计网八股（2）

jask

09/09/2024

什么是 PING

ping 是基于 ICMP 协议工作的，所以要明白 ping 的工作，首先我们先来熟悉 ICMP 协议。

参考计网八股 1： ICMP

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。

ICMP 包头的类型字段，大致可以分为两大类：

一类是用于诊断的查询消息，也就是「查询报文类型」

另一类是通知出错原因的错误消息，也就是「差错报文类型」

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）。

相比原生的 ICMP，这里多了两个字段：

标识符：用以区分是哪个应用程序发 ICMP 包，比如用进程 PID 作为标识符；

序号：序列号从 0 开始，每发送一次新的回送请求就会加 1，可以用来确认网络包是否有丢失。

在选项数据中，ping 还会存放发送请求的时间值，来计算往返时间，说明路程的长短。

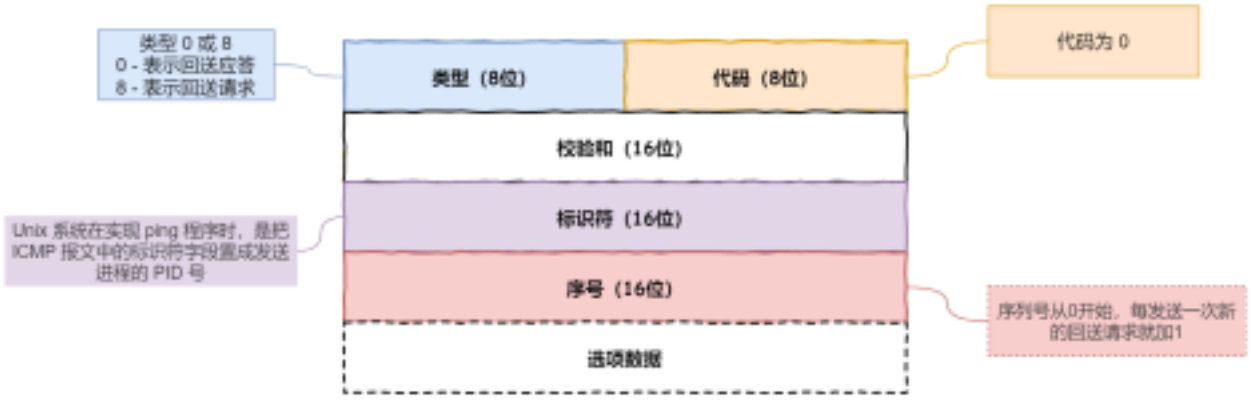


Figure 1： 字段

差错报文类型

说明几个常用的 ICMP 差错报文的例子： 目标不可达消息——类型为 3 原点抑制消息——类型 4 重定向消息——类型 5 超时消息——类型 11

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。

由此，根据 ICMP 不可达的具体消息，发送端主机也就可以了解此次发送不可达的具体原因。

举例 6 种常见的目标不可达类型的代码：

网络不可达 0 IP 地址是分为网络号和主机号的，所以当路由器中的路由器表匹配不到接收方 IP 的网络号，就通过 ICMP 协议以网络不可达（Network Unreachable）的原因告知主机。

自从不再有网络分类以后，网络不可达也渐渐不再使用了。

主机不可达 1 当路由表中没有该主机的信息，或者该主机没有连接到网络，那么会通过 ICMP 协议以主机不可达（Host Unreachable）的原因告知主机。

协议不可达 2 当主机使用 TCP 协议访问对端主机时，能找到对端的主机了，可是对端主机的防火墙已经禁止 TCP 协议访问，那么会通过 ICMP 协议以协议不可达的原因告知主机。

端口不可达 3 当主机访问对端主机 8080 端口时，这次能找到对端主机了，防火墙也没有限制，可是发现对端主机没有进程监听 8080 端口，那么会通过 ICMP 协议以端口不可达的原因告知主机。

需要进行分片但是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。随后，通过一个 ICMP 的不可达消息类型，代码为 4 的报文，告知发送端主机。

