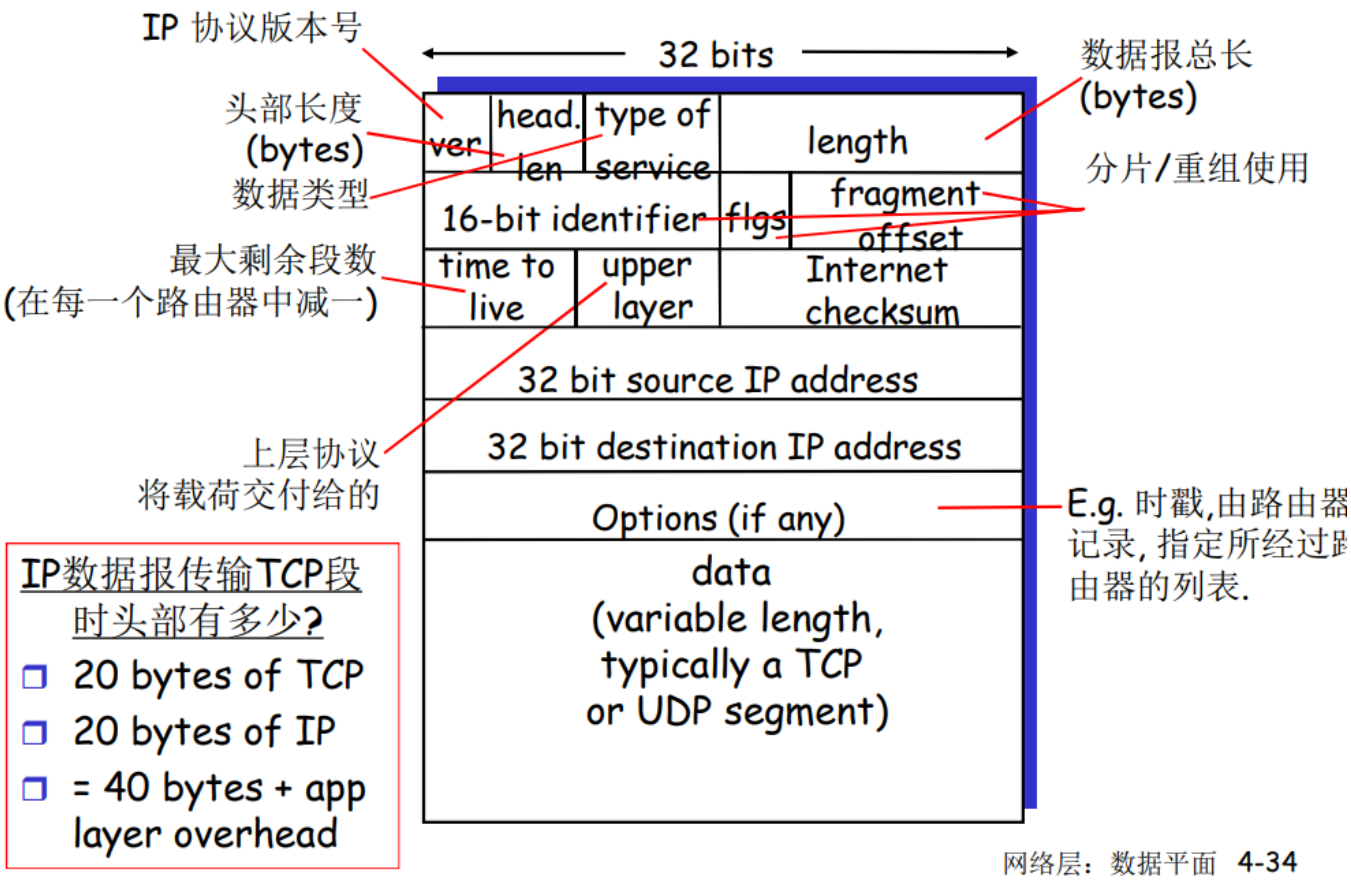
3.IP:Internet Protocol

```
3.1 网络层

    路由协议
        路径选择
        控制平面的路由功能 RIP,OSPF,BGP
    路由表
    IP协议
        实现数据平面的转发功能
        地址约定
        数据报格式
        分组处理的约定
    ICMP协议（信令协议）
        错误报告
        路由器信令

3.2 IP数据报格式


    头部：固定20字节
        version:            IP协议版本号          4 bytes
        head len:           头部长度              4个字节为单位，最少5个
        type of service:    TOS 数据类型          数据报载荷类型 基本上已经不用了
        length:             数据报总长

        16-bit identifier   ID表示                三个字段分片/重组用
        flgs                标志位
        fragment offset     偏移量

        time to live        最大剩余段数（每过一个路由器-1）
        upper layer         上层协议将载荷交付给的  （TCP、UDP或者其他上层实体）
        Internet checksum   校验和  （判断头部有没有坏）

        32 bit source IP address                源IP
        32 bit destination IP address           目标IP

        Options（if any）   E.g. 时间戳，由路由器记录，指定所经过路由器的列表
            头部还有一些option选项，所以头部是可变长的
                Options长度=length-head len

        每一行4个字节，一共5行，不包括option选项
    数据：payload 载荷
```

