CS 61C: Great Ideas in Computer Architecture (Machine Structures)

Lecture 24
More I/O: DMA, Disks, Networking

Instructors: John Wawrzynek & Vladimir Stojanovic

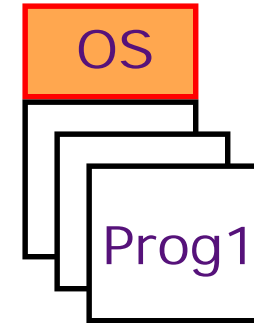http://inst.eecs.berkeley.edu/~cs61c/

Virtual Memory Review

Modern Virtual Memory Systems

Illusion of a large, private, uniform store
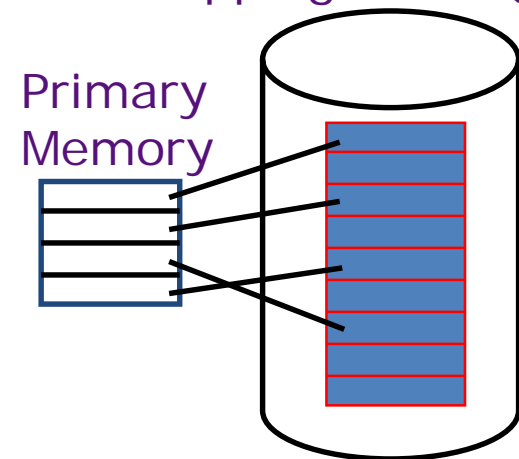
Protection & Privacy
    * Many processes, each with their own private address space and one or more shared address spaces

OS

Prog1

Demand Paging
    * Many processes share DRAM.
    * Provides ability to run programs with large address space.   Pages that aren't yet allocated or pages that don't fit swap to secondary storage.
    * Hides differences in machine configurations
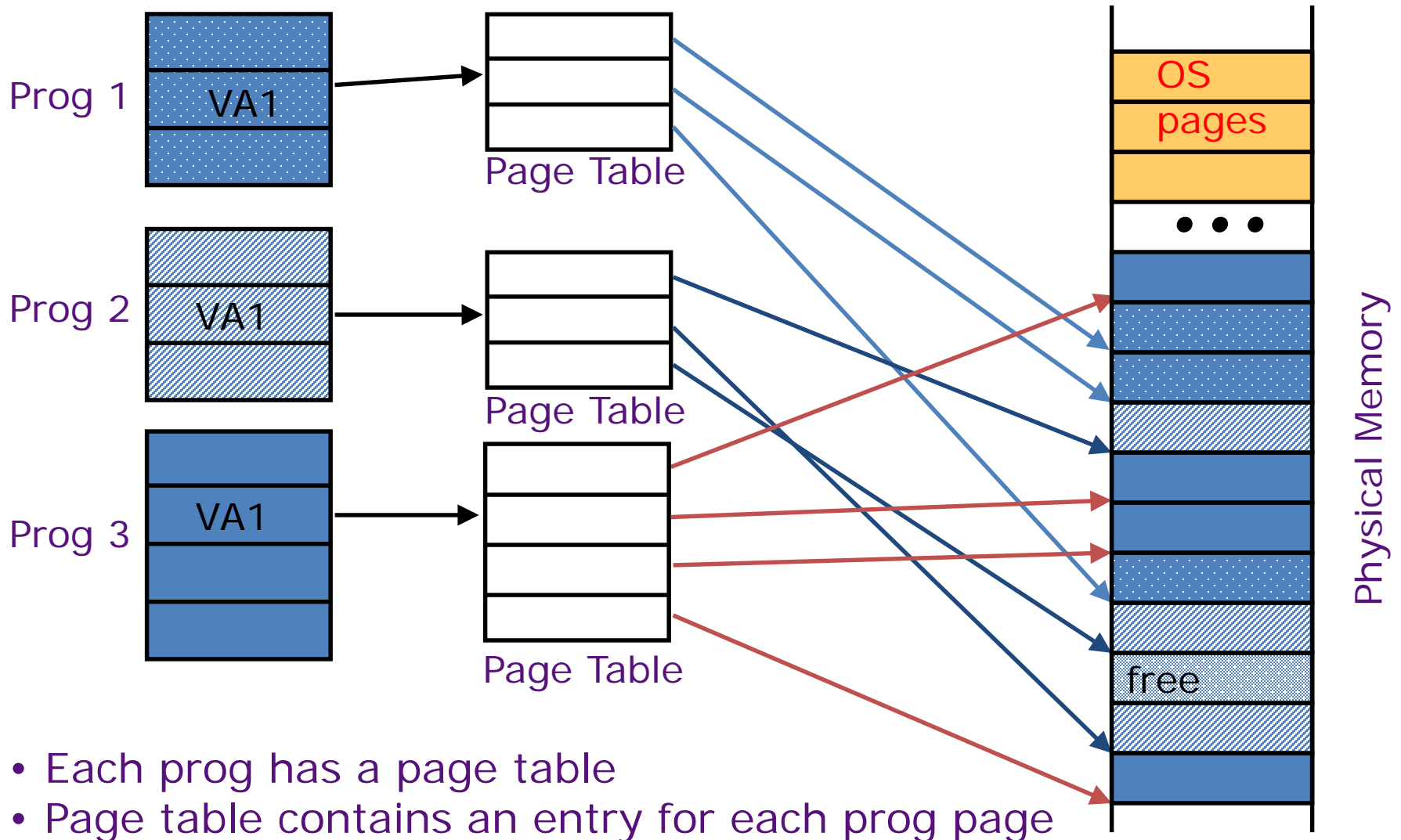
Swapping Store (Disk)

Primary Memory

The price is address translation on each memory reference

VA → mapping TLB → PA

Private (Virtual) Address Space per Program

- Each prog has a page table
- Page table contains an entry for each prog page
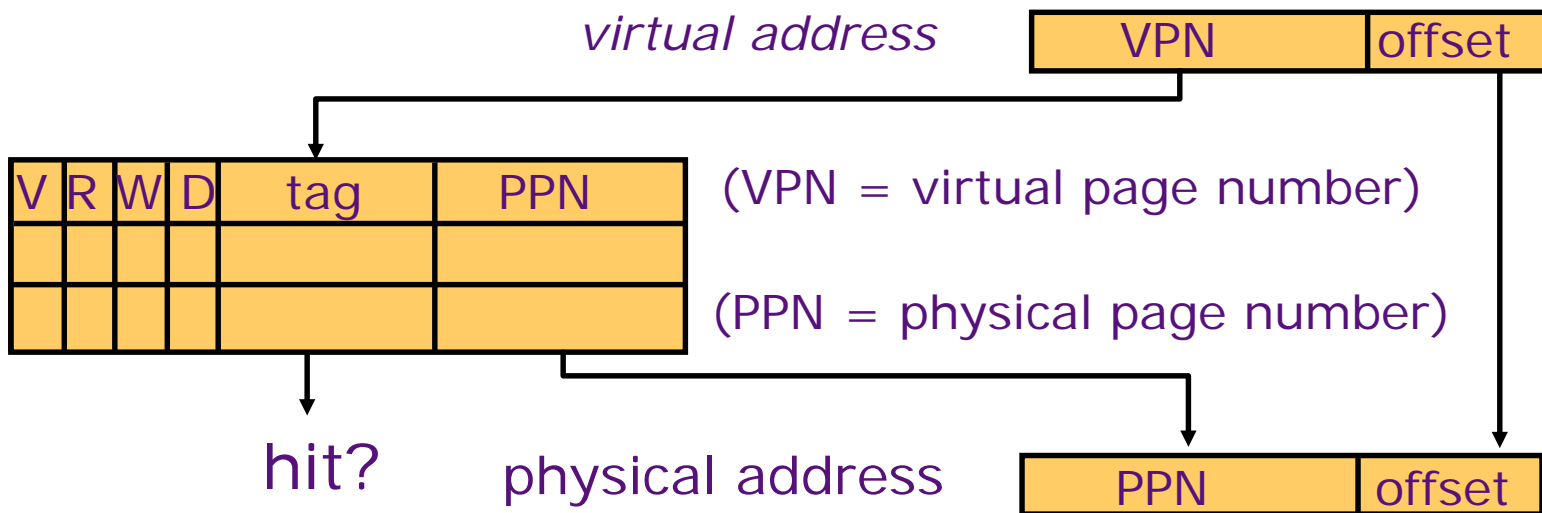
Translation Lookaside Buffers (TLB)

