ECE 435 – Network Engineering Lecture 11

Vince Weaver

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

vincent.weaver@maine.edu

24 February 2022

Announcements

- Midterm on Tuesday March 8th
  More details when closer

- Will try to post project info soon

HW#4 Review

- e-mail (headers)
  - First warning sign – says its from a bank, but the return address is from a Florida dental school
    Also not a bank of mine
  - encrypted and verified from UFL, but sent from videotron.ca cablemodem
  - Virus scanned and SPAM scanned, just sort of barely passed
  - pop from deater.net via fetchmail (this isn't suspicious,

it's the sender not receiver you have to look at)
  - LMTP – local mail transport. LHLO. No mail queue, says right away whether deliver mail is possible.
  - pdf attached probably had some sort of exploit or phishing document. Didn't open.
  - Note, the attachment being listed as "Application" does not mean it's an executable
- e-mail (attachment)
  - was looking for MIME as what's going on
  - Also was looking for base64 as the encoding
- e-mail (additional) mention Phishing e-mails,

Ransomware

- DNS
  - maine.edu created? What is a Registrar?
    2 December 1988, EDUCAUSE
  - A / AAAA / NS / MX
    130.111.218.23
    2607:f8b0:4006:802::2004
    nameo.unet.maine.edu / namep
    ALT4.ASPMX.L.GOOGLE.COM

QUIC

- (originally Quick UDP Internet Connections)
- RFC9000 (May 2021)
- Designed to lower latency of HTTP connections
- Over UDP for now (NATs won't route protocols they don't know)
- Used by Youtube, Google, etc. with Chromium, Firefox
- Head of Line problem with TCP
  One missing packet blocks processing all previous
- Single-way handshake if version match, otherwise has to

negotiate.

- Encrypted

  encrypted by packet rather than full stream so can handle missing packets better
- Once encrypted connection set up once, assumed still there and so sends single HELLO packet followed by data.
- Sends redundancy in packets which can be used with XOR to reconstruct missing packets. (but shown not to help much?)
- Help in roaming, TCP have to rely on timeouts to notice

this. QUIC has identifier for connection that can restart

The Network Layer

- Also "the internet protocol layer"
- Get packets from source to destination
- May require multiple hops
- Transport Layer runs mostly on the endpoint machine, but Network Layer happens along the routers along way
- Critical, and much more complicated than Link Layer
- Connectivity, Scalability, and Resource Sharing problems

Network Layer Design Issues

- Should be independent of router tech, should hide topology and num, type of routers

- Need to send packets between any two machines, globally:

1. How to identify a host globally (addressing)
2. How to connect different networks together
3. How to find a path between two hosts

Internetworking

- Connecting various types of networks (ethernet, 802.11, etc)
- A group of LANs connected togeter is an inter-network, or "Internet"

Connection vs Connectionless

Connectionless / Packet-Switched

- Packets sometimes called Datagrams
- Packets injected into network with no prior setup
- Router responsible for picking how it gets there, routing algorithm
- Router makes "best-effort". Tries to get things there, but if packet gets lost, goes to wrong place, or arrives out of order it doesn't have to do anything about it.
- Example: Internet
- Send/Receive packet primitives.

- Packet ordering/flow control by higher level
- Each packet carry full destination address, as may travel independetly of predecessors

Connection-Oriented

- Virtual circuit created
- Avoid creating a new route for every packet
- A route from source to destination created in all routers along the way
- Each packet carries an ID saying what route it belongs to
- Example: Old POTS telephone land-line network
  older cell phones?

Connectionless vs Connection Tradeoffs

- Setup: none / required
- Addressing: full source + dest / short virt circuit num
- State: no router state / each virt circuit has state
- routing: each packet independent / routing done at startup
- router failure: can route around / all virt circuits terminated
- QoS: difficult / easy if resources allocated in advance
- congestion: difficult / easy of allocated in advance

Routing

- Routing protocols, lead to routing tables
  Table is destination paired with next hop
- goals
  - minimize routing table space (take up room, also pass around)
  - minimize control messages
  - robustness (don't want to misroute)
- choices:
  - centralized vs distributed