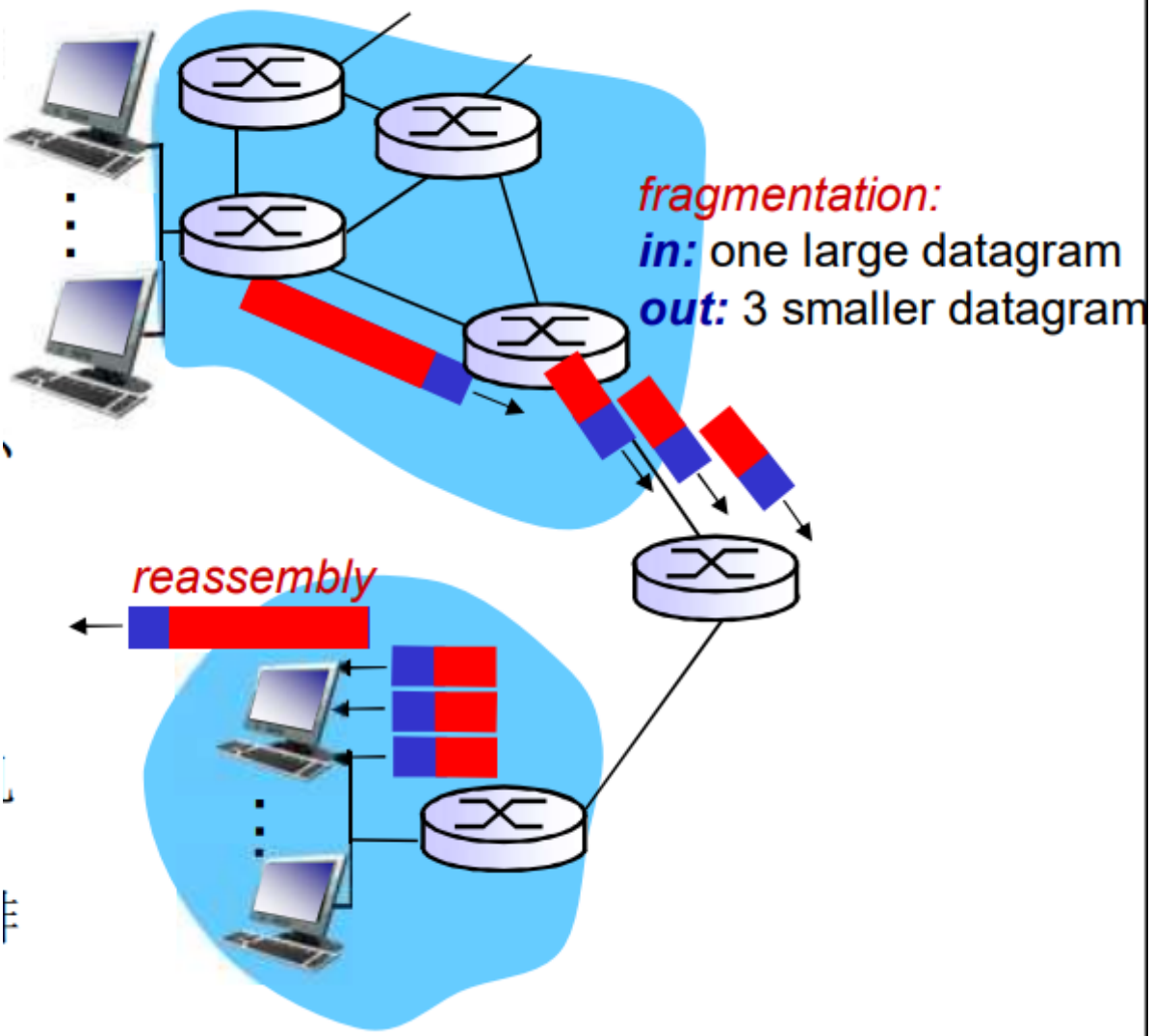
IP 数据报格式



3.3 IP 分片和重组（Fragmentation & Reassembly）

网络链路层有MTU(最大传输单元)-链路层帧所携带的最大数据长度
不同的链路类型
不同的MTU
大的IP数据报在网络上被分片（"fragmented"）
一个数据报被分割成若干个小的数据报
相同ID
不同的偏移量
最后一个分片标记为0
上面都是标志位，都在头部
"重组"只在最终的目标主机进行
为什么不在路由器重组？
路由器负担太重，没空
分片可能走不同路径
IP头部的信息被用于标识，排序相关分片

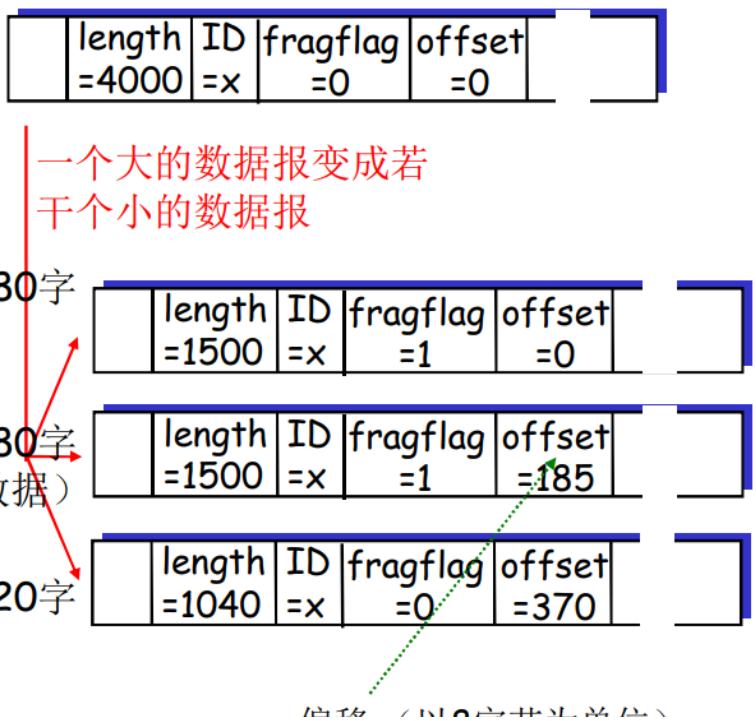
偏移量：颗粒度是8个字节，所以一般要除以8



IP 分片和重组

例子

- 4000 字节数据报
  - 20字节头部
  - 3980字节数据
- MTU = 1500 bytes
- 第一片：20字节头部+1480字节数据
  - 偏移量：0
- 第二片：20字节头部+1480字节数据（1480字节应用数据）
  - 偏移量：1480/8=185
- 第三片：20字节头部+1020字节数据（应用数据）
  - 偏移量：2960/8=370



偏移（以8字节为单位）=
1480/8

3.4 IP编址：引论 （IPv4地址）

IP地址：32位标示，对主机或者路由器的接口编址
接口：主机/路由器和物理链路的连接处
路由器通常拥有多个接口
主机也有可能有多个接口
IP地址和每一个接口关联
一个IP地址和一个接口相关联

