CSE160: Computer Networks

Lecture #10 – Intra-Domain Routing

2020-09-28

Professor

Alberto E. Cerpa

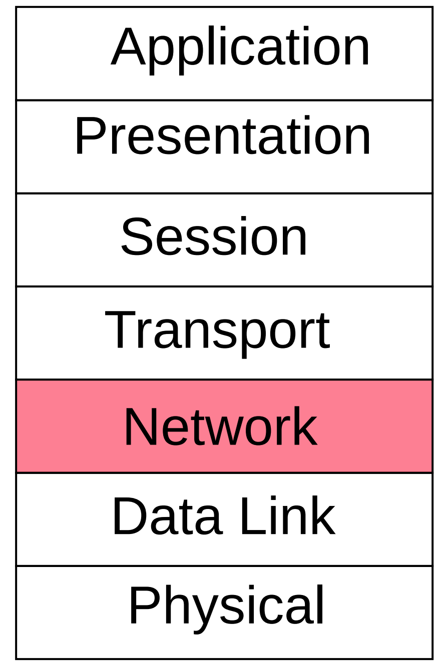
Last Time

- Focus
  - How do we calculate routes for packets?
  - Routing is a network layer function
- Routing Algorithms
  - Intro to routing
  - Best paths
  - Dijkstra SP Algorithm
  - Link-State routing (OSPF)
  - Cost metrics and cost estimation

Application
Presentation
Session
Transport
Network
Data Link
Physical

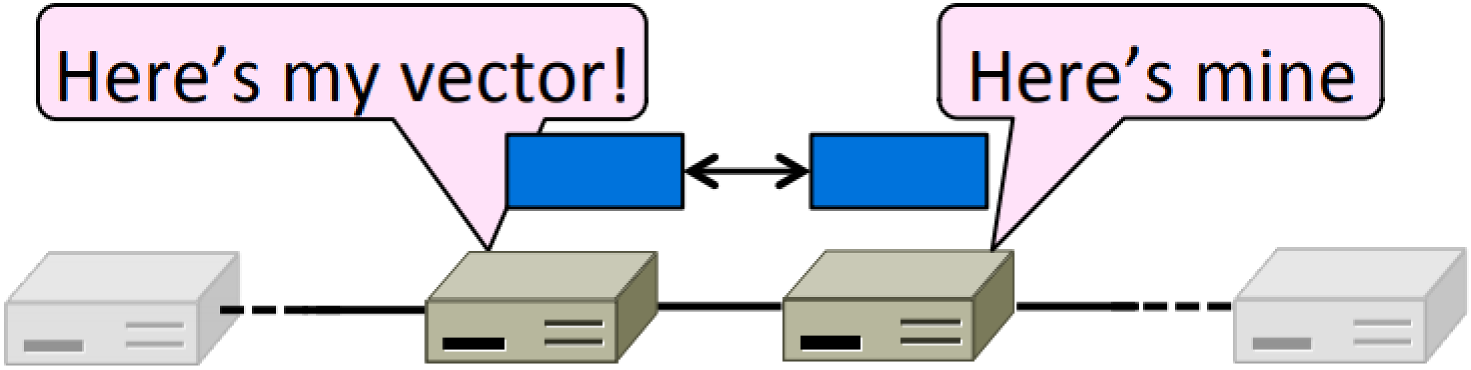
This Time

- Focus
  - How do we calculate routes for packets over single and multiple paths?
  - How do we get IP addresses and MAC addresses for a destination IP?
  - How do we get more IP addresses?
- Topics
  - Distance Vector routing (RIP)
  - Equal-Cost Multipath routing (ECMP)
  - Dynamic Host Configuration Protocol (DHCP)
  - Address Resolution Protocol (ARP)
  - IPv6



Distance Vector Routing

- How to compute shortest paths in a distributed network
  - The Distance Vector (DV) approach



Distance Vector

- 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

Why have two protocols?

- LS: "Tell the world about your neighbors."
  - Harder to get confused ("the nightly news")
  - More complicated
    - Faster convergence (instantaneous update of link state changes), but higher costs (flooding)
    - Able to impose global policies in a globally consistent way
      - Richer cost model, load balancing
- DV: "Tell your neighbors about the world."
  - Easy to get confused ("the telephone game")
  - Simple but limited, costly and slow
    - Better scaling properties but 15 hops is all you get in practice (makes it faster to loop to infinity)
    - Periodic broadcasts of large tables locally to neighbors
    - Slow convergence due to ripples and hold down

Distance Vector Routing

- Assume:
  – Each router knows only address/cost of neighbors
- Goal:
  – Calculate routing table of next hop information for each destination at each router
- Idea:
  – Tell *<u>neighbors</u>* about learned distances to *<u>all destinations</u>*

Distance Vector Setting

- Each node computes its forwarding table in a distributed setting:

1. Nodes know only the cost to their neighbors; not the 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

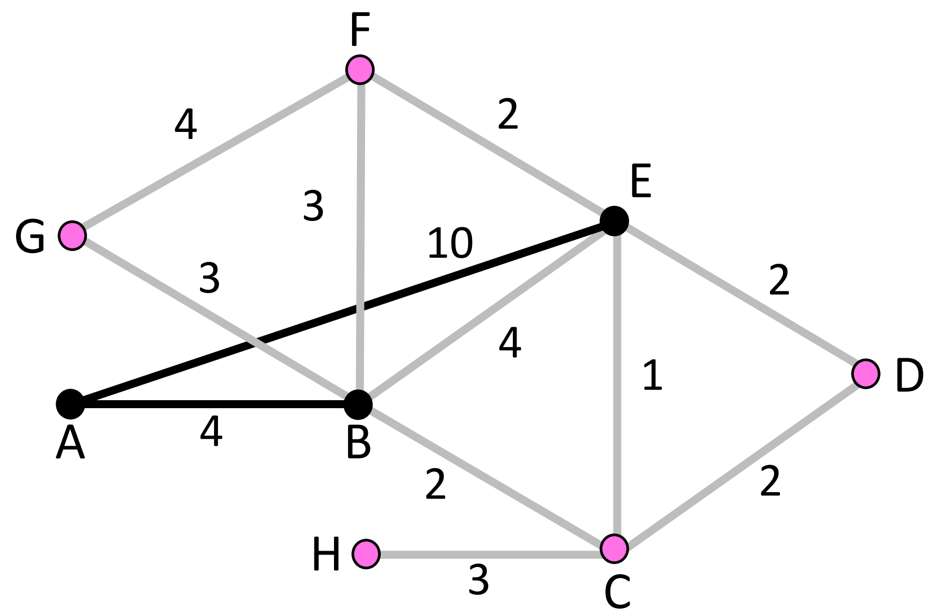
- Each node maintains a vector of distances (and next hops) to all destinations
  1. Initialize neighbors with known cost, others with infinity
  2. Periodically send copy of distance vector to neighbors
  3. On reception of a vector, if neighbors path to a destination plus neighbor cost is better, then switch to better path
     - update cost in vector and next hop in routing table
- Assuming no changes, it will converge to shortest paths
  - But what happens if there are changes?

Distance Vector Example

- Consider from the point of view of node A
  - Can only talk to nodes B and E (assumes no neighbor discovery running)

Initial vector

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



Distance Vector Example (2)

- First exchange with B, E; learn best 1-hop routes
  - this info is available with neighbor discovery

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

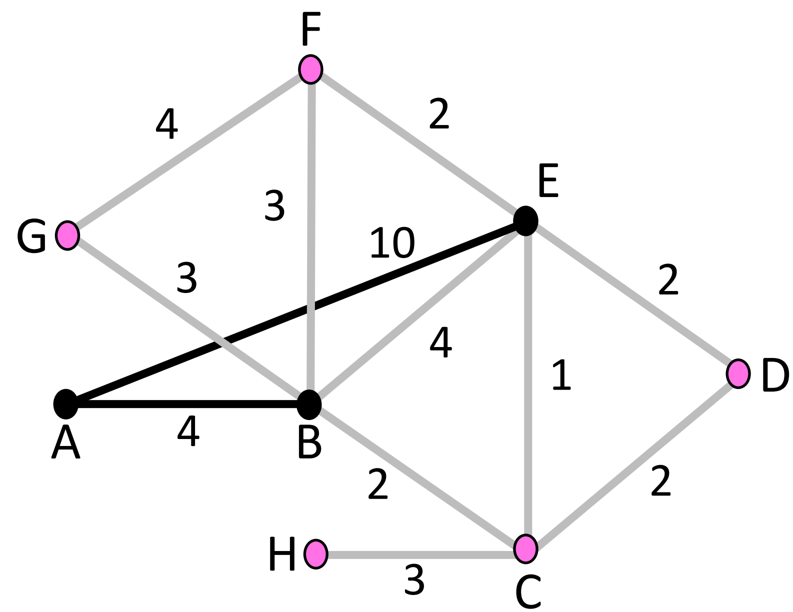
→

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

→

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

Learned better route



Distance Vector Example (3)

- Second exchange; learn best 2-hop routes

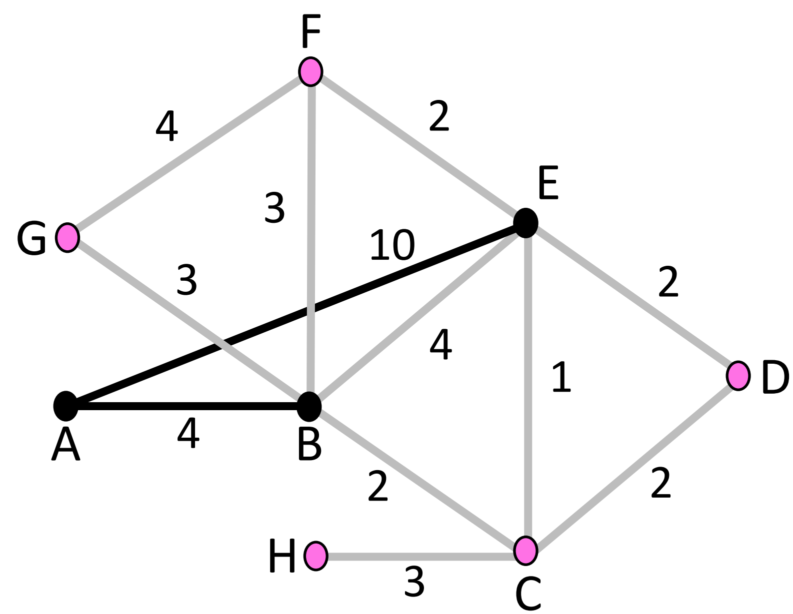
To	B says	E says
A	4	10
B	0	4
C	2	1
D	∞	2
E	4	0
F	3	2
G	3	∞
H	∞	∞

