CS120: Computer Networks

Lecture 12. Other Topics in IP

Zhice Yang

Outline

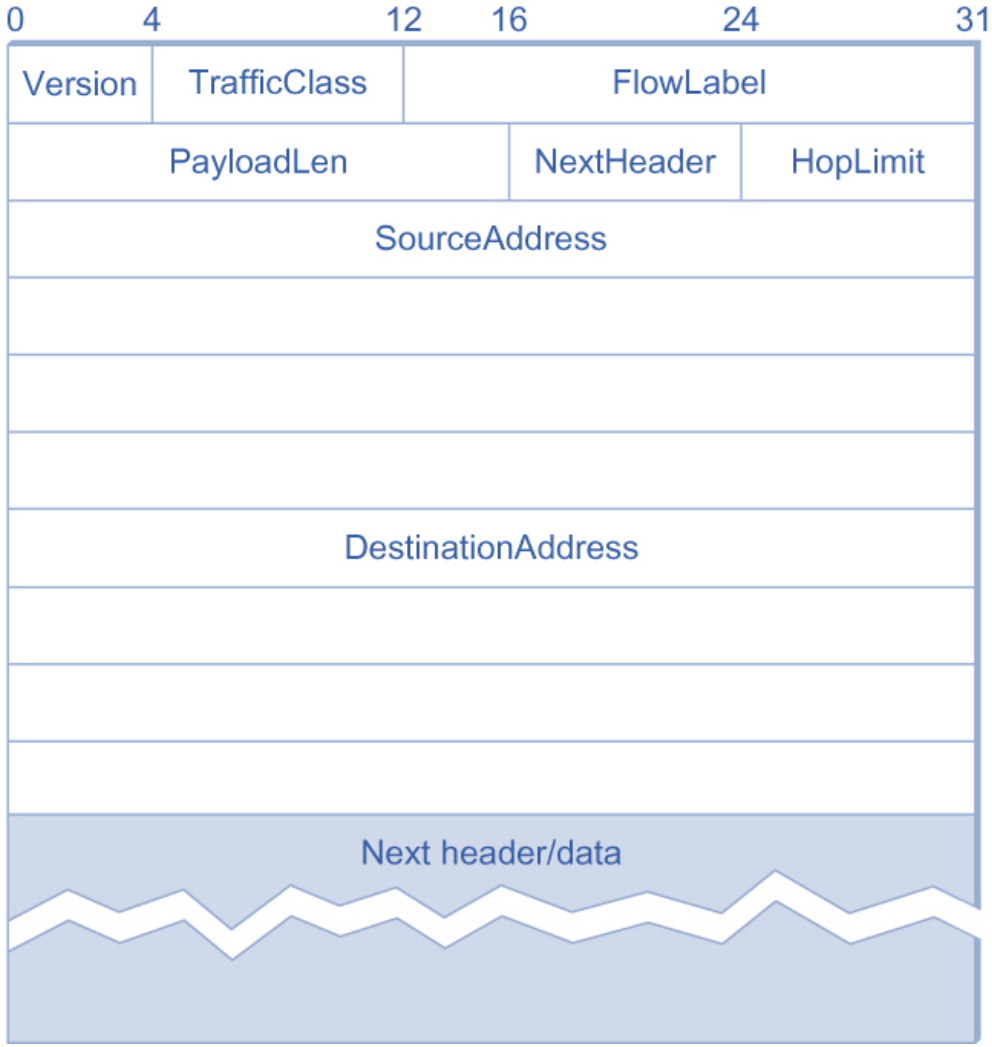
- IPv6
- NAT
- MPLS
- Router Implementation

IPv6 Address

- 16 bytes
  - 1500 addresses per square foot (Earth's surface)
- Classless addressing/routing (similar to CIDR)
  - Notation: x:x:x:x:x:x:x:x (x = 16-bit hex number)
  - Contiguous 0s are compressed: 47CD::A456:0124
  - IPv4-mapped IPv6 address: ::FFFF:123.45.67.8
- Address assignment: more hierarchy

IP Version 6 (IPv6)

- Motivation
  - 32 bits IPv4 Address is not enough
  - Other Features
    - Stateless auto configuration
    - Source Routing
- IPv6 “base” Header
  - 40 bytes “base” header
  - 16 bytes addresses
- Extension headers
  - Fragmentation
  - Source routing
  - Authentication and security
  - etc.



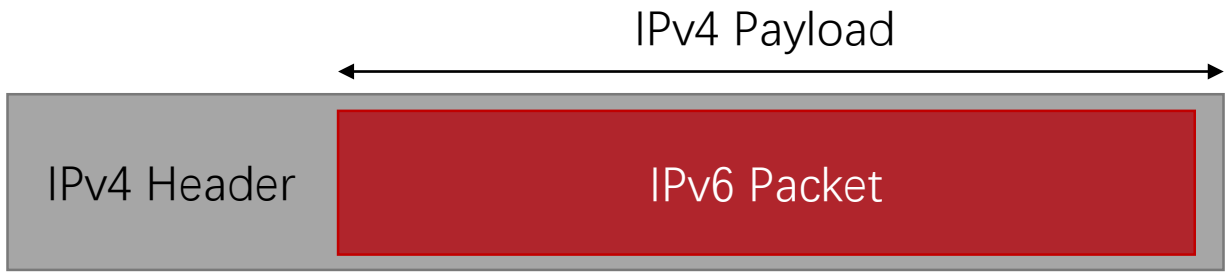
Transition from IPv4 to IPv6

- IETF began looking at the problem of IPv4 address space in 1991
- Not all routers can be upgraded simultaneously

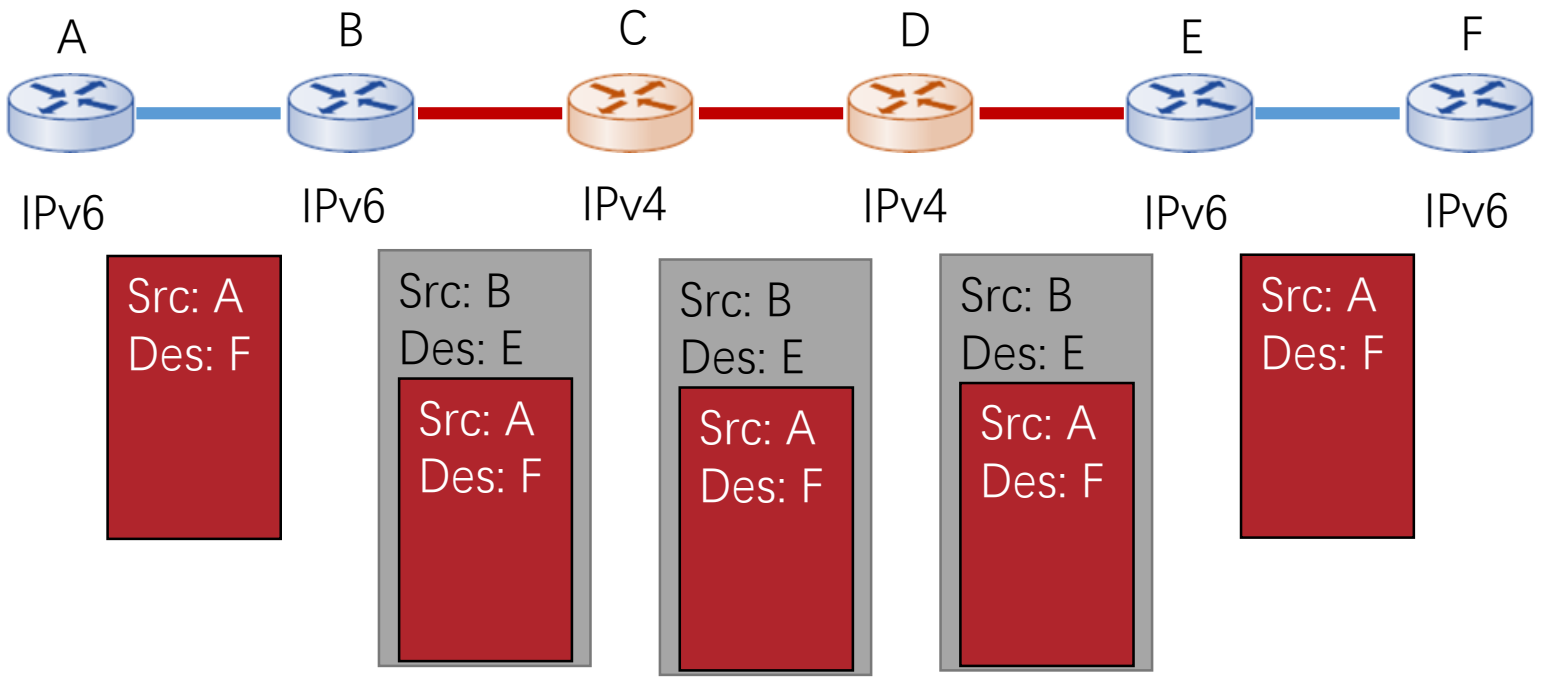
How will network operate with mixed IPv4 and IPv6 routers?

Transition from IPv4 to IPv6

- Tunneling: IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers

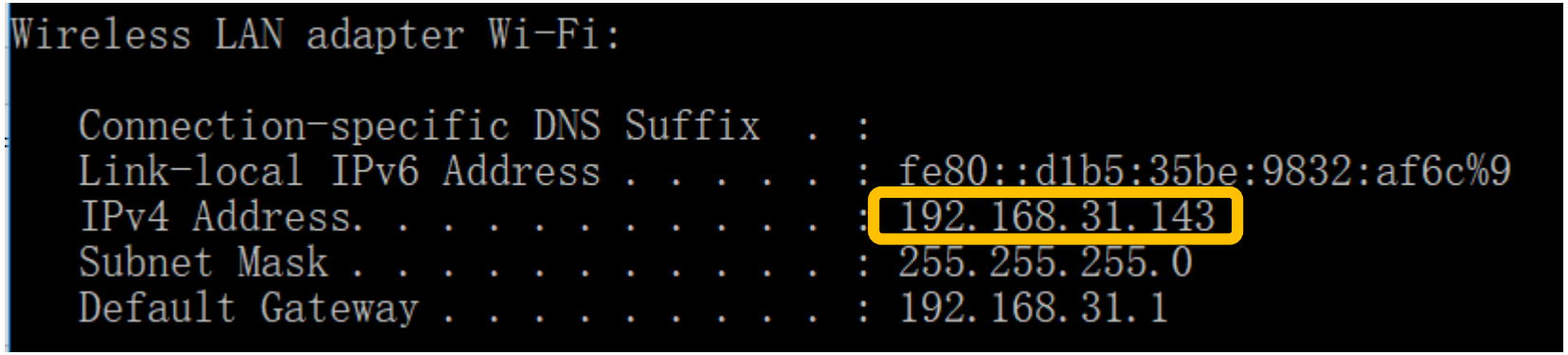


Transition from IPv4 to IPv6



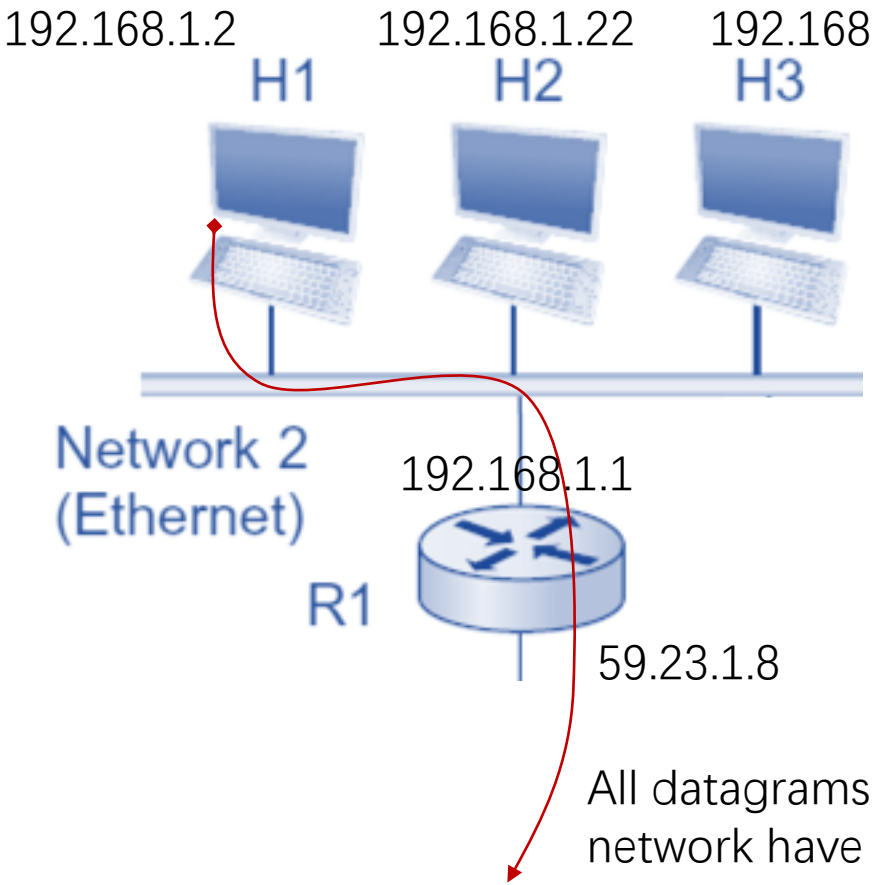
Why haven’t IPv6 replaced IPv4 ?

NAT: Network Address Translation



Address range	Subnet mask	Provides	Addresses per LAN
10.0.0.0 - 10.255.255.255.255	255.0.0.0	1 class A LAN	16,777,216
172.16.0.0 - 172.31.255.255	255.255.0.0	16 class B LANs	65,536
192.168.0.0 - 192.168.255.255	25.255.255.0	256 class C LANs	256

NAT: Network Address Translation

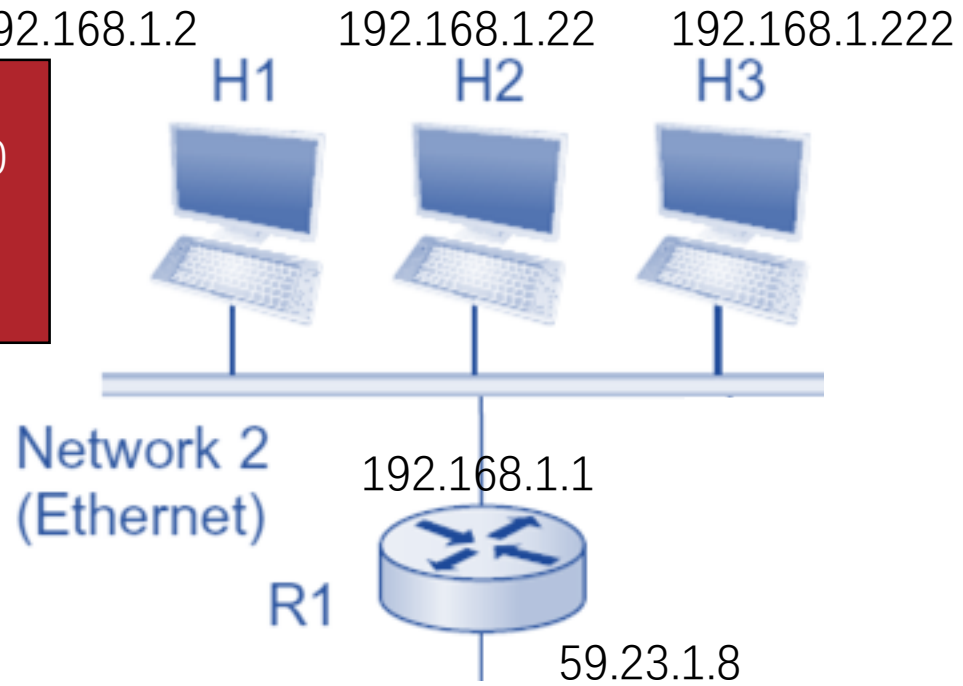


datagrams with source or destination in this network have 192.168.1.0/24 address for source, destination (as usual)

