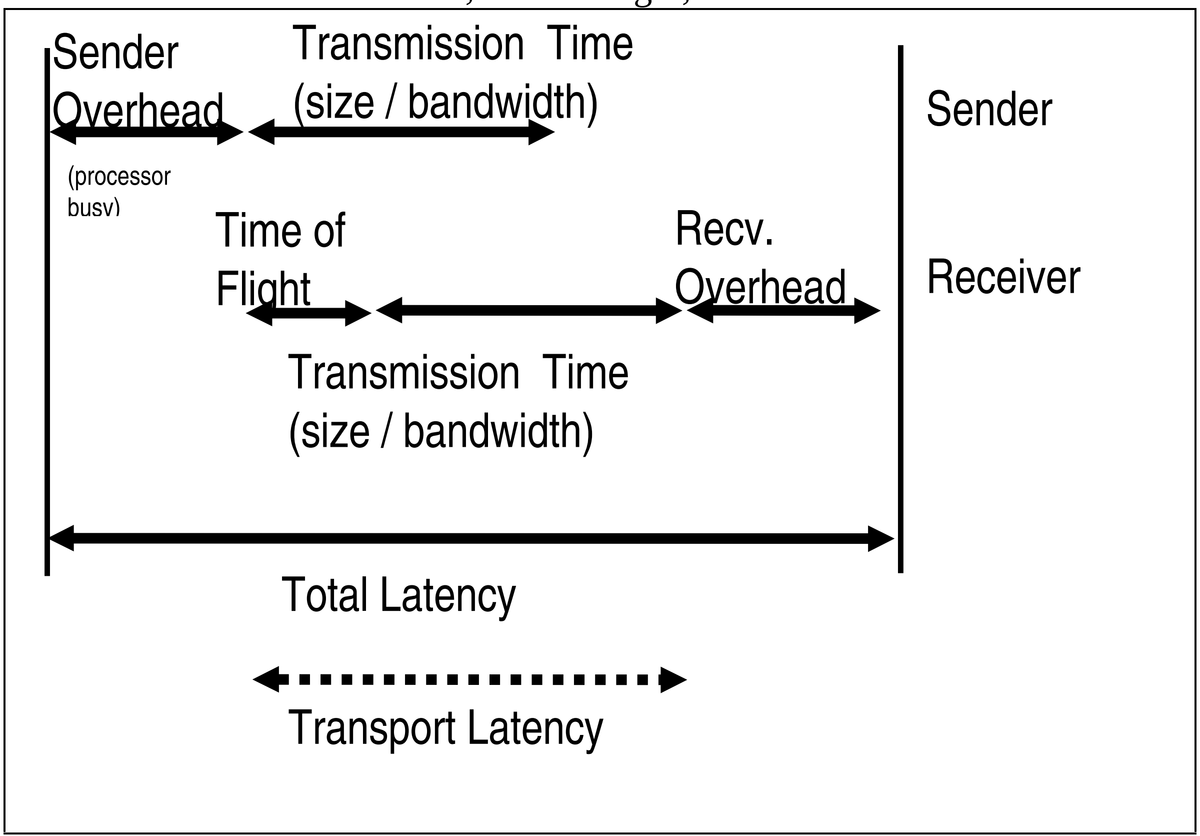
Lecture #30

****
Review -- 1 min
****
layered protocols; headers, trailers

Performance Metrics: ovehead, time of flight, bandwidth



Example: NFS performance ATM v. Ethernet

ovherhead nearly ovewhelms improved bandwidth

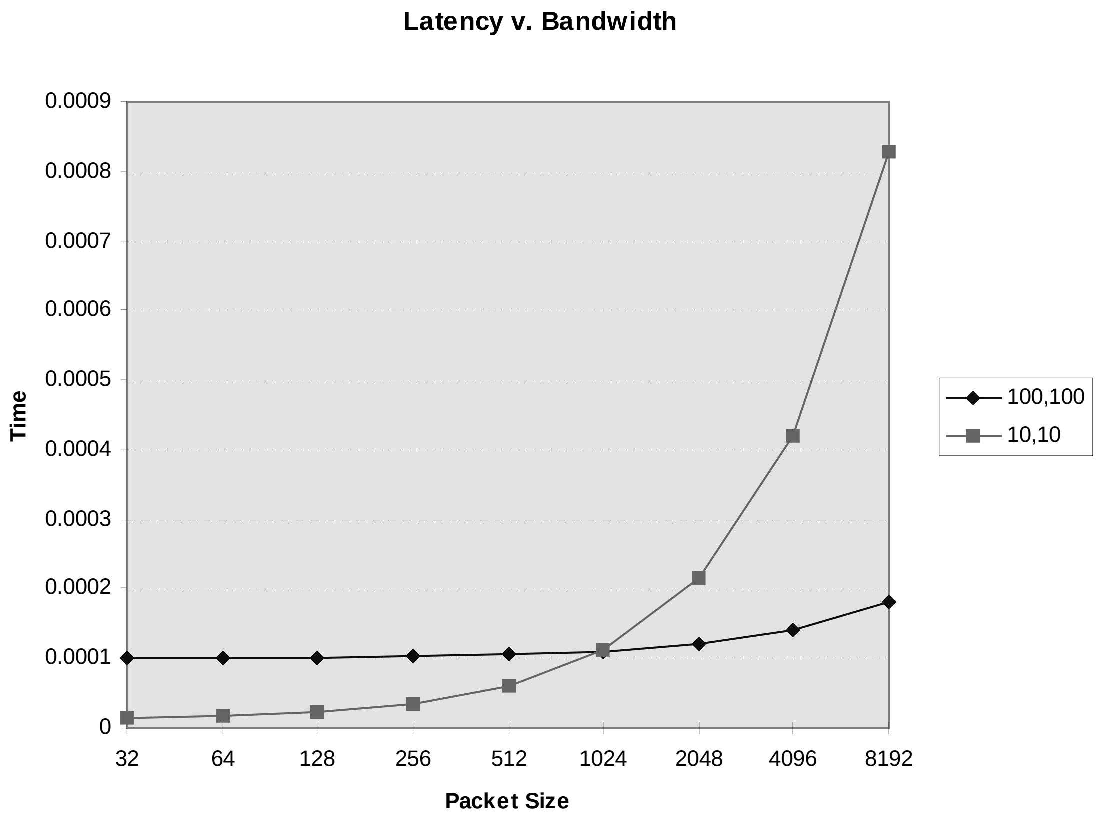
Lesson: 95% of NFS messages small

→ need many small messages to set up a bit one (common pattern)

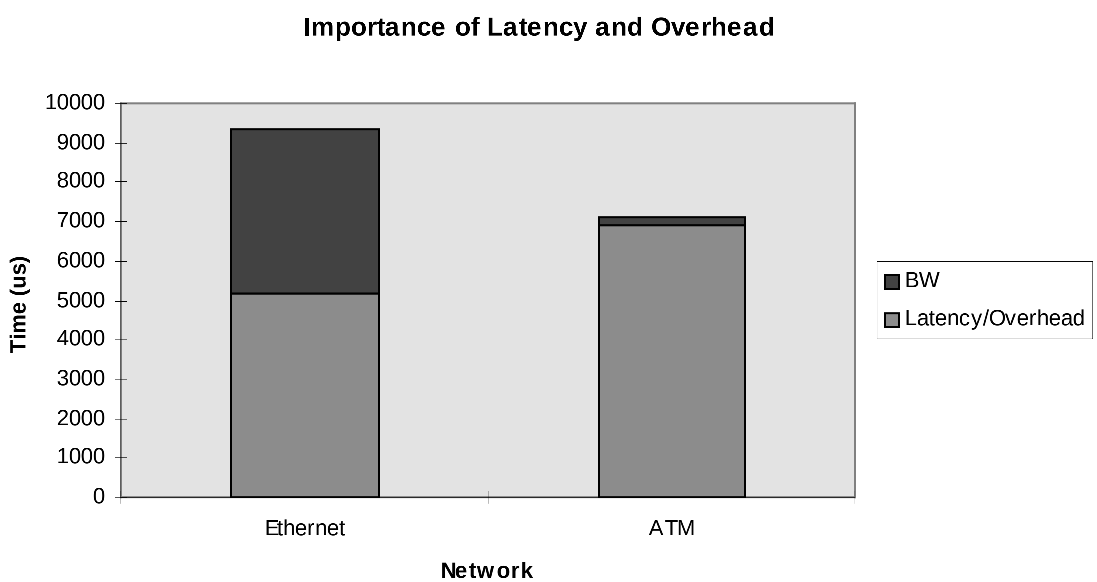
Example of importance of overhead

----

100 MB/s network with 100us overhead v. 10MB/s network with 10us overhead:



Example – intel paragon had a high bandwidth network (40 MB/s?),
but software overhead was > 100us → needed to send huge amounts of data to get within 10% of peak overhead



Study: NSF trace over 1 week: 95 msgs < 200 bytes

Ethernet: 9Mbit/s BW; 456us overhead
ATM Synoptics: 78Mbit/s BW; 626us overhead
→ ATM's 8x better bandwidth → 20% better performance
(latency predicts performance better than BW!)