→

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 Example (4)

- Third exchange; learn best 3-hop routes

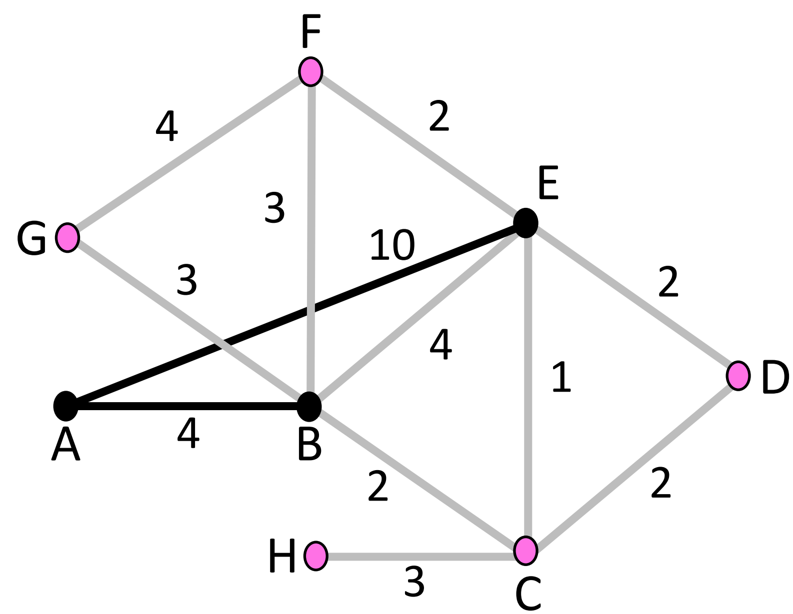
To	B says	E says
A	4	8
B	0	3
C	2	1
D	4	2
E	3	0
F	3	2
G	3	6
H	5	4

➔

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 Example (5)

- Subsequent exchanges; converged

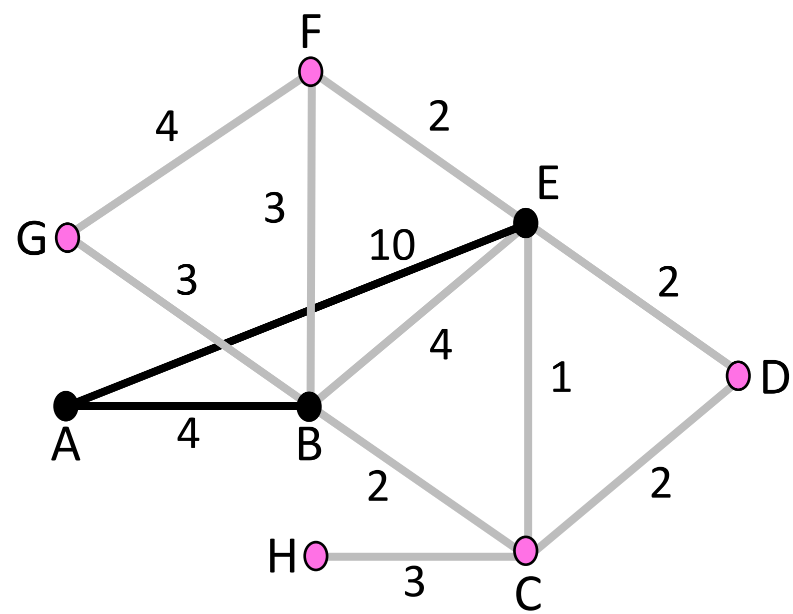
To	B says	E says
A	4	7
B	0	3
C	2	1
D	4	2
E	3	0
F	3	2
G	3	6
H	5	4

→

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

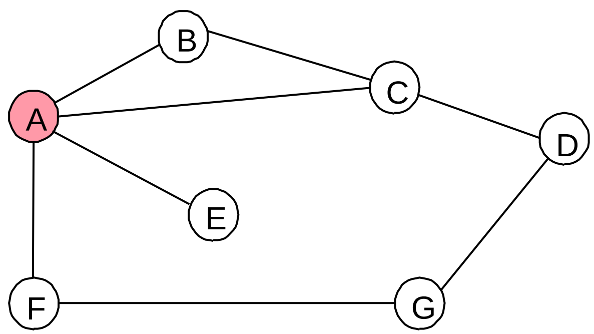


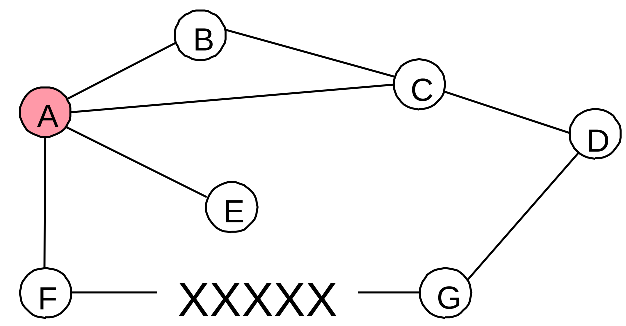
Distance Vector Dynamics

- Adding routes:
  - News travel one hop per exchange
- Removing routes
  - When a node fails, no more exchanges, other nodes forget
- But partitions (unreachable nodes in divided network) are a problem
  - "Count to infinity" scenario (later)

DV Dynamics Example

- How long does it take to converge?





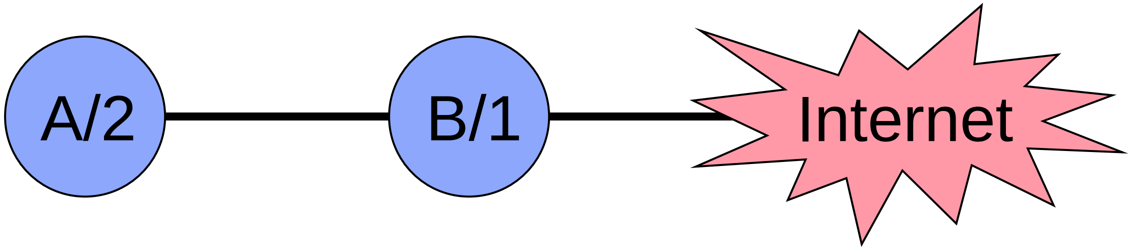
Dest	Cost	Next
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Dest	Cost	Next
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	3	C

  – A directly connected neighbor has an alternative route (i.e. one cycle)!

Count To Infinity Problem

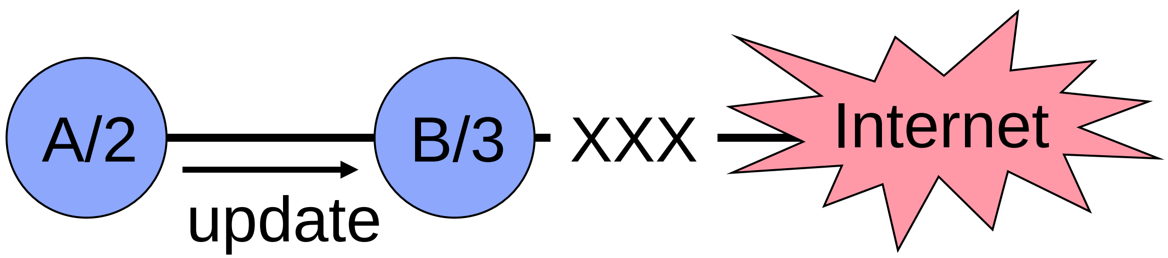
- Simple example
  - Costs in nodes are to reach Internet



- Now link between B and Internet fails …

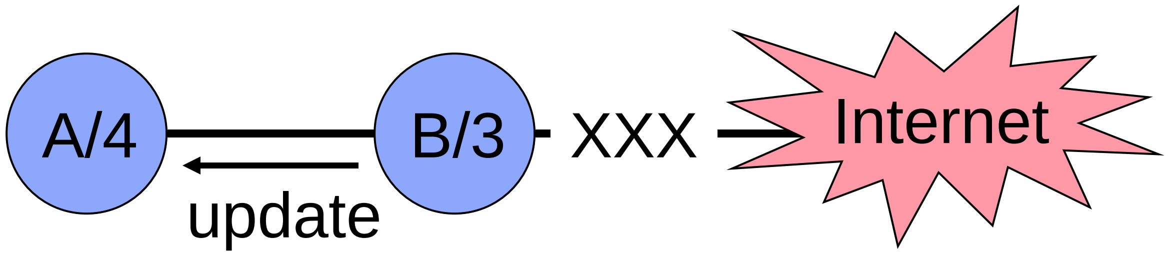
Count To Infinity Problem

- B hears of a route to the Internet via A with cost 2
- So B switches to the "better" (but wrong!) route



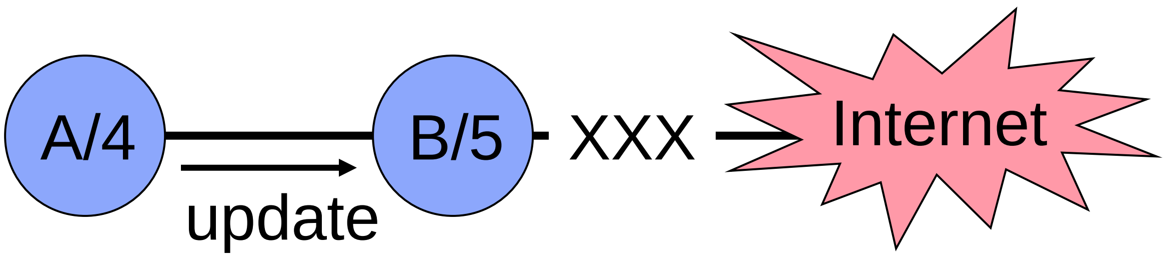
Count To Infinity Problem

- A hears from B and increases its cost



Count To Infinity Problem

- B hears from A and (surprise) increases its cost
- Cycle continues and we “count to infinity”



- Packets caught in the crossfire loop between A and B
- How do we fix this?

