

## Links

- [TCP/IP 网络模型有哪几层?](#)
- [进程间有哪些通信方式? | 小林coding](#)

对于同一台设备上的进程进行通信，可以采用管道，消息队列，共享内存，信号等方式进行。

但是对于不同设备之间进行进程通信，就需要网络通信，而且因为设备的多样性，需要对各式设备进行兼容，因此需要一套**通用的网络协议**。

网络协议是分层的，每一层都有自己相应的职责，接下来根据「TCP/IP 网络模型」分别对每一层进行介绍。

## 应用层

TCP/IP网络模型的最上层就是**应用层**（*Application Layer*），也是我们最直接接触到的一层。我们手机或电脑等使用的应用软件都在应用层实现。那么，当两个不同设备的应用间需要通信时，应用需要把数据传给下一层，即传输层。

应用层只需要专注于为用户提供应用功能，比如 HTTP、FTP、[Telnet](#)、DNS、SMTP等。

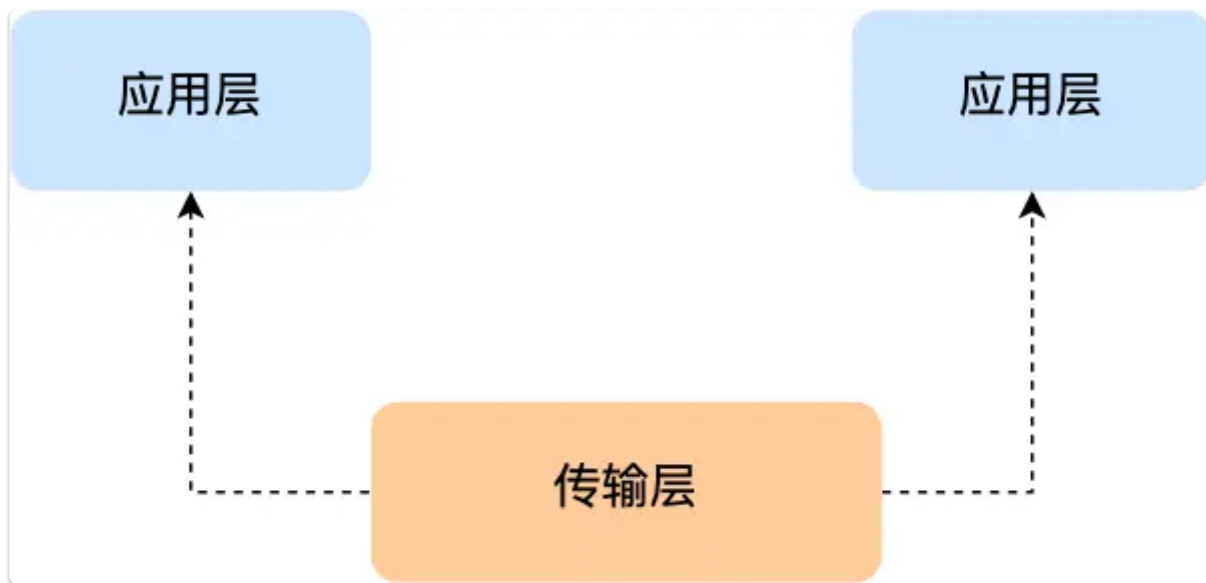
应用层不会关心数据是如何向下传输，如同日常生活当中我们不会去关心包裹📦如何被运输，我们只需要把📦发给快递员即可。

