inst.eecs.berkeley.edu/~cs61c

CS61C : Machine Structures

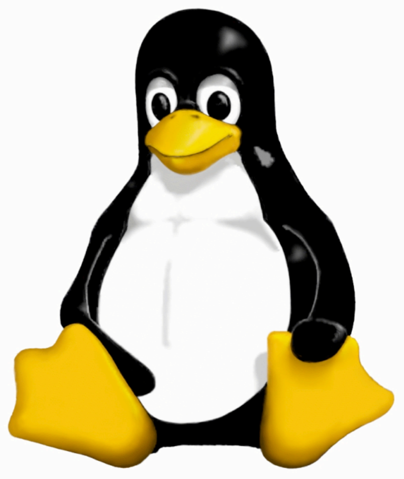
Lecture #28 Networking & Disks

2007-8-13

Scott Beamer, Instructor

Court Rules in favor of Novell: Linux is Safe

Novell.

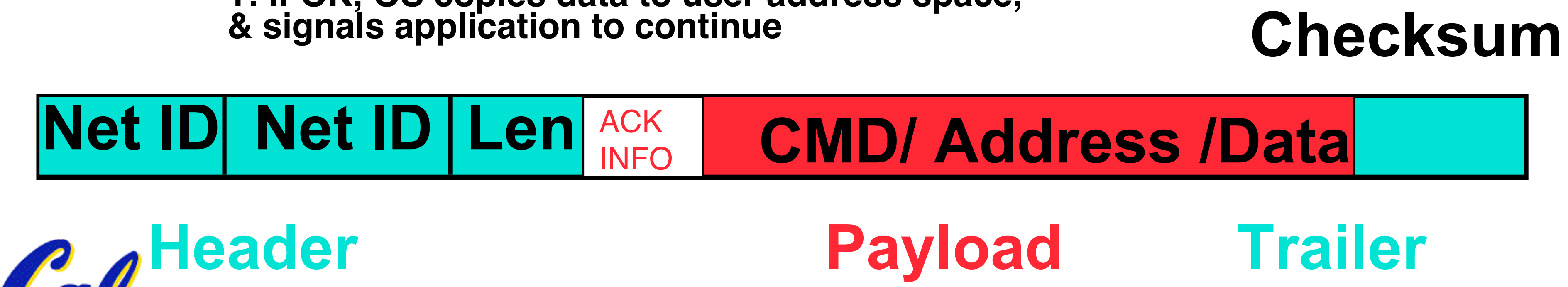


SCO

www.nytimes.com

Recap of Networking Intro

- Networks are essential in the modern age
- Can span large distances and can contain many nodes
- Our attempt at a simple networking protocol:
  - 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



Checksum

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

Header

Payload

Trailer

Protocol for Networks of Networks?

- Abstraction to cope with complexity of communication
- Networks are like onions
  - Hierarchy of layers:
    - Application (chat client, game, etc.)
    - Transport (TCP, UDP)
    - Network (IP)
    - Physical Link (wired, wireless, etc.)

Networks are like onions.

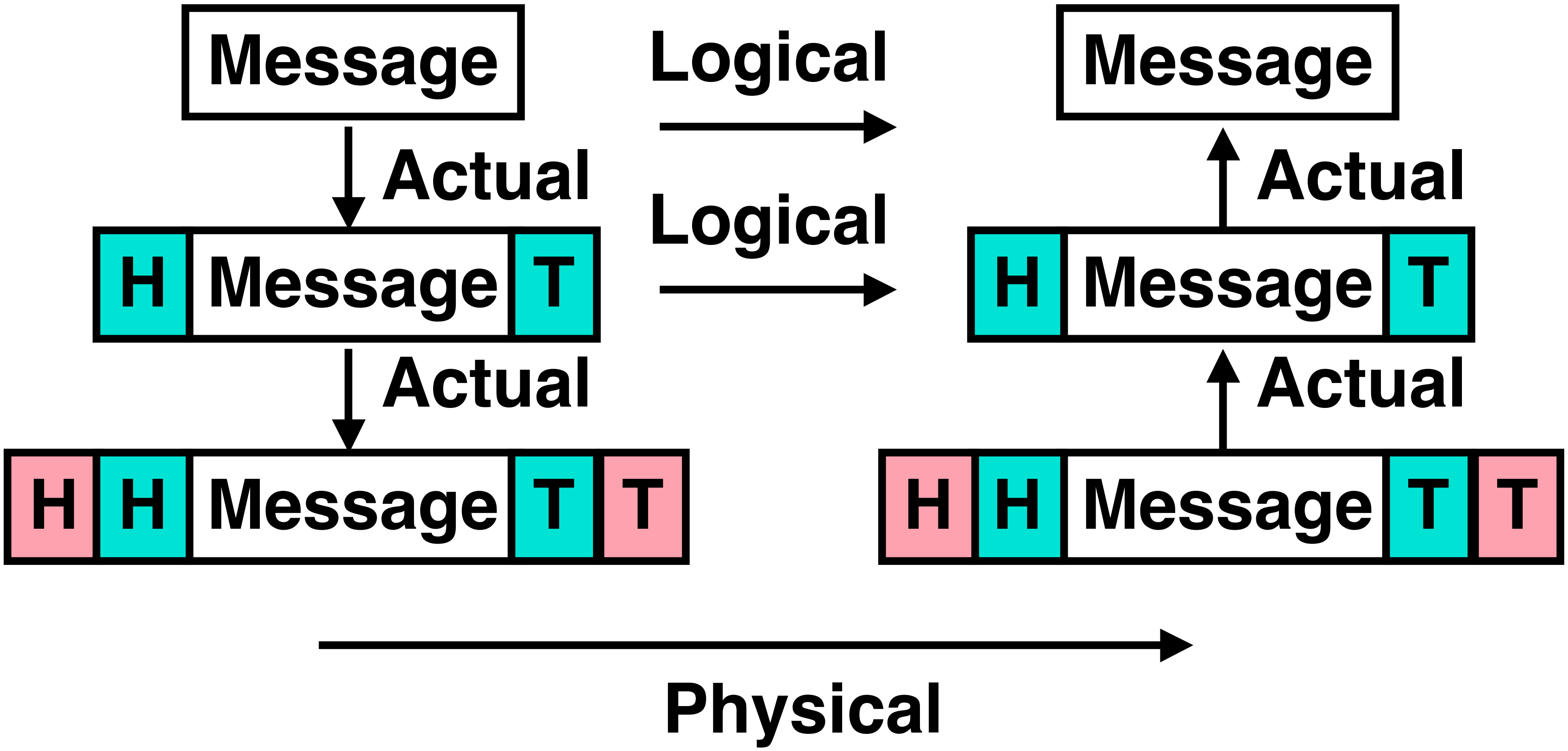
They stink?

Yes. No!

Oh, they make you cry.

No!… Layers. Onions have layers. Networks have layers.

Protocol Family Concept



Protocol Family Concept

- 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”
- Fragmentation: break packet into multiple smaller packets and reassemble

Protocol for Network of Networks

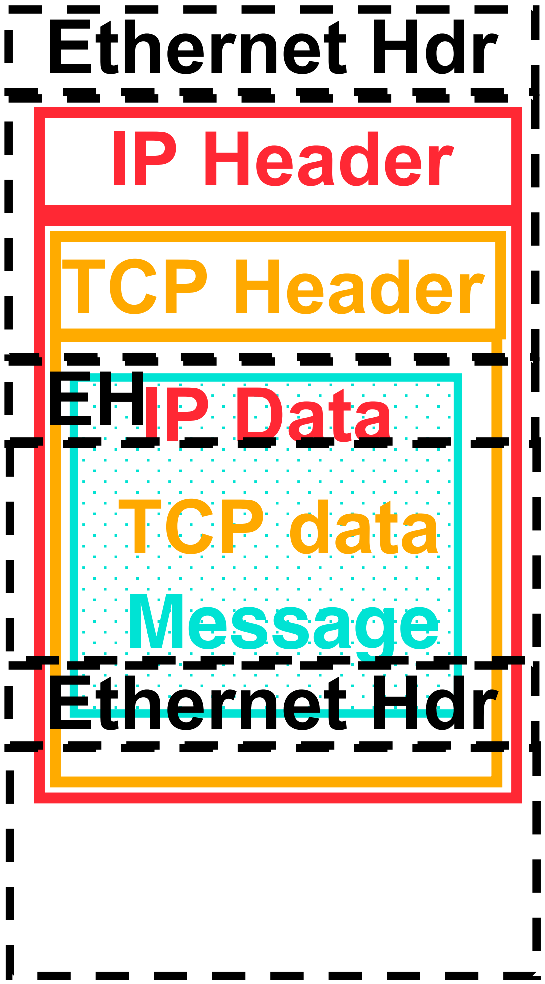
- **IP: Best-Effort Packet Delivery (Network Layer)**
- Packet switching
  - Send data in packets
  - Header with source & destination address
- "Best effort" delivery
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order