Split Horizon & Poison Reverse

- Solves trivial count-to-infinity problem
- Split Horizon (SH): router never advertises the cost of a destination back to its next hop – that's where it learned it from!
- SH w/poison reverse: goes even further – advertise back infinity
- However, DV protocols still subject to the same problem with more complicated topologies
  – Many enhancements suggested

Routing Information Protocol (RIP)

- DV protocol with hop count as metric
  - Infinity value is 16 hops; limits network size
  - Includes split horizon with poison reverse
- Routers send vectors every 30 seconds
  - Runs on top of UDP
  - With triggered updates for link failures
  - Time-out in 180 seconds to detect failures
- RIPv1 specified in RFC1058
  - www.ietf.org/rfc/rfc1058.txt
- RIPv2 (adds authentication etc.) in RFC1388
  - www.ietf.org/rfc/rfc1388.txt

RIP is an "Interior Gateway Protocol"

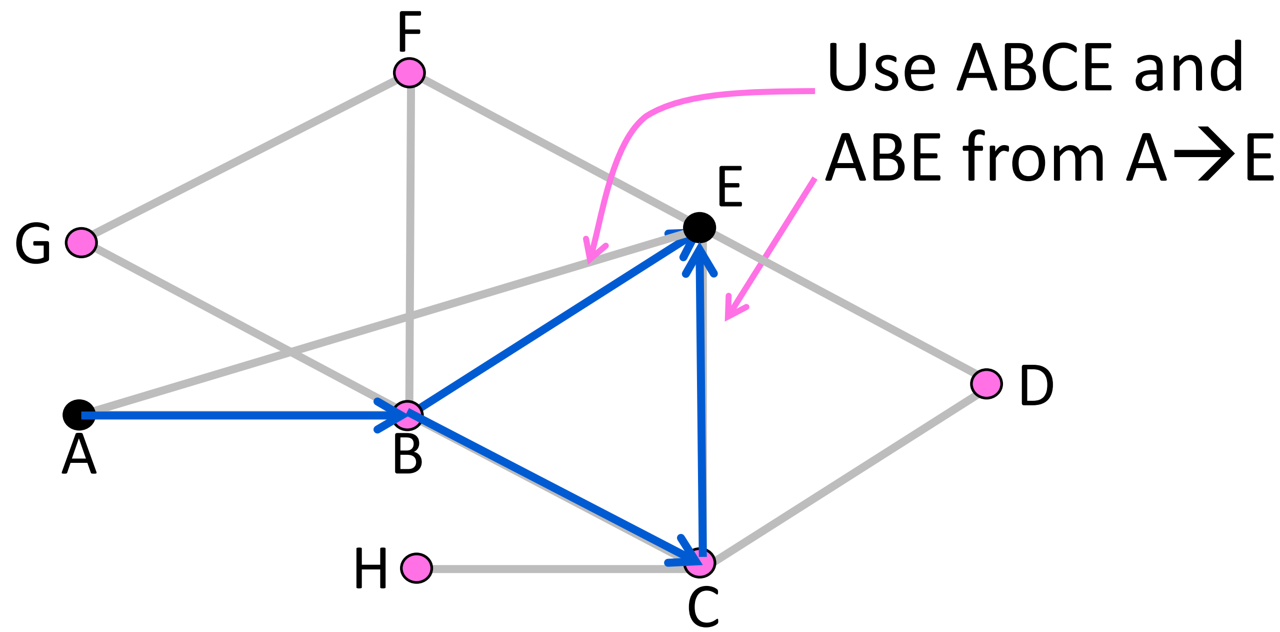
- Suitable for small- to medium-sized networks
  - such as within a campus, business, or ISP
- Unsuitable for Internet-scale routing
  - hop count metric poor for heterogeneous links
  - 16-hop limit places max diameter on network
- Later, we'll talk about "Exterior Gateway Protocols"
  - used between organizations to route across Internet

DV/LS Comparison

Property	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

Equal-Cost Multi-Path Routing

- More on shortest path routes
  - Allow multiple shortest paths



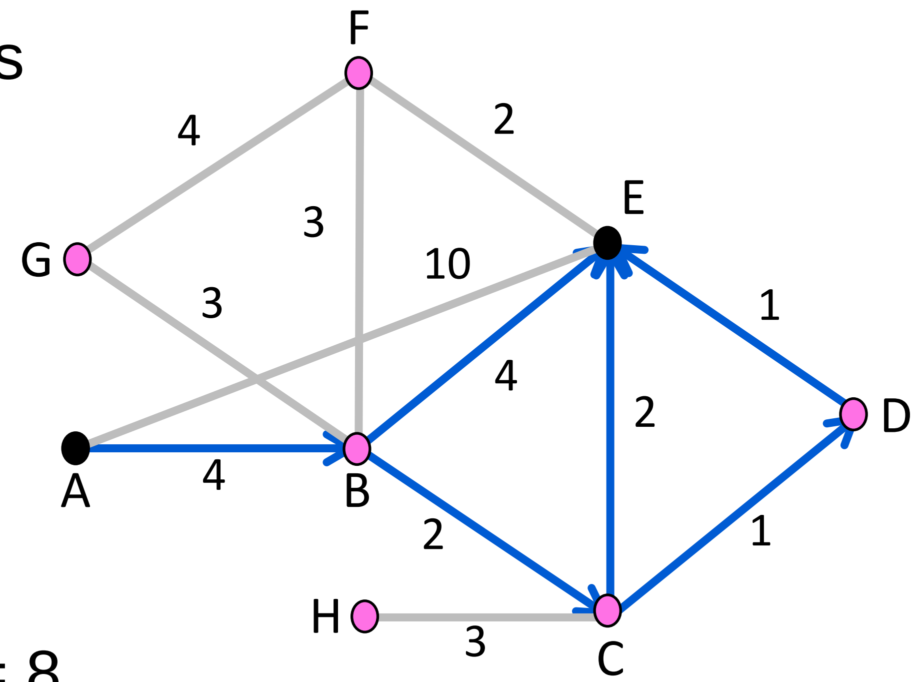
Multi-Path Routing

- 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 Multi-Path Routes

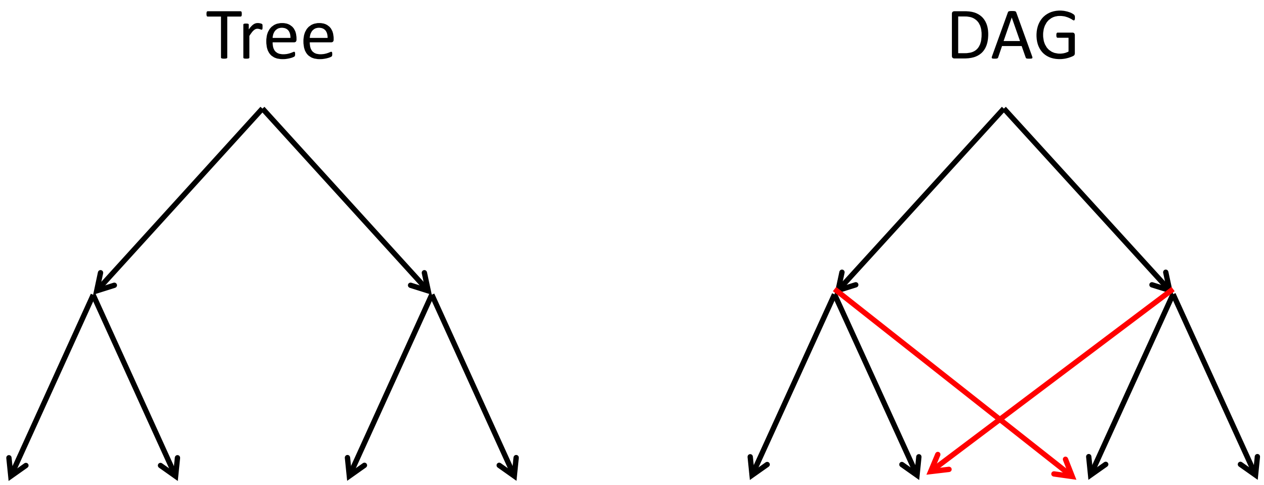
- One form of multipath routing
- Extends shortest path model
  - Keeps set if there are ties



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

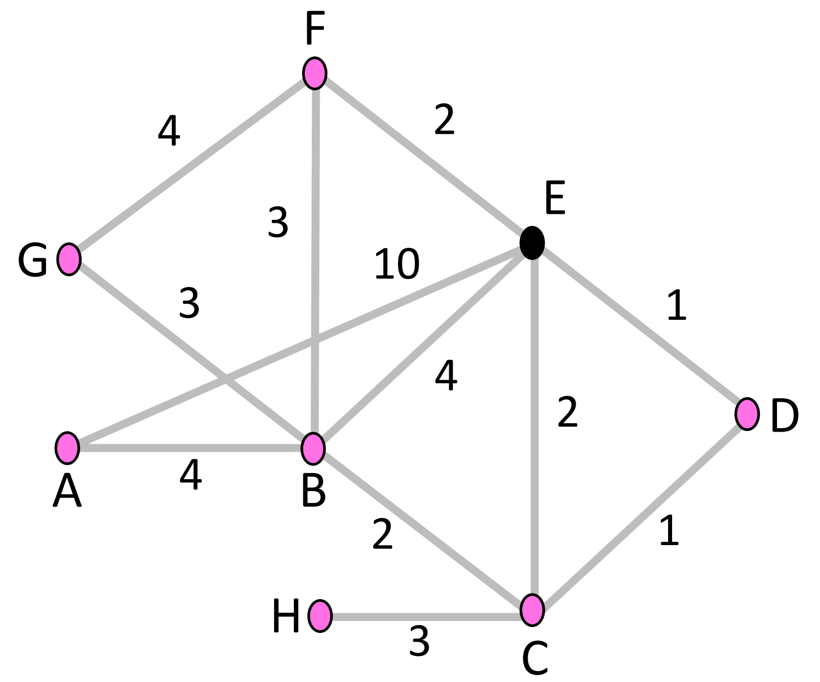
Source "Trees"

