Can Google Route?

Building a High-Speed Switch
from Commodity Hardware

Guido Appenzeller, Matthew Holliman
Q2/2002

Outline

- Motiavation: Cost of switches
- The Basic Unit
- Switching Fabrics
- Analysis & Reality Check

Price Survey of Gigabit Switches

Apples and Oranges

PC Gigabit Ethernet Card

- $34.99 (32 Bit)
- $49.35 (64 Bit)

Layer 2

- 24 Port Switch $2,200
- 16 Port Switch $1,500
- 8 Port Switch $830
- 4 Port Switch $460

Layer 3

- 24 Port Switch $7,467

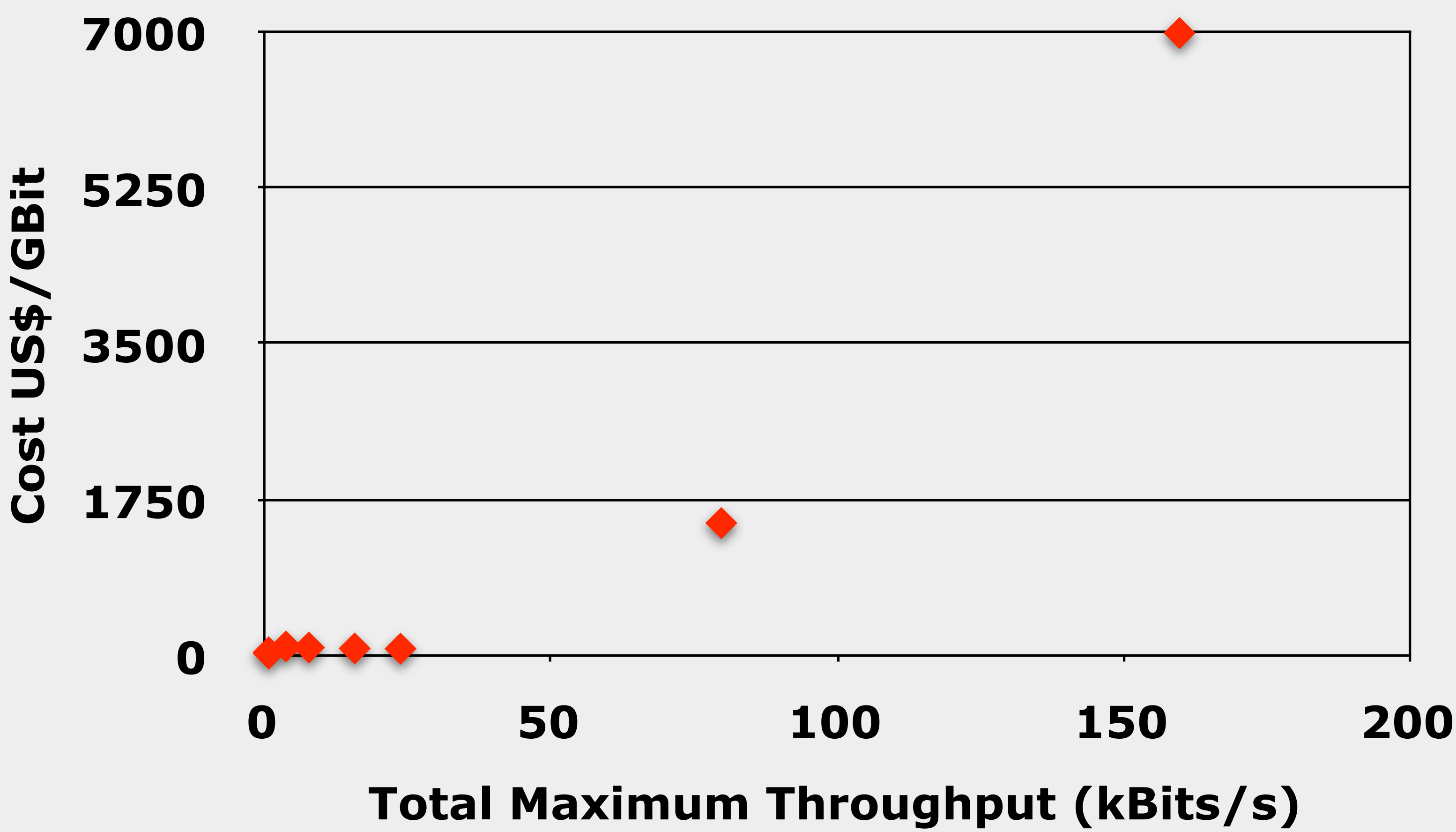
Routers (High End)

