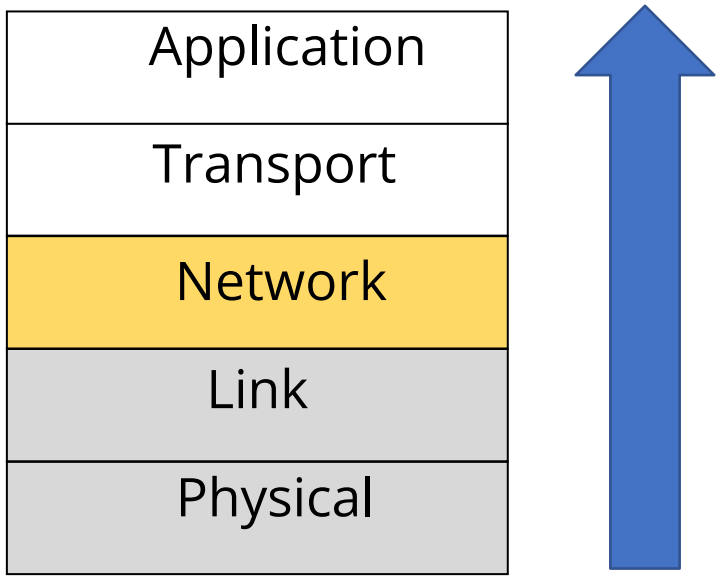
Where we are in the Course

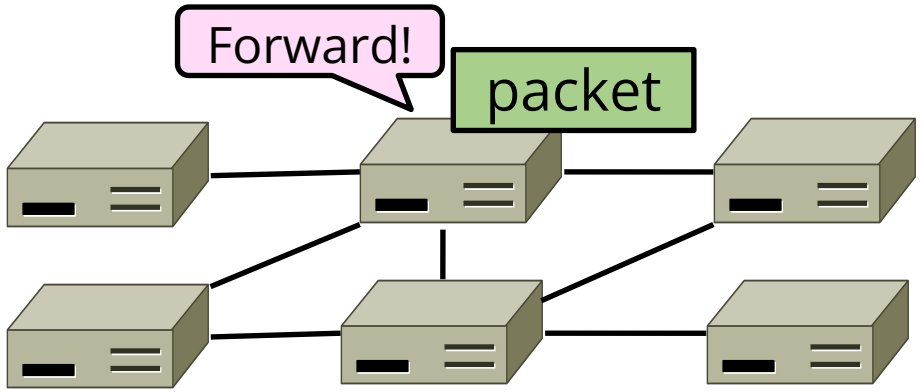
Today: Routing! Sending traffic across the network of networks

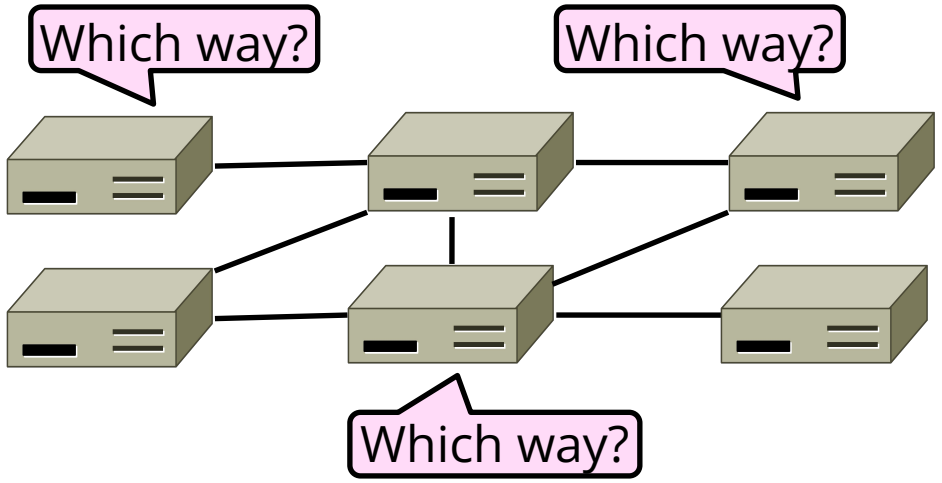


Network Layer (Routing)

Recall: Routing versus Forwarding

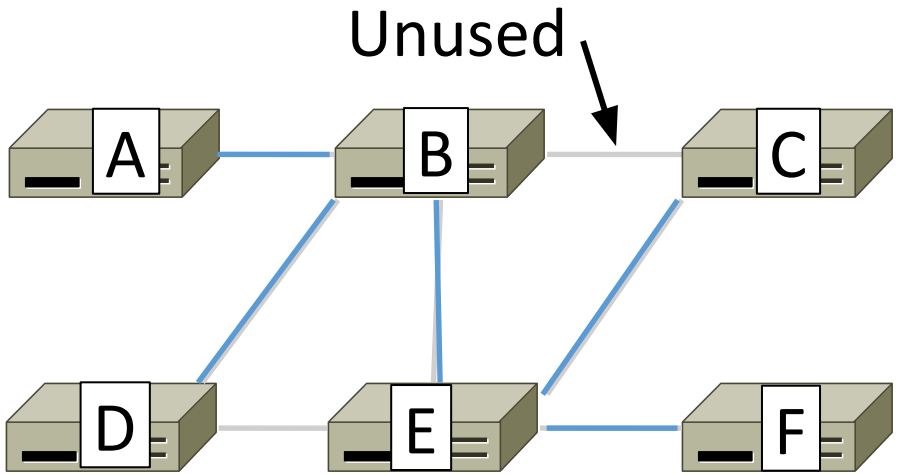
- Forwarding is the process of sending a packet on its way
- Routing is the process of deciding in which direction to send traffic

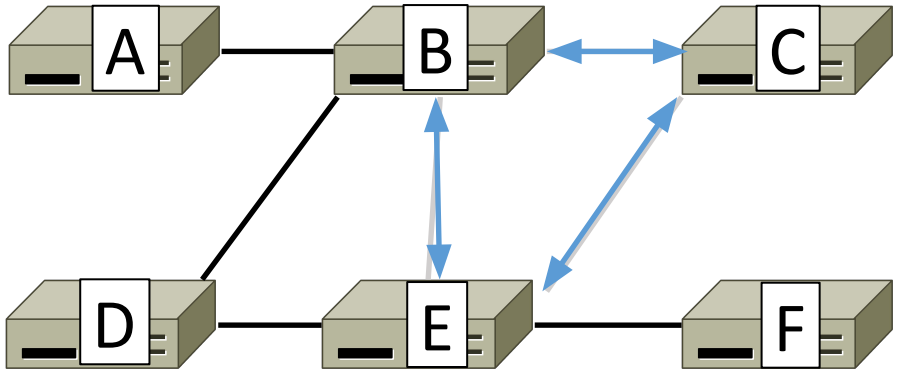




Improving on the Spanning Tree

- Spanning tree provides basic connectivity
  - e.g., some path B→C
- Routing uses all links to find "best" paths
  - e.g., use BC, BE, and CE





Perspective on Bandwidth Allocation

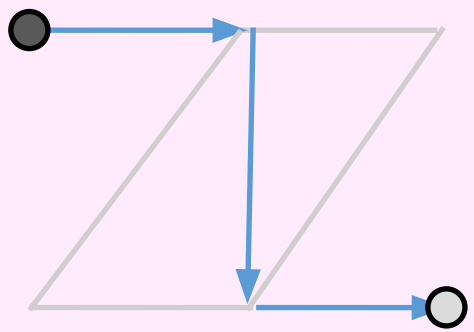
- Routing allocates network bandwidth adapting to failures; other mechanisms used at other timescales

Mechanism	Timescale / Adaptation
Load-sensitive routing	Seconds / Traffic hotspots
Routing	Minutes / Equipment failures
Traffic Engineering	Hours / Network load
Provisioning	Months / Network customers

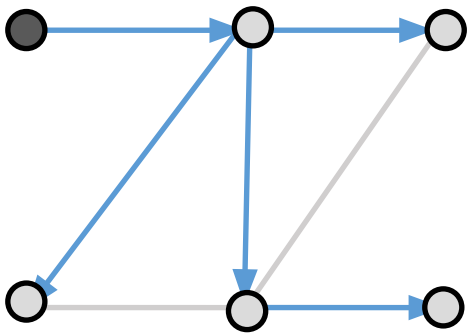
Delivery Models

- Different routing used for different delivery models

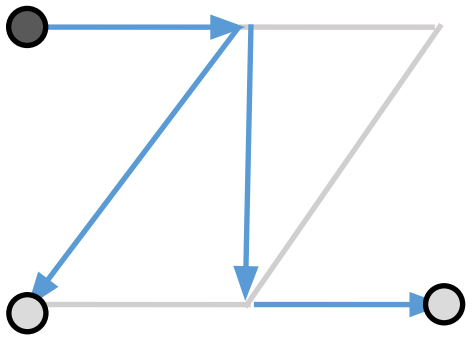
Unicast



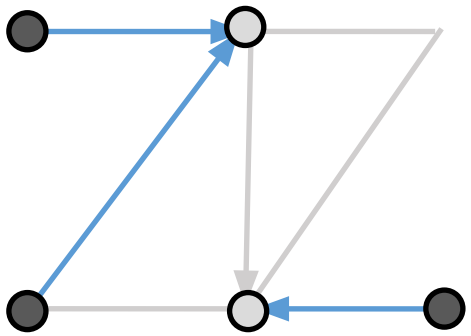
Broadcast



Multicast



Anycast



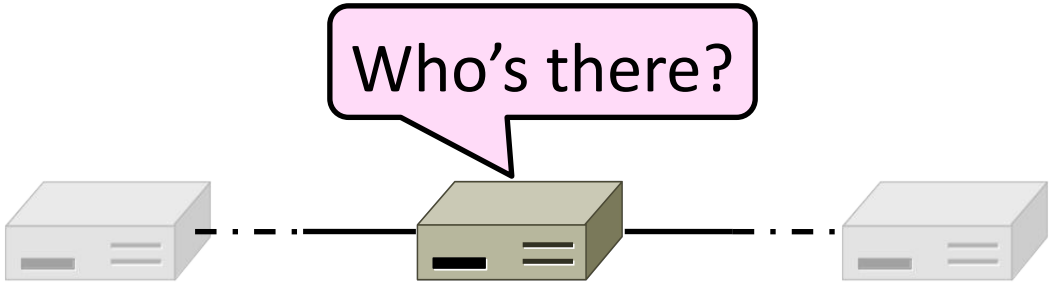
Goals of Routing Algorithms

- We want several properties of any routing scheme:

Property	Meaning
Correctness	Finds paths that work
Efficient paths	Uses network bandwidth well
Fair paths	Doesn't starve any nodes
Fast convergence	Recovers quickly after changes
Scalability	Works well as network grows large

Rules of Classic Routing Algorithms

- Decentralized, distributed setting
  - All nodes are alike; no controller
  - Nodes only know what they learn by exchanging messages with neighbors
  - Nodes operate concurrently
  - May be node/link/message failures

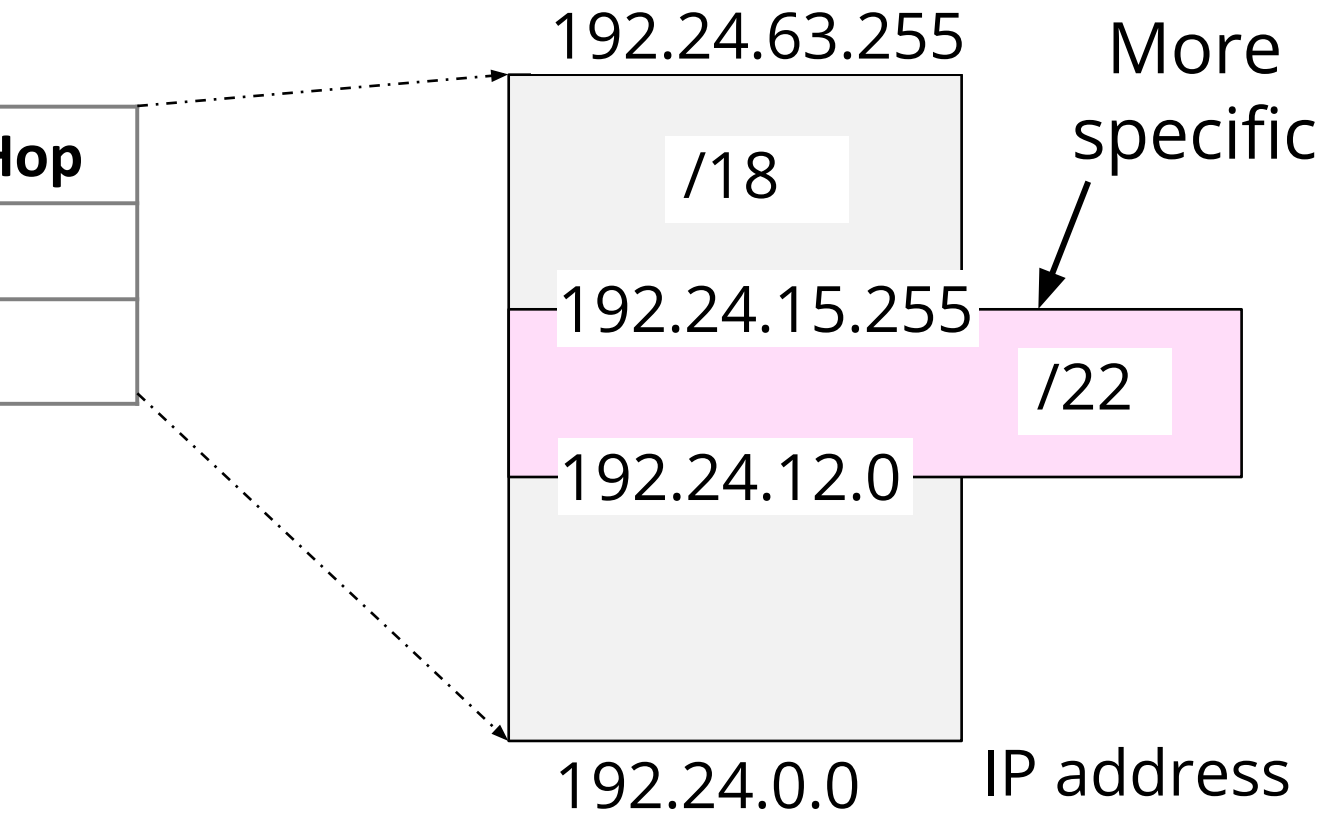


Recap: Classless Inter-Domain Routing (CIDR)

- In the Internet:
  - Hosts on same network have IPs in the same IP prefix
  - Hosts send off-network traffic to nearest router to handle
  - Routers discover the routes to use
  - Routers use longest prefix matching to send packets to the right next hop

Longest Matching Prefix

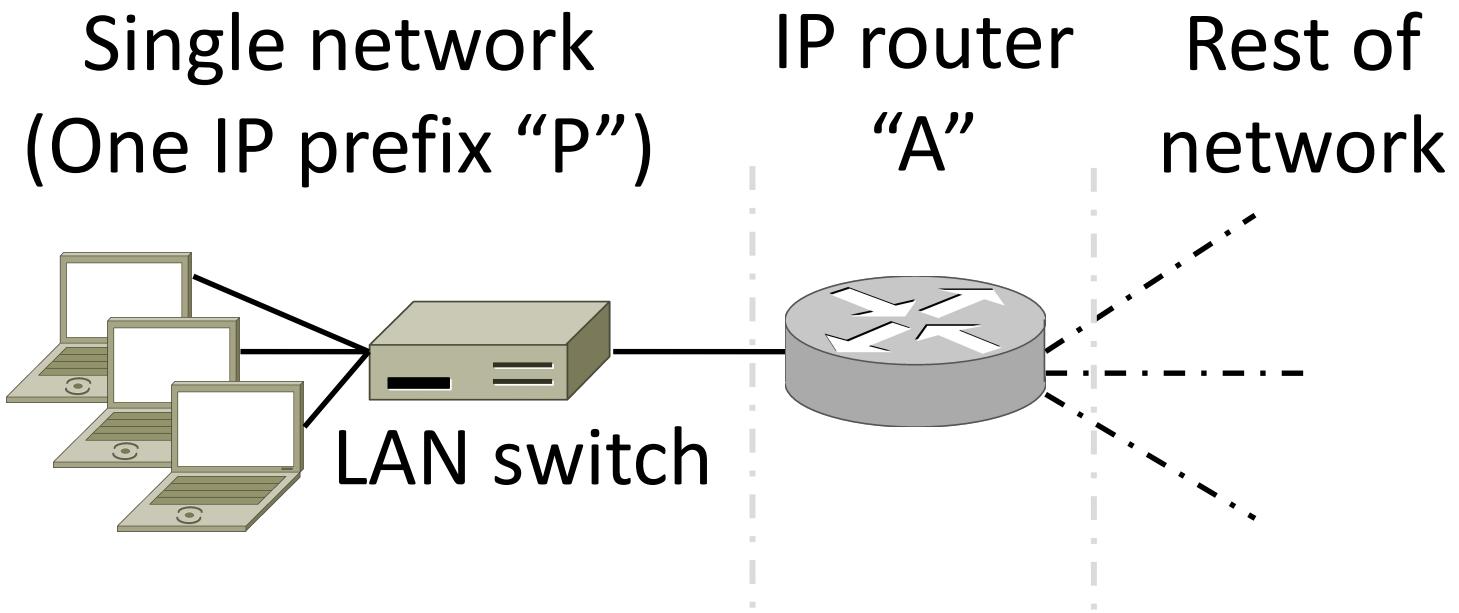
Prefix	Next Hop
192.24.0.0/18	D
192.24.12.0/22	B



192.24.6.0 → D
192.24.14.32 → B
192.24.54.0 → D

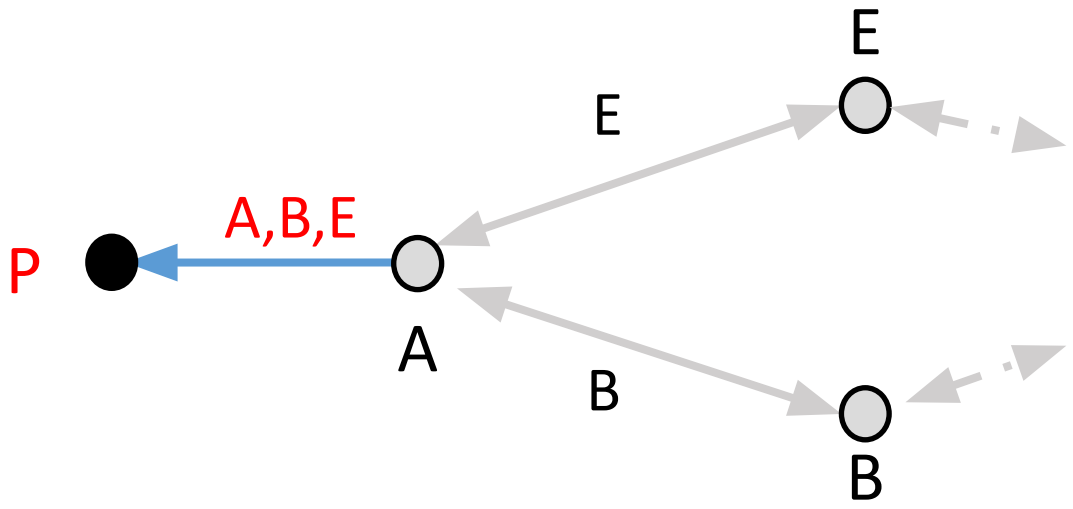
Host/Router Combination

- Hosts attach to routers as IP prefixes
  - Router needs table to reach all hosts



Network Topology for Routing

- Send out routes for hosts you have paths to
  - "Advertise" the routes
  - And the routes you've received



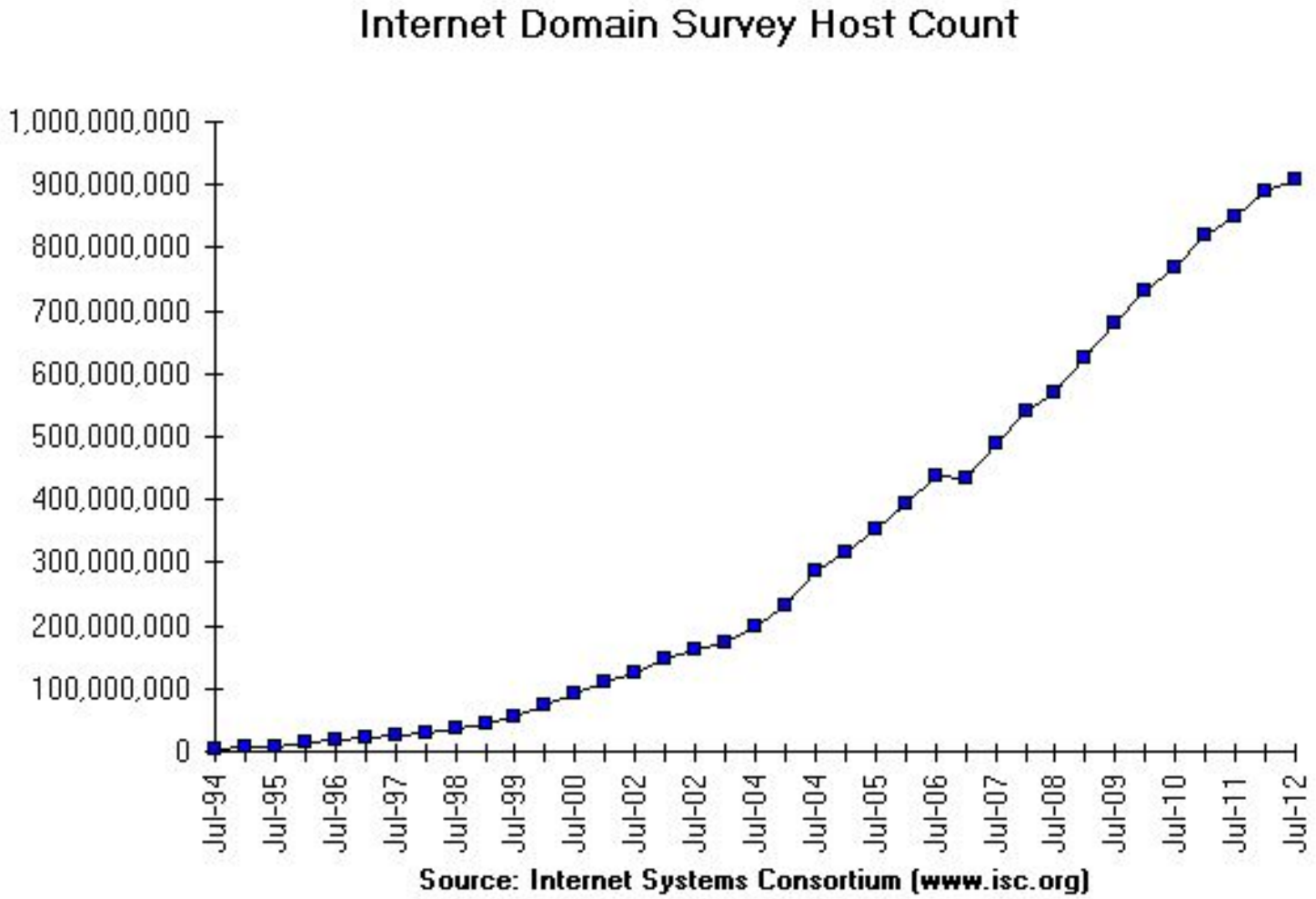
Network Topology for Routing (2)

- Routing now works!
  - Routers advertise IP prefixes for hosts they have routes to
    - Compile these together including own IP
  - Lets all routers find a path to hosts
  - Hosts find by sending to their router

Hierarchical Routing

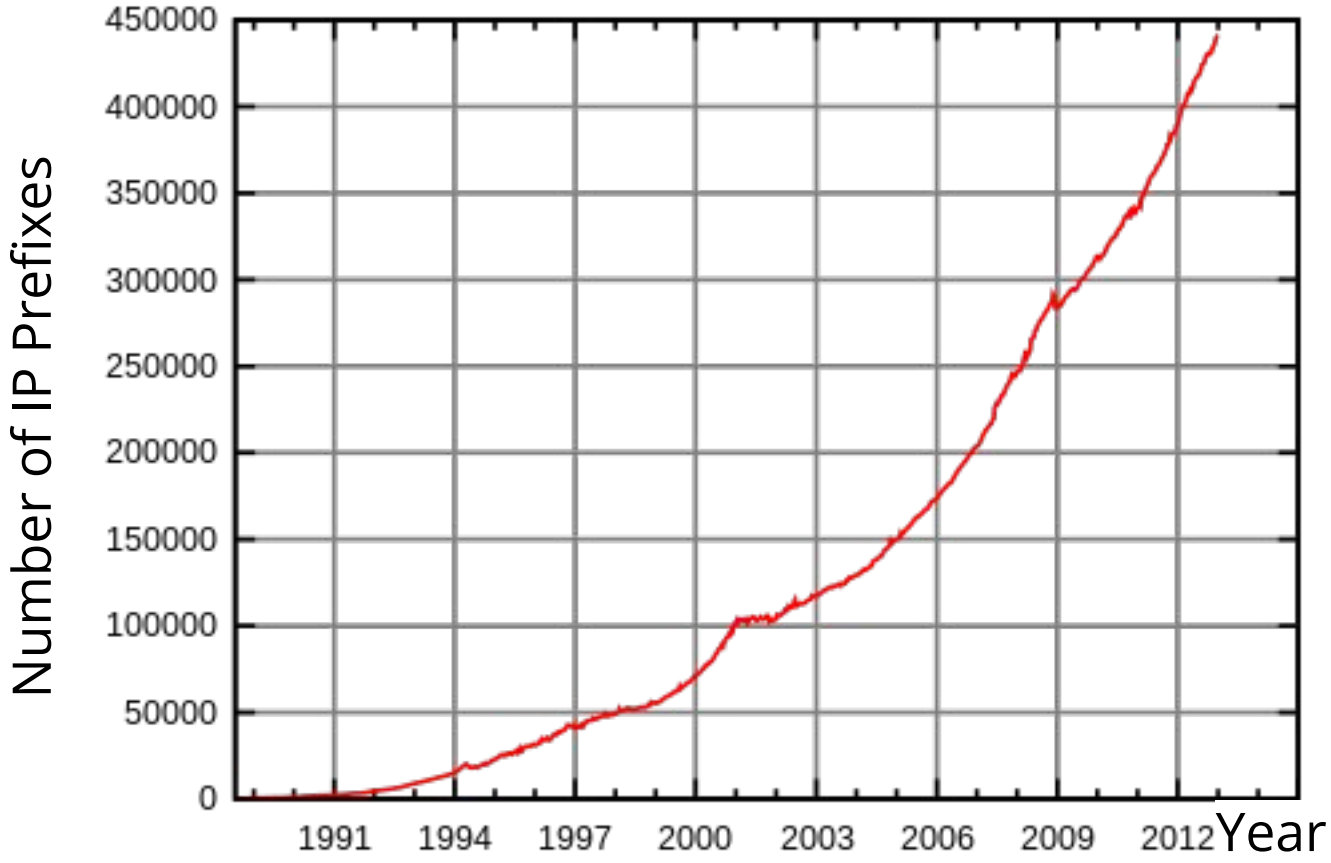
Internet Growth

- Billions of Internet hosts and growing…



Internet Routing Growth

- Internet growth translates into routing table growth
  - Even using prefixes…



Source: By Mro (Own work), CC-BY-SA-3.0 , via Wikimedia Commons

Impact of Routing Growth

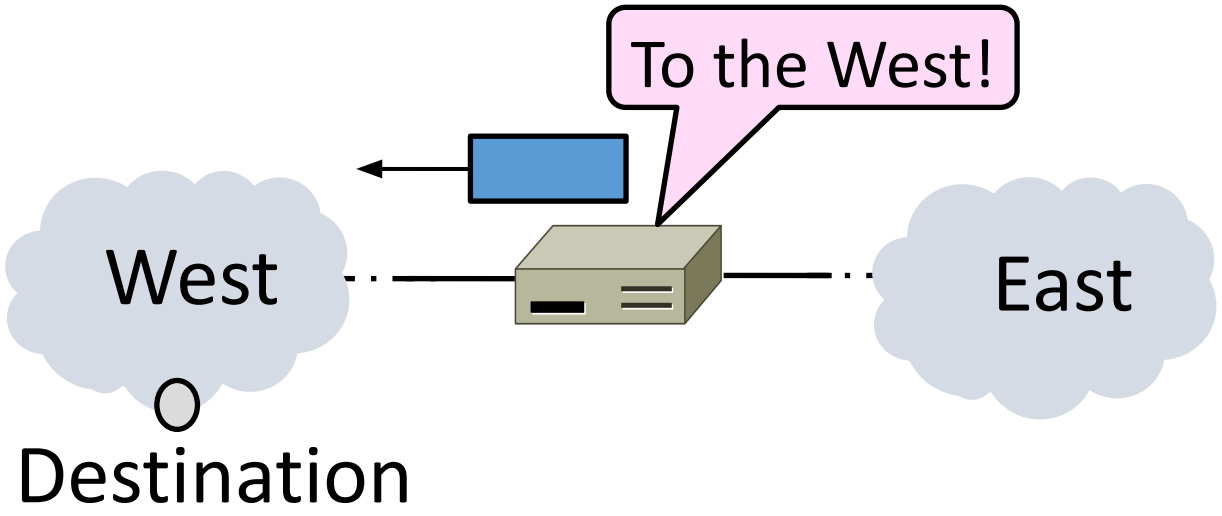
1. Forwarding tables grow
   - Larger router memories, may increase lookup time
2. Routing messages grow
   - Need to keeps all nodes informed of larger topology
3. Routing computation grows
   - Shortest path calculations grow faster than the network