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

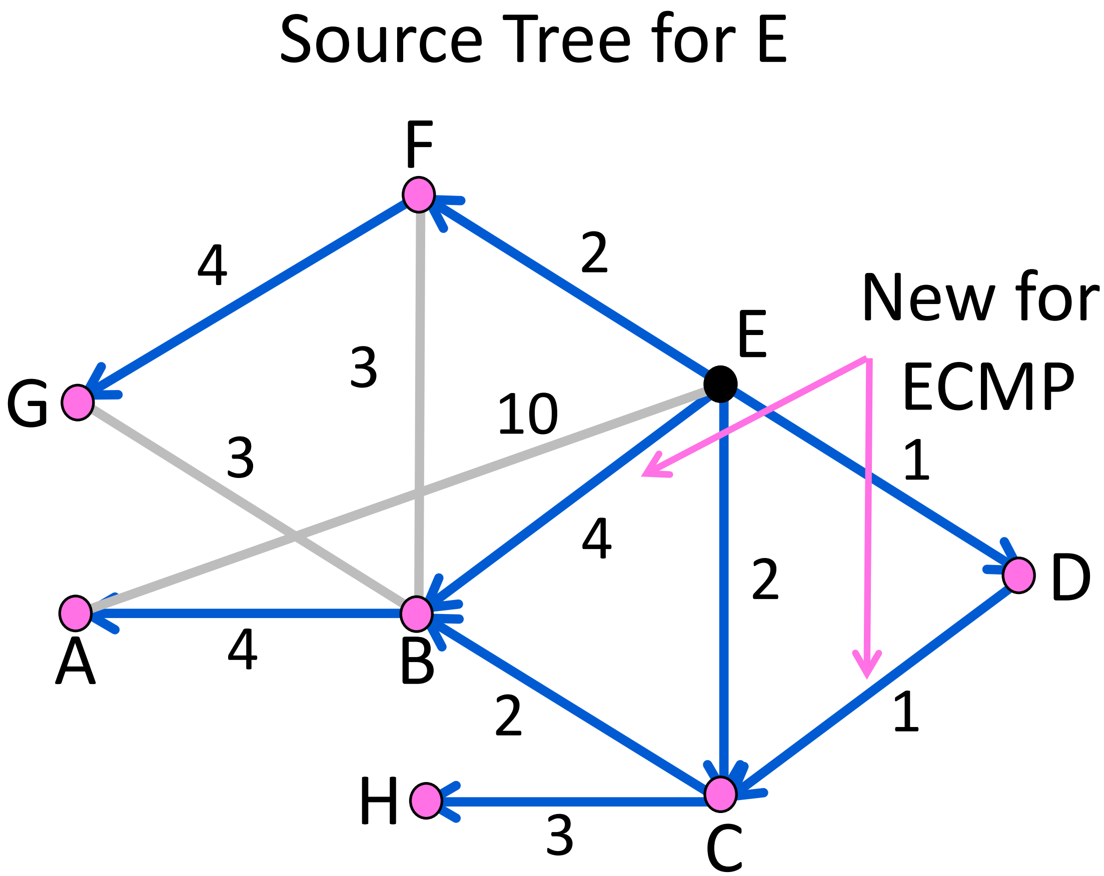


Source "Trees" (2)

- 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)



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

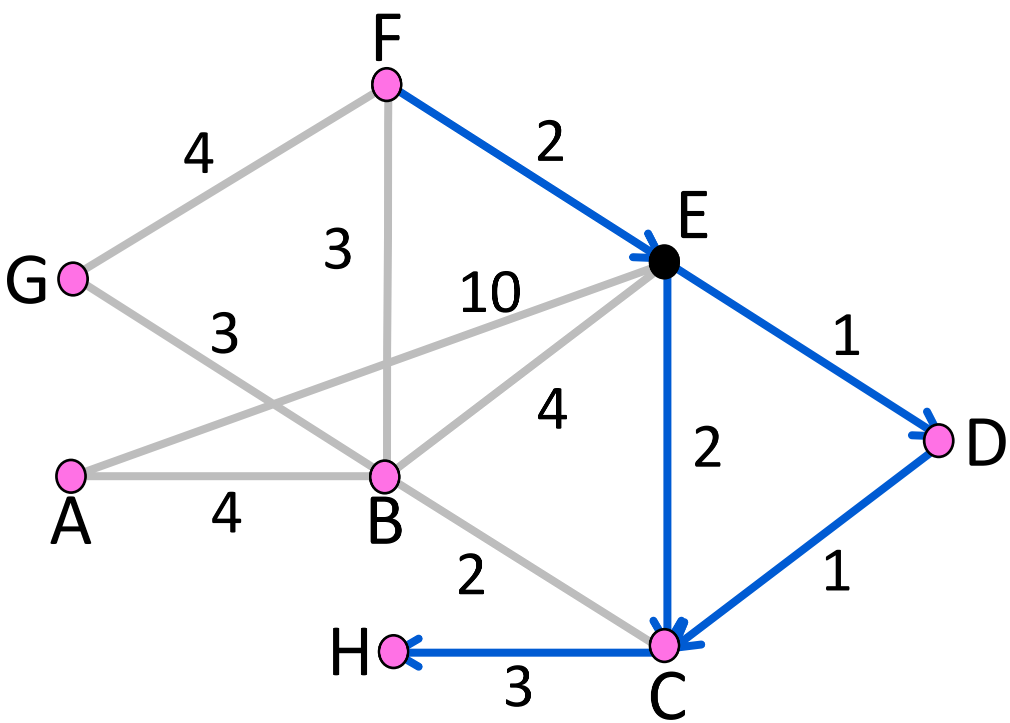
Forwarding with ECMP

- 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 called a <u>flow</u>
  - Map flow identifier to single next hop
  - No jitter within flow, but less balanced

Forwarding with ECMP (2)

Multipath routes from F to H



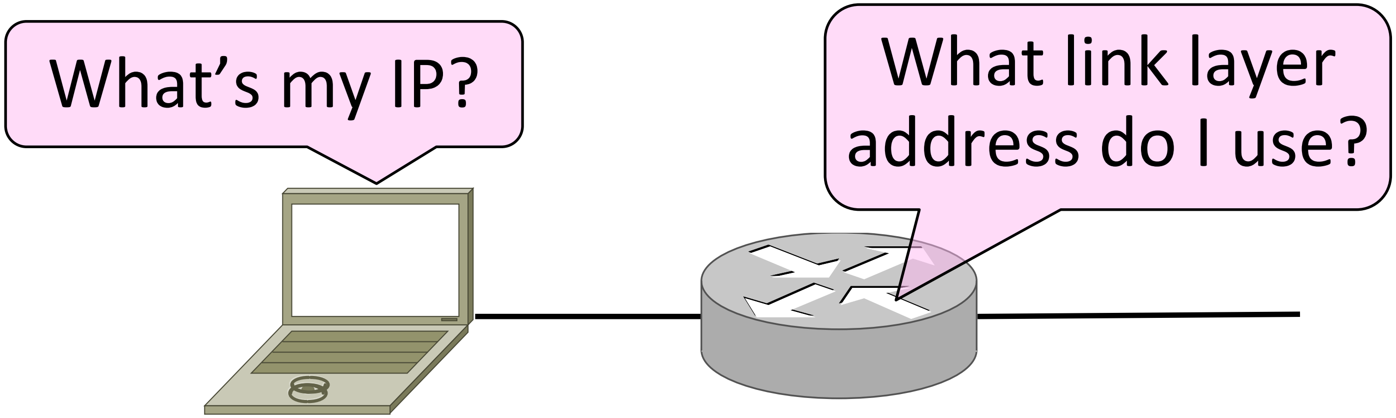
E's Forwarding Choices

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

IP Helpers – DHCP and ARP

- Filling in the gaps we need to make for IP forwarding work in practice
  - Getting IP addresses (DHCP)
  - Mapping IP to link addresses (ARP)

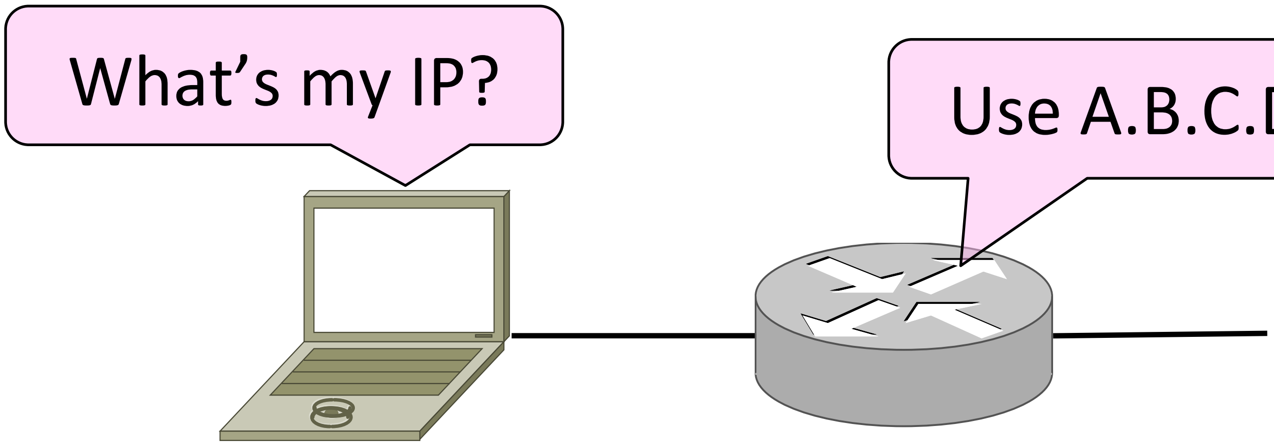


Getting IP Addresses

- Problem:
  - A node wakes up for the first time …
  - What is its IP address? What's the IP address of its router? Etc.
  - At least Ethernet address is on NIC

