Outline

• Internet Protocol
  – Service Model
  – Addressing
    • Original addressing scheme
    • Subnetting
    • CIDR
  – Fragmentation
  – ICMP
  – Address Shortage
    • NAT
    • IPv6
IP Internet

- Concatenation of Networks
- Three requirements
  - Connectivity
  - Addressing
  - Path discovery
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
IP Header
Global Addresses

- Properties
  - globally unique
  - hierarchical
    - network + host
  - Address classes

- Dot Notation
  - 10.3.2.4
  - 128.96.33.81
Address Allocation

• Who manages the IP address space?
  – Hierarchical system
    • Internet Assigned Numbers Authority (IANA)
    • Local Internet Registries (LIRs)
      – ARIN, APNIC, etc
    • ISPs
Datagram Forwarding (v.1)

- **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

- **Example (R2)**

<table>
<thead>
<tr>
<th>Net 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”.
  – 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)**
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>
</table>

**Class B address**

```
11111111111111111111111111111111 1000000000000000
```

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 (v.2)

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
        • Subnets not visible from the rest of the Internet
Classless Interdomain Routing (CIDR)

- 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)

Most specific route = “longest matching prefix”
Classless Interdomain Routing (CIDR)

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: route aggregation

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
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

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

Multi-homing
How a Router Forwards Datagrams

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next-hop</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>65/8</td>
<td>128.17.16.1</td>
<td>3</td>
</tr>
<tr>
<td>128.9/16</td>
<td>128.17.14.1</td>
<td>2</td>
</tr>
<tr>
<td>128.9.16/20</td>
<td>128.17.14.1</td>
<td>2</td>
</tr>
<tr>
<td>128.9.19/24</td>
<td>128.17.10.1</td>
<td>7</td>
</tr>
<tr>
<td>128.9.25/24</td>
<td>128.17.14.1</td>
<td>2</td>
</tr>
<tr>
<td>128.9.176/20</td>
<td>128.17.20.1</td>
<td>1</td>
</tr>
<tr>
<td>142.12/19</td>
<td>128.17.16.1</td>
<td>3</td>
</tr>
</tbody>
</table>

Forwarding table

e.g. 128.9.16.14 => Port 2
Forwarding in an IP Router (v.3)

- 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?
### Fragmentation

**Problem:** A router may receive a packet larger than the maximum transmission unit (MTU) of the outgoing link.

**Source**

![Source Ethernet](A)

MTU=1500 bytes

**Destination**

![Destination Ethernet](B)

MTU=1500 bytes

**MTU<1500 bytes**

**Solution:** R1 fragments the IP datagram into multiple, self-contained datagrams.

Offset>0
More Frag=0

Data | HDR (ID=x)

Offset=0
More Frag=1

Data | HDR (ID=x)
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 $\geq 1500$ bytes today.
- (DF=1 set in IP header; routers send “ICMP” error message, which is shown as “!F”).

ICMP

- 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

<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>

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>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Length</td>
<td>Next Header</td>
<td>Hop Limit</td>
</tr>
<tr>
<td>Source Address (16 bytes, 128 bits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Address (16 bytes)</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 to 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.

![Diagram](attachment:image.png)