



## Designing IP

### EE 122: Intro to Communication Networks

Fall 2007 (WF 4-5:30 in Cory 277)

Vern Paxson

TAs: Lisa Fowler, Daniel Killebrew & Jorge Ortiz

<http://inst.eecs.berkeley.edu/~ee122/>

Materials with thanks to Jennifer Rexford, Ion Stoica, and colleagues at Princeton and UC Berkeley

1

## Announcements

- Lisa away next Friday
  - Section will be taught by Daniel
  - Her office hours will be by appointment
- Daniel is taking over the Wednesday section
- Jorge is shifting to the new Monday section

2

## Goals of Today's Lecture

- Work through process of designing **IP**, the Internet's (sole) network-layer protocol
- Assess security implications of the design

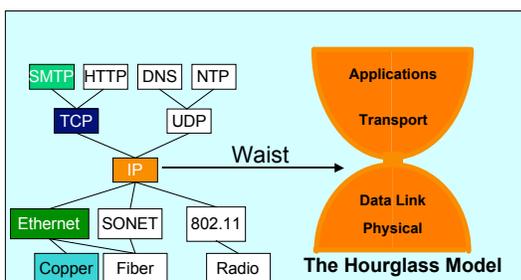
3

## Our Story So Far (Context)

- The Internet uses **packet-switching** rather than circuit-switching in order to
  - Achieve higher levels of **utilization** (we can use statistical multiplexing to more aggressively share network links)
  - Avoid **state** inside the network (**robust** fail-over)
  - Make interconnection between different parties easy (**minimal service promises**)
- The Internet architecture uses **layering** to partition functionality into modules
- The “internetworking layer” (or just **network layer**) forms the **waist** of the layering hourglass ...

4

## The Internet Hourglass



There is just **one** network-layer protocol, **IP**.  
The “narrow waist” facilitates **interoperability**.

5

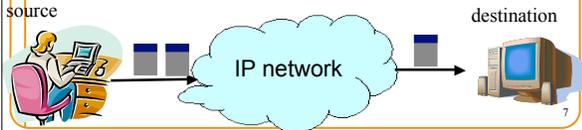
## Our Story So Far (Context), Con't

- The **End-to-End Principle** guides us in where to place functionality
  - If hosts can implement functionality correctly, implement it in a lower layer **only** as a performance enhancement
  - But do so only if it **does not impose burden** on applications that do not require that functionality
- The principle of **Fate Sharing** guides us to keep state with the elements that rely on it, when possible

6

## IP Service: Best-Effort Packet Delivery

- Packet switching
  - Divide messages into a sequence of packets
  - Each packet (datagram) is dealt with **individually**
- “Best-effort” delivery
  - Packets may be **lost**
  - Packets may be **corrupted**
  - Packets may be **delivered out of order**



7

## IP Service Model: Why Best-Effort?

- IP means never having to say you're sorry...
  - Don't need to reserve bandwidth and memory
  - Don't need to do error detection & correction
  - Don't need to remember from one packet to next
- Easier to survive failures
  - Transient disruptions are okay during failover
- ... but, applications *do* want efficient, accurate transfer of data in order, in a timely fashion

8

## IP Service: “Best Effort” Suffices

- No error detection or correction
  - Higher-level protocol can provide error checking
- Successive packets may not follow the same path
  - Not a problem as long as packets reach the destination
- Packets can be delivered out-of-order
  - Receiver can put packets back in order (if necessary)
- Packets may be lost or arbitrarily delayed
  - Sender can send the packets again (if desired)
- No network congestion control (beyond “drop”)
  - Sender can slow down in response to loss or delay

9

## Let's Design IP

- What does it mean to “design” a protocol?
- Answer: specify the **syntax** of its messages and their meaning (**semantics**).
  - Syntax = elements in packet header, their types & layout
    - o representation
  - Semantics = interpretation of elements
    - o information
- For IP, what fields do we need & why?

