

## ● IPv4 数据报格式

- 首部长度的单位是4字节
- 总长度：受MTU限制
- 标识
- 标志
- 片偏移：单位是8字节
- 首部校验和：采用16位二进制反码求和算法



## ● IPv4 地址

- 32位
- 两级结构
- 五类地址：A、B、C、D、E
- 常用的环回地址：127.0.0.1



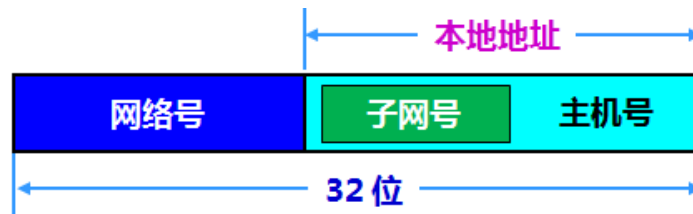
## ● 子网划分

- ✓ IPv4地址：网络号+子网号+主机号

### ✓ 子网掩码

- 确定子网的网络地址和主机标识
- 判断两台通信的主机是否在同一网络中
- 决定子网中主机的数目

- ✓ 相同的IP地址和不同的子网掩码可以得到相同的网络地址。



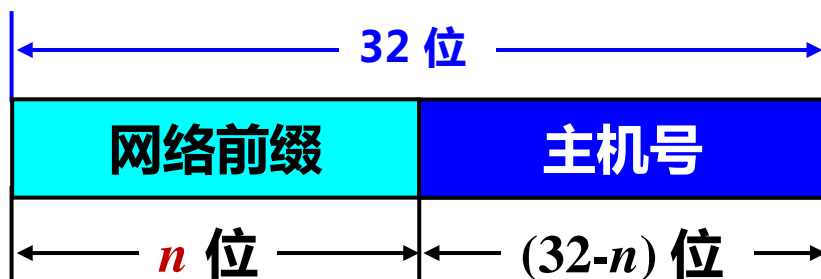
### 无分类域间选路（CIDR）

- 1987 年，RFC 1009 就指明了在一个划分子网的网络中可同时使用几个**不同的子网掩码**。
- 使用**变长子网掩码（VLSM**，Variable Length Subnet Mask)可进一步提高 IP 地址资源的**利用率**。
- 在 VLSM 的基础上又进一步研究出**无分类编址方法**，其正式名字是**无分类域间选路（CIDR**，Classless Inter-Domain Routing)。
- “无分类”的意思是现在的选路决策是基于整个32位IP地址的掩码操作，而不管其IP地址是A类、B类或C类，这样能够将路由表中的许多表项归并成更少的数目。**基本思想**是取消IP地址的分类结构，取而代之的是允许以可变长分界的方式分配网络数。
- **要点**
  - **网络前缀**
  - **地址块**
  - **地址掩码**

## 无分类域间选路 (CIDR) -- 网络前缀

- 无分类域间选路 CIDR 使用各种长度的“网络前缀” (network-prefix) 来代替分类地址中的网络号和子网号。
- IP 地址从三级编址 (使用子网掩码) 又回到了两级编址。
- 无分类的两级编址记法

IP地址 ::= {<网络前缀>, <主机号>}



两级结构

网络前缀的位数  $n$  不固定，  
可以在0~32之间选取任意值。

## 无分类域间选路 (CIDR) -- 网络前缀

- CIDR 使用 “斜线记法” (CIDR 记法)

a.b.c.d / n : 二进制 IP 地址的前 n 位是网络前缀。

例如：IP地址128.14.46.34/20，其网络号为128.14.32.0，表示前20bit为网络前缀，后面12bit为主机号。

10000000 00001110 00101110 00100010

20bit的网络前缀      12bit的主机号

- 10.0.0.0/10 可简写为10/10，即把点分十进制中低位连续的0省略。

10.0.0.0/10隐含地指出IP地址10.0.0.0的掩码是255.192.0.0。此掩码可表示为：

11111111 11000000 00000000 00000000

掩码中有 10 个连续的 1      255      192      0      0

- 网络前缀的后面加一个星号 \* 的表示方法，如 00001010 00\*，在星号 \* 之前是网络前缀，而星号 \* 表示 IP 地址中的主机号，可以是任意值。

### 无分类域间选路（CIDR）-- 地址块

- CIDR 把网络前缀都相同的连续的 IP 地址组成 “**CIDR 地址块**”。
- 一个 CIDR 地址块包含的IP地址数目取决于网络前缀的位数。

例如，128.14.32.0/20 表示的地址块**共有  $2^{12}$  个地址**

最小地址 →

所有地址  
的 20 位  
前缀都是  
一样的

最大地址 →

```

10000000 00001110 00100000 00000000
10000000 00001110 00100000 00000001
10000000 00001110 00100000 00000010
10000000 00001110 00100000 00000011
10000000 00001110 00100000 00000100
10000000 00001110 00100000 00000101
...
10000000 00001110 00101111 11111011
10000000 00001110 00101111 11111100
10000000 00001110 00101111 11111101
10000000 00001110 00101111 11111110
10000000 00001110 00101111 11111111
    
```

128.14.32.0  
(全0地址)

可指派的地址  
数是  $2^{12} - 2$  个

128.14.47.255  
(全1地址)

在不需要指出地址块的起始地址时，也可将这样的地址块简称为 “**/20 地址块**”。

### 无分类域间选路（CIDR）-- 地址块

#### 常用的CIDR地址块

CIDR 前缀长度	点分十进制	包含的地址数	相当于包含分类的网络数
/13	255.248.0.0	512 K	8个B类或2048个C类
/14	255.252.0.0	256 K	4个B类或1024个C类
/15	255.254.0.0	128 K	2个B类或512个C类
/16	255.255.0.0	64 K	1个B类或256个C类
/17	255.255.128.0	32 K	128个C类
/18	255.255.192.0	16 K	64个C类
/19	255.255.224.0	8 K	32个C类
/20	255.255.240.0	4 K	16个C类
/21	255.255.248.0	2 K	8个C类
/22	255.255.252.0	1 K	4个C类
/23	255.255.254.0	512	2个C类
/24	255.255.255.0	256	1个C类
/25	255.255.255.128	128	1/2个C类
/26	255.255.255.192	64	1/4个C类
/27	255.255.255.224	32	1/8个C类

### 无分类域间选路（CIDR）-- 地址块

特殊的地址块	说 明
网络前缀 $n=0$	全 0 IP 地址，即 0.0.0.0/0。用于默认路由。
网络前缀 $n=31$	这个地址块中只有两个 IP 地址，主机号分别为 0 和 1。这个地址块用于点对点链路。
网络前缀 $n=32$	32 位 IP 地址都是前缀，没有主机号。就是一个 IP 地址，用于主机路由。

### 无分类域间选路（CIDR）

- 一个 CIDR 地址块可以表示很多地址，这种地址的聚合常称为**路由聚合（构成超网）**，它使得路由表中的一个项目可以表示**很多个**原来传统分类地址的路由。
- 路由聚合**减少**了路由器之间的**路由选择信息的交换**，提高了整个互联网的性能。
- **注意**：使用CIDR技术汇聚的网络地址的比特位必须一致。
- **CIDR可以看作是子网划分的逆过程**
  - 使用CIDR技术聚合地址时，网络前缀缩短。网络前缀越短，其地址块所包含的地址数就越多。
  - 划分子网是使网络前缀变长。
- CIDR的一个**好处**是可以更加有效地分配IPv4的地址空间，可以**根据客户需要分配**适当大小的CIDR地址块。



## 无分类域间选路 (CIDR)

- 使用 CIDR 时，路由表中的每个项目由“网络前缀”和“下一跳地址”组成。在查找路由表时可能会得到不止一个匹配结果，应从匹配结果中选择具有最长网络前缀的路由（最长前缀匹配），网络前缀越长，其地址块就越小，路由就越具体。最长前缀匹配又称为最长匹配或最佳匹配。

例

收到的分组的目的地址  $D = 206.0.71.130$ 

路由表中的项目：

206.0.68.0/22	1
206.0.71.128/25	2

查找路由表中的第 1 个项目：

第 1 个项目 206.0.68.0/22 的掩码  $M$  有 22 个连续的 1。 $M = 11111111\ 11111111\ 11111100\ 00000000$ 因此只需把  $D$  的第 3 个字节转换成二进制。

	$M =$	11111111	11111111	11111100	00000000
AND	$D =$	206.	0.	01000111.	130
		206.	0.	01000100.	0

与 206.0.68.0/22 匹配!

