IP addressing

- **IP address**: 32-bit identifier for host, router *interface*
- **Interface**: Connection between host, router and physical link
  - Routers typically have multiple interfaces
  - Host may have multiple interfaces
  - IP addresses associated with interface, not host, router

```
223.1.1.1 = 11011111 00000001 00000001 00000001
```
IP addressing (2)

- **IP address:**
  - Network part (high order bits)
  - Host part (low order bits)

- **What’s a network?**
  (from IP address perspective)
  - Device interfaces with same network part of IP address
  - Can physically reach each other without intervening router

Network consisting of 3 IP networks
(for IP addresses starting with 223, first 24 bits are network address)
IP addressing (3)

How to find the networks?

- Detach each interface from router, host
- Create “islands” of isolated networks

Interconnected system consisting of six networks
IP networks: Subnets

- Sub divide address space
  - network part
  - host address

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

```
11001000 00010111 00010000 00000000
```

200.23.16.0/24
Fixed subnetting (classful)

class

A  0  network  host  1.0.0.0 to 127.255.255.255
B  10 network  host  128.0.0.0 to 191.255.255.255
C  110 network  host  192.0.0.0 to 239.255.255.255
D  1110 multicast address  240.0.0.0 to 247.255.255.255

32 bits
Address management

- Problem: We are running out of networks
- Solution (a):
  **Subnetting**: E.g., Class B Host field (16 bits) is subdivided into <subnet;host> fields
- Solution (b):
  **CIDR** (Classless Inter Domain Routing)
CIDR

CIDR: Classless InterDomain Routing

- Motivation
  - Class A is too large, Class C is too small
  - Everyone had a Class B address!!!

- Solution:
  - Sites are given contiguous blocks of class-C addresses (256 addresses each) and a mask or parts of former class A/B networks.
CIDR (2.)

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

```
11001000  00010111
  0001000  0  00000000

200.23.16.0/23
```
IP addresses: How to get one?

Q: How does *host* get IP address?

- Hard-coded by system admin in a file
  - Wintel: Control Panel → Network → Configuration → TCP/IP → Properties
  - UNIX: /etc/rc.config

- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - “Plug-and-play”

- IP / Subnets allocated by provider (RIPE/ARIN/...
Hierarchical address structure

- Recall: CIDR

  128.119.48.12/18 = 10000000 01110111 00110000 00001100

  - High order bits form the **prefix**
  - Once inside the network, can **subnet**: divide remaining bits
  - Subnet example:

![Diagram of hierarchical address structure]

- Forwarding decision: Longest prefix match

Note: picture shows prefix masks, not interface addrs!
Forwarding vs. routing

- **Forwarding**: the process of moving packets from input to output
  - The forwarding table
  - Information in the packet

- **Routing**: process by which the forwarding table is built and maintained
  - One or more routing protocols
  - Procedures (algorithms) to convert routing info to forwarding table.

(More later ...)
Forwarding with CIDR

- Packet should be sent toward the interface with the **longest matching prefix**

Advertised address
Lookup: Longest prefix match

- Forwarding table:
  `<Network>/<mask> <next-hop>`

- IP Packets: destination IP address
  - Find next-hop via longest prefix match

- Example:

<table>
<thead>
<tr>
<th>Forwarding table</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>134.96.252.0/24</td>
<td>134.96.252.200</td>
</tr>
<tr>
<td>134.96.0.0/16</td>
<td>134.96.254.2</td>
</tr>
<tr>
<td>134.96.240.0/20</td>
<td>134.96.239.200</td>
</tr>
<tr>
<td>134.96.252.192/28</td>
<td>134.97.239.200</td>
</tr>
<tr>
<td>134.96.252.128/28</td>
<td>134.96.252.191</td>
</tr>
</tbody>
</table>
IP addressing: The last word ...

**Q:** How does an ISP get block of addresses?

**A:** ICANN: Internet Corporation for Assigned Names and Numbers
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

**Q:** What do I do if I don’t have a public address?

**A:** Private IP addresses (RFC 1918)
- 10/8
- 172.16/12
- 192.168/16

- Private use only – not routable in the Internet
NAT: Network address translation

Motivation: Local network uses just one IP address as far as outside world is concerned:
- Just one IP address for all devices
- Not needed range of addresses from ISP
NAT: Network address translation (2.)

All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)
NAT: Network address translation (3.)

Motivation: Local network uses just one IP address as far as outside world is concerned:

- Range of addresses not needed from ISP: just one IP address for all devices
- Can change addresses of devices in local network without notifying outside world
- Can change ISP without changing addresses of devices in local network
- Devices inside local net not explicitly addressable, visible by outside world (a security plus).
NAT: Network address translation (4.)

Implementation: NAT router must:

- **Outgoing datagrams: Replace** (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  
  ... remote clients/servers will respond using (NAT IP address, new port #) as destination addr.

- **Remember (in NAT translation table)** every (source IP address, port #) to (NAT IP address, new port #) translation pair

- **Incoming datagrams: Replace** (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table
NAT: Network address translation (5.)

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

NAT translation table

<table>
<thead>
<tr>
<th>WAN side addr</th>
<th>LAN side addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>138.76.29.7, 5001</td>
<td>10.0.0.1, 3345</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

2: host 10.0.0.1 sends datagram to 128.119.40.186, 80

3: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

3: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

3: Reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345
NAT: Network address translation (6.)

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!

- NAT is controversial:
  - Routers should only process up to layer 3
  - Violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - Address shortage should instead be solved by IPv6