10

## Information to Capture in IP Header

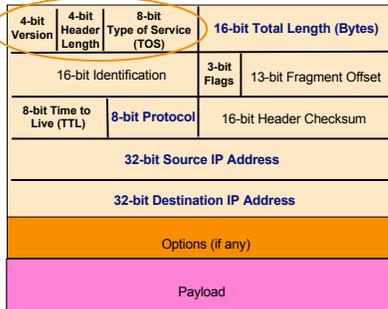
- Addresses: datagram destination & source
- Framing: datagram length, demux information
- Priority: any special forwarding?
- Extensibility: what if we need to tweak/change IP?
- Dealing with problems:
  - Integrity: is the header what it's supposed to be?
  - Loop avoidance: make sure packets don't endlessly circulate
  - Fragmentation: what if the datagram is too large?

11

## IP Packet Structure

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (Bytes)	
16-bit Identification		3-bit Flags	13-bit Fragment Offset	
8-bit Time to Live (TTL)	8-bit Protocol	16-bit Header Checksum		
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Payload				

## IP Packet Structure

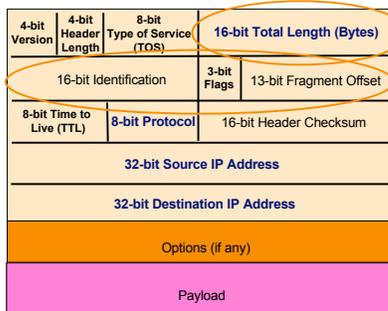


## IP Packet Header Fields

- Version number (4 bits)
  - Indicates the version of the IP protocol
  - Necessary to know what other fields to expect
  - Typically “4” (for IPv4), and sometimes “6” (for IPv6)
- Header length (4 bits)
  - Number of 32-bit words in the header
  - Typically “5” (for a 20-byte IPv4 header)
  - Can be more when IP **options** are used
- Type-of-Service (8 bits)
  - Allow packets to be treated differently based on needs
  - E.g., low delay for audio, high bandwidth for bulk transfer

14

## IP Packet Structure



## IP Packet Header Fields (Continued)

- Total length (16 bits)
  - Number of bytes in the packet
  - Maximum size is 65,535 bytes ( $2^{16} - 1$ )
  - ... though underlying links may impose smaller limits
- Fragmentation: when forwarding a packet, an Internet router can **split** it into multiple pieces (“fragments”) if too big for next hop link
- End host **reassembles** to recover original packet
- Fragmentation information (32 bits)
  - Packet **identifier**, **flags**, and fragment **offset**
  - Supports dividing a large IP packet into fragments
  - ... in case a link cannot handle a large IP packet

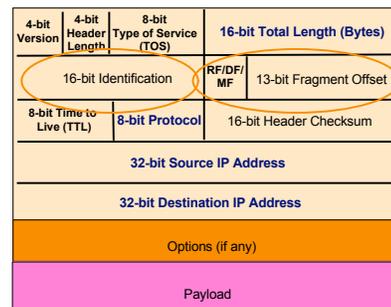
16

**5 Minute Break**

Questions Before We Proceed?

17

## IP Packet Structure



## Fragmentation, con't

- Identifier (16 bits): used to tell which fragments belong together
- Flags (3 bits):
  - Reserved (**RF**): unused bit (why “reserved”?)
  - Don't Fragment (**DF**): instruct routers to **not** fragment the packet even if it won't fit
    - o Instead, they **drop** the packet and send back a “Too Large” ICMP control message
    - o Forms the basis for “Path MTU Discovery”, covered later
  - More (**MF**): this fragment is not the last one
- Offset (13 bits): what part of datagram this fragment covers in **8-byte units**
- Thus, a fragment has **either MF set or Offset > 0**

19

## Example of Fragmentation

- Suppose we have a 4,000 byte datagram sent from host 1.2.3.4 to host 3.4.5.6 ...