Getting IP Addresses (2)

1. Manual configuration (old days)
   – Can't be factory set, depends on use
2. A protocol for automatically configuring addresses (DHCP)
   – Shifts burden from users to IT folk

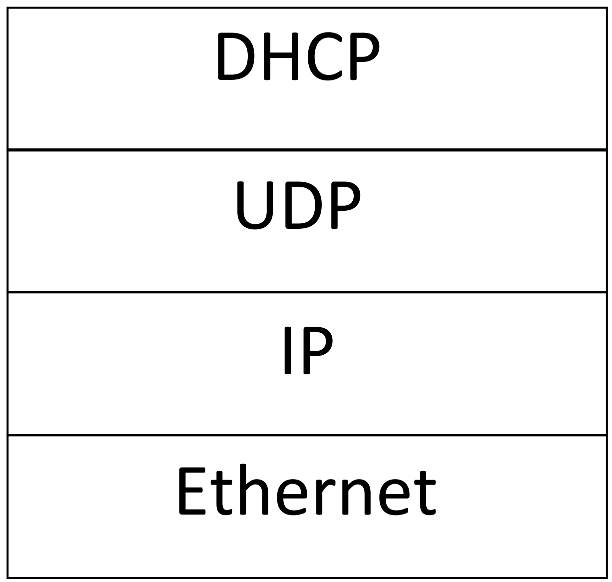


Dynamic Host Configuration Protocol

- Dynamic Host Configuration Protocol (DHCP), from 1993, widely used
- It leases IP address to nodes
- Provides other parameters too:
  - Network mask (more on this later)
  - Address of local router
  - DNS server
  - Time server
  - Etc.

DHCP Protocol Stack

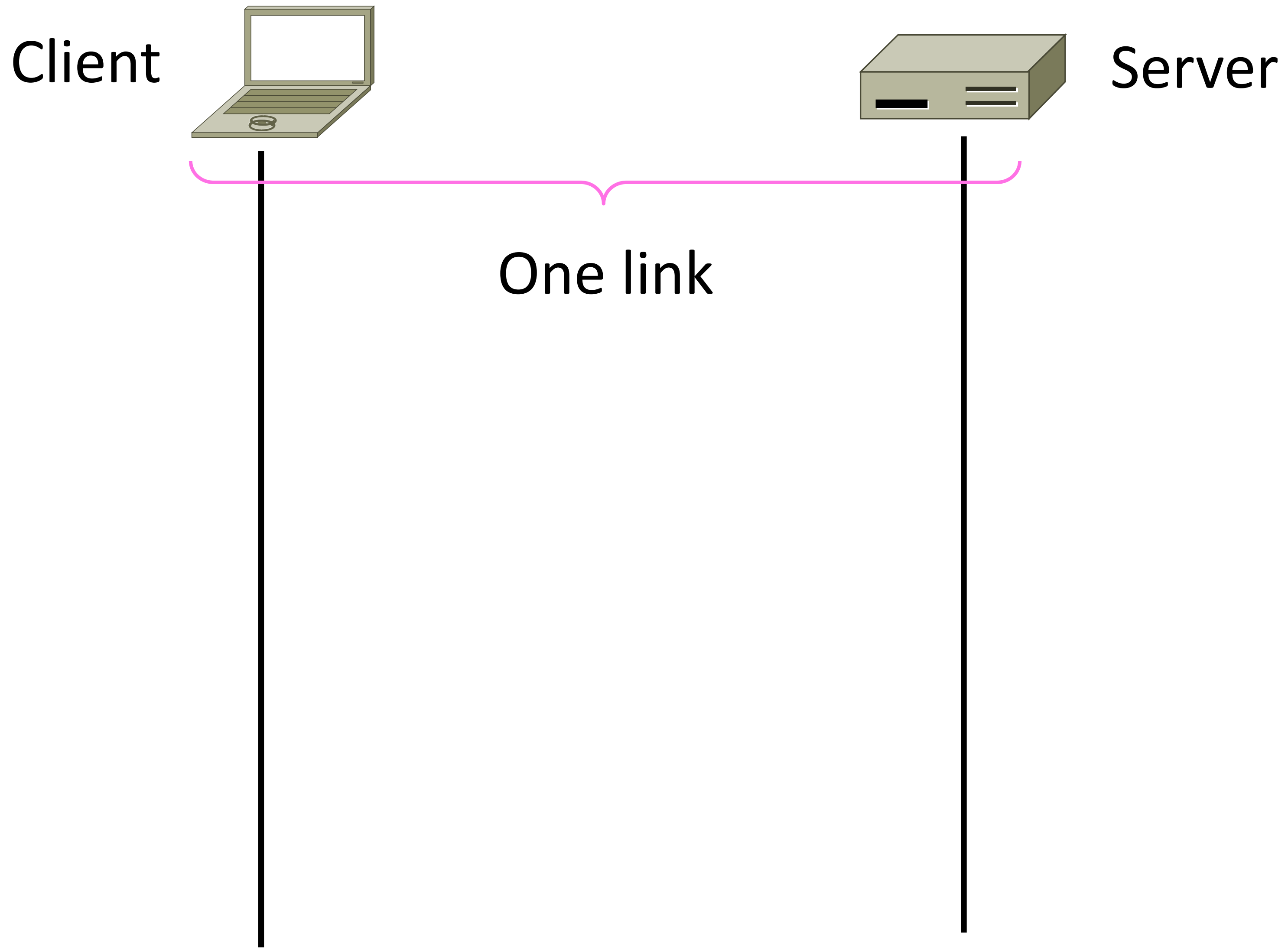
- DHCP is a client-server application
  - Uses UDP ports 67, 68



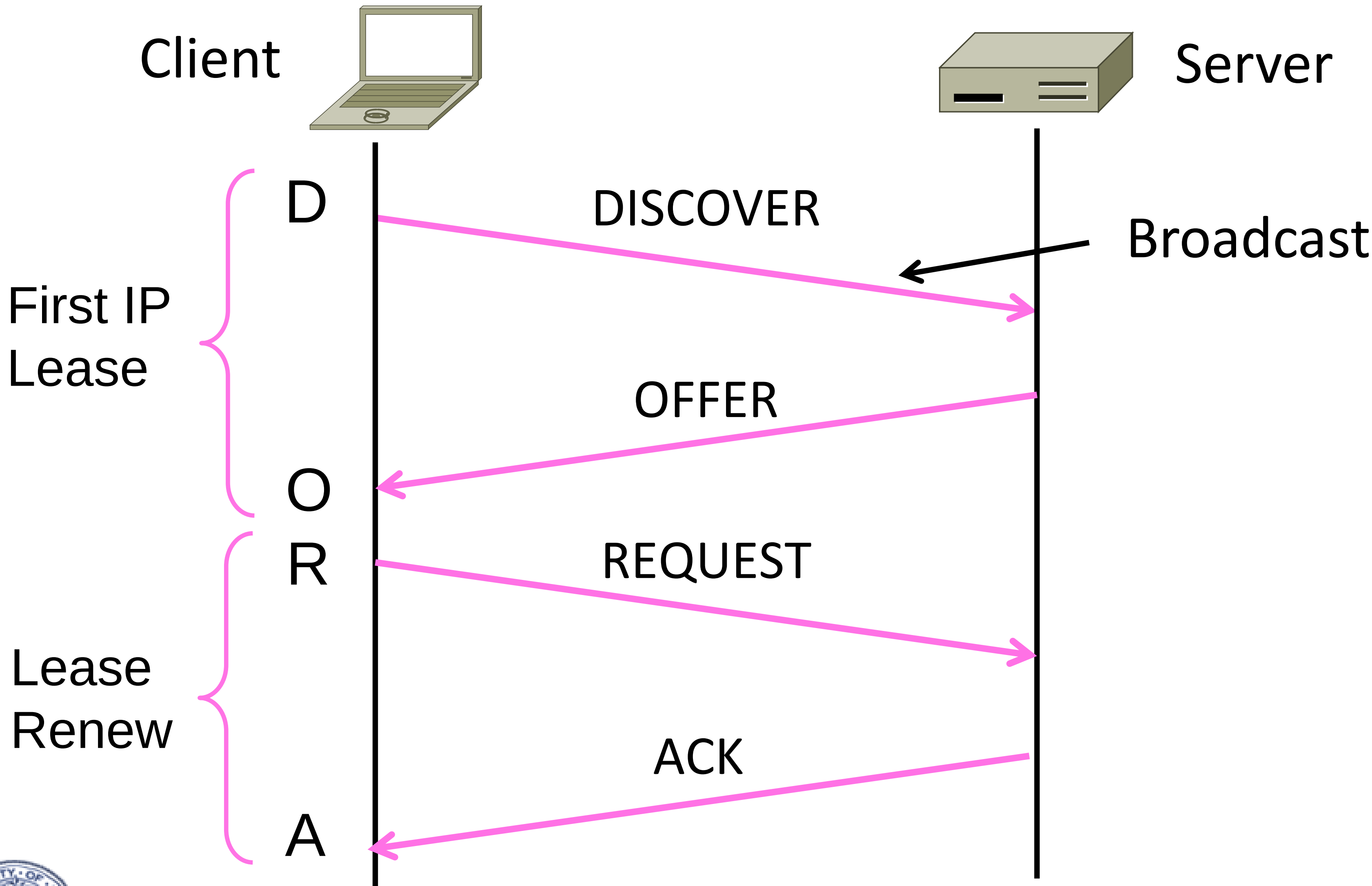
DHCP Addressing

- Bootstrap issue:
  - How does a node send a message to the DHCP server before it is configured?
- Answer:
  - Node sends broadcast messages that are delivered to all nodes in the network
  - Broadcast address is all 1s
  - IP (32 bit): 255.255.255.255
  - Ethernet (48 bit): ff:ff:ff:ff:ff:ff

