Router Design II

- Tag Switching
- Extensible (Active) Routers

IP Forwarding is Slow

- Problem: classless IP addresses (CIDR)
- Route by variable-length Forwarding Equivalence Classes (FEC)
  - FEC = IP address plus prefix of 1-32 bits; e.g., 172.200.0.0/16
- IP Router
  - forwarding tbl: <FEC> ⟶ <next hop, port>
  - match IP address to FEC w/longest prefix

ATM Forwarding

- Primary goal: fast, cheap forwarding
- 1Gb/s IP router: $187,000
- 5Gb/s ATM switch: $41,000
- Create Virtual Circuit at Flow Setup
  - <in VCI> ⟶ <port, out VCI>
- Cell Forwarding
  - index, swap, switch

Cisco: Tag Switching

- Add a VCI-like tag to packets
  - <in tag> ⟶ <next hop, port, out tag>
- TSR uses ATM switch hardware
- IP routing protocols (OSPF, RIP, BGP)
  - build forwarding table from routing table
- Goal: IP router functionality at ATM switch speeds/costs

Forwarding

- *Shim* before IP header

- Tag Forwarding Information Base (TFIB)
  - <in tag> ⟶ <next hop, port, out tag>
- Just like ATM
  - index, swap, switch

Tag Binding

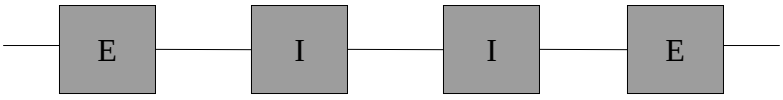
- New FEC from IP routing protocols
  - Select local tag (index in TFIB)
  - <in tag> ⟶ <next hop, port, ???>
- Need <out tag> for next hop
- Other routers need my <in tag>
- Solution: distribute tags like other routing info

Tag Distribution Protocol

- Send TDP messages to peers
  - – <FEC, my tag>
- Upon receiving TDP message, check if sender is next hop for FEC
  - – yes, save tag in TFIB
  - – no, can discard or save for future use
- ‘Control-driven’ label assignment

The First Tag

- Two kinds of routers: edge vs. interior



- Edge: add shim based on IP lookup, strip at exit
- Interior: forward by tag only

Robustness Issues

- What if tag fault?
  - try to forward (default route)
  - discard packet
- Forwarding Loops
  - topology changes cause temporary loops
  - TTL field in tag, same as IP

Tag Switching Recap

- Shim header contains tag
- <in tag> → <next hop, port, out tag>
- TSR runs standard IP routing protocols
- <In tag> bound to new FEC in TFIB
- Tags distributed among peers by TDP
- Edge TSR inserts first shim

Routing Hierarchy

- Improve scalability by reducing routes known by interior TSRs
- Routing protocols help scalability
  - OSPF/RIP vs. BGP
- Problem: transit routing domain
  - often forwarding between border routers
  - interior nodes still need to know all interdomain routes

More Routing Hierarchy

- Solution: tag stack
- Border TSR at entrance of transit routing domain pushes tag
- Interior TSRs route between borders
- Border TSR at exit pops tag, forwards packet between domains
- Reduce routes at interior TSRs

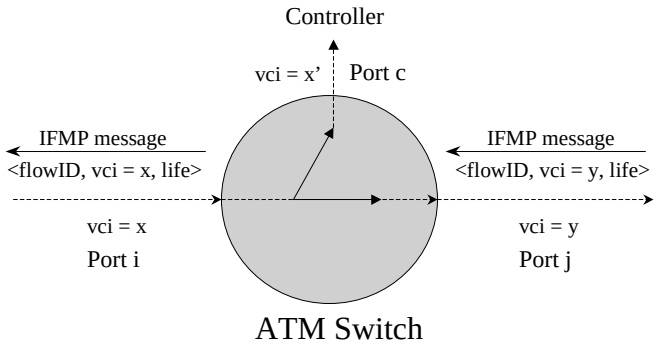
Ipsilon: IP Switching

- Run on ATM switch over ATM network
  - ATM hardware + IP switching software
- Idea: Exploit temporal locality of traffic to cache routing decisions
- Associate labels (VCI) with flows
  - forward packets as usual
  - main difference is in how labels are created, distributed to other routers

IP Switch

- Assume default ATM virtual circuits between routers
- Router runs IP routing protocol, can forward IP packets on default VCs
- Identify flows, assign flow-specific VC
  - “flow” = port pair or host pair
- ‘Data-driven’ label assignment

