Internet Protocol

Layer reminder

- Bridges - emulate single link
  - Everything broadcast
  - Same collision domain
- Switches - emulate single network
  - Flat addressing
  - Broadcast supported
- Internet - connect multiple networks
  - Hierarchical addressing
  - No broadcast
  - Highly scalable

IP service model

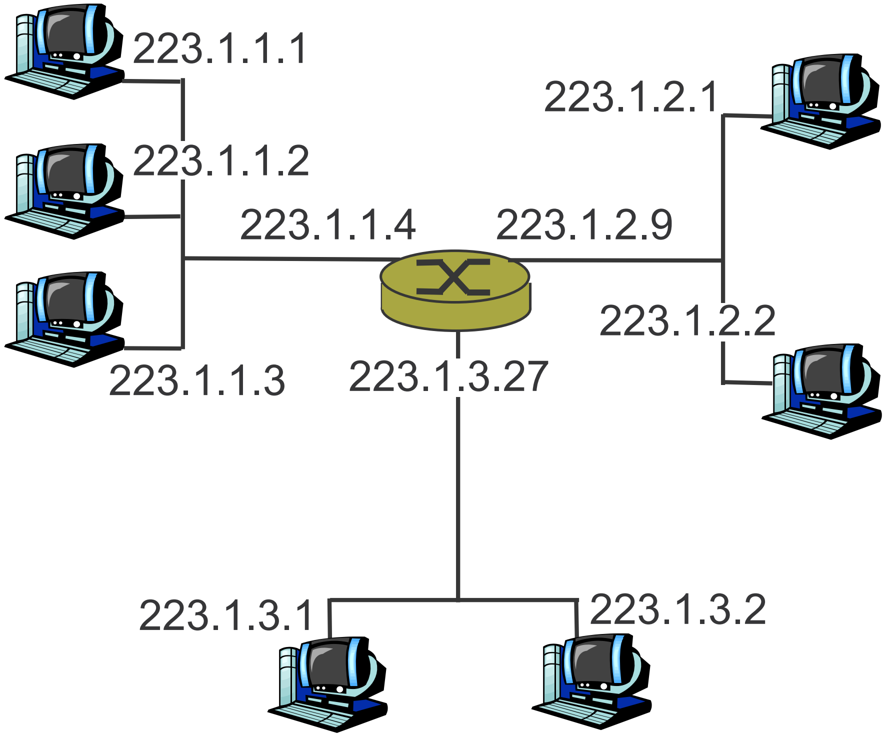
- Service provided to transport layer (TCP, UDP)
  - Global name space
  - Host-to-host connectivity (connectionless)
  - Best-effort packet delivery
- Not in IP service model
  - Delivery guarantees on bandwidth, delay or loss
- Delivery failure modes
  - Packet delayed for a very long time
  - Packet loss
  - Packet delivered more than once
  - Packets delivered out of order

IP addressing

- Ethernet address space
  - Flat
  - Assigned at manufacture time
- IP address space
  - Hierarchical
  - Assigned at configuration time

IP Addressing: introduction

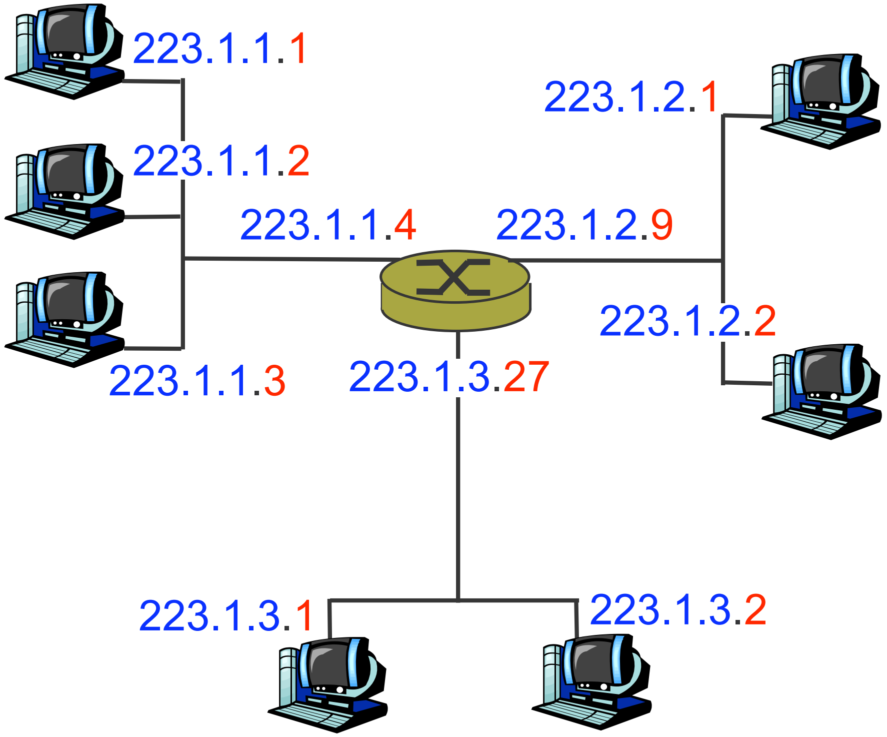
- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - routers typically have multiple interfaces
  - host typically has one interface
  - IP addresses associated with each interface



$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

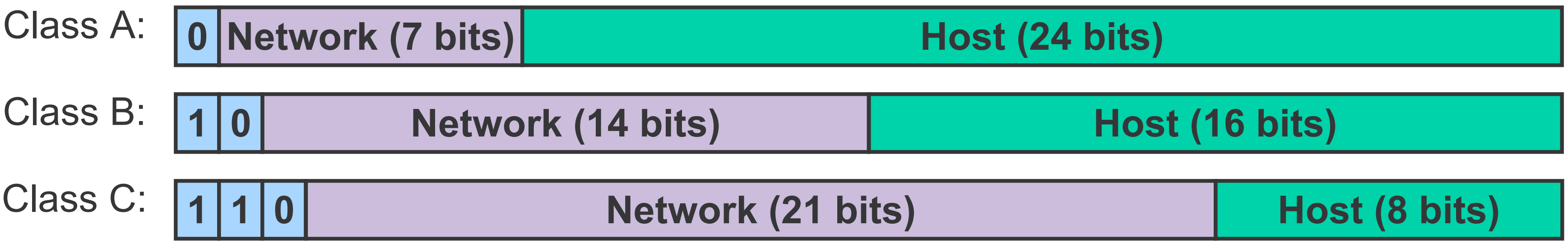
IP networks

- Address has 2 components
  - Network (high-order bits)
  - Host (low-order bits)



IPv4 Address Model

Class	Network ID	Host ID	# of Addresses	# of Networks
A	0 + 7 bit	24 bit	2^{24}-2	126
B	10 + 14 bit	16 bit	65,536 - 2	2^{14}
C	110 + 21 bit	8 bit	256 - 2	2^{21}
D	1110 + Multicast Address		IP Multicast	
E	Future Use			



Class A: 0 | Network (7 bits) | Host (24 bits)

Class B: 1 | 0 | Network (14 bits) | Host (16 bits)