- CISCO 7600
  Max 15-80 Gbit
  Base price $60k
  Line Cards $10k
  Total $100k-250k???

- CISCO 12400
  Max 160 Gbit
  Base Price $120k
  Line Cards $25k-$50k
  Total $300-$1m???

Cost/Gigabit vs. Total Switching Capacity

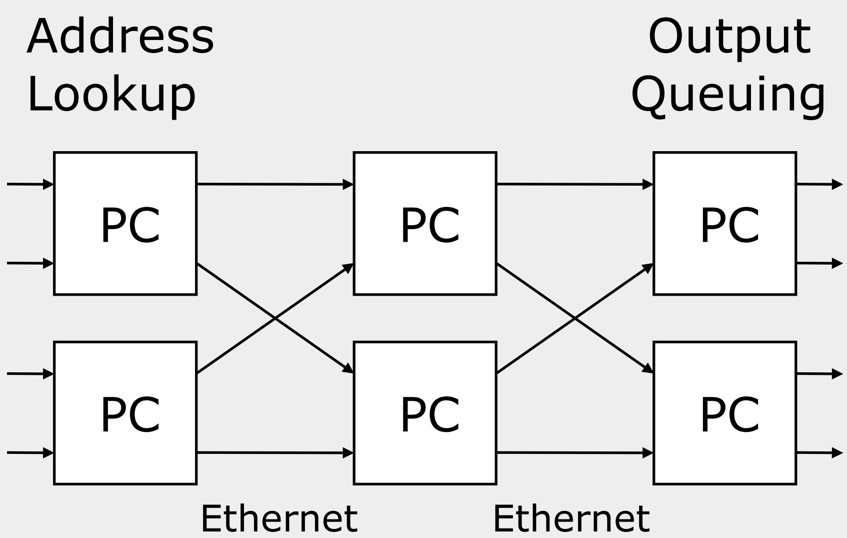
Cost/Gigabit dramatically increases with aggregate speed



Let’s build a Switch the Google way

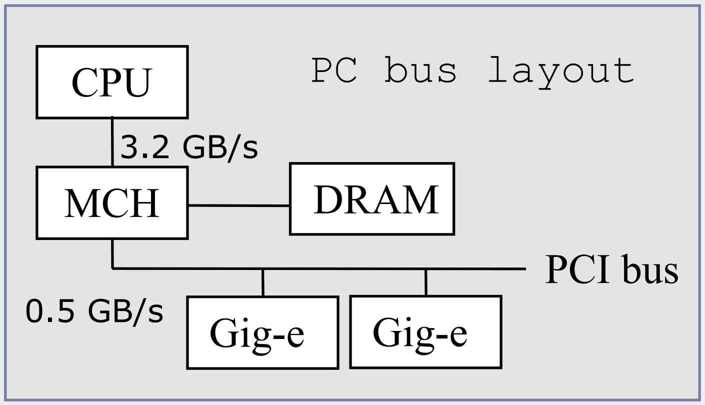
Use large numbers of cheap PC components

