Lecture #31

****
Review -- 1 min
****
Network Interface

HW: Key questions

- how far from CPU
- how much power on NI?
- DMA v. PIO, Interrupts v. polling

SW: Key questions

- avoid OS!!
- reduce layering, copies, …

Amdahl's law strikes again!

Application Performance: NFS Performance

Network	Avg NFS	NFS v. E	BW v. E	UDP(200) v. E
Ethernet	14.5 ms	1.00	1.00	1.00
ATM	11.8 ms	1.22	15	1.09
Myrinet	13.3 ms	1.09	64	1.09

- UDP(200) v. E
  - compares round-trip performance for 200 byte message
- UDP Latency predicts performance better than link BW

****
Outline - 1 min
****

Beyond the NI:

Media
Topologies
Routing
Connections

Flow control

****
Preview - 1 min
****
Finish “Beyond the NI”
then Multiprocessors

****
Lecture - 20 min
****

Beyond the NI

Network Media

Media

Media	BW (Mbit/s)	Distance	Cost/ meter	Cost/ interface
Twisted Pair	0.1-100	100m-1000m	$0.23	$2
Coax Cable	10-100	1000m	$1.64	$5
Multi-mode Optical	600	2000m	$1.03	$1000
Single-mode Optical	2000	100,000m	$1.64	$1000
Many Wires	1280	10m	$10	$500
Wireless	0.01 - 4	1-1000	??	??

- Optical – not a panacea
- Twisted pair – cheap
- Many wires, Wireless – new technology

1) Twisted Pair
<PICTURE>
telephone wire, “Cat 5 wire”

QUESTION: why twisted?
A: avoid antenna effect

Bandwidth – 10-100 Mbit/s (1-0.1 km)

Cost -- $0.23/meter; $2/interface

2) Coax Cable
<picture>
cable TV wire

concentric wires for same reason as twisted pair
(avoid antenna)

Bandwidth – 10-100 Mbit/s (1km)
Cost -- $1.64/meter; $5/interface

3) Fiber Optics

    picture – total internal reflection

    multimode fiber (LED) –

        Bandwidth – 600 Mbit/s
        distance 2 km
        cost $1.03/meter $1000/interface

    single mode fiber (laser)

        2000+ Mbit/s
        100 km (long distance b/c laser avoids dispersion)
        cost/meter $1.64 cost/interface $1000

4) Many wires
e.g. Myrinet = 32 wires @ 20Mhz = 640 Mbit/s
main trick – making the bits transmitted together arrive at dest together
→ short distances only
expensive cables ($17/meter)
relatively cheap interface ($500-$1000)

    (interfaces are cheap enough that repeaters may be practical
    for longer distances)

5) wireless networks

    infrared, radio, metricom
    some line of sight, some 100's of meters

    9600 baud – 4 mbit

interesting failure modes…

Question: Error-free Networks?

Bus assumption – errors are rare
→Crash machine on bus error
Same for (some) networks?
 e.g. FLASH, Fugu, …
 Advantage – get retransmission copy out of fast path

 Error → retransmission → more complex software protocols

Engineering to avoid data corruption:
Reduce error rate by
 Going Slower
 Shorter Wires
 More Error Correction (redundant data)

What about dropped packets?
 Hard problem

****
Lecture - 24 min
****

Topologies: Shared bus v. switched
Trend – evolving towards switched

- Better performance
- More scalable
- Easier to upgrade

Integrated circuit revolutionizing networks as well as processors

- Switch == Specialized Computer

Shared still important

- Historical reasons
- Wireless networks

Shared media (e.g. Ethernet)

broadcast – each message goes to all hosts

hardware filters requests that a machine doesn't care about

arbitration – who gets to talk

on bus – bus controller (extra wires)

not appropriate on LAN

- ♦ no extra wires
- ♦ who gets to be arbiter?

3-pronged attach

1) carrier sensing – listen to check if wire being used
2) collision detection – listen on transmit to see if collision
3) random, exponential backof – after a collision, wait a random period of time (if another collision, wait even longer)

