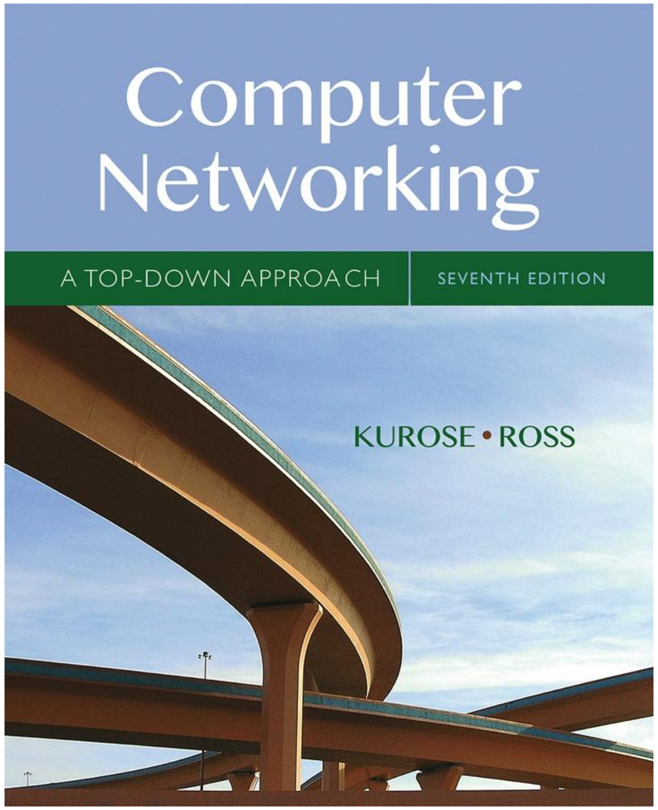
Chapter 4 Network Layer: The Data Plane



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Chapter 4: network layer

- 因特网中所有主机和交换设备都有网络层功能。
- 网络层解决如何实现主机到主机的通信。
- 由于网络管理功能的进步（SDN的研究），进一步将网络层分为数据平面和控制平面。
  - 数据平面，每台路由器的功能，决定达到路由器输入链路的数据报如何转发到该路由器的输出链路之一，转发。
    - IP转发，基于数据报携带的目的地址；
    - 通用转发，使用数据报首部的几个不同域值执行转发。
  - 控制平面，也是每台路由器或集中服务器的功能，控制数据报沿着源主机到目的主机的端到端路径中路由器之间的路由方式。
    - 路由选择算法，OSPF、BGP
    - 软件定义的网络SDN，控制平面置于一台远程控制器

Chapter 4: network layer

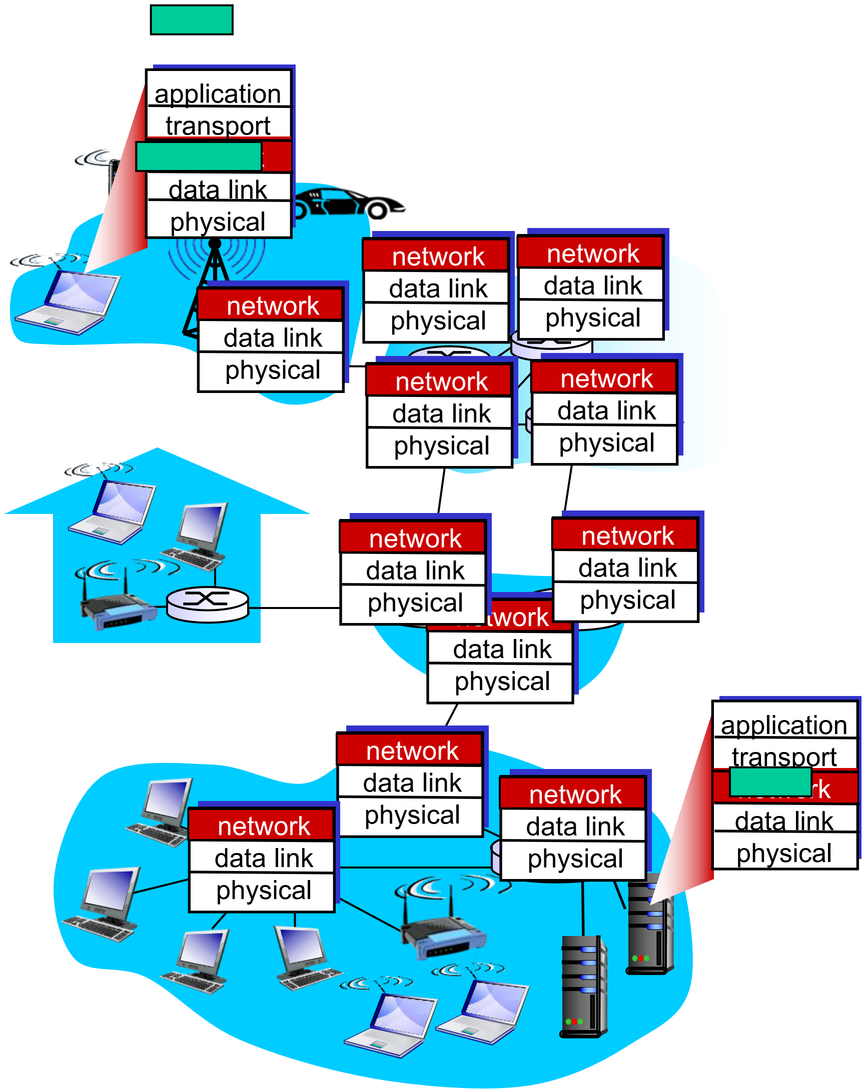
chapter goals:

- understand principles behind network layer services, focusing on data plane:
  - network layer service models
  - forwarding versus routing 转发和路由
  - how a router works
  - generalized forwarding 通用转发
- instantiation, implementation in the Internet

Chapter 4: outline

Network layer 概述

- transport segment from sending to receiving host 从源主机到目的主机传递分组
- on sending side encapsulates segments into datagrams 封装
- on receiving side, delivers segments to transport layer 解封
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it 路由器检查路过IP分组首部



Two key network-layer functions

network-layer functions:

- *forwarding 转发:* move packets from router’ s input to appropriate router output 移动分组
- *routing 路由:* determine route taken by packets from source to destination 决定路径
  - *r o u t i n g a l g o r i t h m s* 路由选择算

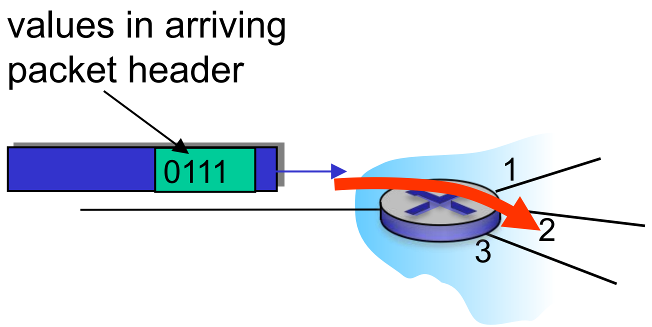
analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

Network layer: data plane, control plane

Data plane

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port
- 转发：将分组从一个输入链路接口转移到适当的输出链路接口的路由器本地动作。
- 索引转发表，通常由硬件完成，很快，几纳秒。

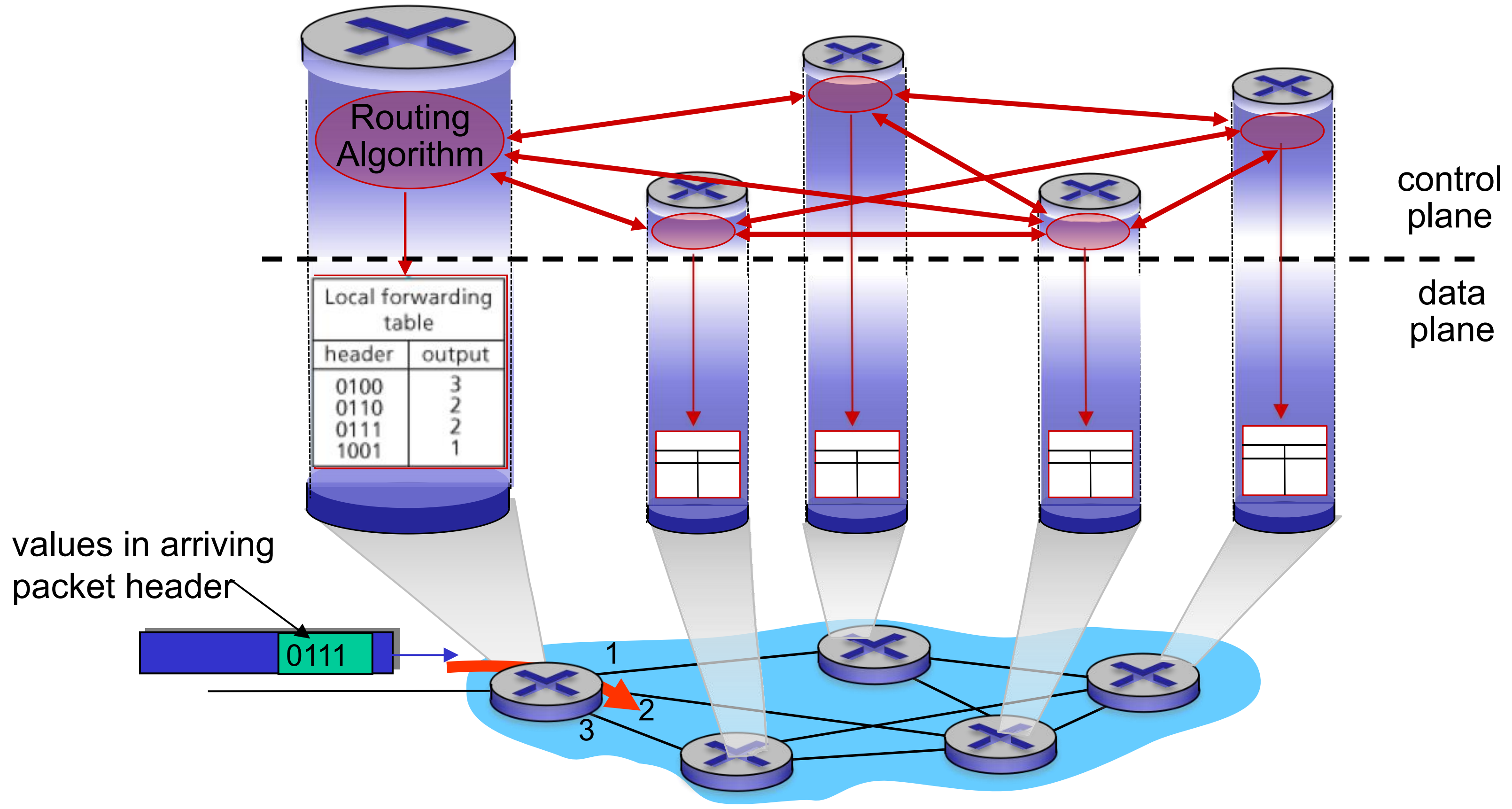


Control plane

- network-wide logic
- 路由：确定分组从源到目的地所采取的端到端路径的网络范围处理过程。
- 路由选择算法决定转发表内容。
- 路由的时间稍长，通常为几秒，由软件完成。
- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

Per-router control plane

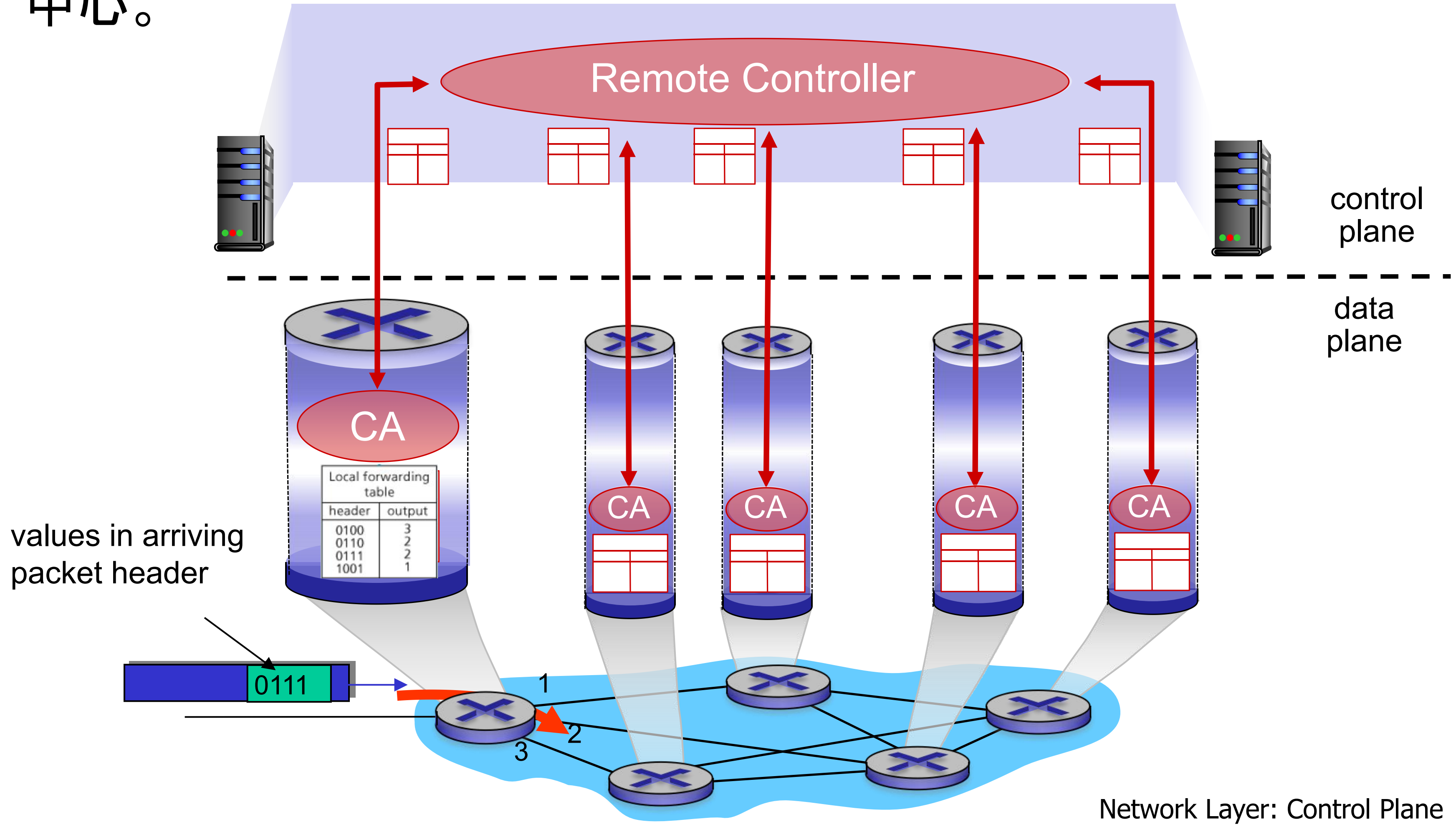
Individual routing algorithm components *in each and every router* interact in the control plane



Local forwarding table	
header	output
0100	3
0110	2
0111	2
1001	1

Logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs), 远程控制器计算并分发转发表，部署在远程数据中心。



Network service model 网络服务模型

Q: What *service model* for “channel” transporting datagrams from sender to receiver?

- 网络服务模型，定义了分组在发送与接收端之间的端到端传输特性。
- 网络层有可能提供的服务（设计希望）：
  - 确保交付，确保分组最终到达目的地。
  - 具有时延上界的确保交付，例如100ms之内。
  - 有序分组交付，确保按序到达目的地。
  - 确保最小带宽，例如确保每流的最小带宽。
  - 安全性，加密解密，提供机密性。
- 但因特网的网络层只提供了单一的服务，称为尽力而为服务。

Network layer service models:

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Chapter 4: outline

Router architecture overview 路由器

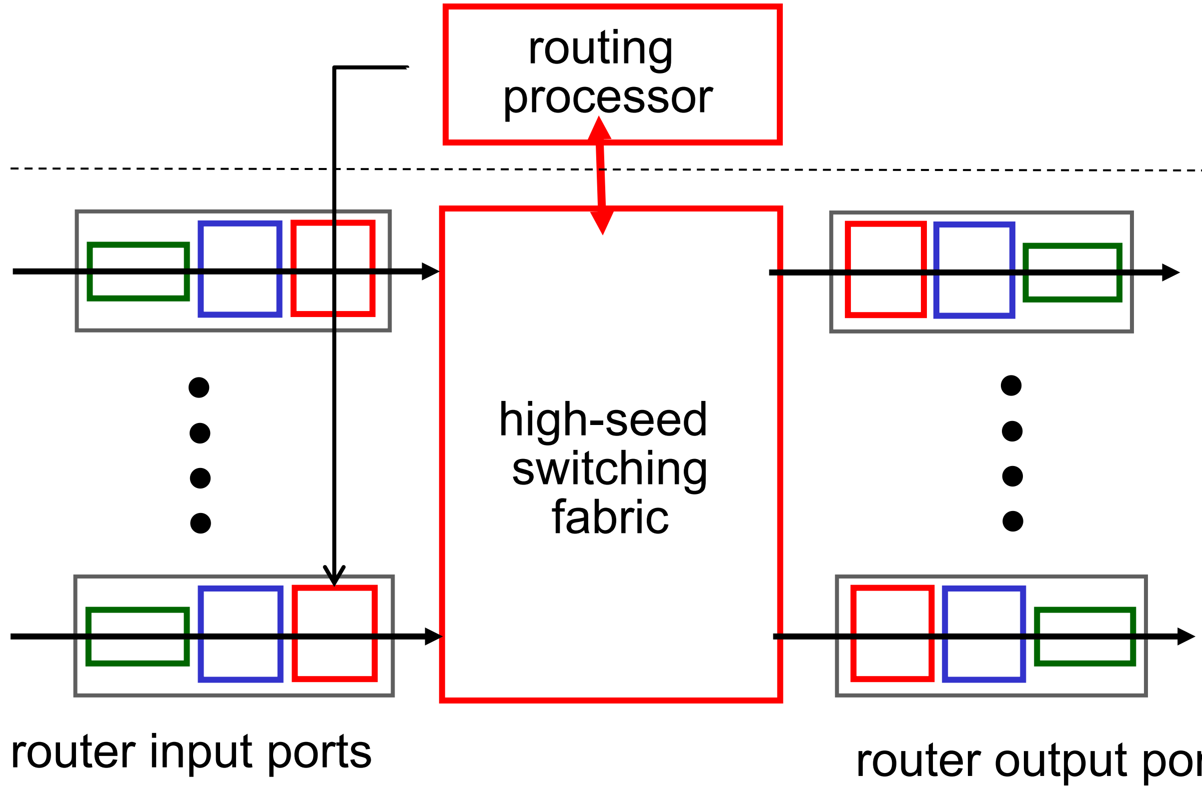
- 接下来探讨路由器的内部软硬件结构，包括输入和输出分组处理、内部交换结构、分组排队和调度。
- 首先约定：

三层交换机

  - 分组交换机：指一台通用分组交换设备，根据分组首部字段中的值，从输入链路接口到输出链路接口转移分组，是第三层设备。
  - 路由器：基于网络层数据报（分组）的首部字段做出转发决定，是第三层设备。
  - 链路层交换机：基于链路层帧的首部字段做出转发决定，是第二层设备。

