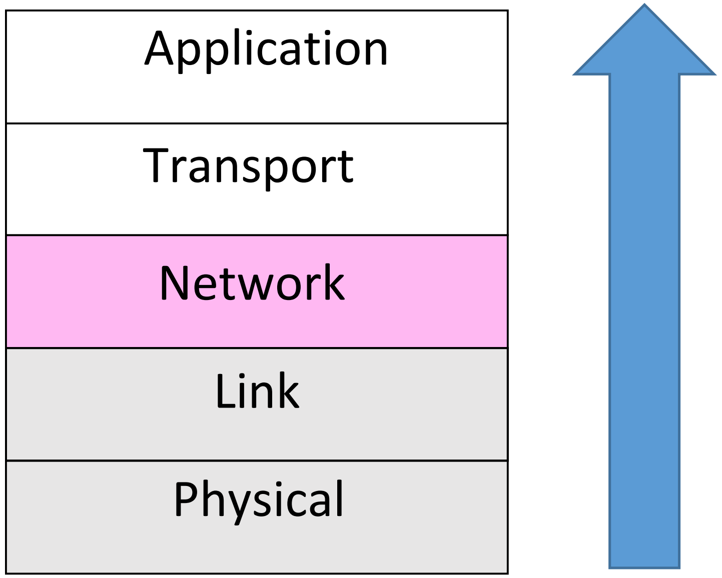
Network Layer (Routing)

Where we are in the Course

- Moving on up to the Network Layer!

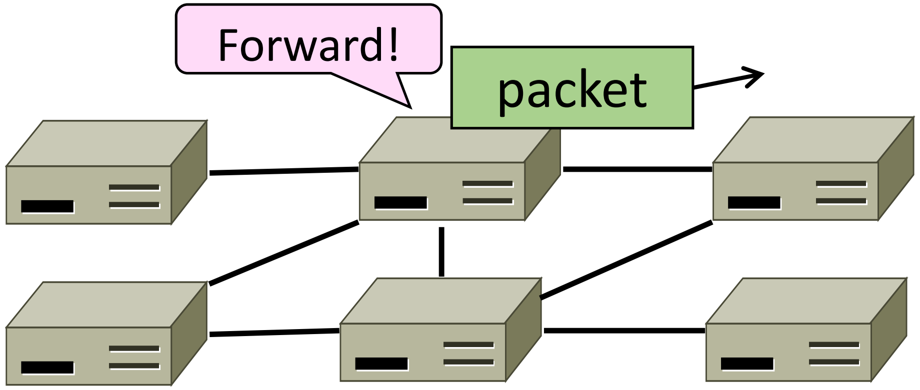


Topics

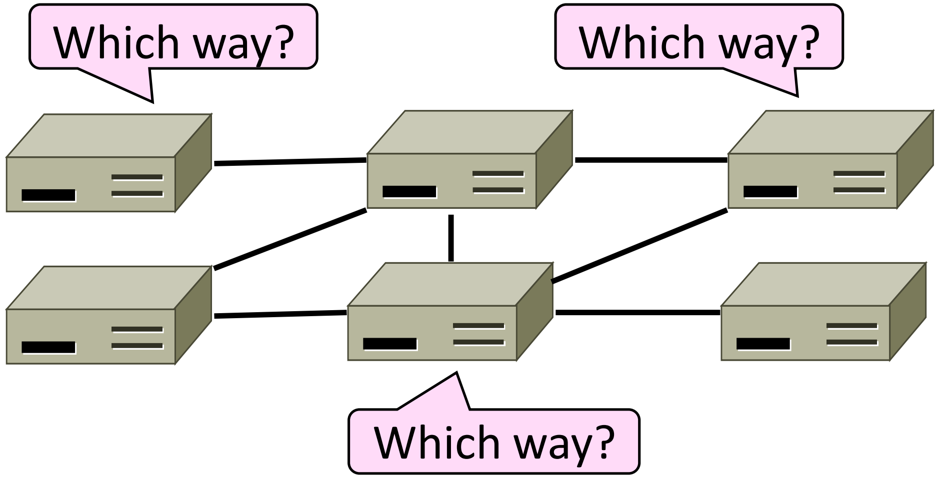
- Network service models
  - Datagrams (packets), virtual circuits
- IP (Internet Protocol)
  - Internetworking
  - Forwarding (Longest Matching Prefix)
  - Helpers: ARP and DHCP
  - Fragmentation and MTU discovery
  - Errors: ICMP (traceroute!)
  - IPv6, scaling IP to the world
  - NAT, and “middleboxs”
- Routing Algorithms

Routing versus Forwarding

- <u>Forwarding</u> is the process of sending a packet on its way

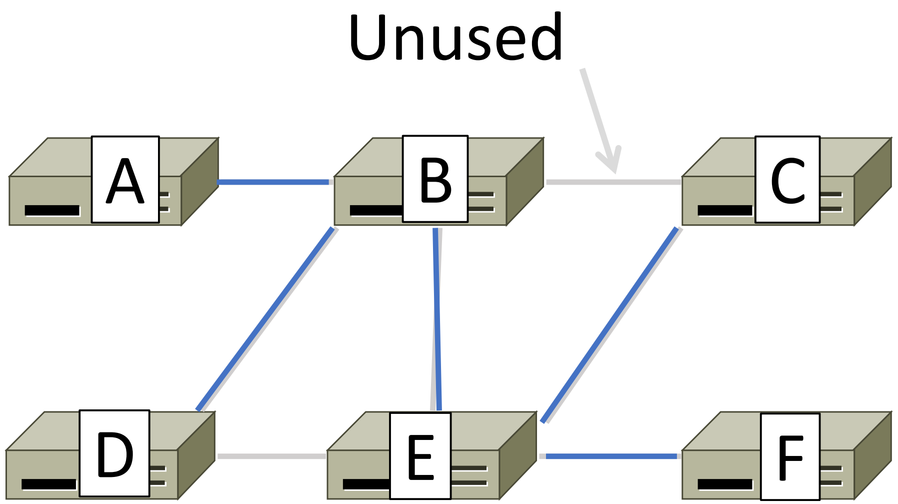


- <u>Routing</u> is the process of deciding in which direction to send traffic

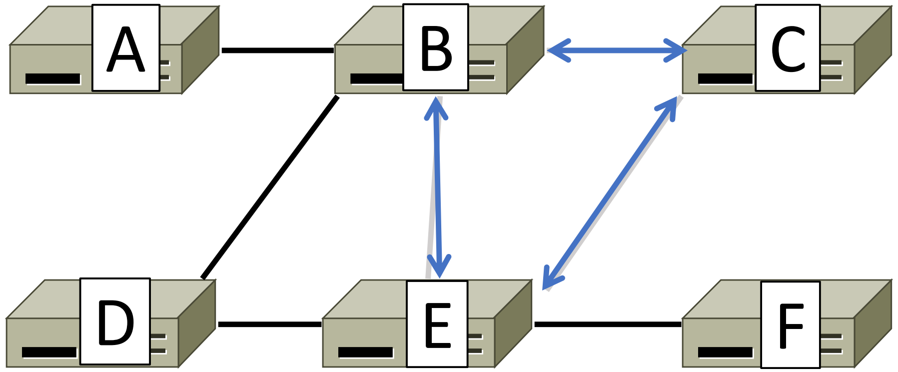


Improving on the Spanning Tree

- 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

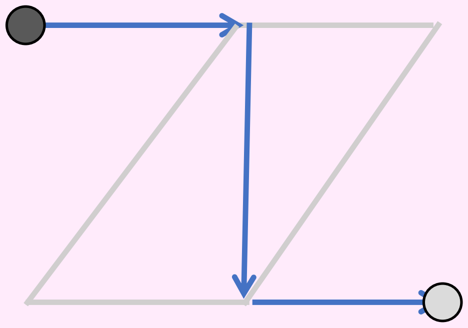
- Routing allocates network bandwidth adapting to failures; other mechanisms used at other timescales

