Final Review

15-441 Fall 2017
Profs Peter Steenkiste & Justine Sherry
Facts

• The Final is next week.

• It mostly covers things after the midterm, but there is significant material from before.

• There were two TCP questions on the original version of the midterm, but we cut one because it was too long. That TCP question is now on the final.

• The big things to know are Security, Wireless, Link Layer, and QoS/Prioritization.
Your Piazza Questions broke down into two camps: TCP/CWNDs and CSMA/CD, CA, etc.
What is the relationship between CWND and BDP?
What is CWND?
CWND: The number of unacked bytes that TCP can have outstanding.
CWND is another way of saying: how many bytes should TCP be able to have “in the network” without B having received them yet?
Abstract View of Link

A

Sending Host

→

B

Receiving Host
Abstract View of Link

Sending Host  

Receiving Host
Abstract View of Link

- A sends to B at a rate of $r$ Mbps over a link with latency $d$ millisec.
Abstract View of Link

- A sends to B at a rate of $r$ Mbps over a link with latency $d$ millisec.
- In $b$ milliseconds, A can stuff $r \times d$ packets “into the pipe”
- These packets have been sent by A, but have not yet been received by B
Abstract View of Link

- $r \times d$ is our *bandwidth-delay product*
- AKA how much stuff “fits in the pipe” at a time.
Big Idea: if A maintains that there are $r \times d$ bytes in that pipe constantly, then A can “keep the pipe full” — meaning achieve the maximum possible sending rate through that pipe.
Hence, perfect CWND is $2 \times r \times d$ (or $r \times RTT$).

Why * 2?
Need to wait for the ACKs!
Abstract View of Link

2 * r * d total packets in the network.
From the homework: Wireshark does not show CWND directly. Give your best estimation about the initial CWND and explain how did you get the value.
Anyone want to explain how they did it?
Count the number of packets in the pipe!

After the handshake, you should have seen some number of packets leave A — how many packets left A before you received the first ACK back for that first packet after the handshake?
Selective ACKs in TCP
SACKs are useful when you have multiple losses.
By default, TCP uses cumulative ACKS.

Return one value: the next byte expected.
By default, TCP uses cumulative ACKS.

Return one value: the next byte expected.
On loss: only know one byte that was lost: the next byte expected.
On loss: only know one byte that was lost: the next byte expected.
On loss: only know one byte that was lost: the next byte expected.
On loss: only know one byte that was lost: the next byte expected.
With SACK, we send back a “scoreboard” of ACKs rather than a cumulative ACK.
With SACK, we send back a “scoreboard” of ACKs rather than a cumulative ACK.

Can retransmit more efficiently!
The difference between CSMA/CA and CSMA/CD
Quick, someone describe CSMA/CD!
Key to CSMA/CD: everyone who wants to transmit can detect when collision happens. (This is not the case in wireless.)
Hidden Terminals

A

B

C
Hidden Terminals
Hidden Terminals

A

B

C

transmit range
• A and C can both send to B but can’t hear each other
  – A is a hidden terminal for C and vice versa
• A and C can both send to B but can’t hear each other
  – A is a *hidden terminal* for C and vice versa
• Carrier Sense will be ineffective
Exposed Terminals

A • B • C • D •
Exposed Terminals

A B C D
Exposed Terminals
• **Exposed node**: B sends a packet to A; C hears this and decides not to send a packet to D (despite the fact that this will not cause interference)!
Exposed Terminals

- **Exposed node**: B sends a packet to A; C hears this and decides not to send a packet to D (despite the fact that this will not cause interference)!

- Carrier sense would prevent a successful transmission.
Key Points
Key Points

• No concept of a global collision
  – Different receivers hear different signals
  – Different senders reach different receivers
Key Points

• No concept of a global collision
  – Different receivers hear different signals
  – Different senders reach different receivers

• Collisions are at receiver, not sender
  – Only care if receiver can hear the sender clearly
  – It does not matter if sender can hear someone else
  – As long as that signal does not interfere with receiver
Key Points

• No concept of a global collision
  – Different receivers hear different signals
  – Different senders reach different receivers

• Collisions are at receiver, not sender
  – Only care if receiver can hear the sender clearly
  – It does not matter if sender can hear someone else
  – As long as that signal does not interfere with receiver

• Goal of protocol:
  – Detect if receiver can hear sender
  – Tell senders who might interfere with receiver to shut up
Basic Collision Avoidance
Basic Collision Avoidance

• Since can’t detect collisions, we try to *avoid* them
### Basic Collision Avoidance

- Since can’t detect collisions, we try to **avoid** them

- **Carrier sense:**
  - When medium busy, choose random interval
  - Wait that many **idle** timeslots to pass before sending
Basic Collision Avoidance

• Since can’t detect collisions, we try to *avoid* them

• Carrier sense:
  – When medium busy, choose random interval
  – Wait that many **idle** timeslots to pass before sending

• When a collision is inferred, retransmit with binary exponential backoff (like Ethernet)
  – Use **ACK** from receiver to infer “no collision”
  – Use exponential backoff to adapt contention window
### CSMA/CA -MA with Collision Avoidance

| sender | receiver | other node in sender’s range |

- Before every data transmission
• Before every data transmission
  – Sender sends a Request to Send (RTS) frame containing the length of the transmission
CSMA/CA - MA with Collision Avoidance

- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver respond with a Clear to Send (CTS) frame
• Before every data transmission
  – Sender sends a Request to Send (RTS) frame containing the length of the transmission
  – Receiver responds with a Clear to Send (CTS) frame
  – Sender sends data
CSMA/CA -MA with Collision Avoidance

- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver respond with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data
**CSMA/CA - MA with Collision Avoidance**

- Before every data transmission
  - Sender sends a Request to Send (RTS) frame containing the length of the transmission
  - Receiver responds with a Clear to Send (CTS) frame
  - Sender sends data
  - Receiver sends an ACK; now another sender can send data

- When sender doesn’t get a CTS back, it assumes collision
CSMA/CA, con’t

receiver

sender

other node in sender’s range

RTS

CTS

data

data
• If other nodes hear RTS, but not CTS: **send**
  – Presumably, destination for first sender is out of node’s range …
CSMA/CA, con’t

• If other nodes hear RTS, but not CTS: send
  – Presumably, destination for first sender is out of node’s range
    ...
  – … Can cause problems when a CTS is lost
CSMA/CA, con’t

- If other nodes hear RTS, but not CTS: send
  - Presumably, destination for first sender is out of node’s range
  …
  – … Can cause problems when a CTS is lost

- When you hear a CTS, you keep quiet until scheduled transmission is over (hear ACK)
RTS / CTS Protocols (CSMA/CA)

B sends to C
Overcome hidden terminal problems with contention-free protocol.
Overcome hidden terminal problems with contention-free protocol

1. B sends to C Request To Send (RTS)
Overcome hidden terminal problems with contention-free protocol

1. B sends to C \textit{Request To Send} (RTS)
2. A hears RTS and defers (to allow C to answer)
Overcome hidden terminal problems with contention-free protocol

1. B sends to C Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
Overcome hidden terminal problems with contention-free protocol

1. B sends to C Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
Overcome hidden terminal problems with contention-free protocol

1. B sends to C Request To Send (RTS)
2. A hears RTS and defers (to allow C to answer)
3. C replies to B with Clear To Send (CTS)
4. D hears CTS and defers to allow the data
5. B sends to C
Alt: Preventing Collisions Altogether
Alt: Preventing Collisions Altogether

– Partition space into non-overlapping cells.
Alt: Preventing Collisions Altogether

– Partition space into non-overlapping cells.
Why doesn’t Google use anycast to redirect traffic to its servers instead of DNS redirection?
Answer: performance
Answer: performance

• Recall: routing over the Internet follows sub-optimal paths. Why?
Answer: performance

• Recall: routing over the Internet follows sub-optimal paths. Why?

• “Follow the money” policies (choose cheap routes over short ones)
Answer: performance

• Recall: routing over the Internet follows sub-optimal paths. Why?
  • “Follow the money” policies (choose cheap routes over short ones)
  • AS path lengths don’t necessarily correlate to shortest geographic paths / lowest latencies anyway
Answer: performance

• Recall: routing over the Internet follows sub-optimal paths. Why?
  • “Follow the money” policies (choose cheap routes over short ones)
  • AS path lengths don’t necessarily correlate to shortest geographic paths / lowest latencies anyway
  • And it’s common for operators to just misconfigure stuff too.
If we use anycast, we’re letting the sub-optimal routing protocols take charge: no guarantee they’ll pick the best choice.
With DNS redirection, Google can pick best route themselves, based on what IP address their DNS server returns for the domain name.
Bonus question: if anycast leads to sub-optimal routing, why does Google DNS (8.8.8.8) use anycast?
Answer: it’s a “chicken and egg” problem. You can’t use DNS redirection to find your optimal DNS resolver, because you’d need to use your DNS resolver to look up its own DNS name…
What is a PKI?
Public Key Infrastructure
Public Key Infrastructure

• How do I get the public key of someone I don’t know? How do I know it’s them?
Public Key Infrastructure

• How do I get the public key of someone I don’t know? How do I know it’s them?

• I can go find them in the real world, check their ID card, get their public key.
Public Key Infrastructure

• How do I get the public key of someone I don’t know? How do I know it’s them?

• I can go find them in the real world, check their ID card, get their public key.

• Or, I can ask a central database that authoritatively stores everyone’s public key.
Public Key Infrastructure

• How do I get the public key of someone I don’t know? How do I know it’s them?
  
  • I can go find them in the real world, check their ID card, get their public key.
  
  • Or, I can ask a central database that authoritatively stores everyone’s public key.
  
  • Everyone registers with the central database, and anyone can look up anyone else’s public key in the database.
Public Key Infrastructure

• How do I get the public key of someone I don’t know? How do I know it’s them?

  • I can go find them in the real world, check their ID card, get their public key.

  • Or, I can ask a central database that authoritatively stores everyone’s public key.

  • Everyone registers with the central database, and anyone can look up anyone else’s public key in the database.

• Implication: everyone has to trust the key provider.
What does the Hamming Distance figure in the lecture slides mean?
...I don’t understand the figure either.
Here’s a review of Hamming Distances though.
How many bit flips do you need to turn one binary string into another?

That’s it. That’s Hamming distance.
Why do Hamming distances matter?

• Hamming distances put a bound on how much a Forward Error Correction algorithm can fix corrupted bitstrings back to their original value.

  • If the valid words of a code have minimum Hamming distance $D$, then $D-1$ bit errors can be detected.

  • If the valid words of a code have minimum Hamming distance $D$, then $[(D-1)/2]$ bit errors can be corrected.
Good luck!
My office hours ARE being held next week, but
Prof Steenkiste’s are not.
TA’s OH are pending their own exam schedules.