Protocol for Network of Networks

- Transmission Control Protocol/Internet Protocol (TCP/IP) (TCP :: a Transport Layer)
  - This protocol family is the basis of the Internet, a WAN protocol
  - IP makes best effort to deliver
  - TCP guarantees delivery
  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN

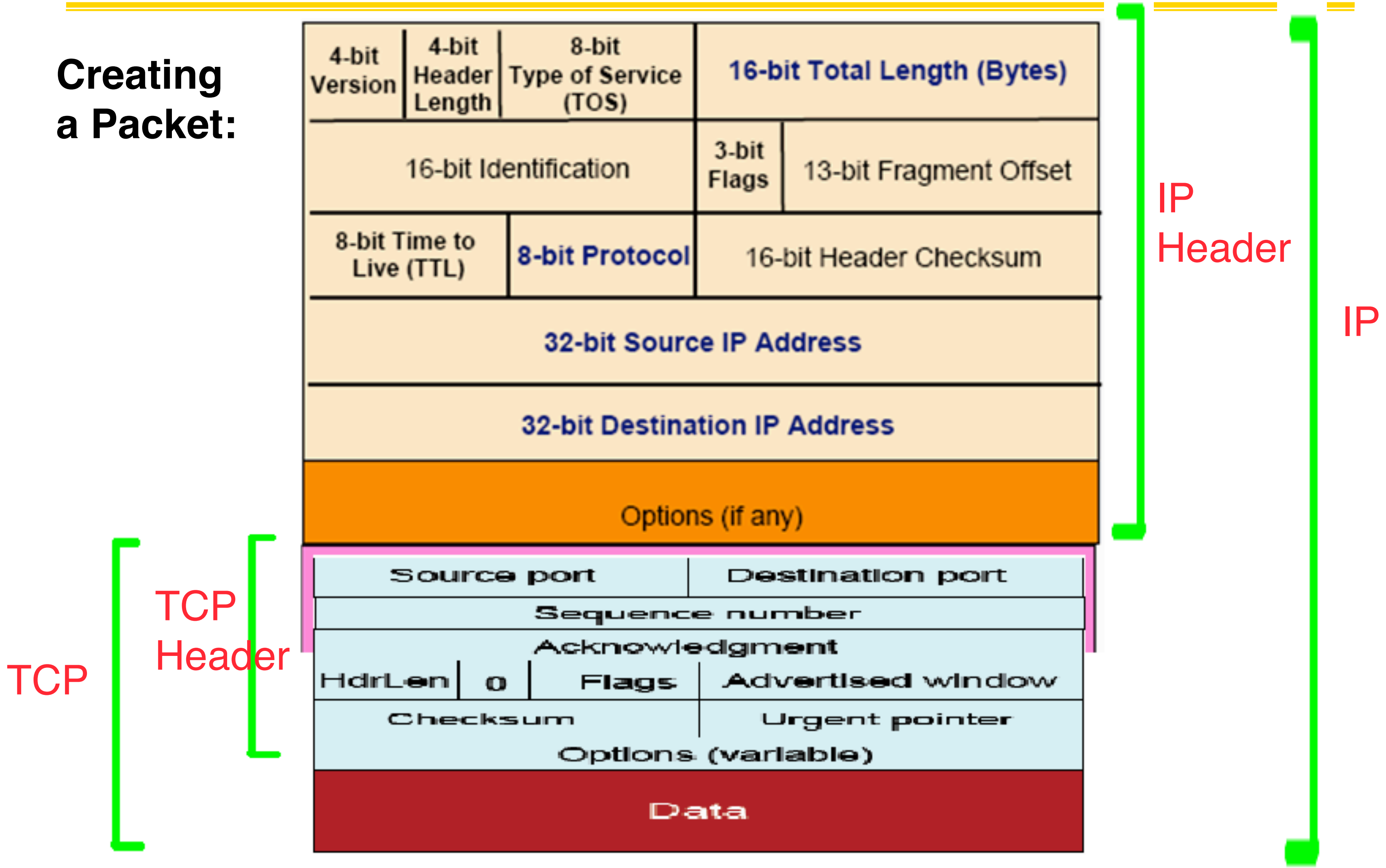
TCP/IP packet, Ethernet packet, protocols

- 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 (24B)
- All Headers, trailers have length field, destination, ...



TCP/IP in action

Creating a Packet:



4-bit Version | 4-bit Header Length | 8-bit Type of Service (TOS) | 16-bit Total Length (Bytes)

16-bit Identification | 3-bit Flags | 13-bit Fragment Offset

8-bit Time to Live (TTL) | 8-bit Protocol | 16-bit Header Checksum

32-bit Source IP Address

32-bit Destination IP Address

Options (if any)

Source port | Destination port

Sequence number

Acknowledgment

HdrLen | 0 | Flags | Advertised window

Checksum | Urgent pointer

Options (variable)

Data

IP Header

IP

TCP Header

TCP

Overhead vs. Bandwidth

- Networks are typically advertised using peak bandwidth of network link: e.g., 100 Mbits/sec Ethernet (“100 base T”)
- Software overhead to put message into network or get message out of network often limits useful bandwidth
- Assume overhead to send and receive = 320 microseconds (μs), want to send 1000 Bytes over “100 Mbit/s” Ethernet
  - Network transmission time: 1000Bx8b/B /100Mb/s = 8000b / (100b/μs) = 80 μs
  - Effective bandwidth: 8000b/(320+80)μs = 20 Mb/s

And in early conclusion…

- Protocol suites allow networking of heterogeneous components
  - Another form of principle of abstraction
  - Protocols ⇒ operation in presence of failures
  - Standardization key for LAN, WAN
- Integrated circuit (“Moore’s Law”) revolutionizing network switches as well as processors
  - Switch just a specialized computer
- Trend from shared to switched networks to get faster links and scalable bandwidth
- Interested?
  - EE122 (CS-based in Fall, EE –based in Spring)

Upcoming Calendar

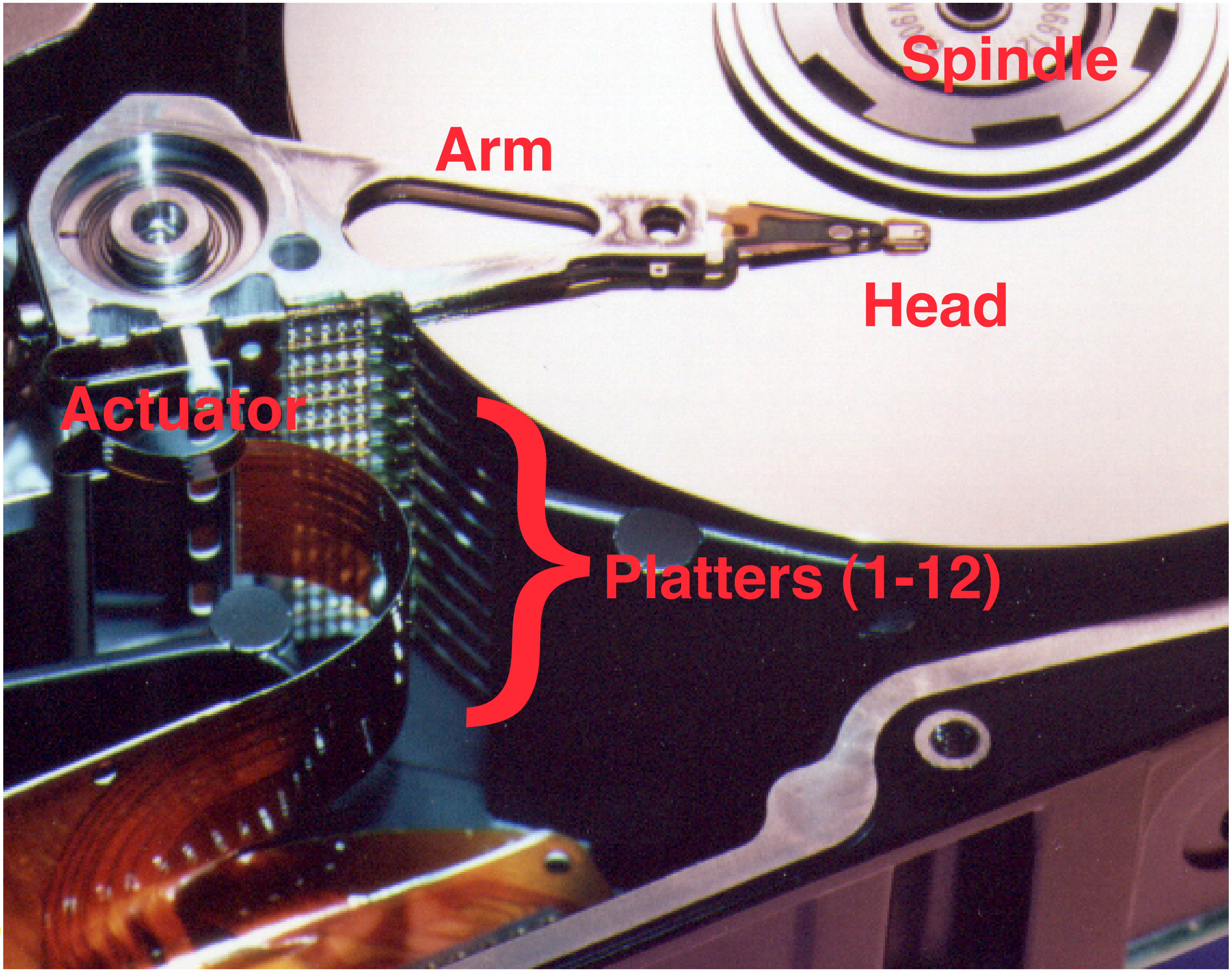
Time	Monday	Tuesday	Wednesday	Thursday
Lecture	I/O Networks & I/O Disks	Performance & Parallel Intro	Parallel	Summary & Course Evaluations
Afternoon/ Evening	Review Session 4-7pm @ 60 Evans	Networking Lab	Last Discussion Section	FINAL 7-10pm @ 10 Evans

