

三、关于TCP、UDP基础概念的面试题

关于TCP基本认识的面试题：

1. 为什么需要 TCP 协议? TCP 工作在哪一层?

因为 TCP 是一个工作在**传输层的可靠数据传输的服务**，它能确保接收端接收的网络包是**无损坏、无间隔、非冗余和按序的**

IP 层是「不可靠」的，它不保证网络包的交付、不保证网络包的按序交付、也不保证网络包中的数据完整性，所以需要TCP协议

如果需要保障网络数据包的可靠性，那么就需要由上层(传输层)的 TCP 协议来负责

2. 什么是 TCP ?

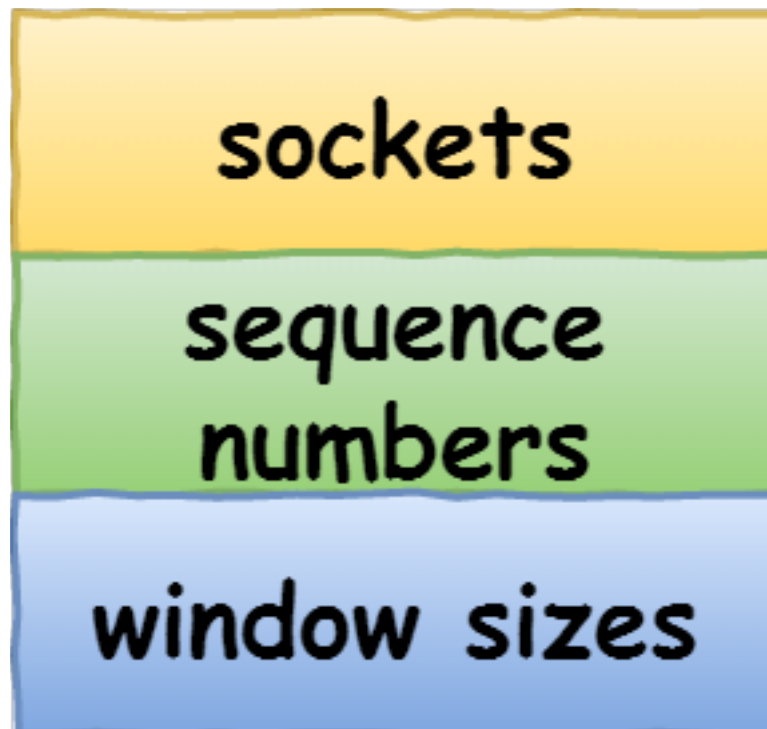
TCP 是**面向连接的、可靠的、基于字节流**的传输层通信协议



- **面向连接**：一定是「一对一」才能连接，不能像 UDP 协议可以一个主机同时向多个主机发送消息，也就是一对多是无法做到的；
- **可靠的**：无论网络链路中出现了怎样的链路变化，TCP 都可以保证一个报文一定能够到达接收端
- **字节流**：消息是「没有边界」的，所以无论我们消息有多大都可以进行传输。并且消息是「有序的」，当「前一个」消息没有收到的时候，即使它先收到了后面的字节，那么也不能扔给应用层去处理，同时对「重复」的报文会自动丢弃

3. 什么是TCP连接?

