计算机网络编程

第7章　IP数据包的分片与重组

信息工程学院　　方徽星
fanghuixing@hotmail.com

大纲

- 设计目的
- 相关知识
- 例题分析

1. 设计目的

- IP数据包在传输过程中经常需要分片和重组
- 熟悉IP数据包的分片与重组，对于理解网络层次结构以及网络问题处理方法，具有重要的意义

2. 相关知识—IP包分片的概念

- **IP数据包**作为网络层数据必然**通过数据链路层**，**封装成帧**再通过物理层来传输
- IP包可能**经过多个不同的**物理网络
- 每个路由器需要对接收的帧解析拆帧与处理，然后**封装成某种类型的帧**来通过物理网络

2. 相关知识—IP包分片的概念

- 每种物理网络都规定了各自**帧的数据字段的最大长度**
- **最大传输单元(Maximum Transfer Unit, MTU)**

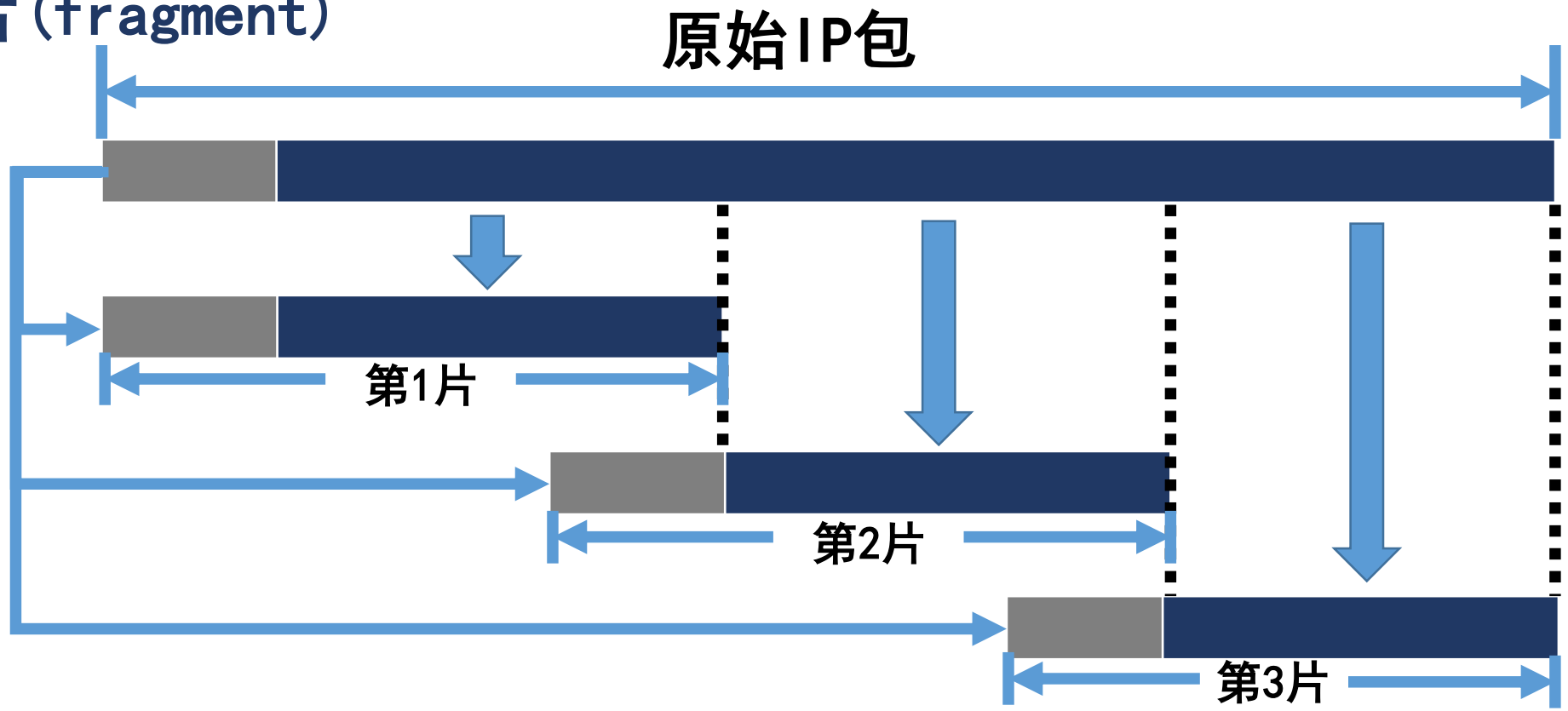
RFC1191

物理网络	最大传输单元(MTY)
16Mb IBM Token Ring	17914字节
FDDI	4352字节
Ethernet	1500字节
X.25	576字节
PPP	296字节