Version 4	Header Length 5	Type of Service 0	Total Length: 4000	
Identification: 56273		R/D/M 0/0/0	Fragment Offset: 0	
TTL 127	Protocol 6	Checksum: 44019		
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

(3980 more bytes here)

- ... and it traverses a link that limits datagrams to 1,500 bytes

20

## Example of Fragmentation, con't

- Datagram split into 3 pieces. Possible first piece:

Version 4	Header Length 5	Type of Service 0	Total Length: 1496	
Identification: 56273		R/D/M 0/0/1	Fragment Offset: 0	
TTL 127	Protocol 6	Checksum: xxx		
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

21

## Example of Fragmentation, con't

- Possible second piece:

Version 4	Header Length 5	Type of Service 0	Total Length: 1200	
Identification: 56273		R/D/M 0/0/1	Fragment Offset: 187 (187 * 8 = 1496)	
TTL 127	Protocol 6	Checksum: yyy		
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

22

## Example of Fragmentation, con't

- Possible third piece:

Version 4	Header Length 5	Type of Service 0	Total Length: 1304 (4000 - 1496 - 1200 = 1304)	
Identification: 56273		R/D/M 0/0/0	Fragment Offset: 337 (337 * 8 = 2696)	
TTL 127	Protocol 6	Checksum: zzz		
Source Address: 1.2.3.4				
Destination Address: 3.4.5.6				

23

## Some Fragmentation Design Decisions

- Q: where are fragments reassembled?
- A: usually at the final destination. Why?
  - Because different fragments can take different paths through the network - the whole collection might only be available at the receiver
- Q: why use a byte-offset for fragments rather than a numbering each fragment?
- Ans #1: with a byte offset, the receiver can lay down the bytes in memory when they arrive
- Ans #2 (more fundamental): allows further fragmentation of fragments

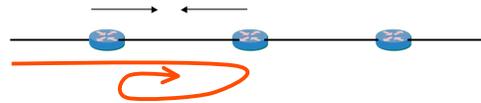
24

## IP Packet Structure

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (Bytes)	
16-bit Identification		3-bit Flags	13-bit Fragment Offset	
8-bit Time to Live (TTL)	8-bit Protocol	16-bit Header Checksum		
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Payload				

## Time-to-Live (TTL) Field (8 bits)

- Potentially lethal problem
  - Forwarding loops can cause packets to cycle forever
  - As these accumulate, eventually consume **all** capacity

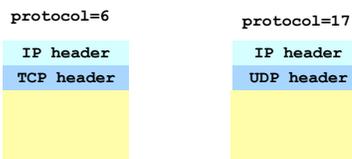


- Time-to-live field in packet header
  - Decremented at each hop, packet discarded if reaches 0
  - ...and “time exceeded” message is sent to the source
    - o Using “ICMP” control message; basis for **traceroute**

26

## IP Packet Header Fields (Continued)

- Protocol (8 bits)
  - Identifies the higher-level protocol
    - o E.g., “6” for the Transmission Control Protocol (TCP)
    - o E.g., “17” for the User Datagram Protocol (UDP)
  - Important for demultiplexing at receiving host
    - o Indicates what kind of header to expect next



27

## IP Packet Header Fields (Continued)

- Checksum (16 bits)
  - Complement of the *ones-complement* sum of all 16-bit words in the IP **packet header**
    - o I.e., bitwise addition but with carry added back in
    - o Then bitwise negation of the result
- Each router computes ones-complement sum of entire header *including checksum* ...
  - ... should get 0 (or 0xffff)
  - If not, router **discards** packet as corrupted
    - o So it doesn't act on bogus information

28

## IP Packet Structure

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (Bytes)	
16-bit Identification		3-bit Flags	13-bit Fragment Offset	
8-bit Time to Live (TTL)	8-bit Protocol	16-bit Header Checksum		
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Payload				

