

Pr. 7 network

2017.5

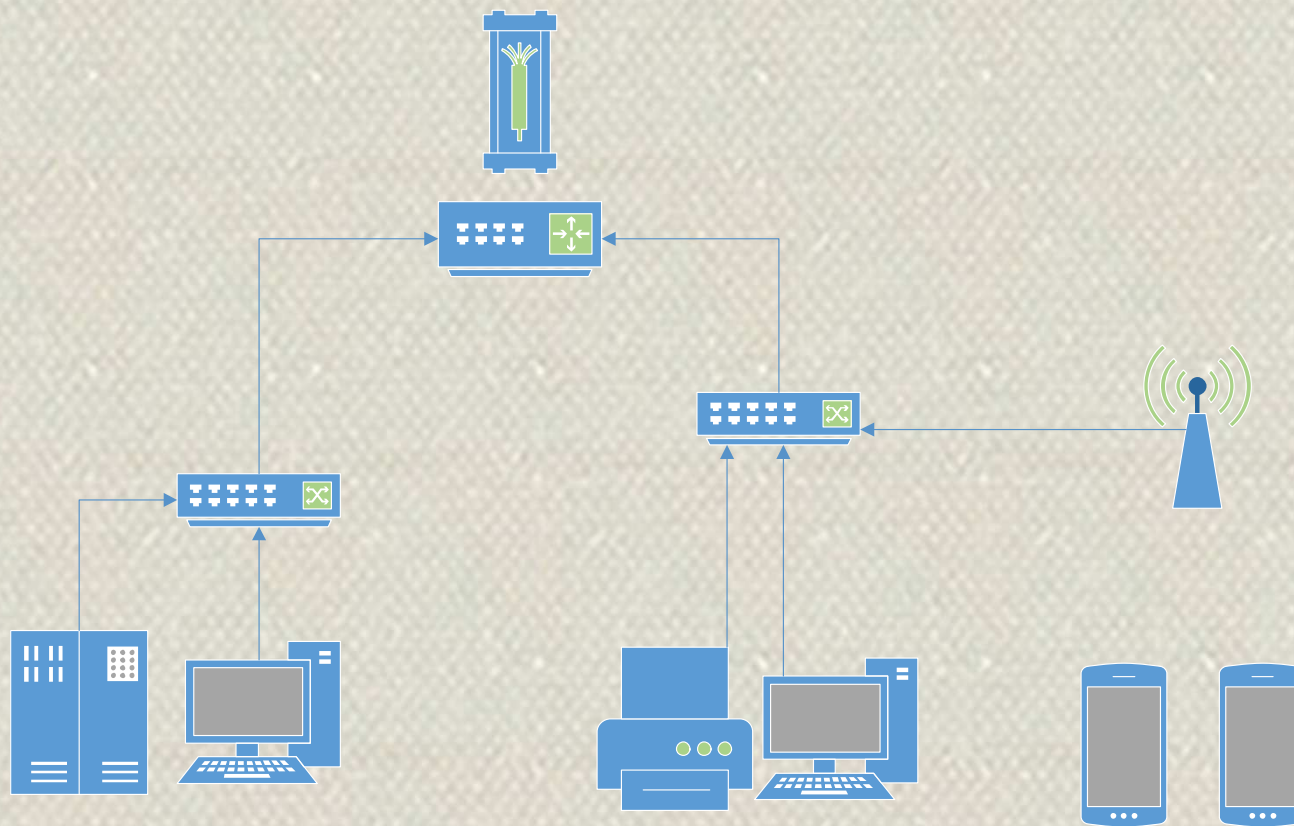
Synonym

- Net
- Network
- web

联网和通信

- 通路—要通信就得连起来
- 寻址—连完了，向哪里发送？
- 协议—找到了能发成功

典型的网络结构



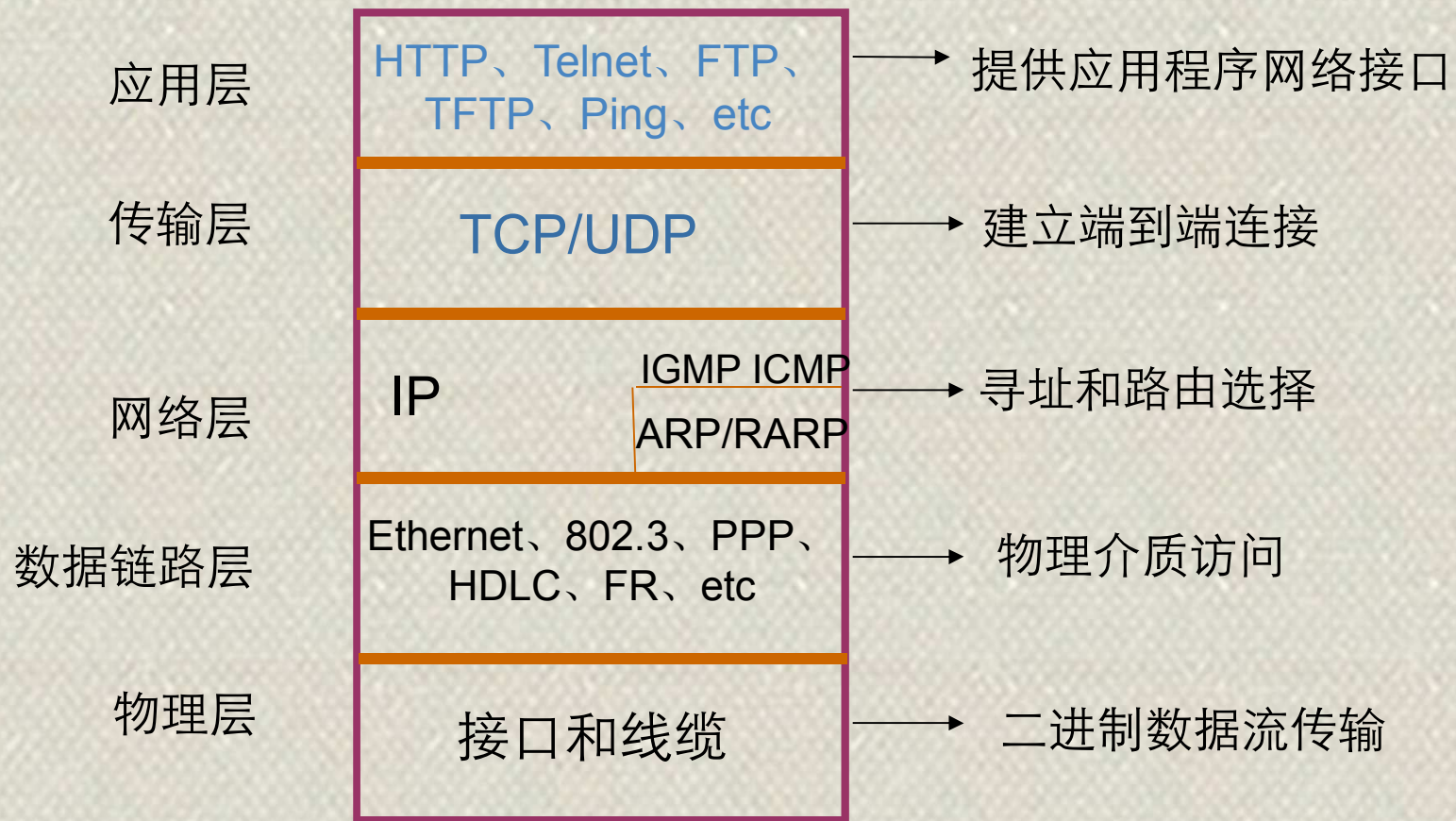
网络设备

- 路游器
- 交换机
- 网卡（无线网卡）

协议

- 大家都遵守的一种约定
- 权威机构制定的公共标准

TCP/IP协议栈



简单的网络命令

- ipconfig/ifconfig
- ping
- telnet

MAC地址

- 就是网卡的编号
- 6个字节48位的数字
- 前24位叫做组织唯一标志符（Organizationally Unique Identifier/OUI），IEEE分配给不同厂家的ID
- 后24位是由厂家自己分配的
- 每块网卡的编号都不同？