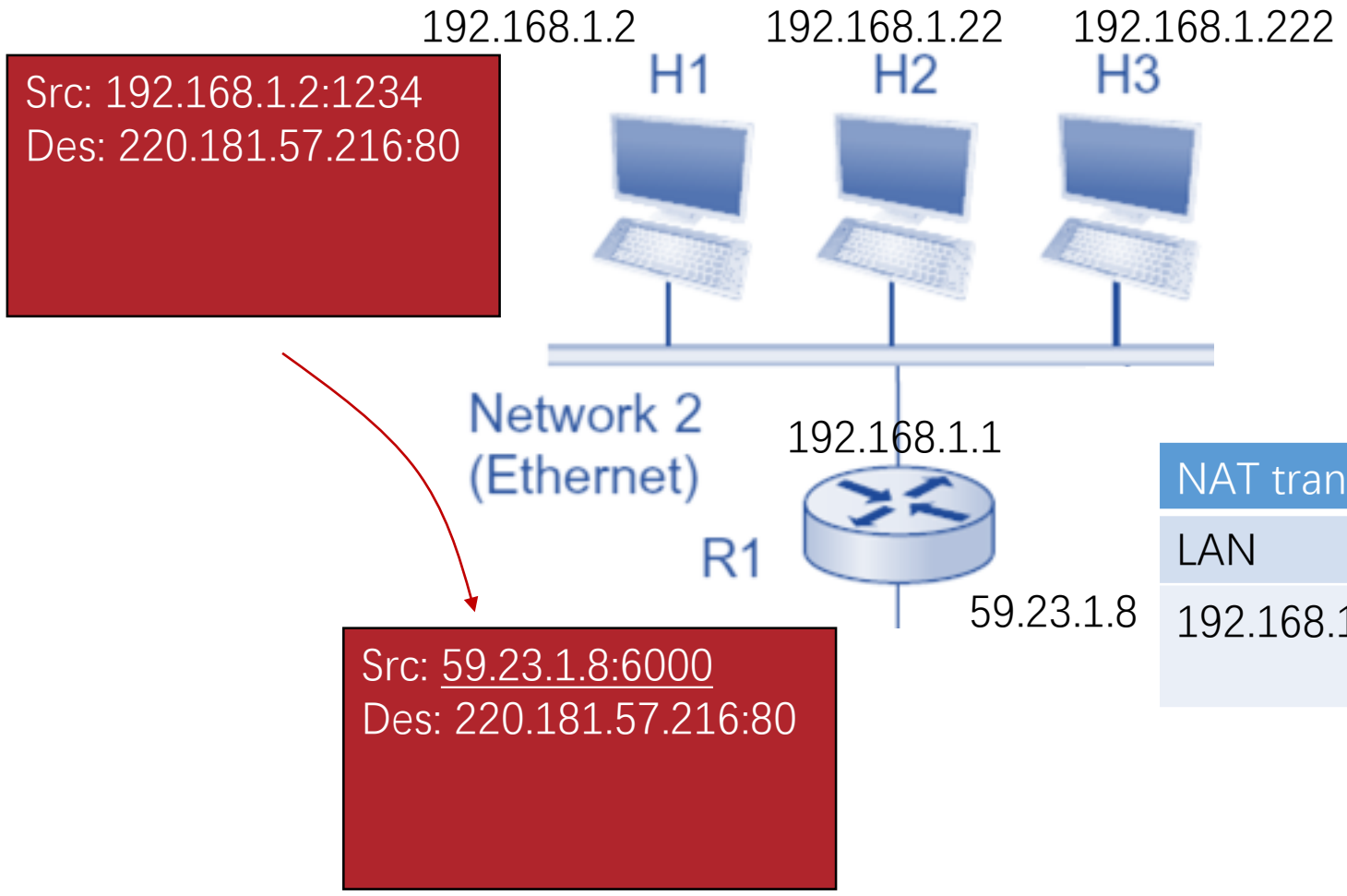
All datagrams leaving local network have same single source NAT IP address: 59.23.1.8, different source port numbers

NAT: Network Address Translation

Src: 192.168.1.2:1234
Des: 220.181.57.216:80

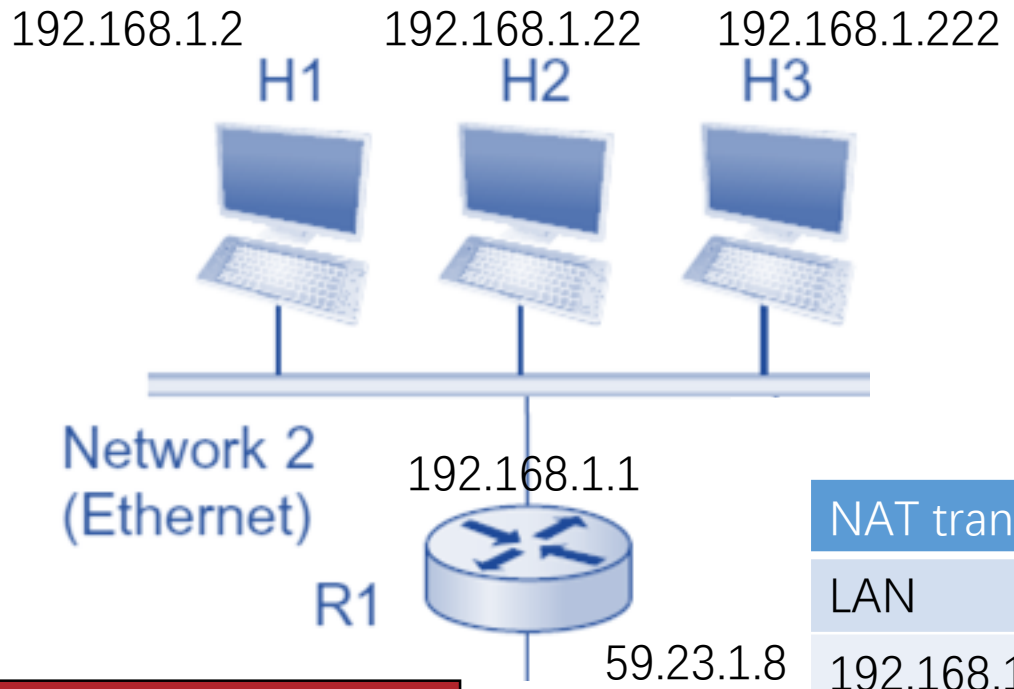


NAT: Network Address Translation



NAT translation table	
LAN	WAN
192.168.1.2:1234	59.23.1.8:6000

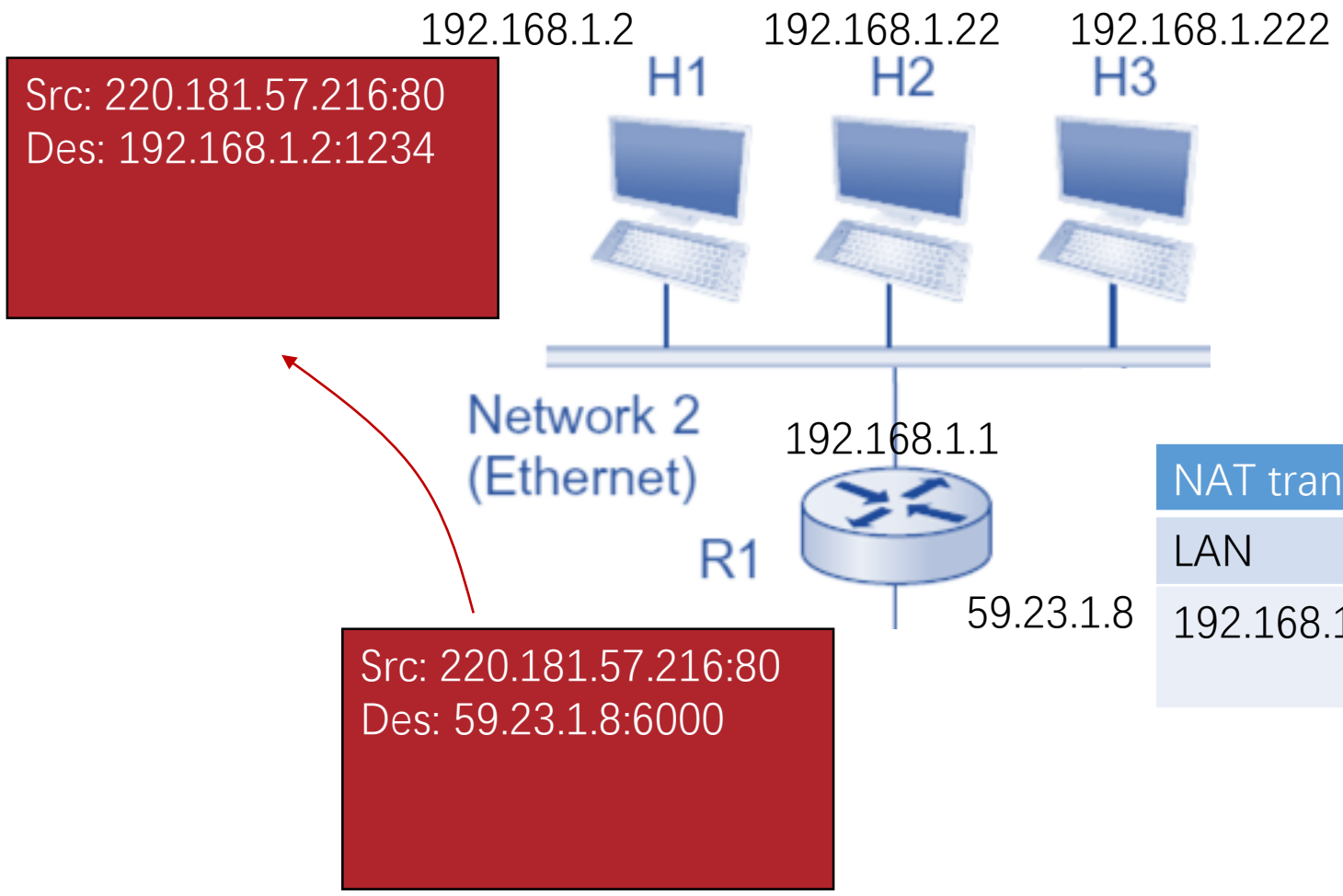
NAT: Network Address Translation



NAT translation table	
LAN	WAN
192.168.1.2:1234	59.23.1.8:6000

Src: 220.181.57.216:80
Des: 59.23.1.8:6000

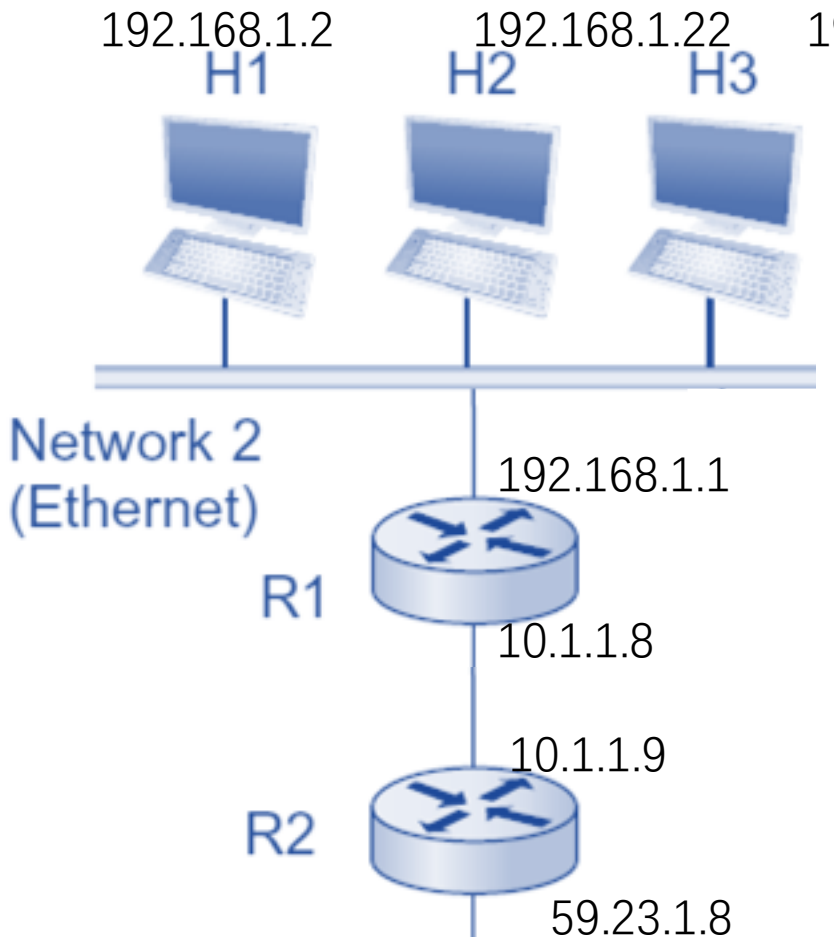
NAT: Network Address Translation



NAT translation table	
LAN	WAN
192.168.1.2:1234	59.23.1.8:6000

NAT: Network Address Translation

- Multi Layer NAT



NAT translation table R1	
LAN	WAN
192.168.1.2:1234	10.1.1.8:6321

NAT translation table R2	
LAN	WAN
10.1.1.8:6321	59.23.1.8:6000

NAT: Network Address Translation

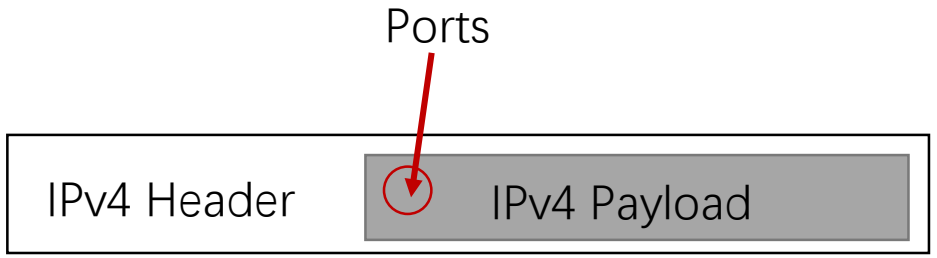
- Demo

```
Tracing route to baidu.com [111.13.101.208]
over a maximum of 30 hops:

  1     5 ms     2 ms    <1 ms  XiaoQiang [192.168.31.1]
  2     2 ms     2 ms     3 ms  10.19.14.1
  3    11 ms     1 ms     2 ms  10.13.7.25
  4     *        *        *     Request timed out.
```

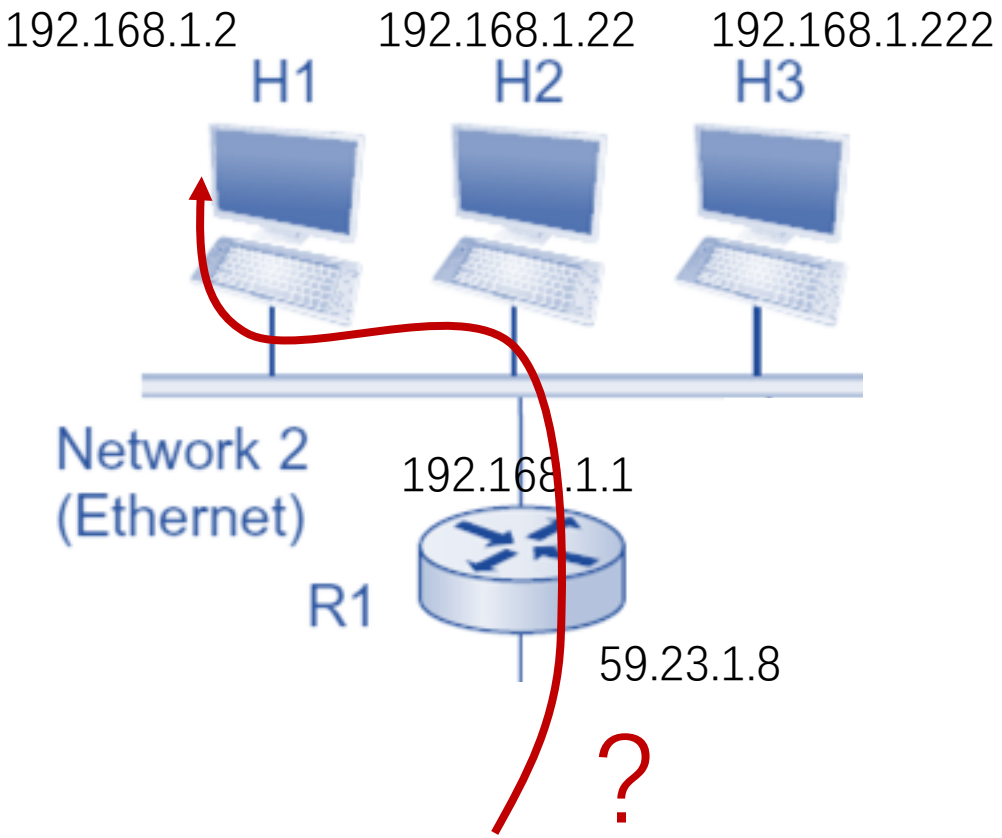
NAT: Network Address Translation

- 16-bit port-number field (see later lecture)
  - 60,000 simultaneous connections with a single IPv4 address
  - Hosts uses LAN addresses
    - 192.168.0.0/16
    - 10.0.0.0/8
    - 172.16.0.0/12
    - etc.
- Problems
  - NAT is “impure”
    - Routers should not touch higher layers
  - Efficiency
  - Traversal Connections



