ECE 435 – Network Engineering Lecture 12

Vince Weaver

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

vincent.weaver@maine.edu

17 October 2016

Announcements

- HW#5 will be posted soon

Midterm Review

1. OSI Layers
   (a) Physical Layer – volts and bits
   (b) Data Link Layer – frames
2. Physical Layer
   (a) Benefit of copper over fiber
   durable(?) cheaper(? depends. for local? long-distance? for what speed?) already there(? more details, if already there is old doesn't help) industry

standard (for what?) easier to repair, bendable

(b) Benefit of fiber (typo!) over copper size, won't corrode, power surges attenuation (but never "no" loss) faster. latency (? speed of light? signals can actually travel faster in copper than fiber)

(c) Satellite over fiber
broadcast

(d) Fiber over satellite
cheaper? Than what? for single connection, yes. For connecting 1 million houses, maybe not.
latency more easily repaired

3. Ethernet

(a) vs token ring
    wire breaking would take down early ethernet too
(b) collision
(c) 2.5GB existence
    power-over-ethernet? Can happen with 1GB and 10GB with injectors. Cheaper? Probably not yet. Lower latency? Probably not. Lower power? Unclear.

4. 802.11 (typo on C/B)

(a) modify collision?

DCF and PCF, ack

(b) 200dBm Believable?

No, too large

5. Bluetooth

(a) Biggest problem is both use 2.4GHz, more minor is frequency hopping interference

(b) redundant header?

simple. (power/performance?) pairing not really relevant

6. Bridging/Switches/VLAN

(a) How does it handle unknown destination? BROADCAST. Watch your wording. It broadcasts. It may never get a response, and in that case broadcast forever. If you say "broadcast and wait for reply" that's ambiguous. It doesn't really wait, especially not for a "reply". If the destination does send a frame (due to the broadcast or otherwise) it will update the table.

(b) Reasons to partition: security, bandwidth. speed? latency? Why might you have *higher* latency if you partition the network?

(c) VLAN: advantage. No running to wiring closet to update things, manage in software. "easy" is relative Expensive is also (how cheap is your intern that re-wires things).

7. Extra Credit

(a) Bob Metcalfe
(b) Radia Perlman

Homework Review

1. See midterm review, many questions were reused

dBm Example

- Juno space probe (13 Oct 2016)
  8.4GHz, received -135.75dBm (2.7e-20kW) 18kb/s

Question from Last Time

- About order when doing Dijkstra Algorithm
  I've updated the Lecture 11 notes

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 on them
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.
  Some might have multiple network ports (do they

have/need separate MAC addresses?)

- Need hierarchical structure to locate IP address globally

IPv4 Addresses

- Each IP address is 32-bits and has network address and host ID
- Who hands these out? ICANN and various regional authorities Internet Corporation for Assigned Names and Numbers
- can write many ways (all equivalent) but most common is dotted decimal
- Classful addressing scheme

- Class A: 8 bit network (high bit 0). 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: reserver (high bits 1111) 240.0.0.0 to 255.255.255.255

- Special cases

- 0.0.0.0/8 reserved for local ID
- 127.0.0.0/8 reserved for loopback
- Internal use (192.168.x.x)
- .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)

Subnets

- Number of hosts available can be larger than possible
- Divide network into subnets
- All hosts on subnet have the same prefix (left bits)
- Diagram?
- 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 something like 192.168.8.0/24

CIDR address

- Running out (have run out) of network addresses
- Merge neighboring class-C together
- Scalability problem: each network takes up space in routing table
- Solution, group neighboring class Cs together Classless Inter-Domain Routing
- Previously, 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

- 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

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?

IPv4 Packet Format

- Header, followed by data, multiple of 4-bytes, big-endian
- (1 nibble [4-bits]) version number: IPv4 this is 4
- (1 nibble [4-bits]) header length: 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

ARP – address resolution protocol

- Some way to map IP to MAC
- Could have a central server on subnet that could be queried
- Internet uses other way. When need a mapping, broadcast to the subnet, “who has this IP”
- device reply with its IP and MAC (unicast)
- These are cached

- Timeout in case you reassign
- TODO: ARP Packet?