计算机网络及应用

Computer Networks and Applications

第四章　网络层：数据平面

路由器原理、IP协议工作原理、通用转发和SDN

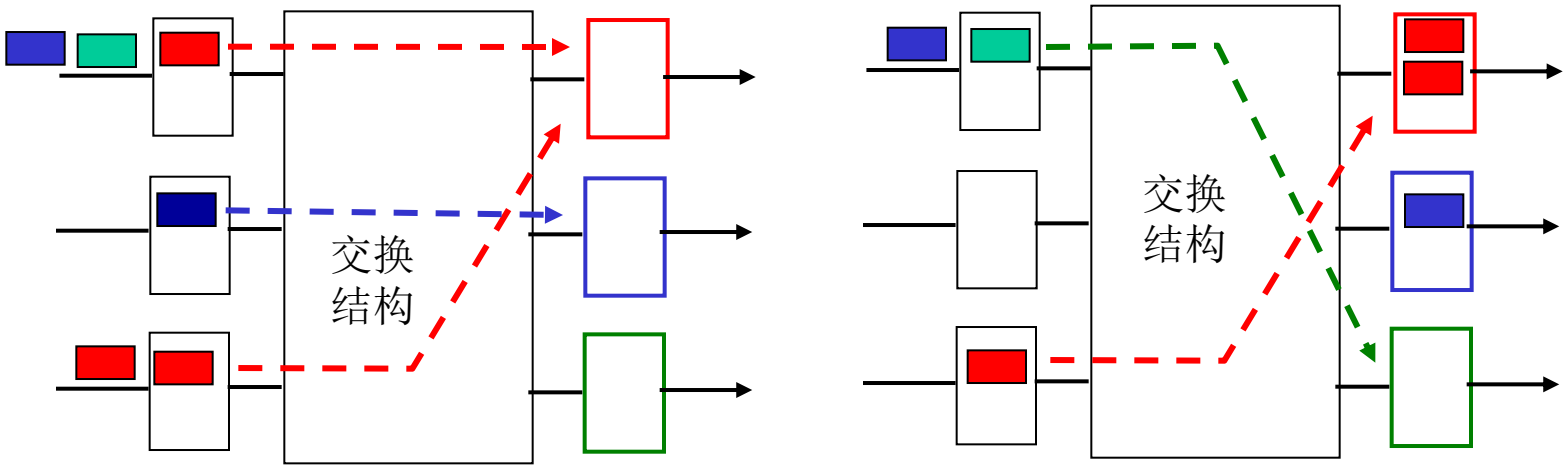
主讲：清华大学　贾庆山

教材：J.F. Kurose, K.W. Ross，Computer Networking: A Top-Down Approach, Addison Wiley, 7th Edition, 2017（机械工业出版社中文版，2018）

Special thanks to Prof. Kurose and Prof. Ross for presentation material

何时出现排队——输出端口

假设输入线路速率=输出线路速率，n个输入端口，n个输出端口。



在时刻t，
输出端口竞争

一个时间单位后

- **交换结构速率：**交换结构能够从输入端口到输出端口移动分组的速率
- 假设交换结构速率>输入、输出线路速率xn
- 最坏情况下输入端口不排队，但输出端口会排队
- 当经由交换结构的到达速率超过输出线速时开始缓存
- **排队时延以及丢包源于输出端口缓存溢出**

缓存设为多大合适？(Buffer Sizing)

- RFC 3439 的经验方法:
  - 平均缓存量等于平均RTT (例如 250 msec) 乘以链路容量 C

    B = RTT*C
  - e.g., C = 10 Gps link, 2.5 Gbit buffer
  - Rule-of-thumb（拍脑袋） G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *in Proc. of the ACM SIGCOMM '04*.
- 最近的理论和实验研究推荐:

  $$B = \frac{RTT * C}{\sqrt{N}}$$
  - 当N 个TCP流经过时, 缓存量等于
    - 缓存的必要性可近似认为是由于TCP窗口的波动性(联想排队论)
    - N个TCP的窗口变化是异步独立，中心极限定律
  - 甚至为：O(log W). W为窗口大小，10-20packets缓存量可实现 85%-90%[M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Routers with very small buffers," in *Proc. of the IEEE INFOCOM '06*.

题外：考虑WiFi情况下的路由器缓存设置

K. Jamshaid, B. Shihada, L. Xia, and P. Levis, "Buffer sizing in 802.11 wireless mesh networks," *in Proc. of the IEEE MASS'2011*.

排队和调度 —系统优化中的有趣课题

- 排队：contention for shared resources (i.e., output links)
- 调度方法

主动队列管理算法
Active Queue Management

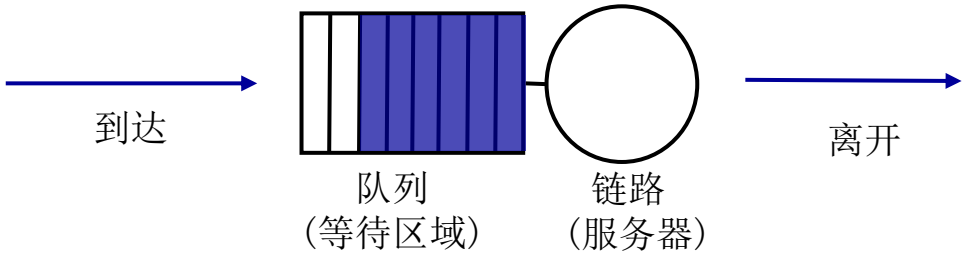
  - 先来先服务 First come first serve
  - 优先权排队 Priority queue
  - 加权公平队列 Weighted fair queue
  - 服务率控制/准入控制/资源分配/缓存优化 .....

排队论的经典教材：

1. Leonard Kleinrock, *Queueing Systems*, vol. 1: Theory, John Wiley, 1975.
2. D. Gross, J.F. Shortle, J.M. Thompson, and C.M. Harris, *Fundamentals of Queueing Theory*, 4th Edition, Hoboken: Wiley, 2008.

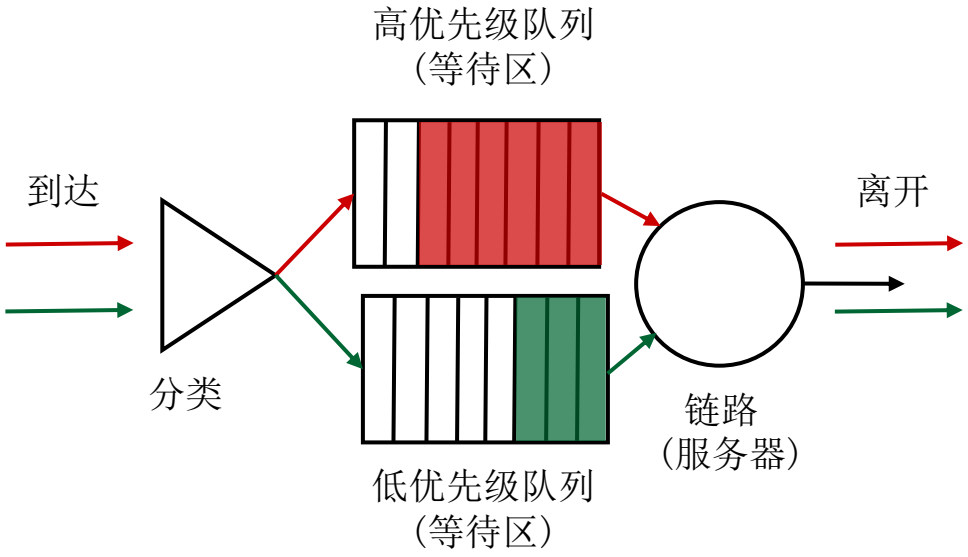
先进先出

- FIFO：按到达顺序发送到队列
- 丢弃策略：如果数据包到达满的队列：丢弃谁？
  - tail drop：删除最后到达的包
  - priority：按优先级删除
  - random：随机删除



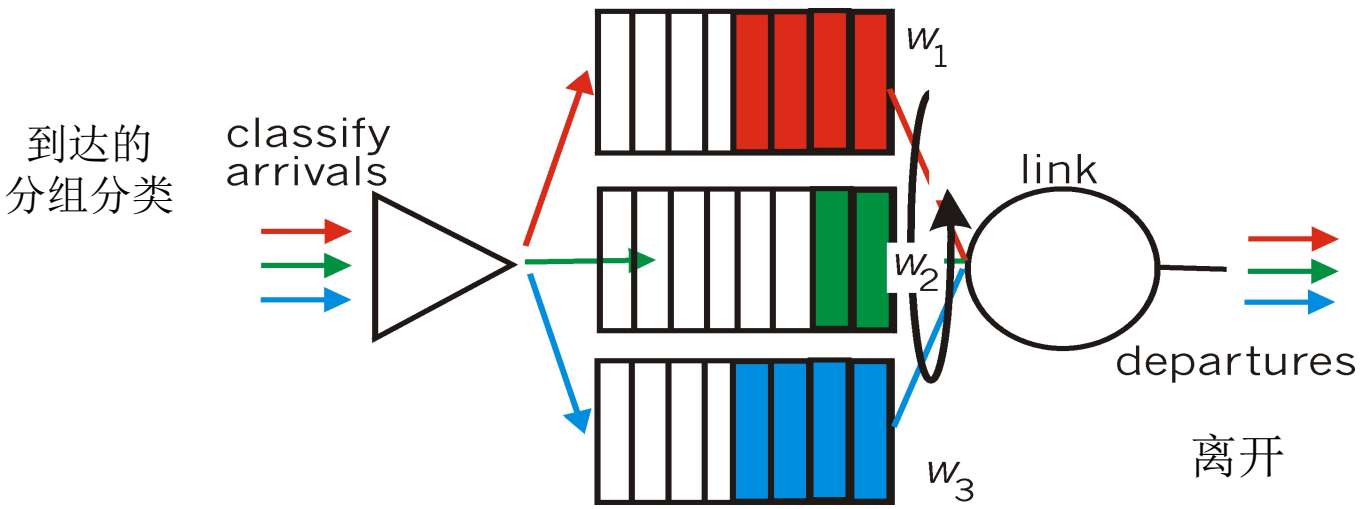
优先权排队

- Priority queue：发送最高优先级的排队数据包
- 分成多个类，优先级不同
  - 类可能取决于标记或其他头信息，例如源/目地IP、端口号等。



加权公平排队

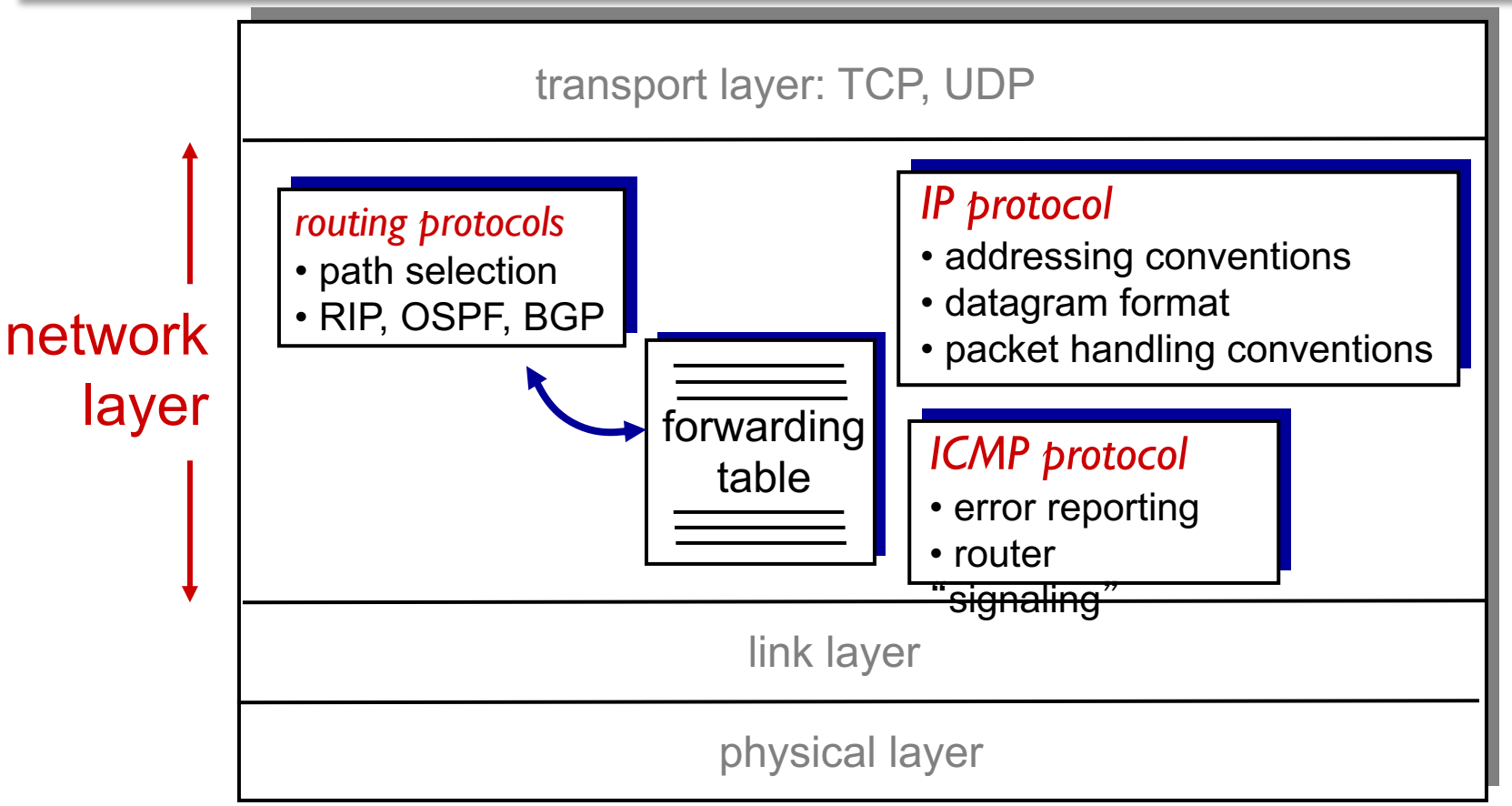
- WFQ(Weighted Fair Queuing)：
  - 以循环的方式为各个类提供服务
  - 从每个类发送一个完整的数据包（如果可用）