NAT Traversal Problem

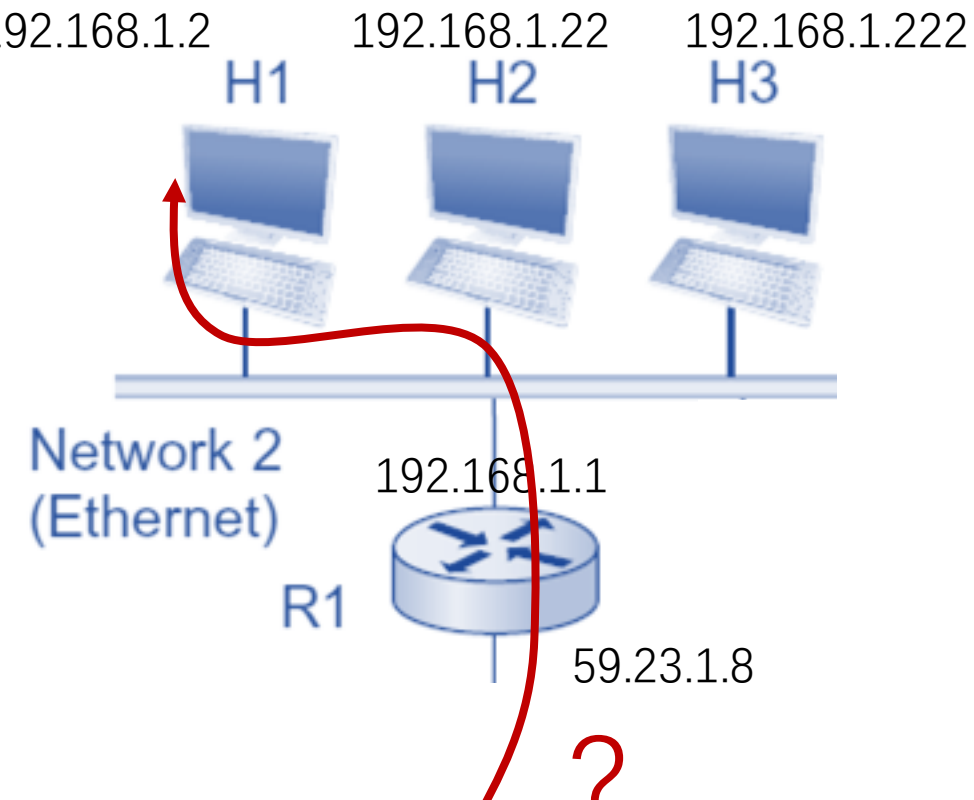
- How to initiate connections to Host 1 from external network ?
  - e.g. IP Phone



NAT Traversal Problem

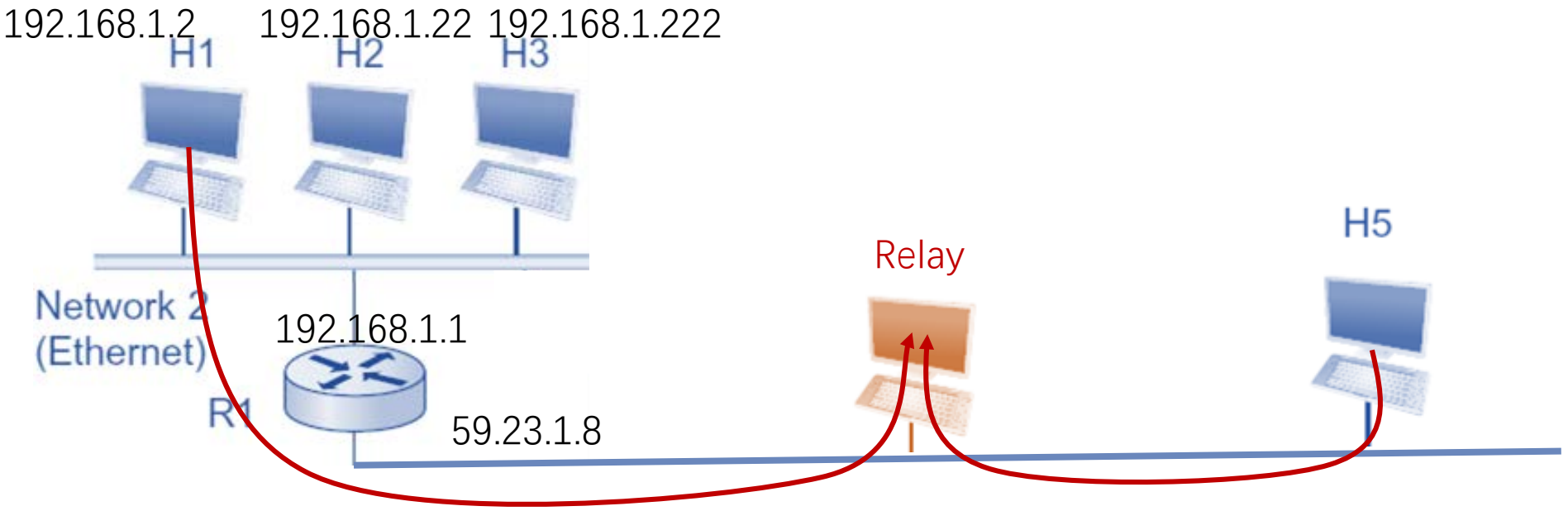
- Solution 1: Static Configure
  - Configure NAT to forward incoming connection requests at given port to the host
  - Need to know the port

NAT translation table	
LAN	WAN
192.168.1.2	59.23.1.8:6000



NAT Traversal Problem

- Solution 2: Relay
  - NATed host establishes connection to relay
  - External host connects to relay
  - Relay bridges packets between to connections



Outline

- IPv6
- NAT
- MPLS
- Router Implementation

Longest Prefix Matching

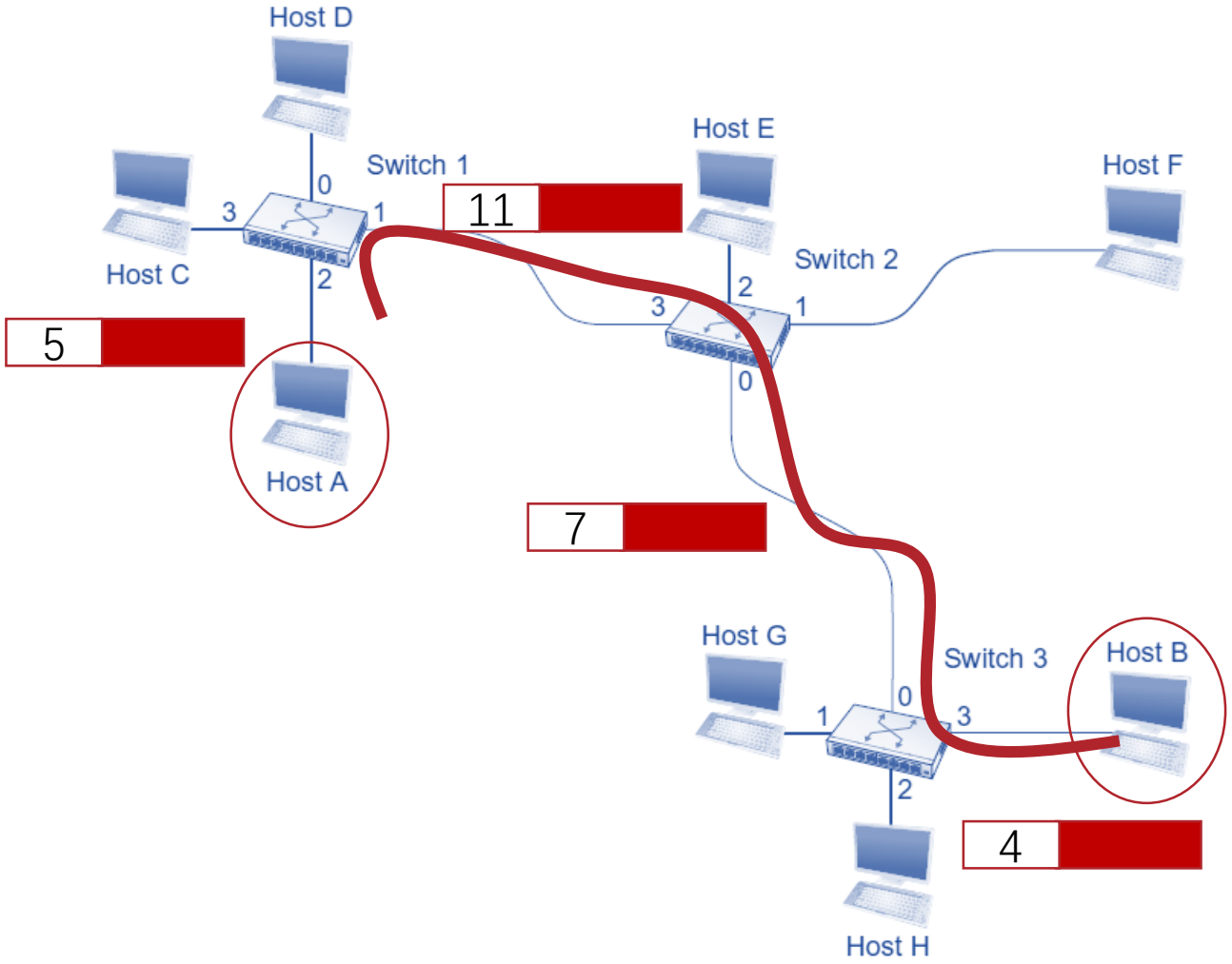
- When looking for forwarding table entry for given destination address, use longest address prefix that matches destination address.

SubnetNum	NextHop	
197.168.0.0/22	R4	11000101.10101000.000000**.********
197.168.3.0/24	R7	11000101.10101000.00000011.********
197.168.4.0/22	R9	11000101.10101000.000001**.********

How to Accelerate Prefix Matching in Backbone Routers ?

11000101.10101000.00000111.11010111 → R9

Virtual Circuit

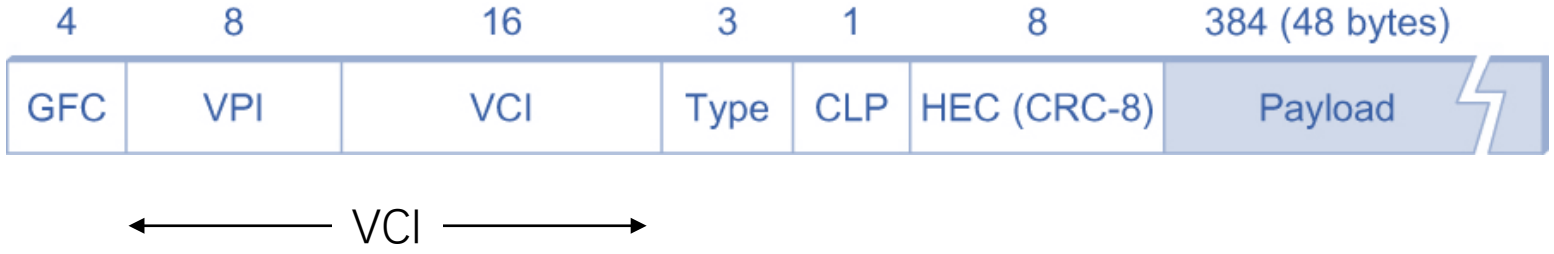


Virtual Circuit Table

Switch1			
Incoming Interface	Incoming VCI	Outgoing Interface	Incoming VCI
2	5	1	11
Switch2			
Incoming Interface	Incoming VCI	Outgoing Interface	Incoming VCI
3	11	0	7
Switch3			
Incoming Interface	Incoming VCI	Outgoing Interface	Incoming VCI
0	7	3	4

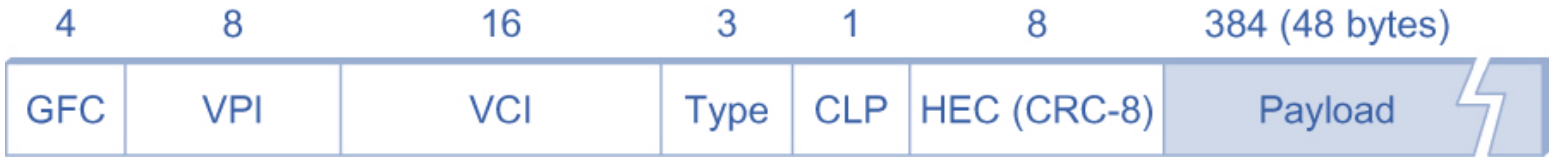
Asynchronous Transfer Mode (ATM)

- Evolved from telephony
  - Connection-oriented packet-switched network
  - Guaranteed Service: strict timing, reliability, etc.
- Packets are called cells



ATM Switch

- Packet Length is Fixed
  - Easier to switch in hardware
- VCI Length is Fixed
  - VCI looking up complexity is O(1)

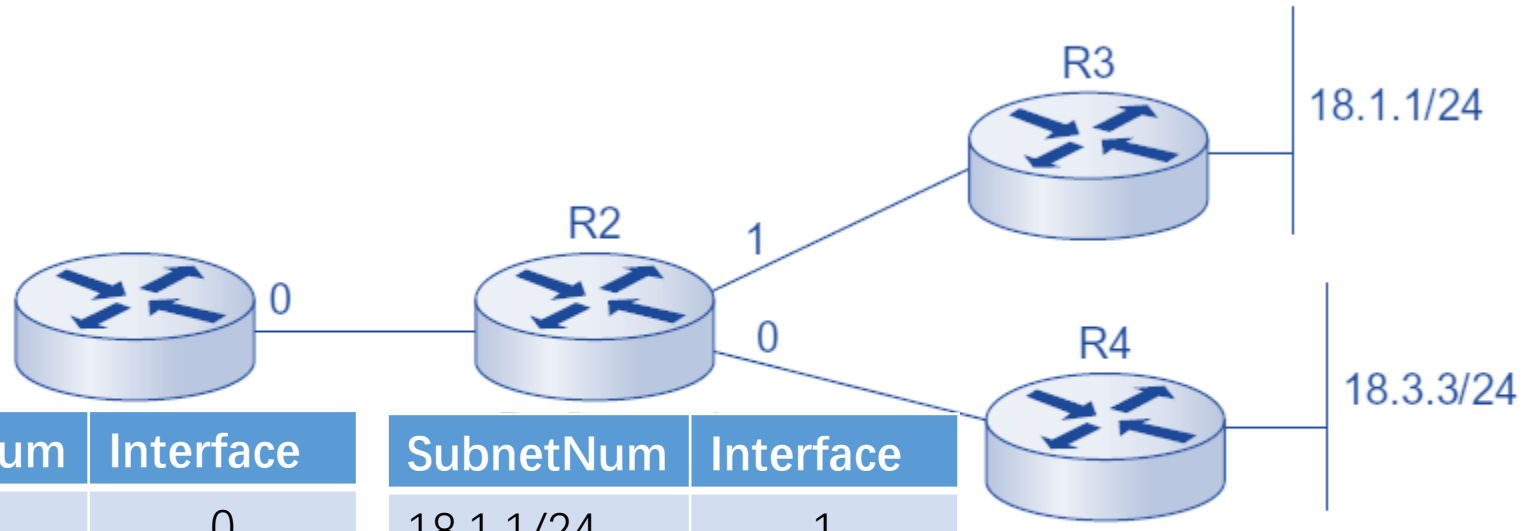


Multiprotocol Label Switching (MPLS)

- Original Motivation
  - Enable IP forwarding on devices that do not have IP forwarding ability (i.e., ATM Switches)
    - Capability (Support IP in ATM)
    - Speed
- Benefit: Destination-Based Forwarding