IP地址

就是一个整数

4个字节32位的整数

[0x00000000, 0xFFFFFFFF] 即 [0, 4294967295]

某台主机

- **3657948666**

→ **1101101000000111101110111111010**

是不是好记一点？

- 1101101000000111101110111111010
- a.b.c.d
- 218.7.221.250

这是哪儿的IP地址？

- 218.7.221.250

ip138.com IP查询(搜索IP地址的地理位置)

您查询的IP:218.7.221.250

- 本站数据：黑龙江省哈尔滨市 联通
- 参考数据1：黑龙江哈尔滨 联通
- 参考数据2：黑龙江省哈尔滨市 联通

▶ 数字/IP地址转换结果:

218.7.221.250 → 3657948666 → 黑龙江省哈尔滨市 哈尔滨师范大学松北校区

更好记的办法

- 不好记就取个名字
- “**www.hrbnu.edu.cn**” = **218.7.221.250**

IP地址

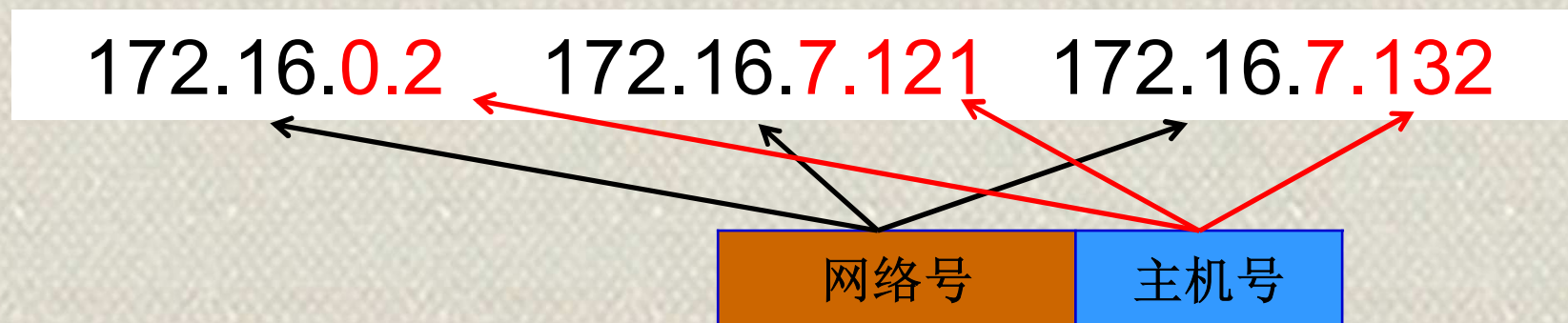
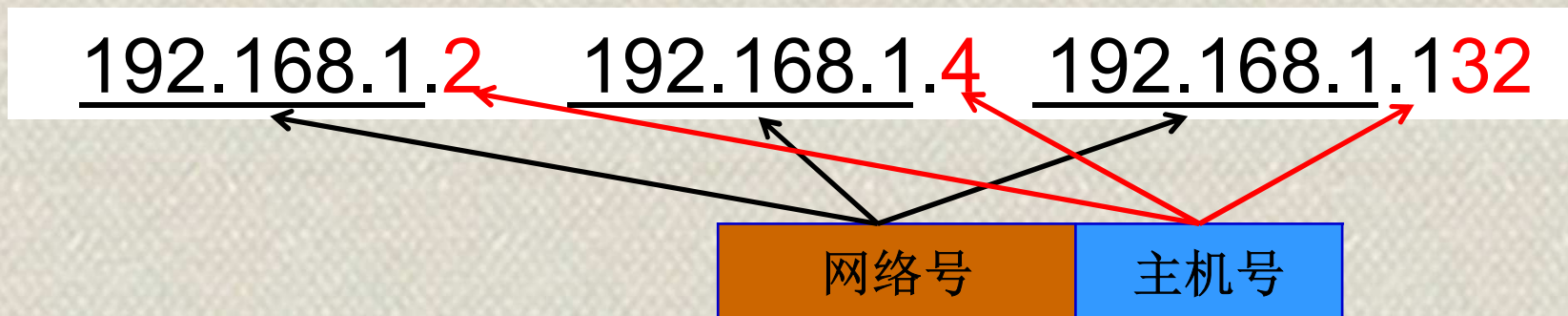
- 类比电话号码