## 无分类域间选路 (CIDR)

- 使用 CIDR 时, 路由表中的每个项目由“网络前缀”和“下一跳地址”组成。在查找路由表时可能会得到不止一个匹配结果, 应从匹配结果中选择具有最长网络前缀的路由 (最长前缀匹配), 网络前缀越长, 其地址块就越小, 路由就越具体。最长前缀匹配又称为最长匹配或最佳匹配。

例

收到的分组的目的地址  $D = 206.0.71.130$ 路由表中的项目 :  
206.0.68.0/22      1  
206.0.71.128/25    2

查找路由表中的第 2 个项目 :

第 2 个项目 206.0.71.128/25 的掩码  $M$  有 25 个连续的 1。 $M = 11111111\ 11111111\ 11111111\ 10000000$ 因此只需把  $D$  的第 4 个字节转换成二进制。

$M =$	11111111	11111111	11111111	10000000
AND $D =$	206.	0.	71.	10000010
	206.	0.	71.	10000000

与 206.0.71.128/25 匹配!

## 无分类域间选路 (CIDR)

- 使用 CIDR 时，路由表中的每个项目由“网络前缀”和“下一跳地址”组成。在查找路由表时可能会得到不止一个匹配结果，应从匹配结果中选择具有最长网络前缀的路由（最长前缀匹配），网络前缀越长，其地址块就越小，路由就越具体。最长前缀匹配又称为最长匹配或最佳匹配。

例

$D \text{ AND } (11111111 \ 11111111 \ 11111100 \ 00000000)$   
= 206.0.68.0/22      匹配

$D \text{ AND } (11111111 \ 11111111 \ 11111111 \ 10000000)$   
= 206.0.71.128/25      匹配



选择两个匹配的地址中更具体的一个，  
即选择最长前缀的地址。

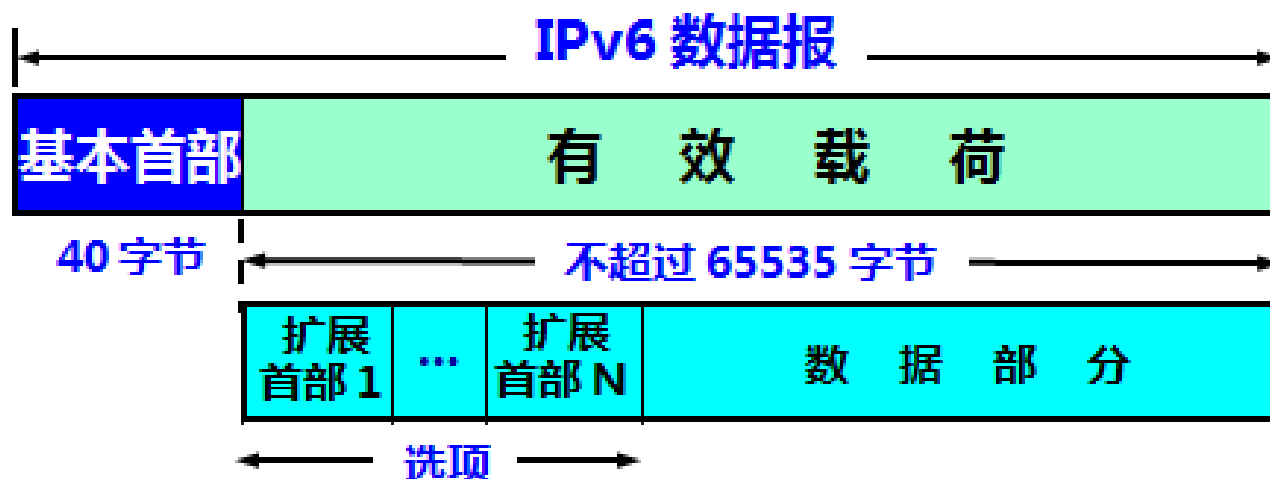
### IPv6

- IPv6 仍支持**无连接的数据报传送**。
- **主要变化如下**：
  - **更大的地址空间**。将地址从 IPv4 的 32 位 增大到了 128 位。
  - **扩展的地址层次结构**。可以划分为更多的层次。
  - **灵活的首部格式**。定义了许多可选的扩展首部。
  - **改进的选项**。允许数据报包含有选项的控制信息，其选项放在有效载荷中。
  - **支持即插即用（即自动配置）**。不需要使用 DHCP。
  - **支持资源的预分配**。支持实时视像等要求，保证一定的带宽和时延的应用。
  - **IPv6 首部改为 8 字节对齐**。首部长度必须是 8 字节的整数倍。原来的 IPv4 首部是 4 字节对齐。
  - **允许协议继续扩充**。更好地适应新的应用。

## IPv6 数据报格式

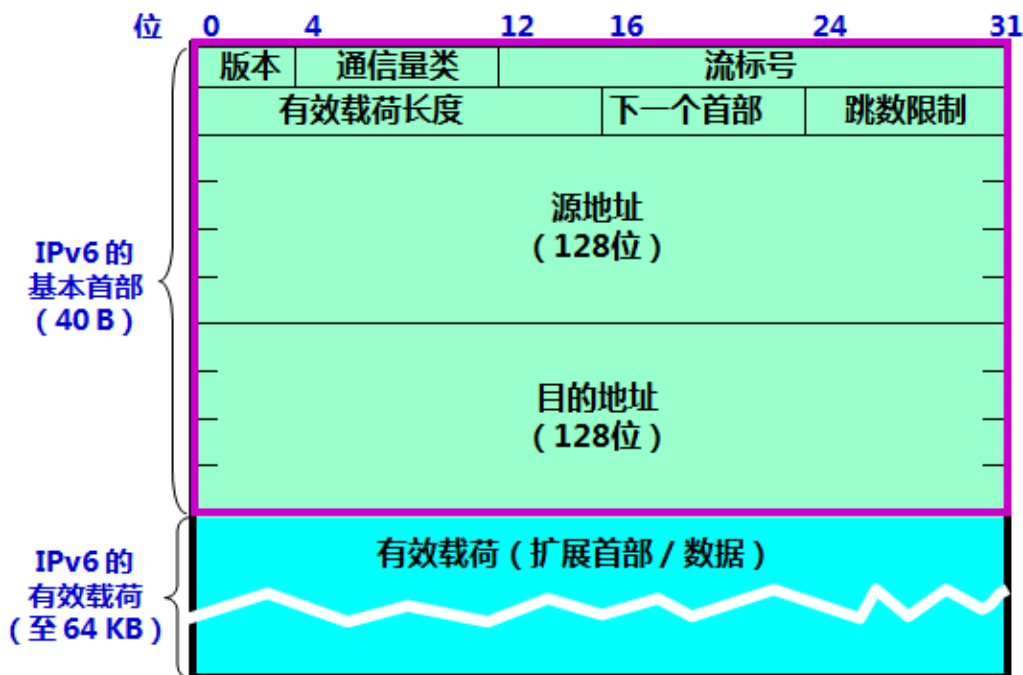
由两部分组成：

- 基本首部
- 有效载荷，也称为净负荷。有效载荷允许有零个或多个**扩展首部**，再后面是数据部分。



## IPv6 数据报格式

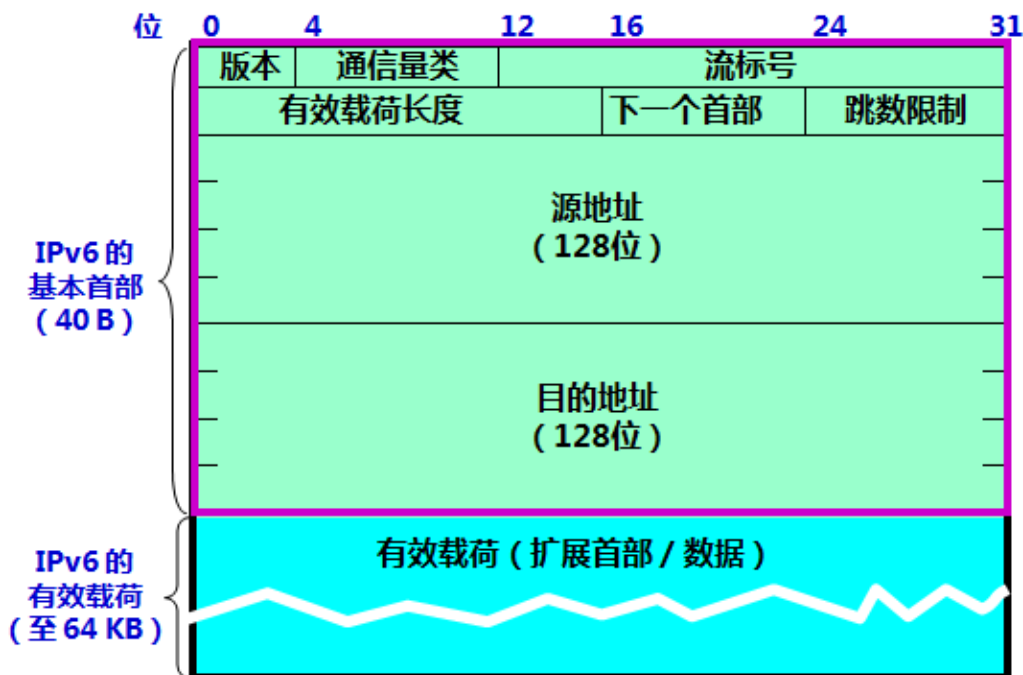
- 首部长度：**固定的 40 字节**，称为**基本首部**。
- 首部字段数：只有 8 个。