## Info

应用层是工作在操作系统当中的用户态，传输层工作在操作系统的核心态。

## 传输层

应用层的数据包会传给传输层，**传输层**（*Transport Layer*）是为应用层的服务提供网络支持。

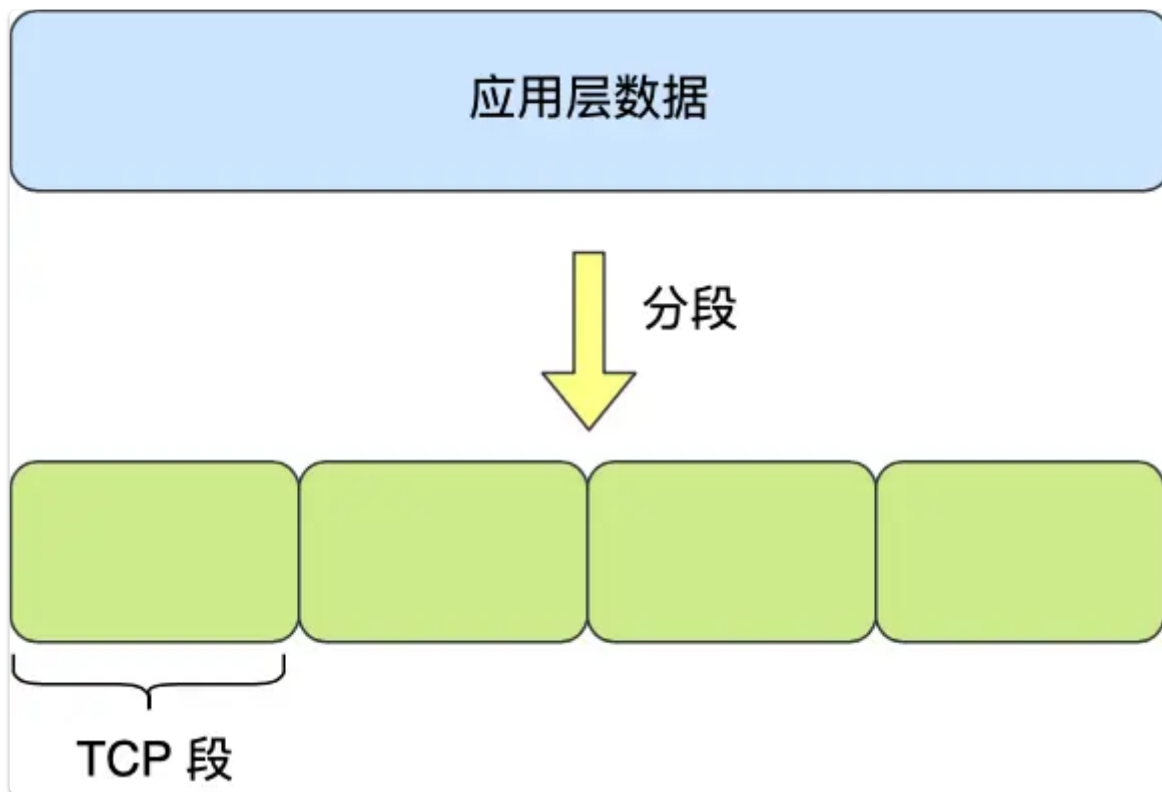


传输层主要有两个传输协议，分别是 TCP 和 UDP。

TCP协议的全称被称为**传输控制协议** (*Transmission Control Protocol*) ,大部分应用使用的正是TCP传输层协议，比如 HTTP 应用层协议。TCP 与 UDP 相比，多了流量控制、超时重传、拥塞控制等特性，这些都可以保证数据包可靠的传输给对方。

UDP 相对来说比较简单，简单到只负责发送数据包，不保证数据包是否能够准确的到达对方，但是它的实时性较高，传输效率较高。如果想要实现的 UDP 的可靠传输，需要把 TCP 协议的相关特性在应用层进行实现，所以总体来说比较困难。

应用层需要传输的数据可能非常大，如果直接传输可能会不好控制，因此当传输层的数据包超过 MSS (TCP最大报文长度时) ,需要将数据包进行分块，这样即使分块在传输的过程中丢失，也只需要发送丢失的分块，而不是重新发送整个数据包📦。在 TCP 协议中，我们把每个分块称为 **TCP段** (*TCP Segment*)