0451-88064175 0451-88064165

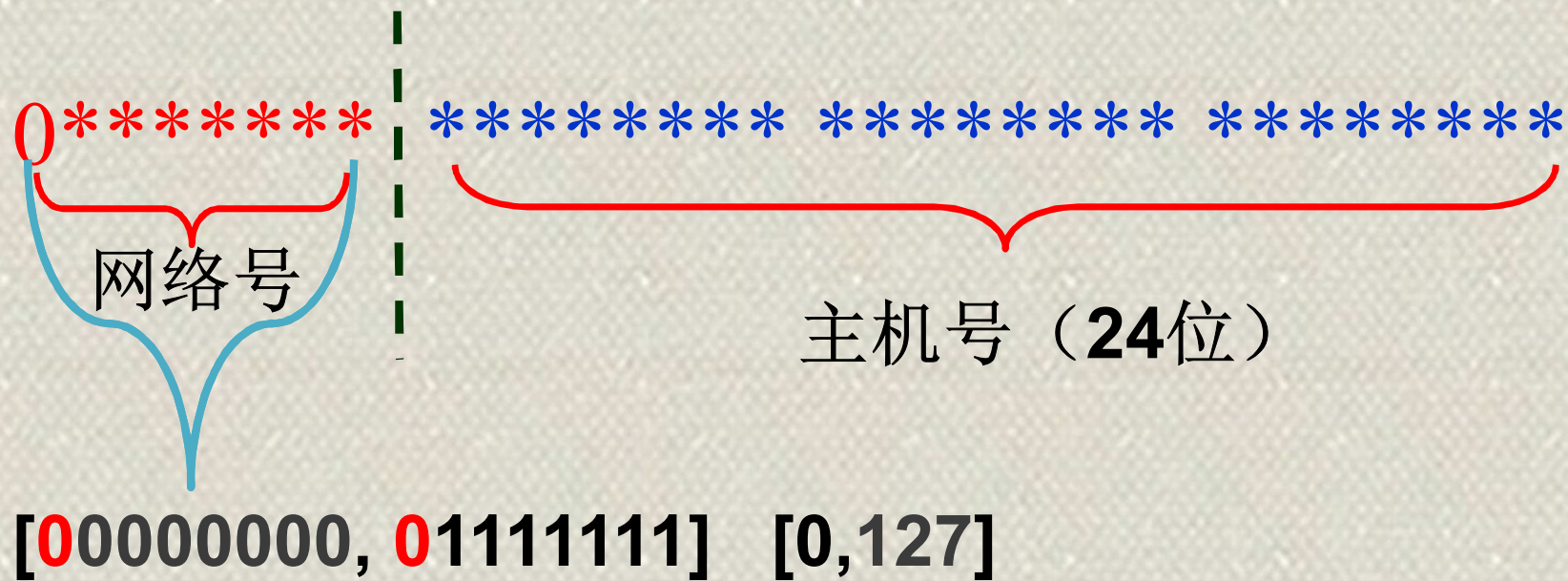
192.168.1.312 192.168.1/24

10.64.2.57

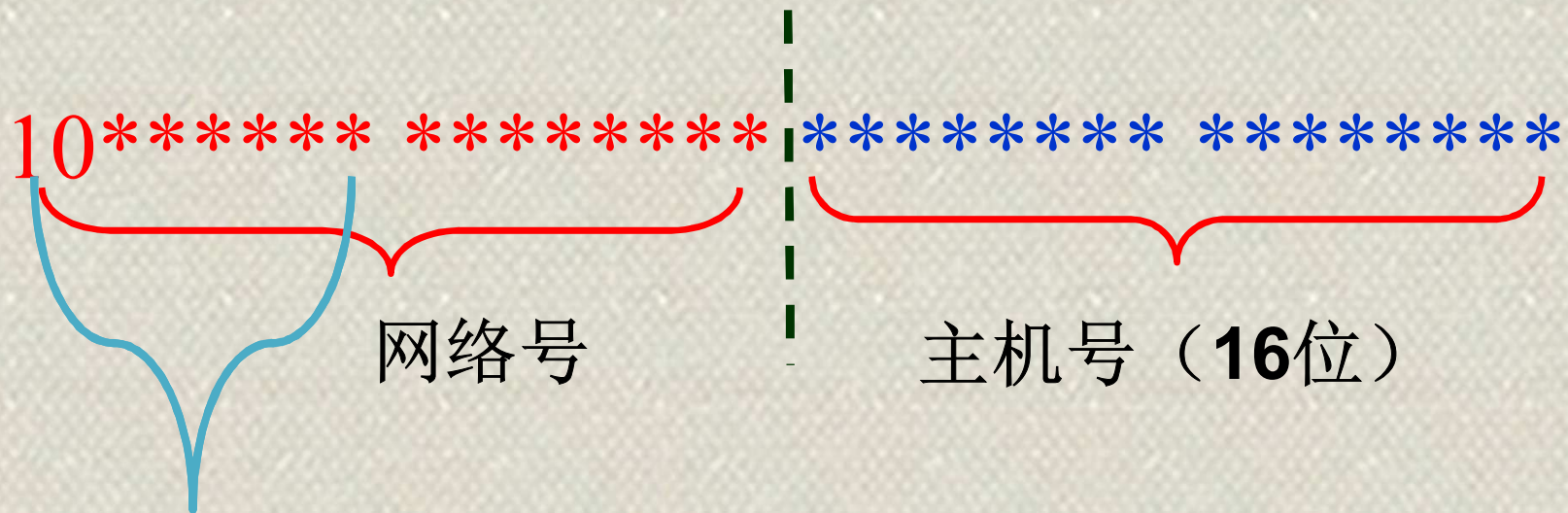
