TCP/UDP 详解

# 用户数据报协议UDP

用户数据报协议udp只在ip的数据报服务至上增加了复用和分用的功能以及差错检测的功能。有面向无连接的报文，不可靠传输的特。UDP对应用层交下来的数据只添加首部，并进行特别的处理，就交给网络层；对网络层传递上来的用户数据报拆封首部后，原封不动的交给应用层。

A close up of a sign

Description automatically generated

## 1、UDP的首部格式

用户数据报UDP分为两个字段：数据字段和首部字段，从图来分析用户数据报UDP的首部格式。

A close up of text on a black surface

Description automatically generatedUDP首部字段很简单，由4个字段组成，每个字段的长度都是两个字节，共8字节。

* **源端口** 原端口号，在需要对方回信时选用，不需要时可全0
* **目的端口** 目的端口号，这在终点交付报文时必须使用，不然数据交给谁呢？
* **长度** UDP的长度，最小值为8字节，仅有首部
* **检验和** 检测用户数据报在传输过程是否有错，有错就丢弃。

在传输的过程中，如果接收方UDP发现收到的报文中的目的端口不存在，会直接丢弃，然后由网际控制报文协议ICMP给发送方发送“端口不可达”差错报文。

## 2、伪首部

计算校验和时，需要在UDP之前增加12个字节的伪首部。这种首部并不是用户数据报的真正首部。伪首部并不在网络中传输，只是在计算检验和，临时添加在UDP用户数据报前，得到一个临时的用户数据报。

UDP的校验和是把首部和数据部分一起校验，发送方计算校验和的一般步骤：

1. 将首部的校验和字段填充为0（零）
2. 把伪首部和用户数据报UDP看出16位的字符串连接起来
3. 如果数据部分不是偶数字节，则填充一个全零字节（该字节不发送到网络层）
4. 按二进制反马计算出这些16位字的和
5. 然后将和写入校验和字段，就可以发送到网络层了。

接收方收到用户数据报后，连同伪首部一起，按二进制反码求这些16位字的和，无差错结果是应全为1.否则出错，直接丢弃该报文。

A screenshot of a cell phone screen with text

Description automatically generated

# TCP协议

Tcp协议作为传输层主要协议之一，具有面向连接，端到端，可靠的双全工通信，面向字节流的数据传输协议。

TCP传输控制协议特点：

面向连接、可靠的、基于字节流的传输层协议

将应用层的数据分割成报文段并发送给目标节点TCP层

数据包都是有序号的，对方收到则发送ACK确认，未收到则重传

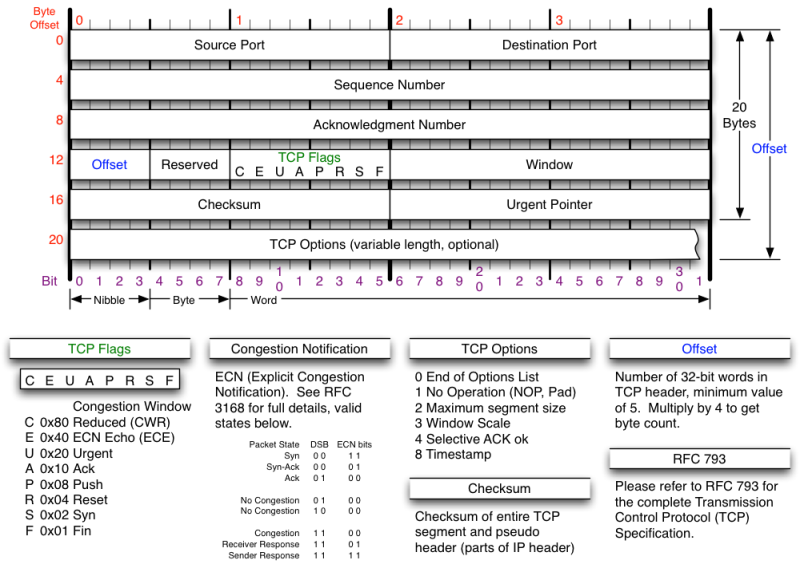
使用校验和函数来检验数据在传输过程中是否有误

## Tcp报文段

虽然TCP面向字节流，但TCP传输的数据单元却是报文段。TCP报文段分为TCP首部和数据部分，TCP报文段首部的前20个字节是固定的，后面有4\*n字节根据需要动态添加的选项，最大长度为40字节。

