计算机网络编程

第10章 发现服务器开启的TCP端口

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

- 网络服务常以客户机/服务器模式工作
- 服务器在某些特定端口上提供网络服务，等待客户机发出的服务请求
- 通过发现服务器开启的TCP端口，了解传输层的基本功能与协议类型
- 掌握网络服务、端口的概念与相互关系

2. 相关知识：传输层的基本概念

- 网络层及以下各层实现**网络主机之间的数据通信**
- **数据通信**并**不是**组建计算机网络的**最终目的**
- 网络的本质是实现主机之间的**资源共享**，以实现在应用层提供的各种**网络服务**

2. 相关知识：传输层的基本概念

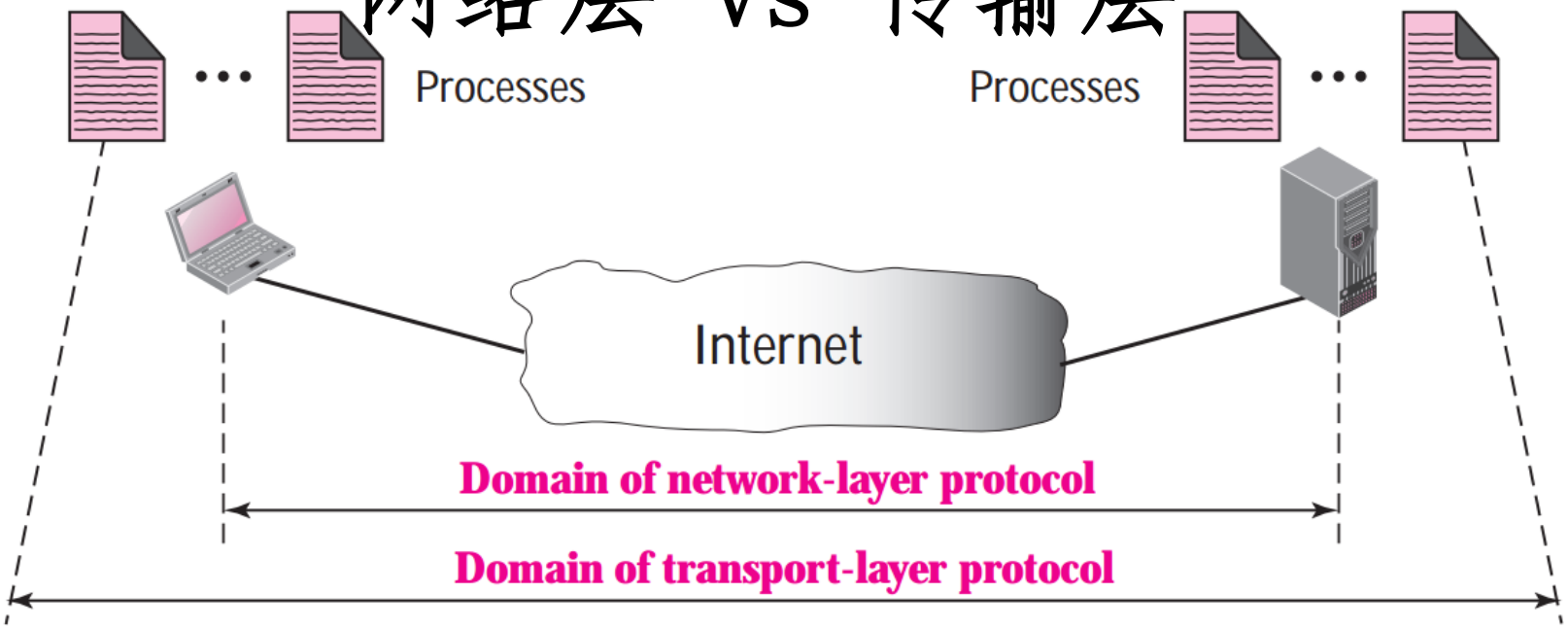
- **传输层主要作用**
  - 实现网络环境中的**分布式进程通信**
  - 为实现应用层的各种网络服务功能**提供传输服务**
  - **承上启下**

