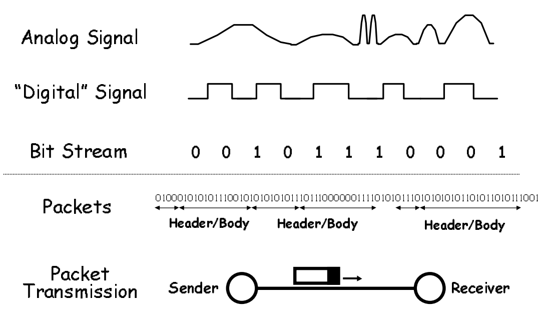
CS 640: Computer Networks

Aditya Akella

Lecture 6 -
Datalink Layer I

Signals and Binary Data



Datalink Protocol Functions

1. **Framing**: encapsulating a network layer
   - Add header, mark and detect frame boundaries, …
2. **Error control**: error detection and correction to deal with bit errors.
   - May also include other reliability support, e.g. retransmission
3. **Error correction**: Correct bit errors if possible
4. **Flow control**: avoid sender outrunning the receiver.
5. **Media access**: controlling which frame should be sent over the link next
   - Easy for point-to-point links
     - Half versus full duplex
   - Harder for multi-access links
     - Who gets to send?
6. **Switching**: How to send frames to the eventual destination?

Framing

- A link layer function, defining which bits have which function
- *Minimal functionality*: mark the beginning and end of packets (or frames).
- Some techniques:
  - frame delimiter characters with character stuffing
  - frame delimiter codes with bit stuffing
  - synchronous transmission (e.g. SONET) out of band delimiters

Byte Stuffing

- Mark end of frame with special character
  - BISYNC uses “ETX”
  - What happens when the user sends this character?
    - Use escape character when controls appear in data
  - Very common on serial lines; old technique
  - View frame as a collection of bytes

Byte Counting

- An alternative is to include a count of number of bytes
  - Next to the start of frame
  - E.g. DDCMP
  - Corruptions of count field may cause receiver to receive incorrectly
  - Include an error-check to help receiver realize this

Bit Stuffing

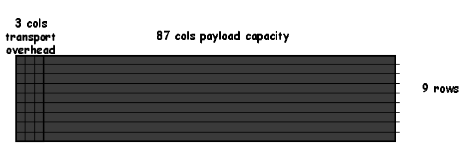
- Treat frames as a sequence of bits
- Mark frames with special bit sequence
  - Example, HDLC: 01111110 is a special sequence or "flag"
    - Used at the beginning and end of frame
  - But, must ensure data containing this sequence can be transmitted
    - Flag can cross byte boundaries
  - transmitter inserts a 0 when this is likely to appear in the data:
    - 111111 -> 1111101
    - must stuff a zero any time five 1s appear:
  - receiver unstuffs.
- Problem with stuffing techniques: frame size depends on data
  - Frames can be of different size
  - Could lead to some inefficiencies

SONET

- SONET is the Synchronous Optical Network standard for data transport over Optical fiber.
- One of the design goals was to be backwards compatible with many older telco standards.
  - E.g. voice at 56Kbps
  - So a single infrastructure could be used for carrying a variety of info
- Beside minimal framing functionality, it provides many other functions:
  - operation, administration and maintenance (OAM) communications
  - synchronization
  - multiplexing of low rate signals
  - multiplexing for high rates

Synchronous Data Transfer

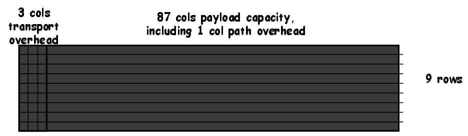
- Sender and receiver are always synchronized.
  - Frame boundaries are recognized based on the clock
  - No need to continuously look for special bit sequences
  - No stuffing or length needed
- SONET frames contain room for control and data.
  - Data frame multiplexes bytes from many users
  - Control provides information on data, management, …