Router architecture 路由器的体系结构

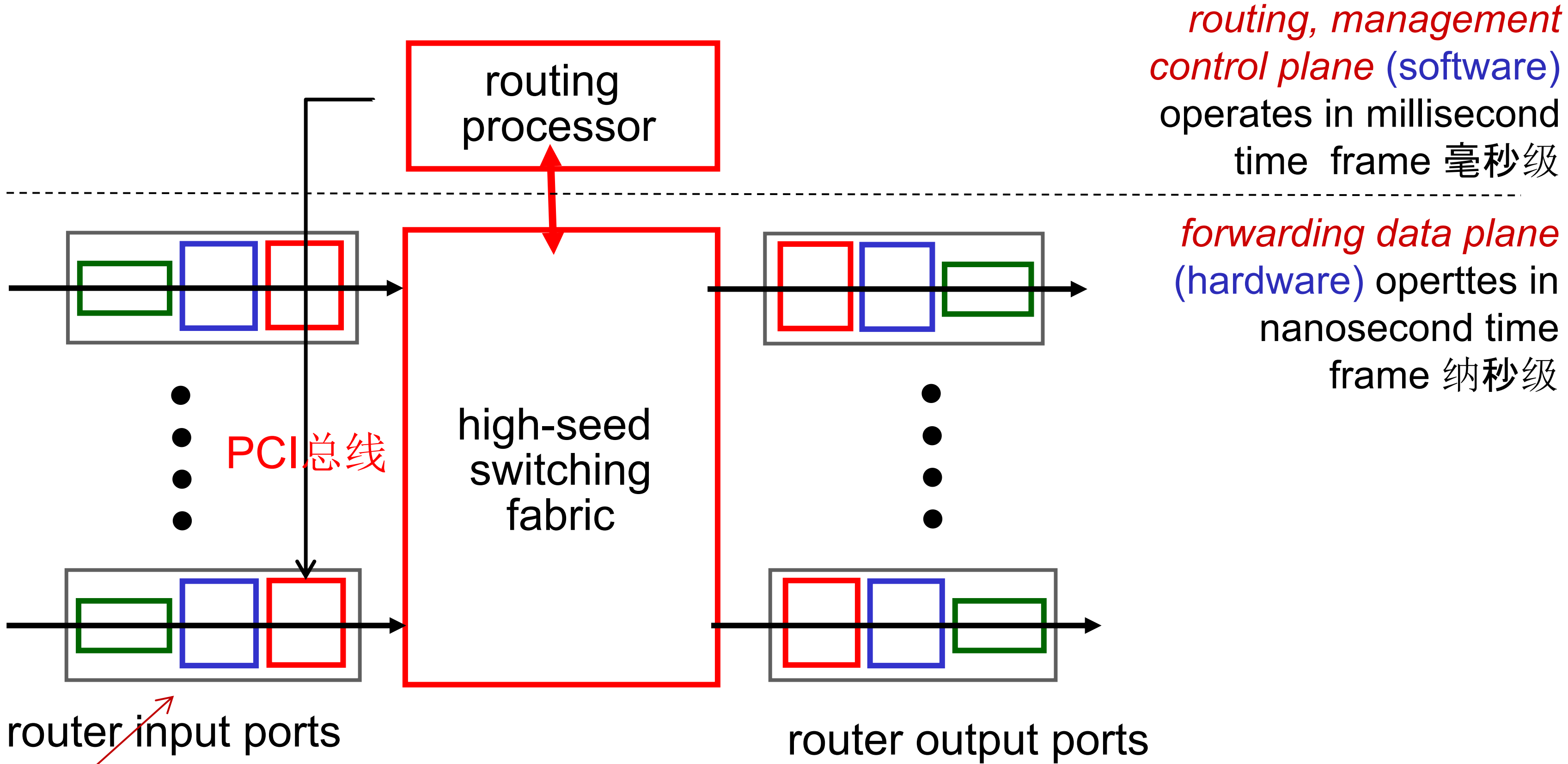
- 通用路由器的体系结构包含四个组件：输入端口、交换结构、输出端口、路由选择处理器。



routing, management
control plane (software)
operates in millisecond time frame 毫秒级

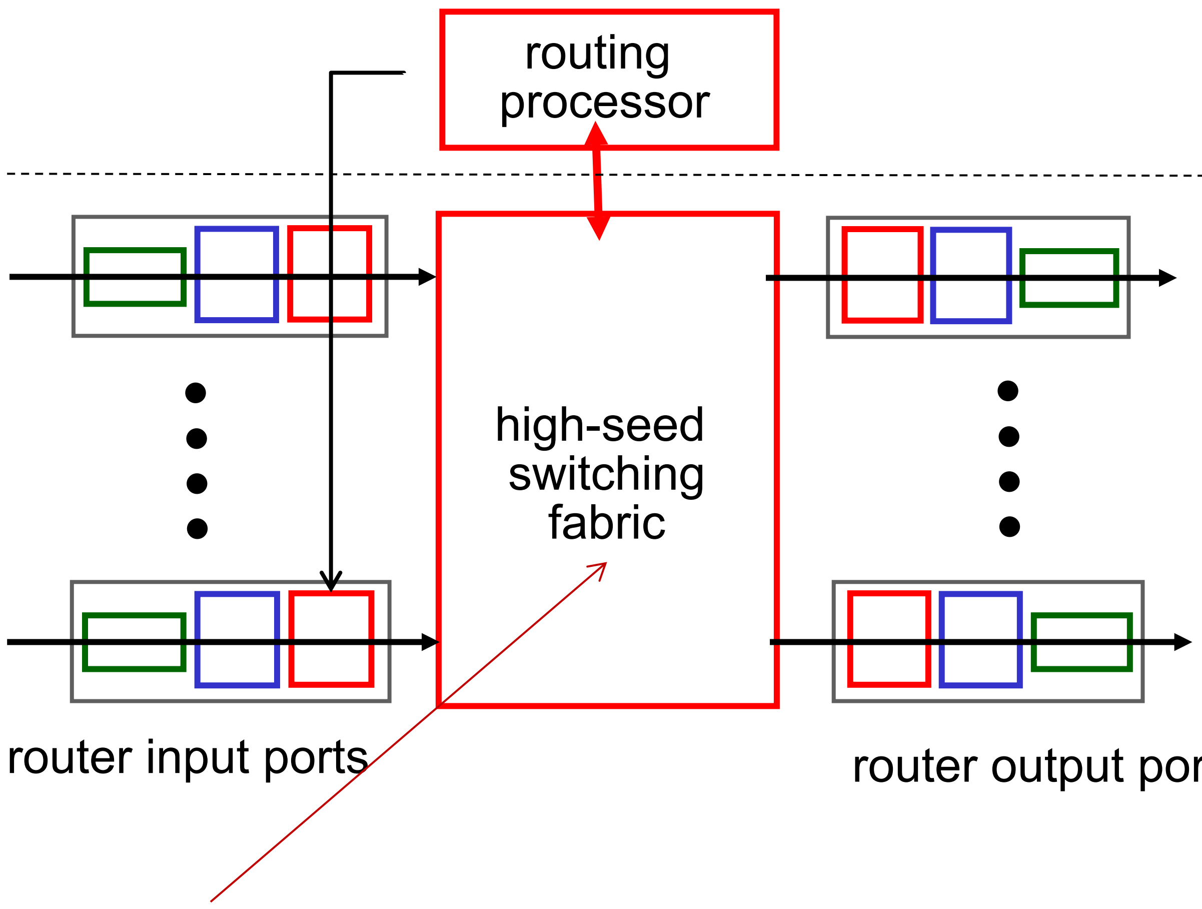
forwarding data plane
(hardware) operttes in nanosecond time frame 纳秒级

Router architecture overview


routing processor
high-seed switching fabric
PCI总线
router input ports
router output ports
routing, management
control plane (software)
operates in millisecond time frame 毫秒级
forwarding data plane
(hardware) operttes in nanosecond time frame 纳秒级

- 输入端口：绿色框完成物理层功能，接收比特流信号；蓝色框完成数据链路层功能，CRC校验、解封帧；红色框完成网络层功能，查找转发表决定转发的路由器输出端口。
- 路由器支持的端口数量范围较大，几个到ISP的几百个10Gbps

Router architecture overview

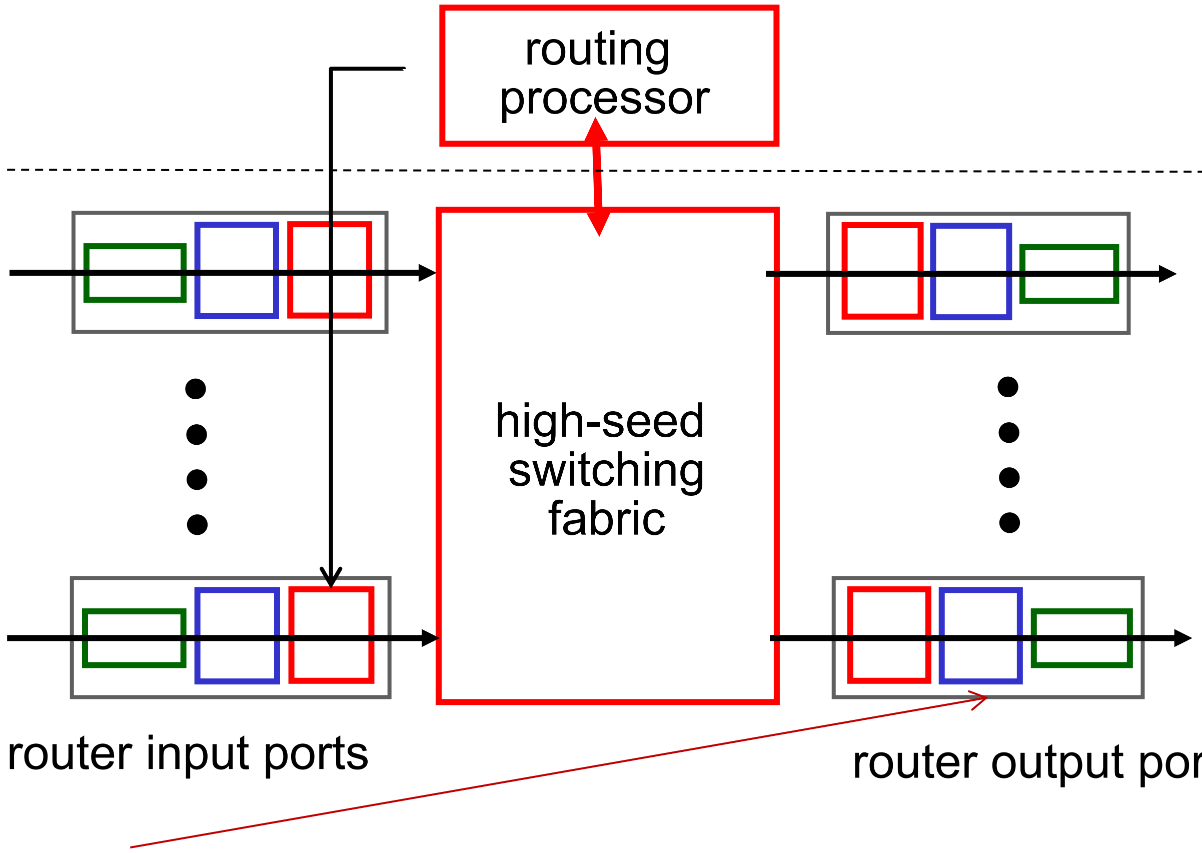


routing, management control plane (software) operates in millisecond time frame 毫秒级

forwarding data plane (hardware) operttes in nanosecond time frame 纳秒级

- 交换结构：将路由器的输入端口连接到输出端口，根据查询转发表的结果将到达的分组通过路由器的交换结构转发到输出端口。

Router architecture overview

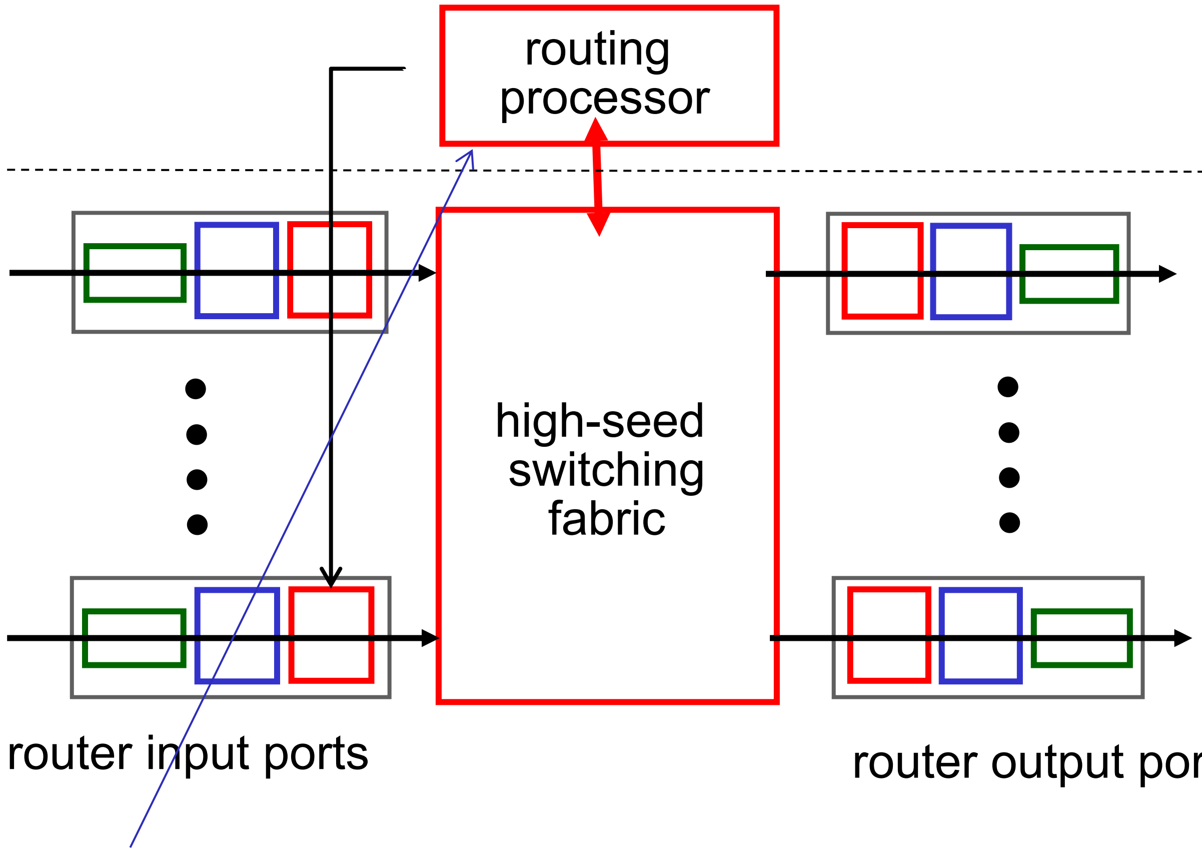


routing, management
control plane (software)
operates in millisecond
time frame 毫秒级

forwarding data plane
(hardware) operttes in
nanosecond time
frame 纳秒级

- 输出端口：从交换结构接收分组，并执行必要的链路层和物理层功能，然后在输出链路上发送出去这些分组。
- 当一条链路是双向传输数据时，输入端口与输出端口成对出现在同一线路卡上。

Router architecture overview

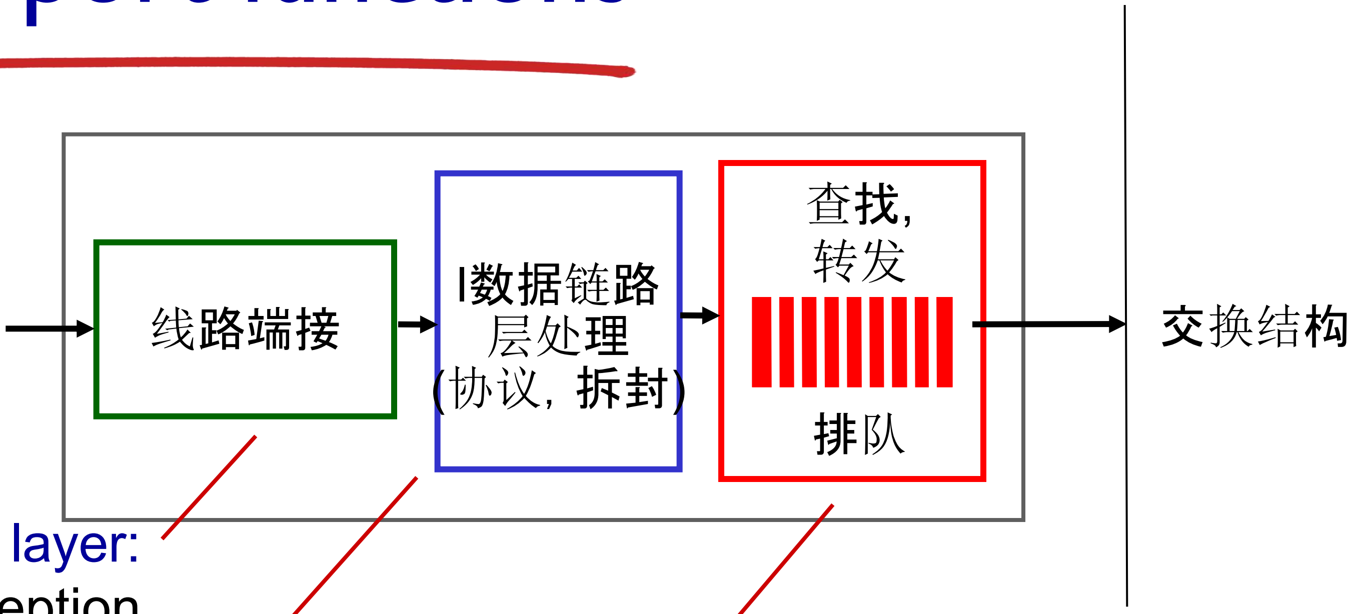


routing, management control plane (software) operates in millisecond time frame 毫秒级

forwarding data plane (hardware) operttes in nanosecond time frame 纳秒级

- 路由选择处理器：维护路由选择表与关联链路状态信息，为该路由器计算转发表（或者在SDN路由器中负责与远程控制器通信），并在路由器是输入端口红框中下发计算结果即转发表
- 路由器的输入/输出端口和交换结构都是由硬件实现的。

Input port functions



physical layer:
bit-level reception

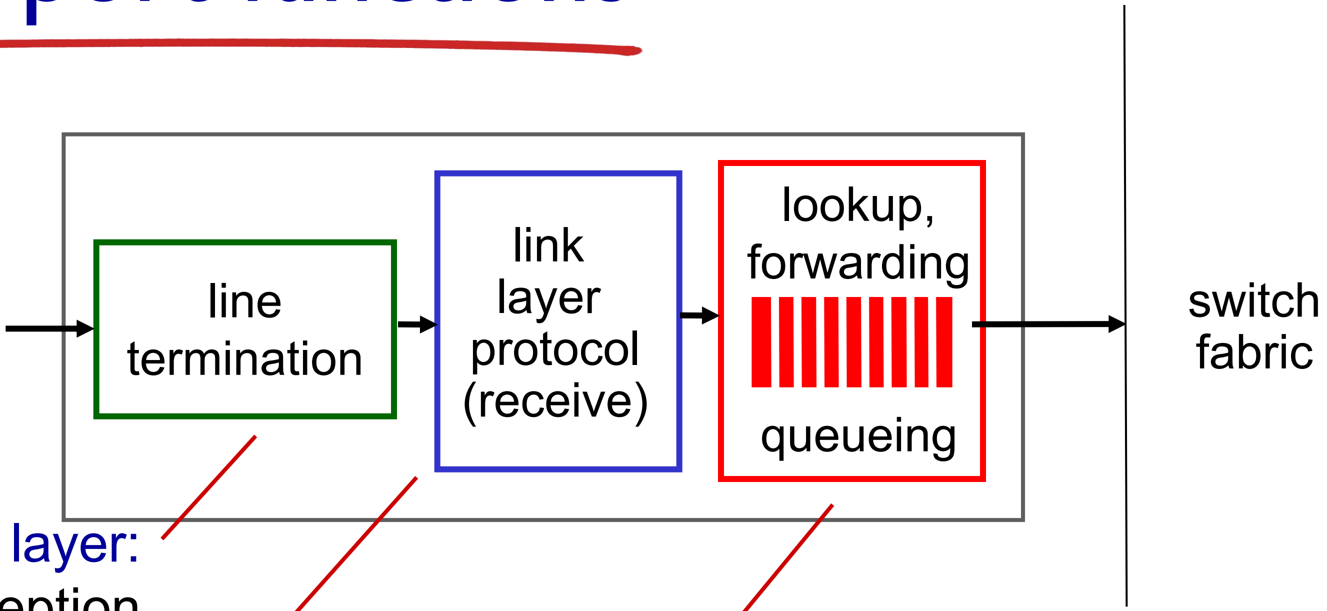
data link layer:
e.g., Ethernet
see chapter 5

- 必须处理物理层和链路层
- 必须检查分组的版本号、检验、寿命字段
- 必须更新用于网络管理的计数器

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory (*"match plus action"*)
- goal: complete input port processing at 'line speed'
- **queuing:** if datagrams arrive faster than forwarding rate into switch fabric

Input port functions



physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- *destination-based forwarding:* forward based only on destination IP address (traditional)
- *generalized forwarding通用转发:* forward based on any set of header field values

Destination-based forwarding 基于目的地址转发

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Destination-based forwarding 基于目的地址转发

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Longest prefix matching 最长前缀匹配规则

longest prefix matching

当有多条匹配时，路由器使用最长前缀匹配规则，即在转发表中寻找最长的匹配项，并向最长前缀匹配相关联的链路接口转发分组.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00010110 10100001 which interface?