- 取消了首部长度字段，因为首部长度是固定的 40 字节；
- 取消了区分服务字段；
- 取消了总长度字段，改用有效载荷长度字段；
- 取消了标识、标志和片偏移字段，这些功能已经包含在分片扩展首部中。
- 把 TTL 字段改称为跳数限制字段；
- 取消了协议字段，改用下一个首部字段；
- 取消了校验和字段；
- 取消了选项字段，而用扩展首部来实现选项功能。

## IPv6 数据报格式

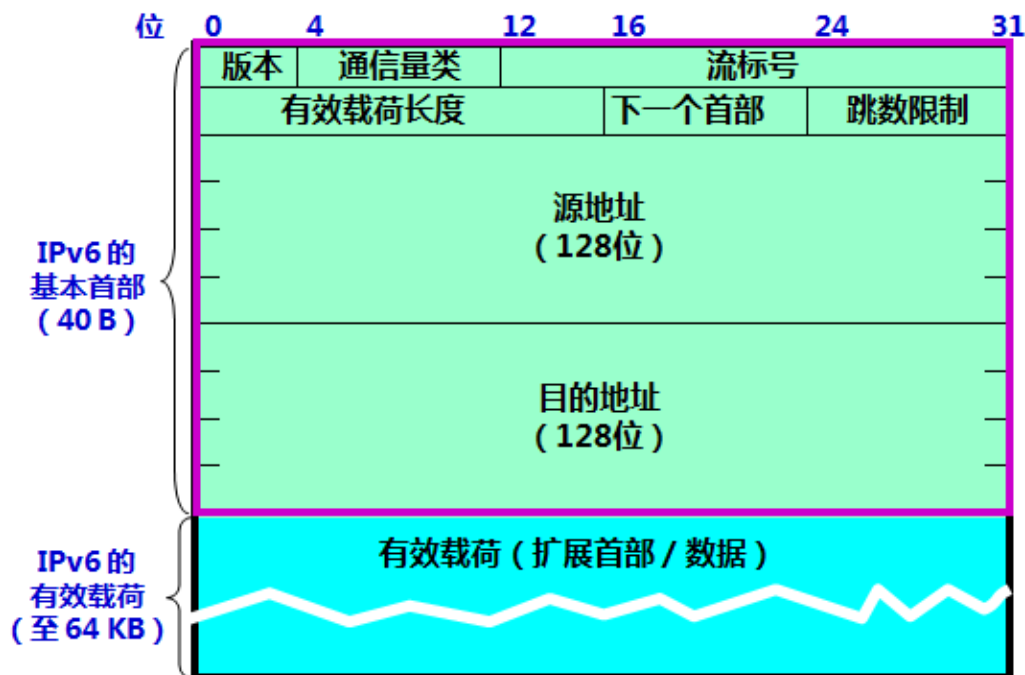
- 首部长度：固定的 40 字节，称为基本首部。
- 首部字段数：只有 8 个。



- 通信量类：8位，用于区分不同的 IPv6 数据报的类别或优先级。
- 流标号：20位，“流”是互联网络上从特定源点到特定终点的一系列数据报，“流”所经过的路径上的路由器都保证指明的服务质量。所有属于同一个流的数据报都具有同样的流标号。资源预分配时使用，对实时音频/视频数据的传送特别有用。
- 下一个首部：8位，相当于 IPv4 的协议字段或选项字段。

## IPv6 数据报格式

- 首部长度：固定的 40 字节，称为基本首部。
- 首部字段数：只有 8 个。



RFC 2460 定义了六种**扩展首部**

- 逐跳选项（逐跳扩展报头）
- 路由选择（路由扩展报头）
- 分片（分片扩展报头）
- 鉴别（认证扩展报头）
- 封装安全有效载荷（加密安全复杂扩展报头）
- 目的站选项（目标扩展报头）



### IPv6 地址

- IPv6 使用**冒号十六进制记法**(colon hexadecimal notation, 简称为 colon hex)。每个 16 位的值用十六进制值表示，各值之间用**冒号**分隔。例如：

0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000

FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF

- 在十六进制记法中，允许省略前导 0。例如把 0000 中的前三个 0 省略，写成 1 个 0。

68E6:8C64:FFFF:FFFF:**0**:1180:960A:FFFF

- 冒号十六进制记法可以允许**零压缩** (zero compression)，即一连串连续的零可以为一对冒号所取代。

FF05:0:0:0:0:0:0:B3 可压缩为 FF05::B3

- 注意：**在任一地址中只能使用一次零压缩。

### IPv6 地址

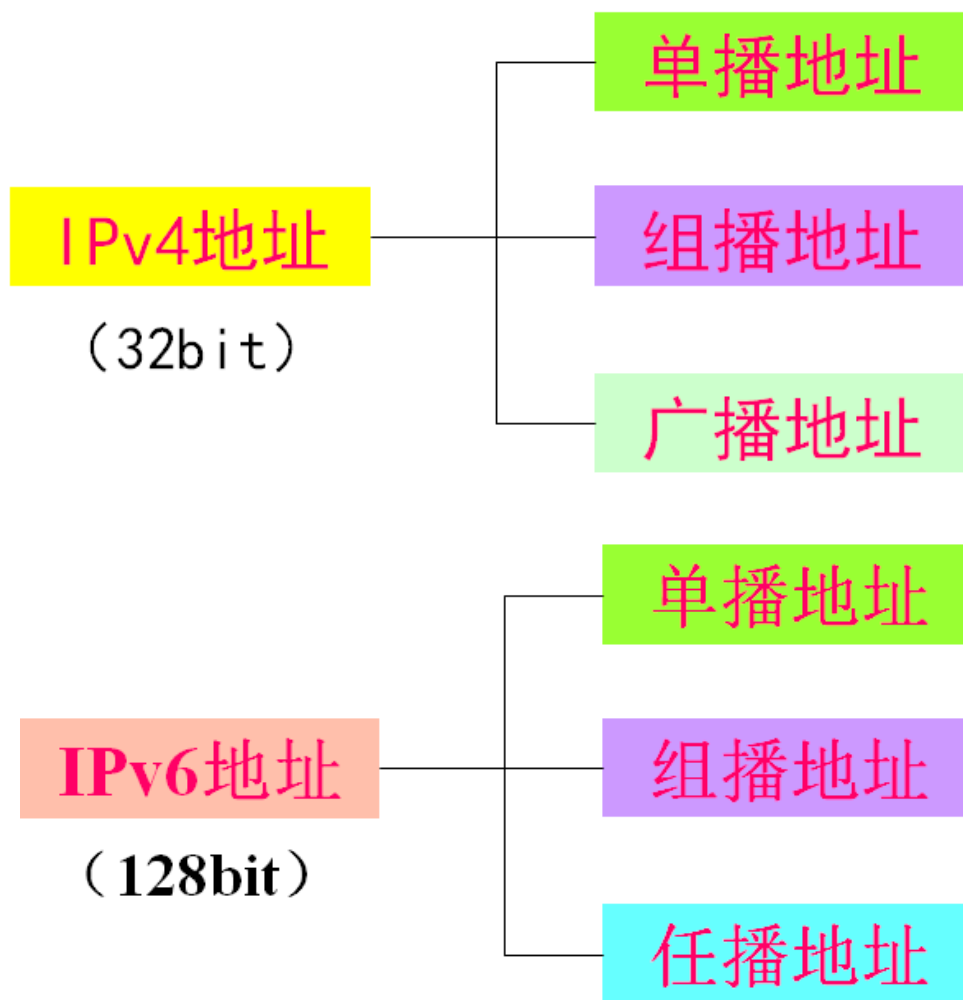
- 例1 : 1050:0000:0000:0000:0005:0600:300c:326b
  - 1050:0:0:0:5:600:300c:326b
  - 1050::5:600:300c:326b
- 例2 : ff06:0:0:0:0:0:0:c3
  - ff06::c3
- 例3 : 0:0:0:0:0:0:128.10.2.1
  - ::128.10.2.1
- CIDR的斜线表示法仍然可用，但取消了子网掩码。
- 例4 : 60 位的前缀 12AB00000000CD3
  - 12AB:0000:0000:CD30:0000:0000:0000:0000/60
  - 12AB::CD30:0:0:0:0/60 (零压缩)
  - 或 12AB:0:0:CD30::/60 (零压缩)

