A 50-Gb/s IP Router

Presenter: Sheng Shen
PhD, ECE@UIUC
Internet Keeps Growing Fast

http://www.internetlivestats.com/internet-users/
Supporting the fast-growing internet

Transmission link  Operating system  Switch
Supporting the fast-growing internet

Transmission link
Operating system
Switch

Routers are slow!
What can we do?
Supporting the fast-growing internet

Transmission link  Operating system  Switch
What’s in a Router?

Input Ports → High-speed Switching Fabric → Output Ports

Routing Processor
Summary of Routing Functionality

• Step 1: Get the packet
• Step 2: Look into packet header for destination
• Step 3: Look up routing table for output interface
• Step 4: Modify header
• Step 5: Pass packet to output interface
Summary of Routing Functionality

• Step 1: Get the packet
• Step 2: Look into packet header for destination
• Step 3: Look up routing table for output interface
• Step 4: Modify header
• Step 5: Pass packet to output interface

How can we accelerate this process?
Gb/s, several million packets / second
Major Contributions

Input Ports → High-speed Switching Fabric → Output Ports

Routing Processor
Major Contributions

Input Ports

Forwarding Engine

Contains Forwarding Table
Handles ‘Fast Path’

High-speed Switching Fabric

Routing Processor

Output Ports
Major Contributions

- **High-speed Switching Fabric**
  - Routing Processor
    - Contains Forwarding Table
      - Handles ‘Fast Path’
    - Updates Routing Table
      - Handles ‘Slow Path’
  - Input Ports
  - Output Ports

- **Forwarding Engine**
  - Provides connection between input and output ports through the High-speed Switching Fabric.
Major Contributions

- High-speed Switching Fabric
  - Routing Processor
    - Contains Forwarding Table
      - Handles ‘Fast Path’
    - Updates Routing Table
      - Handles ‘Slow Path’
  - Forwarding Engine
  - Input Ports
  - Output Ports
  - Shared Bus
  - Switched Backplane
Innovation: Forwarding Engine

- Task: deciding where to forward each packet
- Now separated from line cards
- Custom hardware and software
Forwarding Engine: Hardware

- Alpha 21164 Processor
  - 415 MHz, 64 bit, 32 registers
  - 2 integer logical units, 2 float point units
Forwarding Engine: Hardware

• Alpha 21164 Processor
  - 415 MHz, 64 bit, 32 registers
  - 2 integer logical units, 2 float point units
  - 3 internal caches
    - instruction cache
      8kB, fit key routing code
    - data cache
      8kB, did not use
    - secondary cache
      96kB, slower, store ~12000 recent routes
      >95% hit rate
  - 1 external, tertiary cache
    - 16 MB, divided into two 8MB banks
    - one stores complete forwarding table
    - the other updated by the network processor via PCI bus
Forwarding Engine: Hardware Operation
Forwarding Engine: Hardware Operation

Pipelined forwarding: ‘fast path’!
Forwarding Engine: Software

- Hundreds lines of code
  - 85 most frequent instructions, ~42 cycles
- Peak forwarding speed: 9.8 MMPS (million packets per second)
Forwarding Engine: Software

- Hundreds lines of code
  - 85 most frequent instructions, ~42 cycles

- Peak forwarding speed: 9.8 MMPS (million packets per second)

- Fast Path contains 3 stages:
  - Checking
  - Routing
    match the destination with cached routes
    if missed, check complete routing table
  - Updating
    update TTL, checksum, etc. in the header, together with routing information
Forwarding Engine: Software

• Trick: Fast Path doesn’t check IP header checksum
  - instead, simply update it. (The bad is still bad).
  - errors are rare... Save 21% time.
Forwarding Engine: Software

• Trick: Fast Path doesn’t check IP header checksum
  - instead, simply update it. (The bad is still bad).
  - errors are rare... Save 21% time.

• What datagrams Fast Path cannot handle?
  - destination not in the cache
  - headers with errors
  - headers with IP options
  - datagrams that require fragmentation
  - multicast datagrams
Let network processor handle them!
Forwarding Engine: Software

• Trick: Fast Path doesn’t check IP header checksum
  - instead, simply update it. (The bad is still bad).
  - errors are rare... Save 21% time.

• What datagrams Fast Path cannot handle?
  - destination not in the cache
  - headers with errors
  - headers with IP options
  - datagrams that require fragmentation
  - multicast datagrams
  Let network processor handle them!

• Common questions
  - Why not use ASIC (application specific integrated circuit)?
    Router deployed in ISP’s backbone... IPv4 constantly evolving, require programmability
  - Is route cache effective?
    Yes. Full route lookup in off-chip memory is 5X more expensive than cache hit.
Innovation: Network Processor

- Task:
  - Updates routing table
  - Handles ‘Slow Path’
Network Processor

• Commercial PC motherboard
  - 21064 Alpha processor, 233 MHz
  - runs 1.1 NetBSD
  - access to line cards through a PCI bridge
Network Processor

• Commercial PC motherboard
  - 21064 Alpha processor, 233 MHz
  - runs 1.1 NetBSD
  - access to line cards through a PCI bridge

• Handles ‘slow path’
Network Processor

- Commercial PC motherboard
  - 21064 Alpha processor, 233 MHz
  - runs 1.1 NetBSD
  - access to line cards through a PCI bridge
- Handles ‘slow path’
- Updates routing table and forwarding table periodically
Network Processor

- Commercial PC motherboard
  - 21064 Alpha processor, 233 MHz
  - runs 1.1 NetBSD
  - access to line cards through a PCI bridge

- Handles ‘slow path’

- Updates routing table and forwarding table periodically
  - keeps complete routing information
    - including hop counts, where route is learned from, etc.
Network Processor