提纲

- 概述
- 路由器工作原理
- **网际协议——IPv4,寻址,IPv6及其他**
- 通用转发和SDN

网络层协议IP—因特网中的转发和编址



网络层的三大组件：**IP协议，选路协议，ICMP协议(互联网控制报文协议)**

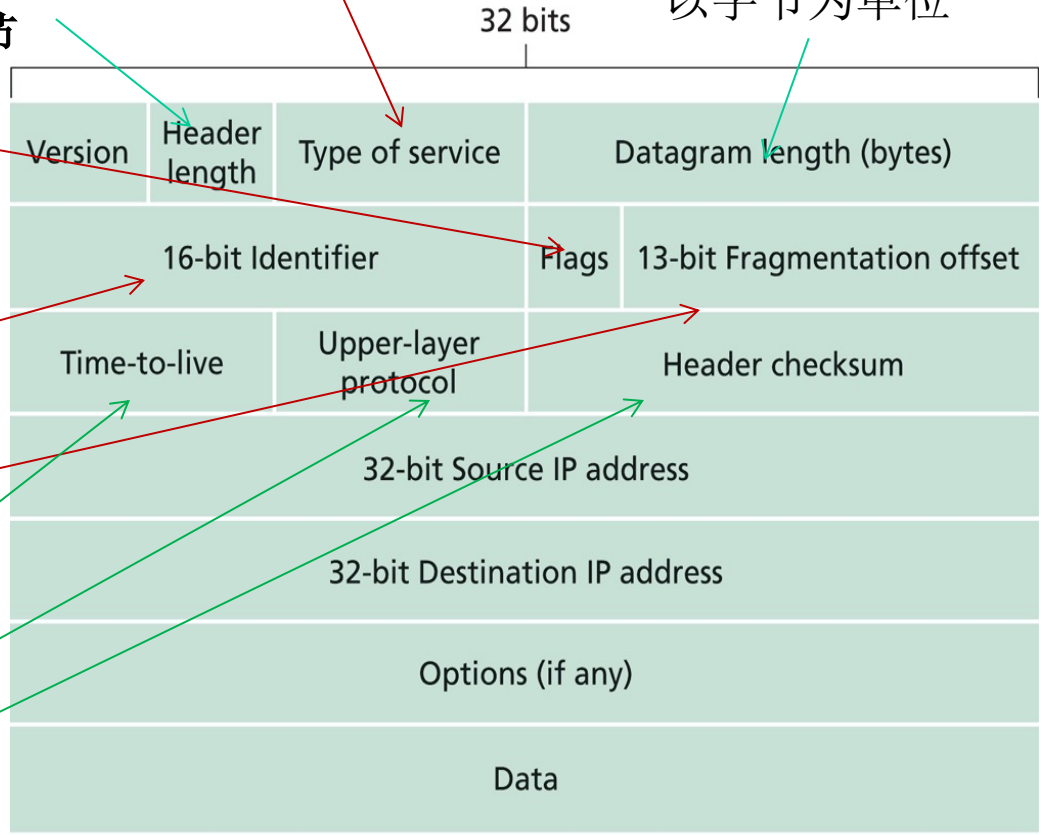
IP 数据报格式

- 标志
  2：Unused
  1： More frag
  0： Don't frag
- 标识符 Ident: ID for a packet (数据报的"碎片"共享相同的标识符)
- 偏移量Offset: for fragments
- 寿命 TTL: Time to live, unit of hops(保鲜期)
- Protocol：high-level protocol ID, i.e. TCP(6),UDP(17)
- Checksum

报头长度
一般为**20**字节

区分不同类型
的IP数据报

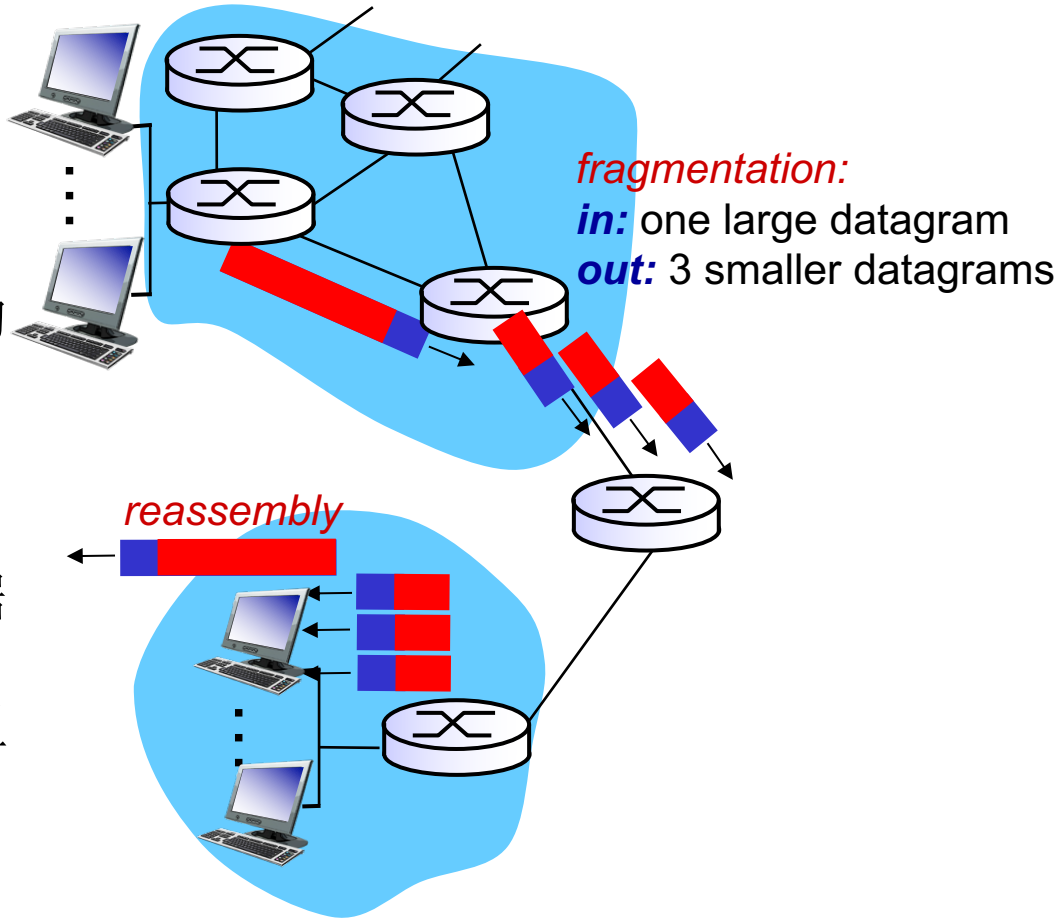
数据报长度
以字节为单位



TCP报头多长?
20 bytes of TCP
20 bytes of IP
= 40 bytes + app layer overhead

IP分片Fragmentation和重组 Reassembly

- 网络链路层有最大传输单元MTU限制 (max. transm. unit) - 最大可能的链路层帧能承载的数据量
  - 不同的链路类型，不同的MTUs
- 大的IP数据报在网络中被划分为小“片”
  - 一个数据报变为几个数据报
  - 仅在最终目的地“重新组装”
  - IP 报头位用来对相关的片进行识别、排序



IPv4将数据报重新组装工作放在端系统，而不是网络路由器。

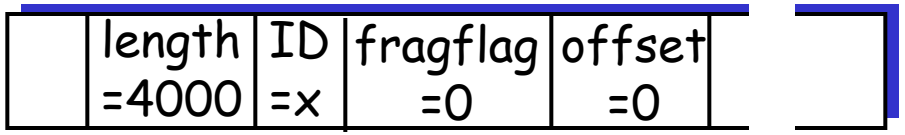
不同链路的MTU

MTU for diverse Links

Protocol	MTU(Byte)
Hyperchannel	65535
Token Ring(16 Mbit/s)	17914
Token Ring (4 Mbit/s)	4464
FDDI	4352
Ethernet	1500
X.25	576
PPP	296

理论上限

IP 分片和重组



Example

- 4000 byte 数据报
- MTU = 1500 bytes

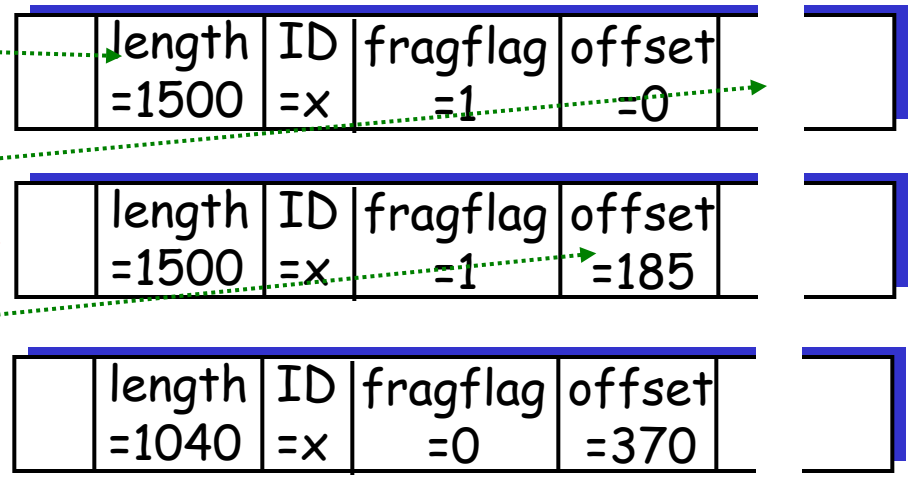
把一个大的数据报分为几个更小的数据报

表示后面还有

1500=1480+20 bytes

1480 bytes in data field

offset = 1480/8



代表长为8个字节的数据块数

4000=1480+1480+1020+ 20(IP首部)

IPv4 数据报要点

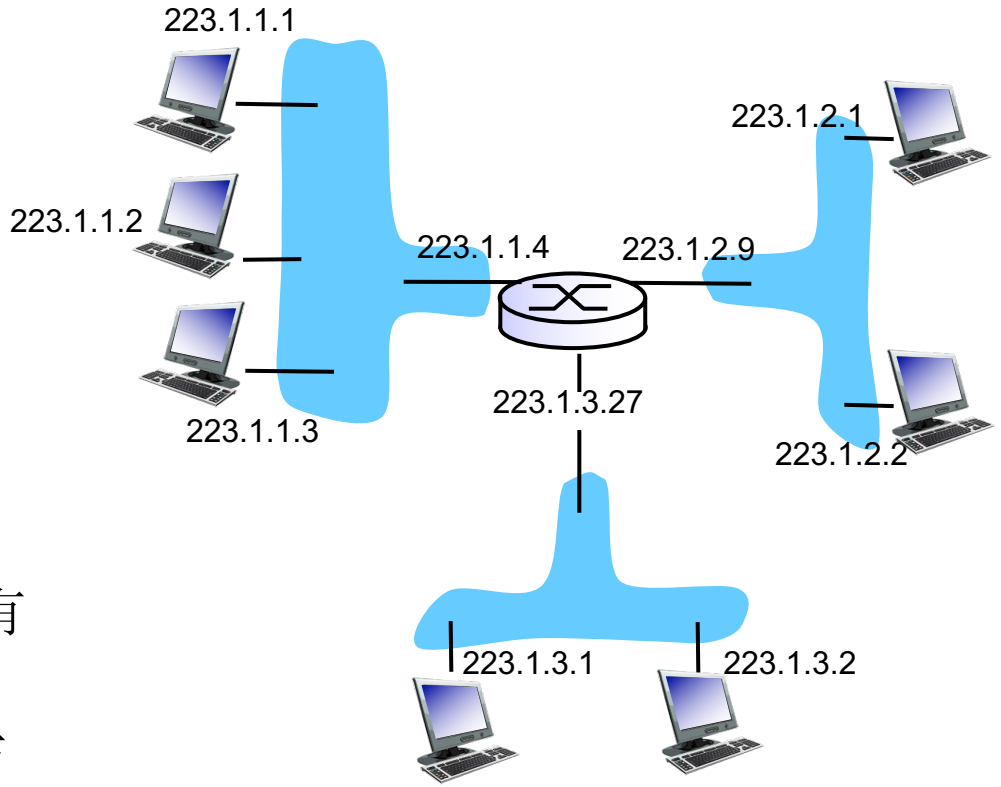
— not in IPv6

- 分片和偏移 offset
- 标识符 Identification
- 源和目的地址
- 寿命 TTL, 服务类型 TOS
- 首部长度 (4bit: max. length=4B*15=60B)
- IP 数据报长度 (16bit=1B*65535=65535B)

