Outline

- Internet Protocol
  - Service Model
  - Fragmentation
  - Addressing
    - Original addressing scheme
    - Subnetting
    - CIDR
  - Forwarding
  - ICMP
  - ARP
  - Address Shortage
    - NAT
    - IPv6
• Concatenation of Networks

• Protocol Stack
Service Model

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
- Datagram format
Problem: A router may receive a packet larger than the maximum transmission unit (MTU) of the outgoing link.

Solution: R1 fragments the IP datagram into multiple, self-contained datagrams.
Fragmentation (II)

• Fragments are re-assembled by the destination host; not by intermediate routers.
• To avoid fragmentation, hosts commonly use path MTU discovery to find the smallest MTU along the path.
• Path MTU discovery involves sending various size datagrams until they do not require fragmentation along the path.
• Most links use MTU≥1500 bytes today.
• Try: traceroute –f www.mit.edu 1500 and traceroute –f www.mit.edu 1501
• (DF=1 set in IP header; routers send “ICMP” error message, which is shown as “!F”).
Global Addresses

- **Properties**
  - globally unique
  - hierarchical: network + host

- **Dot Notation**
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77
Datagram Forwarding

- **Strategy**
  - every datagram contains destination’s address
  - if connected to destination 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 (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>
IP Addressing

• Problem:
  - Address classes were too “rigid”. For most organizations, Class C were too small and Class B too big. Led to very inefficient use of address space, and a shortage of addresses.
  - Organizations with internal routers needed to have a separate (Class C) network ID for each link.
  - And then every other router in the Internet had to know about every network ID in every organization, which led to large address tables.
  - Small organizations wanted Class B in case they grew to more than 255 hosts. But there were only about 16,000 Class B network IDs.
IP Addressing

- Two solutions were introduced:
  - **Subnetting** is used within an organization to subdivide the organization’s network ID.
  - **Classless Interdomain Routing (CIDR)** was introduced in 1993 to provide more efficient and flexible use of IP address space across the whole Internet.
  - CIDR is also known as “supernetting” because subnetting and CIDR are basically the same idea.
Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part
- Subnets visible only within site

<table>
<thead>
<tr>
<th>Network number</th>
<th>Host number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class B address

<table>
<thead>
<tr>
<th>Network number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
</table>

Subnet mask (255.255.255.0)

<table>
<thead>
<tr>
<th>Network number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
</table>

Subnetted address
Subnet Example

Forwarding table at router R1

<table>
<thead>
<tr>
<th>Subnet Number</th>
<th>Subnet Mask</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.96.34.0</td>
<td>255.255.255.128</td>
<td>interface 0</td>
</tr>
<tr>
<td>128.96.34.128</td>
<td>255.255.255.128</td>
<td>interface 1</td>
</tr>
<tr>
<td>128.96.33.0</td>
<td>255.255.255.0</td>
<td>R2</td>
</tr>
</tbody>
</table>
Forwarding Algorithm

D = destination IP address
for each entry (SubnetNum, SubnetMask, NextHop)
    D1 = SubnetMask & D
    if D1 = SubnetNum
        if NextHop is an interface
            deliver datagram directly to D
        else
            deliver datagram to NextHop
    else

• Use a default router if nothing matches
• Not necessary for all 1s in subnet mask to be contiguous
• Can put multiple subnets on one physical network
• Subnets not visible from the rest of the Internet
The IP address space is broken into line segments. Each line segment is described by a *prefix*. A prefix is of the form *x/y* where *x* indicates the prefix of all addresses in the line segment, and *y* indicates the length of the segment. e.g. The prefix 128.9/16 represents the line segment containing addresses in the range: 128.9.0.0 … 128.9.255.255.
Classless Interdomain Routing (CIDR)

Addressing

Most specific route = “longest matching prefix”
Prefix aggregation:

- If a service provider serves two organizations with prefixes, it can (sometimes) aggregate them to form a larger prefix. Other routers can refer to this larger prefix, and so reduce the size of their address table.
- E.g. ISP serves 128.9.14.0/24 and 128.9.15.0/24, it can tell other routers to send it all packets belonging to the prefix 128.9.14.0/23.