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

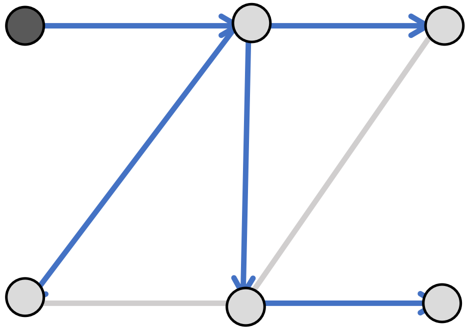
Delivery Models

- Different routing used for different delivery models

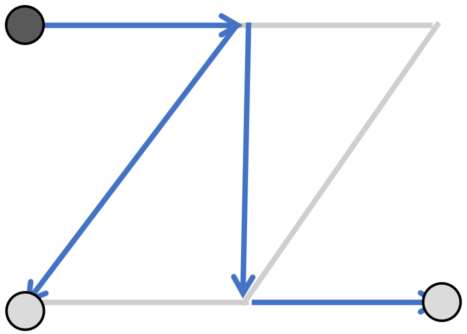
Unicast
(§5.2)



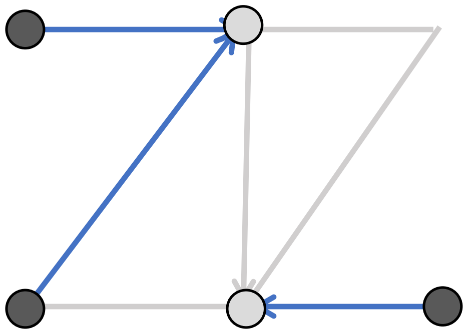
Broadcast
(§5.2.7)



Multicast
(§5.2.8)



Anycast
(§5.2.9)



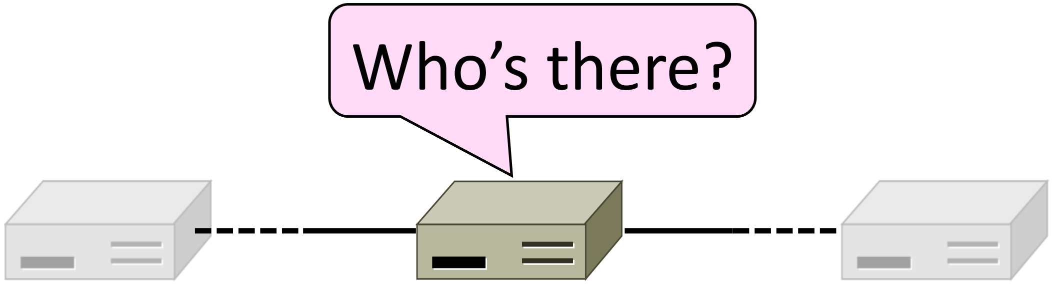
Goals of Routing Algorithms

- 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 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

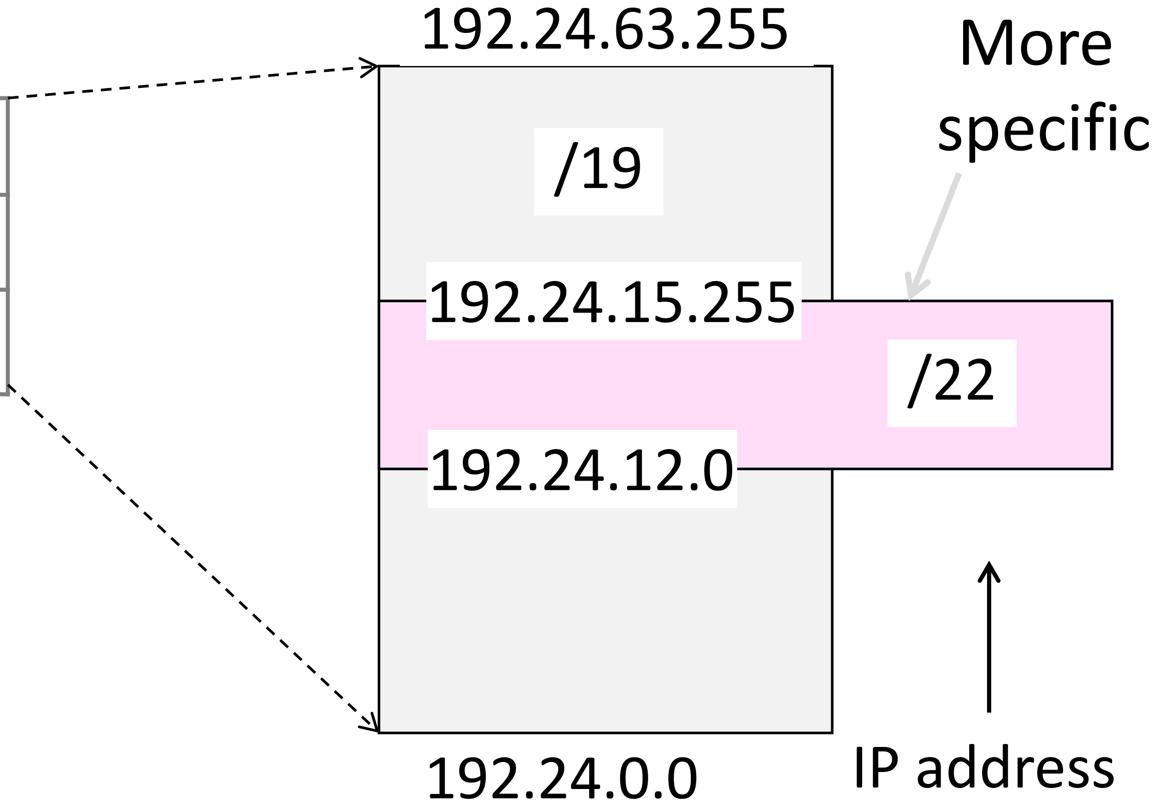


Recap: Classless Inter-Domain Routing (CIDR)