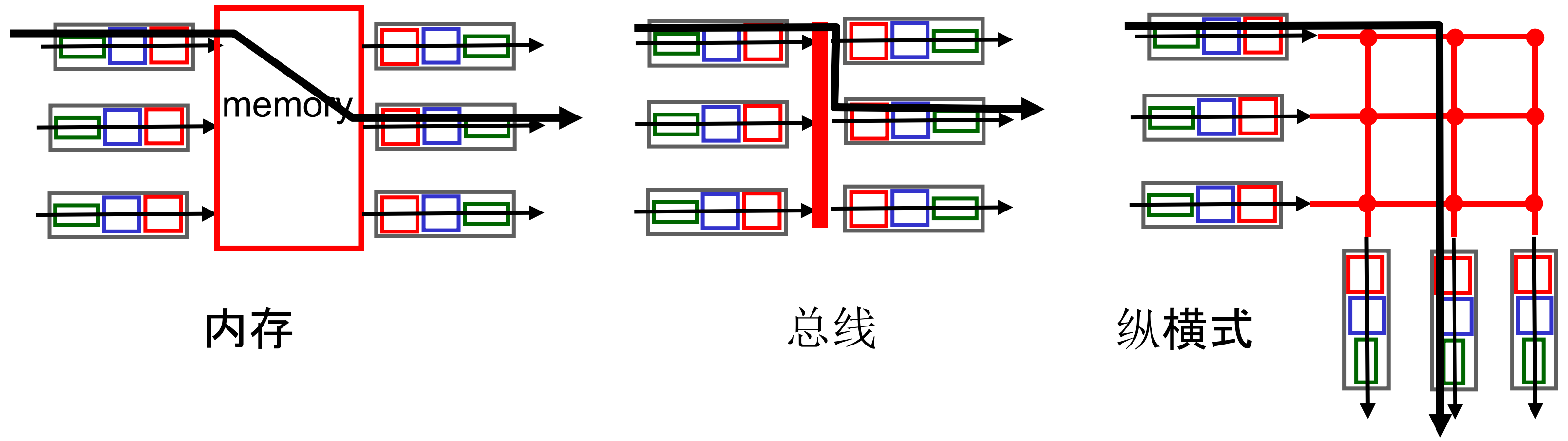
DA: 11001000 00010111 00011000 10101010 which interface?

Longest prefix matching 最长前缀匹配规则

- we’ll see *why* longest prefix matching is used shortly, when we study addressing
- 查找转发表:
  - 用硬件实现，搜索转发表查找最长前缀匹配，在纳秒级执行。
  - 使用超出简单线性搜索的技术，使用快速查找算法。
  - 对内存的访问，使用三态内容可寻址存储器。
- 三态内容可寻址存储器，ternary content addressable memories (TCAMs)。
  - *content addressable:* present address to TCAM，retrieve address in one clock cycle, regardless of table size 将IP地址放入内存，在一个时钟周期返回对该地址的转发表项内容。
  - Cisco Catalyst: can up ~1M routing table entries in TCAM

Switching fabrics 交换结构

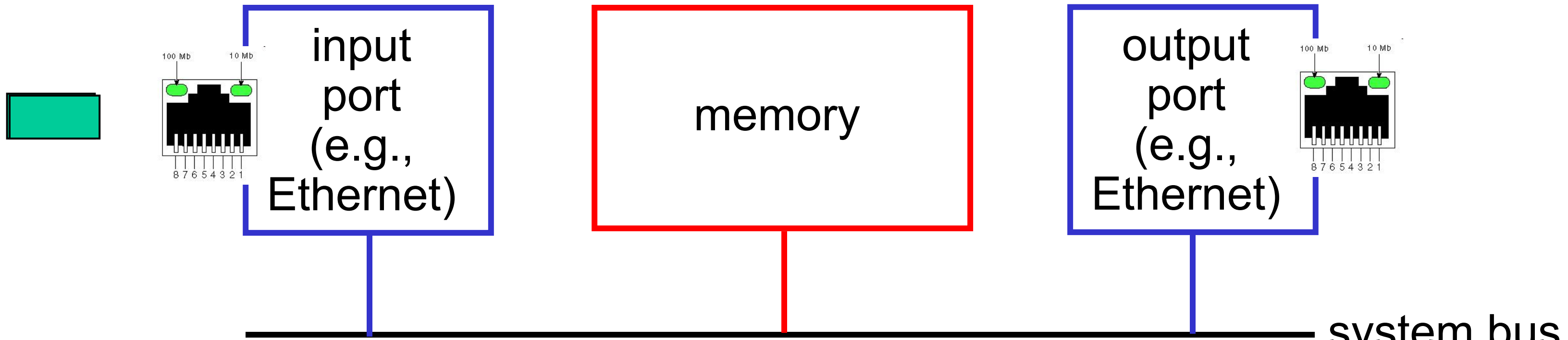
- 将分组从输入端口交换(即转发)到合适的输出端口。
- 交换速率: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output <u>line rate</u>
  - N inputs: switching rate <u>N times line rate</u> desirable
- 三种类型的交换结构:



Switching via memory 经内存交换

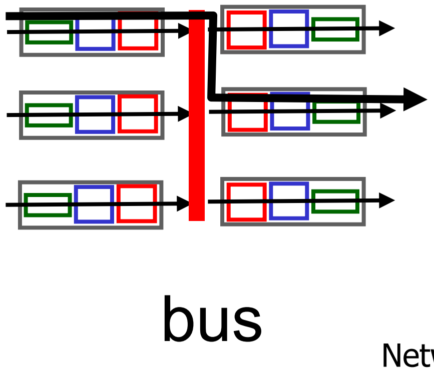
first generation routers 第一代路由器:

- traditional computers with switching under direct control of CPU 最简单、最早的路由器是传统的计算机，交换在CPU直接控制下完成，输入/输出端口像I/O设备一样，分组到达输入端口向通过中断方式向CPU发出信号，然后将分组复制到内存，由CPU提取目的地址，查找转发表决定输出端口，再将分组复制到输出端口缓存中。
- 如果内存带宽为每秒B个分组，则转发带宽为B/2。
- 不能同时转发两个分组，即使目的端口不同，因为共享系统总线一次仅能执行一个内存读写。
- 现代路由器也有用内存交换的，查找转发表和交换由输入线路卡来做。



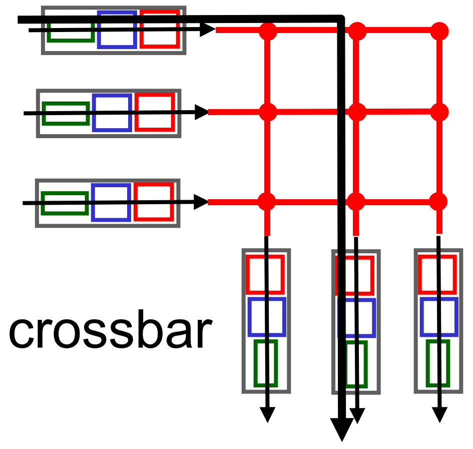
Switching via a bus 经总线交换

- 输入端口经一根共享总线直接将分组传送到输出端口，不需要路由器CPU干预。
- *bus contention:* 输入端口在分组上带标签指示输出端口，然后经总线传送到所有输出端口，只有与标签匹配的端口接收。
- 路由器交换带宽受总线速率的限制。
- 对于小型局域网和企业中的路由器总线交换够用了。
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



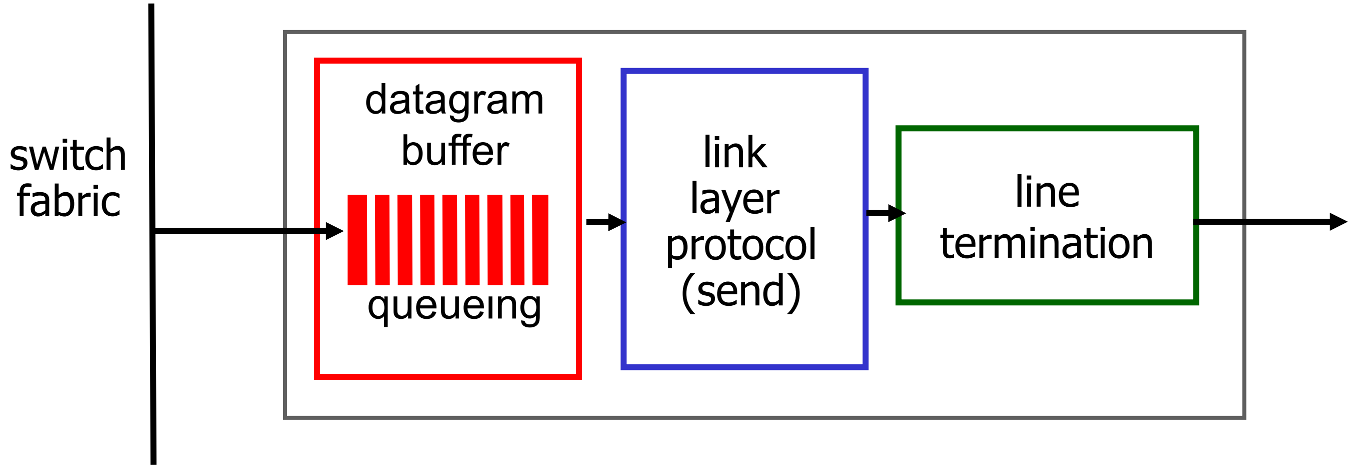
Switching via 经互联网络交换(纵横式交换)

- 克服总线式交换的带宽限制。
- 参考过去多处理器计算机体系结构中互联多个处理器的网络。
- 纵横式交换机:使用一种由2N条总线组成的互联网络，连接N个输入端口和N个输出端口。
- 水平、垂直总线在交叉点处受交换结构控制器控制可根据需要开启、闭合。
- 纵横式交换能够并行转发多个分组，是非阻塞式的。
- 高级设计: 输入端口和输出端口连接到并行的多个交换结构，输入端口将一个分组分成K个小块，通过N个交换结构中的K个同时发送数据小块，再在输出端口处装配还原初始分组.
- Cisco 12000: switches 60 Gbps through the interconnection network



Output ports

This slide in HUGELY important!



- 输出端口取出存放在输出端口缓存中的分组并将其发送到输出链路上。
- 这包括:
- *buffering 缓存* r
  fabric faster thar

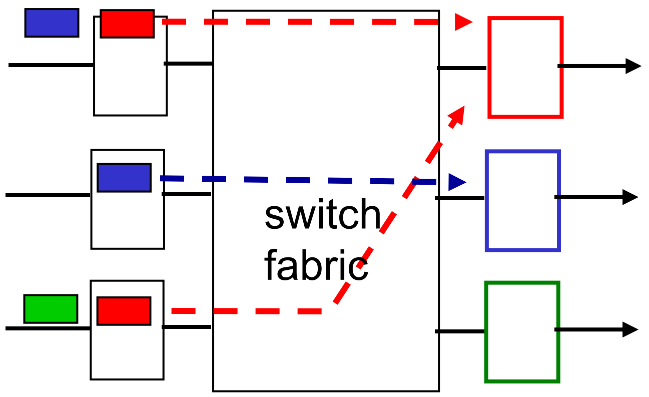
Datagram (packets) can be lost due to congestion, lack of buffers

- *scheduling discipline*
  datagrams for transm

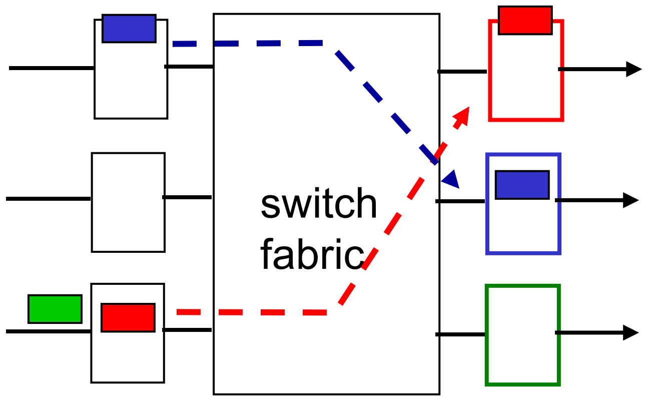
Priority scheduling – who gets best performance, network neutrality

Input port queuing 输入端口排队

- fabric slower than input ports combined -> queueing may occur at input queues
  - *queueing delay and loss due to input buffer overflow!*
- Head-of-the-Line (HOL) blocking 线路前部阻塞: 队列前部的排队分组阻止了队列中的其它分组被转发往不同的输出端口。
- 有研究：只要输入链路分组到达容量的58%，输入队列长度将无限增大。

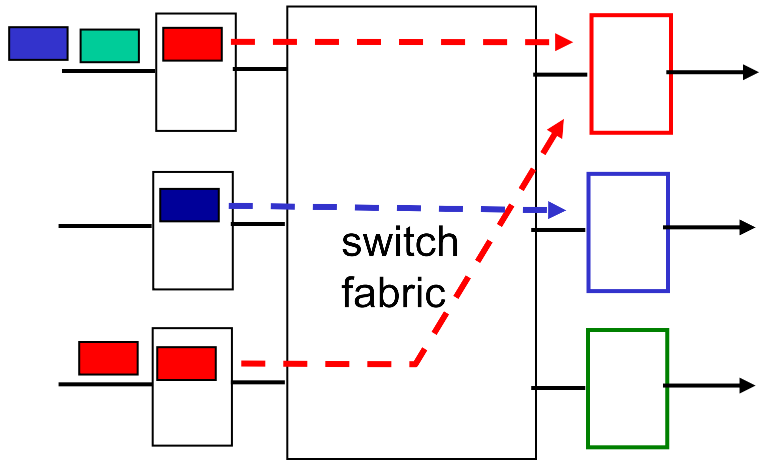


output port contention:
only one red datagram can be transferred.
lower red packet is blocked

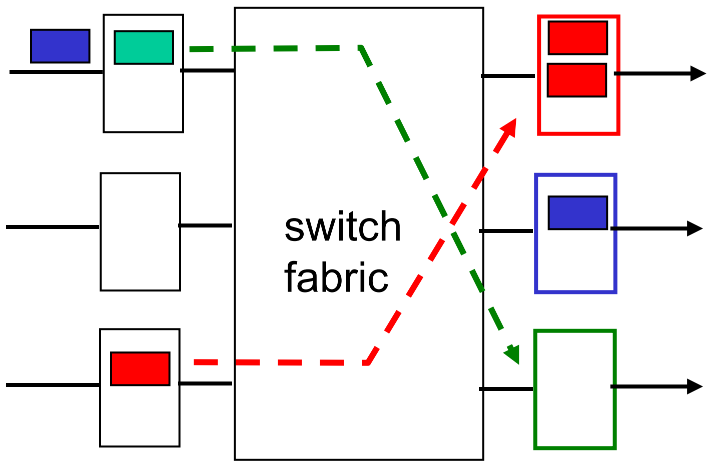


one packet time later:
green packet experiences HOL blocking

Output port queueing 输出端口排队



at *t,* packets more from input to output



one packet time later

- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

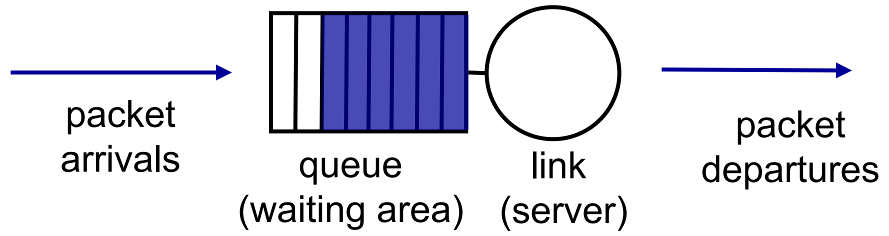
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C
- 缓存容量:链路容量乘以RTT时间。
  - e.g., C = 10 Gpbs link: 2.5 Gbit buffer
- recent recommendation: with *N* flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Scheduling mechanisms 调度策略

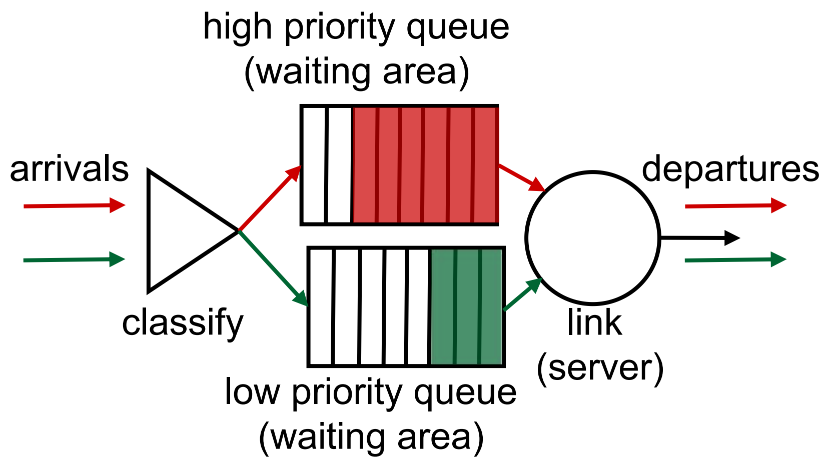
- *scheduling:* choose next packet to send on link
- *FIFO (first in first out) scheduling:* send in order of arrival to queue
  - real-world example?
  - *discard policy:* if packet arrives to full queue: who to discard?
    - *tail drop:* drop arriving packet
    - *priority:* drop/remove on priority basis
    - *random:* drop/remove randomly

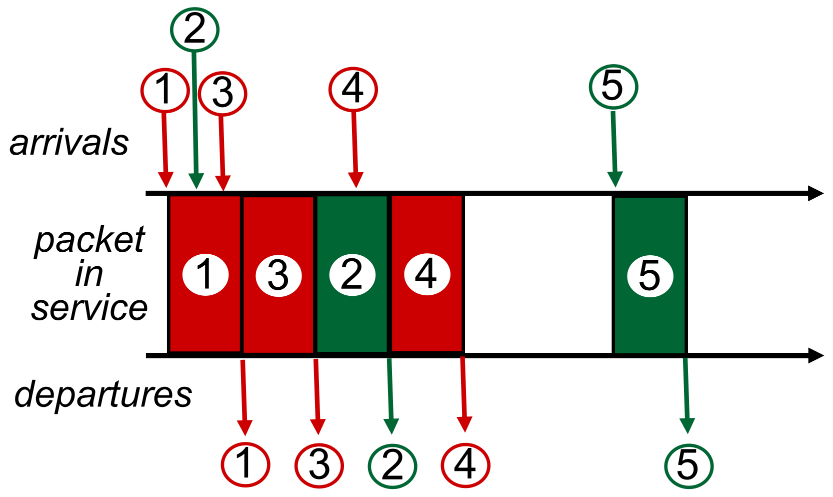


Scheduling policies: priority 优先权排队

priority scheduling: send highest priority queued packet

- multiple *classes*, with different priorities
  - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.
  - real world example?

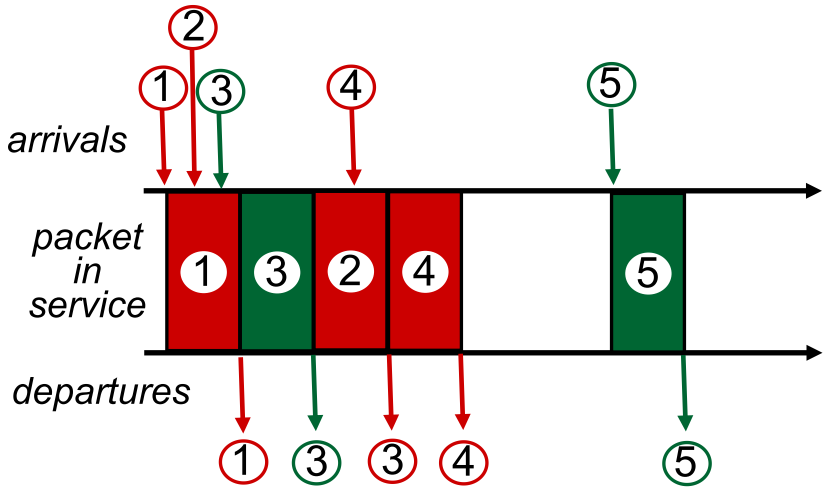




Scheduling policies: still more

Round Robin (RR) scheduling: *循环排队*

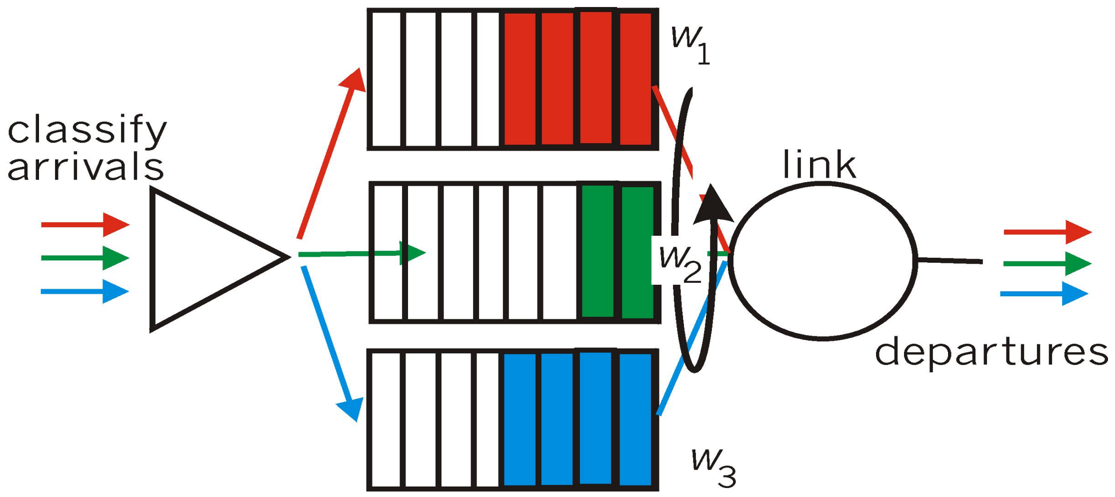
- multiple classes
- cyclically scan class queues, sending one complete packet from each class (if available)
- real world example?



