Announcements

• Homework #2 out, due Oct. 11
  – Your solutions to be submitted using the standard EECS Unix interface, not via email
• Lectures 8 & 9 swapped: we’ll now first do Email & FTP (as well as finishing DNS) before The Web

Goals of Today’s Lecture

• IP addresses
  – Dotted-quad notation
  – IP prefixes for aggregation
• Address allocation
  – Classful addresses
  – Classless InterDomain Routing (CIDR)
  – Growth in the number of prefixes over time
• Packet forwarding
  – Forwarding tables
  – Longest-prefix match forwarding
  – Where forwarding tables come from

Designing IP’s Addresses

• Question #1: what should an address be associated with?
  – E.g., a telephone number is associated not with a person but with a handset
• Question #2: what structure should addresses have? What are the implications of different types of structure?
• Question #3: who determines the particular addresses used in the global Internet? What are the implications of how this is done?

IP Addresses (IPv4)

• A unique 32-bit number
• Identifies an interface (on a host, on a router, …)
• Represented in dotted-quad notation. E.g., 12.34.158.5:

  12  34  158  5
  
  0000110 00110 10011110 00000101

Grouping Related Hosts

• The Internet is an “inter-network”
  – Used to connect networks together, not hosts
  – Needs a way to address a network (i.e., group of hosts)
Scalability Challenge

- Suppose hosts had arbitrary addresses
  - Then every router would need a lot of information
    - ...to know how to direct packets toward the host

Hierarchical Addressing in U.S. Mail

- Addressing in the U.S. mail
  - Zip code: 94704
  - Street: Center Street
  - Building on street: 1947
  - Location in building: Suite 600
  - Name of occupant: Vern Paxson

- Forwarding the U.S. mail
  - Deliver letter to the post office in the zip code
  - Assign letter to mailman covering the street
  - Drop letter into mailbox for the building/room
  - Give letter to the appropriate person

Hierarchical Addressing: IP Prefixes

- Divided into network & host portions (left and right)
- 12.34.158.0/24 is a 24-bit prefix with 2^8 addresses
  - Terminology: “Slash 24”

IP Address and a 24-bit Subnet Mask

<table>
<thead>
<tr>
<th>Address</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.34.158.5</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>12.34.157.5</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

Scalability Improved

- Number related hosts from a common subnet
  - 1.2.3.0/24 on the left LAN
  - 5.6.7.0/24 on the right LAN

Easy to Add New Hosts

- No need to update the routers
  - E.g., adding a new host 5.6.7.213 on the right
  - Doesn’t require adding a new forwarding entry
Classful Addressing

- Originally, only fixed allocation sizes
  - Class A: 0* (first quad ranges from 0-127)
    - Very large /8 blocks (e.g., MIT has 18.0.0.0/8)
  - Class B: 10* (first quad 128-191)
    - Large /16 blocks (e.g., UCB has 128.32.0.0/16)
  - Class C: 110* (first quad 192-223)
    - Small /24 blocks (e.g., ICIR has 192.150.187.0/24)
  - Class D: 1110*
    - Multicast groups
  - Class E: 11110*
    - Reserved for future use

- This is why we use dotted-quad notation
- What problems can classful addressing lead to?
  - Only comes in 3 sizes
  - Routers can end up knowing about a lot of class C’s

Classless Inter-Domain Routing (CIDR)

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

IP Address : 12.4.0.0     IP Mask: 255.254.0.0

<table>
<thead>
<tr>
<th>Address</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001100</td>
<td>11111111</td>
</tr>
<tr>
<td>00001000</td>
<td>11111111</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
</tr>
</tbody>
</table>

Written as 12.4.0.0/15

CIDR: Hierarchical Address Allocation

- Prefixes are key to Internet scalability
  - Addresses allocated in contiguous chunks (prefixes)
  - Routing protocols and packet forwarding based on prefixes

Scalability: Address Aggregation

Routers in the rest of the Internet just need to know how to reach 201.10.0.0/21. The provider can direct the IP packets to the appropriate customer.

CIDR Deployed (1994-1996): Much Flatter

But, Aggregation Not Always Possible


Good use of aggregation, and peer pressure in CIDR report*

But, Aggregation Not Always Possible


But, Aggregation Not Always Possible

Multihomed customer with 201.10.6.0/23 has two providers. Other parts of the Internet need to know how to reach these destinations through both providers.