Moral: bandwidth is not correct measure of network performance (like MIPS)

Net BW ~ CPU MIPS

****
Outline - 1 min
****
Network Interface
    HW
    Software

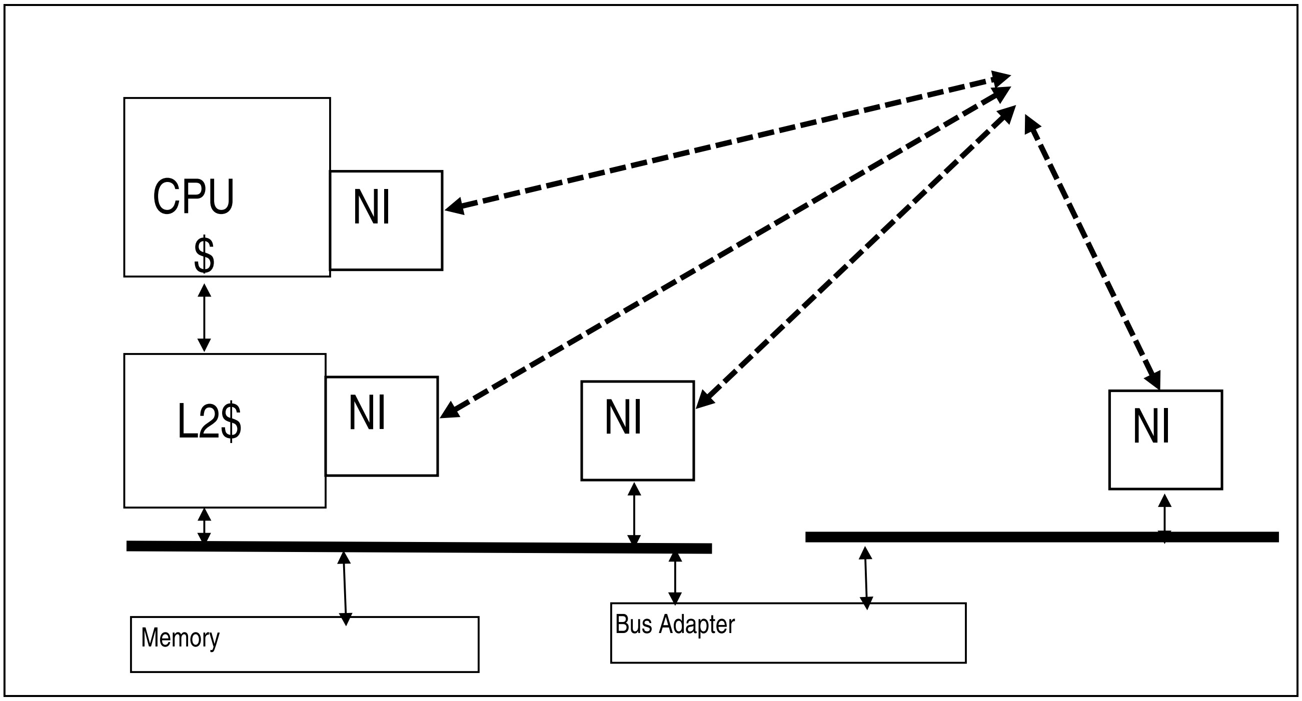
****
Preview - 1 min
****
Media
Topologies
Routing
Connections
Flow control

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

HW Interfaces

Where to connect?
DMA v. PIO?

Where to connect network to computer



Where to put NI?
QUESTION: What are advantages and disadvantages of each of those locations (on CPU/L2/Mem bus/IO Bus)???

