

ECE 435 – Network Engineering

Lecture 14

Vince Weaver

<http://web.eece.maine.edu/~vweaver>

vincent.weaver@maine.edu

3 March 2022

Announcements

- HW#5 grades out soon
- HW#6 due, will grade and return before midterm



Midterm Preview

- Can have one page (8.5" x 11") of notes if you want, otherwise closed everything. I do not think you should need a calculator.
- Mostly short answer questions. No long coding exercises or protocol memorization. Maybe some sockets code, but analyzing it not writing it.
- Know the OSI layers and what each one is for.
- Know at a high level the following protocols:
 - WWW/http



- e-mail
- DNS
- Encryption (at a high level)
- UDP + TCP
 - Know the 3-way handshake
 - Know the tradeoffs between UDP and TCP
 - Why does DNS use UDP
 - Why does HTTP1.1 use TCP



HW#5 Review

- source/destination/size/checksum src: a9a0 = 43424
dest: 35 = 53 (DNS)
size: 2a = 42 bytes
yes checksum
protocol is DNS (how can you tell?)
- Why use UDP vs TCP
lower latency, lower overhead (no need to handshake),
simpler
Recent Linux DNS debate (whether to fall back to

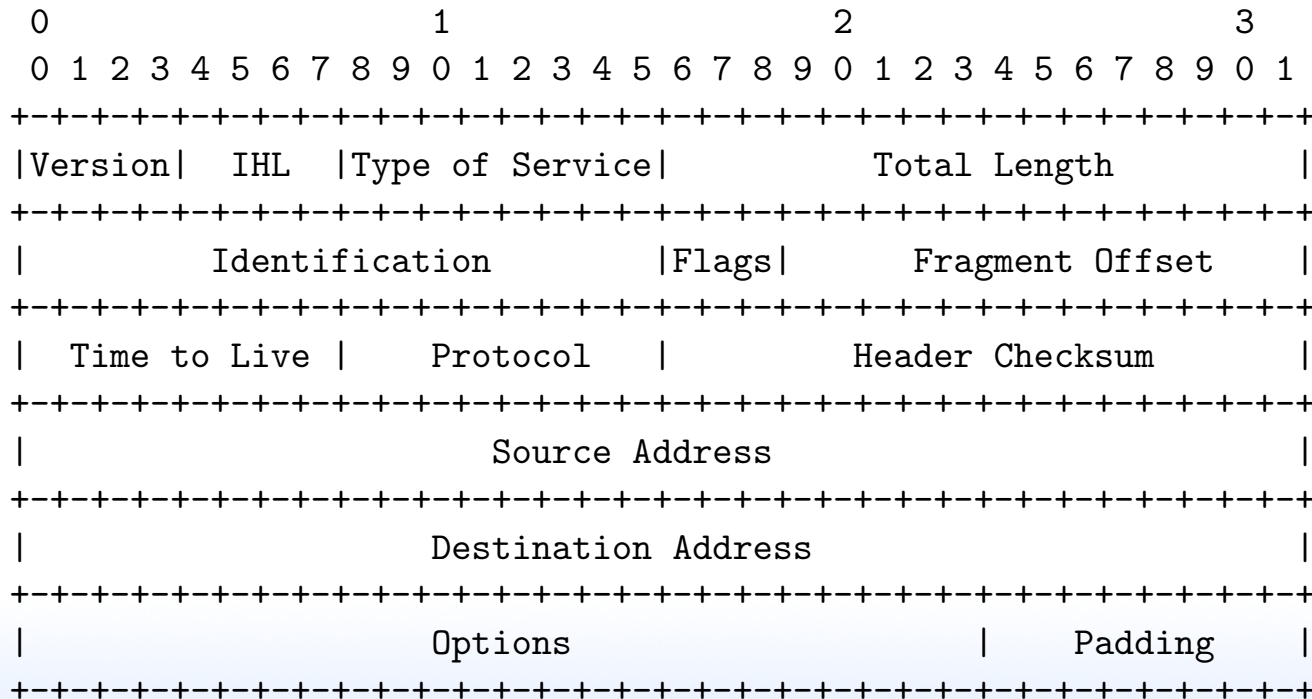


default)