- 分片的代价

IPv4编址：概述

- IP地址：主机、路由器接口 interface 的32-bit识别号
- 接口 *interface*：主机、路由器和物理链路之间的连接
  - 路由器通常有多个接口
  - 主机一般有1-2个接口(有线网口、无线网口)
- IP地址与接口相关，而不是主机或路由器



223.1.1.1 = 11011111 00000001 00000001 00000001

点分十进制记法 223 1 1 1

IP 网络

- IP 地址
  - 网络部分 (high order bits)
  - 主机部分 (low order bits)
- 什么是 子网 ?
  - IP地址具有相同的子网部分的设备接口
  - 在物理上可以相互连通而不经过路由器

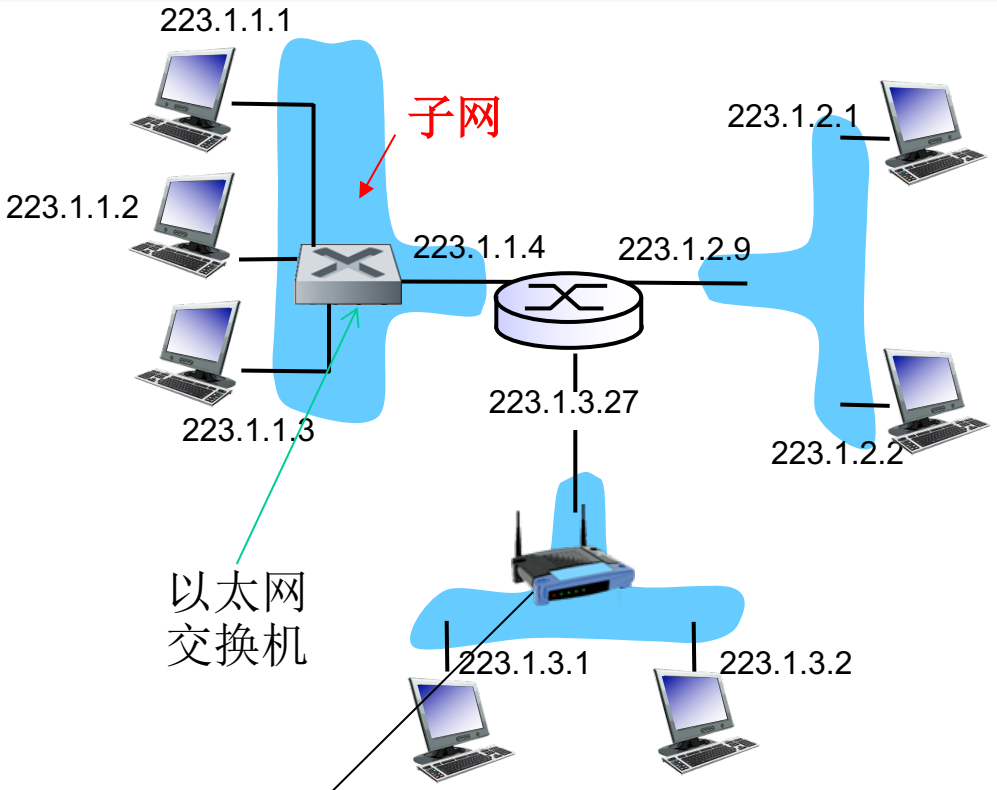
IP地址：223.1.1.1

网络部分　　主机部分

网络地址：223.1.1.0 / 24

书中称子网地址

子网掩码或前缀长度



A: wireless WiFi interfaces connected by WiFi base station

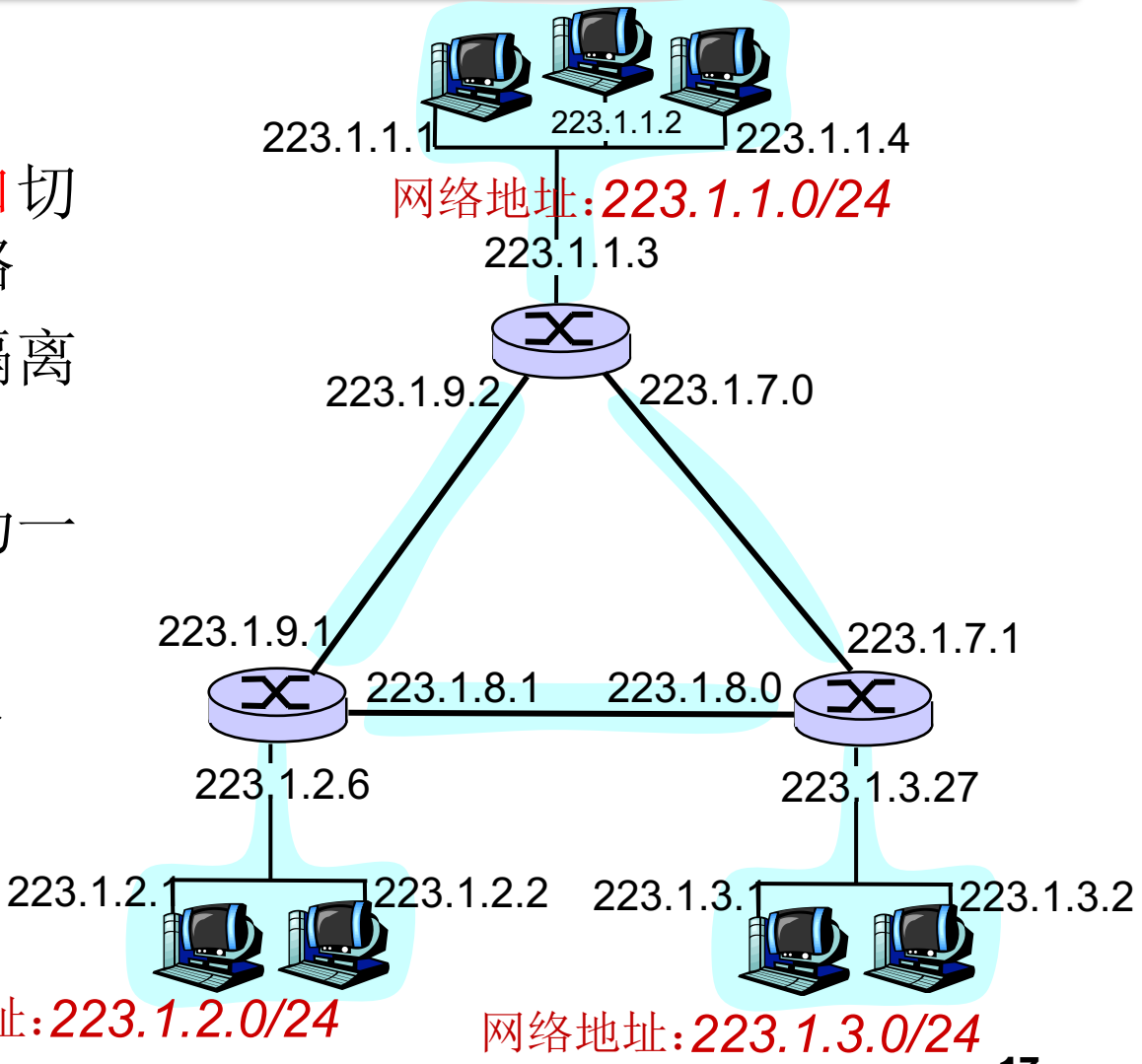
network consisting of 3 subnets

子网 (IP 网络)

怎样确定子网?

- 将路由器的每个接口切一刀，保留其他链路
- 创建了几个“相互隔离的网络孤岛”
- 每个孤立的网络称为一个“子网”

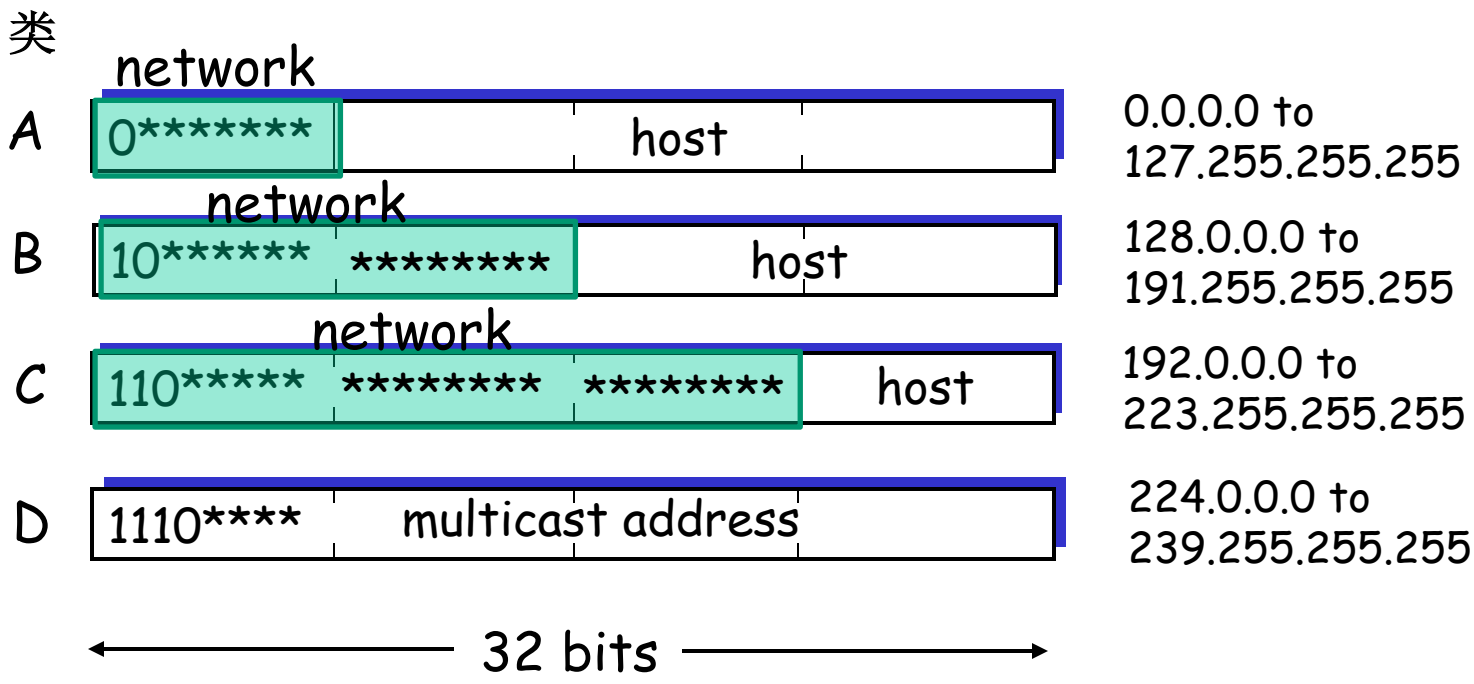
右边网络一共有6个子网



IP 编址: 分类编址（传统方法）

已经定义了 "network", 重新检查 IP 地址

分类编址 classful addressing



IP 嗜守□低□□申

网络类别	最大网络数	第一个可用的网络号	最后一个可用的网络号	每个网络中的最大主机数
A	126 ($2^7 - 2$)	1	126	16777214
B	16384 (2^{14})	128.0	191.255	65534
C	2097152 (2^{21})	192.0.0	223.255.255	254

问：清华校园网的IP地址属于什么网络类别？

A, B, C类地址的问题

- 低利用率
  - 例如，有2000 台主机的一个网络需要分配一个 B 类地址, 但浪费掉63000 个地址
  - 如果给每一个物理网络都分配一个网络号net-ID，路由表会变得过大
- 不灵活
- 解决方案
  - 进一步划分子网 或者无类别域间选路CIDR？

无类别域间选路: CIDR

- 分类编址 (before 1993):
  - 地址空间不能充分利用；接近耗尽
  - 已不再作为IP编址体系结构的正式部分
- CIDR: Classless InterDomain Routing 无类别域间选路 (1993-，不区分ABC类地址)
  - IP 地址的网络部分可以任意长
  - 地址格式: a.b.c.d/x, 其中 x 是 IP 地址网络部分的位数

200.23.16.0/23

怎样获得一个IP 地址?