Advantages of shared

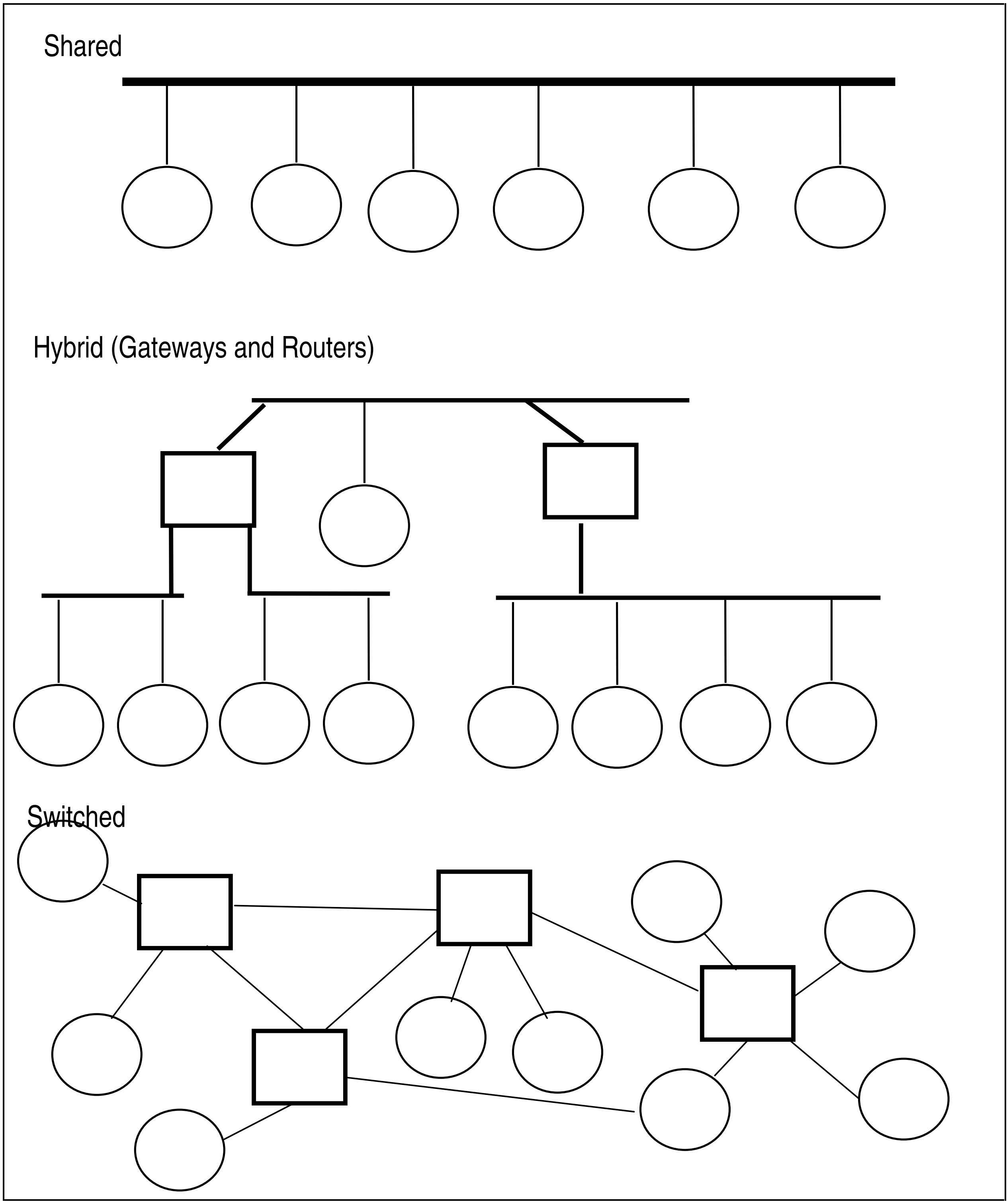
- ♦ cheap
- ♦ reliable (no active components)

DA with shared

- ♦ Poor performance, not scalable
- ♦ Hard to upgrade (we're still using 1980 Ethernet)

<compare aggregate bandwidth for switched and shared bus>

As a result, evolving towards switched LANs over last 5 years



extreme case -- each machine connected to a switch

evolved through intermediate steps

bridges – connect LANs together, passing traffic from one side to the other depending on the addresses in

the packets
- ♦ operates at the Ethernet protocol level
- ♦ usually simpler and cheaper than routers

routers/gateways – connect LANs to WANs or WANs to WANs
- ♦ generally slower than bridges
- ♦ operate at IP level
- ♦ divide interconnect into separate smaller subnets (simplifies management and improves security)

Switched LAN

e.g. ATM, switched ethernet

goal: higher performance, scalability than bus

challenges – cost, reliability

****

Admin - 3 min

****

Sermon3: SW Engineering = Craft

Sign ups for project presentation

Switches

Switch Design
- Routing
- Buffering
- Flow Control

Switch design: Routing

3 ways to specify destination:

1) destination address

→ each router needs map from here to all destinations

“**routing table**” at each switch

e.g. IP

versions

1) deterministic – always follow same path

<destination> → <output port>

2) adaptive – pick different paths to avoid congestion

<destination> → <output port, cost>

3) randomized – pick from among several good paths to balance network load

<destination> → <output port>, <output port>, …

2) virtual circuit

Step 1: establish circuit (using higher level protocol)

Fixed path from source to destination

Step 2: send packets

switch has mapping

virtual circuit → output port

VC important b/c used in ATM

Advantage v. destination address

- Smaller destination address fields
- Use circuit setup to reserve resources
  → good for multimedia

3) source routing

- Source machine puts route in header
  <switch 1, output port 1>
  <switch 2, output port 2>
  <switch 3, output port 3>
  …

- Simple switch
  mapping:
  <output port> → <output port>

Evaluation

- Cheap, fast switch
    - Complexity (mapping route) happens at hosts
      → Good for tracking technology

- Works for small networks
    - All hosts know all hosts

For all of the above:

subtle distributed algorithms for discovering (deadlock free) routes in changing topology

Switch design: Buffering at switches

Problem – on Ethernet, source knows it can't send to destination when line is busy
on switch, several sources can try to send to same destination

<picture 2:1 source:dest>

→ need buffering at switch

What happens when buffer fills?

