

# UDP协议

## 1.UDP特点

**用户数据报协议UDP**（User Datagram Protocol）只在IP的数据报服务至上增加了一点的功能。这就是复用和分用功能和差错检测的功能。

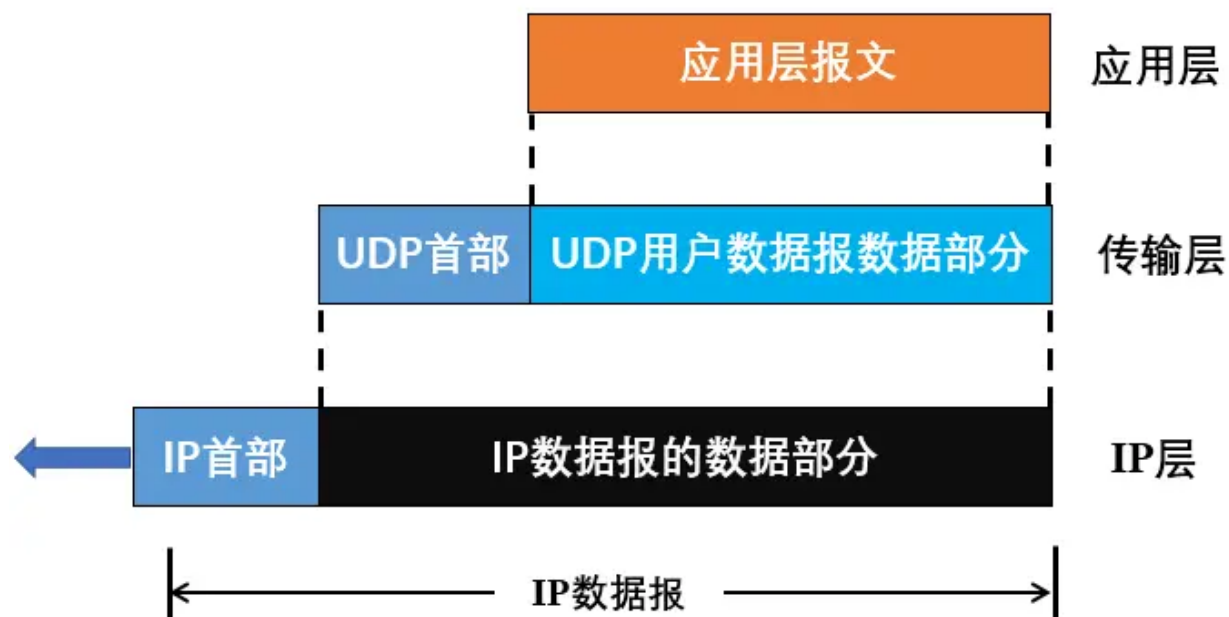
UDP的主要特点：

(1) UDP 是**无连接**的。即发送数据之间不需要建立连接，因此减少了开销和发送数据之间的时延。

(2) UDP使用**尽最大努力交付**，即不保证可靠交付，因此主机不需要维持复杂的连接状态表。

(3) UDP是**面向报文**的。

发送方UDP对应用程序交下来的报文，在添加首部后就向下交付IP层。UDP对应用层交下来的报文，既不开并，也不拆分，而是保留这些报文的边界。这就是说，应用层交给UDP多长的报文，UDP就照样发送，即**UDP一次交付一个完整的报文**。



因此，应用程序必须选择合适大小的报文。若报文太长，UDP把它交给IP层后，IP层在传送时可能要进行分片，这会降低IP层的效率。反之，如果报文太短，UDP把它交给IP层后，会使IP数据报的首部相对长度太大，也降低了IP层的效率。