MPLS: Destination-Based Forwarding

- IP Forwarding with Normal Routers

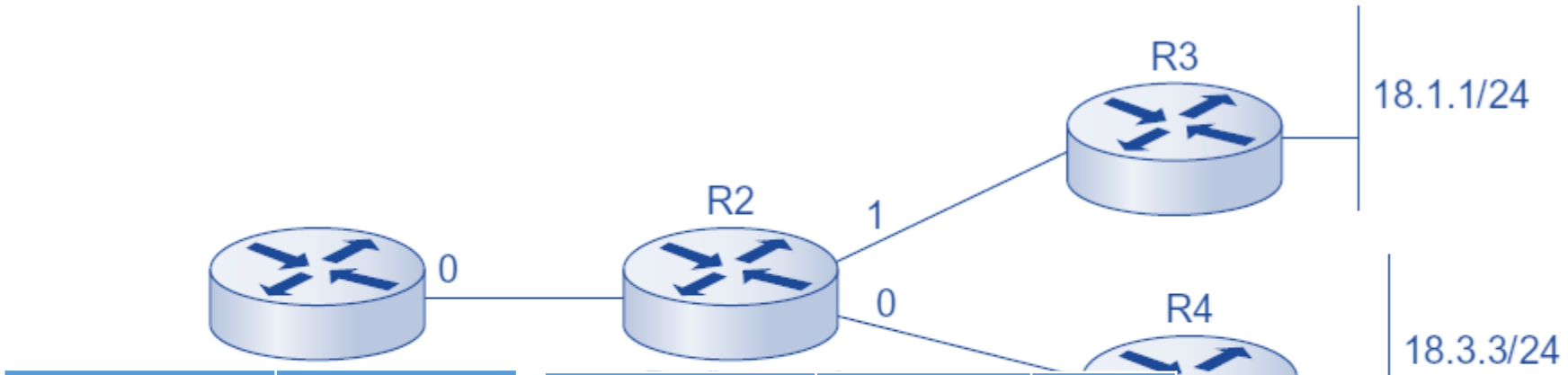


SubnetNum	Interface
18.1.1/24	0
18.3.3/24	0

SubnetNum	Interface
18.1.1/24	1
18.3.3/24	0

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels



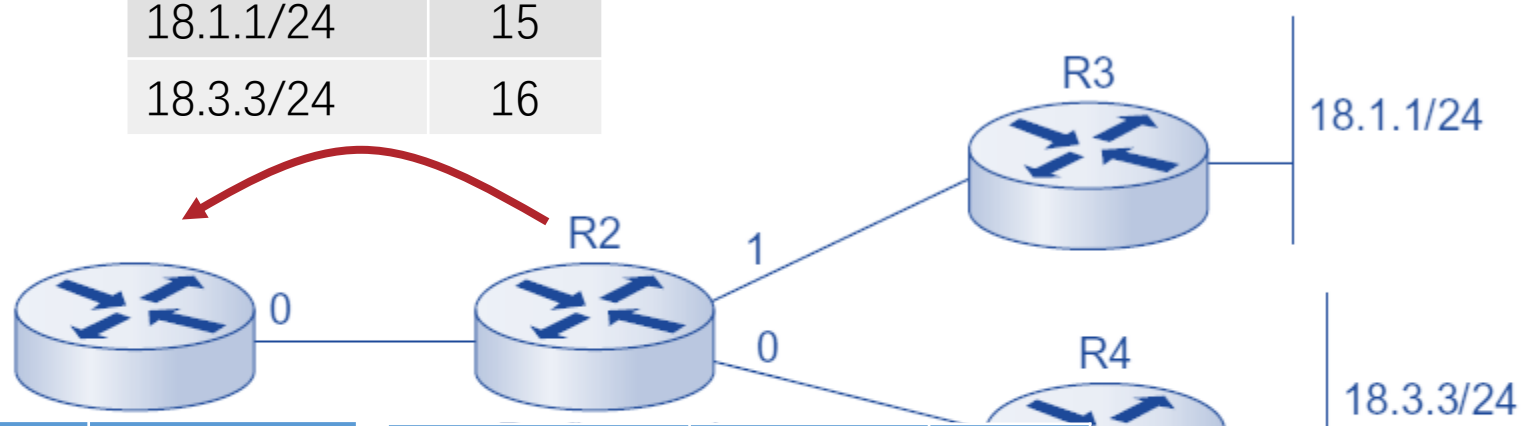
SubnetNum	Interface
18.1.1/24	0
18.3.3/24	0

SubnetNum	Interface	Label
18.1.1/24	1	15
18.3.3/24	0	16

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels

SubnetNum	Label
18.1.1/24	15
18.3.3/24	16

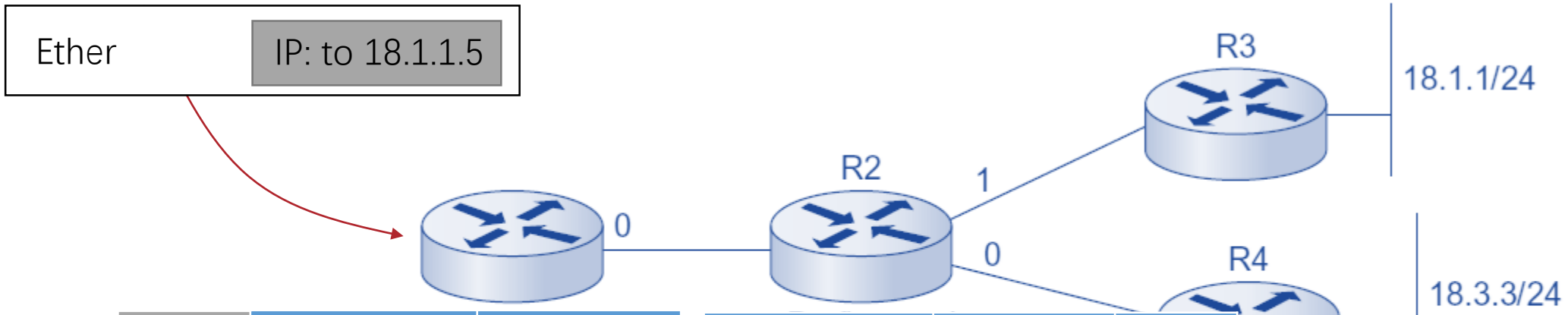


SubnetNum	Interface
18.1.1/24	0
18.3.3/24	0

SubnetNum	Interface	Label
18.1.1/24	1	15
18.3.3/24	0	16

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels

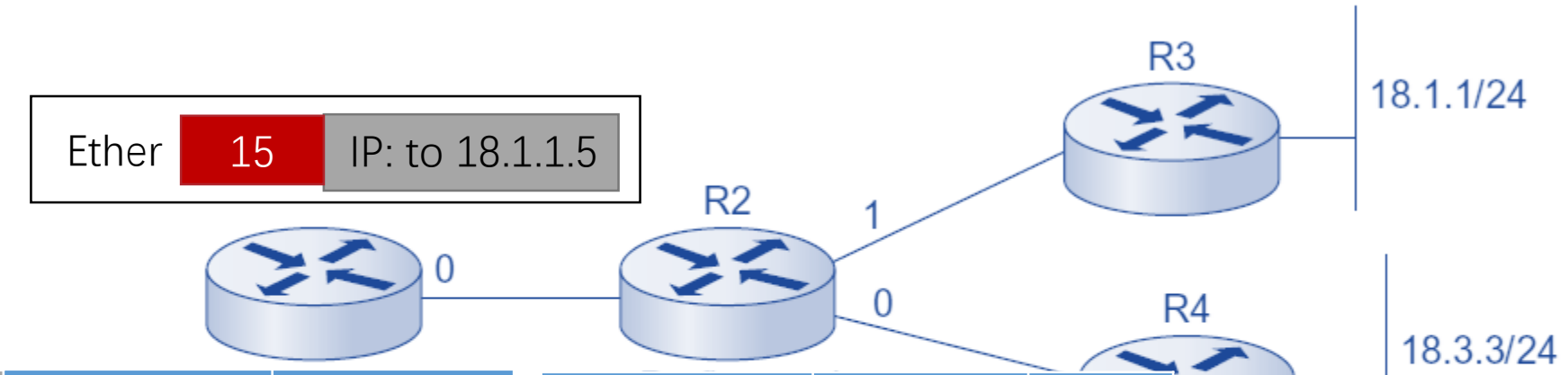


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label
18.1.1/24	1	15
18.3.3/24	0	16

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels

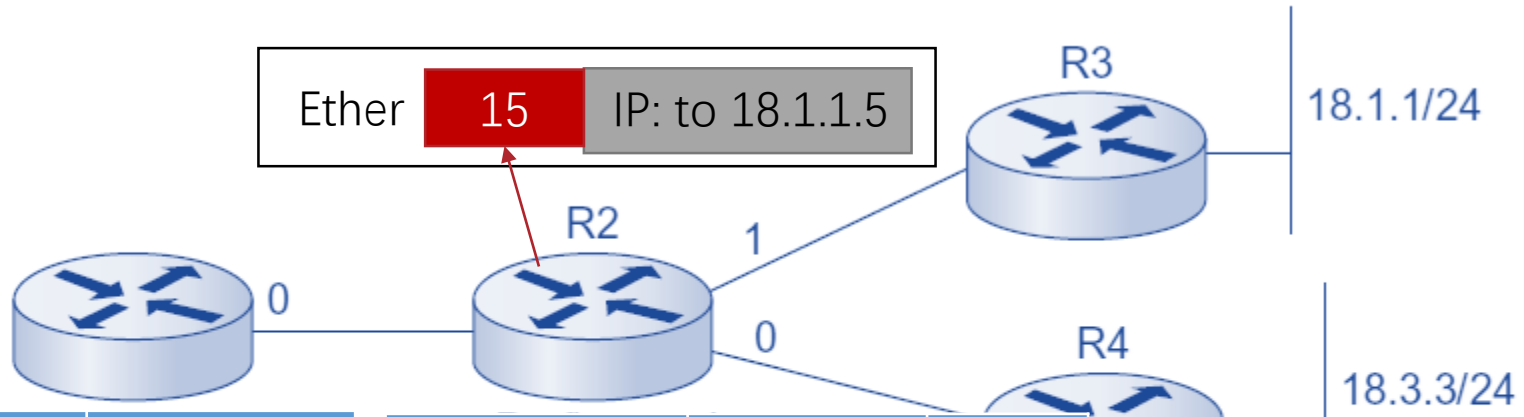


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label
18.1.1/24	1	15
18.3.3/24	0	16

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels

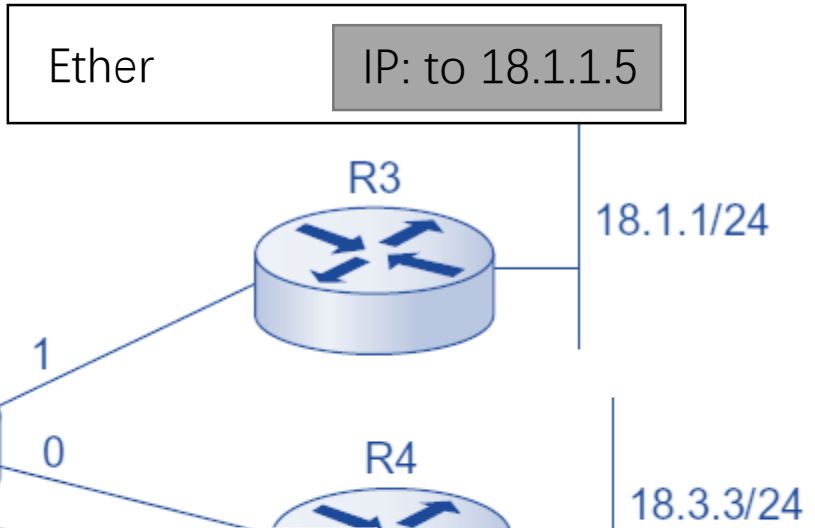


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label
18.1.1/24	1	15
18.3.3/24	0	16

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels



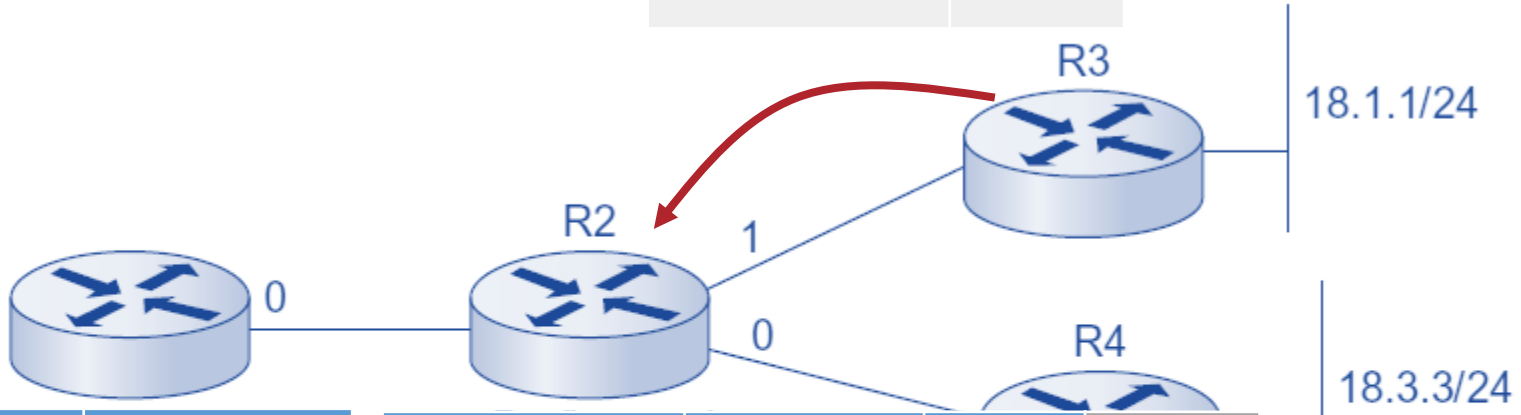
Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label
18.1.1/24	1	15
18.3.3/24	0	16

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels
  - Use labels along the path

SubnetNum	Label
18.1.1/24	24

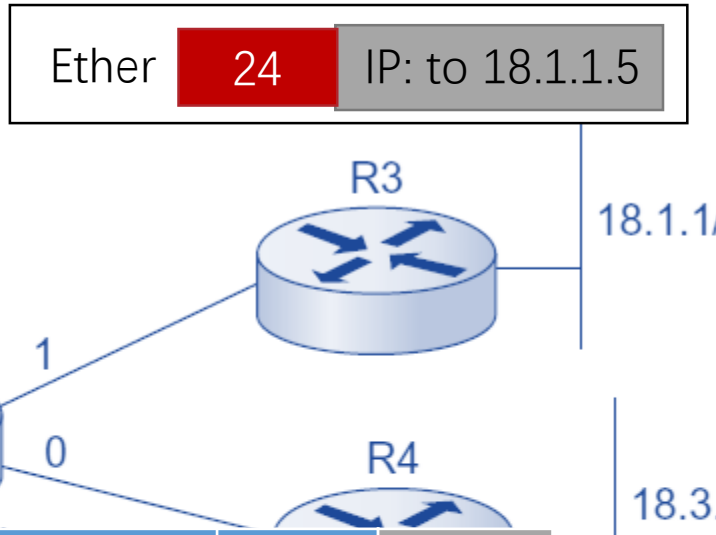


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label	Label
18.1.1/24	1	15	24
18.3.3/24	0	16	

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels
  - Use labels along the path

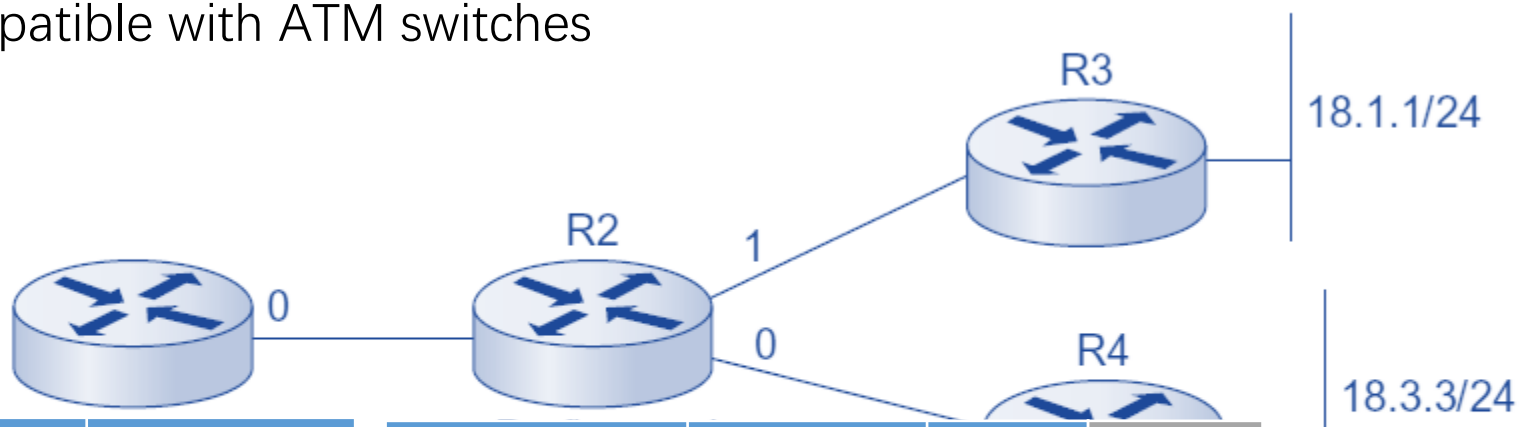


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label	Label
18.1.1/24	1	15	24
18.3.3/24	0	16	

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels
  - The forwarding hardware is based on label
    - Compatible with ATM switches