Scheduling policies: still more

Weighted Fair Queuing (WFQ): 加权公平排队

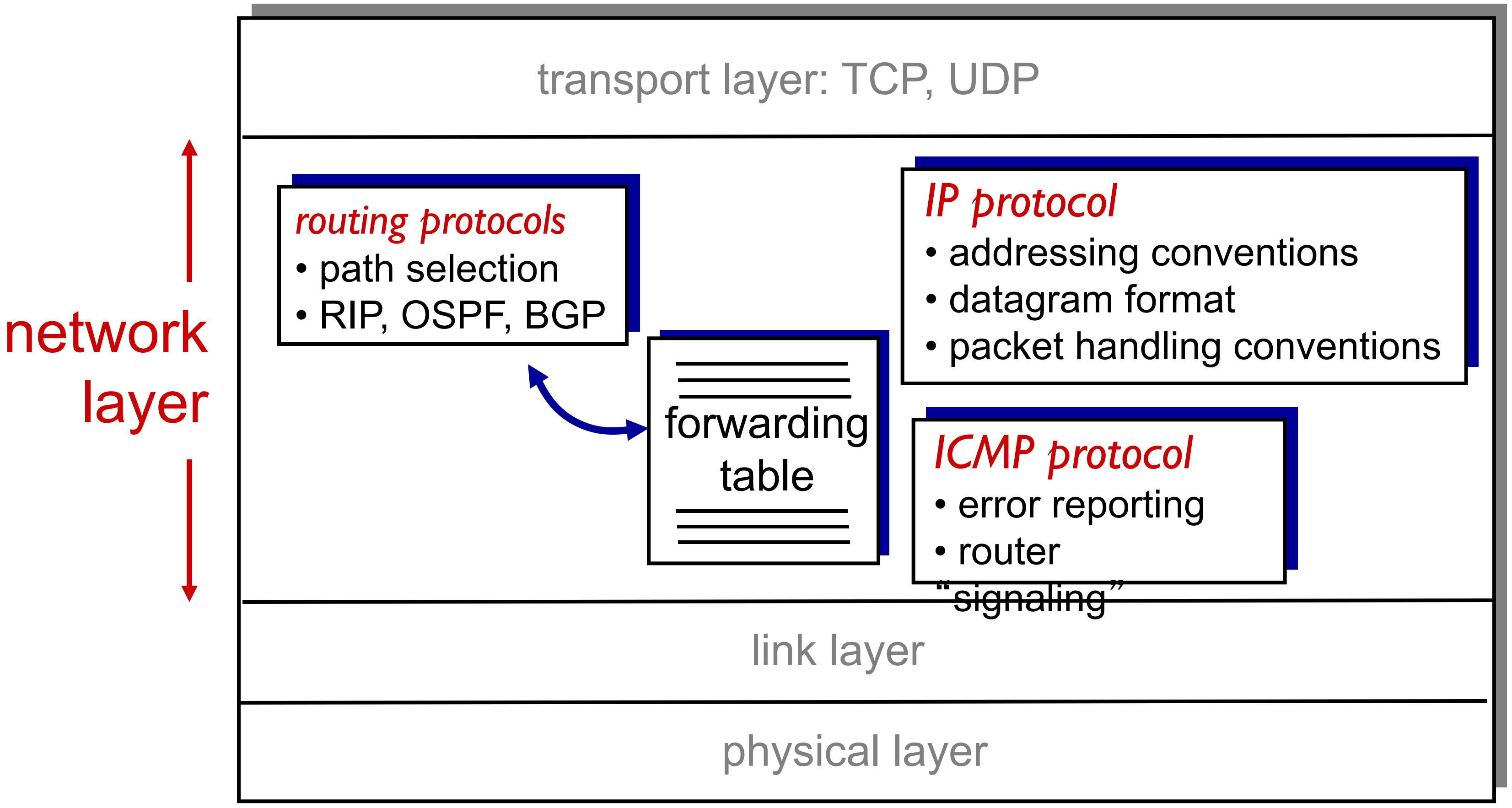
- generalized Round Robin
- each class gets weighted amount of service in each cycle
- real-world example?



Chapter 4: outline

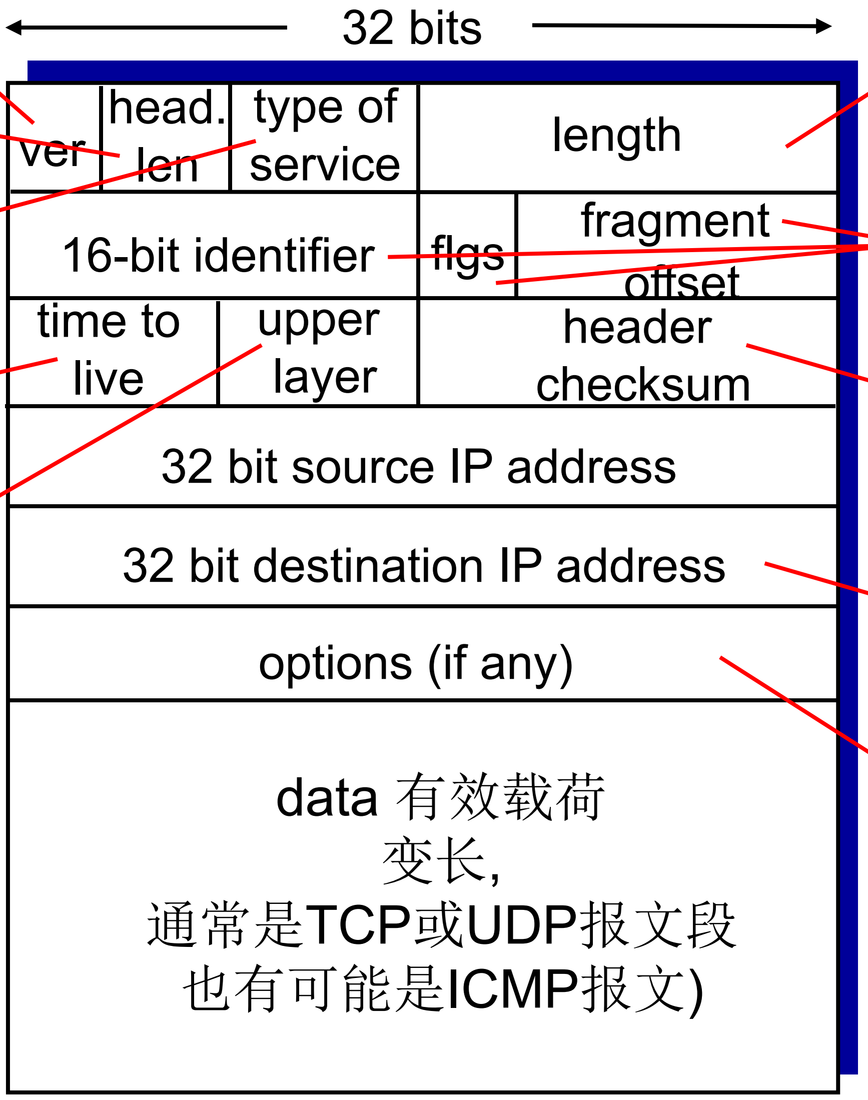
The Internet network layer

host, router network layer functions:



Chapter 4: outline

IP datagram format IP分组格式



IP 协议版本号
IPv4或IPv6

首部长度 (bytes)
固定长度20字节，还有可选字段

服务类型
低时延、高吞吐量、高可靠性、实时、非实时

寿命(最大剩余跳数)
(每跳路由器减一)

递交的上层协议
如6给TCP，17给UDP
像端口号绑定运输层和应用层
上层协议号绑定网络层和传输层

数据报总长度
(bytes) 含首部字段
理论最大65535，不超1500

标识、标志、偏移
与IP分片有关
IPv6取消这个功能

首部校验和
首部每2字节当做一个数，计算校验和
每台路由器重新计算并填回原处，因为TTL会变

源主机通过DNS生成目的IP地址

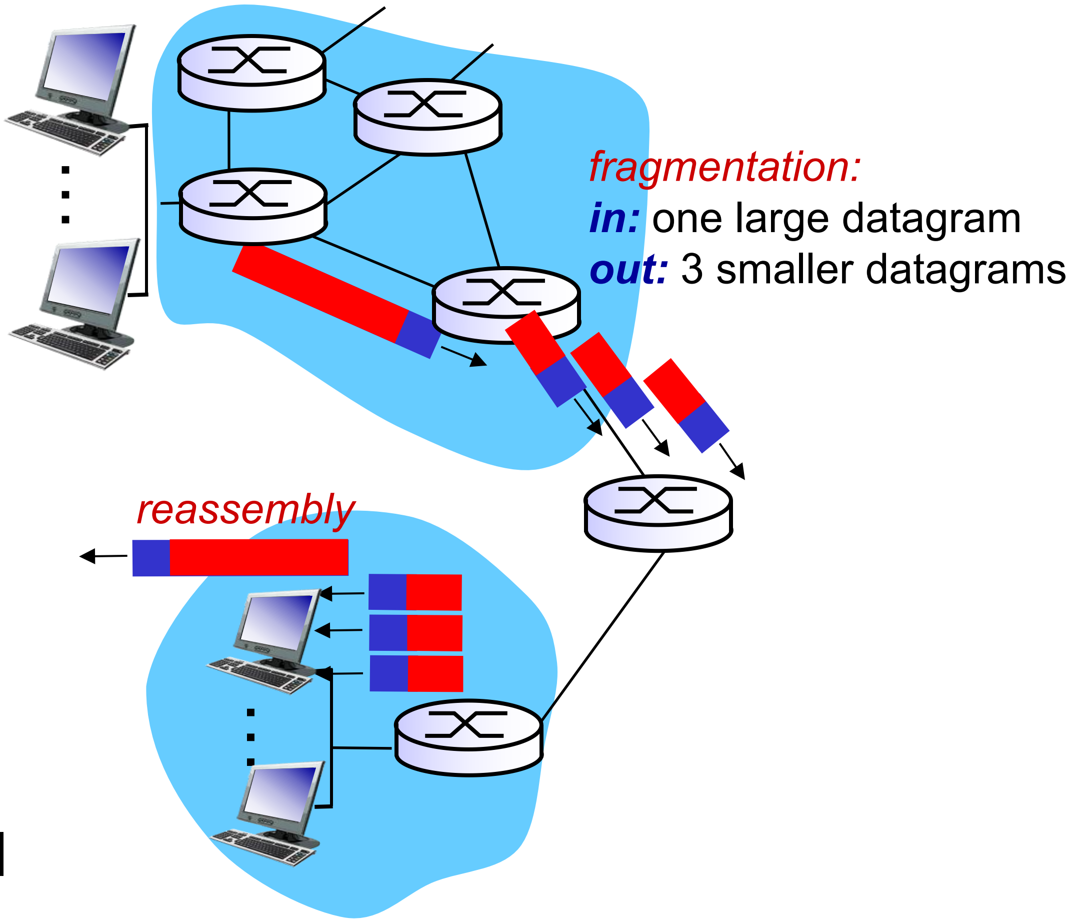
e.g. 时间戳，
路由器产生记录，
指定访问路由列表.
由于选项导致IP数据报长度可变增加了分组处理时间，故IPv6取消了这个字段

how much overhead?
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Chapter 4: outline

IP fragmentation, reassembly IP数据报分片、装配

- 一个链路层的帧能承载的最大数据量叫做**最大传输单元MTU**。
  - 链路层MTU限制IP分组长度.
  - 不同的链路层协议具有不同的 MTUs，例如以太网是1500字节，某些广域网链路是576字节。
- large IP datagram divided (**"fragment 片"**) within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments

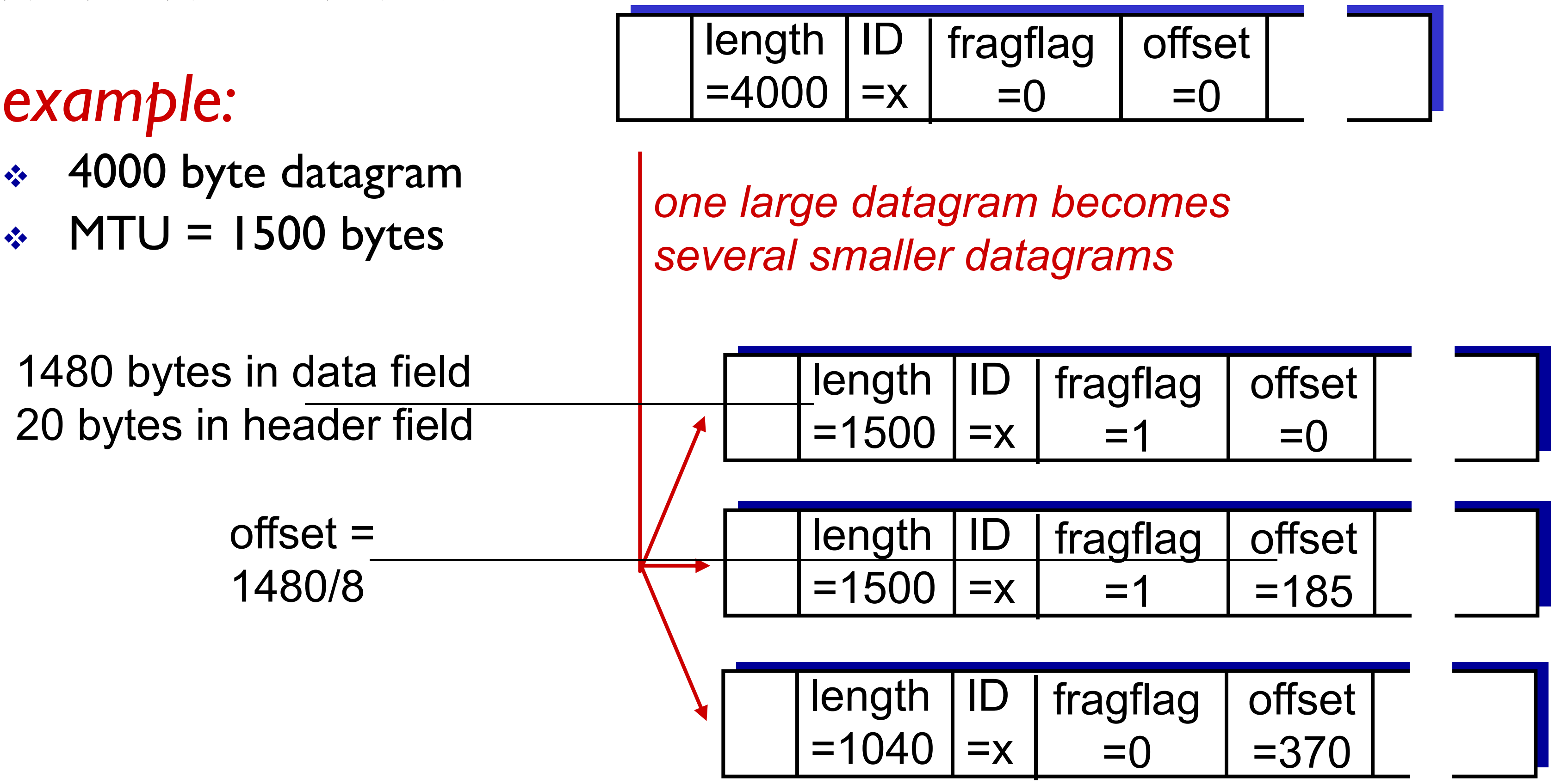


IP fragmentation, reassembly IP数据报分片、装配

- 源主机发送分组时给每个分组添加**标识**，标识号依次+1。
  - 当中间某个路由器需要分片时，按8个字节的倍数分片；同一个分组的每个分片具有相同的 号；**标志位**置为1/最后一个置0；**偏移量**指明分片在分组的起始位置.

example:

- 4000 byte datagram
- MTU = 1500 bytes



Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

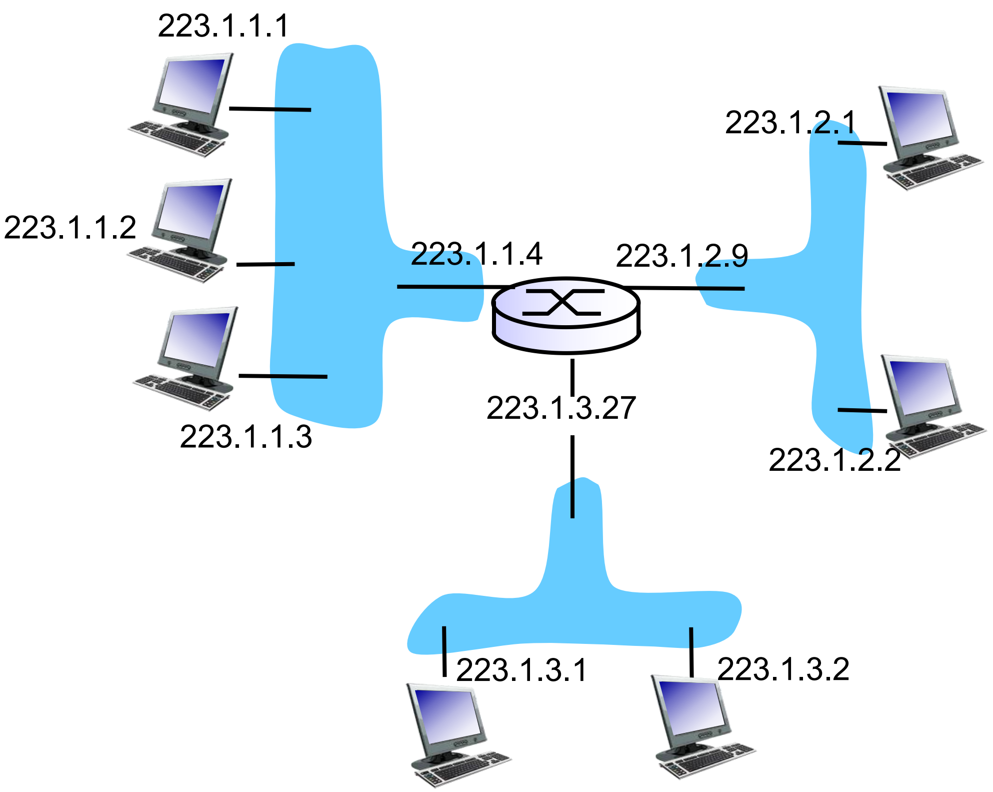
4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

4.4 Generalized Forward and SDN
- match
- action
- OpenFlow examples of match-plus-action in action

IP addressing: introduction IP地址介绍

- *IP address:* 32-bit identifier for host, router interface，共约40亿个，用**点分十进制**法书写。
- *interface* **接口:** connection between host/router and physical link 主机或路由器与物理链路的连接边界。
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)



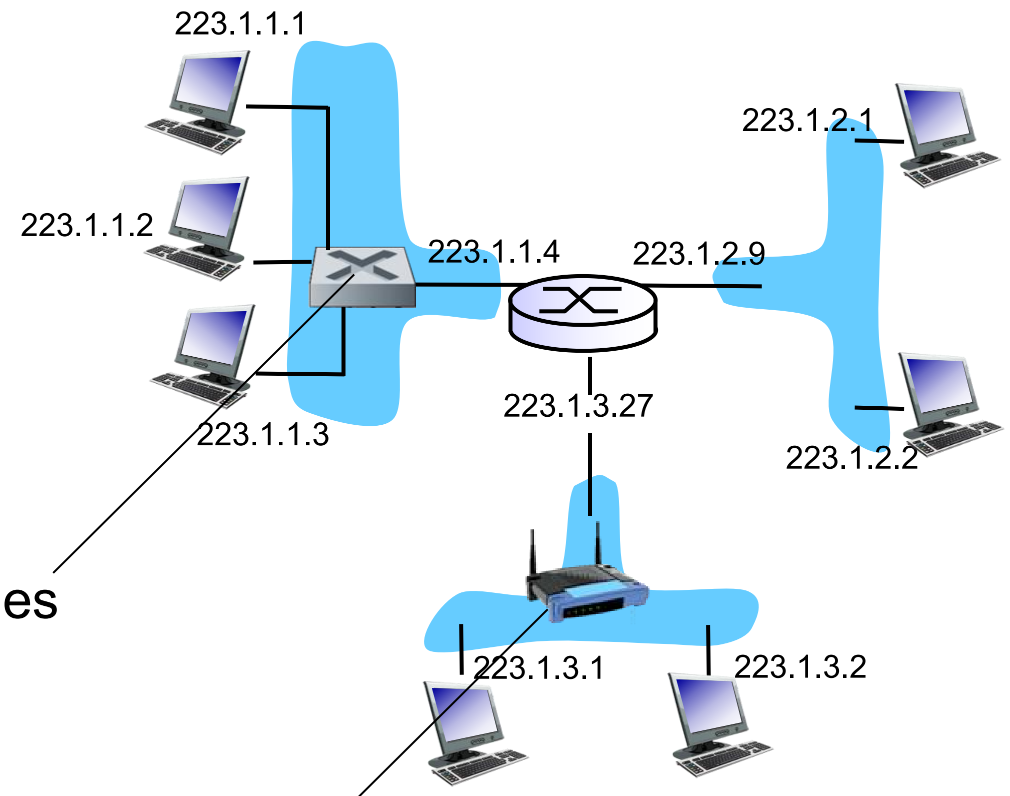
$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000010}_{2}\ \underbrace{00001001}_{9}$$

- IP协议要求每个主机和路由器接口都有自己的IP地址；每个IP地址与一个接口关联。

IP addressing: introduction IP地址介绍

Q: how are interfaces actually connected?