Techniques to Scale Routing

- First: Network hierarchy
  - Route to network regions
- Next: IP prefix aggregation
  - Combine, and split, prefixes

Idea

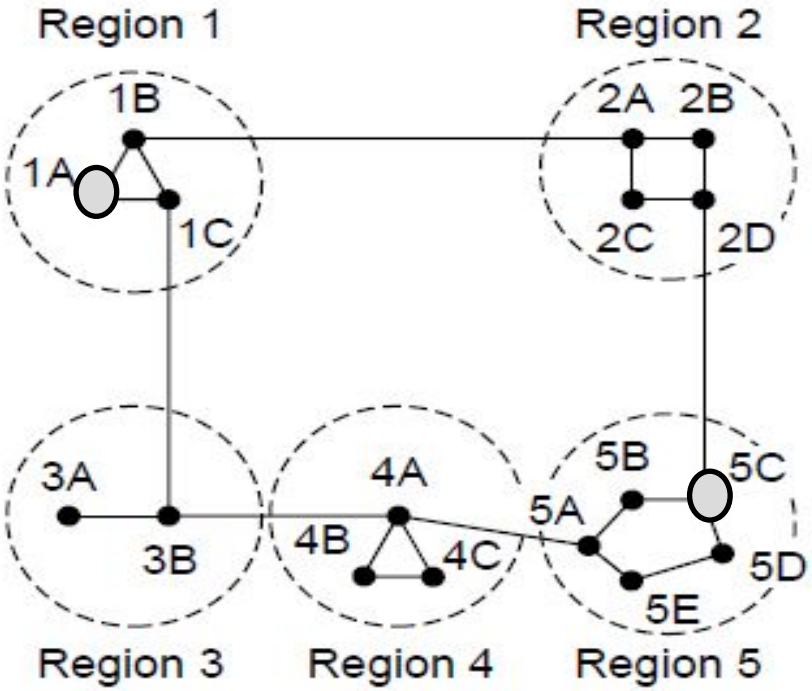
- Scale routing using hierarchy with regions
  - Route to regions, not individual nodes



Hierarchical Routing

- Introduce a larger routing unit
  - IP prefix (many hosts) from one gateway (host)
  - Region, e.g., ISP network
- Route first to the region, then to the IP prefix within the region
  - Hide details within a region from outside of the region

Hierarchical Routing (2)



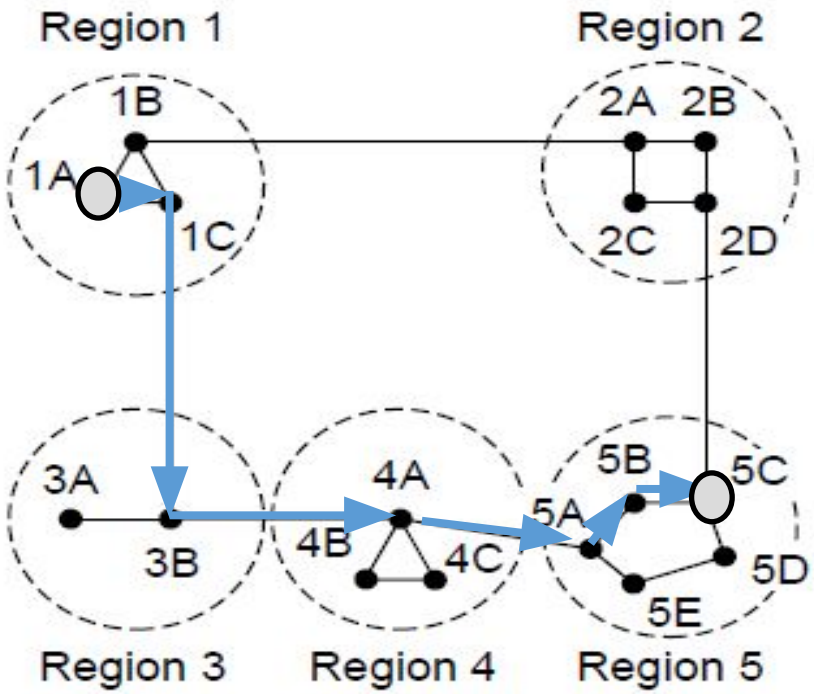
Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Hierarchical Routing (3)



Full table for 1A

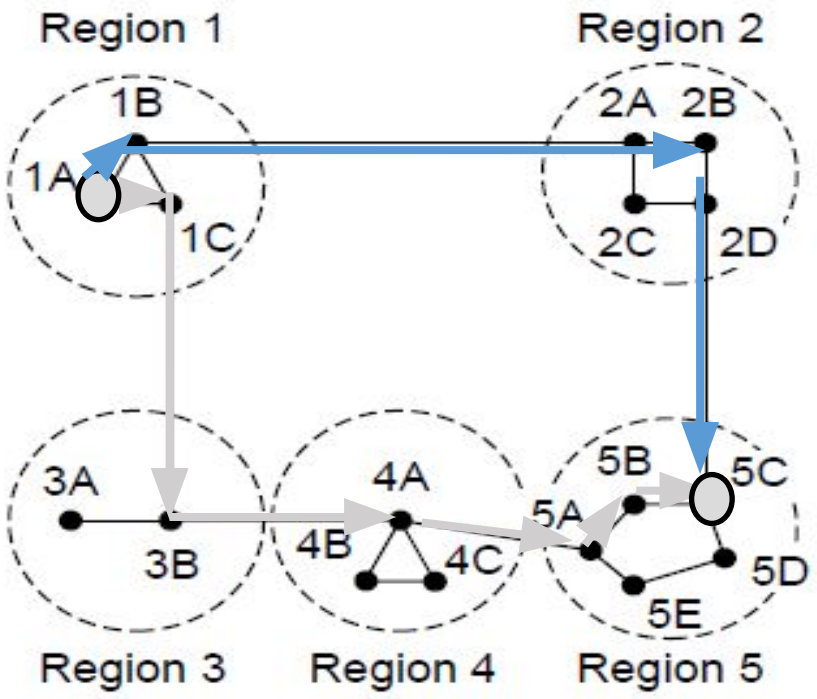
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Hierarchical Routing (4)

- Penalty is possibly longer paths



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

1C is best route to region 5, except for destination 5C

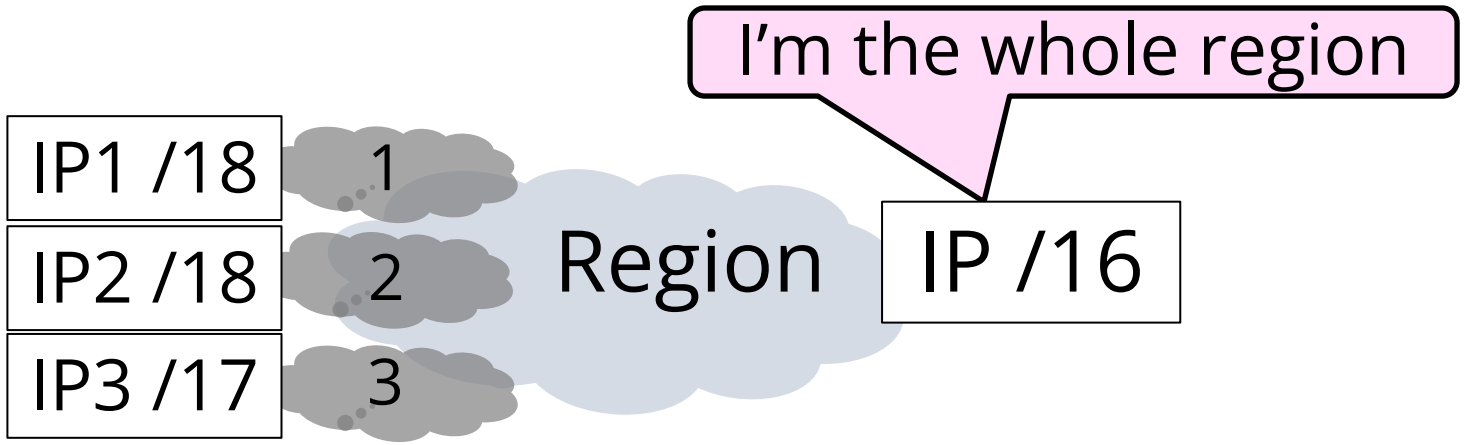
Observations

- Outside a region, nodes have one route to all hosts within the region
  - This gives savings in table size, messages and computation
- However, each node may have a different route to an outside region
  - Routing decisions are still made by individual nodes; there is no single decision made by a region

IP Prefix Aggregation and Subnets

Idea

- Scale routing by adjusting the size of IP prefixes
  - Split (subnets) **and** join (aggregation)



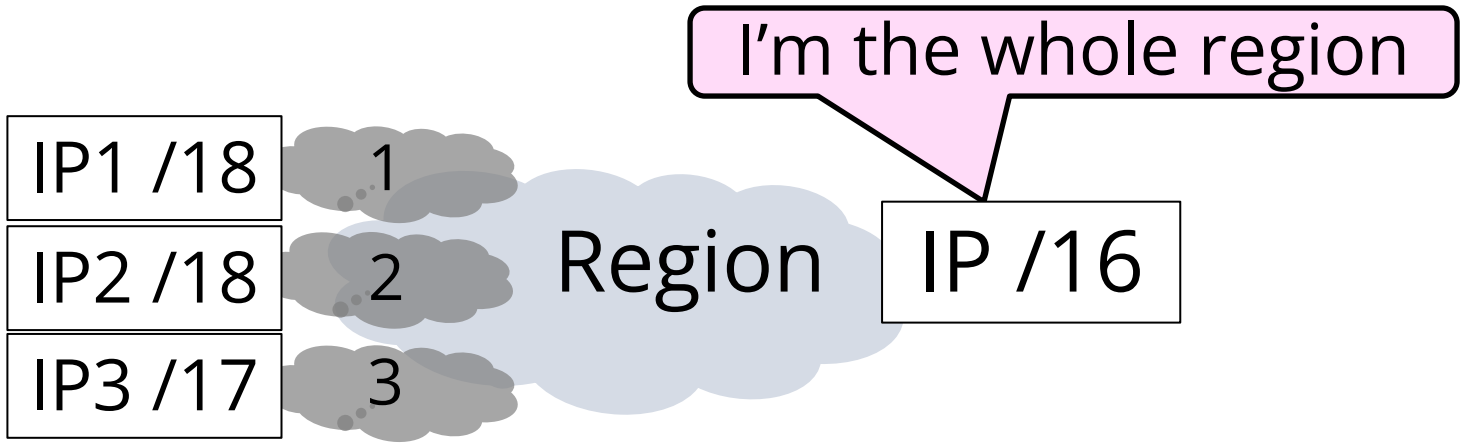
Recall

- IP addresses are allocated in blocks called IP prefixes, e.g., `18.31.0.0/16`
  - Hosts on one network in same prefix
- "/N" prefix has the first N bits fixed
  - 2^{32} - N addresses in IPv4
  - 2^{128} - N addresses in IPv6
- Routers keep track of prefix lengths
  - Use it as part of longest prefix matching

Routers can change prefix lengths without affecting hosts

Prefixes and Hierarchy

- IP prefixes help to scale routing, but can go further
  - Use a less specific (larger) IP prefix as a name for a region



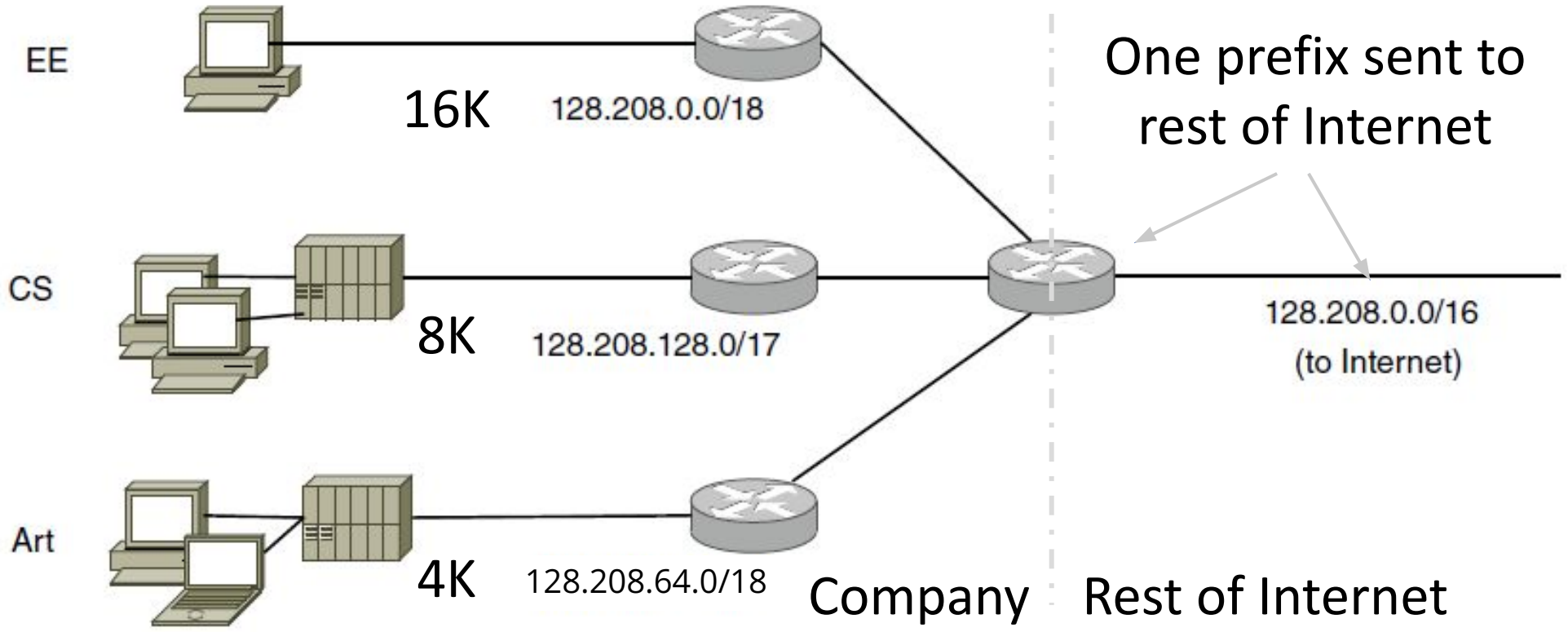
Subnets and Aggregation

- Two use cases for adjusting the size of IP prefixes; both reduce routing table

1. Subnets
   - Internally split one large prefix into multiple smaller ones
2. Aggregation
   - Join multiple smaller prefixes into one large prefix

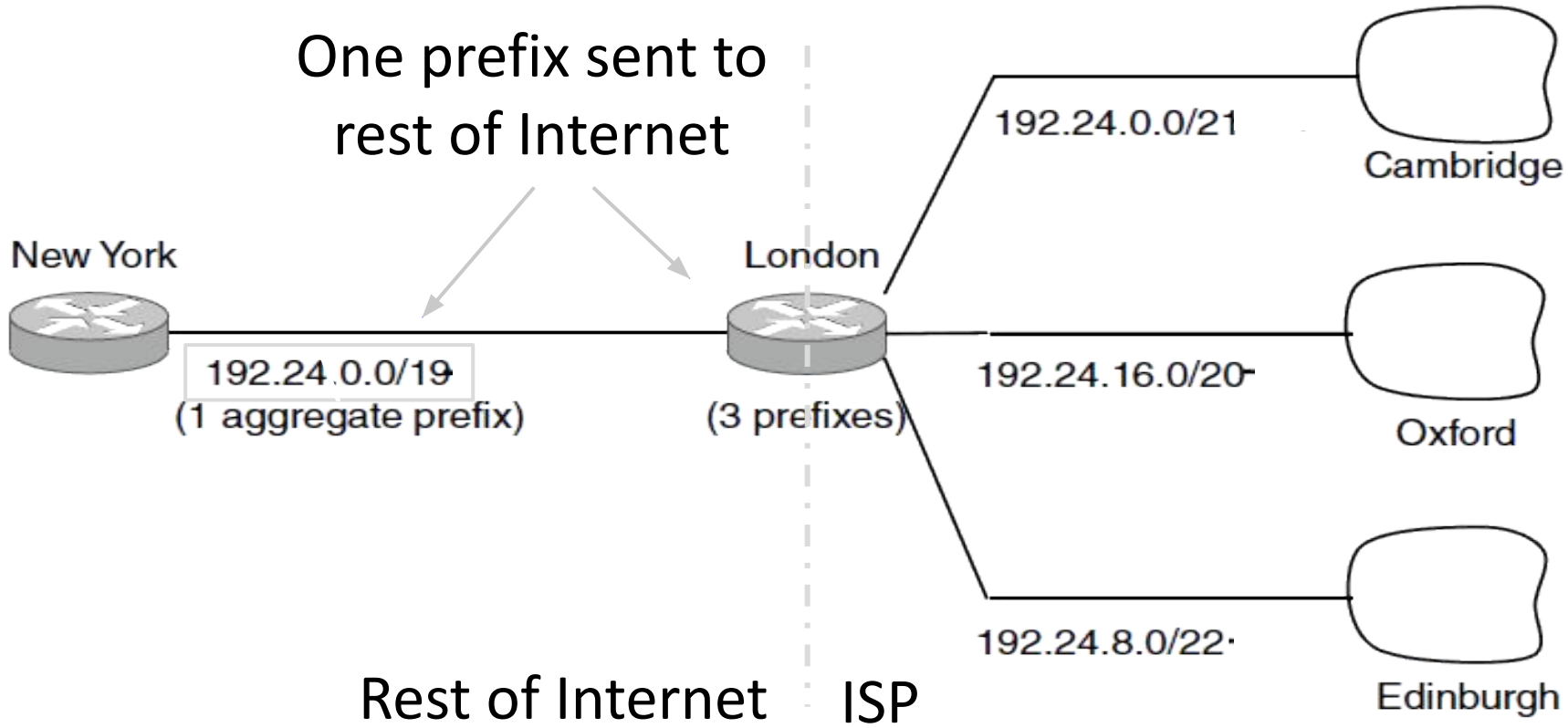
Subnets

- Internally split up one IP prefix



Aggregation

- Externally join multiple separate IP prefixes



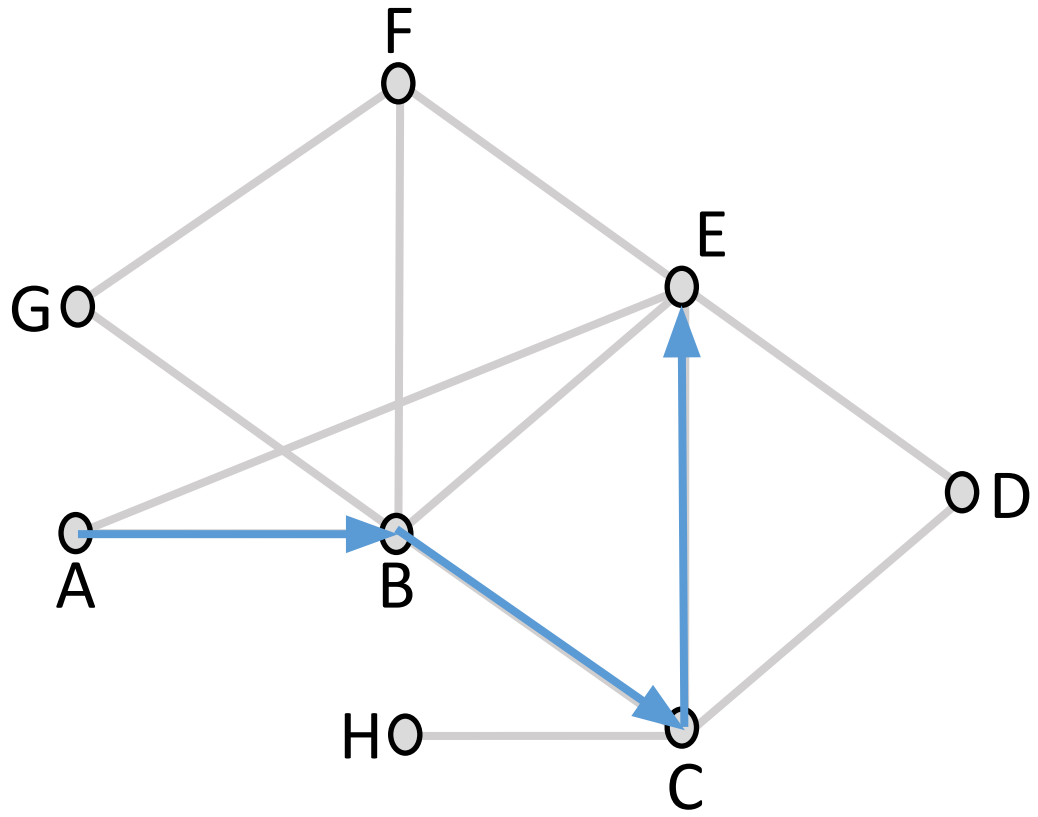
Routing Process

1. Ship these prefixes or regions around to nearby routers
2. Receive multiple prefixes and the paths of how you got them
3. Build a global routing table…

Best Path Routing

What are “Best” paths anyhow?

- Many possibilities:
  - Latency, avoid circuitous paths
  - Bandwidth, avoid slow links
  - Money, avoid expensive links
  - Hops, to reduce switching
- But only consider topology
  - Ignore workload, e.g., hotspots



Shortest Paths

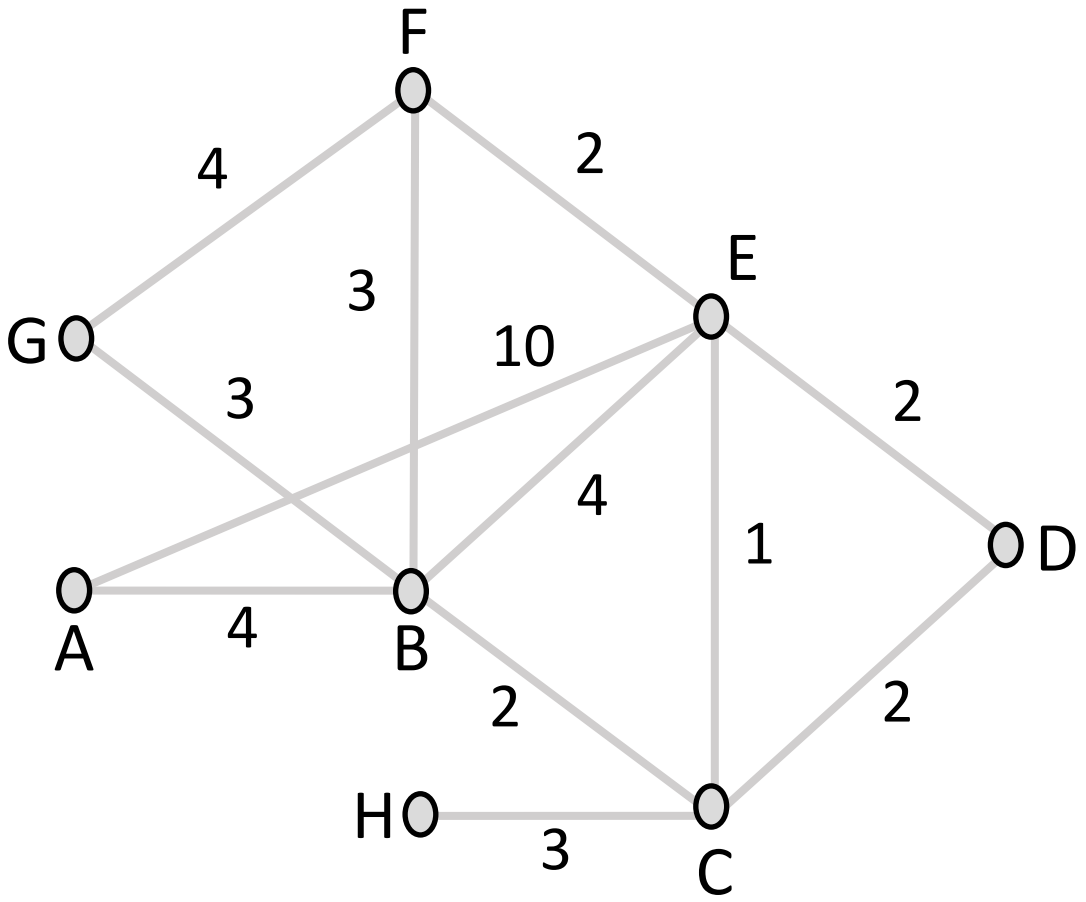
We'll approximate "best" by a cost function that captures the factors

- Often call lowest "shortest"

2. Assign each link a cost (distance)
3. Define best path between each pair of nodes as the path that has the lowest total cost (or is shortest)
4. Pick randomly to any break ties

Shortest Paths (2)

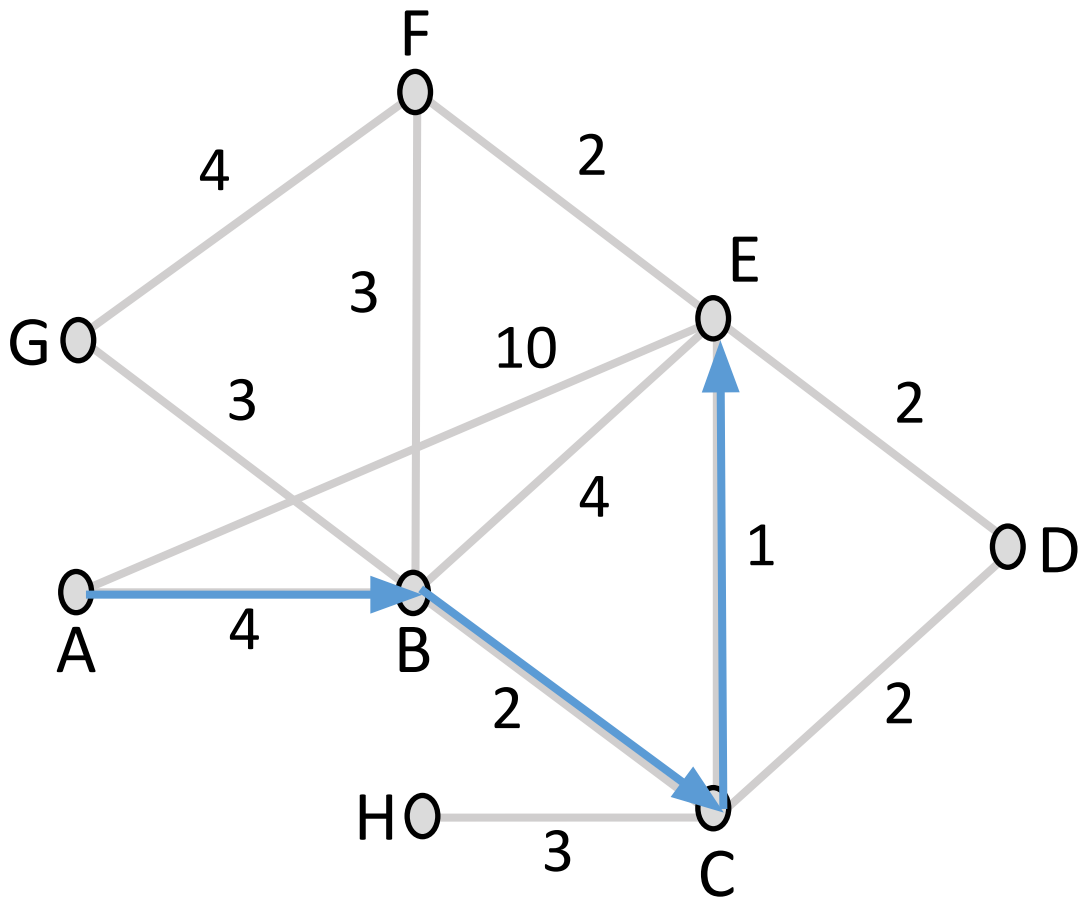
- Find the shortest path A ⟶ E
- All links are bidirectional, with equal costs in each direction
  - Can extend model to unequal costs if needed



Shortest Paths (3)

