L-7 Routers

Acknowledgments: Lecture slides are from the graduate level Computer Networks course thought by Srinivasan Seshan at CMU. When slides are obtained from other sources, a reference will be noted on the bottom of that slide and a full reference detail on the last slide.
Outline

• IP router design
• IP route lookup
• Variable prefix match algorithms
What Does a Router Look Like?

• Currently:
  • Network Processor
  • Line cards
  • Switched backplane

• In the past?
  • Workstation
  • Multiprocessor workstation
  • Line cards + shared bus
Line Cards

• Network interface cards

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

- Runs routing protocol and downloads forwarding table to line cards
  - Some line cards maintain two forwarding tables to allow easy switchover
- Performs “slow” 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

• Solution – buffer packets where needed
FIGURE 2 The Basic Architectures of Packet Processors

(a.)

(b.)

(c.)

(d.)

[McK97]
FIGURE 4 A Four-Input Crossbar Interconnection Fabric

[McK97]
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
FIGURE 6 Model of an N-port Input-Queued Switch with Virtual Output Queueing (VOQ)

Input 1

Data H

A_{1,1}^{(n)}

Q(1,1)

Q(1,N)

Output 1

Crossbar Switch

Input N

Data H

Q(N,1)

Q(N,N)

Output N

Centralized Scheduler

Note: Cells arrive at input 1, and are placed into the appropriate VOQ. At the beginning of each time slot, the centralized scheduler selects a configuration for the crossbar, by matching inputs to outputs. Head of line blocking is eliminated by using a separate queue for each output at each input.

[McK97]
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
ISLIP

Round 1, Iteration 1

Round 1, Iteration 2

Round 2, Iteration 1
ISLIP (cont.)
What Limits Router Capacity?

Approximate power consumption per rack

Power (kW)

0 2.5 5.0 7.5 10.0


Power density is the limiting factor today
FYI--Network Element Power

• 96 x 1 Gbit port Cisco datacenter switch consumes around 15 kW -- equivalent to 100x a typical dual processor Google server @ 145 W

• High port density drives network element design, but such high power density makes it difficult to tightly pack them with servers

• Is an alternative distributed processing/communications topology possible? [CS268]
Power/Cooling Issues

- Communication Equipment (frames)
- Servers & Disk Storage Systems (1.8 - 2.2m tall)
- Workstations (standalone)
- Tape Storage Systems

Year of Product Announcement

Heat load per product footprint - watts/ft²

Heat load per product footprint - watts/m²
Multi-rack Routers Reduce Power Density
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

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.
If Traffic is Uniform…
Real Traffic is Not Uniform
Two-stage Load-Balancing Switch

Load-balancing stage

Switching stage
Static WDM Switching

Array Waveguide Router (AWGR)
Passive and Almost Zero Power

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

- IP router design
- IP route lookup
- Variable prefix match algorithms
Original IP Route Lookup

- Address classes
  - A: 0 | 7 bit network | 24 bit host (16M each)
  - B: 10 | 14 bit network | 16 bit host (64K)
  - C: 110 | 21 bit network | 8 bit host (255)
- Address would specify prefix for forwarding table
  - Simple lookup
Original IP Route Lookup – Example

- **www.cmu.edu** address 128.2.11.43
  - Class B address – class + network is 128.2
  - Lookup 128.2 in forwarding table
  - Prefix – part of address that really matters for routing

- Forwarding table contains
  - List of class+network entries
  - A few fixed prefix lengths (8/16/24)

- Large tables
  - 2 Million class C networks
CIDR Revisited

• Supernets
  • Assign adjacent net addresses to same org
  • Classless routing (CIDR)

• How does this help routing table?
  • Combine routing table entries whenever all nodes with same prefix share same hop
  • Routing protocols carry prefix with destination network address
  • Longest prefix match for forwarding
CIDR Illustration

Provider is given 201.10.0.0/21

Provider

201.10.0.0/22  201.10.4.0/24  201.10.5.0/24  201.10.6.0/23
CIDR Shortcomings

- Multi-homing
- Customer selecting a new provider

```
CIDR  |  Provider 1        |  Provider 2
-----|--------------------|--------------------
201.10.0.0/21 |                   |
201.10.0.0/22 | 201.10.4.0/24     |
201.10.5.0/24 |                   |
201.10.6.0/23 or Provider 2 address |
```
Outline

• IP router design
• IP route lookup
• Variable prefix match algorithms
Trie Using Sample Database

Sample Database

- P1 = 10*
- P2 = 111*
- P3 = 11001*
- P4 = 1*
- P5 = 0*
- P6 = 1000*
- P7 = 100000*

Trie

Root

P5

P1

P6

P7

P8

P4

P2

P3

0

1

0

0

0

1

0

0

0

0

1

1

1

0

1
How To Do Variable Prefix Match

- Traditional method – Patricia Tree
  - Arrange route entries into a series of bit tests
- Worst case = 32 bit tests
  - Problem: memory speed is a bottleneck

![Patricia Tree Diagram]

**Bit to test – 0 = left child, 1 = right child**

- 0
  - default 0/0
  - 128.2/16
- 10
  - 128.32/16
- 16
  - 128.32.130/240
  - 128.32.150/24
Speeding up Prefix Match (P+98)

- Cut prefix tree at 16 bit depth
  - 64K bit mask
  - Keep array of routes/pointers to subtree
- Subtrees are handled separately
  - Bit = 1 if tree continues below cut (root head)
  - Bit = 1 if leaf at depth 16 or less (genuine head)
  - Bit = 0 if part of range covered by leaf
Prefix Tree

Port 1
Port 5
Port 7
Port 3
Port 9
Port 5
Prefix Tree

Subtree 1

Subtree 2

Subtree 3
Speeding up Prefix Match - Alternatives

• Route caches
  • Temporal locality
  • Many packets to same destination

• Other algorithms
  • Waldvogel – Sigcomm 97
    • Binary search on prefixes
    • Works well for larger addresses
  • Bremler-Barr – Sigcomm 99
    • Clue = prefix length matched at previous hop
    • Why is this useful?
  • Lampson – Infocom 98
    • Binary search on ranges