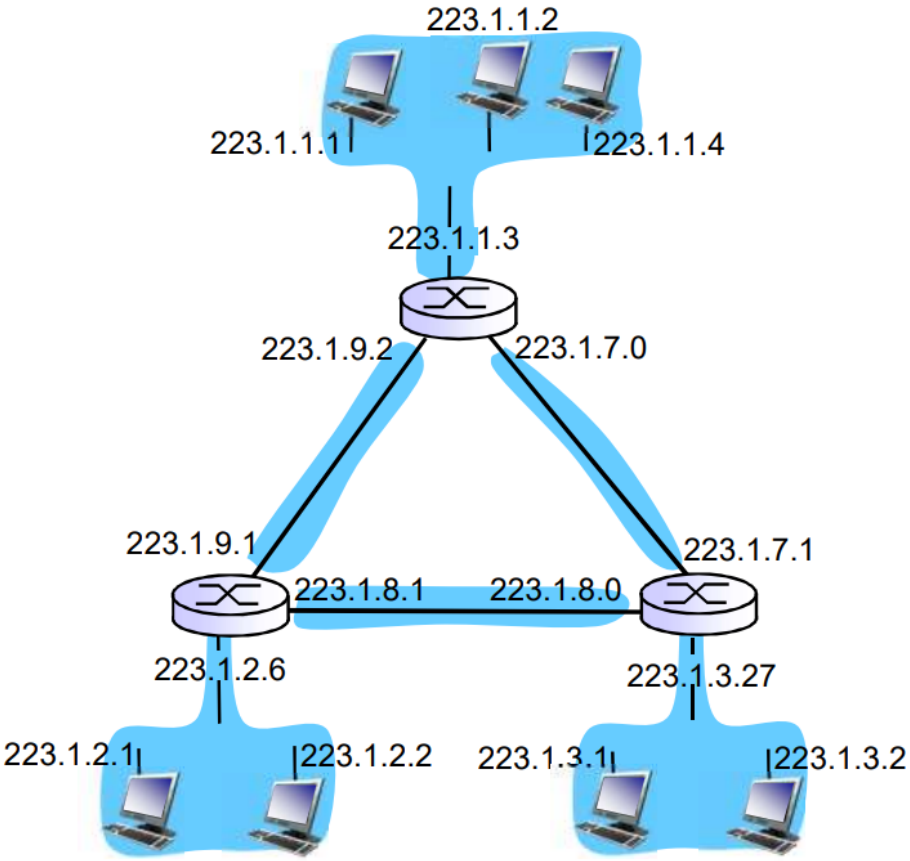
Q：这些接口是如何连接的
5，6章节会学习
A：有线以太网网口链接到以太网网络交换机连接
目前：无需担心一个接口是如何接到另一个接口的（中间没有路由器）

子网（Subnets）
IP地址：
子网部分（高位bits）
主机部分（地位bits）
什么是子网（subnet）
一个子网内的节点（主机或者路由器）他们的IP地址的高位部分相同，这些节点构成的网络的一部分叫子网

无需路由器介入，子网内各主机可以在物理上相互直接到达
一条可达，借助交换机
如何判断：
要判断一个子网，将每一个接口从主机或者路由器上分开，构成一个个网络的孤岛
每一个孤岛（网络）都是一个都可以被称之为subnet
子网掩码：

子网

几个?



6个

```
3.5 IP地址分类
    一共32位
    Class A:
            第一位为0，第一个字节的剩余7位为网络号，剩余24位为主机号
                    2^7-2                       2^24-2
          B:
            10开头，前面两个字节剩下14位为网络号，剩余16位为主机号
                    2^14-2                      2^16-2
          C:
            110开头，前3个字节为剩下的21位为网络号，剩余8位为主机号
                    2^21-2                      2^8-2
          D:
            1110开头，后面为multicast，组播地址
          E:
            预留
    为什么-2，默认全1全0网络号主机号无效
    单播地址：ABC
    组播地址：D

    特殊的IP地址：
        一些约定：
            子网部分：全为0 本网络
            主机部分：全为0 本网络
            主机部分：全为1 广播地址，这个网络的所有主机
            127.x.x.x    回路地址
                数据从传输层到网络层后直接反转回去。 回路地址又称为测试地址

    内网（专用）IP地址：
```

专用地址：地址空间的一部分供专用地址使用
永远不会被当作公用地址来分配，不会与公用地址重复
只在局部网络中有意义，区别不同的设备
路由器不对目标地址是专用地址的分组进行转发
专用地址范围
A类：10.0.0.0-10.255.255.255 MASK 255.0.0.0
B类：172.16.0.0-172.31.255.255 MASK 255.255.0.0
C类：192.168.0.0-192.168.255.255 MASK 255.255.255.0

IP地址在路由时，是以子网为单位进行散播子网可达信息，而不是IP为单位。
子网内是通过交换机一跳达成

CIDR:Classless InterDomain Routing 无类域间路由
子网部分可以在任意的位置
地址格式：a.b.c.d/x,其中x是地址中子网号的长度 一共还是32位

子网掩码：subnet mask
1：bit位置表示子网部分
0：bit位置表示主机部分
原始的A、B、C类网络的子网掩码分别是
A：255.0.0.0 11111111 00000000 00000000 00000000
B：255.255.0.0 11111111 11111111 00000000 00000000
C：255.255.255.0 11111111 11111111 11111111 00000000
DIDR:主要是看x占位情况
x占10位：11111111 11111111 11111100 00000000
另外的一种表示子网掩码的表达方式
/# （理解为/number number为子网位数）
例：/22，则表示前22位是子网部分

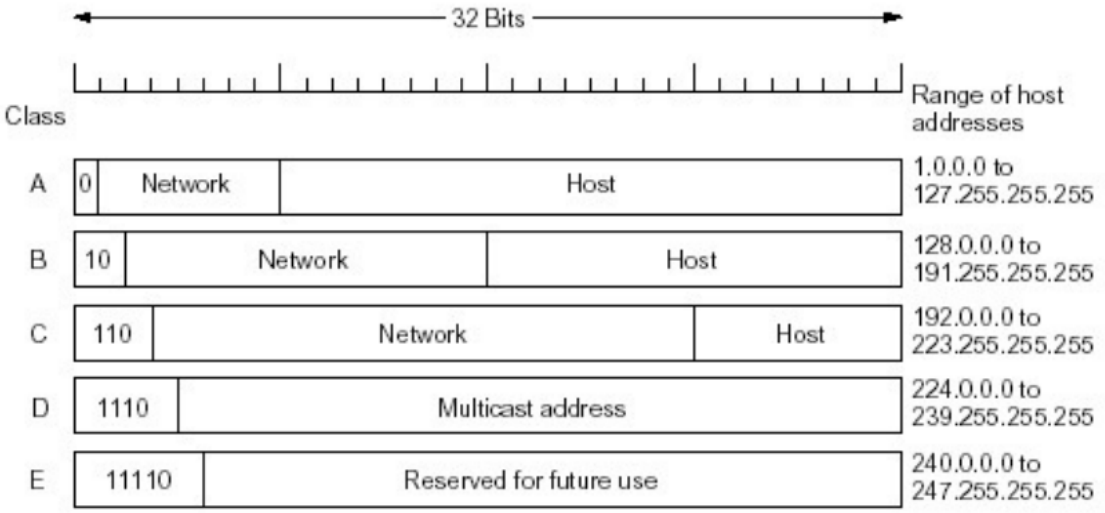
路由表和路由算法：

目标子网号	掩码	下一跳	端口
202.38.73.0	255.255.255.192	IPx	Lan1
202.38.64.0	255.255.255.192	IPy	Lan2
	.....		
Default	-	IPz	Lan0

