

# 实验一 简单局域网组网

## 一、实验目的

- 1) 掌握利用双绞线制作网线的基本方法，学会使用以太网交换机构建局域网（LAN）。
- 2) 掌握配置主机网络连接属性的方法和步骤。
- 3) 学会使用网络命令测试局域网主机的连通性。

## 二、实验环境

- 1) 运行 Windows 2008 Server/XP/7 操作系统、具有以太网卡的 PC 机。
- 2) 实验机房内应配置机柜，包括一台以上以太网交换机和配线架。
- 3) 网线钳 2 人一把，测通仪 2 人一套，RJ-45 水晶头、网线若干。



图 1-1 实验器材

## 三、实验内容

- 1) 制作的网线能够通过测通仪的测试。
- 2) 每台 PC 机都连接到交换机指定接口，如图 1-2 所示。
- 3) 在局域网内的任意两台 PC 机都能互相通信。

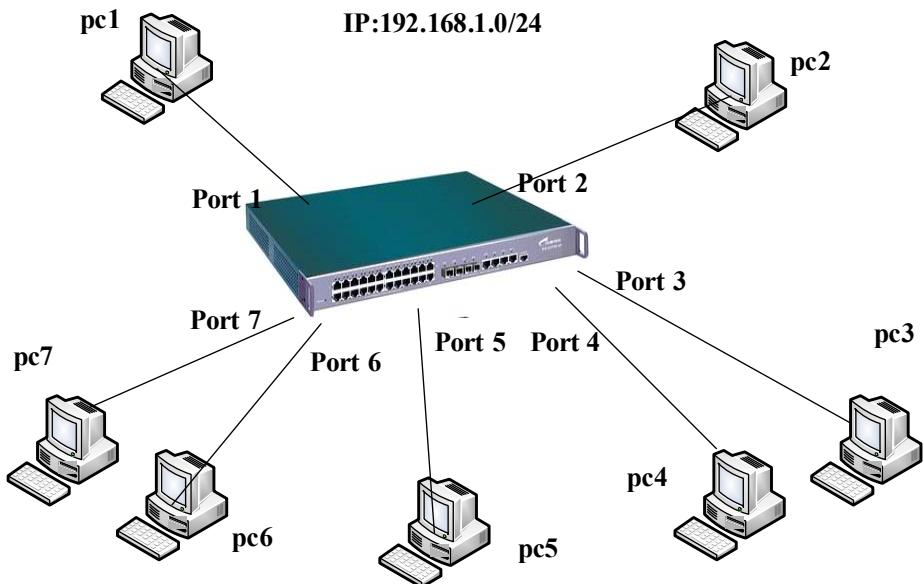


图 1-2 局域网组网实验拓扑

## 四、实验步骤

### 1. 制作 RJ-45 双绞线

**步骤 1:** 选取长度合适的双绞线，然后用网线钳前部剥线器剥除双绞线外皮 2~3cm，如图 1-3 所示。



图 1-3 剥除双绞线外皮

**步骤 2:** 将对线自左向右按橙、蓝、绿、棕的顺序排列，如图 1-4 所示。

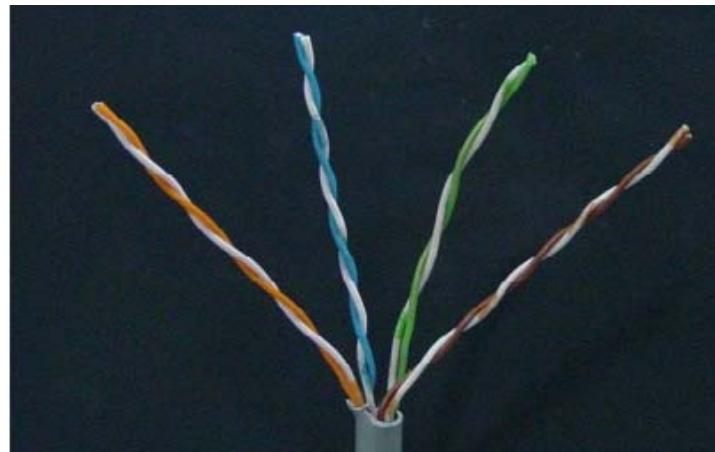


图 1-4 双绞线对的排列方式

**步骤 3：**分离每一对线，将其弄直，并且按照白橙、橙、白绿、蓝、白蓝、绿、白棕、棕的顺序排列，如图 1-5 所示。

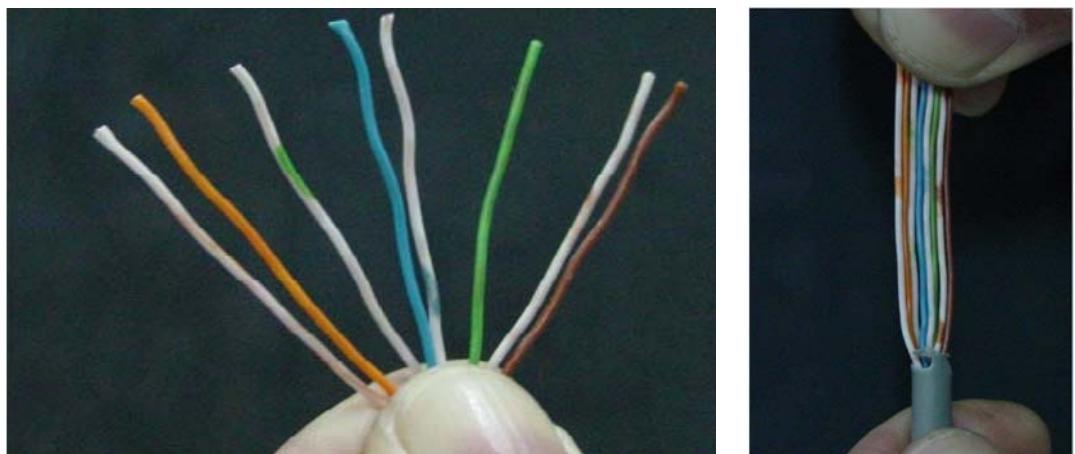


图 1-5 双绞线对拆分后的排列方式

**步骤 4：**将上述网线用网线钳剪齐，长度约为 14mm（注意：不宜过长或过短），再将双绞线的每一根线依序放入 RJ-45 接头的引脚内，第一只引脚内放入白橙线，如图 1-6 所示。

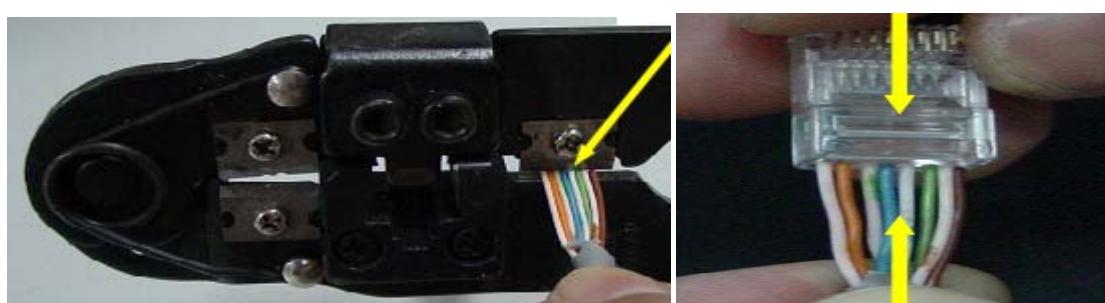


