**5.1 IP 基础知识**

## IP 基本认识

IP 在 TCP/IP 参考模型中处于第三层，也就是**网络层**。

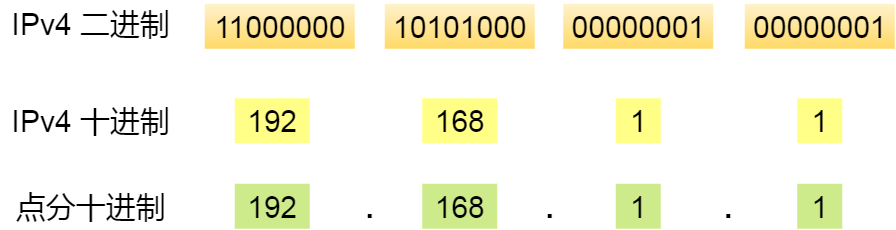
网络层的主要作用是：**实现主机与主机之间的通信，也叫点对点（end to end）通信。**

网络层与数据链路层有什么关系呢？

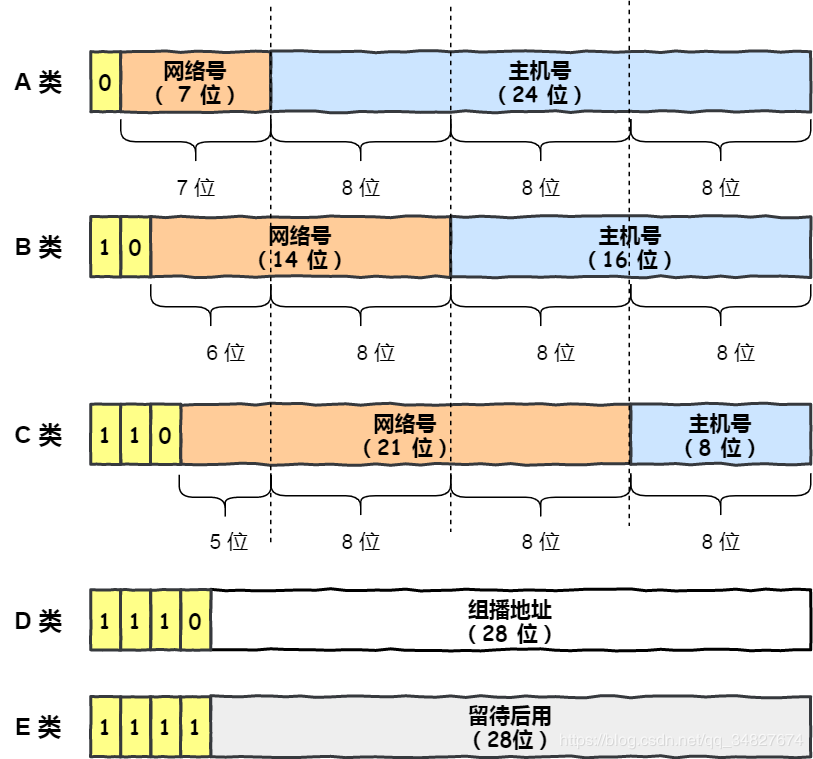
**MAC 的作用则是实现「直连」的两个设备之间通信，而 IP 则负责在「没有直连」的两个网络之间进行通信传输。**

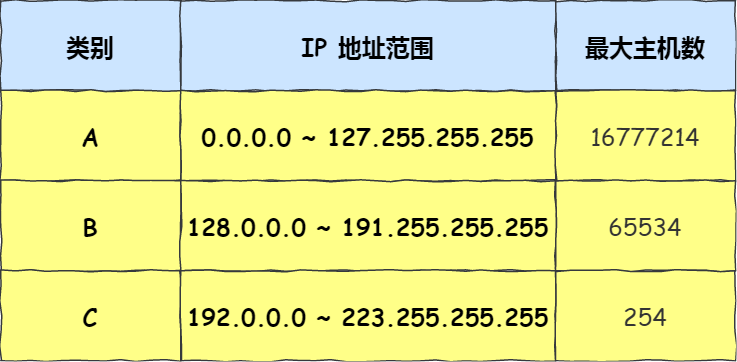
**源IP地址和目标IP地址在传输过程中是不会变化的（前提：没有使用 NAT 网络），只有源 MAC 地址和目标 MAC 一直在变化。**

## IP 地址的基础知识



### IP 地址的分类





A、B、C 分类地址最大主机个数是如何计算的呢？

2的主机号位数次方-2

减 2的原因：**全0和全1不可以用**

* 主机号全为 1 指定某个网络下的所有主机，用于广播
* 主机号全为 0 指定某个网络

广播地址用于什么？

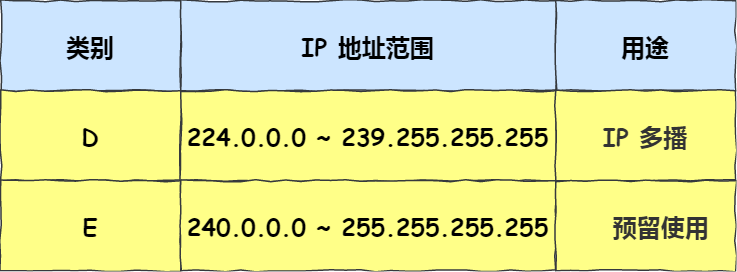
广播地址用于在**同一个链路中相互连接的主机之间发送数据包**

广播地址可以分为本地广播和直接广播两种。

* **在本网络内广播的叫做本地广播**。例如网络地址为 192.168.0.0/24 的情况下，广播地址是 192.168.0.255 。因为**这个广播地址的 IP 包会被路由器屏蔽**，所以不会到达 192.168.0.0/24 以外的其他链路上。
* **在不同网络之间的广播叫做直接广播。**例如网络地址为 192.168.0.0/24 的主机向 192.168.1.255/24 的目标地址发送 IP 包。收到这个包的路由器，将数据转发给 192.168.1.0/24，使得所有 192.168.1.1~192.168.1.254 的主机都能收到这个包（由于直接广播有一定的安全问题，多数情况下会在路由器上设置为不转发）。

什么是 D、E 类地址？

 D 类和 E 类地址是没有主机号的，所以不可用于主机 IP，D 类常被用于**多播（组播）**，E 类是预留的分类，暂时未使用。



多播地址用于什么？

多播用于**将包发送给特定组内的所有主机。广播无法穿透路由**，若想给其他网段发送同样的包，就可以使用**可以穿透路由的多播**。

从 224.0.0.0 ~ 239.255.255.255 都是多播的可用范围，其划分为以下三类：

* 224.0.0.0 ~ 224.0.0.255 为预留的组播地址，**只能在局域网中**，路由器是不会进行转发的。
* 224.0.1.0 ~ 238.255.255.255 为**用户可用的组播地址**，可以**用于 Internet 上**。
* 239.0.0.0 ~ 239.255.255.255 为**本地管理组播地址**，可供内部网在内部使用，仅在**特定的本地范围内有效**。