- ABCE is a shortest path
- dist(ABCE) = 4 + 2 + 1 = 7

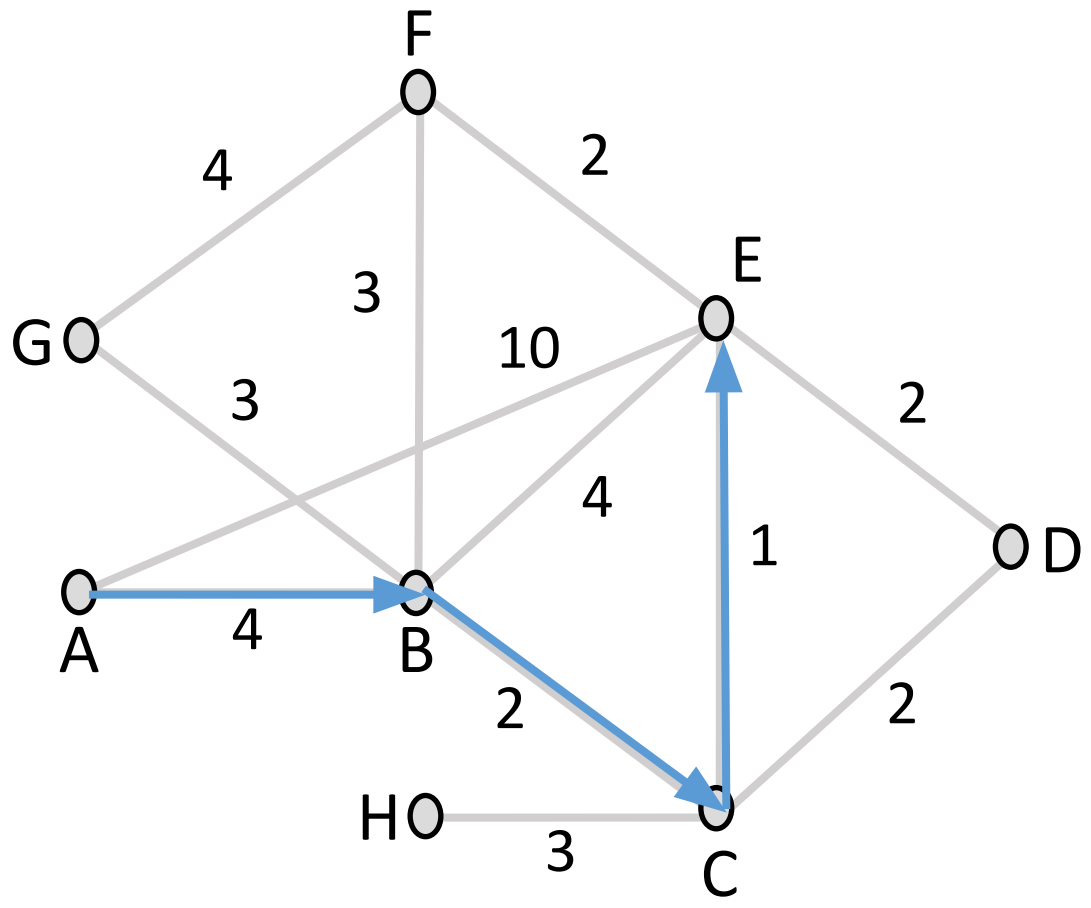
- This is less than:
  - dist(ABE) = 8
  - dist(ABFE) = 9
  - dist(AE) = 10
  - dist(ABCDE) = 10



Shortest Paths (4)

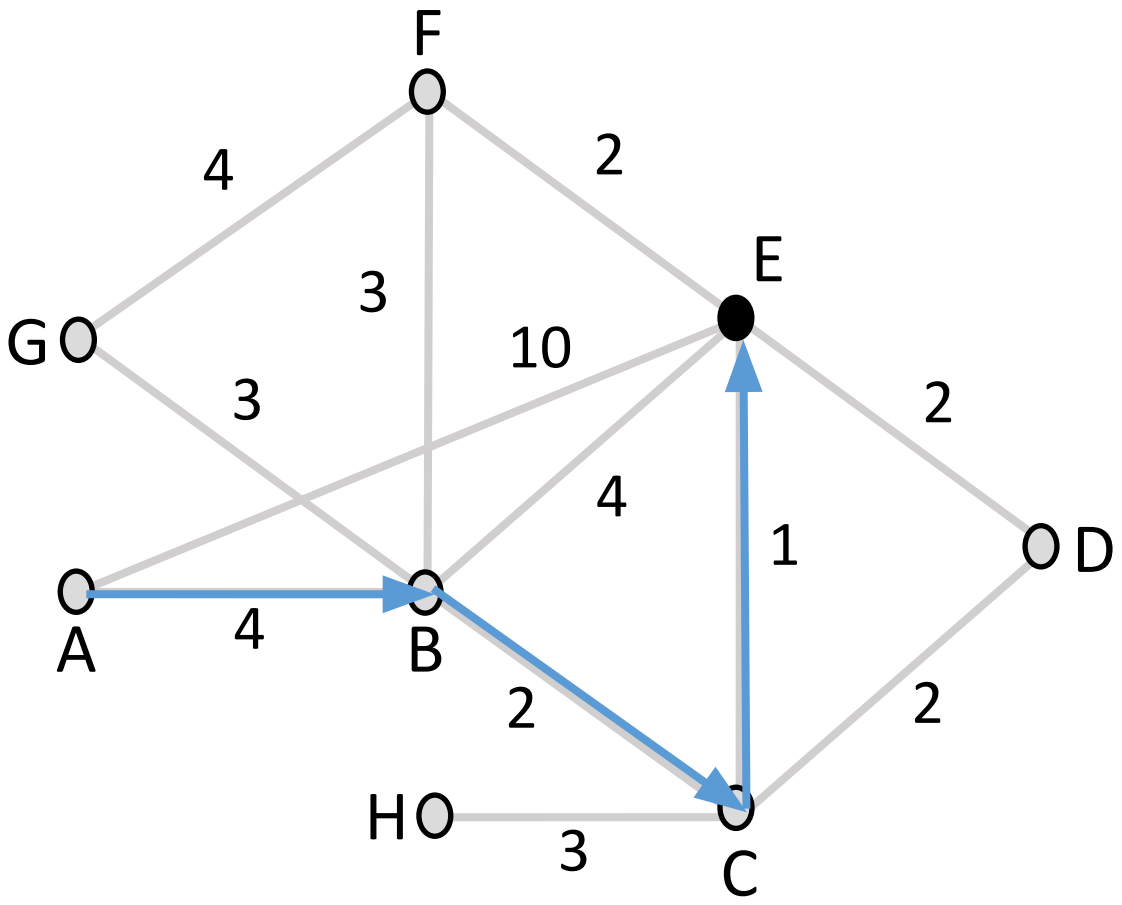
- Optimality property:
  - Subpaths of shortest paths are also shortest paths
- ABCE is a shortest path

  → So are ABC, AB, BCE, BC, CE



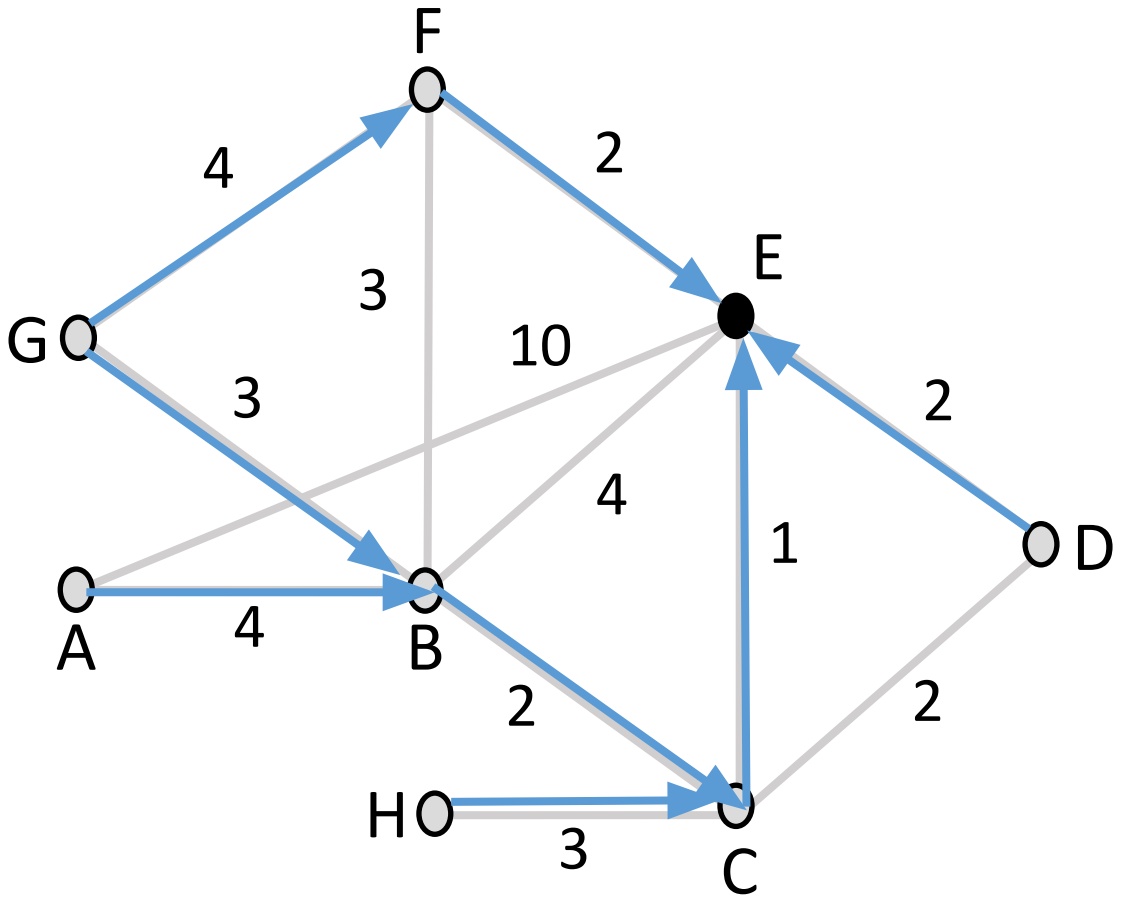
Sink Trees

- Sink tree for a destination is the union of all shortest paths towards the destination
  - Similarly source tree
- Find the sink tree for E



Sink Trees (2)

- Implications:
  - Only need to use destination to follow shortest paths
  - Each node only need to send to the next hop
- Forwarding table at a node
  - Lists next hop for each destination
  - Routing table may know more



So how do we actually build it???

- Will talk about two high-level approaches used for medium-scale *intradomain* routing ("Interior Gateway Protocols") within one (potentially large) organization
- Next will discuss *interdomain* routing between organizations… the global Internet

Distance Vector Routing

Intradomain approach 1

Distance Vector Routing

- Simple, early routing approach
  - Used in ARPANET, and RIP
- One of two main approaches to routing
  - Distributed version of Bellman-Ford
  - Works, but very slow convergence after some failures
- Link-state algorithms are now typically used in practice
  - More involved, better behavior

Distance Vector Setting

Each node computes its forwarding table in a distributed setting:

1. Nodes know only the cost to their neighbors; not topology
2. Nodes can talk only to their neighbors using messages
3. All nodes run the same algorithm concurrently
4. Nodes and links may fail, messages may be lost

Distance Vector Algorithm

Each node maintains a vector of distances (and next hops) to all destinations

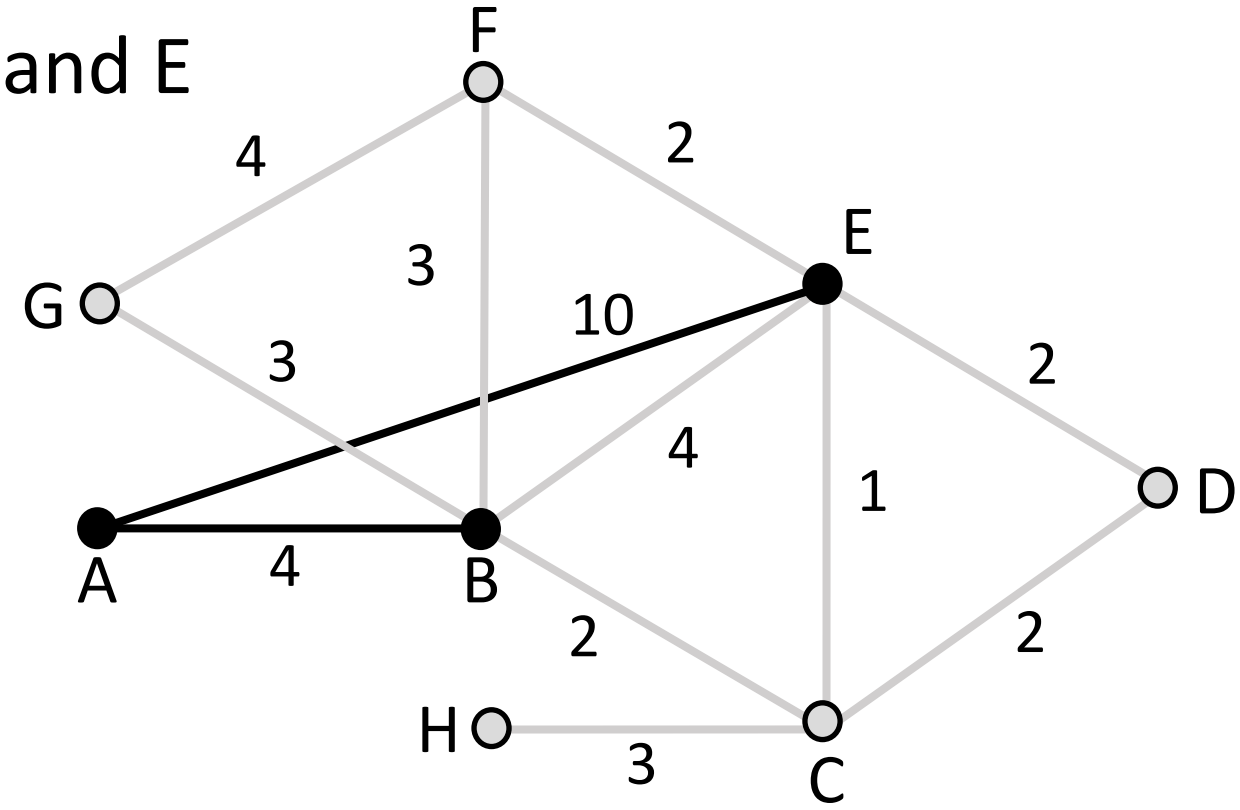
1. Initialize vector with 0 (zero) cost to self, ∞ (infinity) to other destinations
2. Periodically send vector to neighbors
3. Update vector for each destination by selecting the shortest distance heard, after adding cost of neighbor link
4. Use the best neighbor for forwarding

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's	E's
A	∞	∞
B	0	∞
C	∞	∞
D	∞	∞
E	∞	0
F	∞	∞
G	∞	∞
H	∞	∞

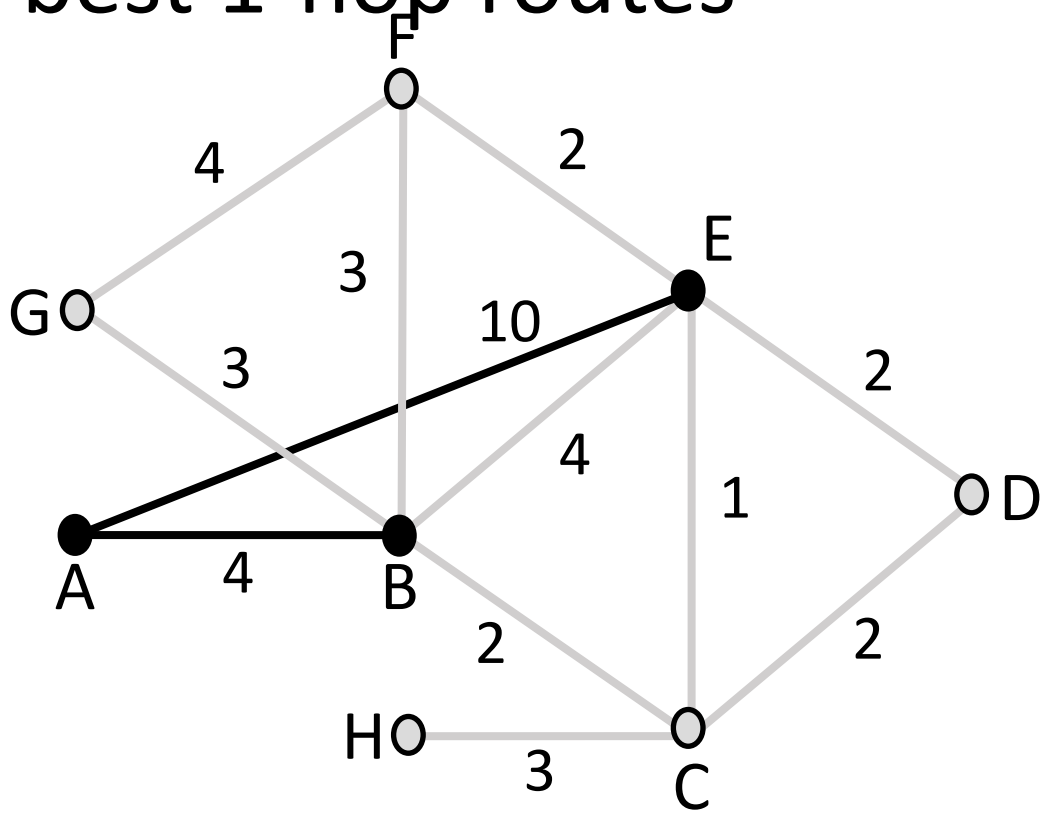
→

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

→

Cost	Next
0	--
4	B
∞	--
∞	--
10	E
∞	--
∞	--
∞	--

Learned better route



Distance Vector (4)

- Second exchange; learn best 2-hop routes

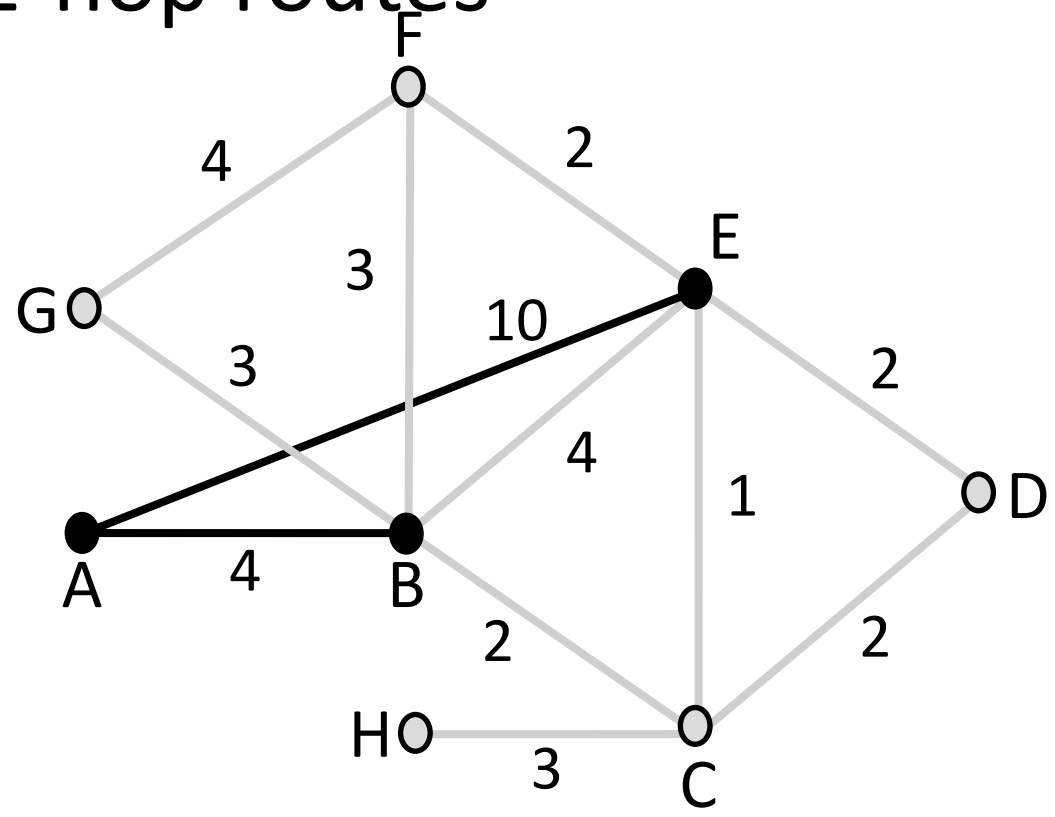
To	B's	E'
A	4	10
B	0	4
C	2	1
D	∞	2
E	4	0
F	3	2
G	3	∞
H	∞	∞

→

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

→

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



Distance Vector (4)

- Third exchange; learn best 3-hop routes

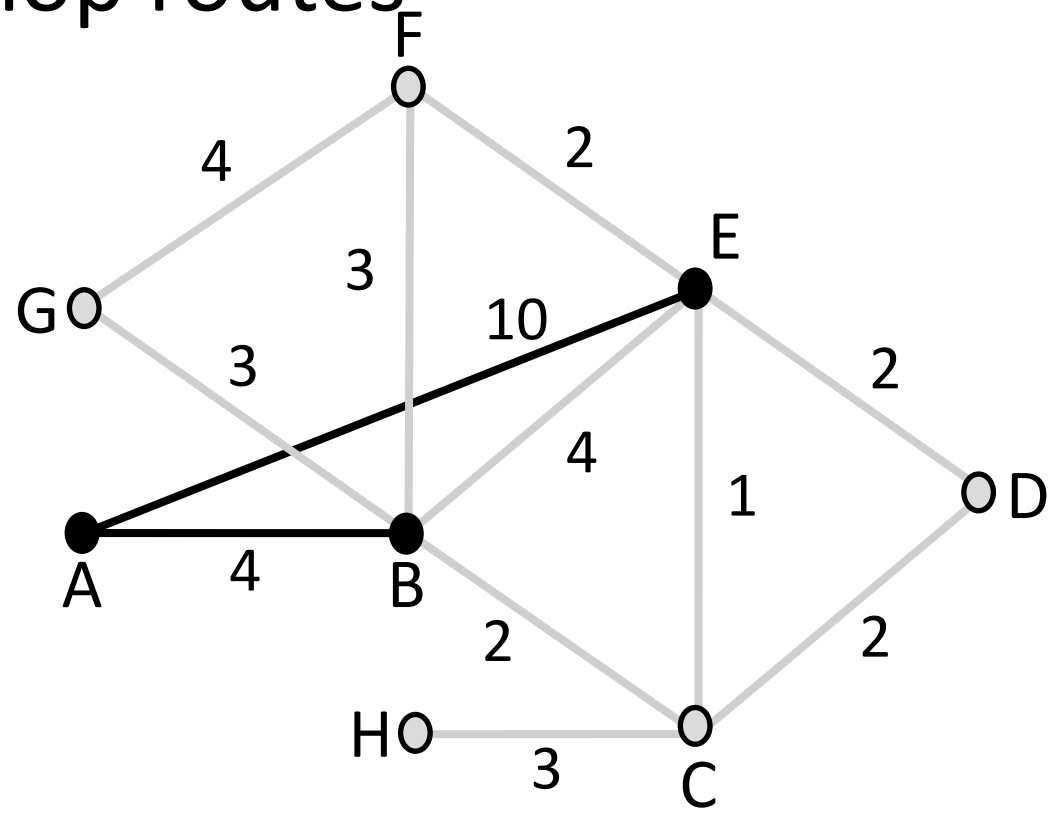
To	B's	E's
A	4	8
B	0	3
C	2	1
D	4	2
E	3	0
F	3	2
G	3	6
H	5	4

→

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

→

Cost	Next
0	--
4	B
6	B
8	B
7	B
7	B
7	B
9	B



Distance Vector (5)

- Subsequent exchanges; converged

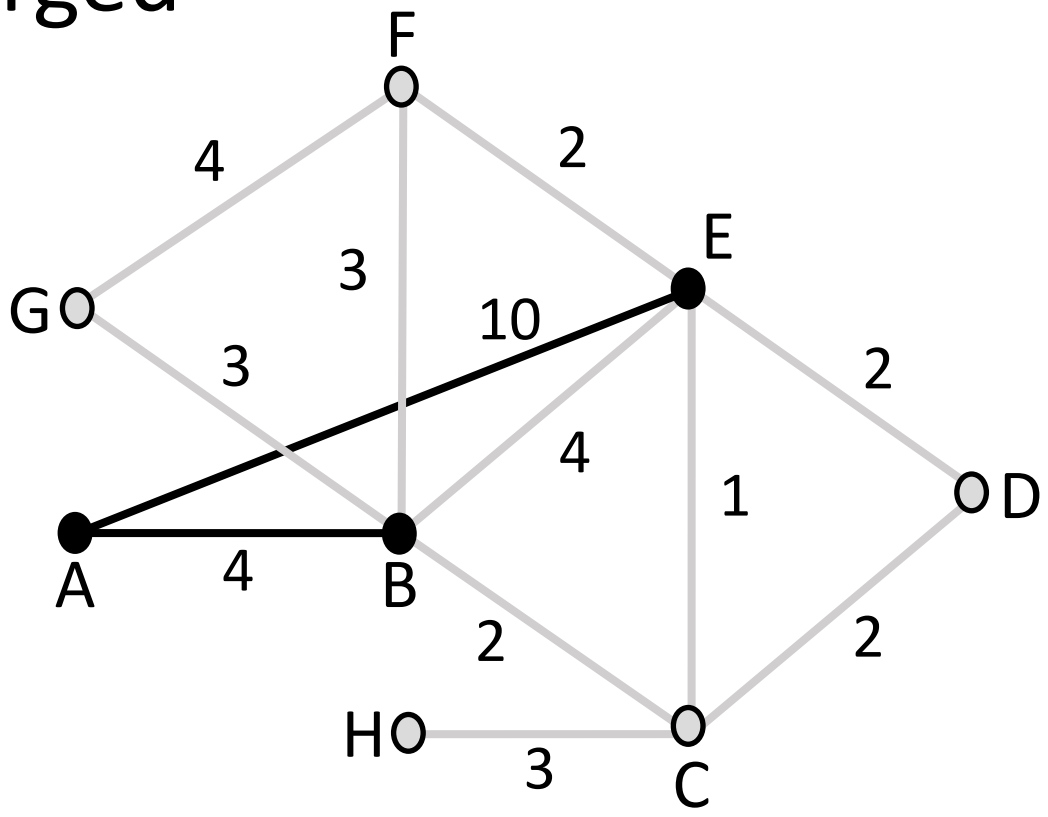
To	B's	E's
A	4	7
B	0	3
C	2	1
D	4	2
E	3	0
F	3	2
G	3	6
H	5	4

→

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

→

Cost	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

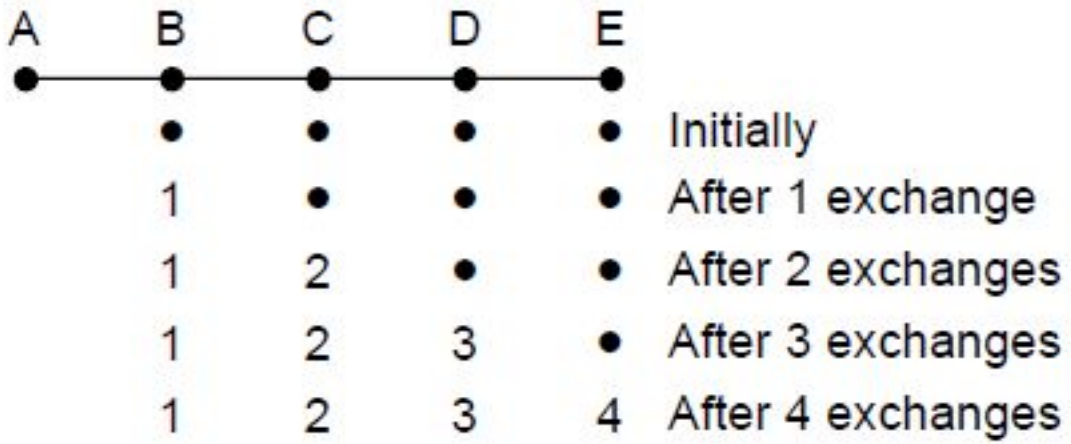
Distance Vector Dynamics

- Adding routes:
  - News travels one hop per exchange
- Removing routes:
  - When a node fails, no more exchanges, other nodes forget

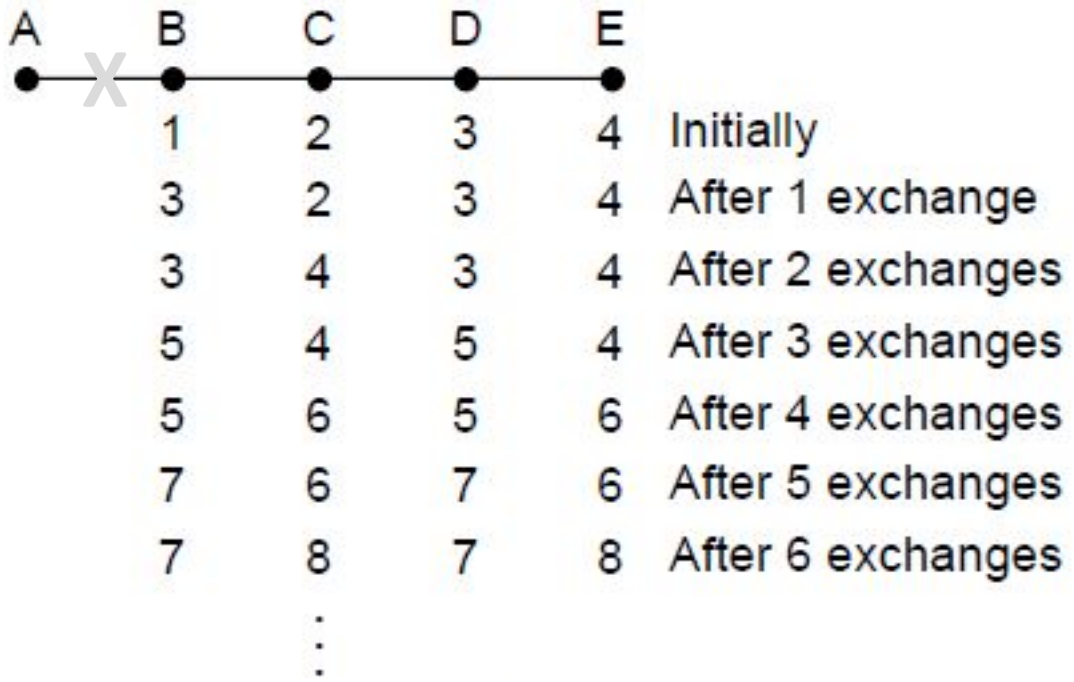
Problem(s)?