- Discard packet
    - dangerous: react to congestion by sending more data

positive feedback – higher-level protocols react to
lost packet by resending data
→ reaction to congested network is to send more
- data into network

- Flow control: send fewer packets
    - Don’t send packet unless there is a buffer for it
        - “back-pressure”
    - 2 methods
        - credit-based
        - signal congestion by discarding packet

- Tech trends
    - Memory capacity improving as fast as signaling technology
    - Buffer size = round-trip-time * bandwidth
    - Buffer size = queue length needed to avoid drops with specified probability given expected burstiness

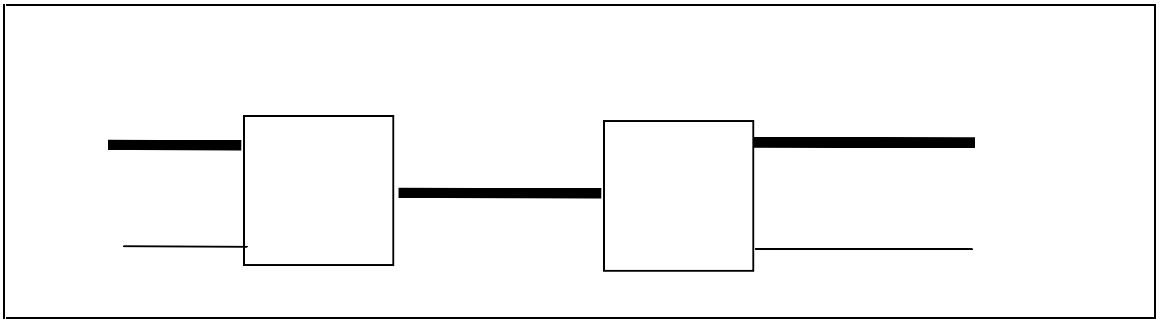
Flow Control

Goal: Minimize buffering

- Avoid dropped packets
- Minimize latency
    - Buffered packets slow other packets
        - SJF scheduling v. FIFO
        - Head-of-line blocking

Design rules

- Avoid bursts to get good latency and bandwidth
    - Queuing theory v. pipeline
- Exponential backoff needed once network congested
    - easier to overflow network than to empty it
        - analogy—rush hour traffic
- “Social cost” of congestion
    - My packets slow down other packets



- Send overhead < recv overhead
  - Delay in send loop can speed up whole network
  - Brewer et al “How to get good performance from the CM-5 data network” http.cs.berkeley.edu/~brewer

Switch design: Store and forward v. cut-through

Store and forward
    each switch waits for full packet to arrive before it is sent
    to next switch

Cut-through / worm hole routing
    switch examines the header, decides where to send the message
    and starts forwarding it immediately

    worm hole – when head of message is blocked, message stays
        strung out over network potentially blocking other messages

    cut through – tail can continue when head is blocked (requires a
        buffer large enough to hold the largest packet)

Store and forward v. cut though
    store and forward simpler control
    cut through – less buffer memory needed?

Latency end-to-end
  store&forward: numbe of switches * size of packet
  cut through: number of switches * header size
    + packet size / net BW

Latency – interference
  little packets have to wait for big ones
  ~shortest job first CPU schduling

Compromise: small packets
  e.g. ATM

  ATM = multimedia -> latency important

Switch topologies

Factors
- degree – number of links from a node
- diameter – max # links crossed between nodes
- avg distance – number of hops to random destination
- bisection – minimum number of links that separate the network into two halves

These factors relate to higher level properties
- latency – diameter, distance
- bandwidth – bisection
- cost – degree (larger degree increases cost per switch and reduces number of switches)

Warnings against beautiful topologies
1) 3-d or N-d drawings must be mapped onto chip and boards
   - elegant when sketched on blackboard may be awkward to build from chips, cables, boards, and boxes

2) subtlety – routing

up*down* routing leads to symmetries → all packets try
to go through same link

e.g. 2-d mesh (see slide)

3) Simple, fast v. beautiful, slow
4) Behavior "in the limit" not terribly relevant
- Biggest machine = 2048 processors
- Most machines < 32 processors

Switch topology: Reliability
----
another consideration – how many nodes become disconnected when a switch fails? How many switches must fail to partition the network?

Solution – redundant connections, careful topologies

****
Summary - 1 min
****