Basic Requirements of a Switch Router

What does / can a router do?
What can be done in hardware / software?
How do protocols / standards influence the design?
Types of Inter-network Nodes

1. Physical
2. Data Link
3. Network
4. Transport
5. Session
6. Presentation
7. Application

- Hub
- Bridge Switch
- Router
- Gateway
## What’s the difference? Switch vs Router

<table>
<thead>
<tr>
<th>Layer</th>
<th>Purpose of the Layer</th>
<th>Role of Switching</th>
</tr>
</thead>
<tbody>
<tr>
<td>(7) Application</td>
<td>Defines user-oriented services such as file transfer, messaging, and transaction processing; provides for structuring applications, coding the data, and exchanging information</td>
<td>Application switching (e.g. e-mail forwarding); gateways between different application types; support for management functions; selection of destination for messages</td>
</tr>
<tr>
<td>(6) Presentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5) Session</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4) Transport</td>
<td>Delivery of data to applications, division of messages into packets</td>
<td>Directs the messages to the specific destination application or protocol type</td>
</tr>
<tr>
<td>(3) Network</td>
<td>End-to-end communications through one or more subnets; selects optimal routes; controls loops; manages addressing</td>
<td>Forwards packets through an interconnected set of networks</td>
</tr>
<tr>
<td>(2) Data Link</td>
<td>Transfer of frames across a single network link such as a LAN; manages contention</td>
<td>Controls switched circuits, switched LANs, and recovers from link errors</td>
</tr>
<tr>
<td>(1) Physical</td>
<td>Transmission over a physical circuit including physical connectors, bit encoding, etc.</td>
<td>Circuit switching as is used for telephony and port switching for LAN physical media</td>
</tr>
</tbody>
</table>
Anatomy of a Node

Real Time OS

Application Level API

Kernel Level API

Network Protocol

NIC Driver

Fast Ethernet NIC

Another Node

Protocol API

Driver Specification

Hardware Interface

Repeater

Transmit Receive

Transmit Receive
**Fast Ethernet System**

![Diagram of Fast Ethernet System]

- **Device**
  - **Network Interface**
    - **Hardware**
    - **Software**
      - **RTOS / Applications**
      - **Protocol**
      - **LLC**
      - **MAC**
      - **Reconciliation**
    - **MII**
      - **PCS**
      - **PMA**
      - **PMD**
      - **AutoNeg**
    - **MDI**
    - **Media**

- **Fast Ethernet Standard** (802.3u)

---

CCL/N300; Paul Huang 3/21/99
Fast Ethernet System

NODE

Device

Software

Network Interface

Hardware

RTOS / Applications

Protocol

LLC

MAC

Reconciliation

PCS

PMA

PMD

AutoNeg

Baseband Repeater Unit

PCS

PMA

PMD

AutoNeg

Reconciliation

PCS

PMA

PMD

AutoNeg

Fast Ethernet Standard (802.3u)

Hardware

Media

Software

Media
Fast Ethernet System

NODE

RTOS / Applications
Protocol
LLC
MAC

Reconciliation

PCS
PMA
PMD
AutoNeg

MII

Media

MDI

L2 Switch

RTOS / Applications
Protocol
LLC
MAC

Reconciliation

PCS
PMA
PMD
AutoNeg

Fast Ethernet Standard
(802.3u)

Software

Hardware

Network Interface

Device
Fast Ethernet System

NODE

Network Interface

Hardware

MII

MDI

Software

Device

RTOS / Applications

Protocol

LLC

MAC

Reconciliation

PCS

PMA

PMD

AutoNeg

Fast Ethernet Standard

(802.3u)

Software

Hardware

Device

RTOS / Applications

Protocol

LLC

MAC

Reconciliation

PCS

PMA

PMD

AutoNeg

Fast Ethernet Standard

(802.3u)
Media Access Controller

- Determine when a node can transmit a packet
- Send frames to the PHY for conversion into packets and transmission on the media
- Receive frames from the PHY and send them to the software that processes frames (protocols and applications).
- Frame checking
  - Valid Frames
    - Frame size between 64 bytes & 1518 bytes
    - Valid frame check sequence (CRC)
    - Even number of octets
  - Non-valid Frames
    - Runts: Any frame that is shorter than 64 bytes (512 bits) in size
    - Jabber: Data transmission greater than 400 ms (largest packet: 120.56 ms)
    - Dribble: Invalid number of octets
- Media independent
Media Access Rules

