计算机网络编程

第9章 发现网络中的活动主机

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协议缺少差错控制与查询机制
- **ICMP**(Internet Control Message Protocol)协议可以补充IP的功能
- 通过封装、发送、接收与解析ICMP数据包
  - 了解**ICMP**包结构中各个字段的用途
  - 深入理解与认识**ICMP**协议的作用

2. 相关知识

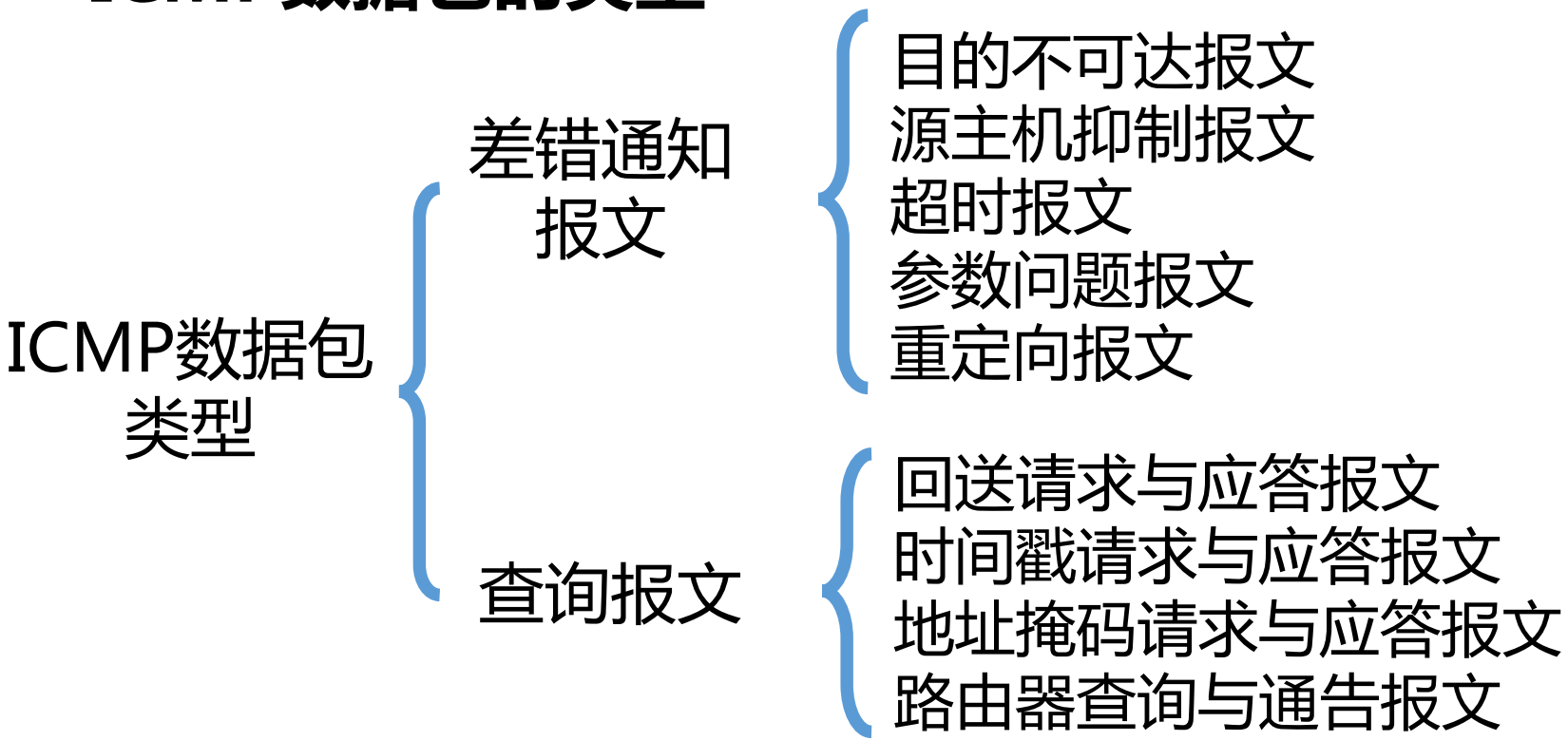
- **ICMP协议的基本概念**
  - IP协议缺少差错控制与查询机制
    - 源节点无法知道IP数据包是否到达目的节点
    - 目的节点也无法知道在传输过程中出现哪种错误
  - 互联网控制报文协议(ICMP)是为解决上述问题而设计的