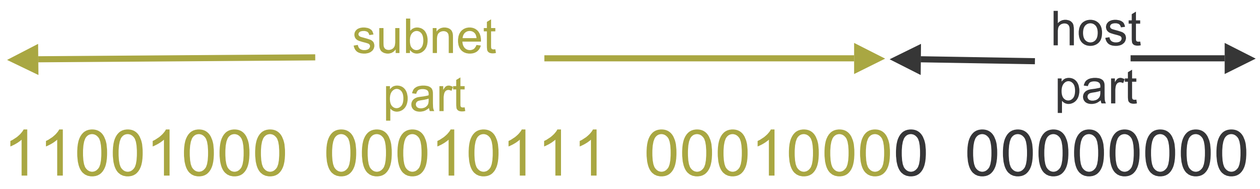
Class C: 1 | 1 | 0 | Network (21 bits) | Host (8 bits)

IP networks

- Class A network: 18.0.0.0 (MIT)
  - www.mit.edu has address 18.7.22.83
- Class B network: 128.174.0.0 (UIUC)
  - www.cs.uiuc.edu has address 128.174.252.84
- Class C network: 216.125.249.0 (Parkland)
  - www.parkland.edu has address 216.125.249.97

CIDR

- 3-class model too inflexible
- CIDR: Classless InterDomain Routing
  - Arbitrary number of bits to specify network
  - Address format: a.b.c.d/x, where x is # bits in network portion



11001000 00010111 00010000 00000000

200.23.16.0/23

Classless Domains

- Internet Archive - 207.241.224.0/20
  - 4K hosts
  - 207.241.224.0 - 207.241.239.255
- AT&T - 204.127.128.0/18
  - 16K hosts
  - 204.127.128.0 - 204.127.191.255
- UUNET - 63.64.0.0/10
  - 4M hosts
  - 63.64.0.0 - 63.127.255.255

IP forwarding

- Forwarding table has:
  - Network number
  - Interface
- Avoid having to store 4 billion entries
  - But there are still 2 million class C's
  - …and perhaps more CIDR networks

Hierarchical Routing

Our routing study thus far - idealization
all routers identical
network “flat”
… *not* true in practice

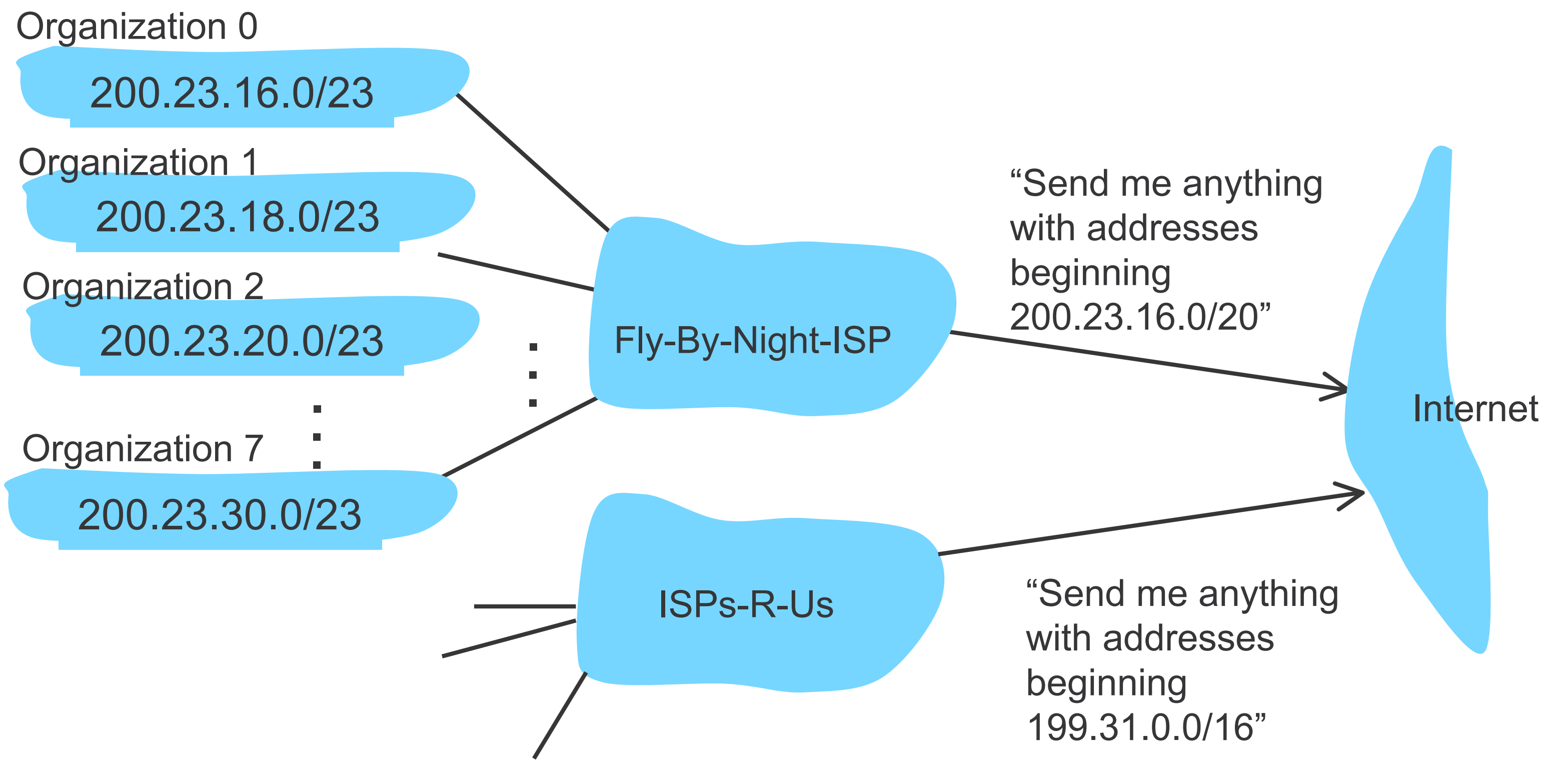
scale: with 200 million destinations:

- can’t store all dest’s in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

Hierarchical Networks

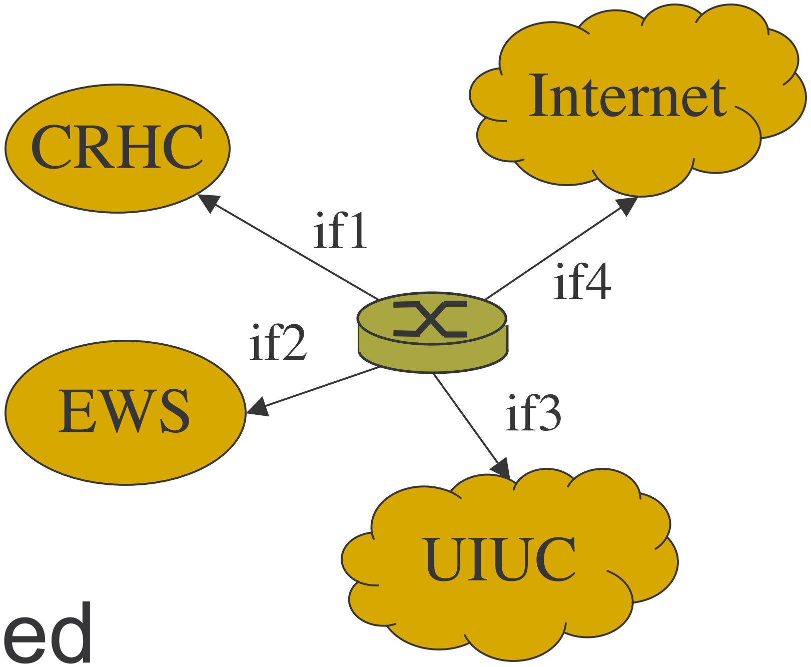


Subnetting

- UIUC - 130.126.0.0/16
  - 130.126.0.0 - 130.126.255.255
- CRHC - 130.126.136.0/21
  - 130.126.136.0 - 130.126.143.255
- EWS - 130.126.160.0/21
  - 130.126.160.0 - 130.126.167.255