2. 相关知识：传输层的基本概念

- 传输层协议
  - 基于网络层协议提供的服务
  - 在源主机和目的主机的应用进程之间
  - 实现“端到端”服务：分布式进程通信

2. 相关知识：传输层的基本概念

网络层 vs 传输层

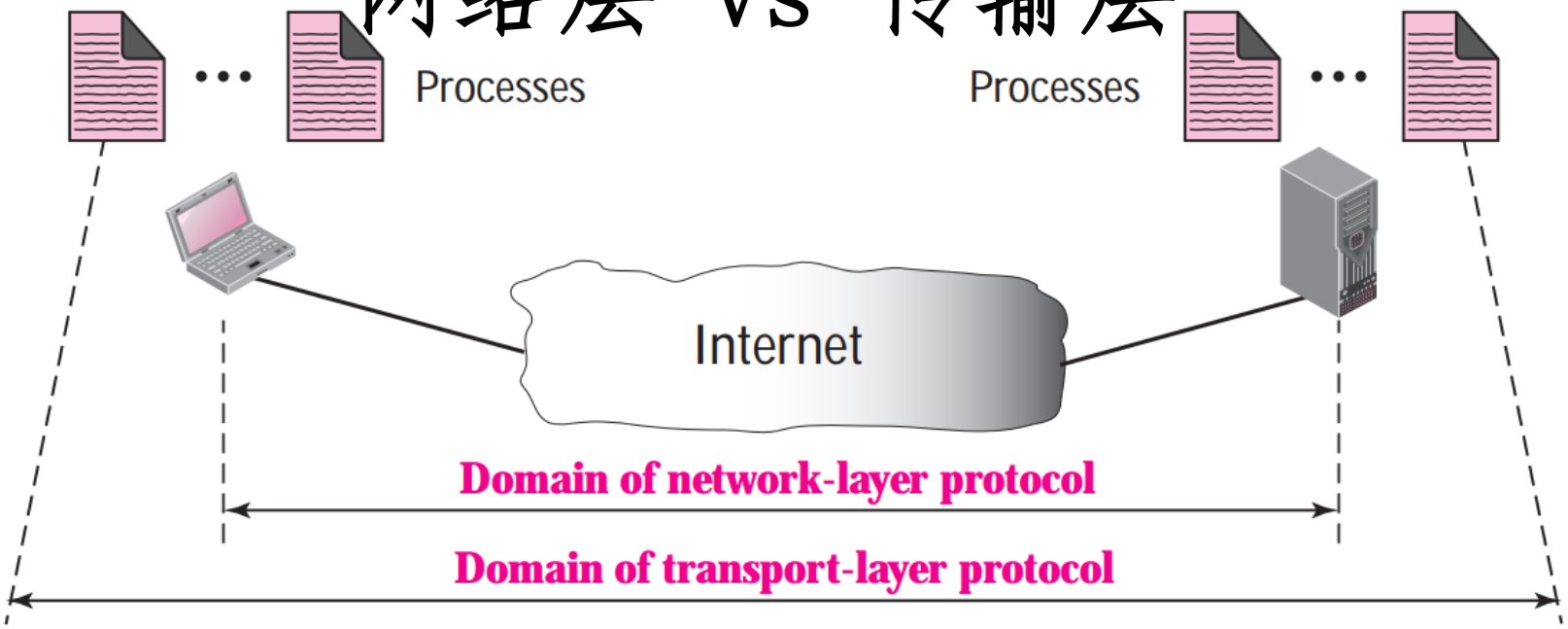