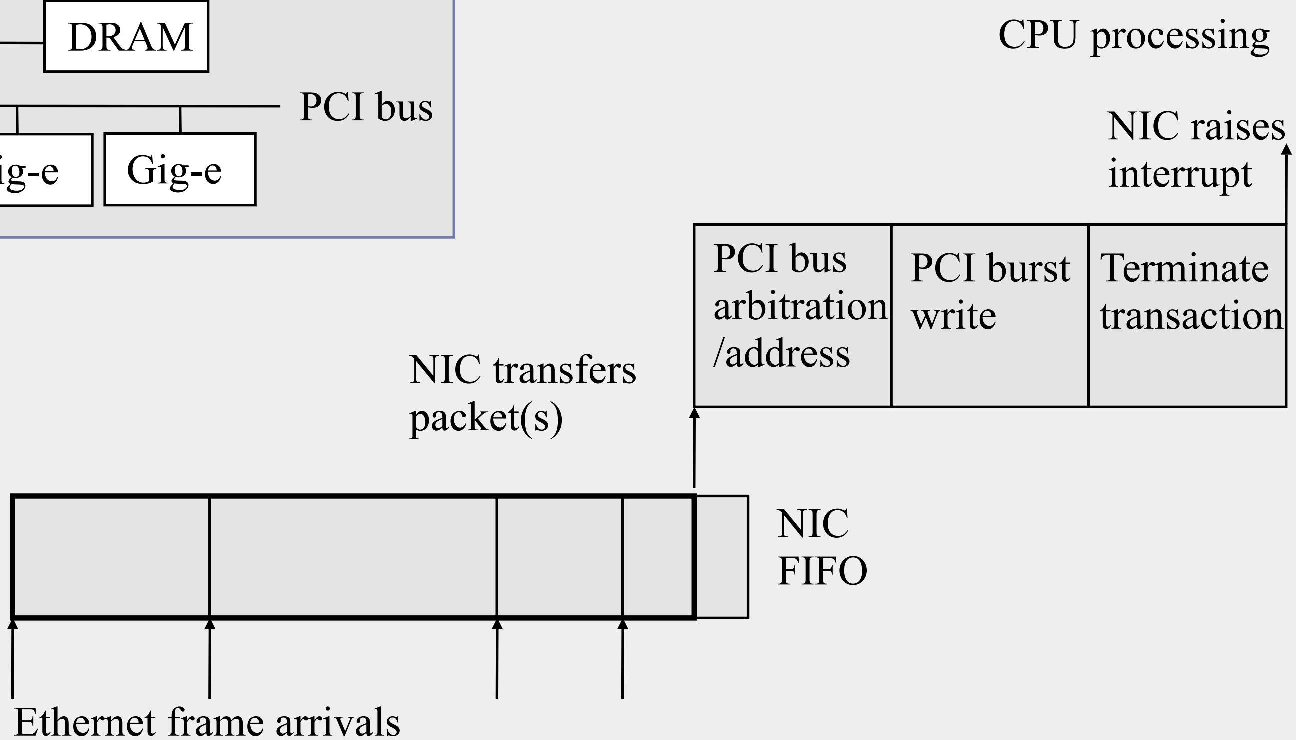
- Use cheap PC boards ($250)
  - 16 MBytes of memory
  - No Disk
- Use cheap copper Gigabit Ethernet Cards
- Use Clos Networks to build larger fabrics out of smaller ones