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

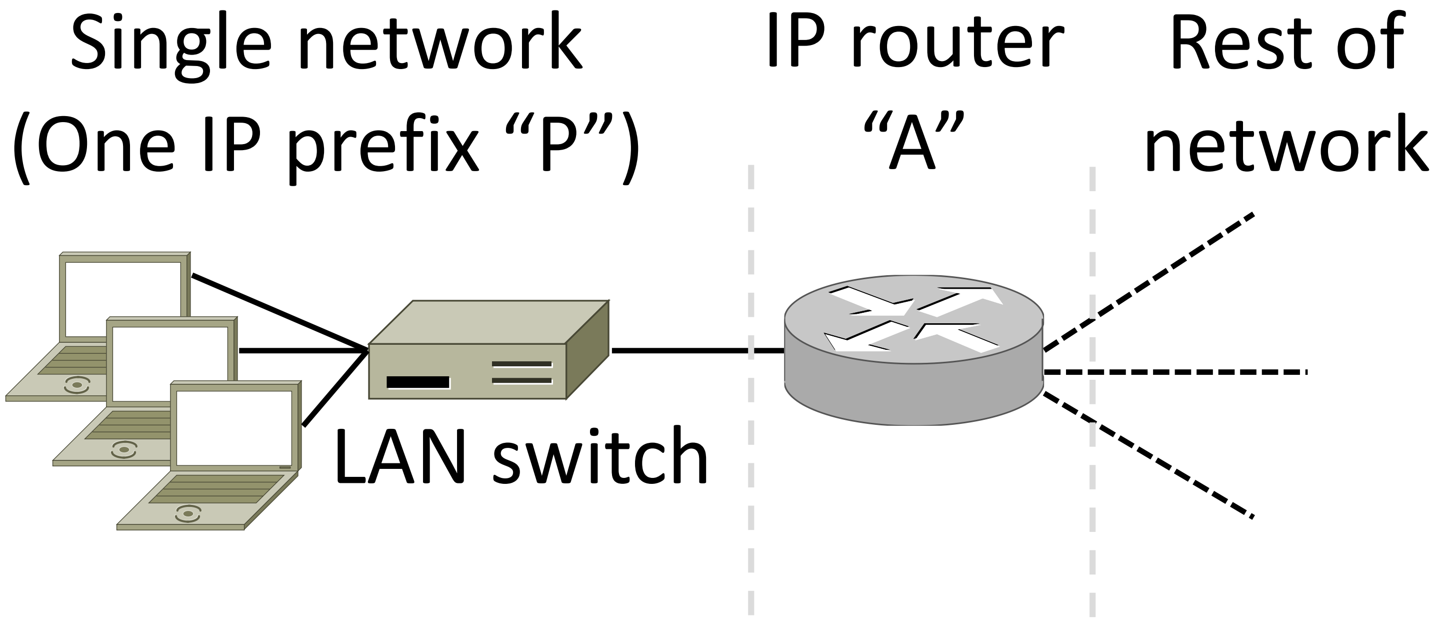
Longest Matching Prefix

Prefix	Next Hop
192.24.0.0/19	D
192.24.12.0/22	B



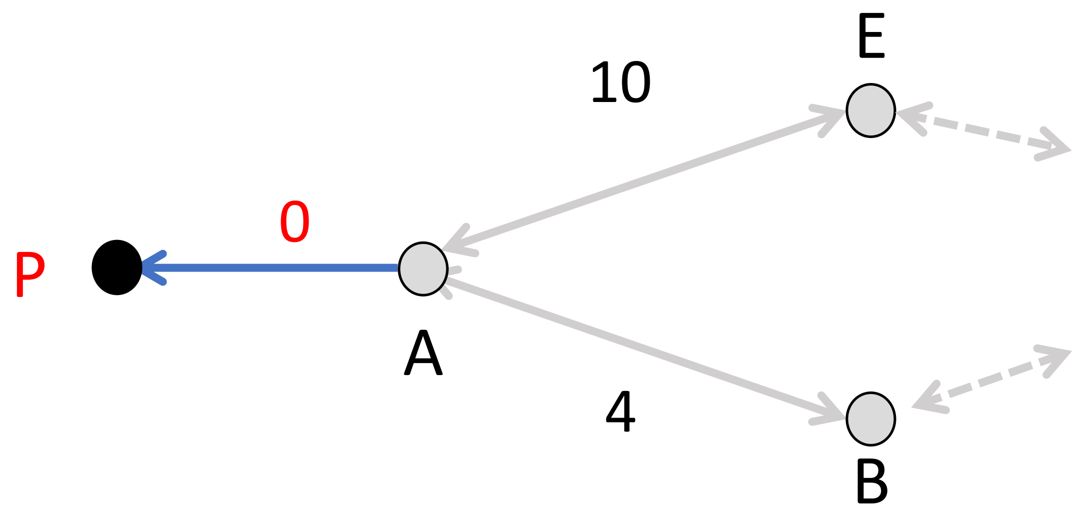
Host/Router Combination

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



Network Topology for Routing

- Group hosts under IP prefix connected to router
  - One entry for all hosts



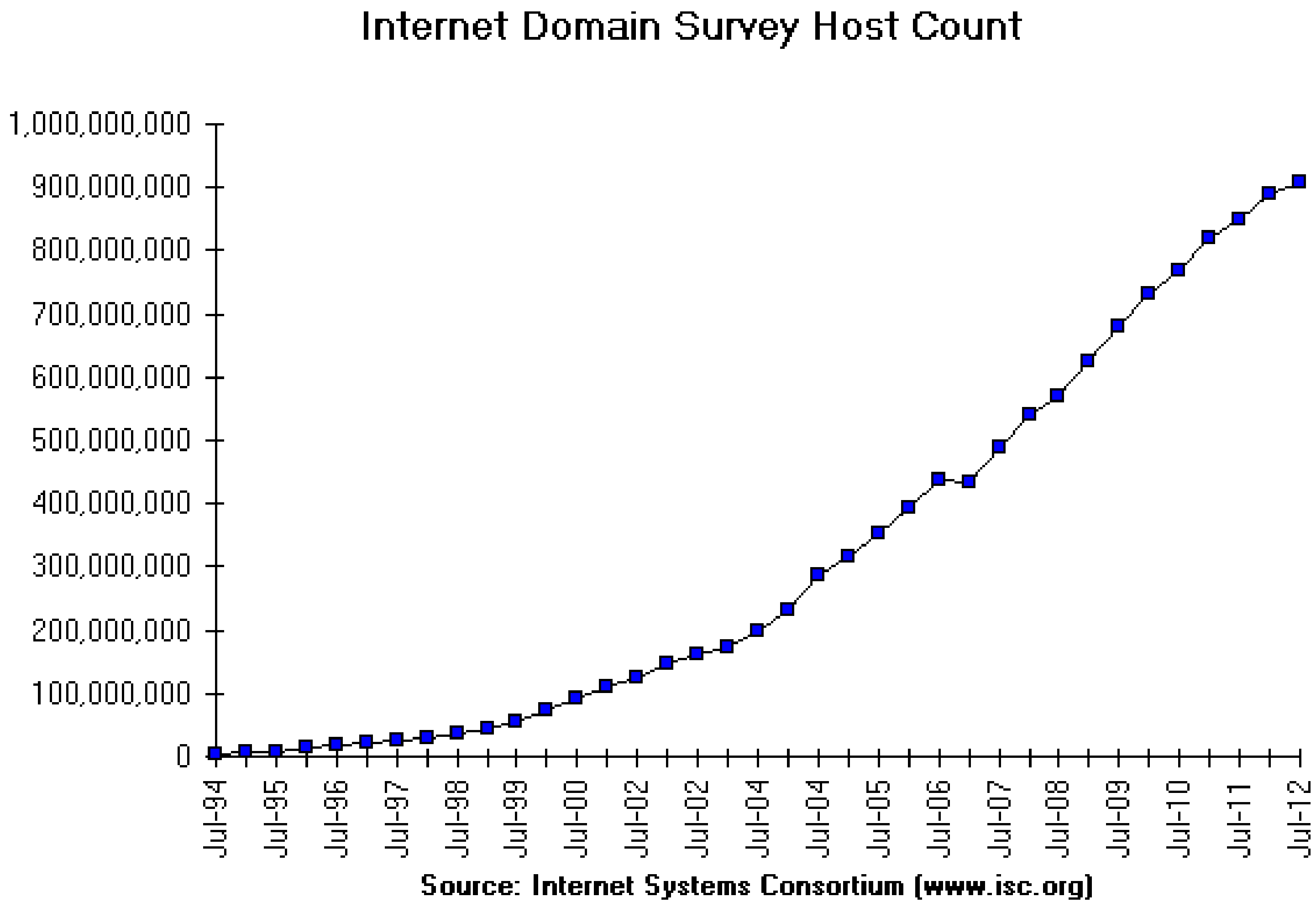
Network Topology for Routing (2)

- Routing now works!
  - Routers advertise IP prefixes for hosts
  - Router addresses are “/32” prefixes
  - Lets all routers find a path to hosts
  - Hosts find by sending to their router