» Listen before sending
  - CSMA — Carrier Sense Multiple Access
  - Interpacket Gap (IPG = 96 bit time or 0.96 µs for FE)

» Backoff
  - CD — Collision Detection
  - Collision domain
  - Collision window
  - Slot time == maximum allowable collision window (512 bit times)
    • minimum frame size (512 bit / 64 bytes)
    • maximum network diameter
  - Truncated binary exponential backoff
    • \( \text{RAND}(0, 2^{\min(N,10)}) \), where \( N \) is the transmit attempt counter
    • Integer multiple of 512 bit slot time (i.e. 512, 1024, 1536, 2048, … , 4096, etc.)
    • Maximum backoff time is 5.3 ms.
CSMA / CD Flow Chart

Transmit Frame

1. Assemble Packet
2. Set Attempt counter to 0

Is media busy?

Y

Other Tx Finished?

N

IPG passed?

N

Delay Rest of IPG

Y

Transmit 1st bit of the Packet

Collision?

N

Transmit next bit

Y

Tx Completed?

N

Frame Tx Failed

Frame Tx Success

Attempts > 16

Increment attempt counter

Send Jam (32 bits)

Delay back-off

Compute back-off

CCL/N300; Paul Huang
3/21/99
# Packet Format

<table>
<thead>
<tr>
<th>Preamble</th>
<th>SFD</th>
<th>Data Frame</th>
<th>EFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>SA</td>
<td>L/T</td>
<td>Data</td>
</tr>
<tr>
<td>6 Bytes</td>
<td>6 Bytes</td>
<td>2B</td>
<td>46 ~ 1500 Bytes</td>
</tr>
<tr>
<td>I/G</td>
<td>U/L</td>
<td>OUI</td>
<td>Address (OUA)</td>
</tr>
<tr>
<td>48</td>
<td>47</td>
<td>45</td>
<td>24 23</td>
</tr>
</tbody>
</table>

- **Preamble**: 10101010
- **SFD**: Start Frame Delimiter 10101011
- **EFD**: End of Frame Delimiter ----
- **DA**: Destination Address
- **SA**: Source Address
- **L/T**: Length / Type
- **Data**: 46 ~ 1500 Bytes
- **FCS**: Frame Check Sequence
- **I/G**: Individual / Group
- **U/L**: Universal / Local Administration
- **OUI**: Organizationally Unique Identifier
- **OUA**: Organizationally Unique Address
Wire Speed — Efficiency

Data: 1500 1262 1006 494 238 110 46 32 24 16 8 3
Packet: 1518 1280 1024 512 256 128 64 64 64 64 64 64

Efficiency:
- 97.5%
- 97.1%
- 96.4%
- 92.9%
- 86.2%
- 74.3%
- 54.8%
- 38.1%
- 28.6%
- 19.0%
- 9.5%
- 3.6%
- 1.2%
Network Utilization

- Full Utilization 30%
- Saturation
- Best
- Ok
- Bad

Utilization vs. Offered Load graph.
**Basic & Worst Case Collision Detection w/ Cat-5 Cable**

### Node-to-node Path Delay Value

(rounded to the nearest whole bit time)

- **A → B**: $126 = 25 + 5.7 + 70 + 0.57 + 25$
- **A → C**: $183 = 15 + 5.7 + 70 + 57 + 25$
- **B → C**: $178 = 25 + 0.57 + 70 + 57 + 25$
- **C → D → C**: $468 = 2 \times (25 + 57 + 70 + 57 + 25)$

### Safety margin

- 4

### Bit-time margin

- 40 = 512 - 472 - 4

---

[Diagram showing node-to-node path delays and safety margins]
Worst-case Collision Window w/ Class-I Hub and Fiberoptic Cable

Repeater (70 bt)

134 m  10 m  134 m
67 bt  5 bt  67 bt

25 bt

A

25 bt

B

25 bt

C

134 m
67 bt

Node-to-node Path Delay Value
(rounded to the nearest whole bit time)

A → B  254 = 25 + 67 + 70 + 67 + 25
A → C  192 = 15 + 67 + 70 + 5 + 25
B → C  192 = 25 + 5 + 70 + 67 + 25

A → B → A  508 = 2 * (25 + 67 + 70 + 67 + 25)
Safety Margin  4
Bit time margin  0 = 512 - 508 - 4
Missed Collision w/ oversized network

