This Lecture

- **Focus**
  - How do we make routing scale?

- **IP Addressing**
  - Hierarchy (prefixes, class A, B, C, subnets)
  - Also allocation (DHCP, ARP)

Scalability Concerns

- Routing burden grows with size of an internetwork
  - Size of routing tables
  - Volume of routing messages
  - Amount of routing computation

- To scale to the size of the Internet, apply:
  - Hierarchical addressing
  - Use of structural hierarchy
  - Route aggregation
IP Addresses

- Reflect location in topology; used for scalable routing
  - Unlike “flat” Ethernet addresses

- Interfaces on same network share prefix
  - Prefix administratively assigned (IANA or ISP)
  - Addresses globally unique

- Routing only advertises entire networks by prefix
  - Local delivery in a single “network” doesn’t involve router
  - (will make “network” precise later on)

Getting an IP address

- Old fashioned way: sysadmin configured each machine

- Dynamic Host Configuration Protocol (DHCP)
  - One DHCP server with the bootstrap info
    - Host address, gateway address, subnet mask, …
    - Find it using broadcast
  - Addresses may be leased; renew periodically

- “Stateless” Autoconfiguration (in IPv6)
  - Get rid of server – reuse Ethernet addresses for lower portion of address (uniqueness) and learn higher portion from routers

Address Resolution Protocol (ARP)

- On a single link, need Ethernet addresses to send a frame
  … source is a given, but what about destination?
  - Requires mapping from IP to MAC addresses

- ARP is a dynamic approach to learn mapping
  - Node A sends broadcast query for IP address X
  - Node B with IP address X replies with its MAC address M
  - A caches (X, M); old information is timed out (~15 mins)
  - Also: B caches A’s MAC and IP addresses, other nodes refresh
ARP Example

- To send first message use ARP to learn MAC address
- For later messages (common case) don’t need to ARP

IPv4 Address Formats

- 32 bits written in “dotted quad” notation, e.g., 18.31.0.135

IPv6 Address Format

- 128 bits written in 16 bit hexadecimal chunks
- Still hierarchical, just more levels
Network Example

- **Network number:** 128.96.0.0
- **Network number:** 128.97.0.0
- **Network number:** 128.98.0.0

Updated Forwarding Routine

- **Used to be** “look up destination address for next hop”
- **Now addresses** have network and host portions:
  - If host: if destination network is the same as the host network, then deliver locally (without router). Otherwise send to the router.
  - If router: look up destination network in routing table to find next hop and send to next router. If destination network is directly attached then deliver locally.
- (Note that it will get a little more complicated later)

Subnetting – More Hierarchy

- **Split up one network number into multiple physical networks**
- **Helps allocation efficiency** – can hand out subnets
- **Rest of internet does not see subnet structure**
  - subnet is purely internal to network
  - aggregates routing 2560

Network number | Host number
--- | ---
Class B address

00000000 01111111 11111111 11111111 00000000

Subnet mask (255.255.255.0)

Network number | Subnet ID | Host ID
--- | --- | ---

Subnetted address
Updated Forwarding Routine

- Used to know network from address (class A, B, C)
- Now need to “search” routing table for right subnet
  - If host: easy, just substitute “subnet” for “network”
  - If router: search routing table for the subnet that the destination belongs to, and use that to forward as before
- (Note that it will get a little more complicated later :)

CIDR (Supernetting)

- CIDR = Classless Inter-Domain Routing
- Generalize class A, B, C into prefixes of arbitrary length; now must carry prefix length with address
- Aggregate adjacent advertised network routes
  - e.g., ISP has class C addresses 192.4.16 through 192.4.31
  - Really like one larger 20 bit address class ...
  - Advertise as such (network number, prefix length)
  - Reduces size of routing tables
- But IP forwarding is more involved
  - Based on Longest Matching Prefix operation
CIDR Example

- X and Y routes can be aggregated because they form a bigger contiguous range.

- But aggregation isn’t always possible. Why?
  - can only aggregate power of 2

---

IP Forwarding Revisited

- Routing table now contains routes to “prefixes”
  - IP address and length indicating what bits are fixed

- Now need to “search” routing table for longest matching prefix, only at routers
  - Search routing table for the prefix that the destination belongs to, and use that to forward as before
  - There can be multiple matches; take the longest prefix

- This is the IP forwarding routine used at routers.

---

Key Concepts

- Hierarchical address allocation helps routing scale
  - Addresses are constrained by topology
  - Only need to advertise and compute routes for networks
  - Hide internal structure within a domain via subnets
  - Keep host simple and let routers worry about routing

- ARP learns the mapping from IP to MAC address