Flow Setup on IP Switch

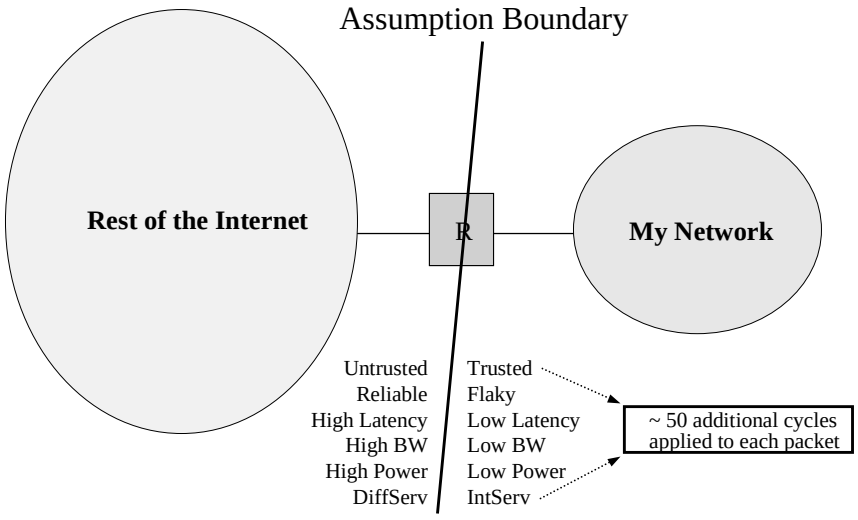


- <vci = x> ⟶ <port c, vci = x'>
- Get IFMP, <vci = x> ⟶ <port j, vci = y>

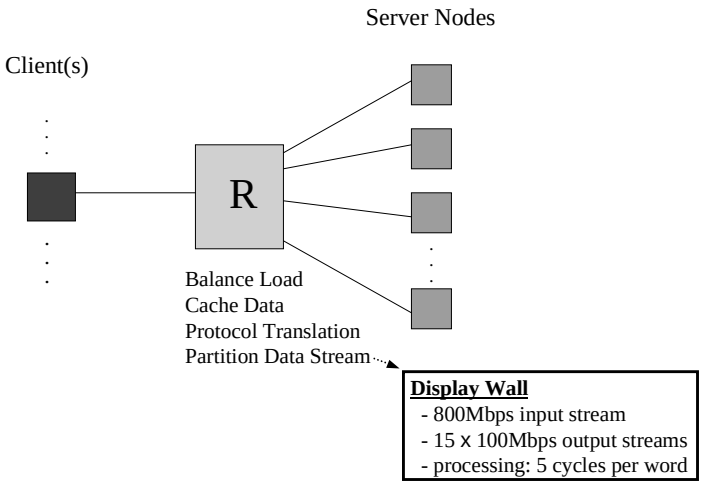
Comparison

IP Switching	Tag Switching
• Switch by flow	• Switch by FEC
• Data driven	• Control driven
• Soft-state timeout	• Route changes
• Between end-hosts	• Between edge TSRs
• Every router can do IP lookup	• Interior TSRs only do tag switching
• Scalable?	

