A 50 Gb/s IP Router

- Partridge et al. IEEE/ACM Trans Networking
Goals of Multi Gigabit Router

- Need sufficient internal bandwidth to move packets between its interfaces
- Need packet processing power
  - Handle several million packets per second
- Conform to IETF standards
Router

- Collection of NICs
- Bus/connection fabric connecting these NICs
- Software or logic that determines how to route packets among NICs
Gbps Router Design

- Network Processor
- Forwarding Engines
  - Many
  - Each is an Alpha processor
- Line Cards
- Switch Fabric
  - Connecting line cards
Juniper’s Router

S. Venkatesan
Department of Computer Science
Router Front

S. Venkatesan

Department of Computer Science
See Figure 1

Switch

Line card (in)
Line card (in)
Line card (in)

FW Engine

Line card (out)
Line card (out)

Network Processor

FW Engine
Five Novel Elements of Design

1. Multiple FW engines, each FW engine has complete FW table
2. Switched backplane (not bus)
3. FW engine card different from line cards
4. FW engine can receive packets from different link layers
5. Include QoS processing in router
Forwarding Engine

- Has complete set of routing tables
- Switched backplane
  - Not a shared bus
- Forwarding engines distinct from line cards
- FE on separate cards
- QoS processing in router
- No ASIC
  - Use standard CPU
- FW in software
- Use Alpha 21164 CPU
**Alpha Processor (64 bit) RISC**

- 32 registers (64 bit)
- Three internal caches
  - First level
    - Icache 8kB
    - Dcache 8kB; ignored
  - Second level
    - Scache: 96kB on-chip secondary cache
  - Third level
    - Tertiary cache: Several MB (Bcache)
First level Cache

- Icache (Instruction cache);
  - 8kB in size
  - Store 2048 4-Byte (32-bit) long instructions

- Dcache (Data cache 8kB in size)
  - ignored
Second level Cache

- Scache
  - 96kB on-chip [for code+data]
  - 8 cycles to load from scache
  - Cache of recent routes
  - Each route entry is 64 bits long
  - Can store 12,000 entries

- 95% hit rate with 12,000 entries
Third level Cache

- Tertiary cache
  - Several MB (Bcache) [2X8MB in this case]
    - Divided into two parts
      - One being used [has complete forwarding table]
      - Other for updating (simultaneously)
        » Invalidate all Scache entries when network processor is done with updating 2nd part
FW engine

- Complete packet stays in inbound line card
- Only header is sent to FW engine via switch fabric
Forwarding Operation

- Headers reach FW engine
  - FIFO queue
- Read front of FIFO queue
- Examine header, determine how to route
  (consult FW table)
- Make writes (instructions to inbound/outbound cards)
- Update header
- Send updated header, FW instructions to inbound card back
FW operation continued

- Inbound card receives updated header/instructions
- Assembles modified IP packet (payload still in inbound card)
- Uses instructions to send to outbound line card via switch
FW operation

- Few hundred lines of code
  - About 85 executed in common case
  - Not less than 42 cycles (9.8 MPPS per FW engine)
    - Executes in quad
  - No checking of IP checksum; update it
    - 17 instructions to check
      - 21% increase in time to check for rare error
Stages of path through code

Stage 1
- Basic error check (header is from IPv4)
- Packet and header lengths are OK
- No IPv4 options?
- Hash offset into router cache, load route
- Start loading next header
Stage 2

- Check if cached route matches destination
  - No? Jump to extended lookup
- Check TTL, update TTL and checksum
- Check if pkt is destined for itself
Stage 3

- Update TTL and checksum in IP header
- Add routing info and layer 2 info to header (including flow classifier)
- No header checksum checking
Network Processor

- Alpha 21064 commercial CPU on PC motherboard with PCI
- Runs 1.1 NetBSD release of Unix