ISP Choice:

- In principle, an organization can keep its prefix if it changes service providers.
Hierarchical addressing allows efficient advertisement of routing information:

```
Organization 0
200.23.16.0/23

Organization 1
200.23.18.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23
```

```
Fly-By-Night-ISP
“Send me anything with addresses beginning 200.23.16.0/20”
```

```
ISPs-R-Us
“Send me anything with addresses beginning 199.31.0.0/16”
```

```
Internet
```

CS 349/Fall05
Hierarchical addressing: route aggregation

- Organization 0
  - 200.23.16.0/23
- Organization 1
  - 200.23.18.0/23
- Organization 2
  - 200.23.20.0/23
- Organization 7
  - 200.23.30.0/23

Multi-homing

Fly-By-Night-ISP

“Send me anything with addresses beginning 200.23.16.0/20”

ISPs-R-Us

“Send me anything with addresses beginning 199.31.0.0/16, or 200.23.30.0/23”

Internet
Size of the Routing Table at the core of the Internet

Source: http://bgp.potaroo.net/
Prefix Length Distribution

Source: Geoff Huston, Oct 2001
How a Router Forwards Datagrams

128.17.20.1

R2

128.17.16.1

R3

R1

R4

128.17.16.1

Prefix Port

3 2 2 7 2 1 3

128.17.16.1

Next-hop

128.17.16.1

128.17.14.1

128.17.14.1

128.17.10.1

128.17.14.1

128.17.20.1

128.17.16.1

e.g. 128.9.16.14 => Port 2

Forwarding table

Prefix | Next-hop | Port
---|---|---
65/8 | 128.17.16.1 | 3
128.9/16 | 128.17.14.1 | 2
128.9.16/20 | 128.17.14.1 | 2
128.9.19/24 | 128.17.10.1 | 7
128.9.25/24 | 128.17.14.1 | 2
128.9.176/20 | 128.17.20.1 | 1
142.12/19 | 128.17.16.1 | 3
Forwarding in an IP Router

- Lookup packet DA in forwarding table.
  - If known, forward to correct port.
  - If unknown, drop packet.
- Decrement TTL, update header Checksum.
- Forward packet to outgoing interface.
- Transmit packet onto link.

**Question:** How is the address looked up in a real router?
Making a Forwarding Decision

*Class-based addressing*

### IP Address Space

<table>
<thead>
<tr>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
<th>D</th>
</tr>
</thead>
</table>

### Routing Table:

- Exact Match: There are many well-known ways to find an exact match in a table.

212.17.9.4

212.17.9.0
## Lookup Performance Required

<table>
<thead>
<tr>
<th>Line</th>
<th>Line Rate</th>
<th>Pkt-size=40B</th>
<th>Pkt-size=240B</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1.5Mbps</td>
<td>4.68 Kpps</td>
<td>0.78 Kpps</td>
</tr>
<tr>
<td>OC3</td>
<td>155Mbps</td>
<td>480 Kpps</td>
<td>80 Kpps</td>
</tr>
<tr>
<td>OC12</td>
<td>622Mbps</td>
<td>1.94 Mpps</td>
<td>323 Kpps</td>
</tr>
<tr>
<td>OC48</td>
<td>2.5Gbps</td>
<td>7.81 Mpps</td>
<td>1.3 Mpps</td>
</tr>
<tr>
<td>OC192</td>
<td>10 Gbps</td>
<td>31.25 Mpps</td>
<td>5.21 Mpps</td>
</tr>
</tbody>
</table>
Direct Lookup

Problem: With $2^{32}$ addresses, the memory would require 4 billion entries.
Associative Lookups
“Contents addressable memory” (CAM)

Advantages:
- Simple

Disadvantages
- Slow
- High Power
- Small
- Expensive
Hashed Lookups

Hashing Function \(\log_2 N\) Address

Memory Data

Search Data 32

16

Associated Data

Hit? Address \(\log_2 N\)
Lookups Using Hashing

An example

Hashing Function

Memory

Linked list of entries with same hash key.