STS-1

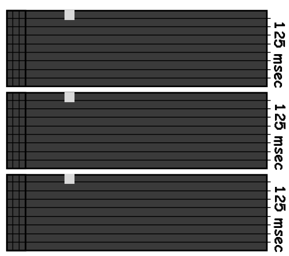
SONET Framing

- Base channel is STS-1 (Synchronous Transport System).
  - Takes 125 microsec and corresponds to 51.84 Mbps
  - 1 byte/frame corresponds to a 64 Kbs channel (voice)
    - b/w of voice is 4Khz → 8000 samples/s when digitizing
  - STS-1 → collection of 810 voice channels.
  - Also called OC-1 = optical carrier



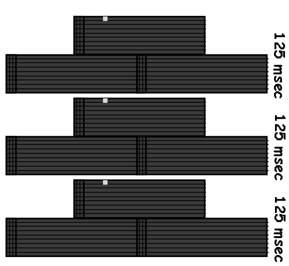
How Do We Support Lower Rates?

- 1 Byte in every consecutive frame corresponds to a 64 Kbit/second channel.
  - 1 voice call.
- Higher bandwidth channels hold more bytes per frame.
  - Multiples of 64 Kbit/second
- Channels have a "telecom" flavor.
  - Fixed bandwidth
  - Just data – no headers
  - SONET multiplexers remember how on one link should be mapped to bytes on the next link



How Do We Support Higher Rates?

- Send multiple frames in a 125 msec time slot.
- The properties of a channel using a single byte frame are maintained!
  - Constant 64 Kbit/second rate
  - Nice spacing of the byte samples

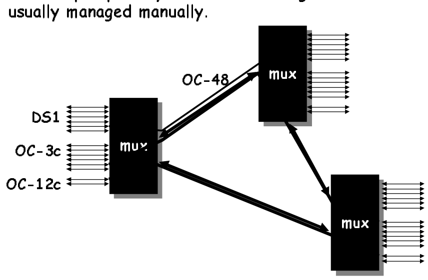


The SONET Signal Hierarchy

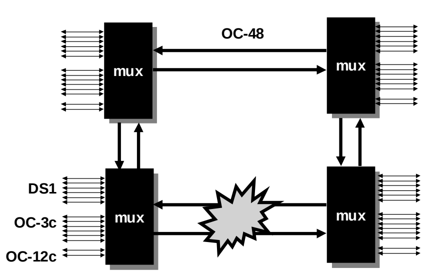
Signal Type	line rate
DS0 (POTS)	64 Kbs
DS1	1.544 Mbs
DS3	44.736 Mbs
OC-1	51.84 Mbs
OC-3	155 Mbs
OC-12	622 Mbs
STS-48	2.49 Gbs
STS-192	9.95 Gbs
STS-768	39.8 Gbs

FYI: Using SONET in Networks

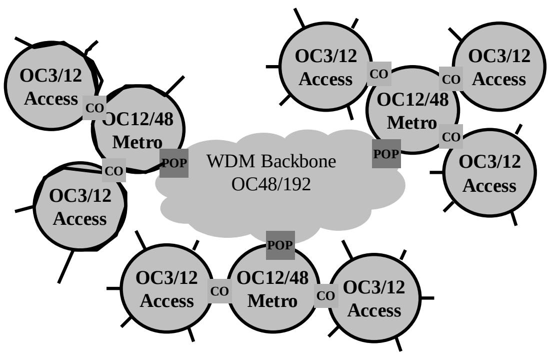
Add-drop capability allows soft configuration of networks usually managed manually.



FYI: Self-Healing SONET Rings



FYI: SONET as Physical Layer



Error Coding

- Transmission process may introduce errors into a message.
  - Single bit errors versus burst errors
- Detection: e.g. CRC
  - Requires a check that some messages are invalid
  - Hence requires extra bits
  - “redundant check bits”
- Correction
  - Forward error correction: many related code words map to the same data word
  - Detect errors and retry transmission

Parity

- Even parity
  - Append parity bit to 7 bits of data to make an even number of 1’s
  - Odd parity accordingly defined.