IPv6地址：  
前缀+接口标识

### IPv6 地址

地址类型	二进制前缀	IPv6记法	说明
未指明地址	00...0 (128位)	::/128	16字节的全0地址，只能用作源地址。当某台主机尚未分配到一个标准的IP地址时，可使用未指明地址进行查询，得到它的IP地址。这类地址仅此一个。
环回地址	00...1 (128位)	::1/128	作用与 IPv4 的环回地址一样，这类地址也是仅此一个。
多播地址	11111111 (8位)	FF00::/8	功能和 IPv4 的一样，这类地址占 IPv6 地址总数的 1/256。
本地链路单播地址	1111111010 (10位)	FE80::/10	未连接到互联网，不能和互联网上的其他主机通信，这类地址占 IPv6 地址总数的 1/1024。
全球单播地址	除上述四种外，所有其他的二进制前缀		

## IPv4地址和IPv6地址



◆任播的终点是一组计算机，但数据报在交付时只交付其中的一个。

◆通常是按照路由算法得出的距离最近的一个。

### 节点和接口（补充）

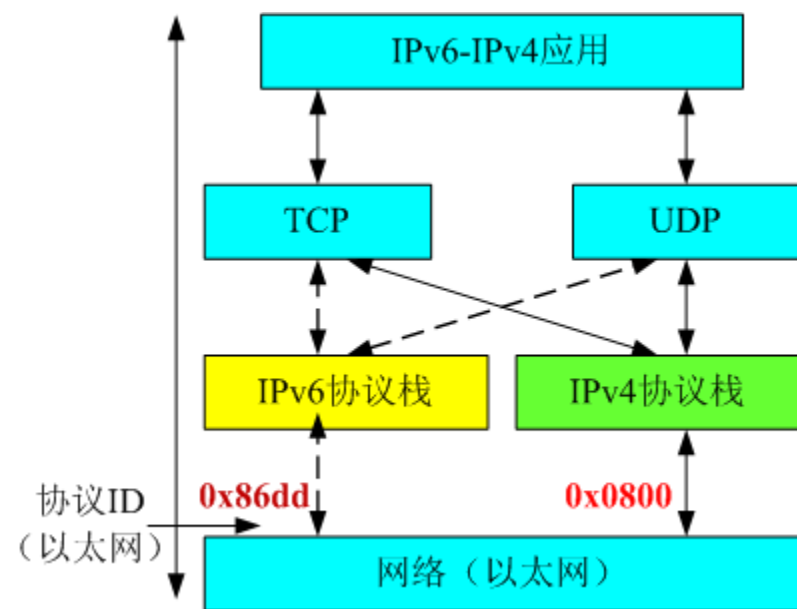
- IP 将实现 IP 的主机和路由器均称为**节点**。
- 一个节点可能有多个与链路相连的接口。
- IP 地址是分配给节点上**接口**的。
  - 一个具有多个接口的节点可以有多个单播地址；
  - 其中任何一个地址都可以当作到达该节点的目的地址。

### IPv4向IPv6过渡的技术

- 向 IPv6 平滑过渡过程中需要注意的问题
  - 充分利用现有IPv4地址资源，节约成本并保护原使用者的利益；
  - 在实现网络设备互联互通的同时实现信息高效无缝传递；
  - 应逐步、渐进过渡，尽可能简便。
- 三种向 IPv6 过渡的技术
  - 双协议栈
  - 隧道技术
  - 网络地址翻译-协议翻译 ( NAT-PT )

## IPv4向IPv6过渡的技术——双协议栈

- 双栈配置
  - 主机：两个协议栈、两套IP地址。
  - 路由器：两套IP地址、两套路由选择机制（IPv4 和 IPv6）
- 采用哪一种地址由域名系统（DNS）返回的地址类型决定。

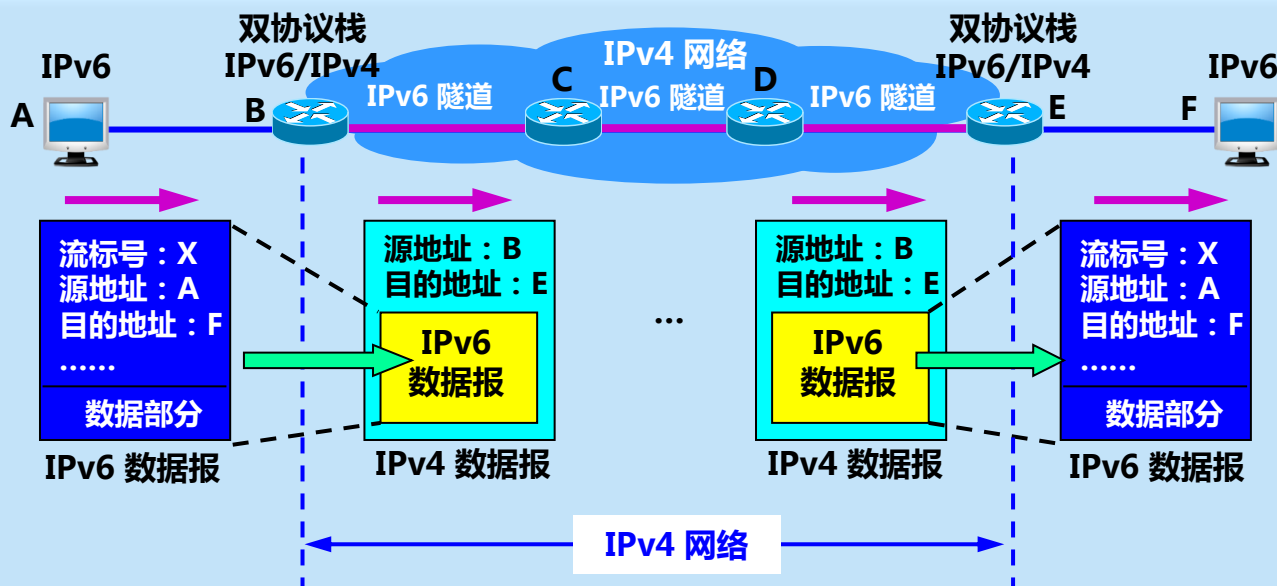


IPv6/IPv4双协议栈工作过程

- 域名系统 (DNS) 是互联网使用的命名系统，一个联机分布式数据库系统，用来把机器名字转换为IP地址。
- 域名到IP地址的解析是由分布在互联网上的许多域名服务器程序（简称为域名服务器）共同完成的。

## IPv4向IPv6过渡的技术——隧道技术

- 在 IPv6 数据报要进入 IPv4 网络时，把 IPv6 数据报封装成为 IPv4 数据报，整个的 IPv6 数据报变成了 IPv4 数据报的数据部分。
- 当 IPv4 数据报离开 IPv4 网络中的隧道时，再把数据部分（即原来的 IPv6 数据报）交给主机的 IPv6 协议栈。

