ECE 435 – Network Engineering
Lecture 21

Vince Weaver

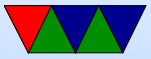
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

7 April 2022

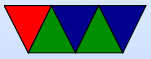
Announcements

- HW#9 (Ethernet) will be posted
- Don't forget project status update due April 15th (next Fri)



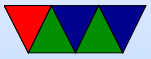
Project Status

- Summary of topic
- What HW/SW you'll be using
- On track to finish
- Say if you're willing to present early (Tuesday vs Thursday)
- Note: course website will be doen Friday to Monday. If you plan to do HW#9 over the weekend, please get it



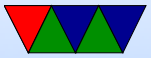
in advance (or e-mail me for copy)

- Note: due to faculty search interview talks, office hours for April 18th and 20th are cancelled, let me know if you need to meet for something and I can arrange a time



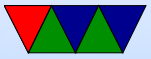
Fast Ethernet (100Mbps)

- 10Mbps not fast enough! What can we do?
  - FDDI and Fibrechannel (fast optic-ring), too expensive
  - Can we just multiply all speeds by 10? Or else come up with some completely new better thing?
  - IEEE decided to just keep everything same, just faster
  - The other group went off and made 802.12 100BaseVG (which failed)
- 802.3u 1995

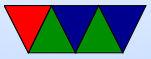


Fast Ethernet (100Mbps)

- 100BASE-TX most common
- Bit time from 100nsec to 10nsec
- Uses twisted pair/switches, no coax
  - To use cat3 100BASE-T4 wiring needed 4 twisted pair and complex encoding, no Manchester, ternary
  - To use cat5 wiring 100BASE-TX. Two twisted pair, one to hub, one from.
- Often split between MAC (media access controller) and PHY (physical interface). Card configures the PHY via

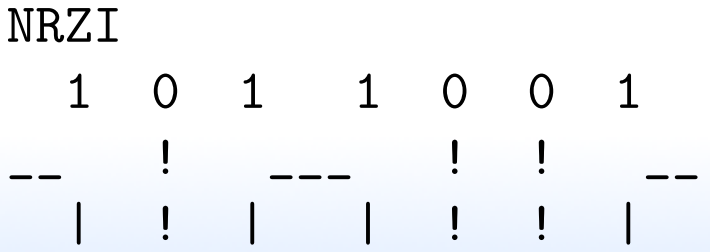


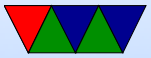
the MII (media independent interface)
Goal was you could have same card but interchangeable PHY (twisted pair, fiber, etc). 4bit bus
Interface requires 18 signals, only two can be shared if multiple PHY
So RMII (reduced) was designed. Clock doubled, only 2-bit bus. Fewer signal wires.

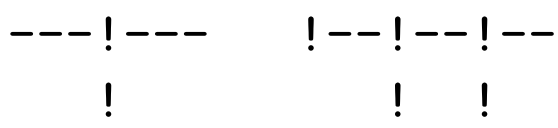


100BASE-TX

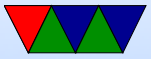
- 2 pairs. One pair 100MB each direction, full duplex, 100m distance
- Raw bits (4 bits wide at 25MHz at MII) go through 4B/5B encoding clocked at 125MHz. (DC equalization and spectrum shaping)
- Then non-return-to-zero inverted (NRZI) encoding (transition at clock if 1, none if 0).





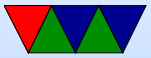


- 4B/5B means 4 bits encoded as 5, this allows at least two transitions per 5 bits, allows special patterns for end frame or error (TODO: table of conversion?)
- TX then goes through MLT-3 encoding (-1,0,+1,0. Transition means 1, no transition means 0) 31.25MHz, much like FDDI



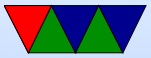
Router vs Hub vs switch

- Hub all frames are broadcast to all others
  Bandwidth is shared (only say 100MB for all)
- Switch – direct connection, no broadcast. Has to be intelligent. Each point to point connection full bandwidth.
  no collisions. Internally either own network to handle collisions, or else just buffer RAM that can hold onto frames until the coast is clear.
- Multi-speed hubs?



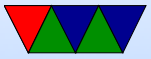
When 10/100MB first came out, cheap hubs could only run at 10MB or 100MB. But switches *really* expensive. They had a compromise 10/100MB hub that internally had a hub for both then a mini-switch to bridge the gap.

- Router will move frames from one network to another
- Lights. How many ports? Uplink ports?



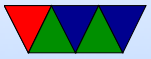
Direct Connection Ethernet

- Direct connect two machines with one cable
- Used to need special “crossover” cable to swap TX and RX lines
- Modern cards can detect direct connect and swap the wires for you

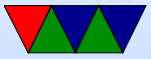


Ethernet Security

- Traditional hub, all machines saw all packets
- With tcpdump could monitor all packets on network, back in day all plain text. e-mail, web-browsing, chat, passwords, telnet
- tcpdump put card in "promiscuous" mode which let it intercept all packets instead of ignoring ones not to system
- Why so low security? Old day trust people at your work/office, also was probably expensive/difficult to get

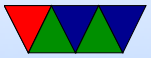


an unauthorized UNIX workstation with ethernet card and root access on the local network



Power over Ethernet

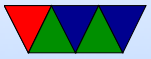
- Method B: In 10/100 Base T, only 2 of the 4 pairs in Cat5 used. So send voltage down spare pairs
- Method A: send DC voltage down with the signals floating on top
- Original 44 VDC, 15.4W
- POE+ 25W
- Need special switch to send power, and device on other end has to support it.