DHCP Messages



DHCP Messages (2)

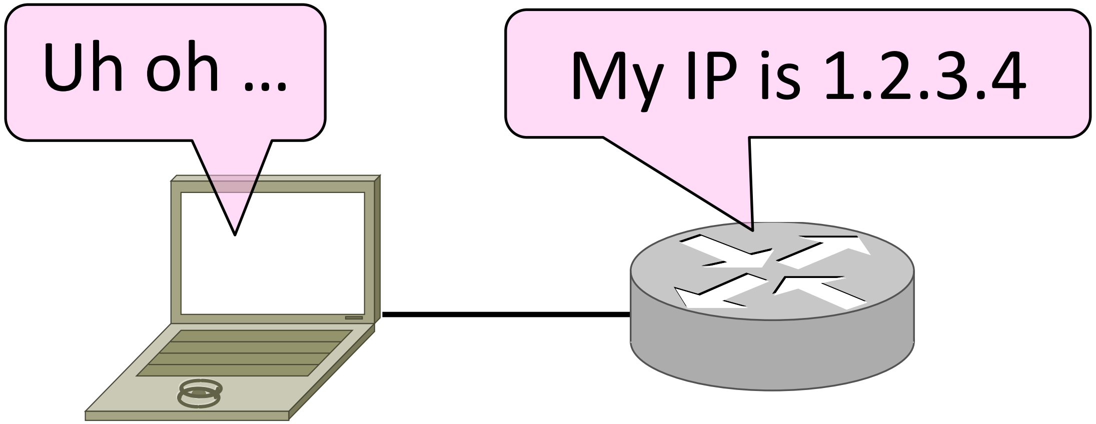


DHCP Messages (3)

- To renew an existing lease, an abbreviated sequence is used:
  - REQUEST, followed by ACK
- Protocol also supports replicated servers for reliability
  - By using broadcast, it allows all of DHCP servers to know all the client requests so they can coordinate among themselves

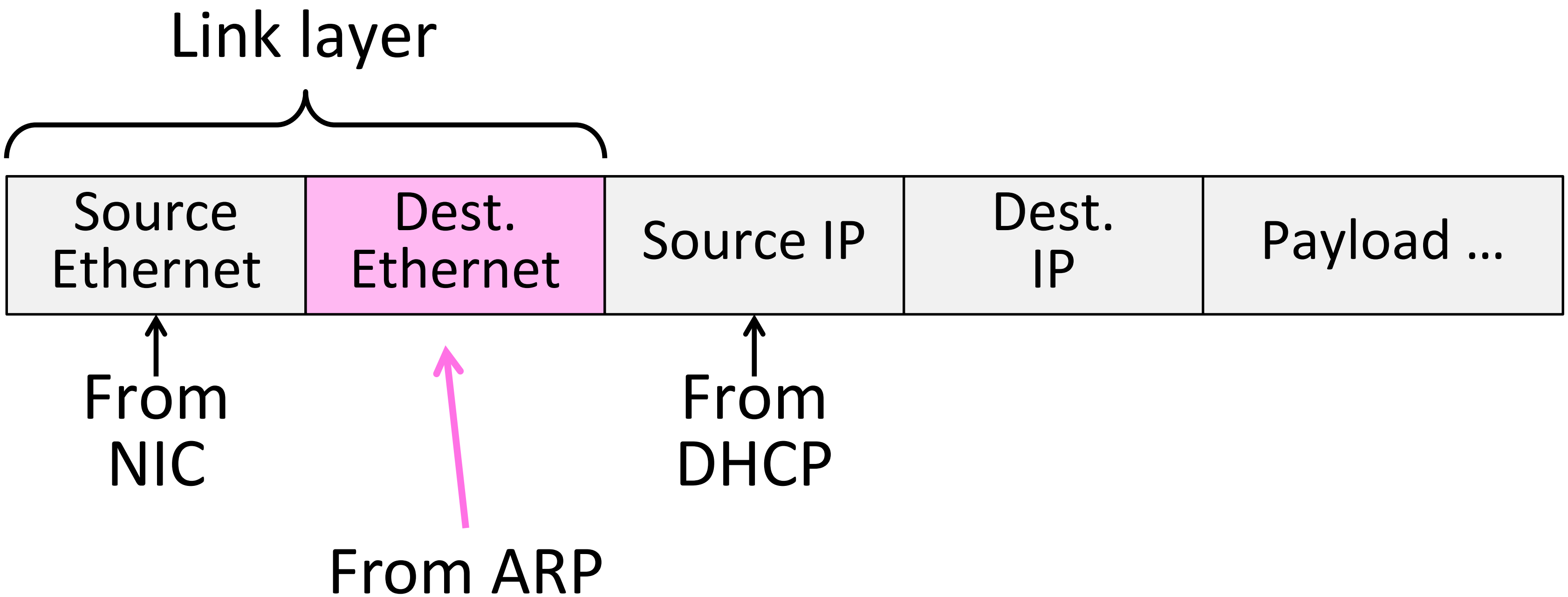
Sending an IP Packet

- Problem:
  - A node needs Link Layer addresses to send a frame over the local link
  - How does it get the destination link address from a destination IP address?



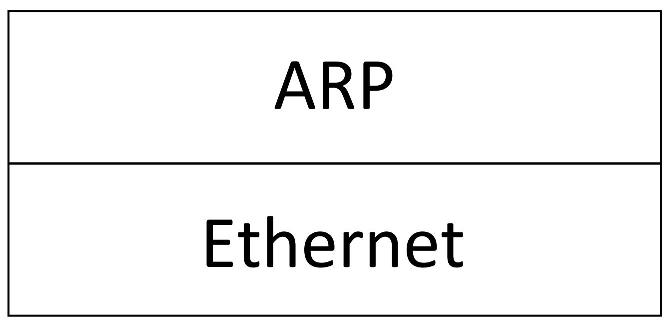
Address Resolution Protocol (ARP)