Repeater (70 bt)

A
B
C

134 m  10 m  150 m
67 bt  5 bt  75 bt

25 bt

25 bt

Node-to-node Path Delay Value
(rounded to the nearest whole bit time)

A \rightarrow B: 262 = 25 + 67 + 70 + 75 + 25
A \rightarrow C: 192 = 15 + 67 + 70 + 5 + 25
B \rightarrow C: 200 = 25 + 5 + 70 + 75 + 25

A \rightarrow B \rightarrow A: 524 = 2 \times (25 + 67 + 70 + 75 + 25)

Safety Margin: 4
Bit time margin: -16 = 512 - 524 - 4
100Base-TX / FX Connection

Pin 1
Pair 1
Pin 2
Pin 3
Pair 3
Pin 6

Twisted Pair

Fiberoptic pair

Node

MAC REC PCS PMA

MAC REC PCS PMA

MAC REC PCS PMA

MAC REC PCS PMA

Repeater Unit

PMA PCS R

PMA PCS R

PMA PCS R
100Base-T4 Connection

Node

MAC  REC  PCS  PMA

MII

Pair 1
Pin 1
Pin 2
Pin 3
Pin 4
Pin 5
Pin 6
Pin 7
Pin 8

Pair 2
Pair 3
Pair 4

Twisted Pair

Repeater Unit

MAC  REC  PCS  PMA

MII

PMA  PCS  R
Repeater Testing

- Function
  - Transmit / Receive event
    - Data handling: forward packet
    - Receive event handling: carrier sense
  - Error handling via partition
    - False carrier events: invalid start-of-stream delimiter
      - Partition and set LINK UNSTABLE state after two false carrier event
      - Send jam signal to all other ports on the repeater for 5 $\mu s$ or until end of FCE
      - Unset LINK UNSTABLE state after detecting no activity for more than 331 $\mu s$ or detecting a valid incoming packet after the line has been idle for the interpacket gap time of 640 $\mu s$.  
    - Excessive collision: more than 60 collisions in a row
      - Partition after receiving more than 60 collisions in a row
      - Clear after detecting activity without a collision for more than 5 $\mu s$
    - Receiver Jabber: data transmission greater than 400 $\mu s$ (largest packet: 120.56 $\mu s$)
      - Clear after jabber stops
**Basic Bridge Operation**

- **Receive Frame**
  - **MAC Address Table**
    - **Record Address & Port # in Table**
      - **Input**
        - **Address in Table?**
          - **Ports the same?**
            - **Unicast DA?**
              - **Lookup DA in table & get corresponding port #**
              - **Forward Frame to All Ports except the inbound port**
                - **Forward Frame to DA port #**
                  - **Output**
                    - **Receive Frame**
  - **Address in Table**
    - **Forward Frame to DA port #**
      - **Inbound Port = DA port #**
        - **Filter the Frame**
          - **Address in Table?**
            - **Unicast DA?**
              - **Lookup SA in Table**
                - **Record Address & Port # in Table**
                  - **Input**
                    - **Receive Frame**
Multiple Bridges

Segment | Alpha | Gamma | Beta | Epsilon | Delta
---|---|---|---|---|---
MAC Address | ABCDEF | GHIJKL | MNOPQR | STUVWX | YZ
Bridge #1 | 111111 | 222222 | 333333 | 222222 | 22
Bridge #2 | 111111 | 111111 | 111111 | 333333 | 22
Multiple Bridges

Problems caused by looping
- Broadcast storming
- Learning problems
- Cloned unicast frames

Solution
- Spanning Tree Protocol
Multiple Bridges

Problems caused by looping:
- Broadcast storming
- Learning problems
- Cloned unicast frames

Solution:
- Spanning Tree Protocol
Ethernet Switching

- Basic techniques
  - Cut-through
    - Advantages
      - low latency
    - Disadvantages
      - forwards runt & error frames
      - internal speedup not possible
      - mixed speeds difficult
  - Interim Cut-through
    - Same as CT, but less runt frames
  - Store & Forward
    - Advantage
      - reduces error frames
      - architecturally flexible
    - Disadvantage
      - longer latency (not really bad !!)
Router vs. Routing function

