ECE 435 – Network Engineering
Lecture 12

Vince Weaver

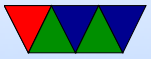
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

12 October 2017

Announcements

- HW#6 will be posted soon
- EDUCAUSE not EDUCASE
- Registering for classes: 598 and 571 next semester



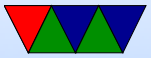
Midterm Review

1. OSI Layers

   (a) Physical Layer – volts and bits
   (b) Transport Layer – packets, delivery, end-to-end connection
   not routing

2. Application Layer

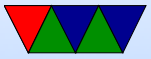
   (a) http 2.0 – encryption/compression
   (b) DNS – map names to addresses, also some other



(c) open relay – primarily spam. security? spoofing?
(d) DNS MX records

3. Encryption

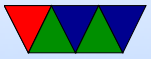
(a) ssh uses public key to transmit the symmetric key
    authentication (password) is sort of related but not involved in the symmetric key transfer
(b) certificate authority key used to verify the SSL certificate
    where does CA key live? Provided by OS or browser
    who must you trust?



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

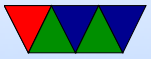
Faster? That's a complex term. On average? Limited by lower layers.

(b) Checksum was enabled, not zero.

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

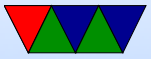
(c) NTP, lower overhead. Want time from atomic clock *now*, not after TCP handshake.

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.



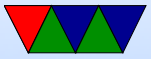
(c) SYN flood. Not necessarily crash machine or web-server but can definitely make unresponsive.

7. Extra Credit

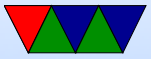
(a) Should not have asked if possible, but practical.
Brute forcing would take longer than heat death of universe?
Putting every possible md5sum in file would be 10**30 bytes?

See `POC||GTFO #14`

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

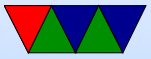


Made not only a PDF with own md5sum, but is also a Nintendo rom that prints the md5sum.

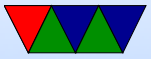


Tannenbaum on the Internet

- Everyone should be forced to read RFC 1958
- Top 10 principles
  1. Make sure it works – do not finalize the standard until someone has tried it first
  2. Keep it simple – Occam's razor. If feature not essential, leave out, especially if can be achieved via other ways.
  3. Make clear choices – if several ways to do something, choose one. Having more than one way to do something confuses things.

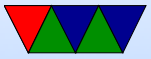


4. Exploit Modularity – protocol stacks, easy to swap out modules independently
5. Expect heterogeneity – network should be flexible enough to handle different hardware
6. Avoid static options and parameters – if limits are unavoidable, best to have sender and receiver agree
7. Perfect is the enemy of the good
8. Be strict when sending but tolerant when receiving – (note, this is not always a good idea, see web browsers)
9. Think about scalability
10. Consider performance and cost



The Internet Protocol v4

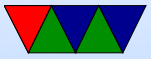
- RFC791
- Network of "autonomous systems" interconnected
- Transport layer takes data and breaks into dataframes of up to 64kB. Sent through Internet (possibly broken up) and when get to other side reconstructed by network layer and passed up to transport layer.
- Global and unique address.
- Need hierarchical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



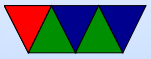
IPv4 Packet Format

- 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    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



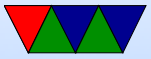
- (1 nibble [4-bits]) version number: IPv4 this is 4
- (1 nibble [4-bits]) header length in 4-byte chunks: variable in size. Often is 5 (20 bytes) the minimum, max is 15 (60 bytes)
- (1 byte) precedence/type of service/Reserved: RFC 791/RFC 1349, often ignored
  - Precedence (RFC 791, high bits):
    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

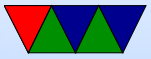
  - R: reserved

- (2 bytes) total length (max is 64kB)
- (2 bytes) identification (ids the packet)
- (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



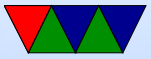
must be a multiple of 8-bytes as only have 13 bits to work with)

- (1 byte) time-to-live (TTL) 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
- (1 byte) upper layer protocol. RFC 1700 and www.iana.org (ICMP=1, TCP=6, UDP=17)
- (2 bytes) header checksum, 16-bits. Sum using 16-bit 1s complement, then complementing. Not as strong as



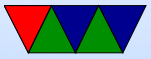
CRC-16, but faster and easier in software. Must be recomputed each hop as TTL changes

- (4 bytes) source IP
- (4 bytes) destination IP
- 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
- 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