1010100	1
1001011	0

- 1 in 8 bits of overhead?
  - When is this a problem?
- Can detect a single error 1010101 1
- But nothing beyond that 1000010 0

2-D Parity

- Make each byte even parity
- Finally, a parity byte for all bytes of the packet
- Example: five 7-bit character packet, even parity

0110100	1
1011010	0
0010110	1
1110101	1
1001011	0
1000110	1

Effectiveness of 2-D Parity

- 1-bit errors can be detected
- Example with even parity per byte:

	0110100	1	
	1011010	0	
error bit	0000110	1	odd number of 1's
	1110101	1	
	1001011	0	
	1000110	1	

Effectiveness of 2-D Parity

- 2-bit errors can also be detected
- Example:

	0110100	1	
	1011010	0	
error bits	0000111	1	even number of 1's - Ok
	1110101	1	
	1001011	0	
	1000110	1	
	odd number of 1's		

- What about 3-bit errors? >3-bit errors?
  - See HW 1 problem

Cyclic Redundancy Codes (CRC)

- Commonly used codes that have good error detection properties
  - Can catch many error combinations with a small number or redundant bits
- Based on division of polynomials
  - Errors can be viewed as adding terms to the polynomial
  - Should be unlikely that the division will still work
- Can be implemented very efficiently in hardware
- Examples:
  - CRC-32: Ethernet
  - CRC-8, CRC-10, CRC-32: ATM

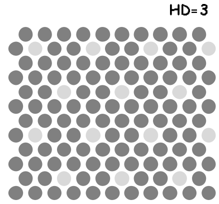
An Aside: Hamming Distance

- *Hamming distance* of two bit strings = number of bit positions in which they differ.
- 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.

1	0	1	1	0
1	1	0	1	0

HD=2

HD=3

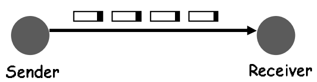


Link Flow Control and Error Control

- Dealing with receiver overflow: flow control.
- Dealing with packet loss and corruption: error control.
- Actually these issues are relevant at many layers.
  - Link layer: sender and receiver attached to the same "wire"
  - End-to-end: transmission control protocol (TCP) - sender and receiver are the end points of a connection
- How can we implement flow control?
  - "You may send" (windows, stop-and-wait, etc.)
  - "Please shut up" (source quench, 802.3x pause frames, etc.)

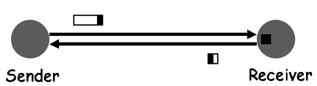
Flow Control: A Naïve Protocol

- Sender simply sends to the receiver whenever it has packets.
- Potential problem: sender can outrun the receiver.
  - Receiver too slow, small buffer overflow, ..
- Not always a problem: receiver might be fast enough.



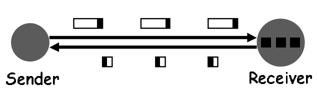
Adding Flow Control

- Stop and wait flow control: sender waits to send the next packet until the previous packet has been acknowledged by the receiver.
  - Receiver can pace the sender
- Drawbacks: adds overheads, slowdown for long links.



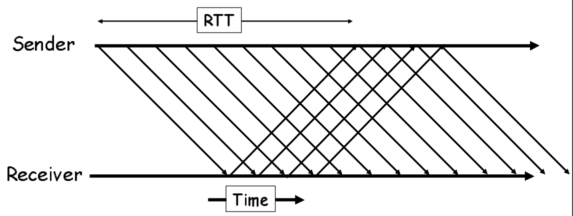
Window Flow Control

- Stop and wait flow control results in poor throughput for long-delay paths: packet size/ roundtrip-time.
- Solution: receiver provides sender with a window that it can fill with packets.
  - The window is backed up by buffer space on receiver
  - Receiver acknowledges the a packet every time a packet is consumed and a buffer is freed



Window Limitations

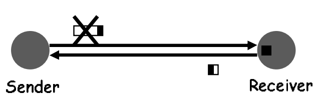
Window Size = 4pkts



Throughput = $\frac{\text{Window Size}}{\text{Roundtrip Time}}$

Error Control: Stop and Wait Case

- Packets can get lost, corrupted, or duplicated.
- Duplicate packet: use sequence numbers.
- Lost packet: time outs and acknowledgements.
  - Positive versus negative acknowledgements
  - Sender side versus receiver side timeouts
- Window based flow control: more aggressive use of sequence numbers (see transport lectures).