- What does a router do
  - Routing function
    - IP packet forwarding
    - Route calculation/convergence
    - Route management
  - Multicast
    - IP packet duplication
    - Multicast routing
  - Traffic Mgt. (QoS)
    - Packet Classification
    - Packet Filtering
    - Queue Management
  - Network Mgt.
  - Security
    - Firewall
    - Authentication

- Router
  - Conventional stand-alone router performs an IP routing function
    - Bus based
    - Central CPU
    - Cached forwarding tables
    - Centralized routing tables
    - SW table lookup
  - Calculations required
    - 10 Gbps throughput
    - 64 byte packets = 50 ns / packet
    - < 50 ns to make each routing decision.
Network Processor

- Evolution to network processors
  - Software programmable
  - Optimized instruction set for networking
  - Breakthrough performance
  - Switching, routing, and features

- Customer-specific differentiation
  - Base level instruction set
  - Empowers the higher level software
  - Addresses all networking markets

- Enables high-level functions at the same speed as basic switching wire speed

High Performance
Low Cost
Highly Flexible
Fast time-to-market
So, What’s so hard about switching & routing ?!#
**Typical Router Architecture**

- **Slow path**
- **Fast path**
- **CPU**
- **Memory**
- **Interface**
- **Bus / Switch Fabric**
- **Packet Forwarding**
- **Security Mgt.**
- **Traffic Mgt.**
- **Node Mgt.**
- **Route Mgt.**

*Cache assisted IP routing*
The Easy Part: Basic IP Forwarding

To forward an IP Unicast packet, you need to:

- Parse the IP header
- Lookup the Address in a large table of address prefixes (100,000+ entries)
- Check the checksum
- Decrement the TTL and adjust the checksum

This stuff is easy to do at high speed

- This is straightforward for ASIC implementation
- Clever implementers can do OC-192 (or even OC-768)
- With today’s technology, this is not even close to being the bottleneck
So What’s So Hard?

System performance = function of ALL elements
“A chain is only as strong as its weakest link”
There are things which can make high speed forwarding hard:

- Where data flows come together (backplane)
- Where parallelism is difficult
  - e.g., Optics, software, protocol
- Protocol standards
  - Unstable or poorly designed or under-defined standards
  - Need mature implementations
  - Multi-lingual
  - Too many standards
- Lots of options and alternative paths
- Maintaining per-packet state that comes and goes
So What’s So Hard?

- Proliferation of standards make system implementation hard:
  - Support for legacy protocol (i.e. Multi-protocol & conformance)
  - Interoperability (i.e. Multi-vendor)
  - Addressing
  - Routing
  - Multicasting
  - Traffic mgt. (QoS)
  - Network mgt.
  - Mobility
  - Security
  - Virtual Private Network
So What’s So Hard?

- Reliability, maintainability, redundancy
  - Hot swappable, Hot standby router
  - Coherent network state
  - Online upgrade
  - Redundancy (power supply, link failure, etc.)

- Scalability

- Additional
  - Frame translation
  - Load balancing
  - Port mirroring
One Hard Part: The Backplane

- Given 100 OC-xx ports, served by 100 line cards, somehow packets have to get between the line cards.

- The design of the switched backplane is “non-trivial”:
  - If there are “n” line cards, you have an n*log(n) problem.
  - You want very high switch utilization to push performance (this affects where packet is buffered).
  - Power and heat become important.
  - Cost of hardware is meaningful.

- It is easy to lose your QoS guarantees across the switched backplane.
ASIC Designer’s Nightmare: Options

- MPLS and IP forwarding
- Filters (source or destination address, RSVP, … )
- Tunneling: Encapsulation and decapsulation (particularly if reassembly is needed)
- Multicast
- IP Options
- Multipath (ECMP)
- NAT (application addresses plus state)
- IPv6 alternate headers
What does MPLS do to a router?