主机部分(host portion):

- 手工分配，配置静态IP地址
- DHCP: 动态主机配置协议：动态地获得地址,指定IP地址、子网掩码、网关、DNS、IP租期等
  - 即插即用

DHCP:动态主机配置协议

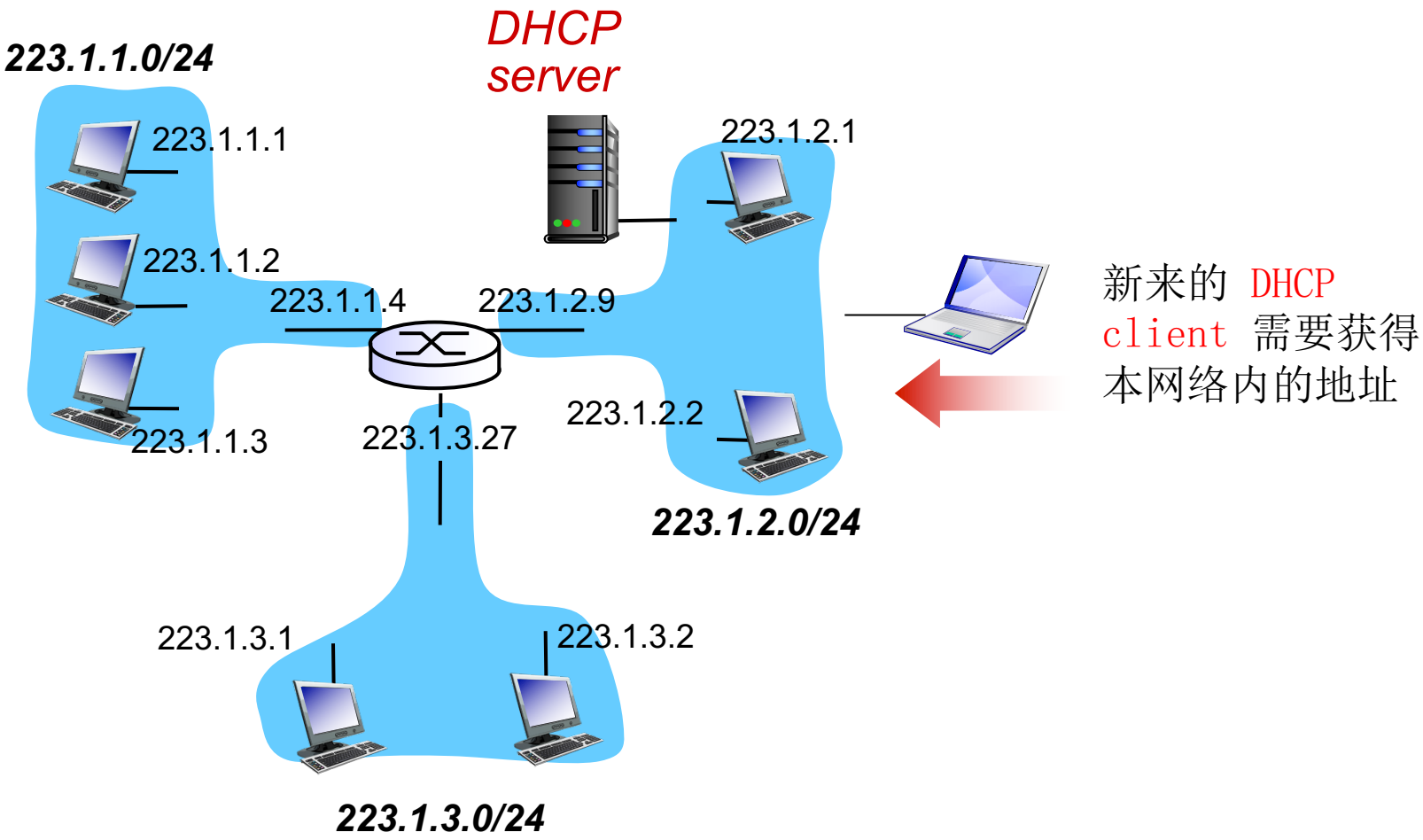
目标: 允许主机在加入网络时从网络服务器动态获取其IP地址

- 可以对正在使用中的地址续租
- 允许重复使用地址
- 支持希望加入网络的移动用户

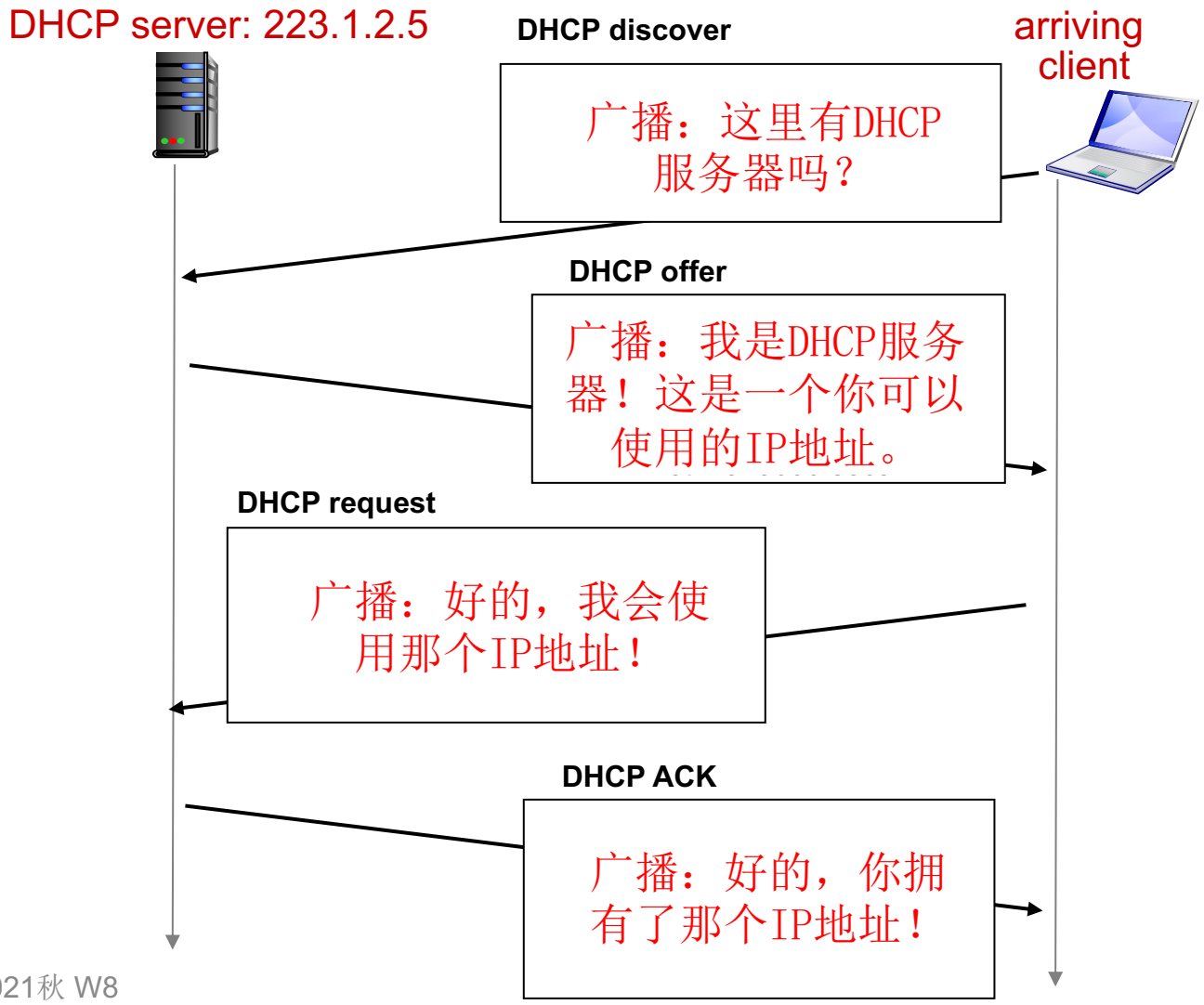
DHCP概述:

- 主机广播"DHCP discover"报文
- DHCP服务器用"DHCP offer"报文响应
- 主机请求IP地址: "DHCP request" 报文
- DHCP服务器发送地址: "DHCP ack"报文

DHCP client-server 场景



DHCP client-server 场景

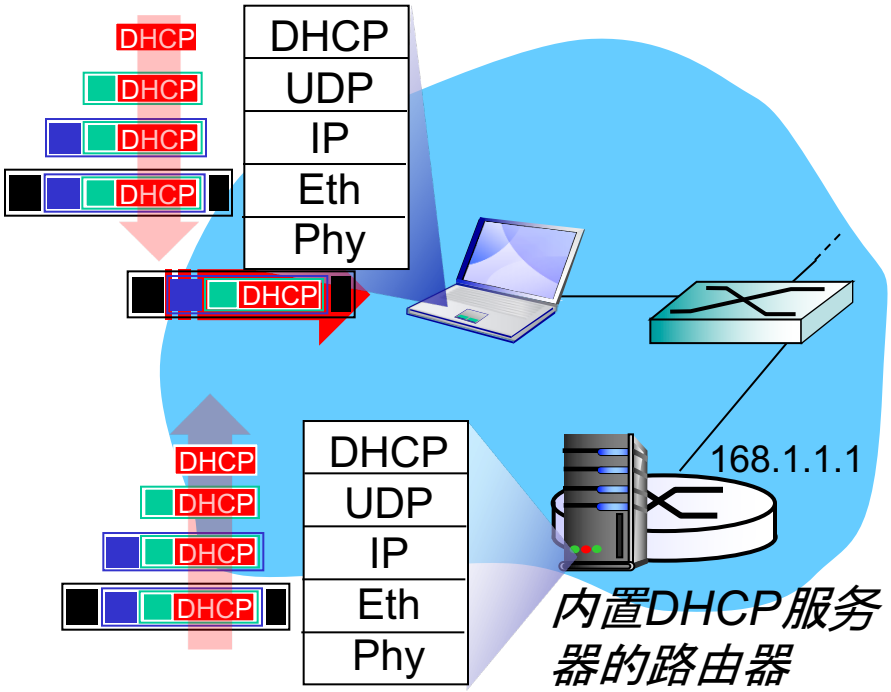


DHCP:动态主机配置协议

DHCP不仅仅可以返回子网内主机分配的IP地址：

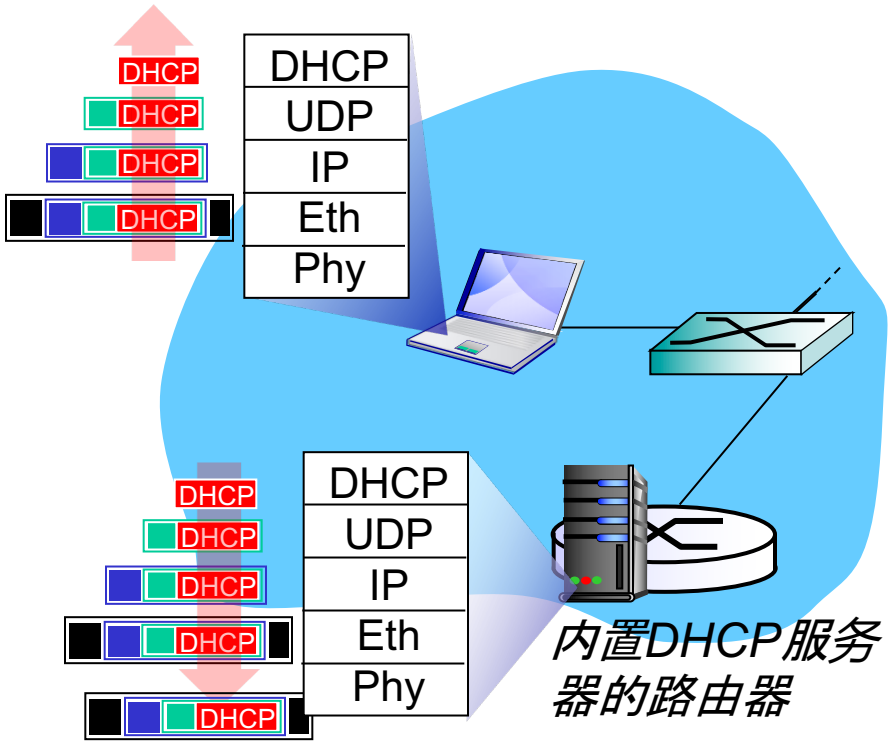
- 第一跳路由器地址（默认网关）
- DNS服务器的地址
- IP地址网络掩码

DHCP: example



- 一台新到达的主机需要其IP地址，第一跳路由器地址，DNS服务器地址：使用DHCP发现报文
- DHCP发现报文封装在UDP中，而后封装在IP中，然后封装在802.1以太网中
- 链路层将该帧广播到所有与该子网连接的节点。 广播的目的地址是255.255.255.255

DHCP: example



- DHCP服务器收到发现报文，用包含客户端IP地址、客户端第一跳路由器IP地址、DNS服务器名称和IP地址的DHCP提供报文进行响应，广播地址为255.255.255.255
- 客户端从一个或多个服务器的提供报文中选择一个，向该服务器用DHCP请求报文进行响应，回显配置的参数
- 服务器用DHCP ACK报文对DHCP请求报文进行响应，证实所要求的参数。

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."