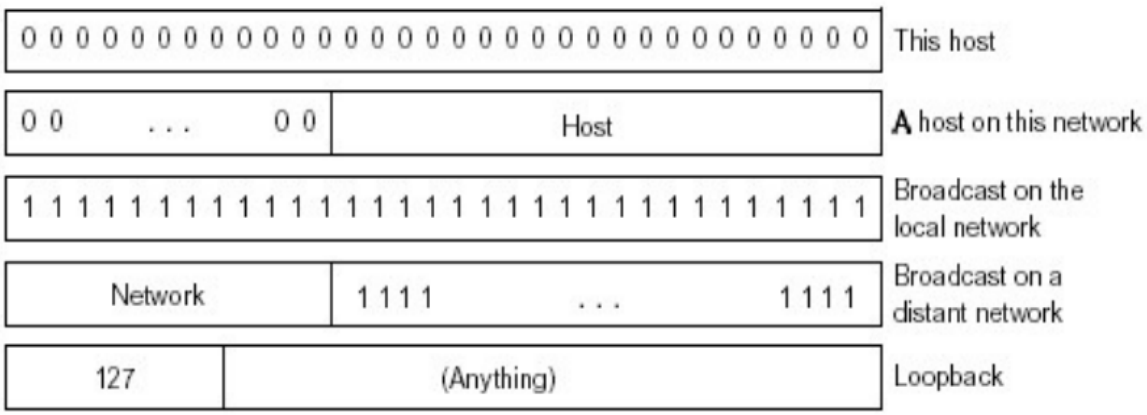
获得IP数据报的目标地址
对于转发表中的每一个表项
如（IP Des addr）目标IP地址 &（mask）子网掩码==destination，则按照表项对应的接口转发该数据报
做一个与运算，如果一样就说明匹配成功
如果都没有找到，则使用默认转发

IP 地址分类

- Class A：126 networks，16 million hosts
- Class B：16382networks，64 K hosts
- Class C：2 million networks，254 host
- Class D：multicast
- Class E：reserved for future



- 特殊IP地址



IP 编址: CIDR

CIDR: Classless InterDomain Routing

（无类域间路由）

- 子网部分可以在任意的位置
- 地址格式: a.b.c.d/x, 其中 x 是 地址中子网号的长度

200.23.16.0/23

子网掩码：11111111 11111111 11111110 00000000

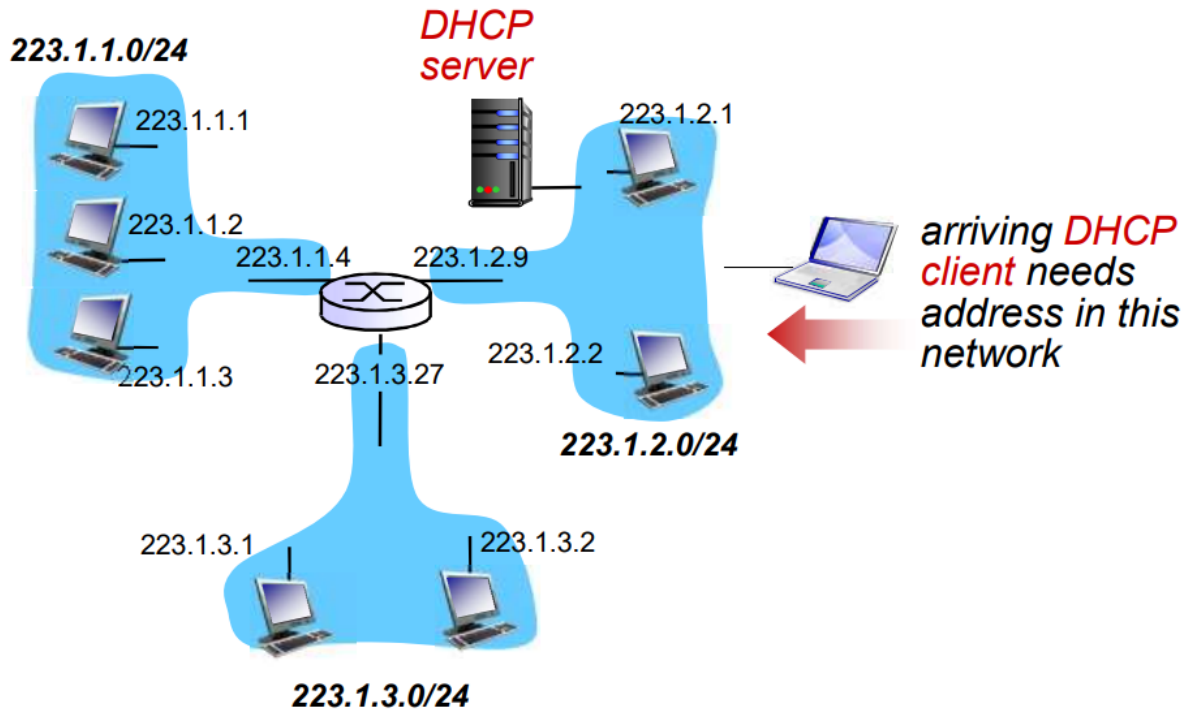
```
    3.6 如何获得IP地址 / DHCP

        主机如何获得一个IP地址：
            系统管理员将地址配置在一个文件中
                Wintel: control-panel ----> network ---->configuration---->TCP/IP---->properties
                UNIX:/etc/rc.config

                配置信息：IP地址，子网掩码（subnet mask），默认网关（default gateway），本地名字服务器
（local name server）
            DHCP:Dynamic Host Configuration Protocol
                从服务器中动态获得一个IP地址
                plug-and-play

        机构如何获得IP

            从ISP获得地址块中分配一个小地址块
                假设ISP的子网掩码为 /20，那么从剩下的12位主机为中拿出X位作为子网号
                举例：假设拿出3位，那么组织的子网掩码为/23

        ISP如何获得地址块
            ICANN:Internet Corporation for Assigned Names and Numbers
                分配地址
                管理DNS
                分配域名，解决冲突

        DHCP:   动态主机配置协议
            目标：
                允许主机在加入网络的时候，动态地从服务器哪里获得IP地址：
                    可以更新对主机在用IP地址地租用期-租期快到了
                    重新启动后，允许重新使用以前用过的IP地址
                    支持移动用户加入到该网络（短期在网）
            DHCP工作概况：
```

主播广播"DHCP discover" 报文 （可选）
此时还没有IP地址，使用32位全0本机地址
不知道DHCP在哪， 就用32位全1广播地址
DHCP服务器用"DHCP offer"提供报文相应（可选）
在UDP上的服务收到DHCP discover，予以回应
主机请求IP地址：发送 "DHCP request"报文
DHCP服务器发送地址："DHCP ack"报文