IP 分类的优点

**简单明了、选路简单**

IP 分类的缺点

缺点一

**同一网络下没有地址层次，缺少地址的灵活性**

缺点二

**不能很好的与现实网络匹配，C类太少，B类太多**

### 无分类地址 CIDR

**无分类地址**不再有分类地址的概念，32 比特的 IP 地址被划分为两部分，前面是**网络号**，后面是**主机号**。

表示形式 a.b.c.d/x，其中 /x 表示前 x 位属于**网络号**， x 的范围是 0 ~ 32，这就使得 IP 地址更加具有灵活性。

**子网掩码：**划分网络号与主机号的形式

**将子网掩码和 IP 地址按位计算 AND，就可得到网络号。**

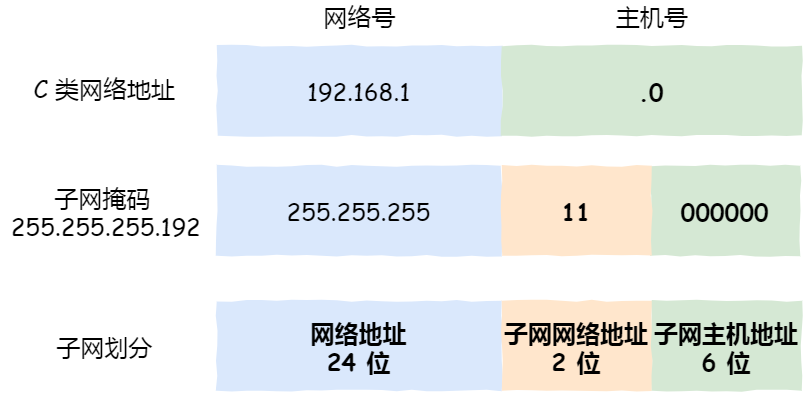
为什么要分离网络号和主机号？

通讯时候用于判断两台计算机是否处于同一个广播域内

划分子网

**子网掩码**的另一个作用：划分子网

网络地址 192.168.1.0，使用子网掩码 255.255.255.192 对其进行子网划分：



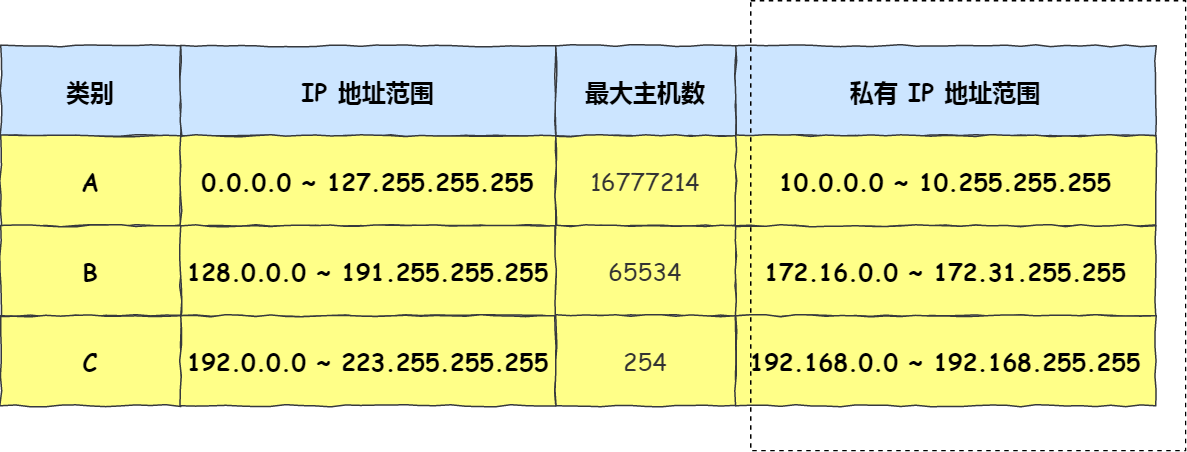
每个子网中，子网主机地址全0和全1也有特殊用途，不能用

全0：这个子网的网络地址

全1：这个子网的广播地址

### 私有 IP 地址

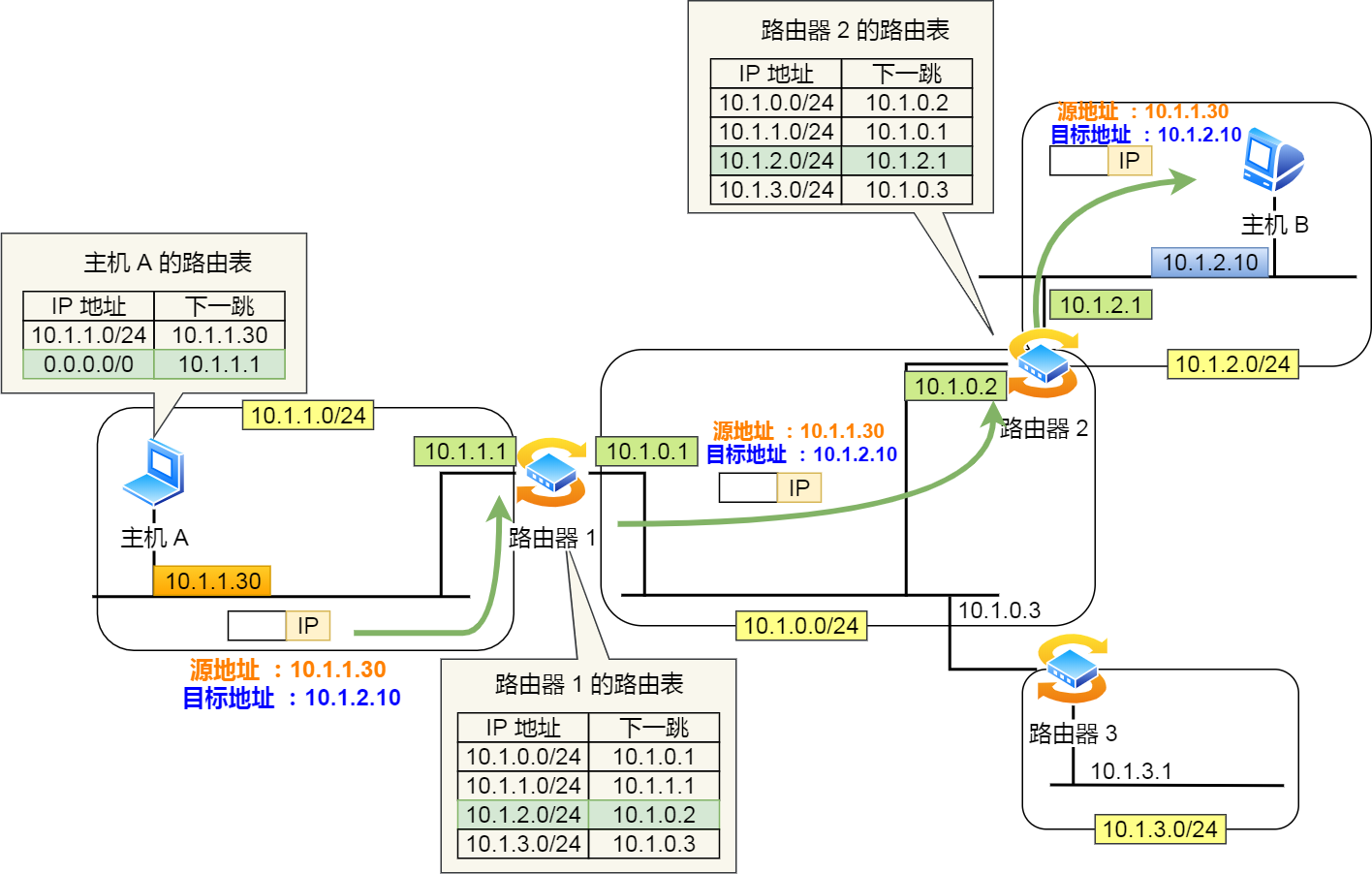
 A、B、C 类地址中，分为公有 IP 地址和私有 IP 地址。允许组织内部的 IT 人员自己管理、自己分配，而且可以重复。