(4) **UDP没有拥塞控制。**

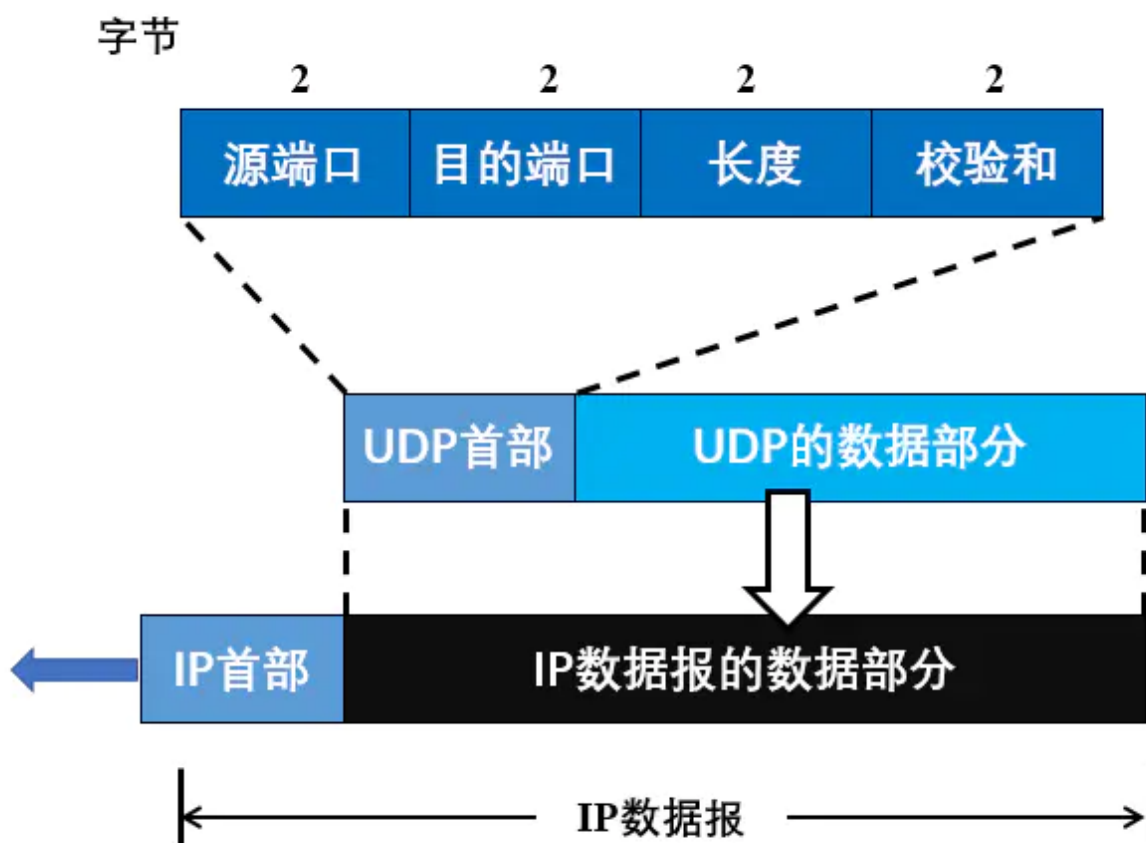
UDP没有拥塞控制，因此网络出现拥塞不会使源主机降低发送速率，这对实时应用很重要。对于一些实时应用（如IP电话，实时视频会议等）要求源主机以恒定的速率发送数据，并且允许网络发生拥塞时丢失一些数据，但却不允许数据有太大的时延。

(5) **UDP首部开销很小，只有8个字节。**

当然，一些实时应用需要对UDP的不可靠传输会进行适当的改进（如**前向纠错或重传已丢失的报文**），以减少数据的丢失。

## 2.UDP的首部格式

用户数据报UDP有两个字段：**数据字段和首部字段**。首部字段只有8个字节，由四个字段组成，**每个字段长度都是两个字节**。



(1) **源端口**：源端口号。不是必须的，只有在需要对方回信时选用，不需要时可用全0。(2) **目的端口**：目的端口号。这在终点交付报文时必须使用。

(3) **长度**：UDP数据报的长度，首部和数据部分长度之和，其最小值是8（即UDP用户数据报没有数据部分）。

(4) **校验和**：检测UDP在传输中是否出错，如果有错就丢弃。

当运输层IP层收到UDP数据报时，就根据首部中的目的端口，把UDP数据报通过相应的端口交付给上层应用进程。如果接收方UDP发现收到的报文的目的端口号不正确（即不存在对应于该端口号的应用进程），即丢弃该报文，并由网际控制报文协议ICMP发送“端口不可达”差错报文给发送方。