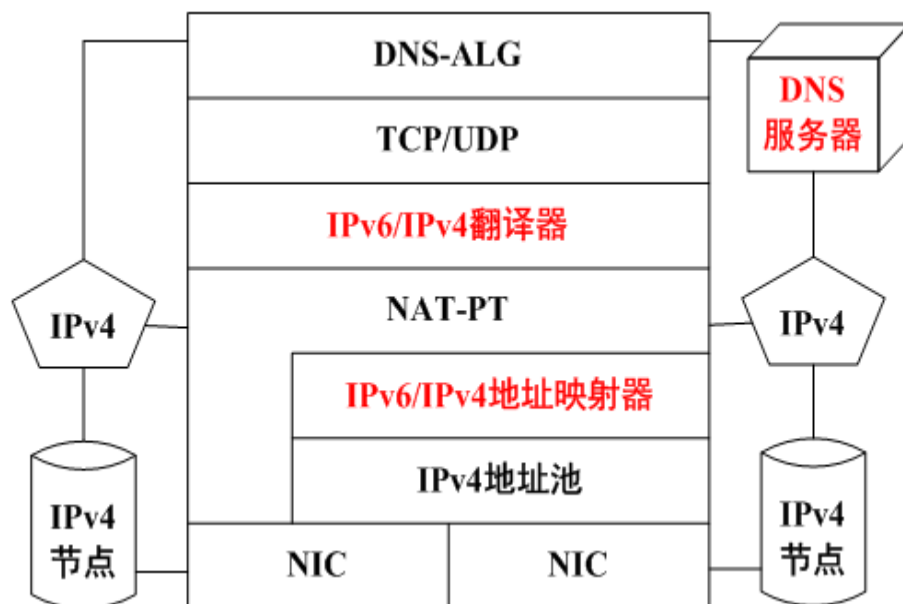


使用隧道技术进行从 IPv4 到 IPv6 的过渡



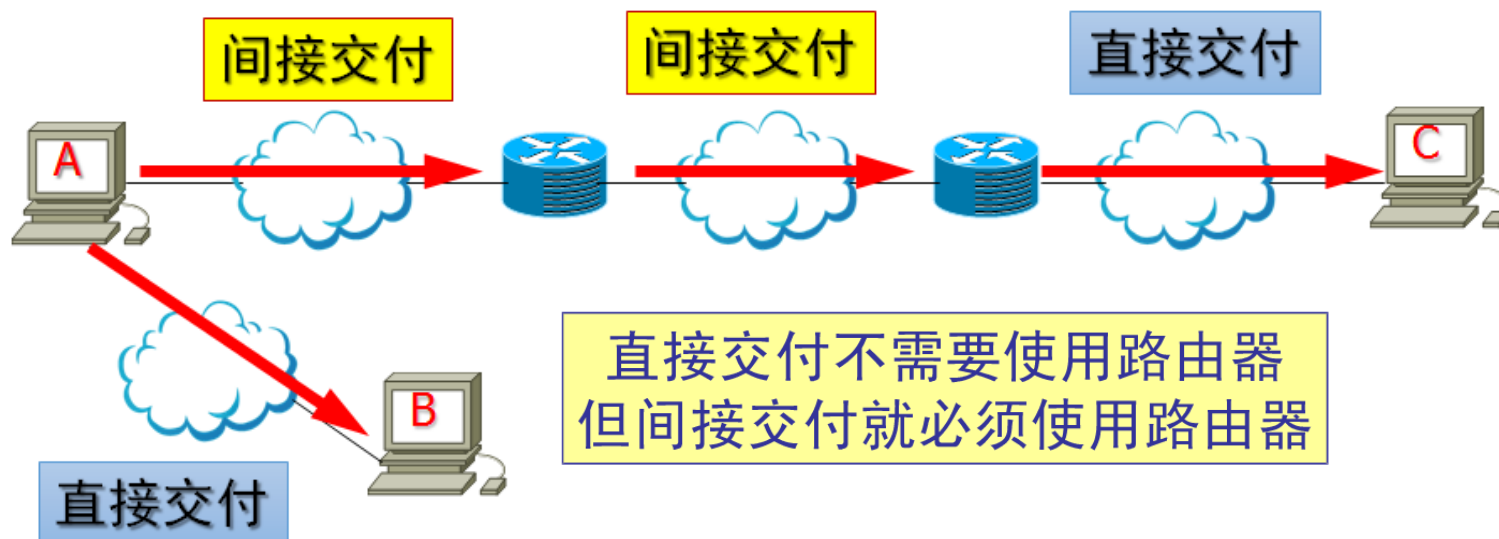
## IPv4向IPv6过渡的技术—— NAT-PT

- 主机必须是双协议栈
- 在协议栈中插入3个特殊的扩展模块
  - 域名解析服务器
  - IPv6/IPv4地址映射器
  - IPv6/IPv4翻译器
- 存在的问题：需要转换数据报的首部，破坏了端到端的服务，有可能限制业务提供平台的容量和扩展性，从而成为网络性能的瓶颈。



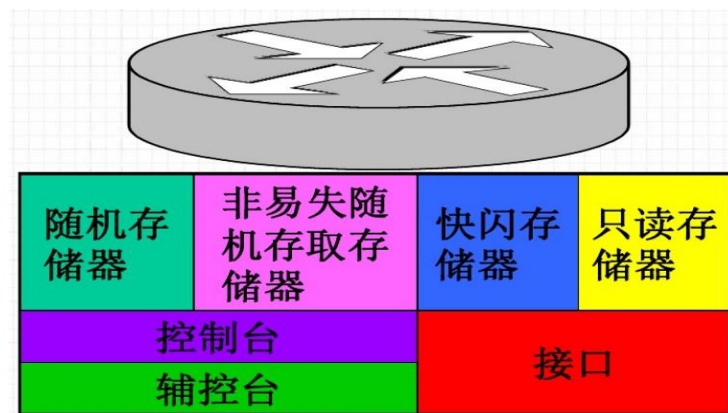
## 直接交付和间接交付

- IP的目的就是要提供一个可以包含多个物理网络的虚拟网络，并提供无连接的数据报交付服务。
- 当源主机和目的主机在**同一个网络**，或是当目的路由器向目的主机传送时，数据报将**直接交付**。
- 如果目的主机与源主机**不在同一个网络**，分组就要**间接交付**。



## 路由器

- 路由器是一种典型的**网络层设备**，其任务是**连通不同的网络，转发分组**。
- 路由器由**硬件**和**软件**组成。硬件主要由中央处理器（CPU）、内存、闪存（相当于硬盘）、接口、控制端口、主板等物理硬件和电路组成；软件主要由路由器的操作系统组成。



内部配置组件

### 路由器

#### ● 路由器主要采用的内存类型

- **ROM（只读内存）**：存储路由器的开机诊断程序、引导程序和操作系统软件。**ROM中软件升级需要更换芯片。**
- **FLASH（闪存）**：可擦除的、可编程的ROM，用来保存操作系统的映像和微码；更新软件时不需要拆除和替换处理器上的芯片。当关机或是重新启动时，FLASH中的内容会保存下来，操作系统的多个备份也能在FLASH中保存。
- **RAM/DRAM（随机存取内存或动态随机存取内存）**：存储路由表项目、ARP高速缓存项目、数据包缓冲队列（共享的RAM）以及数据包控制队列。当路由器处于开机状态时，RAM也为路由器的配置文件提供临时和/或连续的存储。当路由器关机或重启时，RAM中的内容将会丢失。
- **NVRAM（非易失RAM）**：存储路由器备份配置文件，当路由器关机或重启时，NVRAM中的内容将会保存。

## 路由器

## ● 路由器配置端口

- **Console端口**：用来进行路由器的基本配置，通过专用连线与计算机相连。路由器的Console端口多为RJ-45端口。
- **AUX端口**：是一个异步端口，主要用于远程配置，也可用于拨号连接，还可以通过收发器与MODEM进行连接。