Hierarchical Routing

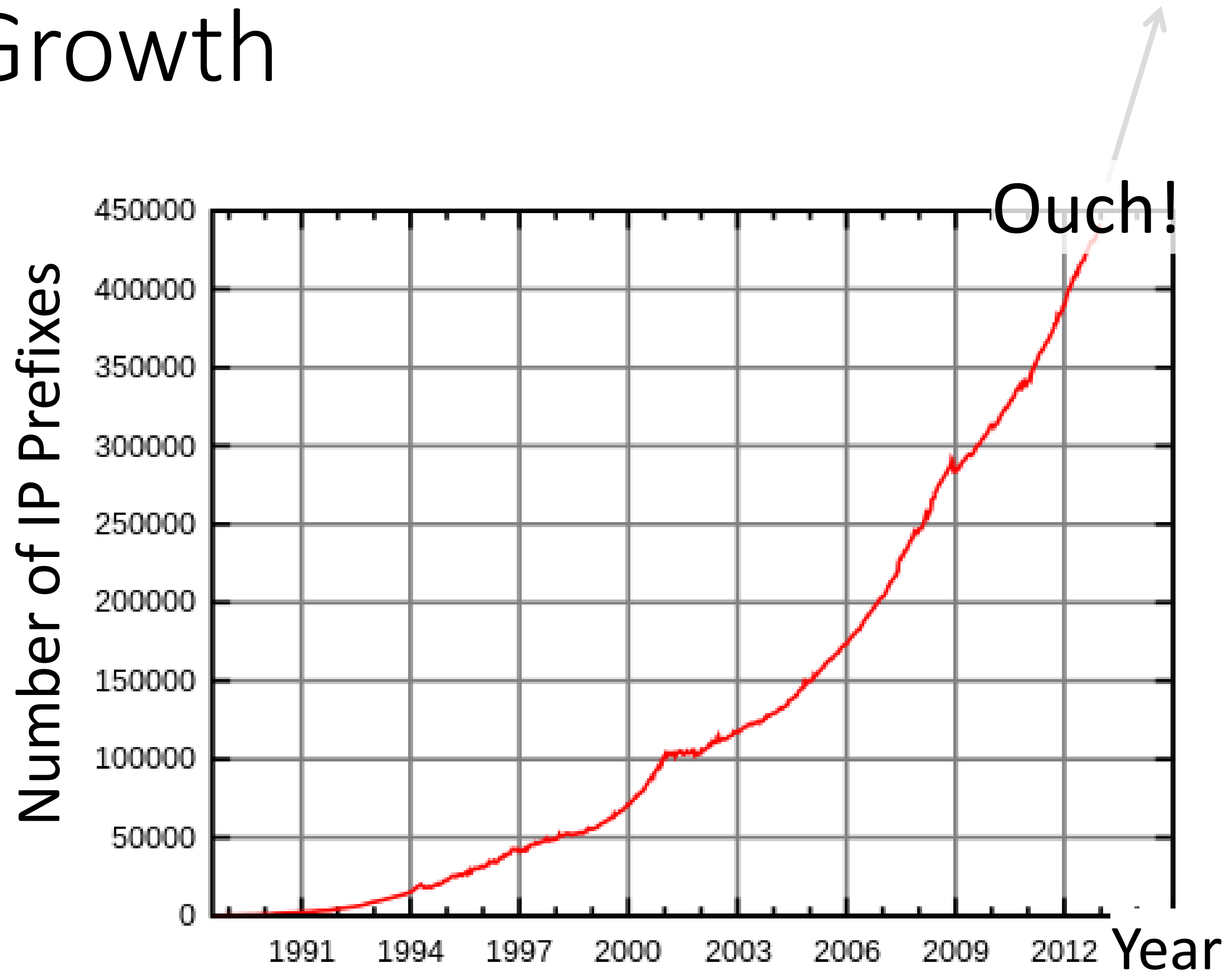
Internet Growth

- At least a billion 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

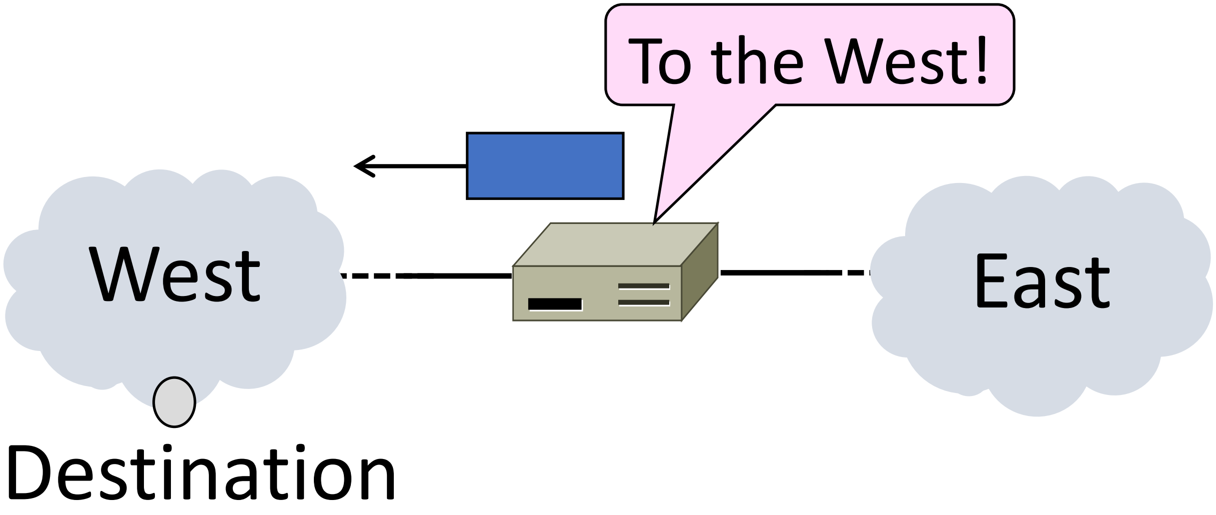
1. Forwarding tables grow
   - Larger router memories, may increase lookup time
2. Routing messages grow
   - Need to keeps all nodes informed of larger topology
3. Routing computation grows
   - Shortest path calculations grow faster than the network

Techniques to Scale Routing

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

Idea

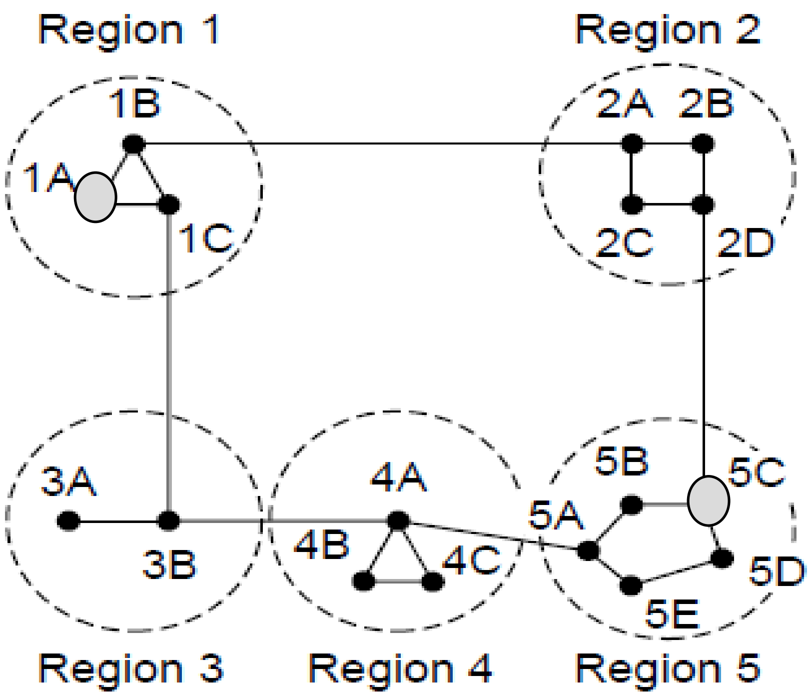
- Scale routing using hierarchy with regions
  - Route to regions, not individual nodes