- Node uses to map a local IP address to its Link Layer addresses

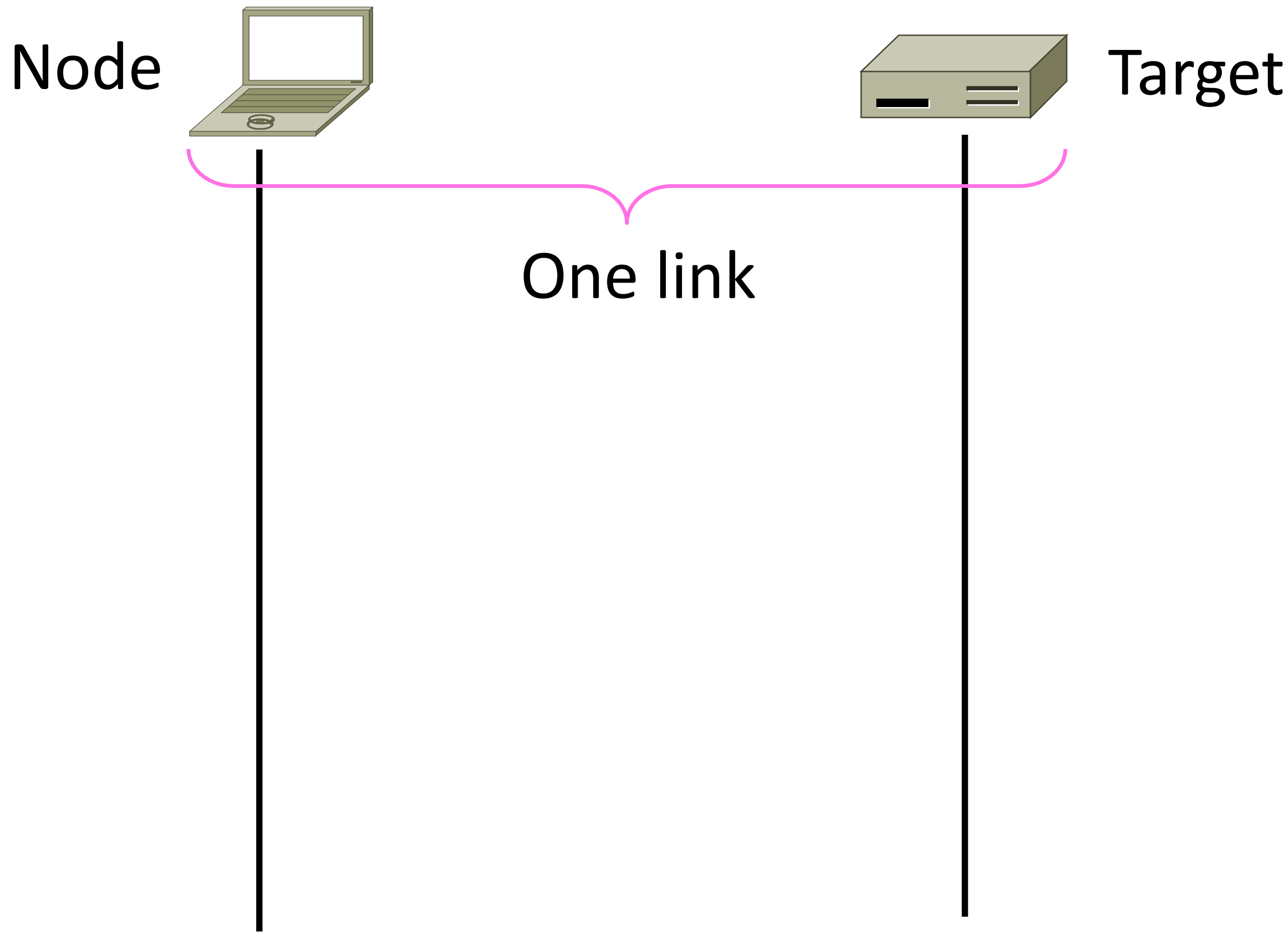


ARP Protocol Stack

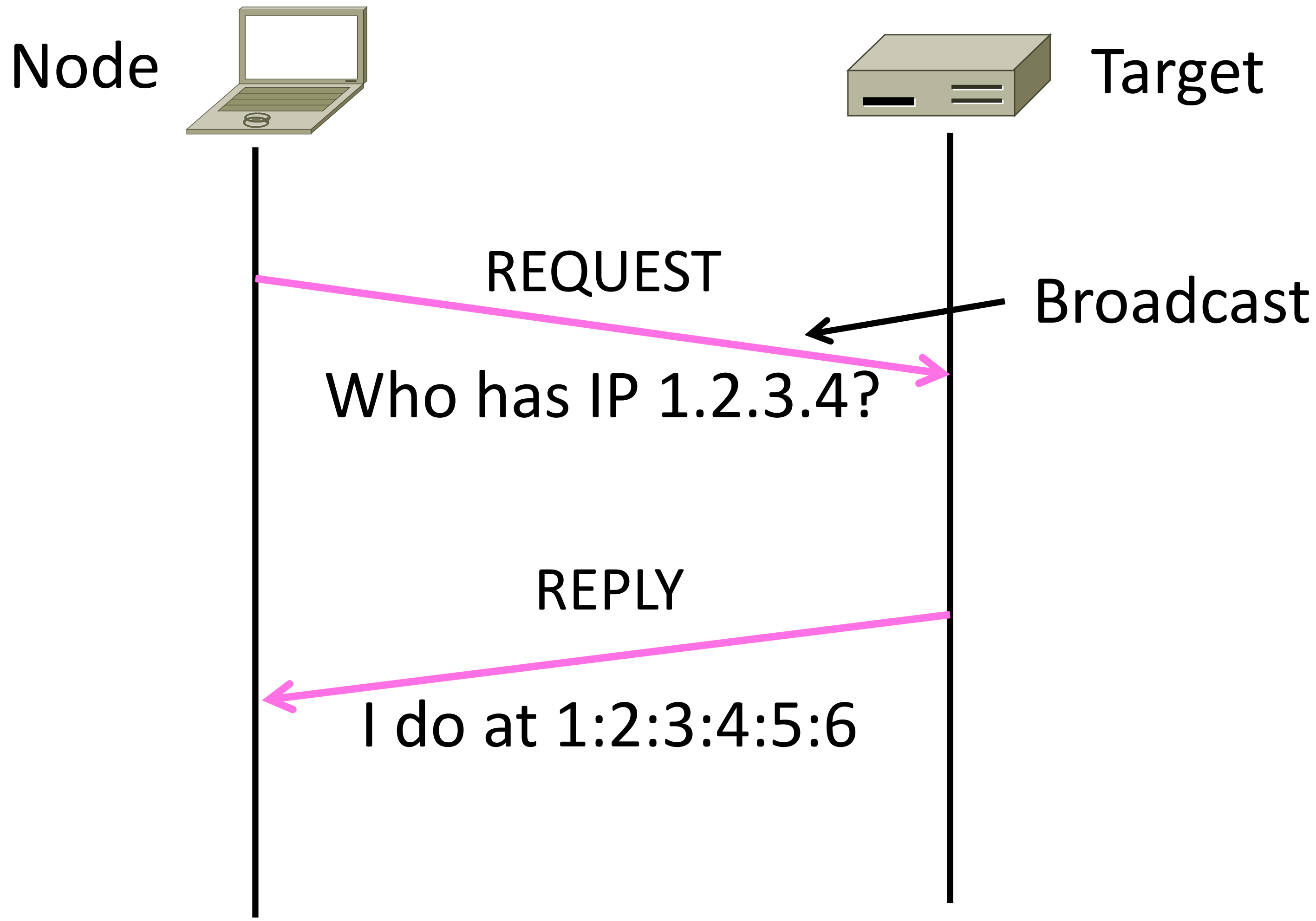
- ARP sits right on top of the link layer
  - No servers, just asks node with target IP to identify itself
  - Uses broadcast to reach all nodes



ARP Messages



ARP Messages (2)

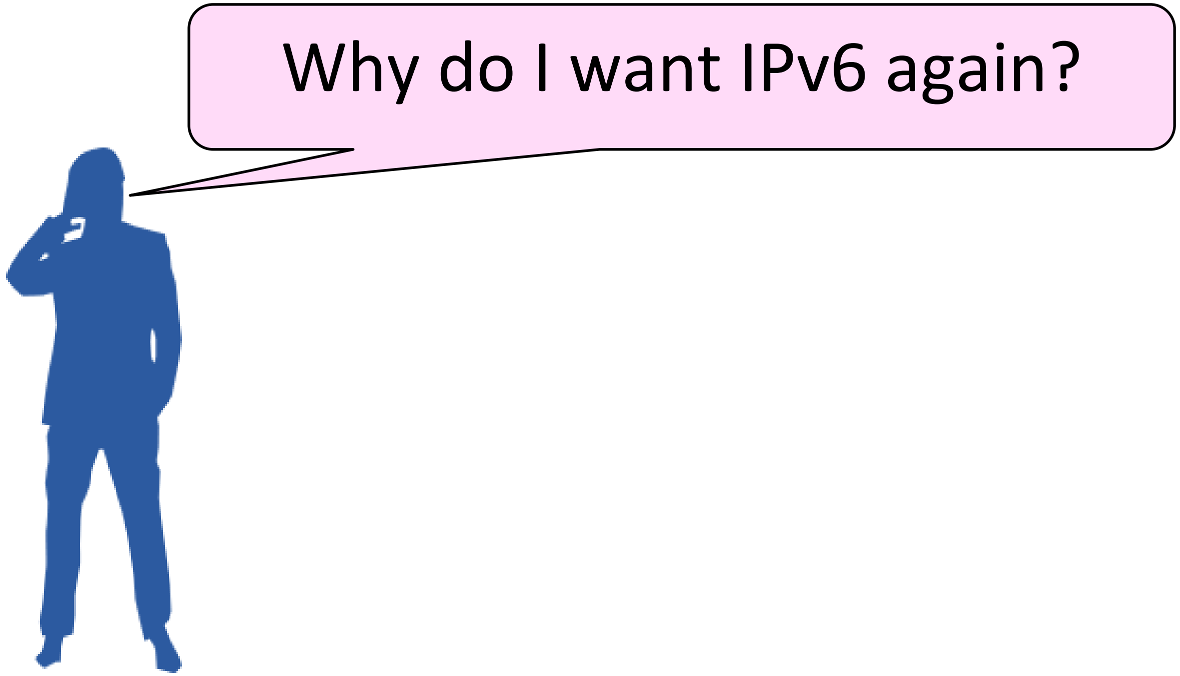


Discovery Protocols

- Help nodes find each other
  - There are more of them
    - E.g., zeroconf, Bonjour
- Often involve broadcast
  - Since nodes aren't introduced
  - Very handy glue

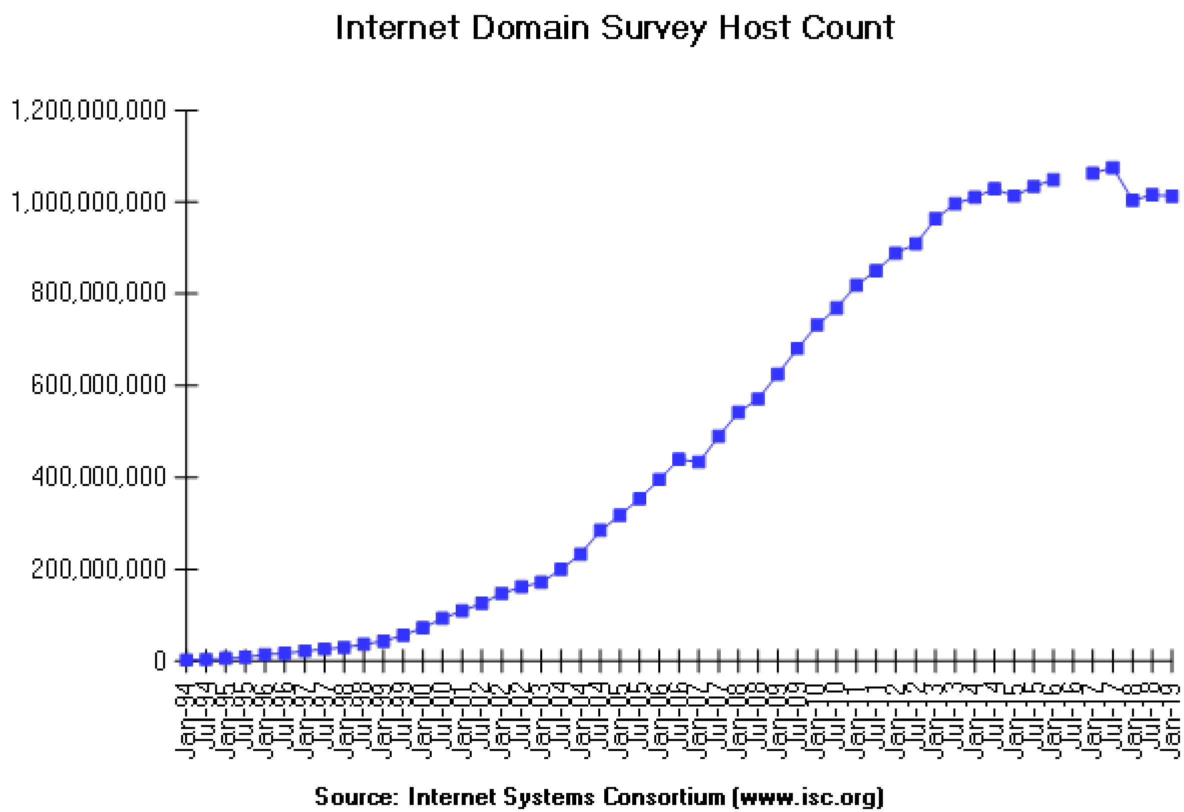
IPv6

- IP version 6, the future of IPv4 that is now (still) being deployed



Internet Growth

- At least a billion+ Internet hosts and growing…
- And we're using 32-bit addresses!