### 路由器

- 路由器局域网端口



AUI端口



SC端口

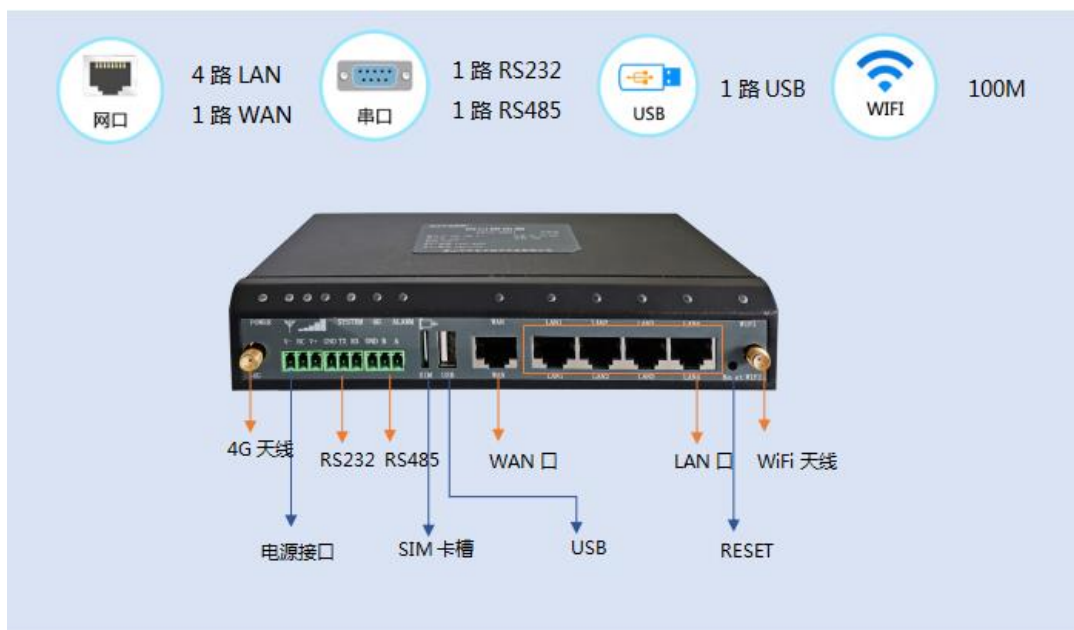


RJ-45端口

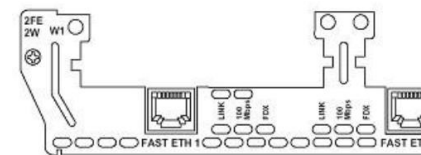


## 路由器

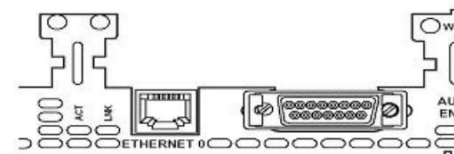
## ● 路由器广域网端口



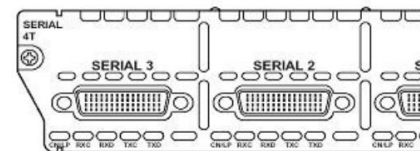
4G工业路由器示意图



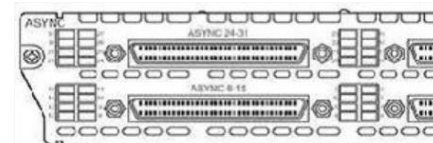
RJ-45端口



AUI端口



高速同步串口

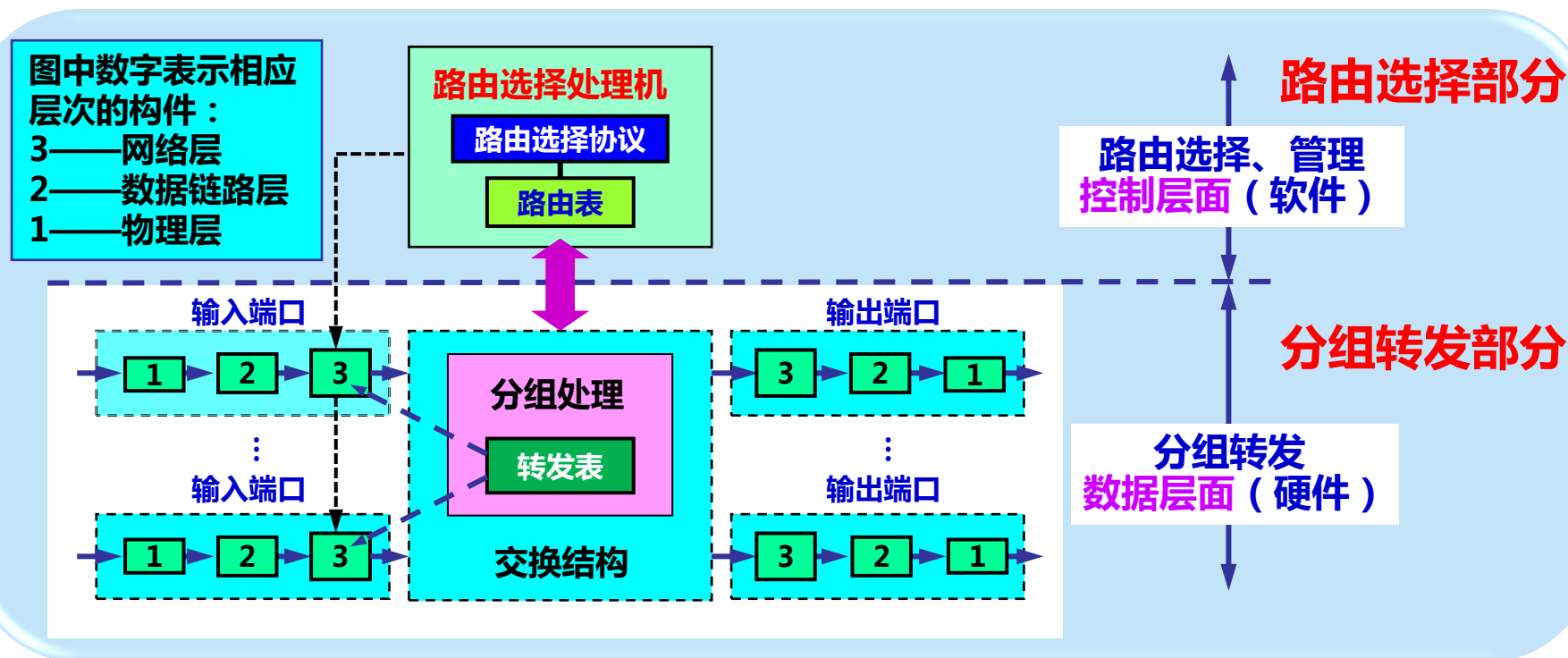


异步串口

## 路由器

## ● 路由器的结构

- 路由选择处理机
- 分组处理与交换部分



注意：这里的端口就是**硬件接口**。



## 路由器

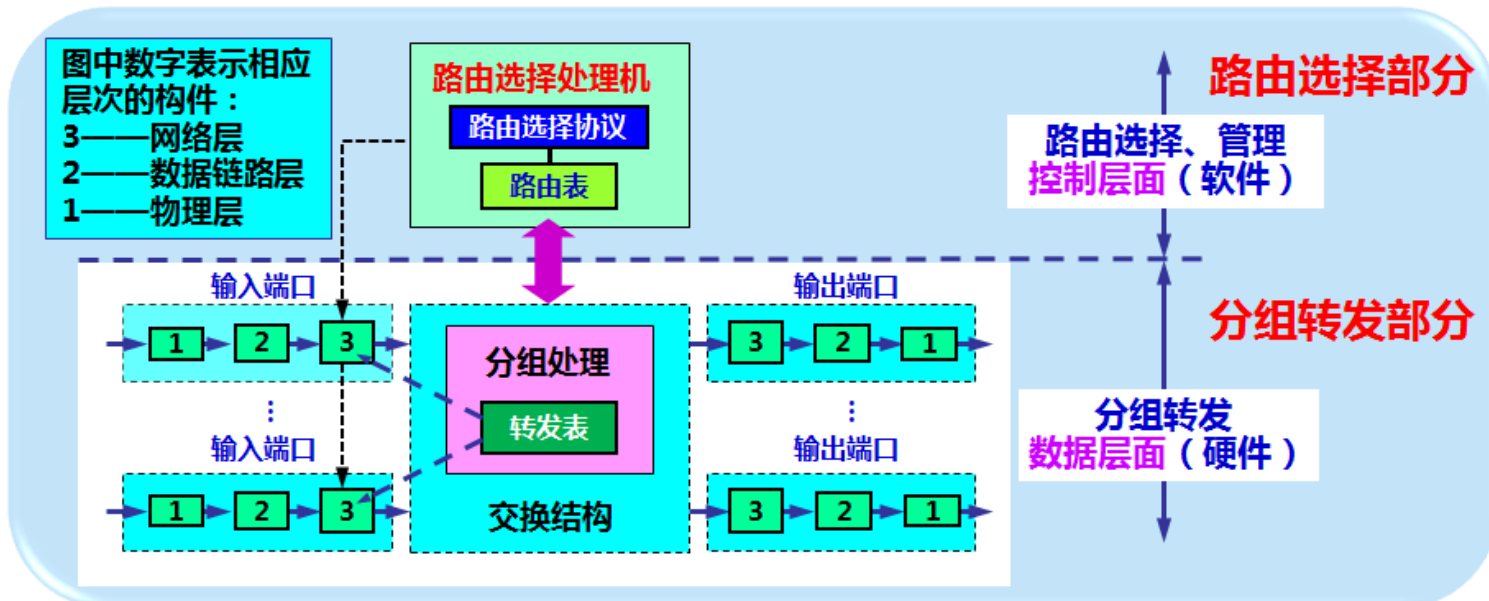
- 网络层抽象地划分为**数据层面**（或转发层面）和**控制层面**。

- **数据层面**

- 每个路由器**独立工作**，根据收到的IP数据报的目的地址，按**转发表转发**至下一跳路由器；
- 采用硬件转发，速度快（纳米数量级， $10^{-9}$ 秒）。

