A **network** is a group of connected devices (like computers, phones, or printers) that can communicate with each other directly.

In IP addressing (IPv4), a network is defined by:

- A **network address** (the "base" IP of the group, e.g., 223.1.17.0).
- A **subnet mask** (e.g., /24), which tells devices: "These bits are for the network ID, and these are for individual devices."

Key Idea: Devices use network address to know if traffic is "local or needs to go to a router.

Example: In 223.1.17.0/24:

- Network address: 223.1.17.0 (all host bits set to 0).
- This covers addresses from 223.1.17.0 to 223.1.17.255 (256 total spots, but 2 reserved).
- Devices on this network can "talk" without leaving the group.

A **host** is any single device connected to a network that can send or receive data. Examples: your laptop, smartphone, server, or even a router's interface.

Each host gets a unique IP address within its network (e.g., 223.1.17.5)

- The IP is split: Network portion (fixed for the group) + Host portion (unique to the device).

Key Idea: Hosts are the "end users" of the network. Without hosts, a network is just empty space.

Reserves in a Subnet:

- First address: Network ID (not for hosts).
- Last address: Broadcast (for "shouting" to all hosts, not for a single device).
- Middle ones: Usable for hosts (e.g., in /26 subnet: 62 usable hosts).

What is Subnetting?

In Simple Terms: Subnetting breaks a large IP network into smaller, more manageable sub-networks (subnets). Each subnet acts like its own mini-network with its own range of addresses.

Why It Exists: A huge network (e.g., 256 addresses) might be too big for one office floor—it's wasteful and hard to manage. Subnetting lets you create:

- Smaller groups for different departments (e.g., HR gets 50 addresses, IT gets 100).
- Better security (traffic stays local to the subnet).
- Efficiency (routers handle less "broadcast" noise).

Real-World Analogy: Your city's zip code (big network) gets split into neighborhoods (subnets) for mail sorting—each neighborhood has its own block of addresses.

CIDR notation

CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

subnet part ⟷ host part

11001000 00010111 00010000 00000000

200.23.16.0/23

IP addresses: how to get one?

Key Idea: No one starts from scratch. IP addresses are handed out in a pyramid: Global registries → ISPs → You/Your Organization.

Q: how does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

ISP's block	11001000	00010111	00010000	00000000	200.23.16.0/20
Organization 0	11001000	00010111	00010000	00000000	200.23.16.0/23
Organization 1	11001000	00010111	00010010	00000000	200.23.18.0/23
Organization 2	11001000	00010111	00010100	00000000	200.23.20.0/23
...		.....		....	....
Organization 7	11001000	00010111	00011110	00000000	200.23.30.0/23

Problem 1

You only need to care about how to allocate the last 8 bits, no overlap among subnet1, subnet2, and subnet3.

Subnet1: 223.1.17.0/26 → 00 000000 $2^6 = 64 > 62$

Subnet2: 223.1.17.128/25 → 1 0000000 $2^7 = 128 > 106$

Subnet3: 223.1.17.64/28 → 0100 0000 $2^4 = 16 > 14$

223.1.17.0/26	11011111	00000001	00010001	00000000
223.1.17.128/25	11011111	00000001	00010001	10000000
223.1.17.64/28	11011111	00000001	00010001	01000000

Other choices for subnet3:

223.1.17.80/28 → 0101 0000 $2^4 = 16 > 14$

223.1.17.96/28 → 0110 0000 $2^4 = 16 > 14$

223.1.17.112/28 → 0111 0000 $2^4 = 16 > 14$

Hierarchical routing

our routing study thus far - idealization

- all routers identical
- network "flat"

… *not* true in practice

scale: with 600 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 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:

- at "edge" of its own AS
- has link to router in another AS

Why different Intra-, 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

Prob. 2 (Offline Reading)

- Policy: Among ASs, policy issues dominate. It may well be important that traffic originating in a given AS not be able to pass through another specific AS. Similarly, a given AS may want to control what transit traffic it carries between other ASs. Within an AS, everything is nominally under the same administrative control and thus policy issues a much less important role in choosing routes with in AS.

- Scale: The ability of a routing algorithm and its data structures to scale to handle routing to/among large numbers of networks is a critical issue in inter-AS routing. Within an AS, scalability is less of a concern. For one thing, if a single administrative domain becomes too large, it is always possible to divide it into two ASs and perform inter-AS routing between the two new ASs.