多数物理网络MTU比IP包最大长度(65535B)短

2. 相关知识—IP包分片的概念

- 当使用物理网络来传输IP包时，经常需要将IP包分成若干个较小的片(fragment)



2. 相关知识—IP包分片的相关字段

- 标识符(Identification)
  - 字段长度：16 bits
  - 一个IP包的所有分片可以分配一个标识符
  - 最多可以分配65535个标识符

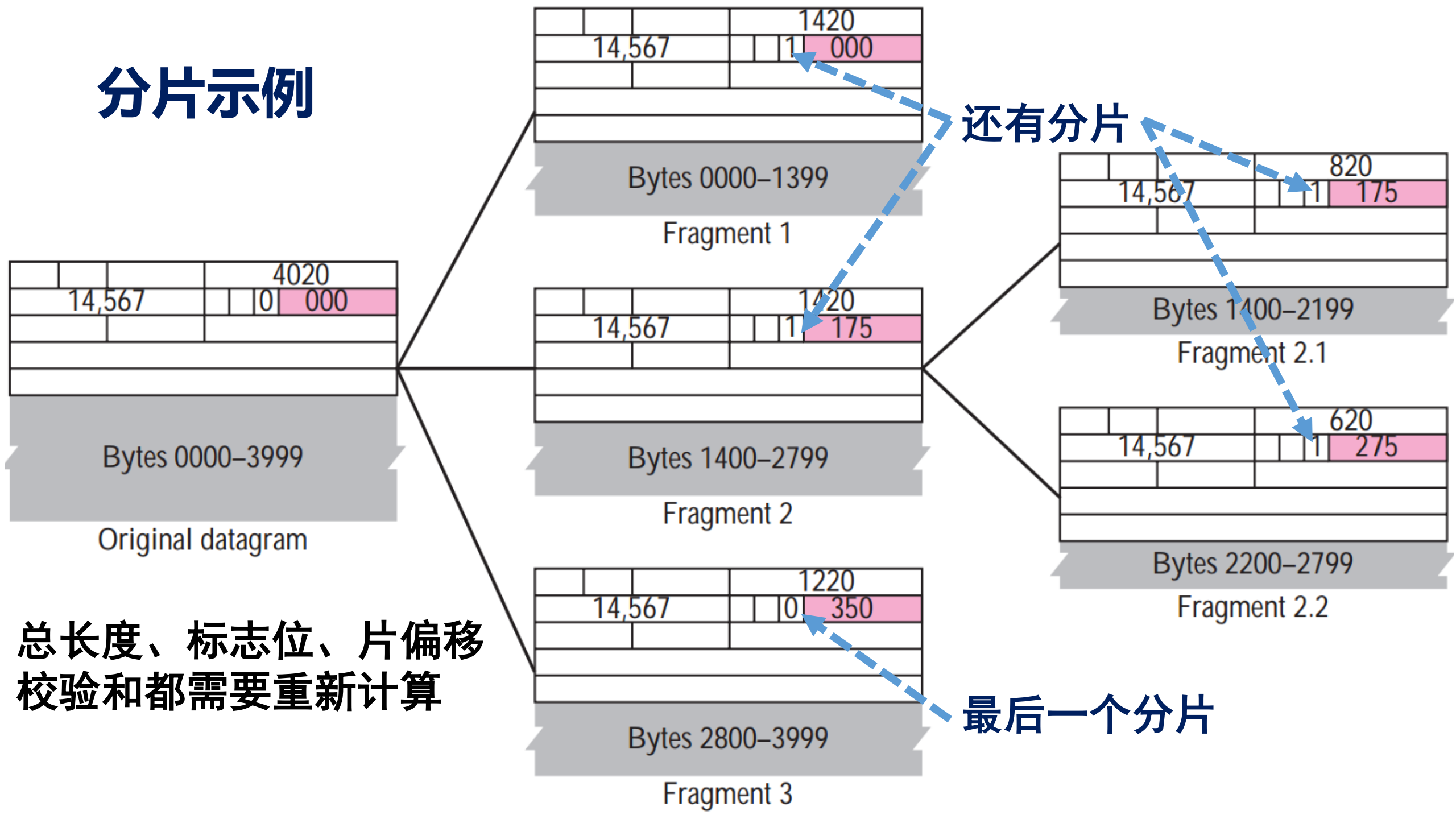
2. 相关知识—IP包分片的相关字段

- 标志位(flags)
  - 字段长度：3 bits
  - 如果IP包长度超过MTU，同时IP包不能分片，则该包被丢弃，并发送ICMP包向源主机报告

2. 相关知识—IP包分片的相关字段

- 片偏移
  - 字段长度13 bits
  - 表示分片在整个IP包中的相对位置
  - 以8字节为基本单位计数

分片示例



总长度、标志位、片偏移
校验和都需要重新计算

3. 例题分析—设计要求

- 根据IPv4协议规定的IP包的标准格式，编写程序来对现有的IP包进行分片
- 将分片后的IP头部与数据字段写入输出文件
- 数据字段值从指定的输入文件中获得
- 本练习为了简便起见，自行填写IP头部中除校验和外的各字段
- 以80B为单位对IP包进行分片

3. 例题分析—设计要求

- 具体要求
  - 要求程序为命令行程序。例如，可执行文件名PackFrag.exe，则程序的命令行格式为：

```
PackFrag  input_file  output_file
```