2. 相关知识

- **ICMP协议的基本概念**
  - ICMP协议是**网络层**的一个协议
  - ICMP数据包**不是直接传送**给数据链路层，而是**封装成IP数据包**再进行传送
  - IP包头部中**协议字段值为1**，表示ICMP数据包

2. 相关知识

- **ICMP数据包的类型**



2. 相关知识

- **ICMP数据包的类型**

差错通知报文——目的不可达报文

- 当路由器无法为一个数据包找到路由
- 或主机无法交付一个数据包时，该数据包被丢弃
- 然后由路由器或主机向源主机返回一个目的不可达报文

2. 相关知识

- **ICMP数据包的类型**

差错通知报文——源主机抑制报文

- 当路由器或主机因拥塞而丢弃数据包时，它就向该数据包的发送方发送一个源主机抑制报文
- 通知源主机，数据包已被丢弃
- 警告源主机，在路径中的某处出现了拥塞，因而必须放慢（抑制）发送过程

2. 相关知识

- **ICMP数据包的类型**

差错通知报文——超时报文

- 路由器发现数据包中生存时间字段减1之后变为0，就丢弃该数据包，并向源主机发送一个超时报文
- 当组成一个报文的所有分片未能在某时限内全部到达目的主机，则丢弃已收到的所有分片，并发送超时报文

2. 相关知识

- **ICMP数据包的类型**

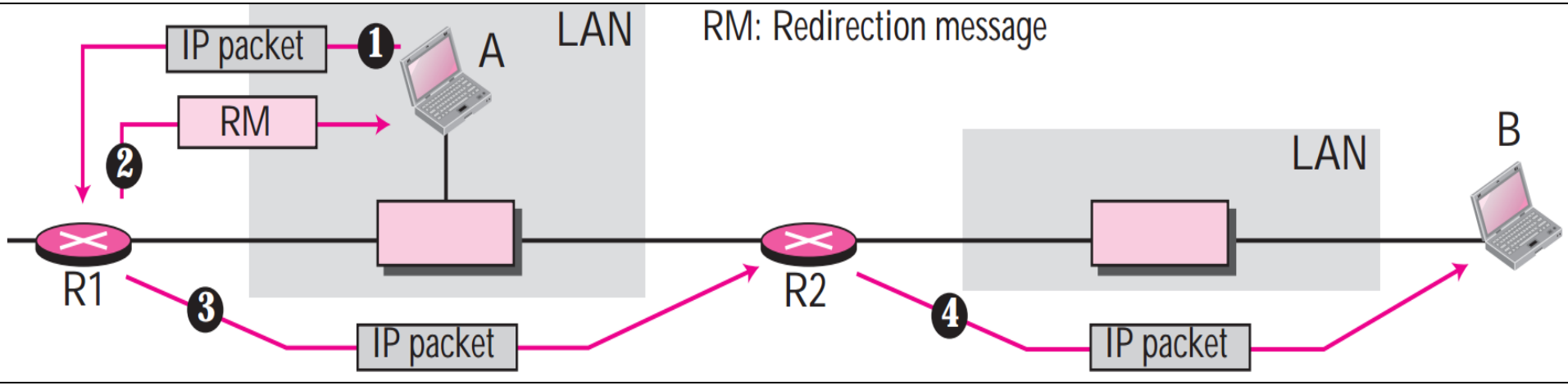
差错通知报文——参数问题

- 如果路由器或者目的主机发现数据包首部中出现二义性或者在数据包的某个字段中缺少某个值，就会丢弃这个数据包，并向源主机返回一个参数问题报文

2. 相关知识

- **ICMP数据包的类型**

差错通知报文——重定向报文



- 主机A打算向主机B发送一个数据包，应选路由器R2
- 但A没有选R2，数据包被发送给路由器R1
- R1在查找路由表后发现应把数据包发给R2
- R1把数据包发给R2，并向A发送重定向报文，A更新路由表