图 1-6 将双绞线剪齐后插入水晶头引脚

**步骤 5：**从水晶头正面目视每根双绞线已经放置正确并到达底部位置之后，将水晶头放入网线钳的压头槽，用力按压接头，使水晶头内部的金属片恰好刺破双绞线的外层表皮与内部金属线良好接触（通常会听到清脆的“咔”声），如图 1-7 所示。

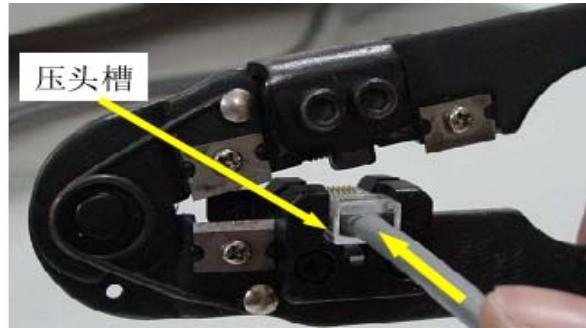


图 1-7 按压水晶头使其与双绞线咬合

重复步骤 1 到步骤 5，制作网线另一端的 RJ-45 接头，完成后的连接线两端的 RJ-45 接头，引脚和颜色完全一致，如图 1-8 所示。

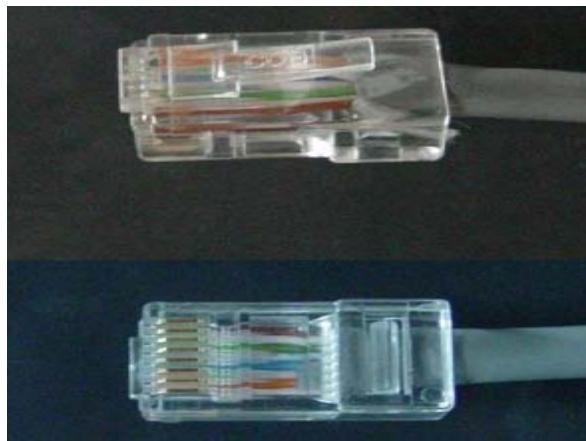


图 1-8 做好的网线两端的 RJ-45 接头

## 2. 用测通仪测试双绞线

用网线测试仪检测上述过程制作的双绞线是否可用。方法是将刚刚制作的网线两端分别插入测试仪主端和从端的接口，打开电源，两端的测试仪上的 LED 依次同时发光，说明线路正常，制作的双绞线可用。



图 1-9 用网线测试仪检测网线

## 3. 组建并配置局域网

用制作好的 RJ-45 双绞线将各自的 PC 机连接到机柜中以太网交换机对应的接口上(接

口顺序由教员或组长指定)。交换机不需要人工配置即可实现组网，当端口收发分组时，对应的指示灯会闪烁。(有的交换机端口会有两个指示灯，另一个表示线路连通。)

接下来，开始配置主机的 IP 地址。首先在“网上邻居”图标上单击右键，选择“属性”一栏，打开“网络连接属性”窗口，如图 1-10 所示。选择物理网卡对应的图标(有的 PC 机上可能存在多个物理或虚拟网卡)，单击右键，选择“属性”一栏，打开“本地连接属性对话框”。



图 1-10 打开“网络连接属性”窗口

在“本地连接属性”对话框中选择“Internet 协议 (TCP/IP)”一栏，单击右下方的“属性”按钮，打开“Internet 协议 (TCP/IP) 属性”配置窗口，如图 1-11 所示。