Address translation is very expensive!
In a two-level page table, each reference becomes several memory accesses
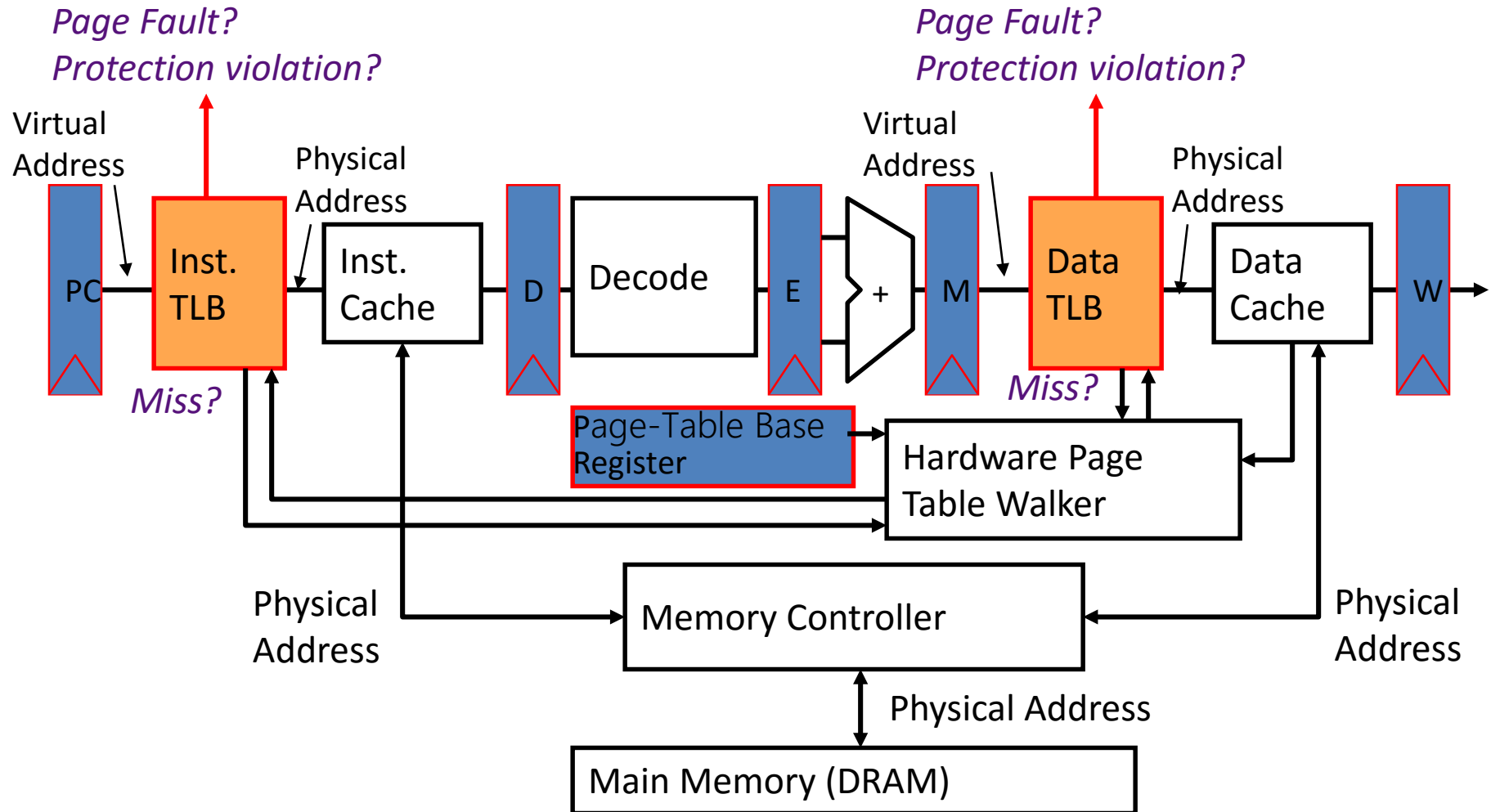
Solution: *Cache translations in TLB*

TLB hit $\Rightarrow$ *Single-Cycle Translation*
TLB miss $\Rightarrow$ *Page-Table Walk to refill*

virtual address

VPN	offset

V	R	W	D	tag	PPN

(VPN = virtual page number)

(PPN = physical page number)

hit?   physical address

PPN	offset

Page-Based Virtual-Memory Machine
(Hardware Page-Table Walk)

- Page tables held in untranslated physical memory

Clicker Question from Last Week

Let's try to extrapolate from caches… Which one is false?

A. # offset bits in V.A. = log2(page size)

B. # offset bits in P.A. = log2(page size)

C. # VPN bits in V.A. = log2(# of physical pages)

D. # PPN bits in P.A. = log2(# of physical pages)

E. A single-level page table contains a PTE for every possible VPN in a particular Virtual Address Space

Review: I/O

- "Memory mapped I/O": Device control/data registers mapped to CPU address space
- CPU synchronizes with I/O device:
  - Polling
  - Interrupts
- "Programmed I/O":
  - CPU execs lw/sw instructions for all data movement to/from devices
  - CPU spends time doing 2 things:
    1. Getting data from device to main memory
    2. Using data to compute

Working with real devices

- "Memory mapped I/O": Device control/data registers mapped to CPU address space
- CPU synchronizes with I/O device:
  - Polling
  - Interrupts
- ~~"Programmed I/O":~~ **DMA**
  - ~~CPU execs lw/sw instructions for all data movement to/from devices~~
  - CPU spends time doing ~~2 things~~:
    1. ~~Getting data from device to main memory~~
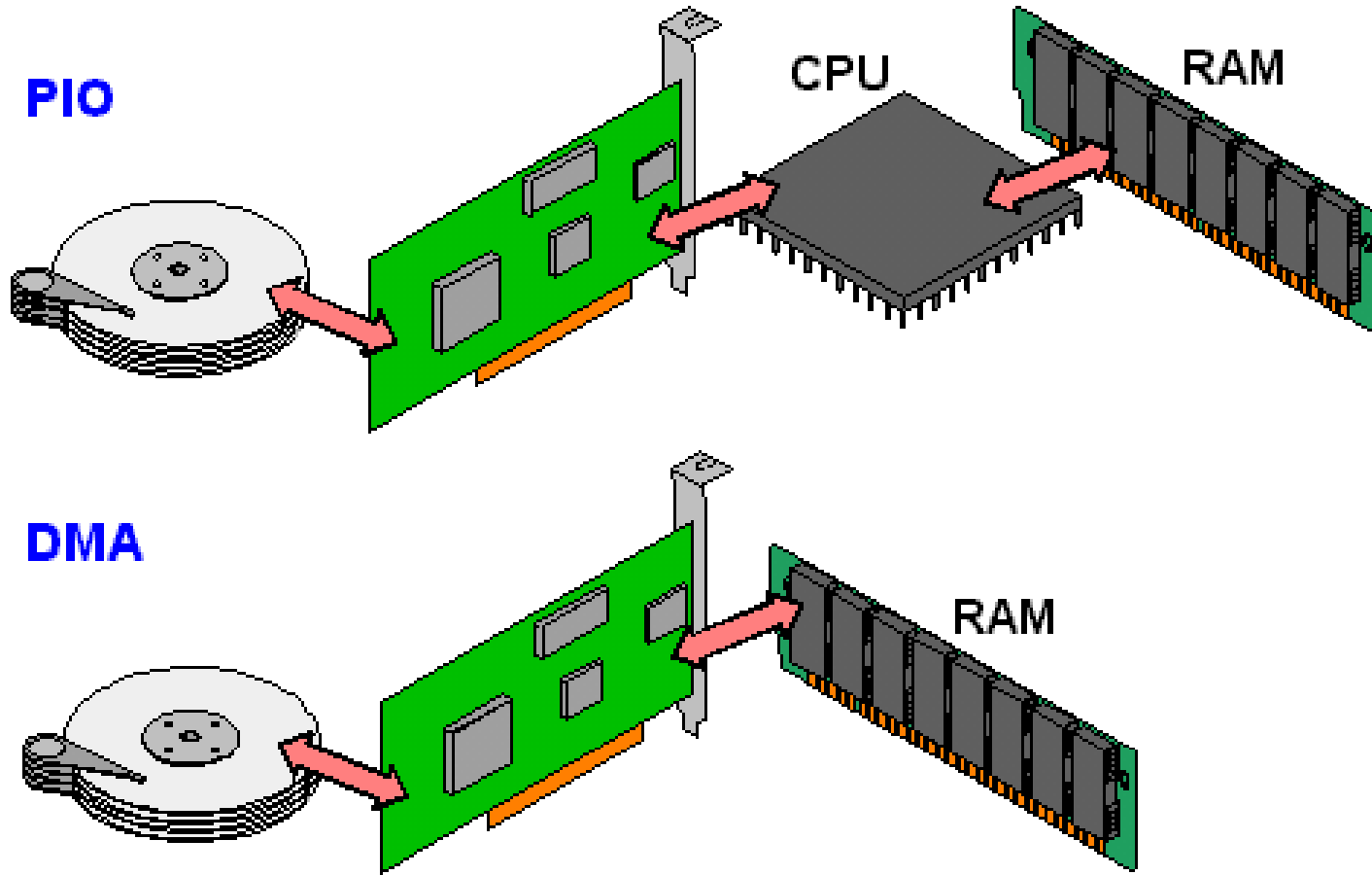   2. Using data to compute

Agenda

- **Direct Memory Access (DMA)**
- Disks
- Networking

What's wrong with Programmed I/O?

