

本节内容

UDP 检验

一种新的差错检验方法

0101 0000 0110 0101
0110 1001 0101 0001

A

32bit数据+16bit检验和



B

0101 0000 0110 0101
0110 1001 0101 0001
0100 0110 0100 1001

0101 0000 0110 0101
+ 0110 1001 0101 0001
0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1

中间结果→ 1011 1001 1011 0110

逐位取反→ 0100 0110 0100 1001 (检验和)

- 如果没有比特错误，那么以16bit为一组，进行二进制加法（最高位产生的进位需要“回卷”），加法结果一定为全1；
- 如果发生了比特错误，加法结果就不是全1

一种新的差错检验方法

0101 0000 0110 0101
0110 1001 0101 0001
0110 0101 1100 0101

A

48bit数据+16bit检验和

B

0101 0000 0110 0101
0110 1001 0101 0001
0110 0101 1100 0101
1110 0000 1000 0011

0101 0000 0110 0101
+ 0110 1001 0101 0001
0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1

中间结果→ 1011 1001 1011 0110

+ 0110 0101 1100 0101
1 1 1 0 0 0 0 1 1 1 0 0 0 1 0 0
0001 1111 0111 1011

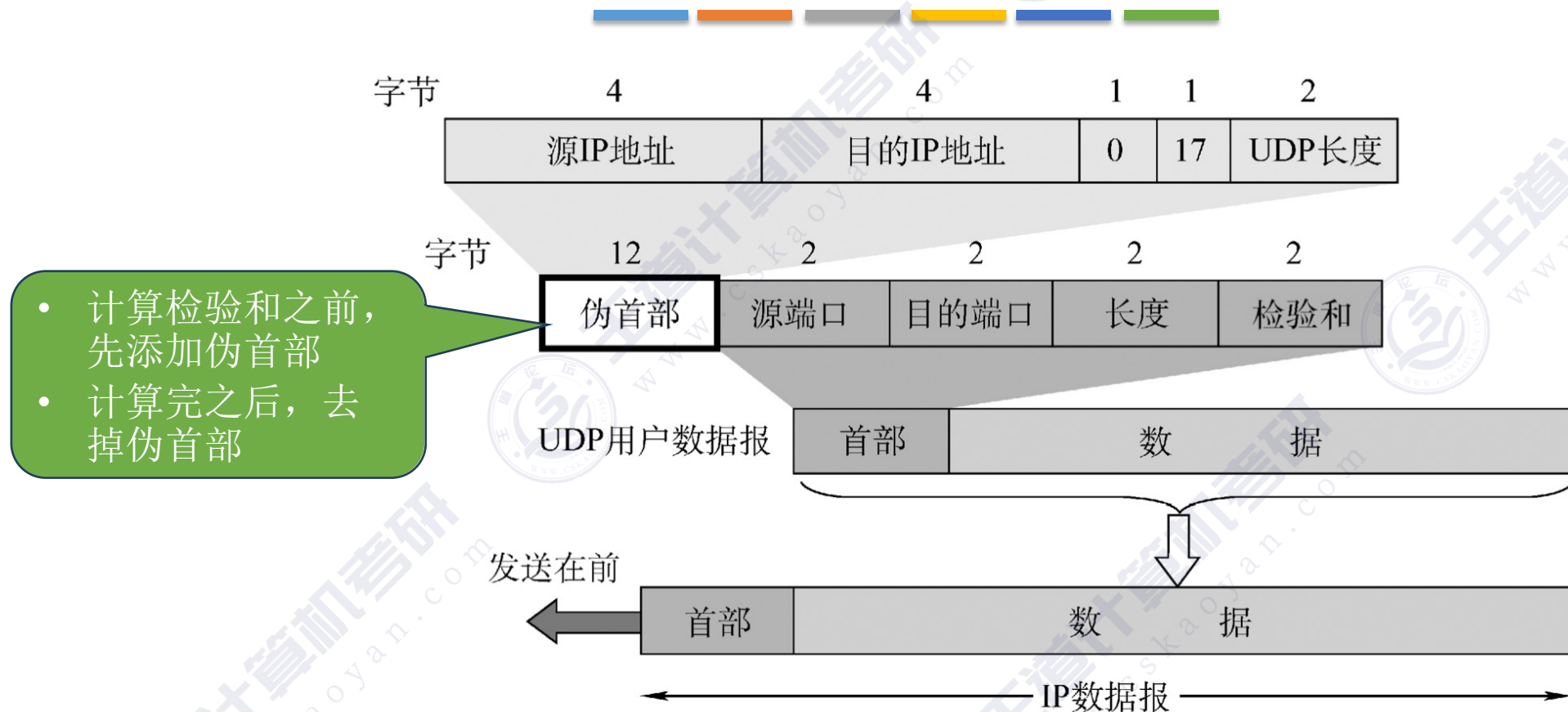
中间结果→ 0001 1111 0111 1100

逐位取反→ 1110 0000 1000 0011 (检验和)

如果最高位产生进位，就“回卷”加到最低位

- 如果没有比特错误，那么以16bit为一组，进行二进制加法（最高位产生的进位需要“回卷”），加法结果一定为全1；
- 如果发生了比特错误，加法结果就不是全1

UDP检验（发送方的传输层）



- ①传输层的UDP协议在计算检验和之前，先添加伪首部