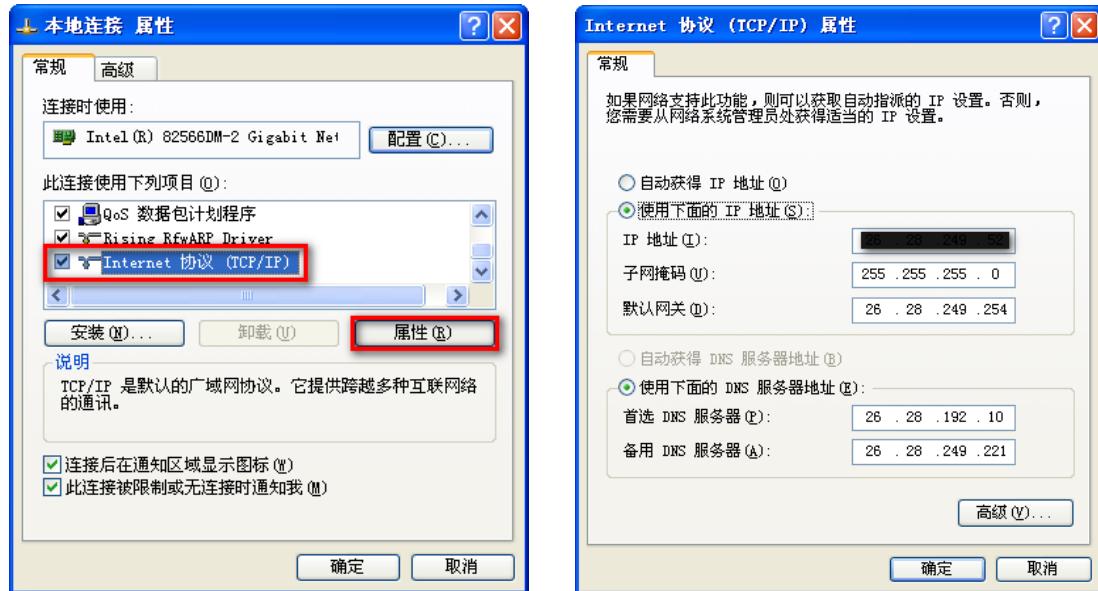


图 1-11 TCP/IP 属性配置窗口

在“TCP/IP 属性”配置窗口中有两种配置 IP 地址的方法：自动获取或手动配置。其中，自动获取 IP 地址需要局域网中有 DHCP 服务器，相关内容在教材 6.6 节中有介绍。本实验中选择手动配置，如图 1-11 所示。配置的要求是：每台主机有唯一的 IP 地址和相同的子网掩码。关于 IP 地址的结构和子网掩码的概念，详见教材 4.2 和 4.3 节。

## 4. 测试局域网连通性

测试两台 PC 机之间是否连通有多种方法，最常用的方法是使用“ping”网络命令。为此，需要启动命令行窗口，方法是单击“开始”菜单的“运行”栏，在“打开”下拉框里输入“cmd”（command 的缩写），并按“回车”键，如图 1-12 所示。



图 1-12 启动命令行窗口的方法

在启动的命令行窗口中输入 ping 命令，语法是“ping+空格+对方主机 IP 地址”，然后按“回车”键。

```
C:\>ping 26.28.249.254

Pinging 26.28.249.254 with 32 bytes of data:

Reply from 26.28.249.254: bytes=32 time<1ms TTL=255
Reply from 26.28.249.254: bytes=32 time<1ms TTL=255
Reply from 26.28.249.254: bytes=32 time=1ms TTL=255
Reply from 26.28.249.254: bytes=32 time<1ms TTL=255

Ping statistics for 26.28.249.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

图 1-13 Ping 命令的执行结果

如果 ping 命令的执行结果如图 13 所示，这表明你的主机与测试主机之间是连通的。默认情况下，Ping 命令会向对方主机发送四次连通性测试分组，因此图中显示对方主机返回了四个应答（reply）分组。有关 ping 命令的更多信息可参考附录。

有时即便两个主机之间是连通的，对方主机也会不响应 ping 命令的测试分组，如图 1-14 所示。可能的原因是对方主机开启了防火墙或其他原因。此时，可以通过发送信息或共享文件等方式查看网络是否连通。

```

C:\>ping 26.28.249.6

Pinging 26.28.249.6 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 26.28.249.6:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>

```

图 1-14 对方主机不响应 Ping 命令的结果

## 五、相关知识

### 1. ping 命令用法简介

在命令行中输入“ping /?”，能够得到 ping 命令的相关参数，如图 1-15 所示。

```

C:\>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] target_name

Options:
  -t          Ping the specified host until stopped.
              To see statistics and continue - type Control-Break;
              To stop - type Control-C.
  -a          Resolve addresses to hostnames.
  -n count    Number of echo requests to send.
  -l size     Send buffer size.
  -f          Set Don't Fragment flag in packet.
  -i TTL      Time To Live.
  -v TOS      Type Of Service.
  -r count    Record route for count hops.
  -s count    Timestamp for count hops.
  -j host-list Loose source route along host-list.
  -k host-list Strict source route along host-list.
  -w timeout   Timeout in milliseconds to wait for each reply.

```

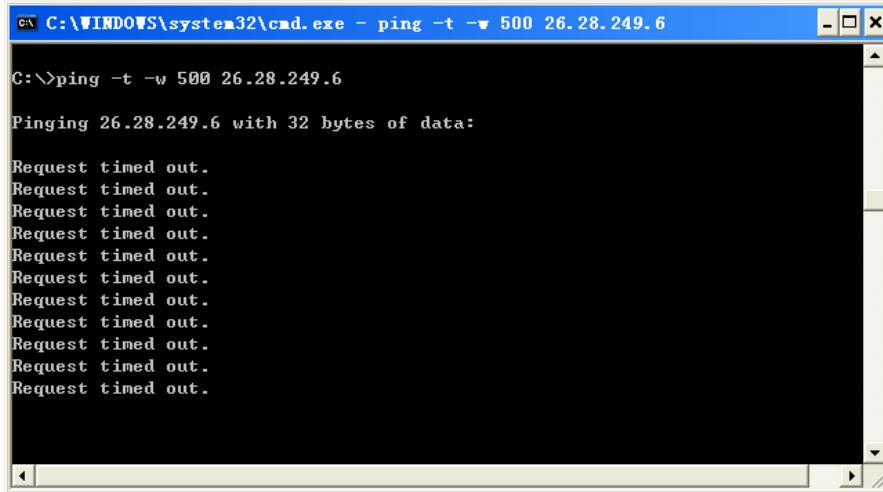
图 1-15 Ping 命令参数

下面我们给出一些常用参数的含义说明。

参数	用法
-t	Ping 指定的计算机直到用户中断命令执行
-n count	发送 count 指定的测试数据包数（默认值为 4）。
-l length	发送包含由 length 指定长度的测试数据包。默认为 32 字节；最大值是 65,527 字节。
-f	在数据包中发送"不要分段"标志。数据包就不会被路由

	上的网关分段。
-w timeout	指定超时间隔，单位为毫秒。

例如，命令“ping -t -w 500 26.28.249.6”表示一直测试与主机 26.28.249.6 之间的连通性，直到用户中断命令执行（同时按“Ctrl”+“C”），且每次测试时的超时间隔为 500 毫秒。



```
C:\>ping -t -w 500 26.28.249.6

Pinging 26.28.249.6 with 32 bytes of data:

Request timed out.
```

图 1-16 Ping 命令的其它使用方法

## 六、注意事项

- 1) 网线钳的刀口十分锋利，使用时要注意安全。
- 2) 爱护公物，不能使用网线钳切割和挤压网线以外的其他东西，或将网线钳掉落在地上。
- 3) 实验过程中，不要随意触摸插座，防止触电。
- 4) 实验完成后，关闭计算机，将网线钳和测试仪规整到位，整洁实验台。

## 七、思考题

双绞线的线序有什么作用，如果线序错误会产生什么结果，如何解决？

# 实验二 串口通信

## 一、实验目的

- (1) 了解串口的通信方式。
- (2) 掌握串口通信的原理，了解串口通信的编程的初步概念和相应函数，掌握一个具体开发平台下的串口编程。

## 二、实验要求

本实验将介绍串行通信的基本原理，以及在 Windows 7、Windows10 环境下用 MFC 实现串口 (COM) 通信的方法，并用串口通信实现简单的通讯协议。

## 三、实验内容

利用 RS-232 通信线缆在两台计算机之间传递数据。

在 Windows 应用程序的开发中，常常需要面临计算机(或单片机)与外围数据源设备进行通信的问题。设计一个相应的串口通信程序，完成数据通信用任务，是一个不错的想法!串行端口的本质功能是作为 CPU 和串行设备间的编码转换器。当数据从 CPU 经过串行端口发送出去时，字节数据转换为串行的位。而在接收数据时，串行的位又被转换为字节数据。

### 二、串口通信的过程

常用的 DOS 系统主要是工作在响应中断方式下。PC 机串行通信程序大多利用其 BIOS 块的 INT 14H 中断，以查询串口的方式完成异步串行通信。与 DOS 响应中断的工作方式不同，在 Windows 环境(Windows NT、Windows 98、Windows2000)下，串口是系统资源的一部分。应用程序要使用串口进行通信。如果想要使用串口进行通信，则必须在使用之前向操作系统提出资源申请要求(打开串口)，通信完成后必须释放资源(关闭串口)。Windows 系统函数已经包含了通信支持中断功能。Windows 系统为每个通信设备开辟了用户定义的输入输出缓冲区(即读 / 写缓冲区)，数据进出通信口均由系统后台来完成，应用程序只需完成对输入输出区操作即可。

详细过程：每接收一个字符，系统产生一个低级硬件中断，Windows 系统中的串行驱动程序就取得了控制权，并将接受到的字符放入输入数据缓冲区，然后将控制权返回正在运行的应用程序。如果输入缓冲区数据已满，串行驱动程序用当前定义的流控制机制通知发送方停止发送数据。队列中的数据按“先进先出”的次序处理。

Windows 系统通过使用一个数据结构对串行口和串行通信驱动程序进行配置，该数据结构被称为设备控制块(Device Control Block)，简称 DCB。

通常可按以下四步实现串行通信：

(1) 按协议的设置初始化并打开串行口。这样做的目的是通知 Windows 本应用程序需要这个串口，并让其封锁其他应用程序，使它们不能使用此串口。

(2) 配置串行口。

(3) 在串口上往返的传输数据，并在传输过程中进行校验。在编写程序时，主要的代码集中在这一步。

(4) 不需要此串口时，关闭串口，即释放串口以供其他应用程序使用。

以上四步的具体流程如图所示：

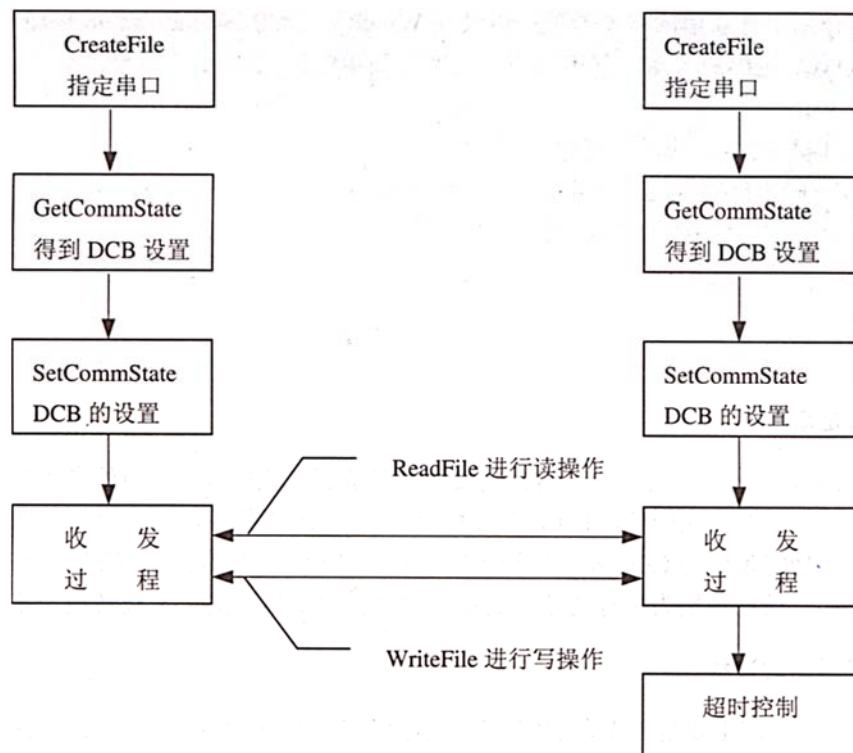


图 1-1 串口通信流程

打开串口的方法一般有四种：

(1) 以文件方式打开串口。这里使用的是 Win32 API 函数，所以无论在 BCB 或 VC 下都可以实现。具体的函数的意义可以参考 Win32 API 的帮助。这里有一个易于使用的 BCB 下的类，可以方便地使用串口。

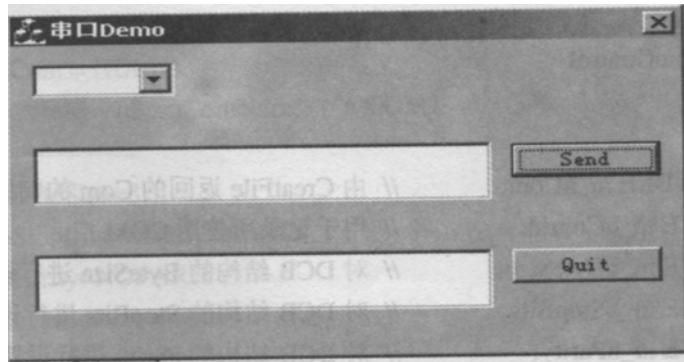
(2) 使用现成的控件。易于使用的是微软的 MS CommFuncication 控件，它是作为一个.ocx 提供的。可以用 Import ActiveX Control 将它加入到 BCB 中，缺省是加载在 ActiveX 页中，就可以作为一个普通的 BCB 控件来使用了(也有不少第三方提供的其他硬件操作控件)。

(3) 直接嵌入汇编法。利用 VS 的直接嵌入汇编功能，可以在 VS 程序中用汇编语言直接对串口进行操作。这种方法可以把原来在 DOS 下开发的汇编程序经过修改后移植到 VS 中继续使用。

## 四、实验步骤

### 1、串口通信

(1) 在 MFC 的应用程序向导里选择 Dialog Based。以下均遵循默认选项。dialog 面板布置控件如图所示。



(2) 各控件的 ID 和 Caption 如下表：

Type	ID	Caption
Breton	IDC_BuTroN_SEND	Send
Button	IDC_BUTTON_QUIT	Quit
EditBox	IDC_EDIT_SEND	
EditBox	IDC_Edrr_RECEIVE	
ComboBox	IDC_COMBO_COM	

打开 Class Wizard 为各控件添加控制变量，如下表：

Type	ID	Variables
EditBox	IDC_EDIT_SEND	CString
EditBox	IDC_EDIT_RECEIVE	CString

(3) 接下来为程序添加一个新类，该类用于控制串口的初始化及其超时控制等等。该类名为：CcomStatus；

- 其头文件为 ComStams.h
- CComControl 的实现文件中本文件中添加几个全局变量：