IPv4 Packet Format

- Header, followed by data, multiple of 4-bytes, big-endian
- ASCII from RFC791 — <https://tools.ietf.org/html/rfc791>



- **Version** (4-bits) version number: IPv4 this is 4
- **Header Length** (4-bits) in 4-byte chunks: variable in size
Often is 5 (20 bytes) the minimum, max is 15 (60 bytes)
- **Precedence / Type of Service** (1 byte)
 - Precedence (RFC 791, high bits):
 - 111 (net control)
 - 110 (internetwork control)
 - 101 (critic/ecp)
 - 100 (Flash override)



011 (flash)

010 (intermediate)

001 (priority)

000 (routine)

○ TOS (RFC 1349):

1000 minimize delay

0100 maximize throughput

0010 maximize reliability

0001 minimize cost

0000 normal

1111 maximize security



- R: reserved
- **Total Length** (2 bytes) – max is 64kB
- **Identification** (2 bytes) – also called sequence, used in fragmentation
- **Fragmentation** (2 bytes) – fragmentation:
 - **flags** (3 bits): for fragmentation control.
 - high bit is always 0, (joke April Fools proposal ‘evil bit’)
 - next is “do not fragment”
 - last is “more fragments”
 - **fragmentation offset** (13-bits): all but last fragment must be a multiple of 8-bytes as only have 13 bits to



work with)

- **TTL** (1 byte) time-to-live, max routers allowed to pass through
 - (was supposed to be time, but ended up as a hop limit)
 - each router decreases TTL by one, if reaches zero discarded and ICMP error sent to source
 - Max is 255. why? prevent packets from wandering lost forever
- **Upper-layer protocol** (1 byte)
Originally in RFC 1700, now see www.iana.org



(ICMP=1, TCP=6, UDP=17)

- **Header Checksum** (2 bytes)
 - Sum using 16-bit 1s complement, then complementing.
 - Not as strong as CRC-16, but faster and easier in software.
 - Only checksums header (not payload).
 - Must be recomputed each hop as TTL changes
- **Source address** (4 bytes)
- **Destination Address** (4 bytes)
- **Options** – not required. rare, debugging
 - security: how secret it is (usually ignored)



- strict source: gives a list of IPs of routers to traverse
- loose: list of routers not to miss
- record route: record ips pass on way (debugging)
- timestamp(debugging)
- Data



IPv4 Packet Fragmentation

- Ethernet MTU 1500 but IP MTU is 64k, so must break up larger packets
- Can be further broken up depending on MTU along way
- Final destination is responsible for reassembling
- Can mark packet “do not fragment”. What happens then if too big?
- All fragments have same sequence number. Last



fragment marked with “more fragments” flag. Position from fragmentation offset field

- Example: original, 3200 bytes of data
 - header id=x, more=1, offset=0, 1480 bytes
 - header id=x, more=1, offset=185 1480 bytes (x8?)
 - header id=x, more=0, offset=370 240 bytes
- Each fragment is a valid IP packet



Fragmentation Limits

- RFC 791 (1981)
- IPv4 Receivers must be able to handle fragmented packets with total re-assembled size of up to 576 bytes (modern OSes can generally handle up to 64k)
- IPv4 packets under 68 bytes can't be fragmented
- Picking the id/sequence number is complex see <https://crnetpackets.com/2015/08/29/a-short-story-abo>



Problems with Fragments

- way to notify other side of missing fragments
- last fragment is usually short (wasting resources)
- receiver must hold in RAM fragments to be reassembled.
- can DoS by sending lots of fragments but none complete
- fragments do not have TCP or UDP header so firewall can't easily filter
- Most modern implementations will always disable fragmentation on TCP connections (setting DNF) and instead rely on path-mtu-discovery



Security Issues with Fragments

- ICMP/UDP larger than MTU, cannot be reassembled
- TCP “Teardrop” attack, send fragments with overlapping offsets, confuse/crash machines



