Lecture 8: Internetworking

CSE 123: Computer Networks
Alex C. Snoeren

HW 2 due Thursday
Lecture 8 Overview

- Internet Protocol
  - Service model
  - Packet format

- Fragmentation

- Addressing
  - Subnetting
  - CIDR
Combing Networks

- Main challenge is heterogeneity of link layers:
  - Addressing
    » Each network media has a different addressing scheme
  - Bandwidth
    » Modems to terabits
  - Latency
    » Seconds to nanoseconds
  - Frame size
    » Dozens to thousands of bytes
  - Loss rates
    » Differ by many orders of magnitude
  - Service guarantees
    » Send and pray vs reserved bandwidth
Cerf & Kahn 74, 
“A Protocol for Packet Network Intercommunication”
  • Foundation for the modern Internet

- **Routers** forward **packets** from source to destination
  • May cross many separate networks along the way

- All packets use a common **Internet Protocol**
  • *Any* underlying data link protocol
  • *Any* higher layer transport protocol
TCP/IP Protocol Stack

- Application Layer
  - HTTP
- Transport Layer
  - TCP
- Network Layer
  - IP
  - Ethernet interface
  - SONET interface
- Link Layer

CSE 123 – Lecture 8: Internetworking
IP Networking

Router

Ethernet

FDDI

data packet

Eth  IP  TCP  HTTP

FDDI  IP  TCP  HTTP

data packet

CSE 123 – Lecture 8: Internetworking
Routers

- A router is a store-and-forward device
  - Routers are connected to multiple networks
  - On each network, looks just like another host
  - A lot like a switch, except at the network layer

- Must be explicitly addressed by incoming frames
  - Not at all like a switch, which is transparent
  - Removes link-layer header, parses IP header

- Looks up next hop, forwards on appropriate network
  - Each router need only get one step closer to destination
IP Philosophy

- Impose few demands on network
  - Make few assumptions about what network can do
  - No QoS, no reliability, no ordering, no large packets
  - No persistent state about communications; no connections

- Manage heterogeneity at hosts (not in network)
  - Adapt to underlying network heterogeneity
  - Re-order packets, detect errors, retransmit lost messages…
  - Persistent network state only kept in hosts (fate-sharing)

- Service model: best effort, a.k.a. send and pray
# IP Packet Header

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version (ver)</td>
<td>0-4</td>
</tr>
<tr>
<td>Header Length</td>
<td>16</td>
</tr>
<tr>
<td>Type of Service (TOS)</td>
<td>5-7</td>
</tr>
<tr>
<td>Identification</td>
<td>0-31</td>
</tr>
<tr>
<td>Time to Live (TTL)</td>
<td>8-15</td>
</tr>
<tr>
<td>Protocol</td>
<td>16</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>32</td>
</tr>
<tr>
<td>Source Address</td>
<td>48</td>
</tr>
<tr>
<td>Destination Address</td>
<td>48</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>32-64</td>
</tr>
<tr>
<td>Data (if any)</td>
<td>64</td>
</tr>
</tbody>
</table>

20 bytes total

**HL**
- R: Reserve (always 0)
- E: Extension flag (0 = no, 1 = yes)
- M: More flag (0 = end, 1 = continue)
- D: Don't fragment flag (0 = no, 1 = yes)
- F: Fragment flag (0 = no, 1 = yes)

**Offset Field**
- Set to 0 if flags are 0 (E, M, D, F)
- R = 0
- E = 0
- M = 0
- D = 0
- F = 0

For flags E, M, D, F:
- E = 0, M = 0, D = 0, F = 0
- E = 0, M = 1, D = 0, F = 0
- E = 0, M = 0, D = 1, F = 0
- E = 0, M = 0, D = 0, F = 1
- E = 0, M = 1, D = 1, F = 0
- E = 0, M = 0, D = 1, F = 1
- E = 0, M = 1, D = 0, F = 1
- E = 1, M = 0, D = 0, F = 0
- E = 1, M = 1, D = 0, F = 0
- E = 1, M = 0, D = 0, F = 1
- E = 1, M = 0, D = 1, F = 0
- E = 1, M = 0, D = 1, F = 1
- E = 1, M = 1, D = 0, F = 0
- E = 1, M = 1, D = 0, F = 1
- E = 1, M = 1, D = 1, F = 0
- E = 1, M = 1, D = 1, F = 1