A: we'll learn about that in chapter 5, 6.



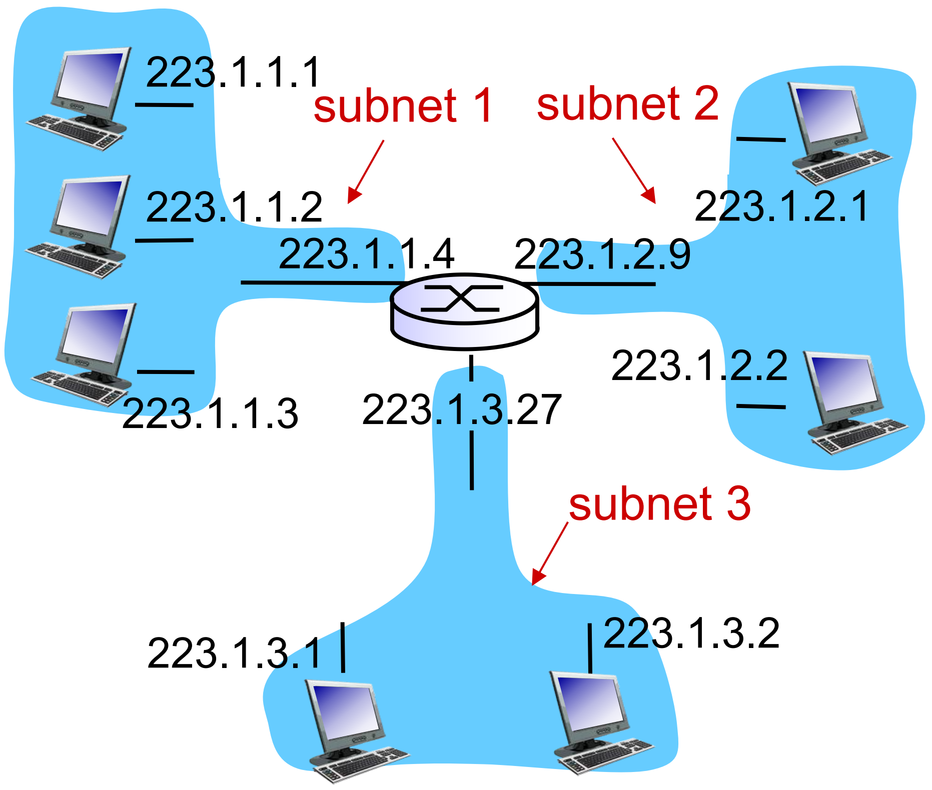
A: wired Ethernet interfaces connected by Ethernet switches

For now: don't need to worry about how one interface is connected to another (with no intervening router)

A: wireless WiFi interfaces connected by WiFi base station

Subnets 子网

- IP address:
  - 子网部分 - high order bits
  - 主机部分 - low order bits
  - 一个接口的IP地址的一部分要由其连接的子网决定.
- *what's a subnet 子网 ?*
- **子网内的设备:**
  - 设备接口具有相同的IP地址子网部分 device interfaces with same subnet part of IP address
  - 物理上可以互相通信而不需通过路由器 can physically reach each other *without intervening router*

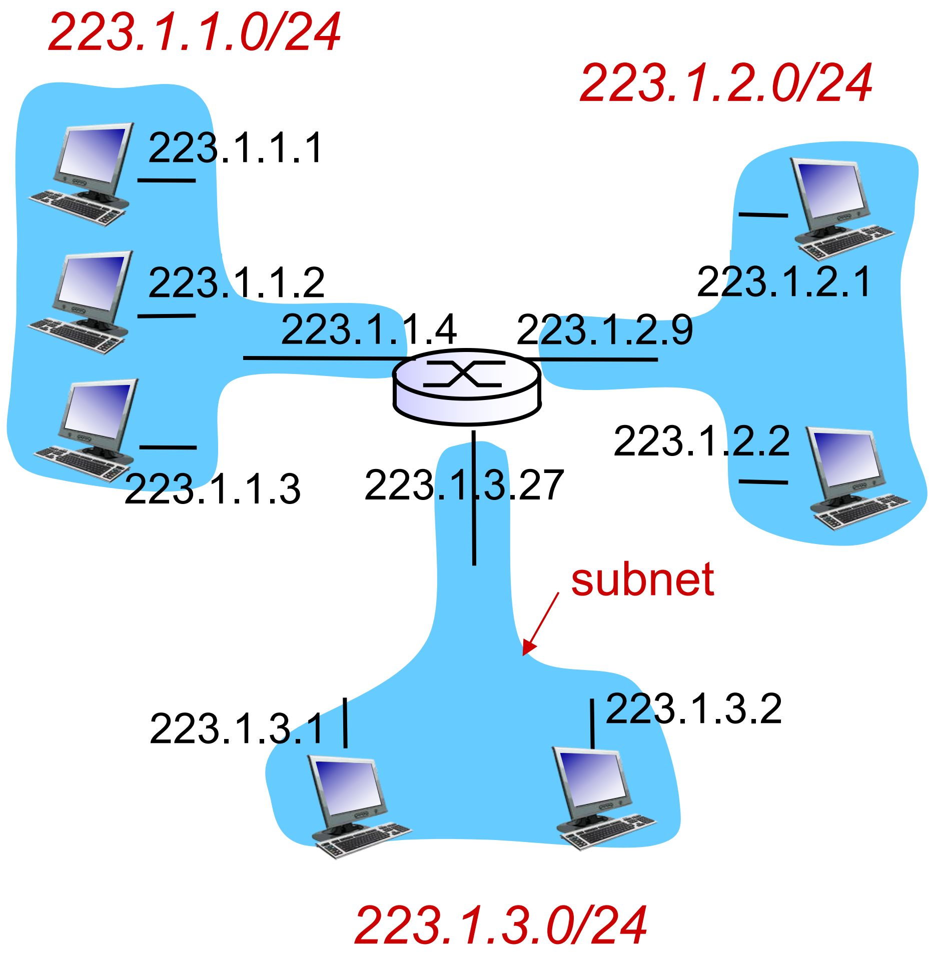


network consisting of 3 subnets

Subnets 子网

recipe

- to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- 每个互相隔离的网络成为一个子网 *subnet*
- 223.1.1.0/24为一个子网号.
- /24表示子网掩码。
- 子网掩码（subnet mask）通常指示了IP地址中哪些位是子网号，可以用/24，也可以表示为：



subnet mask: /24

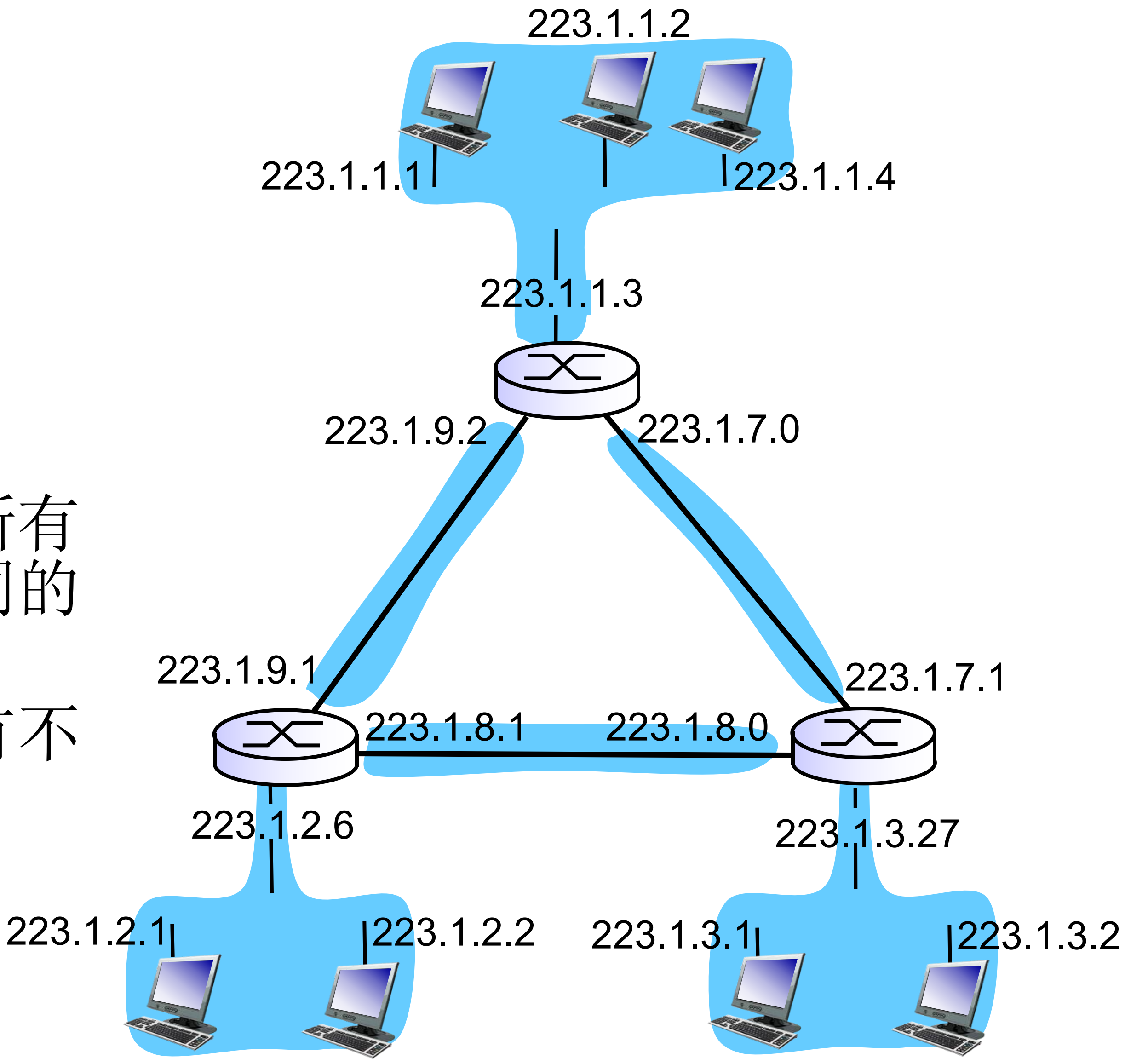
255.255.255.0 = 11111111 11111111 11111111 00000000

Subnets

how many?

6个岛或子网

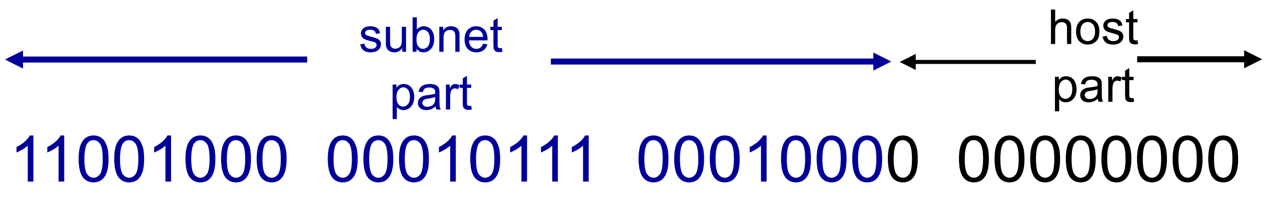
- 给定子网上的所有设备都具有相同的子网地址；
- 不同的子网具有不同的子网地址。



IP编址: CIDR 无类别域间路由选择

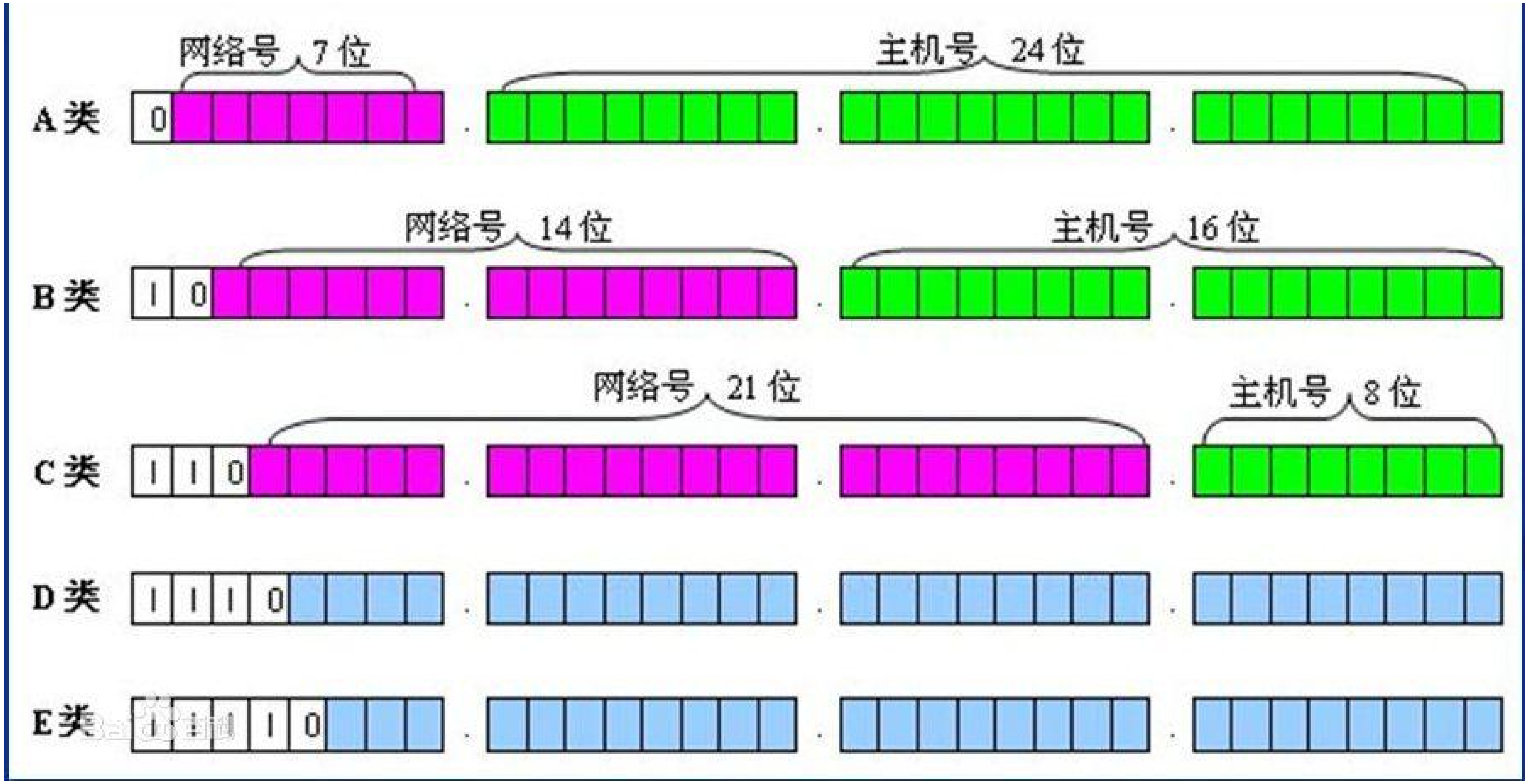
- 因特网的IP地址分配策略称为无类别域间路由策略。

  CIDR: Classless InterDomain Routing
  - 32位的IP地址被划分为两部分。
  - IP地址格式：a.b.c.d/x，x指示了IP地址的子网部分的比特数，即IP地址中前x位为子网号，也称为IP地址的前缀，或网络前缀。
  - 一个接入因特网的组织通常被分配一块连续的地址，即具有相同前缀的一块地址。
  - 前面介绍的路由器的转发表，在搜索时也是匹配网络号。



200.23.16.0/23

IP addressing: CIDR



- A类保留给政府机构，B类分配给中等规模的公司，C类分配给任何需要的人，D类用于组播，E类用于实验，各类可容纳的地址数目不同。

IP addressing: CIDR

A类地址

（1）A类地址第1字节为网络地址，其它3个字节为主机地址。
它的第1个字节的第一位固定为0.

