Naming and Addressing

An Engineering Approach to Computer Networking

Outline

- Names and addresses
- Hierarchical naming
- Addressing
- Addressing in the telephone network
- Addressing in the Internet
- ATM addresses
- Name resolution
- Finding datalink layer addresses

Names and addresses

- Names and addresses both uniquely identify a host (or an interface on the host)
- `%nslookup`
  - `Default Server:  DUSK.CS.CORNELL.EDU`
  - `Address:  128.84.227.13`

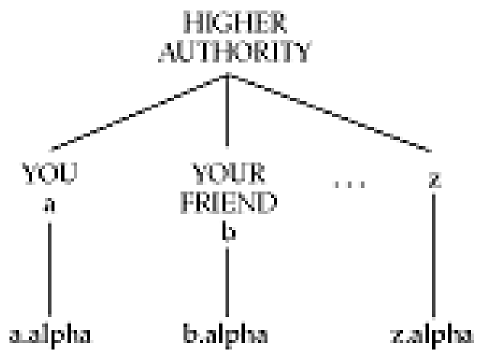
  - `> underarm.com`
  - `Name:    underarm.com`
  - `Address:  206.128.187.146`
- *Resolution*: the process of determining an address from a name

Why do we need both?

- Names are long and human understandable
  - wastes space to carry them in packet headers
  - hard to parse
- Addresses are shorter and machine understandable
  - if fixed size, easy to carry in headers and parse
- Indirection
  - multiple names may point to same address
  - can move a machine and just update the resolution table

Hierarchical naming

- Goal: give a globally unique name to each host
- Naïve approach: ask other naming authorities before choosing a name
  - doesn't scale (why?)
  - not robust to network partitions
- Instead carve up *name space* (the set of all possible names) into mutually exclusive portions => hierarchy

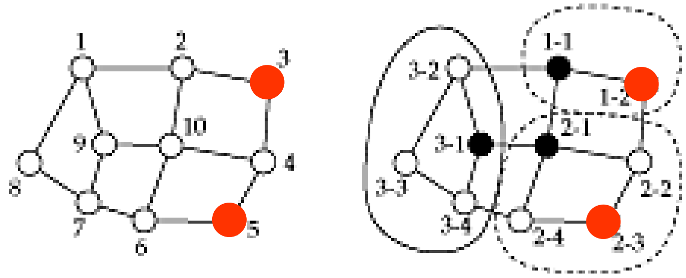


Hierarchy

- A wonderful thing!
  - scales arbitrarily
  - guarantees uniqueness
  - easy to understand
- Example: Internet names
  - use *Domain name system (DNS)*
  - global authority (Network Solutions Inc.) assigns top level domains to naming authorities (e.g. .edu, .net, .cz etc.)
  - naming authorities further carve up their space
  - all names in the same domain share a unique *suffix*

Addressing

- Addresses need to be globally unique, so they are also hierarchical
- Another reason for hierarchy: *aggregation*
  - reduces size of routing tables
  - at the expense of longer routes

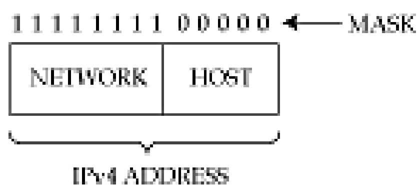


Addressing in the telephone network

- Telephone network has only addresses and no names (why?)
- E.164 specifications
- ITU assigns each country a unique *country code*
- Naming authority in each country chooses unique area or city prefixes
- Telephone numbers are variable length
  - this is OK since they are only used in call establishment
- Optimization to help dialing:
  - reserve part of the lower level name space to address top level domains
  - e.g. in US, no area code starts with 011, so 011 => international call => all other calls need fewer digits dialed

Addressing in the Internet

- Every host interface has its own IP address
- Routers have multiple interfaces, each with its own IP address
- Current version of IP is version 4, addresses are IPv4 addresses



- 4 bytes long, two part hierarchy
  - network number and host number
  - boundary identified with a *subnet* mask
  - can aggregate addresses within subnets