Forwarding Tables

130.126.136.0/21	if1
130.126.160.0/21	if2
130.126.0.0/16	if3
0.0.0.0/0	if4



- Most specific rule is used
- Most hosts outside of the core have default rules

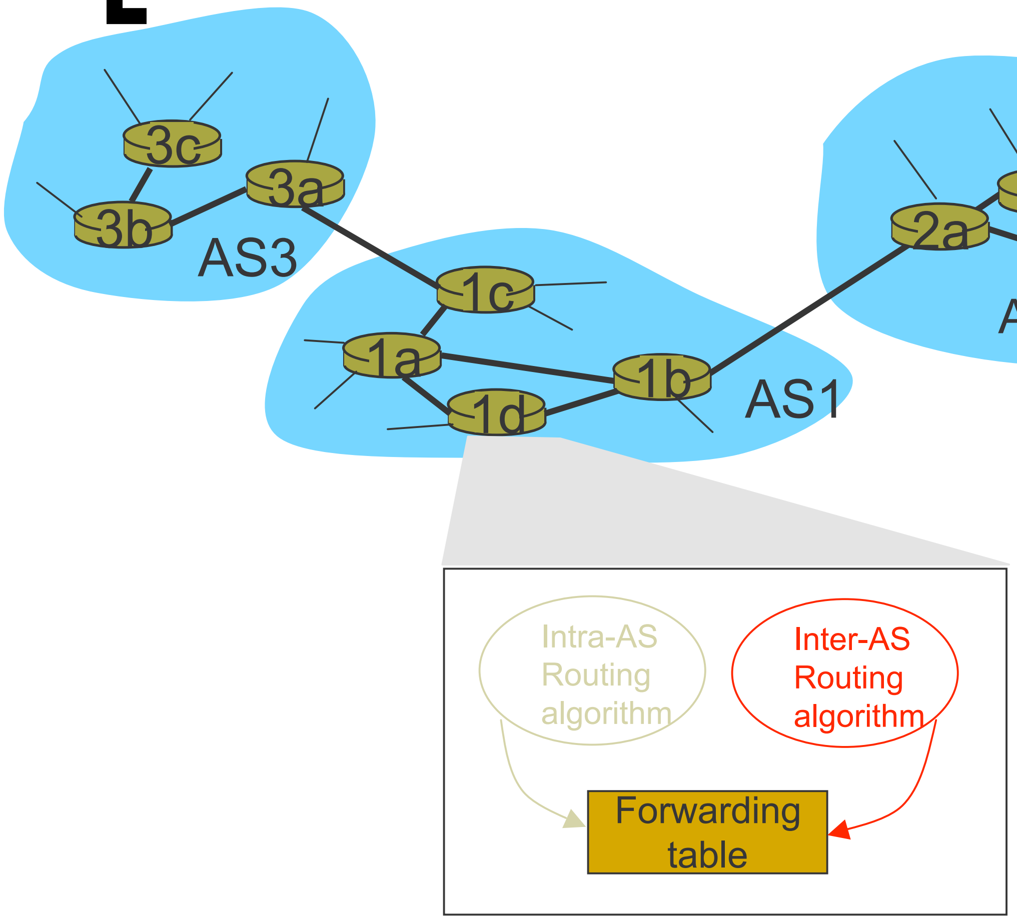
Hierarchical Routing

- aggregate routers into regions, “autonomous systems” (AS)
- routers in same AS run same routing protocol
  - “intra-AS” routing protocol
  - routers in different AS can run different intra-AS routing protocol

Gateway router

- Direct link to router in another AS

Interconnected ASes



- Forwarding table is configured by both intra- and inter-AS routing algorithm
  - Intra-AS sets entries for internal dests
  - Inter-AS & Intra-As sets entries for external dests

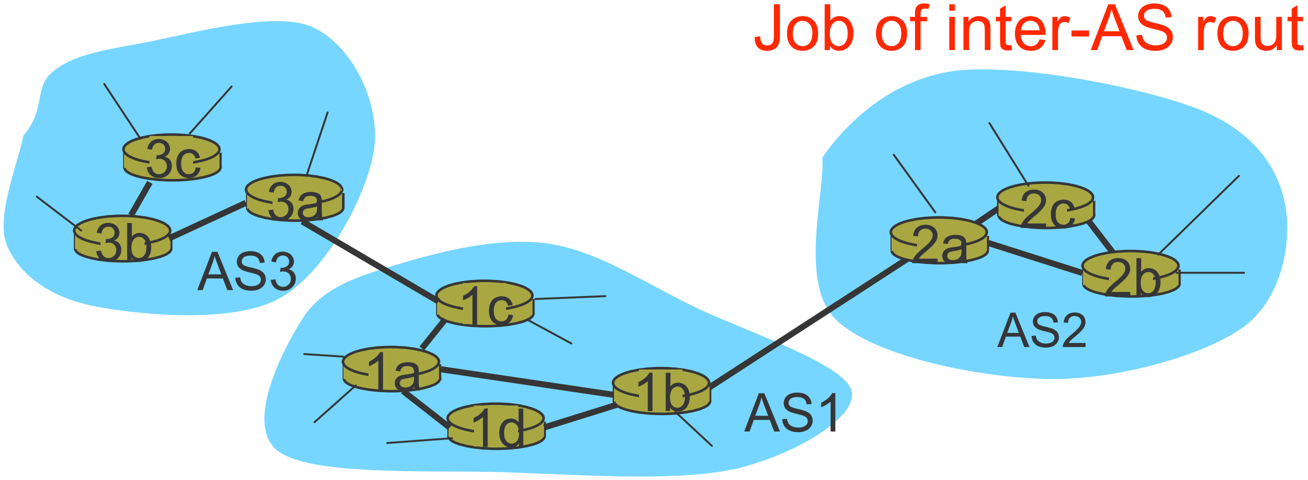
Inter-AS tasks

- Suppose router in AS1 receives datagram for which dest is outside of AS1
  - Router should forward packet towards on of the gateway routers, but which one?

AS1 needs:

1. to learn which dests are reachable through AS2 and which through AS3
2. to propagate this reachability info to all routers in AS1

Job of inter-AS routing!