- **Answer**: Provide lots of alternative forwarding paths

  ![Diagram]

  - `<IP>` → IP
  - `<IP>` → `<Shim> + <IP>`
  - `<IP>` → `<ATM+ shim> + <IP>`
  - `<Shim>` → `<Shim> ; <Shim>+<Shim>; <ATM+ shim>`
  - `<Shim>+< IP>` → `<IP>` (with or without IP lookup)
  - `<ATM+ shim>` → `<ATM+ shim>; <Shim>; …`
  - `<ATM+ shim>+< IP>` → `<IP>` (with or without IP lookup)
  - `<ATM+ shim>+< Shim>` → `<Shim>`
  - etc …

- This is not popular with hardware developers 😞
A Hardware Developer’s View

- Generally: Hardware engineers wish that folks who write standards paid attention to hardware issues
- IP Forwarding can be done very fast, no problem
- CLNP forwarding can be done very fast, …
- But: Please don’t give us so many options!
- “It is clear that IP standards (including IPv6) were designed by folks who don’t pay any attention to what it takes to build a fast router?”
- (On the other hand, things are still within a top hardware team’s capabilities)
The Bottom Line on Speed

- There are really three bottlenecks:
  - The switched backplane
  - The optics
  - How much extra complexity and flexibility you want
    (Filtering, MPLS, options, all make it harder to go fast)

- It really doesn’t matter what the forwarding looks like, if it's straightforward and well defined

- At very high speed, IP, MPLS, ATM, Frame Relay, are all constrained by the same issues are all constrained by the same issues
Can Routers go fast enough?

- There is some limit to how fast routers can go

- Or, more correctly, there is some limit on how fast electronics can go
  - Given today’s chip technology, and reasonable economics, the limit might be on the order of a few thousand * OC-192
  - In four years, possibly ditto but * OC-768

- Past this point, we need optics
  - Core switches become WDM switches
  - Very fast, very branchy routers (and ATM switches) become feeders for WDM in the core
Issues affecting Reliability

- Hardware robustness
  - Reliable hardware, Redundancy at many levels
- Software quality and robustness
  - e.g., How good is your routing software?
- Protocol Design
- Response to congestion
- Failover of links ("Sonet-like" failover rates)
- Network Management
- Avoid mistakes, Diagnose failures
- Testing, testing, testing
Key Design Considerations

Are your assumptions reasonable?
Design Constraints

- **Target:** *Right product at the right time at the right price*
  - Cost
  - System
  - Market Segment
  - Market Timing (Market window)
  - Specifications

- **Competition**
  - Advantages & Weaknesses
  - Targeted Market

- **Resources**
  - Engineering team (Experience / Stability)
  - Management team (Financing / Supportiveness)
  - Standards / Customer / Industry tracking
What is Required ??? (Perceived vs. Real)

- Optimizing Performance:
  - Wire-speed switching at Layer 2
  - Wire-speed forwarding at Layer 3

- Minimize Latency: Cut-through switching vs. Store-and-forward

- Increased Scalability: SOHO, Departmental, Enterprise, Backbone

- Maximize Integration: Multi-chip vs. Single chip solution

- Increased Functionality:
  - VLAN (Port, MAC, IP, IEEE 802.1q Tagging, etc.)
  - Port Trunking / Port Snooping
  - Support Layer 3, Layer 4, …, Layer 7
  - Support IP, IPX, SNA, …
  - Support IEEE 802.3x flow control, jamming
  - CoS / QoS / RSVP / SBM / Differentiated Service
    - Multiple loss / delay queues
    - per VC queueing
Are these assumptions reasonable?

- Maintain multicast / unicast packet sequence.
- Multicast packet needs to switched at the same time
- Support 8 k / 16 k / 32 k MAC addresses.
- Support 8 k / 16 k / 32 k IP addresses.
- Support full SNMP / RMON statistic collection.
Key Design Considerations

Can you successfully overcome today’s technology limitations?
Technology Assumptions

- Memory speeds, size, types
  - DRAM, SRAM, SDRAM, SSRAM, Rambus, NetRAM

- Semiconductor technology
  - Dimension: 0.8 μm, …, 0.35 μm, 0.25 μm, 0.18 μm, etc.
  - Power: 5 V, 3.3 V, 2.5 V, etc.
  - Embedded Memory

- Design Tools
  - Simulation: RTL level, Behavioral, Cycle-base
  - Layout Capabilities
  - Emulation Technology
Do these assumptions hold?