网络层负责计算机级的通信，报文交付给目的计算机

不是完整的交付

2. 相关知识：传输层的基本概念

网络层 vs 传输层

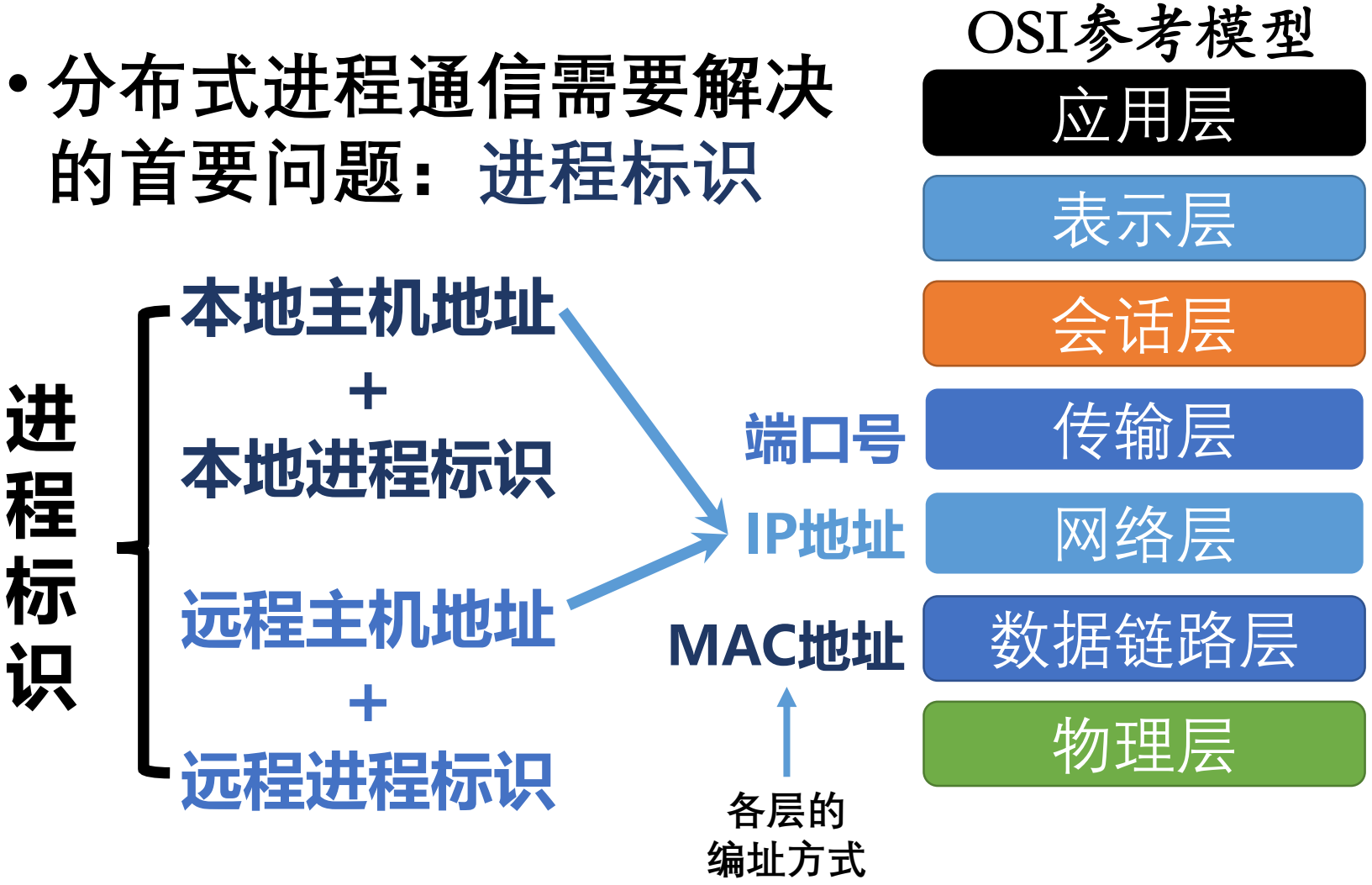


传输层协议负责把报文交付给合适的进程

才算完整的交付!

