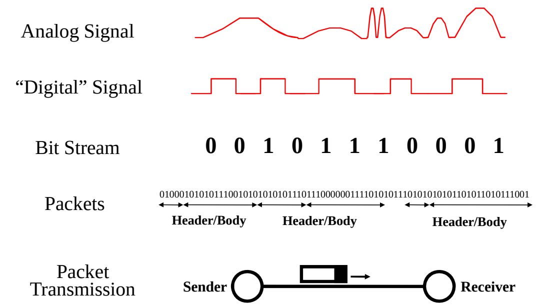
Lecture 6
Datalink – Framing, Switching

Peter Steenkiste
Departments of Computer Science and
Electrical and Computer Engineering
Carnegie Mellon University

15-441 Networking, Spring 2008
http://www.cs.cmu.edu/~dga/15-441/S08

From Signals to Packets



Datalink Functions

- Framing: encapsulating a network layer datagram into a bit stream.
  - » Add header, mark and detect frame boundaries, …
- 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?
- Error control: error detection and correction to deal with bit errors.
  - » May also include other reliability support, e.g. retransmission
- Flow control: avoid that the sender outruns the receiver.

Datalink Lectures

- Framing and error coding.
- Datalink architectures.
- Switch-based networks.
  - » Packet forwarding
  - » Flow and error control
- Taking turn protocols.

- Contention-based networks: basic Ethernet.
- Ethernet bridging and switching.
- Connectivity to the home.

- Circuit-based communication

Framing

- A link layer function, defining which bits have which function.
- Minimal functionality: mark the beginning and end of packets (or frames).
- Some techniques:
  - » out of band delimiters (e.g. FDDI 4B/5B control symbols)
  - » frame delimiter characters with character stuffing
  - » frame delimiter codes with bit stuffing
  - » synchronous transmission (e.g. SONET)

Character and Bit Stuffing

- Mark frames with special character.
  - » What happens when the user sends this character?
  - » Use escape character when controls appear in data:
    *abc*def -> *abc*def
  - » Very common on serial lines, in editors, etc.
- Mark frames with special bit sequence
  - » must ensure data containing this sequence can be transmitted
  - » example: suppose 11111111 is a special sequence.
  - » transmitter inserts a 0 when this appears in the data:
  - » 11111111 -> 111111101
  - » must stuff a zero any time seven 1s appear:
  - » 11111110 -> 111111100
  - » receiver unstuffs.

Example: Ethernet Framing

- Preamble is 7 bytes of 10101010 (5 MHz square wave) followed by one byte of 10101011
- Allows receivers to recognize start of transmission after idle channel

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.
- Beside minimal framing functionality, it provides many other functions:
  - » operation, administration and maintenance (OAM) communications
  - » synchronization
  - » multiplexing of lower rate signals
  - » multiplexing for higher rates

Standardization History

- Process was started by divestiture in 1984.
  - » Multiple telephone companies building their own infrastructure
- SONET concepts originally developed by Bellcore.
- First standardized by ANSI T1X1 group for the US.
- Later picked up by CCITT and developed its own version.
- SONET/SDH standards approved in 1988.

A Word about Data Rates

- Bandwidth of telephone channel is under 4KHz, so when digitizing:

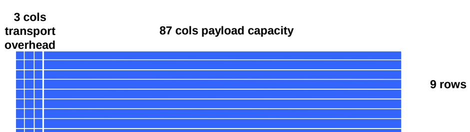
  8000 samples/sec * 8 bits = 64Kbits/second
- Common data rates supported by telcos in North America:
  - » Modem: rate improved over the years
  - » T1/DS1: 24 voice channels plus 1 bit per sample

    (24 * 8 + 1) * 8000 = 1.544 Mbits/second
  - » T3/DS3: 28 T1 channels:

    7 * 4 * 1.544 = 44.736 Mbits/second

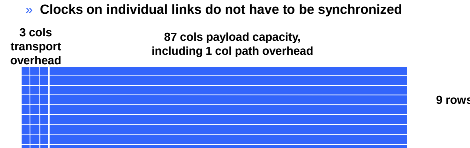
Synchronous Data Transfer

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



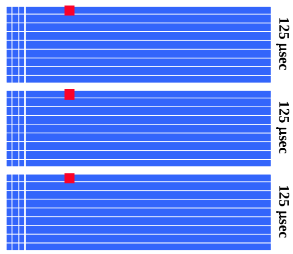
SONET Framing

- Base channel is STS-1 (Synchronous Transport System).
  - » Takes 125 μsec and corresponds to 51.84 Mbps
  - » 1 byte/frame corresponds to a 64 Kbs channel (voice)
  - » Transmitted on an OC-1 optical carrier (fiber link)
- Standard ways of supporting slower and faster channels.
  - » Support both old standards and future (higher) data rates
- Actual payload frame “floats” in the synchronous frame.
  - » Clocks on individual links do not have to be synchronized



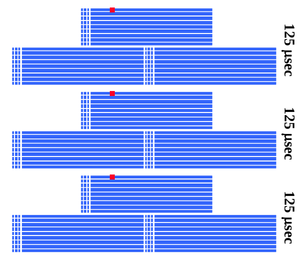
How Do We Support Lower Rates?

- 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 bytes on one link should be mapped to bytes on the next link
    - Byte 33 on incoming link 1 is byte 97 on outgoing link 7



How Do We Support Higher Rates?

- Send multiple frames in a 125 μsec time slot.
- The properties of a channel using a single byte/ST-1 frame are maintained!
  - Constant 64 Kbit/second rate
  - Nice spacing of the byte samples
- Rates typically go up by a factor of 4.
- Two ways of doing interleaving.
  - Frame interleaving
  - Column interleaving
    - concatenated version, i.e. OC-3c



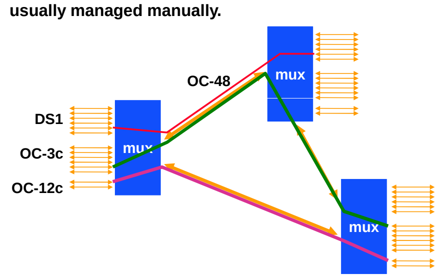
The SONET Signal Hierarchy

STS-1 carries one DS-3 plus overhead

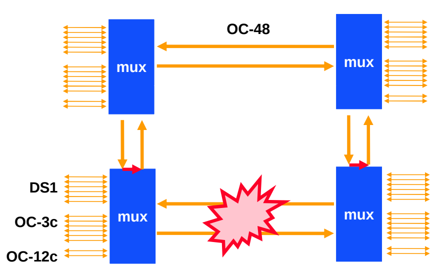
Signal Type	line rate	# of DS0
DS0 (POTS)	64 Kbs	1
DS1	1.544 Mbs	24
DS3	44.736 Mbs	672
OC-1	51.84 Mbs	672
OC-3	155 Mbs	2,016
OC-12	622 Mbs	8,064
STS-48	2.49 Gbs	32,256
STS-192	9.95 Gbs	129,024
STS-768	39.8 Gbs	516,096

Using SONET in Networks

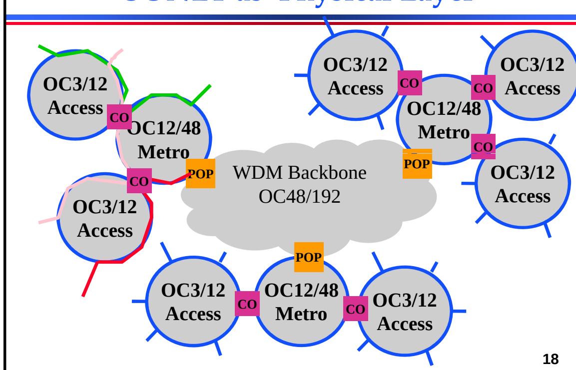
Add-drop capability allows soft configuration of networks, usually managed manually.



Self-Healing SONET Rings



SONET as Physical Layer

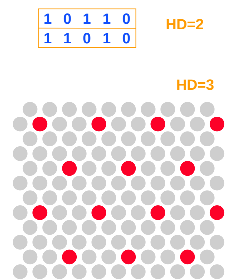


Error Coding

- Transmission process may introduce errors into a message.
  - » Single bit errors versus burst errors
- Detection:
  - » Requires a convention that some messages are invalid
  - » Hence requires extra bits
  - » An (n,k) code has codewords of n bits with k data bits and r = (n-k) redundant check bits
- Correction
  - » Forward error correction: many related code words map to the same data word
  - » Detect errors and retry transmission

Basic Concept: 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.