### IP 地址与路由控制

IP地址的**网络地址**这一部分是用于进行路由控制。

发送 IP 包时，首先要确定 IP 包首部中的目标地址，再从路由控制表中找到与该地址具有**相同网络地址**的记录，根据该记录将 IP 包转发给相应的下一个路由器。如果路由控制表中存在多条相同网络地址的记录，就选择相同位数最多的网络地址，也就是最长匹配。



环回地址

计算机使用一个特殊的 IP 地址 **127.0.0.1 作为环回地址**。与该地址具有相同意义的是一个叫做 localhost 的主机名。使用这个 IP 或主机名时，**数据包不会流向网络**。

### IP 分片与重组

每种数据链路的最大传输单元 MTU 都是不相同的

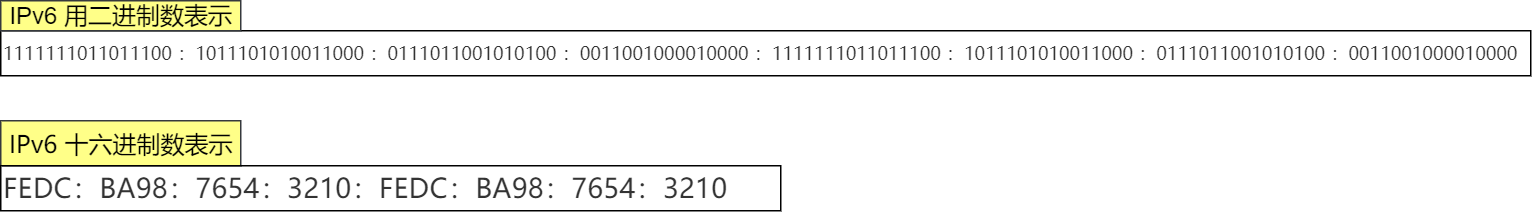
最常见数据链路是**以太网**，它的 MTU 是 1500 字节

 IP 数据包大小大于 MTU 时， IP 数据包就会被分片，由**目标主机**重组。**中间路由只会对 IP 包进行分片（在MTU进一步变小时候），不会对分片进行重组**

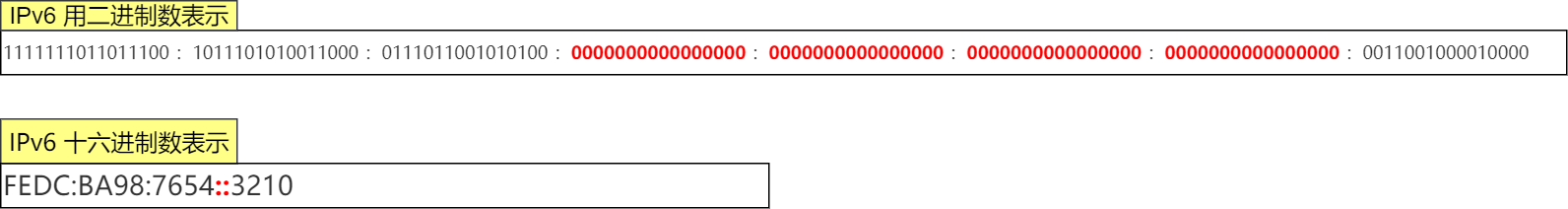
在分片传输中，**一旦某个分片丢失，则会造成整个 IP 数据报作废**，所以 TCP 引入了 MSS 也就是在 TCP 层进行分片不由 IP 层分片，那么对于 **UDP 尽量不要发送一个大于 MTU 的数据报文**。

