

高频考点

输入网址之后发生哪些事情？

举例：我们输入 www.baidu.com 之后发生了哪些事情？

1. HTTP

第一步：解析URL

首先，我们是在浏览器中输入网址的，那么浏览器首先需要对URL进行解析，然后生成发送给Web服务器的请求信息

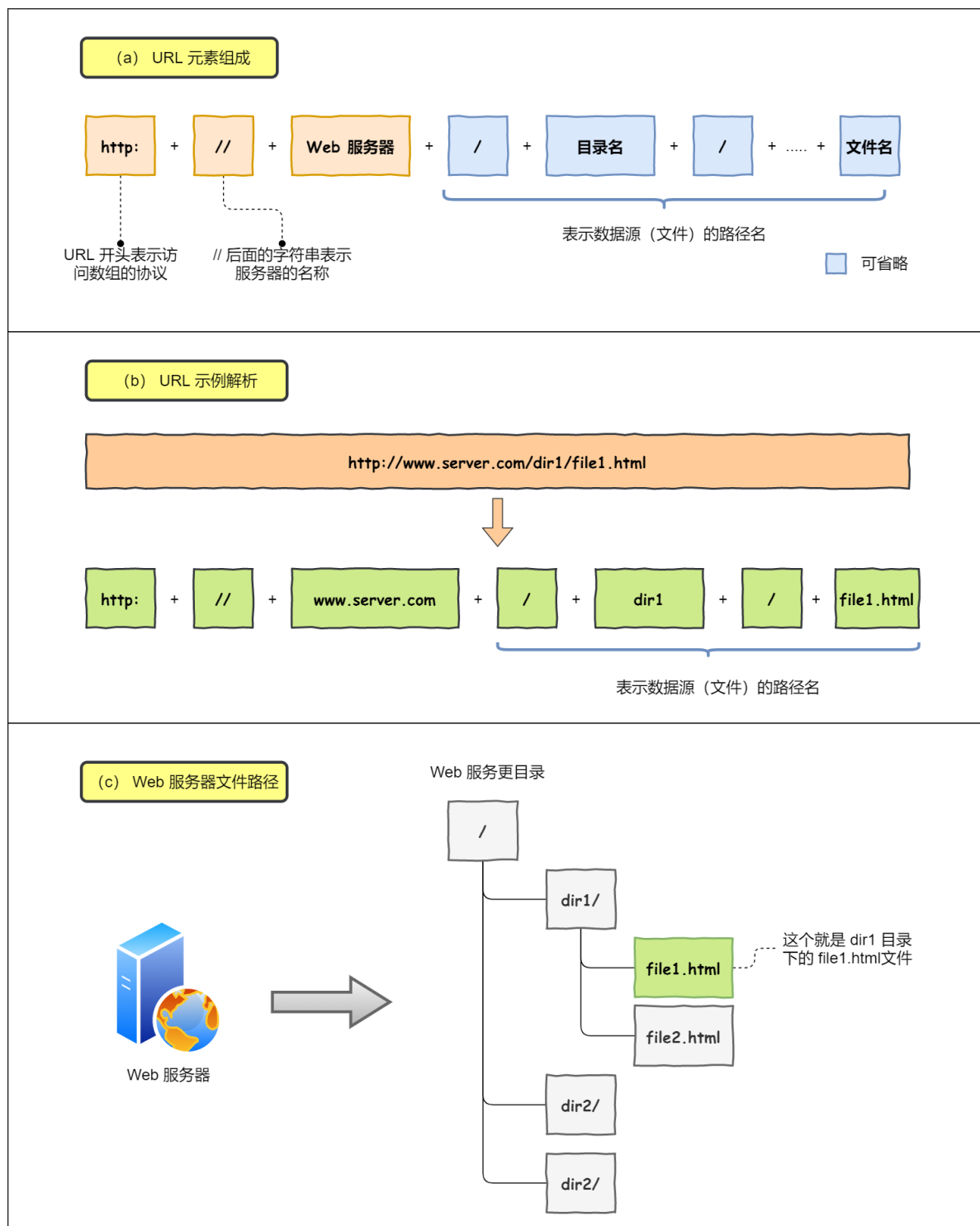
先来看一下，输入的URL的各个元素代表的含义：

<https://www.baidu.com/index.html>

- `https:` 表示访问中遵循的协议
- `//`：表示后面的字符串是目标服务器
- `www.baidu.com`：是紧跟在`//`后面的，所以表示的是需要请求访问的Web服务器
- `/index.html`：可缺省的，表示该服务器上的数据源的路径名称，即该服务器中的文件路径，类似于 `c:\users\admin\Desktop....`