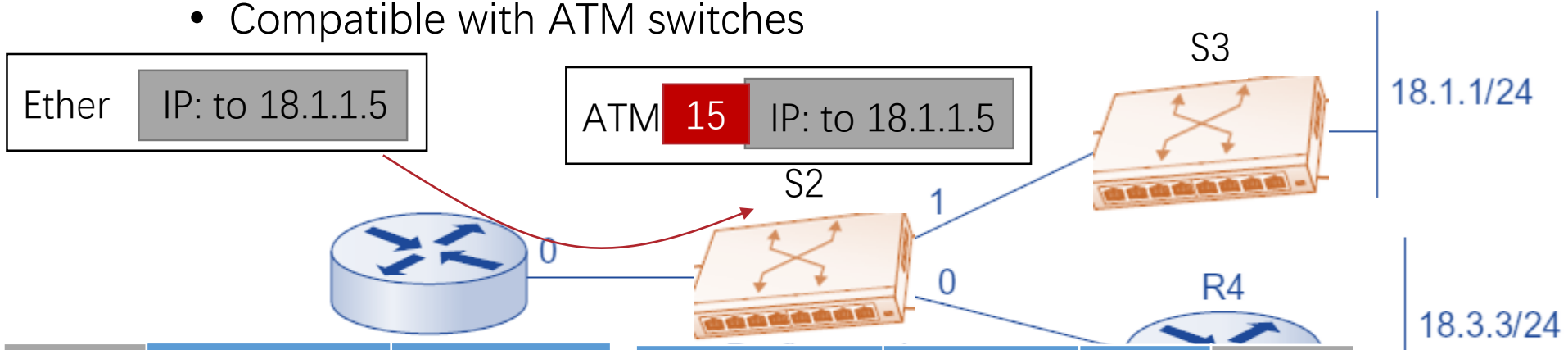


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label	Label
18.1.1/24	1	15	24
18.3.3/24	0	16	

MPLS: Destination-Based Forwarding

- IP Forwarding with Labels
  - The forwarding hardware is based on label
    - Compatible with ATM switches

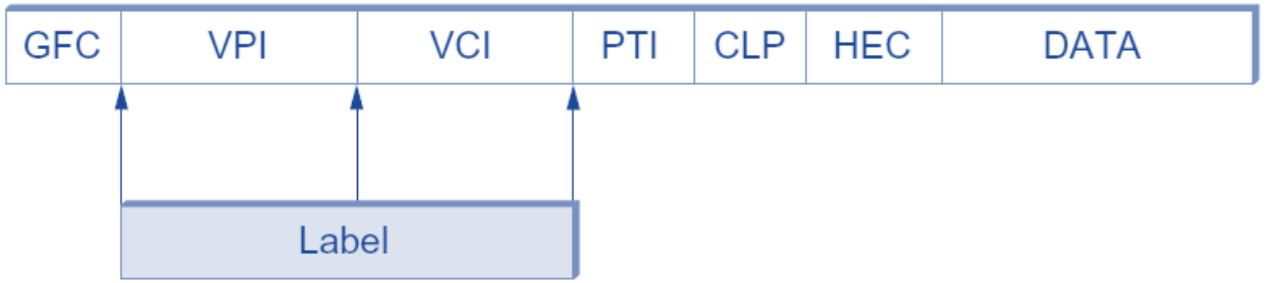


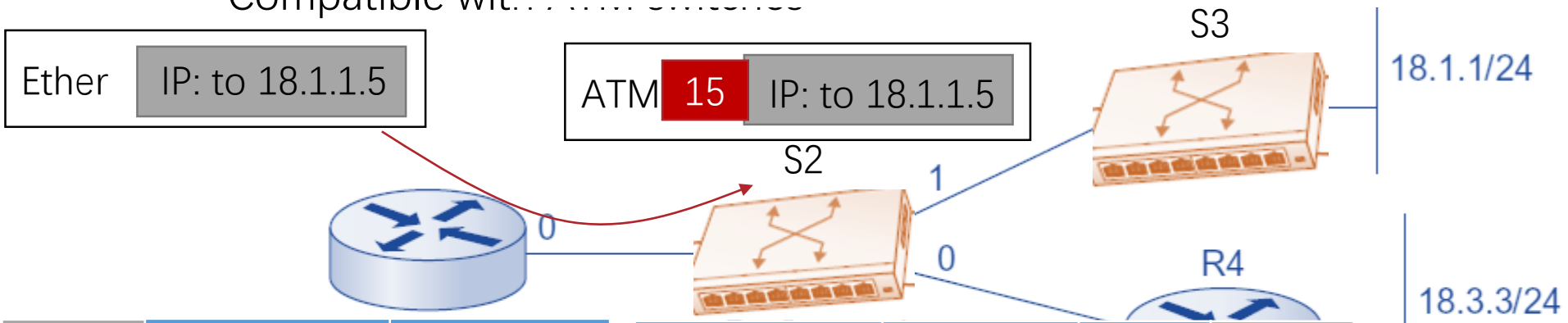
Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label	Label
18.1.1/24	1	15	24
18.3.3/24	0	16	

MPLS: Destination-Based Forwarding

- IP Forwarding wi
  - The forwarding
    - Compatible wit



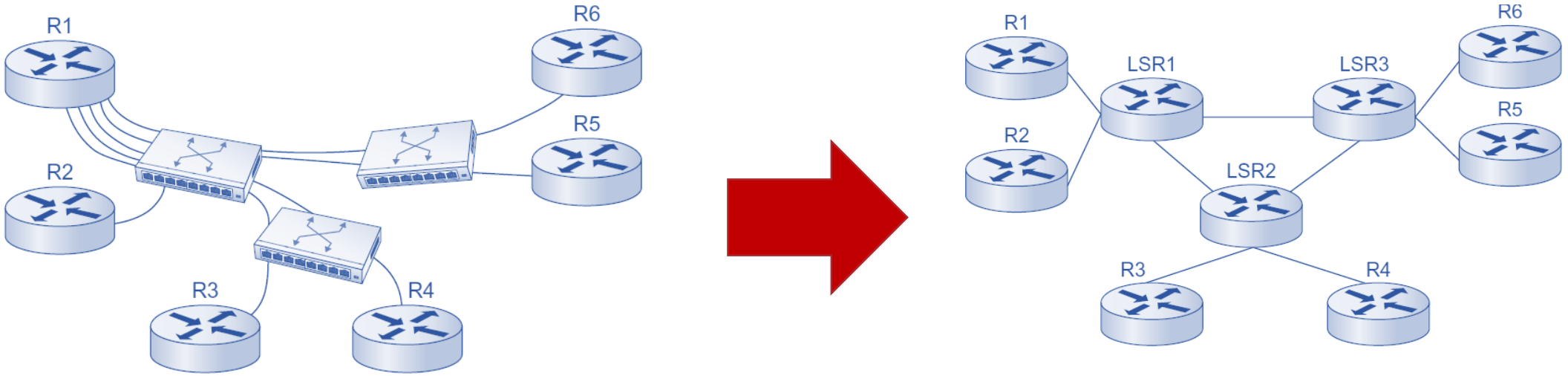


Label	SubnetNum	Interface
15	18.1.1/24	0
16	18.3.3/24	0

SubnetNum	Interface	Label	Label
18.1.1/24	1	15	24
18.3.3/24	0	16	

MPLS: Destination-Based Forwarding

- Transform ATM switches into IP Routers
  - Only software changes are needed
  - IP routers connected by ATM network are able to see better network topoloty

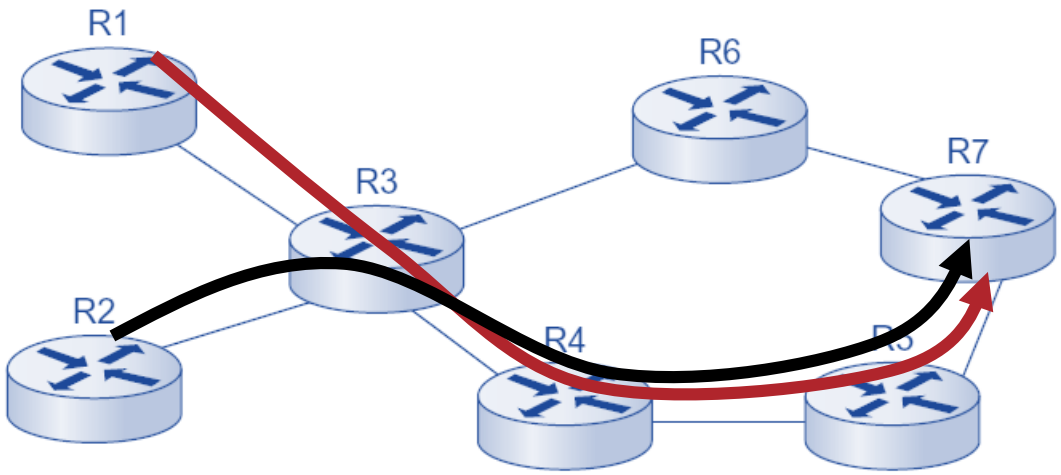


MPLS: Explicit Routing

- MPLS Forwarding Decisions
  - Can follow IP routing algorithm
    - Same path as IP forwarding
  - Can differ from IP routing algorithm
- Application
  - Traffic engineering
    - Use destination and source addresses to route flows to same destination differently
  - Fast reroute
    - pre-computed backup paths

MPLS: Explicit Routing

- IP Routing: path to destination determined by destination address alone
  - Contains no information about the traffic of certain path



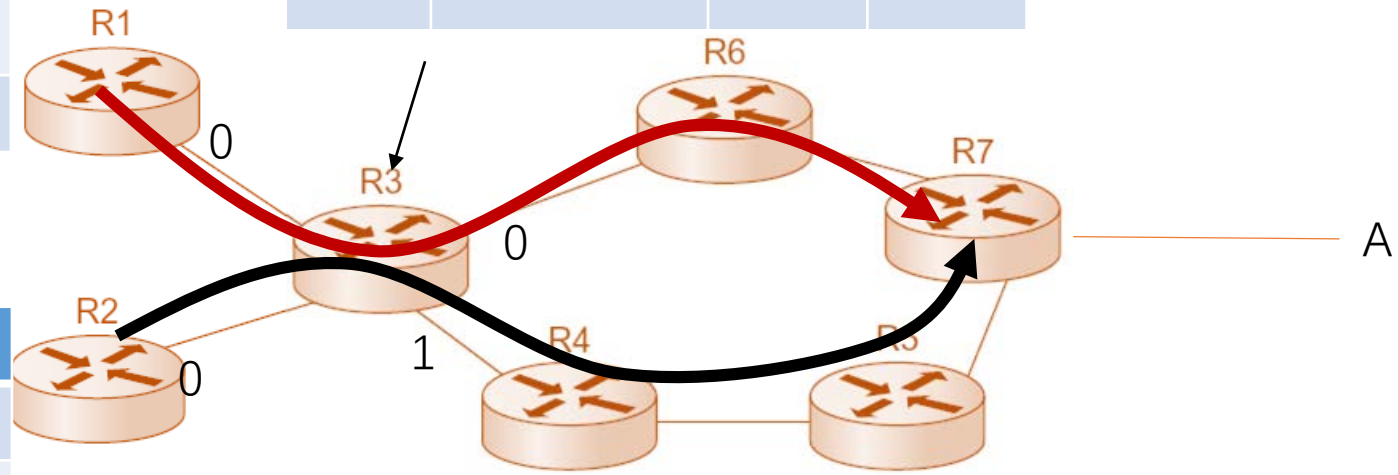
MPLS: Explicit Routing

- MPLS routing: path to destination can be based on source and destination address

Dest	Interface	L_IN	L_OUT
A	0	/	25

Dest	Interface	L_IN	L_OUT
A	0	25	3
A	1	12	4

Dest	Interface	L_IN	L_OUT
A	0	/	12

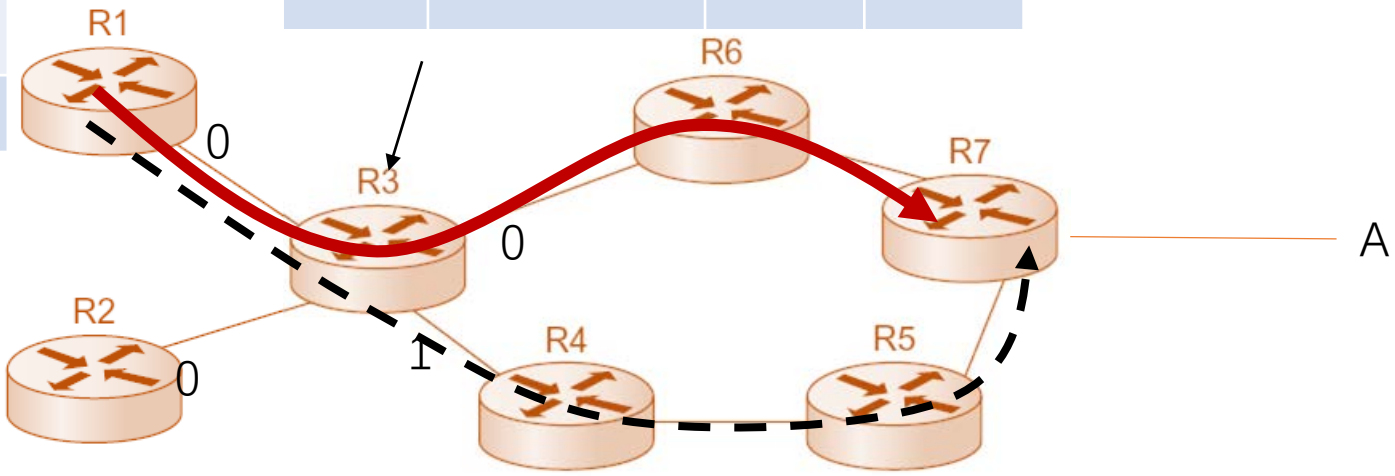


MPLS: Explicit Routing

- MPLS routing: path to destination can be based on source and destination address

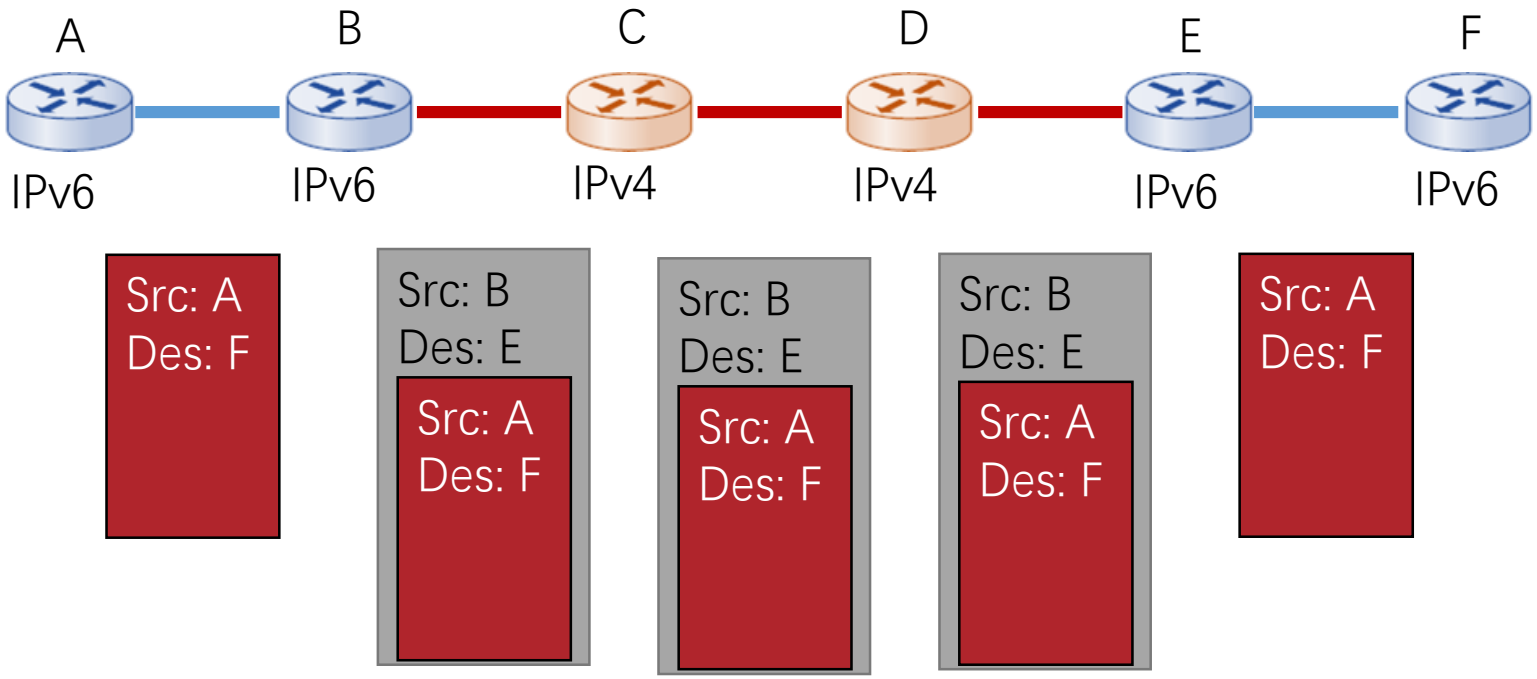
Dest	Interface	L_IN	L_OUT
A	0	/	25
A	0	/	26

