Project proposal: Questions to answer

• What are you trying to do? Articulate your goals using no jargon.

• How is it done today, and what are the limits of current practice?

• What is new in your approach? Why would it succeed?

• What are the risks?

• What are the mid-term and final “exams” to check for success?
High-Speed Hardware Switches

Lecture 12, Computer Networks (198:552)
The router data plane

- Data plane implements per-packet decisions
  - On behalf of control & management planes
- Forward packets at high speed
- Manage contention for switch/link resources
Requirements on router data planes

- Speed!

Inherently parallel workload

⇒ Leverage hardware parallelism!
Requirements on router data planes

• Speed!
• Area & footprint
• Power
• Port density
• Programmability
Overview of router functionality

• Different routers are very different
  • Historically evolving, multiple concurrent designs
  • … but there are many commonalities (Ex: MGR, RMT)
• Packet receive/transmit from/to physical interfaces
• Packet and header parsing
• Packet lookup and modification: ingress & egress processing
• High-speed switching fabric to connect different interfaces
• Traffic management: fair sharing, rate limiting, prioritization
• Buffer management: admission into switch memory
Life of a packet
(1) Receive data at line cards

• Circuitry to interface with physical medium: CoAx, optical
  • SerDes/IO modules: serialize/deserialize data from the wire
  • Interfaces just keep getting faster: more parallelism
  • ... but stay the same size

• Multiple network interfaces on a single line card
  • Component detachable from the rest of the switch
  • Ex: upgrade multiple 10 Gbit/s interfaces to 40 Gbit/s in one shot

• Preliminary header processing possible
  • MGR: convert link-layer headers to standard format
(2) Packet parsing

• Extract header fields: branching, looped processing
  • Ex: Determine transport-level protocol based on IP protocol type
  • Ex: Multiple encapsulations of VLAN or MPLS headers
• Outcome: parse graph and data in the parsed regions
• MGR: done in software using bit slicing of header memory
• RMT: programmable packet parsing in hardware
(2) Packet parsing

• Key principle: Separate the packet header and payload
  • Conserve bandwidth for data read/written inside switch!

• Header continues on to packet lookup/modification
• Payload sits on a buffer until router knows what to do with the packet
  • Buffer could be on the ingress line card (MGR)
  • But more commonly a buffer shared between line cards (RMT)
(3) Packet lookup

• Typical structure: Sequence of tables (Ex: L2, L3, ACL tables)
  • Exact match lookup
  • Longest prefix match
  • Wildcard lookups

• Outcome: a (set of) output ports, possible header rewrites
• Wide range of table sizes (# entries) and widths (headers)
• Header modifications possible
  • TTL decrements, IP checksum re-computation
  • Encapsulate/decapsulate tunneling headers (MPLS, NV-GRE, …)
  • MAC source address rewrite

Interesting algorithmic problems!
(3) Packet lookup: *Pipelined parallelism*

- Different functionalities (ex: L2, L3) in different table stages
- Highly parallel over packets (1 packet/stage): high throughput
- Pipeline circuitry *clocked* at a high rate: ex: RMT@1 GHz
- MGR: software with memory access non-determinism
- RMT: deterministic hardware pipeline stages
(3) Packet lookup: Memory layout matters!

- MGR: Cache hierarchy
- Large tertiary/main memory containing full route table
  - ... but far too slow for random access lookup with small delays
- Employ a fast $L1$ route cache
(3) Packet lookup: Memory layout matters!

- RMT: flexible partitioning of memory across SRAM and TCAM
- Numerous fixed size memory *blocks*
- Circuitry for independent block-level access
- Deterministic access times
  - All of it is SRAM or TCAM
- Contrast to MGR (DRAM)
(4) Interconnect/Switching Fabric

• Move headers and packet from one interface to another
• Kinds of fabrics: memory, bus, crossbar
(4) Crossbars: The scheduling problem

- Demands from port i to port j

- Can one utilize fabric capacity regardless of demand pattern?
  - Blocking vs. nonblocking

- Different topology designs
(5) Queueing: Traffic management

- Where should the packets not currently serviced wait?
- Input-queued vs. output-queued
- HOL blocking? Suppose port 1 wants to send to both 2 and 3
  - But port 2 is clogged
  - Port 1’s packets towards port 3 should not be delayed!
(5) Queueing: Traffic Management

• Better to have queues represent output port contention

• Scheduling policies:
  • Fair queueing across ports
  • Strict prioritization of some ports over others
  • Rate limiting per port!
(5) Queueing: Buffer Management

• Typical buffer management: Tail-drop

• How should buffer memory be partitioned across ports?
  • Static partitioning: if port 1 has no packets, don’t drop port 2
  • Shared memory with dynamic partitioning

• However, need to share fairly:
  • If output port 1 is congested, why should port 2 traffic suffer?

• Algorithmic problems in dynamic memory sizing across ports!
(6) Egress processing

• Combine headers with payload for transmission
  • Need to incorporate header modifications
  • … also called “de parsing”

• Multicast: egress-specific packet processing
  • Ex: source MAC address

• Multicast makes almost everything inside the switch (interconnect, queueing, lookups) more complex