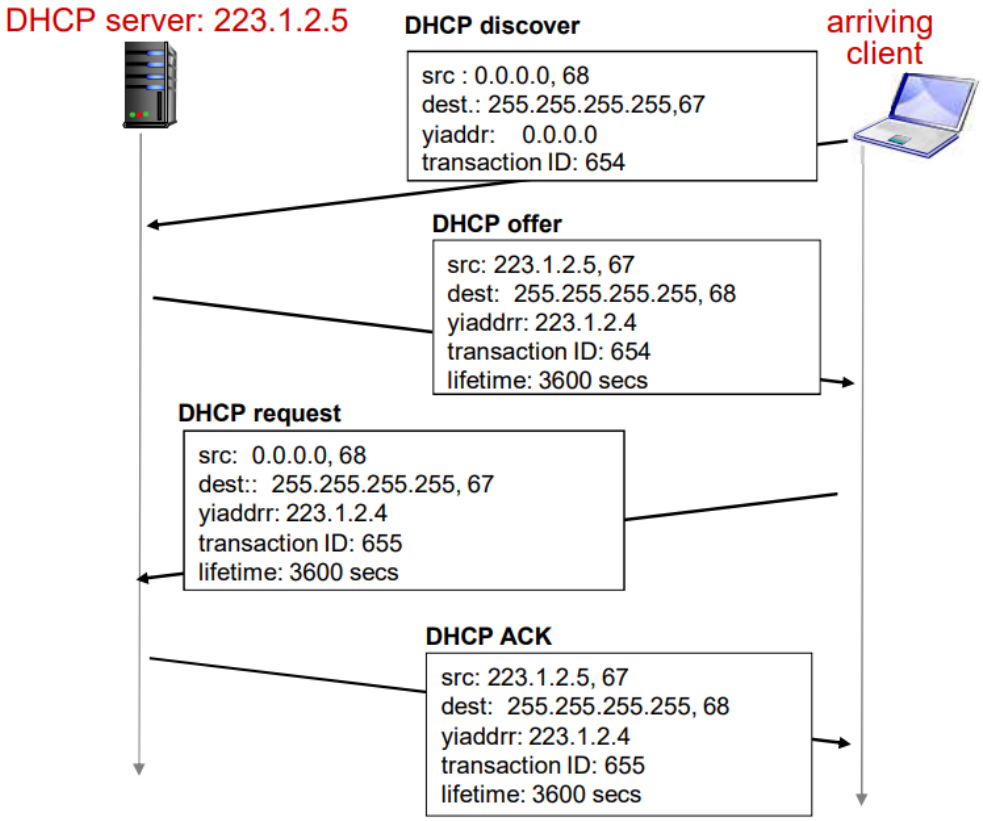
DHCP返回：
IP地址
第一跳路由器的IP地址（默认网关）
DNS服务器的域名和IP地址
子网掩码（指示地址部分的网络号和主机号）

实例：
如图

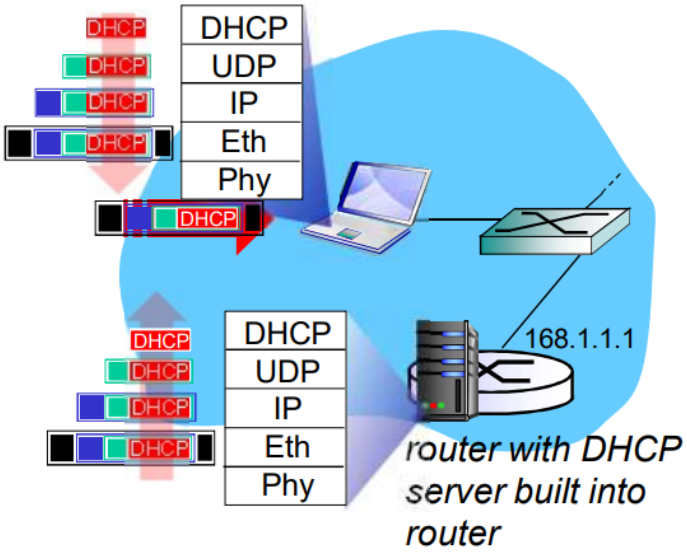
DHCP client-server scenario



DHCP client-server scenario

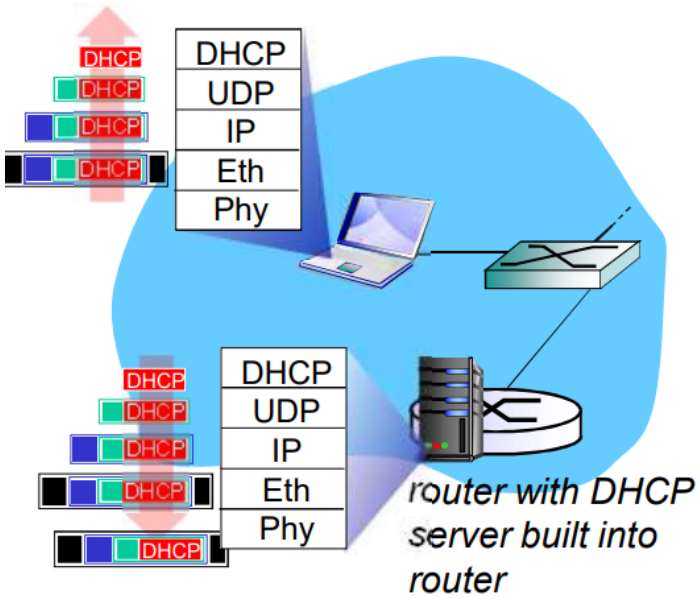


DHCP: 实例



- 联网笔记本需要获取自己的IP地址，第一跳路由器地址和DNS服务器：采用DHCP协议
- DHCP 请求被封装在UDP段中，封装在IP数据报中，封装在以太网的帧中
- 以太网帧在局域网范围内广播（dest：FFFFFFFFFFFF），被运行DHCP服务的路由器收到
- 以太网帧解封装成IP，IP解封装成UDP，解封装成DHCP

DHCP: 实例



- DHCP服务器生成DHCP ACK，包含客户端的IP地址，第一跳路由器的IP地址和DNS域名服务器的IP地址
- DHCP服务器封装的报文所在的帧转发到客户端，在客户端解封装成DHCP报文
- 客户端知道它自己的IP地址，DNS服务器的名字和IP地址，第一跳路由器的IP地址

