

IP数据包的校验和算法C#版 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/253/2021\\_2022\\_IP\\_E6\\_95\\_B0\\_E6\\_8D\\_AE\\_E5\\_8C\\_c97\\_253823.htm](https://www.100test.com/kao_ti2020/253/2021_2022_IP_E6_95_B0_E6_8D_AE_E5_8C_c97_253823.htm) 为做伪IP，特地做了一个IP包，用C#改写IP头的校验和算法。

资料一：IP头格式：  
版本号 (4位) IP头长度 (4位) 服务类型 (8位) 数据包长度 (16位) 标识段 (16位) 标志段 (16位) 生存时间 (8位) 传输协议 (8位) 头校验和 (16位) 发送地址 (16位) 目标地址 (16位) 选项填充

资料二：IP协议采用统一的校验算法，其计算比较简单：  
设校验和初值为0，然后对数据每16位求异或，结果取反，便得校验和。校验时将数据（含校验和）按同样的算法求和，结果为0则数据正确，不为0表示通讯出错，需要丢弃该数据包。

算法源代码：  

```
public static UInt16 checksum(UInt16[] buffer,int size) { Int32 cksum=0. int counter. counter=0. while(size>0) { UInt16 val=buffer[counter]. cksum =Convert.ToInt32(buffer[counter]). counter =1. size=-1. } cksum=(cksum>>16) (cksum&0xffff). cksum =(cksum>>16). return (UInt16)(~cksum). }
```

注意：buffer数组为整个ip包数组，需要转换成UInt16[]；size为buffer数组的长度。关于byte[]转换成UInt16[]的方法比较简单，在此不介绍了。100Test 下载频道开通，各类考试题目直接下载。详细请访问

[www.100test.com](http://www.100test.com)