其中，input_file为输入文件，output_file为输出文件

3. 例题分析—设计要求

- 具体要求
  - 要求将部分字段内容显示在控制台上，具体格式为：

```
IP包1开始封装
总长度：xx
标识符：xx
标志位：xx，DF，MF
片偏移：xx
头部校验和：xx
数据字段：…
IP包2开始封装
…
```

3. 例题分析—设计要求

- 具体要求
  - 有良好的编程规范与注释。编程所使用的操作系统、语言和编译环境不限，但是在提交的说明文档中需要加以注明
  - 撰写说明文档，包括程序的开发思路、工作流程、关键问题、解决思路以及进一步的改进等内容

3. 例题分析—关键问题

- 填充分片头部字段
  - IP包的分片采用的是与IP头部相同的结构
  - 构造一个IP头部的数据结构
  - 依次填充IP头部中的各个字段

3. 例题分析—关键问题

```
If(数据长度<分片长度)
{
        bpacket = false; //最后一个分片
        npacklen = nlength; //分片(数据)长度等于数据长度
        ip.Flags = unsigned short ( (npacknum-1) * 80/8 );
}
Else
{
        nlength = nlength – 80; //数据长度减少80字节
        npacklen = 80; //分片(数据)为80字节
        ip.Flag = unsigned short( ((npacknum-1)*80/8)  | 0x2000);
}
```