——先找到web服务器，然后在该服务器的路径中找到对应的文件

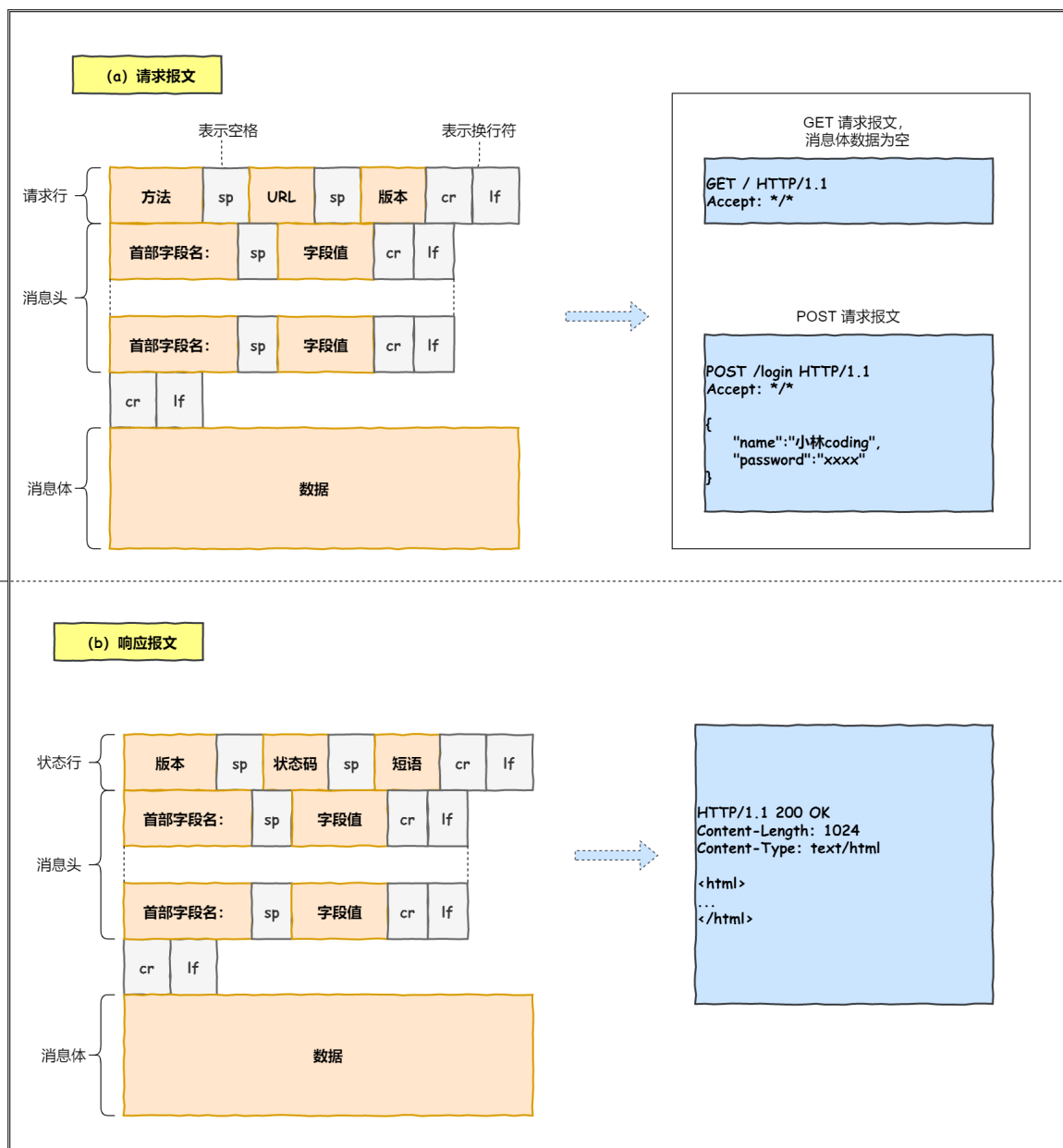
如果该部分是被省略的，eg: <https://www.baidu.com>，代表访问根目录下的事先设置的默认文件，一般是 `/index.html` 或者 `/default.html`



第二步：生成HTTP请求信息

在第一步中对URL进行解析之后，确定了访问需要遵循的协议、要访问的服务器的地址和文件路径，那么需要根据这两部分生成HTTP请求信息：

然后组合成了HTTP报文，发向网络



2. DNS

第三步：查询服务器对应的IP地址

在发送HTTP报文之前，需要知道**该服务器域名所对应的IP地址**，∴委托操作系统发送消息，必须要提供通信对象的IP地址

用到之前的DNS技术

DNS服务器：专门保存了Web服务器域名和IP的对应关系

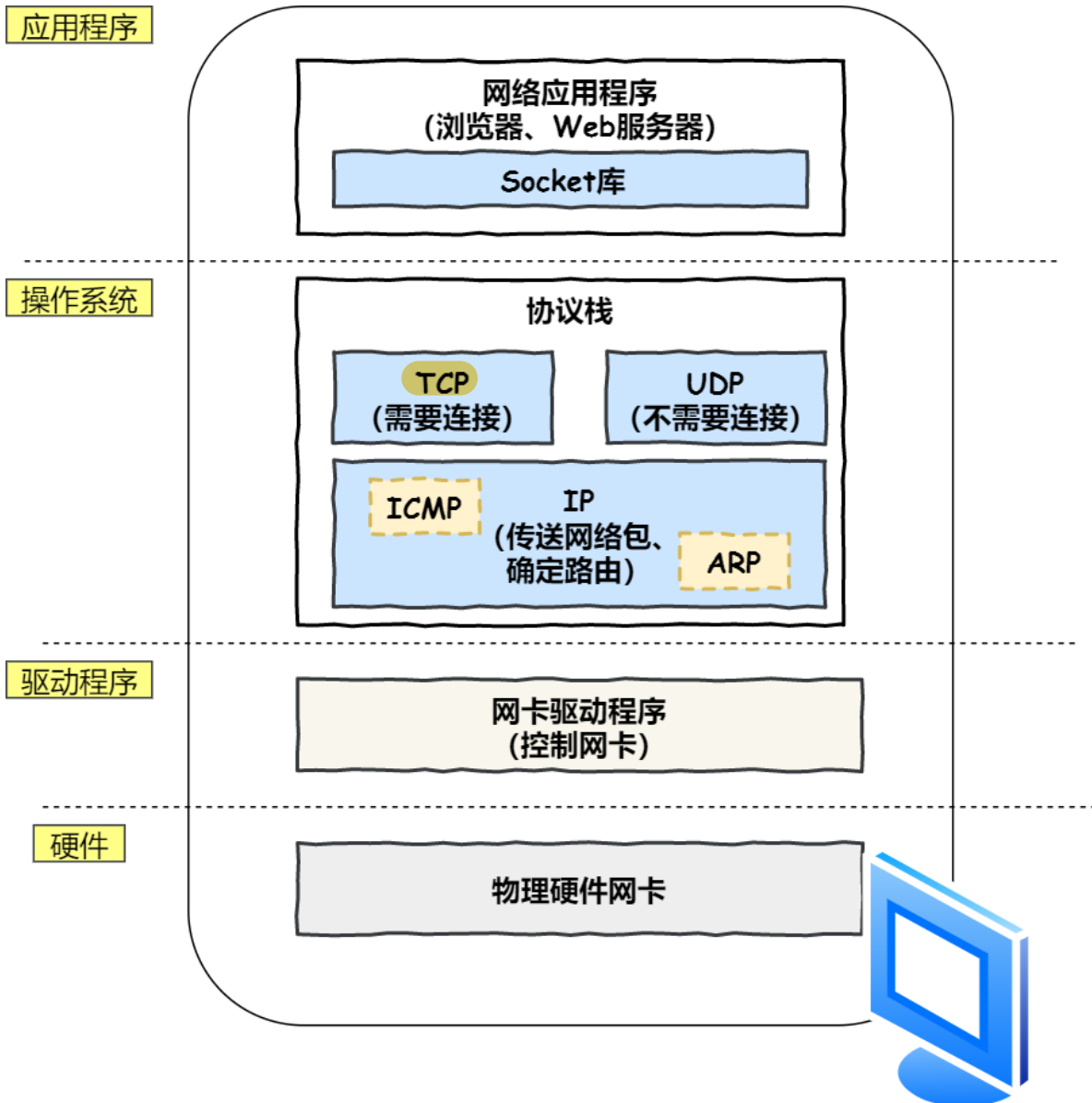
客户端会向本地DNS请求查询网址对应的IP地址（本地DNS服务器是客户端的TCP/IP中配置的，传递信息的方式就是IP数据报），而本地DNS会向根域DNS、顶级域DNS、权威DNS查询需要的网址对应的IP地址，直到找到之后返回给客户端