Internet boom, increased multi-homing. What next?

Long-Term View (1989-2005): Post-Boom

5 Minute Break

Questions Before We Proceed?

Scalability Through Non-Uniform Hierarchy

- Hierarchical addressing
  - Critical for scalable system
  - Don’t require everyone to know everyone else
  - Reduces amount of updating when something changes

- Non-uniform hierarchy
  - Useful for heterogeneous networks of different sizes
  - Initial class-based addressing was far too coarse
  - Classless InterDomain Routing (CIDR) gains much more flexibility
Address Allocation

Obtaining a Block of Addresses
• Separation of control
  – Prefix: assigned to an institution
  – Addresses: assigned by the institution to their nodes
• Who assigns prefixes?
  – Internet Corporation for Assigned Names and Numbers
    • Allocates large address blocks to Regional Internet Registries
    • ICANN is politically charged
  – Regional Internet Registries (RIRs)
    • E.g., ARIN (American Registry for Internet Numbers)
      • Allocates address blocks within their regions
      • Allocated to Internet Service Providers and large institutions
    – Internet Service Providers (ISPs)
      • Allocate address blocks to their customers (could be recursive)

Figuring Out Who Owns an Address
• Address registries
  – Public record of address allocations
  – Internet Service Providers (ISPs) should update when giving addresses to customers
  – However, records are notoriously out-of-date
• Ways to query
  – UNIX: "whois -h whois.arin.net 169.229.60.27"
  – http://www.arin.net/whois/
  – ...

Example Output for 169.229.60.27
University of California, Office of the President UCNET-BLK (NET-169-228-0-0-1)
  169.228.0.0 - 169.233.255.255
University of California at Berkeley ISTDATA (NET-169-229-0-0-1)
  169.229.0.0 - 169.229.255.255

Example Output for ISTDATA
OrgName: University of California at Berkeley
OrgID: UCAB-1
Address: IST Communication and Network Services
  Address: ATTN Network Services Group
  Address: 2484 Shattuck Ave, #1640
  City: Berkeley
  StateProv: CA
  PostalCode: 94720-1640
Country: US
NetRange: 169.229.0.0 - 169.229.255.255
CIDR: 169.229.0.0/16
NetName: ISTDATA
NetHandle: NET-169-229-0-0-1
Parent: NET-169-228-0-0-1
NetType: Reassigned

Example Output for ISTDATA, con’t
NameServer: ADNS1.BERKELEY.EDU
NameServer: ADNS2.BERKELEY.EDU
NameServer: UCB-NS.NYU.EDU
Comment: DMCA Designated Agent is Jacqueline Craig <policy@uclink.berkeley.edu>
RegDate: 1996-05-01
Updated: 2006-09-13
OrgTechHandle: UCB-NOC-ARIN
OrgTechName: IST Communication and Network Services
OrgTechPhone: +1-510-643-3267
OrgTechEmail: noc@nak.berkeley.edu
Are 32-bit Addresses Enough?

- Not all that many unique addresses
  - \(2^{32} = 4,294,967,296\) (just over four billion)
  - Plus, some (many) reserved for special purposes
  - And, addresses are allocated in larger blocks
- And, many devices need IP addresses
  - Computers, PDAs, routers, tanks, toasters, ...
- Long-term solution (perhaps): larger address space
  - IPv6 has 128-bit addresses \((2^{128} = 3.403 \times 10^{38})\)
- Short-term solutions: limping along with IPv4
  - Private addresses
  - Network address translation (NAT)
  - Dynamically-assigned addresses (DHCP)

Hard Policy Questions

- How much address space per geographic region?
  - Equal amount per country?
  - Proportional to the population?
  - What about addresses already allocated?
- Address space portability?
  - Keep your address block when you change providers?
  - Pro: avoid having to renumber your equipment
  - Con: reduces the effectiveness of address aggregation
- Keeping the address registries up to date?
  - What about mergers and acquisitions?
  - Delegation of address blocks to customers?
  - As a result, the registries are often out of date

Packet Forwarding

- Each router has a forwarding table
  - Maps destination addresses...
  - ... to outgoing interfaces
- Upon receiving a packet
  - Inspect the destination IP address in the header
  - Index into the table
  - Determine the outgoing interface
  - Forward the packet out that interface
- Then, the next router in the path repeats
  - And the packet travels along the path to the destination

Separate Table Entries Per Address

- If a router had a forwarding entry per IP address
  - Match destination address of incoming packet
  - ... to the forwarding-table entry
  - ... to determine the outgoing interface

Separate Entry Per 24-bit Prefix

- If the router had an entry per 24-bit prefix
  - Look only at the top 24 bits of the destination address
  - Index into the table to determine the next-hop interface
  - Could also do this based on class (A/B/C/...)