当设备作为接收方，传输层需要将数据包传给应用层，但是一台设备可能存在多个应用在接收或者传输数据，因此需要一个编号将这些应用进行区分，这个编号就是**端口**。

比如 80 端口通常就是 Web 服务器，22 端口用来进行远程登录服务。同样的，每个浏览器的标签栏都相当于一个独立的进程，操作系统会分配临时的端口号。

由于传输层的报文会携带端口号，因此接收方可以识别出这些报文是传给哪些应用。

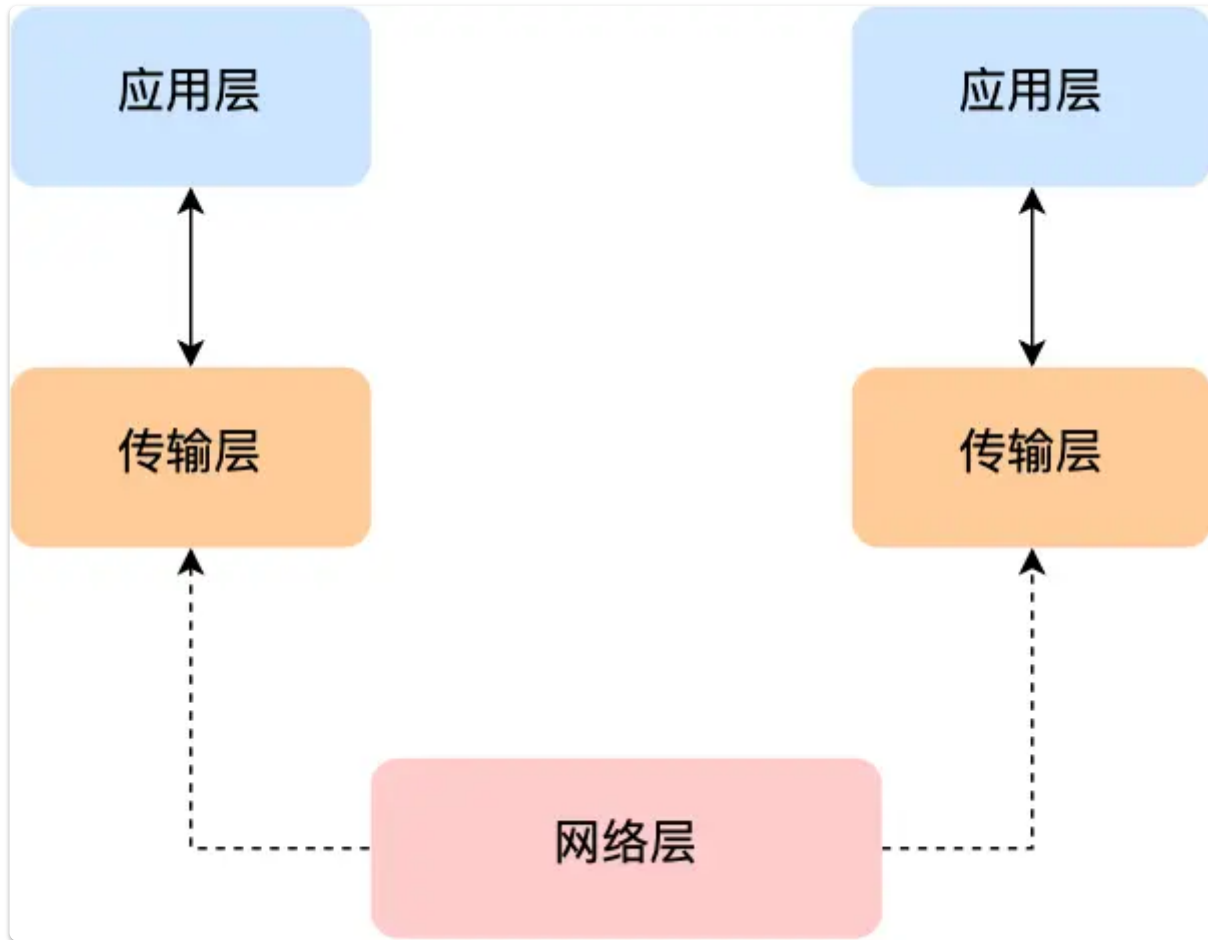
### Example

- 22 远程登录服务
- 80 HTTP 超文本传输协议 Web服务器应用
- 443 HTTPS

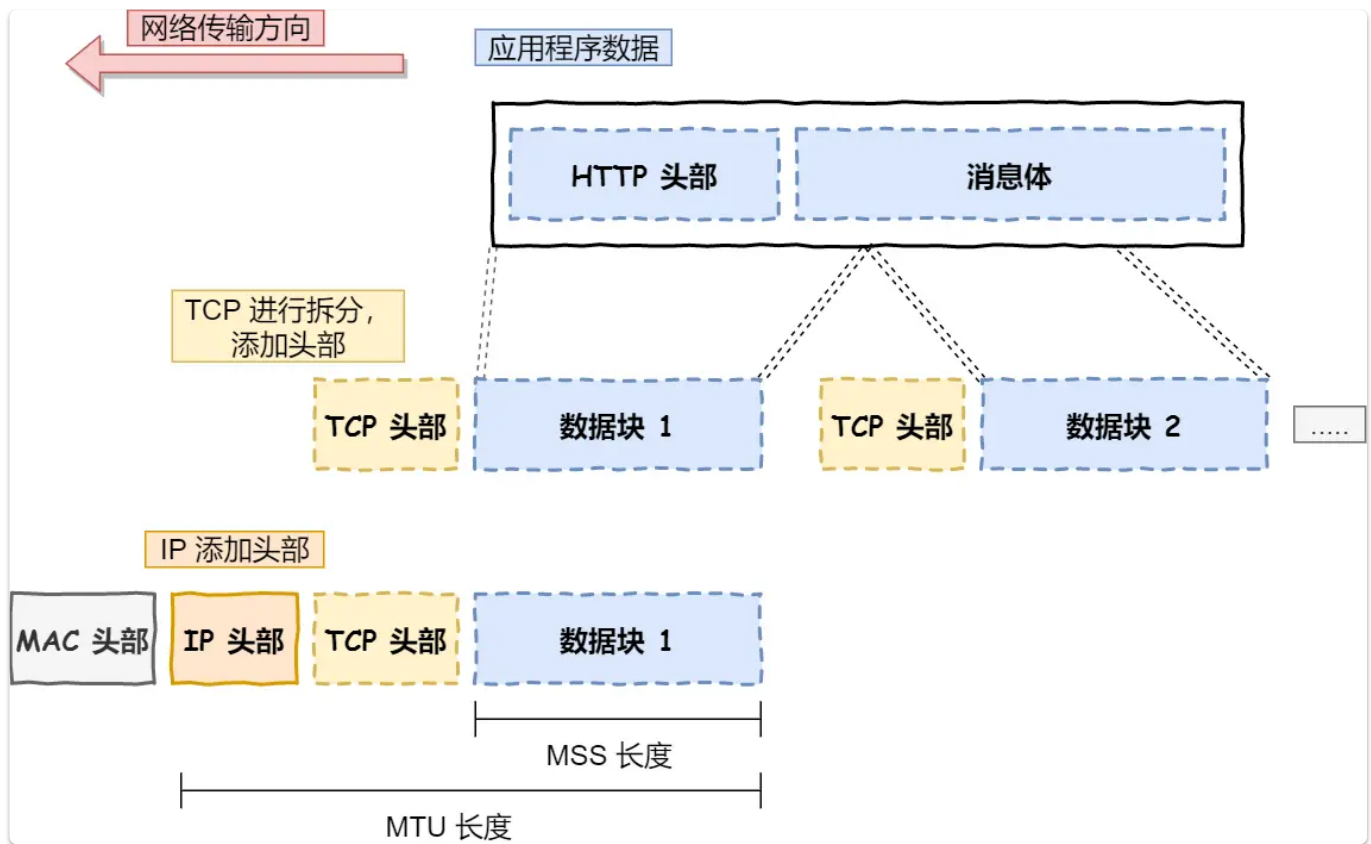
## 网络层

传输层并不负责数据从一个设备传输到另一个设备。在实际的网络环节中是错综复杂的，中间有各种各样的网络选择和分叉路口，如果一个设备的数据需要传输给另一个设备，就需要对各式各样的路径和节点进行选择。

因此，我们不希望传输层过于复杂，只需要服务号应用层即可，让其作为应用之间传输的媒介，帮助实现应用与应用之间的通信，实现的数据传输功能交给了**网络层** (*Internet Layer*)



网络层最常使用的协议就是 **IP 协议** (*Internet Protocol*)，IP 协议会把传输层的报文作为数据部分，再加上 IP 包头组装成 IP 报文，如果 IP 报文大小超过 **MTU**，就会再次进行**分片处理**，得到一个即将发送网络的 IP 报文。



网络层负责将数据从一个设备传输到另一个设备，世界上的电子设备如此多，因此需要区分设备的编号进行查找。

我们一般采用 IP 地址给设备进行编号，对于 [IPv4协议](#)，IP 地址共 32 位，分成四段，每段 8 位，只有一个单纯的 IP 地址对设备进行区分，但是在寻址就会十分麻烦，因此还需要把 IP 地址继续划分。

