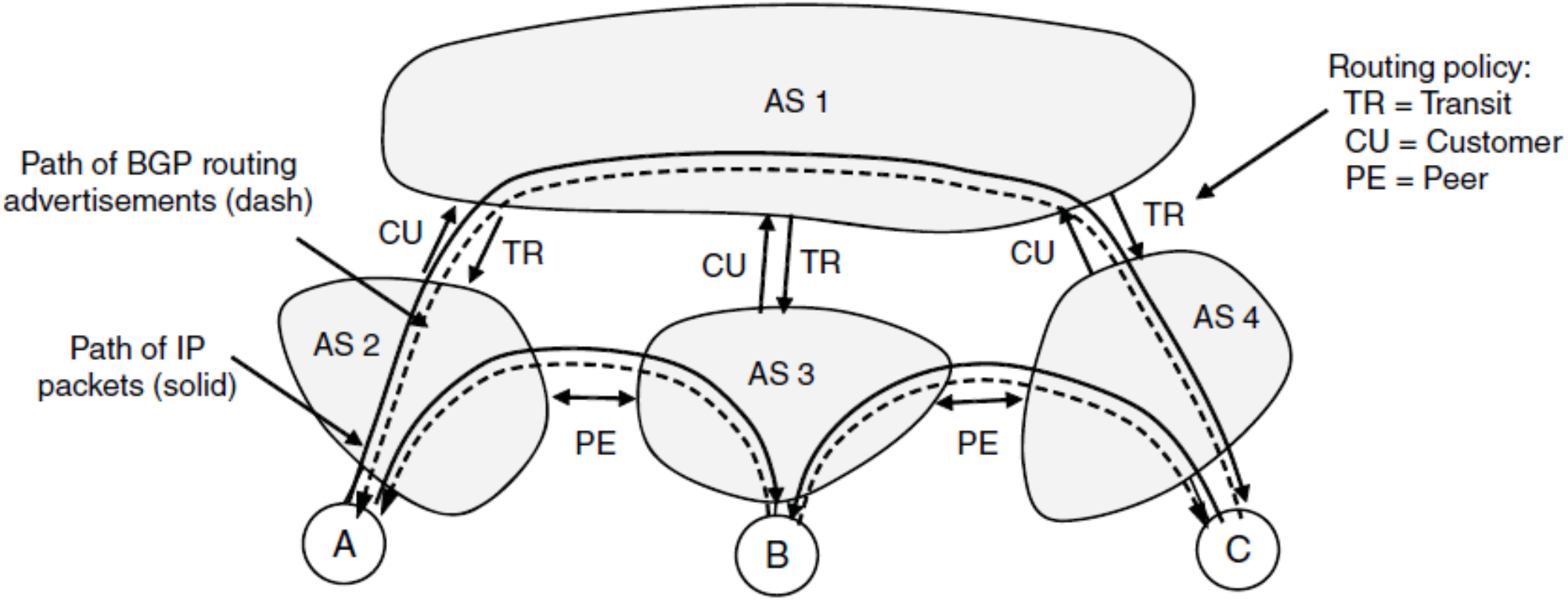
Routing with BGP (3)

PEER: AS2 says A, [AS2] to AS3, AS3 says B, [AS3] to AS2



Routing with BGP (4)

AS2 has two routes to B (via AS1, AS3); chooses AS3 (Free!)



Are these protocols computing good paths?