218.7.221.250



A类地址



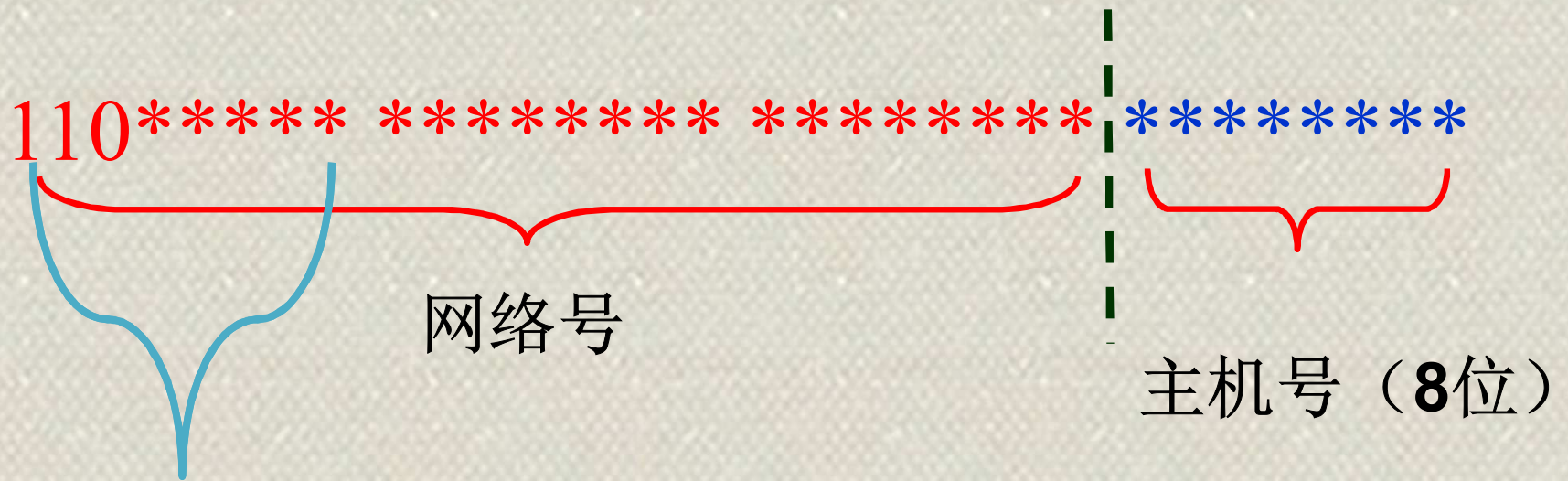
B类地址



[**10**000000, **10**111111] [128,191]