PC Data Flow

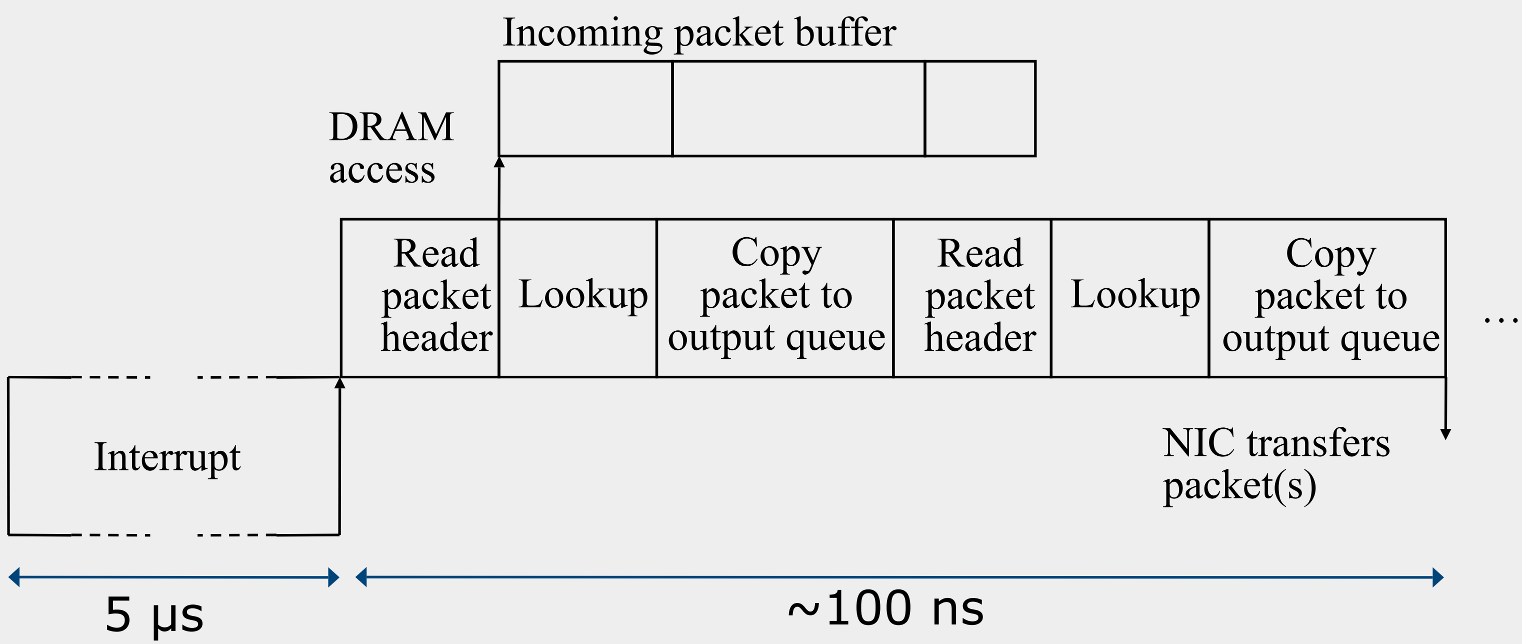
The PCI bus is the bottleneck





Computational Flow