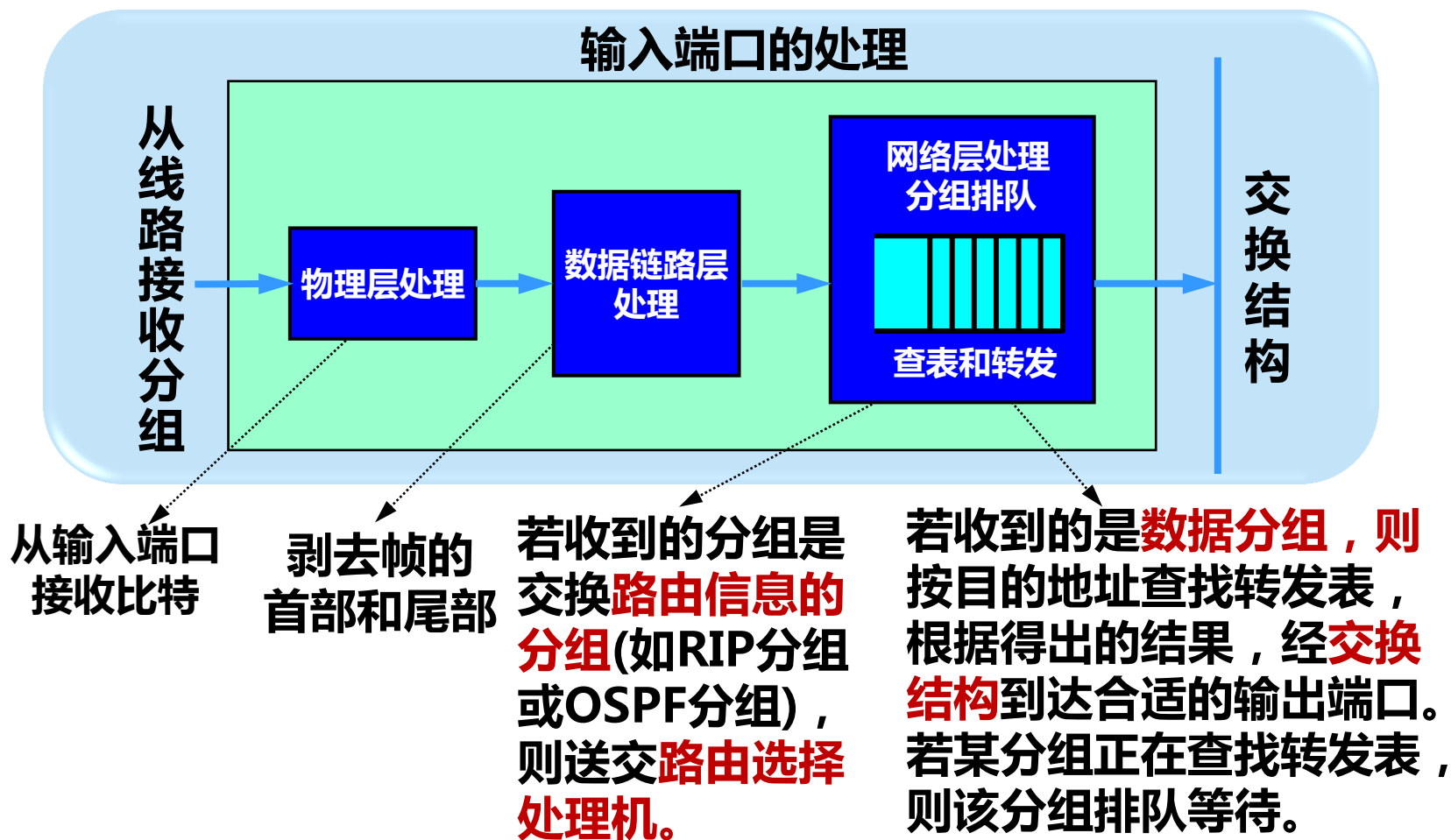
- **控制层面**

- 运行路由选择协议（软件），**生成路由表**（耗时长，秒级）；
- 路由器间需要**相互协作**，交换路由信息。



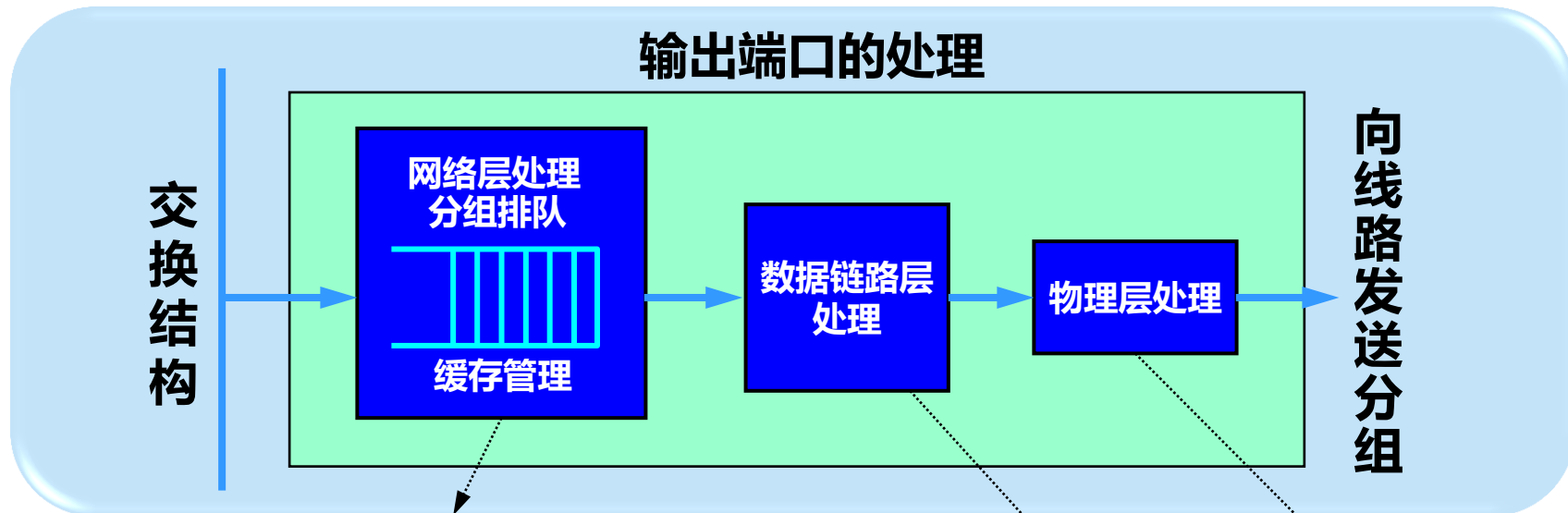
## 路由器

## ● 输入端口对线路上收到的分组的处理



## 路由器

- 输出端口将交换结构传送来的分组发送到线路上



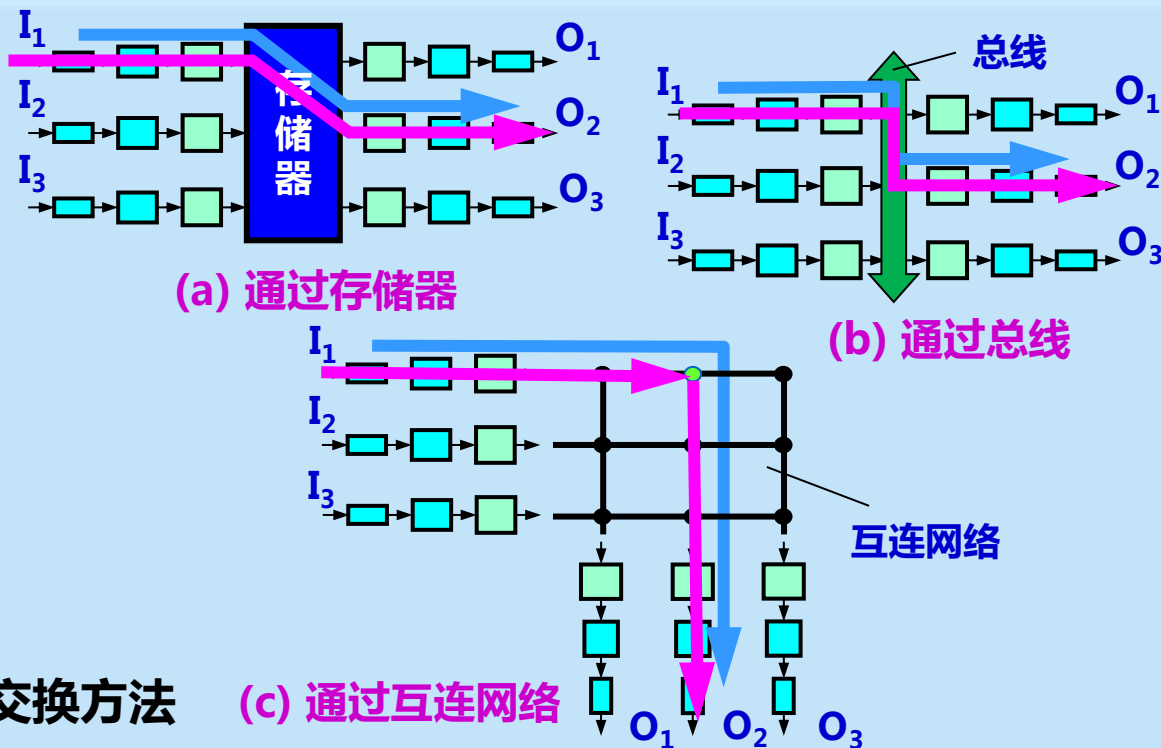
从交换结构接收分组。当交换结构传送的分组的速率**超过**输出链路的发送速率时，来不及发送的分组就必须**缓存**在队列中。若分组处理的速率**低于**分组进入队列的速率，当队列满时，**丢弃**后面进入的分组。

给分组加上链路层的首部和尾部。

发送到输出线路上。

## 路由器

- 交换结构是路由器的关键构件，把分组从一个输入端口转移到某个合适的输出端口。



三种常用的交换方法

(c) 通过互连网络

### 路由器 -- 路由选择和转发的区别

#### 路由选择

- ◆ 按照路由选择算法，得出整个网络的拓扑变化情况，动态地改变所选择的路由，并由此构造出整个路由表，根据路由表的信息为每一个需要转发的数据报找到相应的下一跳地址。
- ◆ 涉及到很多路由器。
- ◆ 路由表根据路由选择算法得出。
- ◆ 路由表一般包含目的IP地址、下一跳 IP 地址等。

#### 转发

- ◆ 根据转发表将用户的 IP 数据报从路由器合适的端口转发出去。
- ◆ 仅涉及到一个路由器。
- ◆ 转发表根据路由表形成。
- ◆ 转发表必须包含完成转发功能所必需的信息，每一行必须包含从要到达的目的网络到输出端口和某些 MAC 地址信息（如下一跳的以太网地址）的映射。

## 路由器

### 路由表

目的IP地址

下一站（或下一跳）路由器的IP地址，或者有直接连接的网络IP地址。

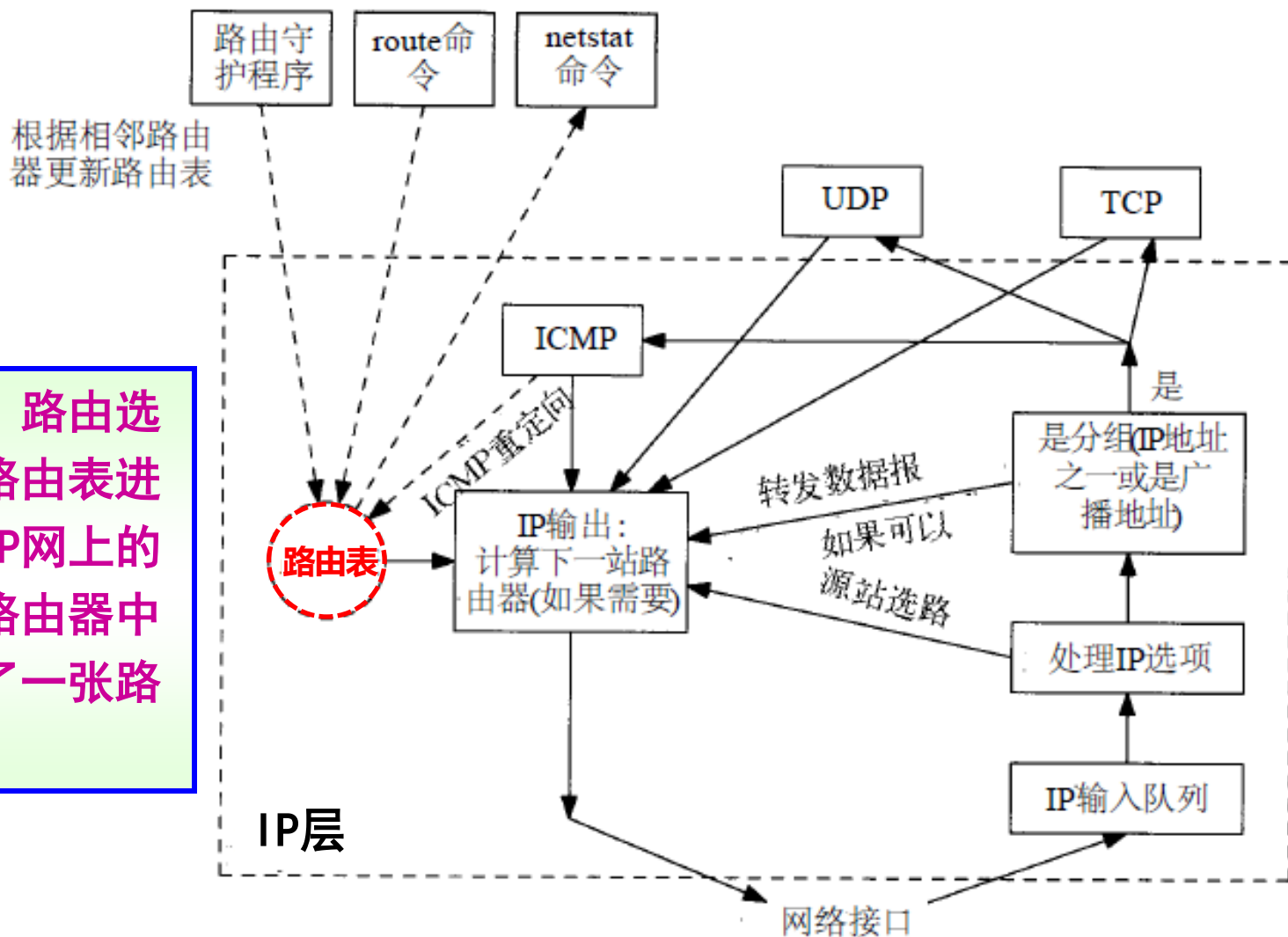
标志

为数据报的传输指定一个网络接口

```