原点抑制消息 4

在使用低速广域线路的情况下，连接 WAN 的路由器可能会遇到网络拥堵的问题。

ICMP 原点抑制消息的目的就是为了缓和这种拥堵情况。

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。

收到这个消息的主机借此了解在整个线路的某一处发生了拥堵的情况，从而增大 IP 包的传输间隔，减少网络拥堵的情况。

然而，由于这种 ICMP 可能会引起不公平的网络通信，一般不被使用。

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。这主要发生在路由器持有更好的路由信息的情况下。

路由器会通过这样的 ICMP 消息告知发送端，让它下次发给另外一个路由器。

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。

设置 IP 包生存周期的主要目的，是为了在路由控制遇到问题发生循环状况时，避免 IP 包无休止地在网络上被转发。

查询报文类型的使用

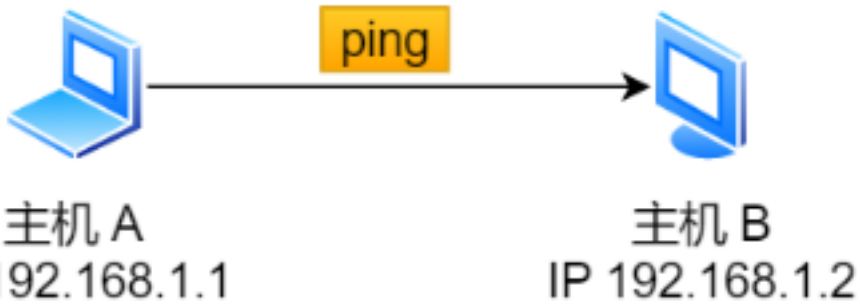
同个子网下的主机 A 和主机 B，主机 A 执行 ping 主机 B 后，我们来看看其间发送了什么？

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。

ICMP 数据包内包含多个字段，最重要的是两个：

第一个是类型，对于回送请求消息而言该字段为 8 ；

另外一个是序号，主要用于区分连续 ping 的时候发出的多个数据包。



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；还要附加上一些控制信息，依据以太网的介质访问规则，将它们传送出去。

主机 B 收到这个数据帧后，先检查它的目的 MAC 地址，并和本机的 MAC 地址对比，如符合，则接收，否则就丢弃。

接收后检查该数据帧，将 IP 数据包从帧中提取出来，交给本机的 IP 层。同样,IP 层检查后，将有用的信息提取后交给 ICMP 协议。

主机 B 会构建一个 ICMP 回送响应消息数据包，回送响应数据包的类型字段为 0 ，序号为接收到的请求数据包中的序号，然后再发送出去给主机 A。

可以看出 ping 这个程序是使用了 ICMP 里面的 ECHO REQUEST(类型为 8) 和 ECHO REPLY (类型为 0)。

traceroute 差错报文类型的使用

traceroute 的第一个作用就是故意设置特殊的 TTL，来追踪去往目的地时沿途经过的路由器。

traceroute 的参数指向某个目的 IP 地址：

原理

它的原理就是利用 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？

traceroute 在发送 UDP 包时，会填入一个不可能的端口号值作为 UDP 目标端口号（大于 3000 ）。当目的主机，收到 UDP 包后，会返回 ICMP 差错报文消息，但这个差错报文消息的类型是「端口不可达」。所以，当差错报文类型是端口不可达时，说明发送方发出的 UDP 包到达了目的主机。

作用二：

traceroute 还有一个作用是故意设置不分片，从而确定路径的 MTU。

这样做的目的是为了路径 MTU 发现。

因为有的时候我们并不知道路由器的 MTU 大小，以太网的数据链路上的 MTU 通常是 1500 字节，但是非以外网的 MTU 值就不一样了，所以我们要知道 MTU 的大小，从而控制发送的包大小。

