Problem: There is more than one network (heterogeneity & scale)

Internetworking:
- Internet Protocol (IP)
- Routing and scalability
- Group Communication

Internetworking

Hongwei Zhang

http://www.cs.wayne.edu/~hzhang

Every seeming equality conceals a hierarchy.

--- Mason Cooley

Acknowledgement: this lecture is partially based on the slides of Dr. Larry Peterson
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation (ARP)
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation (ARP)
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
IP Internet

- Concatenation of Networks

- Protocol Stack: H1 -> H8
Service Model

- Connectionless (datagram-based)
  - No need for connection setup and related control logic (e.g., assigning VC id and setting up forwarding table)

- Best-effort delivery (unreliable service)
  - packets can be lost
  - packets can be delivered out of order
  - duplicate copies of a packet can be delivered
  - packets can be delayed for a long time
IP datagram format

- HLen: header length in # of 32-bit words
- TOS: types of service to differentiate different application traffic
- Length: datagram length (including header) in # of bytes
- <Ident, Flags, Offset>: fragmentation & reassembly
- TTL: avoiding loops (# of hops)
- Protocol: demultiplexing key
- Checksum: calculated by regarding IP header as a sequence of 16-bit works
Fragmentation and Reassembly

- Each network has some maximum transmission unit (MTU)

- Design decisions
  - fragment when necessary (MTU < Datagram)
  - try to avoid fragmentation at source host
    - by choosing datagram size to be no more than MTU for the link associated with source
  - re-fragmentation is possible
    - when a downstream link has smaller MTU than an upstream link

- fragments are self-contained datagrams
- delay reassembly until destination host, for
  - simplicity: not knowing the right size to reassemble
  - efficiency: not knowing whether will be fragmented again

- do not recover from lost fragments; i.e., a whole IP datagram is discarded if a single fragment gets lost
  - thus fragmentation is a good thing to avoid, for instance, by performing “path MTU discovery”
Example

(a) Unfragmented packet

- Start of header
  - Ident = x
  - Offset = 0
- Rest of header
- 1400 data bytes

(b) Fragmented packets

- Start of header
  - Ident = x
  - Offset = 0
- Rest of header
- 512 data bytes

- Start of header
  - Ident = x
  - Offset = 64
- Rest of header
- 376 data bytes

ID of the original/unfragmented packet

“M (more)” bit in the IP “Flag” field

* 8 bytes
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Global Addresses

- Properties
  - globally unique
  - global support by different technologies
  - hierarchical: network + host

- Classful addressing & Dot Notation
  - Class A: 10.3.2.4
  - Class B: 128.96.33.81
  - Class C: 192.12.69.77

- Classless addressing (to be discussed)
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Datagram Forwarding

- **Strategy**
  - every datagram contains destination’s address
  
  - if directly connected to destination network (via a local physical network), then forward to host
  
  - if not directly connected, then forward to some router

  - forwarding table maps network number into next hop

  - each host has a *default router*
  
  - each router maintains a forwarding table
Example forwarding table: R2

<table>
<thead>
<tr>
<th>Network Number</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
</tr>
<tr>
<td>3</td>
<td>interface 1</td>
</tr>
<tr>
<td>4</td>
<td>interface 0</td>
</tr>
</tbody>
</table>

Hierarchical addressing improves scalability: routers maintain table for “networks” rather than “hosts”
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Address Translation

- Map IP addresses into physical/link-layer addresses
  - destination host
  - next hop router

- Techniques
  - encode physical address in host part of IP address
    - (-) constrains the length of physical address
  - table-based look up

- Address resolution protocol (ARP)
  - table of IP to physical address bindings
    - broadcast request if IP address not in table
    - target machine responds with its physical address
    - table entries are discarded if not refreshed (e.g., within 10 minutes)
# ARP Packet Format

<table>
<thead>
<tr>
<th>0</th>
<th>8</th>
<th>16</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware type = 1</td>
<td>ProtocolType = 0x0800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLen = 48</td>
<td>PLen = 32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceHardwareAddr (bytes 0-3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceHardwareAddr (bytes 4-5)</td>
<td>SourceProtocolAddr (bytes 0-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceProtocolAddr (bytes 2-3)</td>
<td>TargetHardwareAddr (bytes 0-1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetHardwareAddr (bytes 2-5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TargetProtocolAddr (bytes 0-3)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **HardwareType**: type of physical network (e.g., Ethernet)
- **ProtocolType**: type of higher layer protocol (e.g., IP)
- **HLEN & PLEN**: length of physical and protocol addresses
- **Operation**: request or response
- **Source/Target-Physical/Protocol addresses**
ARP Details

- ARP Request carries information about IP & physical addresses of the source

- A node receiving the ARP request
  - updates table with source when it is the target
  - refreshes table if it already has an entry for the source
  - otherwise, does not refresh/update table

Why?
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Dynamic host configuration protocol (DHCP)

- Automatically configure information such as IP address, gateway/default router, DNS, etc.
  - Reduces the overhead and probability of errors in manual configuration
  - Uses available IP addresses efficiently: not all nodes are up all the time

- Also support fixed binding of IP and MAC-address through manual configuration at DHCP server side

- System setup: clients ↔ DHCP server
DHCP: with relay agent

- Not want to maintain a DHCP server for every network
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Internet Control Message Protocol (ICMP)

- ICMP sits over IP

**Error reporting:** from routers to source hosts
  - Destination unreachable (e.g., due to link failure)
  - TTL exceeded (so datagrams don’t cycle forever)
  - Checksum failed
  - Reassembly failed

**Utilities**
  - Redirect (from router to source host): a router informs a host of better route (e.g., in helping a host setting the best “default router”)
  - “Echo (ping)” is implemented using ICMP packets
ping
Outline

- Best Effort Service Model
- Global Addressing Scheme
- Datagram forwarding
- Address translation
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual private networks and IP tunnels
Virtual private networks (VPN)

- To provide *controlled connectivity* among hosts for reasons such as security

- VPN via
  - Virtual circuits, LAN switching, etc (over a single network)
  - IP tunneling (over network of networks)
VPN via virtual circuits (frame relay, ATM, etc.)

Two separate private networks

Two VPNs sharing common switches
VPN via IP tunneling

Network 1

R1

Internetwork

10.0.0.1

R2

Network 2

<table>
<thead>
<tr>
<th>IP header, Destination = 2.x</th>
<th>IP header, Destination = 10.0.0.1</th>
<th>IP header, Destination = 2.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP payload</td>
<td>IP payload</td>
<td>IP payload</td>
</tr>
</tbody>
</table>
Benefits of IP tunneling

- Supplemented with encryption, a tunnel becomes a “private” link

- Incremental deployment of new network services (i.e., nodes in between two devices may not support the service)
  - E.g., MBone for network-layer multicast
  - Q: how are peer-to-peer networks different from “VPNs via IP tunneling”?

- Force a packet to be delivered to a particular place even if its original header might suggest otherwise
  - E.g., Mobile IP: packet redirection at the home host
Overhead of IP tunneling

- Increased header overhead, especially for short packets
- Management cost: configuring tunnels, etc
Summary of IP

- Deal with “heterogeneity”
  - Best-effort service: minimum assumption about underlying networks
  - A global, common address space
  - A common IP packet format

- Deal with “scale”
  - Hierarchical (network + host) address: reduces information maintained at routers  
    \textit{(scale in control state)}
  - Automatic configuration: DHCP, etc. \textit{(scale in management)}

- IP tunneling for VPN