Search Data

32

Hashing Function

16

Associated Data

Hit?
Lookups Using Hashing

Advantages:

• Simple

• Expected lookup time can be small

Disadvantages

• Non-deterministic lookup time

• Inefficient use of memory
Patricia Tries

Example Prefixes:

a) 00001
b) 00010
c) 00011
d) 001
e) 0101
f) 011
g) 100
h) 1010
i) 1100
j) 11110000
How to Implement a router

• Issues
  – Performance
  • Throughput
  – Scaling
Workstation-Based

- **Aggregate bandwidth**
  - 1/2 of the I/O bus bandwidth
  - capacity shared among all hosts connected to router
  - example: 1Gbps bus can support 5 x 100Mbps ports (in theory)

- **Packets-per-second**
  - must be able to switch small packets
  - 300,000 packets-per-second is achievable
  - e.g., 64-byte packets implies 155Mbps
Switching Hardware

- **Design Goals**
  - throughput (depends on traffic model)
  - scalability (a function of $n$)

- **Ports**
  - buffering (input and/or output)

- **Fabric**
  - as simple as possible
  - sometimes do buffering (internal)
LAN Addresses and ARP

32-bit IP address:
• *network-layer* address
• used to get datagram to destination IP network (recall IP network definition)

LAN (or MAC or physical or Ethernet) address:
• used to get datagram from one interface to another physically-connected interface (same network)
• 48 bit MAC address (for most LANs)
  “burned” in the adapter EPROM
LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  (a) MAC address: like Social Security Number
  (b) IP address: like postal address
- MAC flat address => portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP network to which node is attached
ARP: Address Resolution Protocol

Each IP node (Host, Router) on LAN has ARP table.

ARP Table: IP/MAC address mappings for some LAN nodes < IP address; MAC address; TTL>

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Question: how to determine MAC address of B knowing B’s IP address?
ARP protocol

- A wants to send datagram to B, and A knows B’s IP address.
- Suppose B’s MAC address is not in A’s ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A’s MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:  
  - nodes create their ARP tables without intervention from net administrator
ARP Protocol (II)

- Proxy ARP
  - Reply on behalf of another node

- ARP Highjacking
  - Node can “steal” packets destined to another node
  - Solutions
    - Static ARP entries
    - Switched LANs
    - Lock MAC addresses to Switch ports
• **Internet Control Message Protocol:**
  - Used by a router/end-host to report some types of error:
  - E.g. Destination Unreachable: packet can’t be forwarded to/towards its destination.
  - E.g. Time Exceeded: TTL reached zero, or fragment didn’t arrive in time. Traceroute uses this error to its advantage.
  - An ICMP message is an IP datagram, and is sent back to the source of the packet that caused the error.
IP Address Shortage

- Global IPv4 addresses are getting depleted
  - Increase in number of hosts
    - PCs, PDAs, cellphones, microwaves, etc
  - Address inefficiencies

- What to do
  - Get larger address space -> IPv6
  - Remove the assumption that address is globally unique
    - Can reuse the same address multiple times -> NAT
**NAT: Network Address Translation**

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

1: host 10.0.0.1 sends datagram to 128.119.40, 80

2: 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
NAT 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 Problems

• Problems due to NAT
  - Increased network complexity, reduced robustness
  - Cannot run services inside NAT (maybe)

• Address shortage should instead be solved by IPv6
IPv6

- Motivation: 32-bit address space exhaustion
- Take the opportunity for some clean-up
- IPv6 datagram format:
  - fixed-length 40 byte header
  - Address length changed from 32 bits to 128 bits
  - fragmentation fields moved out of base header
  - IP options moved out of base header
    - Header Length field eliminated
  - Header Checksum eliminated
  - Type of Service field eliminated
  - Time to Live → Hop Limit, Protocol → Next Header
  - Precedence → Priority, added Flow Label field
  - Length field excludes IPv6 header
### IPv6 header format

<table>
<thead>
<tr>
<th>Version</th>
<th>Priority</th>
<th>Flow Label</th>
<th>Payload Length</th>
<th>Next Header</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Address (16 bytes, 128 bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Address (16 bytes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
- Two proposed approaches to allow the Internet operate with mixed IPv4 and IPv6 routers:
  - **Dual Stack**: some routers with dual stack (v6, v4) can "translate" between formats
  - **Tunneling**: IPv6 carried as payload in IPv4 packets among IPv4 routers