Dest	Interface	L_IN	L_OUT
A	0	25	3
A	1	26	4



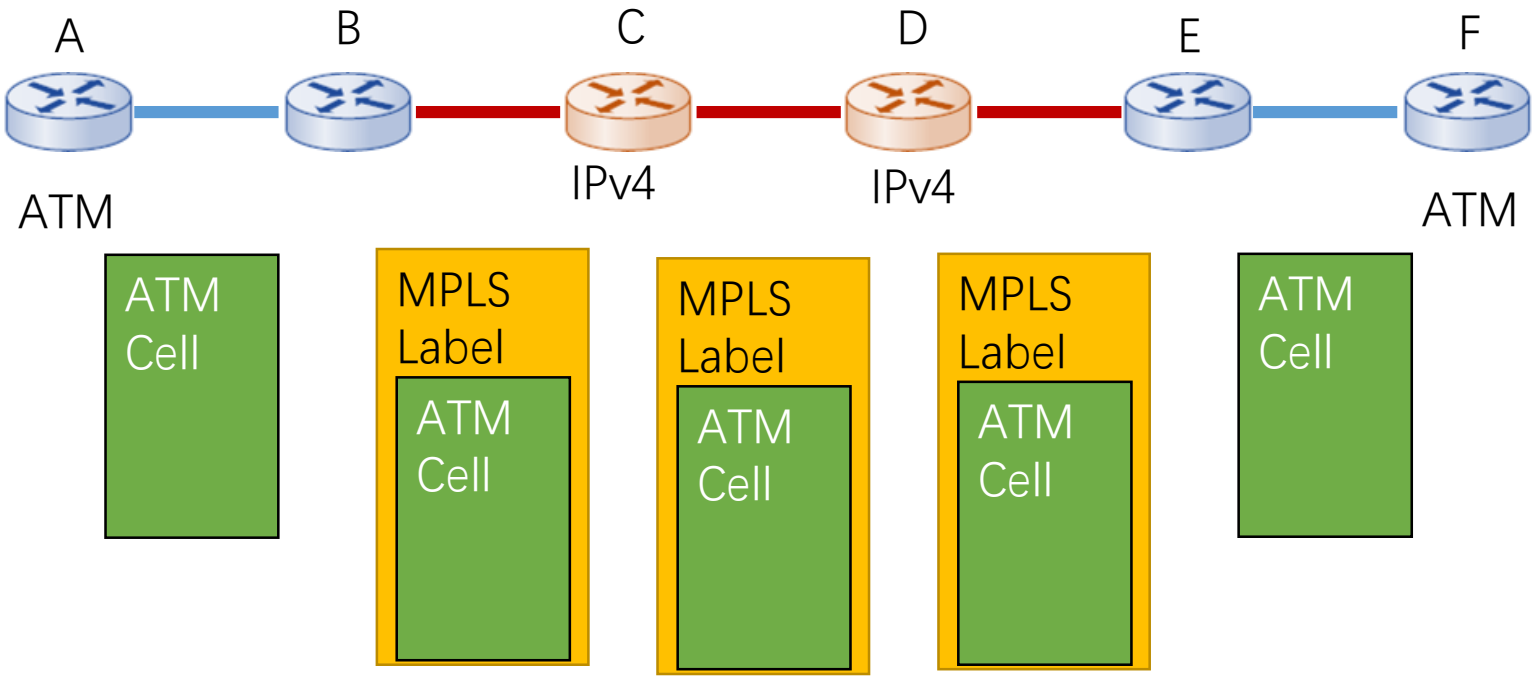
MPLS: Tunneling

- IPv4 Tunneling



MPLS: Tunneling

- MPLS Tunneling

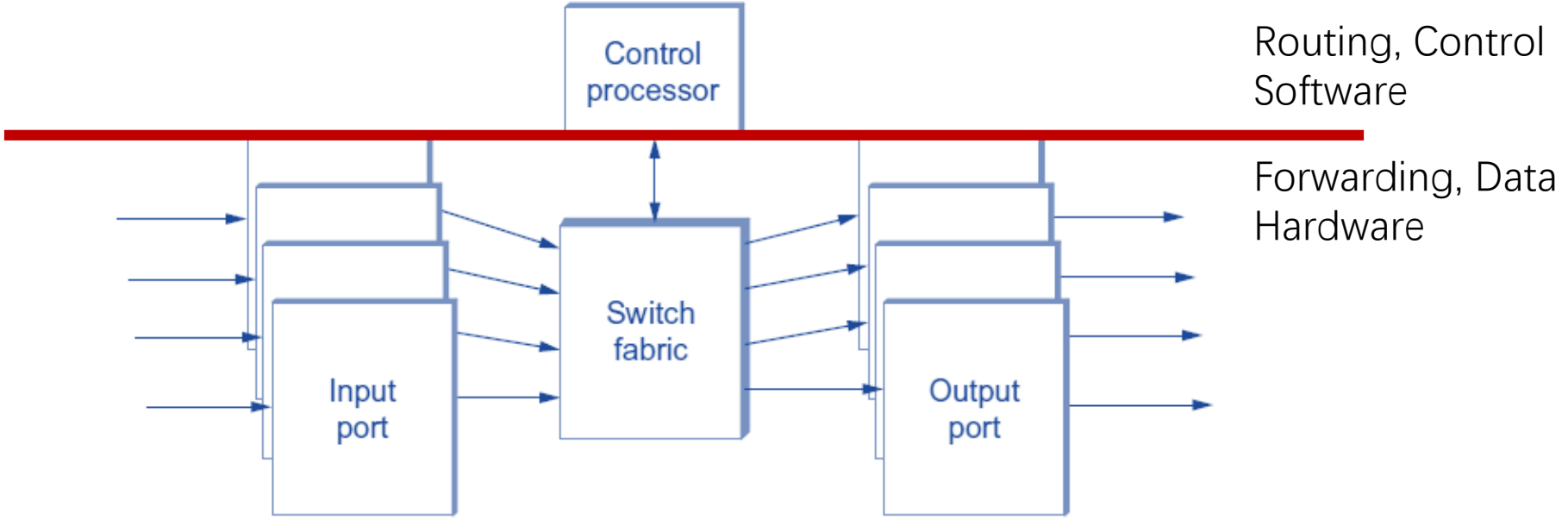


Centralize or Decentralize ?

- That is the question

Router Architecture

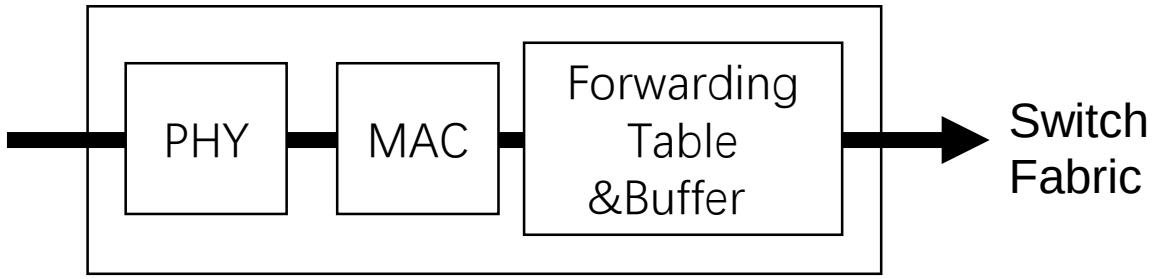
- Two Key Functions:
  - Routing algorithms (e.g, RIP, OSPF, etc.)
  - Forwarding packets from input to output ports
- Performance: packet per second



Control Processor

- Function
  - Control
    - Ports and switch fabrics
  - Calculation
    - Routing Algorithm (Router)
      - Push forwarding tables into ports
    - Switches does not run routing algorithms

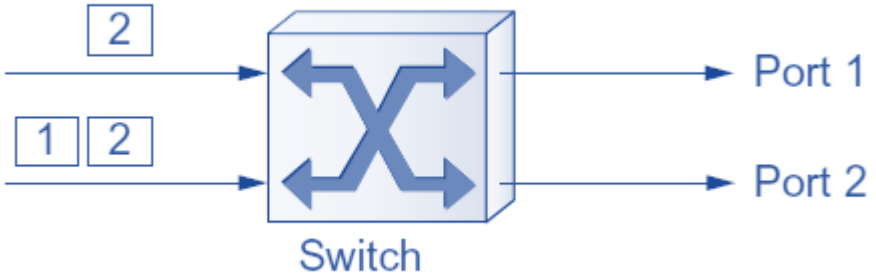
Input Port



- Deliver Packet to Fabric
  - Handle Different Protocols
- Output Port Looking Up
  - Given packet destination lookup output port
    - Ethernet
      - Using table: destination MAC ⇔ port
    - ATM
      - Using table: VCI ⇔ port
  - Goal: complete input port processing at “line speed”
    - 2.48Gbps link, 64byte packet
      - 4.83*10^6 pps
- Buffer