3.7 路由信息的通告

层次编址：

路由聚集

允许路由信息的有效广播-->路由通告

告知上一层（）子网前缀是X的都转发给我，（感觉像是，建立子网的路由项）；对于上一层来说这个过程是路由聚集

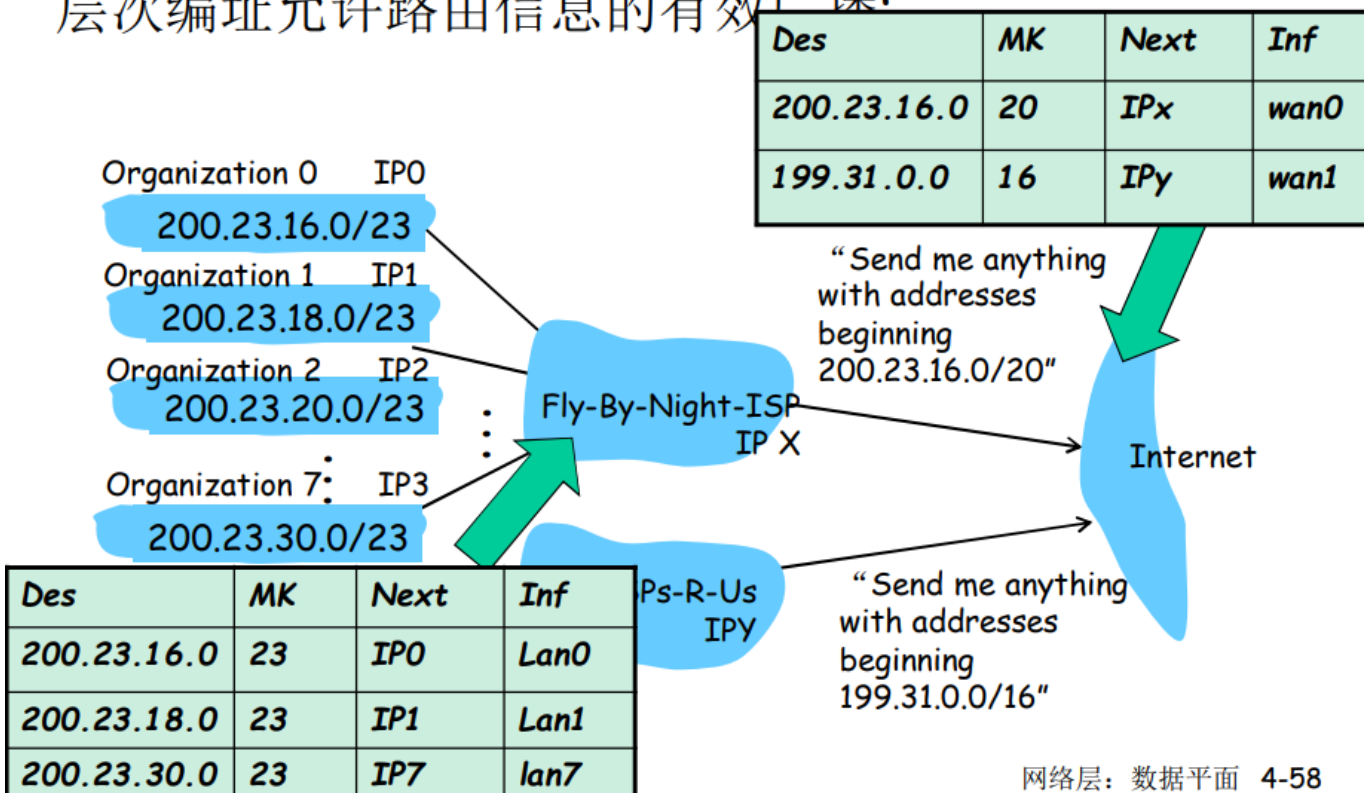
特殊路由信息

可以拥有更精确的路由

一张路由表项的子网掩码 位数 可以有好几种，即我可以有子网掩码为20位的，我也可以有23位的，匹配规则 最长前缀匹配（最长精确，减少通告数量）

层次编址: 路由聚集（route aggregation）

层次编址允许路由信息的有效广播:



Des	MK	Next	Inf
200.23.16.0	20	IPx	wan0
199.31.0.0	16	IPy	wan1

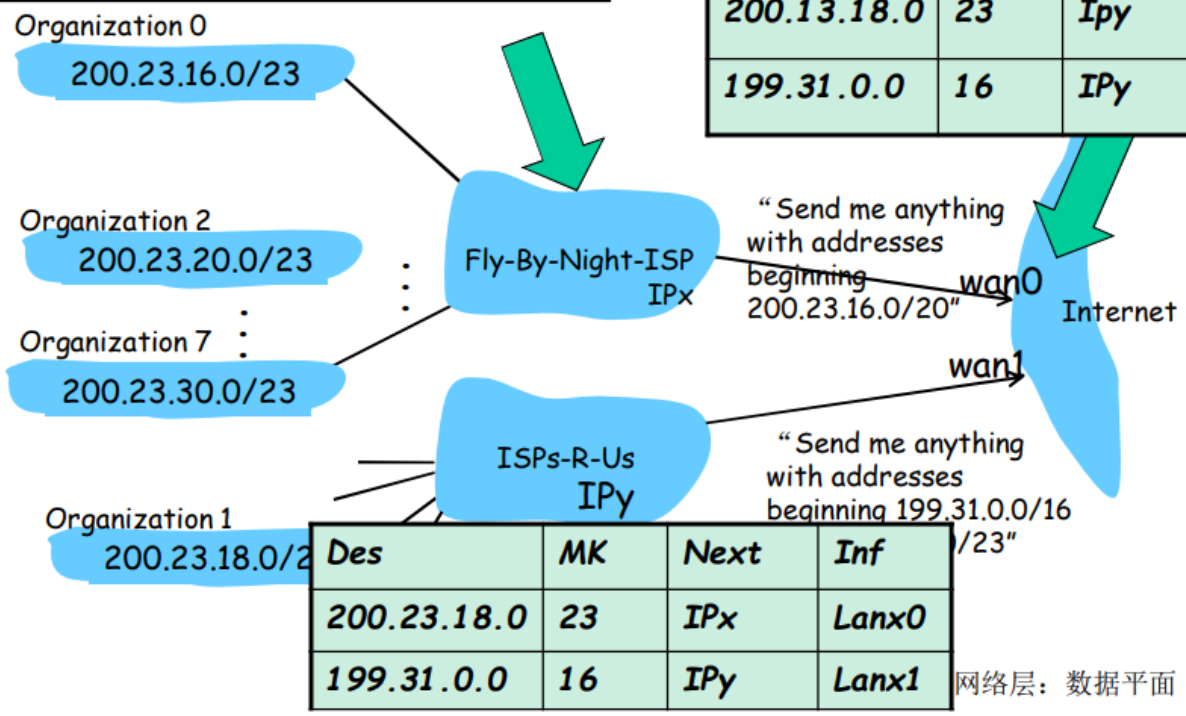
Des	MK	Next	Inf
200.23.16.0	23	IP0	Lan0
200.23.18.0	23	IP1	Lan1
200.23.30.0	23	IP7	lan7

层次编址: 特殊路由信息(more specific routes)