2. 相关知识

- **ICMP数据包的类型**

查询报文——回送请求与应答报文

- 回送请求报文可以由**主机或路由器**发送。收到回送请求报文的主机或路由器发送回送应答报文
- 回送请求和回送应答报文可被网络管理员用来**检查IP协议的工作情况**
- 用回送请求和回送应答报文可测试某个**主机的可达性**。通常是调用ping命令来实现的

2. 相关知识

- **ICMP数据包的类型**

查询报文——时间戳请求与应答报文

- 可以用来计算源点和终点主机之间的往返时间，哪怕它们的时钟没有同步
- 如果已知准确的单向经历时间，则时间戳请求与应答报文可用来同步两个主机的时钟

2. 相关知识

- **ICMP数据包的类型**

查询报文

路由器
查询与通告报文

地址掩码
请求与应答报文

由动态主机配置协议（DHCP）完成

2. 相关知识

- **ICMP数据包的结构**

类型 （8位）	代码 （8位）	头部校验和 （16位）
数据部分 （由类型与代码字段共同决定）		

类型：表示ICMP报文的基本类型；例如，3表示目的不可达报文，5表示重定向报文

2. 相关知识

- **ICMP数据包的结构**

类型（8位）	代码（8位）	头部校验和（16位）
数据部分（由类型与代码字段共同决定）		

类型	代码	功能描述
3(目的不可达报文)	0	网络不可达
	1	主机不可达
	2	协议不可达
	…	…

2. 相关知识

- **ICMP数据包的结构**

类型 （8位）	代码 （8位）	头部校验和 （16位）
数据部分 （由类型与代码字段共同决定）		

头部校验和：用来检查ICMP包头部在传输中是否出错；与IP头部校验和的计算方法相同

2. 相关知识

- ICMP回送请求与应答
  - 本章课题是判断网络中的主机状态，使用ICMP的回送请求与应答报文

类型(8或者0)	代码(0)	头部校验和
标识符（16位）		序列号（16位）
可选数据 由请求报文发送，被回送报文重复		

类型8：回送请求
类型0：回送应答

标识符和序列号在协议中没有正式定义

2. 相关知识

- **ICMP回送请求与应答**
  - **源节点向目的节点发送ICMP回送请求后，等待目的节点返回ICMP回送应答**
  - **如果源节点在规定时间内收到应答信息，则说明目的节点处于活动状态**
  - **否则，目的节点处于关闭或不应答状态**

3. 例题分析

- **设计要求**
  - 根据协议规定的ICMP数据包的标准格式，编写程序向目的主机发送ICMP回送请求
  - 解析目的主机返回的ICMP回送应答，判断目的主机是否处于活动状态

3. 例题分析

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

ScanHost host_addr

目的主机IP地址

3. 例题分析

- **具体要求**
  - 要求将目的主机状态显示在控制台上，具体格式为：

开始主机扫描
目的主机+IP地址：活动状态（或关闭状态）

3. 例题分析

- 关键问题：创建原始套接字

```
//使用2.2版本winsock
WSAStartup(MAKEWORD(2, 2), &WSAData);
//创建原始套接字
sock=socket(AF_INET, SOCK_RAW, IPPROTO_ICMP);
//设置发送超时
int send_timeout = 1000; //1000ms
setsockopt(sock,SOL_SOCKET,SO_SNDTIMEO,
&send_timeout, sizeof(send_timeout));
```

SOL_SOCKET：表示在套接字级别上设置选项

3. 例题分析

- 关键问题：创建原始套接字

```
//使用2.2版本winsock
WSAStartup(MAKEWORD(2, 2), &WSAData);
//创建原始套接字
sock=socket(AF_INET, SOCK_RAW, IPPROTO_ICMP);
//设置发送超时
int send_timeout = 1000; //1000ms
setsockopt(sock,SOL_SOCKET,SO_SNDTIMEO,
&send_timeout, sizeof(send_timeout));
```

SO_SNDTIMEO：表示发送超时

3. 例题分析

- 关键问题：创建原始套接字

```
…
//设置接收超时
int recv_timeout = 1000; //1000ms
setsockopt(sock,SOL_SOCKET,SO_RCVTIMEO, &
recv_timeout, sizeof(recv_timeout));
```