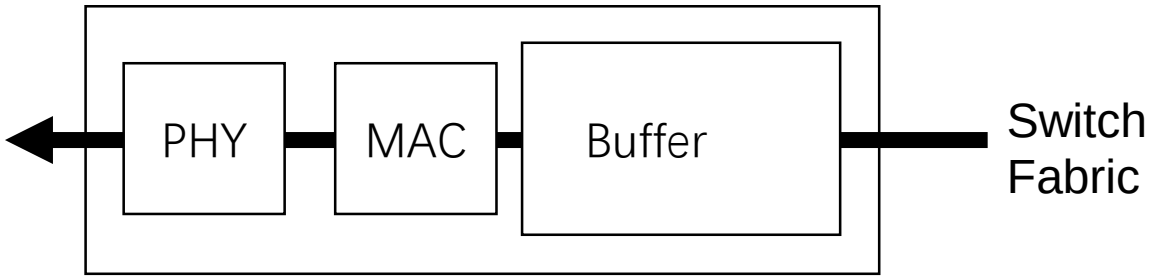
Input Port

- Head-of-line Blocking

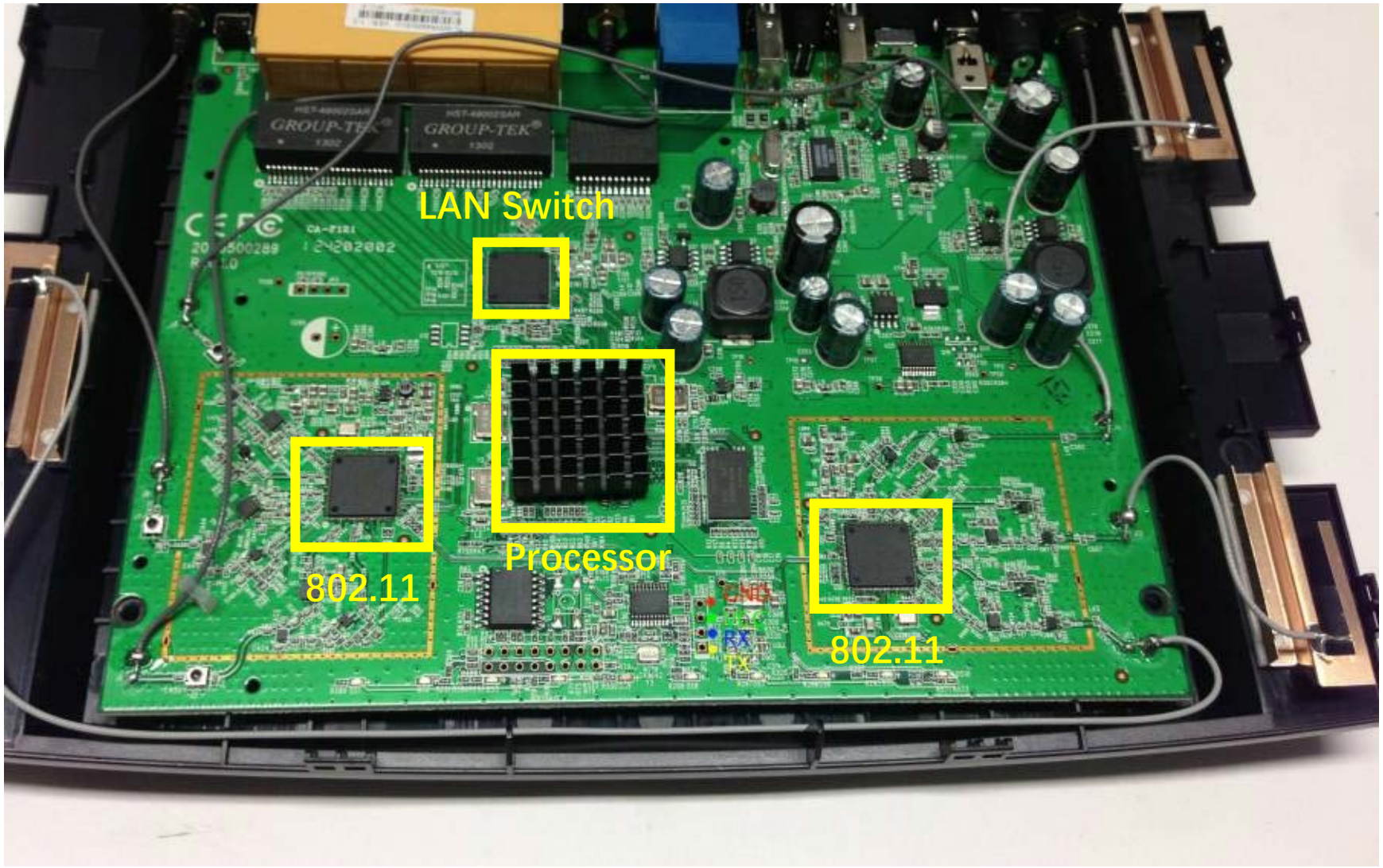


Input Port

- Deliver Packet to Network
- Buffer

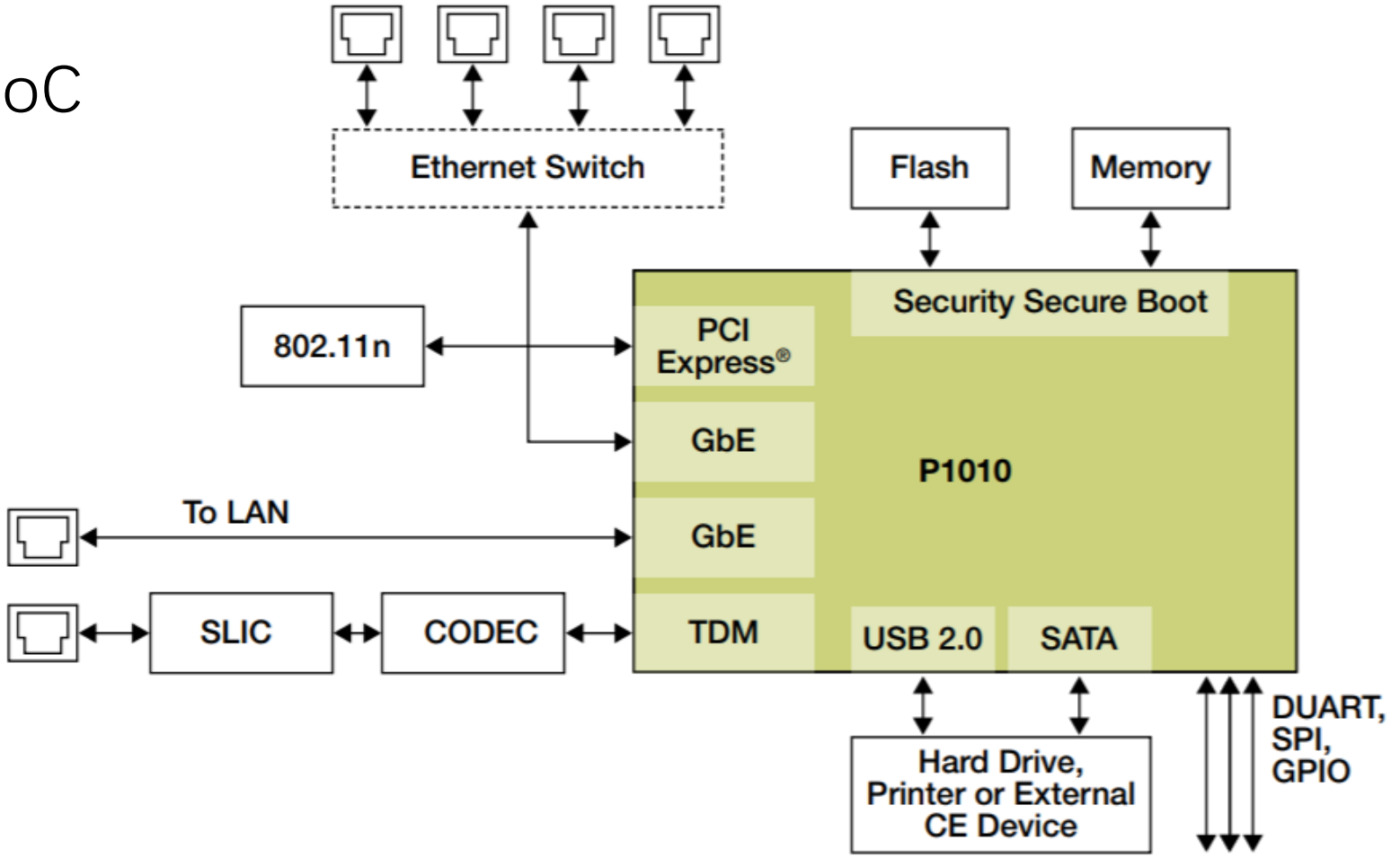


Inside Routers



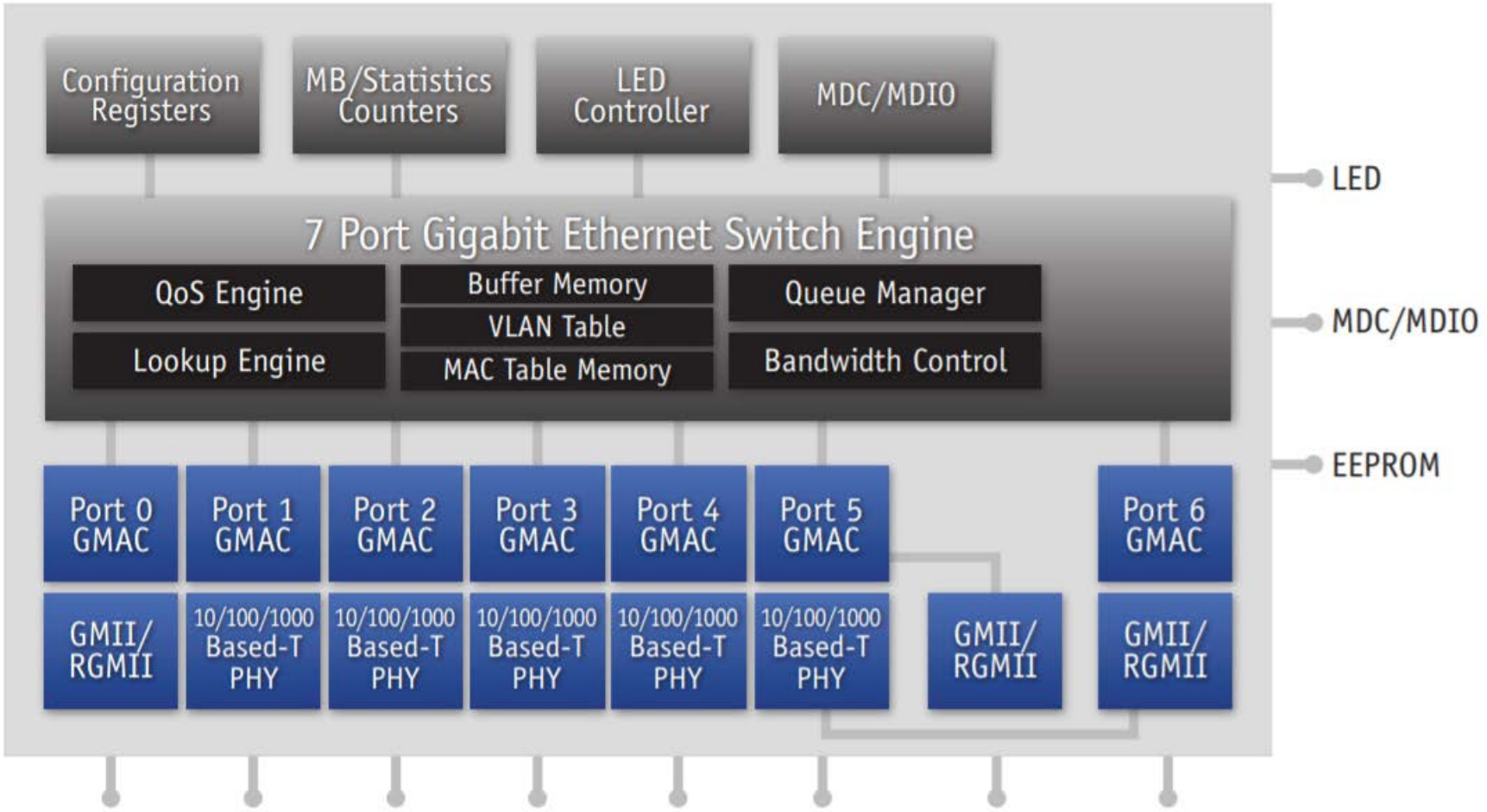
Inside Routers

- Freescale P1010 SoC



Inside Routers

- AR8327 Switch

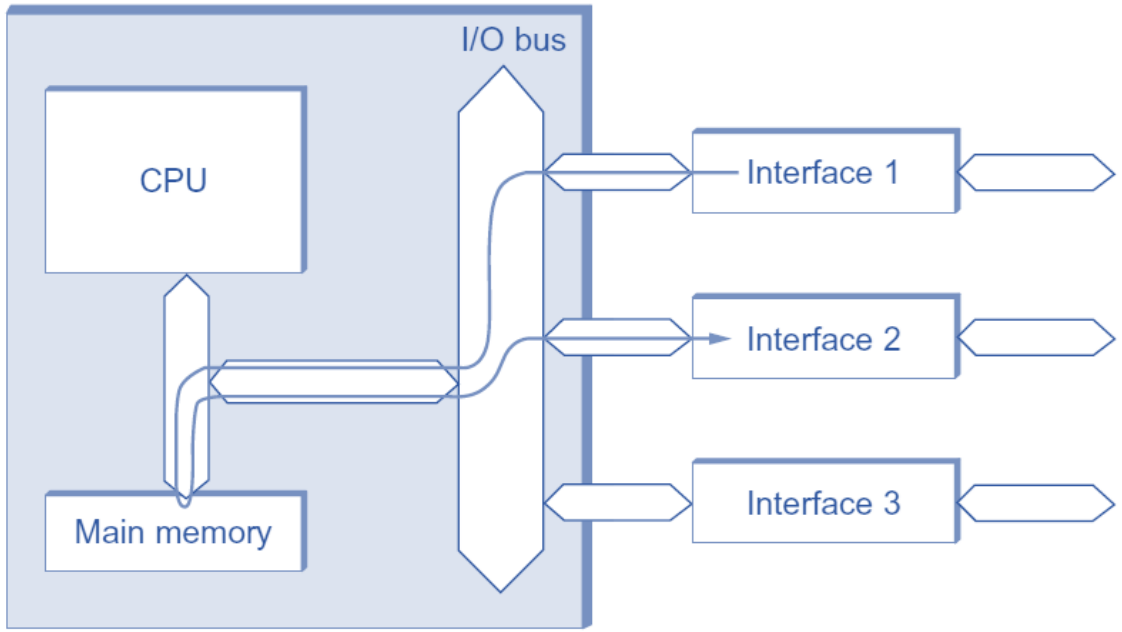


Switching Fabrics

- Function
  - Transfer packet from input buffer to appropriate output buffer
- Switching Throughput
  - Rate at which packets can be transfer from inputs to outputs
  - N inputs: switching throughput N times line rate desirable
- Four Types
  - Shared Bus
  - Shared Memory
  - Crossbar
  - Self-routing

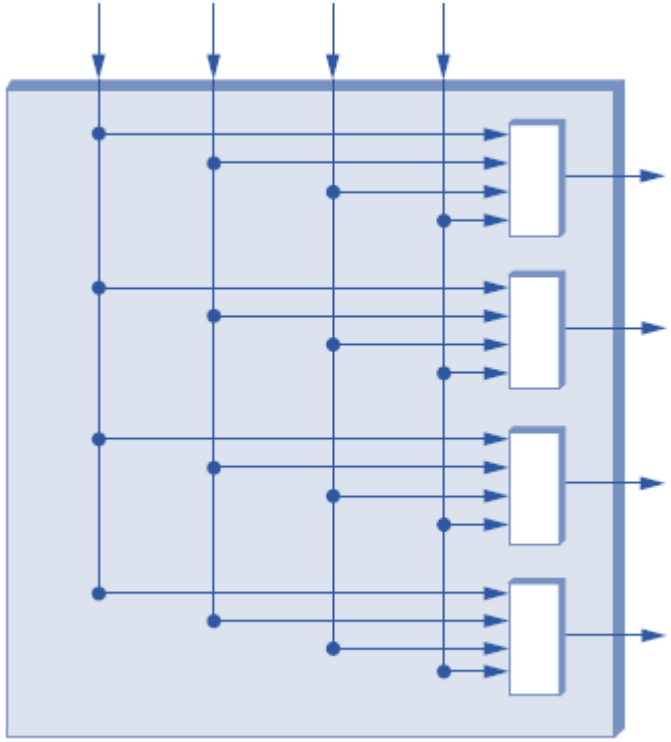
Shared Bus/Memory

- Datagram from input port to output port via a shared bus
  - 2 bus crossings per datagram
  - Bus and memory bandwidth determines switch throughput



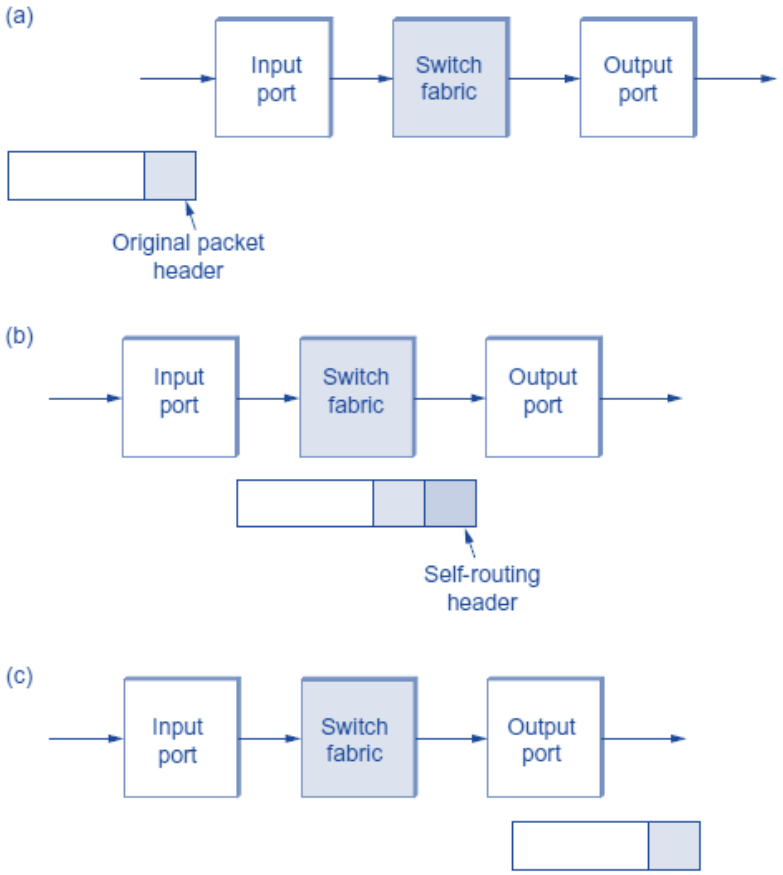
Crossbar

- A crossbar switch is a matrix of pathways that can be configured to connect any input port to any output port