Address classes

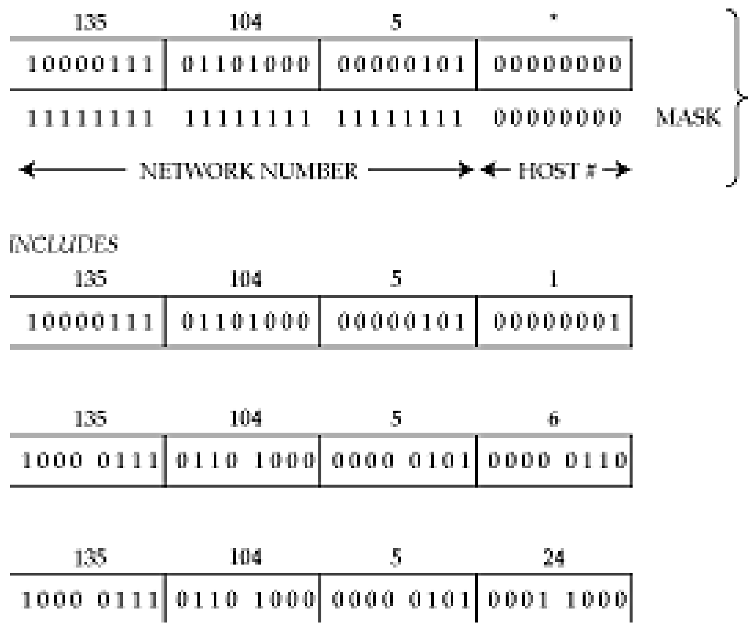
- First cut
  - fixed network-host partition, with 8 bits of network number
  - too few networks!
- Generalization
  - Class A addresses have 8 bits of network number
  - Class B addresses have 16 bits of network number
  - Class C addresses have 24 bits of network number
- Distinguished by leading bits of address
  - leading 0 => class A (first byte < 128)
  - leading 10 => class B (first byte in the range 128-191)
  - leading 110 => class C (first byte in the range 192-223)

Address evolution

- This scheme was too inflexible
- Three extensions
  - subnetting
  - CIDR
  - dynamic host configuration

Subnetting

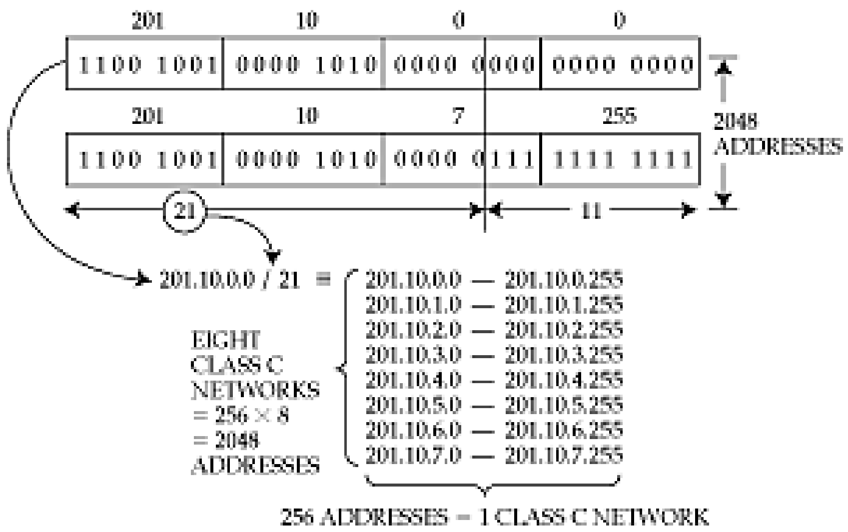
- Allows administrator to cluster IP addresses *within* its network



CIDR

- Scheme forced medium sized nets to choose class B addresses, which wasted space
- Address space exhaustion
- Solution
  - allow ways to represent a set of class C addresses as a block, so that class C space can be used
  - use a CIDR mask
  - idea is very similar to subnet masks, except that all routers must agree to use it
    - subnet masks are not visible outside the network (why?)

CIDR (contd.)


201
10
0
0
1100 1001
0000 1010
0000 0000
0000 0000
201
10
7
255
1100 1001
0000 1010
0000 0111
1111 1111
2048
ADDRESSES
21
11
201.10.0.0 / 21 ≡
201.10.0.0 — 201.10.0.255
201.10.1.0 — 201.10.1.255
201.10.2.0 — 201.10.2.255
201.10.3.0 — 201.10.3.255
201.10.4.0 — 201.10.4.255
201.10.5.0 — 201.10.5.255
201.10.6.0 — 201.10.6.255
201.10.7.0 — 201.10.7.255
EIGHT
CLASS C
NETWORKS
= 256 × 8
= 2048
ADDRESSES
256 ADDRESSES = 1 CLASS C NETWORK