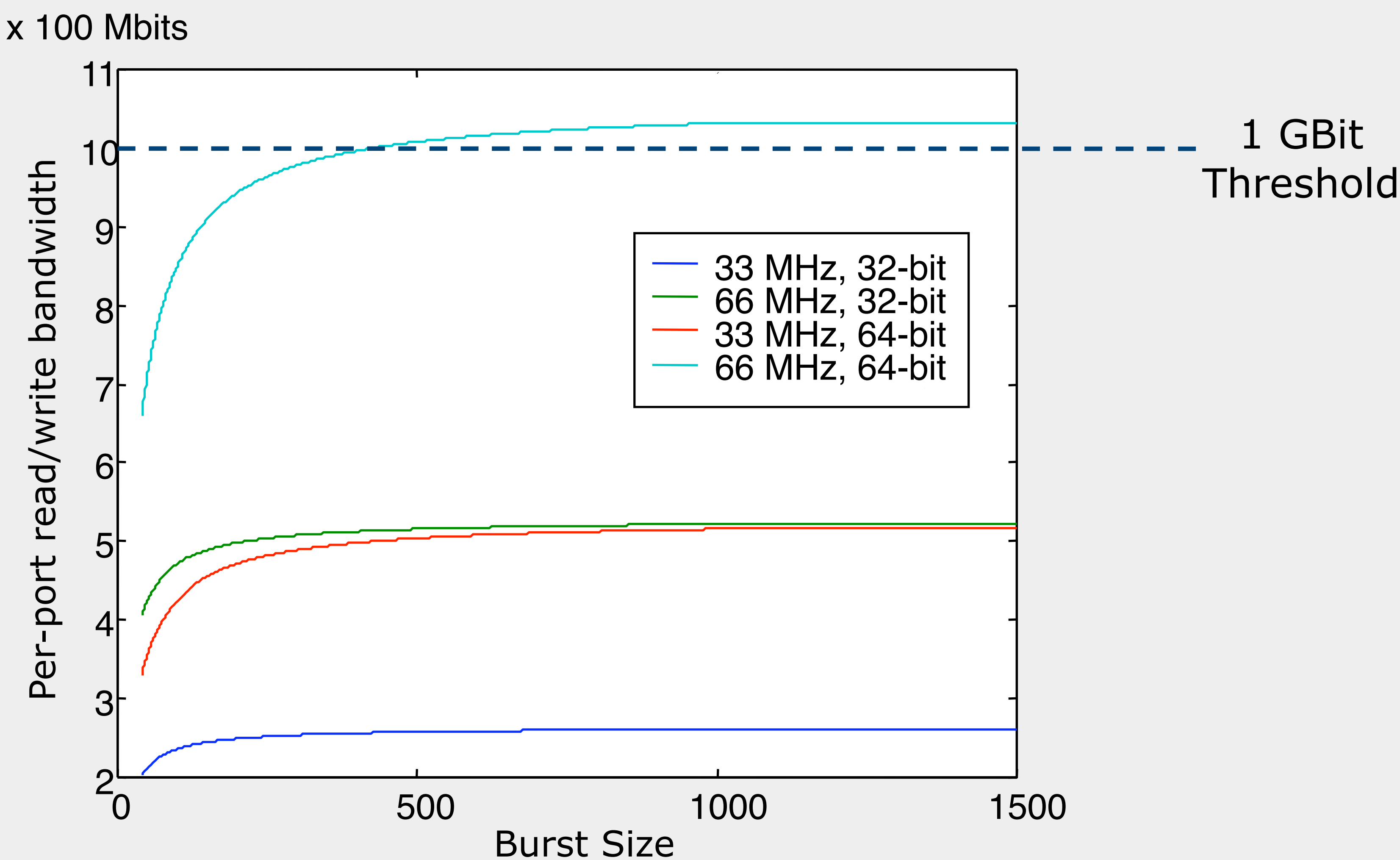
Interrupts are a bottle neck for short packets