（2）A类地址范围：1.0.0.1---126.255.255.254

（3）A类地址中的私有地址和保留地址：

① 10.X.X.X是私有地址（所谓的私有地址就是在互联网上不使用，而被用在局域网络中的地址）。
范围（10.0.0.0---10.255.255.255）

② 127.X.X.X是保留地址，用做循环测试用的。

IP addressing: CIDR

B类地址

（1）B类地址第1字节和第2字节为网络地址，其它2个字节为主机地址。它的第1个字节的前两位固定为10。

（2）B类地址范围：128.1.0.1---191.254.255.254。

（3）B类地址的私有地址和保留地址

① 172.16.0.0---172.31.255.255是私有地址

② 169.254.X.X是保留地址。

如果你的IP地址是自动获取IP地址，而你在网络上又没有找到可用的DHCP服务器。就会得到其中一个IP。

（4）191.255.255.255是广播地址，不能分配。

IP addressing: CIDR

C类地址

（1）C类地址第1字节、第2字节和第3个字节为网络地址，第4个字节为主机地址。另外第1个字节的前三位固定为110。

（2）C类地址范围：192.0.0.1---223.255.255.254。

（3）C类地址中的私有地址：

192.168.X.X是私有地址。(192.168.0.0---192.168.255.255)

D类地址

（1）D类地址不分网络地址和主机地址，它的第1个字节的前四位固定为1110。

（2）D类地址范围：224.0.0.1---239.255.255.254

（3）D类地址用于组播，或称多播。

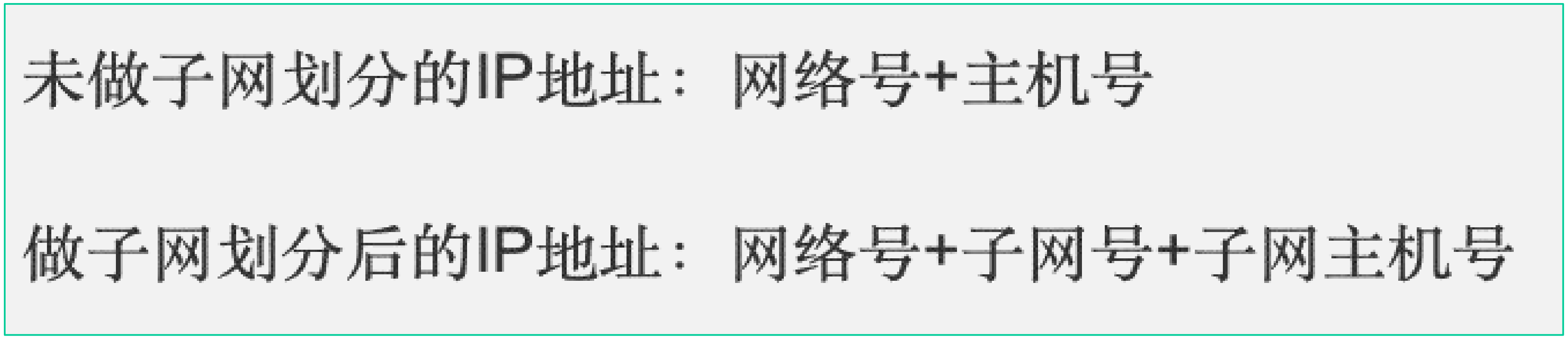
IP addressing: CIDR

E类地址

（1）E类地址不分网络地址和主机地址，它的第1个字节的前五位固定为11110。

（2）E类地址范围：240.0.0.1---255.255.255.254

- **广播地址：255.255.255.255，** 目的地址为广播时会发送给网络内的所有主机。
- IP地址如果只使用ABCDE类来划分，会造成大量的浪费：



未做子网划分的IP地址：网络号+主机号

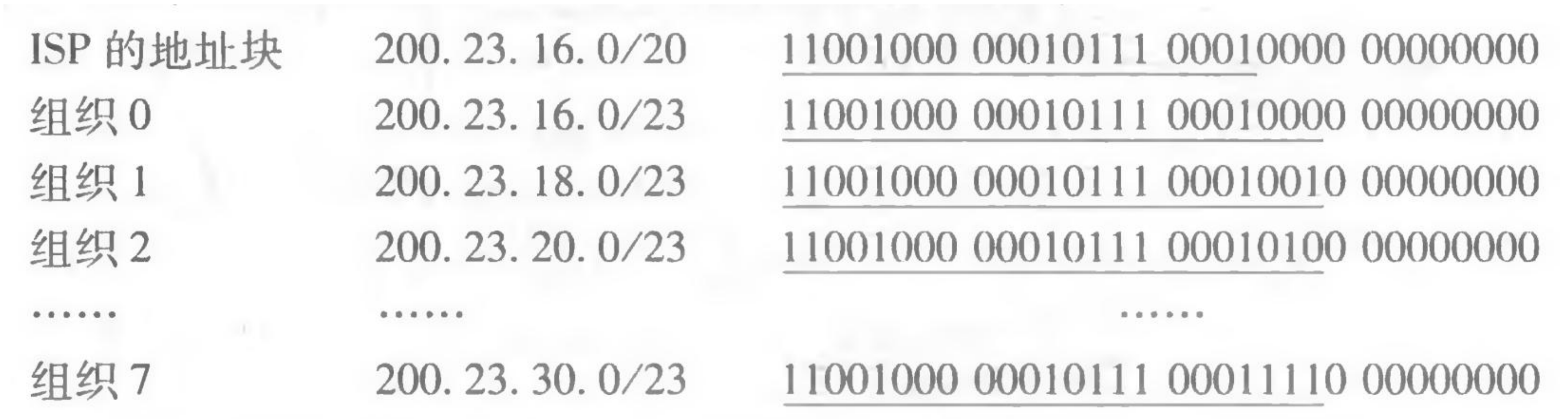
做子网划分后的IP地址：网络号+子网号+子网主机号

- 因此，IP地址还支持可变长子网掩码VLSM技术，可以在ABC类网络的基础上，进一步划分子网。

IP addresses: how to get one?

Q: 一个组织如何为其所管设备获取一块地址?

- 组织的网络管理员向组织的ISP申请，ISP从自己已经拥有的更大的地址块中划分一块提供给该组织。
- 例如，该ISP拥有地址块200.23.26.0/20，可以依次将该地址块划分成8个长度相等的连续小地址块，如下所示。

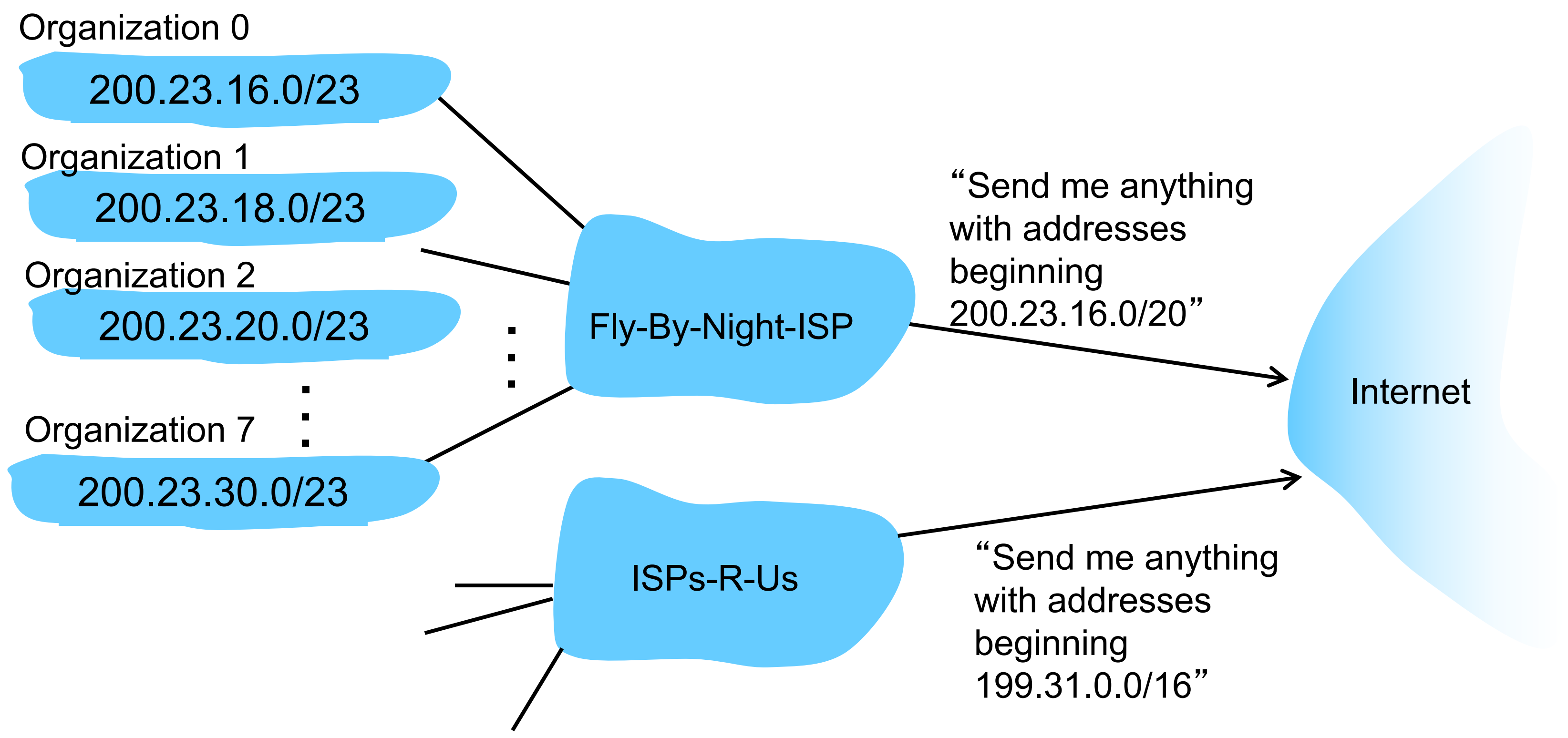


ISP 的地址块	200. 23. 16. 0/20	11001000 00010111 00010000 00000000
组织 0	200. 23. 16. 0/23	11001000 00010111 00010000 00000000
组织 1	200. 23. 18. 0/23	11001000 00010111 00010010 00000000
组织 2	200. 23. 20. 0/23	11001000 00010111 00010100 00000000
……	……	……
组织 7	200. 23. 30. 0/23	11001000 00010111 00011110 00000000

Hierarchical addressing: route aggregation

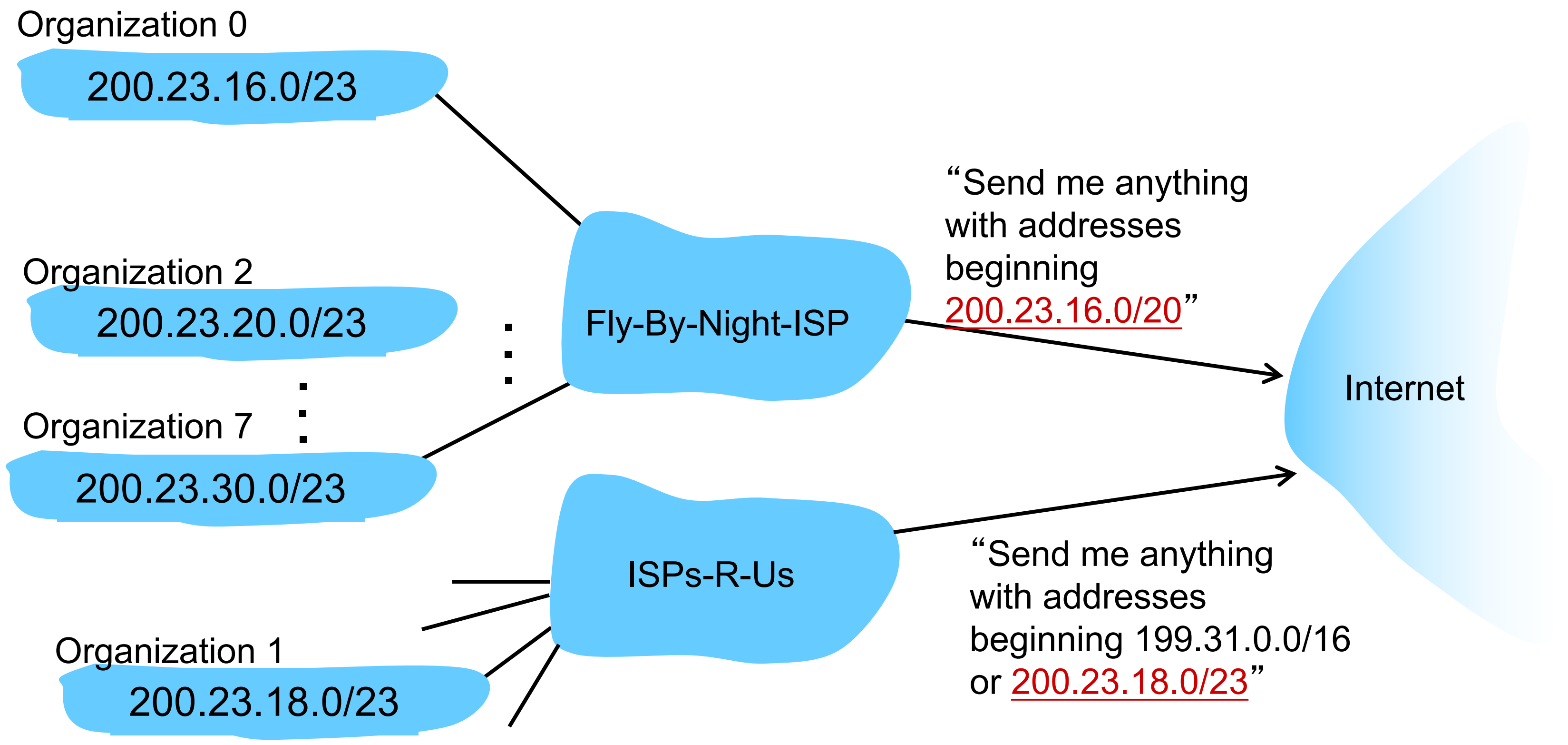
层次化地址：路由聚合

hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



IP addresses: how to get one?

Q: 一个组织如何为其所管设备获取一块地址?

- 为ISP分配地址空间的机构-
- 因特网名字和编号分配机构ICANN，是个非营利组织，管理规则基于RFC 7020，http://www.icann.org/
- 作用包括：
  - 分配IP地址块；
  - 向区域性因特网注册机构如ARIN、RIPE、APNIC、LACNIC分配IP地址块，这些支撑组负责处理本区域内的地址分配/管理；
  - 管理DNS根服务器；
  - 分配域名和解决域名纷争。

IP addresses: how to get one?

Q: 如何为一个设备从某组织的地址块分配一个地址?

- 某组织一旦获得了一块地址，就可以为本组织内部的主机或路由器接口逐个分配IP地址。
- 由系统管理员或网络管理员手动配置在主机或router中.
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- 动态主机配置协议 DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - 管理员可以配置DHCP使得某主机或某接口每次接入网络都使用一个固定IP地址。
  - 也可以为接入网络的主机分配一个临时IP地址，主机每次接入这个网络获得的IP地址都是不同的。
  - DHCP还告知接入主机：子网掩码、默认网关、本地**DNS**。

DHCP: more than IP addresses

- DHCP can return more than just allocated IP address on subnet:
  - address of first-hop router for client 默认网关、第一跳路由器
  - name and IP address of DNS sever DNS服务器地址和名字
  - network mask (indicating network versus host portion of address) 子网掩码

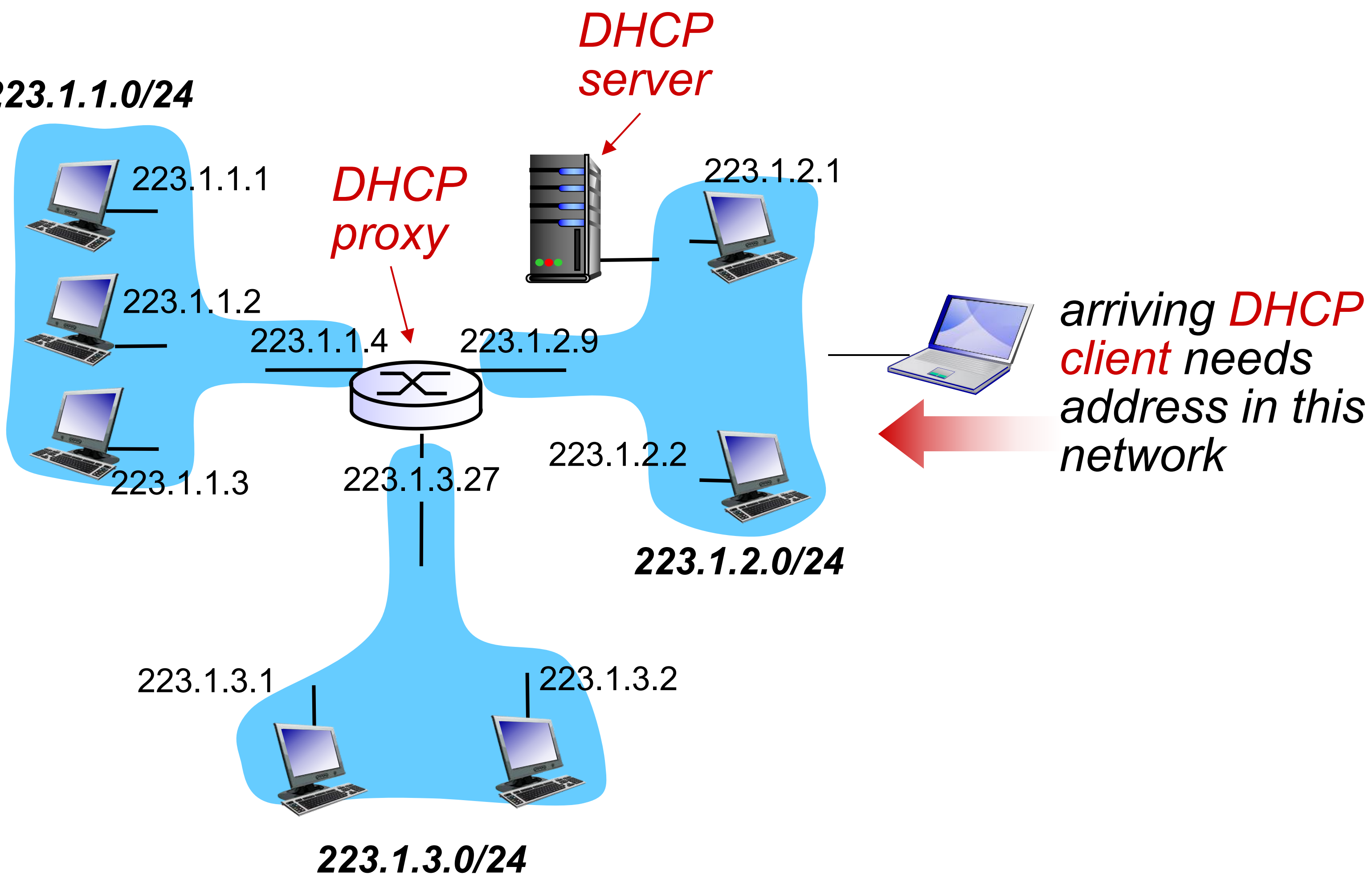
DHCP: 动态主机配置协议

goal 作用: allow host to *dynamically* obtain its IP address and relative information from network server when it joins network.

特点:

- can renew its lease on address in use 续租
- allows reuse of addresses 重用 (only hold address while connected/ “on”)
- support for mobile users who want to join network 支持移动，广泛应用于因特网住宅接入网、企业网、无线局域网中。
- “plug-and-play” 即插即用、零配置，自动工作。
- 是一个客户-服务器协议。使用DHCP协议分配IP地址的子网内通常有一台DHCP服务器；如果没有服务器则需要一个DCHP中继代理（通常也是一台路由器），代理知道该子网的DHCP服务器的地址。