128.0.0.0到191.255.255.254

C类地址



[**110**00000, **110**11111] [192, 223]

192.0.0.0 -- 223.255.255.254

D,E类地址

1110***** *****

组播地址

11110***** *****

保留地址

注意

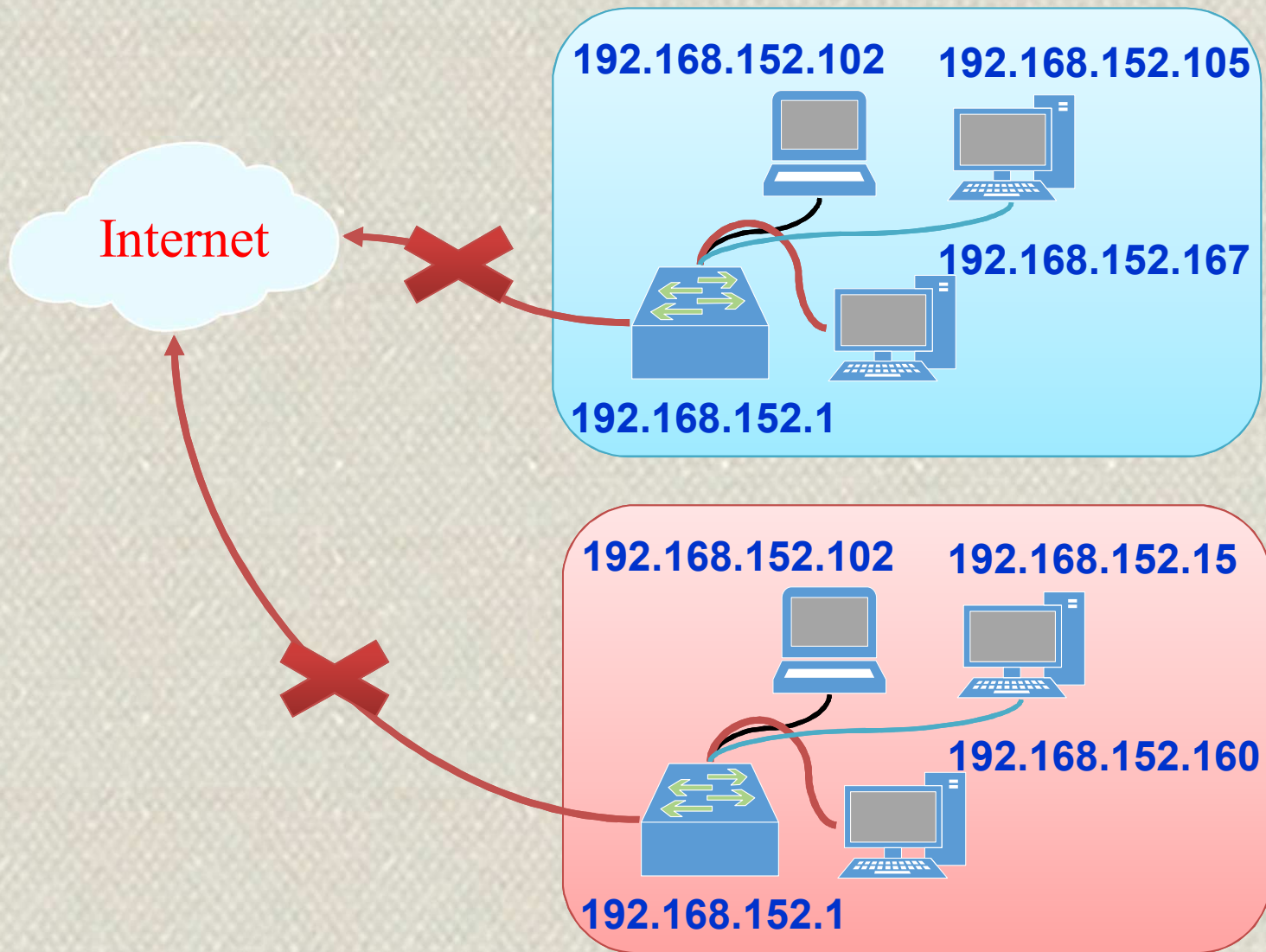
- 主机号为全**0**，表示本机所在网络的网络号，不能用来表示单个主机
- 主机号为全**1**（**255**），代表广播地址，不能用来表示单个主机地址
- A类地址中的127段是地址，如127. 1. 11. 13，用于网络软件测试以及本地机进程间通信
- 127. 0. 0. 1

