CS 43: Computer Networks

21: The Network Layer & IP
November 7, 2018
The Network Layer!

Application: the application (e.g., the Web, Email)

Transport: end-to-end connections, reliability

Network: routing

Link (data-link): framing, error detection

Physical: 1’s and 0’s/bits across a medium (copper, the air, fiber)
On the Internet, best-effort packet switching is the norm.

**Forwarding:** move packets from router’s input to appropriate router output: Look up in a table.

**Routing:** determine route taken by packets from source to destination: Populating the table.

Hardware helps with quick forwarding using longest prefix matching.
Today

• IP header format

• Subnets and IP addressing
  – CIDR
  – Route aggregation

• DHCP: Assigning an IP address to an interface

• Fragmentation
**IP datagram format**

- **IP protocol version number**
- **header length** (in 32-bit words)
- “**type**” of data
- **max number remaining hops** (decremented at each router)
- **upper layer protocol** to deliver payload to
- **total datagram length** (bytes)
  - for fragmentation/reassembly
- **data** (variable length, typically a TCP or UDP segment)
- **options (if any)**

### How much overhead?
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead
### IP datagram format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ver</code></td>
<td>Version, 4 bits</td>
</tr>
<tr>
<td><code>len</code></td>
<td>Length, 16 bits</td>
</tr>
<tr>
<td><code>type of service</code></td>
<td>Type of service, 8 bits</td>
</tr>
<tr>
<td><code>length</code></td>
<td>Length of the datagram, 6 bits</td>
</tr>
<tr>
<td><code>16-bit identifier</code></td>
<td>Identifier for fragmentation, 16 bits</td>
</tr>
<tr>
<td><code>flgs</code></td>
<td>Flags, 4 bits</td>
</tr>
<tr>
<td><code>fragment offset</code></td>
<td>Fragment offset, 13 bits</td>
</tr>
<tr>
<td><code>time to live</code></td>
<td>Time to live, 16 bits</td>
</tr>
<tr>
<td><code>upper layer</code></td>
<td>Upper layer, 8 bits</td>
</tr>
<tr>
<td><code>header checksum</code></td>
<td>Header checksum, 16 bits</td>
</tr>
<tr>
<td><code>32 bit source IP address</code></td>
<td>Source endpoint.</td>
</tr>
<tr>
<td><code>32 bit destination IP address</code></td>
<td>Final destination endpoint.</td>
</tr>
<tr>
<td><code>options (if any)</code></td>
<td>Options if any are present</td>
</tr>
<tr>
<td><code>data</code></td>
<td>Data, variable length, typically a TCP or UDP segment</td>
</tr>
</tbody>
</table>

Addresses must be unique on the network!
IP Address (IPv4)

- A unique 32-bit unsigned integer value
- Identifies an interface (on a host, on a router, ...)
- Represented in dotted-quad/octet notation

```
12  34  158  5
```

```
00001100 00100010 10011110 00000101
```
IP Addresses

• $2^{32} \Rightarrow 4,294,967,296$ possible addresses.

• In the early 80’s, that’s a lot!
  – Population was $\sim$4.5 billion.

• Now...not so much.
  – Population > 7 billion.
Network Interfaces

- **IP address**: 32-bit identifier for host, router
  - *interface*
    - connection between host/router and physical link
      - router’s typically have multiple interfaces
      - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- **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
  § On the same link layer

network consisting of 3 subnets

Lecture 21 - Slide 10
Assigning Addresses

- **IANA** – Internet Assigned Numbers Authority
  - (Run by Jon Postel until 1988)
  - Now a part of ICANN
  - ARIN: North America
  - RIPE: Europe

- **ICANN**: Internet Corporation for Assigned Names and Numbers
  - Manages IP addresses, DNS, resolves disputes
Who gets an address? How many?

• Back in the old days, you called up Jon Postel
  – “How many addresses do you need?”
  – “Here you go! I may have rounded a bit.”
Who gets an address? How many?

- Classful Addressing
  - Class A: 8-bit prefix, 24 bits for hosts (16,777,216)
  - Class B: 16-bit prefix, 16 bits for hosts (65,536)
  - Class C: 24-bit prefix, 8 bits for hosts (256)
CIDR

- Classless Interdomain Routing
  - Prefix (subnet) length is no longer fixed
  - (Can be division of bits rather than just 8/24, 16/16, and 24/8)
Why do we give out addresses in CIDR blocks? How many of these statements are true? (Which ones?)

- It requires fewer resources at routers.
- It requires fewer resources at end hosts.
- It reduces the number of block allocations that need to be managed.
- It better utilizes the IP address space.

A – 0, B – 1, C – 2, D – 3, E – 4
Why do we give out addresses in CIDR blocks? How many of these statements are true? (Which ones?)

• It requires fewer resources at routers (F): requires more resources!