### IPv6 基本认识

IPv6 的地址是 128 位的，是以每 16 位作为一组，每组用冒号 「:」 隔开

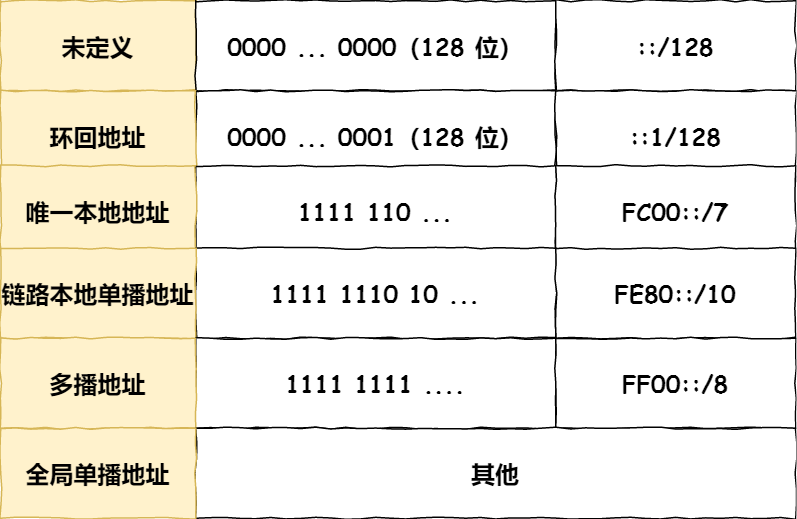


如果出现连续的 0 时还可以将这些 0 省略，并用两个冒号 「::」隔开。但是，一个 IP 地址中只允许出现一次两个连续的冒号



IPv6 的地址主要有以下类型地址：

* 单播地址，用于一对一的通信
* 组播地址，用于一对多的通信
* 任播地址，用于通信最近的节点，最近的节点是由路由协议决定
* **没有广播地址**

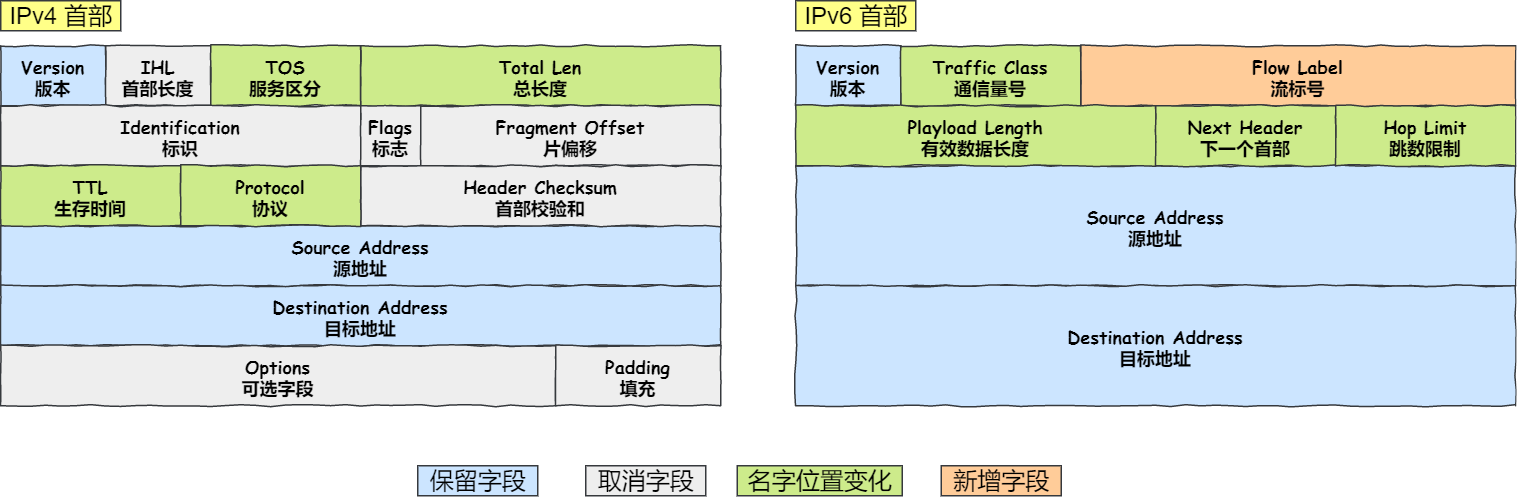


IPv6 单播地址类型

有三类：

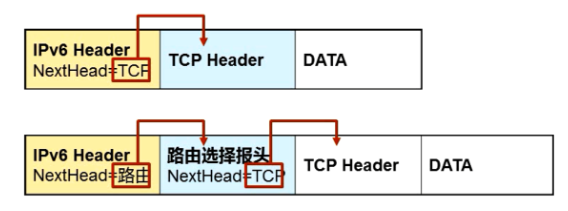
* 在**同一链路单播通信，不经过路由器**，可以使用**链路本地单播地址**，IPv4 没有此类型
* 在**内网里单播通信**，可以使用**唯一本地地址**，相当于 IPv4 的私有 IP
* 在**互联网通信**，可以使用**全局单播地址**，相当于 IPv4 的公有 IP

### IPv4 首部与 IPv6 首部



IPv6 相比 IPv4 的首部改进：

* **取消了首部校验和字段。**
* **取消了分片/重新组装相关字段。** 分片与重组是耗时的过程，IPv6 不允许在中间路由器进行分片与重组（IPv4中有可能分片），这种操作只能在源与目标主机，这将大大提高了路由器转发的速度。
* **取消选项字段。** 选项字段不再是标准 IP 首部的一部分了，但它并没有消失，而是可能出现在 IPv6 首部中的「下一个首部」指出的位置上。删除该选项字段使的 IPv6 的首部成为固定长度的 40 字节。



## IP 协议相关技术

### DNS

**域名解析系统**

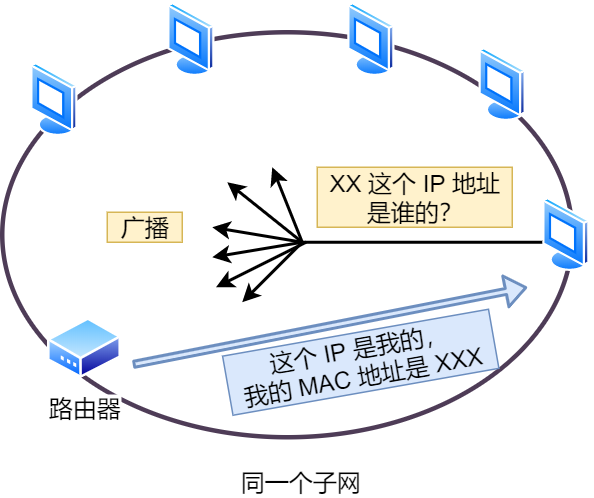
域名解析的工作流程

**浏览器**首先看一下自己的**缓存**里有没有，如果没有就向**操作系统的缓存**要，还没有就检查**本机域名解析文件 hosts**，如果还是没有，就会 DNS 服务器进行查询，查询的过程如下：

1. 客户端首先会发出一个 DNS 请求，问 www.server.com 的 IP 是啥，并发给**本地 DNS 服务器**（也就是客户端的 TCP/IP 设置中填写的 DNS 服务器地址）。
2. 本地域名服务器收到客户端的请求后，如果缓存里的表格能找到 www.server.com，则它直接返回 IP 地址。如果没有，本地 DNS 会去问它的**根域名服务器**：“老大， 能告诉我 www.server.com 的 IP 地址吗？” **根域名服务器是最高层次的，它不直接用于域名解析，但能指明一条道路**。
3. 根 DNS 收到来自本地 DNS 的请求后，发现后置是 .com，说：“www.server.com 这个域名归 .com 区域管理”，我给你 **.com 顶级域名服务器**地址给你，你去问问它吧。”
4. 本地 DNS 收到顶级域名服务器的地址后，发起请求问“老二， 你能告诉我 www.server.com 的 IP 地址吗？”
5. 顶级域名服务器说：“我给你负责 **www.server.com 区域的权威 DNS 服务器**的地址，你去问它应该能问到”。
6. 本地 DNS 于是转向问权威 DNS 服务器：“老三，www.server.com对应的IP是啥呀？” server.com 的权威 DNS 服务器，它是域名解析结果的原出处。为啥叫权威呢？就是我的域名我做主。
7. 权威 DNS 服务器查询后将对应的 IP 地址 X.X.X.X 告诉本地 DNS。
8. **本地 DNS 再将 IP 地址返回客户端**，客户端和目标建立连接。

### ARP

**地址解析协议**



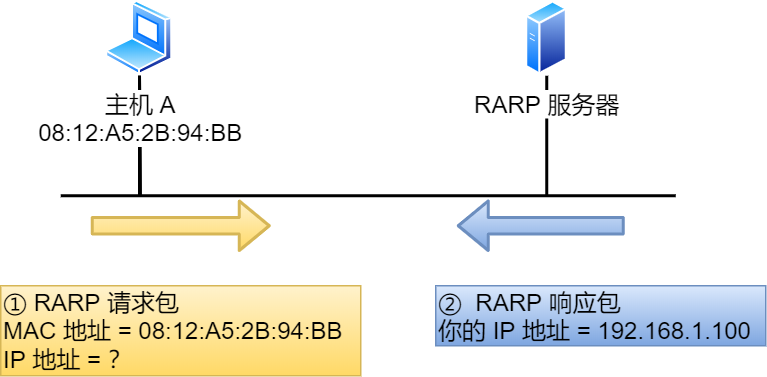
* 主机会通过**广播发送 ARP 请求**，这个包中包含了想要知道的 MAC 地址的主机 IP 地址。
* 当同个链路中的所有设备收到 ARP 请求时，会去拆开 ARP 请求包里的内容，如果 ARP 请求包中的**目标 IP 地址与自己的 IP 地址一致**，那么这个设备就将自己的 MAC 地址塞入 **ARP 响应包**返回给主机。

