Forwarding and Routers

- Forwarding
- IP lookup
- High-speed router architecture
- Readings
  - [McK97] A Fast Switched Backplane for a Gigabit Switched Router
  - [KCY03] Scaling Internet Routers Using Optics
  - Know RIP/OSPF (2 sections)
- Optional
  - [D+97] Small Forwarding Tables for Fast Routing Lookups

Outline

- IP router design
  - Architectures
  - Scheduling
  - Limits
- IP route lookup
- Variable prefix match algorithms
- Packet classification

IP Router Design

- Different architectures for different types of routers
- High speed routers incorporate large number of processors
- Common case is optimized carefully
What Does a Router Look Like?

- Currently:
  - Network controller
  - Line cards
  - Switched backplane
- In the past?
  - Workstation
  - Multiprocessor workstation
  - Line cards + shared bus

Control Plane & Data Plane

- Control plane must remember lots of routing info (BGP tables, etc.)
- Data plane only needs to know the “FIB” (Forwarding Information Base)
  - Smaller, less information, etc.
  - Simplifies line cards vs the network processor

Generic Router Architecture

Control Plane & Data Plane

- Control plane must remember lots of routing info (BGP tables, etc.)
- Data plane only needs to know the “FIB” (Forwarding Information Base)
  - Smaller, less information, etc.
  - Simplifies line cards vs the network processor

First Generation Routers

- Typically <0.5Gb/s aggregate capacity
Big, Fast Routers: Why Bother?

- Faster link bandwidths
- Increasing demands
- Larger network size (hosts, routers, users)
- More cost effective

Innovation #1: Each Line Card Has the Routing Tables

- Prevents central table from becoming a bottleneck at high speeds
- **Complication**: Must update forwarding tables on the fly.

Second Generation Routers

- Bypasses memory bus with direct transfer over bus between line cards
- Moves forwarding decisions local to card to reduce CPU pain
- Punt to CPU for “slow” operations
- Typically <5Gb/s aggregate capacity

Bus-based

- Some improvements possible
  - Cache bits of forwarding table in line cards
  - Send directly over bus to outbound line card
- But shared bus remains big bottleneck
  - E.g., *modern* PCI bus (PCIx16) is only 32Gbit/sec (in theory)
  - Almost-modern cisco (XR 12416) is 320Gbit/sec.
  - Ow! How do we get there?
Innovation #2: Switched Backplane

- Every input port has a connection to every output port
- During each timeslot, each input connected to zero or one outputs
- **Advantage:** Exploits parallelism
- **Disadvantage:** Need scheduling algorithm

---

**Third Generation Routers**

"Crossbar": Switched Backplane

- Line Card
- CPU Card
- Local Buffer Memory
- Routing Table
- Fwding Table
- Fwding Table

Periodic Control updates

Typically <50Gb/s aggregate capacity

---

**Line Cards**

- Network interface cards
- Provides parallel processing of packets
- Fast path per-packet processing
  - Forwarding lookup (hardware/ASIC vs. software)

---

**What a Router Line Card Looks Like**

- 1-Port OC48 (2.5 Gb/s)
  (for Juniper M40)
- 4-Port 10 GigE
  (for Cisco CRS-1)

Power: about 150
Network Processor

- Runs routing protocol and downloads forwarding table to line cards
  - Some line cards maintain two forwarding tables to allow easy switchover
- Responsible for other control and management tasks, e.g. monitoring, configuration, etc.
- Performs “slow” data path processing
  - Handles ICMP error messages
  - Handles IP option processing

Switch Design Issues

- Have N inputs and M outputs
  - Multiple packets for same output – output contention
  - Switch contention – switch cannot support arbitrary set of transfers
    - Crossbar
    - Bus
      - High clock/transfer rate needed for bus
    - Banyan net
      - Complex scheduling needed to avoid switch contention
- Variable versus fixed sized packets
- Solution – buffer packets where needed

Switch Buffering

- Input buffering
  - Which inputs are processed each slot – schedule?
  - Head of line packets destined for busy output blocks other packets
- Output buffering
  - Output may receive multiple packets per slot
  - Need speedup proportional to # inputs
- Internal buffering
  - Head of line blocking
  - Amount of buffering needed

Line Card Interconnect

- Virtual output buffering
  - Maintain per output buffer at input
  - Solves head of line blocking problem
  - Each of MxN input buffer places bid for output
- Crossbar connect
  - Challenge: map of bids to schedule for crossbar
- Many algorithms with different design goals
  - Throughput, speed, complexity, …
iSlip Algorithm - Overview

- Challenge: conflict free schedule for crossbar
- Iterative algorithm – three steps per iteration
  1. Request: each input sends request to each output for which it has cells
  2. Grant: outputs pick an input based on RR priority
  3. Accept: inputs accept an output based on RR priorities
- Has good properties:

ISLIP

ISLIP (cont.)

What Limits Router Capacity?

Approximate power consumption per rack

Power density is the limiting factor today
Examples of Multi-rack Routers

- Alcatel 7670 RSP
- Juniper TX8/T640
- Avici TSR
- Chiaro

Limits to Scaling

- Overall power is dominated by linecards
  - Sheer number
  - Optical WAN components
  - Per packet processing and buffering.
- But power density is dominated by switch fabric
Multi-rack Routers Reduce Power Density

Switch

Limit today ~2.5Tb/s
- Electronics
- Scheduler scales <2x every 18 months
- Opto-electronic conversion

Question

• Instead, can we use an optical fabric at 100Tb/s with 100% throughput?

• Conventional answer: No
  - Need to reconfigure switch too often
  - 100% throughput requires complex electronic scheduler.

• Presentation by Ankit

If Traffic is Uniform…

Real Traffic is Not Uniform
Two-stage Load-Balancing Switch

100% throughput for weakly mixing, stochastic traffic

[C.-S. Chang, Valiant]

Static WDM Switching

Array Waveguide Router (AWGR)
Passive and Almost Zero Power

4 WDM channels, each at rate 2R/N
Outline

- IP router design
- IP route lookup

Next:
- Variable prefix match algorithms
- Packet classification
- Flow monitoring