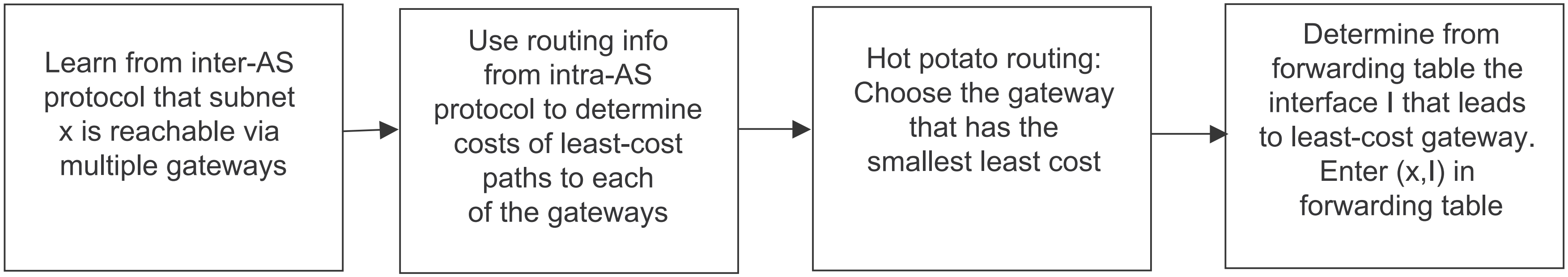


Example: Setting forwarding table in router 1d

- Suppose AS1 learns from the inter-AS protocol that subnet x is reachable from AS3 (gateway 1c) but not from AS2.
- Inter-AS protocol propagates reachability info to all internal routers.
- Router 1d determines from intra-AS routing info that its interface I is on the least cost path to 1c.
- Puts in forwarding table entry (x,I).

Example: Choosing among multiple ASes

- Now suppose AS1 learns from the inter-AS protocol that subnet *x* is reachable from AS3 *and* from AS2.
- To configure forwarding table, router 1d must determine towards which gateway it should forward packets for dest x.
- This is also the job on inter-AS routing protocol!
- Hot potato routing: send packet towards closest of two routers.



Learn from inter-AS protocol that subnet x is reachable via multiple gateways

Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways

Hot potato routing: Choose the gateway that has the smallest least cost

Determine from forwarding table the interface I that leads to least-cost gateway. Enter (x,I) in forwarding table

Intra-AS Routing

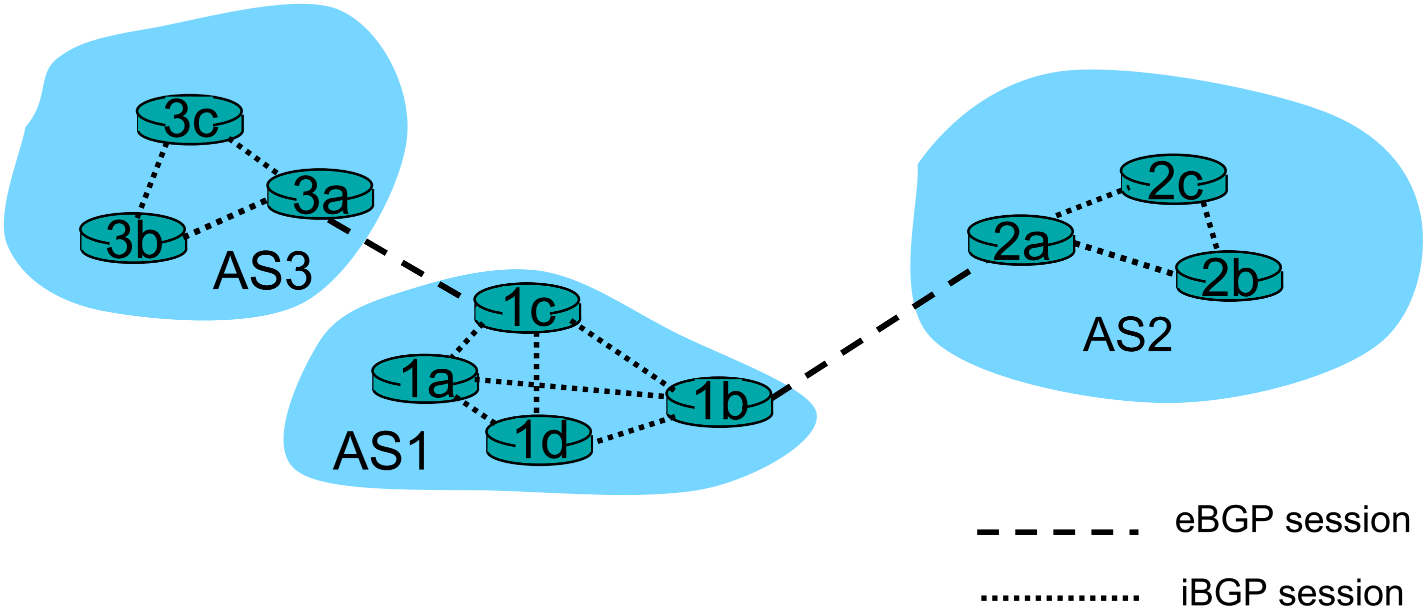
- Also known as Interior Gateway Protocols (IGP)
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol

Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): the de facto standard
- BGP provides each AS a means to:
  - Obtain subnet reachability information from neighboring ASs.
  - Propagate the reachability information to all routers internal to the AS.
  - Determine "good" routes to subnets based on reachability information and policy.
- Allows a subnet to advertise its existence to rest of the Internet: "I am here"

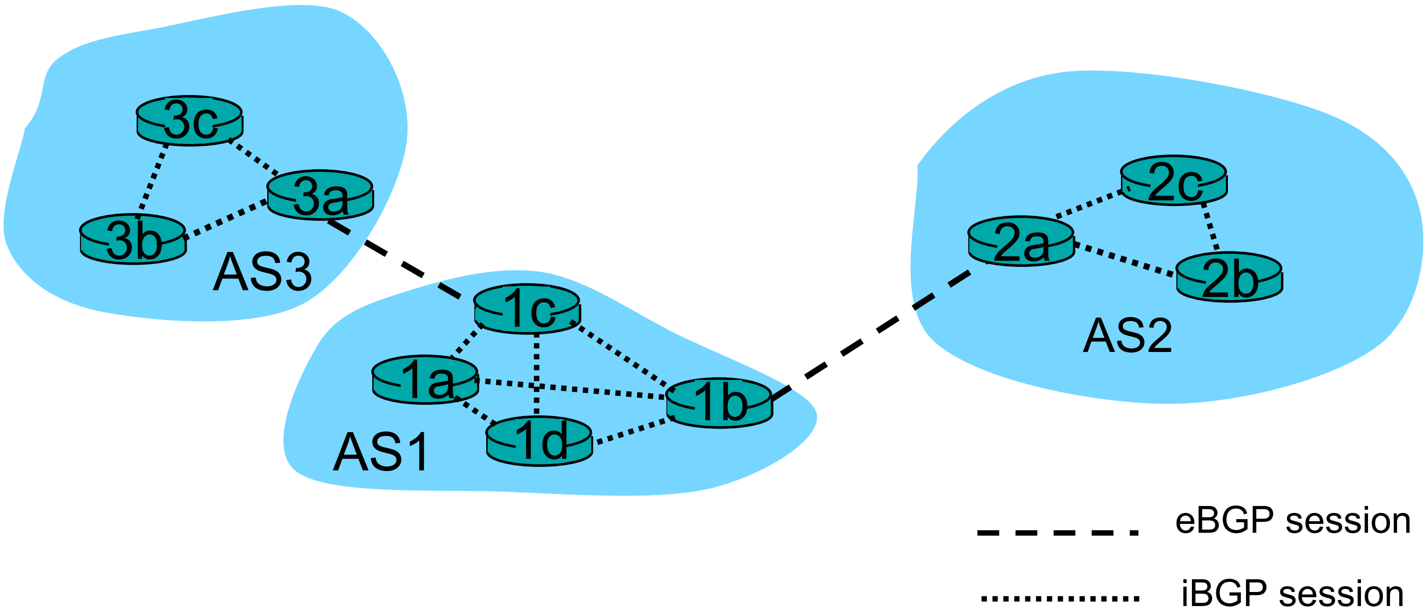
BGP basics

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP conctns: BGP sessions
- Note that BGP sessions do not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement



Distributing reachability info

- With eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.
- 1c can then use iBGP do distribute this new prefix reach info to all routers in AS1
- 1b can then re-advertise the new reach info to AS2 over the 1b-to-2a eBGP session
- When router learns about a new prefix, it creates an entry for the prefix in its forwarding table.