Hierarchical Routing

- Introduce a larger routing unit
  - IP prefix (hosts) ← from one 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)



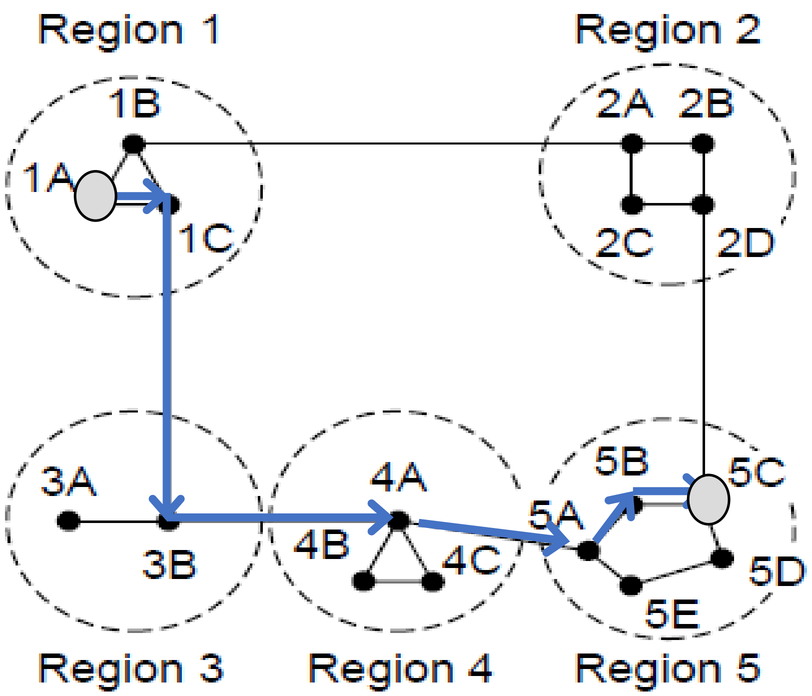
Full table for 1A

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

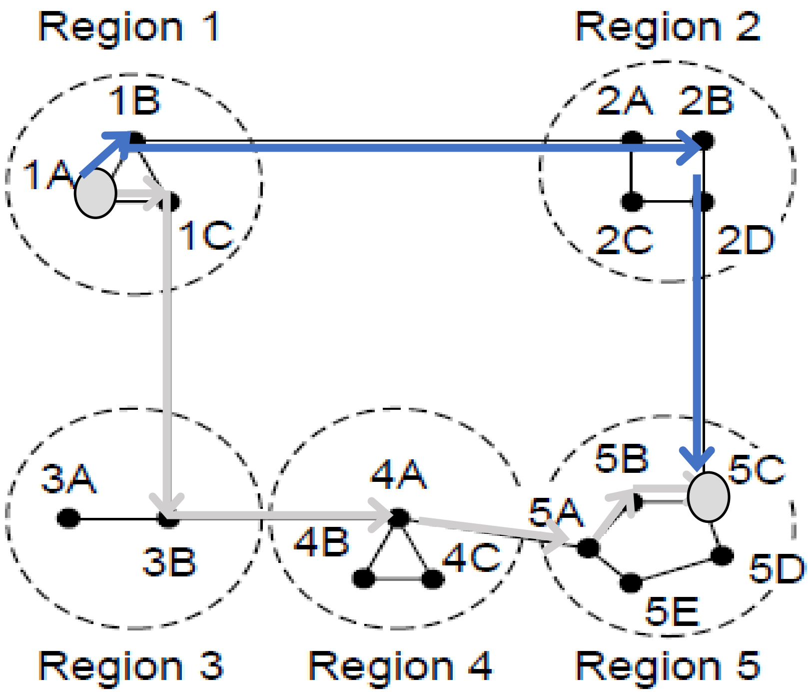
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

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 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

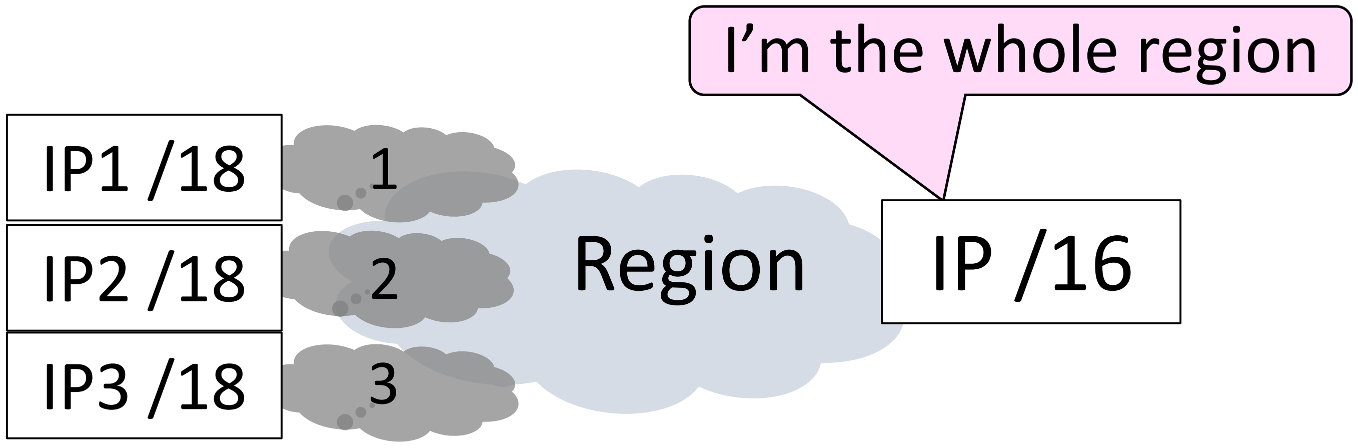
Observations

- 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)



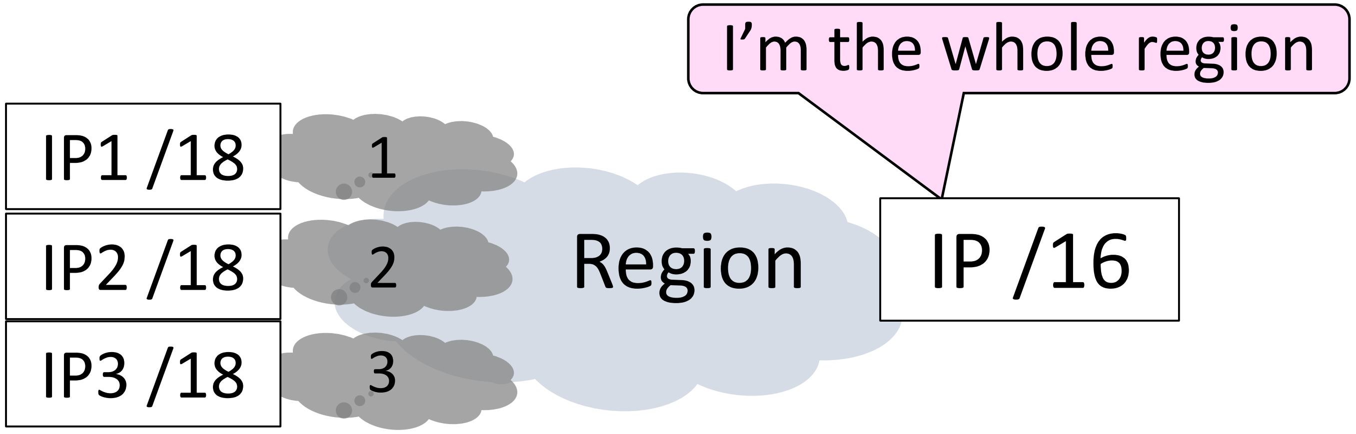
Recall

- 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 and contains 2^{32-N} addresses
  - E.g., a “/24” has 256 addresses
- 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