3. 协议栈

第四步：HTTP的传输工作会交给操作系统中的协议栈完成

已知了目的IP，那么交给协议栈工作了

协议栈内部分为几个部分，分别承担不同的工作。**部分之间是有上下级关系的**，上面的会向下面委托工作，下面接收委托工作并执行——类似于OSI模型的等级



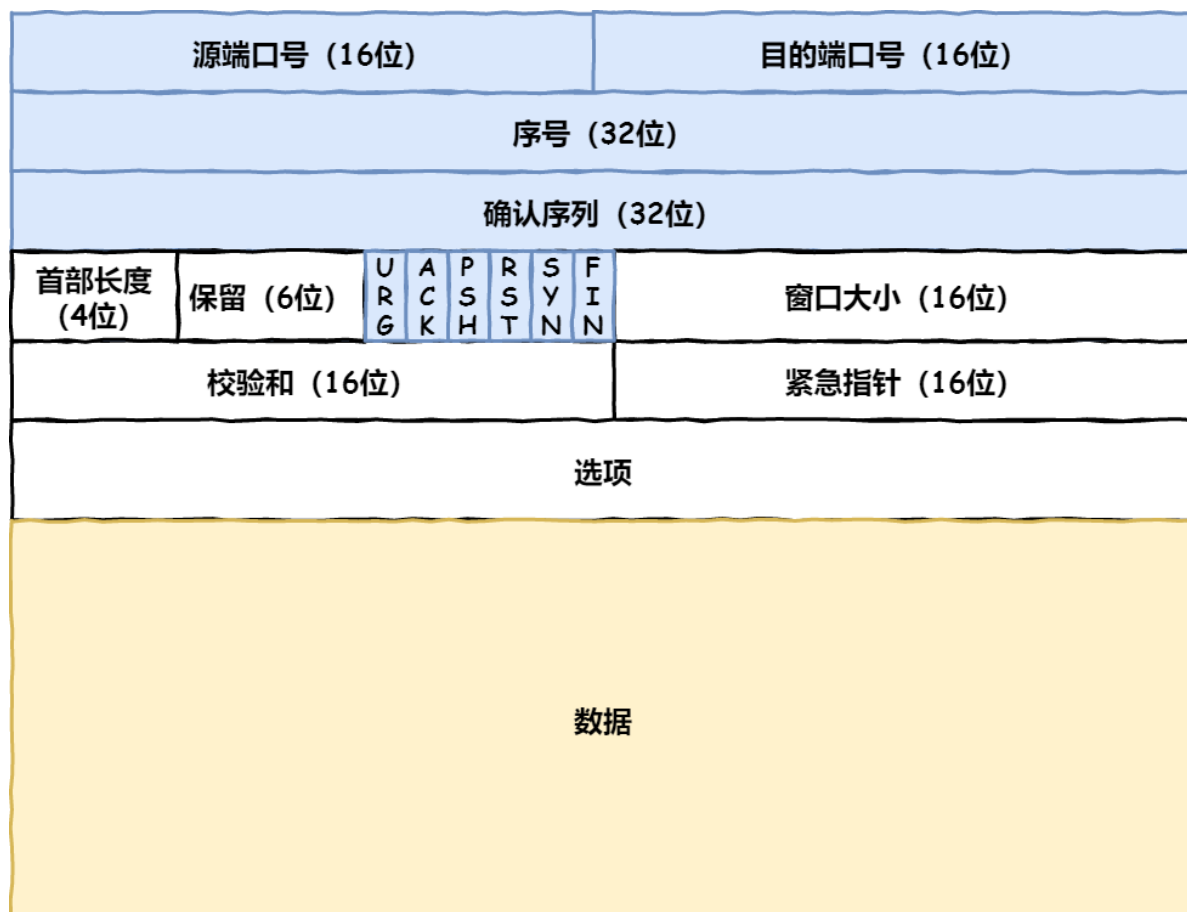
1. 浏览器通过调用socket库，来委托协议栈工作
2. 协议栈上半部分是**TCP/UDP协议**，两者有功能上的区别，前者是一对一连接后才能收发数据，后者是不需连接多端收发，主要功能都是：**接收应用层的委托，执行收发数据的操作**，收发的是TCP/UDP数据报
3. 协议栈下半部分是**IP协议**，用来控制网络包的收发操作，收发的是IP数据报
并且，IP中还包含了ICMP和ARP协议
 - ICMP：在出现传输故障的时候，负责通知故障原因和情况，以及各种控制信息
 - ARP：根据IP地址查询MAC地址
4. 协议栈下面是**网卡驱动程序**，主要是用来控制网卡硬件
5. 最下面是网卡，主要是负责完成实际的收发工作，对网线中的信号执行收发操作