- ②把伪首部、首部、数据部分以16bit为一组，进行二进制加法（最高位产生的进位需要回卷）
- ③将最终的加法结果逐位取反，就得到16bit检验和，将其填入UDP首部
- ④去掉伪首部，并将UDP数据报交给网络层，封装成IP数据报

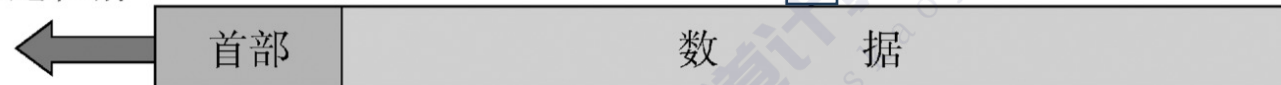
UDP检验（接收方的传输层）



- 检验之前，先添加伪首部
- 检验之后，去掉伪首部



发送在前



IP数据报

- ①网络层向传输层递交UDP数据报
- ②传输层在UDP数据报之前，添加伪首部
- ③把伪首部、UDP首部、数据部分以16bit为一组，进行二进制加法（最高位产生的进位需要回卷）
- ④如果加法结果为全1，说明没有比特错误，于是接收该UDP数据报，并根据目的端口号，向应用层递交报文。如果加法结果不是全1，说明有差错，于是丢弃该UDP数据报

2024真题_39题

39. 若UDP协议在计算校验和过程中，计算得到中间结果为1011 1001 1011 0110时，还需要加上最后一个16位数0110 0101 1100 0101，则最终计算得到的校验和是（ ）。
- A. 0001 1111 0111 1011 B. 0001 1111 0111 1100
C. 1110 0000 1000 0011 D. 1110 0000 1000 0100

注意：进行二进制加法时，最高位产生的进位需要回卷

UDP检验要点总结

UDP检验

一种新的差错检验算法

检验和的计算：将原始数据以16bit为一组，进行**二进制加法（最高位产生的进位需要回卷）**，**加法运算的最终结果逐位取反**，得到16bit“检验和”

差错检验方法：接收方将收到的内容以16bit为一组，**进行二进制加法（最高位产生的进位需要回卷）**

如果加法结果为全1，说明没有差错

如果加法结果不是全1，说明有差错

UDP校验要点

UDP检验和计算方法、差错检验算法如上所示↑

发送方在**计算检验和之前**，需要添加**伪首部**，
计算完检验和之后，拆除伪首部

接收方在**差错检验之前**，需要添加**伪首部**，
检验完之后，拆除伪首部

如果检验出差错，就丢弃该UDP数据报

如果无差错，就接收该UDP数据报

回顾：IP数据报（IP分组）的格式