- Not ideal because …

  1. CPU has to execute all transfers, could be doing other work

  2. Device speeds don't align well with CPU speeds

  3. Energy cost of using beefy general-purpose CPU where simpler hardware would suffice

- Until now CPU has sole control of main memory

PIO vs. DMA

Direct Memory Access (DMA)

- Allows I/O devices to directly read/write main memory

- New Hardware: the <u>DMA Engine</u>

- DMA engine contains registers written by CPU:
  - Memory address to place data
  - # of bytes
  - I/O device #, direction of transfer
  - unit of transfer, amount to transfer per burst
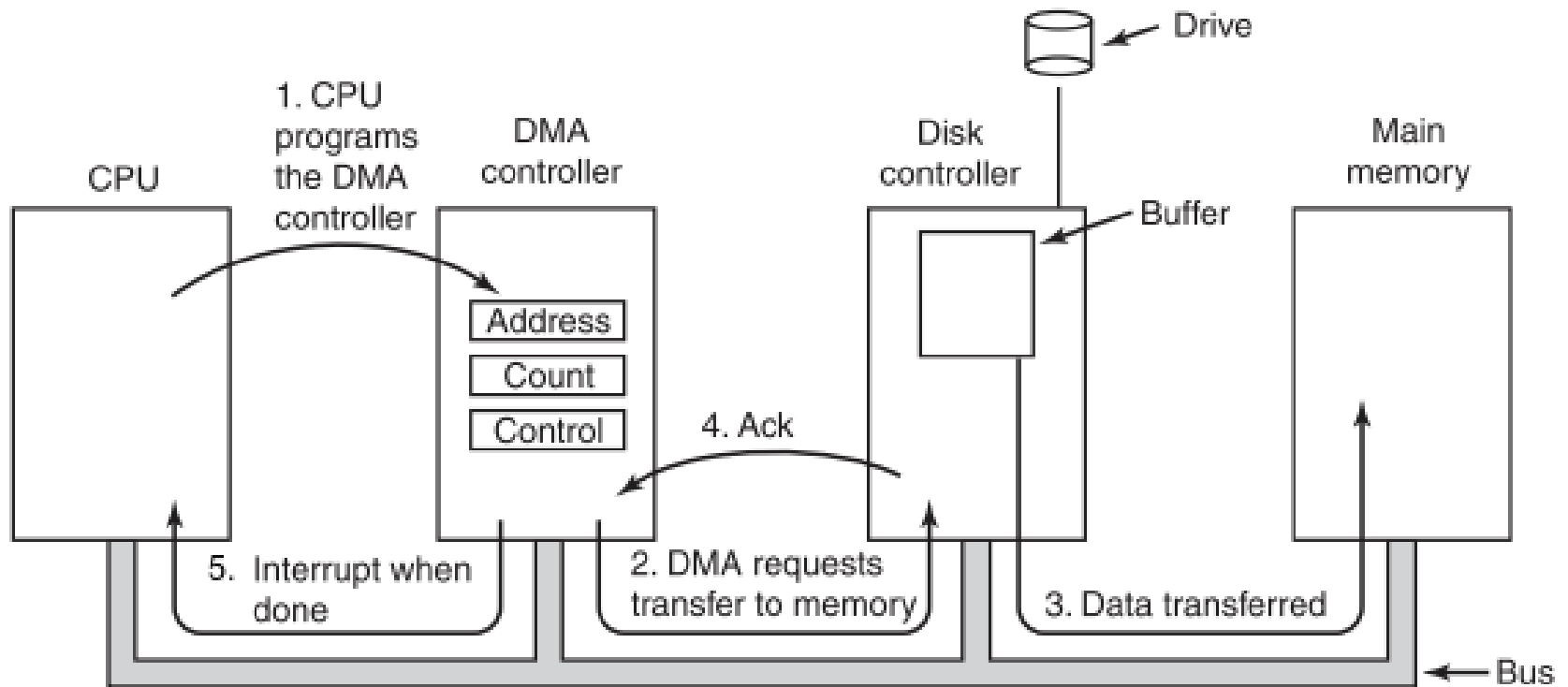
Operation of a DMA Transfer

Figure 5-4. Operation of a DMA transfer.

[From Section 5.1.4 Direct Memory Access in *Modern Operating Systems* by Andrew S. Tanenbaum, Herbert Bos, 2014]

DMA: Incoming Data

1. Receive interrupt from device
2. CPU takes interrupt, begins transfer
    – Instructs DMA engine/device to place data @ certain address
3. Device/DMA engine handle the transfer
    – CPU is free to execute other things
4. Upon completion, Device/DMA engine interrupt the CPU again
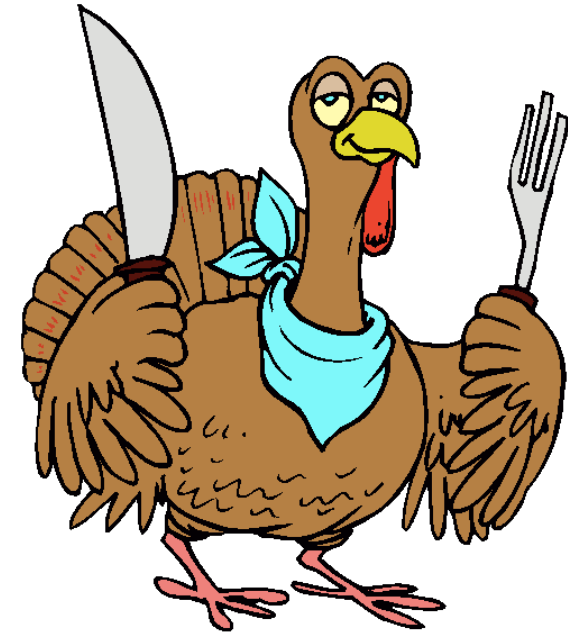
DMA: Outgoing Data

1. CPU decides to initiate transfer, confirms that external device is ready

2. CPU begins transfer
   – Instructs DMA engine/device that data is available @ certain address

3. Device/DMA engine handle the transfer
   – CPU is free to execute other things

4. Device/DMA engine interrupt the CPU again to signal completion

DMA: Some new problems

- Where in the memory hierarchy do we plug in the DMA engine? Two extremes:
  - Between L1 and CPU:
    - Pro: Free coherency
    - Con: Trash the CPU's working set with transferred data
  - Between Last-level cache and main memory:
    - Pro: Don't mess with caches
    - Con: Need to explicitly manage coherency

Administrivia

- Upcoming Lecture Schedule
  - 11/24: I/O: DMA, Disks, Networking (today)
  - 11/26: Thanksgiving Holiday
  - 12/01: Dependability: Parity, ECC, RAID
    - Last day of new material
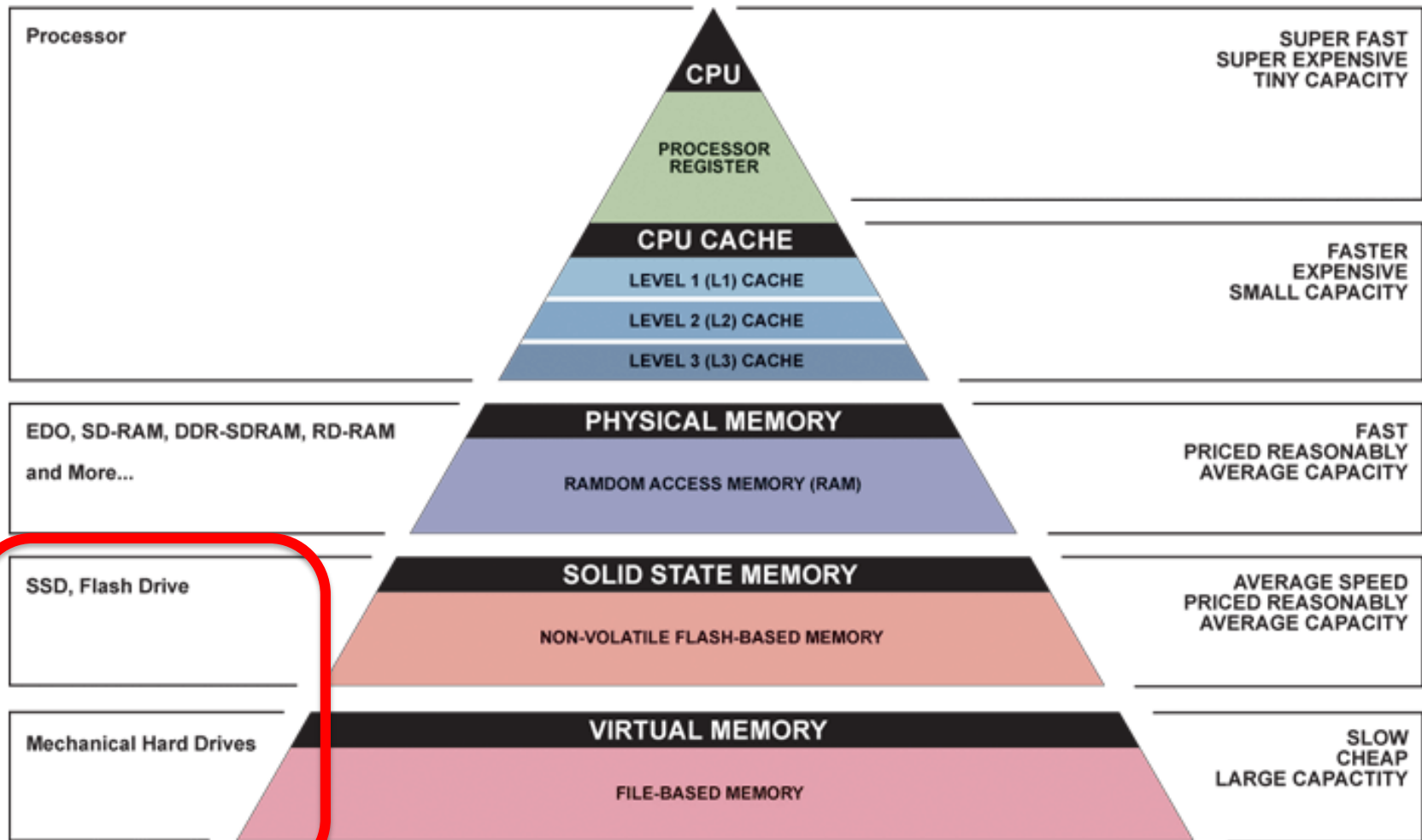 - 12/03: Summary, What's Next? (+ HKN reviews)

Administrivia

- Last HW (5) Virtual Memory
  - Due 12/06 (Sunday after end of classes)
- Last project (5): Spark
  - 5-1: posted, due today!
  - 5-2: due 12/06 (Sunday after end of classes)
- <u>Final Exam</u> is Friday (12/18)
  - ***<u>7-10PM</u>***, RSF Field House
  - More info soon on review sessions, etc.