4. TCP（协议栈里面的）

第4.1步：HTTP数据报通过TCP层

HTTP是基于TCP协议传输的

TCP的报文：（具体功能见TCP那份思维导图）



TCP是面向连接的、可靠的、基于字节流的传输层通信协议，能够保证可靠的有序的传输。并且，还能做到流量控制（窗口大小）、拥塞控制（自我控制传输的速度）

5. IP（协议栈里面的）

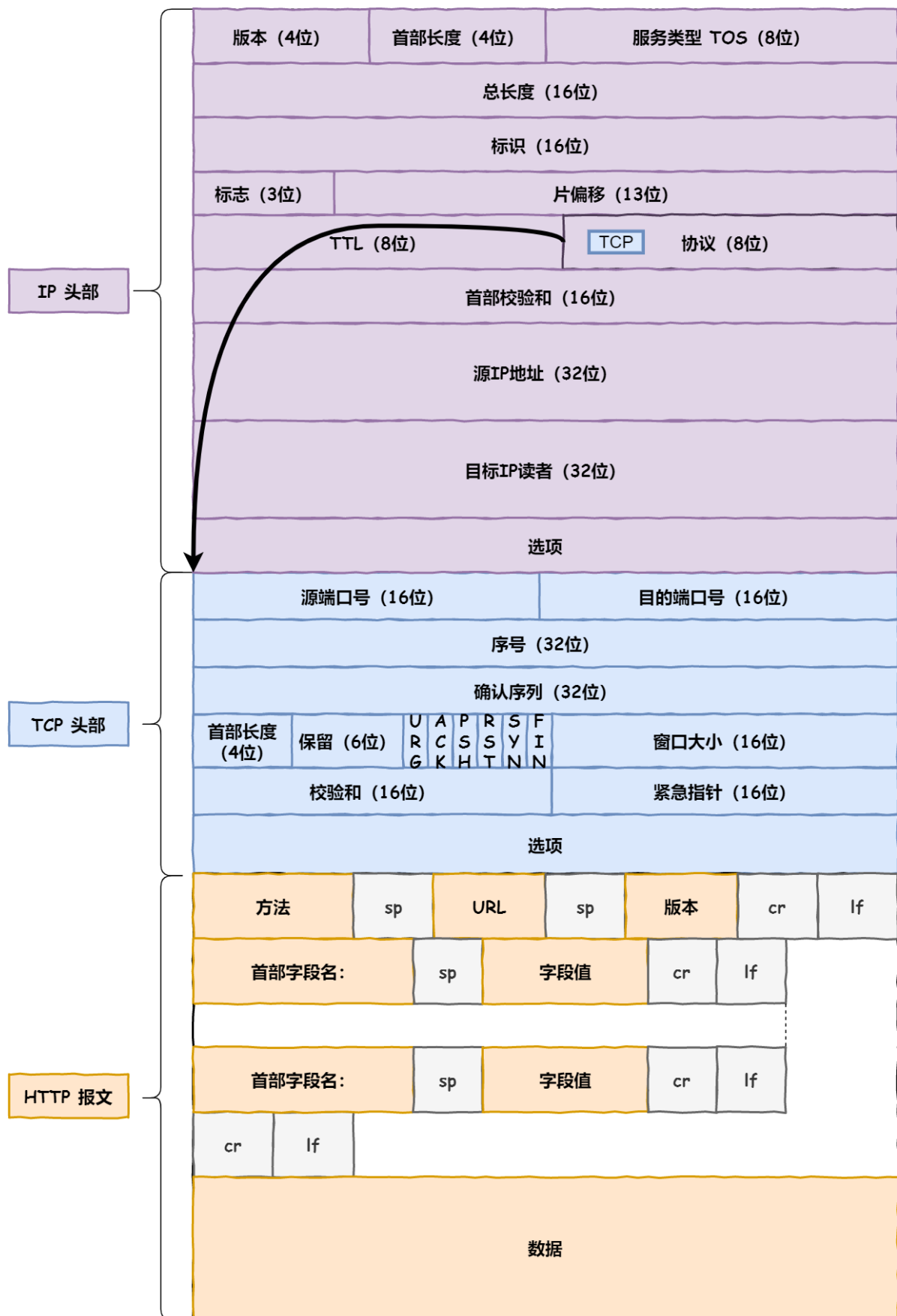
第4.2步：TCP数据报包装IP首部，并且选择合适的网卡（即填入合适的源IP地址）

