Models
- Must know “the rules of the game”
  - Networking (protocols)
  - Hardware
  - Architecture
  - Operating systems
- Simple models that have explanatory and predictive power
  - Leave details to the specialist

Protocols
- Interfaces
  - Used by applications or higher level protocols
- Packet (message) format
- State machine
  - Describes valid behaviors
- Examples
  - IP
  - TCP, UDP

Protocols (continued)
- Typical protocol operations
  - Data manipulation
  - Demultiplexing
  - Looking up state
  - Set timers and receive alarms
- Our focus: the time consuming functions
- Measures of performance
  - Throughput
  - Latency

The Internet in early 2005
- Backbone link speeds: 10 Gbps and 40 Gbps
- Latencies large, e.g. 100 ms for wide area
- Backbone traffic highly aggregated
- Most traffic TCP (web, peer to peer)
- Small transfers (10 packets) are common
  - ~ half of packets 40 bytes, many 1500 bytes
  - Takes 8 ns to send 40 bytes at 40 Gbps
- Malicious traffic can be large
  - SYN floods, scans

Models
- Must know “the rules of the game”
  - Networking (protocols)
  - Hardware
  - Architecture
  - Operating systems
- Simple models that have explanatory and predictive power
  - Leave details to the specialist
Hardware

- Combinatorial logic
- Memories
- Chips

Combinatorial logic

- A function from digital inputs to digital outputs
- Wires, transistors, resistors, capacitors
- Gates: AND, NAND, NOT, etc. (60 picoseconds)
- More complex blocks: adders, multiplexers, etc.
- Different designs for same boolean function offer different tradeoffs
  - Time for the output to stabilize
  - Number of transistors
  - Power dissipation $P=CV^2F$

Memories

- Registers (0.5-1 nanoseconds)
- SRAM (on-chip 1-2ns, 5-10ns off-chip; ≤64M)
- DRAM (latency 40-60ns, 100ns; ≤1G)
  - Cheaper, cooler, larger than SRAM
  - Page mode (access other words in row buffer)

Interleaved DRAMs

- Increase DRAM throughput (same latency)
- While Bank 1 works, send addresses to Bank 2, etc.
- If consecutive accesses to different banks, bandwidth increased by a factor of B

Chips

- Pipelining increases throughput, not latency
  - Work separated into $n$ successive operations
  - Chip separated into $n$ pipeline stages
  - Each pipeline stage performs its task in 1 cycle
- Important chip limitations
  - Power (30 Watts/cm²)
  - Pin count (≤1,000)
  - Number of transistors

Example: pin count limitations

- Router with five 10 Gbps interfaces
- Memory bandwidth to write packets to buffers and read them $2\times5\times10$ Gbps + 100% overhead = 200 Gbps
- RDRAM with 16 banks – 1.6 GBps=13Gbps
- Need 200/13=16 RDRAMs, each with a data bus of width 64 and 25 address lines
- Total number of pins $90\times16=1,440>1,000$
Models

- Must know “the rules of the game”
  - Networking (protocols)
  - Hardware
  - Architecture
  - Operating systems
- Simple models that have explanatory and predictive power
- Leave details to the specialist

Two popular architectures

- Endnodes and low-end routers
  - Optimized for general purpose computation
- Routers
  - Optimized for Internet communication
- Optical switches handle “wavelengths”
  - Circuit switching technology
  - Optics not only for links, but inside switch too

Models

- Must know “the rules of the game”
- Networking (protocols)
- Hardware
- Architecture
- Operating systems
- Simple models that have explanatory and predictive power
- Leave details to the specialist

Endnode architecture

- CPU
- MMU/CACHE
- MEMORY
- SYSTEM BUS
- NETWORK INTERFACE
- BUS ADAPTOR
- I/O BUS

Router architecture

- Major tasks
  - IP lookup & classification
  - Switching
  - Queuing
- Less time critical tasks
  - Header validation
  - Route computations
  - Protocol processing
  - Fragmentation

Why do I need an O.S.?

- It offers abstractions that:
  - Make it easier to write applications
  - Make it easier to protect apps from each other
- What are the most important abstractions?
  - Uninterrupted computation via processes
  - Infinite memory via virtual memory
  - Simple I/O via system calls
O.S. overhead

- What is the cost we pay for the abstractions?
  - Context switch overheads
  - System call overheads
  - Multiple copies of data
  - Less control over what actually happens
- Tricks to reduce the cost
  - Threads
  - Cisco’s IOS uses a single address space

Example: livelock in BSD