DV Dynamics (2)

- Good news travels quickly, bad news slowly



Desired convergence



"Count to infinity" scenario

DV Dynamics (3)

- Various heuristics to address
  - "Split horizon"
    - Don't send route back to where you learned it from.
  - Poison reverse
    - Send "infinity" when you notice a disconnect
- But none are very effective
  - Link state now favored in practice
  - Except when very resource-limited

RIP (Routing Information Protocol)

- DV protocol with hop count as metric
  - Infinity is 16 hops; limits network size
  - Includes split horizon, poison reverse
- Routers send vectors every 30 seconds
  - Runs on top of UDP
  - Time-out in 180 secs to detect failures
- RIPv1 specified in RFC1058 (1988)

Link-State Routing

Intradomain option 2

Link-State Routing

- Other broad class of routing algorithms
  - Trades more computation than distance vector for better dynamics
- Widely used in practice
  - Used in Internet/ARPANET from 1979
  - Modern networks use OSPF (L3) and IS-IS (L2)

Link-State Setting

Nodes compute their forwarding table in the same distributed setting as for distance vector:

1. Nodes know only the cost to their neighbors; not topology
2. Nodes can talk only to their neighbors using messages
3. All nodes run the same algorithm concurrently
4. Nodes/links may fail, messages may be lost

Link-State Algorithm

Proceeds in two phases:

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

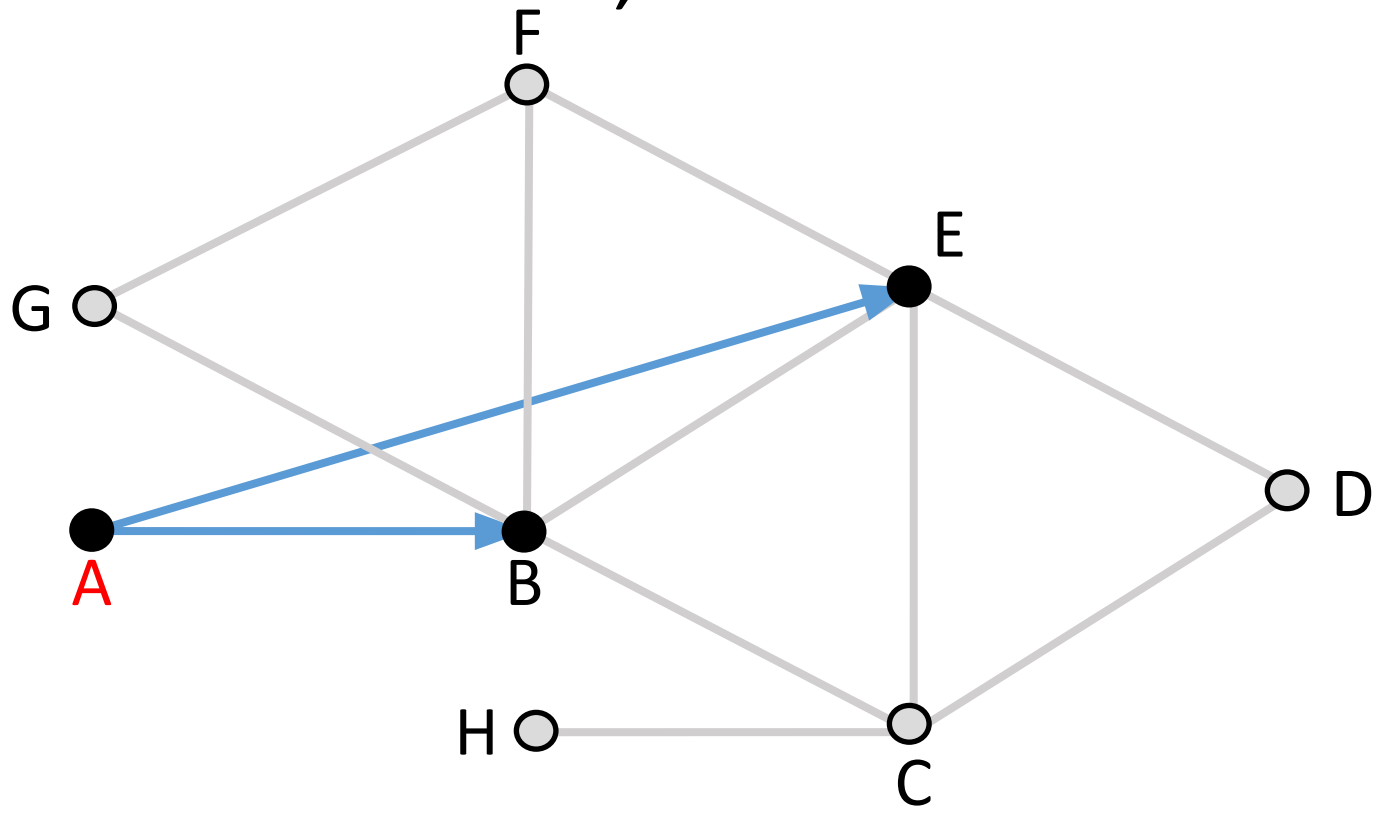
Link-State Part 1: Flood Routing

Flooding

- Rule used at each node:
  - Sends an incoming message on to all other neighbors
  - Remember the message so that it is only flooded once

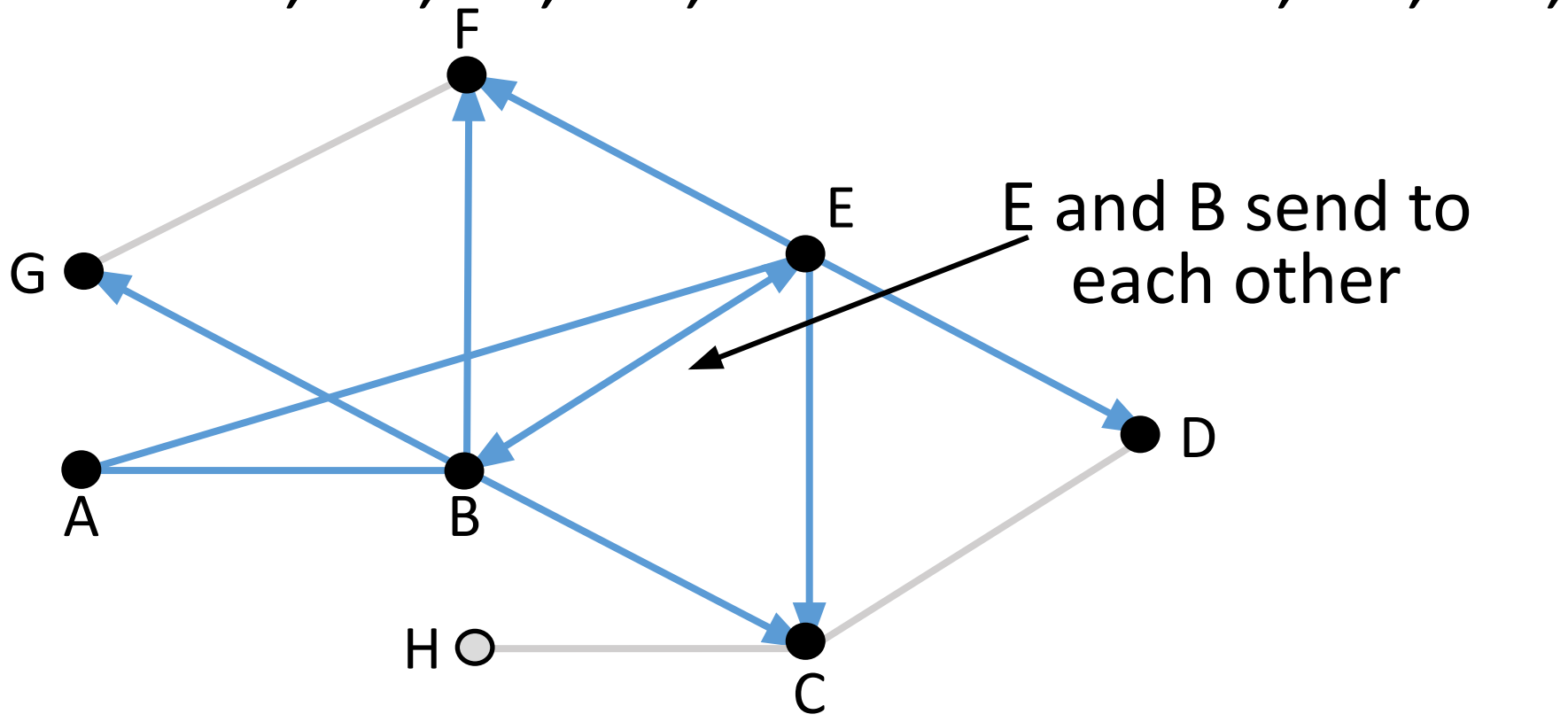
Flooding (2)

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



Flooding (3)

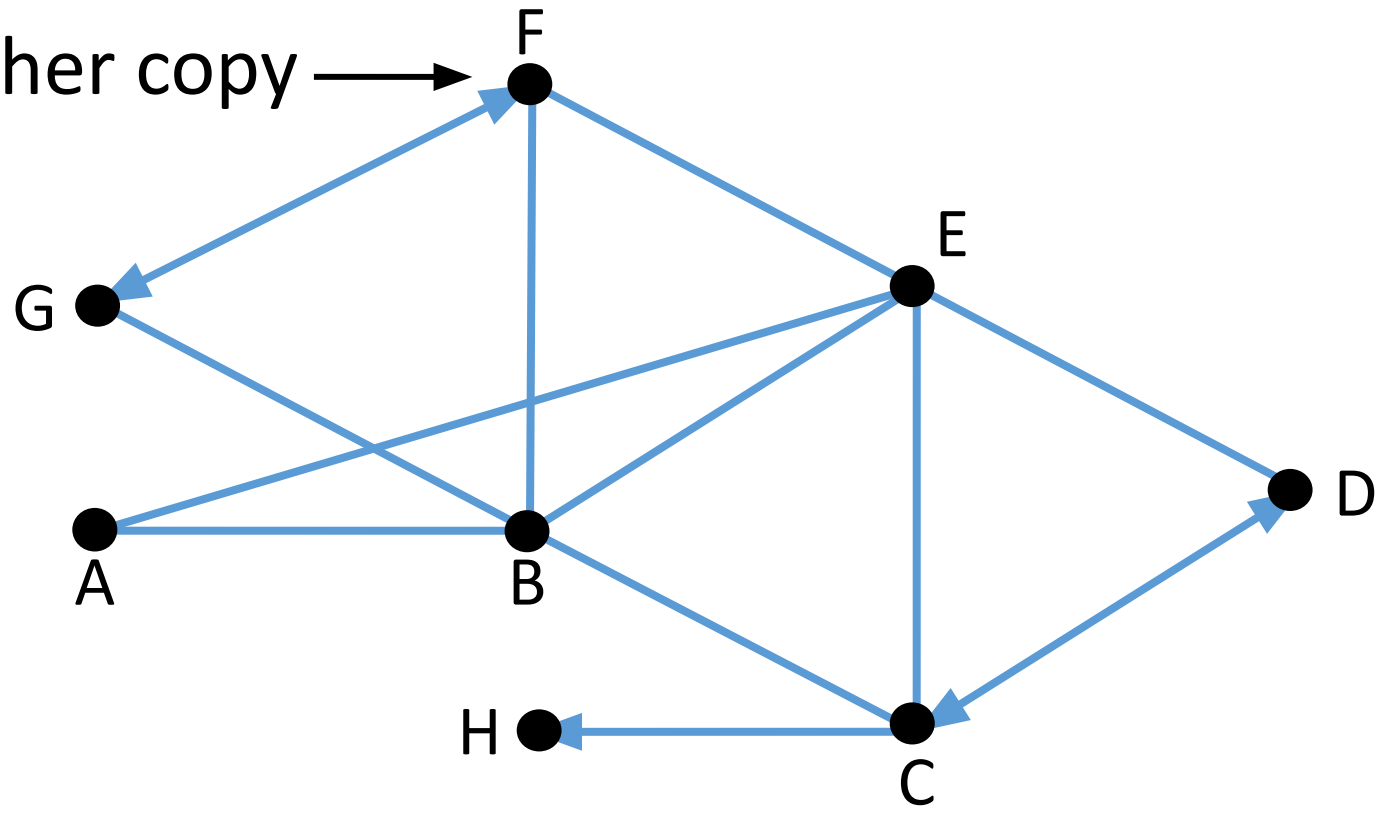
- Next B floods BC, BE, BF, BG, and E floods EB, EC, ED, EF



Flooding (4)

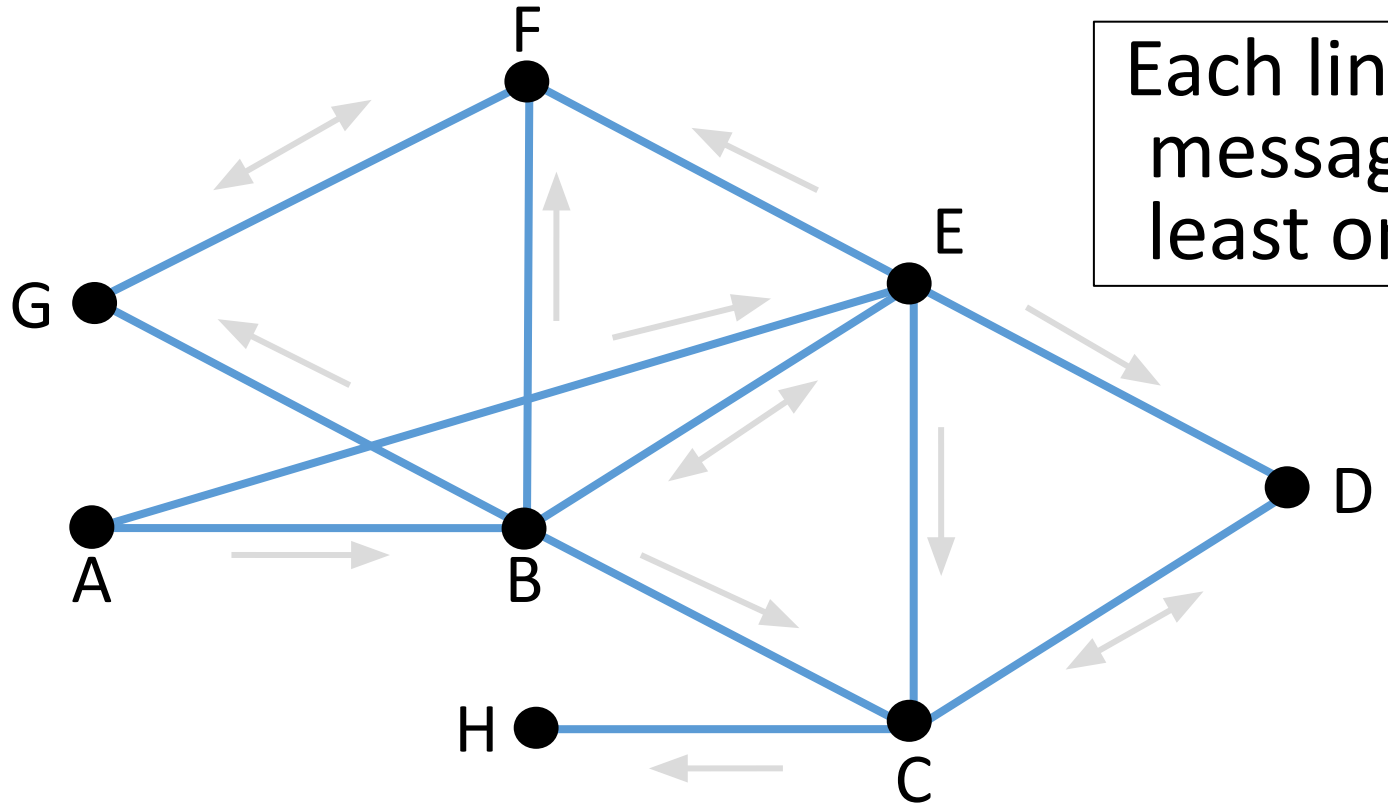
- C floods CD, CH; D floods DC; F floods FG; G floods GF

F gets another copy →



Flooding (5)

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



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

Flooding Details

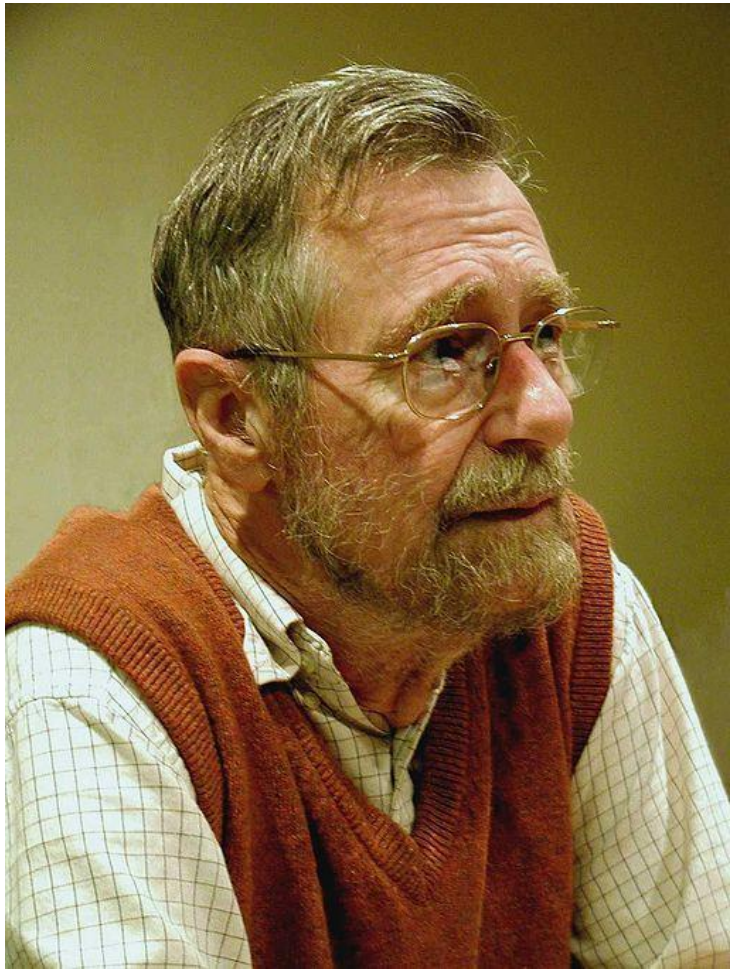
- Remember message (to stop flood) using source and sequence number
  - So next message (with higher sequence) will go through
- To make flooding reliable, use ARQ on each link :)
  - So receiver acknowledges, and sender resends if needed

Link-State Part 2: Dijkstra's Algorithm

Edsger W. Dijkstra (1930-2002)

- Famous computer scientist
  - Programming languages
  - Distributed algorithms
  - Program verification

- Dijkstra's algorithm, 1969
  - Single-source shortest paths, given network with non-negative link costs



By Hamilton Richards, CC-BY-SA-3.0, via Wikimedia Commons

Dijkstra's Algorithm

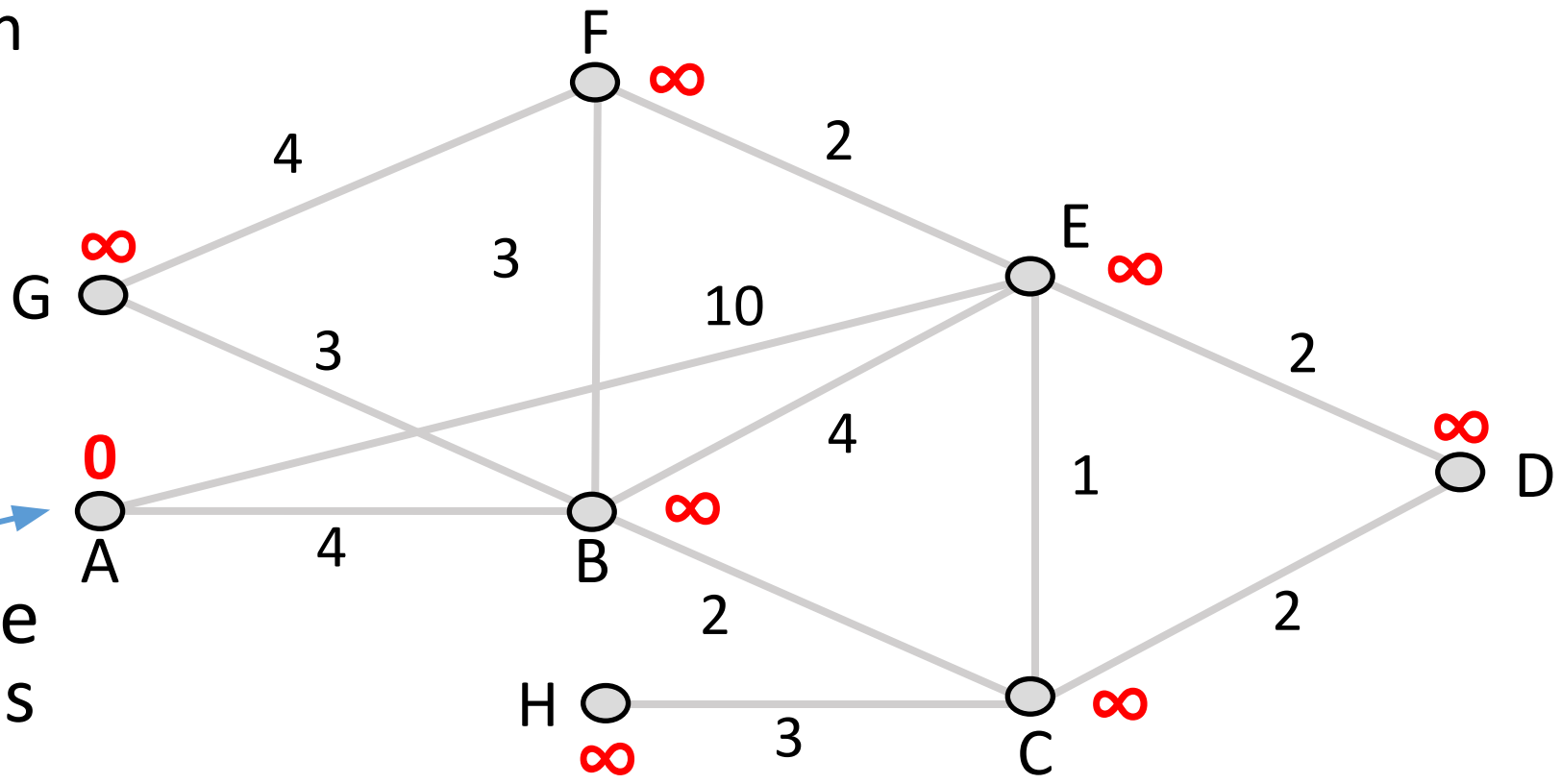
Going to skip since you should have seen this in a previous class before– if you have not, or need a refresher, come to office hours and we can go over it!

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)

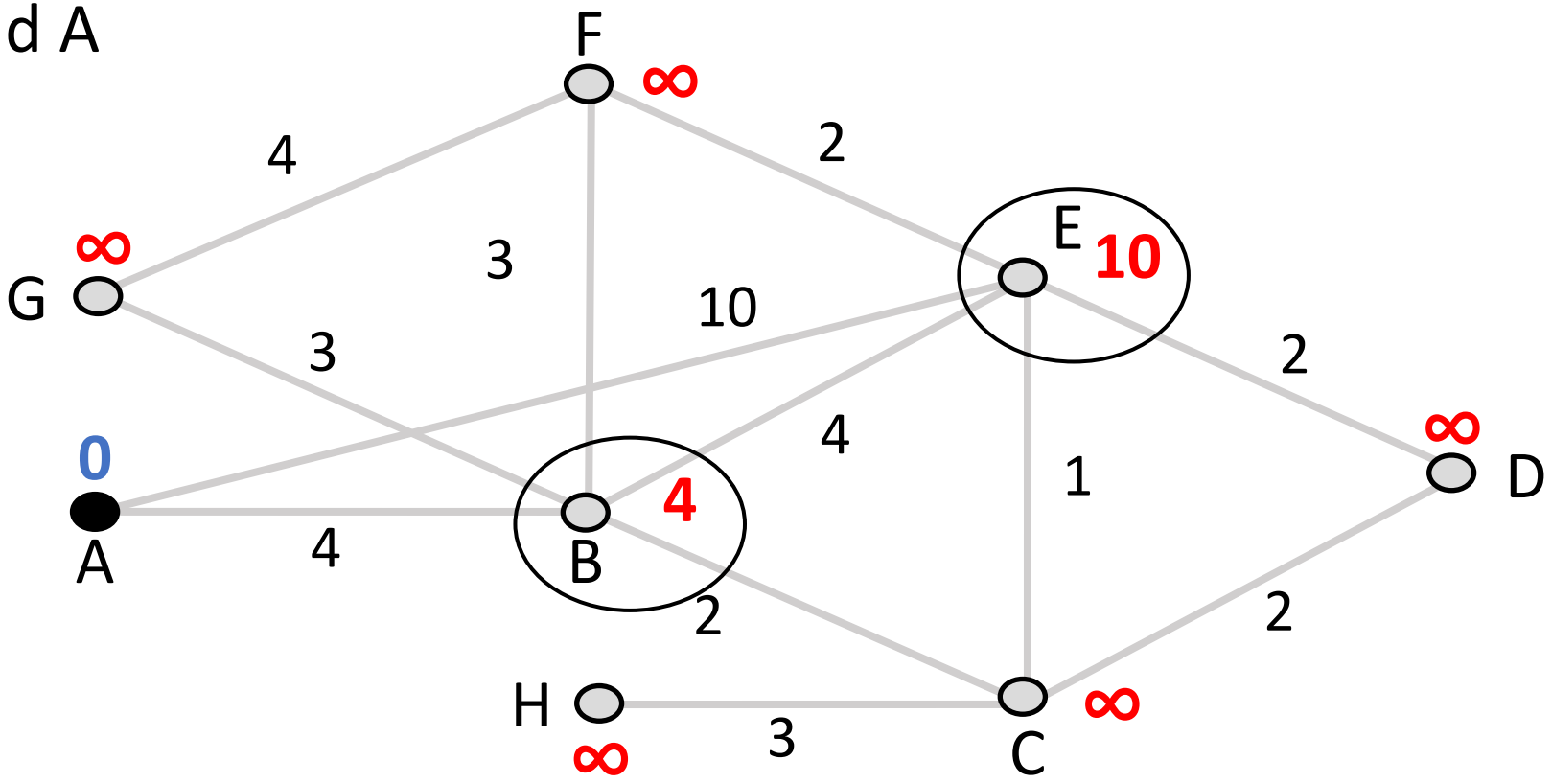
- Initialization



We'll compute shortest paths from A

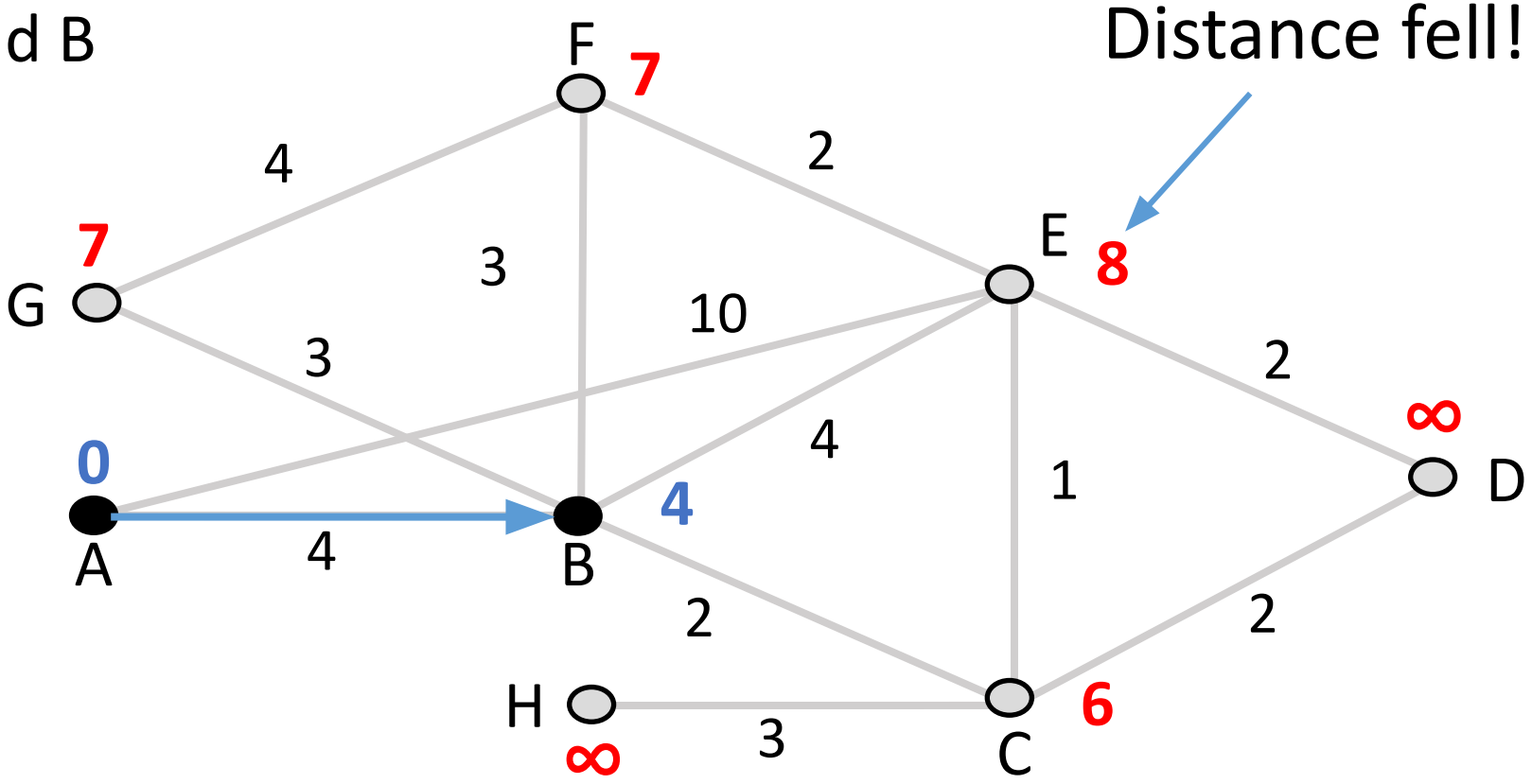
Dijkstra's Algorithm (3)