Gigabit Ethernet

- Two task forces working in 1998/1999
- 802.3z 1998 (fiber), 802.3ab 1999 (copper)
- Could still use hub, problem was the CSMA/CD restriction.
  - About 200m for 100Mbps.
  - For Gb would have been 20m which is not very far.
  - Carrier extension: hardware transparently pads frames to 512 bytes

    Wasteful, 512 bytes to send 64 bytes of data

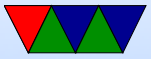


- Frame bursting: allow sender to sends sequence of multiple frames grouped together

- Better solution is just use full duplex
- 1000Base-SX (fiber)/LX (fiber)/CX (shielded)/T (cat 5), more
- Fiber
  - No Manchester, 8B/10B encoding. chosen so no more than four identical bits in row, no more than six 0s or six 1s

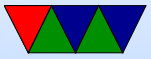
    need transitions to keep in sync

    try to balance 0s and 1s? keep DC component low so



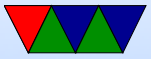
can pass through transformers?

- 1000BASE-T
  - 5 voltage levels, 00, 01, 10, 11, or control. So 8 bits per clock cycle per pair, 4 pairs running at 125MHz, so 1GBps
  - simultaneous transmission in both directions with adaptive equalization (using DSPs), 5-level pulse-level modulation (PAM-5) [technically 100BASE-TX is PAM-3]. Diagram? looks sort of like a sine wave as cycle through the voltages.
  - four-dimensional trellis coded modulation (TCM) 6dB



coding gain across the four pairs

  - Autonegotiation of speed. Only uses two pairs for this, can be trouble if pairs missing.
- Fast enough that computers at time had trouble saturating such a connection
- Jumbo Frames? 9000 byte?

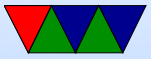


Even Faster Ethernet

http://www.theregister.co.uk/2017/02/06/decoding_25gb_ethernet_and_beyond/

- Misquote: Not sure what the network will be like in 30 years, but they will call it Ethernet.

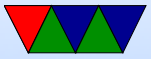
- 2.5Gb: 802.3bz (Sep 2016?)
  Like 10Gb but slower. Can't run 10Gb over Cat5e
  Power over Ethernet (for using on wireless access points)
  Power with signal overlaid on top.
  2.5Gb on Cat 5e, 5Gb on Cat6



- 10Gb: 802.3ae-2002. Full duplex, switches only
  Need Cat6a or Cat7 for links up to 100m
  Expensive. Lots of kind. 10GBASE-T, 802.3an-2006
  100m over cat6a, 55m Cat6
  additional encoding overhead, higher latency
  Tomlinson-Harashin precoding (THP), PAM15 in two-dimensional checkerboard DSQ128 at 800Msymbol/s

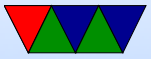
- 25Gb, 802.3by. 25GBASE-T, 50GBASE-T. Available, if copper only a few meters

- 40GB, 100GB. 802.3ba-2010, 802.3bg-2011, 802.3bj-



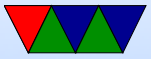
2014, 802.3bm-2015
40GBASE-T twisted pair 40GBit/s 30m. QFSP+ connectors, like Infiniband

- Terabit? still under discussion



Autonegotiation

- How figure out line speed and duplex
- Series of pulses sent along periodically
- If not received for 150ms, link is detected as down
- Encoded in the 16-bit series of pulses is the speed / duplex available



What does your machine have

- skylake machine:

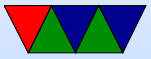
```
[   18.240021] e1000e: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: Rx/Tx
```

- Raspberry Pi:

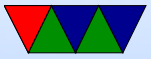
```
[   77.110505] smsc95xx 1-1.1:1.0 eth0: link up, 100Mbps, full-duplex, lpa 0xC5E1
```

- Haswell machine:

```
[    3.907651] tg3 0000:03:00.0 eth0: Tigon3 [partno(BCM95761) rev 5761100]
(PCI Express) MAC address f0:92:1c:f5:e8:f3
[    3.919115] tg3 0000:03:00.0 eth0: attached PHY is 5761
(10/100/1000Base-T Ethernet) (WireSpeed[1], EEE[0])
```

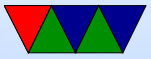


```
[    3.929838] tg3 0000:03:00.0 eth0: RXcsums[1] LinkChgREG[0] MIirq[0] ASF[1] TSOcap[1]
[    3.938174] tg3 0000:03:00.0 eth0: dma_rwctrl[76180000] dma_mask[64-bit]
[   13.758613] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[   15.404905] tg3 0000:03:00.0 eth0: Link is up at 100 Mbps, full duplex
[   15.411479] tg3 0000:03:00.0 eth0: Flow control is on for TX and on for RX
```



Linux OS Support

- When frame comes in, interrupt comes in
- Allocates `sk_buff` copies in
- Old: `net_if_rx()` interrupt, `net_rx_action()` interrupt/polling
- `net_if_receive_skb()`
- passes it to proper net level (`ip_recv()`, `ip_ipv6_recv()`, `arp_recv()`



- for send
- `net_tx_action()`
  `dev_queue_xmit()` and then deallocate `sk_buff`
- `qdisc_run()` selects next frame to transmit and calls `dequeue_skb()`