CS228A: Models

George Varghese

January 9, 2002
A rather small set of key concepts is enough. Only by learning the essence of each topic, and by carrying along the least amount of mental baggage at each step, will the student emerge with a good overall understanding of the subject.

— Carver Mead and Lynn Conway
Goals of Models

- Before you can play with improvements, you need to know the rules of the game.

- But network algorithmics uses four separate areas: protocols, architecture, OS and algorithms. Innovations occur when experts work together to create synergy.

- **Question:** How can a logic designer understand protocol issues, and how can a clever algorithm designer understand hardware tradeoffs?

- **Answer:** Using *simple* models that yet have *explanatory* and *predictive* power
Outline of Models

• Network Protocols
  – Transport Protocols
  – Routing Protocols
  – Protocol Model
  – Protocol Measures

• Hardware

• Computer Architecture

• Operating Systems.
- Presents illusion of two shared data queues in each direction (connection) despite network unreliability using seq numbers and retransmission.

- Connection between local queues called ports (i.e., extensions) at IP addresses (i.e., phone numbers). Well known port numbers like 80 for web traffic. UDP, ports but no reliability.
Routing Protocols

- **Forwarding**: Router determines next router/hop to send a packet based on a forwarding table (or default route). Speed-critical.

- **Routing**: Routers work together to build forwarding table. *Distance Vector (RIP)*: exchange distance estimates. *Link state (OSPF)*: Exchange local topology, then use Dijkstra. *Path Vector (BGP)*: exchange path estimates, compute routes based on policy.
Abstract Model of Protocol

- Scenarios cannot specify all executions of protocol. Need state machines and message formats. RFCs. (e.g., TCP = 793). Example transitions: receiving SYN-ACK (TCP) and higher sequence number LSP (OSPF).

- Examples of copying (TCP: from application to kernel, forwarding: from input link to output link via a switch), lookups (TCP: connection state, routing: longest matching prefix)
Measures

- **Throughput**: number of jobs completed per second (i.e., in a network, number of packets delivered per second). Focus of industry. ISPs wish to maximize.
- **Latency**: time (typically worst-case) to complete a job (i.e., time for a packet to go from one end to another and back). Ideal interactive time 100 msec (today more like 250 msec across U.S). Affects distributed computation.
Performance Observations

• **Link Speeds:** Backbone (OC-12, 10Gbps to OC-48, 40 Gbps). Local: Gigabit Ethernet. Wireless, modems slower but improving (HDR etc.)

• **TCP and Web Dominance:** Web (70%) and TCP (90%)

• **Small Transfers:** 50% of accessed files are 50 Kbytes or less.

• **Poor Latencies:** 241 msec average compared to less than 30 msec within U.S.

• **Poor Locality:** 1 million concurrent packet header types (flows) in backbone. 5 packets to one destination.

• **Wire Speed Forwarding:** Half of packets are 40-bytes. Avoid losing important packets in a stream of small packets.
Critical Measures and Tools

• Global versus local: global (e.g., end-to-end throughput) versus local (workstation throughput). Network managers care about global but we focus on local performance of boxes (routers, workstations, bridges) which enable but do not suffice to guarantee global performance.

• Tools: Most network management tools such as HP’s OpenView deal with global measures. Local measures need tools for profiling software (e.g., Rational’s Quantify, cache performance tools such as Intel’s VTune) and even hardware oscilloscopes. Network monitors such as Snoop and tcpdump are also useful.
Case Study 1: Protocols affect Performance

- **Setting:** Large data centers connect disks and computers with a *Storage Area Network (SAN)* to enable disk sharing. Fiberchannel today (expensive).

- **Market force:** iSCSI (Internet Storage) protocols: replace FiberChannel protocols and hardware with (hopefully cheaper) TCP/IP protocols and hardware.

- **Fiberchannel features:** Single SCSI command can READ 10 Mbytes of data from a remote disk without per-packet processing by requesting computer or disk. Network interfaces implements a Fiber Channel transport protocol in hardware.

- **Easy implications:** Must implement TCP in hardware, must add a iSCSI header with length field to mark SCSI boundaries over TCP.

- **Harder implication:** SCSI messages C1 and C2, C2 arrives before C1, Fiberchannel allows steering C2 to a preallocated SCSI buffer but TCP semantics require buffering C2 till C1 with additional later copying overhead. Length field framing fails (why?). Purists vs. pragmatists. Protocol semantics affect performance!
Hardware Models

Hardware design needed for critical functions at gigabit speeds. Classical division:

- **Combinational Logic:** (e.g., logic to implement computation such as a multiply or a router lookup)
  - From transistors to gates.
  - Intuition on timing and power.
  - Higher level building blocks (e.g., standard cells).

- **Memories**
  - Simple memoryies (registers, SRAM, DRAM)
  - Fancier memory tricks (page mode, interleaved memories).
CMOS Transistor Model

- Three terminals (gate, source, drain). When a input voltage $I$ applied to gate, source-drain path conducts. Voltage-controlled switch.

- Inverter: Connect drain to power supply and source to ground (0 volts). What happens when $I$ is high? When $I$ is low? Resistance missing.
From Transistors to Logic

- Inverter generalizes to a NAND gate of inputs $I_1$ and $I_2$ using 2 transistors whose source-drain paths are in series. NOR by placing paths in parallel.

- Boolean Algebra tells us that we can implement any Boolean function $f(I_1, \ldots, I_n)$ of $n$ inputs using CNF or DNF. Logic minimization crucial to reduce transistors. From $O = I_1 \cdot I_2 + I_1 \cdot \overline{I_2}$ to $O = I_1$. 
Example 1, Quality of Service and Priority Encoders

- Router maintains $n$ output packet queues for a link, where queue $i$ has higher priority than queue $j$ if $i < j$.

- Scheduler maintains a $N$-bit vector (bitmap) $I$ such that $I[j] = 1$ if and only if queue $j$ is non-empty.

- Scheduler can find the highest priority non-empty queue by finding the smallest position in $I$ in which a bit is set. *Priority encoder.*

- Even from what little we have learned, priority encoder is a boolean function and can be implemented by logic for small $N!$ MMOL. How does delay and logic scale with $N$? Needs a little more insight.
Timing

- 40 byte packet OC-768 speeds, 8 nsec forwarding time. Thus max delay from inputs to outputs on any logic path is 8 nsec.
- In practice, when an input changes from 0 to 1 it takes time for the gate to accumulate charge to allow source-drain path to conduct.
- Modelled as gate input charging a gate capacitor \((C)\) in series with a resistor \((R)\).
- Think of charge as water, voltage as water pressure, capacitance as container size, resistance as friction.
- Formally, the voltage at time \(t\) after the input \(I\) is set to \(V\) is \(V(1 – e^{-t/RC})\). \(RC\) is time constant; within \(RC\), output reaches \(1 – 1/e = 66\%\) of final value.
- More gates to charge, the more effective capacitance. 0.18 micron process, single F04 inverter delay is 60 picoseconds.
- Charging input can cause further outputs to charge further inputs. Thus for a comb function like priority encoder, delay is the worst-case charging and discharging delay across any path of transistors.