DHCP client-server scenario

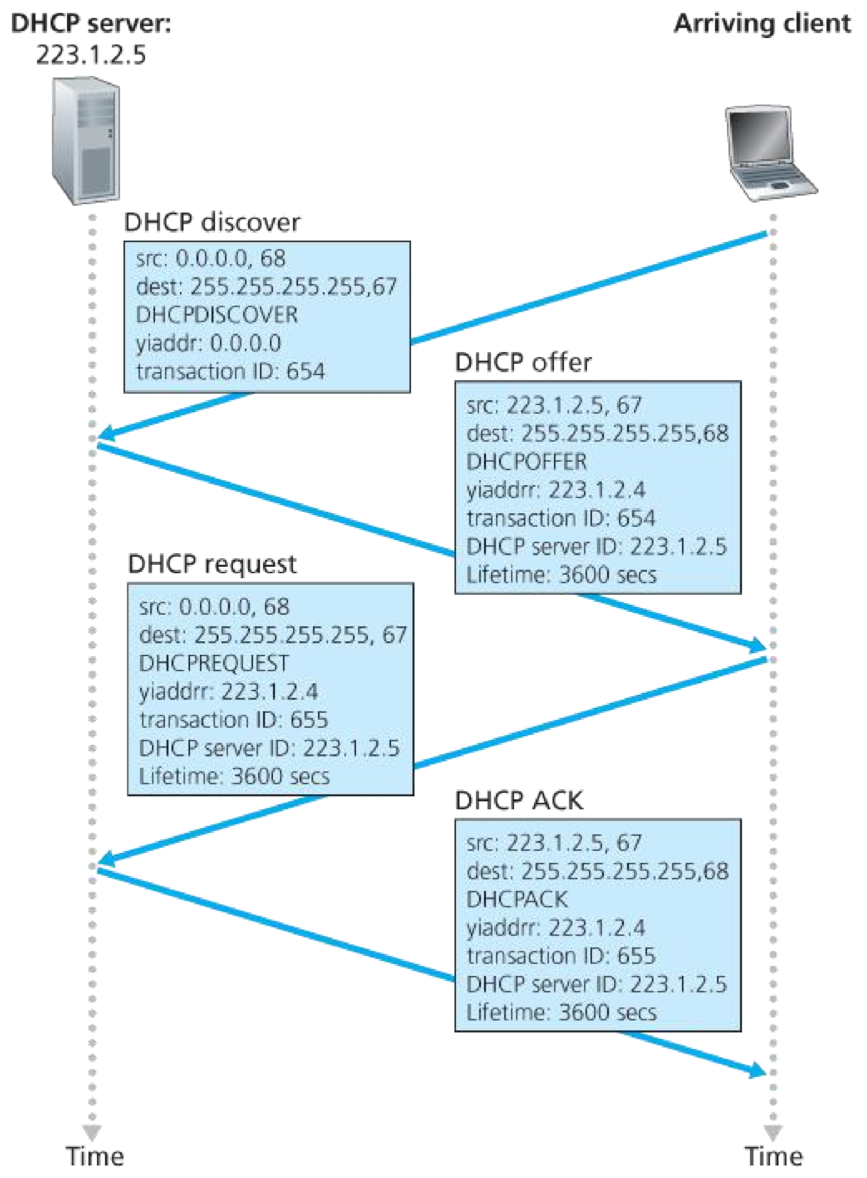


DHCP: 动态主机配置协议

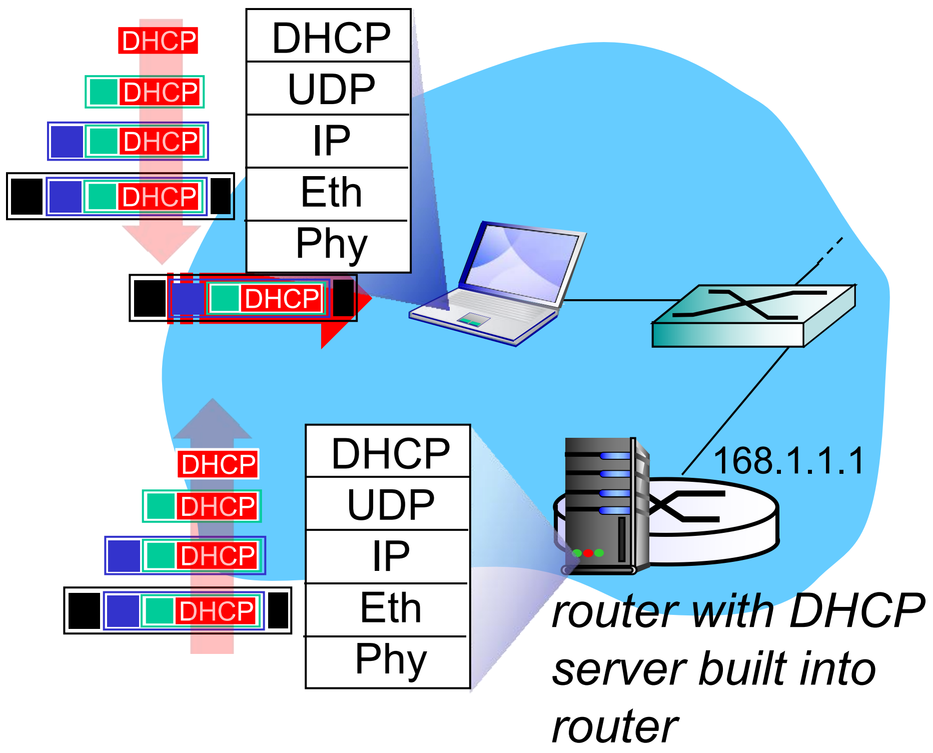
DHCP **的4个步骤：**

- DHCP服务器发现，新到达的主机向子网广播**DHCP发现报文**即“DHCP discover” msg [optional]来发现一个DHCP服务器，客户主机在UDP分组中向端口67发送该发现报文，使用广播目的地址255.255.255.255和源地址0.0.0.0。
- DCHP服务器提供，DHCP 服务器收到一个DHCP发现报文后用**DHCP提供报文**即“DHCP offer” msg [optional]向客户主机做出响应，该报文仍使用255.255.255.255目的地址广播。每台DHCP服务器提供的报文包括：DHCP发现报文里的**事务ID**、向客户主机提供的**IP地址**、**子网掩码**、**IP地址租用期**(即有效时间，通常为几小时或几天)。
- DHCP请求，客户主机从一个或多个服务器提供中选择一个，并向选中的服务器发送**DHCP请求报文**即“DHCP request” msg，回显配置的参数。
- DHCP应答，DHCP服务器用“DHCP ack” msg响应，确认。

DHCP client-server scenario

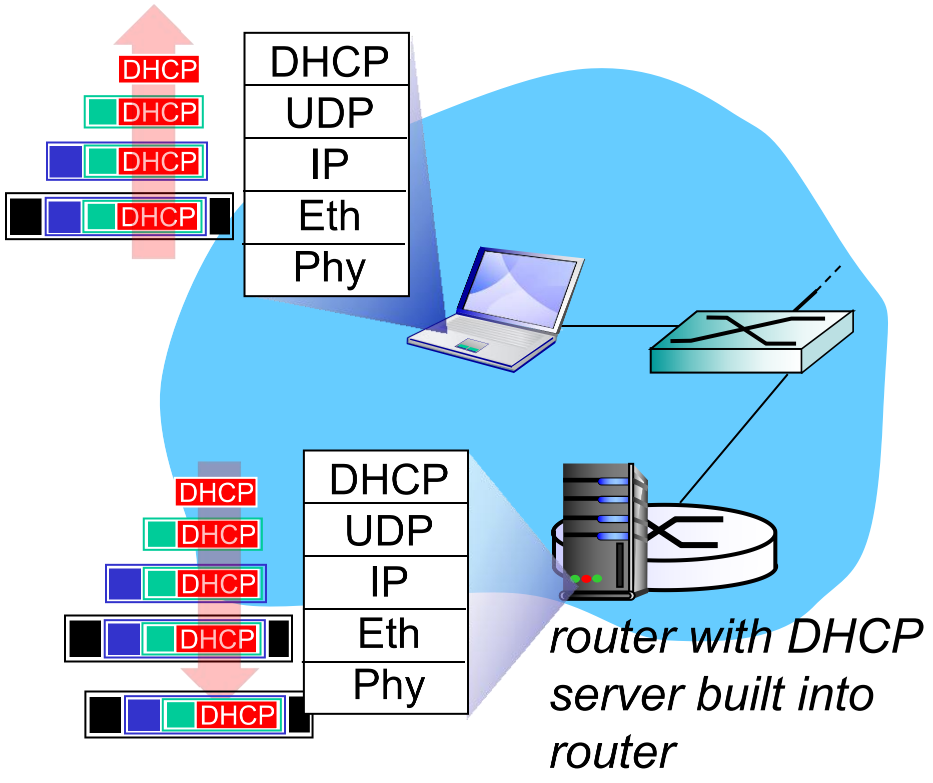


DHCP: example



- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

DHCP: example



router with DHCP server built into router

- DCP server formulates DHCP ACK containing client' s IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

DHCP: Wireshark output (home LAN)

request

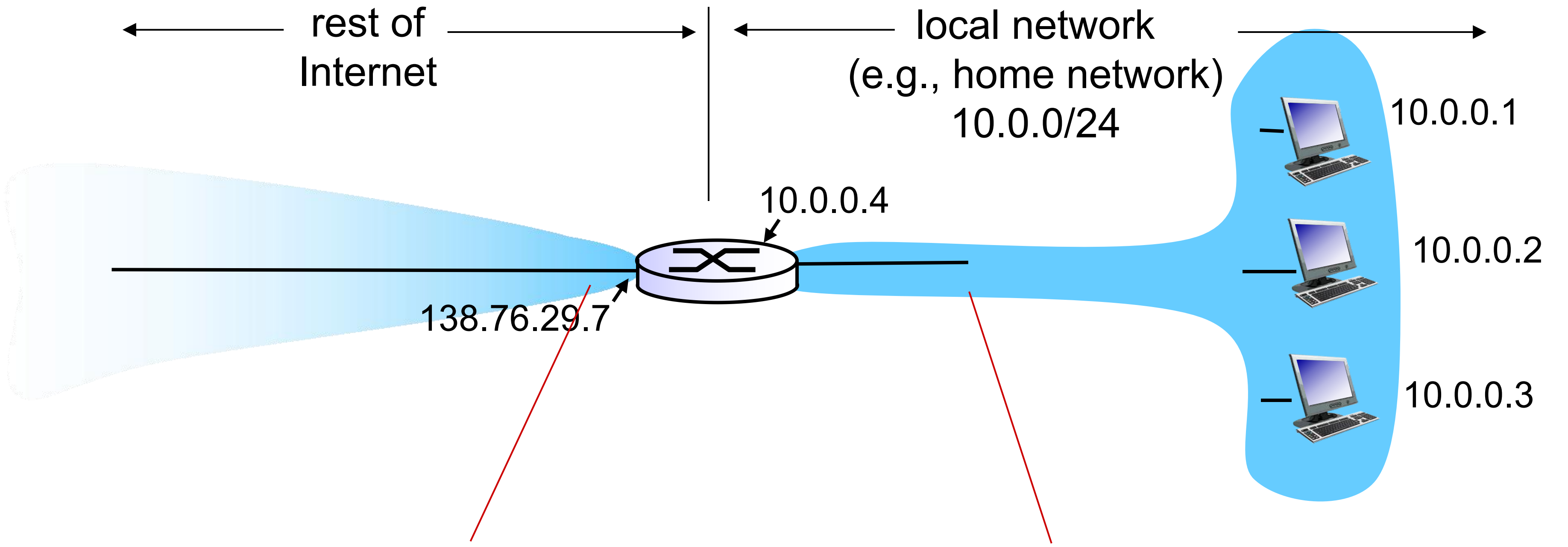
```
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    ……
```

reply

```
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
     IP Address: 68.87.71.226;
     IP Address: 68.87.73.242;
     IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."
```

Chapter 4: outline

NAT: network address translation 网络地址转换



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

motivation: local network uses just one IP address as far as outside world is concerned:

- range of addresses not needed from ISP: just one IP address for all devices 共用一个外部IP地址
- can change addresses of devices in local network without notifying outside world 可以随意更改内部地址
- can change ISP without changing addresses of devices in local network 可以更改ISP或外部地址而不改变内部地址
- devices inside local net not explicitly addressable, visible by outside world (a security plus) 安全性，设备的内部地址不暴露

NAT: network address translation

implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr

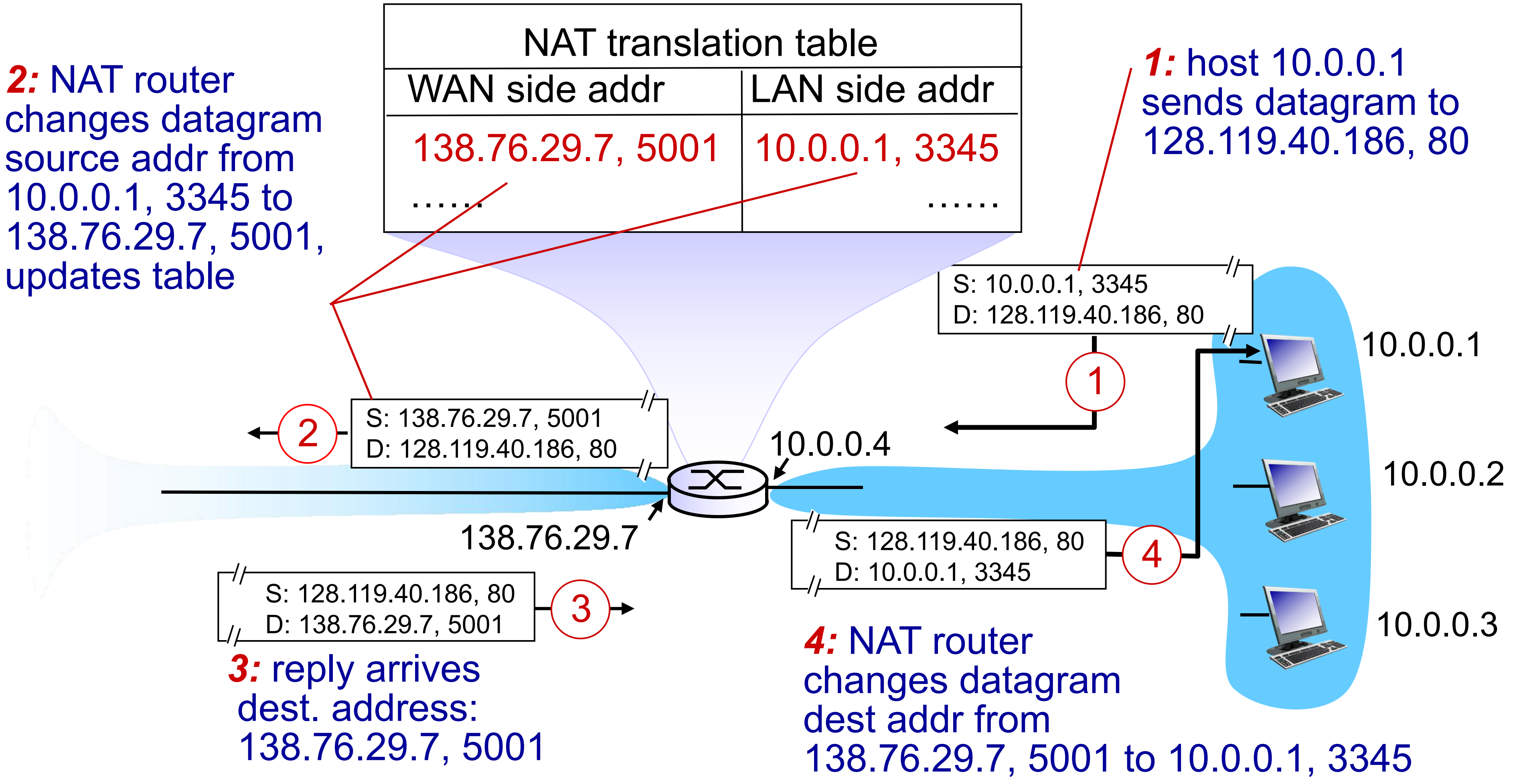
- *remember (in NAT translation table NAT转换表)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation

NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……



2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table
1: host 10.0.0.1 sends datagram to 128.119.40.186, 80
S: 10.0.0.1, 3345
D: 128.119.40.186, 80
1
10.0.0.1
S: 138.76.29.7, 5001
D: 128.119.40.186, 80
2
10.0.0.4
10.0.0.2
138.76.29.7
S: 128.119.40.186, 80
D: 10.0.0.1, 3345
4
S: 128.119.40.186, 80
D: 138.76.29.7, 5001
3
10.0.0.3
3: reply arrives dest. address: 138.76.29.7, 5001
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

NAT: network address translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - address shortage should be solved by IPv6
  - violates end-to-end argument 破坏了端到端原则
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - NAT traversal: what if client wants to connect to server behind NAT?

Chapter 4: outline

IPv6: motivation 动机

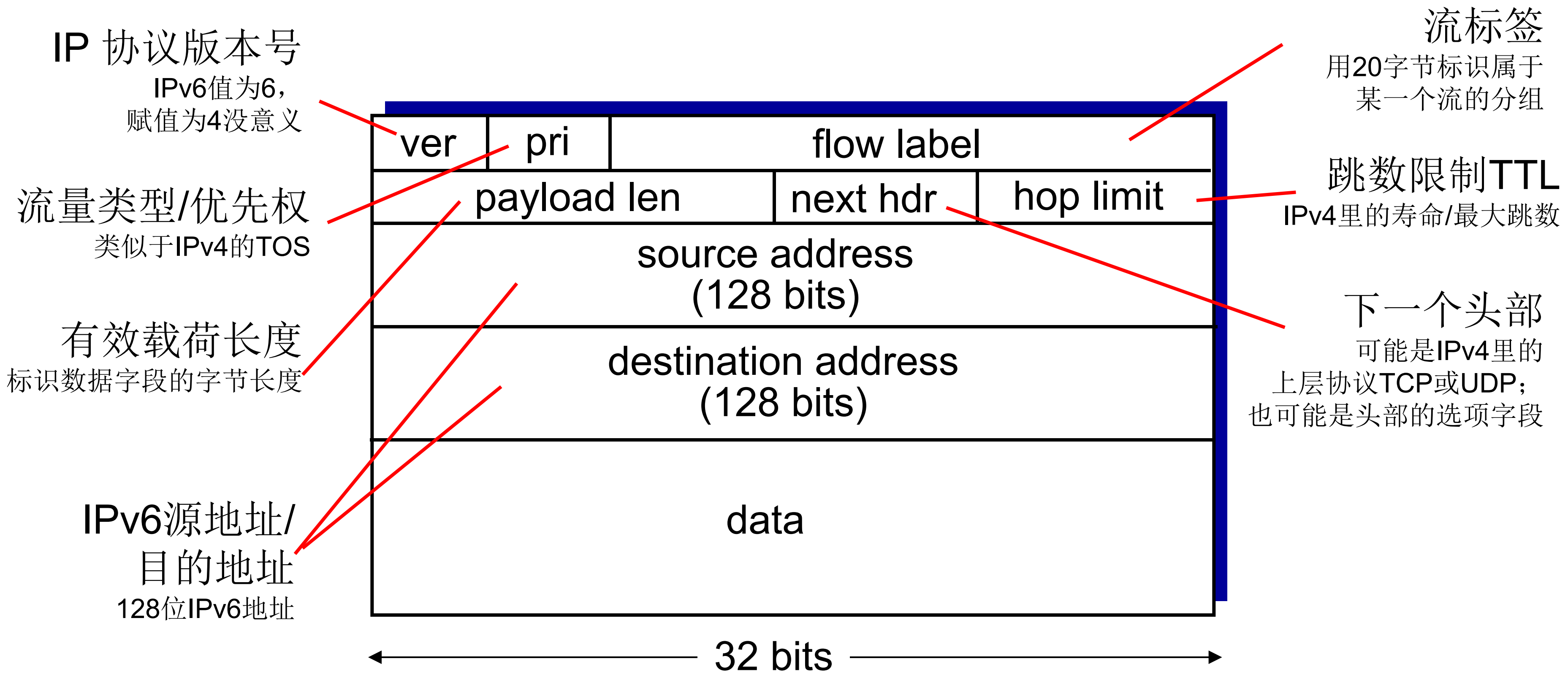
- *initial motivation:* 32-bit address space soon to be completely allocated. 初始动机：32位IP地址将用尽
  - 2011年，IANA向一个区域注册机构分配完了最后一个地址池，每个机构的地址池还有未用完的IP地址。
- additional motivation 额外动机：
  - header format helps speed processing/forwarding 头部格式有助于快速处理和转发
  - header changes to facilitate QoS 新头部格式化有助于QoS管理

IPv6 datagram format:

- fixed-length 40 byte header 头部长度固定40字节
- no fragmentation allowed 不允许分组分片

IPv6 datagram format 数据片格式

priority: identify priority among datagrams in flow
flow Label: identify datagrams in same "flow."
(concept of"flow" not well defined).
next header: identify upper layer protocol for data



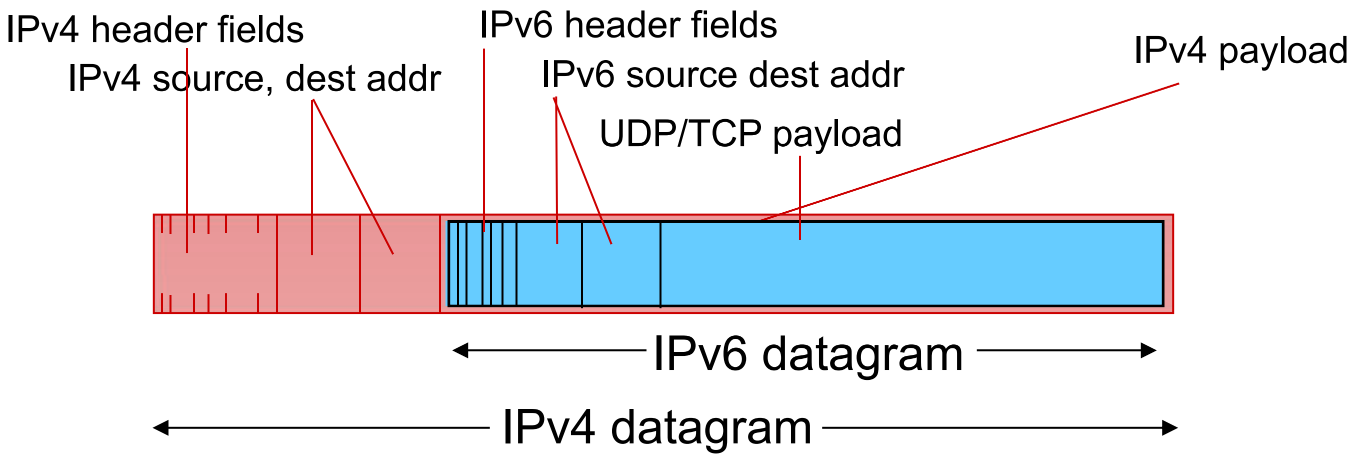
Other changes from IPv4

IPv6取消的内容：

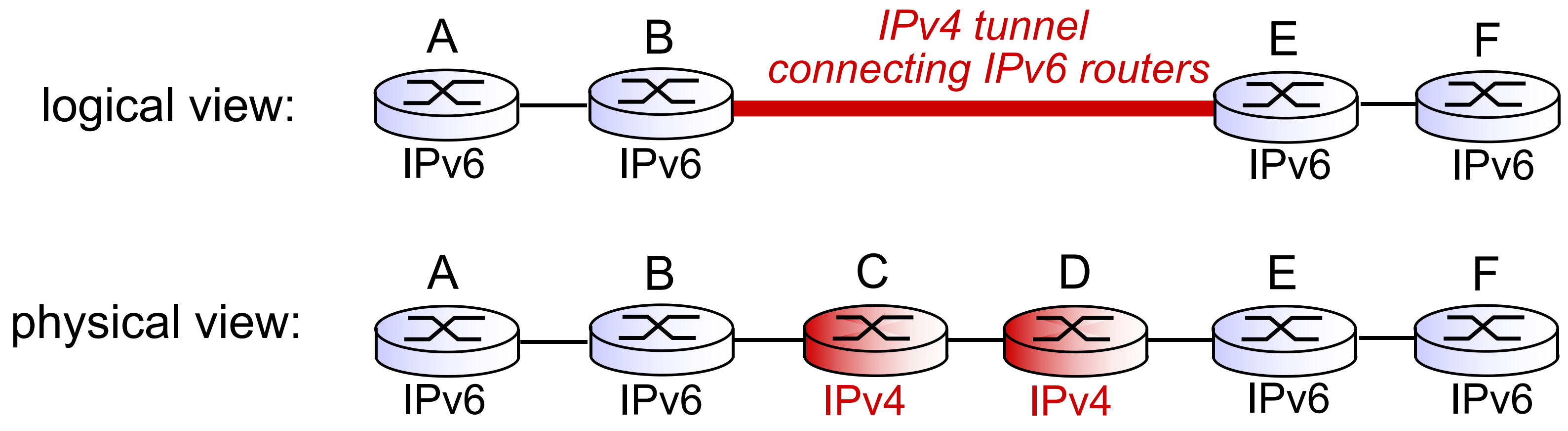
- **分片/重新装配:** 中间路由器不允许对分组进行分片，如果收到了过大的分组不能转发到输出链路上只需要将其丢弃，分组的分片和重新装配只能在源/目的端进行。
- **校验和** *checksum:* 传输层和数据链路层都有这个功能，而且由于TTL字段的存在导致校验和在每个路由器需要重新计算再写入，影响了路由器的快速转发。
- **选项** *options:* 没有完全取消，如果需要可在"下一个首部"字段中指出。
- *ICMPv6:* new version of ICMP 新版ICMP
  - additional message types, e.g. "Packet Too Big" 额外消息类型
  - multicast group management functions 多播管理功能