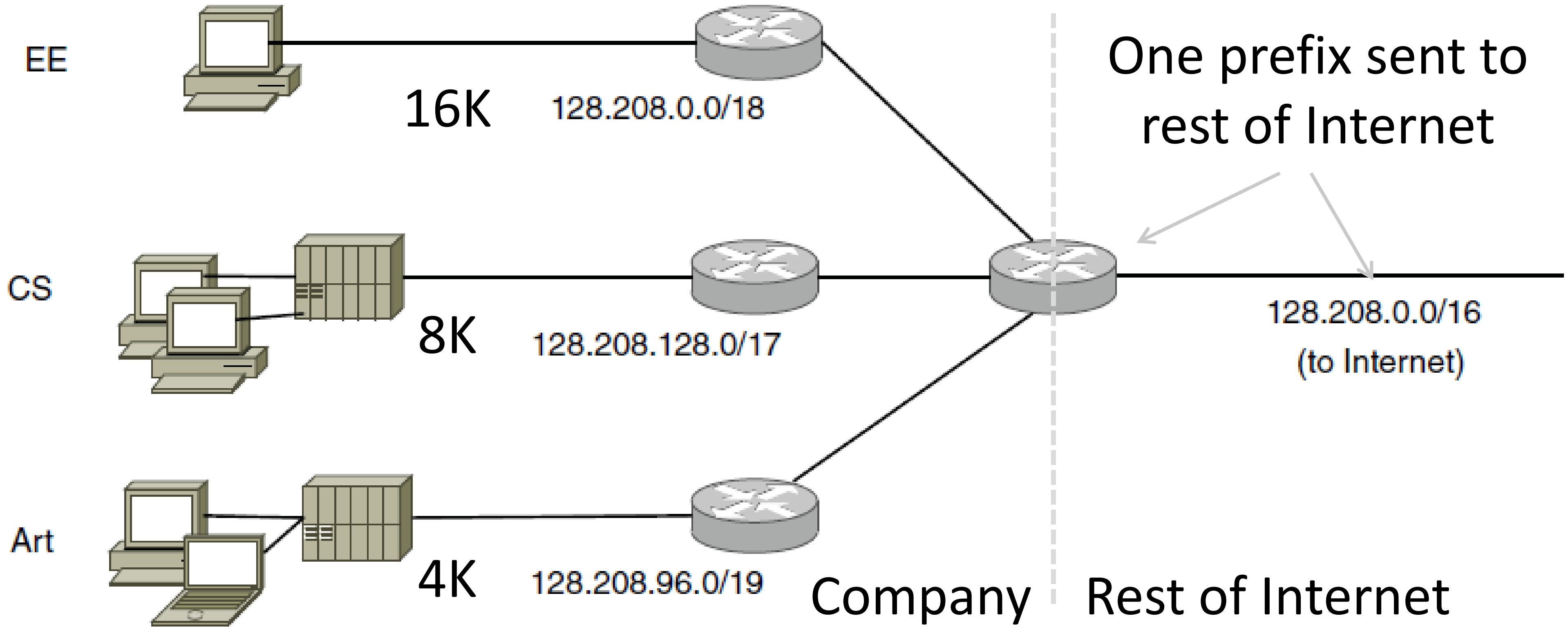
Subnets and Aggregation

- 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

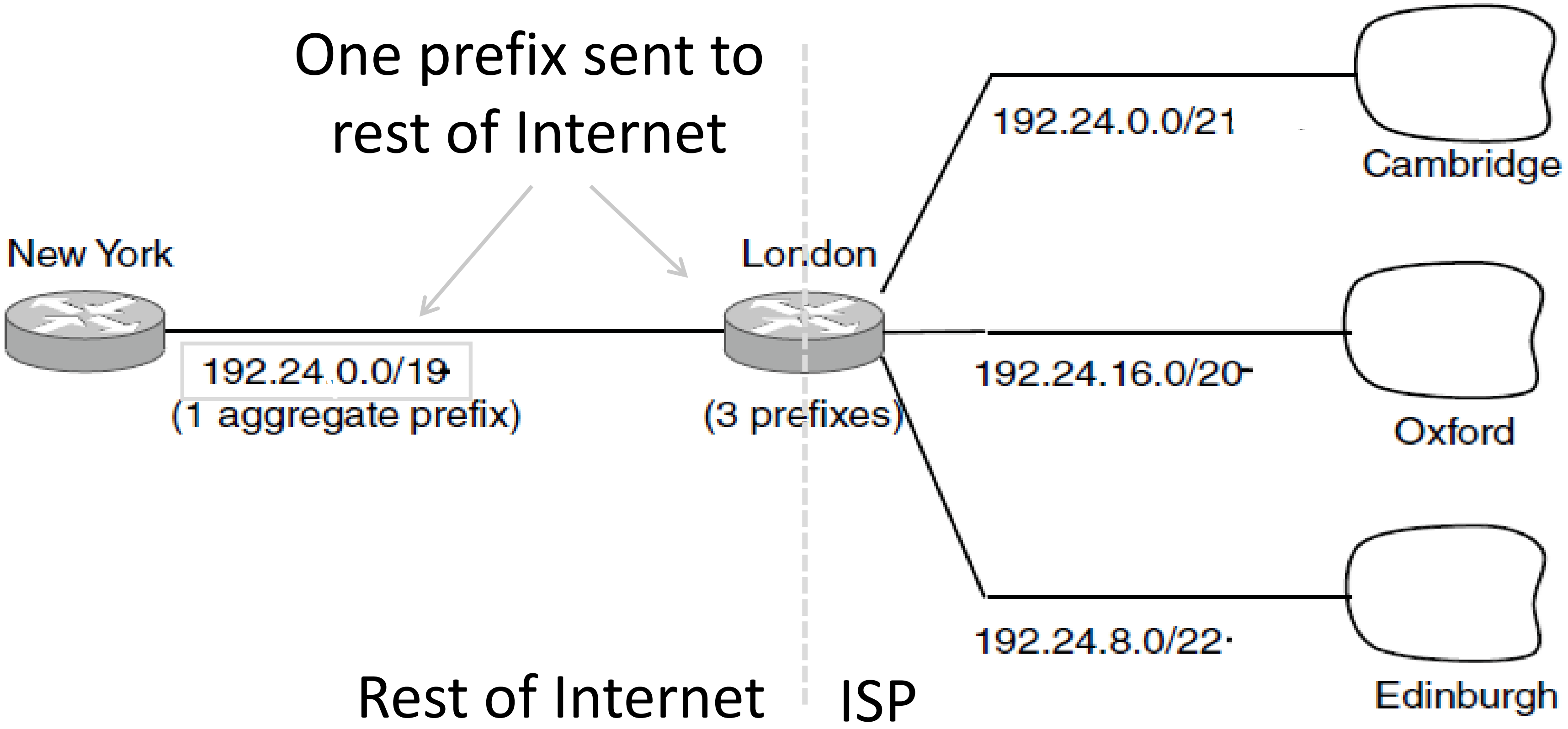
Subnets

- Internally split up one IP prefix



Aggregation

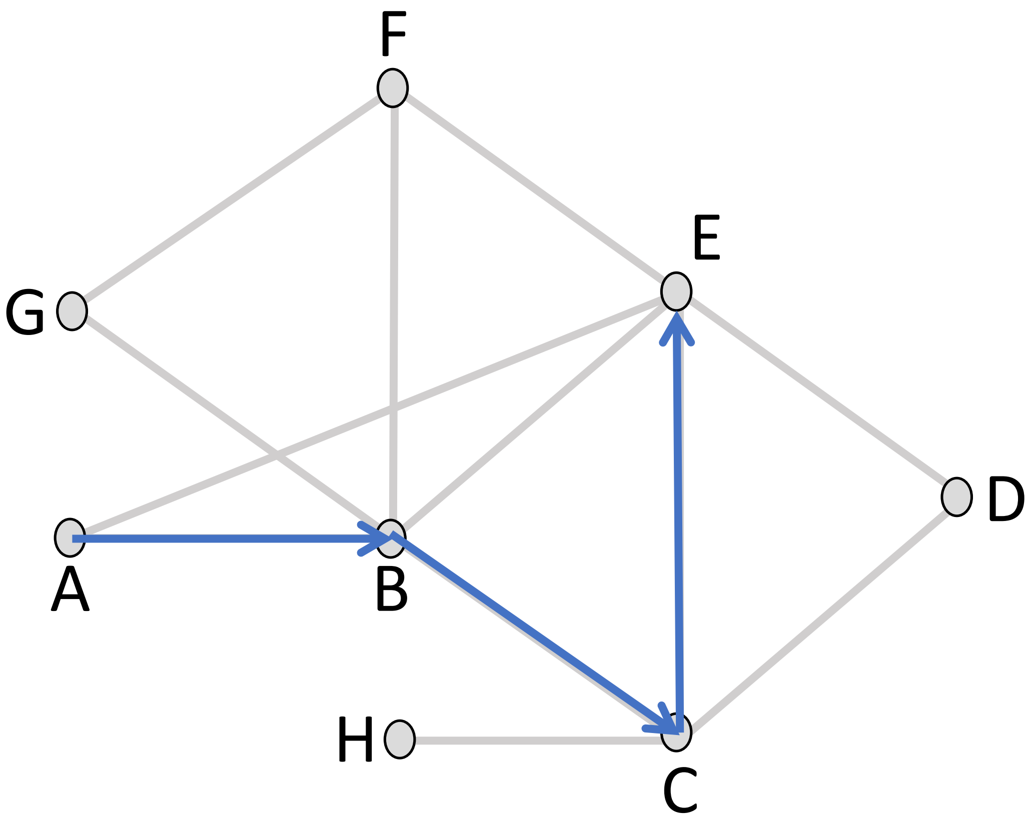
- Externally join multiple separate IP prefixes



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