简单来说就是，用于保证可靠性和流量控制维护的某些状态信息，这些信息的组合，包括 **Socket**、序列号和窗口大小称为连接。



所以我们可以知道，建立一个 TCP 连接是需要客户端与服务器端达成上述三个信息的共识：

- **Socket**: 由 IP 地址和端口号组成
- **序列号**: 用来解决乱序问题等
- **窗口大小**: 用来做流量控制

4. 如何唯一确定一个 TCP 连接呢？

- TCP四元组：【源地址，源端口，目标地址，目标端口】

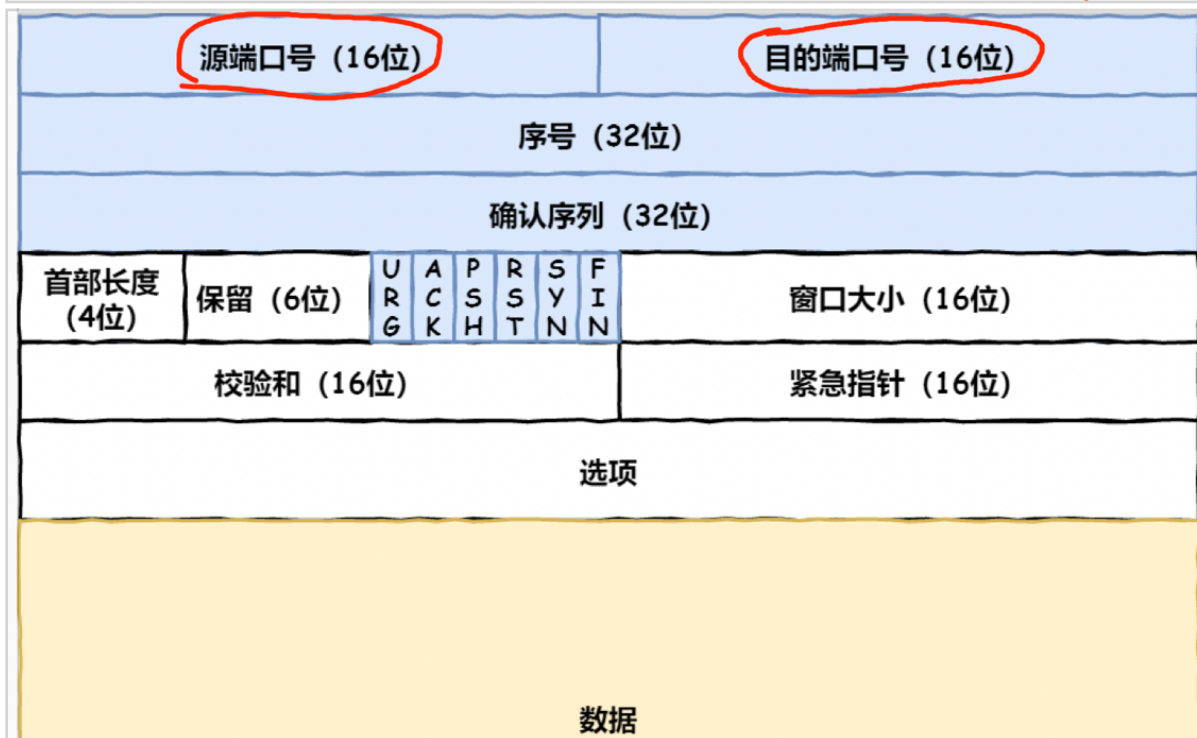


源地址和目的地址的字段(各32位，各4字节)是在 IP 头部中，作用是通过 IP 协议发送报文给 对方主机

源端口和目的端口的字段(各16位，各2字节)是在 TCP 头部中，作用是告诉 TCP 协议应该把报文发给 哪个进程

H2 IP报文

IP报文是在网络层传输的数据单元，也叫IP数据报。IP报文格式如下图（图片来源：百度百科）



5. 有一个 IP 的服务器监听了一个端口，它的 TCP 的最大连接数是多少？

服务器通常固定在某个本地端口上监听，等待客户端的连接请求

因此，客户端 IP 和 端口是可变的，其理论值计算公式如下：

$$\text{最大 TCP 连接数} = \text{客户端的IP 数} \times \text{客户端的端口数}$$

对于IPv4，客户端的IP数最多为 2^{32} ，客户端的端口数最多为 2^{16} ，也就是说服务端单机最大的TCP连接数约为 2^{48}

当然，服务端最大并发TCP连接数远远不能达到理论上限（为什么？）

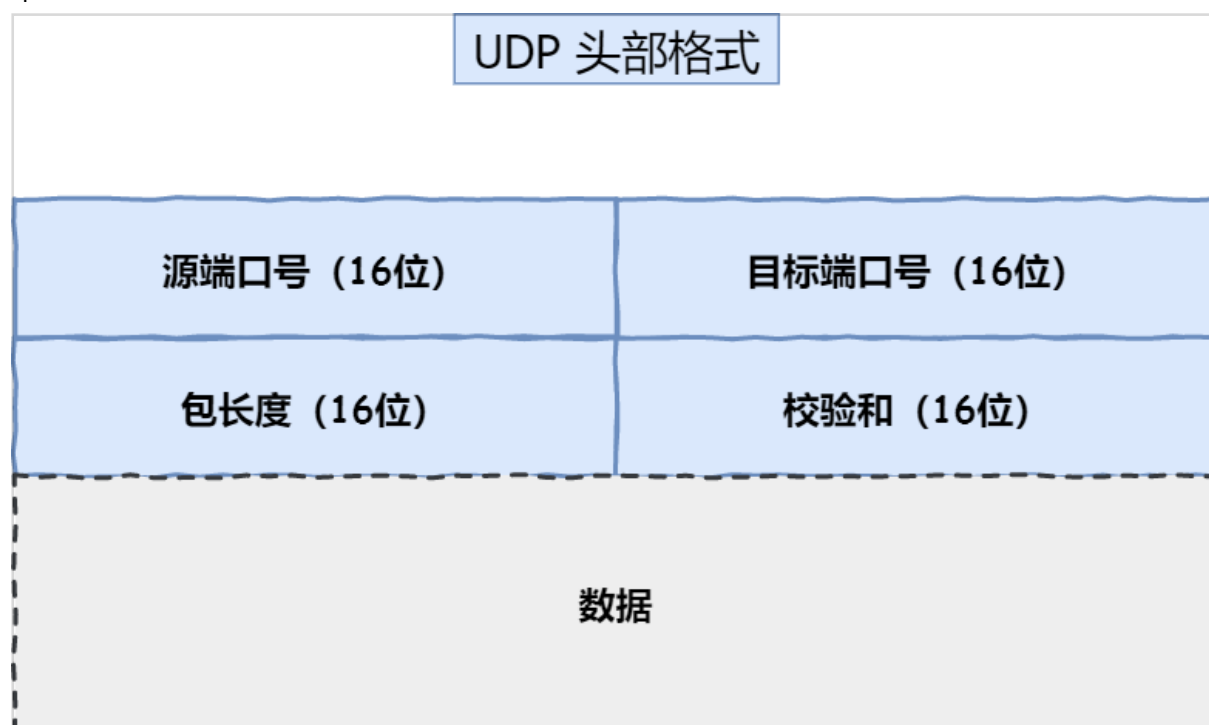
- 首先主要是**文件描述符限制**，Socket 都是文件，所以首先要通过 `ulimit` 配置文件描述符的数目；
- 另一个是**内存限制**，每个 TCP 连接都要占用一定内存，操作系统的内存是有限的