- Administrivia
  - Scott's OH today moved to 1-2pm in 329 Soda
  - HW8 due tomorrow @ 11:59pm (no slip)

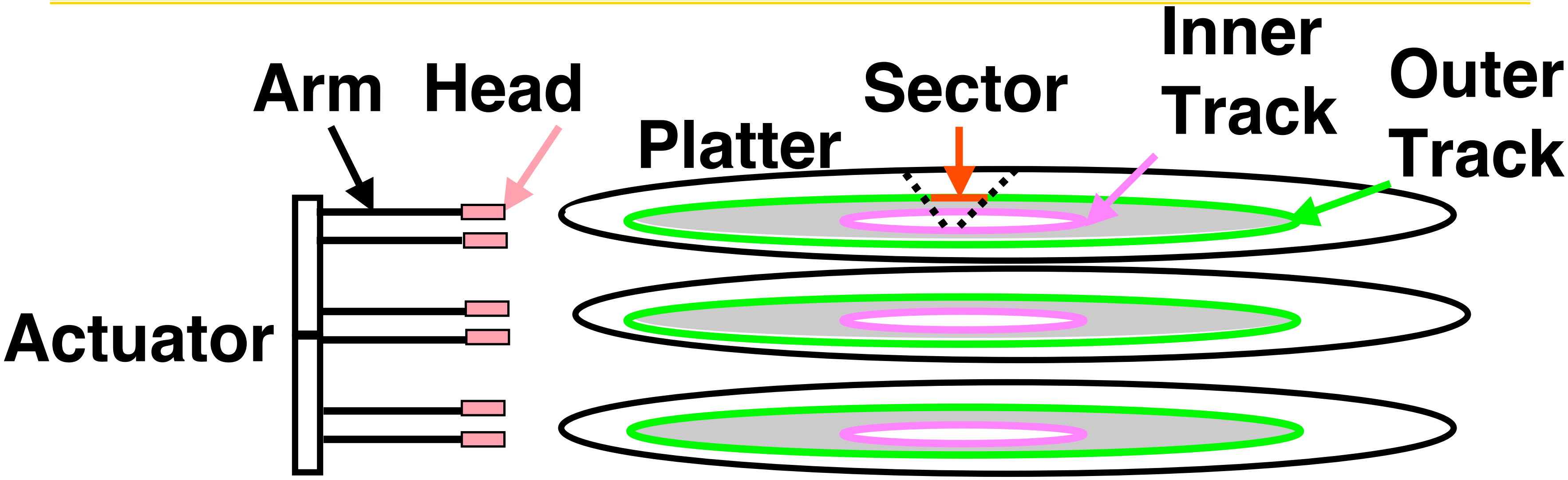
Magnetic Disk – common I/O device

- A kind of computer memory
  - Information sorted by magnetizing ferrite material on surface of rotating disk (similar to tape recorder except digital rather than analog data)
- Nonvolatile storage
  - retains its value without applying power to disk.
- Two Types
  - Floppy disks – slower, less dense, removable.
  - Hard Disk Drives (HDD) – faster, more dense, non-removable.
- Purpose in computer systems (Hard Drive):
  - Long-term, inexpensive storage for files
  - "Backup" for main-memory. Large, inexpensive, slow level in the memory hierarchy (virtual memory)

Photo of Disk Head, Arm, Actuator

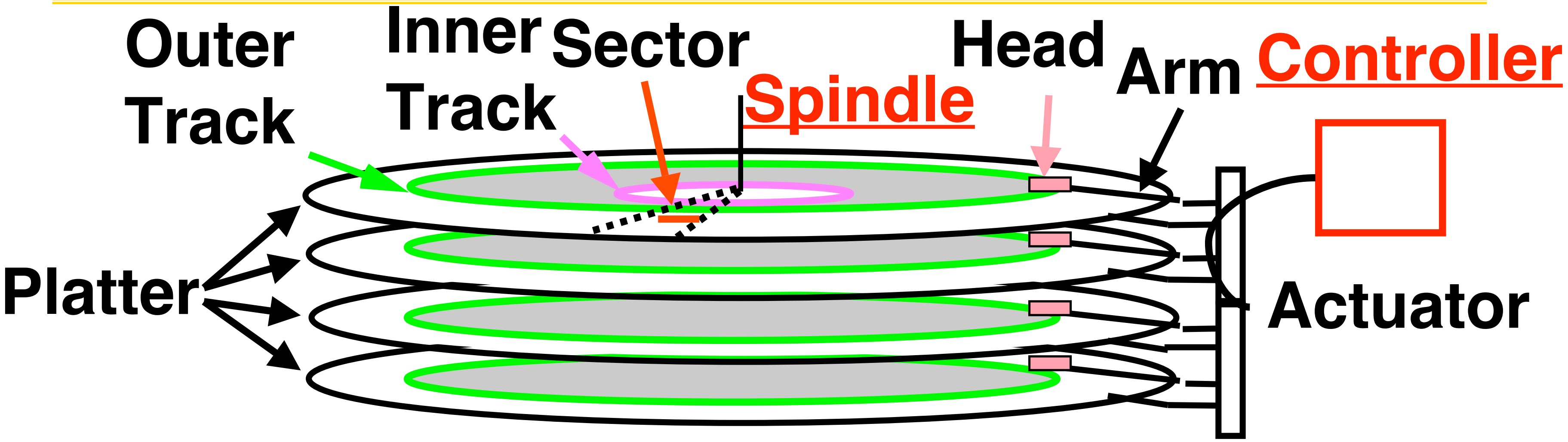


Disk Device Terminology



- 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

Disk Device Performance (1/2)



- **Disk Latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead**
  - Seek Time? depends on no. tracks to move arm, speed of actuator
  - Rotation Time? depends on speed disk rotates, how far sector is from head
  - Transfer Time? depends on data rate (bandwidth) of disk (f(bit density,rpm)), size of request

Disk Device Performance (2/2)

- Average distance of sector from head?
- 1/2 time of a rotation
  - 7200 Revolutions Per Minute ⇒ 120 Rev/sec
  - 1 revolution = 1/120 sec ⇒ 8.33 milliseconds
  - 1/2 rotation (revolution) ⇒ 4.17 ms
- Average no. tracks to move arm?
  - Disk industry standard benchmark:
    - Sum all time for all possible seek distances from all possible tracks / # possible
    - Assumes average seek distance is random
- Size of Disk cache can strongly affect perf!
  - Cache built into disk system, OS knows nothing