- Packet processing is done from/to DRAM
- Packets are written from to network cards in bursts to save IRQ overhead and PCI bandwidth

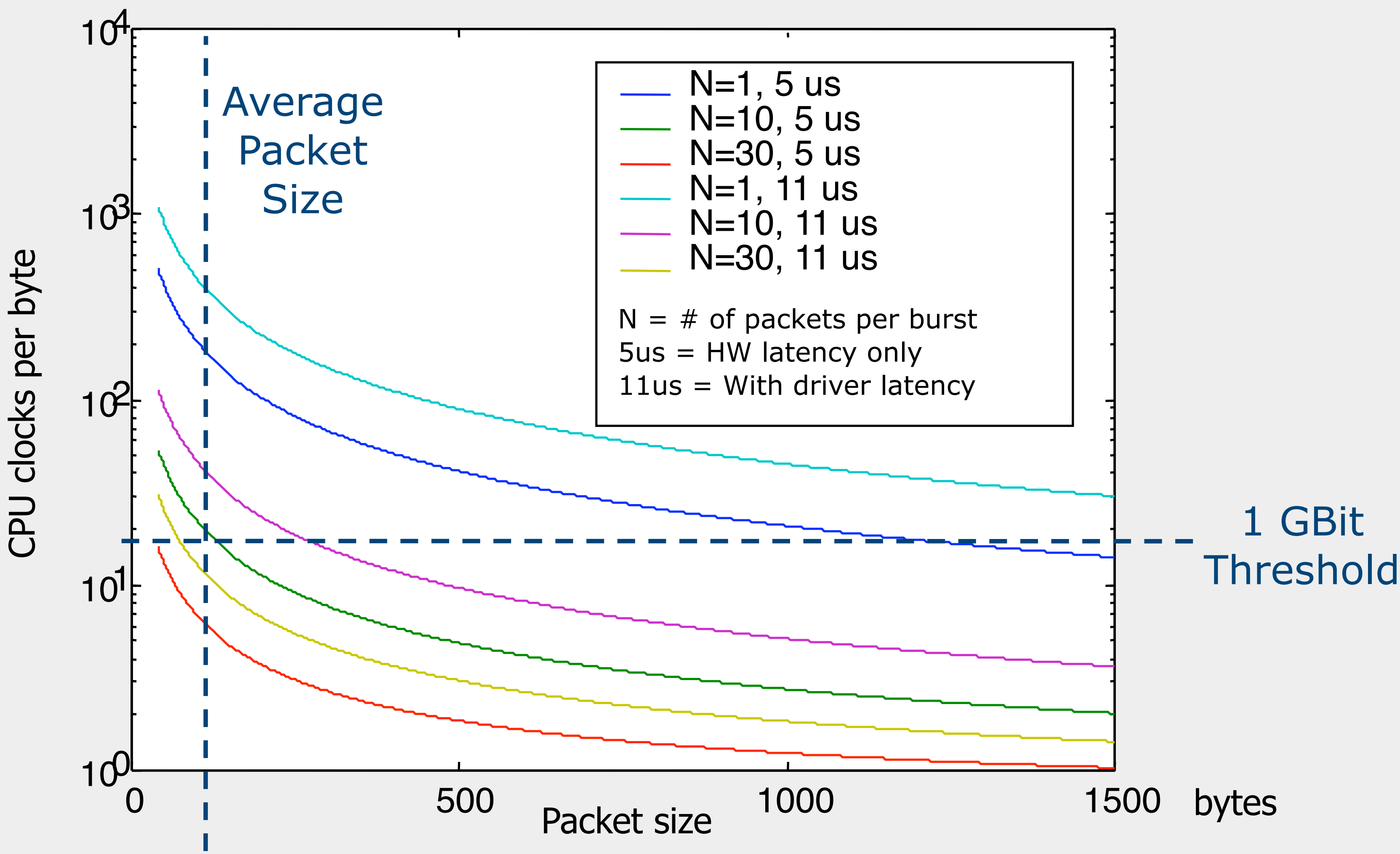
Per Port Throughput vs. Burst Size

We need 66Mhz, 64-bit system



CPU clock cycles per byte vs. Packet Size

For 100%throughput we need to aggregate short packets



PC Performance Summary

Today's PCs are *just* fast enough to operate as a 4x4 switch

- To build a 4x4 half duplex (2x2 full duplex) switch we need:
  - 66 MHz/64 Bit PCI bus
  - 1 Gbyte/s Memory Bandwidth
  - NIC must have sufficient buffers to aggregate short packets to bursts (about 2kBytes)