Agenda

- Direct Memory Access (DMA)
- **Disks**
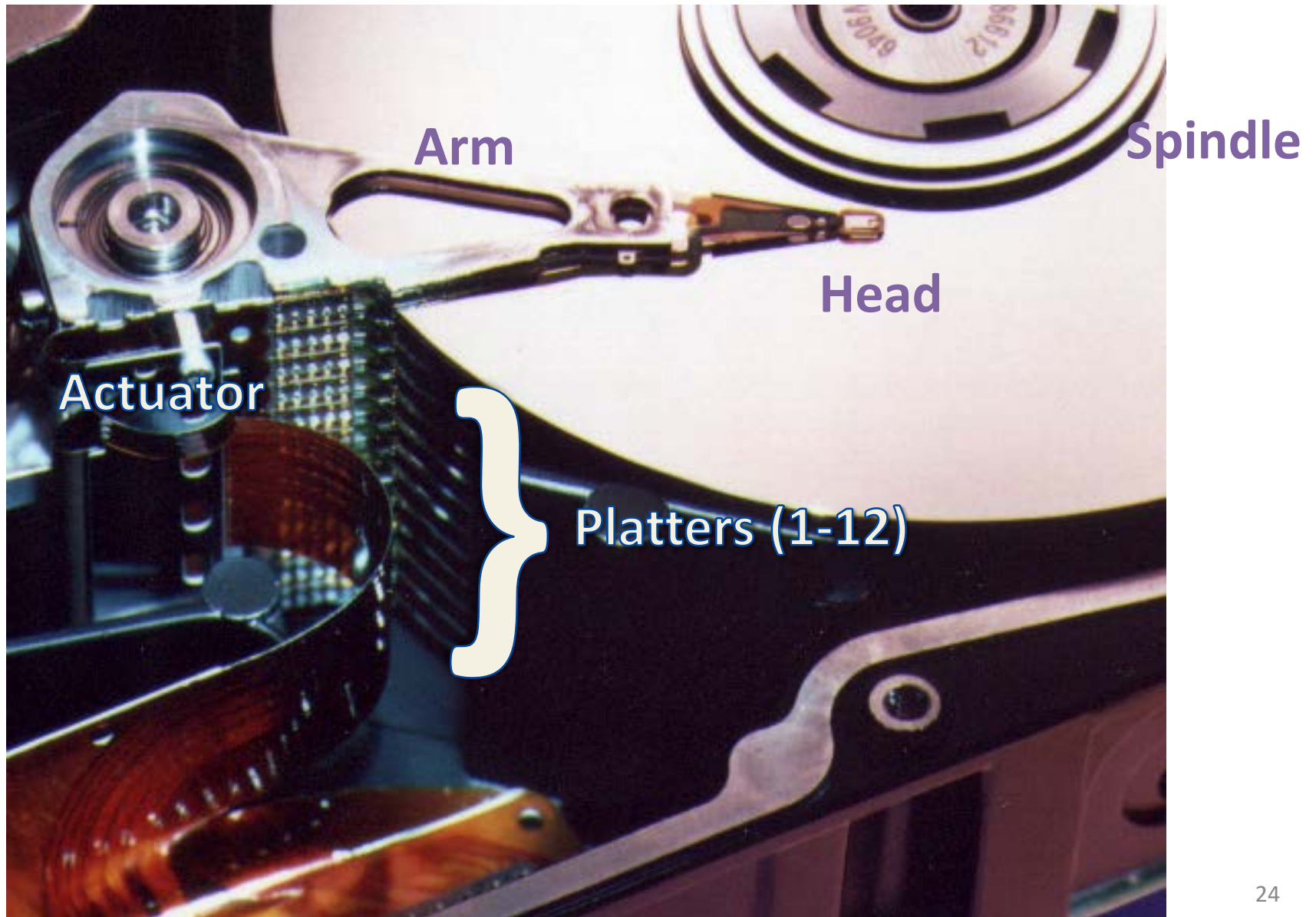- Networking

Computer Memory Hierarchy

Processor

| SUPER FAST SUPER EXPENSIVE TINY CAPACITY |

CPU

PROCESSOR REGISTER

| FASTER EXPENSIVE SMALL CAPACITY |

CPU CACHE

LEVEL 1 (L1) CACHE

LEVEL 2 (L2) CACHE

LEVEL 3 (L3) CACHE

EDO, SD-RAM, DDR-SDRAM, RD-RAM

and More...

PHYSICAL MEMORY

RAMDOM ACCESS MEMORY (RAM)

| FAST PRICED REASONABLY AVERAGE CAPACITY |

SSD, Flash Drive

SOLID STATE MEMORY

NON-VOLATILE FLASH-BASED MEMORY

| AVERAGE SPEED PRICED REASONABLY AVERAGE CAPACITY |

Mechanical Hard Drives

VIRTUAL MEMORY

FILE-BASED MEMORY

| SLOW CHEAP LARGE CAPACITY |

▲ Simplified Computer Memory Hierarchy
Illustration: Ryan J. Leng
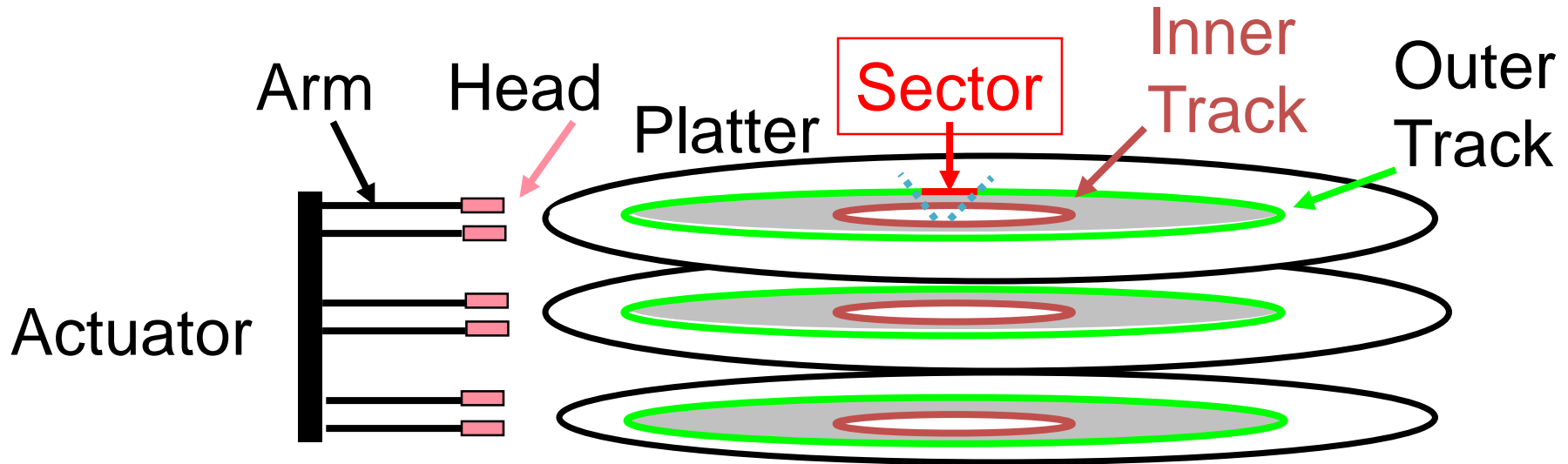
Today

22

Magnetic Disk – common I/O device

- A kind of computer memory
  - Information stored by magnetizing ferrite material on surface of rotating disk
    - similar to tape recorder except digital rather than analog data

- A type of non-volatile storage
  - retains its value without applying power to disk.

- Two Types of Magnetic Disk
  1. Hard Disk Drives (HDD) – faster, more dense, non-removable.
  2. Floppy disks – slower, less dense, removable (now replaced by USB "flash drive").

- Purpose in computer systems (Hard Drive):
  1. Working file system + long-term backup for files
  2. Secondary "backing store" for main-memory. Large, inexpensive, slow level in the memory hierarchy (virtual memory)

Photo of Disk Head, Arm, Actuator

Arm

Spindle

Head

Actuator

Platters (1-12)
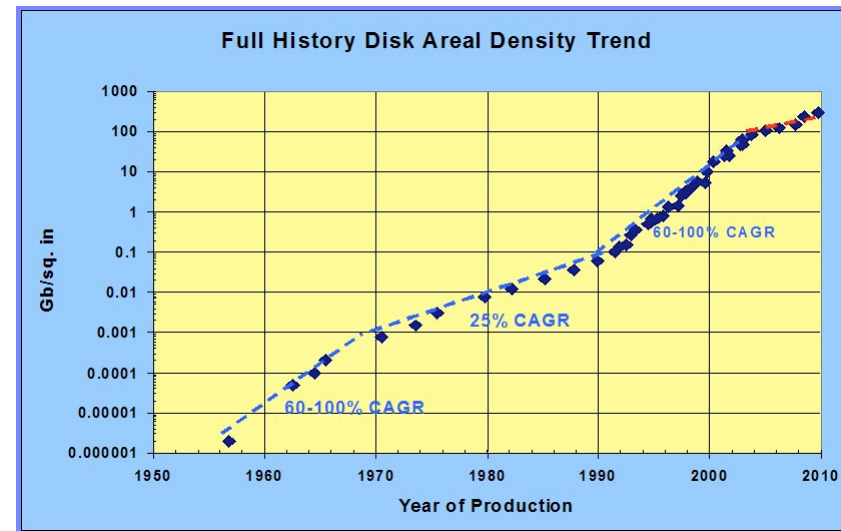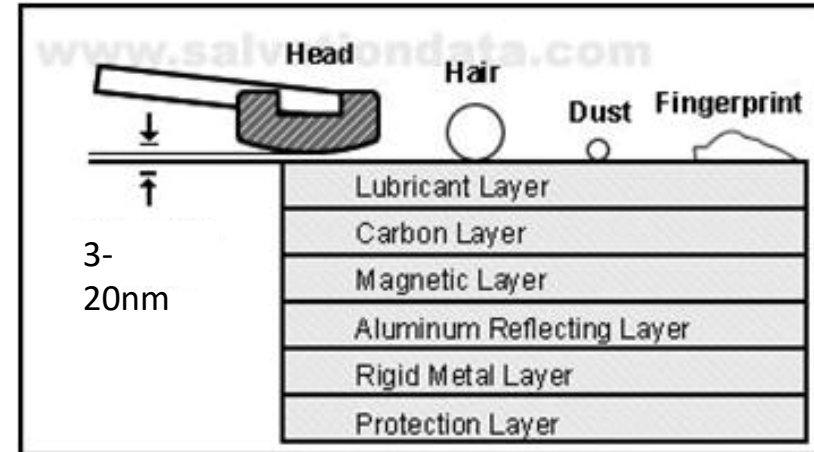
Disk Device Terminology