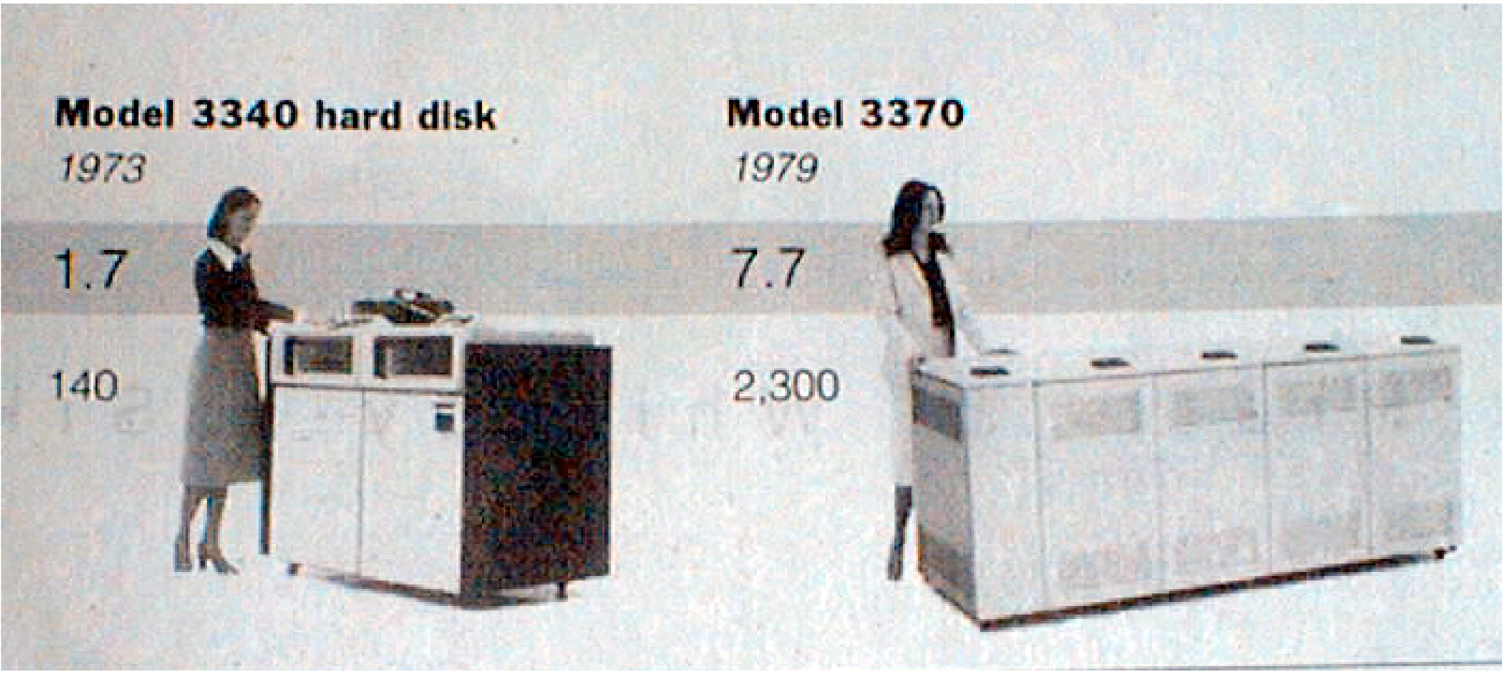
Data Rate: Inner vs. Outer Tracks

- To keep things simple, originally same number of sectors per track
  - Since outer track longer, lower bits per inch
- Competition ⇒ decided to keep bits per inch (BPI) high for all tracks ("constant bit density")
  - ⇒ More capacity per disk
  - ⇒ More sectors per track towards edge
  - ⇒ Since disk spins at constant speed, outer tracks have faster data rate
- Bandwidth outer track 1.7x inner track!

Early Disk History (IBM)

Data density Mbit/sq. in.

Capacity of unit shown Megabytes

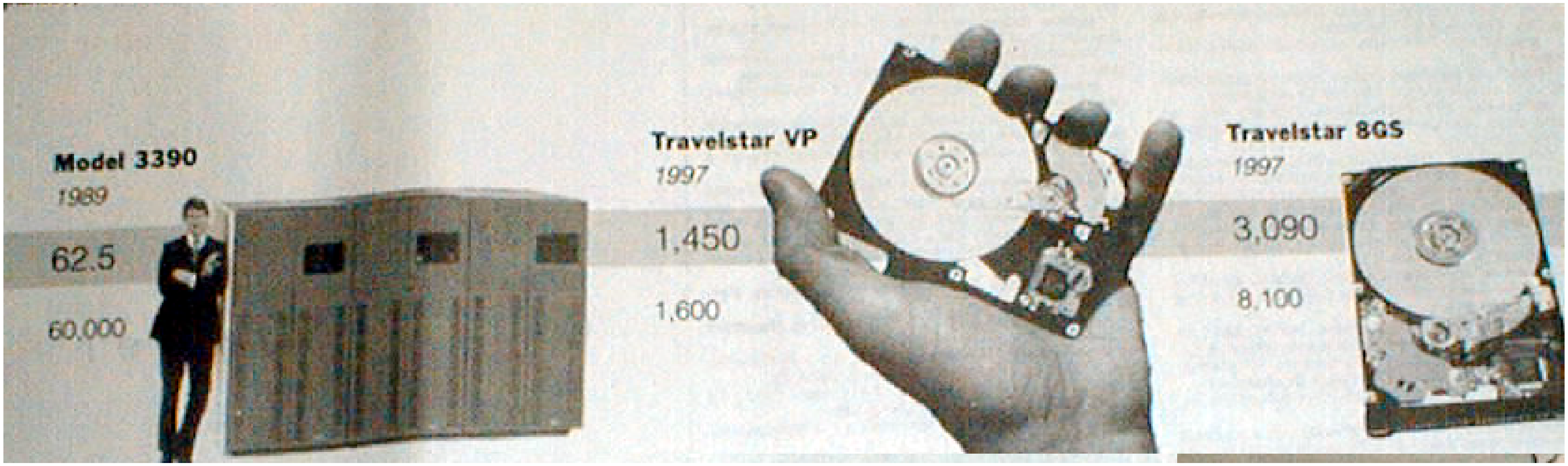


1973:
1. 7 Mbit/sq. in
140 MBytes

1979:
7. 7 Mbit/sq. in
2,300 MBytes

source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"

Early Disk History



1989:
63 Mbit/sq. in
60,000 MBytes

1997:
1450 Mbit/sq. in
1600 MBytes

1997:
3090 Mbit/sq. in
8100 MBytes

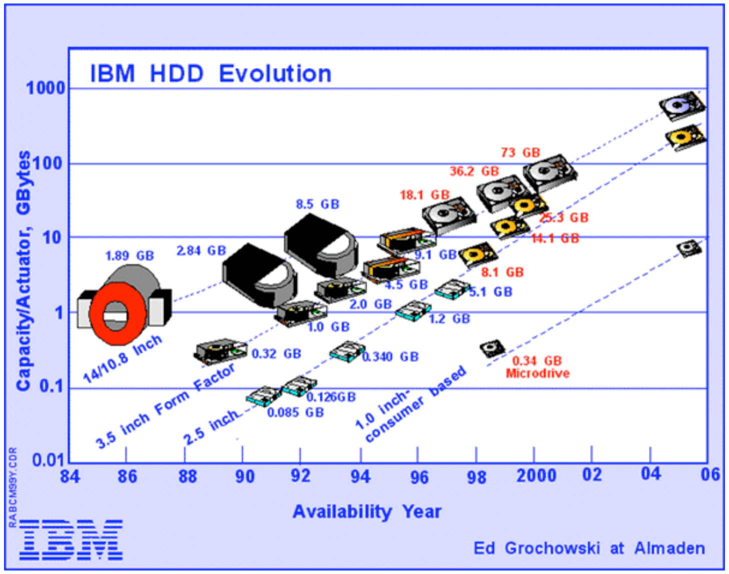
source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"

Disk Performance Model /Trends

- Capacity : + 100% / year (2X / 1.0 yrs)

  Over time, grown so fast that # of platters has reduced (some even use only 1 now!)

- Transfer rate (BW) : + 40%/yr (2X / 2 yrs)
- Rotation+Seek time : – 8%/yr (1/2 in 10 yrs)
- Areal Density
  - Bits recorded along a track: Bits/Inch (BPI)
  - # of tracks per surface: Tracks/Inch (TPI)
  - We care about bit density per unit area Bits/Inch^2
  - Called Areal Density = BPI x TPI
  - "~120 Gb/In^2 is longitudinal limit"
  - "230 Gb/In^2 now with perpendicular"
- GB/$: > 100%/year (2X / 1.0 yrs)
  - Fewer chips + areal density



State of the Art: Two camps (2006)

- Performance
  - Enterprise apps, servers
- E.g., Seagate Cheetah 15K.5
  - Ultra320 SCSI, 3 Gbit/sec, Serial Attached SCSI (SAS), 4Gbit/sec Fibre Channel (FC)
  - 300 GB, 3.5-inch disk
  - 15,000 RPM
  - 13 watts (idle)
  - 3.5 ms avg. seek
  - 125 MB/s transfer rate
  - 5 year warrantee
  - $1000 = $3.30 / GB

- Capacity
  - Mainstream, home uses
- E.g., Seagate Barracuda 7200.10
  - Serial ATA 3Gb/s (SATA/300), Serial ATA 1.5Gb/s (SATA/150), Ultra ATA/100
  - 750 GB, 3.5-inch disk
  - 7,200 RPM
  - 9.3 watts (idle)
  - 8.5 ms avg. seek
  - 78 MB/s transfer rate
  - 5 year warrantee
  - $350 = $0.46 / GB
- Uses Perpendicular Magnetic Recording (PMR)!!
  - What's that, you ask?