操作系统通常会把第一次通过 ARP 获取的 MAC 地址缓存起来，超时清除。

RARP 协议

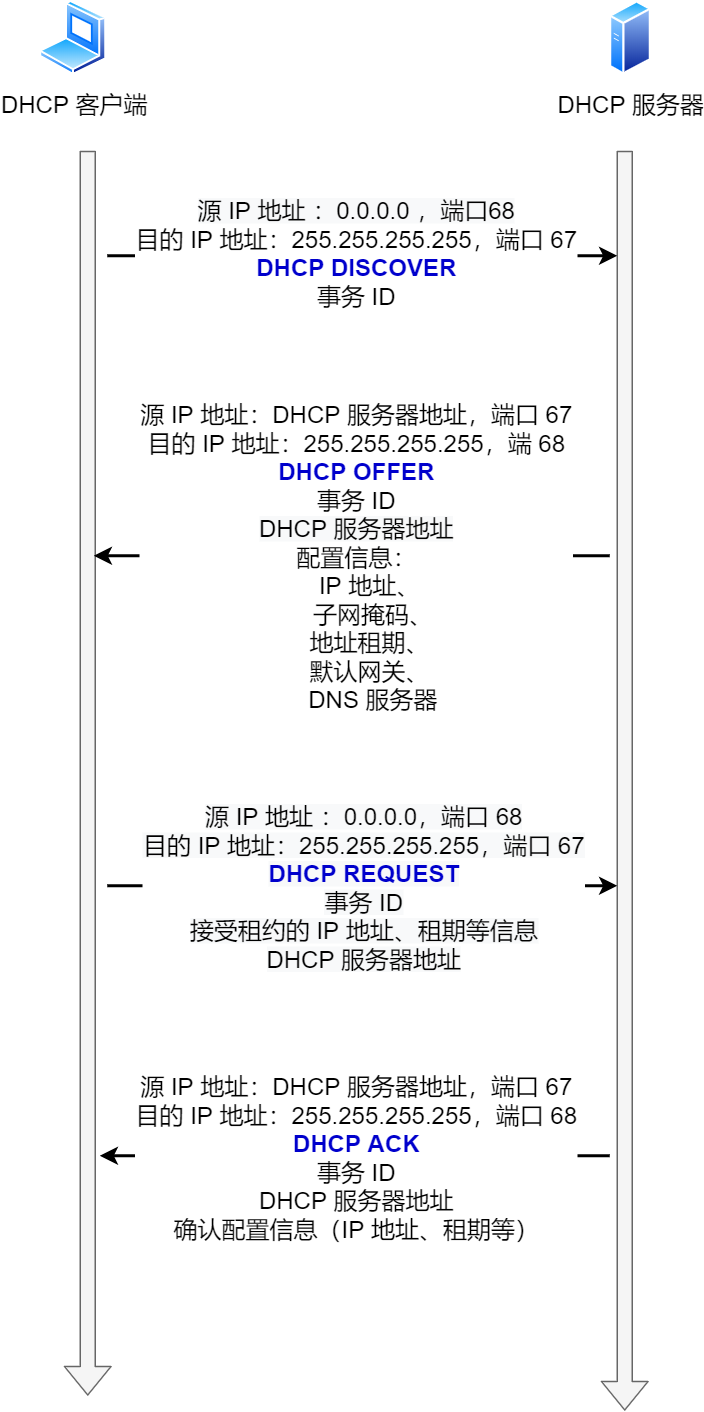
RARP 是**已知 MAC 地址求 IP 地址**。例如将打印机服务器等小型嵌入式设备接入到网络时就经常会用得到。

通常这需要架设一台 RARP 服务器，在这个服务器上注册设备的 MAC 地址及其 IP 地址。然后再将这个设备接入到网络



### DHCP

**动态主机配置协议**



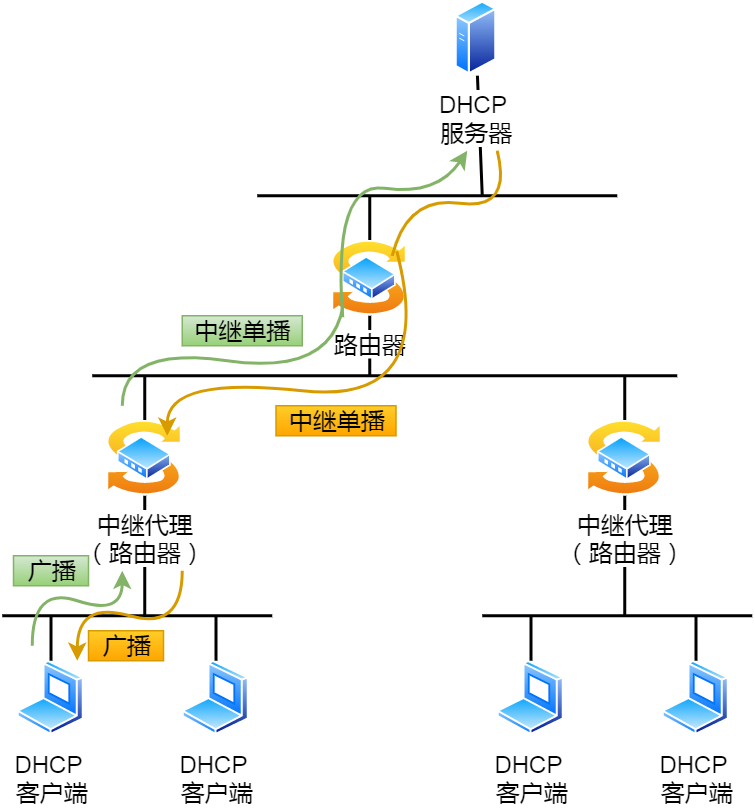
DHCP **客户端**进程监听的是 68 端口号，DHCP **服务端**进程监听的是 67 端口号。

* 客户端首先发起 **DHCP 发现报文（DHCP DISCOVER）** 的 IP 数据报，由于客户端没有 IP 地址，也不知道 DHCP 服务器的地址，所以使用的是 UDP **广播**通信，其使用的广播目的地址是 255.255.255.255（端口 67） 并且使用 0.0.0.0（端口 68） 作为源 IP 地址。DHCP 客户端将该 IP 数据报传递给链路层，链路层然后将帧广播到所有的网络中设备。
* DHCP 服务器收到 DHCP 发现报文时，用 **DHCP 提供报文（DHCP OFFER）** 向客户端做出响应。该报文仍然使用 IP 广播地址 255.255.255.255，该报文信息携带服务器提供可租约的 IP 地址、子网掩码、默认网关、DNS 服务器以及 **IP 地址租用期**。
* 客户端收到**一个或多个**服务器的 DHCP 提供报文后，**从中选择一个服务器**，并向选中的服务器（注意还是用的广播目的地址，不是DHCP服务器地址）发送 **DHCP 请求报文（DHCP REQUEST**进行响应，回显配置的参数。
* 最后，服务端用 **DHCP ACK 报文**对 DHCP 请求报文进行响应，应答所要求的参数。

