ECE 435 – Network Engineering
Lecture 19

Vince Weaver

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

vincent.weaver@maine.edu

13 November 2018

Announcements

- HW#8 was Posted

Data Link Layer

- All about frames.
- Transmitting values to nearby machines: ones/zeros go out to physical layer, same bits arrive back on other machine

Link Layer – Issues

- Addressing – specify source/destination
- Framing – split data into frames
- Error control and reliability
- Flow Control – stop from sending too fast
- Medium Access Control – method to decide which host gets to transmit (handle collisions)

Framing

- Break up data stream into frames, checksum each on send and receive
- How do you break up into frames? (Delimiting)

1. Character count – send a byte describing how many chars follow, followed by that many chars
   Trouble is, what if count affected by noise. Then the data gets out of sync, no way to resync
2. Flag bytes – special byte indicates start and stop, you can then use to find frame boundaries

What to do if flag byte appears in data you are sending? Use escape chars (sometimes called "byte stuffing"?)

3. Bitstuffing – instead of sending multiples of 8 bits, send arbitrary bit widths, with special bit patterns as flags
4. Physical layer coding – use some of the ones we discussed, where you can 4B/5B or such where you can use the unused values as frame markers

Frame Format

- Frame and Packet sometimes used interchangeably
- Usually a header, with address, length, type, error detection
- Followed by data
- Might be trailer at end

Addressing

- How do you determine which machines gets data?
- How do you know who to respond to?
- Global or local? Only few extra bits of extra overhead so often global these days (MAC address?) IEEE 802 is 48-bits. Is that enough?

Flow Control

- What if sender tries to send faster than receiver can handle?
- Feedback based: receiver sends back info saying it is ready for more (serial with HW flow control)
- Rate-based flow control. The rate is set in the protocol. Not really used in the link layer

Medium Access Control (MAC)

- Whose turn is it to send or receive?
- What if on a shared medium (wire, spectrum)

Error Control / Reliability

- You detect an error, what can you do?
  - Drop it on the floor? ("Best Effort") Maybe hope another layer helps
  - Get an acknowledgement saying was correct?
  - What if something happens and the entire frame lost? Receiver never gets it one way or another. Sender waits forever?
  - Use a timer. If no response send again
  - What happens if you send multiple times and then

eventually both get there? Often have a sequence number to track if there are multiple.

- Very quickly end up re-implementing the net layer at this layer.

Error Detection/Correction

- Are errors a problem? If sending 1000 bit frames, and error rate is .001 per bit, then if even distributed on average each frame have an error. Are errors evenly distributed? What if 1000 in a row then none? (bursty)
- Error-Detection Codes – let you tell if an error happened what to do if error happens? resend. doable if errors infrequent (reliable connection)
- Error-Correcting Codes – let you fix an error

Hamming Distance

- Number of bits that differ
- Can calc by exclusive oring then counting the ones.
- 0101 1101 = 1000 = 1
- If hamming distance of N then takes N single-bit errors to convert between the two
- To detect N errors you need hamming distance of N+1 to ensure than N errors can create another valid code

- To correct N errors you need 2N+1 distance, that way even with N errors it is still closer to changed value than any other
- parity bit. Chosen so code word is always even (or odd) can detect single bit error
- Hamming code for detecting errors

Error Detecting Codes

- One way: arrange bits in rectangle, take parity bits across both rows and columns

- Polynomial codes: CRC (cyclic redundancy check)

Checksum

- Like in TCP/UDP
- Easy to calculate in software

CRC check

- Easier to calculate with hardware
- Polynomial, 110001 means $x^5 + x^4 + x^0$
- Agree on generator in advance. High and low bits must be 1. Checksum is calculated. Value to check must be longer than generator
- Append checksum on end, and when run through the result is zero. Any remainder means an error

- IEEE 802 uses $x^{32} + x^{26} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$
  which can detect any burst error less than 32 and all odd number bits

- might seem hard, but easy to make in hardware with a shift register and some xor gates.

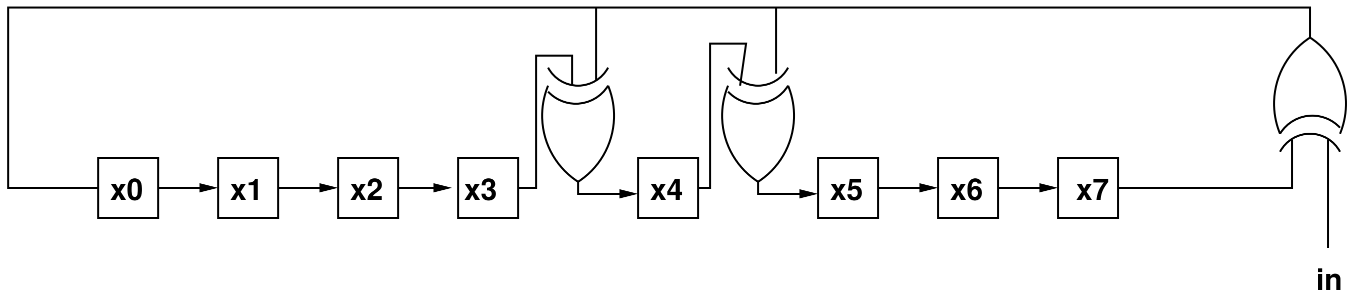
- CRC can find single bit errors, double bit errors, bursts of errors less than length of polynomial.

- Explaining how it works is “mathematically complex” says open source approach book

1-wire CRC check example

- Usually used in hardware, harder to implement in software
- Can detect all double-bit errors, any double bit errors, any cluster within an 8-bit window
- if CRCs with itself gets 0 at the end, how hardware detects correct address.
- $X^8 + X^5 + X^4 + X^1$

- Fill with zero, shift values in.



-

Example Link Protocols

- Obsolete/Fading
  - Token Ring
  - HIPPI, FDDI – fiber distributed data interface
  - Fibre Channel
  - ATM
  - ISDN
  - X.25
  - NCP, IPX, Appletalk
- Current

- Ethernet (802.3)
- PPP (? fading)
- WLAN (802.11)
- Bluetooth (802.15)
- FiberChannel (?)
- DSL
- LTE/WiMAX (802.16)

PPP / HDLC

- Older textbooks like to talk about PPP
- Point to Point Protocol, used when serial/modems were the range
- Remnant still exists, ASDL oftened tunneled via PPPoE (PPP over Ethernet)
- PPP based on earlier protocol called HDLC
- We will skip over it as not being super relevant anymore