2. 相关知识：传输层的基本概念

- 分布式进程通信需要解决的首要问题：进程标识



2. 相关知识：传输层的基本概念

- 分布式进程通信需要解决的首要问题：进程标识

进程标识 = 本地主机地址 + 本地进程标识 + 远程主机地址 + 远程进程标识

本地进程标识 → 端口号

本地主机地址 → IP地址

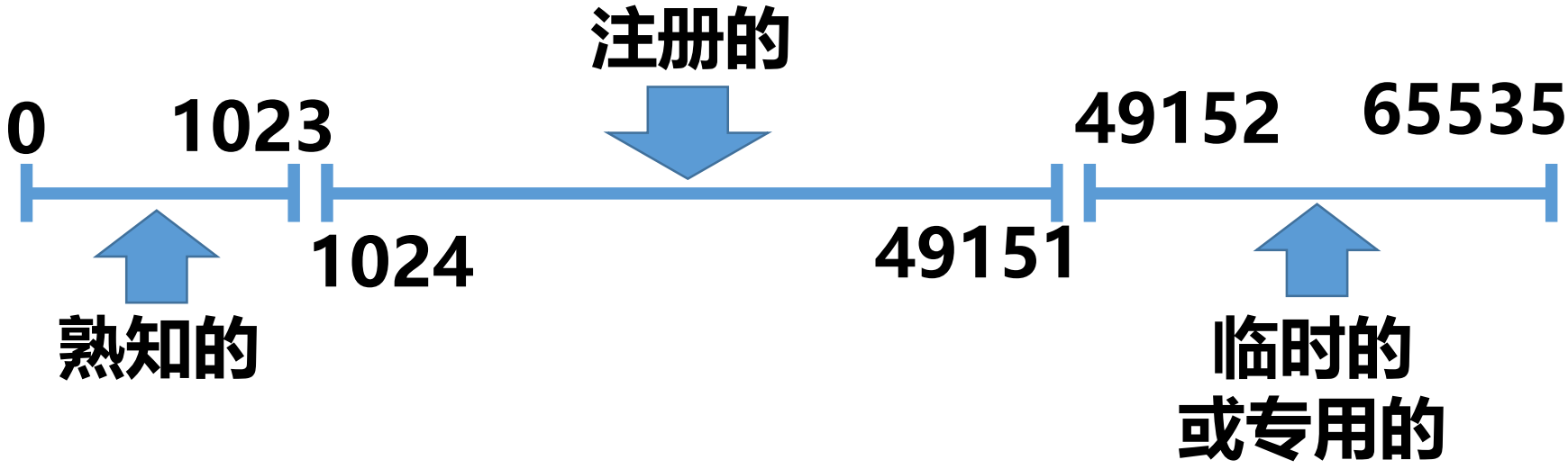
远程主机地址 → MAC地址

远程进程标识 → 端口号

2. 相关知识：端口号的分配

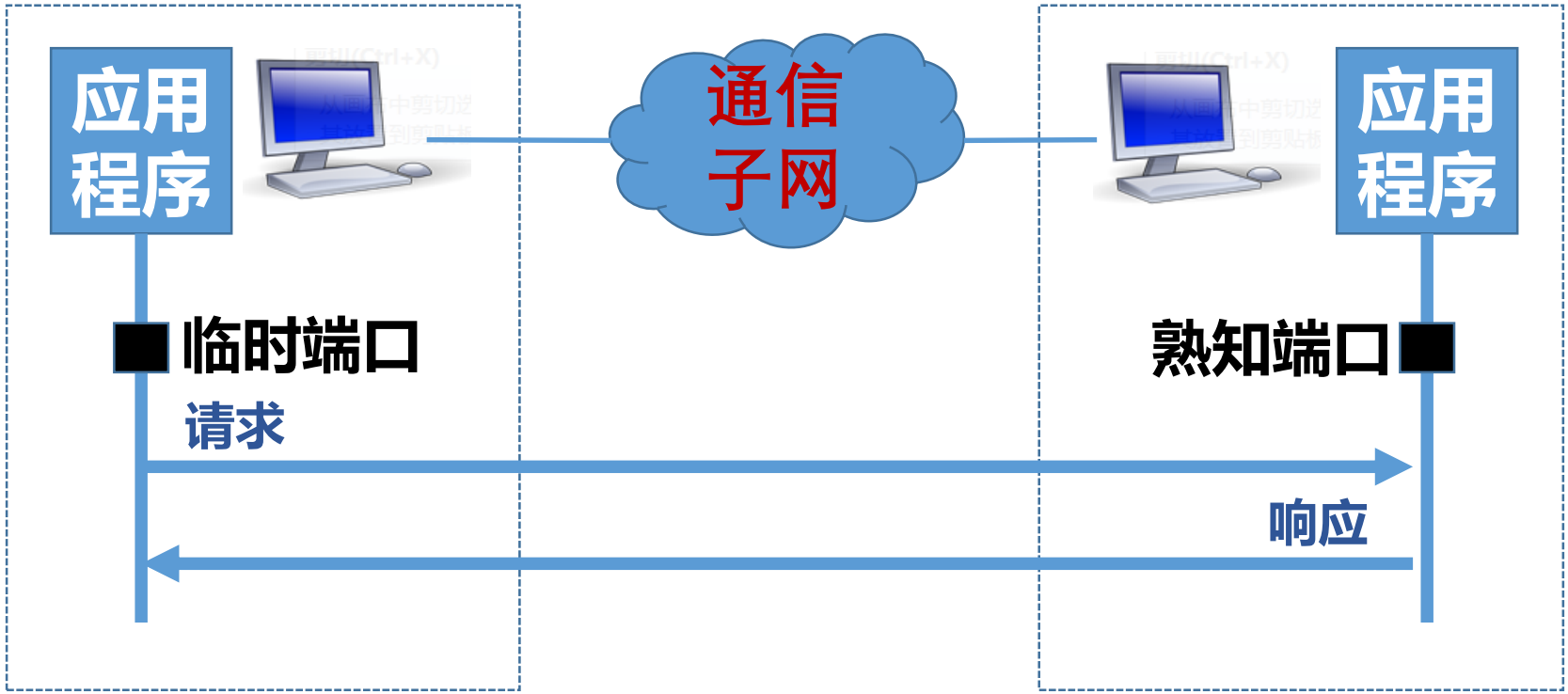
- IP地址定义了某台主机
- 主机选定后，端口号定义了该主机上某个进程

ICANN定义的端口范围段



注：IANA在1998年10月以前曾负责管理因特网域名和地址

2. 相关知识：端口号的分配



客户机通过临时端口向服务器请求服务
服务器通过熟知端口向客户机提供服务

2. 相关知识：端口号的分配

- TCP的主要熟知端口号

端口号	服务进程	说明
20	FTP	文件传输协议(数据连接)
21	FTP	文件传输协议(控制连接：认证)
23	Telnet	虚拟终端网络
25	SMTP	简单邮件传输协议
53	DNS	域名系统
80	HTTP	超文本传输协议
110	POP3	邮局协议第3版
443	HTTPS	安全超文本传输协议