```
BYTE XwCom=0x40;
BYTE sCom1[5], sCom2[MAXBLOCKLENGTH+12], sCom3[MAXBLOCKLENGTH+12];
BYTE opation;
shortComNum; //串口 ID
CComControl:: CComControl()
{
    m_hCom=NULL;
    m_bComId=(char)ComNum; //COM1
    m_bByteSize=8;
    m_bStopBits=ONESTOPBIT;
    m_bParity=NOPARITY;
    m_dwBaudRate=9600;
    m_bEvtChar=EVENTCHAR;
    m_fBinary=1;
    m_bConnected=FALSE;
```

```
m_bFlowCtrl=FC_XONXOFF;  
m_fXonXoff=FALSE;  
}
```

以下是重载的 ComControl 构造函数：

```
CComConla~ol:: CComControl (BYTE bComId, BYTE bByteSiZC, BYTE bStopBitS,  
BYTE bParity, DWORD dwBaudRate', 'WORD fChEVL', chal"bEvtChar4t  
DWORD fBinary)  
{  
m_hCom=NULL;  
m_bComId=bComId;  
m_bByteSize=bByteSize;  
m_bStopBitS=bStopBitS;  
m_bParity=bParity;  
m_dwBaudRate=dwBaudRate;  
m_bEvtChar=-bEvtChar;  
m_fBinary=fBiIlary;  
m_bConnected=FALSE;  
m_bROWCrtl=FC_XoNXOFF;  
m_fXonXoff=FALSE;
```

OpenConnection 函数：此函数用于设置相应参数。例如，超时控制参数 CommTimeOuts、OVERLAPED，确定通信 COM 口的 ID。

SetupConnection 函数：用于通过变量 dcb 对 DCB 属性进行操作，完成对串口各个属性的配置。在这里请注意参数设置的顺序，以及两个重要函数 GetCommstate 和 SetCommState 的使用。理解 DCB 结构的各个参数的意义是关键。通过计算机串口通信之前，必须根据监控设备的有关通信参数约定双方的通信方式，包括波特率的设置、奇偶校验及停止位的设置等。确定数据传输帧格式，确定 UART 操作方式。逐个对线路控制寄存器、波特率因子寄存器和 MODEM 寄存器进行写入操作。

接着，要查询发送流程，只要 CPU 检测到 UART 发送器保持寄存器为空，即向 UART 输出一个字符。发送方首先输出 RTS 和 DTR 有效，检测 MODEM 寄存器。只有收到 DECshurude CTS、DSR 有效，CPU 才向 UART 输出一发送符。

然后，我们要查询接收流程。只要 CPU 检测到 UART 接收器数据准备就绪，即可从接收器数据寄存器中接收字符。接收方首先输出数据中断就绪有效 (DTR=1)，然后检测 MODEM 状态寄存器 • 只有 DSR=I，CPU 才接收一字符。在 API 函数的串口通信方法中，DCB 参数包含了这些参数的设置。我们只需得到串口的 DCB 结构变量即可。

IsConncted 函数用于判断连接成功；

CloseConnection 显式的关闭了串口文件；

我们在实验指导书中只提供了重要的函数，具体细节请同学们自行编写，在完成了程序的编辑之后，对其进行编译连接、运行。此实验过程需要两台计算机和一根串口线。实验比较简单，只实现了数字的传输。在阅读程序时，建议先掌握原理，对程序中的错误处理可以先略去不看，而只看关键代码。

### 注意：

串行通信程序的调试相对来说是比较麻烦的，一般可以采用以下步骤：

(1) 检查连线是否正确。在三线制方式中，要注意“交叉”问题：还要注意握手信号线

的正确连接。