Des	MK	Next	Inf
200.23.16.0	23	IP0	Lan0
200.23.20.0	23	IP2	Lan1
200.23.30.0	23	IP7	lan7

精确的

Des	MK	Next	Inf
200.23.16.0	20	IPx	wan0
200.13.18.0	23	Ipy	wan1
199.31.0.0	16	IPy	wan1



Des	MK	Next	Inf
200.23.18.0	23	IPx	Lanx0
199.31.0.0	16	IPy	Lanx1

3.8 网络地址转换

所有离开本地网络的数据报具有一个相同的源地址(NAT IP addressl),但是具有不同的端口号

动机：
- 本地网络只有一个有效的IP地址
  - 不需要从ISP分配一块地址，可以用一个IP地址用于所有的（局域网）设备--省钱
  - 可以在局域网改变设备的地址情况下而无须通知外界
  - 可以改变ISP（地址变化）而不需要改变内部的设备地址
  - 局域网内部的设备没有明确的地址，对外是不可以见的---安全

实现：
- NAT路由器必须：
  - 外出数据报：替换 源地址和端口号 为 NAT IP 地址和新的端口号，目标IP和端口不变
    - 远端的C/S将会用NAP IP地址，新端口号作为目标地址
  - 记住：每个转换替换对（在NAT转换表中）
    - 源IP，端口 vs NAP IP ,新端口
  - 进入数据包：替换目标IP地址和端口号，采用存在NAT表中的mapping表项，用（源IP，端口）

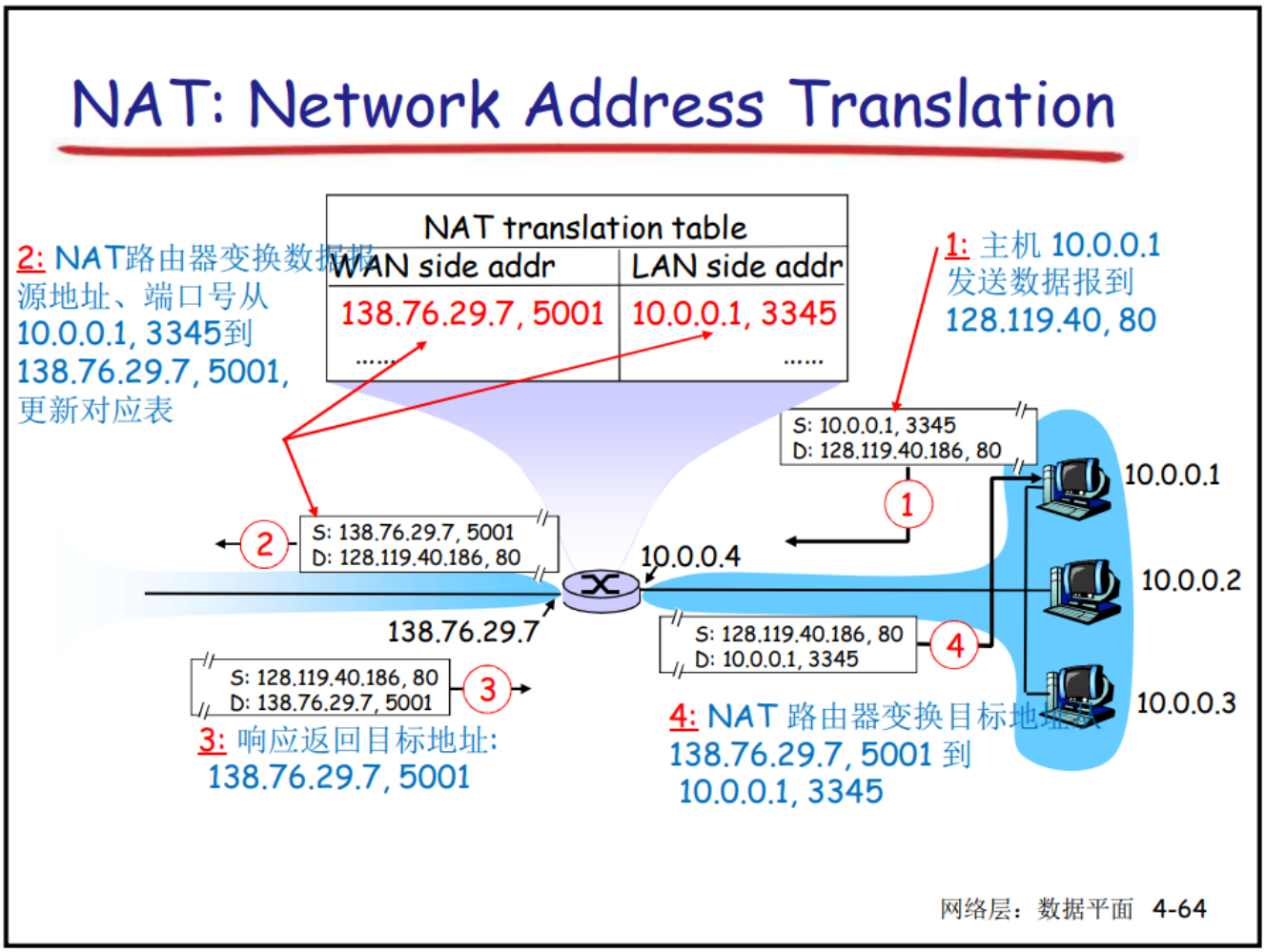
16-bit端口字段：
- 6w多同时连接，一个局域网

对NAT是有争议的
- 路由器只应该对第3层做信息处理，而这里对端口号（4层）作了处理
- 违反了end-to-end原则
  - 端到端原则：复杂性放到网络边缘
    - 无需借助中转和变换，就可以直接传送到目标主机
  - NAT可能要被一些应用设计者考虑
    - P2P
  - 外网的机器无法主动连接到内网的机器上
- 地址短缺问题可以被IPv6解决

NAT穿越：如果客户端需要连接在NAT后面的服务器，如何操作
- 客户端需要连接地址为10.0.0.1的服务器
  - 服务器地址10.0.0.1 LAN本地地址（客户端不能够使用其作为目标地址）
  - 整网只有一个外部可见地址：
    - 138.76.29.7

- 方案1：
  - 静态配置NAT:转发进来的对服务器特定端口的连接请求
    - e.g. (123.76.29.7,port 2500)总是转发到10.0.0.1 port 25000
- 方案2：
  - Universal Plug and Play(UPnP) Internet Gateway Device(IGD)协议.允许NATted主机可以：
    - 获知网络的公共IP地址为：138.76.29.7
    - 列举存在的端口映射
    - 增/删端口映射（在租用时间内）

  - 自动化静态NAT端口映射配置
- 方案3：
  - 中继(used in Skype)
    - NAT后面的服务器建立和中继的连接
    - 外部的客户端链接到中继
    - 中继在2个连接之前的桥梁