3. 例题分析：设计要求

- 编写程序来扫描服务器已开启的TCP端口，并将获得的相应端口号显示出来
- 本练习只扫描0~127范围内的端口

3. 例题分析：设计要求

- 具体要求
  - 要求程序为命令行程序

    ScanPort　server_addr
  - 要求将部分字段内容显示在控制台上

    已开启的TCP端口：xx

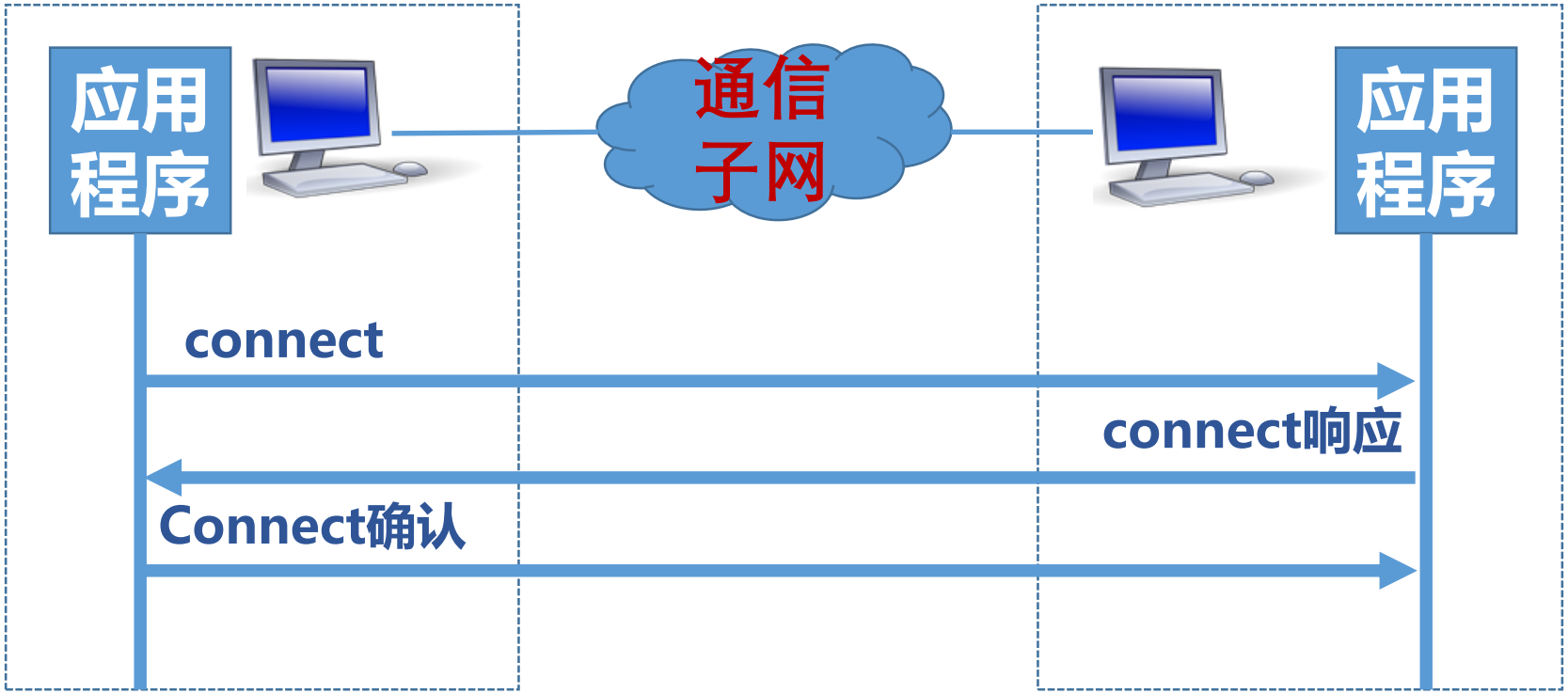
3. 例题分析：关键问题

- 创建套接字

```
//初始化Winsock DLL,成功则返回0
iResult = WSAStartup(MAKEWORD(2,2), &WSAData));
//创建流式套接字
SOCKET sock = socket(AF_INET, SOCK_STREAM, 0);
//填充套接字地址
sockaddr_in serveraddr;
serveraddr.sin_family = AF_INET;
serveraddr.sin_port = 端口号;
Serveraddr.sin_addr.S_un.S_addr = IP地址;
```

3. 例题分析：关键问题

- TCP端口扫描
  - Connect扫描，利用套接字的connect()函数进行，扫描每个端口都需完成三次握手，又称全连接扫描



3. 例题分析：关键问题

- SYN(Synchronize Sequence Numbers)扫描
  - 利用包含SYN标志的TCP包进行，扫描每个端口仅需一次握手
  - 若服务器没有开启端口，则返回RST包关闭连接，又称半连接扫描
  - SYN报文段不携带数据，但会消耗一个序号
  - RST表示复位、用来异常的关闭连接

3. 例题分析：关键问题

- FIN扫描
  - 利用包含FIN标志（发送方字节流结束，用于关闭连接）的TCP包进行扫描
  - 若服务器开启端口，则会丢弃该TCP包
  - 若服务器没有开启端口则返回RST包
  - 不需要建立连接

3. 例题分析：关键问题

- TCP端口扫描
  - 本课题采用Connect扫描
  - 优点
    - 正常建立TCP连接，在编程上可调用connect()函数轻松完成
    - 可采用多线程并发执行提供多端口扫描效率
  - SYN和FIN扫描执行速度快，但实现复杂，不确定