(2) 简单的用逻辑笔检查发接信号的有无(注意逻辑笔只能检查 TrL 信号。因此，检查点一般为经接口芯片转换后的 TXD, RXD)。

(3) 在确认有接发信号的前提下，如果接发数据不正常，则重点应检查通信协议是否一致，例如波特率的设置、奇偶校验、停止位数、通信的应答等设置。

(4) 在只有单机的情况下串口程序调试，可采用将串口的 TXD 与 RXD 直接相连的办法，简单方便。

## 五、思考题

(1) 串口通信与网络上的通信有什么不同?

(2) 程序中 SENDDATA 这个 union，起到什么样的作用?可不可以用其他类型来代替?

# 实验三 子网划分与路由器配置

## 【实验目的】

- 1) 熟悉利用 CIDR 技术规划分配 IP 地址的基本方法，以及网络参数的配置；
- 2) 熟悉静态路由协议的设置过程；
- 3) 熟悉 RIPv2 协议的配置和运行过程。
- 4) 掌握使用 PacketTracer 模拟网络场景的基本方法，加深对网络环境、网络设备和网络协议交互过程等方面的理解。

## 【实验要求】

- (1) 熟悉路由器的基本配置命令，并学会合理的制作静态路由和动态路由；
- (2) 熟悉交换机的基本配置命令，能够对 Ethernet、Token Ring、FastEthernet、Gi 做出熟练的配置，并学会合理配置 VLAN；利用基本的命令去测试配置的网络结构联通情况；

## 【实验原理】

路由是第三层的概念。网络层在 Internet 中的功能是实现主机到端主机的通信，即无论两台计算机相距多远，中间相隔多少个网络，这一层保障它们可以互相通信。例如我们常用的 PING 命令就是一个网络层的命令，PING 通了，就是指网络层的功能正常了。通常，网络层不保障通讯的可靠性，也就是说，虽然正常情况下数据可以到达目的地，但即便出现异常，网络层也不作任何更正和恢复的工作。网络层常用的协议有 IP、IPX、APPLETALK 等等，其中 IP 协议更是 Internet 的基石。在 TCP/IP 协议体系中，第三层的其他辅助协议还包括 ARP（地址解析）、RARP（反向地址解析）、ICMP（网际报文控制）和 IGMP（组管理协议）等等。由于网络互连设备都具有路径选择功能，所以我们经常将 RIP、OSPF 等路选协议也放在这一层讨论。

交换机的一个重要的功能是避免交换循环，这就涉及到了 STP (Spanning Tree Protocol, 分支树协议)。分支树协议的功能是避免数据帧在交换机构成的网络中循环传送。如下图所示，如果网络中有冗余链路的话，STP 协议现选出根交换机 (Route Bridge)，然后确定每一台非根交换机到根交换机之间的路径，最后，将此路径上的所有链路置成转发 (Forward) 状态，其余的交换机之间的连接就是冗余链路，置为阻塞 (Block) 状态。

交换机的另外一个重要功能是 VLAN (Virtual LAN, 虚拟局域网)。VLAN 的好处主要有三个：

- 1) 端口的分隔。即便在同一个交换机上，处于不同 VLAN 的端口也是不能通信的。这样一个物理的交换机可以当作多个逻辑的交换机使用。
- 2) 网络的安全。不同 VLAN 不能直接通信，杜绝了广播信息的不安全性。
- 3) 灵活的管理。更改用户所属的网络不必换端口和联线，只该软件配置就可以了。

## 【实验内容】

\*\*\*大学具有 4 个校区，分别是 A 校区、B 校区、C 校区以及 D 校区，根据机构和人员的分布情况，每个校区约有 4000 台设备接入校园网络，为了实现校区之间的高速网络数据通信，学校购置了一批路由器和交换机对网络设施进行升级改造，每个校区有一台高速路由器作为接入路由器，且校区路由器之间通过万兆链路互联。本次实验你的任务是：

- (1) 利用 Packet Tracer 软件绘制各个校区之间网络互联的逻辑结构图;
- (2) 根据总部为学校分配的 IP 地址块 26.28.0.0/16, 按照校区划分子网, 并根据每个校区的接入设备数目为每个校区的子网划分 IP 地址块, 并在每个校区网络内选择两台主机作为代表性主机 (A 校区部署一台服务器), 为其分配具体 IP 地址;
- (3) 在校区互联路由器上配置静态路由协议, 实现校区子网之间的连通, 并测试任意两个校区内网络设备的连通性;