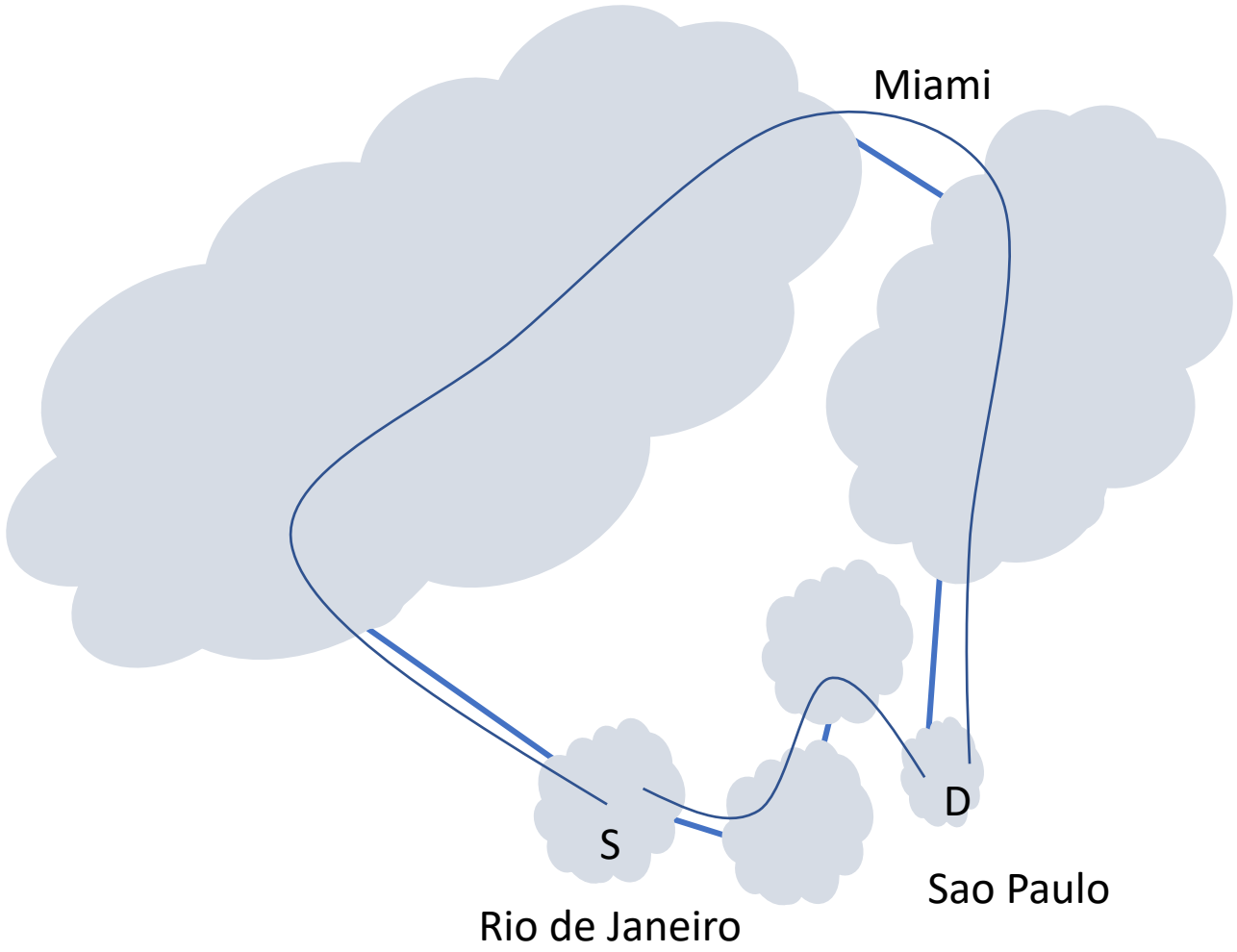
DV and LS

- Yes, as long as cost is meaningful
- But load is not part of cost

BGP

- Number of ISPs along the path is the default metric
  - Can produce highly circuitous paths because ISPs are different sizes
- Policy makes it even worse