设置标志位字段

第npacknum(1,2,3,…)个分片

前3位001
还有分片

16bits

3. 例题分析—关键问题

- 计算头部校验和函数

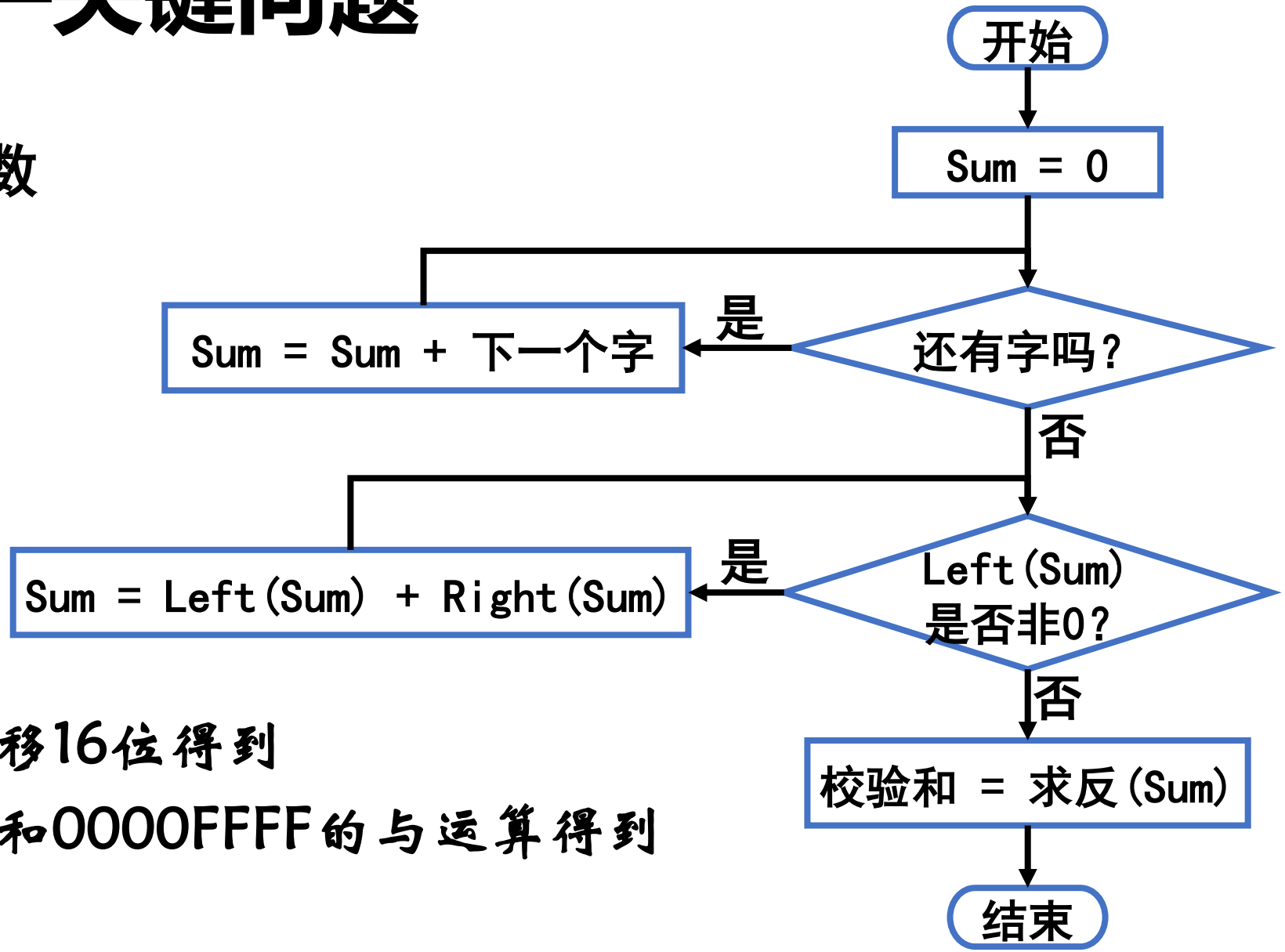
字 16位

校验和 16位

Sum 32位

Left(Sum) 把Sum右移16位得到

Right(Sum) 通过Sum和0000FFFF的与运算得到



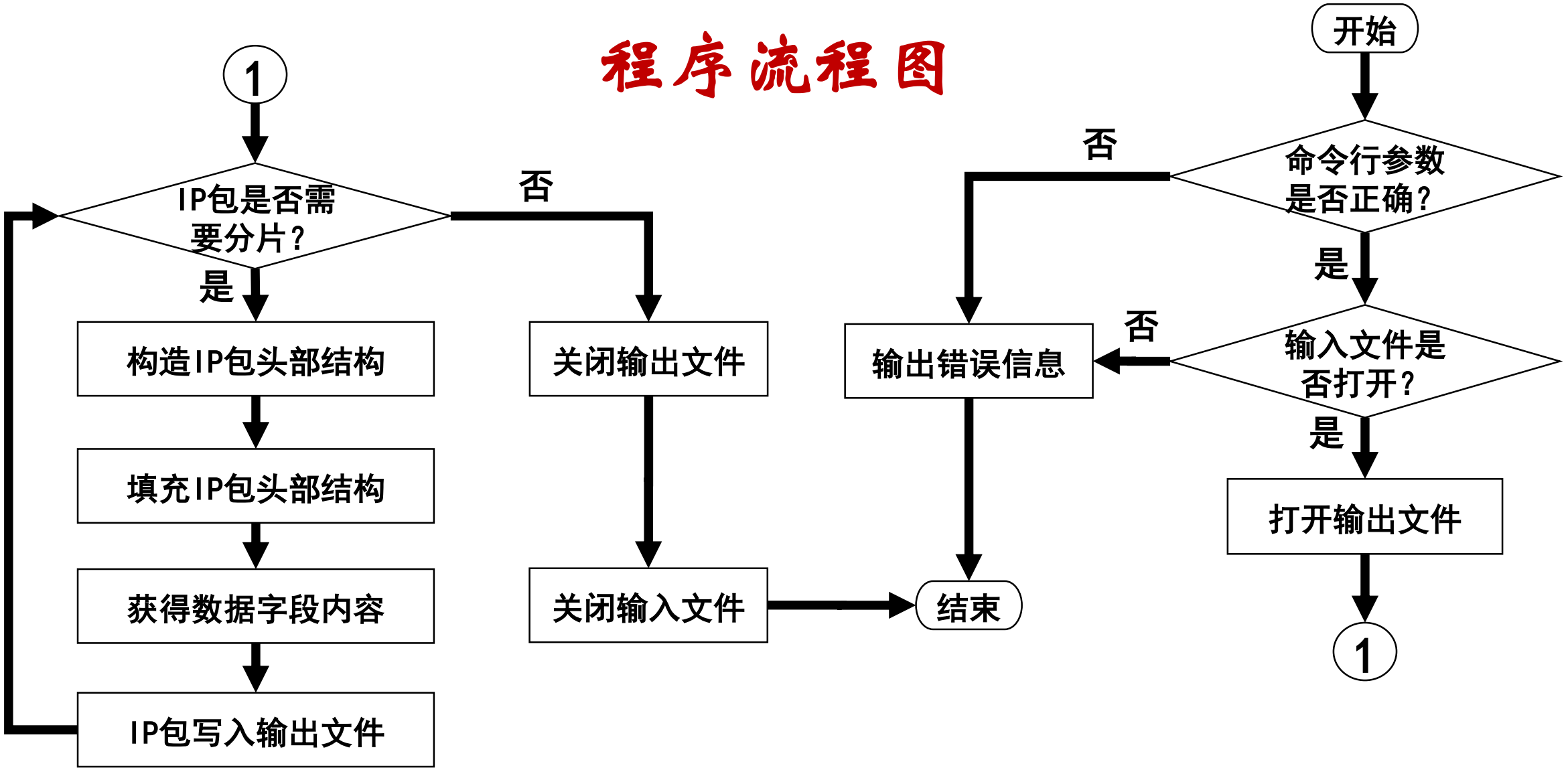
3. 例题分析—关键问题

```
unsigned short checksum(unsigned short * buffer, int size)
{
        unsigned long sum = 0;
        while(size > 0)  {
                sum = sum +  (*buffer);
                buffer ++;
                size = size – sizeof(unsigned short);
        }
        while((sum>>16) != 0) {
                sum = (sum>>16) + (sum & 0xffff);
        }
        return (unsigned short)(~sum);
}
```

计算头部校验和函数

3. 例题分析——关键问题

程序流程图



3. 例题分析—程序演示


Microsoft Visual Studio 调试控制台
IP包1开始封装
总长度：100
标识符：4096
标志位：0，DF=0，MF=1
片偏移：0(8B)
头部校验和：2642
数据字段：Nankai University was founded in 1919 by the famous patriotic educators in China
IP包2开始封装
总长度：72
标识符：4096
标志位：0，DF=0，MF=0
片偏移：10(8B)
头部校验和：10852
数据字段：se modern history，Mr. Zhang Boling and Mr. Yan Xiu.
IP包分片封装完成

本章小结

- 设计目的
  - IP数据包在传输过程中需要分片和重组，深入理解网络层次关系
- 相关知识
  - 分片的概念，MTU
  - 分片相关字段：标识符、标志位、片偏移
- 例题分析
  - 填充字段、计算校验和