**Reserved Bit (R)**
- Always set to 0

**Extension Flag (E)**
- Value can be set to 0 or 1

**More Fragment Flag (M)**
- Value can be set to 0 or 1

**Don’t Fragment Flag (D)**
- Value can be set to 0 or 1

**Fragmented Flags (F)**
- Value can be set to 0 or 1
Version field

- Which version of IP is this?
  - Plan for change
  - Very important!

- Current versions
  - 4: most of Internet today
  - 6: new protocol with larger addresses
  - What happened to 5? Standards body politics.
How big is IP header?
- In bytes/octetes
- Variable length
  » Options
- Engineering consequences of variable length...

Most IP packets are 20 bytes long
Type-of-Service

● How should this packet be treated?
  ◆ Care/don’t care for delay, throughput, reliability, cost
  ◆ How to interpret, how to apply on underlying net?
  ◆ Largely unused until 2000 (hijacked for new purposes, ECN & Diffserv)
Length

- How long is whole packet in bytes/octets?
  - Includes header
  - Limits total packet to 64K
  - Redundant?
TTL (Time-to-Live)

- How many more routers can this packet pass through?
  - Designed to limit packet from looping forever
- Each router decrements TTL field
- If TTL is 0 then router discards packet
Protocol

- Which transport protocol is the data using?
  - i.e. how should a host interpret the data

- TCP = 6
- UDP = 17
IP Checksum

- Header contains simple checksum
  - Validates content of header only

- Recalculated at each hop
  - Routers need to update TTL
  - Hence straightforward to modify

- Ensures *correct* destination receives packet
Different networks may have different frame limits (MTUs)
- Ethernet 1.5K, FDDI 4.5K

Router breaks up single IP packet into two or more smaller IP packets
- Each fragment is labeled so it can be correctly reassembled
- End host reassembles them into original packet
IP ID and Bitflags

- Source inserts unique value in identification field
  - Also known as the IPID
  - Value is copied into any fragments

- Offset field indicates position of current fragment (in bytes)
  - Zero for non-fragmented packet

- Bitflags provide additional information
  - More Fragments bit helps identify last fragment
  - Don’t Fragment bit prohibits (further) fragmentation
  - Note recursive fragmentation easily supported—just requires care with More Fragments bit
Fragmentation Example

One large datagram becomes several smaller datagrams

<table>
<thead>
<tr>
<th>length</th>
<th>ID</th>
<th>MF</th>
<th>offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>x</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1500</td>
<td>x</td>
<td>1</td>
<td>1480</td>
</tr>
<tr>
<td>1040</td>
<td>x</td>
<td>0</td>
<td>2960</td>
</tr>
</tbody>
</table>
Costs of Fragmentation

- Interplay between fragmentation and retransmission
  - A single lost fragment may trigger retransmission
  - Any retransmission will be of entire packet (why?)

- Packet must be completely reassembled before it can be consumed on the receiving host
  - Takes up buffer space in the mean time
  - When can it be garbage collected?