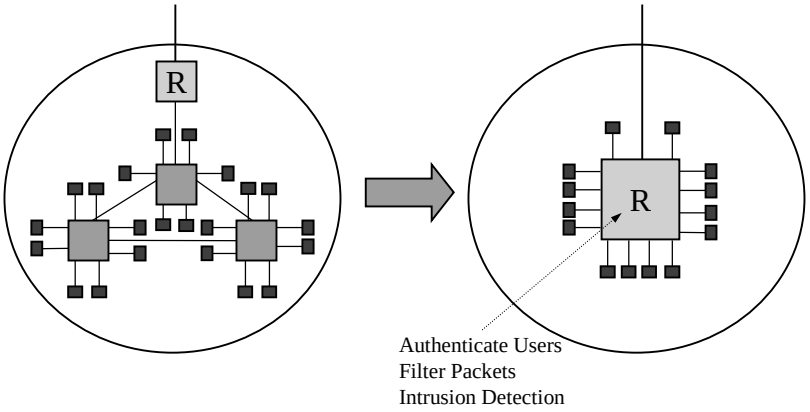
Edge Routers



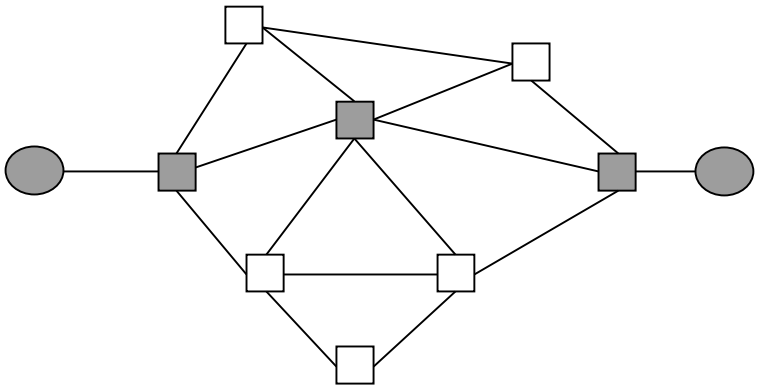
Scalable Servers



Router as LAN



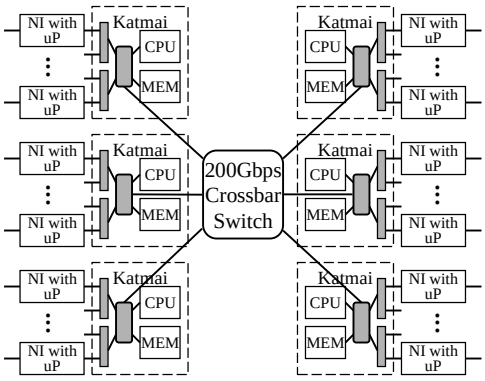
Active Networks



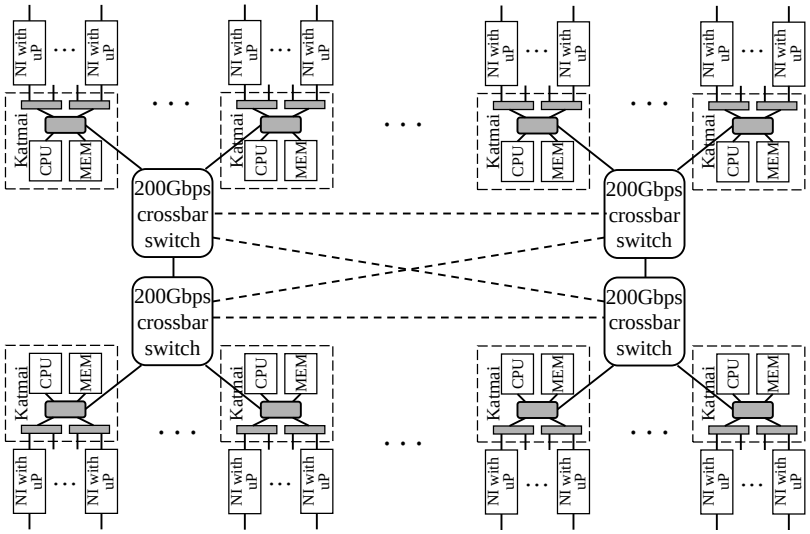
Key Ideas

- Closer integration of...
  - processing cycles
  - switching bandwidth
- Implement hierarchy of…
  - classification mechanisms
  - processing paths

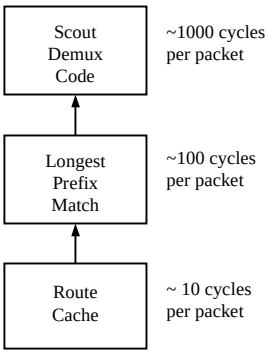
Hardware Architecture



Scalable Router



Classification Hierarchy



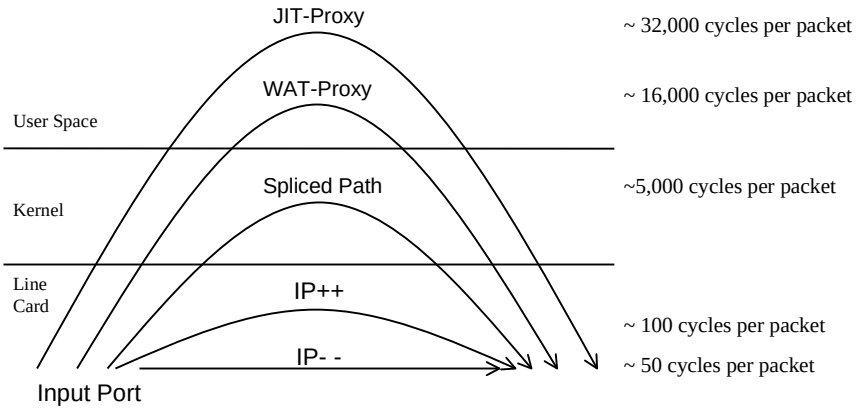
~1000 cycles per packet

~100 cycles per packet

~ 10 cycles per packet

Cycle Budget

Line Card: 200-600 cycles per packet

Katmai: (1Gbps flow)
- 1000 cycles per packet
- 50 cycles per word

Path Hierarchy