因此，一般的 IP 地址分成两个意义：

- 一个是网络号，负责标识该 IP 地址属于哪个子网的；
- 一个是主机号，负责标识同一个子网下的不同主机；

需要[子网掩码](#)才可以计算 IP 地址 的网络号和主机号。

IP 地址 : 10.100.122.2

00001010 1100100 1111010 00000010

子网掩码 : 255.255.255.0

11111111 11111111 11111111 00000000



IP 地址 和 子网掩码  
做 AND 运算

网络号 : 10.100.122.0

00001010 1100100 1111010

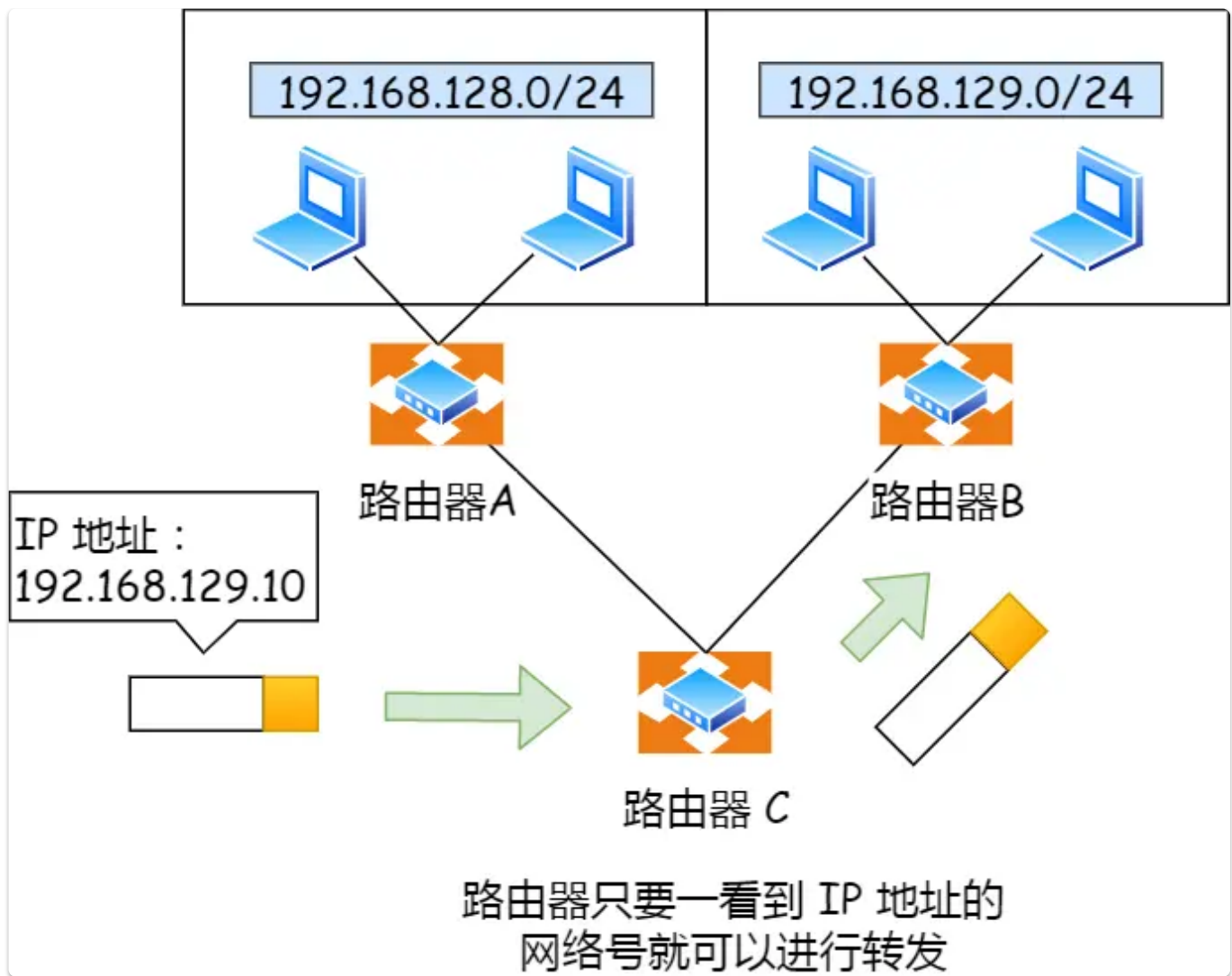
00000000

网络号

那么在寻址的过程中，会匹配相同的网络号，再去寻找对应的主机🖥️:。

除了寻址能力之外，IP 协议还有一个重要的能力就是**路由**。实际场景中，两台设备之间不会根据一条网线连接，可能中间需要很多网关、路由器、交换机等众多网络设备连接起来，那么形成很多条网络的路径，因此数据包到达下一个网络节点，需要通过**路由算法**决定走哪一步路径。