## IP Packet Header (Continued)

- Two IP addresses
  - Source IP address (32 bits)
  - Destination IP address (32 bits)
- Destination address
  - Unique **identifier/locator** for the receiving host
  - Allows each node to make forwarding decisions
- Source address
  - Unique identifier/locator for the sending host
  - Recipient can decide whether to accept packet
  - Enables recipient to send a reply back to source

30

## IP Address Integrity

- What are the *security implications* of the address fields?
  - Let's focus on how an attacker sending traffic can manipulate header fields ...
    - o ... rather than the problem of attackers analyzing traffic they see, such as who-talks-to-whom (*traffic analysis*) or reading payload (*eavesdropping/sniffing*)
- Source address should be the sending host
  - But, who's checking, anyway?
  - You could send packets with any source you want
  - **Why is checking hard?**

31

## IP Address Integrity, con't

- Why would someone use a bogus source address?
- Launch a **denial-of-service** attack
  - Send excessive packets to the destination
  - ... to overload the node, or the links leading to the node
  - But: victim can identify/filter you by the source address
- Evade detection by "spoofing"
  - Put **someone else's** source address in the packets
    - o Or: use a **lot of different** ones so can't be filtered
- Or: as a way to bother the spoofed host
  - Spoofed host is wrongly blamed
  - Spoofed host may receive return traffic from the receiver

32

## Security Implications of IP's Design

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (Bytes)	
16-bit Identification		3-bit Flags	13-bit Fragment Offset	
8-bit Time to Live (TTL)	8-bit Protocol	16-bit Header Checksum		
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Payload				

## Security Implications, con't

- Version field (4 bits) .... ?
  - Issue: fledgling **IPv6** deployment means sometimes connectivity exceeds security enforcement
    - o E.g., firewall rules only set up for **IPv4**
- Header length (4 bits) .... ?
  - Controls presence of **IP options**
    - o E.g., **Source Route** lets sender control path taken through network - say, sidestep security monitoring
  - Non-obvious difficulty: IP options often processed in router's **slow path**
    - o Allows attacker to stress router for **denial-of-service**
  - Often, today's firewalls configured to **drop** packets with options. (So much for extensibility! :-)

34

## IP Packet Structure

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (Bytes)	
16-bit Identification		3-bit Flags	13-bit Fragment Offset	
8-bit Time to Live (TTL)	8-bit Protocol	16-bit Header Checksum		
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Payload				

## Security Implications of TOS? (8 bits)

- What if attacker sets TOS for their flooding traffic for prioritized delivery?
  - If regular traffic does not set TOS, then network **prefers the attack traffic**, greatly compounding damage
- What if network **charges** for TOS traffic ...
  - ... and attacker spoofs the victim's source address? (denial-of-money)
- In general, in today's network TOS **does not work**
  - Due to very hard problems with **billing**
  - TOS has now been redefined for **Differential Service**
    - o Discussed later in course

36



### Security Implications of TTL? (8 bits)

- Allows discovery of **topology** (ala' *traceroute*)
- Can provide a hint that a packet is spoofed
  - It arrives at a router w/ a TTL different than packets from that address usually do
    - Because path from attacker to router has different # hops
  - Though this is *brittle* in the presence of routing changes
- Initial value that's picked is somewhat distinctive to sender's operating system. This plus other such initializations allow OS **fingerprinting** ...
  - Which in turn can allow attacker to infer its likely vulnerabilities

43

### Security Implications of Remainder?

- No apparent problems with **protocol** field (8 bits)
  - It's just a demux'ing handle
  - If value set incorrectly, next higher layer will find packet ill-formed
- Similarly, bad IP **checksum** field (16 bits) will very quickly cause packet to be **discarded** by the network

44

### Next Lecture

- IP Addressing & Forwarding
- Read K&R: 4.4.2
- If you want to check out IPv6: K&R 4.4.4

45