Internet Addressing & Naming

CS587X Lecture 2
Fall, 2002

Network Layer

- Data grouped into “packets”
- Network-uniform addresses
  - Requires translation to link-layer addresses
- Packets can be routed between different links
  - E.g., Packets can be routed from Ethernet to PPP to Token Ring to FDDI

LAN Addresses

- Each Local Area Network has its own layer 2 address
  - Ethernet, Token Ring, FDDI: 6-byte, top 3 bytes assigned to manufacturer
  - Arcnet: 1-byte assigned by network administrator
- TCP/IP layer 3 addresses
  - Allow programmers to address machines regardless of LAN
  - Single IP address per machine interface
  - Groups of IP addresses typically assigned to organization by Internet provider

IP Address Structure

- IPv4
- 32-bit (4 bytes or octets)
- Presentation format
  - Four numbers, 0-255, separated by dots
- Classes
  - A: 0.0.0.0-127.255.255.255
  - B: 128.0.0.0-191.255.255.255
  - C: 192.0.0.0-223.255.255.255
  - D (Multicast): 224.0.0.0-239.255.255.255
  - E (Reserved): 240.0.0.0-247.255.255.255

IP Address Classes

- Class A
  - 0nnnnnnn hhhhhhhh hhhhhhhh hhhhhhhh
  - First bit 0; 7 network bits; 24 host bits
  - Initial byte: 0 - 127
  - 126 Class As exist (0 and 127 are reserved)
  - 16,777,214 hosts on each Class A
- Class B
  - 10nnnnnn nnnnnnnn hhhhhhhh hhhhhhhh
  - First two bits 10; 14 network bits; 16 host bits
  - Initial byte: 128 - 191
  - 16,384 Class Bs exist
  - 65,532 hosts on each Class B

IP Address Classes

- Class C
  - 110nnnnn nnnnnnnn nnnnnnnn hhhhhhhh
  - First three bits 110; 21 network bits; 8 host bits
  - Initial byte: 192 - 223
  - 2,097,152 Class Cs exist
  - 254 hosts on each Class C
- Class D
  - 1110mmmm mmmmmmmm mmmmmmmm mmmmmmmm
  - First four bits 1110; 28 multicast address bits
  - Initial byte: 224 - 247
  - Class Ds are multicast addresses - see RFC 1112

IP Addresses Generalized

- IP address space exhaustion
- IP addresses no longer assigned by class
- Now assigned as a network number and number of network bits (CIDR – RFC 1519)
  - E.g. 129.186.0.0/16 (ISU), 192.188.162.0/24 (ISU Research Park), 63.224.0.0/13 (USWest)

Special IP Addresses

- Special addresses
  - 0.0.0.0 - “This host”
  - 255.255.255.255 - “All hosts”
  - Host part of all 1's - “All hosts on this subnet”
  - 127.0.0.1 - “localhost”
- Reserved addresses
  - Can be used locally (behind Network Address Translator, for example)
    - 192.168.0.0-192.168.255.255
    - 172.16.0.0-172.31.255.255
    - 10.0.0.0-10.255.255.255
  - Not routed through the Internet

Domain Name System

- Maps Internet Addresses to names (and vice versa)
- Replaced hosts.txt file distributed by a central registrar
- Implemented as hierarchical, distributed database
- Root (.com, .org, etc.) and country-code (.uk, .us, .au, etc.) top-level domains
- Registrations for each top-level domain handled by various organizations

Domain Name Resolution

- Organizations have local name servers
- Applications query the local name server
  - Local name server caches entries
  - An answer may be found in the cache
- If not cached, local name server recursively queries name servers, starting at the known root nameservers, until an answer is found
- E.g., www.yahoo.com
  - Local name server queries root name server for .com
  - Queries .com server for yahoo.com
  - Queries yahoo.com server for www.yahoo.com

Hostname Lookups in C

- gethostbyname(3)

```
#include <netinet/in.h>
#include <netdb.h>

struct hostent *h;
struct inet_addr in;
int i;

h = gethostbyname("www.yahoo.com");
if (h != NULL) {
  if (h->h_addrtype == AF_INET && h->h_length == sizeof(struct
    in_addr))
  for (i = 0; h->h_addr_list[i] != NULL; i++) {
    memcpy(&in, h->h_addr_list[i], sizeof(in));
    printf("IP address #%d for www.yahoo.com: %s\n", i,
      inet_ntoa(in));
  }
} else {
  printf("Could not resolve www.yahoo.com\n");
}
```

Reverse Lookups

- gethostbyaddr(3)

```
#include <netinet/in.h>
#include <netdb.h>

struct hostent *h;
struct inet_addr in;
int i;

h = gethostbyaddr("129.186.3.4");
if (h != NULL) {
  printf("Hostname for 129.186.3.4: %s\n", i, h->h_name);
} else {
  printf("Could not reverse lookup 129.186.3.4\n");
}
```

Summary

- IP Addresses
  - Presentation format
  - Classes, CIDR
  - Special addresses
- DNS
  - Hierarchical database structure
  - DNS resolution process
  - Lookups in C
  - Reverse lookups in C