

计算机网络编程

第8章 IPv6数据包的封装与解析

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

- IPv6协议是针对当前IP协议问题制定的下一代协议标准
- 根据IPv6包标准格式，通过封装与解析IPv6包
 - 了解IPv6协议的基本概念
 - 深入理解下一代网络层协议的工作过程

2. 相关知识

- **IPv4协议的主要缺点**

- 标准分类地址利用率低，地址数量不能满足网络规模不断扩展的需要
- 随着网络结构越来越复杂，路由选择算法的研究越来越显得困难
- IPv4协议对分组传输可靠性没有提供任何保障措施
- IPv4协议不支持多播传输
- IPv4协议不能保证分组传输的服务质量
- IPv4协议对网络安全问题没有提出对策

2. 相关知识

- IPv6协议的基本概念

- IPv6是由多个层次的一系列相关协议所构成的协议集

1992: IETF成立IPng工作组

1995: Cisco和Nokia
起草IPv6最初方案

1998: IETF正式公布
IPv6协议标准RFC2460

2001: 主流操作系统
开始支持IPv6

2003: 我国启动CNGI
中国下一代互联网示范工程

1994: IPng工作组公布
RFC1726, 提出18个选择方案

1996: IETF启动建立全球
IPv6实验床6Bone

1999: IETF成立IPv6论坛
开始分配IPv6地址

2003: 主要网络设备制造商
开始提供IPv6设备

2. 相关知识

- IPv6协议的特点

<p>新的协议 头部格式</p>	<ul style="list-style-type: none">✓ 基本头部长长度固定✓ 将一些非根本性与可选的字段移到扩展头部✓ 使路由器在处理协议头部时效率更高
----------------------	--

2. 相关知识

- IPv6协议的特点

巨大的地址空间	<ul style="list-style-type: none">✓ 从IPv4的32位增大到128位✓ 从根本上解决IP地址匮乏问题✓ 不再使用带来很多问题的NAT技术
----------------	--

2. 相关知识

- IPv6协议的特点

<p>有效的分层 路由结构</p>	<ul style="list-style-type: none">✓ 更好地将路由结构划分出层次，可以覆盖从各级主干网直到内部网的多级结构✓ 64位可作为子网地址空间使用✓ 64位用于映射网卡硬件地址
-----------------------	---

2. 相关知识

- IPv6协议的特点

灵活的地址 自动配置	<ul style="list-style-type: none">✓ 有状态地址自动配置✓ 无状态地址自动配置✓ 无须用户干预，自动获得可用IP地址
-----------------------	---

2. 相关知识

- IPv6协议的特点

<p>内置的安全性服务</p>	<ul style="list-style-type: none">✓ IPSec协议作为一个IPv6的组成部分而使用✓ IPSec可用提供主机IP地址认证、数据完整性验证与数据加密等功能
------------------------	---

2. 相关知识

- IPv6协议的特点

<p>更好地支持 QoS</p>	<ul style="list-style-type: none">✓ IPv6头部的流标记字段定义如何识别通信流✓ 路由器可对属于一个流的数据包进行特殊处理
----------------------	--