怎样获取一个IP 地址?

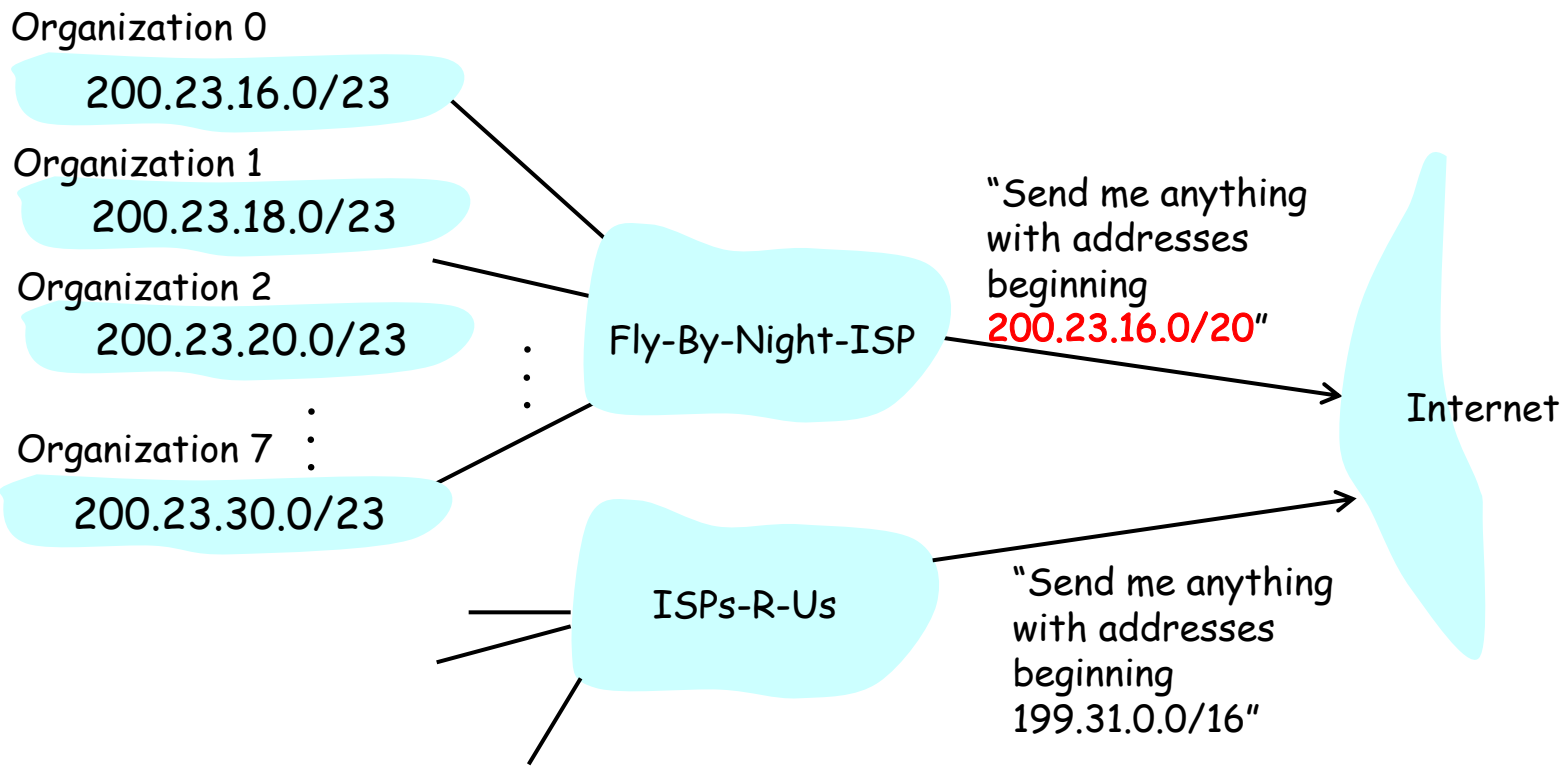
网络部分 (network portion):

- 从 ISP 的地址空间里得到被分配的部分

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

层次化编址: 路由聚合route aggregation

层次化编址允许高效率地向外通告(advertise)选路信息



- 200.13.16.0/23

  11001000.00010101.0001000|0.00000000
- 200.13.18.0/23

  11001000.00010101.0001001|0.00000000

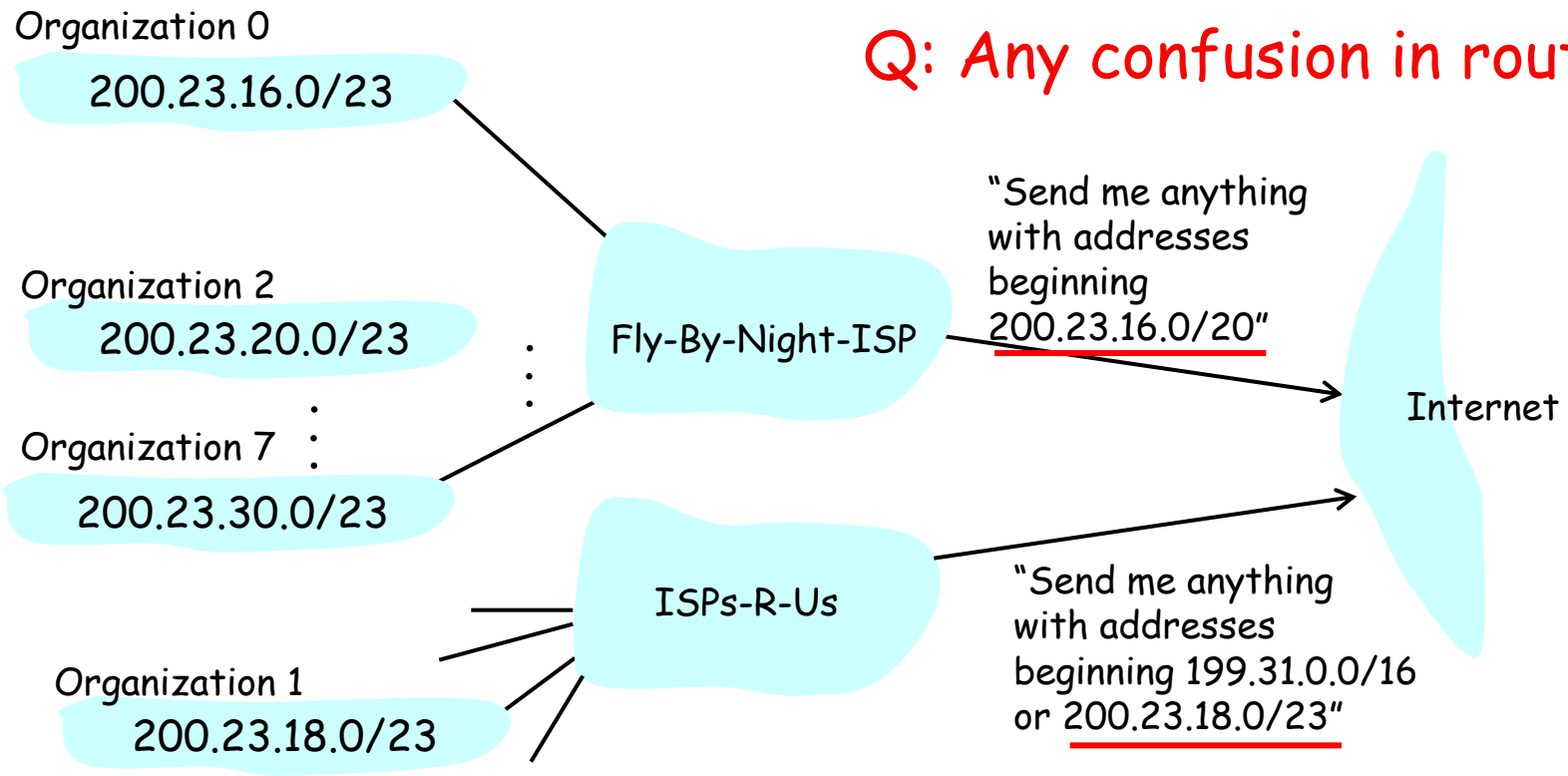
…

- 200.13.20.0/23

  11001000.00010101.0001111|0.00000000

层次化编址: 更多特别的路径选择

ISPs-R-Us 要加一条到 Organization 1 的特别路径



Q: Any confusion in routing?

最长前缀匹配

IP 编址: the last word...

Q: ISP如何得到大块的地址?

A: ICANN: Internet Corporation for Assigned Names and Numbers 因特网名字与号码分配团体，非营利组织

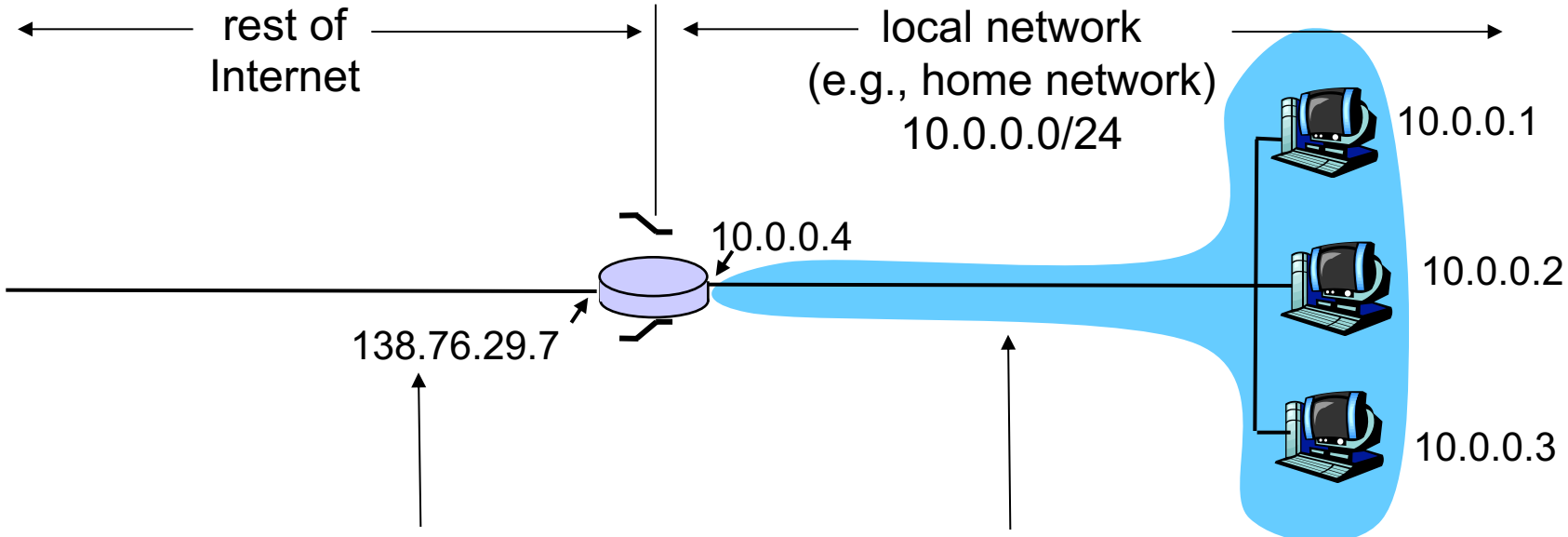
- 分配IP地址
- 管理 DNS
- 配置域名、调解争端

有特殊意义的 IP 地址

- 网络号或主机号全为"0"
  - 意味着“当前”，"this", “default", or “current"
  - 特定的某个网络 - Host Bits all “0" 154.3.0.0
  - 当前网络的某特定主机 - Network Bits all “0" 0.0.99.6
  - 当前主机 - “Me" 0.0.0.0
- 网络号或主机号全为"1"
  - 意味着“所有”,"all"
  - 特定网络的所有主机 - Host Bits all “1" 154.3.255.255
  - 广播到当前网络的所有主机 - all “1" 255.255.255.255
- 其他保留地址
  - 本地主机测试地址 Loopback address： 127.0.0.1
  - 其他保留地址：用来未来实验或管理因特网时内部使用