6 TCP和UDP有什么区别？分别的应用场景是什么？

Recap UDP:

UDP不提供复杂的控制机制，利用IP 提供面向「无连接」的通信服务

UDP 协议真的非常简单，头部只有 8 个字节(64 位，每个部分各2字节)，UDP 的头部格式如下:



源端口号和目的端口号：主要是告诉 UDP 协议应该把报文发给哪个进程

包长度（UDP长度）：该字段保存了UDP首部+数据的长度之和，最小值是8（单位是字节）（仅有头部没有数据的时候）

校验和：校验和是为了提供可靠的 UDP 首部和数据而设计。

• TCP和UDP的区别：

1. 连接的有无

- TCP 是面向连接的传输层协议，传输数据前先要建立连接
- UDP 是不需要连接，即刻传输数据

2. 服务对象方式不同

- TCP 是一对一的两点服务，即一条连接只有两个端点
- UDP 支持一对一、一对多、多对多的交互通信

3. 可靠性不同

- TCP 是可靠交付数据的，数据可以无差错、不丢失、不重复、按顺序到达
- UDP 是尽最大努力交付，不保证可靠交付数据

4. 拥塞控制、流量控制的有无

- TCP 有拥塞控制和流量控制机制，保证数据传输的安全性
- UDP 则没有，即使网络非常拥堵了，也不会影响 UDP 的发送速率

5. 首部开销不同

- TCP 首部长较长，会有一定的开销，首部在没有使用「选项」字段时是用了「选项」字段则会变长（由4位首部长度可以推算，TCP首部最长是60字节，选项最长40字节）
- UDP 首部只有 8 个字节，并且是固定不变的，开销较小

6. 传输方式不同

- TCP 是流式传输，没有边界，但保证顺序和可靠
- UDP 是一个包一个包的发送，是有边界的，但可能会丢包和乱序

7. 分片不同

- TCP 的数据大小如果大于 **MSS** 大小，则会在**传输层**进行分片，目标主机收到后，也同样在**传输层**组装 TCP 数据包，如果中途丢失了一个分片，只需要传输丢失的这个分片
- UDP 的数据大小如果大于 **MTU** 大小，则会在 **IP 层**进行分片，目标主机收到后，在 **IP 层**组装完数据，接着再传给传输层，但是如果中途丢了一个分片，则就需要重传所有的数据包，这样传输效率非常差，所以通常 UDP 的报文应该小于 MTU

• TCP和UDP分别的应用场景是什么？

由于 TCP 是面向连接，能保证数据的可靠性交付，因此经常用于：

- FTP 文件传输
- HTTP / HTTPS

由于 UDP 面向无连接，它可以随时发送数据，再加上UDP本身的处理既简单又高效，因此经常用于：

- 包总量较少的通信，如 DNS 、 SNMP等
- 视频、音频等多媒体通信
- 广播通信

7. 为什么 UDP 头部没有「首部长度」字段，而 TCP 头部有「首部长度」字段呢？

原因是 TCP 有可变长的「选项」字段，而 UDP 头部长度则是不会变化的，无需多一个字段去记录UDP 的首部长度

8. 为什么 UDP 头部有「包长度」字段，而 TCP 头部则没有「包长度」字段呢？

UDP的数据长度 = 包长度 - 首部长度（固定8字节），所以需要包长度

而TCP计算负载数据长度：

TCP 数据的长度 = IP 总长度 - IP 首部长度 - TCP 首部长度

其中 IP 总长度 和 IP 首部长度，在 IP 首部格式是已知的。TCP 首部长度，则是在 TCP 首部格式（对应的4位字段：首部长度/数据偏移）已知的，所以就可以求得 TCP 数据的长度

大家这时就奇怪了问：“UDP 也是基于 IP 层的呀，那 UDP 的数据长度也可以通过这个公式计算呀？为何还要有「包长度」呢？”

这么一问，确实感觉 UDP 「包长度」是冗余的。

因为为了网络设备硬件设计和处理方便，首部长度需要是 4 字节的整数倍。

如果去掉 UDP 「包长度」字段，那 UDP 首部长度就不是 4 字节的整数倍了，所以小林觉得这可能是为了补全 UDP 首部长度是 4 字节的整数倍，才补充了「包长度」字段。