- Relax around A



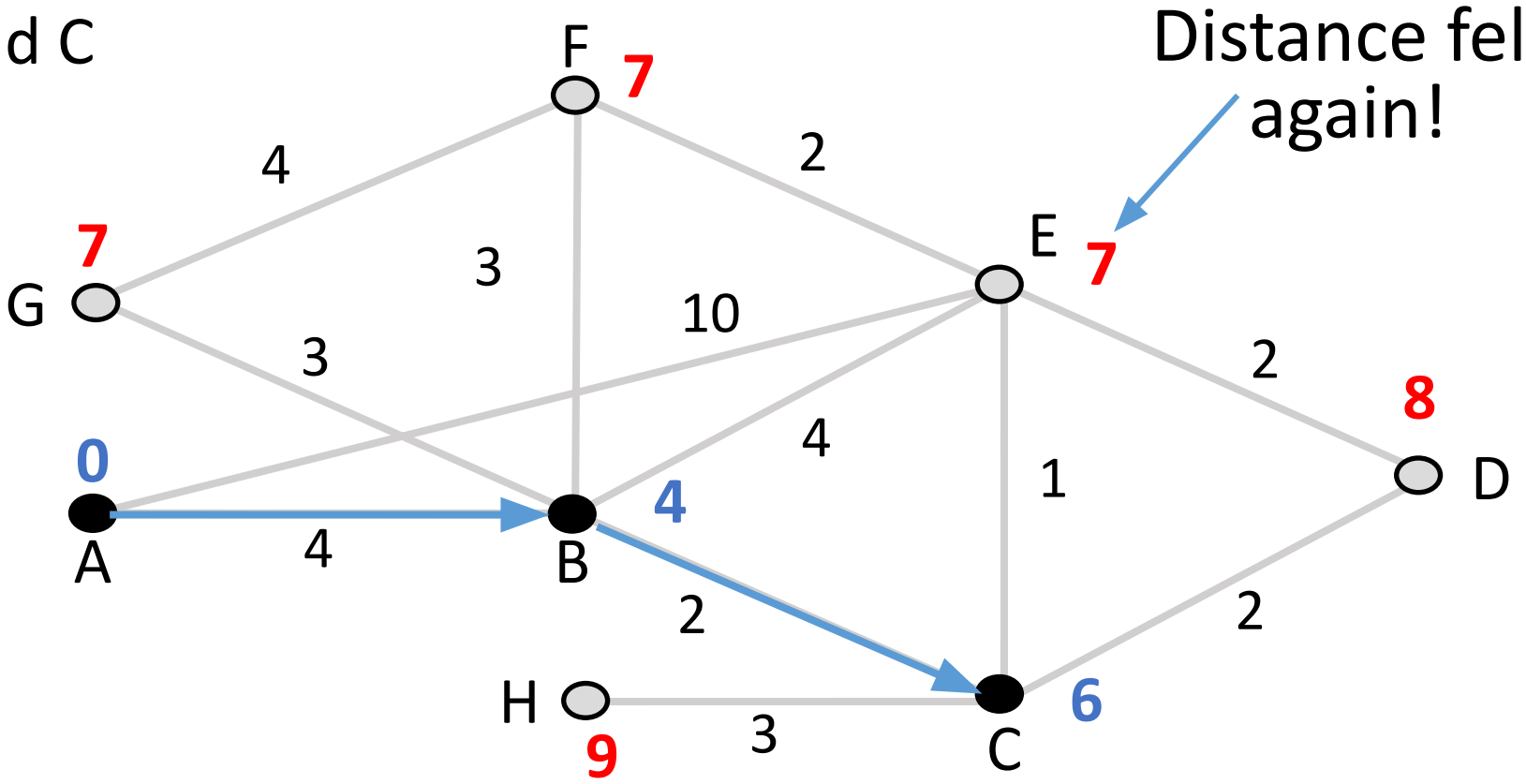
Dijkstra's Algorithm (4)

- Relax around B



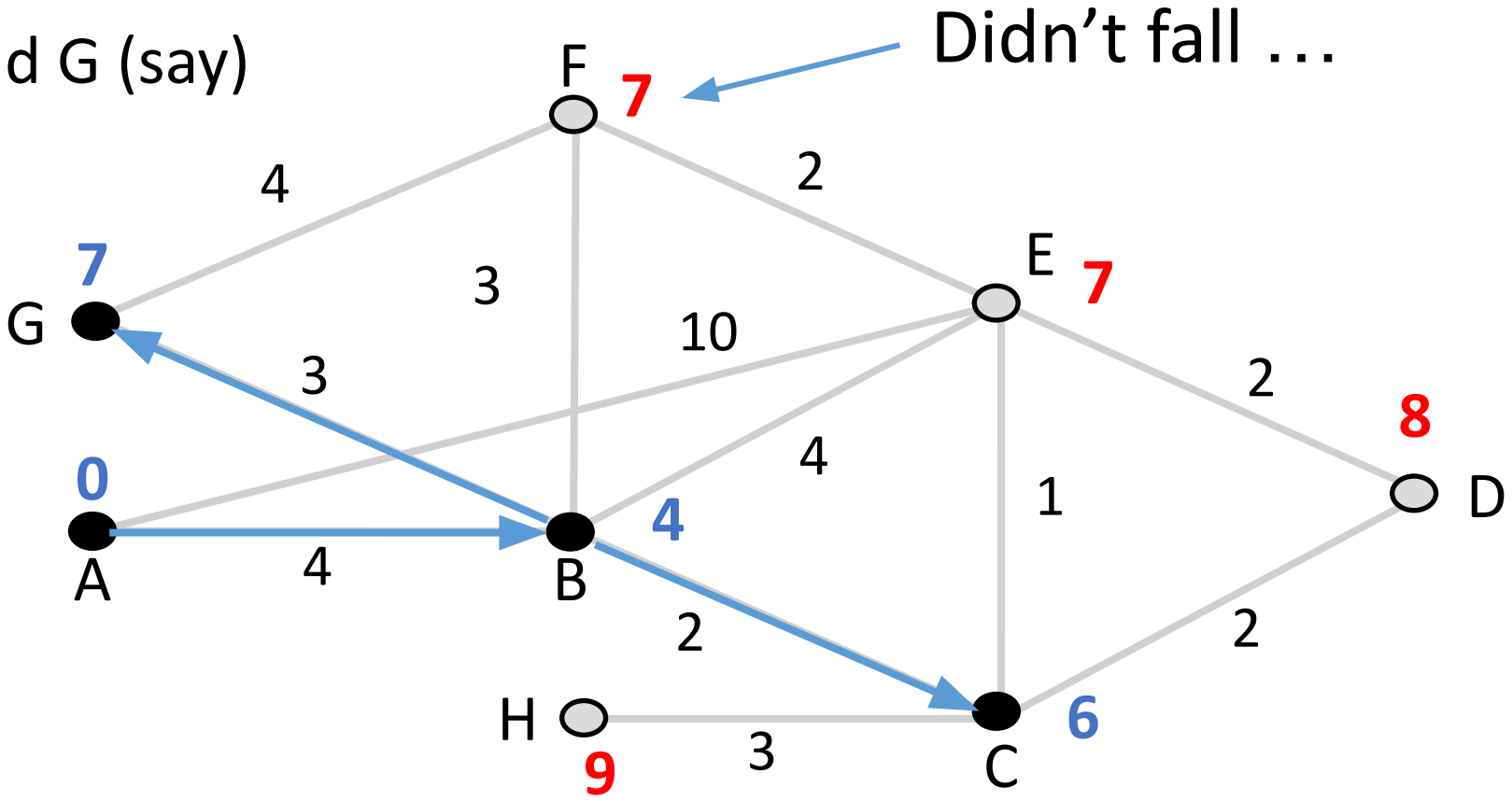
Dijkstra's Algorithm (5)

- Relax around C



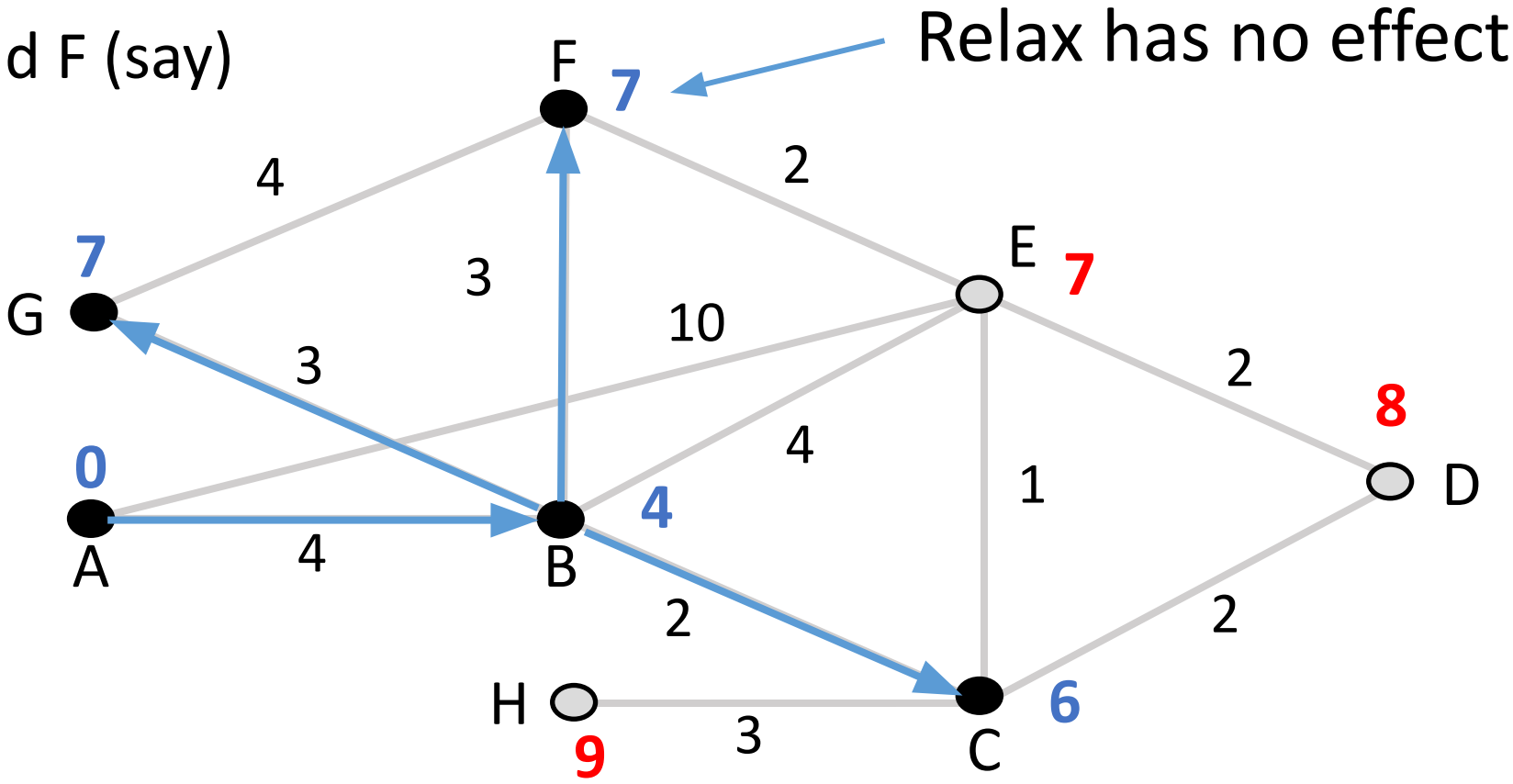
Dijkstra's Algorithm (6)

- Relax around G (say)



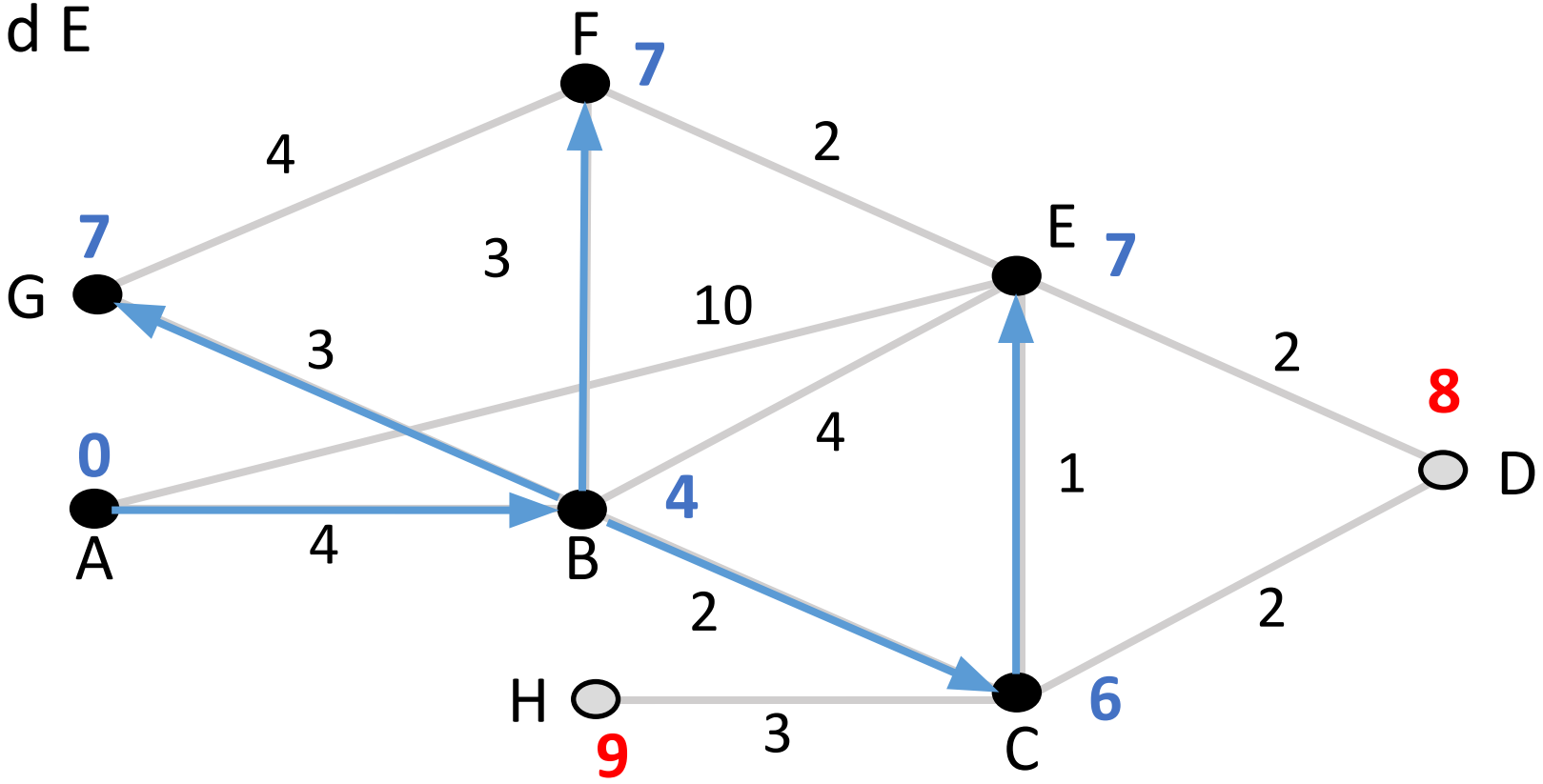
Dijkstra's Algorithm (7)

- Relax around F (say)



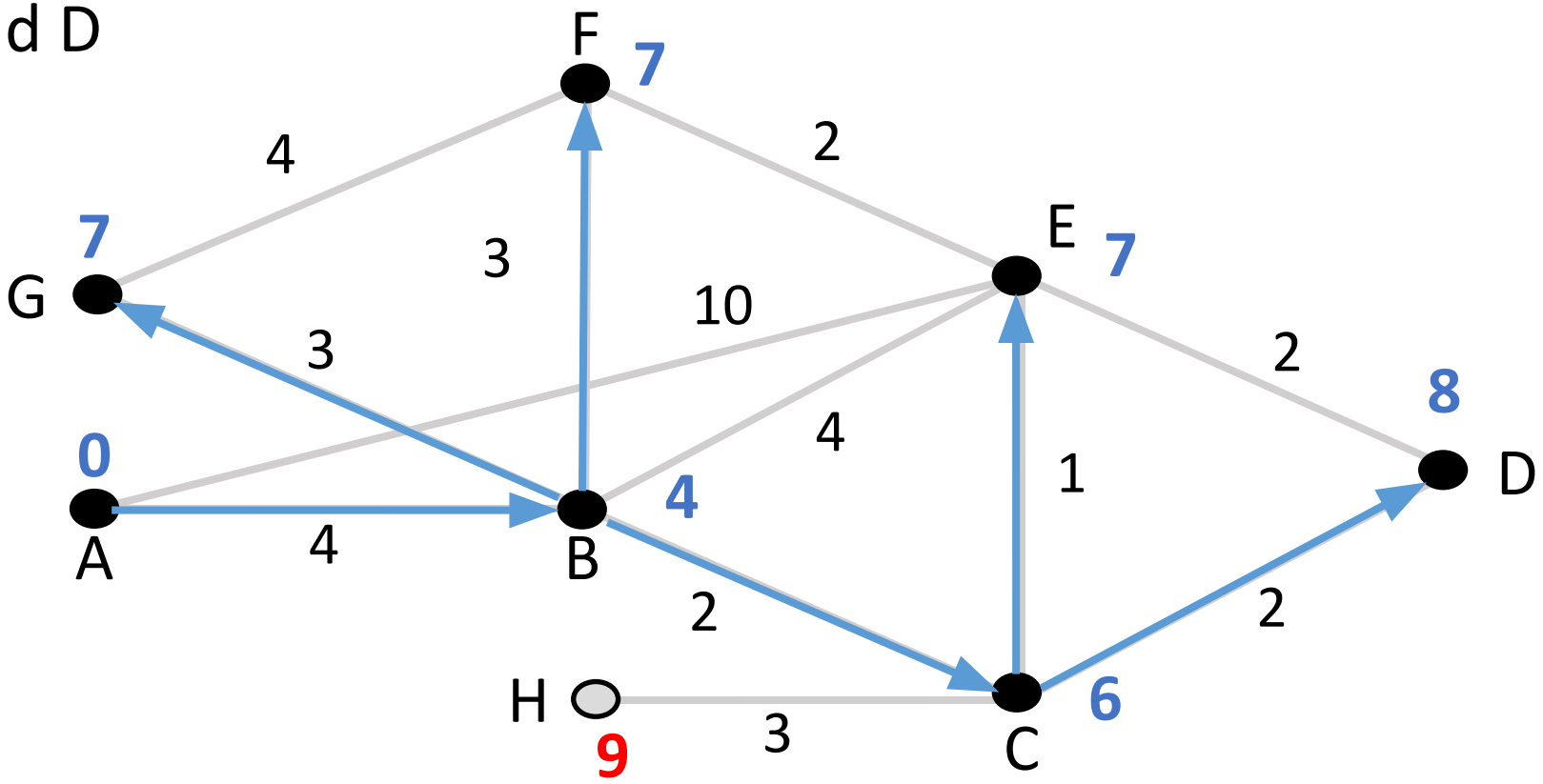
Dijkstra's Algorithm (8)

- Relax around E



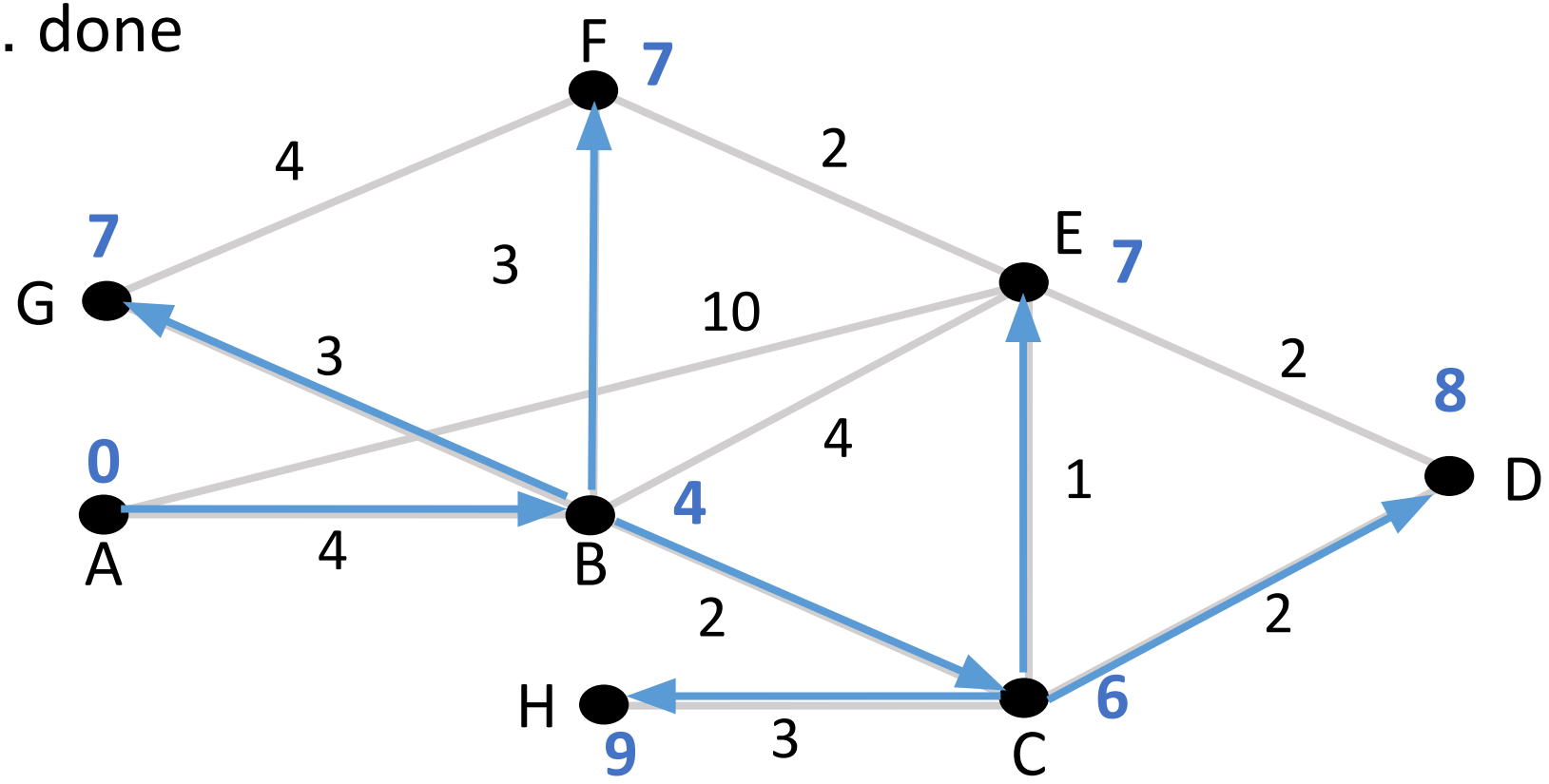
Dijkstra's Algorithm (9)

- Relax around D



Dijkstra's Algorithm (10)

- Finally, H … done

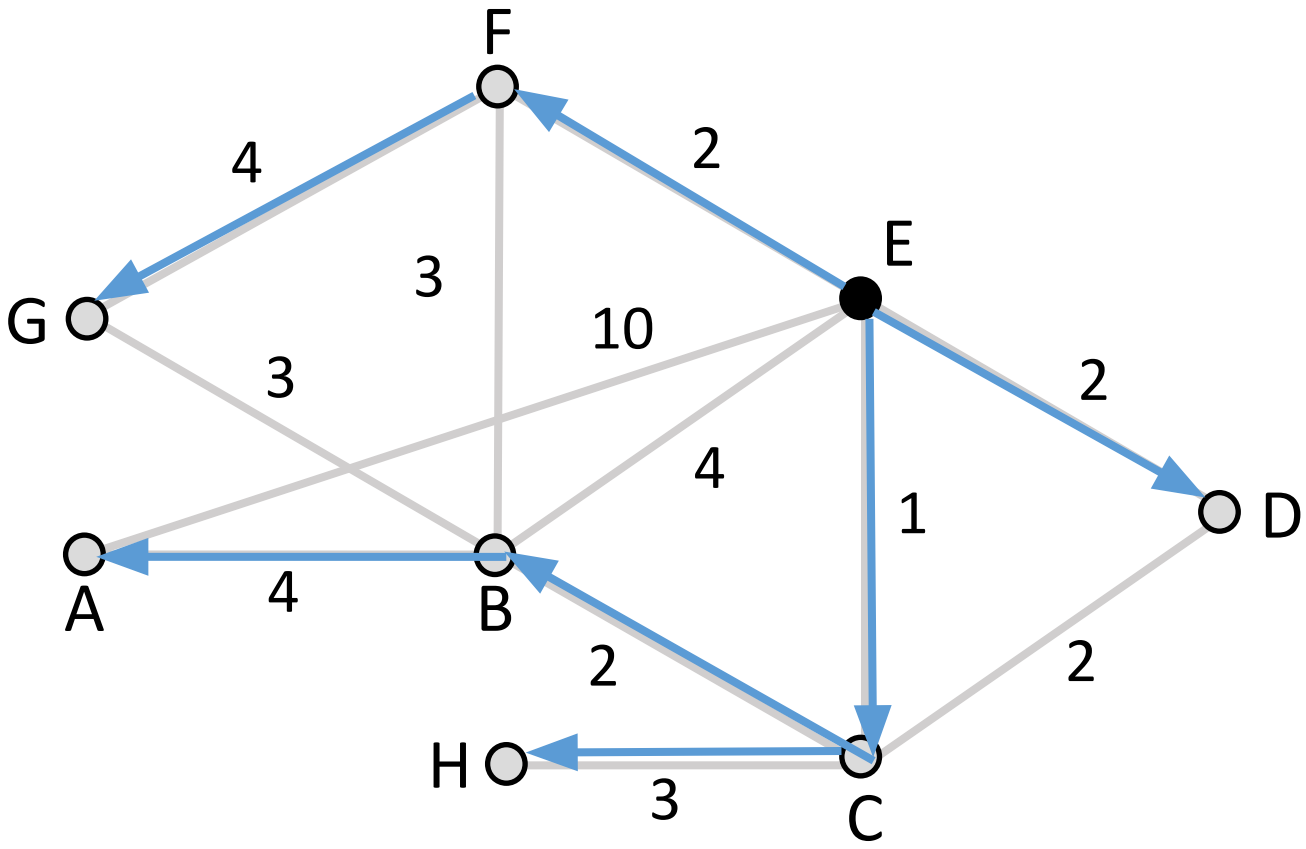


Dijkstra Comments

- Finds shortest paths in order of increasing distance from source
  - Leverages optimality property
  - **Cost must be monotonic**… no negative edges!
- Runtime depends on cost of extracting min-cost node
  - Superlinear in network size (grows fast)
  - Using Fibonacci Heaps the complexity turns out to be O(|E|+|V|log(|V|))
- Gives complete source/sink tree
  - More than needed for forwarding!
  - But requires complete topology