**全程都是使用 UDP 广播通信**

如果 DHCP 服务器和客户端不是在同一个局域网内

由于路由器不会转发广播包，使用**DHCP 中继代理**，**对不同网段的 IP 地址分配也可以由一个 DHCP 服务器统一进行管理。**

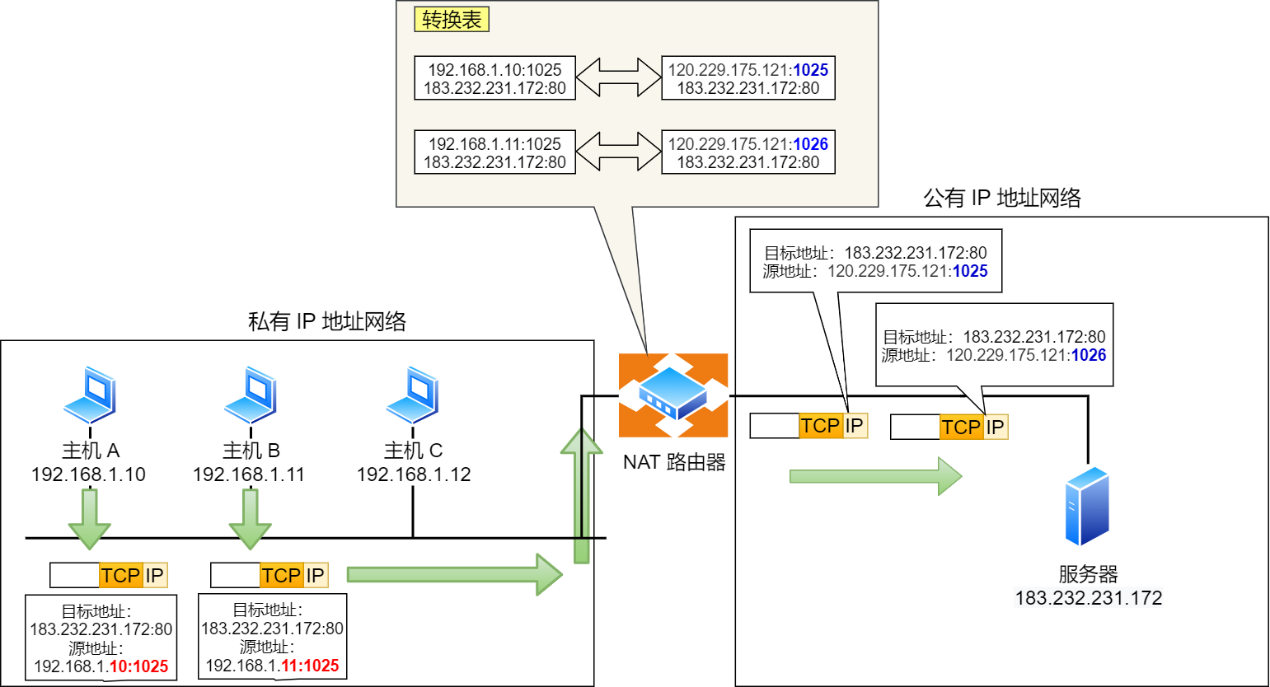


### NAT

**网络地址转换**

NAT 就是同个公司、家庭、教室内的主机对外部通信时，把私有 IP 地址转换成公有 IP 地址。

普通的 NAT 转换没什么意义，可以把 IP 地址 + 端口号一起进行转换，这种转换技术就叫**网络地址与端口转换 NAPT。**



有两个客户端 192.168.1.10 和 192.168.1.11 同时与服务器 183.232.231.172 进行通信，并且这**两个客户端的本地端口都是 1025**。

此时，**两个私有 IP 地址都转换 IP 地址为公有地址 120.229.175.121，但是以不同的端口号作为区分。**生成一个 NAPT 路由器的**转换表**，就可以正确地转换地址跟端口的组合。转换表在 NAT 路由器上自动生成。例如，在 TCP 的情况下，建立 TCP 连接首次握手时的 SYN 包一经发出，就会生成这个表。而后又随着收到关闭连接时发出 FIN 包的确认应答从表中被删除。

NAT缺点

NAT/NAPT 都依赖于自己的转换表：

* 外部无法**主动**与 **NAT 内部服务器**建立连接，因为 NAPT 转换表没有转换记录。
* 转换表的生成与转换操作都会产生性能开销。
* 通信过程中，如果 NAT 路由器重启了，所有的 TCP 连接都将被重置。

如何解决 NAT 潜在的问题

第一种就是改用 IPv6

每台设备都可以配置一个公有 IP 地址

第二种 NAT 穿透技术

在 NAT 穿透技术中，NAT设备后的应用程序处于主动地位，它已经明确地知道 NAT 设备要修改它外发的数据包，于是它主动配合 NAT 设备的操作，主动地建立好映射，这样就不像以前由 NAT 设备来建立映射了。

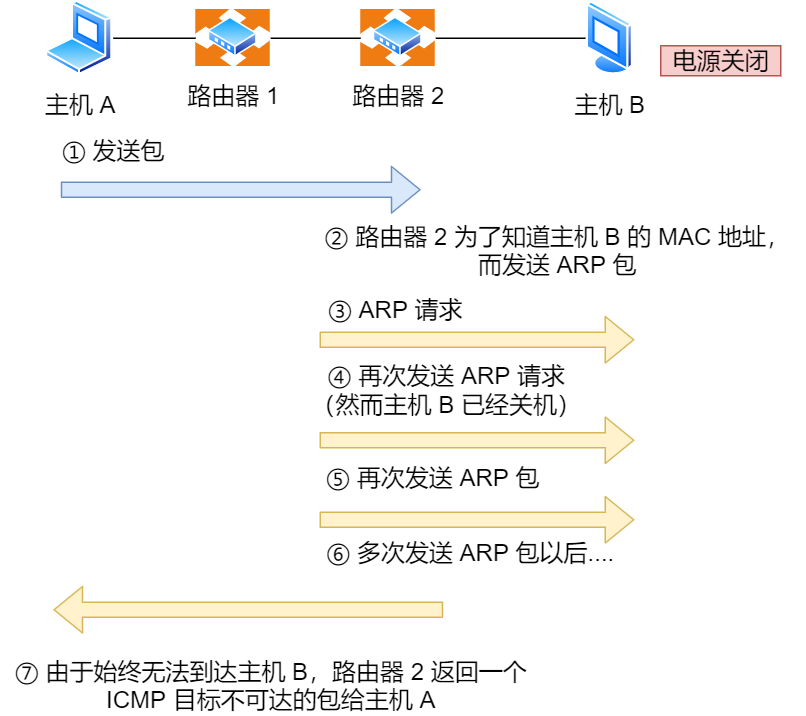
### ICMP

**互联网控制报文协议**

ICMP 功能

**确认 IP 包是否成功送达目标地址、报告发送过程中 IP 包被废弃的原因、改善网络设置等**

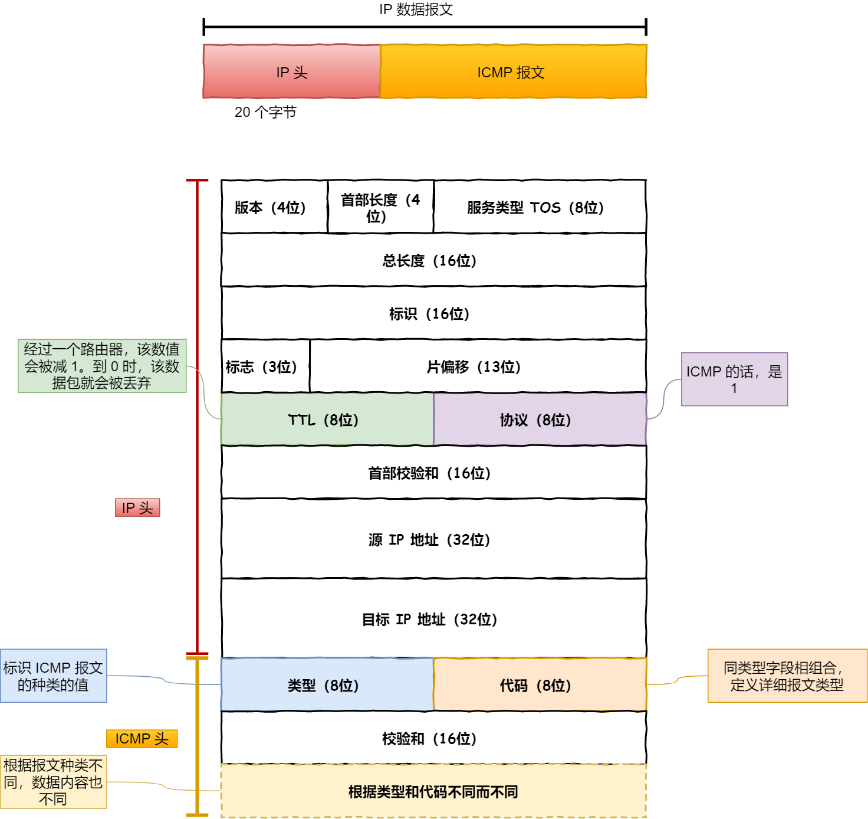




ICMP 的这种通知消息会使用**IP**进行发送 。

ICMP 包头格式

ICMP 报文是封装在 IP 包里面，它工作在网络层，是 IP 协议的助手。



## 查询报文类型

回送消息 —— 类型 0 和 8

**回送消息**用于进行通信的主机或路由器之间，判断所发送的数据包是否已经成功到达对端的一种消息，ping 命令就是利用这个消息实现的。

可以向对端主机发送**回送请求**的消息（ICMP Echo Request Message，类型 8），也可以接收对端主机发回来的**回送应答**消息（ICMP Echo Reply Message，类型 0）。