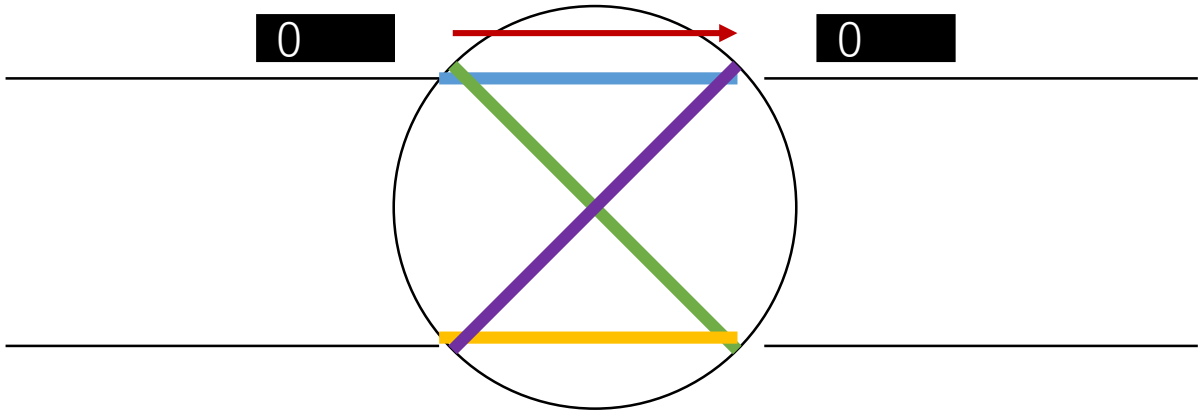
Self-routing

- Routing Header



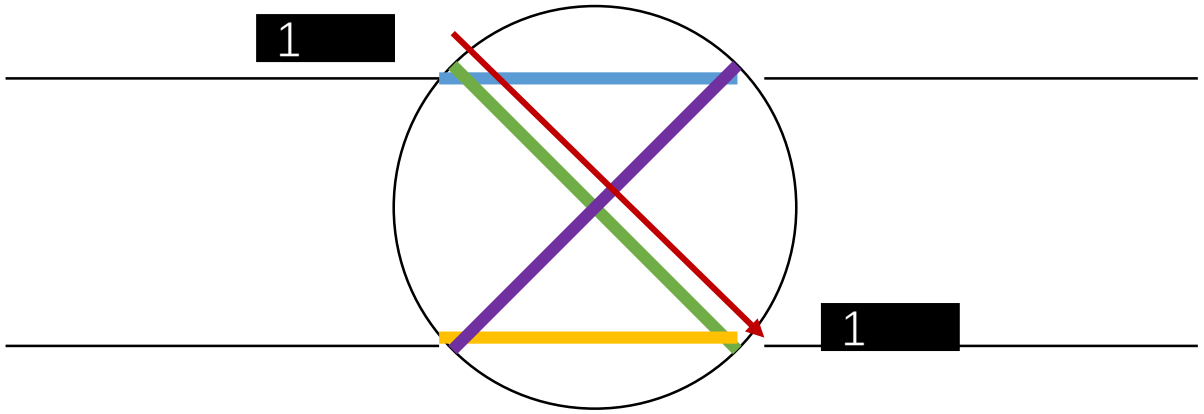
Self-routing

- Switching Element
  - 0=> up
  - 1=> down



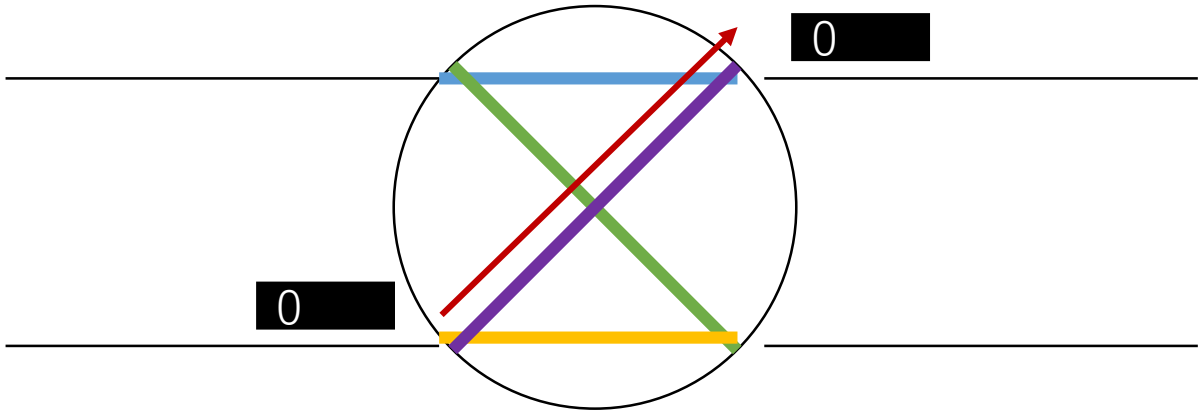
Self-routing

- Switching Element
  - 0=> up
  - 1=> down



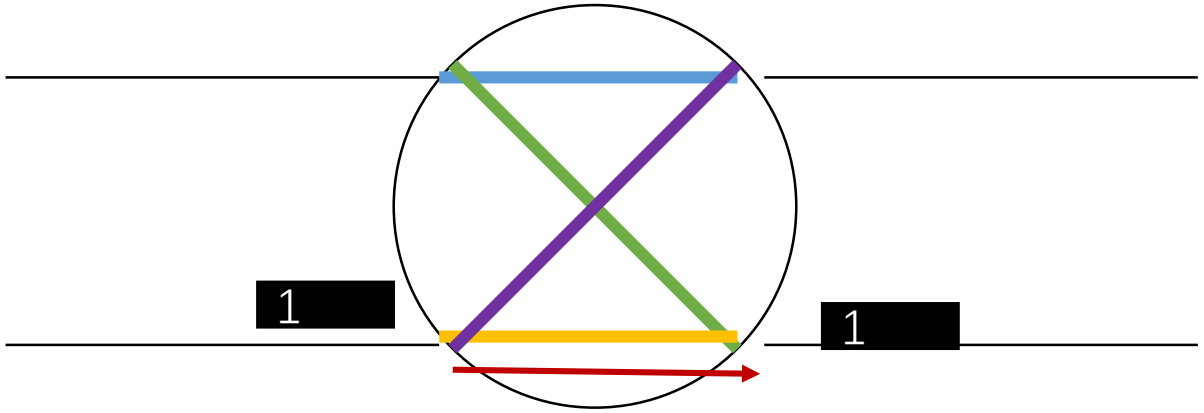
Self-routing

- Switching Element
  - 0=> up
  - 1=> down



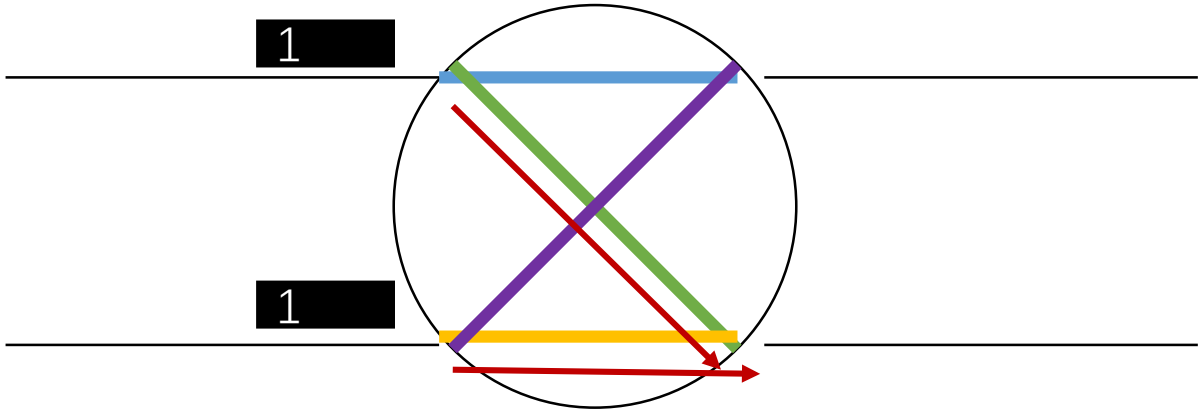
Self-routing

- Switching Element
  - 0=> up
  - 1=> down



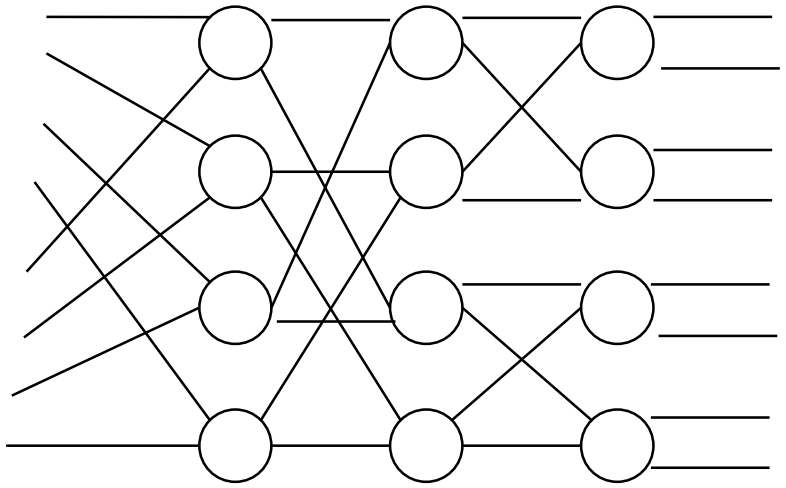
Self-routing

- Switching Element
  - Collision: Two packets with same output ports



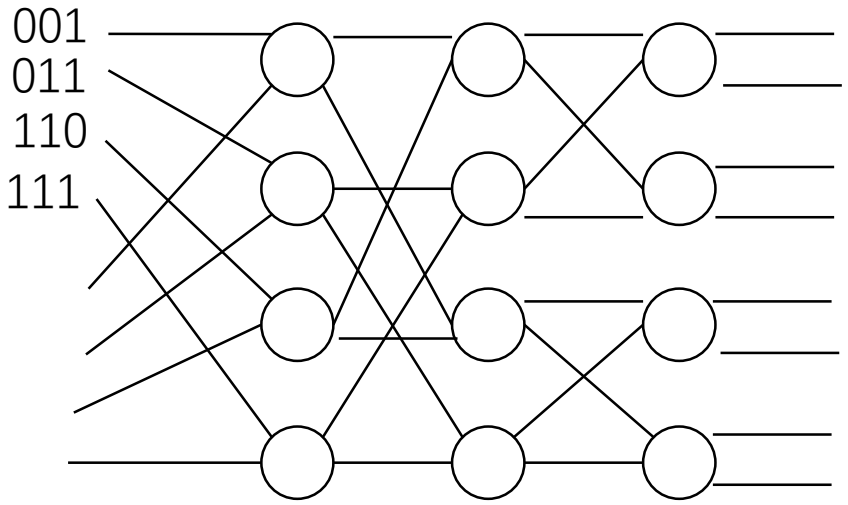
Self-routing

- Banyan Network
  - Collision Free
    - Input Packets are sorted according to routing header



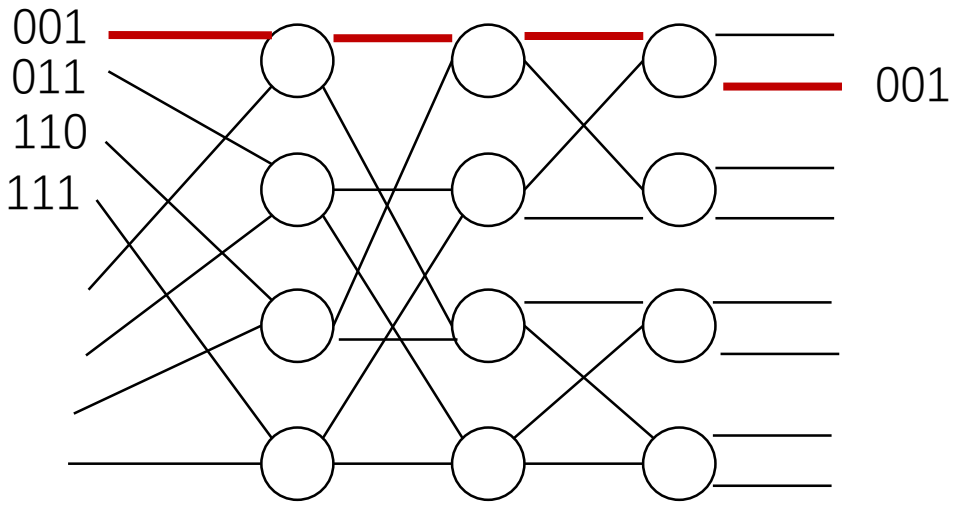
Self-routing

- Banyan Network
  - Collision Free
    - Input Packets are sorted according to routing header



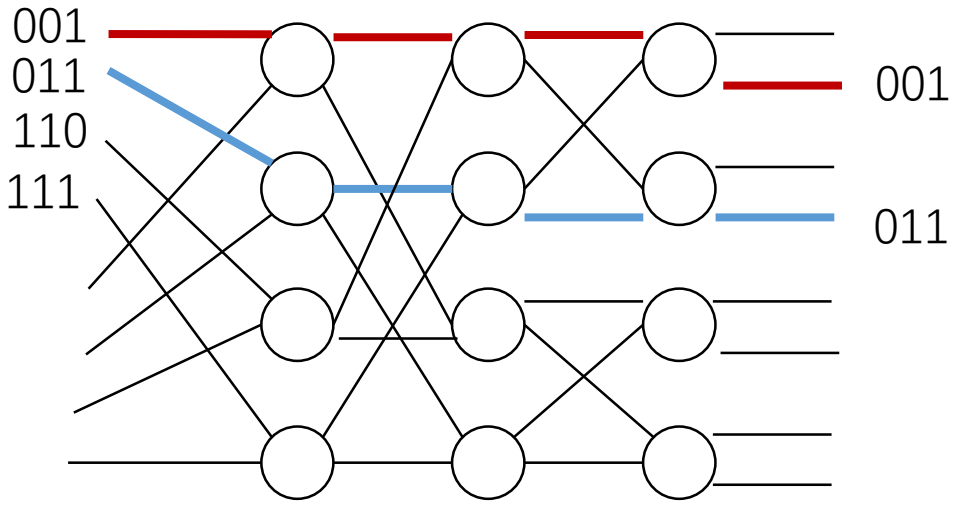
Self-routing

- Banyan Network
  - Collision Free
    - Input Packets are sorted according to routing header



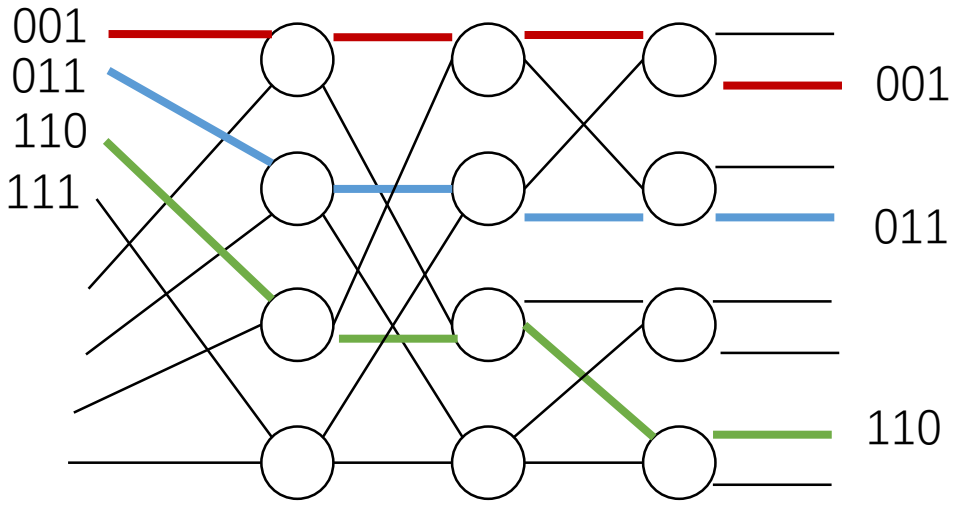
Self-routing

- Banyan Network
  - Collision Free
    - Input Packets are sorted according to routing header



Self-routing

- Banyan Network
  - Collision Free
    - Input Packets are sorted according to routing header



Self-routing

- Banyan Network
  - Collision Free
    - Input Packets are sorted according to routing header

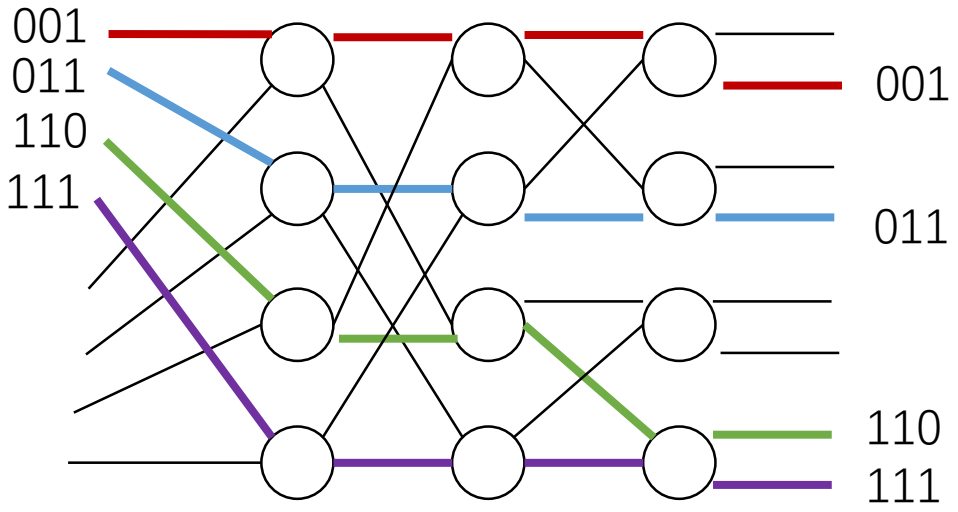
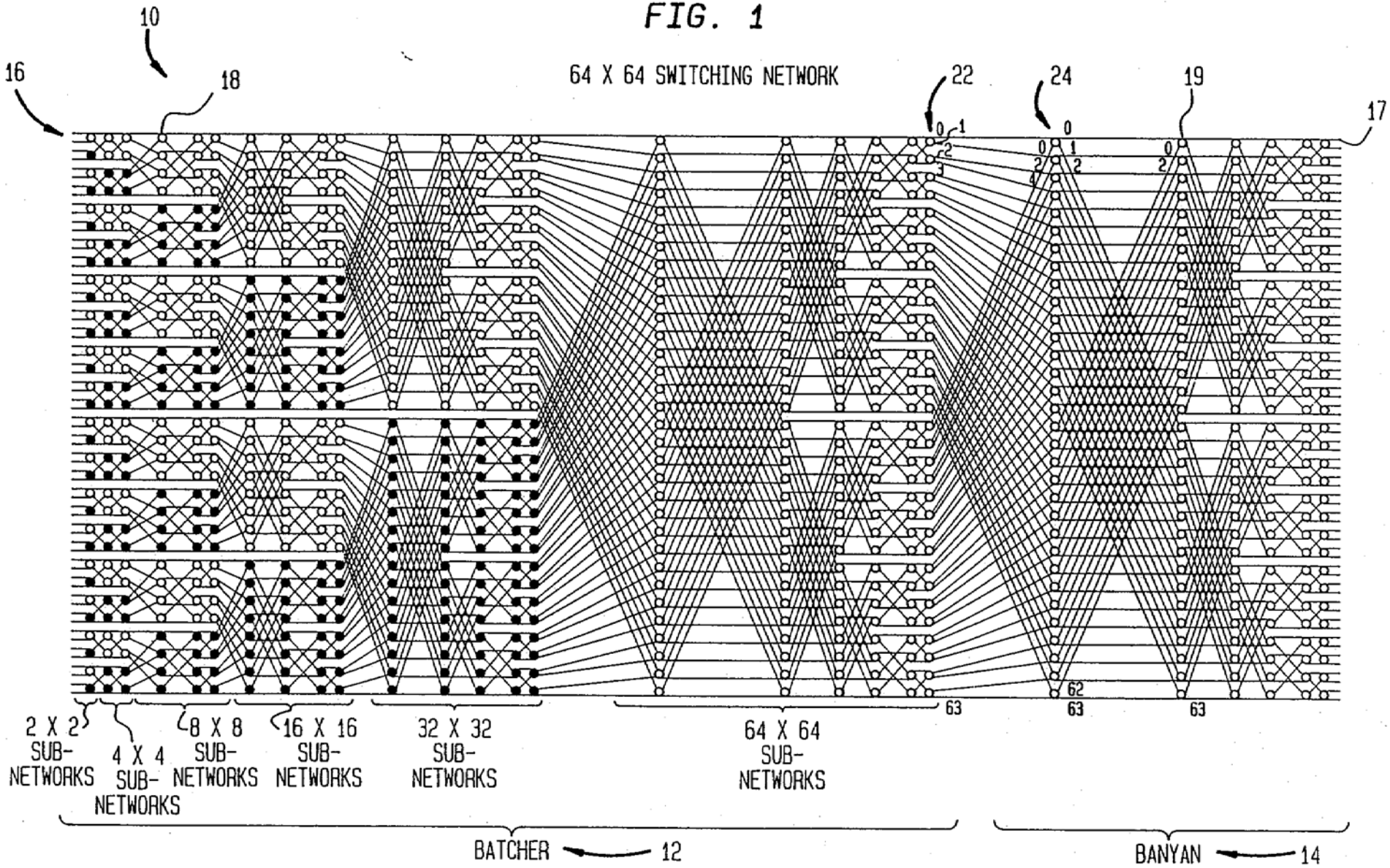


FIG. 1


10
16
18
64 X 64 SWITCHING NETWORK
22
24
19
17
0 1 2 3
0 1 2 4
0 1 2
0 2
62 63
63
63
2 X 2 SUB-NETWORKS
4 X 4 SUB-NETWORKS
8 X 8 SUB-NETWORKS
16 X 16 SUB-NETWORKS
32 X 32 SUB-NETWORKS
64 X 64 SUB-NETWORKS
BATCHER 12
BANYAN 14

Open Sourced Wireless Router

- Linksys WRT54G

OpenWrt
Wireless Freedom

Reference

- Textbook 4.1
- Textbook 4.3
- Textbook 3.4