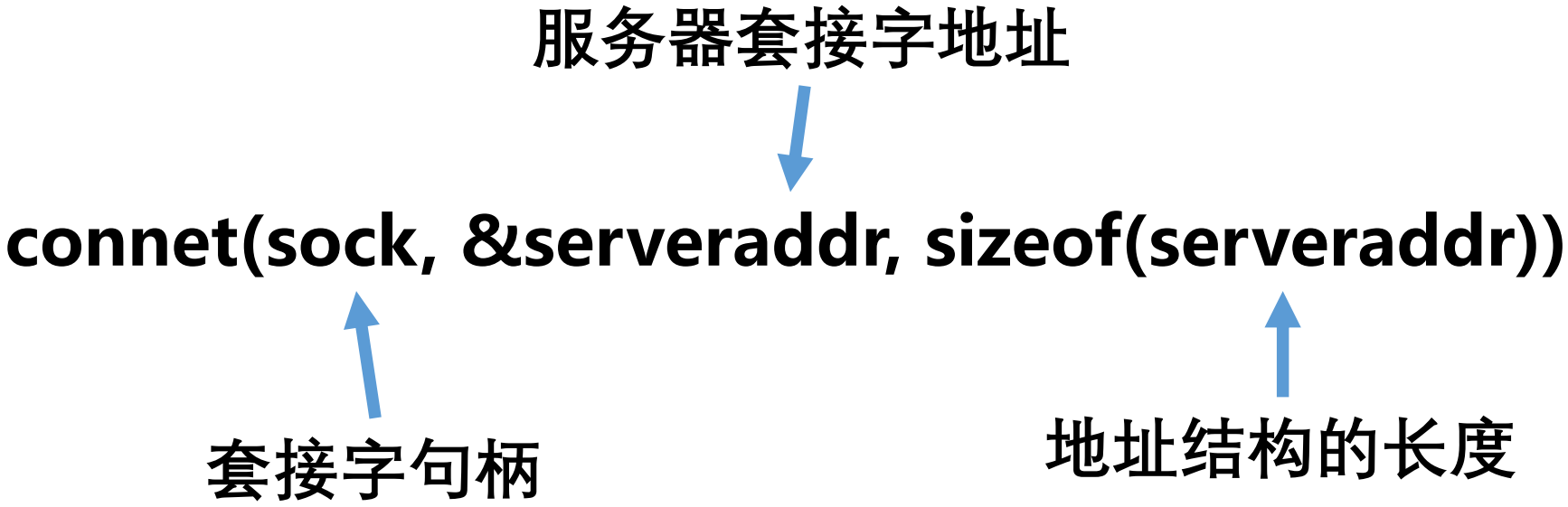
3. 例题分析：关键问题

- TCP端口扫描

```
//设置超时时间
struct timeval timeout;
timeout.tv_sec = 100/1000;     //秒
timeout.tv_usec = 0;           //微秒1 μs= 1.0E-6 sec
//与端口建立连接
connet(sock, &serveraddr, sizeof(serveraddr));
//判断连接是否超时
if(select(0, NULL, &write, NULL, &timeout)>0)
    …
```

3. 例题分析：关键问题

- TCP端口扫描



触发协议栈向目标地址发送SYN请求，完成TCP的三次握手
connect()函数成功返回(0)表示已确认服务器是存在的

3. 例题分析：关键问题

- TCP端口扫描

select()函数可用于检查多个套接字状态

3. 例题分析：关键问题

- TCP端口扫描

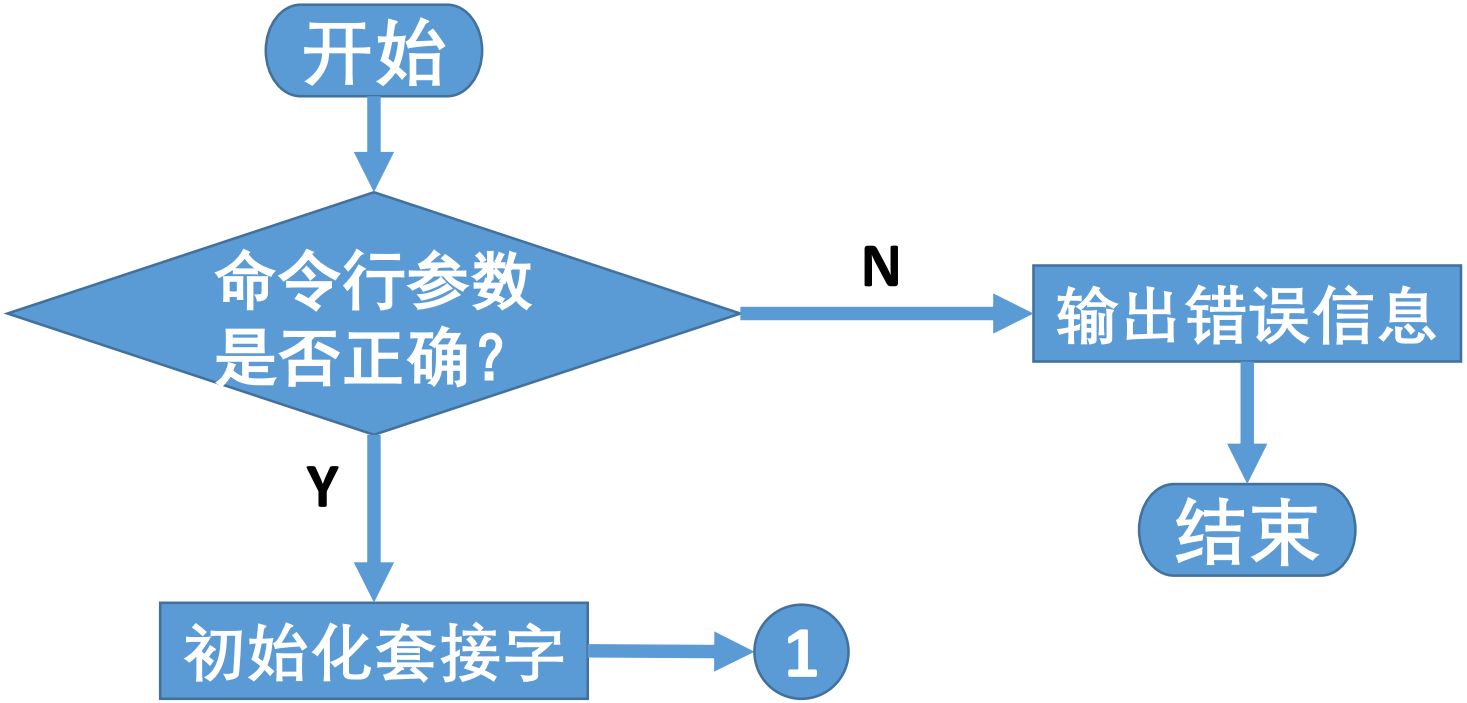
select(0, NULL, &write, NULL, &timeout)

select()函数返回就绪的套接字数量
如果返回0，则表示已经超时了

更多细节参见：https://docs.microsoft.com/en-us/windows/desktop/api/winsock2/nf-winsock2-select