Hitachi now has a 1TB drive! (Deskstar 7K1000)

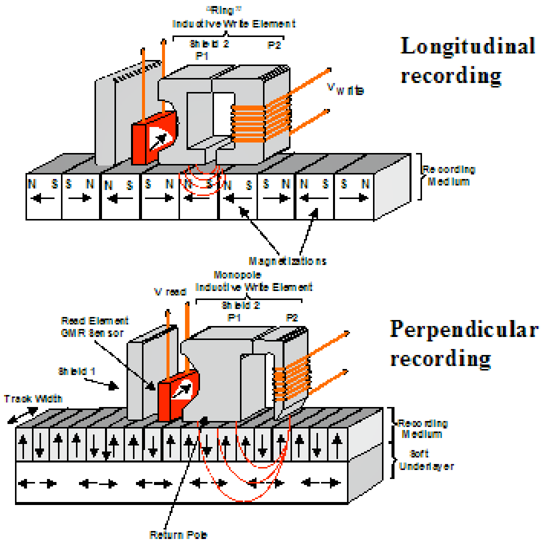
source: www.seagate.com

1 inch disk drive!

HITACHI
Inspire the Next

- Hitachi 2007 release
  - Development driven by iPods & digital cameras
  - 20GB, 5-10MB/s (higher?)
  - 42.8 x 36.4 x 5 mm

- Perpendicular Magnetic Recording (PMR)
  - FUNDAMENTAL new technique
  - Evolution from Logitudinal
    - Starting to hit physical limit due to superparamagnetism
  - They say 10x improvement



© 2005, Hitachi Global Storage Technologies

www.hitachi.com/New/cnews/050405.html

www.hitachigst.com/hdd/research/recording_head/pr/

Where does Flash memory come in?

- Microdrives and Flash memory (e.g., CompactFlash) are going head-to-head
  - Both non-volatile (no power, data ok)
  - Flash benefits: durable & lower power (no moving parts, need to spin μdrives up/down)
  - Flash limitations: finite number of write cycles (wear on the insulating oxide layer around the charge storage mechanism)
- How does Flash memory work?
  - NMOS transistor with an additional conductor between gate and source/drain which “traps” electrons. The presence/absence is a 1 or 0.

en.wikipedia.org/wiki/Flash_memory

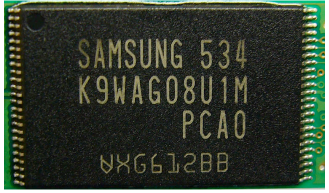
What does Apple put in its iPods?

en.wikipedia.org/wiki/Ipod
www.apple.com/ipod

iPod nano shuffle

Toshiba 1.8-inch HDD
30, 80GB

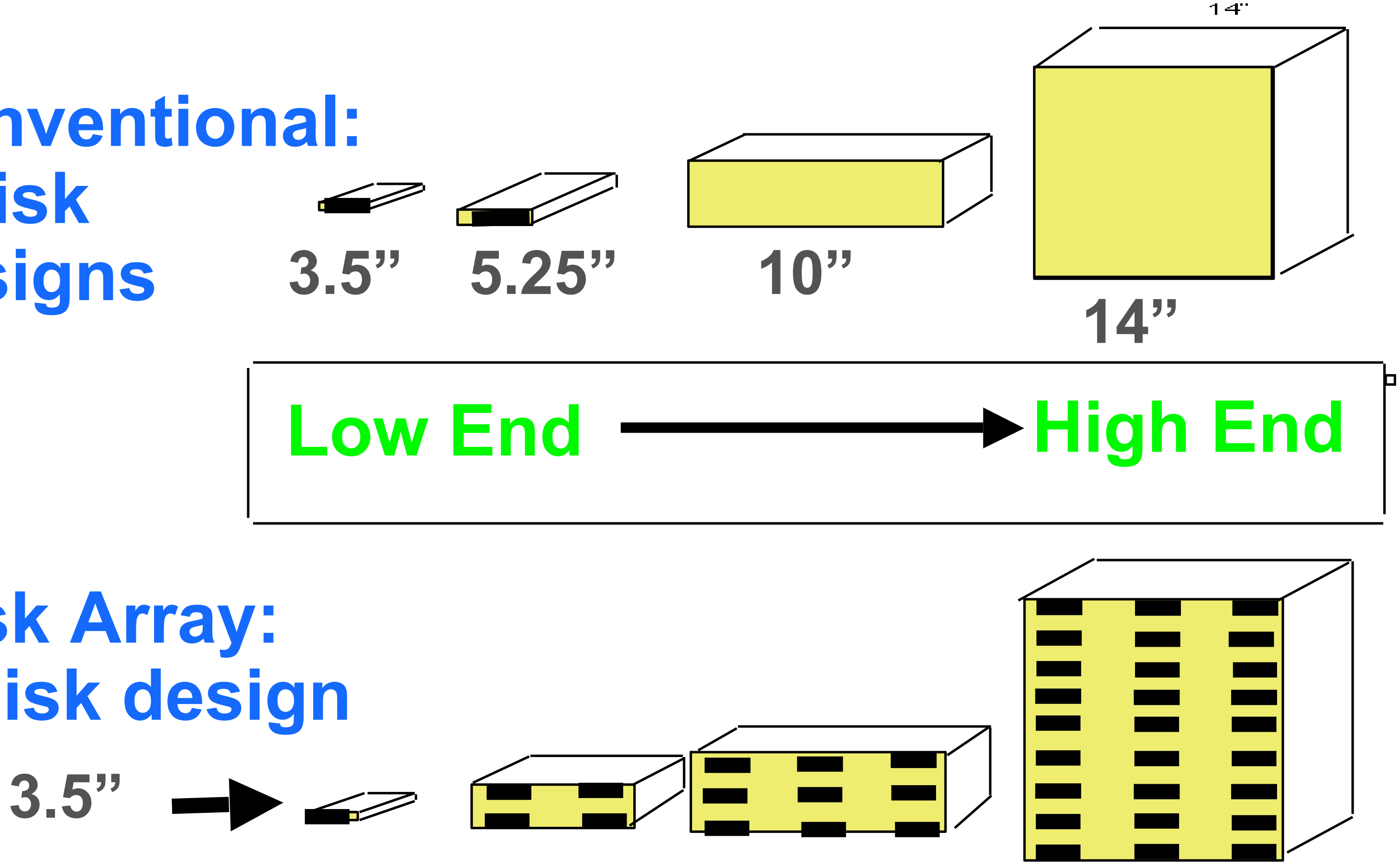
Samsung flash
2, 4, 8GB



Toshiba flash
1GB

Use Arrays of Small Disks…

- Katz and Patterson asked in 1987:
  - Can smaller disks be used to close gap in performance between disks and CPUs?



Conventional:
4 disk
designs

3.5” 5.25” 10” 14”

Low End → High End

Disk Array:
1 disk design

3.5” →

Replace Small Number of Large Disks with Large Number of Small Disks! (1988 Disks)

	IBM 3390K	IBM 3.5" 0061	x70	
Capacity	20 GBytes	320 MBytes	23 GBytes	
Volume	97 cu. ft.	0.1 cu. ft.	11 cu. ft.	9X
Power	3 KW	11 W	1 KW	3X
Data Rate	15 MB/s	1.5 MB/s	120 MB/s	8X
I/O Rate	600 I/Os/s	55 I/Os/s	3900 IOs/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

Disk Arrays potentially high performance, high MB per cu. ft., high MB per KW,

but what about reliability?

Array Reliability

- Reliability - whether or not a component has failed
  - measured as Mean Time To Failure (MTTF)
- Reliability of N disks
  = Reliability of 1 Disk ÷ N
  (assuming failures independent)
  - 50,000 Hours ÷ 70 disks = 700 hour
- Disk system MTTF:
  Drops from 6 years to 1 month!
- Disk arrays too unreliable to be useful!