它的工作原理如下：

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。

协议栈

在 DNS 获取到 IP 之后，就可以通过 HTTP 的传输工作交给操作系统的协议栈。

TCP 协议

HTTP（1,2）是基于 TCP 协议传输的，所以在这我们先了解下 TCP 协议。

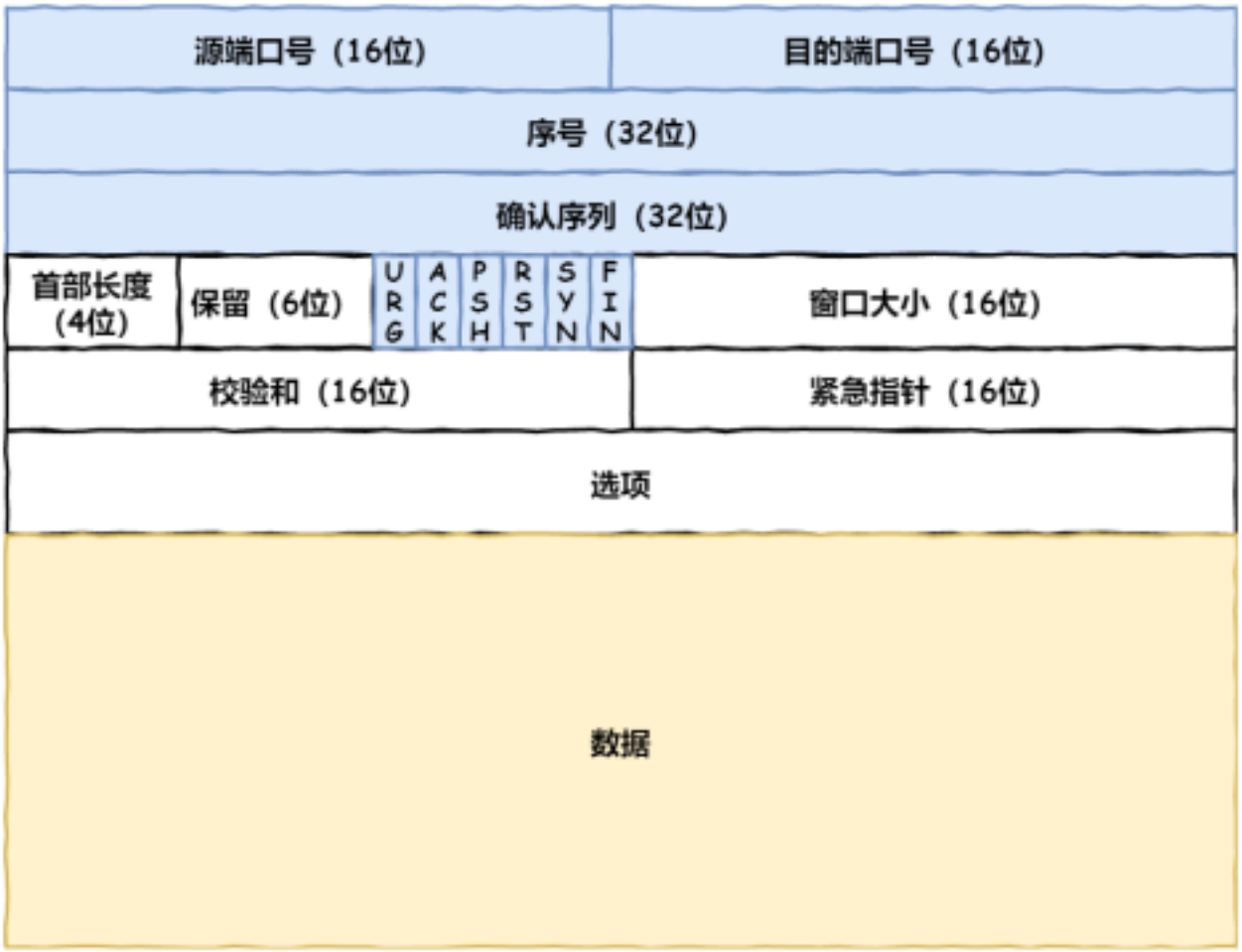


Figure 2: 报文头部

首先，源端口号和目标端口号是不可少的，如果没有这两个端口号，数据就不知道应该发给哪个应用。接下来有包的序号，这个是为了解决包乱序的问题。

还有应该有的是确认号，目的是确认发出去对方是否有收到。如果没有收到就应该重新发送，直到送达，这个是为了解决不丢包的问题。

接下来还有一些状态位。例如 SYN 是发起一个连接，ACK 是回复，RST 是重新连接，FIN 是结束连接等。

TCP 是面向连接的，因而双方要维护连接的状态，这些带状态位的包的发送，会引起双方的状态变更。

还有一个重要的就是窗口大小。TCP 要做流量控制，通信双方各声明一个窗口（缓存大小），标识自己当前能够的处理能力，别发送的太快，撑死我，也别发的太慢，饿死我。

除了做流量控制以外,TCP 还会做拥塞控制，

三次握手

在 HTTP 传输数据之前，首先需要 TCP 建立连接,TCP 连接的建立，通常称为三次握手。

这个所谓的「连接」，只是双方计算机里维护一个状态机，在连接建立的过程中，双方的状态变化时序图就像这样。

一开始，客户端和服务端都处于 CLOSED 状态。先是服务端主动监听某个端口，处于 LISTEN 状态。