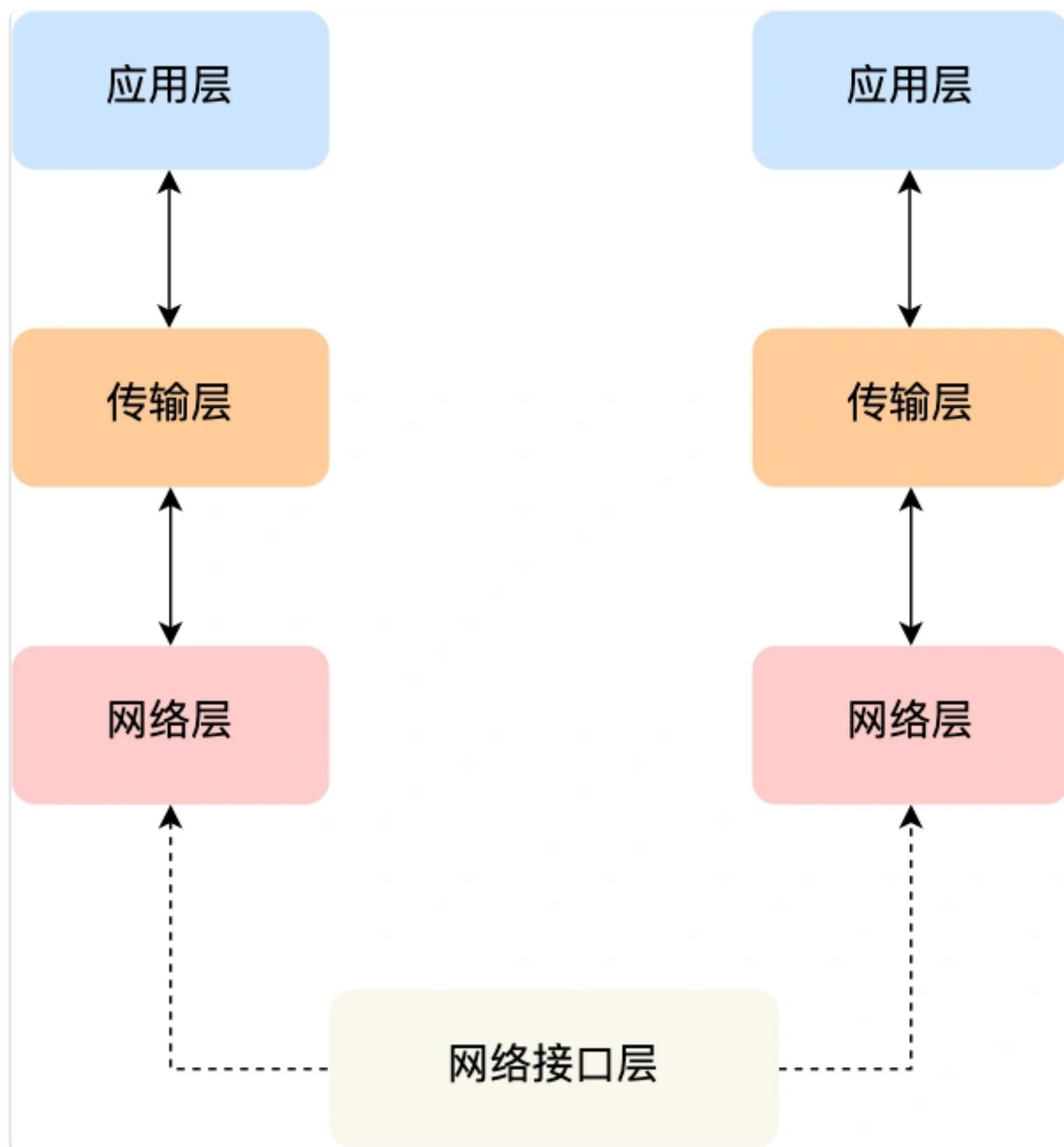
路由寻址工作中，需要先找到目标地址的子网，找到子网后进而把数据分给对应的主机🖥️。