Redundant Arrays of (Inexpensive) Disks

- Files are “striped” across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array

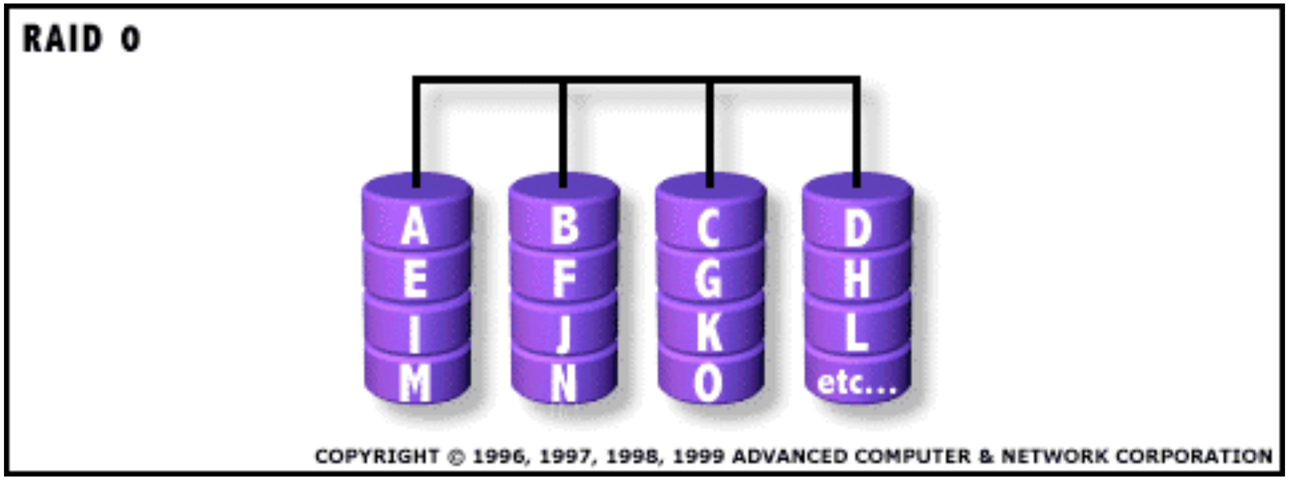
  ⇒ Capacity penalty to store redundant info

  ⇒ Bandwidth penalty to update redundant info

Berkeley History, RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software
- Today RAID is > tens billion dollar industry, 80% non-PC disks sold in RAIDs

“RAID 0”: No redundancy = “AID”

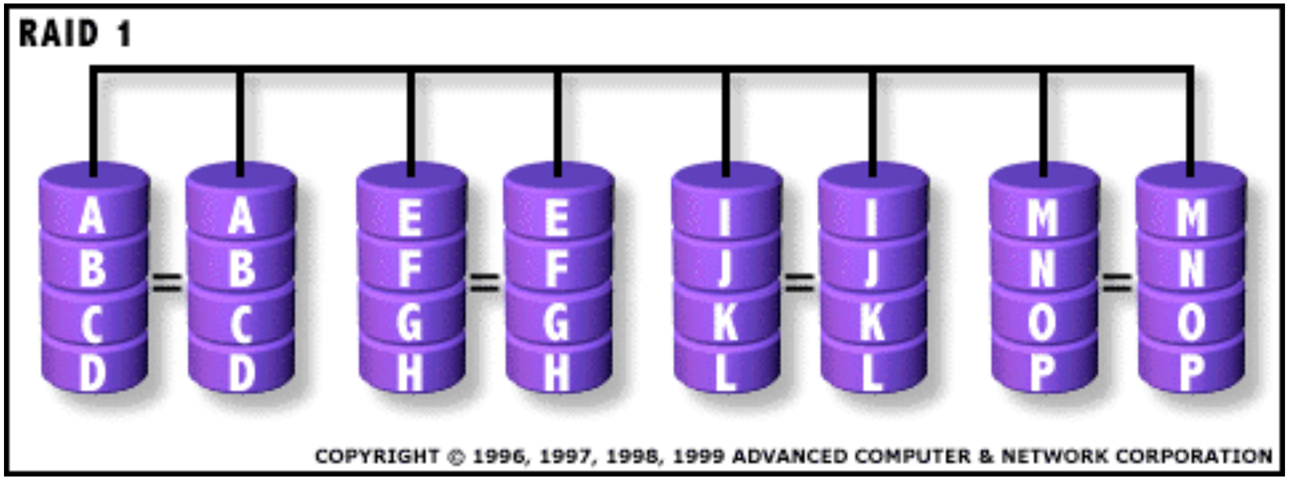

RAID 0
A
E
I
M
B
F
J
N
C
G
K
O
D
H
L
etc...
COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

- Assume have 4 disks of data for this example, organized in blocks
- Large accesses faster since transfer from several disks at once

This and next 5 slides from RAID.edu, http://www.acnc.com/04_01_00.html

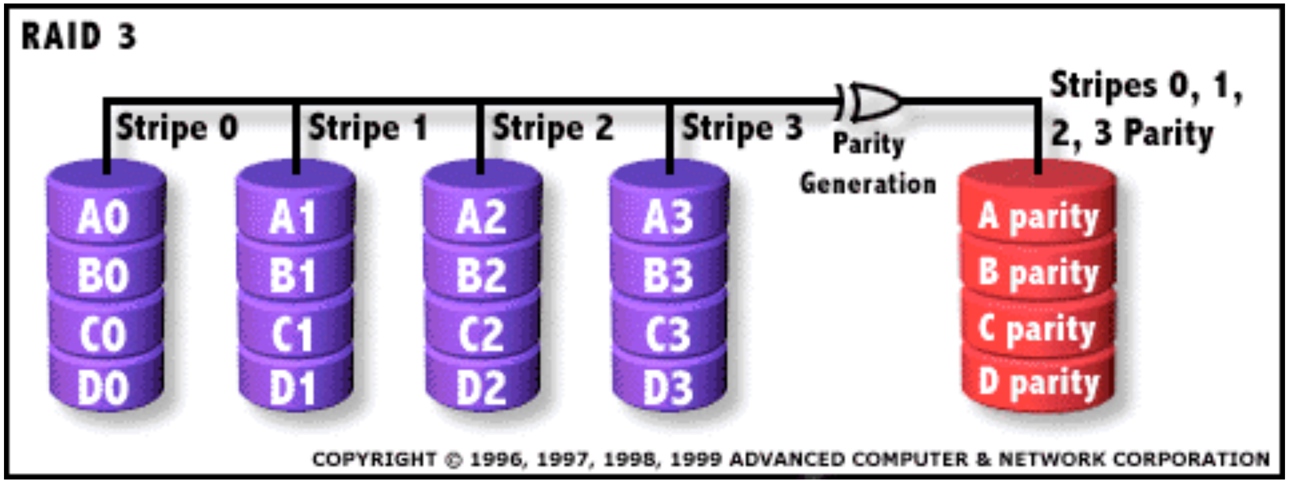
`http://www.raid.com/04_00.html` **also has a great tutorial**

RAID 1: Mirror data


RAID 1
A
B
C
D
=
A
B
C
D
E
F
G
H
=
E
F
G
H
I
J
K
L
=
I
J
K
L
M
N
O
P
=
M
N
O
P
COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION

- Each disk is fully duplicated onto its "mirror"
  - Very high availability can be achieved
- Bandwidth reduced on write:
  - 1 Logical write = 2 physical writes
- Most expensive solution: 100% capacity overhead

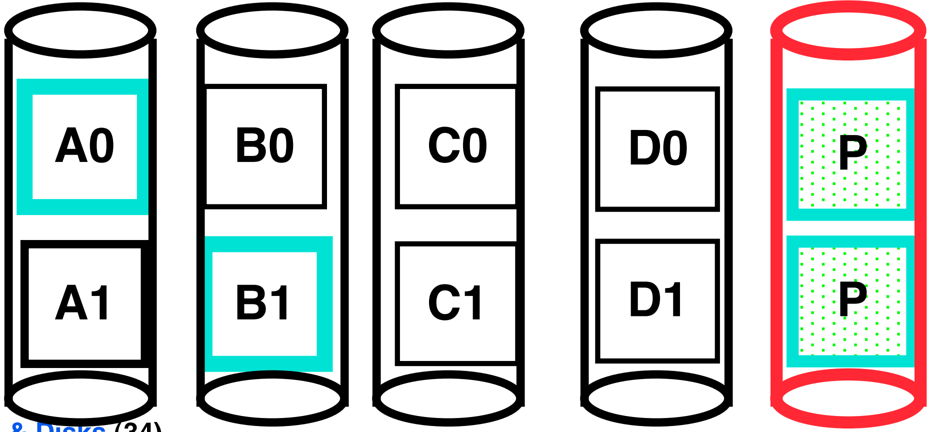
RAID 3: Parity



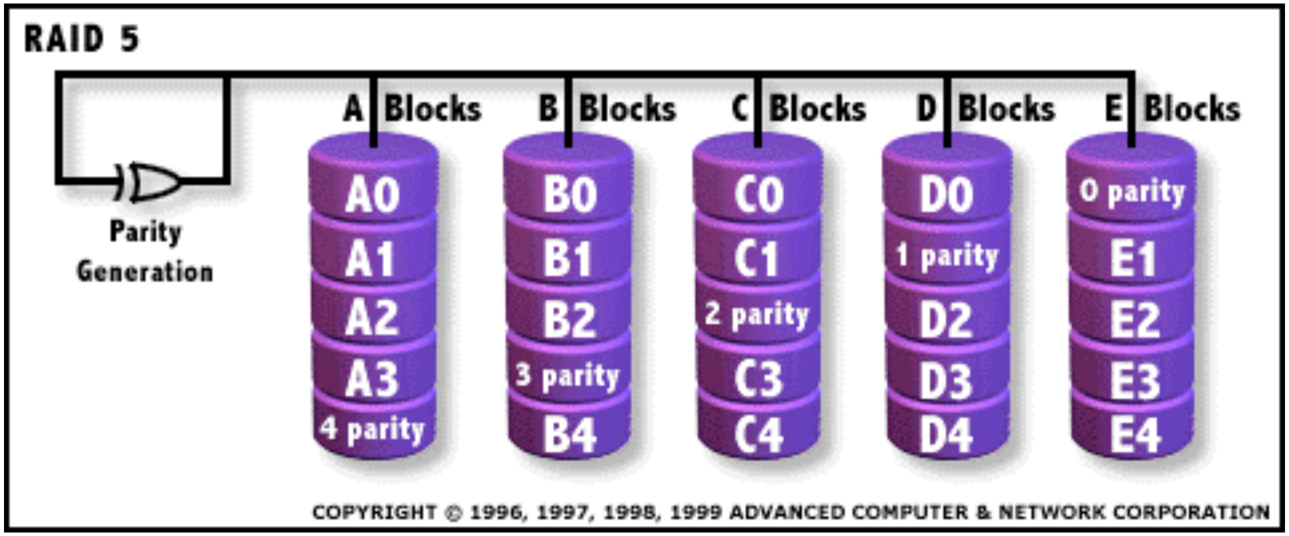
- Parity computed across group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk
- 25% capacity cost for parity in this example vs. 100% for RAID 1 (5 disks vs. 8 disks)