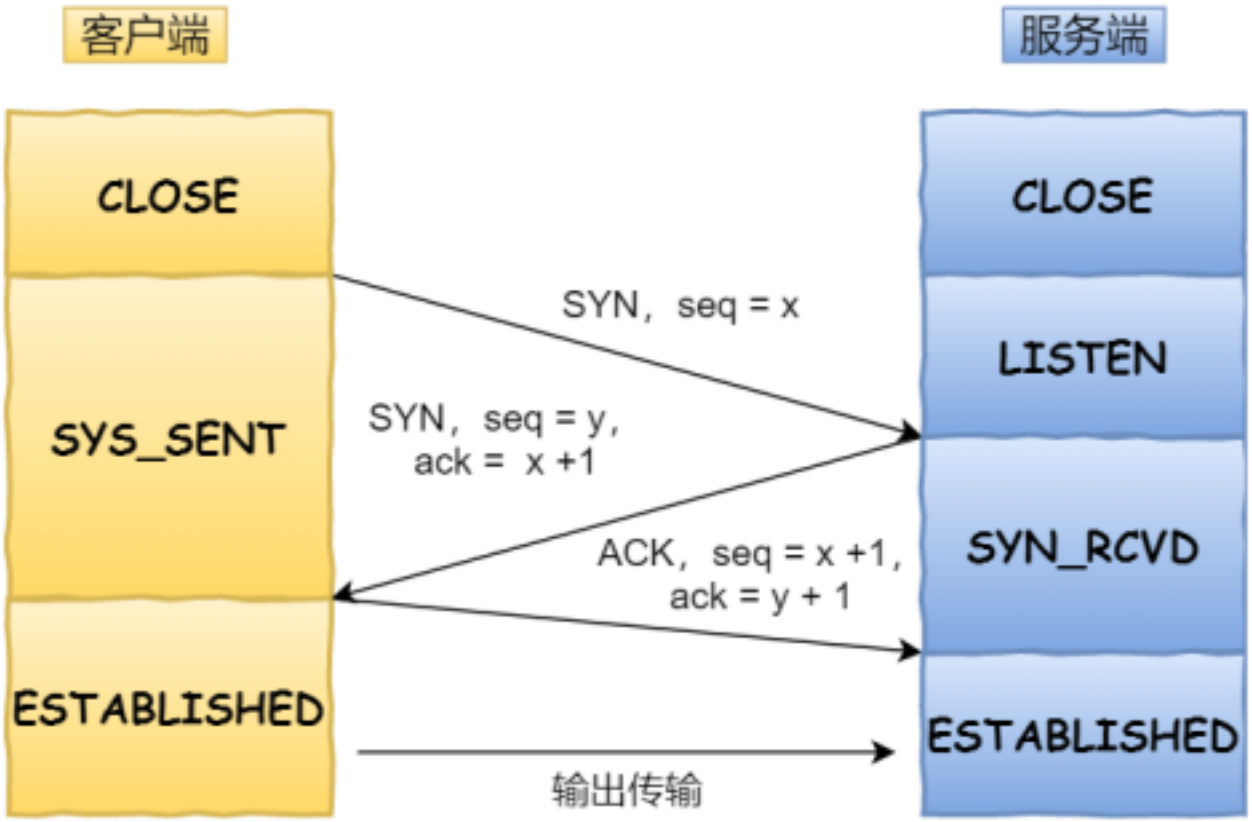


Figure 3: 三次握手

然后客户端主动发起连接 SYN ，之后处于 SYN-SENT 状态。

服务端收到发起的连接，返回 SYN ，并且 ACK 客户端的 SYN ，之后处于 SYN-RCVD 状态。

客户端收到服务端发送的 SYN 和 ACK 之后，发送 ACK 的 ACK ，之后处于 ESTABLISHED 状态，因为它一发一收成功了。

服务端收到 ACK 的 ACK 之后，处于 ESTABLISHED 状态，因为它也一发一收了。

所以三次握手目的是保证双方都有发送和接收的能力。

如何查看 tcp 的连接状态

TCP 的连接状态查看，在 Linux 可以通过 netstat -napt 命令查看。

tcp 分割数据

如果 HTTP 请求消息比较长，超过了 MSS 的长度，这时 TCP 就需要把 HTTP 的数据拆解成一块块的数据发送，而不是一次性发送所有数据。

MTU ：一个网络包的最大长度，以太网中一般为 1500 字节。

MSS ：除去 IP 和 TCP 头部之后，一个网络包所能容纳的 TCP 数据的最大长度。

数据会被以 MSS 的长度为单位进行拆分，拆分出来的每一块数据都会被放进单独的网络包中。也就是在每个被拆分的数据加上 TCP 头信息，然后交给 IP 模块来发送数据。

tcp 报文生成

TCP 协议里面会有两个端口，一个是浏览器监听的端口（通常是随机生成的），一个是 Web 服务器监听的端口（HTTP 默认端口号是 80 ，HTTPS 默认端口号是 443 ）。

