1. High Speed Routers
2. Route lookups
Routers

Cisco 12016: 80 Gbps
Cisco 12416: 320 Gbps
Cisco 12816: 1280 Gbps  Cost: $500K
    Power: 4.2 KW

Juniper M 320
    320 Gbps
    Power 3.2 KW
What do routers do?

- Routing
  - Decide the next hop based on Destination address
  - Cost varies as Table size
- Header modification
  - Decrement TTL, Link layer address for next hop, etc
  - Requires rewriting
- Forwarding
  - Byte movement
    - Move bytes from input interface to output interface
  - Need to keep up with line card speed
Basic Components of a Traditional High Speed router

Control Plane

Datapath” per-packet processing
Forwarding Engine

Packet

payload  header

Router

Routing Lookup Data Structure

Destination Address

Outgoing Port

Forwarding Table

<table>
<thead>
<tr>
<th>Dest-network</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>65.0.0.0/8</td>
<td>3</td>
</tr>
<tr>
<td>128.9.0.0/16</td>
<td>1</td>
</tr>
<tr>
<td>149.12.0.0/19</td>
<td>7</td>
</tr>
</tbody>
</table>
Need for high speed routers

- B/W keeps increasing
- Need to keep line cards fully utilized

<table>
<thead>
<tr>
<th>Line</th>
<th>Linerate (Gbps)</th>
<th>40B (MPPS)</th>
<th>Lookup-speed (nano sec)</th>
<th>84B (MPPS )</th>
<th>354B (MPPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC3</td>
<td>0.155</td>
<td>0.48</td>
<td>2083</td>
<td>0.23</td>
<td>0.054</td>
</tr>
<tr>
<td>OC12</td>
<td>0.622</td>
<td>1.94</td>
<td>515</td>
<td>0.92</td>
<td>0.22</td>
</tr>
<tr>
<td>OC48</td>
<td>2.5</td>
<td>7.81</td>
<td>128</td>
<td>3.72</td>
<td>0.88</td>
</tr>
<tr>
<td>OC192</td>
<td>10.0</td>
<td>31.25</td>
<td>32</td>
<td>14.88</td>
<td>3.53</td>
</tr>
<tr>
<td>OC768</td>
<td>40.0</td>
<td>125</td>
<td>8</td>
<td>59.52</td>
<td>14.12</td>
</tr>
</tbody>
</table>
Hardware

- DRAM access times 50 nsec
- Pricing: DRAM  Yr 2012
  - Retail Price 4GB is $32; 2GB is $20
    - In [Gupta 98] 16 MB for $50 was the price in 1998
- SRAM access times 5 to 10 nsec
- Pricing: SRAM is 40 to 50 times more expensive than DRAM
  - 1 GB $1500; 16 MB $80
First-Generation IP routers

CPU

Buffer Memory

DMA Line Card MAC

DMA Line Card MAC

DMA Line Card MAC
First-generation IP routers

- Shared memory
- Bus is the bottleneck
- Memory r/w speeds is also bottleneck
- Every packet needs two transfers between line cards and memory
  - Crosses the bus twice
- Route table stored in DRAM
- Does not scale to too many line cards
- Suffices for Low speed routers
- < 1 Gbps speed
Second-Generation IP routers

- CPU
- Buffer Memory
- DMA
- MAC
- Route
- Cache

Cache update

Fast Path
Slow Path
Second-generation IP routers

- Each line card has a route table cache
- On a hit, forward directly
  - Fast path – Switching interface
- On a miss, via CPU bus, memory
  - Slow path
- Copy only header, then reconstruct packet on outbound link
  - Buffer packets on cards
- < 5Gbps speed
Third -Generation IP routers

Mckeown 97, Partridge 98
Third generation IP routers

- Multiple forwarding engines
- IP header stripped and given to FEs
- Header processed separately from body
- FE determines outbound header
- Packet reconstructed and moved from Source buffer to destination buffer
- Exploit Parallelism
- Have a separate Data transfer path
Three types of switching fabrics

memory

bus

crossbar
Crossbar arbitration

- 1 FIFO at the input
- Request for any of the N outputs
- What about fairness
- Rotating priority
- Top priority given to the next line following the line which was last serviced
- HOL blocking in simple FIFO
Virtual queues

- Complex arbitration
- $N^2$ input possibilities $M$ output possibilities
- Lots of arbiter schemes
- Decide which of the request from the input queues wins
### M Series Reference Table