SO_RCVTIMEO：表示接收超时

3. 例题分析

- 关键问题：定义ICMP头部的数据结构

```
typedef  struct ICMP_Head
{
   unsigned char Type;//基本类型(8位)
   unsigned char Code;//代码(8位)
   unsigned short HeadChecksum;//校验和(16位)
   unsigned short Identifier;//标识符(16位)
   unsigned short Sequence;//序列号(16位)
}icmp_head;
```

3. 例题分析

- **关键问题：填充与发送ICMP包**

```
//填充ICMP数据包
char icmp_data[MAX_PACKET];
icmp_head *icmp_hdr;
int icmpsize;
memset(icmp_data, 0, MAXPACKET); //全部初始化为0
icmpsize = DEF_PACKET + sizeof(icmp_head); //包长度
icmp_hdr = (icmp_head *) icmp_data;
icmp_hdr -> Type = ICMP_ECHO;
icmp_hdr -> Identifier = (unsigned short)GetCurrentThreadId();
icmp_hdr -> HeadChecksum = checksum((unsigned short *)
                                        icmp_data,  icmpsize);
```

3. 例题分析

- **关键问题：填充与发送ICMP包**

```
//初始化目的地址
sockaddr_in  dest;
memset(&dest, 0, sizeof(dest));
dest.sin_family = AF_INET;
dest.sin_addr.s_addr = inet_addr(argv[1]);
```

将点分十进制的IP转换成一个长整数型数

```
typedef struct sockaddr_in {
  short          sin_family;
  u_short        sin_port;
  struct in_addr sin_addr;
  char           sin_zero[8];
} SOCKADDR_IN, *PSOCKADDR_IN, *LPSOCKADDR_IN;
```

3. 例题分析

- 关键问题：填充与发送ICMP包

```
//发送ICMP数据包
sendto(
        sock,                                   //套接字
        icmp_data,                              //缓冲区
        icmpsize,                               //缓冲区长度
        0,                                      //发送标志(默认值)
        (struct sockaddr *) &dest,              //目的地址
        sizeof(dest)                            //地址长度
);
```