Path attributes & BGP routes

- When advertising a prefix, advert includes BGP attributes.
  - prefix + attributes = “route”
- Two important attributes:
  - AS-PATH: contains the ASs through which the advert for the prefix passed: AS 67 AS 17
  - NEXT-HOP: Indicates the specific internal-AS router to next-hop AS. (There may be multiple links from current AS to next-hop-AS.)
- When gateway router receives route advert, uses import policy to accept/decline.

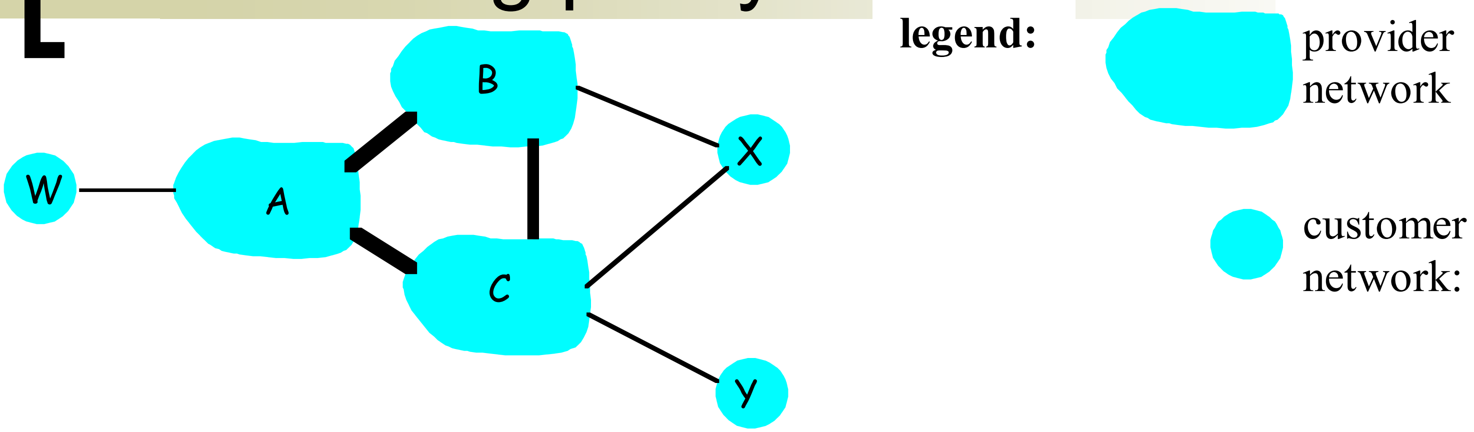
BGP route selection

- Router may learn about more than 1 route to some prefix. Router must select route.
- Elimination rules:
  1. Local preference value attribute: policy decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. Additional criteria

BGP messages

- BGP messages exchanged using TCP.
- BGP messages:
  - OPEN: opens TCP connection to peer and authenticates sender
  - UPDATE: advertises new path (or withdraws old)
  - KEEPALIVE keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - NOTIFICATION: reports errors in previous msg; also used to close connection

BGP routing policy



A,B,C are provider networks

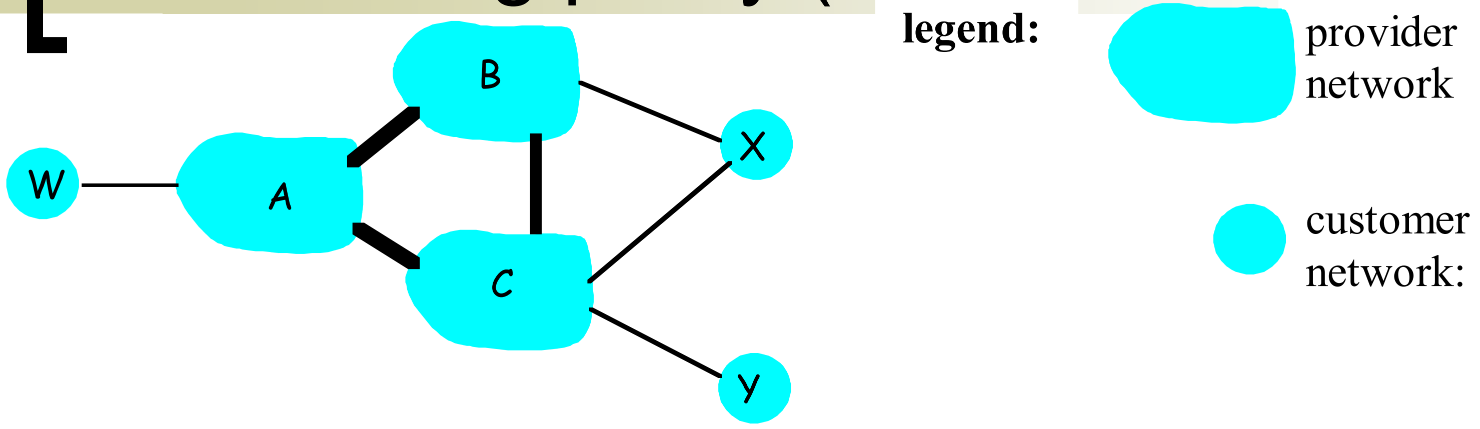
X,W,Y are customer (of provider networks)

X is dual-homed: attached to two networks

X does not want to route from B via X to C

.. so X will not advertise to B a route to C

BGP routing policy (2)



A advertises to B the path AW

B advertises to X the path BAW

Should B advertise to C the path BAW?

- No way! B gets no “revenue” for routing CBAW since neither W nor C are B’s customers
- B wants to force C to route to w via A
- B wants to route *only* to/from its customers!

Why different Intra- and Inter-AS routing ?