svr4 % netstat -rn
Routing tables
Destination      Gateway         Flags    Refcnt  Use    Interface
140.252.13.65    140.252.13.35  UGH      0        0      emd0
127.0.0.1        127.0.0.1      UH       1        0      lo0
default          140.252.13.33  UG       0        0      emd0
140.252.13.32    140.252.13.34  U        4       25043  emd0
```

标志	说 明
U	该路由可以使用
G	该路由是一个网络（路由器）。如果没有设置该标志，则说明目的地是直接相连的。
H	该路由是到一个主机，即目的地址是一个完整的主机地址。如果没有设置该标志，则说明该路由是到一个网络，而目的地址是一个网络地址：一个网络号或网络号与子网号的组合。
D	该路由由重定向报文创建。
M	该路由已被重定向报文修改。

## 网络层的工作流程



在IP中，路由选择依靠路由表进行。在IP网上的主机和路由器中均保存了一张路由表。

### 网络层的工作流程

#### ● netstat命令

- 用于显示与IP、TCP、UDP和ICMP协议相关的统计数据，一般用于检验本机各端口的网络连接情况。
- 提供MTU、输入分组数、输入错误、输出分组数、输出错误、冲突以及当前的输出队列长度等信息。

netstat的一些常用选项	
<b>netstat -s</b>	按照各个协议分别显示其统计数据
<b>netstat -e</b>	显示关于以太网的统计数据
<b>netstat -r</b> <b>route print</b>	显示关于路由器的信息
<b>netstat -a</b>	显示一个所有的有效连接信息列表，包括已建立的连接，也包括监听连接请求的那些连接
<b>netstat -n</b>	显示所有已建立的有效连接