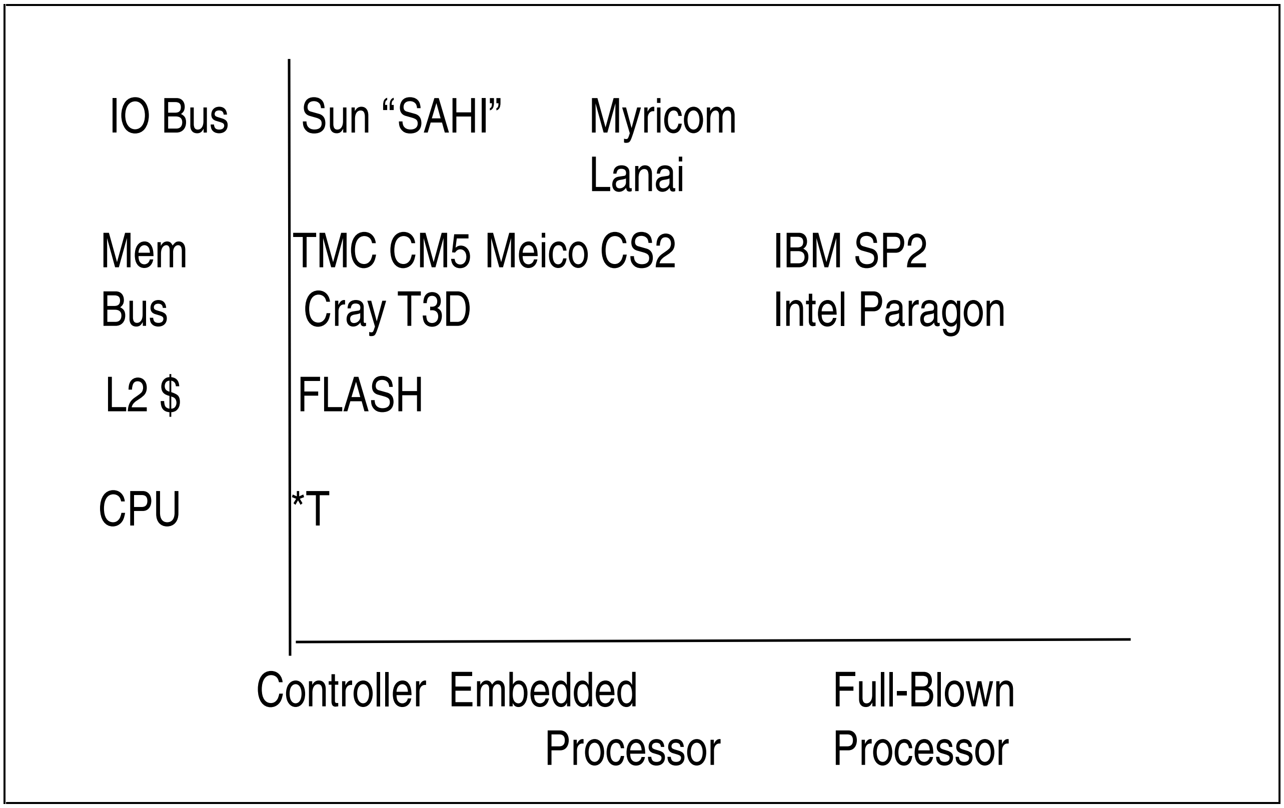
Goal: high bandwidth, low latency, standards

Basic trade-off – closer to CPU is better performance but harder to build

harder --- electrically more challenging

-- more tied to specific processor

What exists – almost all of the above



Don’t underestimate difficulty of getting near processor

*T – never built (Motorola chip design changed)

FLASH – (4 years – 4 node machine started working 2 weeks ago (late March 1998)

Given software overhead – not clear if worth it…

One problem: I/O getting farther from processor over time

SPEC – fast memory hierarchy

E.g. 40MHz SS-2 v. 50MHz SS-20 (no L2) v. 50 MHz SS20 (with L2)

SS2: combined memory, I/O bus → 200ns latency to NI

SS20, no L2$: 2 busses → 500ns to NI

SS-20, L2$ → 1000ns

Other issues
1) How specialized hardware (controller v full processor)
QUESTION: What are trade-offs?

2) Interface: PIO v. DMA
    PIO – best for low latency
    DMA – best for large messages

3) Interface: interrupt v. polling

    FIGURE 7-8

4) Avoid Operating System!!

****
Admin - 3 min
****
comments on checkpoints

Sermon: Performance tuning

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

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

Problem: TCP/IP, UDP go through OS
    same protocol to send data across room as across country
→ too much overhead

Counter-example – active messages (Berkeley), U-net (Cornell), Fast Messages (Illinois)
example: U-net on Fore ATM
Send overhead: 33us
Bandwidth: 120 Mbit/s (= 155 mbit/s - ATM/AAL5 headers)

Main idea: Map network interface into program address space

Traditional Interface:

system call: send(data)

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

New interface: Map NI into user space
send(data)
→ dma data from user space into NI

Challenges

1) Protection – NI must tag data with sender address
2) retransmission – NI must store a copy somewhere
3) multiplexing – many processes must share NI
4) multi-layer – deal with complex protocol but adds copies
5) complex memory management (deal with multi-layer; kernel buffer mgmt routines space (not time) efficient)

Fast Sockets
(USENIX 1997 – Steve Rodrigues, Dave Culler, Tom Anderson)
User-level library implementation of TCP interface on top of AM (TCP/AM)

Results: Myrinet

	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

Fast Sockets (TCP/AM):

****

Summary - 1 min

****

HW and SW Interface affects Overhead, Latency, Bandwidth
Network Overhead violates Amdahl’s Law