- Several platters, with information recorded magnetically on both surfaces (usually)
- **Bits recorded in tracks, which in turn divided into sectors (e.g., 512 Bytes)**
- **Actuator moves head (end of arm) over track ("seek"), wait for sector rotate under head, then read or write**
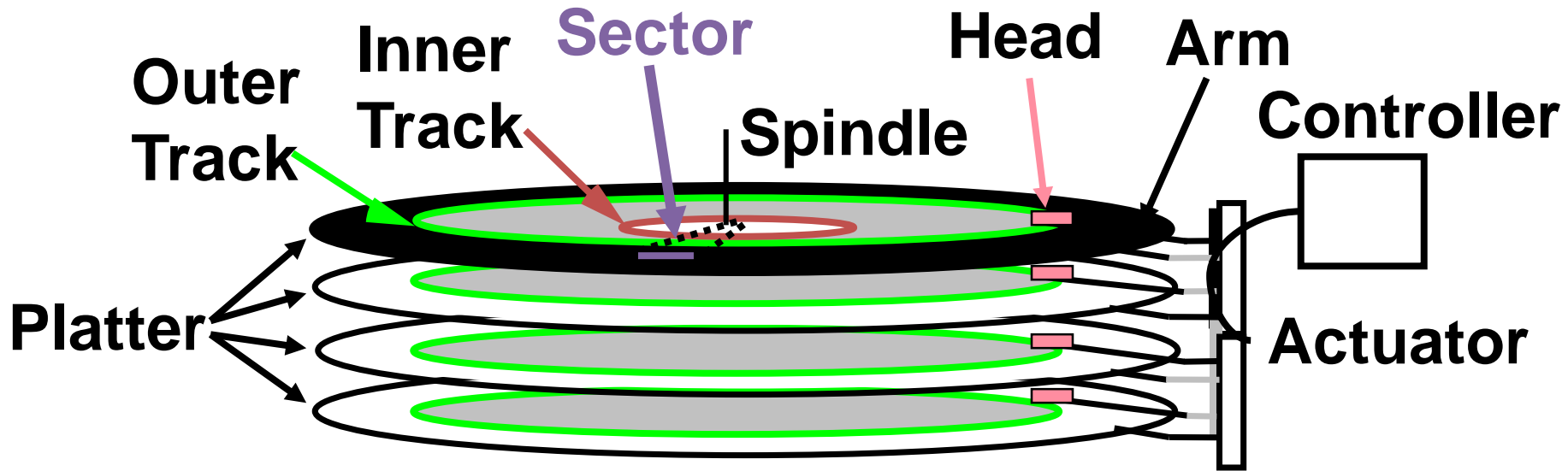
Video of hard disk in action

Hard Drives are Sealed. Why?

- The closer the head to the disk, the smaller the "spot size" and thus the denser the recording.
  - Measured in Gbit/in^2
  - ~900 Gbit/in^2 is state of the art
  - Started out at 2 Kbit/in^2
  - ~450,000,000x improvement in ~60 years
- Disks are sealed to keep the dust out.
  - Heads are designed to "fly" at around 3-20nm above the surface of the disk.
  - 99.999% of the head/arm weight is supported by the air bearing force (air cushion) developed between the disk and the head.

www.salvationdata.com

Head   Hair   Dust   Fingerprint

3-20nm

| Lubricant Layer |
| Carbon Layer |
| Magnetic Layer |
| Aluminum Reflecting Layer |
| Rigid Metal Layer |
| Protection Layer |

Full History Disk Areal Density Trend

Gb/sq. in

60-100% CAGR

25% CAGR

60-100% CAGR

Year of Production

26

Disk Device Performance (1/2)

- **Disk Access Time = Seek Time + Rotation Time + Transfer Time + Controller Overhead**

  – Seek Time = time to position the head assembly at the proper cylinder

  – Rotation Time = time for the disk to rotate to the point where the first sectors of the block to access reach the head

  – Transfer Time = time taken by the sectors of the block and any gaps between them to rotate past the head

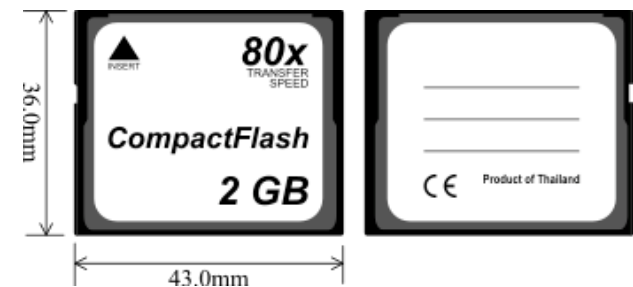# Disk Device Performance (2/2)

- Average values to plug into the formula:
- Rotation Time: Average distance of sector from head?
  - 1/2 time of a rotation
    - 7200 Revolutions Per Minute $\Rightarrow$ 120 Rev/sec
    - 1 revolution = 1/120 sec $\Rightarrow$ 8.33 milliseconds
    - 1/2 rotation (revolution) $\Rightarrow$ 4.17 ms
- Seek time: Average no. tracks to move arm?
  - Number of tracks/3 (see CS186 for the math)
  - Then, seek time = number of tracks moved × time to move across one track
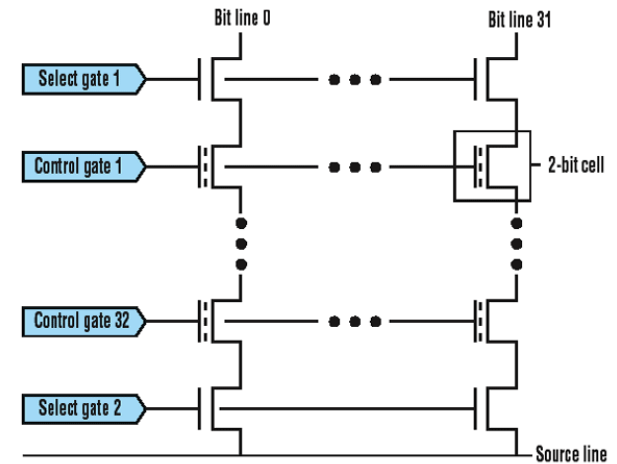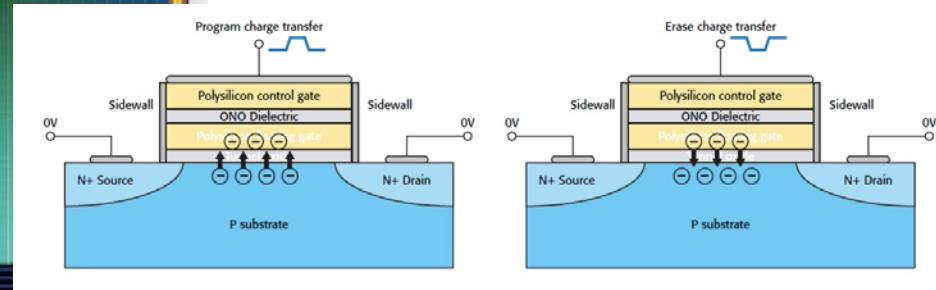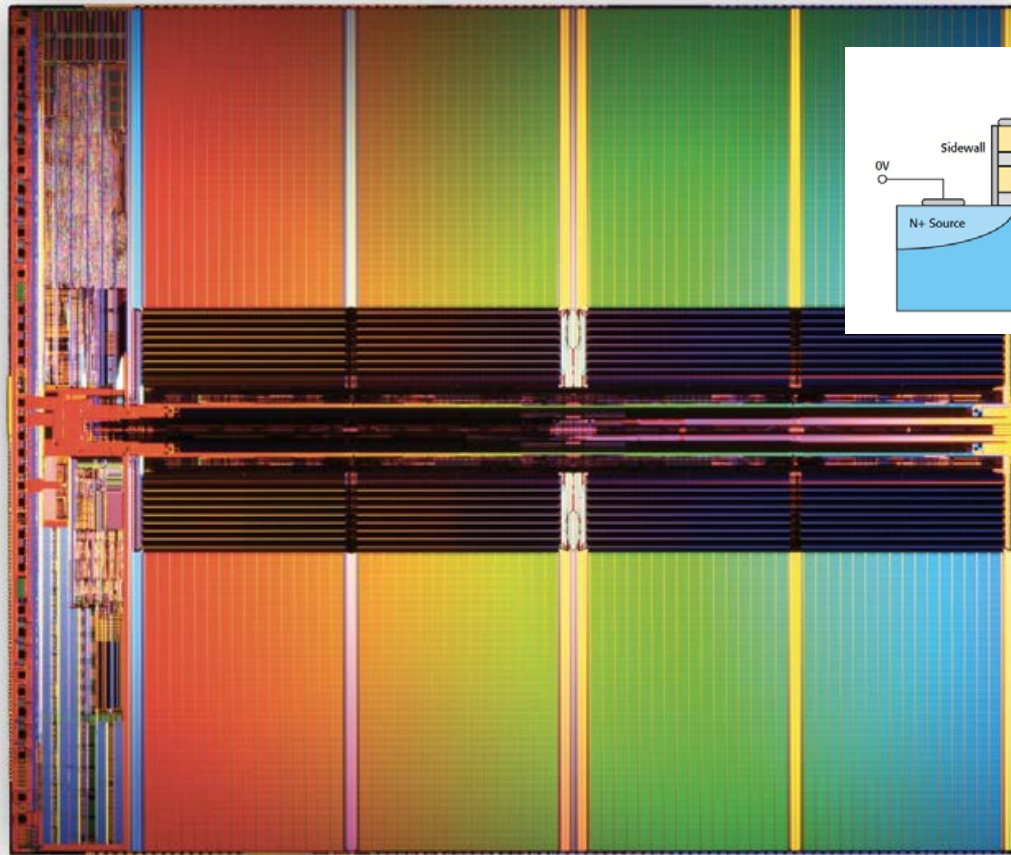
But wait!