Forwarding Table

Source Tree for E (from Dijkstra)



E's Forwarding Table

To	Next
A	C
B	C
C	C
D	D
E	--
F	F
G	F
H	C

Handling Changes

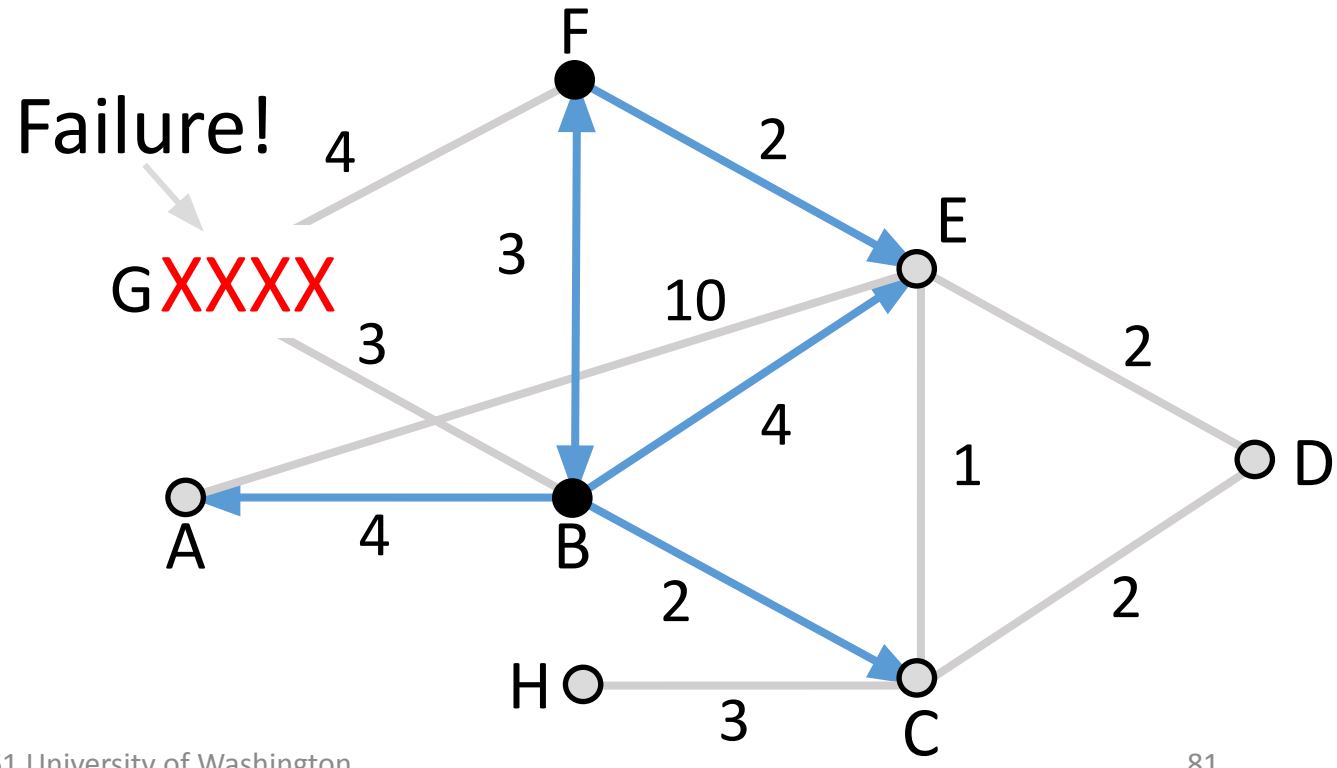
- On change, flood updated LSPs, re-compute routes
  - E.g., nodes adjacent to failed link or node initiate

B's LSP

Seq. #	
A	4
C	2
E	4
F	3
G	∞

F's LSP

Seq. #	
B	3
E	2
G	∞



Handling Changes (2)

- Link failure
  - Both nodes notice, send updated LSPs
  - Link is removed from topology
- Node failure
  - All neighbors notice a link has failed (link state!)
  - Failed node can't update its own LSP
  - But it is OK: all links to node removed

Handling Changes (3)

- Addition of a link or node
  - Add LSP of new node to topology
  - Old LSPs are updated with new link

- Additions are the easy case …

Link-State Complications

- Things that can go wrong:
  - Seq. number reaches max, or is corrupted
  - Node crashes and loses seq. number
  - Network partitions then heals
- Strategy:
  - Include age on LSPs and forget old information that is not refreshed
- Much of the real-world implementation complexity is due to handling corner cases

DV/LS Comparison

Goal	Distance Vector	Link-State
Correctness	Distributed Bellman-Ford	Replicated Dijkstra
Efficient paths	Approx. with shortest paths	Approx. with shortest paths
Fair paths	Approx. with shortest paths	Approx. with shortest paths
Fast convergence	Slow – many exchanges	Fast – flood and compute
Scalability	Excellent – storage/compute	Moderate – storage/compute

IS-IS and OSPF Protocols

- Widely used in large enterprise and ISP networks
  - IS-IS = Intermediate System to Intermediate System
  - OSPF = Open Shortest Path First
- Both are fundamentally link-state protocol with many added features
  - E.g., heirarchical “Areas” for scalability

Equal-Cost Multi-Path Routing

A minor extension to either Distance Vector or Link-State protocols

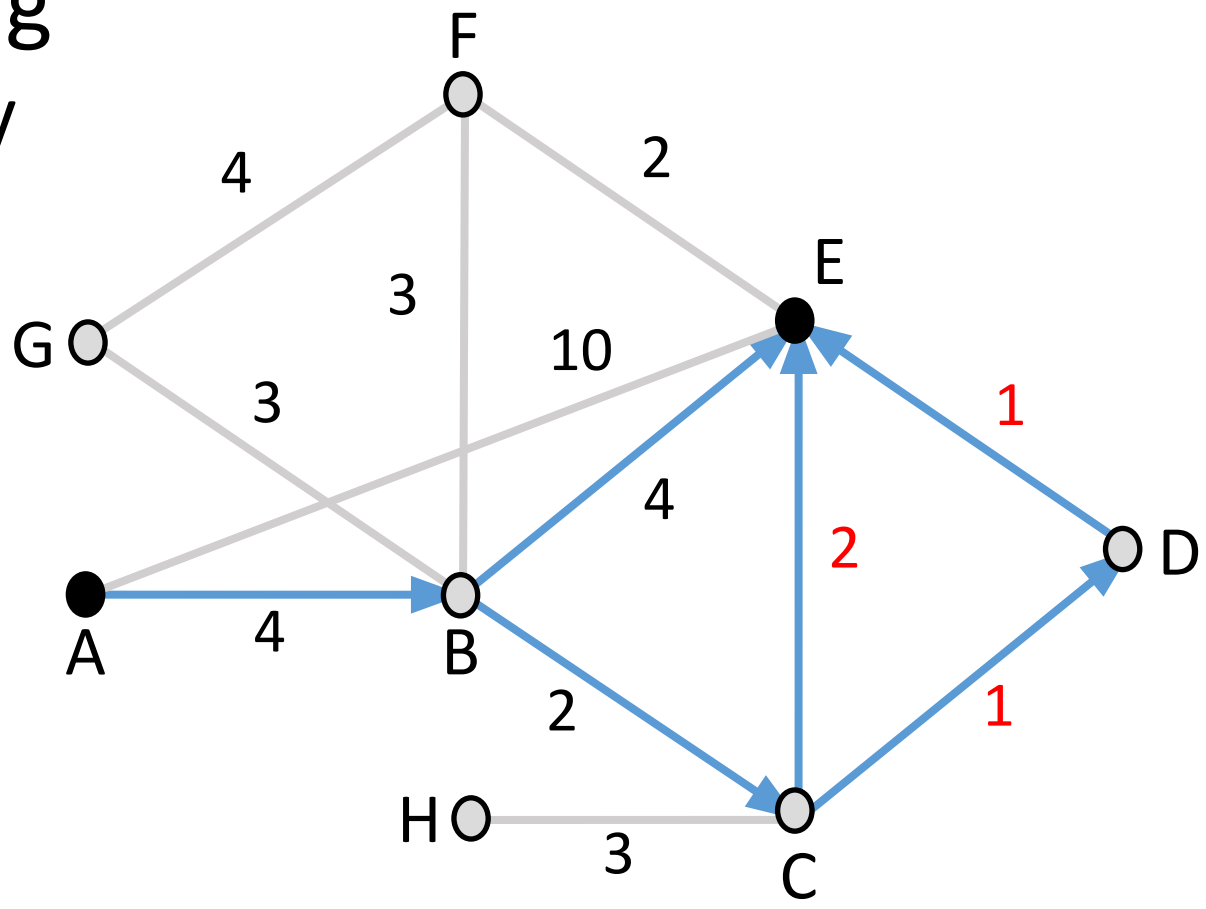
Multipath Routing

- Allow multiple routing paths from node to destination be used at once
  - Topology has them for redundancy
  - Using them can improve performance
- Questions:
  - How do we find multiple paths?
  - How do we send traffic along them?

Equal-Cost Multipath Routes

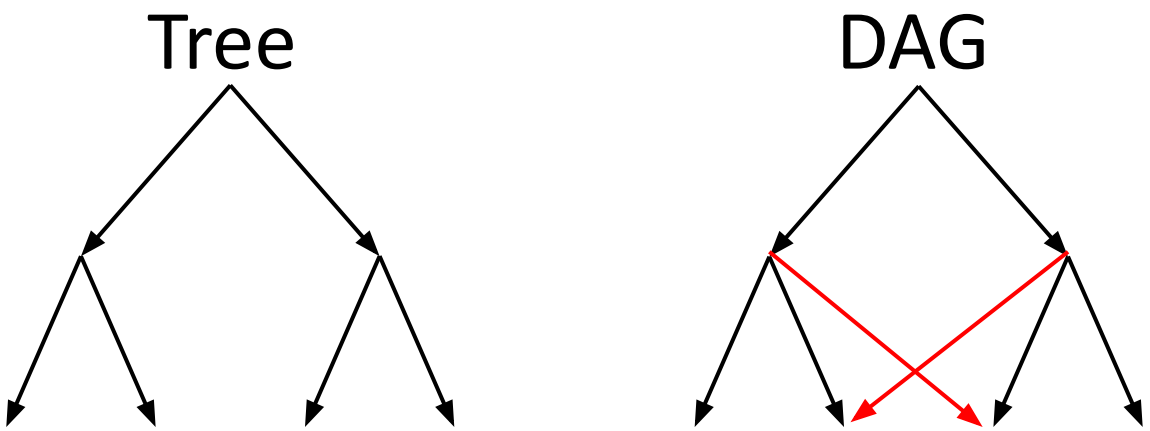
- One form of multipath routing
  - Extends shortest path model by keeping set if there are ties

- Consider A⟶E
  - ABE = 4 + 4 = 8
  - ABCE = 4 + 2 + 2 = 8
  - ABCDE = 4 + 2 + 1 + 1 = 8
  - Use them all!



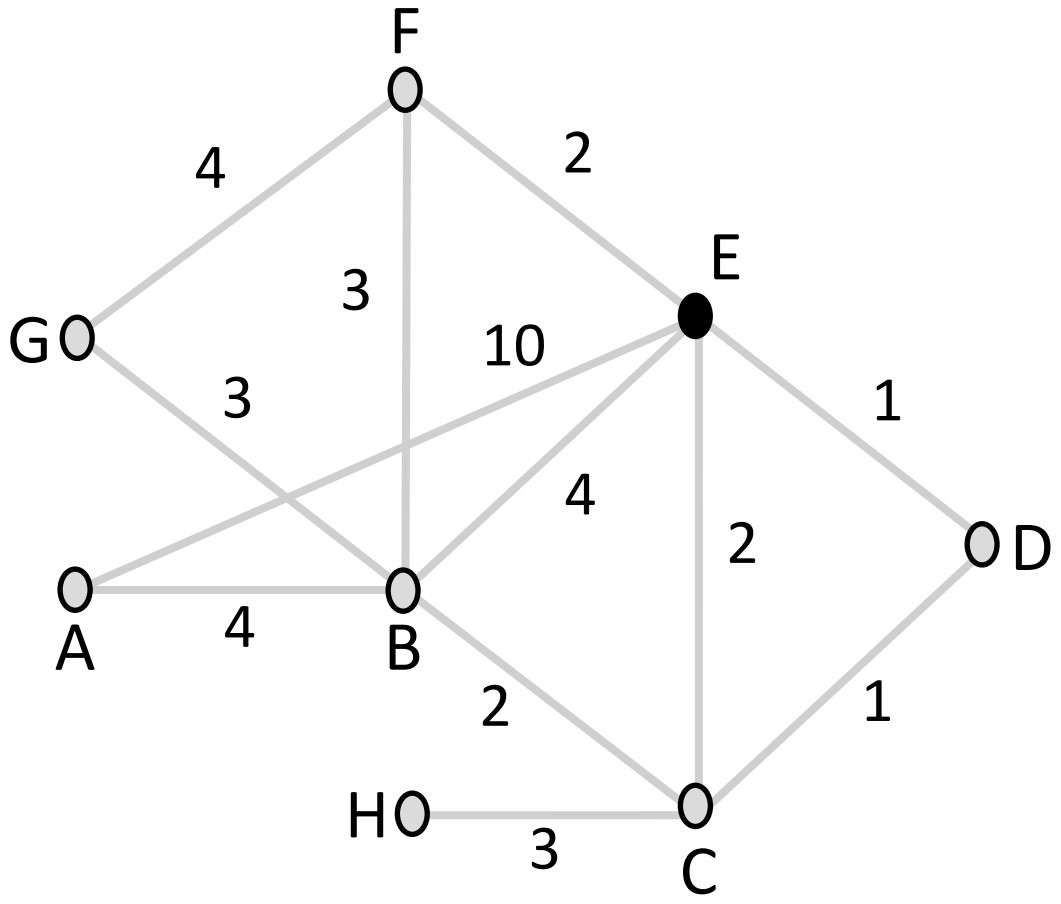
Source "Trees"

- With ECMP, source/sink "tree" is a directed acyclic graph (DAG)
  - Each node has set of next hops
  - Still a compact representation



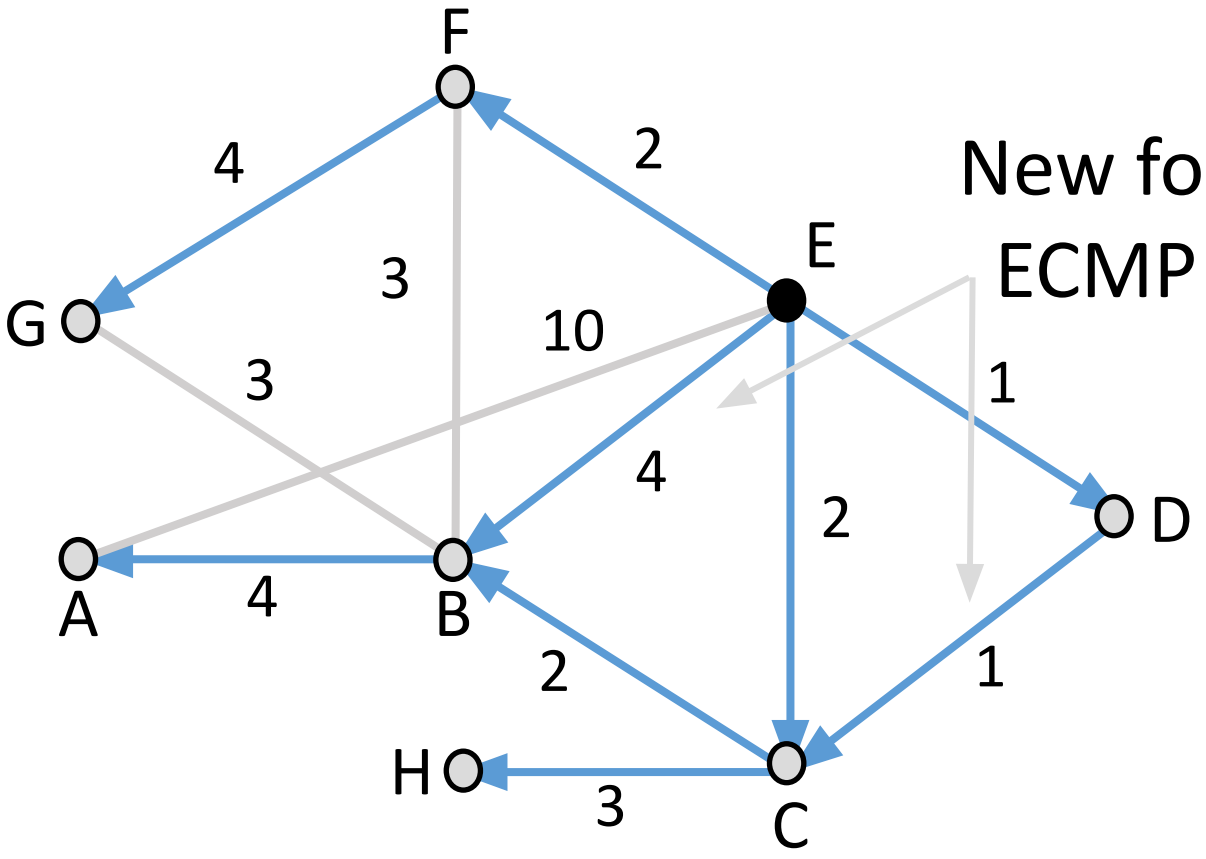
Source “Trees” (2)

- Find the source “tree” for E
  - Procedure is Dijkstra, simply remember set of next hops
  - Compile forwarding table similarly, may have set of next hops
- Straightforward to extend DV too
  - Just remember set of neighbors



Source "Trees" (3)

Source Tree for E



E's Forwarding Table

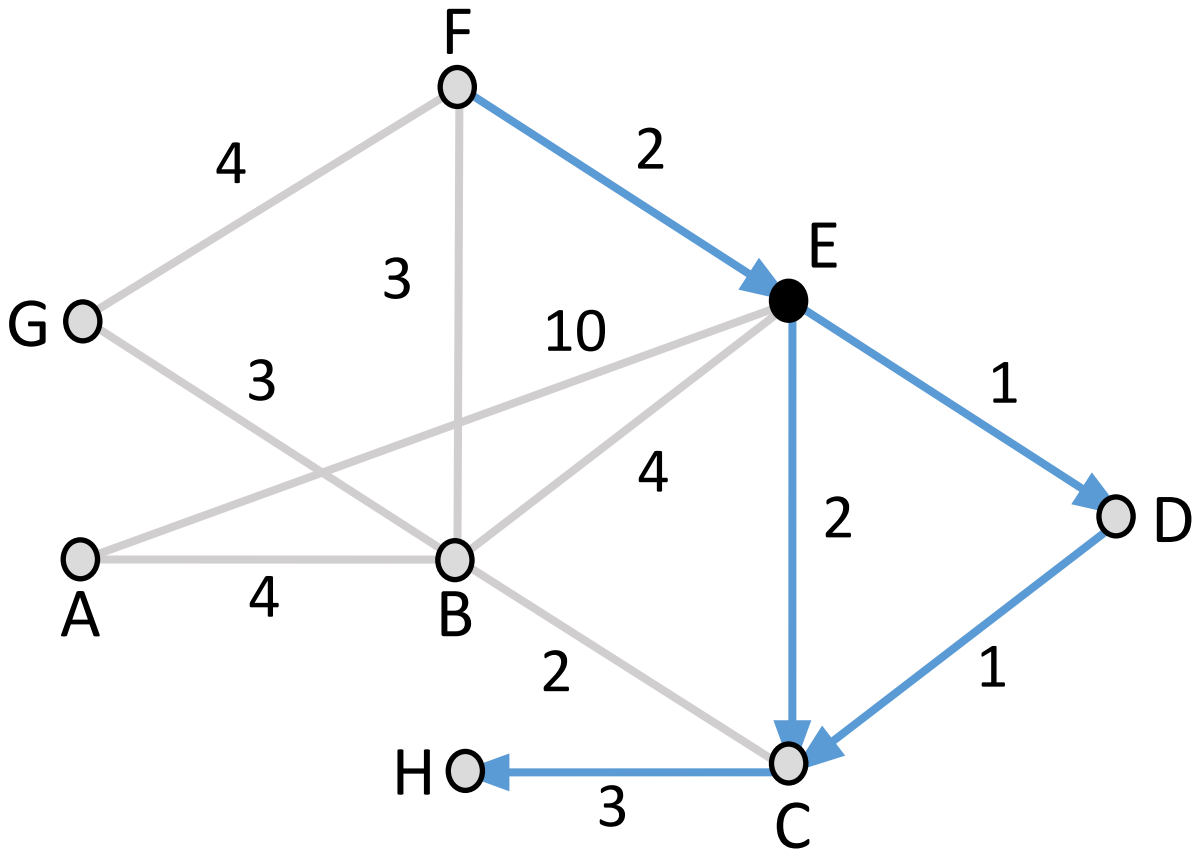
Node	Next hops
A	B, C, D
B	B, C, D
C	C, D
D	D
E	--
F	F
G	F
H	C, D

Forwarding with ECMP

- Could randomly pick a next hop for each packet based on destination
  - Balances load, but adds jitter
- Instead, try to send packets from a given source/destination pair on the same path
  - Source/destination pair is called a flow
  - Map flow identifier to single next hop
  - No jitter within flow, but less balanced

Forwarding with ECMP (2)

Multipath routes from F/E to C/H



E's Forwarding Choices

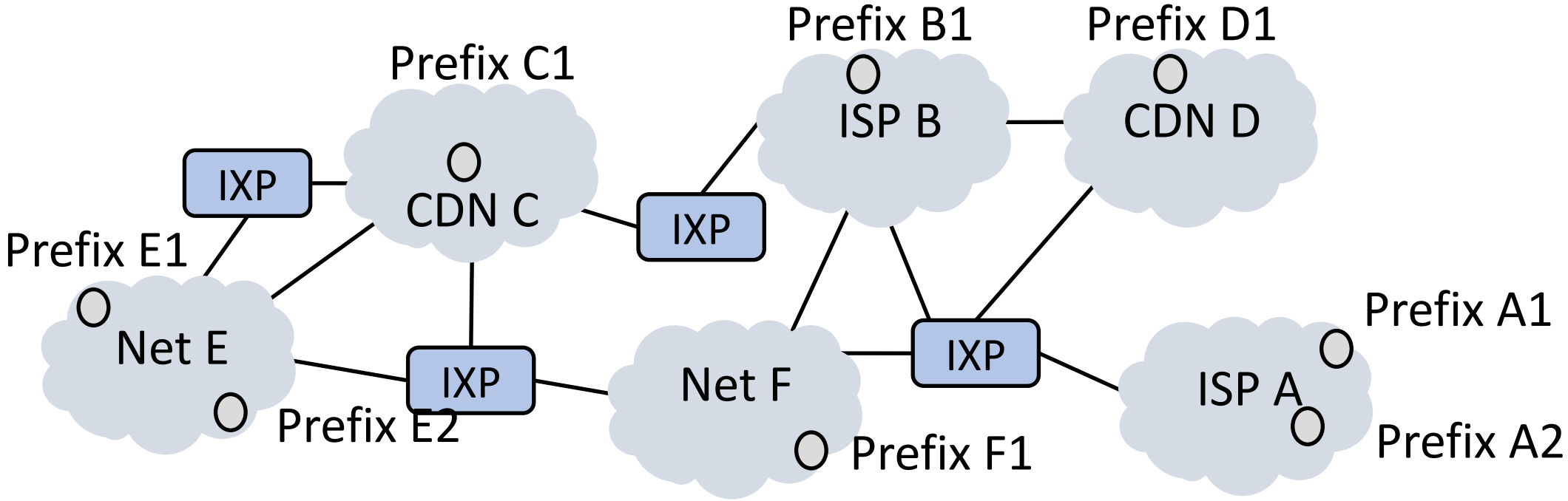
Flow	Possible next hops	Example choice
F → H	C, D	D
F → C	C, D	D
E → H	C, D	C
E → C	C, D	C

Use both paths to get to one destination

Interdomain Routing: Border Gateway Protocol (BGP)

Structure of the Internet

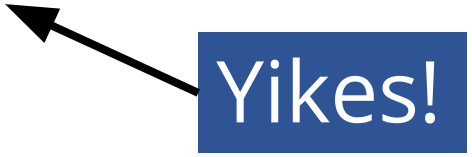
- Networks (ISPs, CDNs, etc.) group with IP prefixes
- Networks are richly interconnected, often using IXPs



Internet-wide Routing Issues

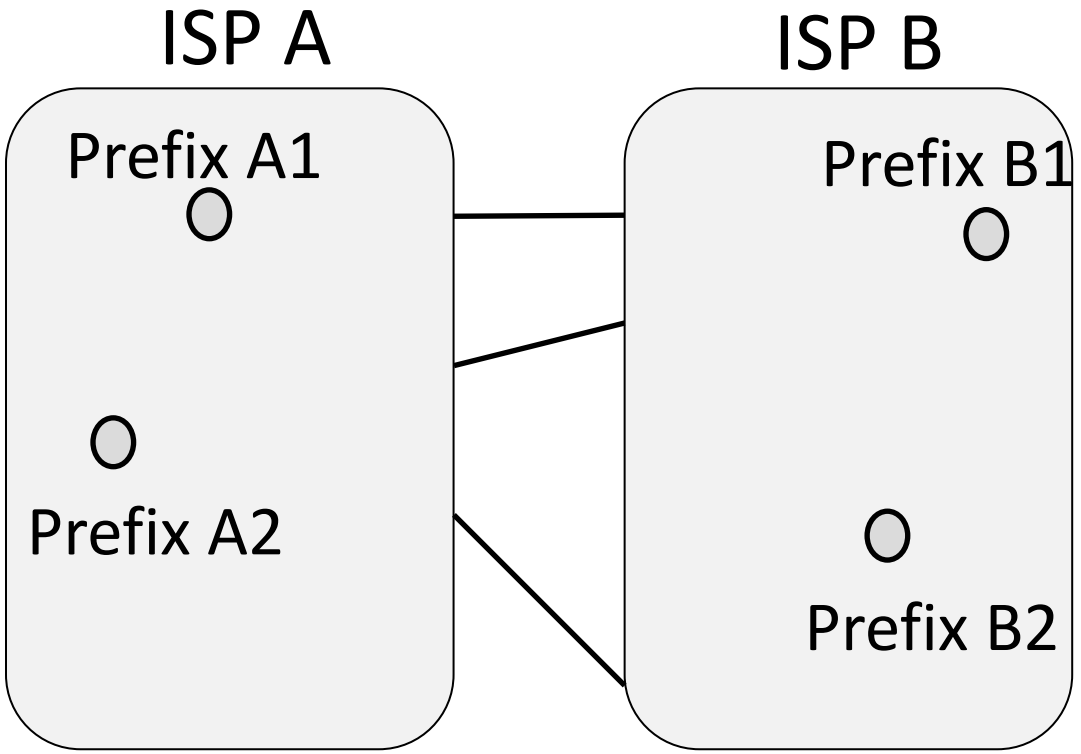
Two problems beyond routing within a network:

1. Scaling to very large networks
   - Techniques of IP prefixes, hierarchy, prefix aggregation
2. Incorporating policy decisions
   - Letting different parties choose their routes to suit their own needs



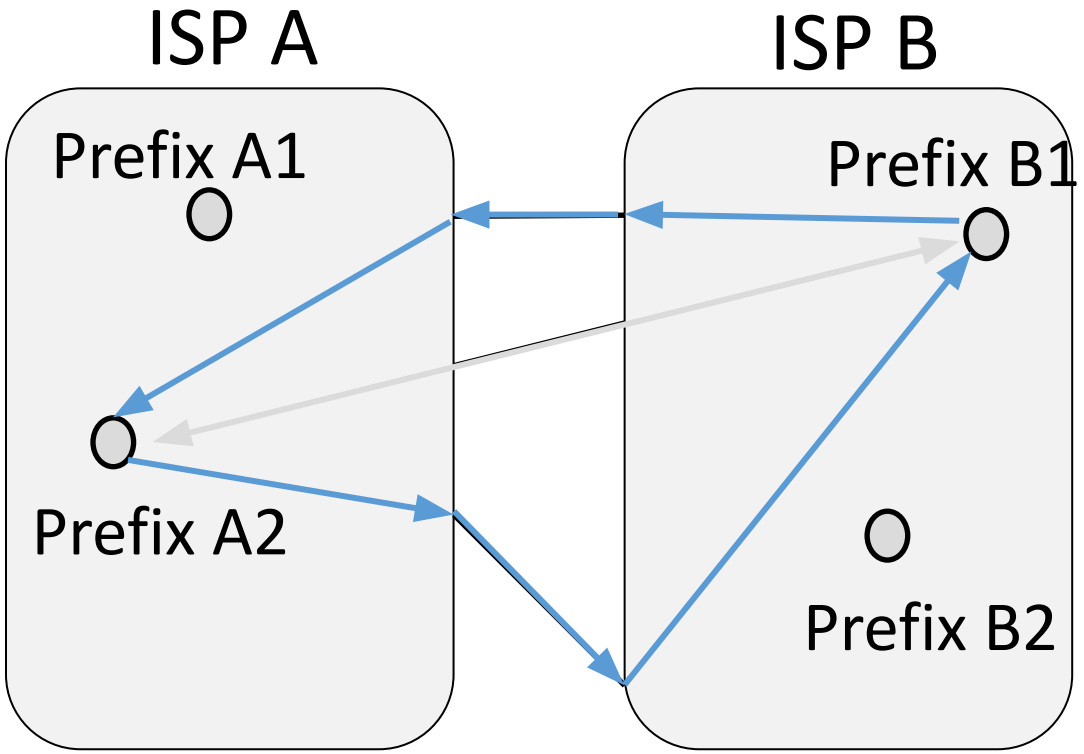
Effects of Independent Parties

- Each party selects routes to suit its own interests
  - e.g, shortest path in ISP
- 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!
- This is a consequence of independent goals and decisions, *not* hierarchy