3. 例题分析

- **接收与解析ICMP包**
  - **如果目的主机处于活动状态，它会向源主机发送一个ICMP回送应答**

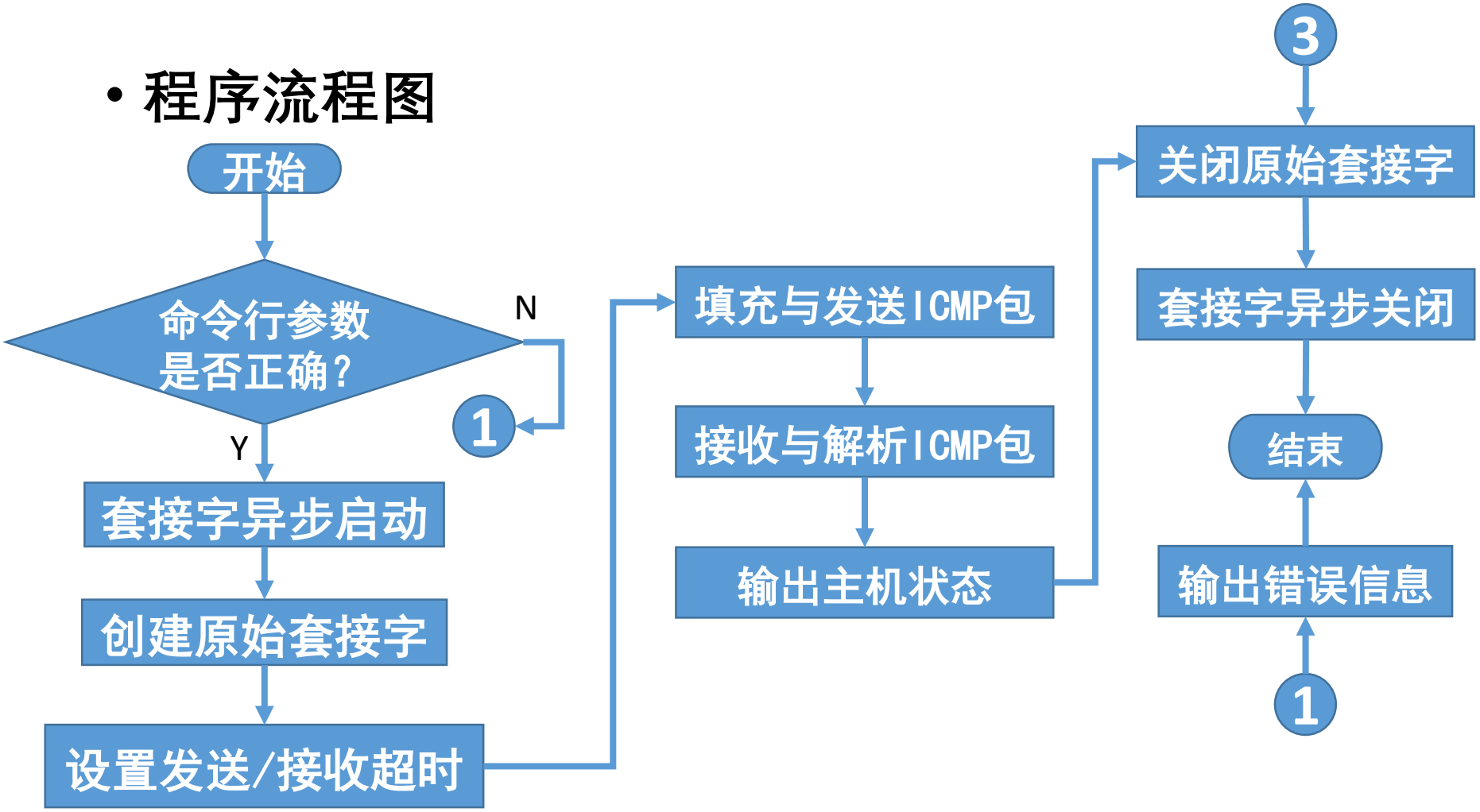
```
//初始化源地址
sockaddr_in from;
int fromlen = sizeof(from);
memset(&from, 0, fromlen);
char * recvbuf = new char[MAX_PACKET+sizeof(ip_head)];
//接收ICMP数据包
int nRecv = recvfrom(sock, recvbuf, MAX_PACKET +
sizeof(ip_head), 0, (struct sockaddr *) &from, &fromlen);
```

3. 例题分析

```
//解析ICMP数据包
ip_head * iphdr = (ip_head *) recvbuf;
icmp_head * icmphdr;
unsigned short ip_size = (iphdr->HeadLen & 0x0f) *4;
icmphdr = (icmp_head *) (recvbuf + ip_size);
//判断ICMP报文类型
if(nRecv < ip_size + ICMP_MIN)//长度非法
    …
if(icmphdr->Type!=ICMP_ECHO_REPLY)//不是回送应答
   …
if(icmphdr->Identifier != (unsigned short)
GetCurrentThreadId()) //标识符不匹配
   …
```

3. 例题分析

- 程序流程图


开始
命令行参数
是否正确?
N
Y
1
套接字异步启动
创建原始套接字
设置发送/接收超时
填充与发送ICMP包
接收与解析ICMP包
输出主机状态
关闭原始套接字
套接字异步关闭
结束
输出错误信息
3
1

3. 例题分析—程序演示


管理员: C:\Windows\system32\cmd.exe
开始主机扫描
主机192.168.0.1:活动状态
E:\cprjs\FrameParse\Debug>Ch-9-ScanHost.exe 192.168.0.2
开始主机扫描
主机192.168.0.2:关闭状态
E:\cprjs\FrameParse\Debug>Ch-9-ScanHost.exe 192.168.0.3
开始主机扫描
主机192.168.0.3:关闭状态
E:\cprjs\FrameParse\Debug>Ch-9-ScanHost.exe 192.168.0.101
开始主机扫描
主机192.168.0.101:活动状态
E:\cprjs\FrameParse\Debug>Ch-9-ScanHost.exe 192.168.1.1
开始主机扫描
主机192.168.1.1:活动状态

本章小结

- 设计目的
  - ICMP可以补充IP协议功能
  - 理解ICMP协议的作用
- 相关知识
  - ICMP基本概念
  - 数据包类型、结构
  - ICMP回送请求与应答
- 例题分析
  - 创建原始套接字
  - ICMP头部数据结构、填充、发送、接收与解析