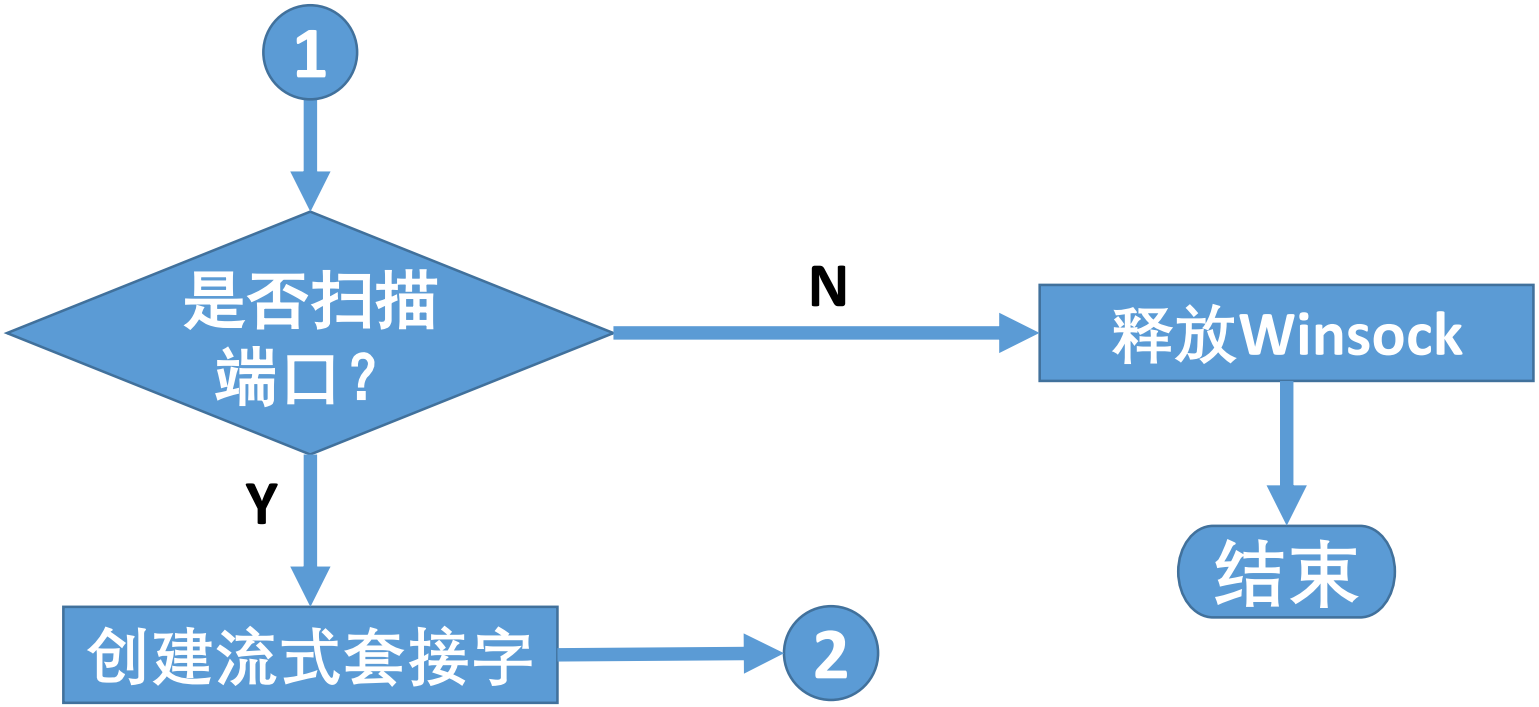
3. 例题分析：关键问题

- 程序流程图



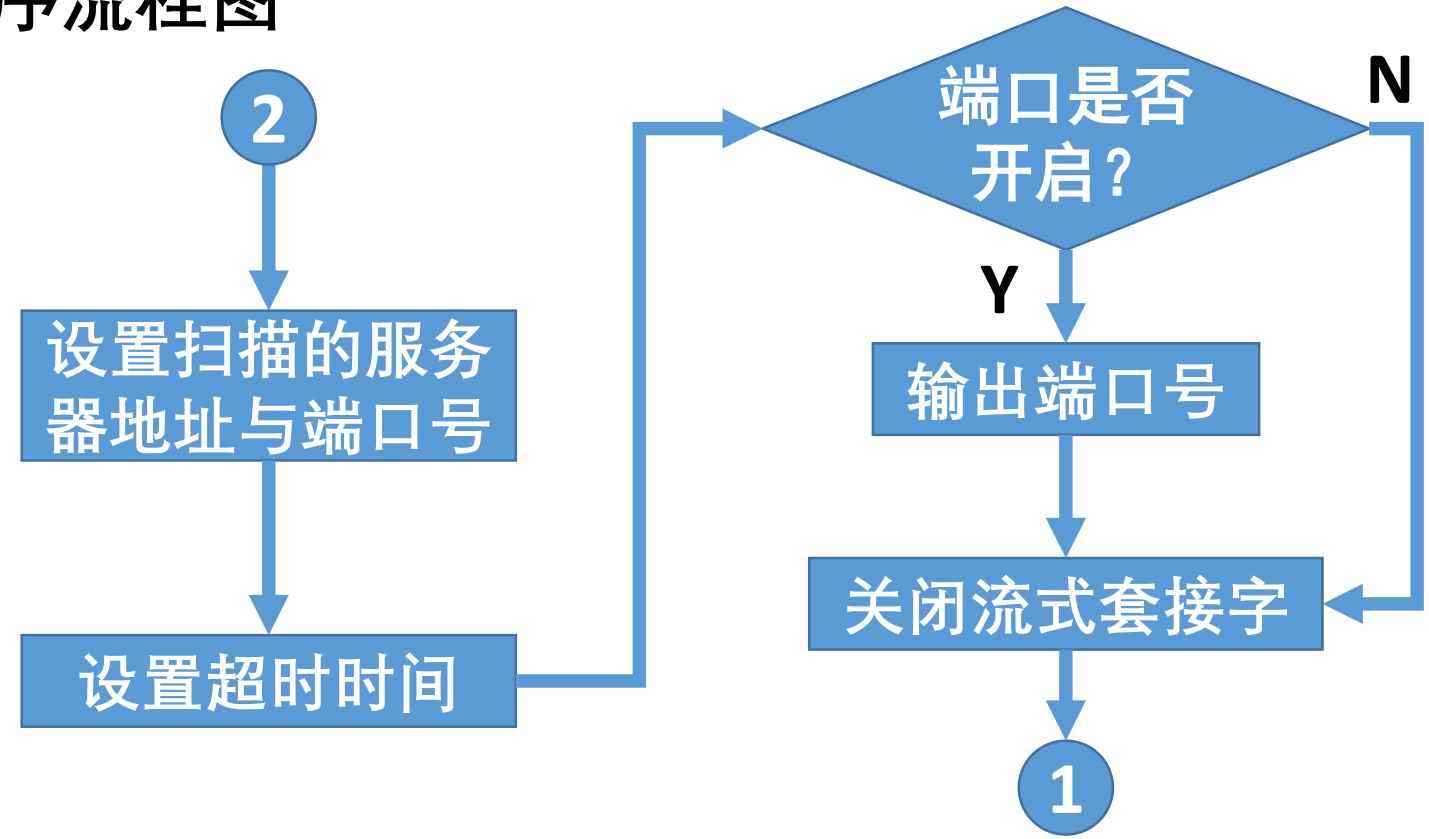
3. 例题分析：关键问题

- 程序流程图

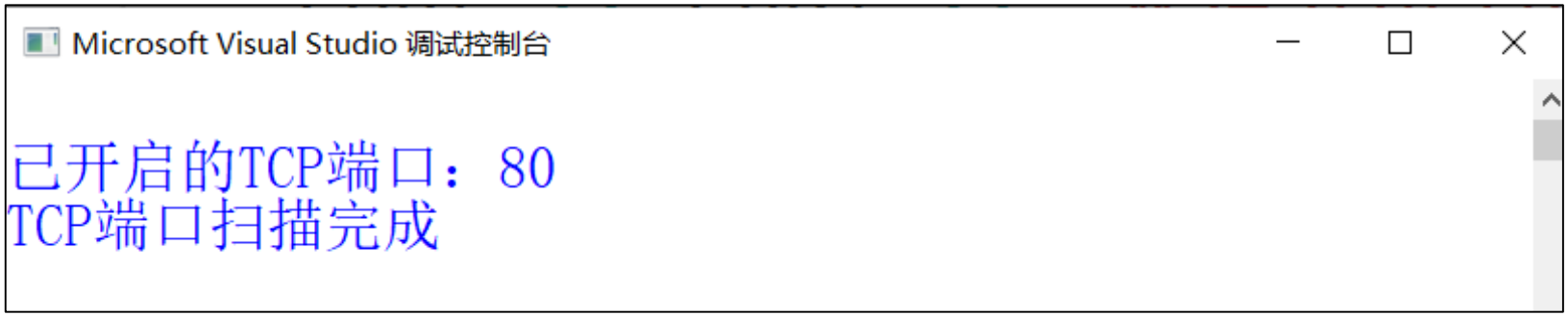


3. 例题分析：关键问题

- 程序流程图



3. 例题分析：程序演示



本章小结

- 设计目的
  - 了解传输层的基本功能、协议类型
  - 掌握网络服务、端口的概念
- 相关知识
  - 传输层概念、端口号
- 例题分析
  - 创建套接字、Connect扫描