Effect of path length



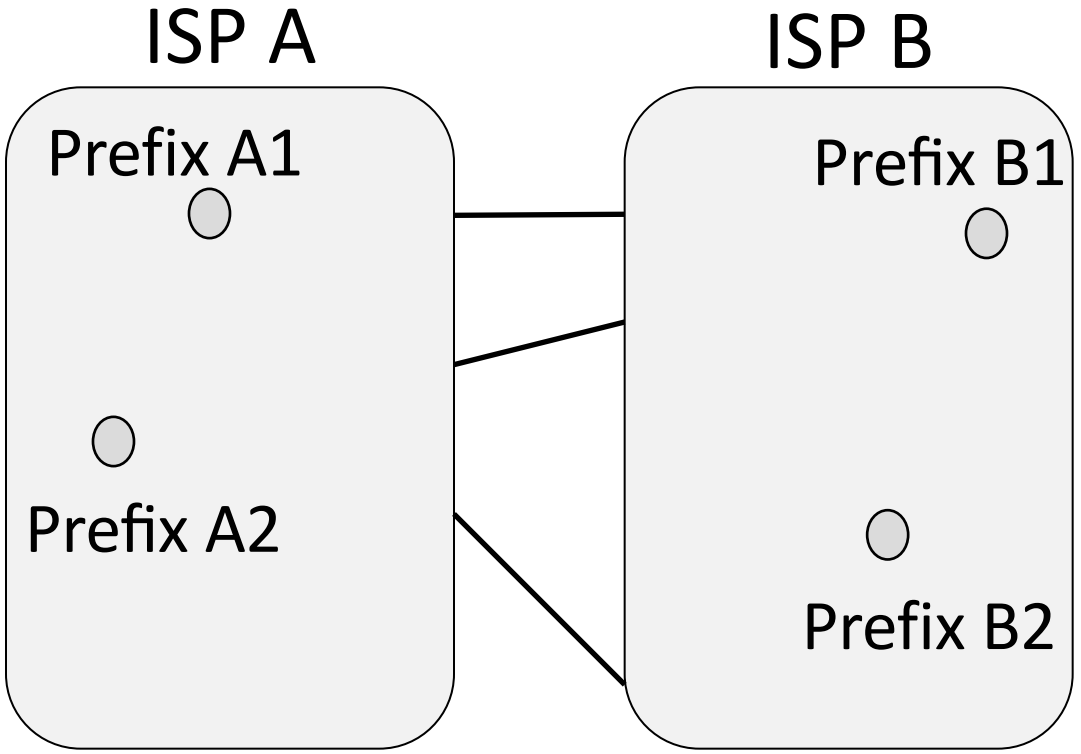
Effects of independent parties

Each party selects routes to suit its own interests

- E.g, shortest path in its network

What path will be chosen for A2→B1 and B1→A2?

- What is the best path?

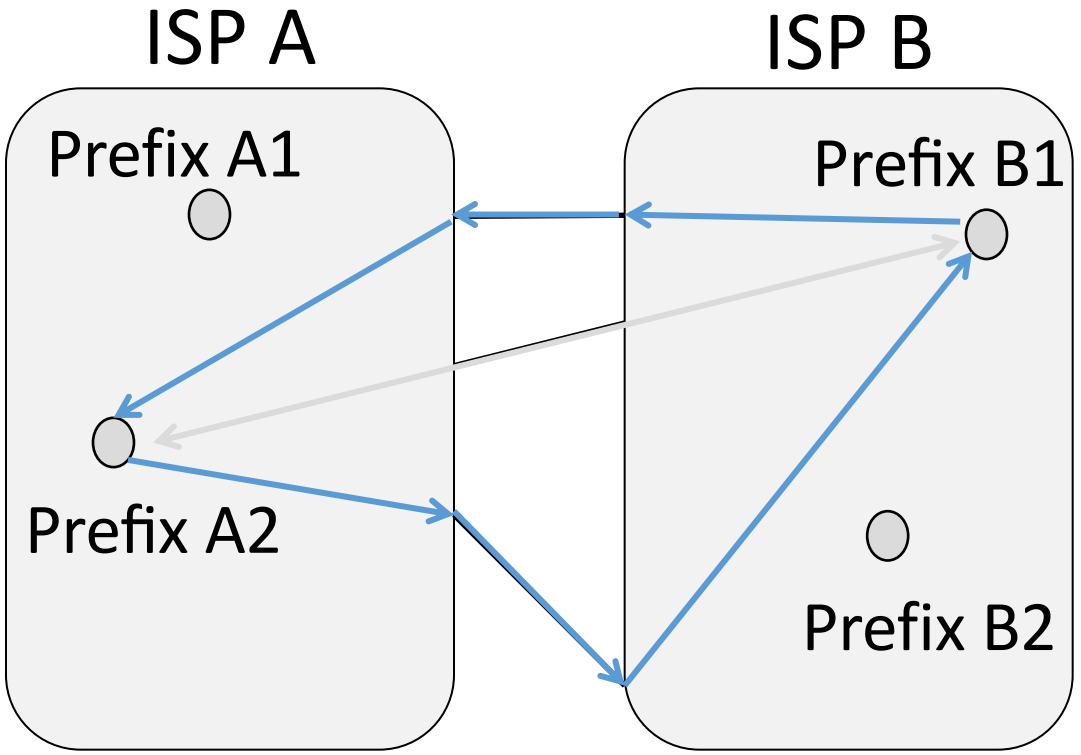


Effects of independent parties (2)