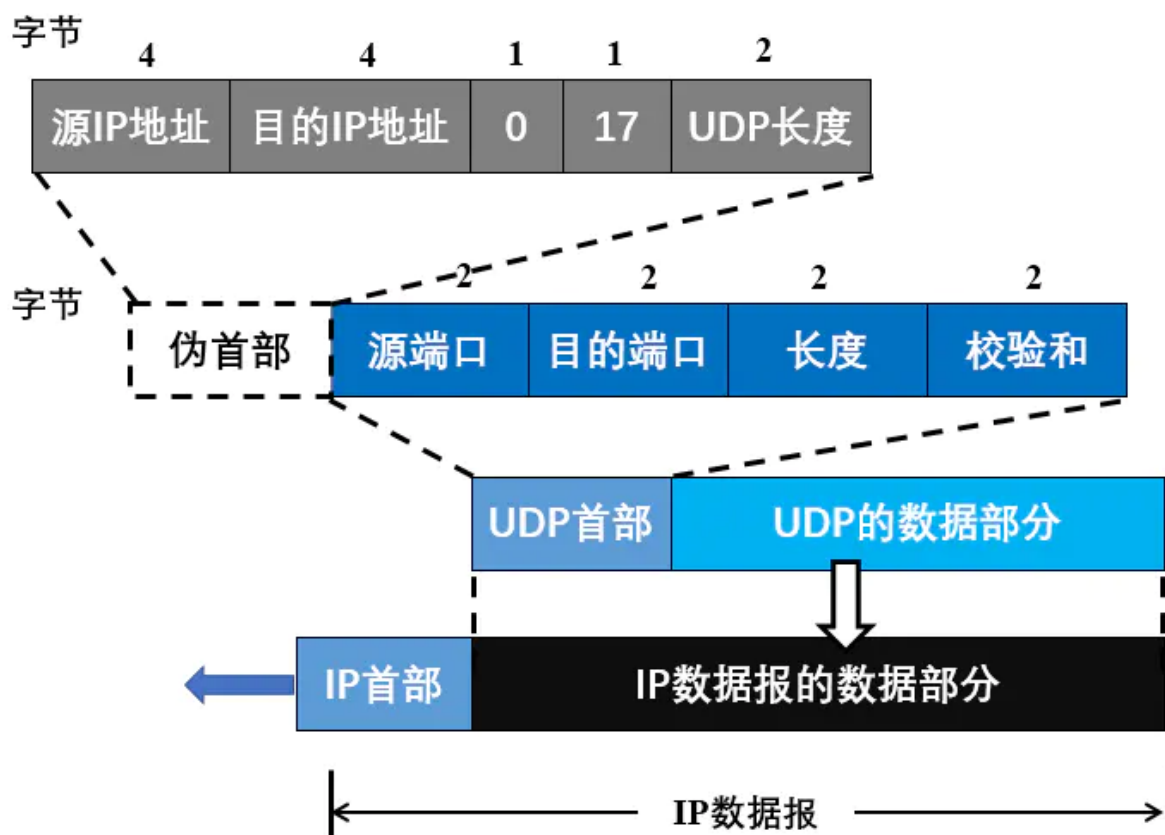
在ICMP协议一文中，在讨论Traceroute时，就是让发送的UDP用户数据报故意使用一个非法的UDP端口，结果ICMP就返回一个端口不可达的差错报文，从而达到测试的目的。