- (4) 在校区互联路由器上配置动态 RIPv2 路由协议, 实现校区子网之间的连通, 并测试任意两个校区内网络设备的连通性;
- (5) 在 A 校区网络内部署一台 web 服务器, 其他校区内的主机可以访问该服务器;
- (6) 实验完成后将最后生成的两个 pkt 文件 (分别对应静态路由和 RIP 动态路由) 与实验报告一起上交。

## 四、实验步骤

### 1. 安装网络模拟器

安装 CISCO 网络模拟器 PacketTracer 版本 5.3.0。双击 PacketTracer 安装程序图标, 进入安装过程。根据提示进行选择确认, 可顺利安装系统。

### 2. 使用 PacketTracer 模拟器规划基本网络拓扑

- (1) 启动系统。点击 “CISCO Packet Tracer”图标, 将会出现如图 3-1 所示的系统界面。

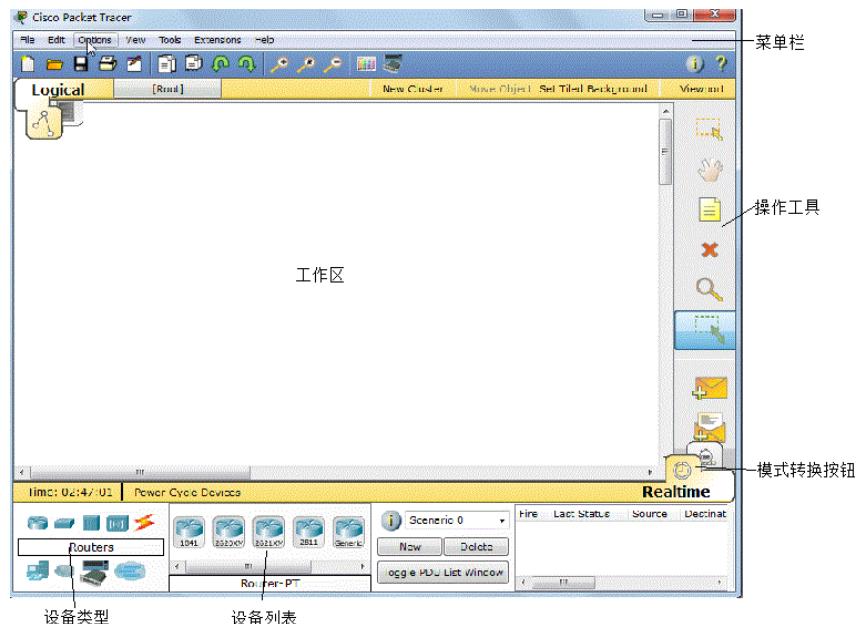


图 3-1 PacketTracer 的主界面

菜单栏中包含新建、打开、保存等基本文件操作, 其下方是一些常用的快捷操作图标。

工作区则是我们绘制、配置和调试网络拓扑图的地方。操作工具位于工作区右边，自上而下有 7 个按钮。这些操作工具的作用分别是：选择(Selected)，用于选中配置的设备；移动(Move Layout)，用于改变拓扑布局；放置标签(Place Note)，用于给网络设备添加说明；删除(Delete)，用于去除拓扑图中的元素，如设备、标签等；检查(Inspect)，用于查询网络设备的选路表、MAC 表、ARP 表等；增加简单的 PDU(Add Simple PDU)，用于增加 IP 报文等简单操作；增加复杂的 PDU(Add Complex PDU)，可在设置 IP 报文后再设置 TTL 值等操作。使用检查工具可查看网络设备（交换机、路由器）的 3 张表，该功能等同于在 IOS 命令行中采用相应的 show 命令，如 show arp。增加简单的 PDU 和增加复杂的 PDU 两个工具用于构造测试网络的报文时使用，前者仅能测试链路或主机之间是否路由可达，后者则具有更多的功能。例如，要测试 PC0 到 Router0 之间的连通性，可先用增加简单的 PDU 工具点击 PC0，再用该工具点击 Router0 就可以查看两设备之间是否连通。如图 3-2 所示。



图 3-2 用增加简单的 PDU 工具测试设备之间的连通性

增加复杂的 PDU 工具的使用方法稍复杂些，也是先用工具依次点击所要测试链路的两端，再设置所要发送的报文格式。如图 3-3 所示。

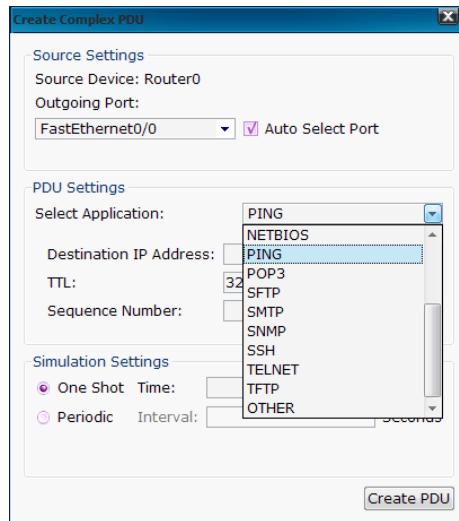


图 3-3 定制增加复杂的 PDU 中的报文

在主界面右下角，是转换实时模式与模拟模式的按钮。在实时模式下，所有操作中报文的传送是在瞬间完成。在模拟状态下，报文的传送是按操作一步一步地向前走，有助于我们仔细地观察报文的具体传输过程。

## (2) 设计绘制网络拓扑图

设计绘制网络拓扑图主要有以下几个步骤：增加网络设备，增加设备硬件模块，连接设备和配置设备等。

**增加网络设备：**在主界面下方有增加网络设备的功能区，该区域有两个部分：设备类别选择区域以及显示某个类别设备的详细型号区域。先点击设备类别，再选择具体型号的设备。

例如，先从左下角区域选择了路由器类别，此时右侧区域将显示可用的各种 CISCO 路由器型号列表，可以从中选用所要的网络设备。

增加设备硬件模块(选项)：如果选用的网络设备恰好适用，则可进行下一步。但有时有些设备基本合用，但还缺少某些功能，如某种硬件接口数量不够等，这就需要通过增加设备硬件模块来解决。例如，我们选择了路由器 2620XM，发现它仅有一个 10/100Mbps 的以太端口，一个控制端口和一个辅助设备端口。我们需要扩展一个光纤介质的 100 Mbps 的以太端口和一些 RJ45 端口的以太端口。这时我们双击工作区路由器 2620XM 图标，可看到如图 3-4 所示的界面。从图中左侧物理模块列表中找出模块 NM-1FE-FX，从左下方窗口中的描述发现它符合我们的要求，就可以将其拖入上部的物理设备视图中。由此，我们可以完成所有相关操作。