- source-based v hop-by-hop. Source you specify entire path at beginning, hop decides each hop along away
- stochastic vs deterministic – deterministic each hop has one route, stochastic multiple routes, picks randomly
- single vs multiple path – one path or if alternate available
- state-dependent vs state-independent – whether you balance based on load. can be better, but can also lead to problems if choose poorly, also extra overhead

Routing and Forwarding

- Routing: whch routes to use, find shortest path
- Forwarding: looking up which outgoing line to use
- Characteristics: simplicity/efficiency , robustness, stability, fairness, optimality
- Simplicity: packets stored on routers, efficient resource sharing
  maintain good performance (low delay and packet loss)

- Robustness: cope with changes w/o requiring all jobs stopped and rebooted
- Stability: routing eventually converges on an equilibirum
- Fairness and optimality often conflicting
- Fairness example?
- Unicast routing: point to point
- Multicast routing: one to many or many to many

Routing Algorithm Types

- Nonadaptive: not based on measurement, but computed in advance. Static routing. sysadmin sets them. Do not adapt well if routers fail.

- Adaptive: change routing decisions to reflect changes in topology and traffic
  centralized – require global information
  quasi-centralized (?)
  distributed – ?
  hop-by-hop (internet. source routing?)

Optimal Route?

- What do we optimize? Latency? Throughput? Number of hops?

- Something like ssh might want lowest latency

- Multimedia might want high bandwidth and low jitter

- Often a "cost" is defined based on the desired characteristics, and then this is optimized for

Optimality Principle

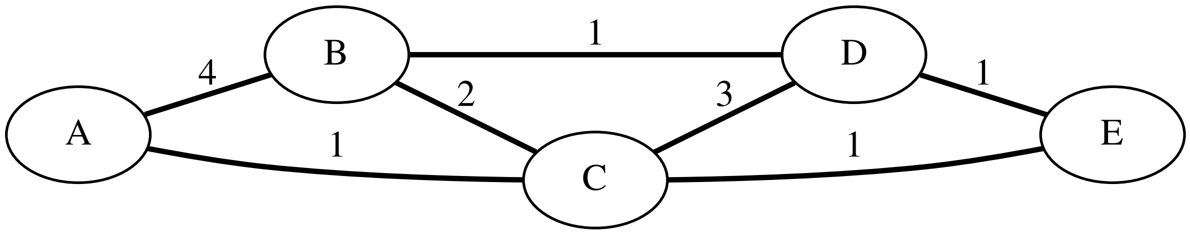
- If router J is on optimal path from I to K, then optimal path J to K is on same route
- Set of all optimal routes from all sources to a destination form a tree rooted at desitnation, called a “sink tree”. Not necessarily unique
- Tree and not a loop, so packets delivered in finite number of hops
- Though routers can come and go so things can go wrong

(static) Shortest Path Routing

- Number of hops?
- Length (in meters?)
- Transmission delay?

(static) Link State Routing

- Requires global information, routers broadcast the info so all have consistent view
- Dijkstra Algorithm
  Form least spanning tree
  Find lowest cost iteratively
- Iterative algorithm, takes N-1 iterations
- Example based on one from Lin/Hwang/Baker



T=set of known machines, C(X)=cost of X, p(X)=previous hop

Iteration	T	C(B),p(B)	C(C),p(C)	C(D),p(D)	C(E),p(E)
0	A	4,A	1,A	∞	∞
1	AC	3,C		4,C	2,C
2	ACE	3,C		3,E	
3	ACEB			3,E	
4	ACEBD				

Iterative. Start not knowing anything but direct connections. Pick shortest cost and add to set. Update all the link costs. Repeat until all nodes added.

For second step, cost to get to C is 1. So then from there cost to B through C is 2+1 (3) which is shorter, so update. Cost to D is 1+3 (4), update, etc.

Final routing table for A.

Path	Cost	Next
A-B	3	C
A-C	1	C
A-D	3	C
A-E	2	C

(static) Flooding

- Every packet sent out on every outgoing line, with a counter (set to the distance) so after so many hops discarded
- Selective flooding, only floods out the connections going in vaguely the right direction
- Very robust (can handle if routers droppign out constantly)
- Flooding always chooses shortest path, as it finds all

possible paths in parallel