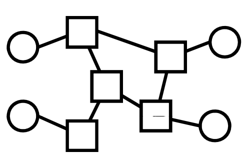
Examples

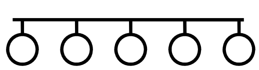
- A (4,3) parity code has D=2:
  0001 0010 0100 0111 1000 1011 1101 1110
  (last bit is binary sum of previous 3, inverted - “odd parity”)
- A (7,4) code with D=3 (2ED, 1EC):
  0000000 0001101 0010111 0011010
  0100011 0101110 0110100 0111001
  1000110 1001011 1010001 1011100
  1100101 1101000 1110010 1111111
- 1001111 corrects to 1001011
- Note the inherent risk in correction; consider a 2-bit error resulting in 1001011 -> 1111011.
- There are formulas to calculate the number of extra bits that are needed for a certain D.

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

Datalink Architectures





- Packet forwarding.
- Error and flow control.

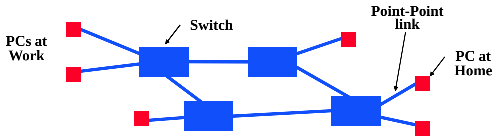
- Media access control.
- Scalability.

Media Access Control

- How do we transfer packets between two hosts connected to the same network?
- Switches connected by point-to-point links -- store-and-forward.
  - » 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
  - » Mostly Thursday lecture

A Switch-based Network

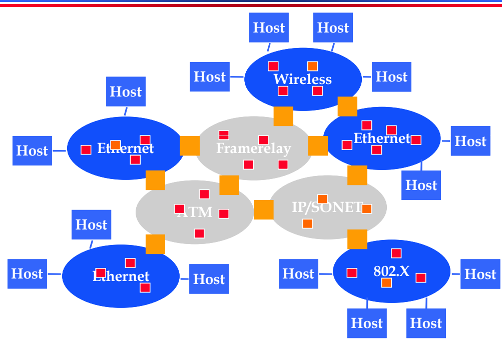
- Switches are connected by point-point links.
- Packets are forwarded hop-by-hop by the switches towards the destination.
  - » Forwarding is based on the address
- How does a switch work?
- How do nodes exchange packets over a link?
- How is the destination addressed?



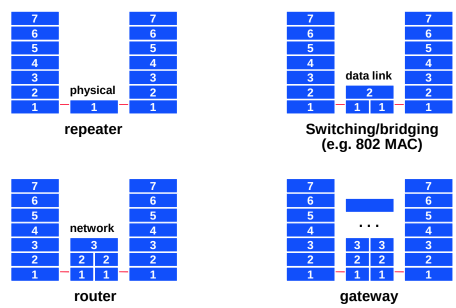
Switching Introduction

- Idea: forward units of data based on address in header.
- Many data-link technologies use switching.
  - » Virtual circuits: Frame Relay, ATM, X.25, ..
  - » Packets: Ethernet, MPLS, ...
- “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
- Switching is different from SONET mux/demux.
  - » SONET channels statically configured - no addresses

An Inter-network

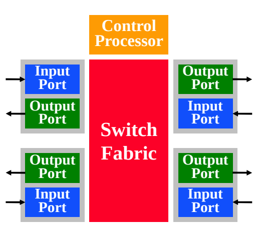


Internetworking Options

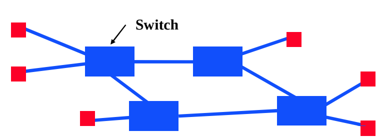


Switch Architecture

- 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.



Packet Forwarding: Address Lookup



Address	Next Hop	Info
B31123812508	3	13
38913C3C2137	3	-
A21023C90590	0	-
128.2.15.3	1	(2,34)

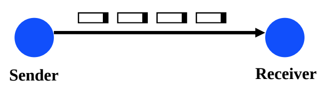
- Address from header.
  - » Absolute address (e.g. Ethernet)
  - » (IP address for routers)
  - » (VC identifier, e.g. ATM))
- Next hop: output port for packet.
- Info: priority, VC id, ..
- Table is filled in by routing protocol.