Selected paths are longer than overall shortest path

- And asymmetric too!

Consequence of independent goals and decisions



BGP paths in practice

Good enough in the average case but long tail

ISPs and others play whack-a-mole with long paths in the tail

BGP hijacking

For two hours, a large chunk of European mobile traffic was rerouted through China

It was China Telecom, again. The same ISP accused last year of "hijacking the vital internet backbone of western countries."

By Catalin Cimpanu for Zero Day | June 7, 2019 -- 19:41 GMT (12:41 PDT) | Topic: Security

MORE FROM CATALIN CIMPANU

Security
Google reveals

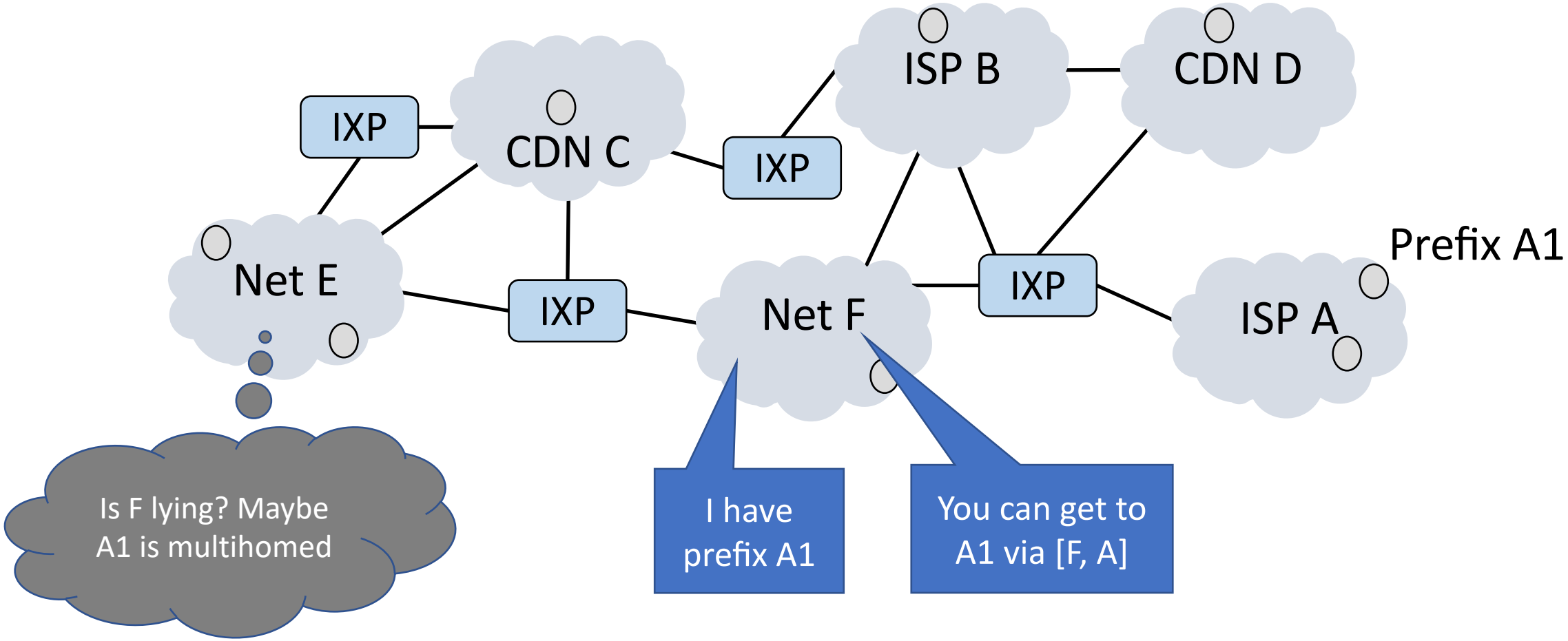
BORDER GATEWAY PROTOCOL ATTACK —

Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency

Almost 1,300 addresses for Amazon Route 53 rerouted for two hours.