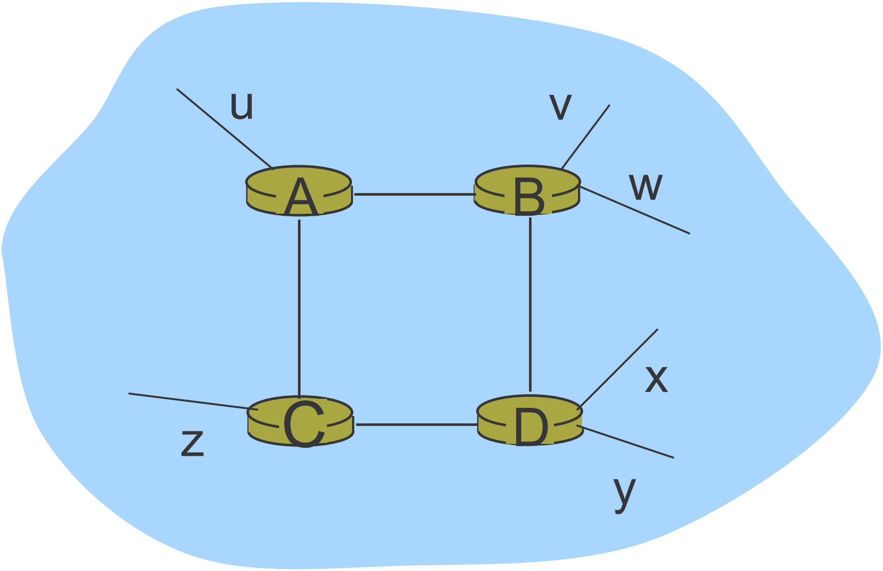
- Policy:
  - Inter-AS: admin wants control over how its traffic routed, who routes through its net.
  - Intra-AS: single admin, so no policy decisions needed
- Scale:
  - hierarchical routing saves table size, reduced update traffic
- Performance:
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance

Intra-AS Routing

- Also known as Interior Gateway Protocols (IGP)
- Most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary)

RIP (Routing Information Protocol)

- Distance vector algorithm
- Included in BSD-UNIX Distribution in 1982
- Distance metric: # of hops (max = 15 hops)



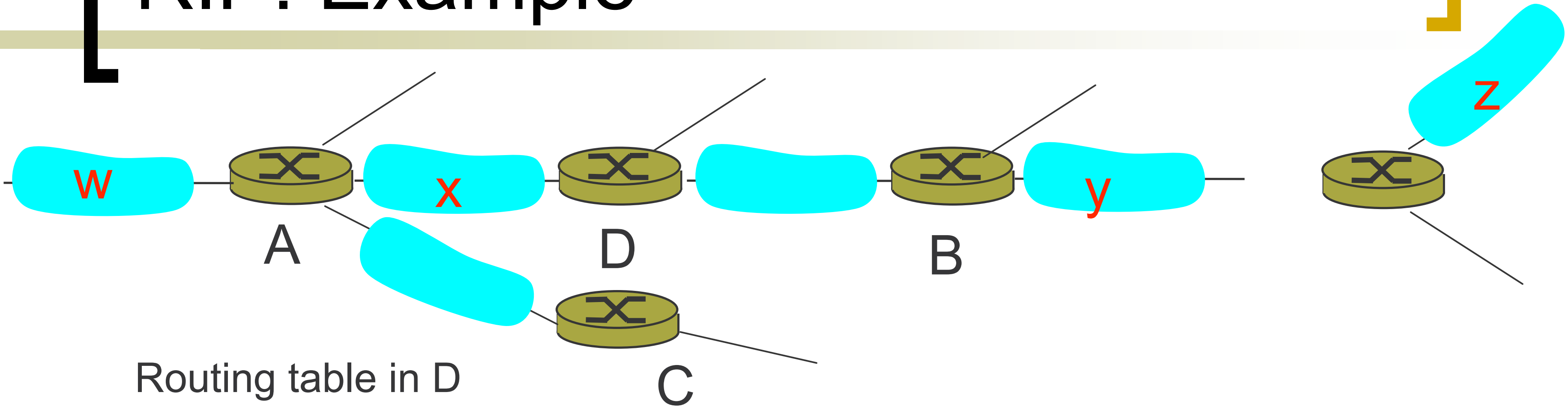
From router A to subsets:

destination	hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP advertisements

- Distance vectors: exchanged among neighbors every 30 sec via Response Message (also called advertisement)
- Each advertisement: list of up to 25 destination nets within AS

RIP: Example



Routing table in D

Dest NW	Next Router	Hops 2 Dest
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
...	...	...

Distance Vector from A to D

Dest	Next	hops
w	-	1
x	-	1
z	C	4
....	...	...

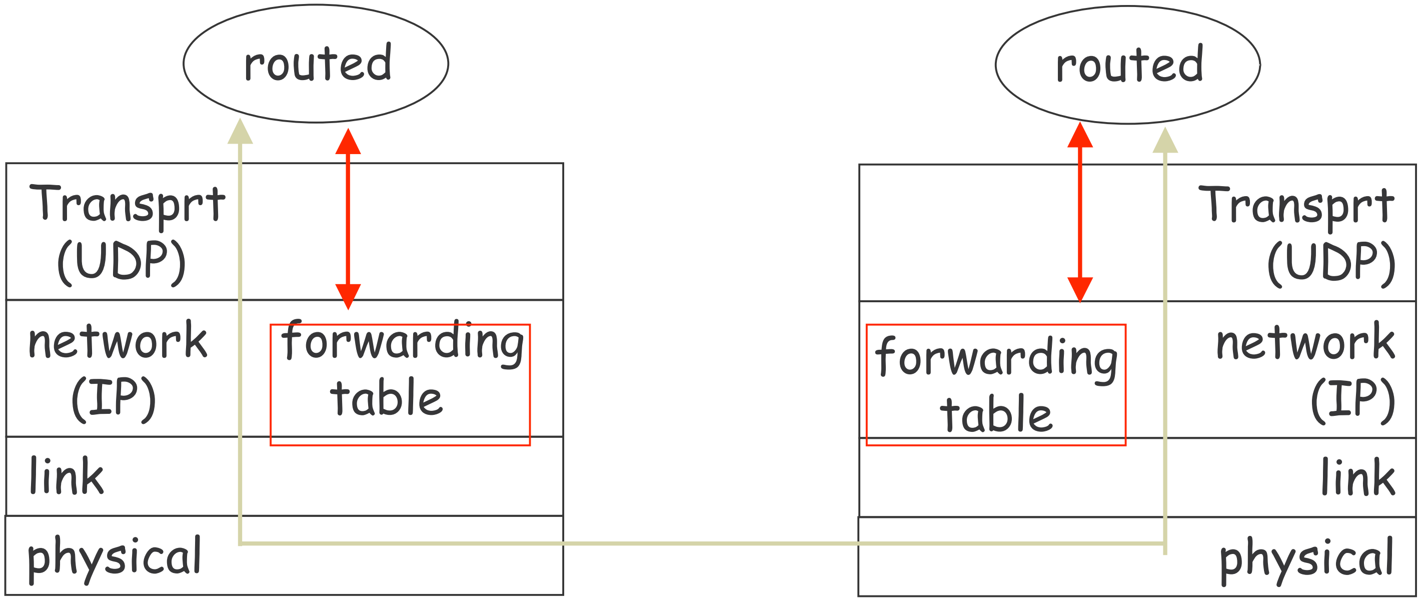
RIP: Link Failure and Recovery

If no advertisement heard after 180 sec --> neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly (?) propagates to entire net
- poison reverse used to prevent ping-pong loops
- infinite distance = 16 hops

RIP Table processing

- RIP routing tables managed by **application-level** process called route-d (daemon)
- advertisements sent in UDP packets, periodically repeated