- Performance estimates are different in practice:

- Many disks have on-disk caches, which are completely hidden from the outside world

  – Previous formula completely replaced with on-disk cache access time

Where does Flash memory come in?

- ~10 years ago: Microdrives and Flash memory (e.g., CompactFlash) went head-to-head
  - Both non-volatile (retains contents without power supply)
  - Flash benefits: lower power, no crashes (no moving parts, need to spin μdrives up/down)
  - Disk cost = fixed cost of motor + arm mechanics, but actual magnetic media cost very low
  - Flash cost = most cost/bit of flash chips
  - Over time, cost/bit of flash came down, became cost competitive

Flash Memory / SSD Technology

2. Micron's triple-level cell (TLC) flash memory stores 3 bits of data in each transistor.

In the basic functional block used in multilevel NAND flash memories, 32 rows of bit lines and 32 control-gate lines form a building block that's repeated many times to form the memory array. The select gate lines are used with the control gate lines to control access to the array.

- NMOS transistor with an additional conductor between gate and source/drain which "traps" electrons. The presence/absence is a 1 or 0

- Memory cells can withstand a limited number of program-erase cycles. Controllers use a technique called *wear leveling* to distribute writes as evenly as possible across all the flash blocks in the SSD.

What did Apple put in its iPods?

Toshiba flash
2 GB

Samsung flash
16 GB

Toshiba 1.8-inch HDD
80, 120, 160 GB
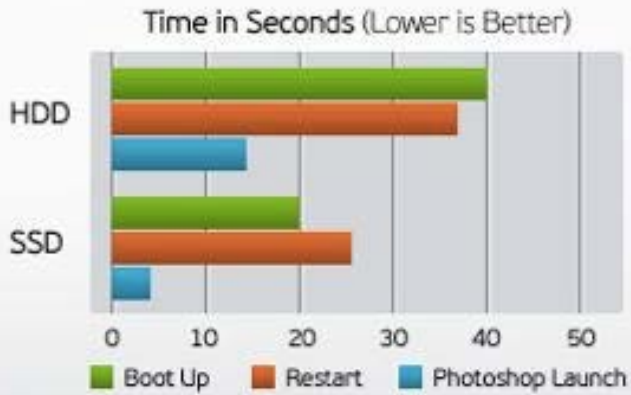
Toshiba flash
32, 64 GB

shuffle   nano   classic   touch

Flash Memory in Smart Phones

Flash Memory in Laptops – Solid State Drive (SSD)

Time in Seconds (Lower is Better)

HDD

SSD

0   10   20   30   40   50

■ Boot Up   ■ Restart   ■ Photoshop Launch

tom's hardware

Hard Drive vs. SSD vs. Hybrid HDD

Cost per bit

Capacity

Endurance P/E Cycle

Power

Reliability

Sequential Read Write

Random R/W

—— SSD   —— HDD   —— HHDD

capacities up to 512GB

iClicker Question

- We have the following disk:
  - 15000 Cylinders, 1 ms to cross 1000 Cylinders
  - 15000 RPM = 4 ms per rotation
  - Want to copy 1 MB, transfer rate of 1000 MB/s
  - 1 ms controller processing time

- What is the access time using our model?

Disk Access Time = Seek Time + Rotation Time + Transfer Time + Controller Processing Time

A	B	C	D	E
10.5 ms	9 ms	8.5 ms	11.4 ms	12 ms

iClicker Question

- ## We have the following disk:
  - 15000 Cylinders, 1 ms to cross 1000 Cylinders
  - 15000 RPM = 4 ms per rotation
  - Want to copy 1 MB, transfer rate of 1000 MB/s
  - 1 ms controller processing time

- ## What is the access time?

Seek = # cylinders/3 * time = 15000/3 * 1ms/1000 cylinders = 5ms

Rotation = time for ½ rotation = 4 ms / 2 = 2 ms

Transfer = Size / transfer rate = 1 MB / (1000 MB/s) = 1 ms

Controller = 1 ms

Total = 5 + 2 + 1 + 1 = 9 ms

Break

Agenda

- Direct Memory Access (DMA)
- Disks
- **Networking**

Networks: Talking to the Outside World

- Originally sharing I/O devices between computers
  - E.g., printers
- Then communicating between computers
  - E.g., file transfer protocol
- Then communicating between people
  - E.g., e-mail
- Then communicating between networks of computers
  - E.g., file sharing, www, …

The Internet (1962)

- History
  - 1963: JCR Licklider, while at DoD's ARPA, writes a memo describing desire to connect the computers at various research universities: Stanford, Berkeley, UCLA, …
  - 1969 : ARPA deploys 4 "nodes" @ UCLA, SRI, Utah, & UCSB
  - 1973 Robert Kahn & Vint Cerf invent <u>TCP</u>, now part of the <u>Internet Protocol Suite</u>

- Internet growth rates
  - Exponential since start!

"Lick"

Vint Cerf

"Revolutions like this don't come along very often"

www.greatachievements.org/?id=3736
en.wikipedia.org/wiki/Internet_Protocol_Suite

11/24/2015
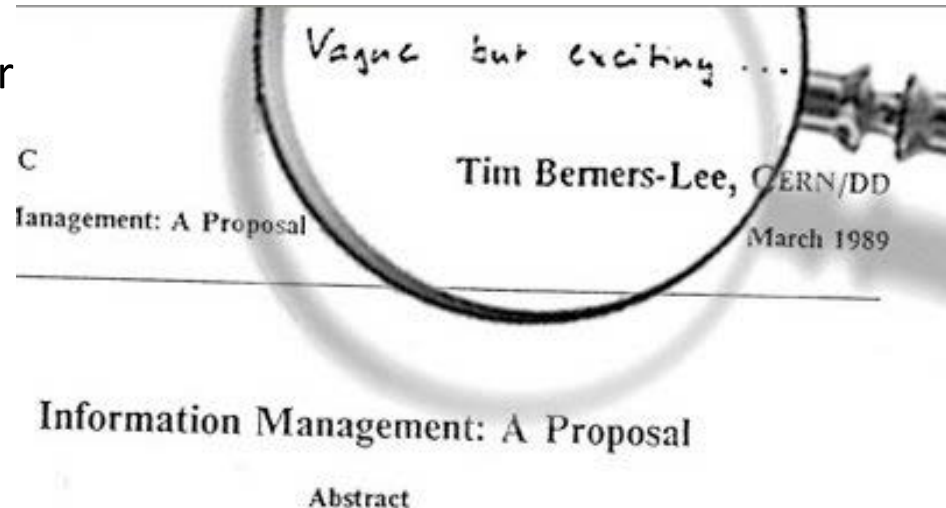
40

The World Wide Web (1989)

- "System of interlinked hypertext documents on the Internet"
- History
  - 1945: Vannevar Bush describes hypertext system called "memex" in article
  - 1989: Sir Tim Berners-Lee proposed and implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client and server using the internet.
  - ~2000 Dot-com entrepreneurs rushed in, 2001 bubble burst
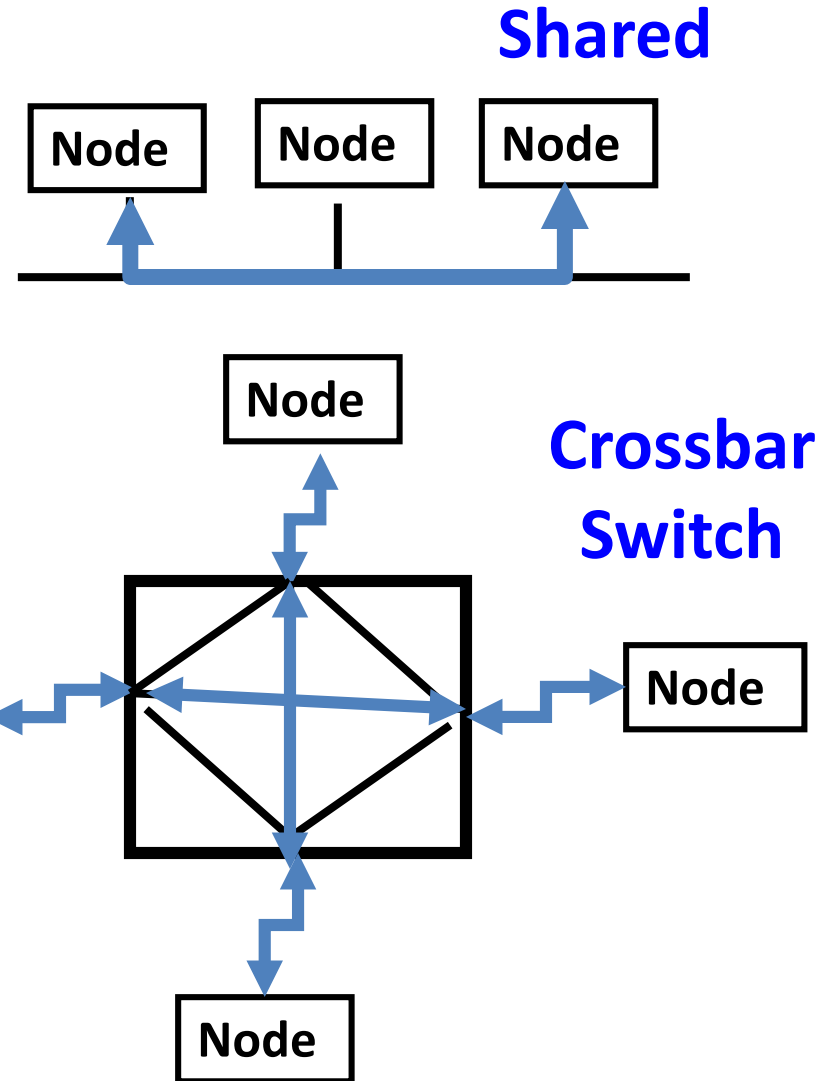- Today : Access anywhere!