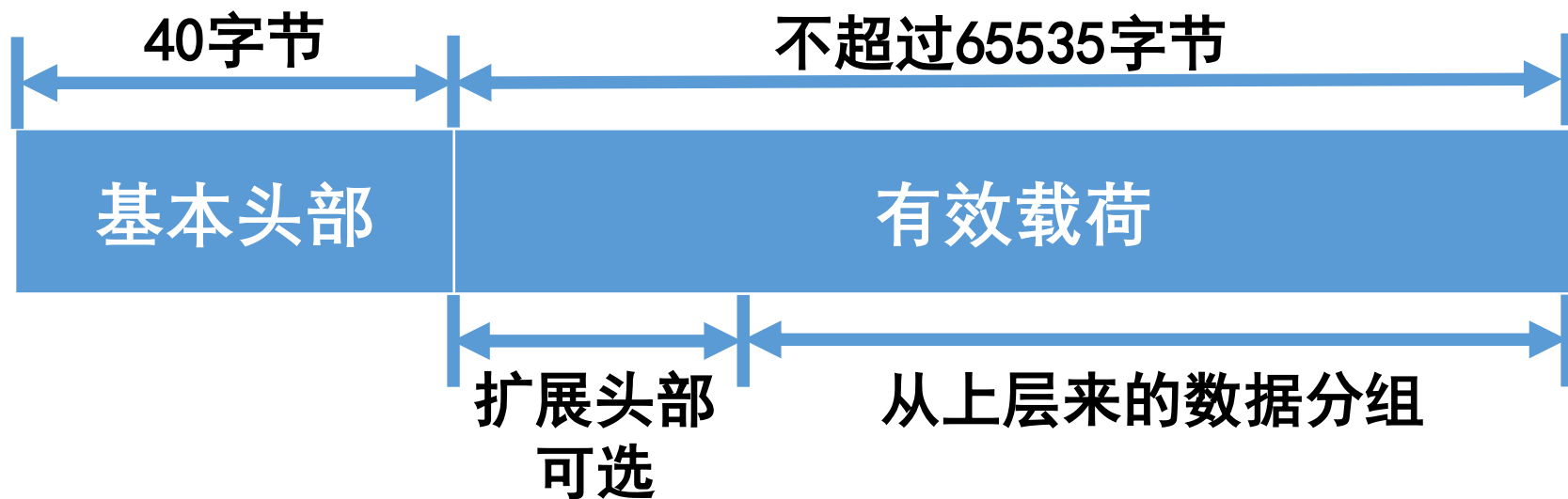
2. 相关知识

- IPv6协议的特点

<p>良好的可扩展性</p>	<ul style="list-style-type: none">✓ IPv6支持在基本头部之后定义扩展头部，方便地实现对新增网络应用的支持和扩展✓ IPv4头部最多只能40B✓ IPv6扩展头部长度只受IPv6包长度的限制
----------------	---

2. 相关知识

- IPv6数据包的结构



2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (2字节)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

版本： 字段值6

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (2字节)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

优先级：路由器通过该字段值决定在网络拥塞时如何处理数据包

0~7：在拥塞时允许延时处理（默认值0）

8~15：优先级较高的实时业务需要使用固定速率传输

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (2字节)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

流标号：表示数据包属于源主机与目的主机之间的某个数据流；默认值0

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (16位)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

有效载荷长度：表示数据包中除了基本头部之外的数据长度；最大长度65535字节

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (16位)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

下一个头部：表示基本头部后面的头部

- 可能是IP自己要使用的一个可选的扩展首部
- 也可能是被封装的分组的首部，如UDP或TCP
- 每一个扩展头部中也包含该字段

2. 相关知识

下一个头部的代码

代码	下一个头部	代码	下一个头部
0	逐跳头部	44	分片头部
2	ICMP	50	加密的安全有效载荷
6	TCP	51	认证
17	UDP	59	空(没有下一个头部)
43	源路由选择	60	目的地址选项

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (16位)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

跳步限制：表示数据包可以通过的最大的路由器转发次数；每经过一个路由器，该字段值减1；为0时，路由器丢弃该数据包，并向源主机发送ICMPv6报文

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (16位)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

源IP地址：发送数据包的源主机IPv6地址

目的IP地址：接收数据包的目的地主机IPv6地址

2. 相关知识

版本 (4位)	优先级 (8位)	流标号 (20位)	
有效载荷长度 (16位)		下一个头部 (8位)	跳步限制 (8位)
源IP地址 (128位)			
目的IP地址 (128位)			
扩展头部与数据部分			