所以，IP 协议的寻址作用就是告诉我们去往下一个目的地该朝哪个方向走，路由根据下一个目的地选择相关的路径。寻址像是导航📍，路由更像操作方向盘🚗。

## 网络接口层

生成 IP 头部之后，接下来会交给**网络接口层**（*Link Layer*）在 IP 头部加上一个 MAC 头部，并且封装成数据帧分发到网络上。



IP 头部中接收方 IP 地址标识网络包到达的地方，通过这个地址我们可以判断这个包发到哪里，但在[以太网](#)的世界中，这个思路无法实行。

以太网是指由现实生活中的以太网接口，Wi-Fi接口，以太网交换机，路由器上的以太网接口，网线组成。以太网是将[局域网](#)内的附近设备连接起来，进行通讯的技术。

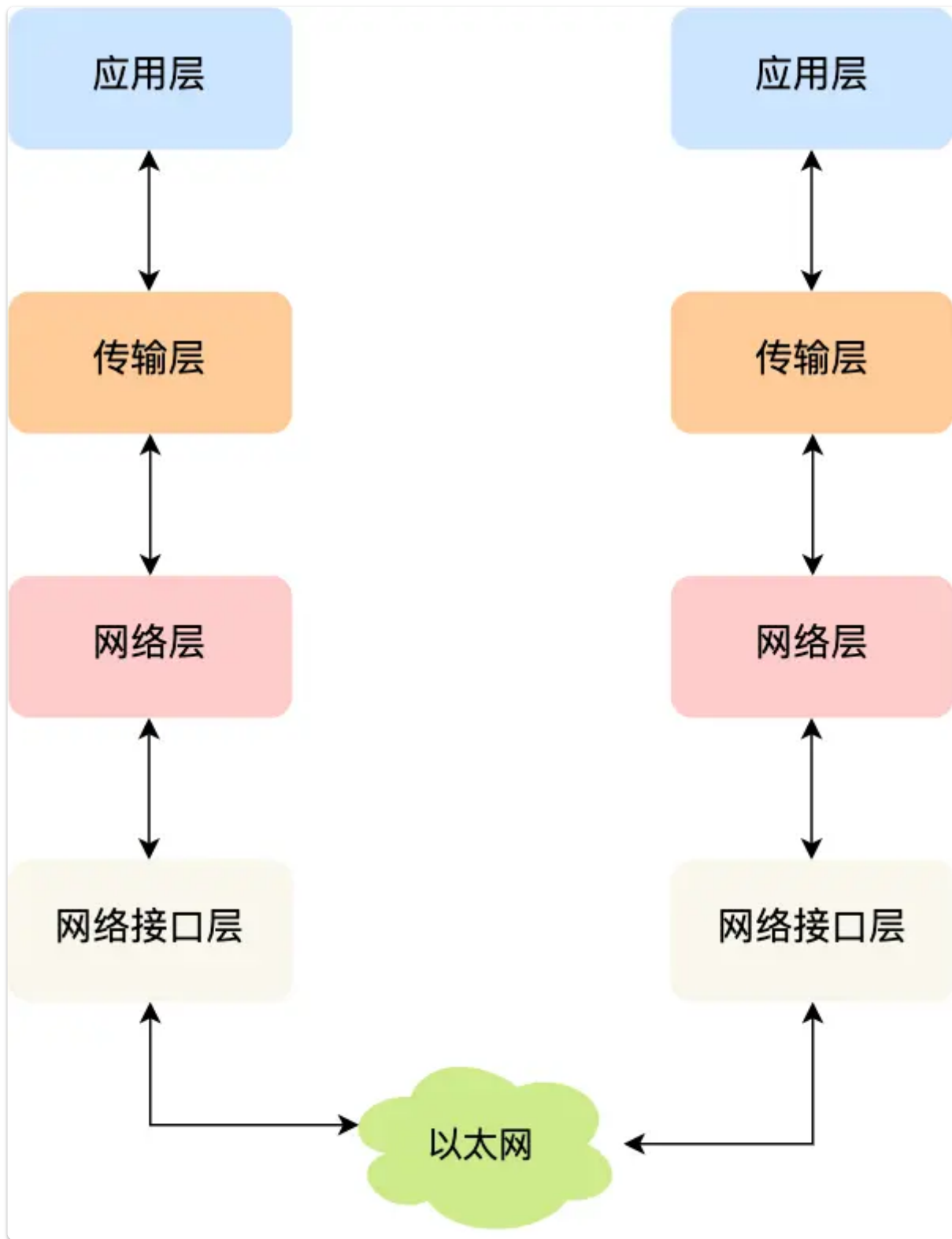
在以太网之间进行通讯需要用到 MAC 地址。MAC 头部是以太网使用的头部，包含接收方和发送方的 MAC 地址信息，我们可以通过[ARP协议](#)获取对方的 MAC 地址。