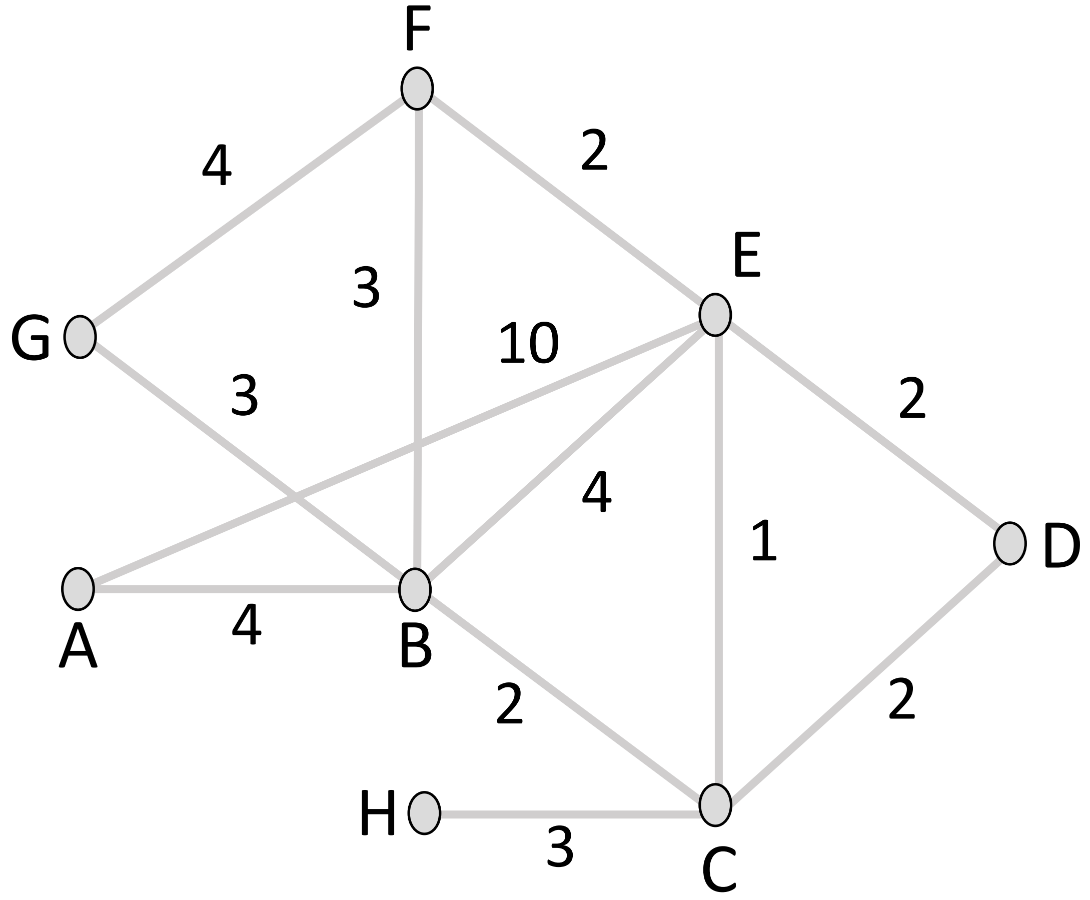
- Often call lowest "shortest"

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

Shortest Paths (2)

- Find the shortest path A → E

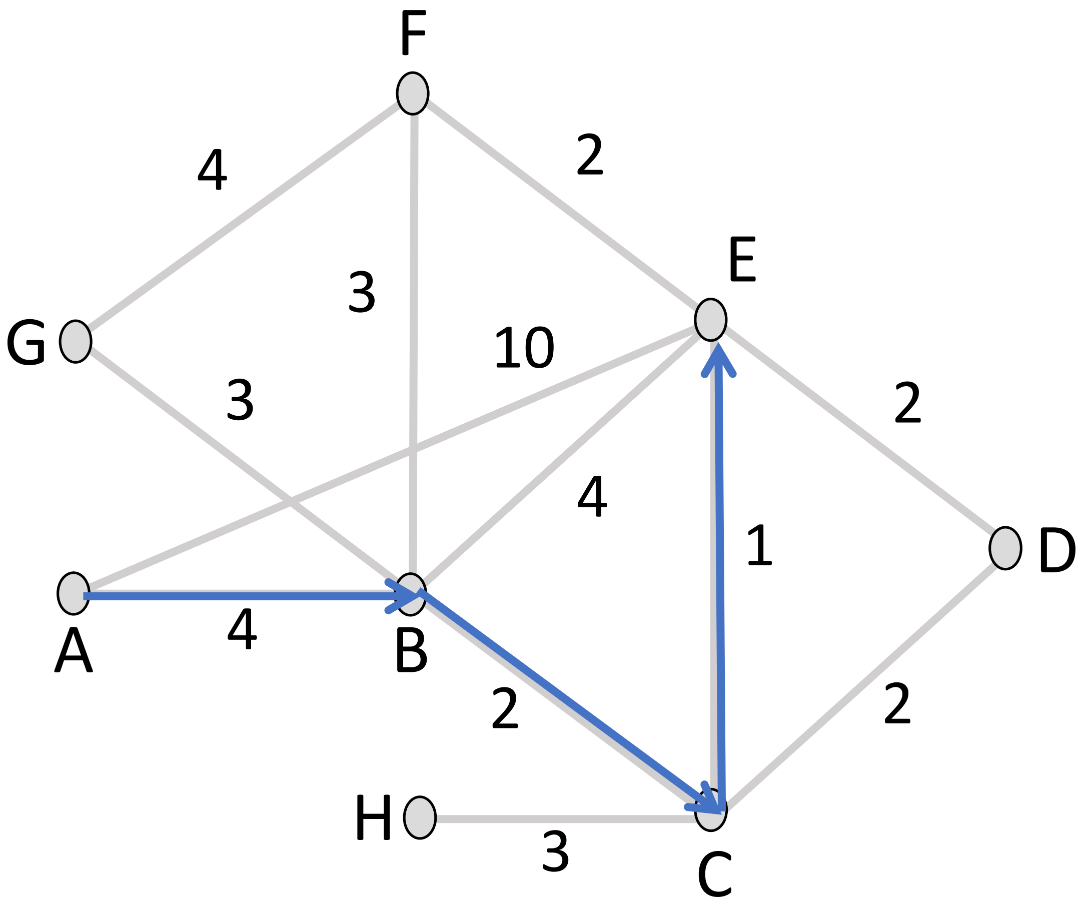
- All links are bidirectional, with equal costs in each direction
  - Can extend model to unequal costs if needed