• It requires fewer resources at end hosts (F): negligible increased state

• It reduces the number of block allocations that need to be managed (F): increases the number of block allocations

• It better utilizes the IP address space (T)

A – 0, B – 1, C – 2, D – 3, E – 4
CIDR

• Classless Interdomain Routing
  – Prefix (subnet) length is no longer fixed
  – Address blocks come with a subnet mask

• Subnet mask written in two ways:
  – Dotted decimal: 255.255.240.0
  – /20
  – Both mean:
    \[ \begin{array}{cccccc}
    1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 1 & 1 & 1 \\
    1 & 1 & 1 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 0 \\
    \end{array} \]
  – /20
CIDR

• Addresses divided into two pieces:
  – Prefix portion (network address)
  – Host portion

• Given an IP address and mask, we can determine:
  – The prefix (network address) by ANDing
  – The broadcast address by ORing inverted mask
Network Address (Subnet Address)

- E.g., 230.8.1.3/18 /18 => mask is 255.255.192.0

```
11100110 00001000 00000001 00000011
```

IP address

```
11111111 11111111 11000000 00000000
```

/18 Subnet mask
Network Address (Subnet Address)

• E.g., 230.8.1.3/18  /18 => mask is 255.255.192.0

Network address advertised by router: 230.8.0.0
Why might a device care about its “Network Address”?

- Answers the question: is the destination on the same subnet as me?

- Address + subnet mask -> Network address

- If destination is on same network:
  - Send directly to them

- Else:
  - Send to gateway router
Broadcast Address

- E.g., 230.8.1.3/18

<table>
<thead>
<tr>
<th>IP address</th>
<th>/18 Subnet mask</th>
<th>complement of the subnet mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>11100110 00001000 00000001 00000011</td>
<td>11111111 11111111 11000000 00000000</td>
<td>00000000 00000000 00111111 11111111</td>
</tr>
</tbody>
</table>

Lecture 21 - Slide 22
## Broadcast Address

- E.g., 230.8.1.3/18

<table>
<thead>
<tr>
<th>IP address complement of the subnet mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>11100111 00001000 00000001 00000011</td>
</tr>
<tr>
<td>00000000 00000000 00111111 11111111</td>
</tr>
</tbody>
</table>
Broadcast Address

• E.g., 230.8.1.3/18

Broadcast address: 230.8.63.255
Hierarchical addressing allows efficient advertisement of routing information:

Send me anything with addresses beginning 200.23.16.0/23

Send me anything with addresses beginning 199.31.0.0/16
"Send me anything with addresses beginning 200.23.16.0/20" translates to the following:

```
200.23.16.0/20 = 11001000 00010111 00010000 00000000  /20 Prefix
200.23.16.0 = 11001000 00010111 00010000 00000000 | Range
200.23.31.255 = 11001000 00010111 00011111 11111111
```

/20 prefix contains the range of IP addresses that match the first 20 bits, and can have any value for the remaining 12 bits in the range of:

[first 20 bits] 0000 00000000
[first 20 bits] 1111 11111111

A total of $2^{12} = 4,096$ IP addresses
What should we do if organization 1 decides to switch to ISPs-R-Us?

Send me anything with addresses beginning 200.23.16.0/20

Send me anything with addresses beginning 199.31.0.0/16
Route aggregation in Fly-By-Night ISP

Fly-By-Night-ISP

200.23.16.0/20 = 11001000 00010111 00010000 00000000

Individual Organizations: All of these organizations IP addresses lie within Fly-by-Night’s /20 prefix (first 20 bits are the same)

• they more specifically match on the three more bits to form a /23 prefix (first 23 bits of all IP addresses within their organization are the same).

• The last 9 (32-23) bits provide $2^9 = 512$ unique IP addresses within each organization.

    /23 prefixes

200.23.16.0/23 = 11001000 00010111 00010000 00000000

200.23.18.0/23 = 11001000 00010111 00010010 00000000

200.23.20.0/23 = 11001000 00010111 00010100 00000000

200.23.30.0/23 = 11001000 00010111 00011110 00000000
What should we do if organization 1 decides to switch to ISPs-R-Us?

A. Move 200.23.18.0/23 to ISPs-R-Us (and break up Fly-By-Night’s /20 block).
B. Give new addresses to Organization 1 (and force them to change all their addresses).
C. Some other solution.
Hierarchical addressing: More Specific Routes

ISPs-R-Us has a more specific route to Organization 1

Send me anything with addresses beginning 200.23.16.0/20 or 200.23.18.0/23

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

Fly-By-Night-ISP

ISPs-R-Us

Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23

Internet
Hierarchical addressing: More Specific Routes

ISPs-R-Us has a more specific route to Organization 1

Send me anything with addresses beginning 200.23.16.0/20 or 200.23.18.0/23

Longest prefix matching!

Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23

Organization 0
200.23.16.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23

Organization 1
200.23.18.0/23

Fly-By-Night-ISP

ISPs-R-Us

Router 1

Internet
Longest Prefix Matching at Router 1

Now, when an incoming packet addressed with destination address 200.23.18.5 arrives – this address belongs to Organization 1 and the packet will be matched using longest prefix matching and will be routed to ISPs-R-Us rather than the Fly-by-Night ISP.
How does an end host get an IP address?