3.9 IPv6

动机：

初始动机：

32-bit地址空间将会被很快用完

另外的动机：

头部格式改变帮助加速处理和转发：

TTL-1

头部checksum

分片

头部格式改变帮助QoS

数据报格式：

固定的40字节头部

数据报传输过程中不允许分片

头部：Cont

Priority：标示流中数据表的优先级

Flow Label：标示数据报在一个"flow"（"flow"的概念没有被严格定义）

Next hreader：标示上层协议

TLV模式：Type length value 字解释

与IPv4的其他变化：

Checksum：被移除，降低在每一段中的处理速度

Options：允许，但是在头部之外，被"Next Header"标示

ICMPv6：ICMP的新版本

附加了报文类型， e.g. "Packet Too Big"

多播组管理功能

从IPv4到IPv6的平移（平滑升级）

不是所有的路由器都能够同时升级的

没有一个标记日"flag days"

在IPv4和IPv6路由器混合时，网络如何运转

隧道：

在IPv4路由器之前的IPv4数据报写道IPv6数据报

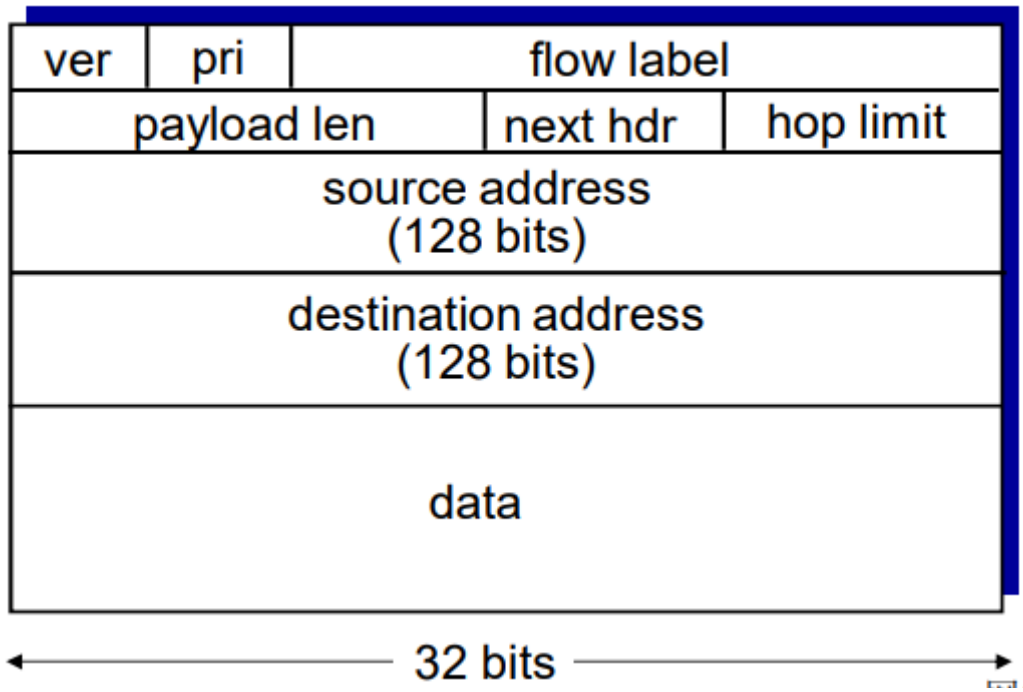
应用：

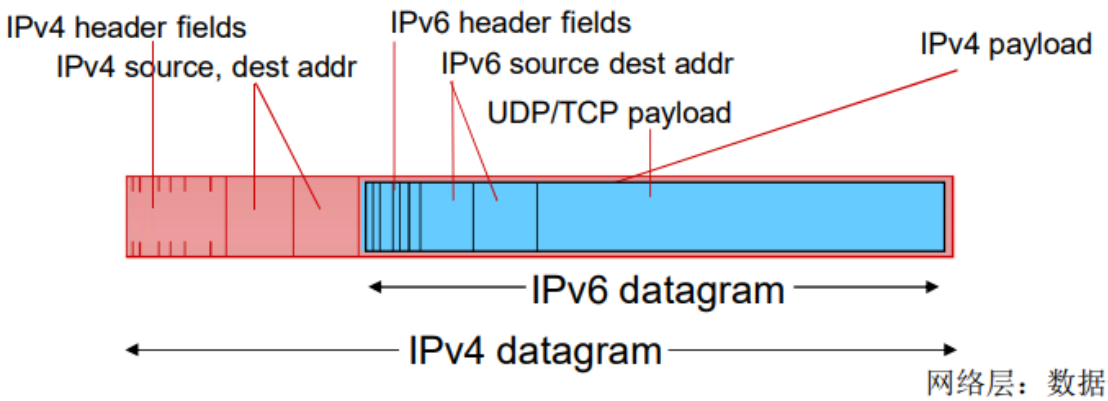
google：8%的客户通过IPv6访问谷歌服务

NIST:全美国1/3的政府域支持IPv6

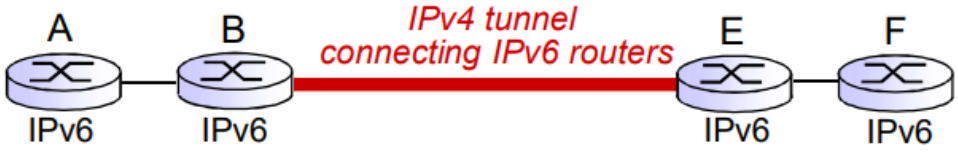
估计还需要很长一段时间部署

20年以上

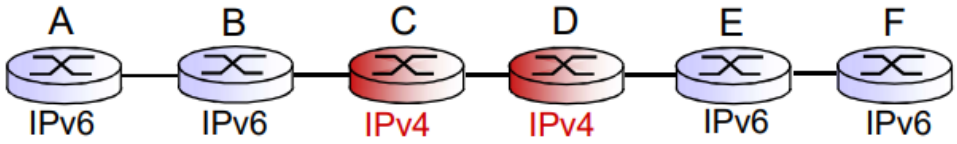




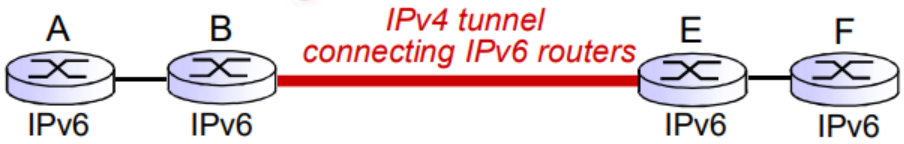
逻辑视图：



物理视图：



逻辑视图：



物理视图：