## 差错报文类型

目标不可达消息（Destination Unreachable Message） —— 类型为 3

IP 路由器无法将 IP 数据包发送给目标地址时，会给发送端主机返回一个**目标不可达**的 ICMP 消息，并在这个消息中显示不可达的具体原因，原因记录在 ICMP 包头的**代码**字段

* 网络不可达代码为 0。路由器中的路由器表匹配不到接收方 IP 的网络号
* 主机不可达代码为 1。路由表中没有该主机的信息，或者该主机没有连接到网络
* 协议不可达代码为 2。对端主机的防火墙已经禁止某个协议访问，或协议服务未启动
* 端口不可达代码为 3。对端主机没有进程监听某个端口
* 需要进行分片但设置了不分片位代码为 4。发送端主机发送 IP 数据报时，将 IP 首部的**分片禁止标志位**设置为1。根据这个标志位，途中的路由器遇到超过 MTU 大小的数据包时，不会进行分片，而是直接抛弃。

原点抑制消息（ICMP Source Quench Message） —— 类型 4

当路由器向低速线路发送数据时，其发送队列的缓存变为零而无法发送出去时，可以向 IP 包的源地址发送一个 ICMP **原点抑制消息**。

重定向消息（ICMP Redirect Message） —— 类型 5

如果路由器发现发送端主机使用了「不是最优」的路径发送数据，那么它会返回一个 ICMP **重定向消息**给这个主机。在这个消息中包含了**最合适的路由信息和源数据**

超时消息（ICMP Time Exceeded Message） —— 类型 11

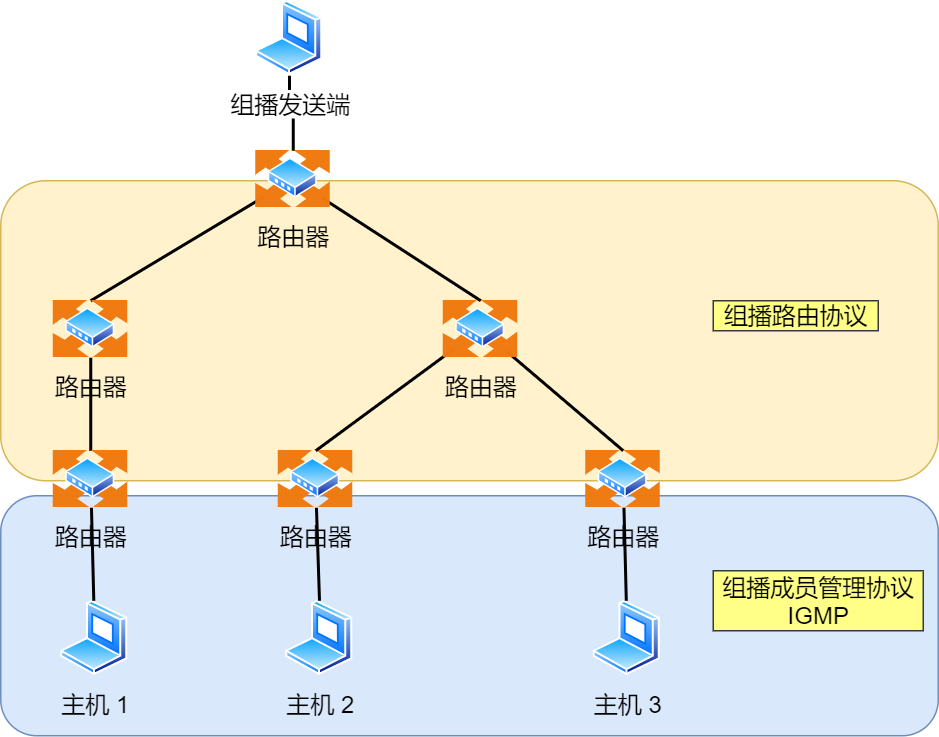
IP 包中有一个字段叫做 **TTL** （Time To Live，生存周期），它的**值随着每经过一次路由器就会减 1，直到减到 0 时该 IP 包会被丢弃。**

此时，路由器将会发送一个 ICMP **超时消息**给发送端主机，并通知该包已被丢弃。

### IGMP

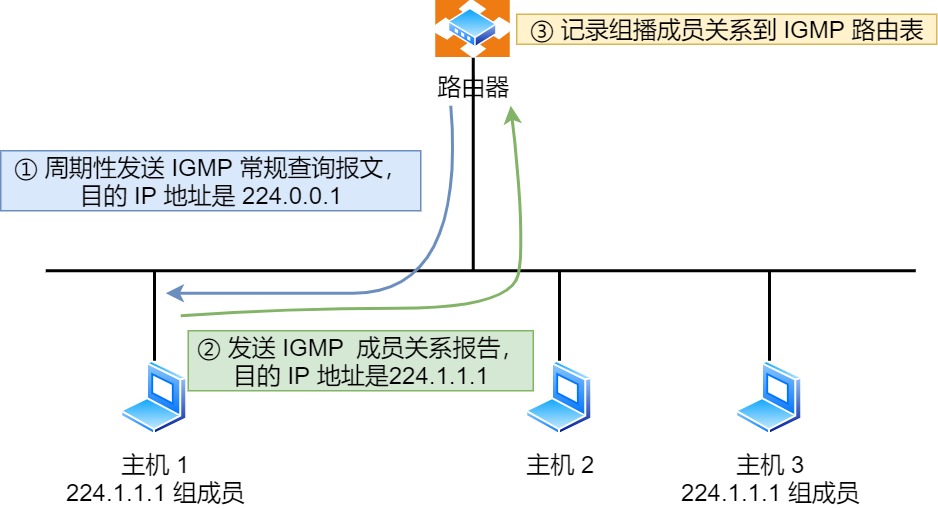
**因特网组管理协议**

**工作在主机（组播成员）和最后一跳路由之间**



* IGMP 报文**向路由器**申请加入和退出组播组，默认情况下路由器是不会转发组播包到连接中的主机，除非主机通过 IGMP 加入到组播组，主机申请加入到组播组时，路由器就会记录 **IGMP 路由器表**，路由器后续就会转发组播包到对应的主机了。
* IGMP 报文采用 IP 封装，IP 头部的协议号为 2，而且 TTL 字段值通常为 1，因为 IGMP 是工作在主机与连接的路由器之间。

以 IGMPv2 为例



1.常规查询与响应工作机制

1. 路由器会周期性发送目的地址为 224.0.0.1（表示同一网段内**所有主机和路由器**） **IGMP 常规查询报文**。
2. 主机1 和 主机 3 收到这个查询，随后会启动「报告延迟计时器」，计时器的时间是随机的，通常是 0~10 秒，计时器超时后主机就会发送 **IGMP 成员关系报告报文（**源 IP 地址为自己主机的 IP 地址，目的 IP 地址为组播地址）。**如果在定时器超时之前，收到同一个组内的其他主机发送的成员关系报告报文，则自己不再发送，这样可以减少网络中多余的 IGMP 报文数量**。
3. 路由器收到主机的成员关系报文后，就会在 IGMP 路由表中加入该组播组，后续网络中一旦该组播地址的数据到达路由器，它会把数据包转发出去。