主要是：主机和主机之间的传输了（TCP类似于应用到应用之间的传输），即点到点——点之间可能不直接相连的

IP层是TCP层的下一层，即网络层的操作。（具体见IP那份思维导图）

通过IP层的包装，那么能够知道从哪个IP地址出来，到哪个IP地址去

IP层的报文：



6. MAC

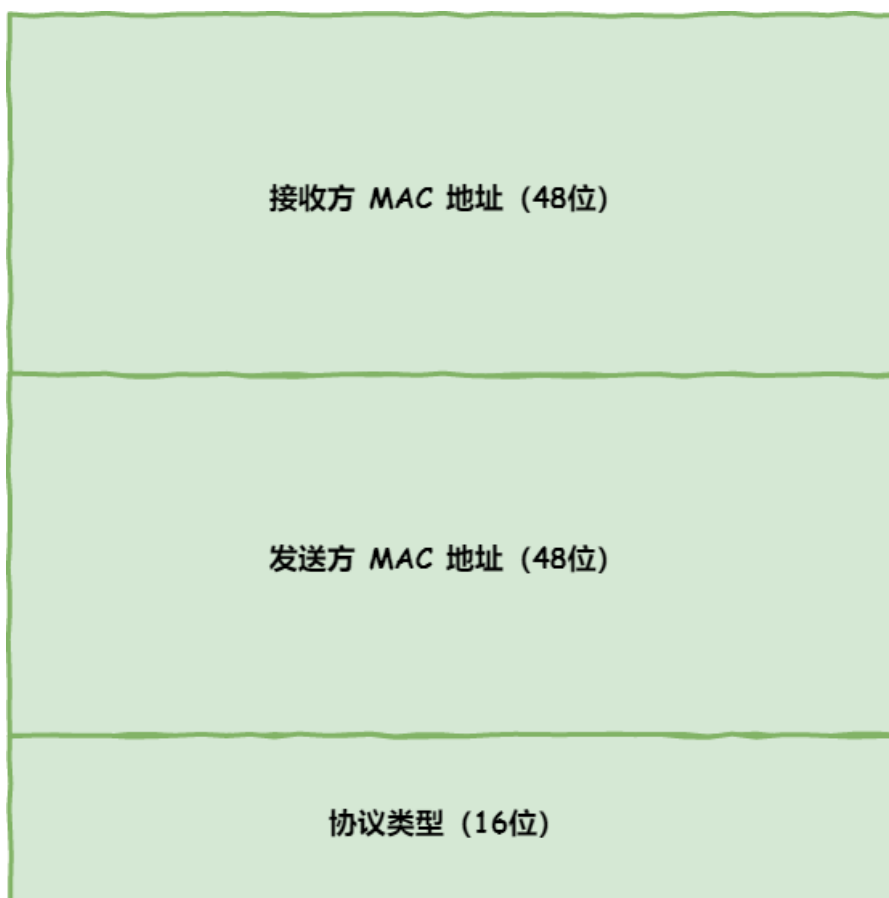
第4.3步: 根据IP地址获取MAC地址, 并包装IP数据报

IP数据报还需要在IP数据报外面再包一层**MAC头部**

该层主要的作用: IP层知道目的地是哪里, 而MAC层就是一步步指示如何到目的地, 即**下一步要去哪里**。

- IP层：没有直连的两个主机之间通信
- MAC层：直连的两个主机之间传输

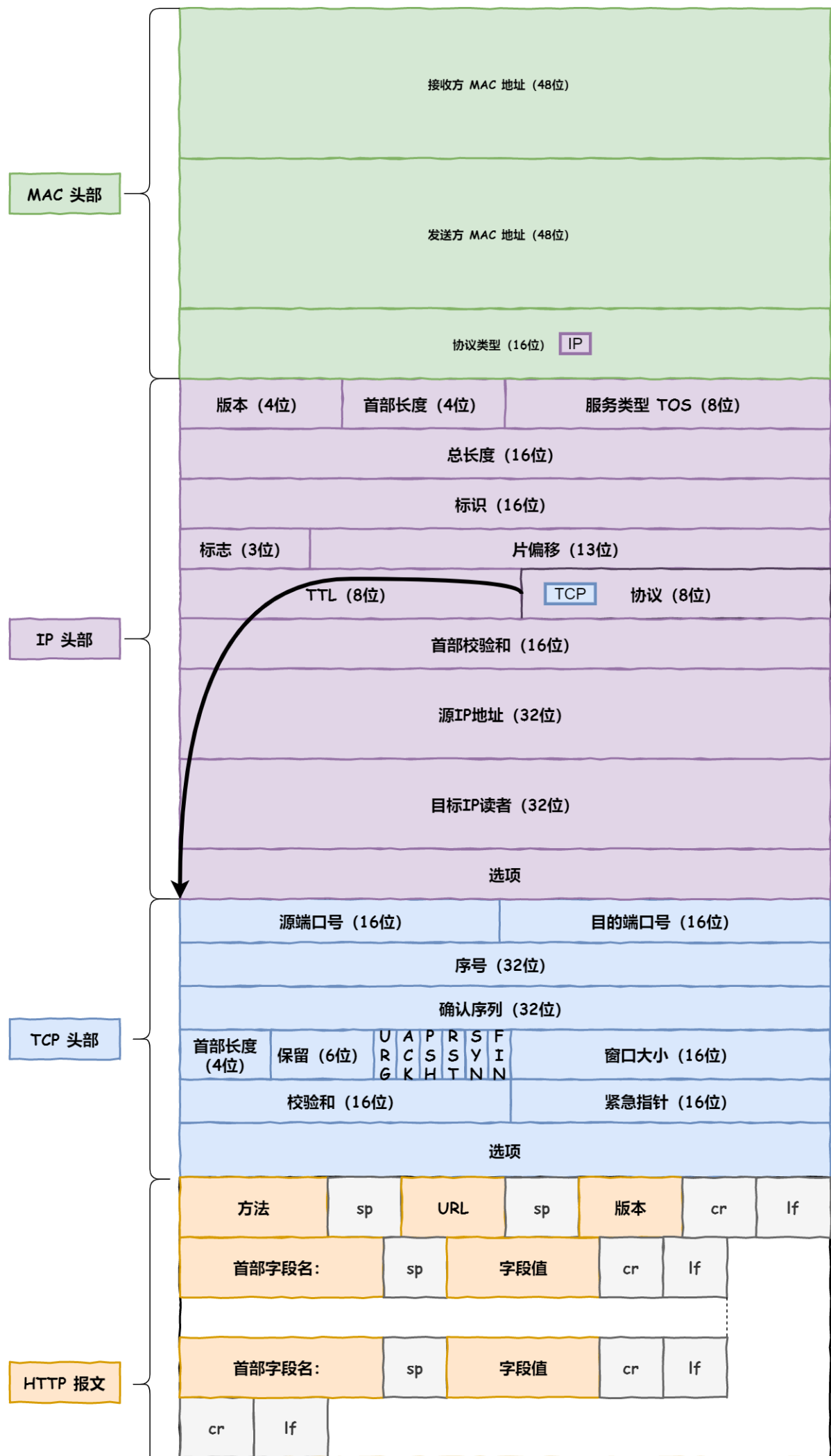
MAC的包头格式：



- 接收方的MAC地址 / 发送方的MAC地址：用于直连的两地传输
 - 发送方的MAC地址：**网卡在生产时已经写入ROM中**，那么只需要从网卡中读取出来写入到发送方的位置中即可

ps：一个主机可能有多个网卡，已经在IP层确定了网卡，即IP地址，那么MAC地址也就相应确定了
 - 接收方的MAC地址：稍微复杂，但是一旦知道接收方的MAC地址，以太网会根据MAC地址将包发送过去
 - 首先：查询路由表，通过目的IP地址，查询最匹配的条目，然后将包发送给对应的IP地址即可——这边的IP就是和该主机直连的
 - 知道该IP地址后，还需要知道MAC地址，通过ARP协议查找，通过一个链路中的所有设备广播，然后符合的主机进行响应，返回的响应报文中塞入MAC地址
 - ps：OS会将该MAC地址放在**ARP缓存**中，但是缓存的时间在几分钟，所以实际上根据IP地址查询MAC地址
 - 1. 先去找ARP缓存中是否存在该条，存在就不需要ARP广播查询了，直接使用即可
 - 2. 当ARP缓存中不存在，那么需要ARP广播查询
- 协议类型：对于TCP/IP填写的是：**0800——IP协议/0806——ARP协议**