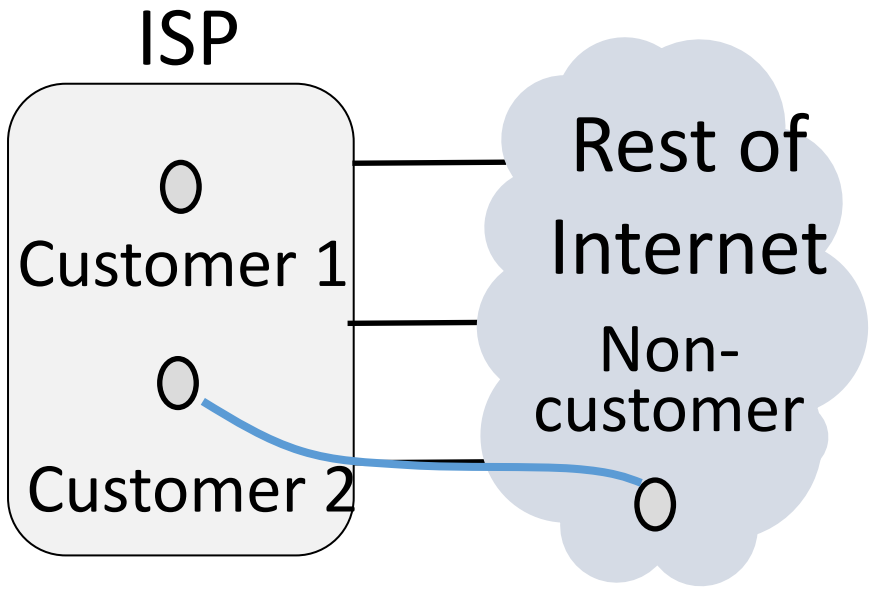


Routing Policies

- Capture the goals of different parties
  - Could be anything
  - E.g., Internet2 only carries non-commercial traffic
- Common policies we'll look at:
  - ISPs give TRANSIT service to customers
  - ISPs give PEER service to each other

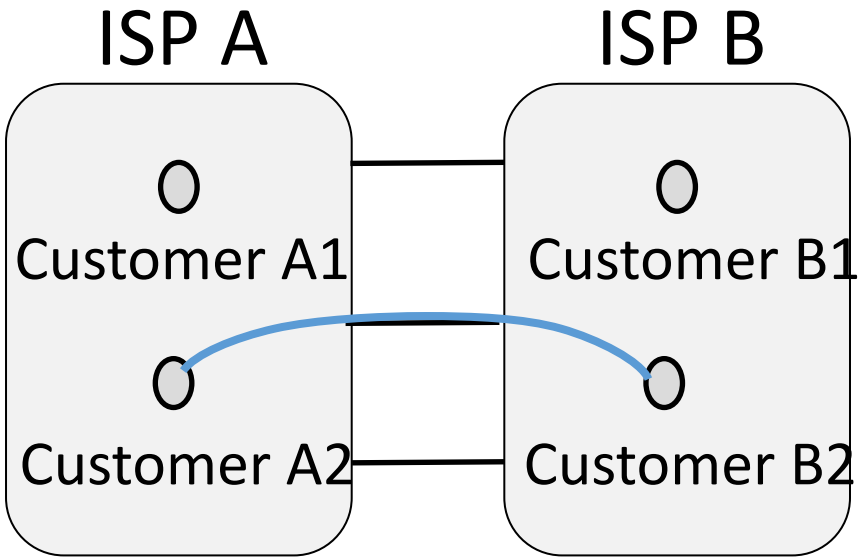
Routing Policies – Transit

- One party (customer) gets TRANSIT service from another party (ISP)
  - ISP accepts traffic for customer from the rest of Internet
  - ISP sends traffic from customer to the rest of Internet
  - Customer pays ISP for the privilege



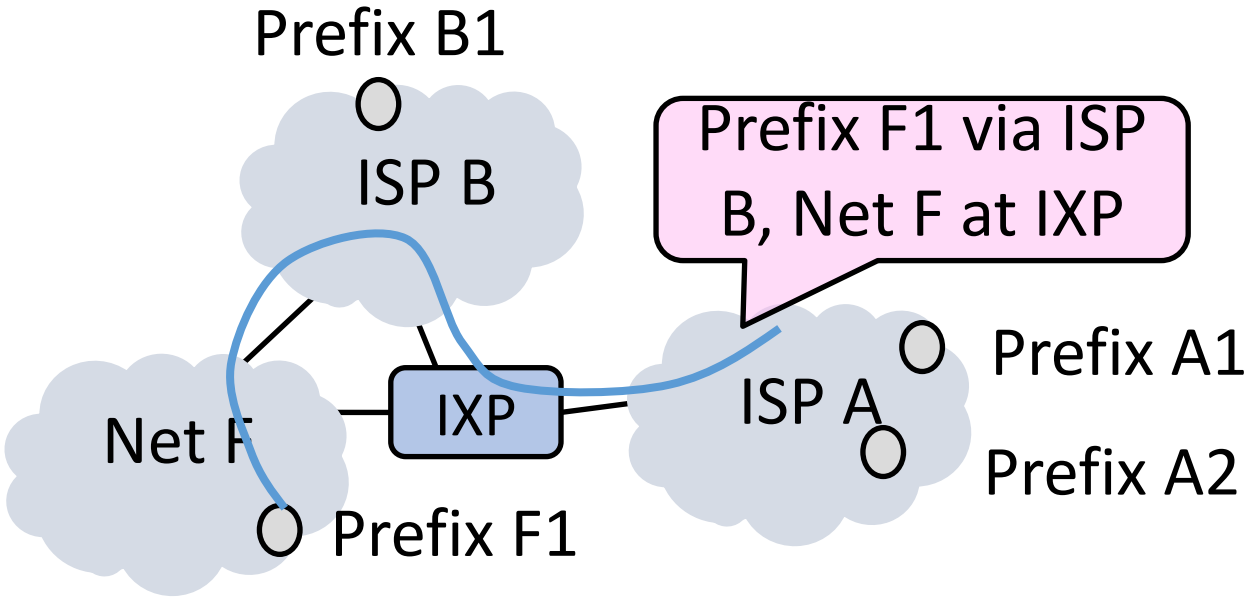
Routing Policies – Peer

- Both party (ISPs in example) get PEER service from each other
  - Each ISP accepts traffic from the other ISP only for their customers
  - ISPs do not carry traffic to the rest of the Internet for each other
  - ISPs don't pay each other



Routing with BGP (Border Gateway Protocol)

- iBGP is for internal routing
- eBGP is interdomain routing for the Internet
  - Path vector, a kind of distance vector



Routing with BGP (2)

- Parties like ISPs are called AS (Autonomous Systems)
  - AS numbers assigned by regional Internet Assigned Numbers Authority (IANA) like APNIC
- AS's configure (often manually) their internal BGP routes/advertisements
- External routes go through complicated filters for forwarding/filtering
- AS BGP routers communicate with each other to keep consistent routing rules

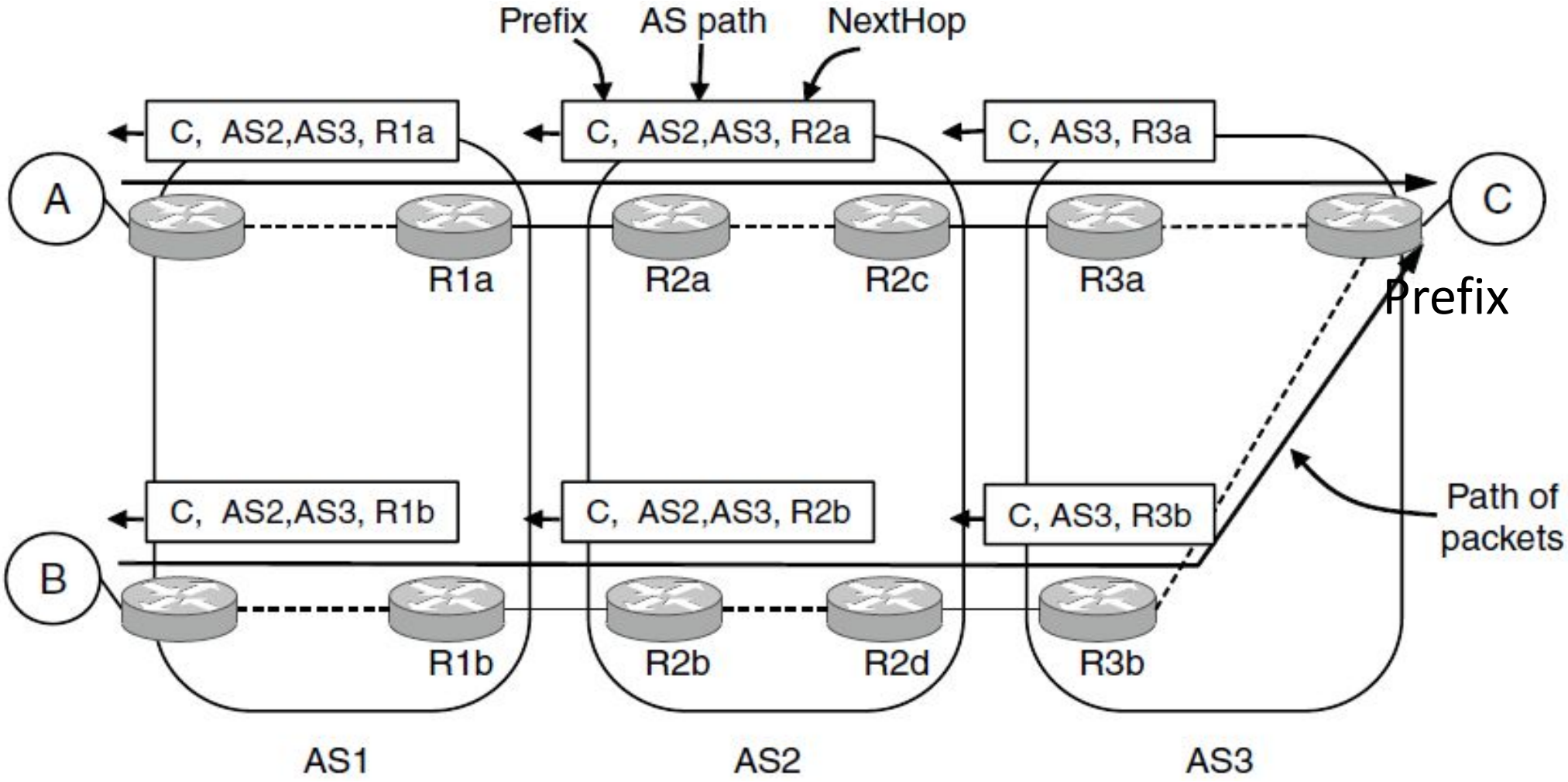
Routing with BGP (2)

- Border routers of ASes announce BGP routes
- Route announcements have IP prefix, path vector, next hop
  - Path vector is list of ASes on the way to the prefix
  - List is to find loops
- Route announcements move in the opposite direction to traffic

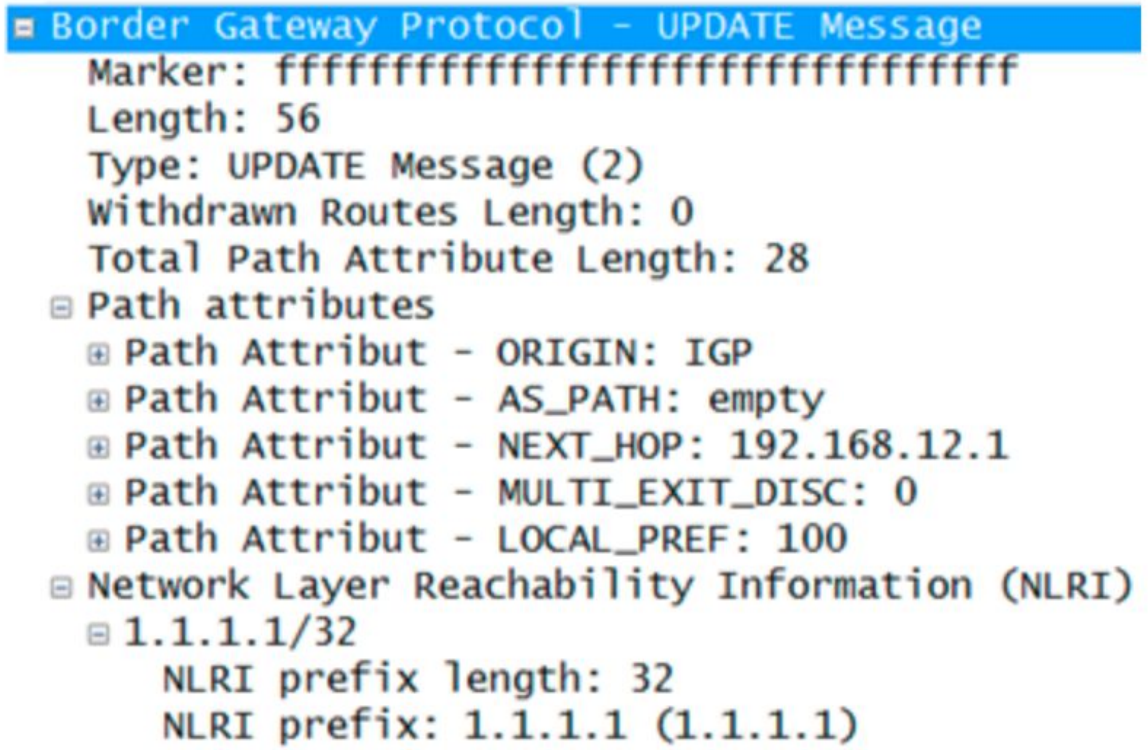
Routing with BGP (3)

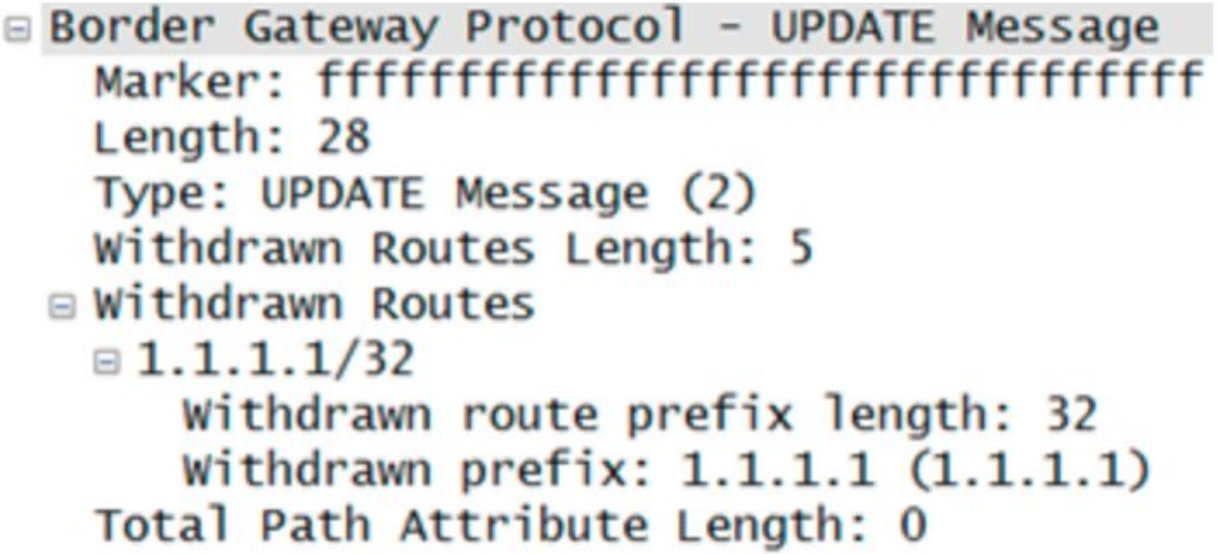
- Application-layer protocol (uses TCP)
- Types of BGP Messages
  - Open: Create a relationship
  - Keepalive: Still here (reset timeouts)
  - Update: A route changed
  - Notification: Error message
  - Route Refresh: Please send me the route again

Routing with BGP (5)



Routing with BGP (5)





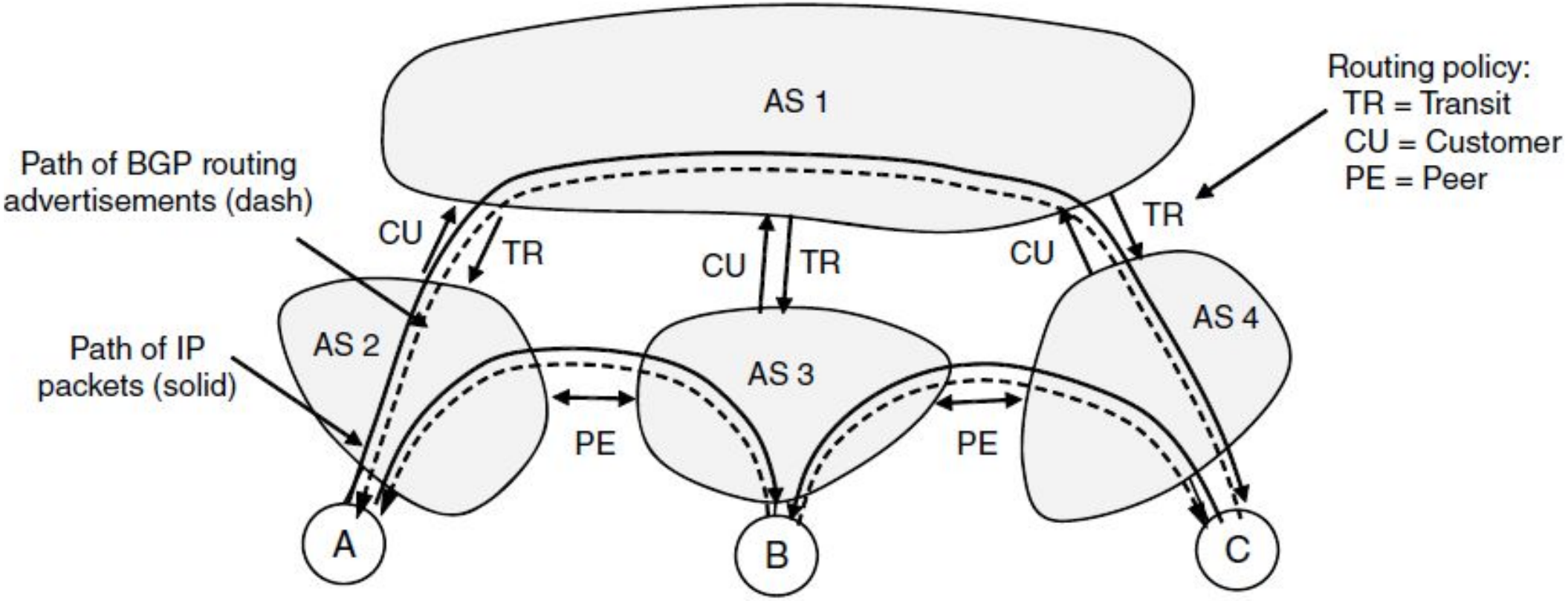
Routing with BGP (6)

Policy is implemented in two ways:

1. Border routers of ISP announce paths only to other parties who may use those paths
   - Filter out paths others can't use
2. Border routers of ISP select the best path of the ones they hear in any, non-shortest way

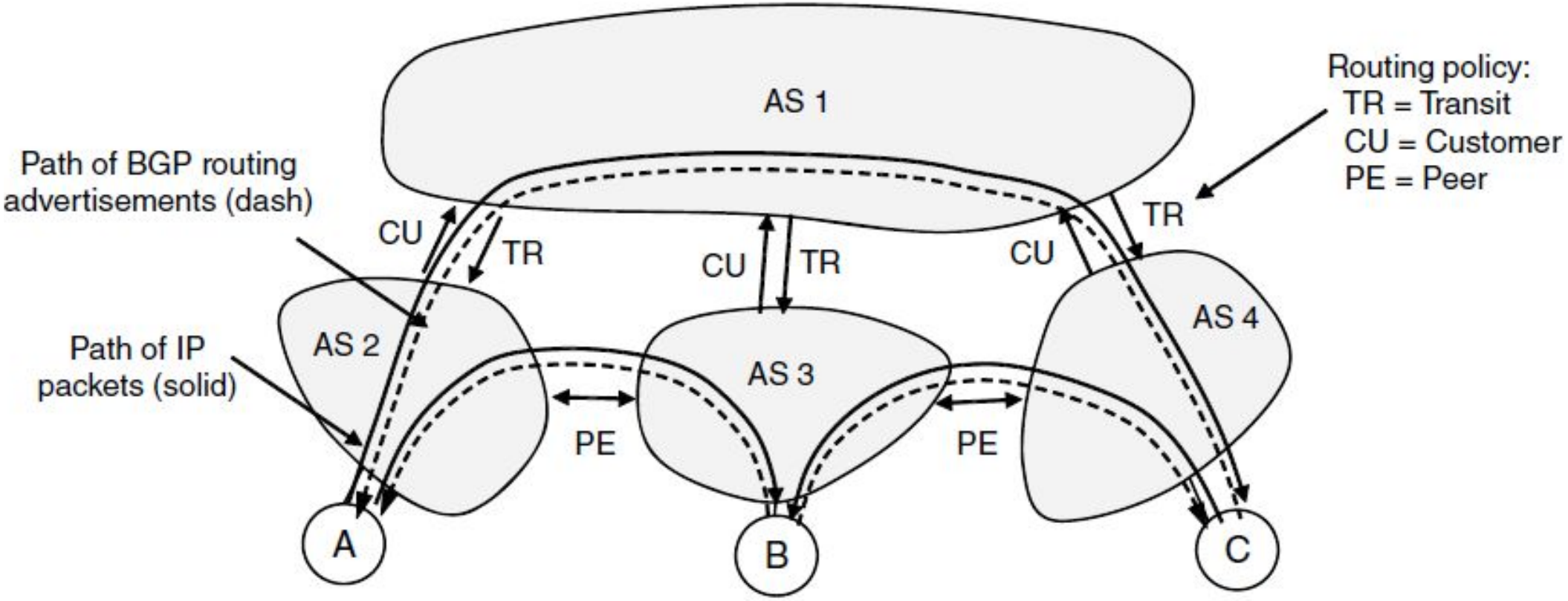
Routing with BGP (7)

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



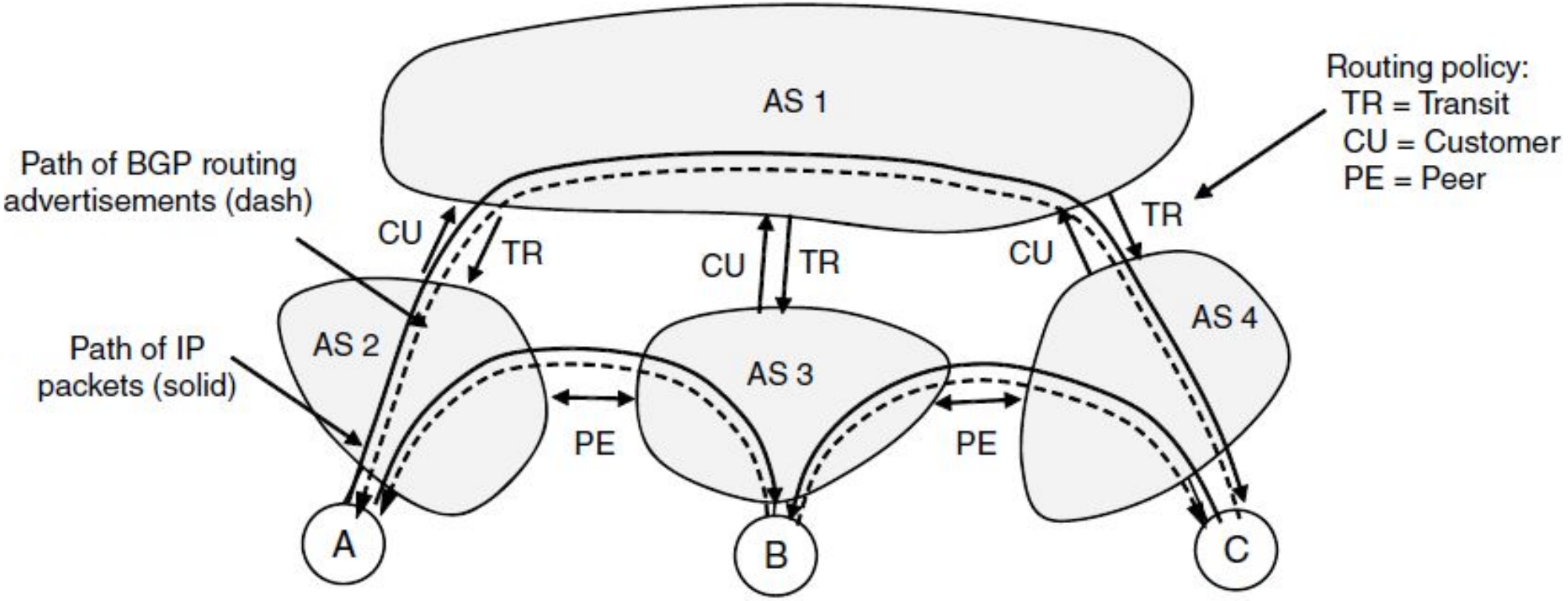
Routing with BGP (8)

- CUSTOMER (other side of TRANSIT): AS2 says [A, (AS2)] to AS1



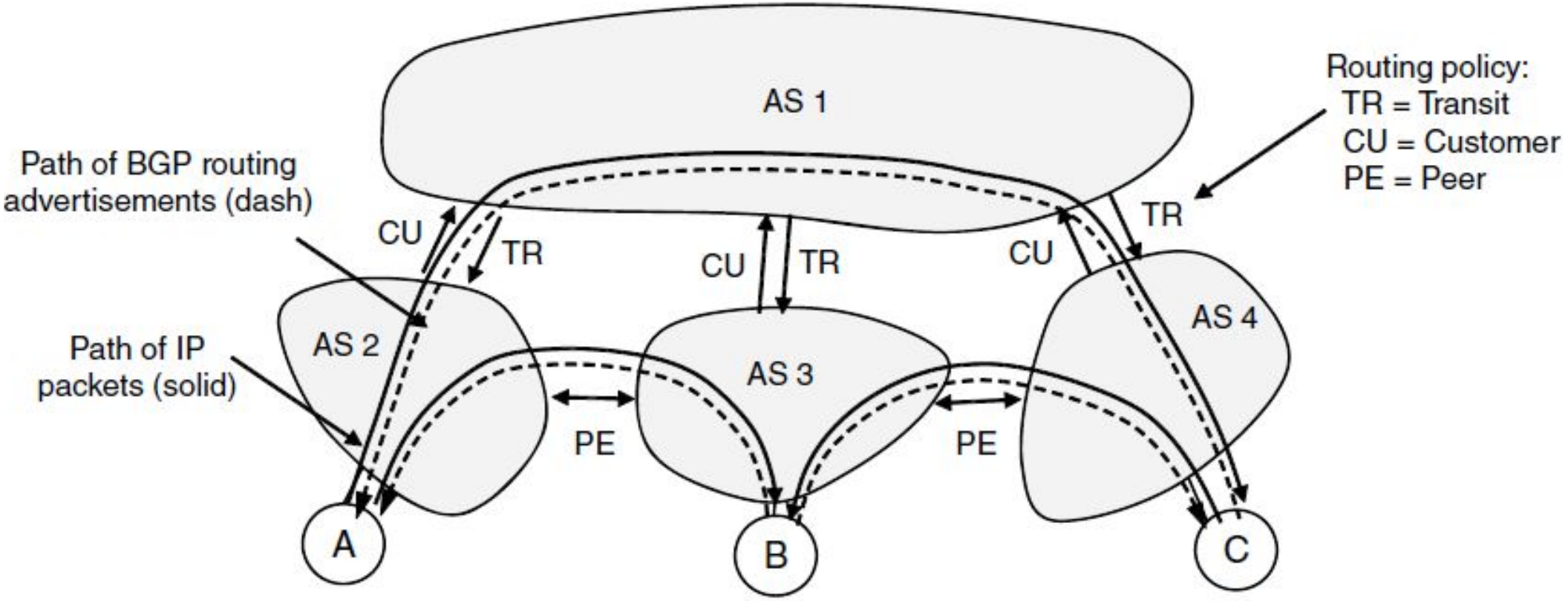
Routing with BGP (9)

- PEER: AS2 says [A, (AS2)] to AS3, AS3 says [B, (AS3)] to AS2



Routing with BGP (10)

- AS2 has two routes to B (AS1, AS3) and chooses AS3 (Free!)