2.离开组播组工作机制

① 情况一，网段中仍有该组播组的主机

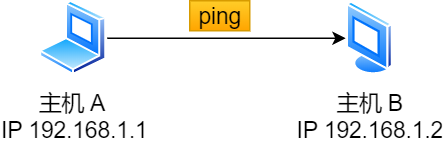
1. 主机 1 要离开组 224.1.1.1，发送 **IGMPv2 离组报文**，报文的目的地址是 224.0.0.2（表示发向网段内的**所有路由器**）
2. 路由器 收到该报文后，以 1 秒为间隔连续发送 **IGMP 特定组查询报文**（共计发送 2 个），以便确认该网络是否还有 224.1.1.1 组的其他成员。
3. 主机 3 仍然是组 224.1.1.1 的成员，因此它立即响应这个特定组查询。路由器知道该网络中仍然存在该组播组的成员，于是继续向该网络转发 224.1.1.1 的组播数据包。

② 情况二，网段中没有该组播组的主机了

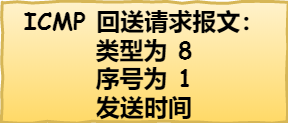
1. 主机 1 要离开组播组 224.1.1.1，发送 **IGMP 离组报文**。
2. 路由器收到该报文后，以 1 秒为间隔连续发送 **IGMP 特定组查询报文**（共计发送 2 个）。此时在该网段内，组 224.1.1.1 已经没有其他成员了，因此没有主机响应这个查询。
3. 一定时间后，路由器认为该网段中已经没有 224.1.1.1 组播组成员了，将不会再向这个网段转发该组播地址的数据包。

# 5.2 ping 的工作原理

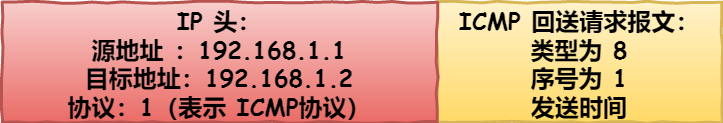
**ping 的发送和接收过程（利用ICMP查询报文类型）**



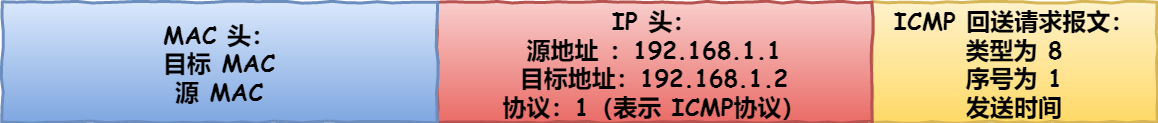
ping 命令执行的时候，源主机首先会构建一个 **ICMP 回送请求消息**数据包. 每发出一个请求数据包，**序号会自动加 1**。为了能够计算往返时间 RTT，它会在报文的数据部分插入**发送时间**。



然后，由 ICMP 协议将这个数据包连同地址 192.168.1.2 一起交给 IP 层。IP 层将以 192.168.1.2 作为**目的地址**，本机 IP 地址作为**源地址**，**协议**字段设置为 1 表示是 ICMP 协议，再加上一些其他控制信息，构建一个 IP 数据包。



接下来，需要加入 MAC 头。如果在本地 ARP 映射表中查找出 IP 地址 192.168.1.2 所对应的 MAC 地址，则可以直接使用；如果没有，则需要发送 ARP 协议查询 MAC 地址，获得 MAC 地址后，由数据链路层构建一个数据帧，目的地址是 IP 层传过来的 MAC 地址，源地址则是本机的 MAC 地址



在规定的时候间内，源主机如果没有接到 ICMP 的应答包，则说明目标主机不可达；如果接收到了 ICMP 回送响应消息，则说明目标主机可达。

## traceroute —— 差错报文类型的使用

1. traceroute 作用一

**故意设置特殊的 TTL，来追踪去往目的地时沿途经过的路由器**

原理就是利用 IP 包的**生存期限** 从 1 开始按照顺序递增的同时发送 **UDP 包**，强制接收 **ICMP 超时消息**的一种方法。比如，将 TTL 设置 为 1，则遇到第一个路由器，就牺牲了，接着返回 ICMP 差错报文网络包，类型是**时间超时**。接下来将 TTL 设置为 2，第一个路由器过了，遇到第二个路由器也牺牲了，也同时返回了 ICMP 差错报文数据包，如此往复，直到到达目的主机。

发送方如何知道发出的 UDP 包是否到达了目的主机呢

发送 UDP 包时，会填入一个**不可能的端口号**值作为 UDP 目标端口号. 然后对于每个下一个探针，它都会增加一个，这些端口都是通常认为不会被使用

目的主机，收到 UDP 包后，会返回 ICMP 差错报文消息，差错报文消息的类型是「**端口不可达**」。

2. traceroute 作用二

**故意设置不分片，从而确定路径的 MTU**。

首先在发送端主机发送 IP 数据报时，将 IP 包首部的**分片禁止标志位设置为 1**。根据这个标志位，途中的路由器不会对大数据包进行分片，而是将包丢弃。

随后，通过一个 ICMP 的不可达消息将**数据链路上 MTU 的值**一起给发送主机，不可达消息的类型为「**需要进行分片但设置了不分片位**」。

发送主机端每次收到 ICMP 差错报文时就**减少**包的大小，以此来定位一个合适的 MTU 值，以便能到达目标主机。

# 5.3 断网了，还能 ping 通 127.0.0.1 吗？

**可以**

如果把 127.0.0.1 换成 0.0.0.0 或 localhost 会怎么样呢

## 什么是127.0.0.1

**这是个 IPV4 地址**

**127 开头的都属于回环地址**

## 为什么断网了还能 ping 通 127.0.0.1

有网的情况下，ping 最后是**通过网卡**将数据发送出去的

当发现**目标IP是外网IP**时，会从"真网卡"发出。

当发现**目标IP是回环地址**时，就会选择**本地网卡**。

**本地网卡是一个“假网卡“，会把数据推到一个叫 input\_pkt\_queue 的 链表 中。这个链表，其实是所有网卡共享的，上面挂着发给本机的各种消息。消息被发送到这个链表后，会再触发一个软中断**。

ping 回环地址和**通过TCP等各种协议发送数据到回环地址**都是走这条路径。整条路径从发到收，都没有经过"真网卡"

ping 本机IP 跟 ping 回环地址完全一样

## 127.0.0.1 和 localhost 以及 0.0.0.0 有区别吗

首先 localhost 就不叫 IP，它是一个域名，就跟 "baidu.com",是一个形式的东西，只不过默认会把它解析为 127.0.0.1 ，当然这可以在 /etc/hosts 文件下进行修改。

其次就是 0.0.0.0，执行 ping 0.0.0.0 ，是会**失败**的，因为它在IPV4中表示的是无效的**目标地址**。

 0.0.0.0的作用：启动服务器的时候，一般会 listen 一个 IP 和端口，等待客户端的连接。

如果此时 listen 的是本机的 0.0.0.0 , 那么它表示本机上的**所有IPV4地址（包括本机IP和回环地址）**。