私有地址： 10.x.x.x, 172.16.x.x～172.31.x.x, 192.168.x.x

NAT (Network Address Translation) 网络地址翻译



所有离开本地网络的数据报
都有相同的NAT IP地址
138.76.29.7,
但有不同的源端口号

源或目的在此网络内的
数据报的源或目的地址
10.0.0.0/24

课后练习：观察宿舍无线路由所组成的网络，并查看其IP地址构成

NAT (Network Address Translation)

- 基本思想: 本地网络只使用一个IP地址和外部世界相连接
  - 不需要从ISP获得IP地址范围：所有本地计算机对外只用一个IP地址
  - 可以在本地（内部）网络内随便改变IP地址，不必通知外部世界
  - 可以改换ISP，而不必修改本地网络内的地址
  - 位于内网的计算机不能被外部世界直接寻址，不可见(增加了安全性)

NAT: network address translation

implementation: NAT router must:

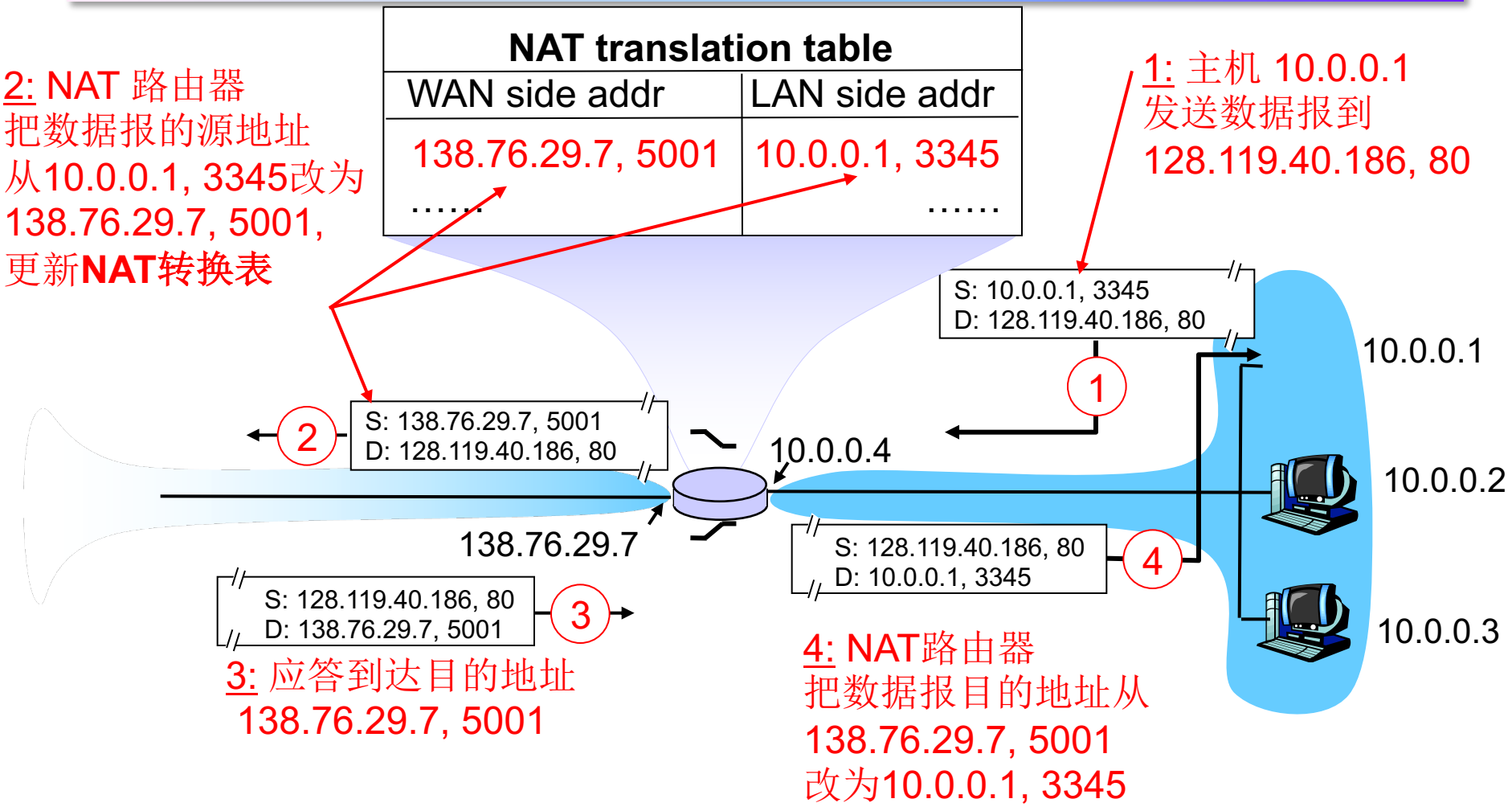
- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

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

- *remember (in NAT translation table)* 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 (Network Address Translation)

- 16位的端口号域
  - 通过一个有效IP地址可支持内部网中60,000多个并发连接!
- **关于NAT的争议**
  - 端口号应用于编址进程，而非编址主机
  - 路由器只应该处理到第3层，不应涉及运输层
  - 违反了end-to-end 约定
    - 中间节点不应介入修改IP地址和端口
    - NAT 可能性必须被应用开发者考虑, 例如 P2P 应用
  - IP地址短缺的问题应该由IPv6来解决

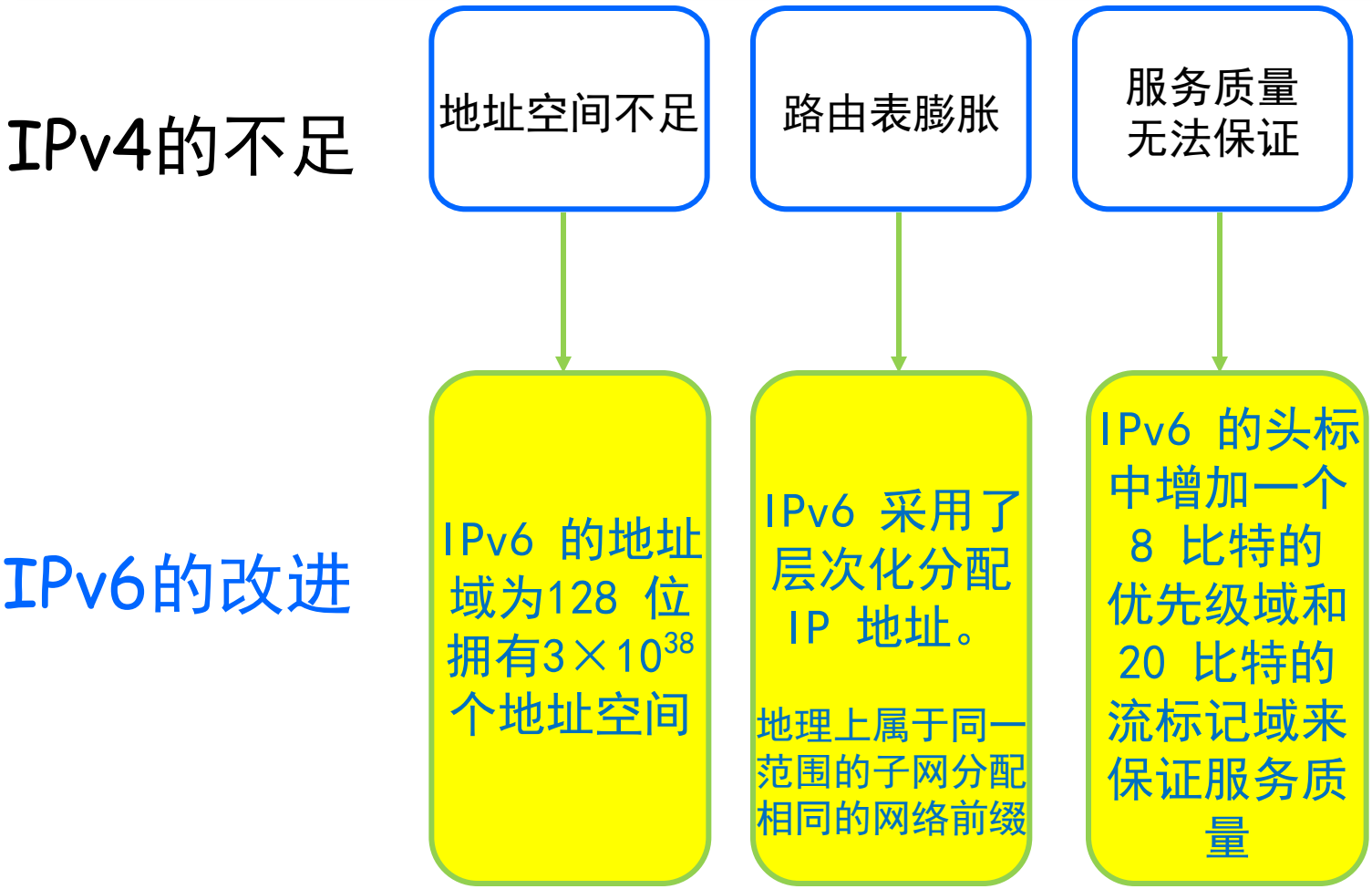
思考：内部网中的主机可能存在什么问题？

无有效IP地址，外界无法直接访问，无法充当服务器，…

IPv6

IPv4 面临的问题

IPv4 vs. IPv6



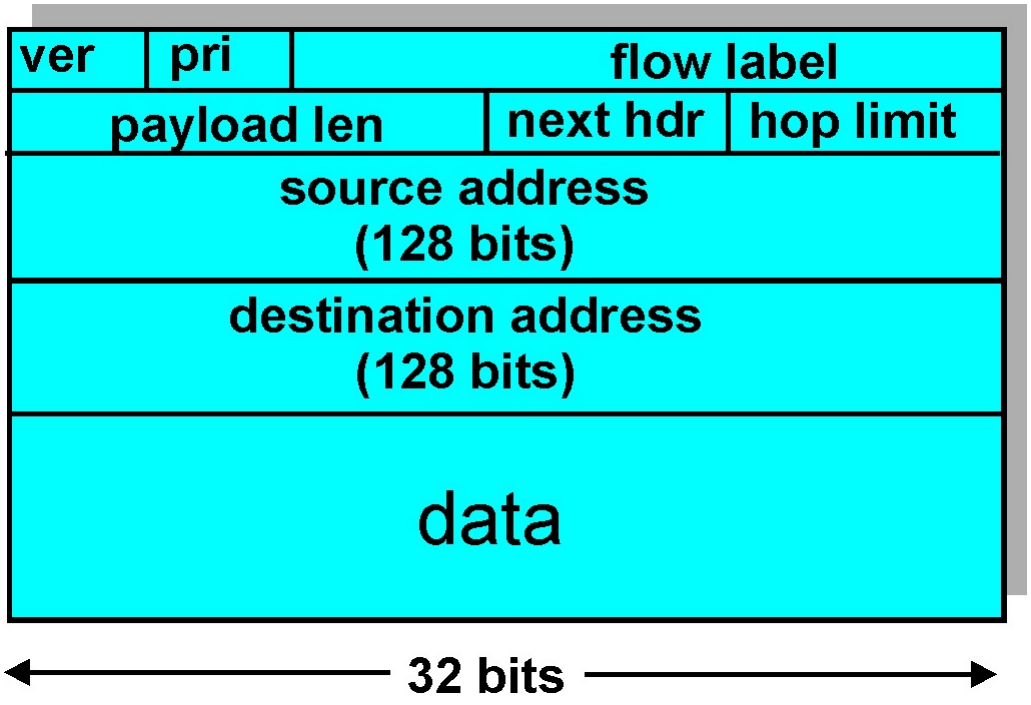
地球上的每一粒沙子都可以分到一个IP！

IPv6 报头 (固定40字节)

Priority 优先级(8bits) 区别数据流中的数据报优先级

Flow Label 流标签(20bits) 区别在同一"流"中的数据报，用于QoS 管理 ("流"的概念尚未明确定义)