Dynamic host configuration

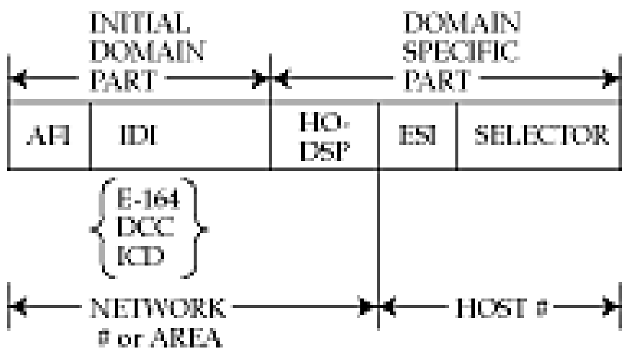
- Allows a set of hosts to share a pool of IP addresses
- Dynamic Host Configuration Protocol (DHCP)
- Newly booted computer broadcasts *discover* to subnet
- DHCP servers reply with *offers* of IP addresses
- Host picks one and broadcasts a *request* to a particular server
- All other servers withdraw offers, and selected server sends an *ack*
- When done, host sends a *release*
- IP address has a *lease* which limits time it is valid
- Server reuses IP addresses if their lease is over
- Similar technique used in *Point-to-point* protocol (PPP)

IPv6

- 32-bit address space is likely to eventually run out
- IPv6 extends size to 128 bits
- Main features
  - classless addresses
  - multiple levels of aggregation are possible
    - registry
    - provider
    - subscriber
    - subnet
  - several flavors of multicast
  - anycast
  - interoperability with IPv4

ATM network addressing

- Uses *Network Service Access Point (NSAP)* addresses
- Variable length (7-20 bytes)
- Several levels of hierarchy
  - national or international naming authority
  - addressing domain
  - subnet

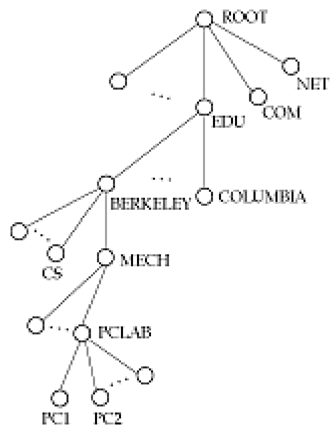


Name resolution

- Done by name servers
  - essentially look up a name and return an address
- Centralized design
  - consistent
  - single point of failure
  - concentrates load

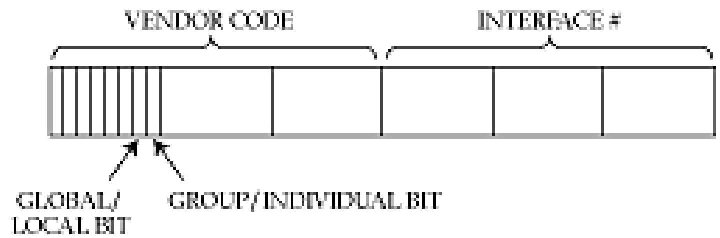
DNS

- Distributed name server
- A name server is responsible (an *authoritative server)* for a set of domains
- May delegate responsibility for part of a domain to a child
- Root servers are *replicated*
- If local server cannot answer a query, it asks root, which delegates reply
- Reply is *cached* and timed out

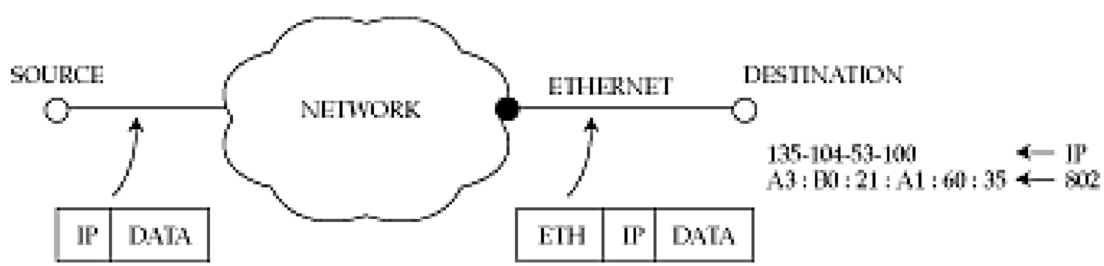


Finding datalink layer addresses

- Datalink layer address: most common format is IEEE 802



- Need to know datalink layer address typically for the last hop



ARP

- To get datalink layer address of a machine on the local subnet
- Broadcast a query with IP address onto local LAN
- Host that owns that address (or proxy) replies with address
- All hosts are required to listen for ARP requests and reply
  - including laser printers!
- Reply stored in an ARP cache and timed out
- In point-to-point LANs, need an ARP server
  - register translation with server
  - ask ARP server instead of broadcasting