Lecture #32

Experiment – do this lecture with overhead transparancies

Review

NI questions

- Where to put it?
- How much smarts on NI?

3 strategies

1) Avoid the OS
2) Collapse Layered Protocol
3) Eliminate WAN overhead in LAN

SW NI: Fast Sockets case study

Software Interface

Source: LogP Quantified: The Case for Low-Overhead Local Area Networks K.Keeton et al Hot Interconnects III Aug 1995

Raw HW Performance

Network Interface	Switch	Link BW	Switch Latency
AMD Lance Ethernet	Baynetwork Ethercell	10 Mb/s	52 us
Fore SBA-200 ATM	Fore ASX-200	155 Mb/s	13 us
Myricom Myrinet	Myricom Myrinet	640 Mb/s	0.5 us

- Myrinet example of “System Area Network”
- shorter → wider, faster, cheaper
- short distance → source-based routing → simpler switches
- Tandem also building SAN
- BW advantage: 60x; switch advantage 10x

by any measure the hardware is more than an order of magnitude faster than Ethernet

UDP Performance:

Network	UDP Roundtrip
AMD Lance Ethernet	1009 + 2.18*N us
Fore SBA-200 ATM	1285 + 0.32 * N us
Myricom Myrinet	1443 + 0.36*N us

- Worse latency
- 7x bandwidth

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

Application Performance: Database

Network	Avg TPS	TPS v. E	BW v. E	UDP(50) v. E
Ethernet	77 tps	1.00	1.00	1.00
ATM	67 tps	0.87	15	0.80
Myrinet	66 tps	0.85	64	0.78

- Numbe of Transactions Per Second (TPS) for DebitCredit Benchmark; front end to server with entire database in main memory (256 MB)
- Each Transaction → 4 messages via TCP/IP
- DebitCredit Message Sizes < 200 bytes

3 Problems w. traditional SW interface:

1) Go through OS
2) Layered protocol
3) Same protocol for LAN and WAN

1) Go through OS

Traditional Interface:

system call: send(data)

1) Context switch to kernel
2) copy data into kernel buffer (several times)

kernel mem mgmt optimized for space not speed

-- manipulate MBUFs

3) copy data from kernel buffer to NI

New approach:

NI mapped into user program address space

→ Avoid context switch

→ avoid copies

→ simpler memory management

(more memory available)

(avoid multi-layer)

Challenges

- Protection – Multiplexing

  many processes must share NI

  → map different parts of NI into different user address spaces

- Protection – distributed security

  NI must tag data with sender's address (machine, processID, userID)

2) Layered protocol

Traditional Interface:

Ethernet
IP
TCP
Application

New approach: Single-layer – application-to-application

Argue – multi-layer has NOT resulted in plug-and-play
building blocks
Avoid copies
Simpler memory mgmt

Challenges:

- Retransmission – NI must store copy somewhere
- software engineering – more complex protocol?

3) Avoid WAN protocol

costs of WAN: more complex retransmission, timeout, flow control, packet sizing,

Challenges:

single interface regardless of destination

Example: Fast Sockets

(USENIX 1997 – Steve Rodrigues, Dave Culler, Tom Anderson)

User-level library implementation of TCP interface on top of AM

	RTT (latency)	BW: 8 KB	200 Byte RTT
Fast Sockets	58 us	32 MB/s	80 us
TCP/IP	533 us	24 MB/s	580 us

Outline

Beyond the NI

- Media
- Bus v. switched networks
- Switches
  - Routing
  - Buffering
  - Store & Forward v. Cut through
- Topologies
- Connections
  - Flow Control
- Network Summary

Preview

- Multiprocessors
- Projects

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

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

Engineering to avoid data corruption:

Reduce error rate by

- Going Slower
- Shorter Wires
- More Error Correction (redundant data)

What about dropped packets?

- Hard problem

Bus v. switched networks

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

Arbitration – who gets to talk?