Next header 下一个报头(8bits): 表明数据的上一层协议，等同IPv4中的协议字段



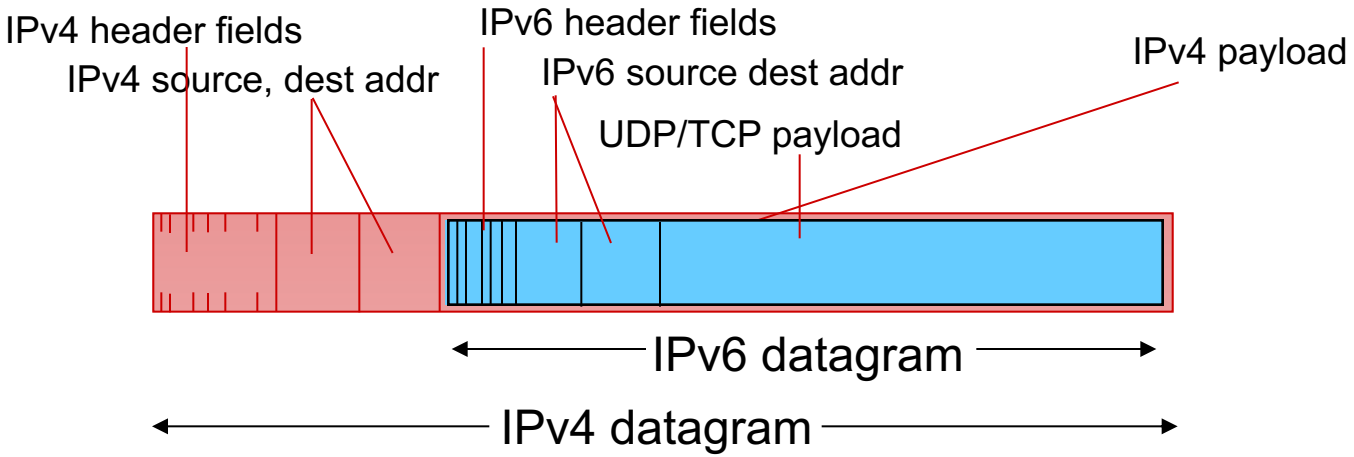
相比于IPv4的其他变化

- *没有分片机制*：加快路由器处理速度
- *Checksum*：完全去掉了，无需做检验和运算，缩短在每一跳的处理时间
- *Options*：允许，但在报头部分之外，由 "Next Header" 域表明，保证40字节的报头定长
- *ICMPv6*：新版的 ICMP
  - 更多的报文类型，e.g. "Packet Too Big"
  - 多播组管理功能

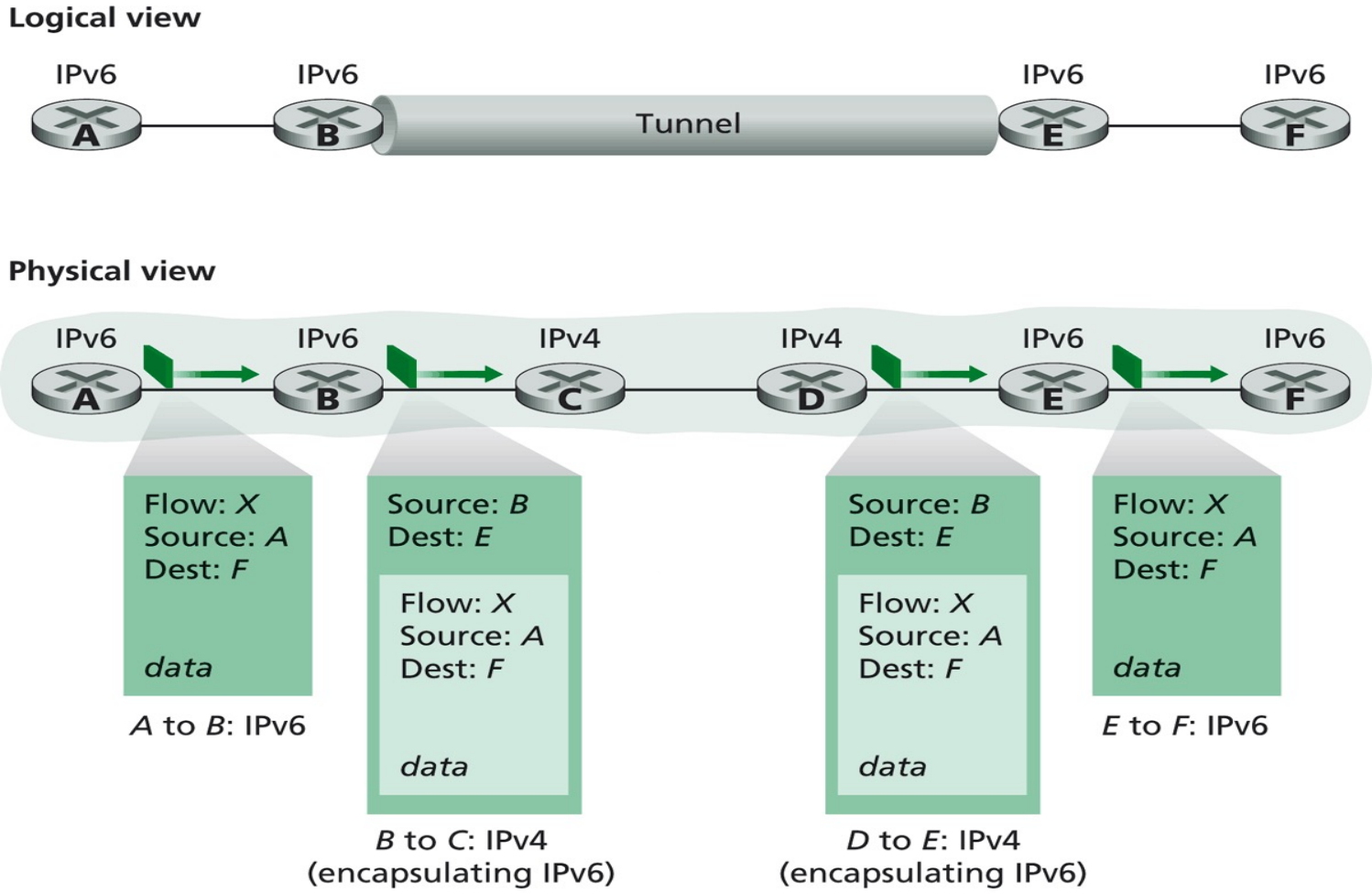
IPv6地址格式是X:X:X:X:X:X:X:X，其中X是一个4位十六进制整数(16位)，例如1030:0:0:0:c9b4:ff12:48aa:1a2b

从IPv4 到 IPv6 (RFC 2893)

- 并非所有路由器都可以同时升级
  - 没有“标志日”
  - IPv4和IPv6混合路由器的网络将如何运行？
- 隧道：IPv4路由器中IPv6数据包作为IPv4数据包的负载

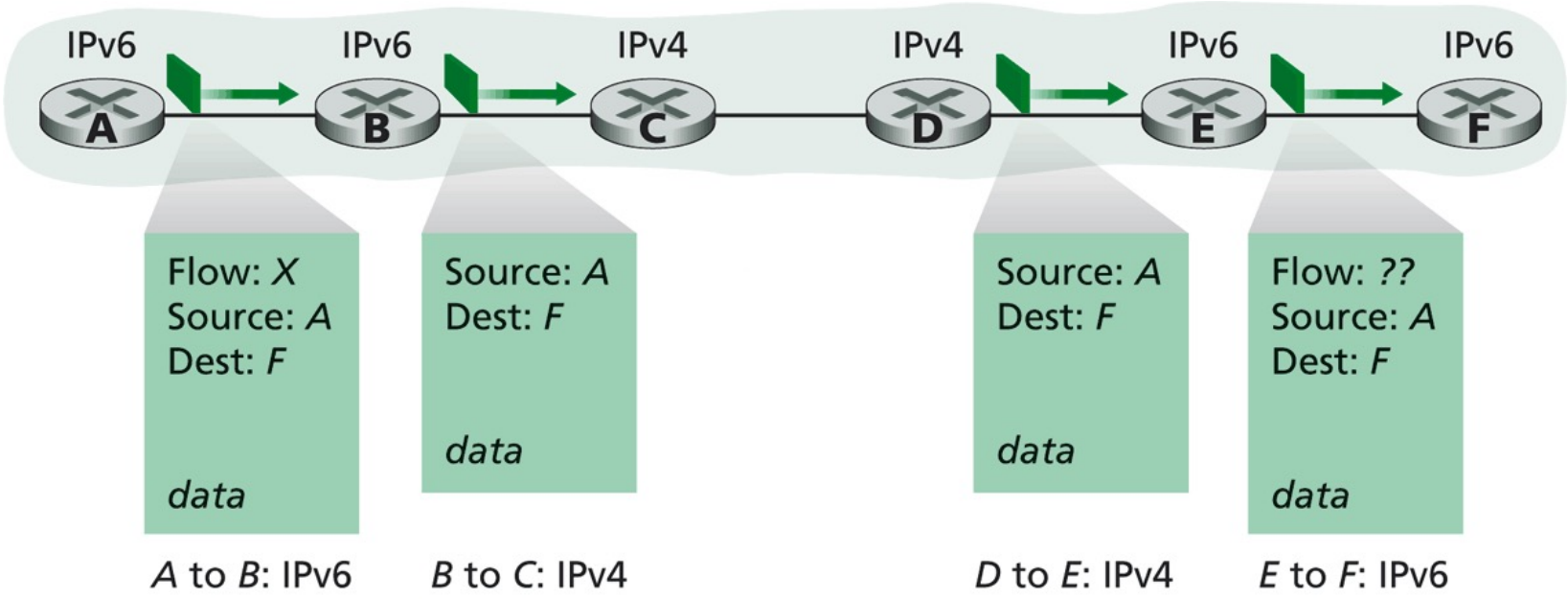


隧道 Tunnel


Logical view
IPv6
IPv6
IPv6
IPv6
A
B
Tunnel
E
F
Physical view
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
A
B
C
D
E
F
Flow: X
Source: A
Dest: F
data
A to B: IPv6
Source: B
Dest: E
Flow: X
Source: A
Dest: F
data
B to C: IPv4
(encapsulating IPv6)
Source: B
Dest: E
Flow: X
Source: A
Dest: F
data
D to E: IPv4
(encapsulating IPv6)
Flow: X
Source: A
Dest: F
data
E to F: IPv6

从IPv4 到 IPv6 (RFC 2893)

- 双栈 Double Stack



IPv6: adoption

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

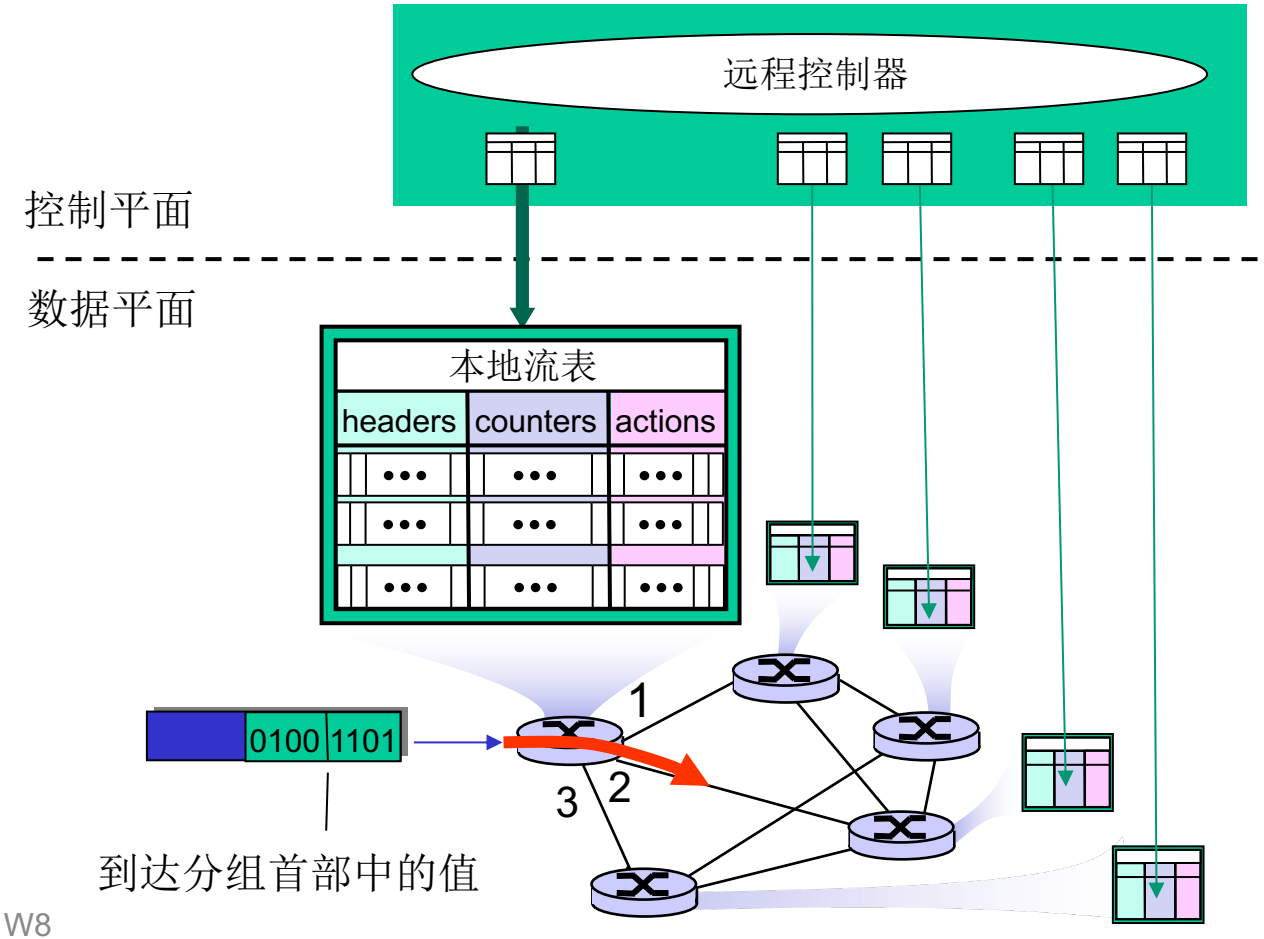
- *Long (long!) time for deployment, use*
  - 20 years and counting!
  - think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, …
  - *Why?*