Internet Exchange Points (IXPs)

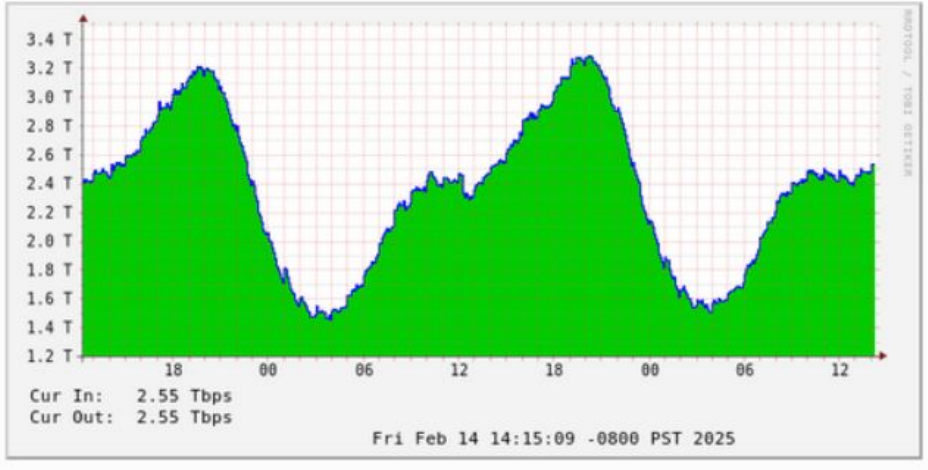
- Centralized location for AS interconnect
- Often “public” - anyone can join (if they can pay)
  - Usually interesting organizationally, though some big (multi-IXP) players
- Many-to-many instead of 1-1
- Often operates a “route server” to reduce the n-to-n complexity of a ton of peering relationships

SEATTLE INTERNET EXCHANGE

The SIX is a non-profit Internet Exchange Point in Seattle, Washington. We provide reliable and low-cost interconnection between member networks in the Northwest United States and beyond. Networks connect at speeds from 1 to Nx400 Gbps.

Quick Facts:

- 372 ASNs, 437 routers, 366 members
- 3 member-facing 400GbE ports
- 193 member-facing 100GbE ports
- 1 member-facing 40GbE port
- 208 member-facing 10GbE ports
- 20 member-facing GigE ports
- 3.70 Tbps of peak traffic
- 22+ terabits of connected member capacity



General Announcements

2025-01-15	SAVE THE DATE!: For the April 17th, 2025 Online Annual Member Meeting, please see this agenda and Calendar ICS.
2025-01-10	Updated port counts over time can be found here: csv png xls
2023-03-27	Broadcast, Unknown-Unicast, and Multicast (BUM) graphs are now available.
2022-04-19	400GbE one-time port fee introduced on join page at $15,000 with availability at the Westin Building and KOMO Plaza.
2020-01-15	The route servers now perform additional strict filtering using RPKI

Participant Updates

2025-02-13	Welcome Bytefilter LLC (AS1002) at 10G via the NOCIX extension.
2025-02-10	Whitesky Communications (AS62887) has added 1x10G, bringing their capacity to a total of 2x10G. They are connected to the route servers.
2025-02-09	Welcome imput (AS16909) at 10G via the NOCIX extension.
2025-02-07	Valve Corporation (AS32590) has added 1x100G.

BGP Communities

- Basically labels attached to BGP messages
  - Very common trick we've seen in routing earlier
- Few with predefined meanings
  - NO_EXPORT (0xFFFFFF01) -> Advertise only within AS
  - NO_ADVERTISE (0xFFFFFF02) -> Don't advertise at all
  - NO_EXPORT_SUBCONFED (0xFFFFFF03) -> Advertise only with subconfederation
  - NOPEER (0xFFFFFF04) -> "Need not" advertise to peers

BGP Communities

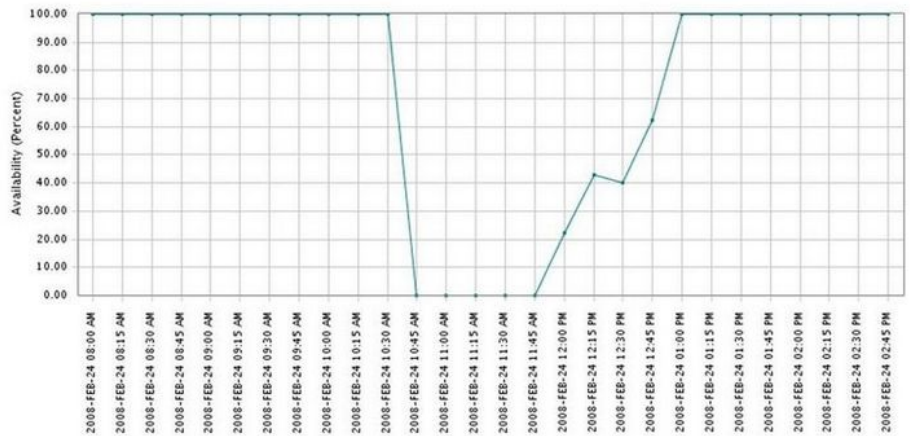
- User defined BGP communities
  - Can be anything, mostly define specific routes
    - e.g., “This route is through ATT Canada”
  - e.g, :3356:2003 (AS 3356 says 2003)
- Provides a mechanism for “prioritizing” BGP routes
  - Backups
    - Send to 3353:2003 instead of 3353:150 for some reason
  - Blackholing - want to blackhole nearest the AS
    - send 3353:9999 to indicate that peer needs to blackhole 3353

CULTURE

How Pakistan knocked YouTube offline (and how to make sure it never happens again)

YouTube becoming unreachable isn't the first time that Internet addresses were hijacked. But if it spurs interest in better security, it may be the last.

BY DECLAN MCCULLAGH | FEBRUARY 25, 2008 4:28 PM PST



This graph that network-monitoring firm Keynote Systems provided to us shows the worldwide availability of YouTube.com dropping dramatically from 100 percent to 0 percent for over an hour. It didn't recover completely until two hours had elapsed.

Keynote Systems

A high-profile incident this weekend in which Pakistan's state-owned telecommunications company managed to cut YouTube off the global Web highlights a long-standing security weakness in the way the Internet is managed.

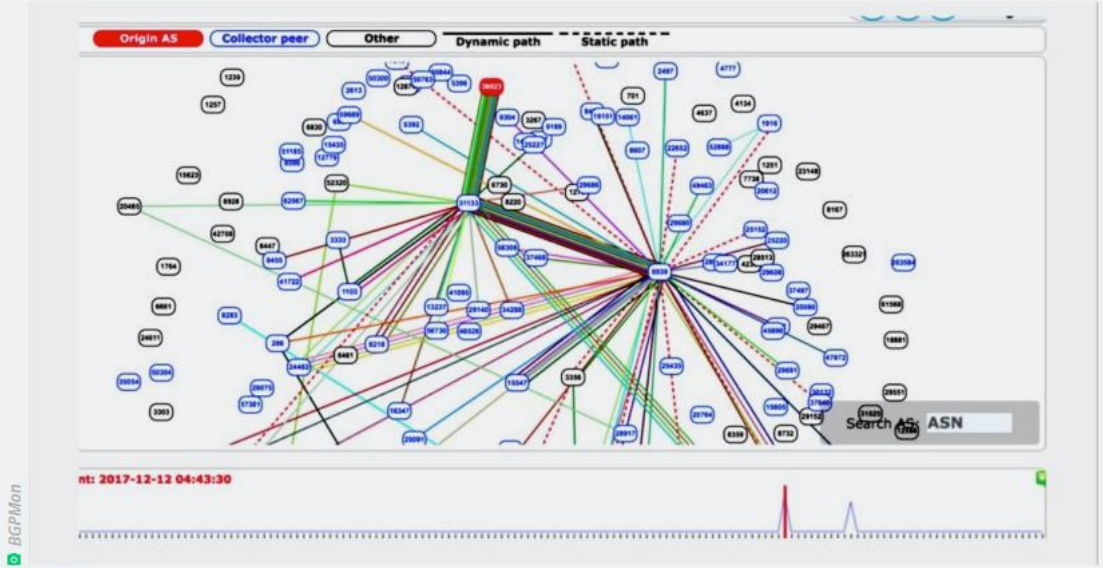
After receiving a censorship order from the telecommunications ministry directing that YouTube.com be blocked, Pakistan Telecom went even further. By accident or

BIZ & IT —

“Suspicious” event routes traffic for big-name sites through Russia

Google, Facebook, Apple, and Microsoft all affected by “intentional” BGP mishap.

DAN GOODIN - 12/13/2017, 2:43 PM



Enlarge

104

Traffic sent to and from Google, Facebook, Apple, and Microsoft was briefly routed through a previously unknown Russian Internet provider Wednesday under circumstances researchers said was suspicious and intentional.

FURTHER READING
Russian-controlled telecom hijacks financial services’ Internet traffic

The unexplained incident involving the Internet's Border Gateway Protocol is the latest to raise troubling questions about the trust and reliability of communications sent over the global network. BGP routes large-scale amounts of traffic among Internet backbones, ISPs, and other large networks. But despite the sensitivity and amount of data it controls, BGP's security is often based on trust and word of mouth. Wednesday's event comes eight months after large chunks of network traffic belonging to MasterCard, Visa, and more than two dozen other financial services were briefly routed through a Russian government-

SECURITY

31

Apple network traffic takes mysterious detour through Russia

Land of Putin capable of attacking routes in cyberspace as well as real world

Thomas Claburn

Wed 27 Jul 2022 // 18:56 UTC

Apple's internet traffic took an unwelcome detour through Russian networking equipment for about twelve hours between July 26 and July 27.

In a write-up for MANRS (Mutually Agreed Norms for Routing Security), a public interest group that looks after internet routing, Internet Society senior internet technology manager Aftab Siddiqui said that Russia's Rostelecom started announcing routes for part of Apple's network on Tuesday, a practice referred to as BGP (Border Gateway Protocol) hijacking.

BGP is the glue that links multiple networks together to form the internet. Unfortunately, the protocol is too credulous. When an autonomous system (AS) – a group of networks managed by a single entity – announces routes for groups of IP addresses (IP prefixes) that it does not own, internet traffic will generally adapt to those routes if the rogue announcement isn't filtered out.

Some bad route announcements are accidental and a result of something like a configuration blunder, and some announcements are straight-up malicious.

For example, in 2018 cyberthieves used BGP hijacking to meddle with Amazon's Route 53 DNS service and redirect internet traffic from a cryptocurrency website to a phishing site hosted in Russia.

The redirection of Apple's networking traffic began about 2125 UTC on Tuesday, according to Siddiqui, when Rostelecom's AS12389 network began announcing 17.70.96.0/19, which is part of Apple's 17.0.0.0/8 block. The /19 block is usually announced as part of Apple's 17.0.0.0/9 range, according to MANRS.

MORE CONTEXT

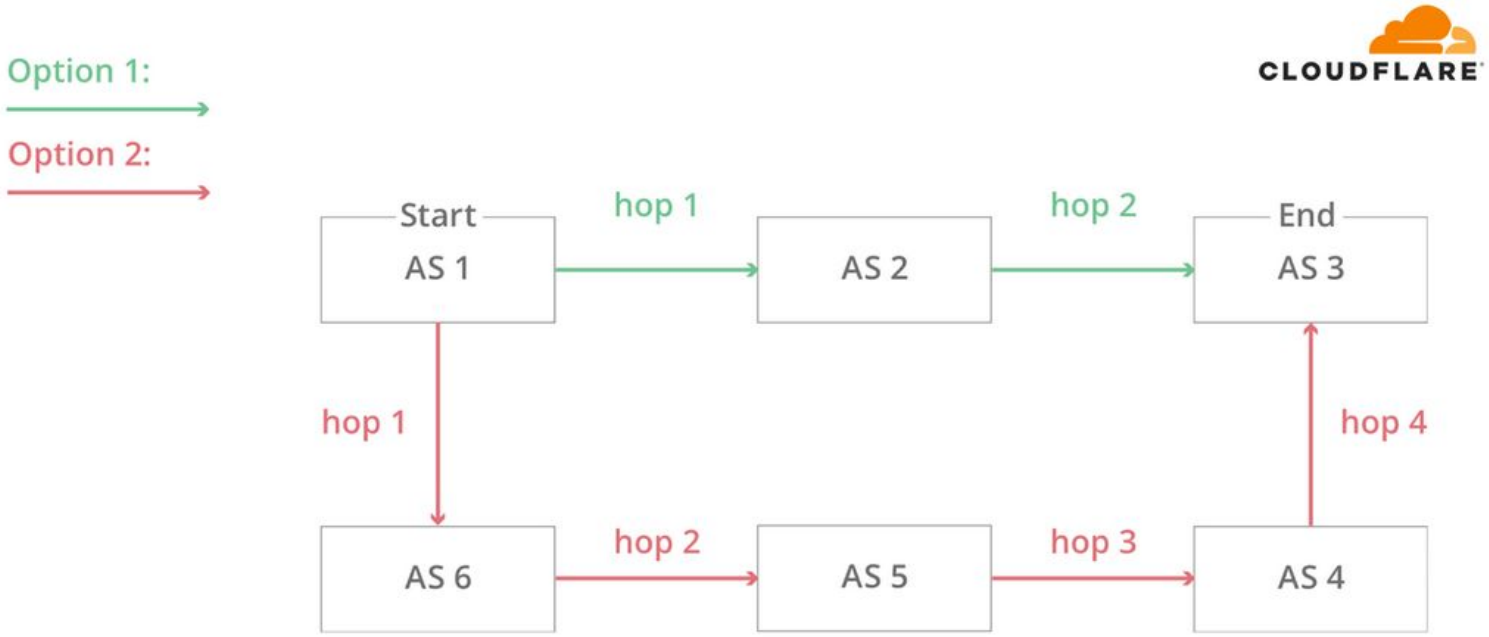
After config error takes down Rogers, it promises to spend billions on reliability

Cloudflare's outage was human error. There's a way to make tech divinely forgive

Big Tech's private networks and protocols threaten the 'net, say internet registries

Facebook rendered spineless by buggy audit code that missed catastrophic network config error

that can be used if the first fails.



At 15:58 UTC we noticed that Facebook had stopped announcing the routes to their DNS prefixes. That meant that, at least, Facebook's DNS servers were unavailable. Because of this Cloudflare's 1.1.1.1 DNS resolver could no longer respond to queries asking for the IP address of facebook.com.

```
route-views>show ip bgp 185.89.218.0/23
% Network not in table
route-views>

route-views>show ip bgp 129.134.30.0/23
% Network not in table
route-views>
```

Meanwhile, other Facebook IP addresses remained routed but weren't particularly useful since without DNS Facebook and related services were effectively unavailable:

Our Culture

PeeringDB, as the name suggests, was set up to facilitate peering between networks and peering coordinators. In recent years, the vision of PeeringDB has developed to keep up with the speed and diverse manner in which the Internet is growing. The database is no longer just for peering and peering related information. It now includes all types of interconnection data for networks, clouds, services, and enterprise, as well as interconnection facilities that are developing at the edge of the Internet.

We believe in, and rely on the community to grow and improve the PeeringDB database. The volunteers who run the database are passionate about security, privacy, integrity, and validation of the data in the database. Even though PeeringDB is a freely available and public tool, users strictly adhere to the acceptable use policy, which prevents the database being used for commercial purposes and discourages unsolicited communications. This is largely policed by the community and has been very effective since PeeringDB was launched.

I'm a network operator. How can PeeringDB help me?

Almost one-third of Autonomous System Numbers (ASNs) register their interconnection data in the PeeringDB database. That means, by using PeeringDB and adding your own interconnection data, you'll be able to confidently find information about networks looking to interconnect, where and how to connect with them, and they'll be able to find the same information about your network. Since the database is user-maintained and validated by our volunteers, you can trust that the information is accurate and up-to-date.

This data will help you to accelerate the process of finding and connecting with other networks, while supporting a faster and more decisive deployment of your own network expansion and development plans.

I'm an Internet Exchange Point (IXP), data center or other interconnection facility. How can PeeringDB help me?

IXPs and data center facilities can add to and maintain their information in the database, increasing visibility and their appeal to new and existing customers. If you're in the database, this makes it much easier for networks to find crucial information about your services and the other networks present at your IXP or facilities.

BGP Thoughts

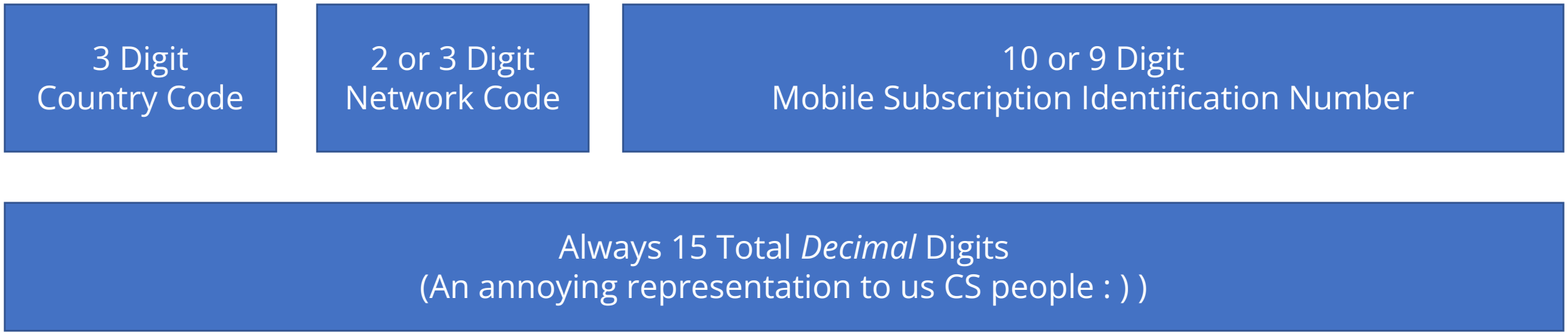
- Much more beyond basics to explore!
- Policy is a substantial factor
  - Can independent decisions be sensible overall?
- Other important factors:
  - Convergence effects
  - How well it scales
  - Integration with intradomain routing
  - And more …

Cellular Routing

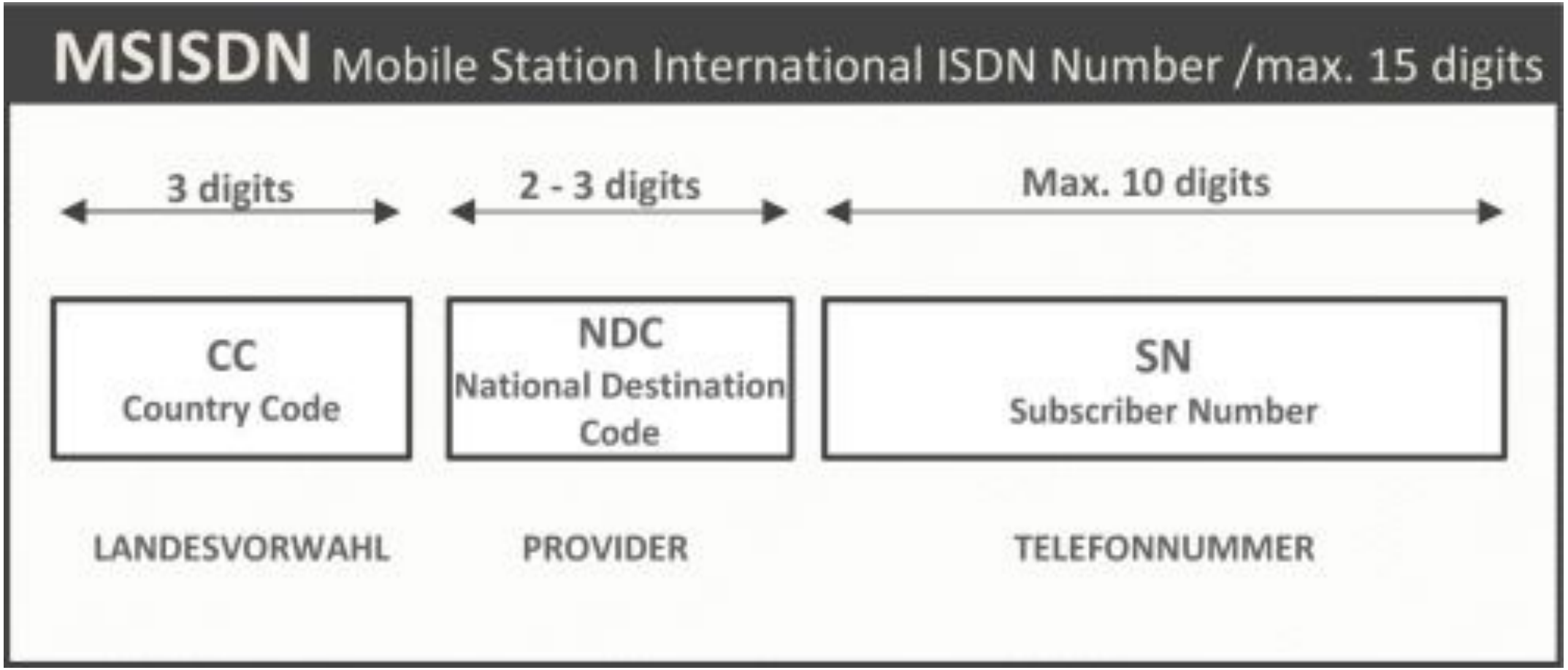
Addressing in Cellular

- Everyone has a unique physical identifier: SIM Card
  - IMSI: “International Mobile Subscriber Identity”
  - Has associated mobile provider
  - Has K_i secret auth key
  - Phone number **not** present
    - Known as “msisdn”

IMSI: identifier per SIM



MSISDN

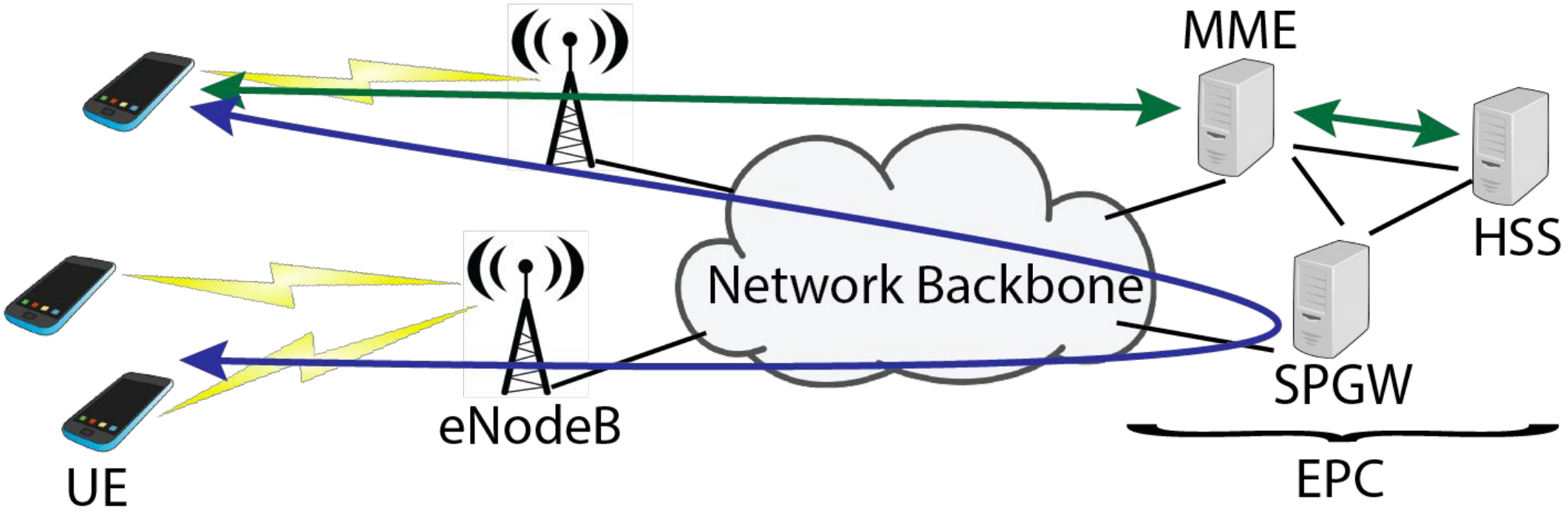

MSISDN Mobile Station International ISDN Number /max. 15 digits
3 digits
2 - 3 digits
Max. 10 digits
CC
Country Code
NDC
National Destination Code
SN
Subscriber Number
LANDESVORWAHL
PROVIDER
TELEFONNUMMER

Question…

- Why use two identifiers (IMSI & MSISDN)?

- Backwards compatibility! MSISDN shared with fixed line phone network
- Allows business-level mapping between phone # and actual sim…
  - Can keep your phone number when you lose/upgrade your phone!
  - But opens the door to “social engineering” sim-swap attacks : (

Cellular Core Networks



In-network routing

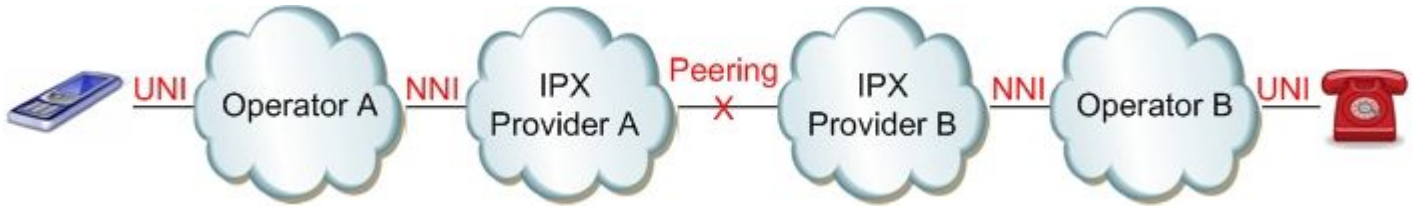
1. User dials phone number
2. Number is “looked up” in some database
   - If “in network” -> HSS/HLR
   - If “out of network”, see next slide
3. If local, we get the associated IMSI
4. Check that sender and send and receiver can receive
5. Look up tower group of IMSIs last registration
6. Page the receiver
7. Bill them both

Out-of-network Routing

- Signaling System No. 7 (SS7)
  - Performs number translation, local number portability, prepaid billing, Short Message Service (SMS), roaming, and other stuff
  - Either directly connected or connected through aggregators such as Sybase
  - Business vs Protocols

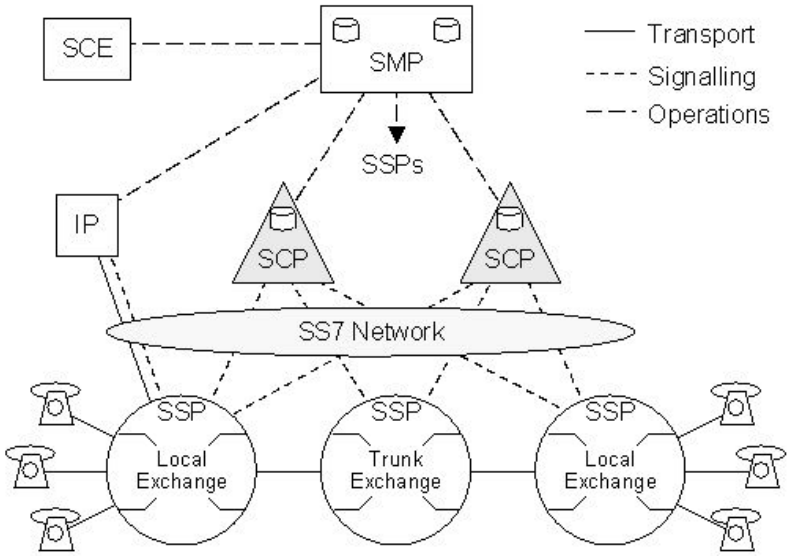
Out-of-network Routing

- IP Exchange (IPX)
  - Cellular equivalent of IXP
  - Interconnect for IP-based telecommunications
    - e.g., Voice-over-IP (VoIP)



Cellular Lookups

- An SSP telephone exchange receives a call to an 0800 number. This causes a trigger within the SSP that causes an SCP (Service Control Point) to be queried using SS7 protocols (INAP, TCAP). The SCP responds with a geographic number, e.g. 0121 XXX XXXX, and the call is actually routed to a phone.



In small groups…

- What is one advantage of the telephone way of doing things relative to what we saw with BGP?
- What is one advantage of BGP?

Some food for thought:
- *Which network has become more relevant?*
- *Do the architecture and affordances of a network influence how it grows and develops?*