Transition from IPv4 to IPv6

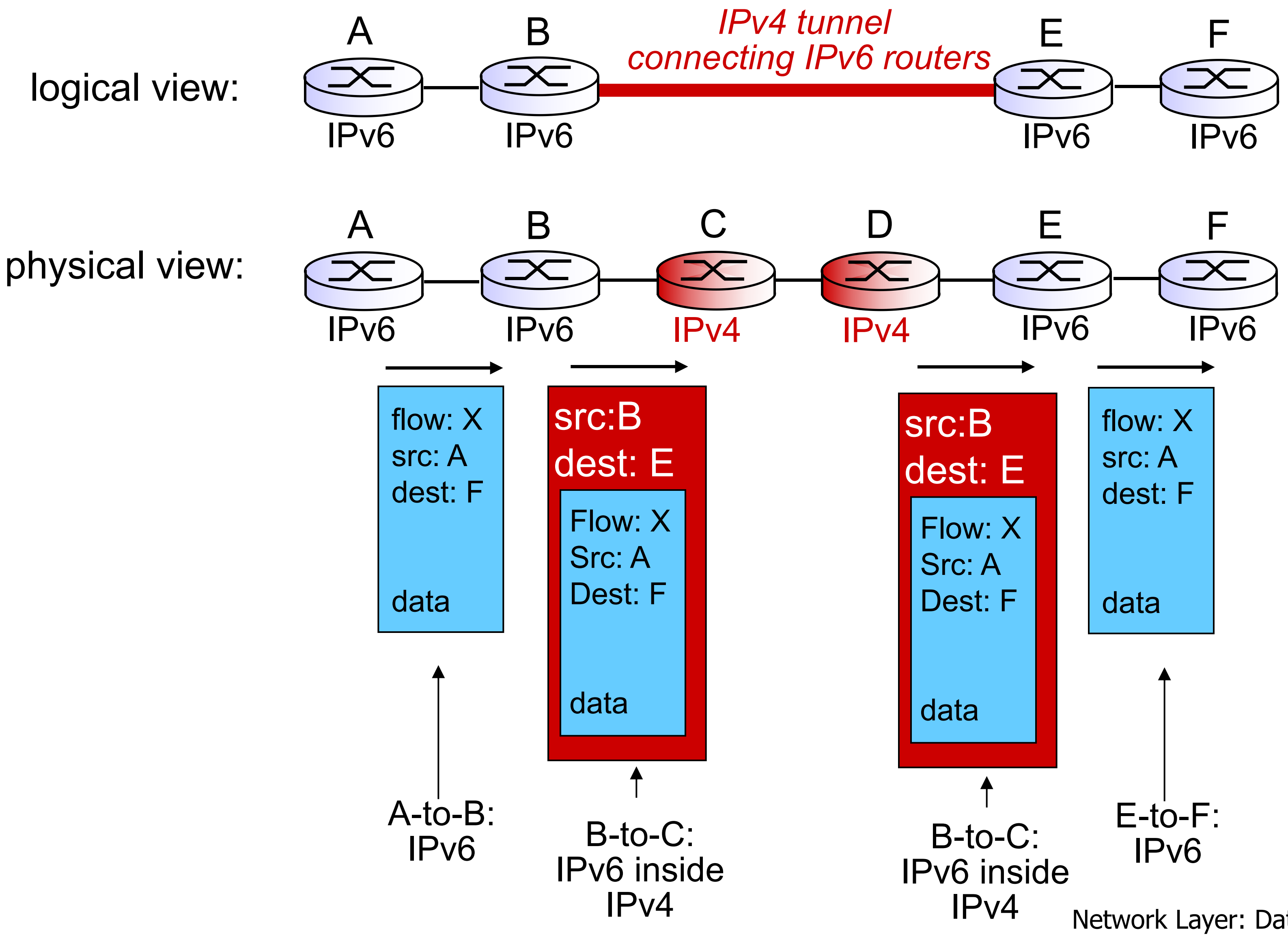
- not all routers can be upgraded simultaneously
  - no "flag days" 没有标志日，规模、成本太大
  - IPv6向后兼容，在IPv6系统里可以转发IPv4分组；但IPv4系统里却不可以转发IPv6分组。
  - how will network operate with mixed IPv4 and IPv6 routers? 如何过渡？
- 隧道 *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers 源目的IPv4地址手工配置，或者用特殊方法自动获取。



Tunneling 隧道技术



Tunneling 隧道技术



IPv6: adoption

- Google: 8% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable

- *Long (long!) time for deployment, use*
  - 20 years and counting!
  - think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, …
  - *Why?*

Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

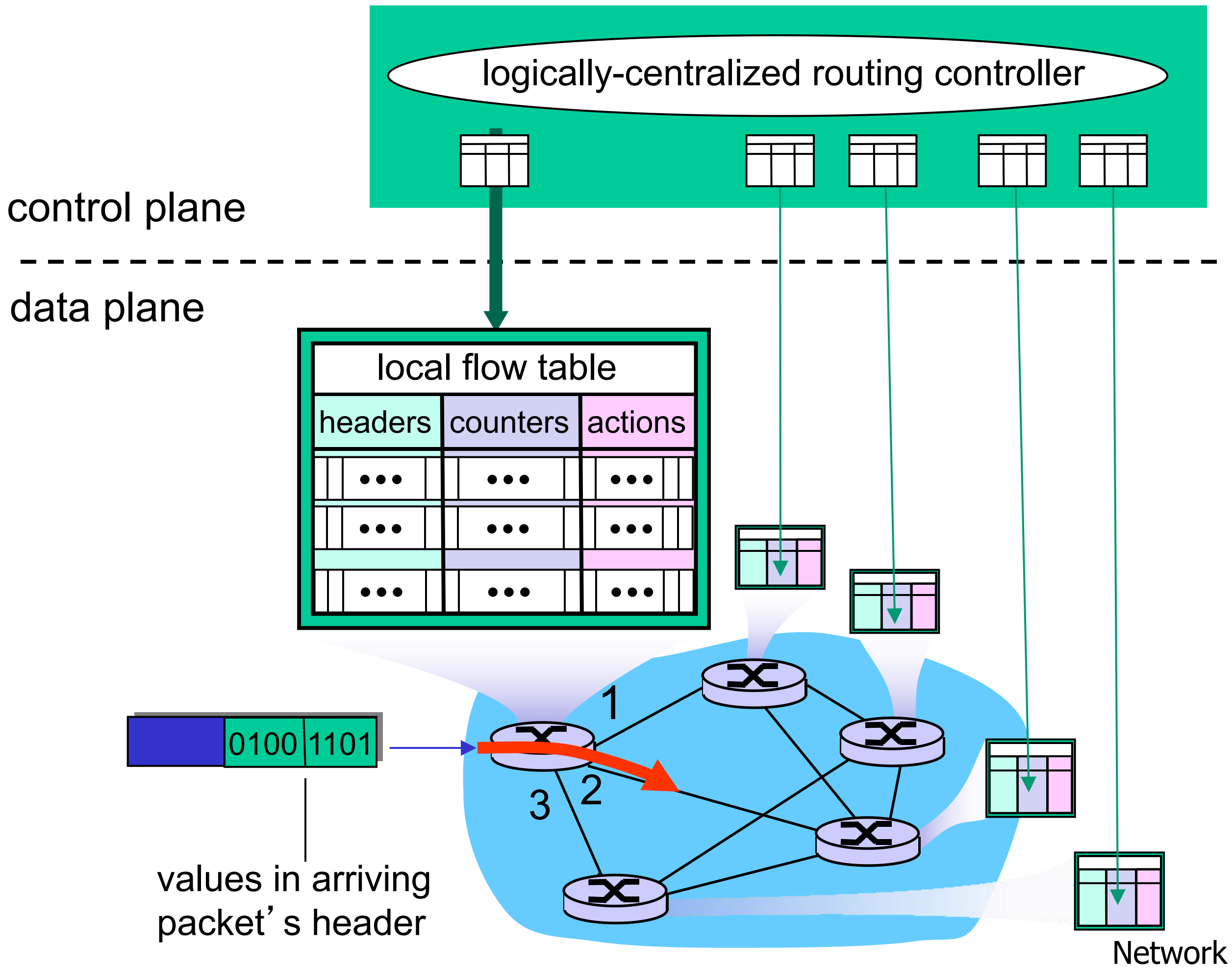
4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

4.4 Generalized Forward and SDN
- match
- action
- OpenFlow examples of match-plus-action in action

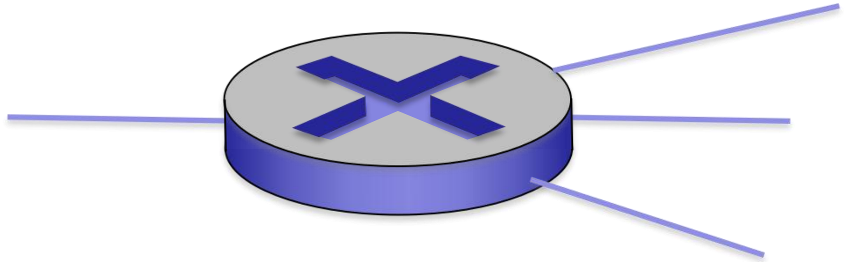
Generalized Forwarding and SDN

Each router contains a *flow table* that is computed and distributed by a *logically centralized* routing controller



OpenFlow data plane abstraction

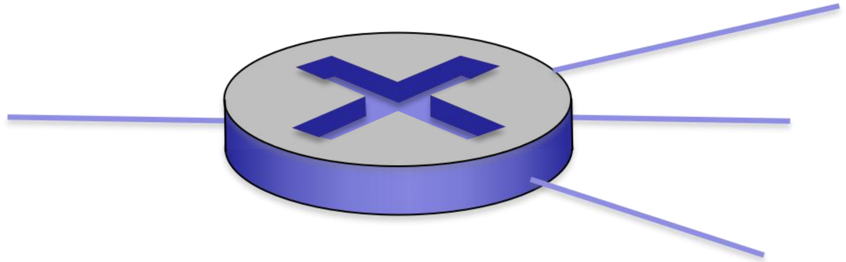
- *flow*: defined by header fields
- generalized forwarding: simple packet-handling rules
  - *Pattern:* match values in packet header fields
  - *Actions: for matched packet:* drop, forward, modify, matched packet or send matched packet to controller
  - *Priority*: disambiguate overlapping patterns
  - *Counters:* #bytes and #packets



Flow table in a router (computed and distributed by controller) define router's match+action rules

OpenFlow data plane abstraction

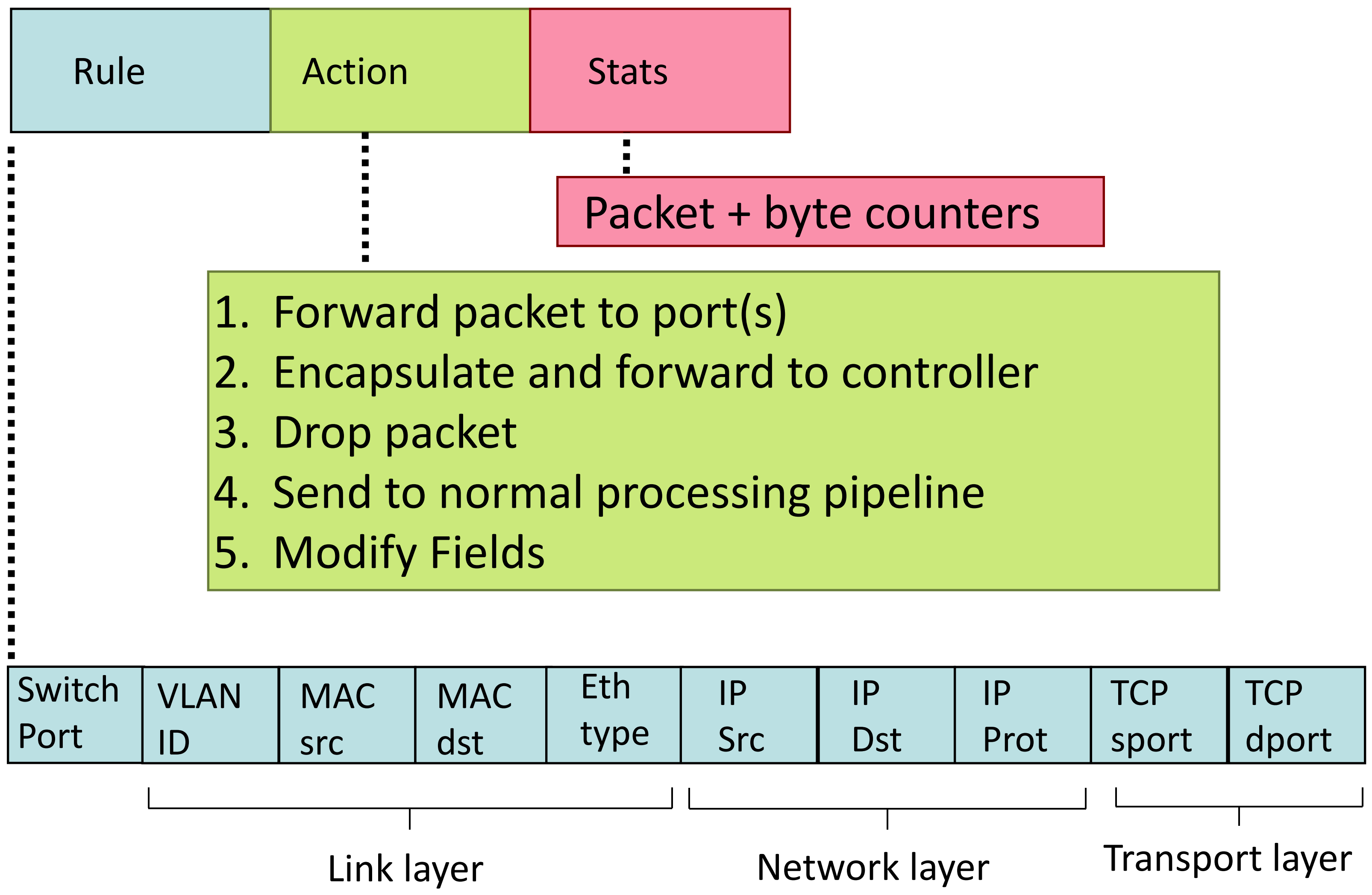
- *flow*: defined by header fields
- generalized forwarding: simple packet-handling rules
  - *Pattern:* match values in packet header fields
  - *Actions: for matched packet:* drop, forward, modify, matched packet or send matched packet to controller
  - *Priority*: disambiguate overlapping patterns
  - *Counters:* #bytes and #packets



* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

OpenFlow: Flow Table Entries



Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

Examples

Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	port3

layer 2 frames from MAC address 22:A7:23:11:E1:02 should be forwarded to output port 6

OpenFlow abstraction

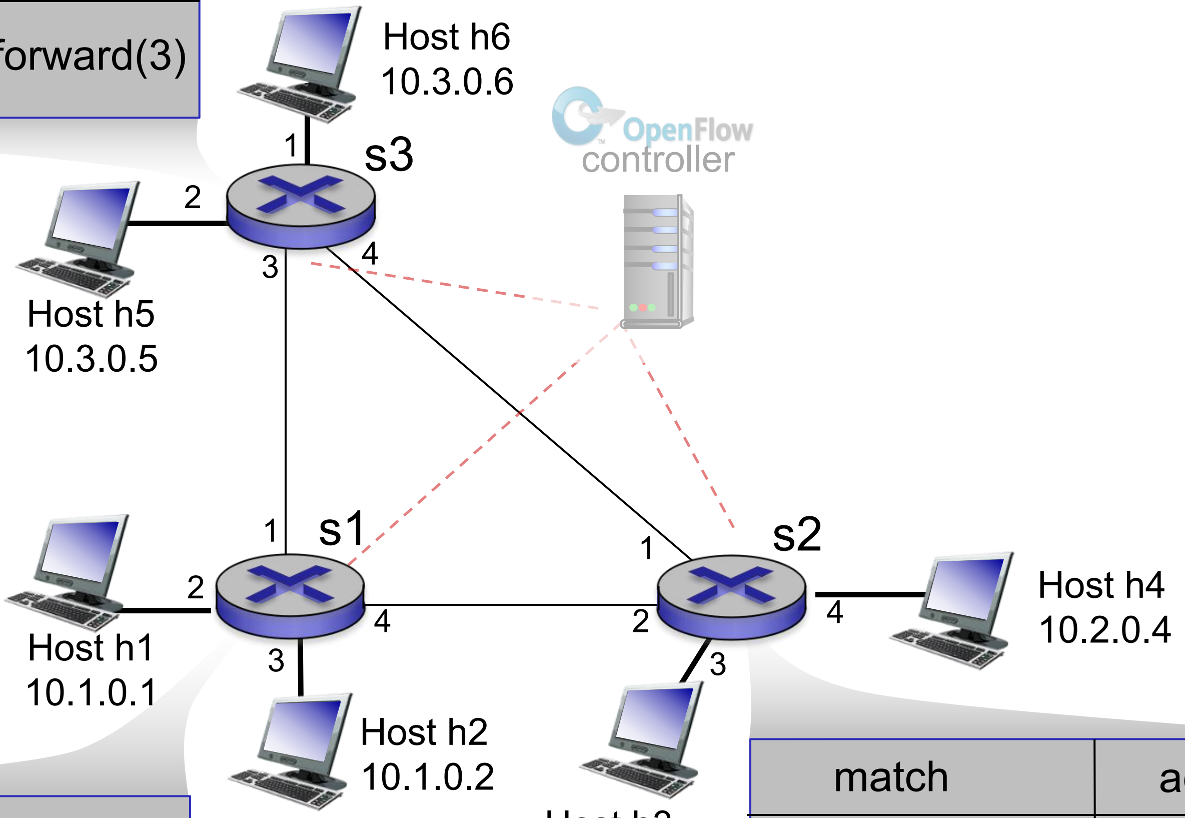
- *match+action:* unifies different kinds of devices

- Router
  - *match:* longest destination IP prefix
  - *action:* forward out a link
- Switch
  - *match:* destination MAC address
  - *action:* forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:* rewrite address and port

OpenFlow example

Example: datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

Chapter 4: *done!*

4.1 Overview of Network layer: data plane and control plane

4.2 What's inside a router

4.3 IP: Internet Protocol

- datagram format
- fragmentation
- IPv4 addressing
- NAT
- IPv6

4.4 Generalized Forward and SDN

- match plus action
- OpenFlow example

Question: how do forwarding tables (destination-based forwarding) or flow tables (generalized forwarding) computed?

Answer: by the control plane (next chapter)