扩展头部：用来扩展协议功能的部分；长度是8B的整数倍；多个扩展头部形成链状结构

2. 相关知识

逐跳头部



目的地址选项头部



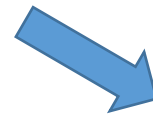
路由头部



分片头部



认证头部

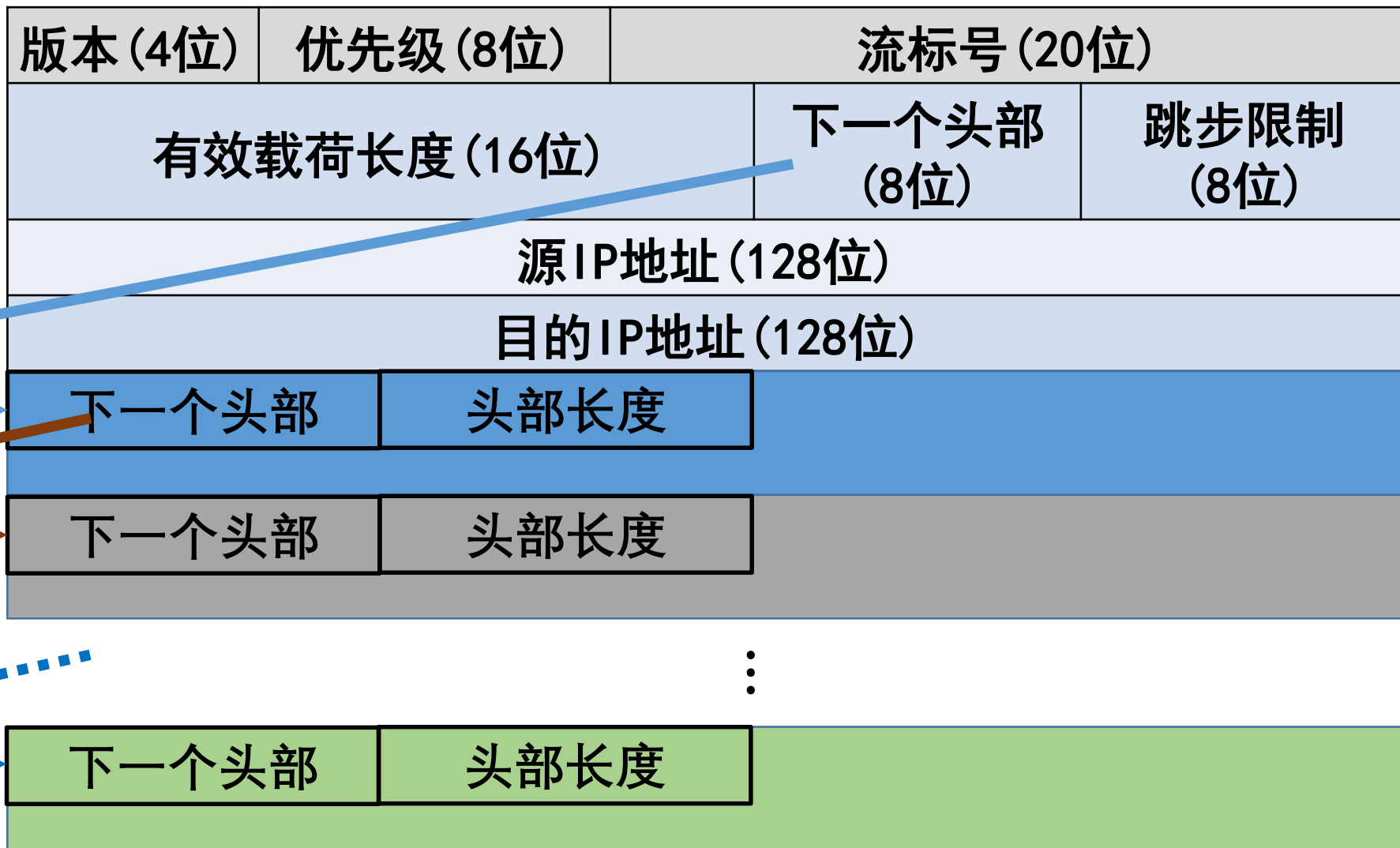


封装安全载荷头部

IPv6扩展头部的
排列顺序

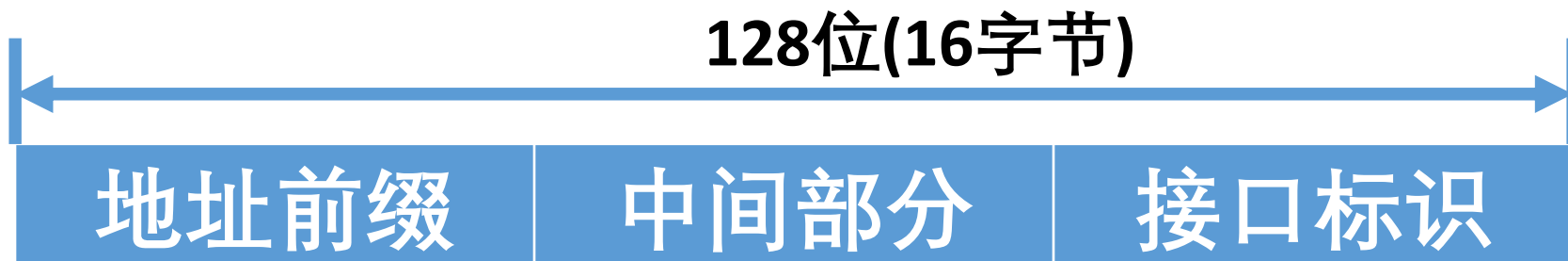
除了目的地址选项头部之外，其他扩展头部在IPv6数据包中只能出现一次

2. 相关知识



2. 相关知识

- IPv6地址结构



16进制冒号记法:

FDEC:BA98:7654:3210:ADBF:BBFF:2922:FFFF

2. 相关知识

- IPv6地址结构

FDEC:0:0:0:0:BBFF:0:FFFF

零压缩



FDEC::BBFF:0:FFFF

2. 相关知识

- IPv6地址结构

CIDR地址

FDEC::BBFF:0:FFFF/60



60位前缀

2. 相关知识

- IPv6地址结构

三种 目的地址

单播地址：定义了一个接口(计算机或路由器)。发送到单播地址的分组必须交付给这个指定的计算机

任播地址：定义了一组共享一个地址的计算机。发送到任播地址的分组必须交付给这个组的成员之一，也就是最容易到达的那一个

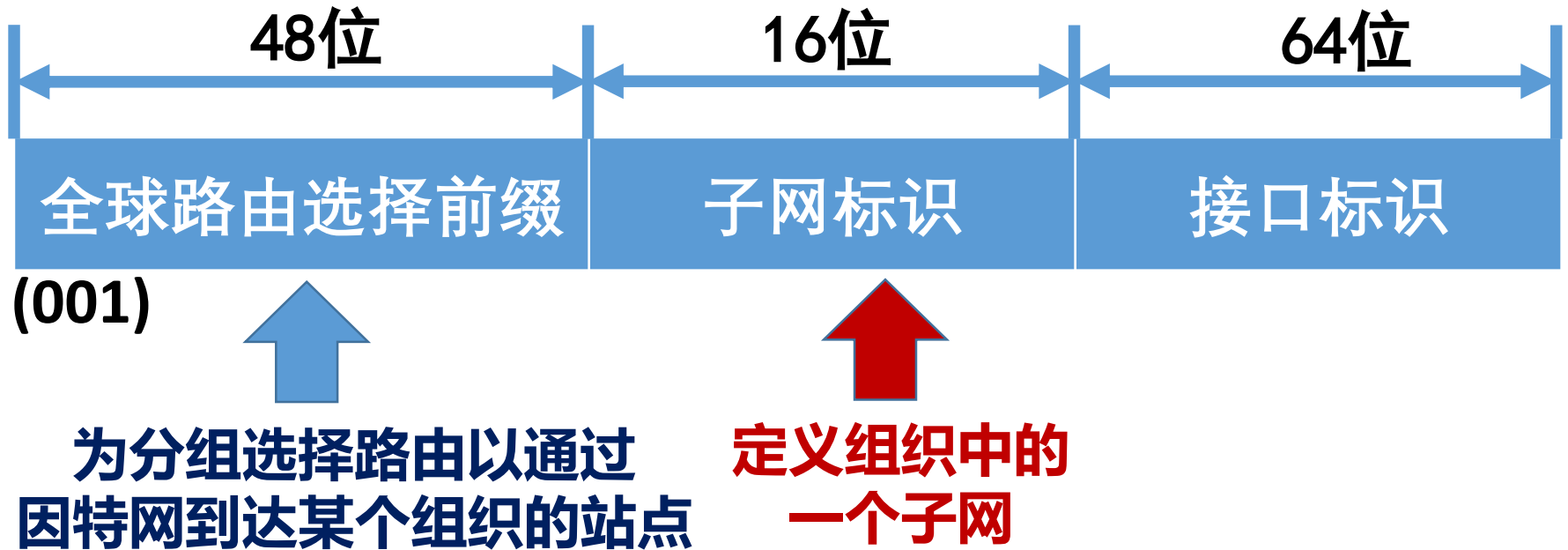
多播地址：也定义了一组计算机。在多播通信中，组中所有成员均会收到一个分组副本

2. 相关知识

	<i>Block Prefix</i>	<i>CIDR</i>	<i>Block Assignment</i>	<i>Fraction</i>
1	0000 0000	0000:: 8</td <td>Reserved (IPv4 compatible)</td> <td>1/256</td>	Reserved (IPv4 compatible)	1/256
	0000 0001	0100:: 8</td <td>Reserved</td> <td>1/256</td>	Reserved	1/256
	0000 001	0200:: 7</td <td>Reserved</td> <td>1/128</td>	Reserved	1/128
	0000 01	0400:: 6</td <td>Reserved</td> <td>1/64</td>	Reserved	1/64
	0000 1	0800:: 5</td <td>Reserved</td> <td>1/32</td>	Reserved	1/32
	0001	1000:: 4</td <td>Reserved</td> <td>1/16</td>	Reserved	1/16
2	001	2000::<!--3</b-->	Global unicast	1/8
3	010	4000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
4	011	6000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
5	100	8000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
6	101	A000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
7	110	C000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
8	1110	E000:: 4</td <td>Reserved</td> <td>1/16</td>	Reserved	1/16
	1111 0	F000:: 5</td <td>Reserved</td> <td>1/32</td>	Reserved	1/32
	1111 10	F800:: 6</td <td>Reserved</td> <td>1/64</td>	Reserved	1/64
	1111 110	FC00:: 7</td <td>Unique local unicast</td> <td>1/128</td>	Unique local unicast	1/128
	1111 1110 0	FE00:: 9</td <td>Reserved</td> <td>1/512</td>	Reserved	1/512
	1111 1110 10	FE80:: 10</td <td>Link local addresses</td> <td>1/1024</td>	Link local addresses	1/1024
	1111 1110 11	FEC0:: 10</td <td>Reserved</td> <td>1/1024</td>	Reserved	1/1024
	1111 1111	FF00:: 8</td <td>Multicast addresses</td> <td>1/256</td>	Multicast addresses	1/256