The End of New IPv4 Addresses

- Now running on leftover blocks held by the regional registries; much tighter allocation policies

IANA
(All IPs)
01/31/2011

APNIC
(Asia Pacific, Australia)
04/15/2011

RIPE NCC
(Europe, Middle East and Central Asia)
09/14/2012

LACNIC
(Latin America and the Caribbean)
06/10/2014

ARIN
(US, Canada)
09/21/2015

AfriNIC
(Africa)
Still new IPv4 available

ISPs

Companies

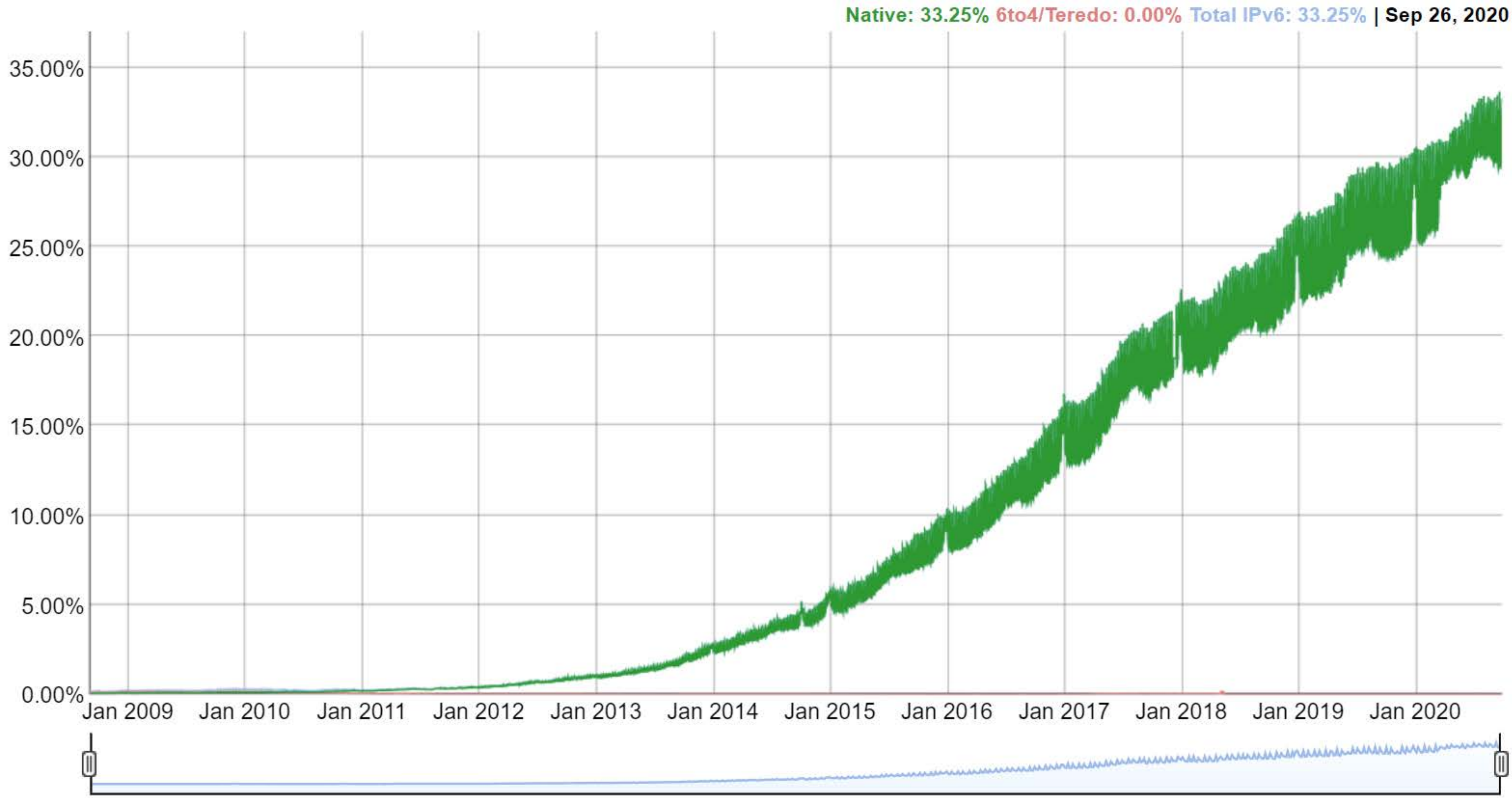
IP Version 6 to the Rescue

- Effort started by the IETF in 1994
  - Much larger addresses (128 bits)
  - Many sundry improvements

- Become an IETF standard in 1998
  - Nothing much happened for a decade
  - Hampered by deployment issues, and a lack of adoption incentives
  - Big push ~2011 as exhaustion looms

IPv6 Deployment

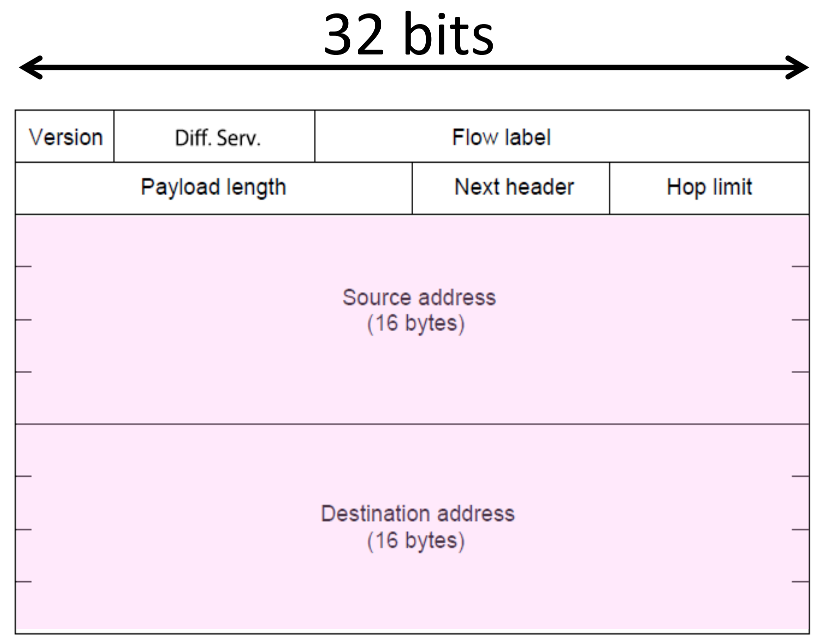
- Percentage of users accessing Google via IPv6



Source: Google IPv6 Statistics, 09/28/20

IPv6 Address

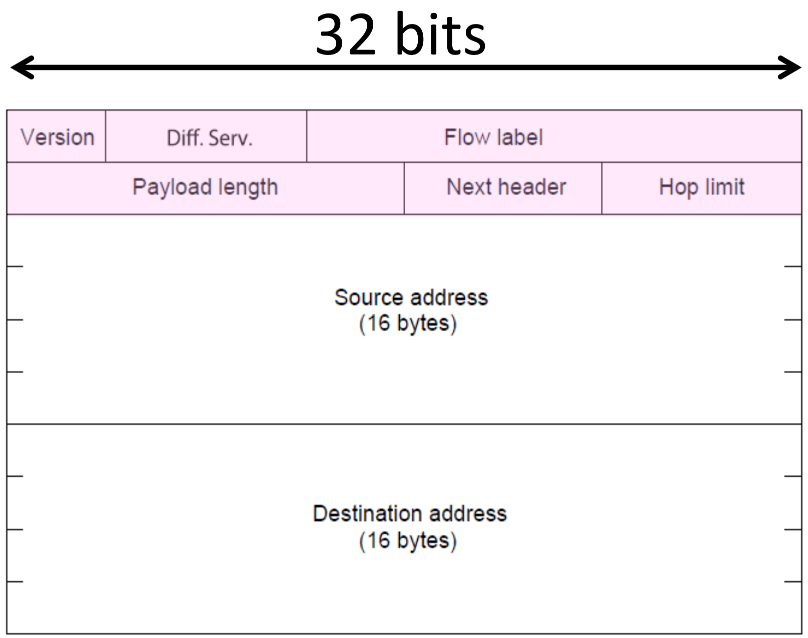
- Features large addresses
  - 128 bits, most of header
- New notation
  - 8 groups of 4 hex digits (16 bits)
  - Omit leading zeros, groups of zeros



Example:
2001:0db8:85a3:0000:0000:8a2e:0370:7334
→ *2001:db8:85a3::8a2e:370:7334*

IPv6 Address (2)

- Lots of other small changes
  - Streamlined header processing
  - Flow label to group of packets
  - Better fit with "advanced" features (mobility, multicasting, security)



IPv6 Transition

- The Big Problem:
  - How to deploy IPv6?
  - Fundamentally incompatible with IPv4
- Dozens of approaches proposed in the past
  - Dual stack (speak IPv4 and IPv6)
  - Translators (convert packets)
  - Tunnels (carry IPv6 over IPv4)
- Majority of Internet supports dual stack nowadays

Key Concepts

- Routing is a global process; forwarding is local one
- The DV algorithm and RIP
  - Distributed data dissemination and computation
  - Good scalability, but slow convergence
- The ECMP protocol
  - Allows using multiple equal cost paths
  - Permits different forms of load balancing
- We need IP helpers for everything to work
  - DHCP: helps you get an IP address
  - ARP: helps you find a Link-Layer address for a destination IP
- IPv6 helps us deal with IPv4 address exhaustion