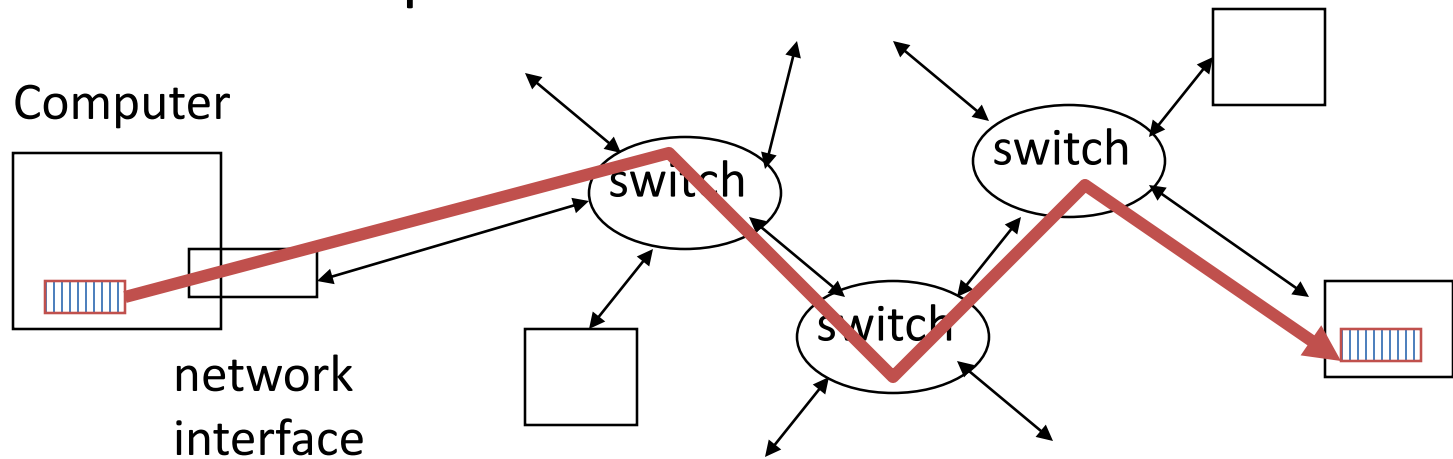Tim Berners-Lee

World's First web server in 1990

Vague but exciting ...

C

Tim Berners-Lee, CERN/DD

Management: A Proposal

March 1989

Information Management: A Proposal

Abstract

Shared vs. Switch-Based Networks

- Shared vs. Switched:
  - Shared: 1 at a time (CSMA/CD)
  - Switched: pairs ("point-to-point" connections) communicate at same time
- Aggregate bandwidth (BW) in switched network is many times that of shared:
  - point-to-point faster since no arbitration, simpler interface

Shared

| Node | Node | Node |

Crossbar Switch

What makes networks work?

- **links** connecting **switches and/or routers** to each other and to computers or devices

- ability to name the components and to route packets of information - messages - from a source to a destination

- Layering, redundancy, protocols, and encapsulation as means of <u>abstraction</u> (61C big idea)

CISCO SYSTEMS

Software Protocol to Send and Receive

- SW Send steps
    - 1: Application copies data to OS buffer
    - 2: OS calculates checksum, starts timer
    - 3: OS sends data to network interface HW and says start
- SW Receive steps
    - 3: OS copies data from network interface HW to OS buffer
    - 2: OS calculates checksum, if OK, send ACK; if not, delete message (sender resends when timer expires)
    - 1: If OK, OS copies data to user address space, & signals application to continue

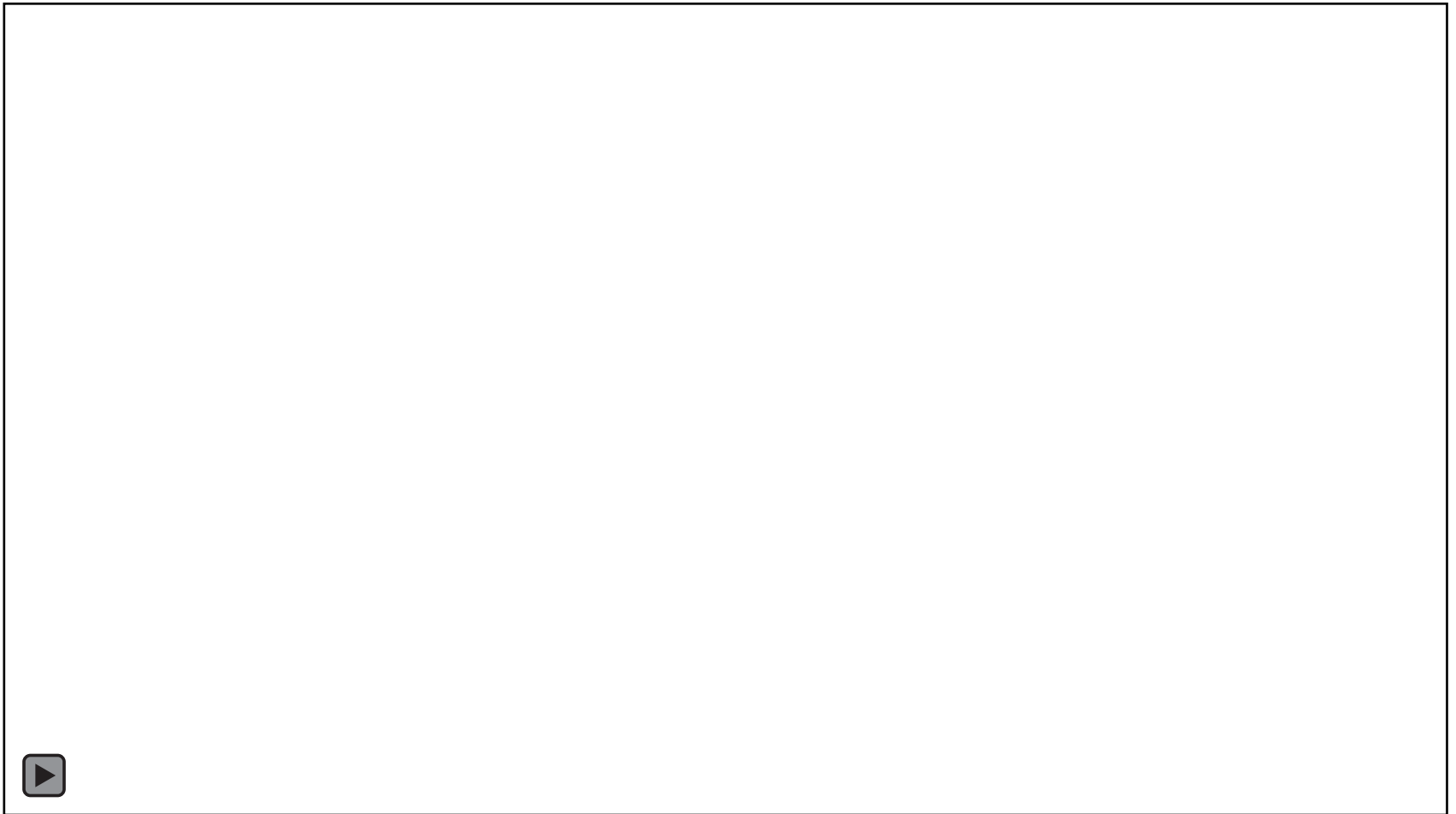**Dest** **Src**   **Checksu**

Net ID	Net ID	Len	ACK INFO	CMD/ Address /Data	

Header   **Payload**   **Trailer**

Networks are like Ogres

Protocols for Networks of Networks?

What does it take to send packets across the globe?

- Bits on wire or air
- Packets on wire or air
- Delivery packets within a single physical network
- Deliver packets across multiple networks
- Ensure the destination received the data
- Create data at the sender and make use of the data at the receiver

Protocol for Networks of Networks?

Lots to do and at multiple levels!

Use <u>abstraction</u> to cope with <u>complexity of communication</u>

- Networks are like ~~ogres~~ onions
  - Hierarchy of layers:
    - Application (chat client, game, etc.)
    - Transport (TCP, UDP)
    - Network (IP)
    - Data Link Layer (ethernet)
    - Physical Link (copper, wireless, etc.)

