ECE 435 – Network Engineering
Lecture 12

Vince Weaver

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

vincent.weaver@maine.edu

18 October 2018

Announcements

- HW#6 will be posted soon
- Registering for classes: 574 next semester

Midterm Review

1. OSI Layers

   (a) Physical Layer – volts and bits

   (b) Transport Layer – packets, delivery, end-to-end connection

   not routing

2. Application Layer

   (a) http 2.0 – encryption/compression

   (b) DNS – map names to addresses, also some other

(c) open relay – primarily spam. security? spoofing?

(d) DNS MX records

3. Encryption

(a) ssh uses public key to transmit the symmetric key
authentication (password) is sort of related but not involved in the symmetric key transfer

(b) certificate authority key used to verify the SSL certificate
where does CA key live? Provided by OS or browser
who must you trust?

4. Socket Programming

(a) Yes, bug in the variable name

(b) important part was need to move the accept out of the loop or you are leaking file descriptors

(c) On read, do you check for less than or equal zero?
or that you read the buffer size?
can Linux return fewer bytes than you asked for?
What other issue? (NUL termination)

5. UDP

(a) UDP Benefit? Lower overhead. No need for 3-way handshake first

Faster? That's a complex term. On average? Limited by lower layers.

(b) Checksum was enabled, not zero.

It's actually rare to disable the UDP checksum. Special flag on Linux

(c) NTP, lower overhead. Want time from atomic clock *now*, not after TCP handshake.

6. TCP

(a) TCP over UDP: guaranteed delivery, in-order delivery.
Error handling? What type? UDP has checksum too.
It can handle lost packets.
Multiple connections to one port? UDP can do this too.

(b) 3-way handshake
Some noticed last packet had no SEQ field? This is a special "Pure ACK" that transmits no data, only the ACK. These have no sequence number.
Yes it was the start of a webserver request, but no actual data (request) had happened yet.

(c) SYN flood. Not necessarily crash machine or webserver but can definitely make unresponsive.

7. Extra Credit

(a) Brute forcing would take longer than heat death of universe?

Putting every possible md5sum in file would be 10**30 bytes?

See `POC||GTFO #14`

`https://www.alchemistowl.org/pocorgtfo/pocorgtfo14.pdf`

Made not only a PDF with own md5sum, but is also a

Nintendo rom that prints the md5sum.

The Internet Protocol v4

- RFC791
- Network of "autonomous systems" interconnected
- Transport layer takes data and breaks into dataframes of up to 64kB. Sent through Internet (possibly broken up) and when get to other side reconstructed by network layer and passed up to transport layer.
- Global and unique address.
- Need hierarchical structure to locate IP address globally

IPv4 Addresses

- Each IP address is 32-bits and has network address and host ID
- Can write many ways: decimal, hex, (all equivalent) but most common is dotted decimal (i.e. `12.34.56.78`)
- Unique to *interface* not necessarily to *host*.

Who Hands these Out?

- ICANN and various regional authorities Internet Corporation for Assigned Names and Numbers Internet Assigned Numbers Authority (IANA)
- Regional Internet Registrars
  - AfriNIC (Africa)
  - ARIN (N America),
  - APNIC (Asia-pacific)
  - LACNIC (latin america),
  - RIPE NCC (Europe and rest)

Subnets

- Number of hosts available can be larger than possible
- Divide network into subnets
- All hosts on subnet have the same prefix (left bits)
- Use subnet mask indicating the leftmost bits to use as subnet
- Can look like 255.255.255.0 meaning only bottom 8 bits are for host

  Alternately can write this as 192.168.8.0/24 (24 is number of leading binary 1s in mask)

Classful IP Routing (No Longer Used)

- Routers just shifted right for A, B, and C class. Looked up A and B in table, C in hash table to find where to send
- Has a routing entry for each Class A (128), an entry for each class B (16k). Class C (2 million) a bit much, so hash table.
- Why so simple? In 80s memory and processors were expensive!
- Original classful addressing scheme (not necessarily used

anymore)

- Class A: 8 bit network (high bit 0) (24 bits of hosts) 0.0.0.0 to 127.255.255.255
- Class B: 16 bit network, (high bits 10) 128.0.0.0 to 191.255.255.255
- Class C: 24 bit network (high bits 110) 192.0.0.0 to 223.255.255.255
- Class D: multicast (high bits 1110) 224.0.0.0 to 239.255.255.255
- Class E: reserved (high bits 1111) 240.0.0.0 to 255.255.255.255

Reserved IP Ranges

- Private Networks
  - 10.0.0.0/8 private network (RFC 1918)
  - 172.16.0.0/12 private network (RFC 1918)
  - 192.168.0.0/16 Private Network (RFC 1918)
- Loopback
  - 127.0.0.0/8 loopback (RFC 6890)
- 0.0.0.0/8 reserved for current network (RFC 6890)
- 100.64.0.0/10 shared address space (RFC 6598)
- 169.254.0.0/16 link-local (RFC 3927)

- 192.0.0.0/24 IETF (RFC 6890)
- 192.0.2.0/24 test (RFC 5737)
- 192.88.99.0/24 IPv6 to IPv4 relay (RFC 3068)
- 224.0.0.0/4 IP Multicast (class D) (RFC 5771)
- 240.0.0.0/4 Reserved (class E) (RFC 1700)
- 255.255.255.255 Broadcast (RFC 919)

Other IPv4 Conventions

- .0 represents a subnet
- .1 is often (but not always) a router
- it all host bits 1, broadcast for that subnet
- 255.255.255.255 is broadcast for device that doesn't know own IP yet (DHCP)

Classless Inter-Domain Routing (CIDR)

- Running out (have run out) of network addresses
- For many groups, Class-A too big, Class-C too small (three bears problem?)
- Merge neighboring class-C together
- RFC 1519
- Scalability problem: each network takes up space in routing table
- Solution, group neighboring class Cs together
- With CIDR bit more complex.

- Triplet with IP address, subnet mask, outgoing line.
- In theory has to scan all. If multiple matches, one with longest mask is used.
- There are algorithms to make this go faster.
- Example – from 444 in Tannenbaum

Local IP Routing

- If on same subnet, send packet directly to destination
- Otherwise, send on outgoing. See Linux route command. Often a "default router" 0.0.0.0/0. If doesn't match any other, sent out over default route
- Algorithm: if to same host, skip network. If to same subnet, deliver directly (Ethernet, etc) otherwise, send to default router
- If multiple network interfaces: If to this machine, deliver it, If to directly connected subnet, directly deliver, else

deliver to next hop router

- How do we know if on network? If ((hostIP XOR destip)&subnetmask)==0
- If local, how do we map IP to MAC? We'll see that in a minute.
- Due to CIDR, longest prefix matching. If match both a /21 and /24 then 24 is the one to send to as it's the longest.
- Data structures. Hashes? Trie?
  - Linux: two level hashing
  - BSD - trie (prefix tree)

Linux/UNIX routing setup

- “route” command
- `route add default gateway` sets default gateway (router) for packets leaving the local network
- also set up local subnets you are on, those packets don’t need a router
- more complicated if you are configuring your Linux box to *be* a router

IPv4 Packet Format

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **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,
    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 though
  - (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