### 3.UDP伪首部和校验和计算

**UDP校验和提供差错检测功能。**在计算校验和时，要在UDP用户数据报之前增加12字节的**伪首部**。

所谓“伪首部”是因为伪首部并不是UDP用户数据报真正的首部，只是在计算校验和时，临时添加在UDP用户数据报前面，得到一个临时的UDP用户数据报。

**校验和就是按照这个临时的UDP用户数据报计算来的**，伪首部既不向下传送也不向上递交，**仅仅是为了计算校验和**。



- (1) **源IP地址和目的IP地址**：和IP数据一样，各占4个字节。
- (2) **伪首部第3个字段是全零**。
- (3) **协议字段**：以前说过，UDP协议的协议字段值是17。
- (4) **UDP长度**：UDP用户数据报长度，首部长度和数据部分长度之和。

UDP计算校验和的步骤：

- (1) 将校验和字段置位0。
- (2) 将伪首部和UDP用户数据报（首部和数据部分）看成是以16位为单位的二进制组成（如果用

户数据报数据部分不是偶数字节，用0填充），依次进行二进制反码求和。

(3) 将求和的结果的反码写入校验和字段。

这里先介绍以下二进制反码求和， $1+1=0$ ，高位进1、 $1+0=0+1=1$ 、 $0+0=0$ ，如果最后一次加法有溢出，结果需要加1。如下图，一个简单的二进制反码求和

## 二进制反码求和：求01010和11011反码求和

$$\begin{array}{r} 01010 \\ + 11011 \\ \hline 00101 \end{array} \quad \text{高位溢出，结果要加1}$$

↓

$$\begin{array}{r} 00101 \\ + 00001 \\ \hline 00110 \end{array}$$

下面用一个例子说明下，校验和的计算，假设源IP地址为153.19.0.104，目的IP地址为171.3.14.11，UDP用户数据报长度为15，源端口1087，目的端口13。根据UDP用户数据报的长度可知，数据部分是7（15 - 8）字节，所以需要0填充一个字节使之成为16位（偶数字节）的整数倍。

伪首部	153.19.8.104	171.3.14.11	0	17	15
首部	1087	13	15	全0	

下图给出了校验和计算的过程，每16个二进制作为1行，逐行按二进制反码求和，图中给出了前3行的求和结果，下面的计算过程相似，就略去了。

伪首部	153.19.8.104				10011001 00010011 →153.19
	171.3.14.11				00001000 01101000 →8.104
首部	0	17	15		10101011 00000011 →171.3
	1087		13		00001110 00001011 →14.11
数据部分	15		0		00000000 00010001 →0和17
	数据	数据	数据	数据	00000000 00001111 →15
	数据	数据	数据	0	00000100 00111111 →1087
	数据	数据	数据	0	00000000 00001101 →13
	数据	数据	数据	0	00000000 00001111 →15
					00000000 00000000 →0(检验和)
					01010100 01000101 →数据
					01010011 01010100 →数据
					01001001 01001110 →数据
					01000111 00000000 →数据和0(填充)

按二进制反码运算求和：10010110 11101101

将得出的结果求反码：**01101001 00010010** →校验和

10011001 00010011 →153.19  
00001000 01101000 →8.104  


---

10100001 01111011  
  
10100001 01111011  
10101011 00000011 →171.3  


---

**0**1001100 01111110 →高位溢出  

↓

**0**1001100 01111110  
00000000 00000001  


---

01001100 01111111

从上可以看出，UDP校验和是把首部和数据部分一起校验，而IP校验和仅仅校验IP首部。

在接收方，把收到的UDP用户数据报连同伪首部（以及可能的填充零字节）一起，按二进制反码求这些16位字的和，当无差错时期其结果应为全为1。否则就表明出现了差错，接收方就应丢弃这个UDP用户数据报（也可以上交给应用层，但附上差错的警告）。

这种简单的差错检验方法的检错能力并不强，但是好处是简单，处理起来较快。

## 4.总结