A screenshot of a cell phone

Description automatically generated



 **源端口和目的端口** 各占两个字节，TCP的分用功能也是通过端口实现的。

 **序号** 占4个字节，范围是[0,232],TCP是面向字节流的，每个字节都是按顺序编号。例如一个报文段，序号字段是201，携带数据长度是100，那么第一个数据的序号就是201，最后一个就是300。当达到最大范围，又从0开始。

 **确认号** 占4个字节，是期望收到对方下一个报文段的第一个字节的序号。若确认号=N,则表示序号N前所有的数据已经正确收到了。

 **数据偏移** 占4位，表示报文段的数据部分的起始位置，距离整个报文段的起始位置的距离。间接的指出首部的长度。

 **保留** 占6位，保留使用，目前为0.

 **URG（紧急）** 当URG=1,表明紧急指针字段有效，该报文段有紧急数据，应尽快发送。

 **ACK(确认)** 仅当ACK=1时，确认号才有效，连接建立后，所有的报文段ACK都为1。

 **PSH(推送)** 接收方接收到PSH=1的报文段，会尽快交付接收应用经常，不再等待整个缓存填满再交付。实际较少使用。

 **RST(复位)** RST=1时，表明TCP连接中出现严重差错，必须是否连接，再重连。

 **SYN(同步)** 在建立连接时用来同步序号。当SYN=1,ACK=0，则表明是一个连接请求报文段。SYN=1,ACK=1则表示对方同意连接。TCP建立连接用到。

 **FIN(终止)** 用来释放一个连接窗口。当FIN=1时，表明此报文段的发送方不再发送数据，请求释放单向连接。TCP断开连接用到。

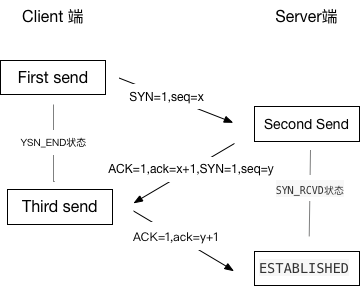
 **窗口** 占2个字节，表示发送方自己的接收窗口，窗口值用来告诉对方允许发送的数据量。

 **校验和** 占2字节，检验和字段查验范围包括首部和数据部分。

 **紧急指针** 占2字节，URG=1时，紧急指针指出本报文段中的紧急数据的字节数（紧急字节数结束后为普通字节）。

 **选项** 长度可变，最长可达40字节。例如最大报文段长度MSS。MSS指的是数据部分的长度而不是整个TCP报文段长度，MSS默认为536字节长。窗口扩大，时间戳选项等。

## Tcp建立连接 --- 三次握手



第一次握手：client 端 发送请求报文给客户端， 其中syn = 1, seq = x，这个时候client端进入syn\_end状态

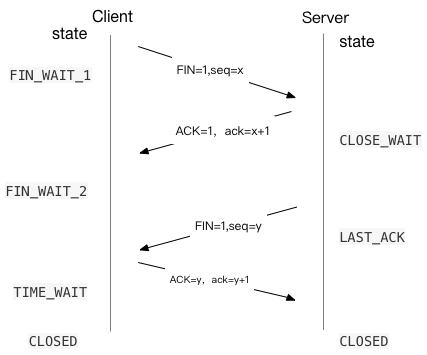
第二次握手： server端收到报文后，发回确认报文给client端口，其中ACK = 1, ack = x+1,因为需要客户端确认，所以报文中 也有syn = 1, seq = y, 这个时候server端进入 syn\_rcvd状态。

第三次握手： client 端收到确认报文后，发送确认报文，其中 ACK = 1, ack =y+1，发送完客户端进入established 状态，服务器接收到报文后也进入到established状态。至此建立完成。

三次握手原因

避免资源被浪费掉。如果在第二步握手时，由于网络延迟导致确认包不能及时到达客户端，那么客户端会认为第一次握手失败，再次发送连接请求，服务端收到后再次发送确认包。在这种情况下，服务端已经创建了两次连接，等待两个客户端发送数据，而实际却只有一个客户端发送数据。

## Tcp断开连接 --- 四次挥手



第一次挥手：client发送 FIN = 1, seq = x 的包给服务端，表示自己没有数据要进行传输，单面连接传输要关闭。发送完后，客户端进入fin\_wait\_1状态。

第二次挥手： 服务端收到请求包后，发回 ACK = 1, ack = x+1 的确认包，表示确认要断开连接。服务端进入 close\_wait状态。客户端收到后进入fin\_wait\_2状态，等待客户端的确认包。

第三次挥手： 服务端发送fin=1, seq = y 的包给客户端，表示自己没有数据要给客户端了，发送完成后进入last\_ack状态，等待客户端的确认包。

第四次挥手： 客户端收到请求包后， 发送ACK =1, ack =y+1的确认包给服务端，并进入time\_wait状态，服务端收到确认包后，进入closed状态，此时服务端到客户端的连接服务已断开，客户端多一段时间也会进入closed状态。

四次挥手原因：由于TCP的连接是全双工，双方都可以主动传输数据，一方的断开需要告知对方，让对方可以相关操作。

## UDP与TCP的区别

TCP面向连接vs无连接

可靠性：TCP通过三次握手保证，UDP可能丢包

有序性：TCP最终会进行排序（Sequence Number）

速度：TCP较慢

量级：TCP是重量级的，UDP是轻量级