Shortest Paths (3)

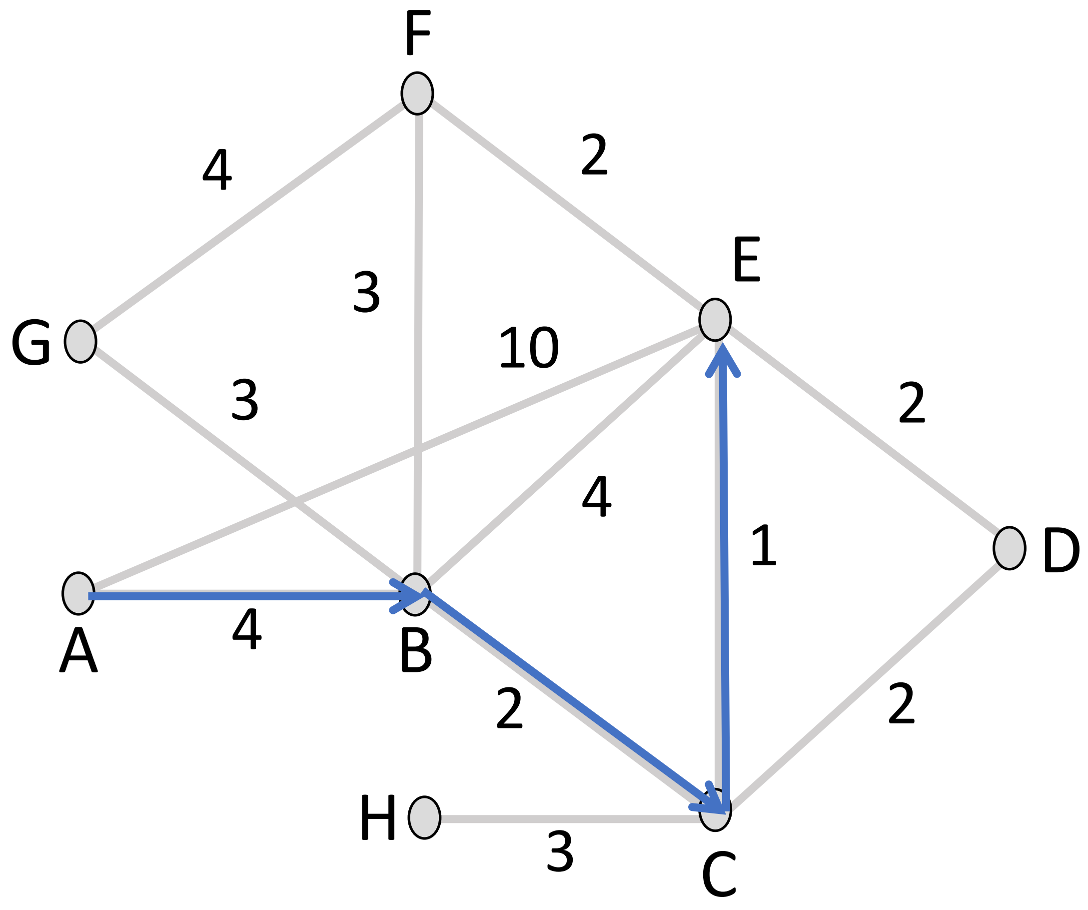
- ABCE is a shortest path
- dist(ABCE) = 4 + 2 + 1 = 7

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



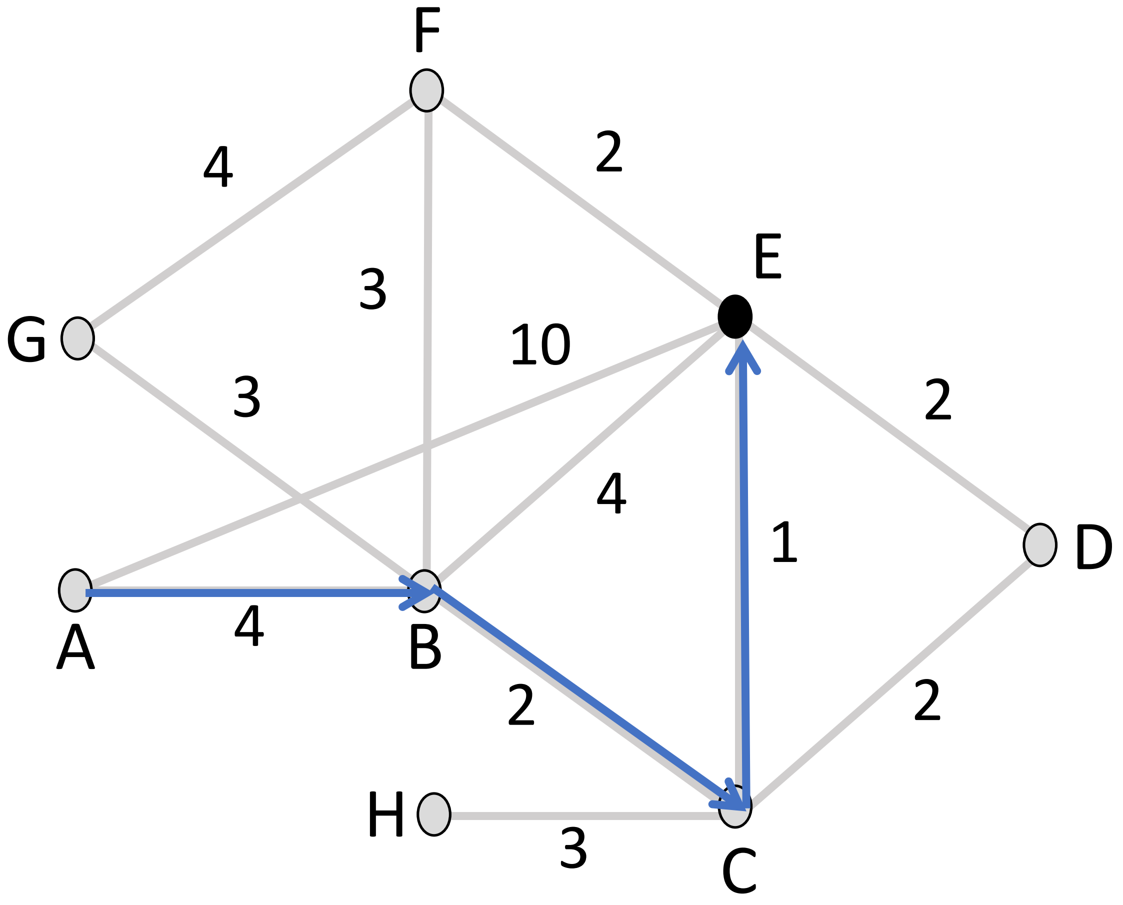
Shortest Paths (4)

- 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