- Freebies
  - Memory speed / size
  - Silicon cost
  - Computational power

- Does these assumption still holds in a hyper-competitive environment?
  - No, because everybody have access to the same components and semiconductor foundry.
Improved Competitiveness

● Creating advantages by speeding up design cycle
  » Increase engineering experience
  » Invest in the state-of-the-art engineering tools
    – Latest simulation / CAD tools
    – Advance computers
    – SOC Emulation / FPGA hardware
      ● to improve simulation time
      ● to reduce potential errors, thus less debugging
Design Goals

Design specifications
Design Specifications

- **System Features**
  - Single chip eight 10/100 Mbps Ethernet ports with RMII interface
  - Provides two 32-bit memory interfaces which support SSRAM
  - Supports a 16-bit CPU interface
  - Statistics collection to support SNMP, RMON-1

- **Layer 3 Features**
  - Supports wire-speed IP routing (1.2Mpps) with line rate address lookup
  - Supports 10K routes
  - Supports IP Multicast
  - Supports two level of user data priority (Class of Service Support)

- **Layer 2 Features**
  - Supports IEEE 802.1d bridging and spanning tree algorithm
  - Supports port or IEEE802.1Q compliant tag based VLANs
  - Supports 8K MAC address entries
  - IEEE 802.3x flow control for full duplex operation
  - Supports port snooping
Design Goals

Architecture
Key Common Architectural Components

In high performance systems, the forwarding decision, backplane and output link scheduling must be performed in hardware, while the less timely management and maintenance functions are performed in software.
Architectural Evolution
Architectural Evolution

Line Card #1
Forwarding Engine

Line Card #2
Forwarding Engine

Line Card #N
Forwarding Engine

CPU
Memory

CPU
Memory

Crossbar

•
•
•
**Architectural Renewal with Advance Technology**

**Trade-off**
- Centralized vs. Distributed
- Hardware vs. Software
- Packet vs. Cell
- Connection-less vs. Connection-oriented
- Shared Bus vs. Crossbar
- Transmission vs. Switching
- “Big Pipe” vs. “Managed BW”
- “Dumb Network” vs. “Intelligent”
- Wired vs. Wireless

**Technology Factors**

**Semiconductor Advances**
- Computing Power (CPU)
- Memory Size
- Analog / RF / Optical technology

**Material Advances**
- Optical transmission
Conceptual Model of L3 Switch
Packet Flow (Tetris)

Data  Hdr

Packet Control

Hdr

Forwarding Engine  Routing
Decision

Buffer Mgt.

Data  Hdr

Output Scheduler

Priority / Normal / Multicast Pkt (Ptr & Hdr)
Packet Controller
**Output Scheduler**

From Packet Control

![Diagram showing the output scheduler flow](image)

- **Next Read**
- **Next Write**
- **Priority**
  - Normal
  - Multicast
- **Scheduler**
- **Packet FIFO**
- **Header Process**
- **MAC**

From Buffer Mgt.
Buffer Management

Routing Decision
I/O ports, Pkt Location, QoS

Modified IP Header

Tail Maintenance

Free Descriptor

Head Maintenance

Temporary Descriptors

To Output Scheduler
Forwarding Engine

L2 Table Lookup (DA)
L2 Learning (SA)
L2 Learning (SA)

Route

Port Snooping Map
Port Trunking Map

Header Verification
Multicast Lookup
Unicast Lookup (ARP)

Header Modification

Routing Decision
Design Trade-offs

Packet Memory Design
**Variable Length Format**

- **Advantage**
  - Simple, no link list required
  - No descriptors required
  - No gaps between packets
  - Easy to debug

- **Disadvantage**
  - No sharing among ports
  - Fast route decision required
  - Large temporary FIFO required
  - Parity bit or packet length write-back required
  - Look-ahead forwarding not allowed for multicast packets

- **Variations**
  - Parity bit vs. Packet Length

<table>
<thead>
<tr>
<th>Port #1</th>
<th>Port #2</th>
<th>Port #3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Fixed Packet Size Format

### Advantage
- Sharing among ports
- Routing decision relaxed
- Look-ahead forwarding allowed
- Small temporary FIFO

### Disadvantage
- Inefficient for small packets
- Link list required
- Difficult to debug

### Variations
- 1536 bytes vs 2048 bytes
Cell Format

- **Advantage**
  - Sharing among ports
  - Efficient for most packets
  - Routing decision relaxed
  - Look-ahead forwarding allowed
  - Small temporary FIFO