What is Used in Practice?

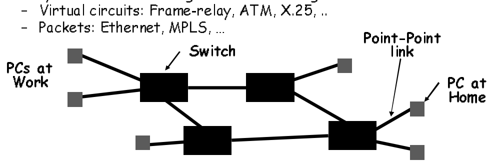
- No flow or error control.
  - E.g. regular Ethernet, just uses CRC for error detection
- Flow control only.
  - E.g. Gigabit Ethernet
- Flow and error control.
  - E.g. X.25 (older connection-based service at 64 Kbs that guarantees reliable in order delivery of data)

Switching and Media Access Control

- How do we transfer packets between two hosts connected to the a switched network?
- Switches connected by point-to-point links -- store-and-forward.
  - Multiplexing and forwarding
  - Used in WAN, LAN, and for home connections
  - Conceptually similar to “routing”
    - But at the datalink layer instead of the network layer
  - Today
- Multiple access networks -- contention based.
  - Multiple hosts are sharing the same transmission medium
  - Used in LANs and wireless
  - Need to control access to the medium
  - Next lecture

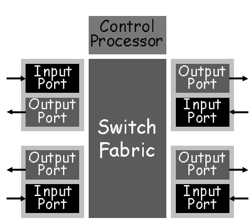
A Switch-based Network

- Switches are connected by “point-to-point” links.
  - In contrast, how are hosts connected?
- Packets are forwarded hop-by-hop by the switches towards the destination.
  - Each packet gets entire capacity of link for a short duration
    - Mux-ing
  - Forwarding is based on the *address*
- Many datalink technologies use switching.
  - Virtual circuits: Frame-relay, ATM, X.25, ..
  - Packets: Ethernet, MPLS, …



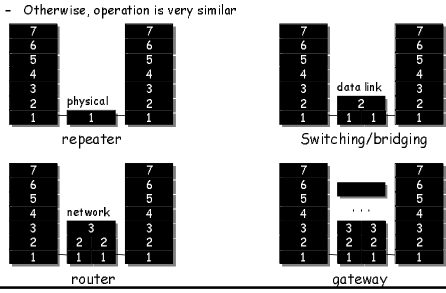
Switch Architecture Overview

- Takes in packets in one interface and has to forward them to an output interface based on the address.
  - A big intersection
  - Same idea for bridges, switches, routers: address look up differs
- Control processor manages the switch and executes higher level protocols.
  - E.g. routing, management, ..
- The switch fabric directs the traffic to the right output port.
- The input and output ports deal with transmission and reception of packets.
- More when we talk of IP routers

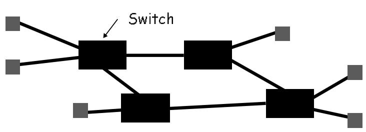


Internetworking Options

- "Switching" also happens at the network layer.
  - Layer 3: Internet protocol
  - In this case, address is an IP address
  - IP over SONET, IP over ATM, ..
  - Otherwise, operation is very similar



Packet Forwarding: Address Lookup Overview



Address	Next Hop	Info
B31123B12508	3	13
3B913C3C2137	3	-
A21023C90590	0	-
128.2.15.3	1	(2,34)

- Address from header.
  - Absolute address (e.g. Ethernet)
  - (IP address for routers)
  - (VC identifier, e.g. ATM))
- Next hop: output port for packet.
- Info: priority, VC id, ..
- Table is filled in by routing protocol.

Next Lecture

- Ethernet
- MAC
- LAN architectures