Inspiration for RAID 5 (RAID 4 block-striping)

- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk (access all disks)
  - Option 2: since P has old sum, compare old data to new data, add the difference to P: 1 logical write = 2 physical reads + 2 physical writes to 2 disks
- Parity Disk is bottleneck for Small writes: Write to A0, B1 => both write to P disk



RAID 5: Rotated Parity, faster small writes



- Independent writes possible because of interleaved parity
  - Example: write to A0, B1 uses disks 0, 1, 4, 5, so can proceed in parallel
  - Still 1 small write = 4 physical disk accesses

en.wikipedia.org/wiki/Redundant_array_of_independent_disks

Peer Instruction

1. RAID 1 (mirror) and 5 (rotated parity) help with performance and availability
2. RAID 1 has higher cost than RAID 5
3. Small writes on RAID 5 are slower than on RAID 1

	ABC
0:	FFF
1:	FFT
2:	FTF
3:	FTT
4:	TFF
5:	TFT
6:	TTF
7:	TTT

Peer Instruction Answer

1. All RAID (0-5) helps with performance, only RAID0 doesn’t help availability. TRUE
2. Surely! Must buy 2x disks rather than 1.25x (from diagram, in practice even less) TRUE
3. RAID5 (2R,2W) vs. RAID1 (2W). Latency worse, throughput (ll writes) better. TRUE

1. RAID 1 (mirror) and 5 (rotated parity) help with performance and availability
2. RAID 1 has higher cost than RAID 5
3. Small writes on RAID 5 are slower than on RAID 1

	ABC
0:	FFF
1:	FFT
2:	FTF
3:	FTT
4:	TFF
5:	TFT
6:	TTF
7:	TTT

"And In conclusion…"

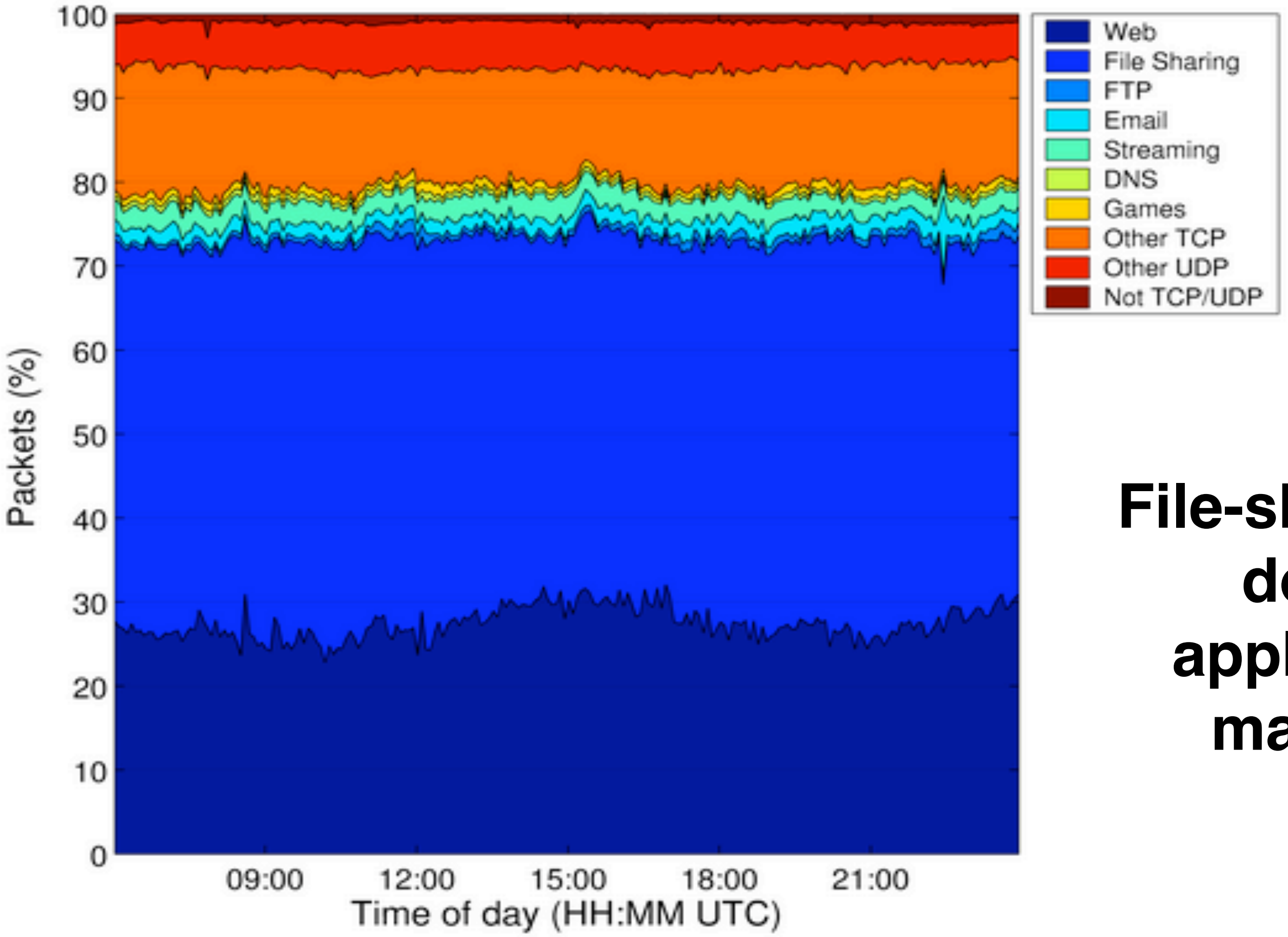
- Magnetic Disks continue rapid advance: 60%/yr capacity, 40%/yr bandwidth, slow on seek, rotation improvements, MB/$ improving 100%/yr?
  - Designs to fit high volume form factor
  - PMR a fundamental new technology
    - breaks through barrier
- RAID
  - Higher performance with more disk arms per $
  - Adds option for small # of extra disks
  - Can nest RAID levels
  - Today RAID is > tens-billion dollar industry, 80% nonPC disks sold in RAIDs, started at Cal

Bonus slides

- These are extra slides that used to be included in lecture notes, but have been moved to this, the “bonus” area to serve as a supplement.
- The slides will appear in the order they would have in the normal presentation

Bonus

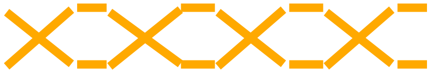
[Bonus] Backbone Link App Composition



File-sharing is the dominant application on many links!