• Commercial PC motherboard
  - 21064 Alpha processor, 233 MHz
  - runs 1.1 NetBSD
  - access to line cards through a PCI bridge

• Handles ‘slow path’

• Updates routing table and forwarding table periodically
  - keeps complete routing information
    including hop counts, where route is learned from, etc.
  - builds forwarding table
    much smaller, only needs next hop information
Network Processor

• Commercial PC motherboard
  - 21064 Alpha processor, 233 MHz
  - runs 1.1 NetBSD
  - access to line cards through a PCI bridge

• Handles ‘slow path’

• Updates routing table and forwarding table periodically
  - keeps complete routing information
    including hop counts, where route is learned from, etc.
  - builds forwarding table
    much smaller, only needs next hop information
  - downloads forwarding table to forwarding engines
    remember: split forwarding memory into two banks
    one being used and the other being updated; then switch
Innovation: Switching Fabric

- Task: move data between function cards

- Input Ports
- Forwarding Engine
- High-speed Switching Fabric
- Output Ports
- Routing Processor
Switching Fabric: Hardware

- Conventional shared bus
  15-port point-to-point switch
Switching Fabric: Hardware

- **Conventional shared bus**
  - 15-port point-to-point switch
  - Pros: parallelism
    - 15 simultaneous data transfers can happen
  - Cons: doesn’t support one-to-many transfer (multicast)
    - simply copy the packets; still faster than shared bus!
Switching Fabric: Hardware

- **Conventional shared bus**
  15-port point-to-point switch
  - Pros: parallelism
    15 simultaneous data transfers can happen
  - Cons: doesn’t support one-to-many transfer (multicast)
    simply copy the packets; still faster than shared bus!

- **Two pin interfaces for each function card**
  - data interface:
    75 input data pins, 75 output data pins, 51.84 MHz
  - allocation interface:
    2 request pins, 2 inhibit pins, 1 input stats pin, 1 output status pin, 25.92 MHz

- **Bandwidth**
  - 1024 bits (data) + 176 bits (control) per transfer cycle
  - 49.77 Gb/s (data) in total
Switching Fabric: Pipelined Switching

- (Phase 1) Source card signals that it has data to send to the destination card
- (Phase 2) Switch allocator decides how to schedule the transfer in phase 4
- (Phase 3) Source and destination line cards are notified that the transfer will take place. Data path cards get ready
- (Phase 4) The transfer takes place.
Switching Fabric: The Allocator

• Task:
  - takes in connections requests from all function cards
  - outputs a transfer configuration

• Configuration space is large
  - input: 15 X 15 = 255 possible parings...
  - output: 15! (1.3 trillion) different ways to transfer
Switching Fabric: The Allocator

• Task:
  - takes in connections requests from all function cards
  - outputs a transfer configuration

• Configuration space is large
  - input: 15 X 15 = 255 possible parings...
  - output: 15! (1.3 trillion) different ways to transfer

• What is a good allocation strategy?
  - fast
    in order to enable pipeline switching
  - fair
    otherwise some ports have more opportunity to transfer than others
The Allocator: Straightforward Solution

• Zig-zag scan
  - Left-to-right, top-to-bottom

• Allocate the connection if the destination is not occupied
The Allocator: Straightforward Solution

- Zig-zag scan
  - Left-to-right, top-to-bottom
- Allocate the connection if the destination is not occupied
- Fairness: ×
  - there is a preference for low-numbered sources
- Speed: ×
  - for 15X15 matrix, need to evaluate 225 positions serially... too slow
The Allocator: New Method

• Fixing the fairness: Random Shuffling
  - randomly shuffle the sources and destination
  - the allocator maintains two 15-entry shuffle arrays for sources and destinations
The Allocator: New Method

• Fixing the fairness: Random Shuffling
  - randomly shuffle the sources and destination
  - the allocator maintains two 15-entry shuffle arrays for sources and destinations

• Fixing the timing problem: Wavefront Evaluation
  - evaluate positions in parallel
  - 225 steps $\rightarrow$ 29 steps!
  - still a little slow...
The Allocator: New Method

• Fixing the fairness: Random Shuffling
  - randomly shuffle the sources and destination
  - the allocator maintains two 15-entry shuffle arrays for sources and destinations

• Fixing the timing problem: Wavefront Evaluation
  - evaluate positions in parallel
  - 225 steps $\rightarrow$ 29 steps!
  - still a little slow...

• Split into groups
  - even faster!
Revisiting High-speed Switching Fabric Routing Processor

- Input Ports
- Output Ports
- High-speed Switching Fabric
- Forwarding Engine
  - Contains Forwarding Table
  - Handles ‘Fast Path’
- Routing Processor
  - Updates Routing Table
  - Handles ‘Slow Path’
- Shared Bus
- Switched Backplane

- Handles ‘Slow Path’
- Contains Forwarding Table
- Handles ‘Fast Path’

Diagram: Connections and labels indicating flow and functionality between components.
Conclusions

• A huge impetus to the router industry...
  - The industry started to build fast core routers in backbone networks

• Shows that examining every packet header is feasible in Gb network
  - desirable for security and robustness

• Shows that router technology is not failing
  - they really made it

• Careful selection and placement of hardware and software is the key
Further Thoughts

• Questions:
  - the network processor ARPs all possible addresses... is it acceptable on
    the Internet backbone?
  - how expensive is such a router? will people buy it?
  - ...

• Can we further accelerate router?
  PacketShader: a GPU-Accelerated Software Router
  - many operations can be done in parallel...
  - utilize GPU’s processing power
  - 4X improvement over existing software router
  - 39Gbps... but remember, this is a software router!
Thank you!