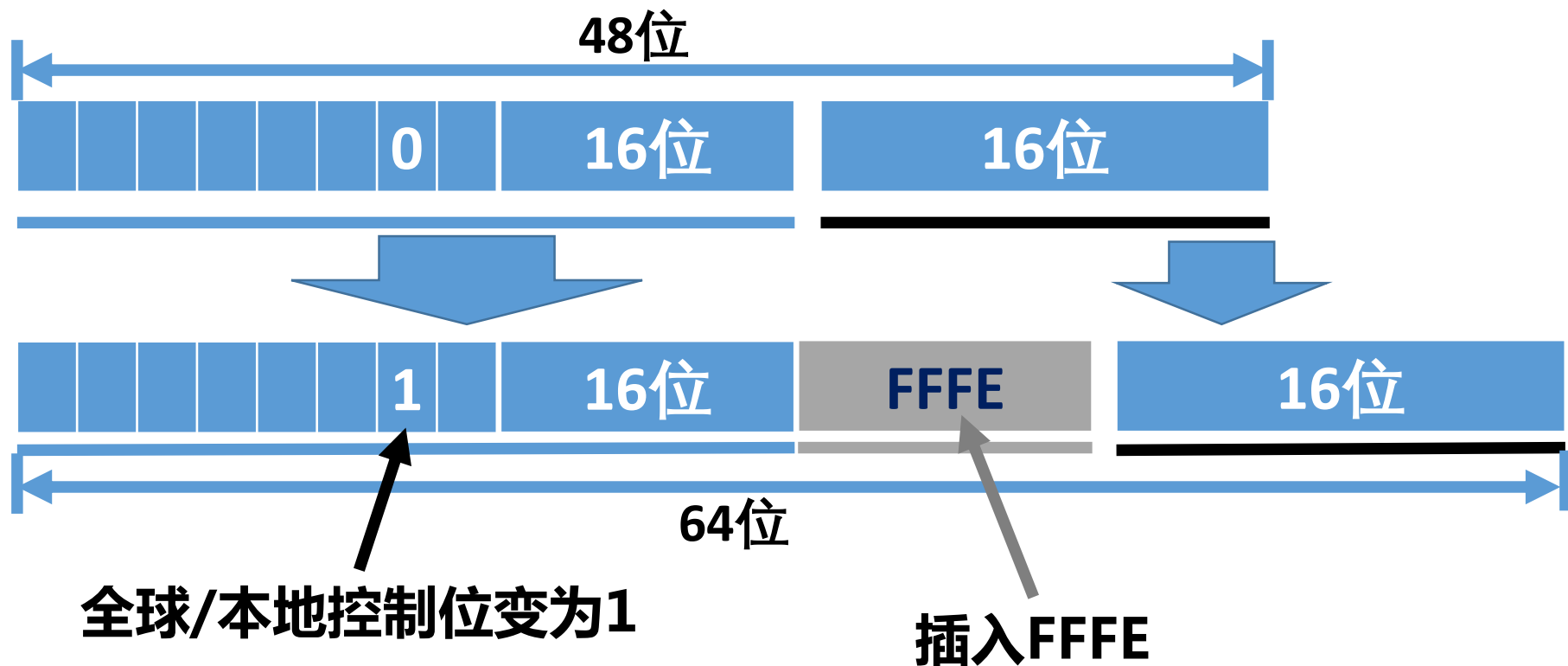
2. 相关知识

- 全球单播地址块(2000::/3)
 - 用于因特网上两个主机之间的单播通信



2. 相关知识

将一个48位以太网地址映射为一个64位接口标识



3. 例题分析—设计要求

- 设计要求

- 根据协议规定的IPv6数据包的标准格式，编写程序构造IPv6包结构(包括IPv6头部与TCP头部)
- 然后将封装后的IPv6包内容写入输出文件
- 不需要构造任何IPv6扩展头部
- 数据字段通过字符串赋值来获得
- 需要计算TCP头部与数据部分的校验和

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，可执行文件名为 `Ipv6Encap.exe`，则命令行格式为：

`Ipv6Encap output_file`

- 要求将部分字段内容显示在控制台上，具体格式：

IP头部与数据字段

版本：`xx`

有效载荷长度：`xx`

下一个头部：`xx`

源IP地址：`xx:xx:xx:xx:xx:xx:xx:xx`

目的IP地址：`xx:xx:xx:xx:xx:xx:xx:xx`

数据字段：`...`

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

- 定义IPv6头部的数据结构

```
typedef struct IP_HEAD  
{
```

```
    union  
    {
```

```
        unsigned int Version;  
        unsigned int Priority;  
        unsigned int FLOWLabel;
```

```
    }  
    unsigned short PayloadLen;  
    unsigned char NextHead;  
    unsigned char HopLimit;
```

```
    struct  