• **Static IP: hard-coded**
  – Windows: 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”
DHCP: Dynamic Host Configuration Protocol

**Goal:** allow host to *dynamically* obtain its IP address from network server when it joins network
- can renew its lease on address in use
- allows reuse of addresses
- support for mobile users who want to join network

**DHCP overview:**
- host broadcasts “DHCP discover” msg [optional]
- DHCP server responds with “DHCP offer” msg [optional]
- host requests IP address: “DHCP request” msg
- DHCP server sends address: “DHCP ack” msg
DHCP client-server scenario

DHCP server: 223.1.2.5

DHCP discover
src: 0.0.0.0, 68
dest: 255.255.255.255, 67
yiaddr: 0.0.0.0
transaction ID: 654

DHCP offer
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 654
lifetime: 3600 secs

DHCP request
src: 0.0.0.0, 68
dest: 255.255.255.255, 67
yiaddr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

DHCP ACK
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs
DHCP: More than IP Addresses

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client (default GW)
- name and IP address of DNS server(s)
- subnet mask
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

fragmentation: in: one large datagram out: 3 smaller datagrams

reassembly
IP datagram format

- **Version (ver)**: 4 bits indicating the version of the IP protocol.
- **Header length (head. len)**: 4 bits indicating the length of the header in 32-bit words.
- **Type of service (type of service)**: 8 bits indicating the type of service for networking applications.
- **Length (length)**: 16 bits indicating the total length of the datagram in bytes.
- **16-bit identifier** (4 bits): A unique identifier for fragmentation and reassembly.
- **Flags (flgs)**: 3 bits indicating the flags for fragmentation and reassembly.
- **Fragment offset (fragment offset)**: 13 bits indicating the offset of the fragment within the original datagram.
- **Time to live (time to live)**: 8 bits indicating the number of hops the datagram can take before it is discarded.
- **Upper layer (upper layer)**: 8 bits indicating the upper layer protocol.
- **Header checksum (header checksum)**: 16 bits indicating the checksum of the header.
- **32-bit source IP address**:
- **32-bit destination IP address**:
- **Options (if any)**:
- **Data (variable length, typically a TCP or UDP segment)**: Variable length data, typically a TCP or UDP segment.
Example:

- 4000 byte datagram
- MTU = 1500 bytes

One large datagram becomes several smaller datagrams

Example:

- 4000 byte datagram
- MTU = 1500 bytes

1480 bytes in data field

Offset = 1480/8

One large datagram becomes several smaller datagrams
How can we use this for evil?

A. Send fragments that overlap.

B. Send many tiny fragments, none of which have offset 0.

C. Send fragments that, when assembled, are bigger than the maximum IP datagram.

D. More than one of the above.

E. Nah, networks (and operating systems) are too robust for this to cause problems.
IP Fragmentation Attacks...

IP fragmentation exploits

IP fragment overlapped

The IP fragment overlapped exploit occurs when two fragments contained within the same IP datagram have offsets that indicate that they overlap each other in positioning within the datagram. This could mean that either fragment A is being completely overwritten by fragment B, or that fragment A is partially being overwritten by fragment B. Some operating systems do not properly handle fragments that overlap in this manner and may throw exceptions or behave in other undesirable ways upon receipt of overlapping fragments. This is the basis for the teardrop Denial of Service attacks.

IP fragmentation buffer full

The IP fragmentation buffer full exploit occurs when there is an excessive amount of incomplete fragmented traffic detected on the protected network. This could be due to an excessive number of incomplete fragmented datagrams, a large number of fragments for individual datagrams or a combination of quantity of incomplete datagrams and size/number of fragments in each datagram. This type of traffic is most likely an attempt to bypass security measures or Intrusion Detection Systems by intentional fragmentation of attack activity.

IP fragment overrun

The IP Fragment Overrun exploit is when a reassembled fragmented datagram exceeds the declared IP data length or the maximum datagram length. By definition, no IP datagram should be larger than 65,535 bytes. Systems that try to process these large datagrams can crash, and can be indicative of a denial of service attempt.

IP fragment overwrite

Overlapping fragments may be used in an attempt to bypass Intrusion Detection Systems. In this exploit, part of an attack is sent in fragments along with additional random data; future fragments may overwrite the random data with the remainder of the attack. If the completed datagram is not properly reassembled at the IDS, the attack will go undetected.

IP fragment too many datagrams

The Too Many Datagrams exploit is identified by an excessive number of incomplete fragmented datagrams detected on the network. This is usually either a denial of service attack or an attempt to bypass security measures. An example of "Too Many Datagrams", "Incomplete Datagram" and "Fragment Too Small" is the Rose Attack.

IP fragment incomplete datagram

This exploit occurs when a datagram cannot be fully reassembled due to missing data. This can indicate a denial of service attack or an attempt to defeat packet filter security policies.

IP fragment too small

An IP Fragment Too Small exploit is when any fragment other than the final fragment is less than 400 bytes, indicating that the fragment is likely intentionally crafted. Small fragments may be used in denial of service attacks or in an attempt to bypass security measures or detection.
Summary

• $2^{32}$ addresses is not that many...

• CIDR helps give out finer granularity
  – Divide bits among network and host
  – Longest prefix matching allows blocks to be divided

• IP supports fragmentation – usually bad news
  – These days, most links have common MTU