问题

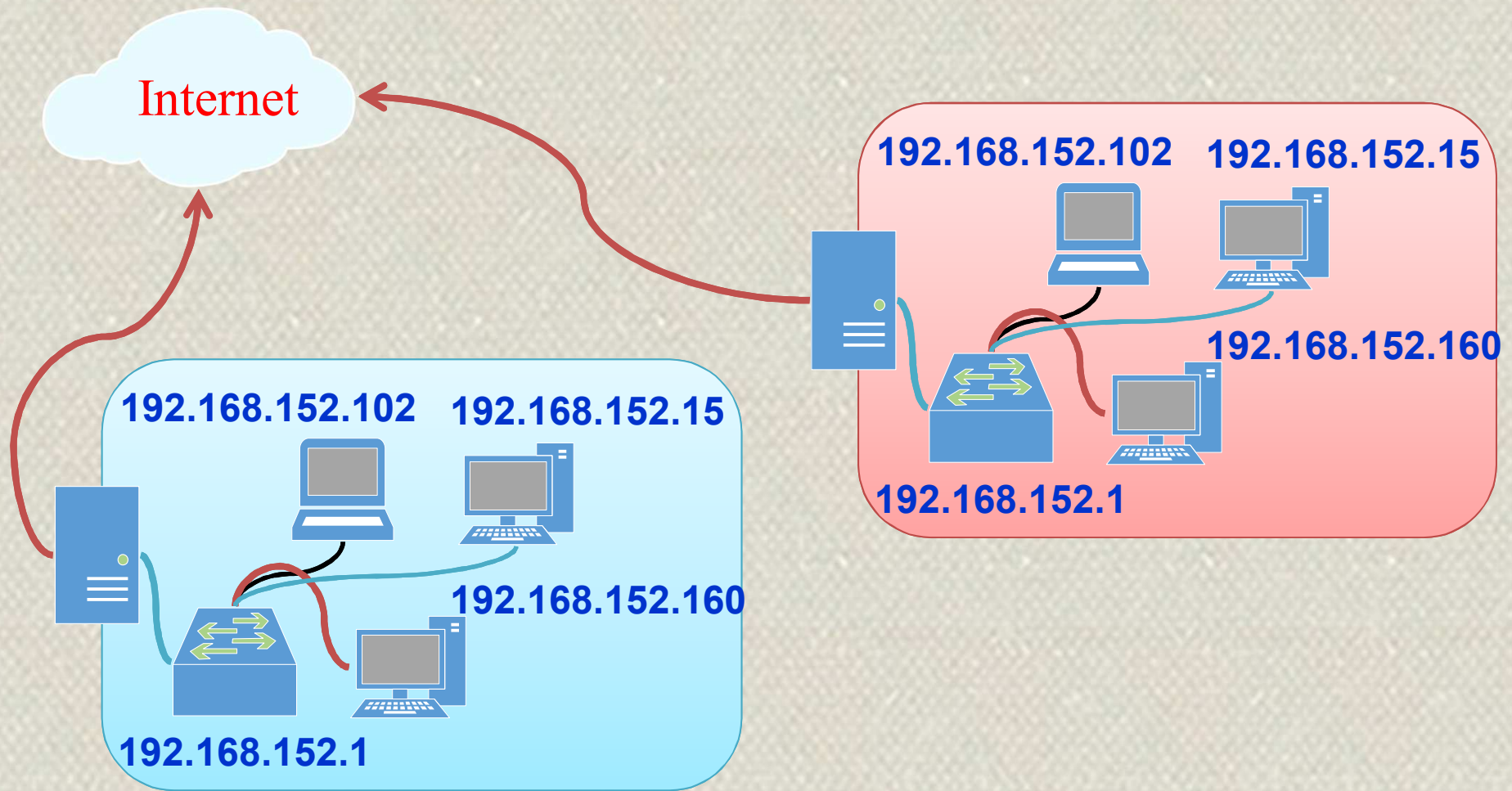
- IP地址[0, 4294967295]
- 保留地址
- 耗尽了怎么办？

什么是私有地址？

地址类型	私有地址范围	网络个数
A类	10.0.0.0 ~ 10.255.255.255	1
B类	172.16.0.0 ~ 172.31.255.255	16
C类	192.168.0.0 ~ 192.168.255.255	256



NAT



IPV6

- 数值更大 → 16字节128位

端口

- 是一个2字节16位的无符号整数
- [0, 65535]

Port







特殊端口

- **80 http** 用于万维网（**WWW**）服务的超文本传输协议（**HTTP**）
- **138 netbios-dgm** 在红帽企业 **Linux** 中被 **Samba** 使用的 **NETBIOS** 数据报服务
- **139 netbios-ssn** 在红帽企业 **Linux** 中被 **Samba** 使用的 **NETBIOS** 会话服务
- **443 https** 安全超文本传输协议（**HTTP**）
- **445 microsoft-ds** 通过 **TCP/IP** 的服务器消息块（**SMB**）

TCP/UDP

- 面向连接→打电话
- 不面向连接→发短信

服务器

- 网络环境初始化
- 创建一个socket(本机的IP, port)
- 将服务与socket绑定bind
- 开始监听谁来连接listen
- 不停地接收连接请求accept（也是一个socket）
- Socket间开始收发消息

服务器主体代码

```
35 SOCKET sockSrv = socket(AF_INET, SOCK_STREAM, 0);
36 SOCKADDR_IN addrSrv;
37 addrSrv.sin_addr.S_un.S_addr = htonl(INADDR_ANY);
38 addrSrv.sin_family = AF_INET;
39 addrSrv.sin_port = htons(6000);
40 //bind the socket to the server
41 bind(sockSrv, (SOCKADDR*)&addrSrv, sizeof(SOCKADDR));
42 listen(sockSrv, 5);
43 SOCKADDR_IN addrClient;
44 int len = sizeof(SOCKADDR);
45 while (1)
46 {