DAN GOODIN - 4/24/2018, 12:00 PM

BGP hijacking



Solution approaches

Data analysis

- Too much noise; does not prevent “accidents”

Routing registries

- Updating and using the information is optional

Cryptographic signatures to protect origins or paths

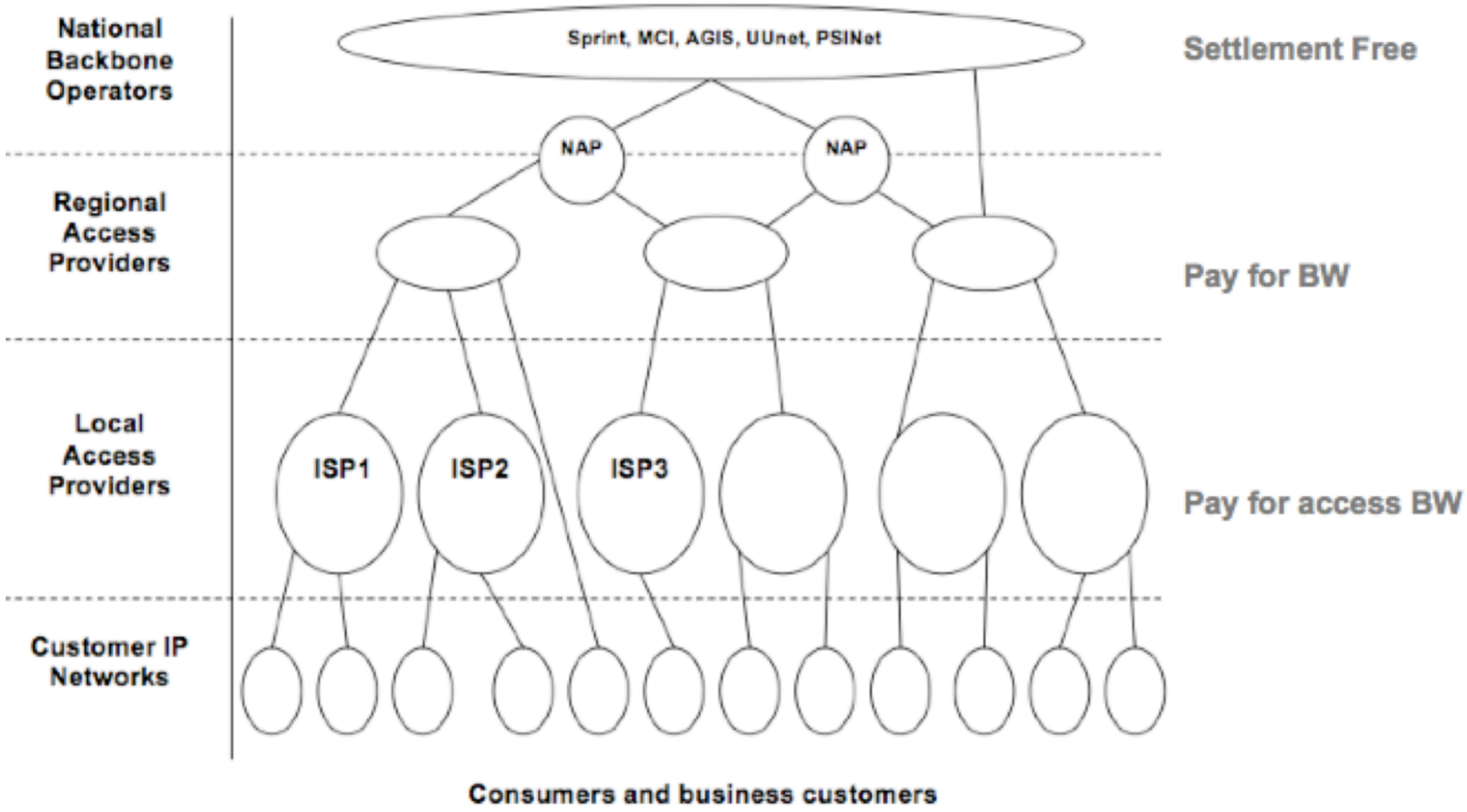
- High overhead (so they say) but RPKI gaining traction