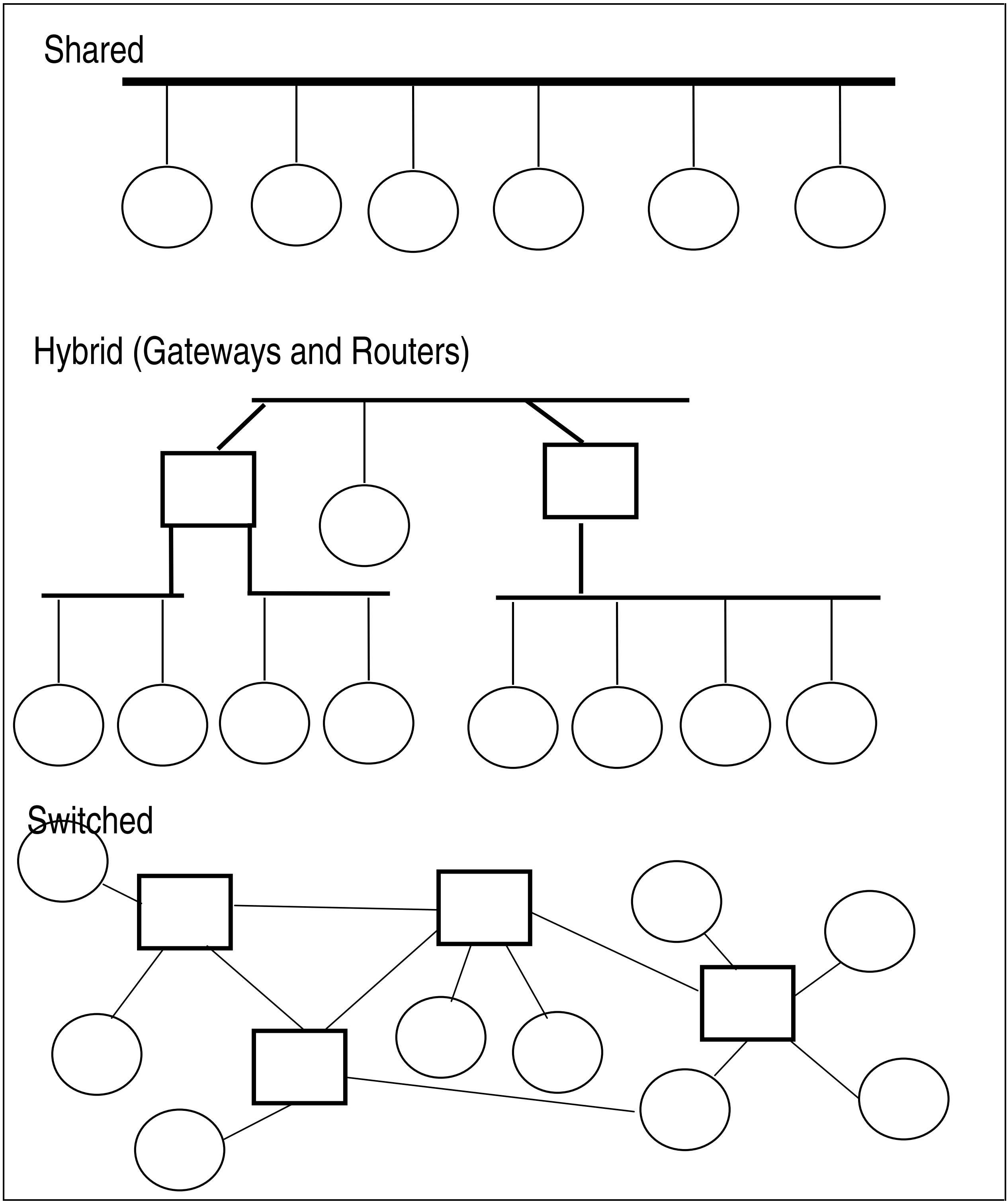
- On bus – bus controller
- Controller not work on LAN
  - No extra wires
  - Who gets to be arbiter?

3-pronged attack

1) Carrier sensing
2) Collision detection
3) Random, exponential backoff

Question: Why random?
Question: Why exponential?

Switched Networks



Switches

Switch Design

- Routing
- Buffering
- Flow Control

Switches: Routing

3 Ways to specify Destination

1) Destination address

Routing table at each switch

3 strategies

1) Deterministic

destination → <output port>

2) Adaptive

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

3) Randomized

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

Routing (cont)

2) Virtual circuit

Step 1: establish circuit

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

Routing (cont)

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

Switches: Buffering

Problems with buffering

What happens when buffer fills?

- Discard packet
    - dangerous: react to congestion by sending more data
        - Positive feedback loop – unstable
- 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

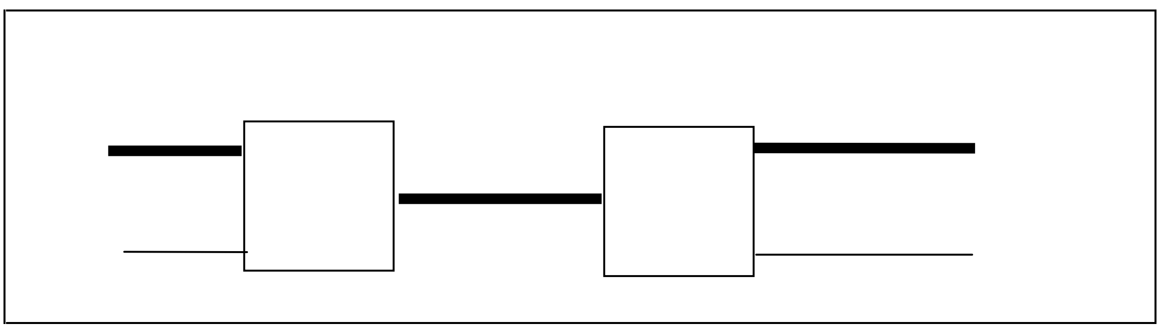
Flow Control

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

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
- Redundancy – how many switch failures to partition the network? How many nodes disconnected when 1 switch fails?

Topologies (cont)

Factors above relate to higher level properties

- latency – diameter, distance
- bandwidth – bisection
- cost – degree (larger degree increases cost per switch and reduces number of switches)
- reliability

Avg. Latency v. Bisection Bandwidth

- NI overhead >> Switch Latency

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

- e.g 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

Network Summary

- Protocols allow heterogeneous networking
- Protocols allow operation in presence of failures
- Amdahl’s law – bandwidth useless if overhead is high
  - Hardware – bring NI close to processor (but not too close)
  - Software – current protocols too heavywieght for high-performance LAN or SAN
    - OS
    - Internetworking protocols used as LAN protocols

      →Large overhead for LAN
- Integrated circuit revolutionizing processor networks as well as processors
  - Switch = specialized computer