Link Flow Control and Error Control

- Naïve protocol.
- Dealing with receiver overflow: flow control.
- Dealing with packet loss and corruption: error control.
- Meta-comment: 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.)
  - Where are each of these appropriate?

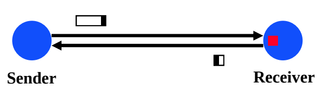
A Naïve Protocol

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



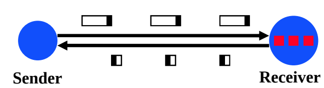
Adding Flow Control

- 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 receiver
- Drawbacks: adds overheads, slowdown for long links.

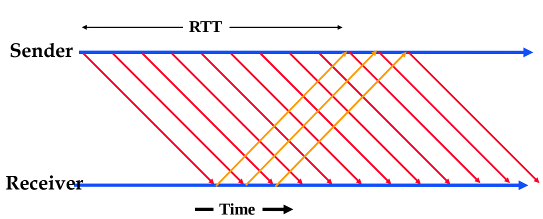


Window Flow Control

- 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



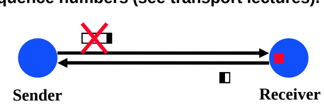
Bandwidth-Delay Product



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

Dealing with Errors Stop and Wait Case

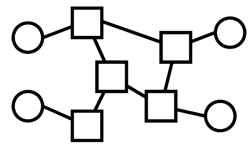
- Packets can get lost, corrupted, or duplicated.
  - Error detection or correction turns corrupted packet in lost or correct packet
- Duplicate packet: use sequence numbers.
- Lost packet: time outs and acknowledgements.
  - Positive versus negative acknowledgements
  - Sender side versus receiver side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

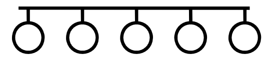


What is Used in Practice?

- 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)

Datalink Layer Architectures

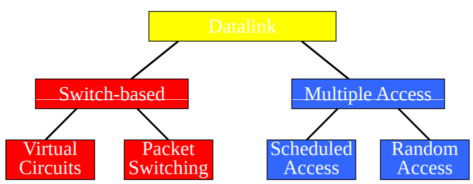




- Packet forwarding.
- Error and flow control.

- Media access control.
- Scalability.

Datalink Classification

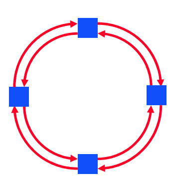


Multiple Access Protocols

- Prevent two or more nodes from transmitting at the same time over a broadcast channel.
  - » If they do, we have a collision, and receivers will not be able to interpret the signal
- Several classes of multiple access protocols.
  - » Partitioning the channel, e.g. frequency-division or time division multiplexing
    - – With fixed partitioning of bandwidth –
    - – Not flexible; inefficient for bursty traffic
  - » Taking turns, e.g. token-based, reservation-based protocols, polling based
  - » Contention based protocols, e.g. Aloha, Ethernet
    - – Next lecture

Fiber Distributed Data Interface (FDDI)

- One token holder may send, with a time limit
  - » Provides known upper bound on delay.
- Optical version of 802.5 token ring, but multiple packets may travel in train: token released at end of frame
- 100 Mbps, 100km
- Optional dual ring for fault tolerance
- Concerns:
  - » Token overhead
  - » Latency
  - » Single point of failure



Other "Taking Turn" Protocols

- Central entity polls stations, inviting them to transmit
  - » Simple design – no conflicts
  - » Not very efficient – overhead of polling operation
  - » Example: the "Point Control Function" mode for 802.11
- Stations reserve a slot for transmission.
  - » For example, break up the transmission time in contention-based and reservation based slots
    - – Contention based slots can be used for short messages or to reserve time slots
    - – Communication in reservation based slots only allowed after a reservation is made
  - » Issues: fairness, efficiency

MAC Protocols - Discussion

- Channel partitioning MAC protocols:
  - » Share channel efficiently at high load
  - » Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!
- "Taking turns" protocols
  - » More flexible bandwidth allocation, but
  - » Protocol can introduce unnecessary overhead and access delay at low load
- Random access MAC protocols (next lecture)
  - » Efficient at low load: single node can fully utilize channel
  - » High load: collision overhead