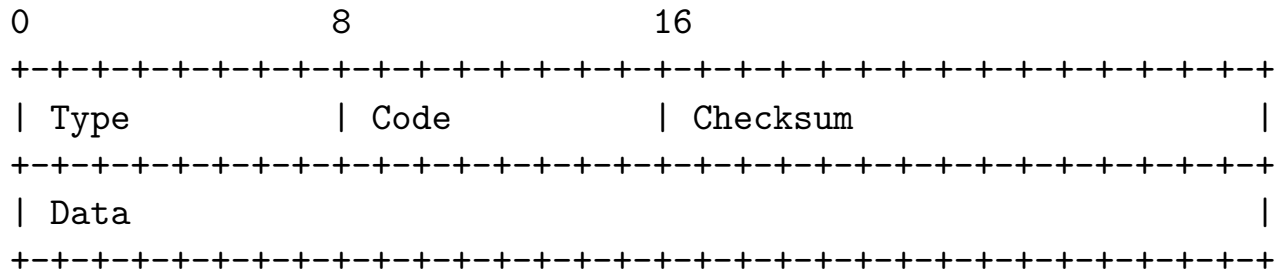
Errors

- What happens when something goes wrong with your packet?
- Does a router just drop it?
- Or does it try to let the sender know?



ICMP

- Internet Control Message Protocol
- Carried as a payload in an IP packet
- IP header type 1
- Some sysadmins block ICMP. Why?



A Selection of ICMP Types/Codes

- DESTINATION UNREACHABLE, Also if MTU is too small but do-not-fragment set
- SOURCE QUENCH – should slow transmission rate (congestion), This is now usually done in transport layer
- REDIRECT – try the other router path
- TIME EXCEEDED – exceeded TTL, traceroute uses this
- PARAMETER PROBLEM – illegal value in header
- ECHO, ECHO_REPLY – see if machine is up
- TIMESTAMP, TIMESTAMP_REPLY – performance debug



ping

- Mike Muuss in 1983
`http://ftp.arl.army.mil/~mike/ping.html`
- Like sonar ping (Hunt for Red October), not any of the backronyms you might find.
- Ping the duck
- ICMP ECHO packet, waits for ECHO reply. Prints timing info, etc.
- Used to just say “host is alive”. People would make machines called elvis.



Malicious pings

- Ping of death – crash any machine on network (late 90s)
 - Technically not a ping bug, but fragmentation
 - Ping typically 56 bytes, but can be 64k
 - Technically not valid, but most will try anyway
 - 64k ping broken into 8 fragments
 - Maximum can specify is 65528, add in 20 for header, 65548
 - This is bigger than 65536, buffer overflow on reassemble



- Ping flood
- Broadcast ping to x.x.x.255 (no longer works)



traceroute

- Van Jacobson in 1987 (also wrote tcpdump)
- Uses ICMP
- *not* tracer-t
- Send packet with $TTL=1$, when sends ICMP error message know where first hop is
- Send packet with $TTL=2$, find next
- Linux traceroute sends UDP packets as originally ICMP requests weren't supposed to generate ICMP errors
- Sends 3 packets, lists all 3 results



Dynamic Host Configuration Protocol (DHCP)

- RFC2131
- To get on network need IP, subnet mask, default router
- Can we automatically get this?
- Broadcasts, asking for address
- Server can respond with a fixed one (setup in config file) or handle out dynamically from range
- To avoid need for server on each subnet, can pass through



- Broadcast DHCPDISCOVER on UDP port 67.
- All servers send DHCPOFFER on port 68
- Send DHCPREQUEST, respond with DHCPACK
- Timer, needs to re-request before timer is out or server might give to someone else
- Get a “lease” from the server. Why short vs long lease/
- Can see this all inaction with `dhclient -v`
- DHCP format based on BOOTP



Setting up DHCP server

- Static vs Dynamic (how hand out static addresses?)
- Be careful to not hand out on network you don't own