- Software has to run w/o interrupts
  - e.g. Linux in halted mode

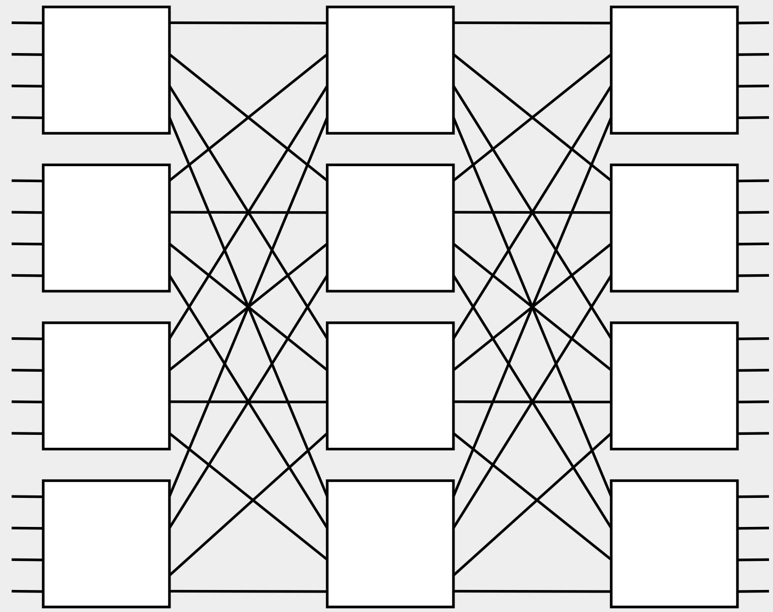
Building larger Switches

Clos Network for an 16x16 Switch made of 4x4 Switches

Requires

- 12 PCs
- 48 Network Cards
- 8 GBit capacity
- Total cost: $5400

➡ 3Com is cheaper



Building larger Switches

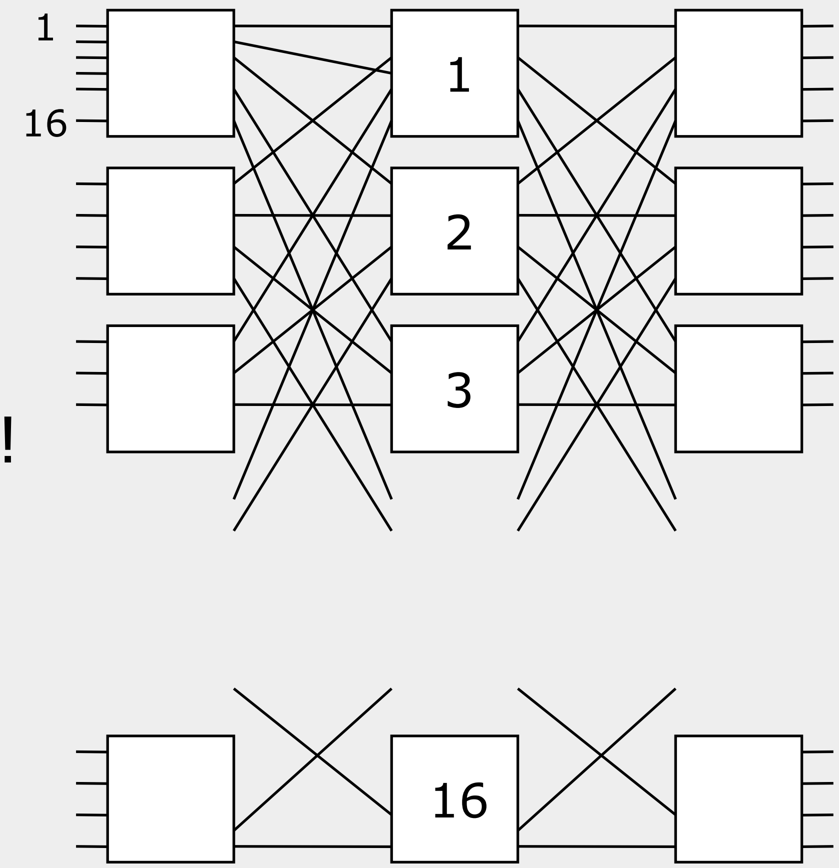
Clos Network for an 256x256 Switch out of 16x16 switches

Requires

- 576 PCs
- 2304 network cards
- 128 Gbit capacity
- Total cost: $260k

➡ Now we are cheaper!

Well, sort of…



Switch size vs. Basic Units Needed

Scaling is slightly worse than n log(n)

- How many 4x4 switches do we need for an NxN switch?
  - 4 x 4 — 1 switch
  - 16 x 16 — 12 switches
  - 256 x 256 — 576 switches
  - $4^{2\hat{}n} \times 4^{2\hat{}n}$ — $3^n 4^{2\hat{}n}/4$ switches
- General:
  - N x N needs *$(N/4)\log_4 N\,(1.5)$^$\log_2\log_4 N$* switches
- Could you build a switch with less basic units
  - Maybe, but not much
  - Theoretical limit is *$O(\,(N/4)\log_4 N\,)$*
  - Differing term *(1.5)^$\log_2\log_4 N$* is small

Scheduling – The Problem

How do we do scheduling?