<table>
<thead>
<tr>
<th>ROUTER</th>
<th>M71</th>
<th>M10I</th>
<th>M40E</th>
<th>M120</th>
<th>M320</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate half-duplex throughput</td>
<td>10 Gbps/16 Mpps</td>
<td>16 Gbps/16 Mpps</td>
<td>51.2 Gbps/40 Mpps</td>
<td>120 Gbps/90 Mpps</td>
<td>320 Gbps/385 Mpps</td>
</tr>
<tr>
<td>FPC* slots and full duplex throughput per slot</td>
<td>1 built-in, 4 Gbps additional 1 Gbps for FIC</td>
<td>2 built-in, 4 Gbps</td>
<td>8 FPC slots, 3.2 Gbps</td>
<td>4 FPC slots, 10 Gbps</td>
<td>8 FPC slots, 20 Gbps</td>
</tr>
<tr>
<td>PICs** per chassis</td>
<td>4, plus 2 additional fixed FE, or 1 fixed GE ports</td>
<td>8</td>
<td>32</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>Chassis per rack</td>
<td>24</td>
<td>9</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Redundancy</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

---

**Juniper Networks M-Series Routers**
High Speed routers

- Two multi-gigabit routers
  1. A 50-Gbps IP router by Craig Partridge et.al, in ACM TON, 1997
     - Specialized Hardware 3rd Gen IP Router
     - Way ahead of its time
  2. A 40- Gbps IP router by Sangjin Han et.al., in SIGCOMM 2010
     - PacketShader
     - Software router on commodity hardware (CPU + GPU)
  - What has changed in 13 years?
MultiGigabit Router (MGR)

- Separate Switching Back plane fabric
- Distributed architecture
- Multiple forwarding engines
- Forwarding engines determine output line based on header
- Each FE has its own forwarding table and buffer
- Routing and forwarding separation
  - FE determines which outbound line to send the packet
  - Only header moves between line card and FE
  - Packet construction and deconstruction done by line cards
  - Line card uses the switching fabric to forward the packet
- Network Processor, FE processor, Packet Processor
- Only header moves around
- Data is transferred from Input to output lone card via switching backplane
- Two memory banks to handle route updates
- Routing table handled by NP (maintains several routes to D)
  - Routing Information for various destinations
  - Determine active routes based on policy
- Forwarding table handled by FEP (maintains only active route to D)
  - Install active routes for each destination
  - FE cache 16 MB, divided into 8-MB banks (used in active-standby modes)
Packet Processing

- Line card: Packet Buffered in FIFO queue
- Header is removed, passed onto FE
- FE: Read header, lookup, write modified header
- Modified header, with forwarding instructions sent back to line card
- Line card: buffer entire packet for delivery to output line
Packet processing

- Fast path code
- Header check, lookup, update TTL update header
- 42 cycles (each cycle 2.4 nanosec – 415 MHz alpha processor)
- Fast path time: 101.2 nsec
- Packet forwarding rate 9.8 MPPS
Packet Processing

- Slow path
  - Cache miss
  - Header errors
  - Headers with IP options
  - Fragments
  - Multicast
MGR Features

● FE associated with each line card (but separate)
  ● Has its own 415 Mhz Alpha processor and memory
● FEs keep entire routing table as opposed to cache
● Switched backplane as opposed to shared bus
● Switch transfer cycle 0.308 microsec
● Max 15 simultaneous transfer of 1 K bits (approx 50 Gbps)
● Network Processor 233Mhz Alpha processor
  ● Used for writing router updates to the routing table in FEs
  ● Control plane function
Switch arbitration

- Cross bar with FIFO
- Each head of line packet contends for bus
- For fairness priority of input line rotated
- Multiple input buffers
- Row and column arbiter
- Diagonal arbiter
- Wrapped Diagonal arbiter
Switch arbitration

- Wave front arbiter
- On a N x N cross bar, grant request along the diagonal
- Priority given to higher level diagonals
- Rotate priority of diagonals
Forwarding speeds

- Lots of H/W
- Specialized H/W
  - Network Processor
- Switching fabric
- Smart Forwarding (copy only header)
- 3rd generation can match line card speeds
# MGR router: [Partridge 1998]

