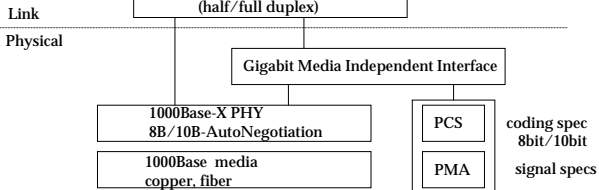


Gigabit Ethernet

- GigE and 10GigE technology overview
- Woo Feng's 10GigE performance analysis paper

Ethernet

- most commonly deployed interconnection technology
- from ~3Mbps in its original deployment (1973, Xerox PARC, Metcalfe and Boggs) to xGbps -> always considered high-performance
 - LAN, MAN, WAN deployment
- addresses physical and link ISO layers:
 - physical – medium and signaling specifications
 - copper, fiber, wireless
 - link – framing of data, medium access, flow control...



Gigabit Ethernet

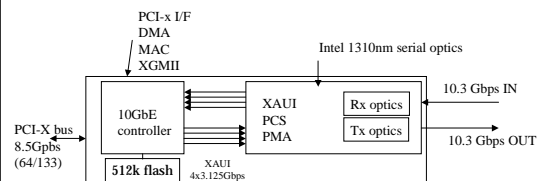
- 1000Mbps, compatible with 10/100Mbps
- Shared access enhancements
 - increase min size to 512bytes for CD
 - frame bursts – pad first frame only, transmit consecutive frames in a burst
 - (bandwidth utilization)
- Flow control via Pause protocol
 - ‘stop’ and ‘go’ pause frames
- Auto Negotiation protocol
 - rate, half/full duplex, ...

10Gb Ethernet

- 10Gbps, full-duplex, over fiber only now
 - backward compatible except for half-duplex mode
- Independent Attachment Unit interface
 - 74-signal wide media interface
- Coding sublayer
 - 64/66 bit encoding
- media dependent layer
 - link distances, wavelengths, multiplexing...
- exists in local/man nets, or in WANs

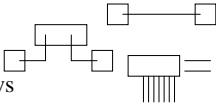
Case Study: Optimizing 10GbE

- test use Intel PRO/10GbE adapter



Experiments

- bandwidth and latency
 - direct flow
 - indirect flow
 - multiple flows
- look for bottlenecks through incremental optimization



Tools

- Iperf & NTTCP – bulk data transfers
 - (amount of data, or time for # of fixed size packets)
- NetPipe – ping-pong latency and bandwidth
- STREAM – measures memory bandwidth
- MAGNET – profile packets from TCP stack
- tcpdump, loadavg...

Bandwidth

- Stock tests (adjusted window size to BDP)
 - 1.8Gbps for 1.5k, 2.7 Gbps for 9k
 - is CPU bottleneck?
 - dip in jumbo frames curve
- Better PCI burst size
 - increase burst size 8x, for 9k frames 30% increase =>

- Uniprocessor kernel -> why?
 - for 1.5k 20% improvement
 - for 9k, similar peak, increased avg through increase for smaller pkts
- Increasing window size
 - should not improve performance
 - but for 1.5k -> 2.47Gbps, 9k -> 3.9Gbps
 - the dip disappears

where is the bottleneck?

- MTU tuning
 - 16000B - ~4.09Gbps
 - 8160B - 4.11Gbps
 - why?

- PCI – not
- CPU – not
- Memory – not
- => I/O latency and CPU's ability to move data to/from device

- Latency

- minimal ~19us
- roughly linear increase with packet size
- disable interrupts saves ~5us for min = 12us