Forwarding table
CIDR Makes Packet Forwarding Harder

- Router can no longer determine network prefix just by inspecting the address
- Forwarding table may have multiple matches
  - E.g., table entries for 201.10.0.0/21 and 201.10.6.0/23
  - The IP address 201.10.6.17 would match both!

Longest-Prefix-Match Forwarding

- Router needs to identify longest-matching prefix

Simple Algorithms Are Too Slow

- Scan the forwarding table one entry at a time
  - See if the destination matches the entry
  - If so, check the size of the mask for the prefix
  - Keep track of the entry with longest-matching prefix
- Overhead is linear in size of the forwarding table
  - Today, that means 150,000-200,000 entries!
  - And, the router may have just a few nanoseconds...
  - ... before the next packet is arriving
- Need greater efficiency to keep up with line rate
  - Better algorithms
  - Hardware implementations

Patricia Tree

- Store the prefixes as a tree
  - One bit for each level of the tree
  - Some nodes correspond to valid prefixes
  - ... which have next-hop interfaces in a table
- When a packet arrives
  - Traverse the tree based on the destination address
  - Stop upon reaching the longest matching prefix

Even Faster Lookups

- Patricia tree is faster than linear scan
  - Proportional to number of bits in the address
- Patricia tree can be made faster
  - Can make a k-ary tree
    - E.g., 4-ary tree with four children (00, 01, 10, and 11)
  - Faster lookup, though requires more space
- Can use special hardware
  - Content Addressable Memories (CAMs)
  - Allows look-ups on a key rather than flat address
- Huge innovations in the mid-to-late 1990s
  - After CIDR was introduced (in 1994)
  - ... and longest-prefix match was a major bottleneck

Where do Forwarding Tables Come From?

- Routers have forwarding tables
  - Map prefix to outgoing link(s)
- Entries can be statically configured
  - E.g., “map 12.34.158.0/24 to Serial0/0.1”
- But, this doesn’t adapt
  - To failures
  - To new equipment
  - To the need to balance load
- That is where other technologies come in...
  - Routing protocols, DHCP and ARP (later in course)
How Does Sending End Host Forward?

- End host with single network interface
  - PC with an Ethernet link
  - Laptop with (just) a wireless link
- Don’t need to run a routing protocol
  - Packets to the host itself (e.g., 1.2.3.4/32)
    • Delivered locally
  - Packets to other hosts on the LAN (e.g., 1.2.3.0/24)
    • Sent out the interface with LAN address (ARP)
  - Packets to external hosts (e.g., 0.0.0.0/0)
    • Sent out interface to local gateway
- How this information is learned
  - Static setting of address, subnet mask, and gateway
  - Dynamic Host Configuration Protocol (DHCP)

What About Reaching the End Hosts?

- How does the last router reach the destination?

  - Each interface has a persistent, global identifier
    - MAC address (Media Access Control)
      - Programmed into adaptor (ROM/EEPROM)
        - Usually flat address structure (i.e., no hierarchy)
  - Constructing an address resolution table
    - Mapping MAC address to/from IP address
    - Address Resolution Protocol (ARP)

Summary

- IP address
  - A 32-bit number identifying an interface
  - Allocated in prefixes
    - Non-uniform hierarchy for scalability and flexibility
  - Packet forwarding
    - Based on IP prefixes
      - Longest-prefix-match forwarding
  - Issues to be covered later
    - Populating the forwarding table (routing)
    - How hosts get their addresses (DHCP)
      - How to map from an IP address to a link address (ARP)

Next Lecture

- Transport Protocols & DNS
- Read P&D: 2.5, 5.1, 9.1