## Some observations

<table>
<thead>
<tr>
<th>H/W units</th>
<th>1998</th>
<th>2012</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FE Processor</strong></td>
<td>415 Mhz</td>
<td>2.66 GHZ (Intel xeon x5550)</td>
</tr>
<tr>
<td><strong>Network Processor</strong></td>
<td>233 Mhz</td>
<td></td>
</tr>
<tr>
<td><strong>Level 2 cache (on-chip)</strong></td>
<td>96KB</td>
<td>256 KB or 512 KB</td>
</tr>
<tr>
<td><strong>L1 Cache</strong></td>
<td>8KB</td>
<td>64KB</td>
</tr>
<tr>
<td>I-Cache</td>
<td>8KB</td>
<td>64 KB</td>
</tr>
<tr>
<td>D-Cache</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Registers</strong></td>
<td>32 Registers</td>
<td>16 or 32 registers</td>
</tr>
<tr>
<td><strong>Switch</strong></td>
<td>15 Port</td>
<td>?</td>
</tr>
<tr>
<td><strong>Cores</strong></td>
<td>Single</td>
<td>Dual to multi core</td>
</tr>
</tbody>
</table>
What’s happening now

- Then - Specialized hardware -- MGR (1997)
- Now - General purpose hardware (PacketShader 2010-SIGCOMM) - GPU
  - Software based high speed router
  - Software Defined Networking (SDN)
- Trend towards programmability
- Use commodity H/W
- Parallel Computing available in GPUs
- ShaderPrograms – manipulate pixel values for scenes
“Silicon Budget” in CPU and GPU

Xeon X5550:
- 4 cores
- 731M transistors

GTX480:
- 480 cores
- 3,200M transistors
### Per-Packet CPU Cycles for 10G

<table>
<thead>
<tr>
<th>Cycles needed</th>
<th>IPv4</th>
<th>IPv6</th>
<th>IPsec</th>
<th>IPv4 lookup</th>
<th>IPv6 lookup</th>
<th>Encryption and hashing</th>
<th>Total Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet I/O</td>
<td>1,200</td>
<td>1,200</td>
<td>1,200</td>
<td>+ 600</td>
<td>+ 1,600</td>
<td>+ 5,400</td>
<td>1,800 cycles</td>
</tr>
<tr>
<td>IPv4 lookup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2,800 cycles</td>
</tr>
<tr>
<td>IPv6 lookup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6,600 cycles</td>
</tr>
<tr>
<td>Encryption and hashing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Available budget
- **1,400 cycles**
- **10G, min-sized packets, dual quad-core 2.66GHz CPUs**
Our Approach 2: GPU Offloading

- GPU Offloading for
  - Memory-intensive
  - or
  - Compute-intensive operations

- IPv4 lookup: 600
- IPv6 lookup: 1,600
- Encryption and hashing: 5,400
A 40 Gbps router that uses commodity PCs
- Exploit parallelism feature of GPUs by doing lookups on a batch of packets
  - E.g., Xeon CPU has 4 cores
  - E.g., GTX480 GPU has 480 cores
- Basic routing operations offloaded to GPUs
- Rendering a scene is a parallel operation (operations on pixels)
- How is packer routing parallel workload?

**Key Idea**
- Process packets in batches from a large input buffer
- Each packet processing handled by a separate core
- Avoid H/W bottlenecks by partitioning
- Run packet processing operations on independent cores
Parallelism in Packet Processing

- The key insight
  - Stateless packet processing = parallelizable

1. Batching

2. Parallel Processing in GPU
Scaling with a Multi-Core CPU

Device driver → Pre-shader → Post-shader → Device driver

Master core

Worker cores
Results (w/ 64B packets)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Throughput (Gbps)</th>
<th>CPU-only</th>
<th>CPU+GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4</td>
<td>28.2</td>
<td>39.2</td>
<td></td>
</tr>
<tr>
<td>IPv6</td>
<td>8</td>
<td>38.2</td>
<td></td>
</tr>
<tr>
<td>OpenFlow</td>
<td>15.6</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>IPsec</td>
<td>3</td>
<td>10.2</td>
<td></td>
</tr>
</tbody>
</table>

GPU speedup:
- IPv4: 1.4x
- IPv6: 4.8x
- OpenFlow: 2.1x
- IPsec: 3.5x
Packet Processing in Routers is a parallel work load!
- Batch Packets

Same idea can be applied to other CPU intensive networking operations
- Connection processing
- Https (SSL Shader- NSDI 2011)