IP addressing and forwarding

The Internet Network layer
Host, router network layer functions:

Transport layer: TCP, UDP

Routing protocols
- path selection
- RIP, OSPF, BGP

IP protocol
- addressing conventions
- datagram format
- packet handling conventions

ICMP protocol
- error reporting
- "router "signaling"

Transport layer: TCP, UDP

Link layer

Network layer

IP datagram format

- IP protocol version
- header length (bytes)
- "type" of data:
  - max number of datagrams
  - remaining hops (decremented at each router)
- upper layer protocol to deliver payload to

- how much overhead with TCP:
  - 20 bytes of TCP
  - 20 bytes of IP
  - = 40 bytes + app layer overhead

- 32 bits
- total datagram length (bytes) for fragmentation/reassembly
- 16-bit identifier
- type of service
- fragment offset
- time to live
- upper layer header
- checksum
- 32-bit source IP address
- 32-bit destination IP address
- Options (if any)

- data (variable length, typically a TCP or UDP segment)
- E.g. timestamp, record route taken, specify list of routers to visit

IP datagram format

<table>
<thead>
<tr>
<th>Field</th>
<th>Length (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1</td>
</tr>
<tr>
<td>Header length</td>
<td>1</td>
</tr>
<tr>
<td>Type of service</td>
<td>8</td>
</tr>
<tr>
<td>Datagram length</td>
<td>13</td>
</tr>
<tr>
<td>16-bit identifier</td>
<td>2</td>
</tr>
<tr>
<td>Flags</td>
<td>1</td>
</tr>
<tr>
<td>Time-to-live</td>
<td>2</td>
</tr>
<tr>
<td>Upper-layer</td>
<td>1</td>
</tr>
<tr>
<td>Protocol</td>
<td>1</td>
</tr>
<tr>
<td>Header checksum</td>
<td>1</td>
</tr>
<tr>
<td>32-bit Source IP</td>
<td>4</td>
</tr>
<tr>
<td>32-bit Destination IP</td>
<td>4</td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>
IP Fragmentation & Reassembly

- Network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- Large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - “reassembled” only at final destination
- IP header bits used to identify, order related fragments

Example

- 4000 byte datagram
- MTU = 1500 bytes
- 1480 bytes in data field

IP Addressing

- IP address: 32-bit identifier for host, router interface
- interface: connection between host/router and physical link
  - router’s typically have multiple interfaces
  - host typically has one interface
- IP addresses associated with each interface

Subnets

- IP address:
  - subnet part (high order bits)
  - host part (low order bits)
- What’s a subnet?
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router

One large datagram becomes several smaller datagrams
Subnetworks

Recipe

• To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.

Subnet mask: /24

How many subnets?

IP addressing: CIDR

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

How do you get an 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"
**How do you get an IP address?**

How does network get subnet part of IP addr?
gets allocated portion of its provider ISP’s address space

<table>
<thead>
<tr>
<th>ISP’s block</th>
<th>11001000 00010111 00010000 00000000</th>
<th>200.23.16.0/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization 0</td>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00010111 00010010 00000000</td>
<td>200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>11001000 00010111 00010100 00000000</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>Organization 7</td>
<td>11001000 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>

**Hierarchical addressing: route aggregation**

Hierarchical addressing allows efficient advertisement of routing information:

**Hierarchical addressing: more specific routes**

ISP’s-R-Us has a more specific route to Organization 1

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

ICANN: Internet Corporation for Assigned Names and Numbers
– allocates addresses
– manages DNS
– assigns domain names, resolves disputes
### NAT: Network Address Translation

- **rest of Internet**
- **local network (e.g., home network)**
- **10.0.0/24**

All datagrams leaving local network have some 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 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 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.
- **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 Example
NAT

- 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

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting:
    - unreachable host, network, port, protocol
  - echo request/rep (used by ping)
- network layer “above” IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code, plus first 8 bytes of IP datagram causing error

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply (ping)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>dest. network unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>dest host unreachable</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>dest protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>dest port unreachable</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>dest network unknown</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>dest host unknown</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>source quench (congestion control - not used)</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo request (ping)</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>route advertisement</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>router discovery</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL expired</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>bad IP header</td>
</tr>
</tbody>
</table>

Traceroute and ICMP

- Source sends series of UDP segments to dest
  - First has TTL=1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - Sends to source an ICMP message (type 11, code 0)
  - Message includes name of router & IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times
- Stopping criterion
- UDP segment eventually arrives at destination host
- Destination returns ICMP “port unreachable” packet
  (type 3, code 3)
- When source gets this ICMP, stops.