- For n=k Clos Network we need dynamic matching
  - For 256x256 alogrithm is time-consuming
- How to pass traffic information between inputs, scheduler and nodes
  - More network connections are costly
  - Timing critical

Solution: Buffered Clos Networks

Two ideas:

1. Add a randomization stage (Chang et. al.)
   - Now we can use round robin as scheduler
   - This is deterministic and requires no synchronization between nodes
2. Use the PC's capability to buffer data
   - Each node has a few Mbytes
   - If there is a collision re-send packets

- We use randomization

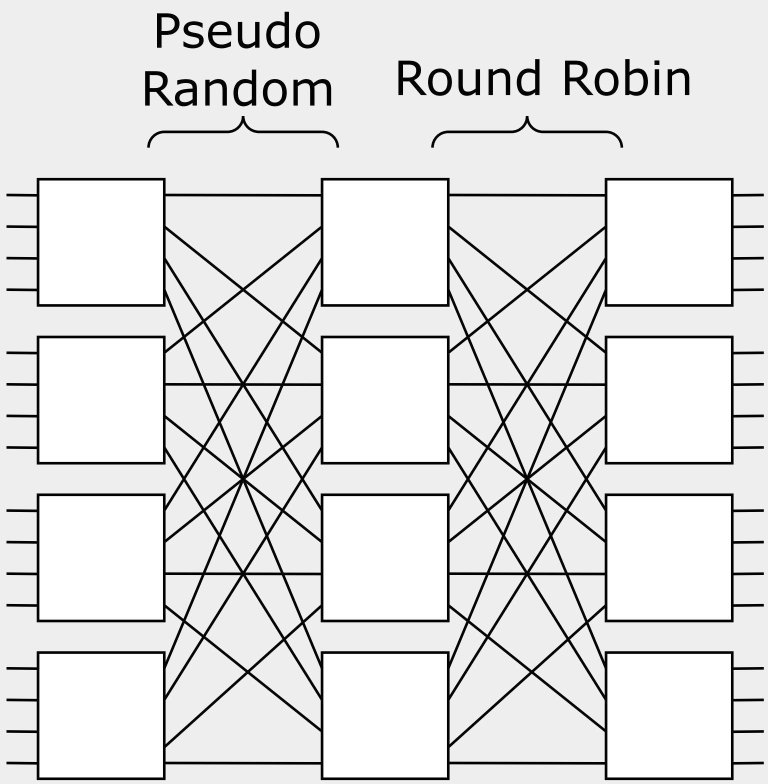
Randomized, Buffering Clos Network

Stage 1:

- Pseudo Random
  (no coordination needed)
- Never blocks

Stage 2:

- Round Robin
  (no coordination needed)
- Never blocks.



Stability Analysis of the Switch

- First stage Analysis
  - Matching is random, distribution of packets on middle column of nodes is I.I.D. Uniform
  - No blocking can occur
  - Queue length at the middle stage is equivalent to an IQR with k inputs, VOQs and Round Robin scheduling
    - We know such a IQR has 100% throughput under I.I.D. Uniform traffic
- Second stage
  - No blocking can occur, 100% throughput if all VOQs in middle stage are occupied
  - Queue length at the middle stage is equivalent to an output queued router with k inputs.
    - Output queued router has 100% throughput

➡ ~~System has 100% throughput for any admissible traffic~~

Reality Check

- This might look like a good idea…
  - Cheap
  - Scalable – switch can grow
  - Some Redundancy - node failure reduces throughput by 1/16 worst case
- …but is probably not exactly what Carriers want
  - High power consumption (50 kW vs. 2.5 kW)
  - Major space requirements (10-20 racks)
  - Packet reordering can happen (TCP won’t like it)
  - Maintenance – One PC will fail per day!

Research Outlook

Why this could still be interesting

- We can do this in hardware
  - Implement in VLSI
  - Build from chipsets that 24x24 switch manufacturers use.
- We could use better base units
  - E.g. 1.15 TBit half duplex (fastest in the world?)
    - 576x576 using 24x24 Netgear switches (GS524T)
    - Cost: $158k
    - (We might get a volume discount from Netgear)
- So far we don’t use intelligence of the nodes
  - We can re-configure matchings periodically
  - Distribute lookup

Backup

Randomized/HoQ Buffering Clos Network

Stage 1:

- Pseudo Random (no coordination needed)
- Never blocks

Stage 2:

- Head of Queue (no coordination needed)
- If it blocks, buffer packet and resend.

Note: We can overprovision middle layer!