Protocol Family Concept

- *Protocol*: packet structure and control commands to manage communication
- *Protocol families (suites)*: a set of cooperating protocols that implement the network stack

- Key to protocol families is that communication occurs logically at the same level of the protocol, called peer-to-peer...

  ...but is implemented via services at the next lower level

- Encapsulation: carry higher level information within lower level "envelope"

Inspiration…

- CEO A writes letter to CEO B
  - Folds letter and hands it to assistant

Dear John,

- Assistant:
  - Puts letter in envelope with CEO B's full name
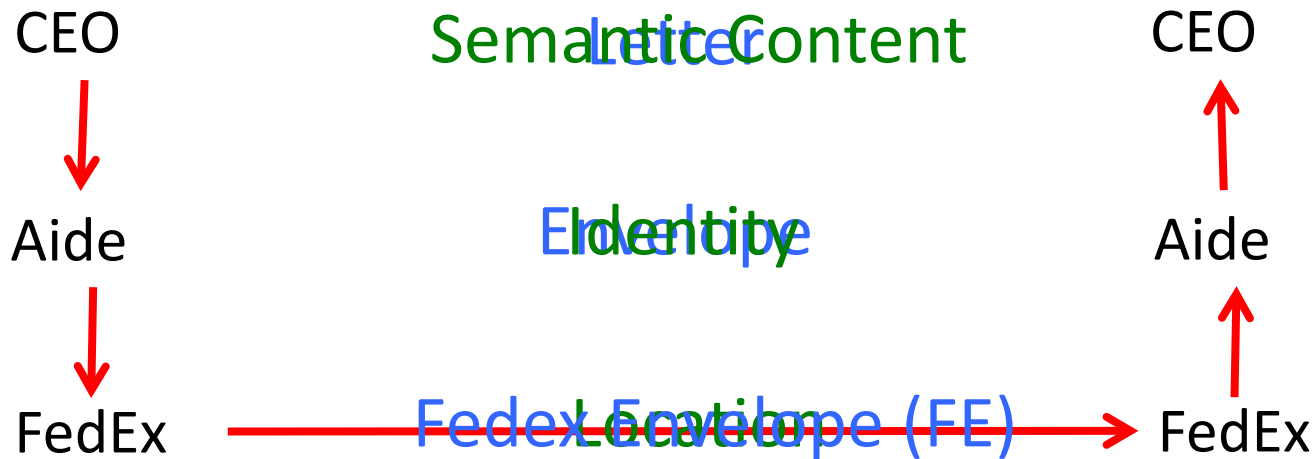 - Takes to FedEx

Your days are numbered.

- FedEx Office
  - Puts letter in larger envelope
  - Puts name and street address on FedEx envelope

Pat

  - Puts package on FedEx delivery truck
- FedEx delivers to other company
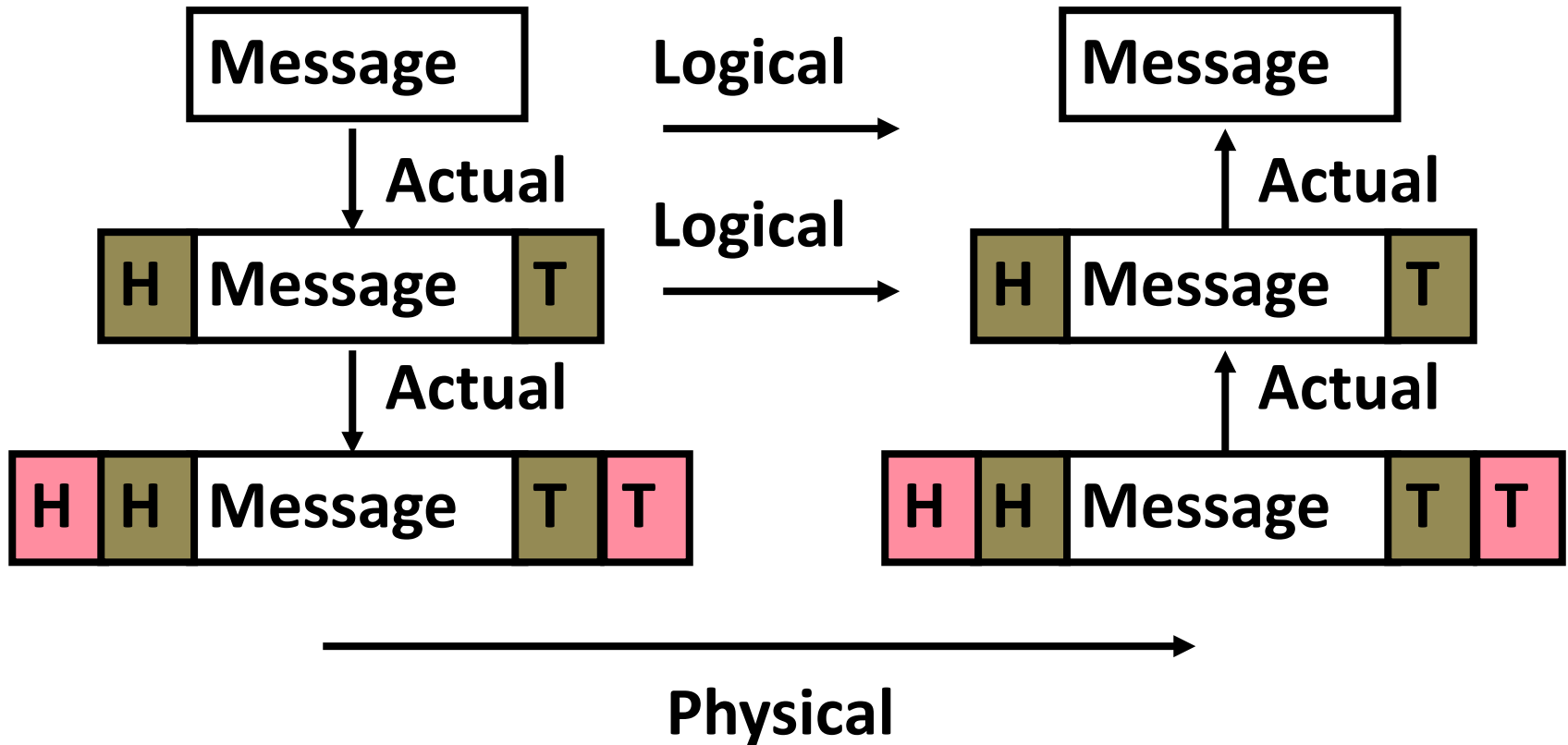
The Path of the Letter

"Peers" on each side understand the same things
No one else needs to
Lowest level has most packaging

CEO   Semantic Letter Content   CEO

Aide   Identity Header   Aide

FedEx   →   Fedex Envelope (FE) Location   →   FedEx
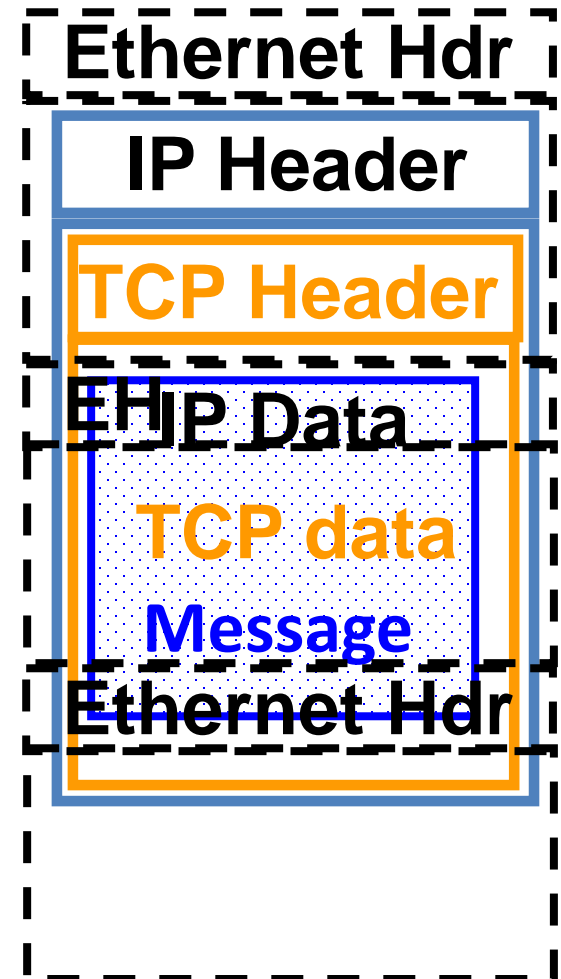
Protocol Family Concept

Each lower level of stack "encapsulates" information from layer above by adding header and trailer.

Most Popular Protocol for Network of Networks

- Transmission Control Protocol/Internet Protocol (TCP/IP)

- This protocol family is the basis of the Internet, a WAN (wide area network) protocol

  - IP makes best effort to deliver

    - Packets can be lost, corrupted

  - TCP guarantees delivery

  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN (local area network)

TCP/IP packet, Ethernet packet, protocols

- Application sends message

- **TCP breaks into 64KiB segments, adds 20B header**

- **IP adds 20B header, sends to network**

- **If Ethernet, broken into 1500B packets with headers, trailers**

Ethernet Hdr

IP Header

TCP Header

EHIP Data

TCP data

Message

Ethernet Hdr

"And in conclusion…"

- I/O gives computers their 5 senses
- I/O speed range is 100-million to one
- Polling vs. Interrupts
- DMA to avoid wasting CPU time on data transfers
- Disks for persistent storage, replaced by flash
- Networks: computer-to-computer I/O
  - Protocol suites allow networking of heterogeneous components. Abstraction!!!

Acknowledgement: Some Networking slides adapted from CS168