47     //wait for a certain client visiting
48     SOCKET sockConn = accept(sockSrv, (SOCKADDR*)&addrClient, &len);
49     char sendBuf[100];
50
51     char client_ip[32];
52     memset(client_ip, 0, 32);
53
54     inet_ntop(AF_INET, &addrClient.sin_addr, client_ip, 16);
55     sprintf_s(sendBuf, "Welcome%s here\n", client_ip);
56
57     //send data to client
58     send(sockConn, sendBuf, strlen(sendBuf) + 1, 0);
```

客户端

- 网络环境初始化
- 创建一个用于连接的socket(服务器的IP, port)
- 建立连接connect(三次握手)
- 连接成功后开始收发消息

客户端主体代码

```
25 SOCKET sockClient = socket(AF_INET, SOCK_STREAM, 0);
26 SOCKADDR_IN addrSrv;
27 inet_pton(AF_INET, "127.0.0.1", (void*)&addrSrv.sin_addr.s_addr);
28 addrSrv.sin_family = AF_INET;
29 addrSrv.sin_port = htons(6000);
30 int status = connect(sockClient, (SOCKADDR*)&addrSrv, sizeof(SOCKADDR));
31 if (status != 0)
32 {
33     printf("the socket cannot connect\n");
34     return 0;
35 }
36
37 char sendBuf[128];
38 char recvBuf[256];
39
40 std::cin >> sendBuf;
41
42 recv(sockClient, recvBuf, 256, 0);
43 printf("%s\n", recvBuf);
44
45 send(sockClient, sendBuf, strlen(sendBuf) + 1, 0);
46
47 closesocket(sockClient);
48 WSACleanup();
49 return 1;
```