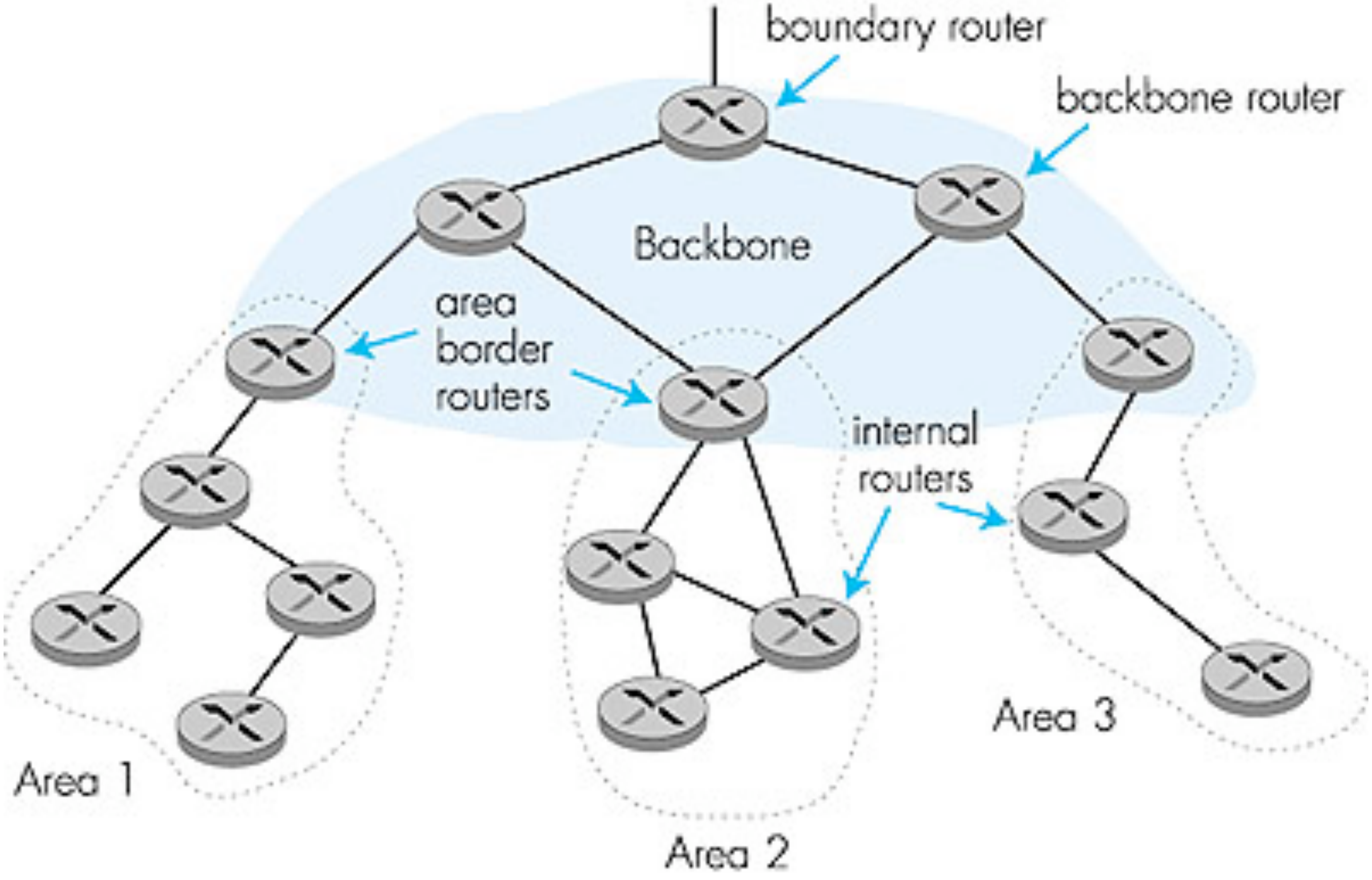
OSPF
(Open Shortest Path First)

- "open": publicly available
- Uses Link State algorithm
  - LS packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra's algorithm
- OSPF advertisement carries one entry per neighbor router
- Advertisements disseminated to entire AS (via flooding)
  - Carried in OSPF messages directly over IP (rather than TCP or UDP

OSPF “advanced” features (not in RIP)

- Security: all OSPF messages authenticated (to prevent malicious intrusion)
- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics
- Integrated uni- and multicast support:
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- Hierarchical OSPF in large domains.

Hierarchical OSPF



Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - Link-state advertisements only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas.
- **Area border routers:** “summarize” distances to nets in own area, advertise to other Area Border routers.
- **Backbone routers:** run OSPF routing limited to backbone.
- **Boundary routers:** connect to other AS’s.

IPv4 Address Translation support

- IP addresses to LAN physical addresses
- Problem
  - An IP route can pass through many physical networks
  - Data must be delivered to destination’s physical network
  - Hosts only listen for packets marked with physical interface names
    - Each hop along route
    - Destination host

IP to Physical Address Translation

- Hard-coded
  - Encode physical address in IP address
  - Ex: Map Ethernet addresses to IP addresses
    - Makes it impossible to associate address with topology
- Fixed table
  - Maintain a central repository and distribute to hosts
    - Bottleneck for queries and updates
- Automatically generated table
  - Use ARP to build table at each host
  - Use timeouts to clean up table

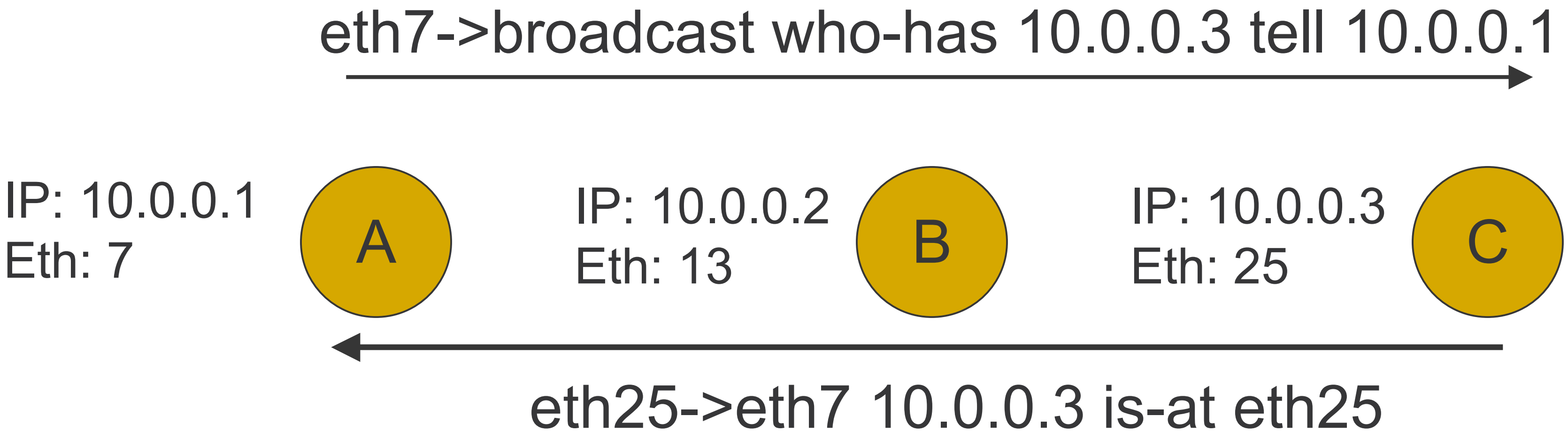
Address Resolution Protocol (ARP)