    {
```

```
        __int64 Prefix;  
        unsigned char MacAddr[8];  
    }SourceAddr;
```

```
    struct  
    {
```

```
        __int64 Prefix;  
        unsigned char MacAddr[8];  
    }DestinAddr;
```

```
};  
}ip_head;
```

3. 例题分析—关键问题

- 填充IPv6包的各个字段
 - 在对IPv6包内容封装之前，需要分别填充IPv6头部、TCP头部与TCP数据

```
//填充3位地址前缀  
ip.SourceAddr.Prefix = 0x01;  
//填充45位路由前缀  
ip.SourceAddr.Prefix <<= 45;  
ip.SourceAddr.Prefix += 0x01;  
//填充16位子网号  
ip.SourceAddr.Prefix <<= 16;  
ip.SourceAddr.Prefix += 0x01;  
ip.SourceAddr.Prefix = hton64(ip.SourceAddr.Prefix) ;
```

3. 例题分析—关键问题

- 填充IPv6包的各个字段
 - 在对IPv6包内容封装之前，需要分别填充IPv6头部、TCP头部与TCP数据

```
//MAC地址转成接口标识
```

```
ip.SourceAddr.MacAddr[0] = char(0x02);
```

```
ip.SourceAddr.MacAddr[1] = char(0x00);
```

```
ip.SourceAddr.MacAddr[2] = char(0x80);
```

```
ip.SourceAddr.MacAddr[3] = char(0xFF);
```

```
ip.SourceAddr.MacAddr[4] = char(0xFE);
```