[Bonus] Example: Network Media

Twisted Pair
(“Cat 5”):



Copper, 1mm think, twisted to avoid antenna effect

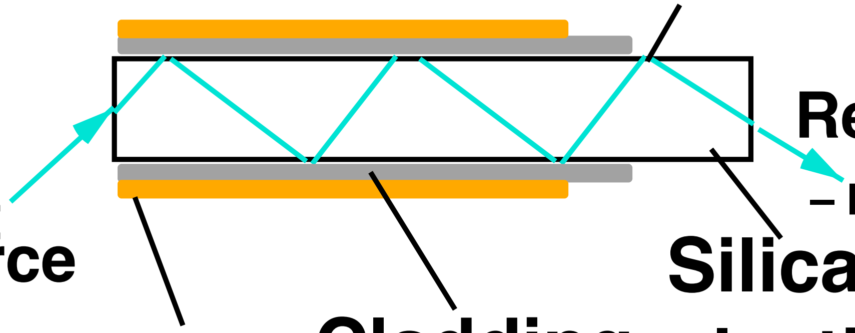
Light:
3 parts are
cable, light
source,
light
detector

Fiber Optics

Air

Total internal
reflection

Transmitter
Is L.E.D or
Laser Diode



Receiver

– Photodiode

light
source

Buffer

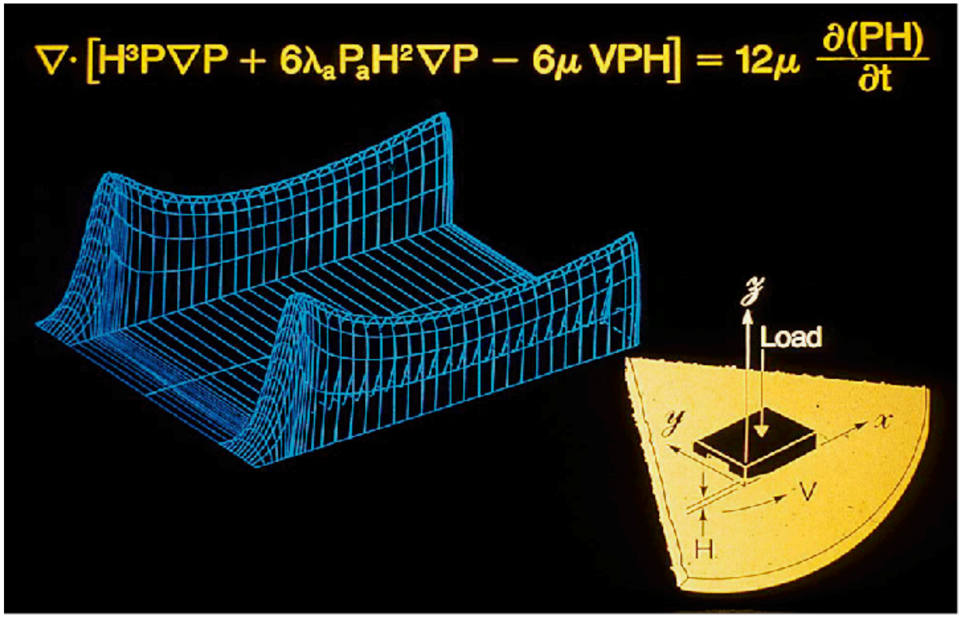
Cladding

Silica: glass or plastic; actually < 1/10 diameter of copper

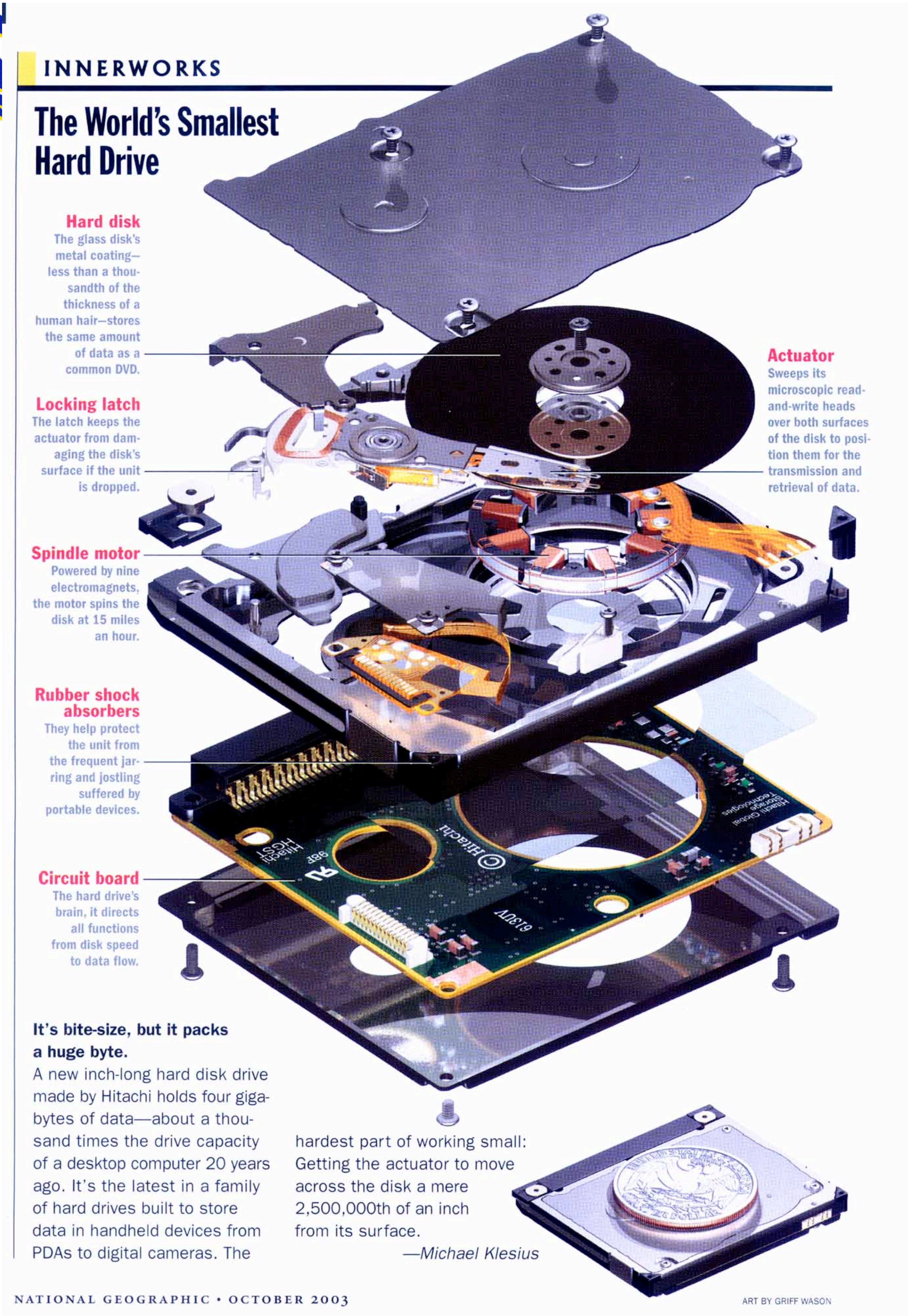
BONUS : Hard Drives are Sealed. Why?

- The closer the head to the disk, the smaller the "spot size" and thus the denser the recording.
  - Measured in Gbit/in^2. ~60 is state of the art.

- Disks are sealed to keep the dust out.
  - Heads are designed to "fly" at around 5-20nm 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.



$$\nabla \cdot [H^3 P \nabla P + 6\lambda_a P_a H^2 \nabla P - 6\mu VPH] = 12\mu \frac{\partial(PH)}{\partial t}$$



INNERWORKS

The World's Smallest Hard Drive

Hard disk The glass disk's metal coating—less than a thousandth of the thickness of a human hair—stores the same amount of data as a common DVD.

Locking latch The latch keeps the actuator from damaging the disk's surface if the unit is dropped.

Spindle motor Powered by nine electromagnets, the motor spins the disk at 15 miles an hour.

Rubber shock absorbers They help protect the unit from the frequent jarring and jostling suffered by portable devices.

Circuit board The hard drive's brain, it directs all functions from disk speed to data flow.

Actuator Sweeps its microscopic read-and-write heads over both surfaces of the disk to position them for the transmission and retrieval of data.

It's bite-size, but it packs a huge byte.

A new inch-long hard disk drive made by Hitachi holds four gigabytes of data—about a thousand times the drive capacity of a desktop computer 20 years ago. It's the latest in a family of hard drives built to store data in handheld devices from PDAs to digital cameras. The hardest part of working small: Getting the actuator to move across the disk a mere 2,500,000th of an inch from its surface.

—*Michael Klesius*

NATIONAL GEOGRAPHIC • OCTOBER 2003

ART BY GRIFF WASON

Historical Perspective

- *Form factor* and *capacity* are more important in the marketplace than is performance
- Form factor evolution:

  1970s: Mainframes ⇒ 14 inch diameter disks

  1980s: Minicomputers, Servers
  ⇒ 8", 5.25" diameter disks

  Late 1980s/Early 1990s:
  - PCs ⇒ 3.5 inch diameter disks
  - Laptops, notebooks ⇒ 2.5 inch disks
  - Palmtops didn't use disks, so 1.8 inch diameter disks didn't make it
- Early 2000s:
  - MP3 players ⇒ 1 inch disks

Disk Performance Example

- Calculate time to read 1 sector (512B) for Deskstar using advertised performance; sector is on outer track

Disk latency = average seek time + average rotational delay + transfer time + controller overhead

= 8.5 ms + 0.5 * 1/(7200 RPM)
+ 0.5 KB / (100 MB/s) + 0.1 ms

= 8.5 ms + 0.5 /(7200 RPM/(60000ms/M))
+ 0.5 KB / (100 KB/ms) + 0.1 ms

= 8.5 + 4.17 + 0.005 + 0.1 ms = 12.77 ms

- How many CPU clock cycles is this?