- **Disadvantage**
  - Large descriptor memory required
  - Link list required
  - Complex logic / Longer design cycle
  - Prone to error
  - Very difficult to debug

- **Variations**
  - 64 / 128 / 256 bytes

<table>
<thead>
<tr>
<th>Port #1</th>
<th>Port #2</th>
<th>Port #3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Design Trade-offs

Buffer Management
Queue Methodology #1
Queue Methodology #2

Link List for Port #1
A2
A4
A9

Link List for Port #2
A3
A8
A1
A6
A10

Link List for Port #3
A5
A1
A7
A10

Link List for Port #4
xx

Link List for Free Unicast Descriptor

Link List for Free Multicast Descriptor
Queue Methodology #3

- Link List for Port #1
  - A2
  - A4
  - A9
  - A6

- Link List for Port #2
  - A3
  - A8

- Link List for Port #3
  - A5
  - A7

- Link List for Port #4
  - xx

- Link List for Multicast
  - A1
  - A10

- Link List for Free Unicast Descriptor

A2
A3
A4
A8
A5
xx
A6
A7
A1
A10

Queue Methodology #4

Link List for Port #1
- A2
- A4
- A9

Link List for Port #2
- A3
- A8
- A1
- A6
- A10

Link List for Port #3
- A5
- A1
- A7
- A10

Link List for Port #4

Link List for Free Packet Buffer
Design Trade-offs

IP Forwarding
**IP Routing**

- Routing Algorithms calculate the routes
  - Unicast: RIP, OSPF
  - Multicast: DVMRP, PIM, MOSPF, CBT
- Routes are converted to table format
- Route tables are written into memory
  - Initialization
  - Route updates
- Route search looks up forwarding instruction / packet
Route Search Operation

- Longest Prefix Match
- Lookup Criteria
  - # memory access required
  - size of the data structure
  - # instruction required
- Lookup Methods
  - Hashing
  - Cache hit
  - CAM
  - Tree search
  - Table lookup
  - CPU search
  - Protocol based (Tagging)

Routing Table

R1 = 0101
R2 = 0101101
R3 = 010110101011

IP = 010101101011
IP = 010110101101

2^32 leaves (IP Address)

Depth 32
Histogram of Prefix Length Distribution
Mutated Binary Search on Hashing Table

“Waldvogel, et. Al. “Scalable High Speed IP Routing Lookup”

<table>
<thead>
<tr>
<th>Length</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>01010</td>
</tr>
<tr>
<td>7</td>
<td>0101011</td>
</tr>
<tr>
<td>12</td>
<td>0110110</td>
</tr>
<tr>
<td></td>
<td>011011010101</td>
</tr>
</tbody>
</table>

Hash Table
Mutated Binary Search on Hashing Table

- **Criteria**
  - Memory reference: 2X (Average); 5X (Worst)
  - Memory usage: 1.2 Mbyte
  - Lookup time: 100 ns (Ave); 450 ns (Worst); 2 ~ 10 Mpps

- **Advantage:**
  - The speed of IP lookup is independent of forwarding table size
  - Relatively few memory access
  - Fast enough to support Gigabit rates

- **Disadvantage:**
  - Routing update requires the tree to be rebuilt
  - Insertion and deletion of routes from memory table is complex
Direct Table Lookup

“P. Gupta, et. al. “Routing Lookups in Hardware at Memory Access Speeds”

\[
\begin{array}{c}
0 \\
23 \\
24 \\
31 \\
\end{array}
\quad
\begin{array}{c}
2^{24} \text{ Entries} \\
24 \\
31 \\
\end{array}
\quad
\begin{array}{c}
2^{8} \text{ Entries} \\
\end{array}
\quad
\text{Next Hop}
\]
Direct Table Lookup

Criteria

» Memory reference: 2X (Maximum)
» Memory usage: 33 Mbyte
» Lookup time: 10 ~ 20 Mpps

Advantage:

» Few memory references
» Enabling pipelined implementation

Disadvantage:

» Inefficient memory usage
» Insertion and deletion of routes from memory table is complex
Conclusion

- Understand and always keep the “Big Picture” in mind
  - Market
  - Technology
  - Brain Power

- Be aware of the “Hype” vs. “Reality”

- Remember the “KISS” Principle
  - Keep it simple, stupid!
  - Successful technologies are not about perfection, but about compromise between complexity, performance, ease of deployment and cost