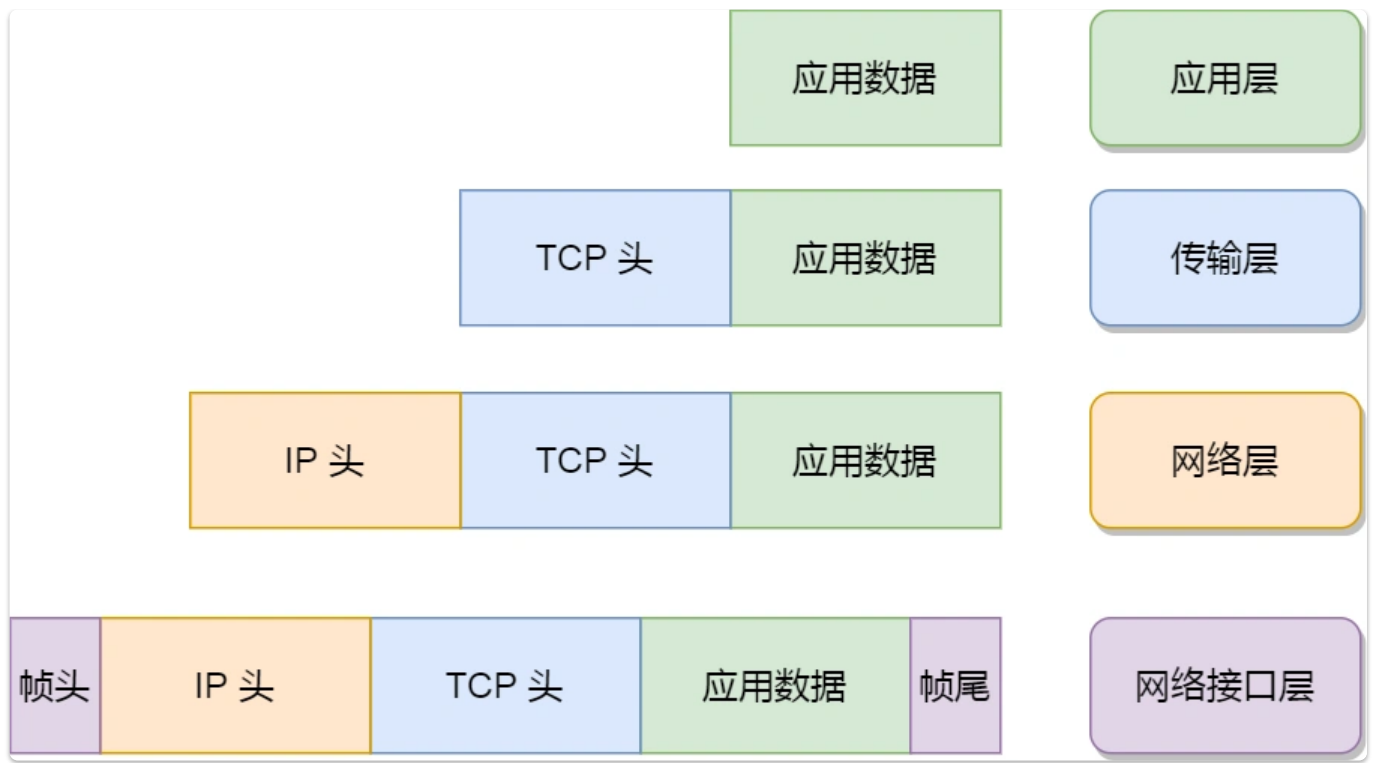
所以说，网络接口层主要为网络层提供链路级别的传输服务，负责在以太网、WiFi 这样的底层网络上发送原始数据包，工作在网卡这个层次，使用 MAC 地址标识网络上的设备。

## 总结



TCP/IP 网络模型一共分为四层，分别是应用层，传输层，网络层和网络接口层。





网络接口层传输单位是帧，网络层传输单位是包，传输层传输单位是段，应用层传输单位是消息或报文。都可以统称为数据包。