最后生成MAC报文：



数据

7. 网卡

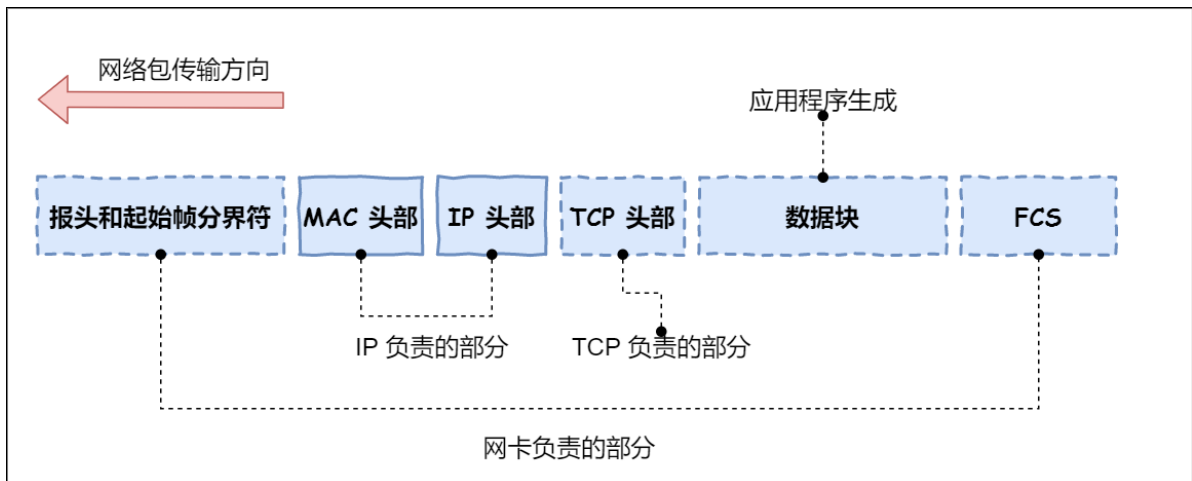
到前面步骤为止，报文的内容已经确定下来了，当前是一串二进制的数字信息，但是无法直接在网络上传输，需要将**数字信号转换为电信号**，才能在网线上传输

将数字信号转换为电信号的操作是**通过网卡**，而控制网卡需要**网卡驱动程序**。

第5步：数字信号转电信号

网卡驱动获得IP最后的数据包之后，将其复制到网卡的缓存区，然后再在开头加上**报头，和起始帧分界符**（用来标记包的起始为止，区分不同的帧），在末尾加上用于**检测错误的帧校验序列**（检查包传输过程中是否发生损坏）

最后在网线上传输的数据包就是如下结构的：



8. 交换机

交换机的概念：

- 功能：将网络包原样转发到目的地
- 工作位置：**MAC层**，所以又被称为**二层网络设备**

交换机的包接收操作：

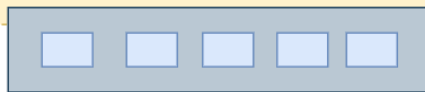
1. 电信号到达了网线接口（在网线上传输的都是电信号），交换机里的模块进行接收，并且将电信号转换为数字信号
2. 网络包变成了数字模式，然后通过包末尾的FCS检查包是否发生损坏，如果没问题，那么就放在缓冲区

ps：网卡和交换机存在区别：**网卡是存在MAC地址的**，所以在接收到包之后会先验证该包的目的地MAC地址是否为自己，如果是就接收；如果不是，就直接丢弃；**交换机不存在MAC地址**，所以对所有经过它的网络包都统一接收并放入缓存

- 3. 包进入缓存后，会等待，当轮到它执行时，会先查询该包的接收方MAC地址是否已经在**MAC地址表**中

交换机内部有一张 MAC 地址与网线端口的映射表。
当接收到包时，会将相应的端口号和发送 MAC 地址写入表中，
这样就可以根据地址判断出该设备连接在哪个端口上了。
交换机就是根据这些信息判断应该把包转发到哪里的。

MAC 地址表	端口	控制信息
00-60-97-A5-43-3C	1	...
00-00-C0-16-AE-FD	2	...
00-02-B3-1C-9C-F9	3	...
....



交换机

即可以查询该表格，将对应MAC地址的网络包，并将信号发送到对应的端口

如果目的MAC地址在表格上不存在：

- 原因：该MAC地址的设备还未向该交换机发送过网络包 / 该设备一段时间内未工作所以被从地址表中删除了
- 操作：如果找不到，那就广播，将包发送给除源端口以外的所有端口，所有连接该交换机上的设备都能接收到该包——但是只有目标MAC地址的主机会接收，其他均忽略了
- 影响：广播，不会造成网络拥塞。因为目标设备收到该包之后会做出响应，交换机收到该包之后就会在MAC地址表中增加该条，那么下次就需要广播了
- ps：如果MAC地址是一个广播地址，那么需要将该包发送到除源端口以外的所有端口，广播地址有：MAC地址的FF:FF:FF:FF:FF:FF，对应IP地址的255.255.255.255

9. 路由器

网络包经过交换机后到达了路由器，然后通过路由器到达下一个路由器/目标地址

路由器与交换机的异同：

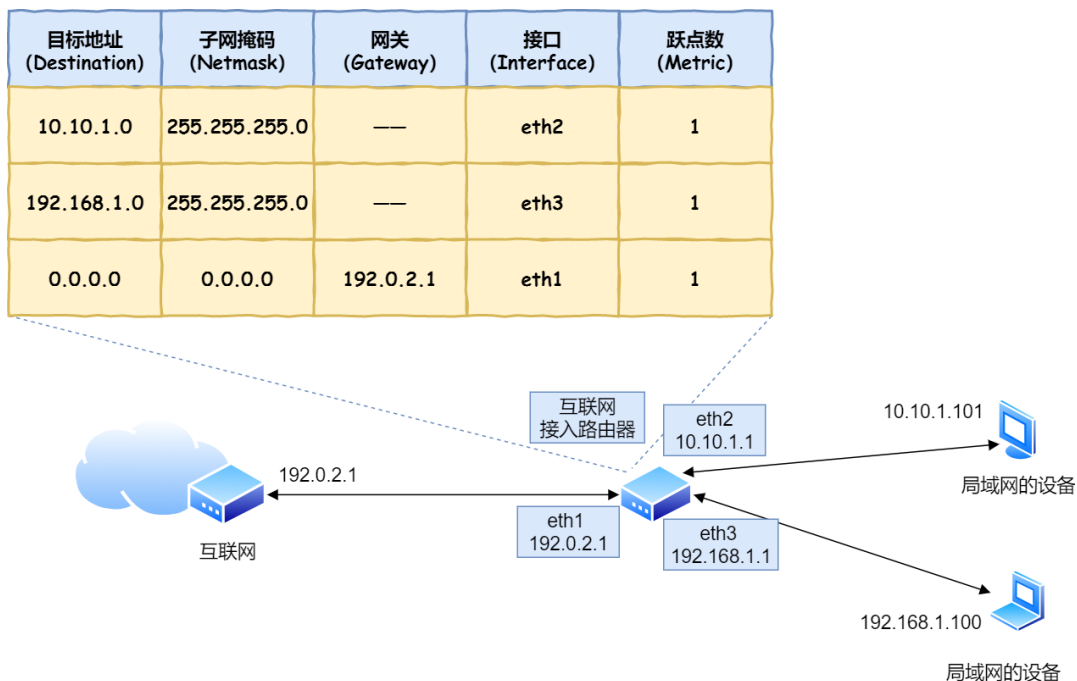
- 异：
 - 路由器是IP层的，又称三层网络设备。**路由器的各个端口都有MAC地址和IP地址**
 - 交换机是MAC层的，又称二层网络设备。交换机的端口没有MAC地址
- 同：工作原理类似：通过查表选择转发端口