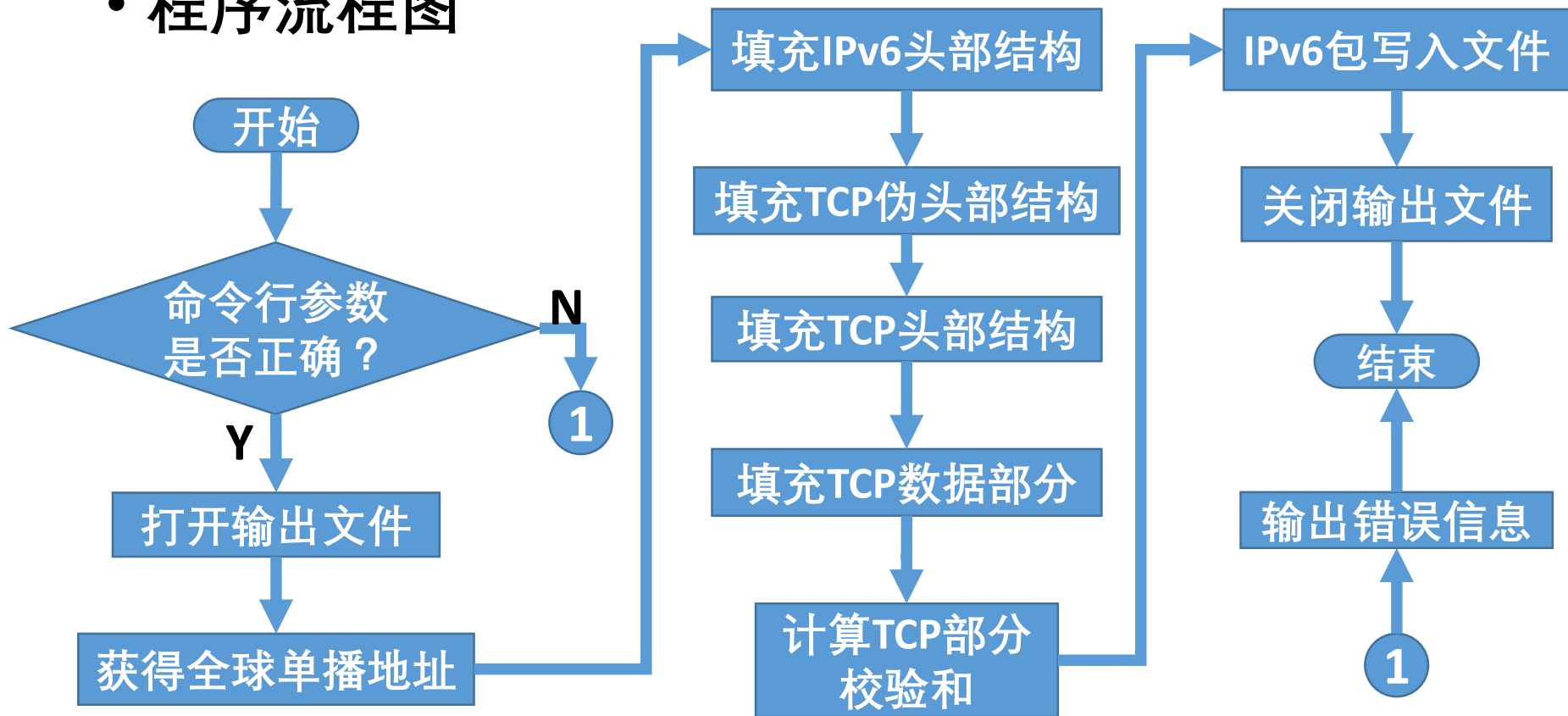
```
ip.SourceAddr.MacAddr[5] = char(0x18);
```

```
ip.SourceAddr.MacAddr[6] = char(0x6E);
```