在双方建立了连接后，TCP 报文中的数据部分就是存放 HTTP 头部 + 数据，组装好 TCP 报文之后，就需交给下面的网络层处理。

至此，网络包的报文如下图。

在 IP 协议里面需要有源地址 IP 和目标地址 IP：

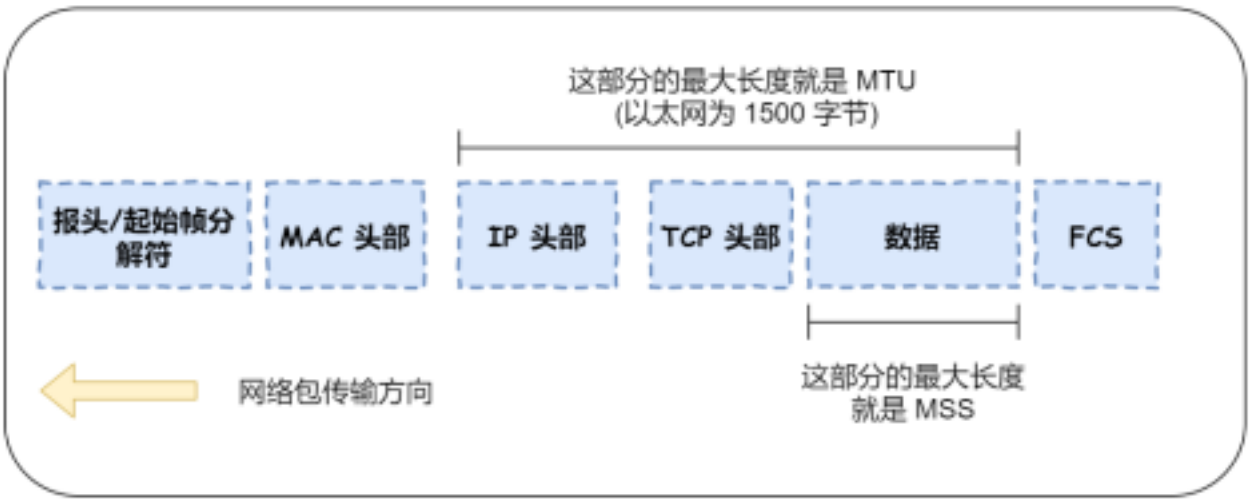


Figure 4: 数据分割

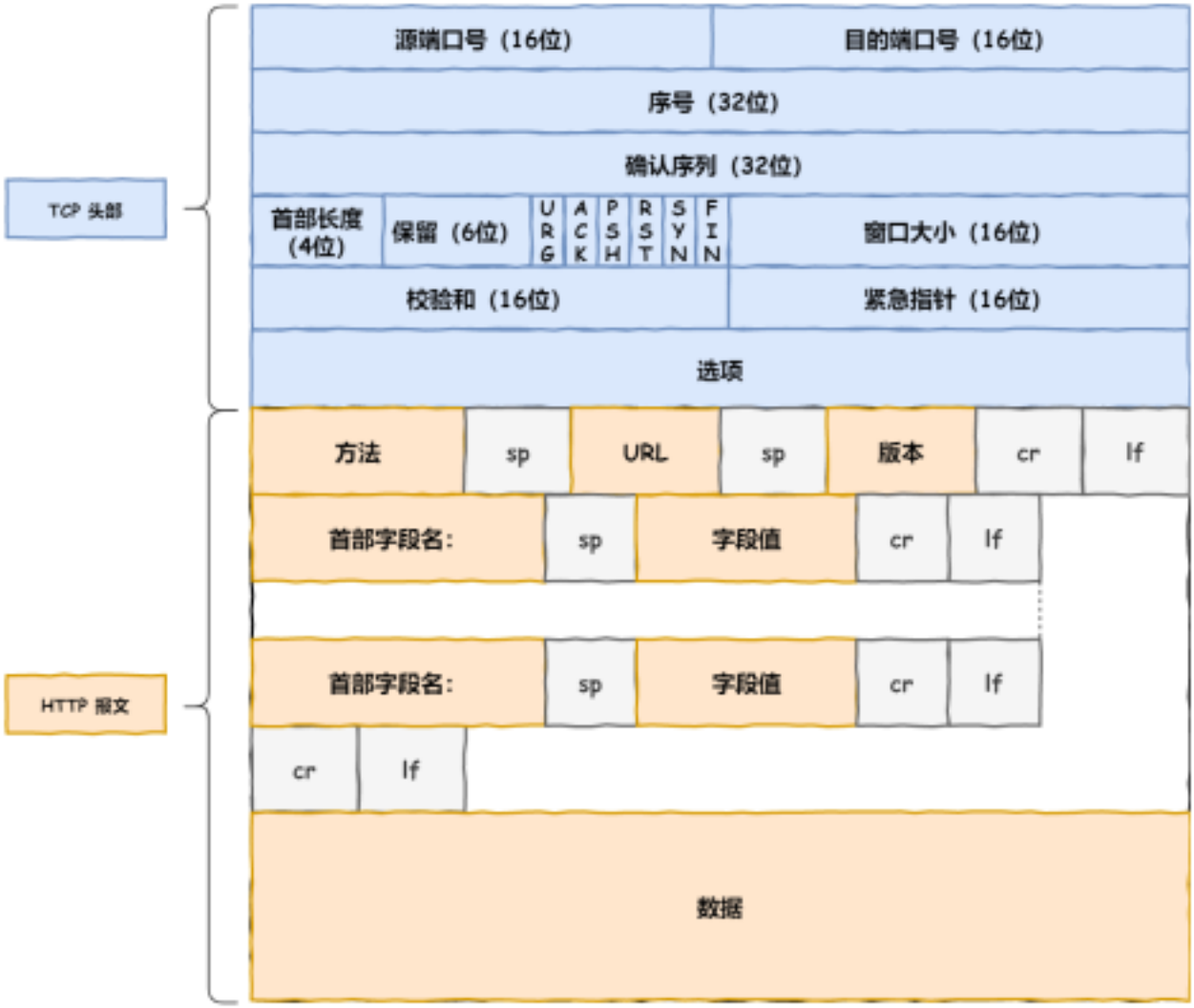


Figure 5: 报文

源地址 IP，即是客户端输出的 IP 地址；

目标地址，即通过 DNS 域名解析得到的 Web 服务器 IP。

因为 HTTP 是经过 TCP 传输的，所以在 IP 包头的协议号，要填写为 06（十六进制），表示协议为 TCP。

假设客户端有多个网卡，就会有多个 IP 地址，那 IP 头部的源地址应该选择哪个 IP 呢？

当存在多个网卡时，在填写源地址 IP 时，就需要判断到底应该填写哪个地址。这个判断相当于在多块网卡中判断应该使用哪个一块网卡来发送包。

这个时候就需要根据路由表规则，来判断哪一个网卡作为源地址 IP。

在 Linux 操作系统，我们可以使用 route -n 命令查看当前系统的路由表。