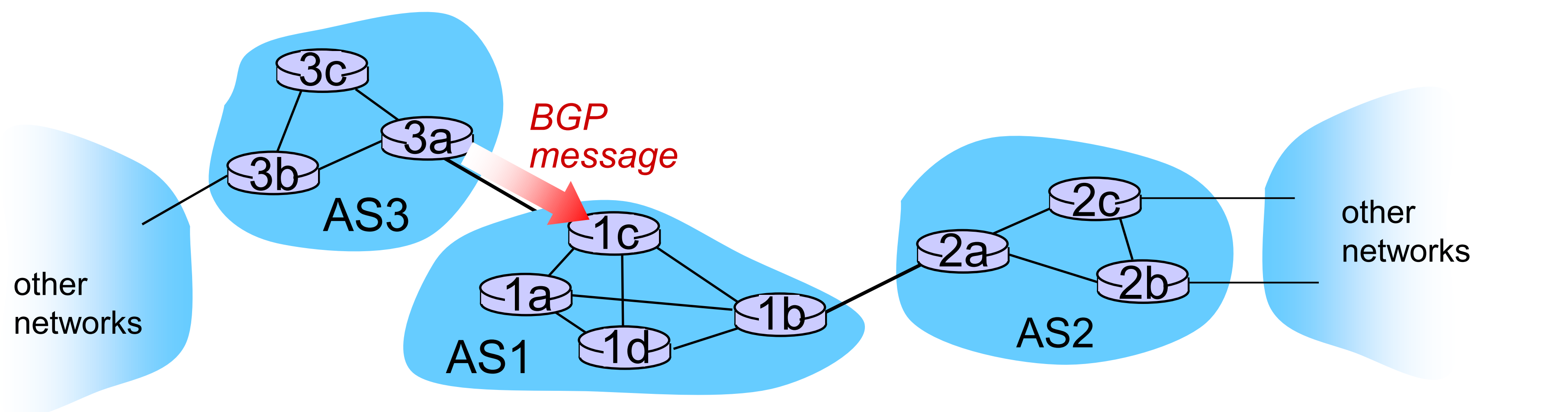
- Performance: Because inter-AS routing is so policy oriented, the quality (for example, performance) of the routes used is often of secondary concern (that is, a longer or more costly route that satisfies certain policy criteria may well be taken over a route that is shorter but does not meet that criteria). Indeed, we saw that among ASs, there is not even the notion of cost (other than AS hop count) associated with routes. Within a single AS, however, such policy concerns are of less importance, allowing routing to focus more on the level of performance realized on a route.

Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASs.
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and policy.
- allows subnet to advertise its existence to rest of Internet: *"I am here"*

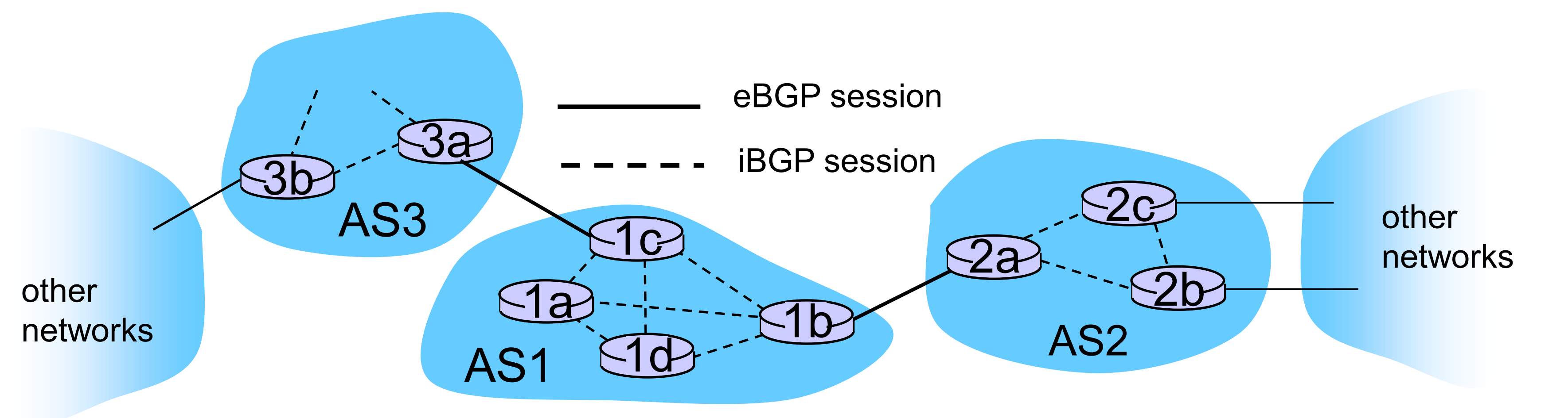
BGP basics

- BGP session: two BGP routers ("peers") exchange BGP messages:
  - advertising *paths* to different destination network prefixes ("path vector" protocol)
  - exchanged over semi-permanent TCP connections
- when AS3 advertises a prefix to AS1:
  - AS3 *promises* it will forward datagrams towards that prefix
  - AS3 can aggregate prefixes in its advertisement



BGP basics: distributing path information

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



Problem 3

- A) eBGP
- B) iBGP
- C) eBGP
- D) iBGP