“Flattening” of the Internet

Internet structure is being reshaped by cloud providers that want to get closer to the customers for performance reasons

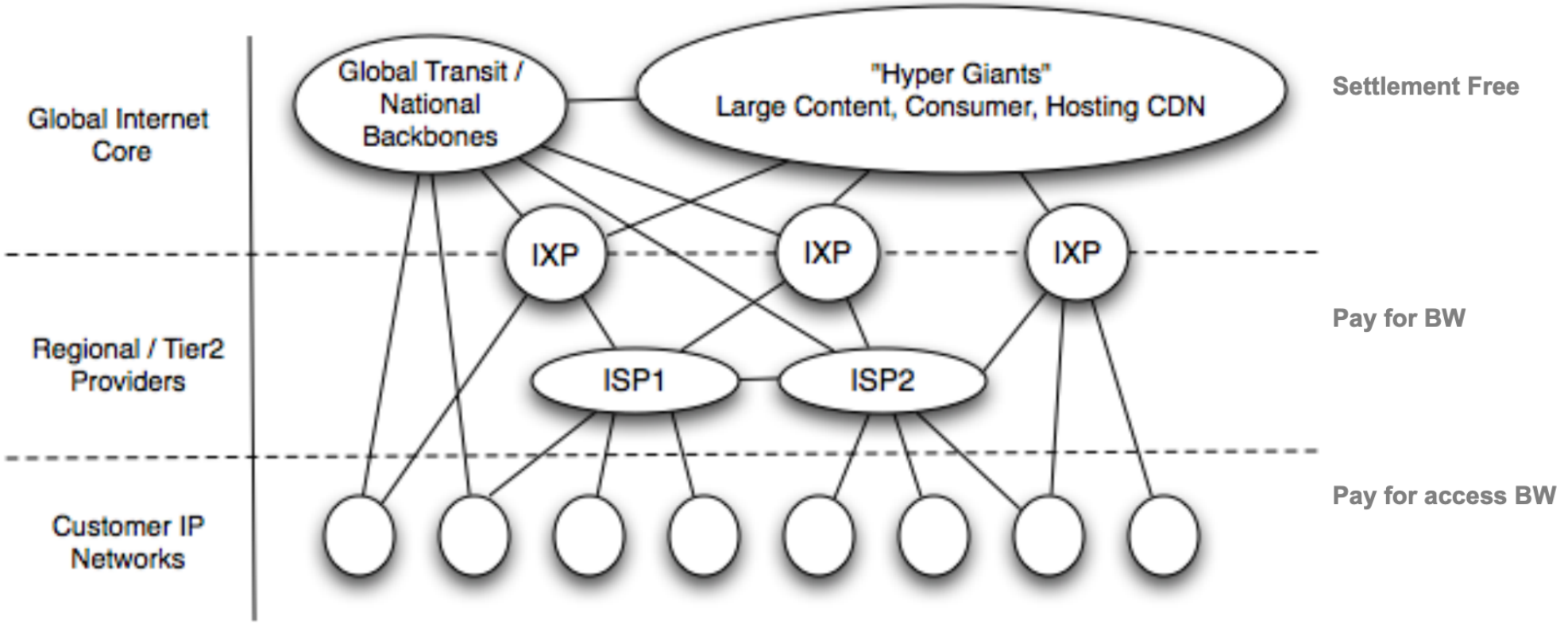
- Build their own backbones (ISP)
- Peer widely
- Cuts out tier-1 ISPs

Traditional structure



Source: Labovitz, SIGCOMM 2010

New structure



Source: Labovitz, SIGCOMM 2010