路由器的工作原理：

一般都是转发包：在端口接收网络包；然后通过路由表查询转发的目标——下一跳，然后通过对应的端口转发出去

- 接收包：
 - 首先，电信号到达网线端口，路由器中对应的模块会接收该包，并且将电信号转换为数字信号，并且通过末尾的FCS进行错误校验
 - 接着，检查该包的MAC头部的接受方的MAC地址，如果接收方不是自己，那么直接丢弃；如果接收方是自己，那么放入缓存

- 接收完成后，就会**丢弃该包的MAC头部**（因为需要通过该路由器转发出去，MAC头部需要变化了）
- 缓存排队排到之后，根据IP头部的目的IP地址，**查询路由表**获得下一跳的IP地址——查询流程类似于网卡的选择过程：找到就选择作为下一跳目标，找不到就选择默认路由
- eg：从10.10.1.101发送到192.168.1.100，那么到达该路由器之后，查询路由表，然后一条一条匹配，发现第二条匹配，那么该条记录就会作为转发目标；而如果都不匹配就选择最下面的默认路由

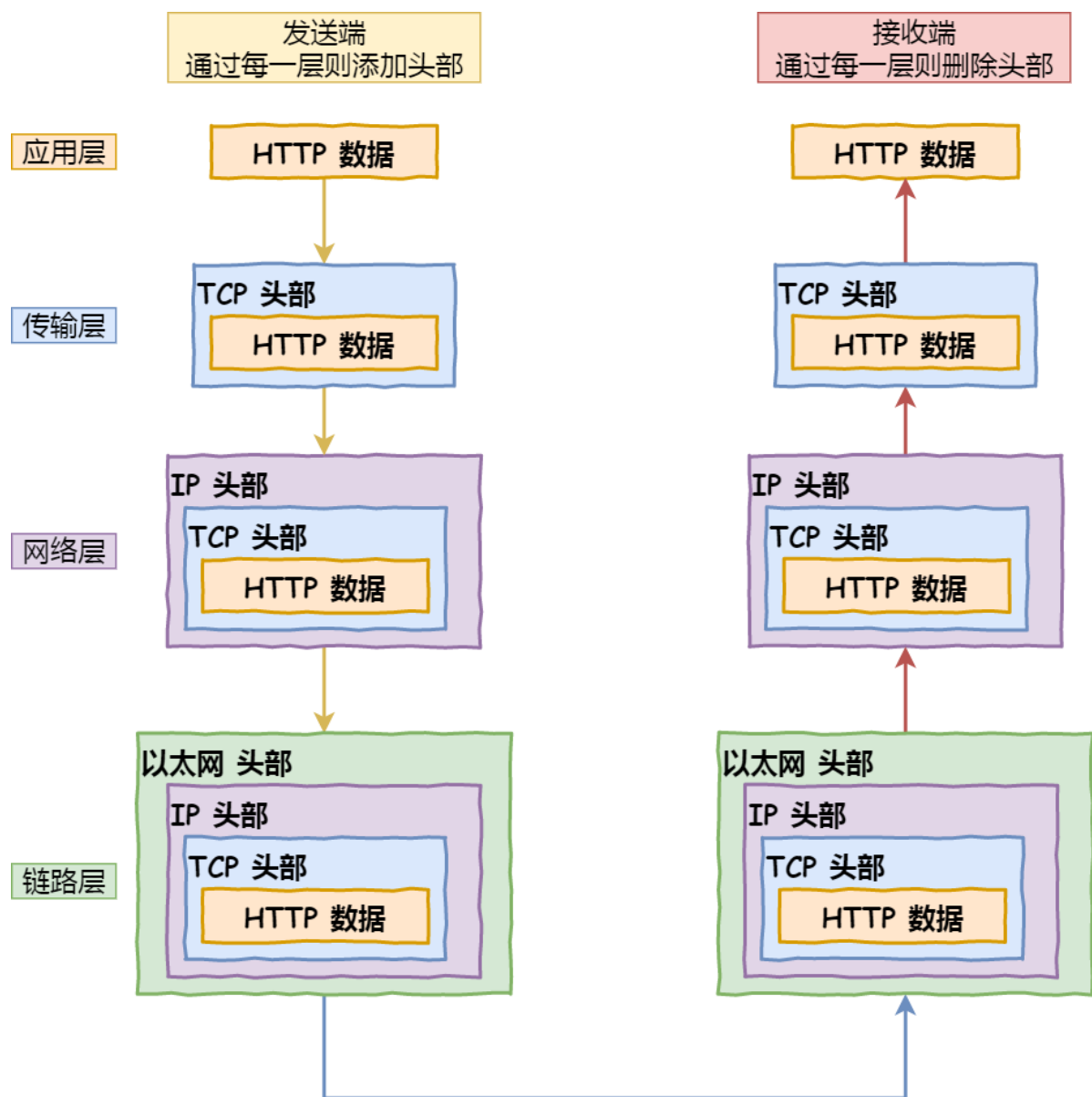


- 发送包：选择号转发目标之后，就要发送出去了
 - 先根据**路由表中的网关列**判断下一跳的地址
 - 如果网关显示的是IP地址，那么**该IP地址就要要转发的目的地**，表示**还未抵达终点**，需要继续转发
 - 如果网关为空，那么IP报文中的**接收方地址就是转发的目的地**，且已经**找到终点了**
 - 接着根据得到的转发IP地址，通过ARP协议得到MAC地址，并将结果写入到MAC头部的接收方MAC地址中
 - 然后，发送方的MAC地址，填写的是输出端口的MAC地址
 - MAC头部的协议字段填写的是0X80，表示IP协议——至此，MAC头部又重新添加完成了
 - 然后驱动网卡将数字信号转换为电信号，并通过端口发送到网线上