- Check table for physical address
- If address not present
  - Broadcast a query, include host’s translation
  - Wait for a response
- Upon receipt of ARP query
  - Targeted host responds with address translation
- Timeout and discard entries after O(10) minutes

ARP snooping

- Due to broadcast nature, other hosts overhear ARP exchange
- If address already present
  - Refresh entry and reset timeout
- If address not present
  - Add entry for requesting host
  - Ignore for other hosts

ARP example



eth7->broadcast who-has 10.0.0.3 tell 10.0.0.1

IP: 10.0.0.1
Eth: 7

IP: 10.0.0.2
Eth: 13

IP: 10.0.0.3
Eth: 25

eth25->eth7 10.0.0.3 is-at eth25

A’s table

10.0.0.3	eth 25

C’s table

10.0.0.1	eth 7

ARP Packet

<table>
<tr><th>0</th><th>8</th><th>16</th><th>31</th></tr>
<tr><td colspan="2">Hardware type = 1</td><td colspan="2">ProtocolType = 0x0800</td></tr>
<tr><td>HLEN = 48</td><td>PLEN = 32</td><td colspan="2">Operation</td></tr>
<tr><td colspan="4">SourceHardwareAddr (bytes 0 – 3)</td></tr>
<tr><td colspan="2">SourceHardwareAddr (bytes 4 – 5)</td><td colspan="2">SourceProtocolAddr (bytes 0 – 1)</td></tr>
<tr><td colspan="2">SourceProtocolAddr (bytes 2 – 3)</td><td colspan="2">TargetHardwareAddr (bytes 0 – 1)</td></tr>
<tr><td colspan="4">TargetHardwareAddr (bytes 2 – 5)</td></tr>
<tr><td colspan="4">TargetProtocolAddr (bytes 0 – 3)</td></tr>
</table>

Host Configuration

- Plug new host into network
  - How much information must be known?
  - What new information must be assigned?
  - How can process be automated?
- Some answers
  - Host needs an IP address (must know it)
  - Host must also
    - Send packets out of physical (direct) network
    - Thus needs physical address of router

Host Configuration

- Reverse Address Resolution Protocol (RARP)
  - Translate physical address to IP address
  - Used to boot diskless hosts
  - Host broadcasts request to boot
  - RARP server tells host the host’s own IP address
- Boot protocol (BOOTP)
  - Use UDP packets for same purpose as RARP
  - Allows boot requests to traverse routers
  - IP address of BOOTP server must be known
  - Also returns file server IP, subnet mask, and default router for host

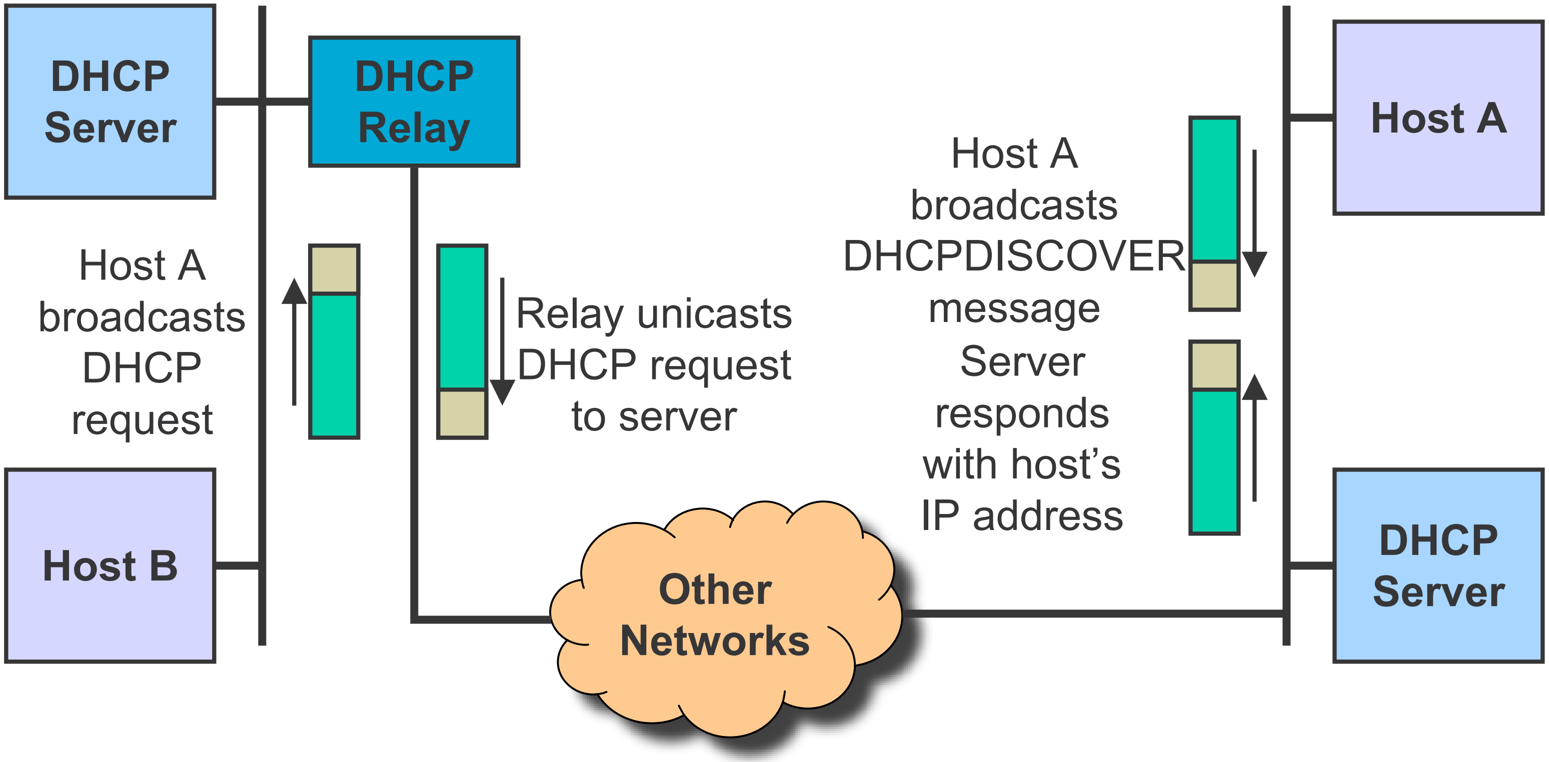
Dynamic Host Configuration Protocol (DHCP)

- A simple way to automate configuration information
  - Network administrator does not need to enter host IP address by hand
  - Good for large and/or dynamic networks

Dynamic Host Configuration Protocol (DHCP)

- New machine sends request to DHCP server for assignment and information
- Server receives
  - Directly if new machine given server's IP address
  - Through broadcast if on same physical network
  - Via DHCP relay nodes that forward requests onto the server's physical network
- Server assigns IP address and provides other info
- Can be made secure (present signed request or just a "valid" physical address)

DHCP



Host A broadcasts DHCP request

Relay unicasts DHCP request to server

Host A broadcasts DHCPDISCOVER message

Server responds with host's IP address