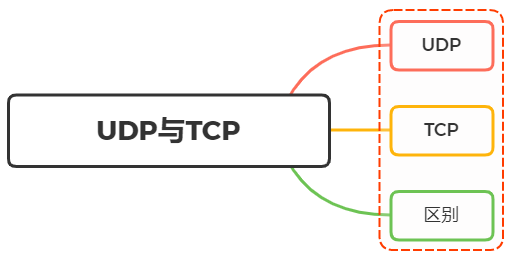
# 面试官：如何理解UDP 和 TCP? 区别? 应用场景?



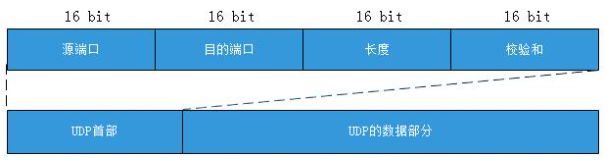
## 一、UDP

UDP（User Datagram Protocol），用户数据包协议，是一个简单的**面向数据报的通信协议**，即对应用层交下来的报文，不合并，不拆分，只是在其上面加上首部后就交给了下面的网络层

也就是说无论应用层交给UDP多长的报文，它统统发送，一次发送一个报文

而对接收方，接到后直接去除首部，交给上面的应用层就完成任务

UDP报头包括4个字段，每个字段占用2个字节（即16个二进制位），标题短，开销小



特点如下：

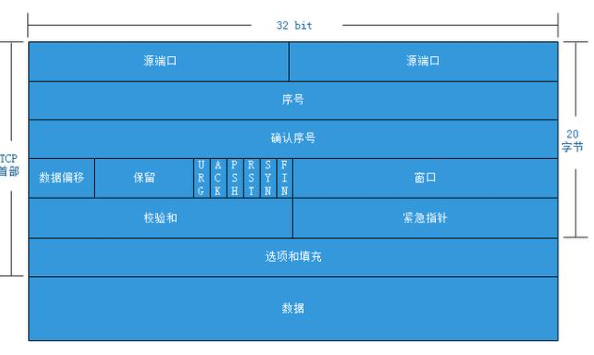
* UDP 不提供复杂的控制机制，利用 IP 提供面向无连接的通信服务
* 传输途中出现丢包，UDP 也不负责重发
* 当包的到达顺序出现乱序时，UDP没有纠正的功能。
* 并且它是将应用程序发来的数据在收到的那一刻，立即按照原样发送到网络上的一种机制。即使是出现网络拥堵的情况，UDP 也无法进行流量控制等避免网络拥塞行为

## 二、TCP

TCP（Transmission Control Protocol），传输控制协议，是一种可靠、**面向字节流的通信协议**，把上面应用层交下来的数据看成无结构的字节流来发送

可以想象成流水形式的，发送方TCP会将数据放入“蓄水池”（缓存区），等到可以发送的时候就发送，不能发送就等着，TCP会根据当前网络的拥塞状态来确定每个报文段的大小

TCP报文首部有20个字节，额外开销大

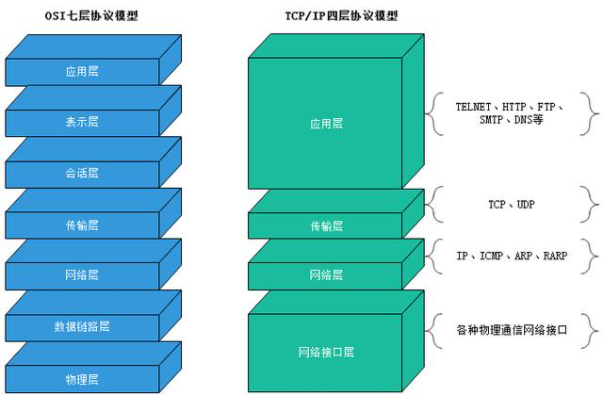


特点如下：

* TCP充分地实现了数据传输时各种控制功能，可以进行丢包时的重发控制，还可以对次序乱掉的分包进行顺序控制。而这些在 UDP 中都没有。
* 此外，TCP 作为一种面向有连接的协议，只有在确认通信对端存在时才会发送数据，从而可以控制通信流量的浪费。
* 根据 TCP 的这些机制，在 IP 这种无连接的网络上也能够实现高可靠性的通信（ 主要通过检验和、序列号、确认应答、重发控制、连接管理以及窗口控制等机制实现）

## 三、区别

UDP与TCP两者的都位于传输层，如下图所示：

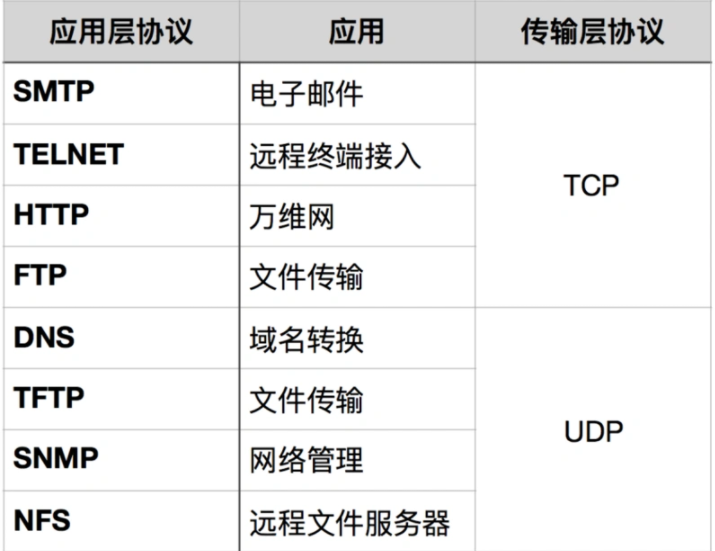


两者区别如下表所示：

|  |  |  |
| --- | --- | --- |
|  | TCP | UDP |
| 可靠性 | 可靠 | 不可靠 |
| 连接性 | 面向连接 | 无连接 |
| 报文 | 面向字节流 | 面向报文 |
| 效率 | 传输效率低 | 传输效率高 |
| 双共性 | 全双工 | 一对一、一对多、多对一、多对多 |
| 流量控制 | 滑动窗口 | 无 |
| 拥塞控制 | 慢开始、拥塞避免、快重传、快恢复 | 无 |
| 传输效率 | 慢 | 快 |

* TCP 是面向连接的协议，建立连接3次握手、断开连接四次挥手，UDP是面向无连接，数据传输前后不连接连接，发送端只负责将数据发送到网络，接收端从消息队列读取
* TCP 提供可靠的服务，传输过程采用流量控制、编号与确认、计时器等手段确保数据无差错，不丢失。UDP 则尽可能传递数据，但不保证传递交付给对方
* TCP 面向字节流，将应用层报文看成一串无结构的字节流，分解为多个TCP报文段传输后，在目的站重新装配。UDP协议面向报文，不拆分应用层报文，只保留报文边界，一次发送一个报文，接收方去除报文首部后，原封不动将报文交给上层应用
* TCP 只能点对点全双工通信。UDP 支持一对一、一对多、多对一和多对多的交互通信

两者应用场景如下图：



可以看到，TCP 应用场景适用于对效率要求低，对准确性要求高或者要求有链接的场景，而UDP 适用场景为对效率要求高，对准确性要求低的场景

## 参考文献

* https://zh.wikipedia.org
* https://www.shangmayuan.com/a/a1e3ceb218284cefb95de7fd.html
* https://segmentfault.com/a/1190000021815671
* https://vue3js.cn/interview