——然后，又通过交换机到达下一个路由器，然后重复操作，直到达到最终目的地

可以发现，在网络包传输的过程中，IP头部都是不变的，只是提供查询，而变化的是MAC头部，不断变化发送方的MAC地址和接收方的MAC地址——因为是在两个设备之间进行传输的，而IP地址是用来两个主机之间的传输的

10. 服务端（目的地）



服务端开始将网络包打开查看

- 先看MAC头部，看接受方的MAC地址是否和自己的MAC地址一致，一致才接收，且把MAC头部丢弃
- 然后看IP头部，看接收方的IP地址是否和自己的IP地址一致，一致才接收，并且根据**协议位**，知道上层需要用到TCP协议
- 然后看TCP头部，里面有序列号，看序列号是否是期待的，如果是就放入缓存中，**然后返回一个ACK**，并且TCP中还包含接收方的端口号，而HTTP的服务器正在监听该端口
- 根据端口号将该包发送给HTTP进程，然后HTTP进程打开数据区了解到该包的目的，于是处理之后，重新一层层封装，发送回去

ps: 交换机

路由器有交换机的功能，交换机就是用来交换数据的，eg: 3台电脑连接在交换机上，那么这几台电脑能够互相交换数据了

总结:

Web 页面请求过程

1. DHCP 配置主机信息

- 假设主机最开始没有 IP 地址以及其它信息，那么就需要先使用 DHCP 来获取。

- 主机生成一个 DHCP 请求报文，并将这个报文放入具有目的端口 67 和源端口 68 的 UDP 报文段中。
- 该报文段则被放入在一个具有广播 IP 目的地址(255.255.255.255)和源 IP 地址 (0.0.0.0) 的 IP 数据报中。
- 该数据报则被放置在 MAC 帧中，该帧具有目的地址 FF:FF:FF:FF:FF:FF，将广播到与交换机连接的所有设备。
- 连接在交换机的 DHCP 服务器收到广播帧之后，不断地向上分解得到 IP 数据报、UDP 报文段、DHCP 请求报文，之后生成 DHCP ACK 报文，该报文包含以下信息：IP 地址、DNS 服务器的 IP 地址、默认网关路由器的 IP 地址和子网掩码。该报文被放入 UDP 报文段中，UDP 报文段有被放入 IP 数据报中，最后放入 MAC 帧中。
- 该帧的目的地址是请求主机的 MAC 地址，因为交换机具有自学习能力，之前主机发送了广播帧之后就记录了 MAC 地址到其转发接口的交换表项，因此现在交换机就可以直接知道应该向哪个接口发送该帧。
- 主机收到该帧后，不断分解得到 DHCP 报文。之后就配置它的 IP 地址、子网掩码和 DNS 服务器的 IP 地址，并在其 IP 转发表中安装默认网关。

2. ARP 解析 MAC 地址

- 主机通过浏览器生成一个 TCP 套接字，套接字向 HTTP 服务器发送 HTTP 请求。为了生成该套接字，主机需要知道网站的域名对应的 IP 地址。
- 主机生成一个 DNS 查询报文，该报具有 53 号端口，因为 DNS 服务器的端口号是 53。
- 该 DNS 查询报文被放入目的地址为 DNS 服务器 IP 地址的 IP 数据报中。
- 该 IP 数据报被放入一个以太网帧中，该帧将发送到网关路由器。
- DHCP 过程只知道网关路由器的 IP 地址，为了获取网关路由器的 MAC 地址，需要使用 ARP 协议。
- 主机生成一个包含目的地址为网关路由器 IP 地址的 ARP 查询报文，将该 ARP 查询报文放入一个具有广播目的地址 (FF:FF:FF:FF:FF:FF) 的以太网帧中，并向交换机发送该以太网帧，交换机将该帧转发给所有的连接设备，包括网关路由器。
- 网关路由器接收到该帧后，不断向上分解得到 ARP 报文，发现其中的 IP 地址与其接口的 IP 地址匹配，因此就发送一个 ARP 回答报文，包含了它的 MAC 地址，发回给主机。

3. DNS 解析域名

- 知道了网关路由器的 MAC 地址之后，就可以继续 DNS 的解析过程了。
- 网关路由器接收到包含 DNS 查询报文的以太网帧后，抽取出 IP 数据报，并根据转发表决定该 IP 数据报应该转发的路由器。
- 因为路由器具有内部网关协议 (RIP、OSPF) 和外部网关协议 (BGP) 这两种路由选择协议，因此路由表中已经配置了网关路由器到达 DNS 服务器的路由表项。
- 到达 DNS 服务器之后，DNS 服务器抽取出 DNS 查询报文，并在 DNS 数据库中查找待解析的域名。
- 找到 DNS 记录之后，发送 DNS 回答报文，将该回答报文放入 UDP 报文段中，然后放入 IP 数据报中，通过路由器反向转发回网关路由器，并经过以太网交换机到达主机。

4. HTTP 请求页面

- 有了 HTTP 服务器的 IP 地址之后，主机就能够生成 TCP 套接字，该套接字将用于向 Web 服务器发送 HTTP GET 报文。
- 在生成 TCP 套接字之前，必须先与 HTTP 服务器进行三次握手来建立连接。生成一个具有目的端口 80 的 TCP SYN 报文段，并向 HTTP 服务器发送该报文段。
- HTTP 服务器收到该报文段之后，生成 TCP SYN ACK 报文段，发回给主机。
- 连接建立之后，浏览器生成 HTTP GET 报文，并交付给 HTTP 服务器。
- HTTP 服务器从 TCP 套接字读取 HTTP GET 报文，生成一个 HTTP 响应报文，将 Web 页面内容放入报文主体中，发回给主机。

- 浏览器收到 HTTP 响应报文后，抽取 Web 页面内容，之后进行渲染，显示 Web 页面。