- Why not reassemble at each router?
Path MTU Discovery

- Path MTU is the smallest MTU along path
  - Packets less than this size don’t get fragmented

- Fragmentation is a burden for routers
  - We already avoid reassembling at routers
  - Avoid fragmentation too by having hosts learn path MTUs

- Hosts send packets, routers return error if too large
  - Hosts can set “don’t fragment” flag
  - Hosts discover limits, can size packets at source
  - Reassembly at destination as before
IP Addresses

- 32-bits in an IPv4 address
  - Dotted decimal format a.b.c.d
  - Each represent 8 bits of address

- Hierarchical: Network part and host part
  - E.g. IP address 128.54.70.238
  - 128.54 refers to the UCSD campus network
  - 70.238 refers to the host ieng6.ucsd.edu

- Which part is network vs. host?
Class-based Addressing

- Most significant bits determines “class” of address
  
  **Class A**
  
<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
  
  127 nets, 16M hosts

  **Class B**
  
<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td></td>
</tr>
</tbody>
</table>
  
  16K nets, 64K hosts

  **Class C**
  
<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0</td>
<td></td>
</tr>
</tbody>
</table>
  
  2M nets, 254 hosts

- Special addresses
  
  - Class D (1110) for multicast, Class E (1111) experimental
  - 127.0.0.1: local host (a.k.a. the loopback address)
  - Host bits all set to 0: network address
  - Host bits all set to 1: broadcast address

CSE 123 – Lecture 8: Internetworking
IP Forwarding Tables

- Router needs to know where to forward a packet
- Forwarding table contains:
  - List of network names and next hop routers
  - Local networks have entries specifying which interface
    » Link-local hosts can be delivered with Layer-2 forwarding

- E.g. www.ucsd.edu address is 132.239.180.101
  - Class B address – class + network is 132.239
  - Lookup 132.239 in forwarding table
  - Prefix – part of address that really matters for routing
Subnetting

- Individual networks may be composed of several LANs
  - Only want traffic destined to local hosts on physical network
  - Routers need a way to know which hosts on which LAN

- Networks can be arbitrarily decomposed into subnets
  - Each subnet is simply a prefix of the host address portion
  - Subnet prefix can be of any length, specified with netmask
Subnet Addresses

- Every (sub)network has an address and a netmask
  - Netmask tells which bits of the network address is important
  - Convention suggests it be a proper prefix

- Netmask written as an all-ones IP address
  - E.g., Class B netmask is 255.255.0.0
  - Sometimes expressed in terms of number of 1s, e.g., /16

- Need to size subnet appropriately for each LAN
  - Only have remaining bits to specify host addresses
IP Address Problem (1991)

- Address space depletion
  - In danger of running out of classes A and B

- Why?
  - Class C too small for most organizations (only ~250 addresses)
  - Very few class A – very careful about giving them out (who has 16M hosts anyway?)
  - Class B – greatest problem
CIDR

- **Classless Inter-Domain Routing (1993)**
  - Networks described by variable-length prefix and length
  - Allows arbitrary allocation between network and host address

<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prefix</strong></td>
<td>Mask=# significant bits representing prefix</td>
</tr>
</tbody>
</table>

- e.g. 10.95.1.2/8: 10 is network and remainder (95.1.2) is host

- **Pro:** Finer grained allocation; aggregation
- **Con:** More expensive lookup: *longest prefix match*
Longest Matching Prefix

- Forwarding table contains many prefix/length tuples
  - They *need not* be disjoint!
  - E.g. 200.23.16.0/20 and 200.23.18.0/23
  - What to do if a packet arrives for destination 200.23.18.1?
  - Need to find the longest prefix in the table which matches it (200.23.18.0/23)

- Not a simple table, requires multiple memory lookups
  - Lots and lots of research done on this problem
  - Our own George Varghese is the master of this domain
• **Straightforward way to look up LMP**
  - Arrange route entries into a series of bit tests
  - Worst case = 32 bit tests
  - Problem: memory speed is a bottleneck

![PATRICIA Trie Diagram]

- Bit to test – 0 = left child, 1 = right child
- Default 0/0
- 128.2/16
- 128.32/16
- 128.32.130/24
- 128.32.150/24
For Next Time

- Read 4.1.5-6, 9.1.3 in P&D

- Homework due next time at beginning of class