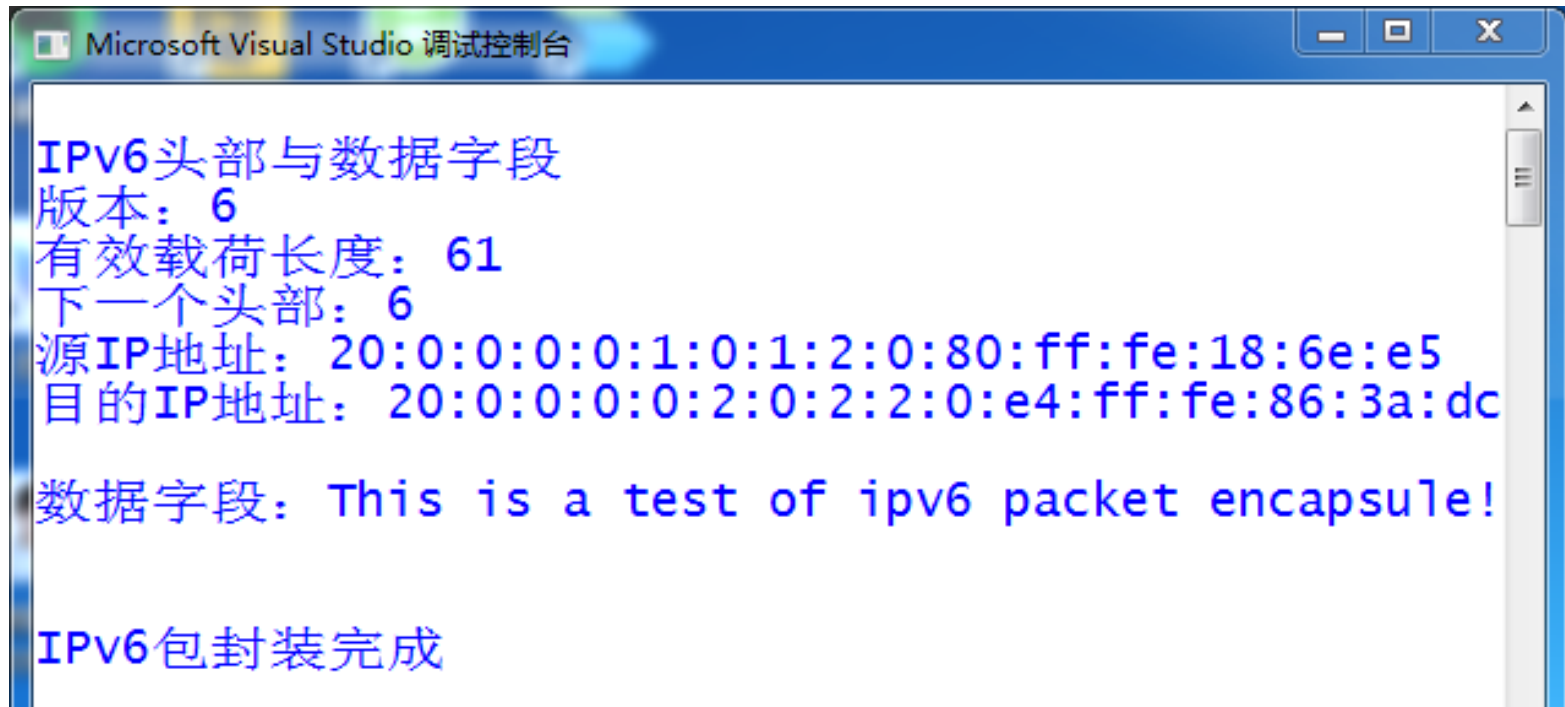
```
ip.SourceAddr.MacAddr[7] = char(0xE5);
```

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

IPv6头部与数据字段
版本： 6
有效载荷长度： 61
下一个头部： 6
源IP地址： 20:0:0:0:0:1:0:1:2:0:80:ff:fe:18:6e:e5
目的IP地址： 20:0:0:0:0:2:0:2:2:0:e4:ff:fe:86:3a:dc

数据字段： This is a test of ipv6 packet encapsule!

IPv6包封装完成
```

本章小结

- 设计目的

- 了解IPv6协议的基本概念，理解协议工作过程

- 相关知识

- IPv4主要缺点
- IPv6特点、数据包结构、地址结构

- 例题分析

- IPv6头部数据结构
- 填充字段