提纲

- 概述
- 路由器工作原理
- 网际协议——IPv4,寻址,IPv6及其他
- **通用转发和SDN**

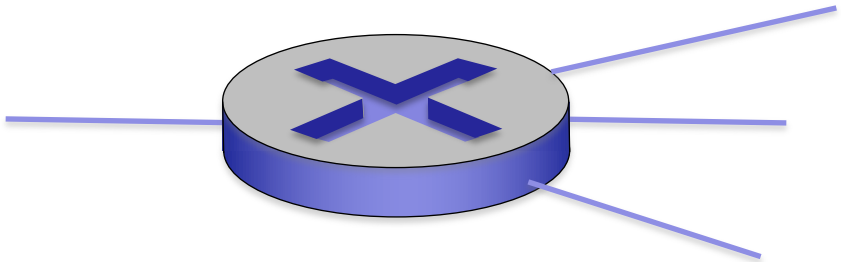
通用转发和SDN

每个“分组交换机”包含一个由远程控制器计算和分发的流表



OpenFlow数据平面抽象

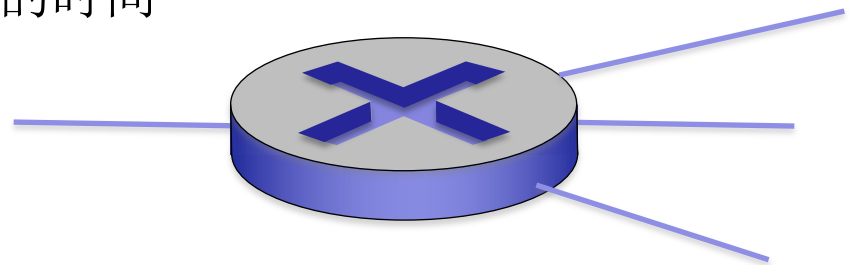
- 流表：匹配加动作转发表在OpenFlow中称为流表
- 通用转发：简单的包处理规则
  - Pattern：匹配数据包首部字段中的值
  - Actions：当分组匹配流表项时，丢弃、转发、修改或将匹配的数据包发送给控制器
  - Priority：消除重叠匹配的歧义
  - Counters（计数器）：已经与该表项匹配的分组数量，该表项上次更新以来的时间



分组交换机中的流表（由远程控制器计算分发）定义了交换机的匹配和动作规则

OpenFlow数据平面抽象

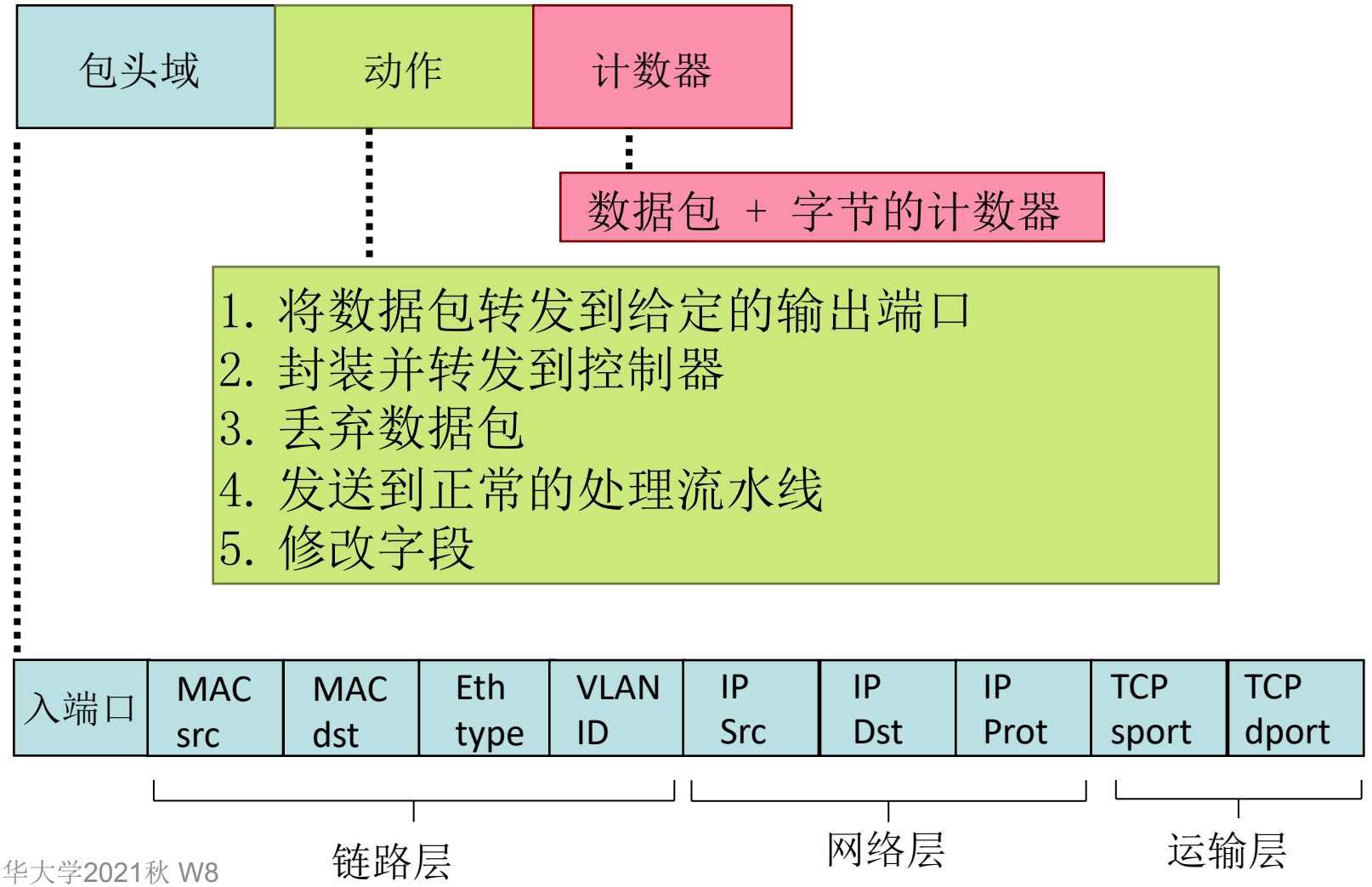
- 流表：匹配加动作转发表在OpenFlow中称为流表
- 通用转发：简单的包处理规则
  - Pattern：匹配数据包首部字段中的值
  - Actions：当分组匹配流表项时，丢弃、转发、修改或将匹配的数据包发送给控制器
  - Priority：消除重叠匹配的歧义
  - Counters（计数器）：已经与该表项匹配的分组数量，该表项上次更新以来的时间



*：通配符

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：流表表项



例子

基于IP地址（第三层）的转发：

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

将发往IP地址51.6.0.8的IP数据报转发到分组交换机输出端口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

不转发（阻止）所有发往TCP端口22的数据报

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

不转发（阻止）主机128.119.1.1发送的所有数据报

例子

基于MAC地址（第二层）的转发：

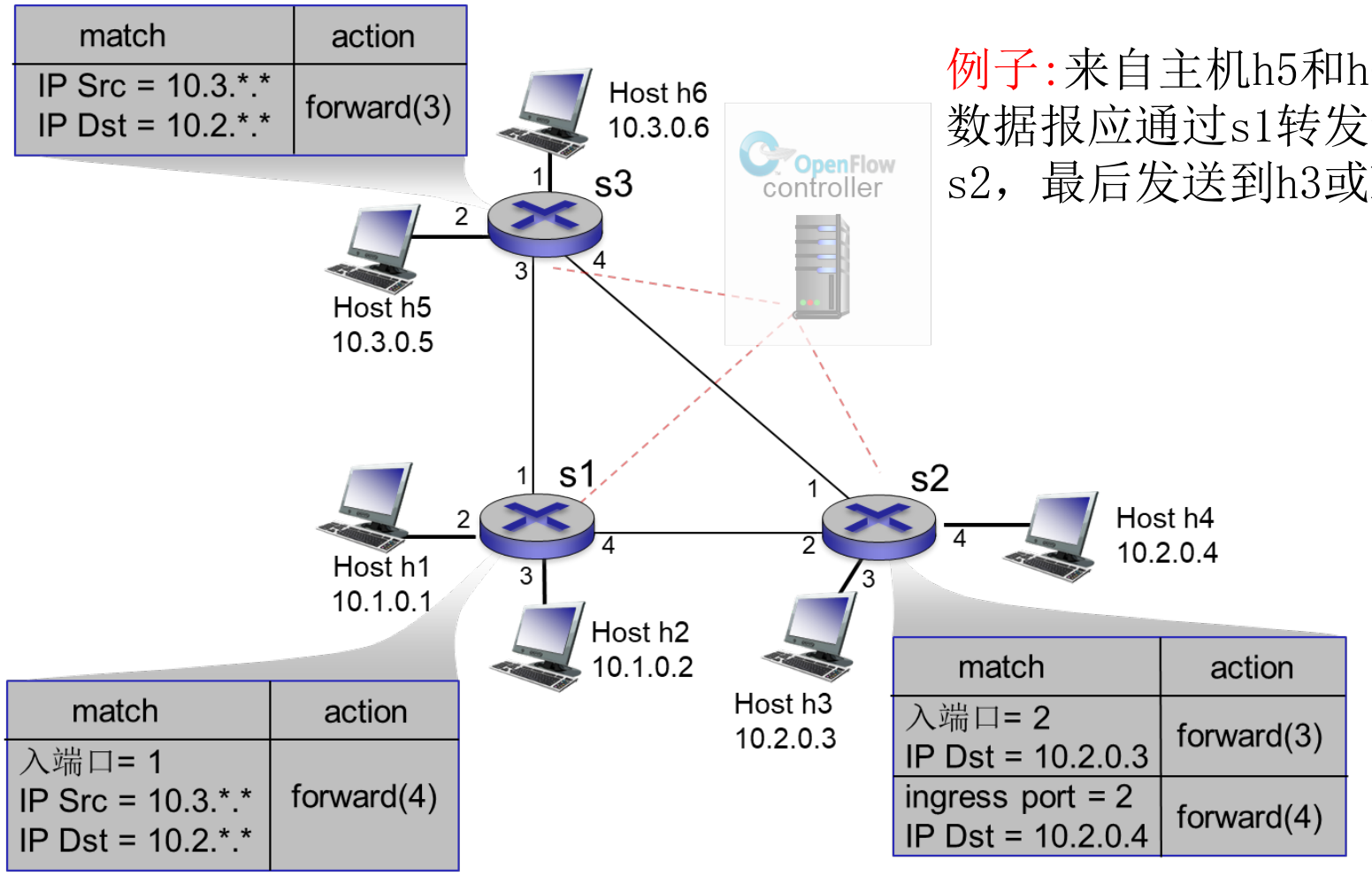
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23:11:E1:02	*	*	*	*	*	*	*	*	port3

将来自MAC地址22:A7:23:11:E1:02的第2层帧转发到输出端口6

OpenFlow抽象

- *match+action：* 统一不同类型的设备
  - 路由器
    - match：最长的目标IP地址的前缀
    - action：转发
  - 交换机
    - match：目标MAC地址
    - action：转发或者泛洪
  - 防火墙
    - match：IP地址和TCP/UDP端口号
    - action：通过或拒绝
  - NAT
    - match：IP地址和端口号
    - action：重写地址和端口号

OpenFlow例子



例子:来自主机h5和h6的数据报应通过s1转发给s2，最后发送到h3或h4