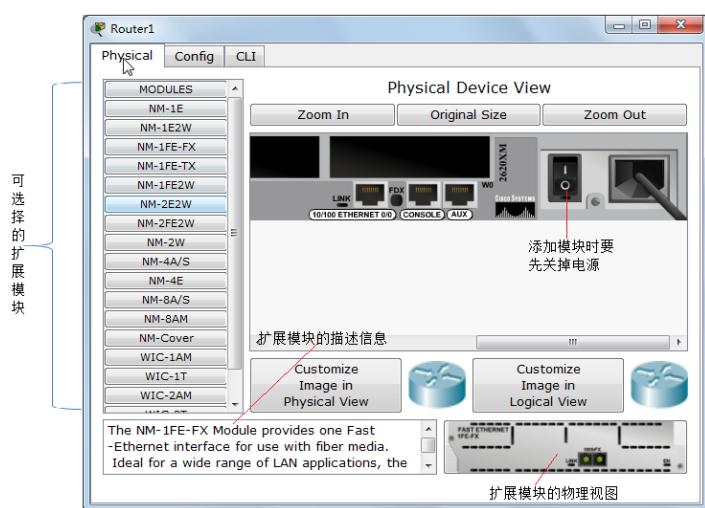


图 3-4 路由器 2620XM 的物理接口

连接设备：在设备类型区域选取“连接(Connections)”，再在右侧选取具体连接线缆类型。注意到连接线缆有如下不同类型：线缆有控制口(Console)、直连铜线(Copper Straight-Through)、交叉铜线(Copper Cross-Over)和光纤(Fiber)等，需要选取适当的线缆类型才能保证设备的正确连通。

配置设备：配置网络设备是一件细致的工作，我们将在其他实验中讲解配置网络设备详细过程。

下面，我们以此为例，绘制一幅简单的网络拓扑图(参见图 3-5)。

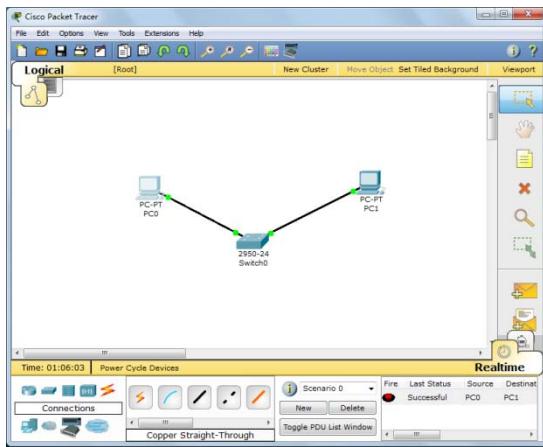


图 3-5 经交换机连接两台 PC

先用上述方法从设备区拖入两台 PC 和一台交换机，再用直通铜线与某个 RJ45 以太端口连接。稍停片刻，线缆端的点将会变绿，表示所有的物理连接都是正确的，否则要检查并排除所存在的物理连接方面的问题。

为了使两台 PC 之间 IP 能够连通，需要进一步配置该网络的网络层协议。双击 PC0 的图标，进入“Config/FastEthernet”界面，我们配置“IP Configuration”。选静态(Static)方式，IP 地址可输入：192.168.1.1，子网掩码可选：255.255.255.0。对 PC1 图标，也进行类似的配置，只是 IP 地址可为：192.168.1.2。为了检验配置是否正确，双击 PC0，进入“Desktop/Command Prompt”界面，键入：ping 192.168.1.2，这时就应当出现 PC1 对该 Ping 响应的信息。由于交换机是一种自配置的设备，无需配置就能使用其基本功能工作。

### 3. 观察与 IP 网络接口的各种网络硬件

为了使 IP 能够通信，网络设备硬件接口之间至少要用一种物理介质连接好，并且要求这些硬件接口与物理介质相匹配。下面，通过实验来研究相关内容。

从 PacketTracer 中打开路由器 2620XM 的物理设备视图，仔细做下列工作：观察有关 NM-1FE-FX 模块描述；将其拖入设备，观察模块面板上的硬件接口情况(可用 Zoom In 放大)；做笔记，并自行分析该模块的适用场合。

对路由器 2620XM 的 NM-1FE-TX、NM-2FE2W、NM-8AM、NM cover plate 模块分别做出上述工作。

### 4. ping 和 traceroute 实验

(1) 启动系统。在网络设备库中选择型号为“1841”的路由器一台，PC 机两台，如图 3-6 所示。

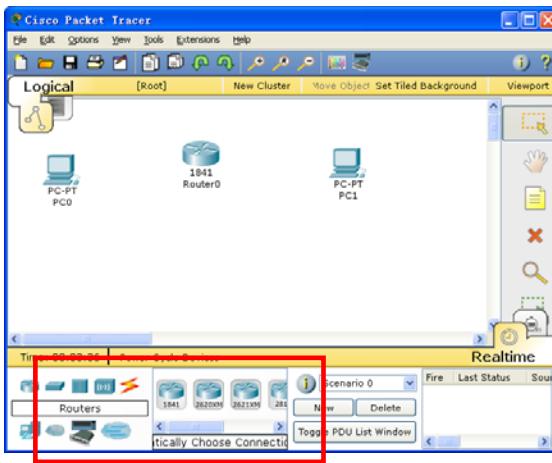


图 3-6 构建网络拓扑

(2) 创建链路。在设备库中选择链路，选择自动添加链路类型，然后分别点击需要添加链路的设备，结果如图 3-7 所示，此时链路两端红色表示链路不通。

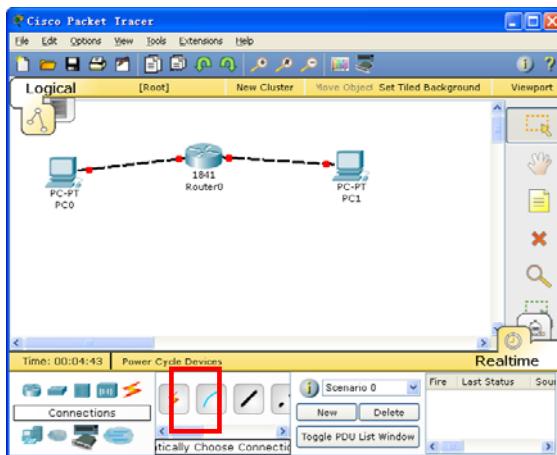


图 3-7 添加链路

(3) 配置网络设备。双击设备，得到设备的配置界面。在 PC 机的配置界面中，选择“Desktop”标签，选择“IP Configuration”，配置 PC 机的地址信息，如图 3-8 所示。按上述方法，将 PC0 的 IP 设置为 192.168.1.2，子网掩码 255.255.255.0，默认网关 192.168.1.1。用同样的方法设置 PC1 的的 IP 为 192.168.2.2，子网掩码 255.255.255.0，默认网关 192.168.2.1。

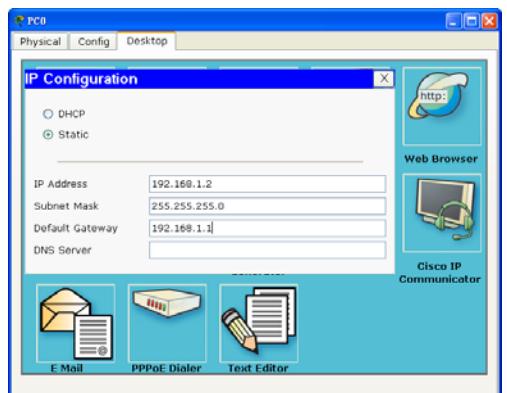


图 3-8 PC 配置

配置路由器端口。设置 Router0，在路由器配置界面中选择“config”标签，选择“FastEthernet0/0”，将 IP 设置成 192.168.1.1，子网掩码 255.255.255.0，同样设置“FastEthernet0/1”，将 IP 设置成 192.168.2.1，子网掩码 255.255.255.0，如图 3-9 所示，注意将路由器端口打开。

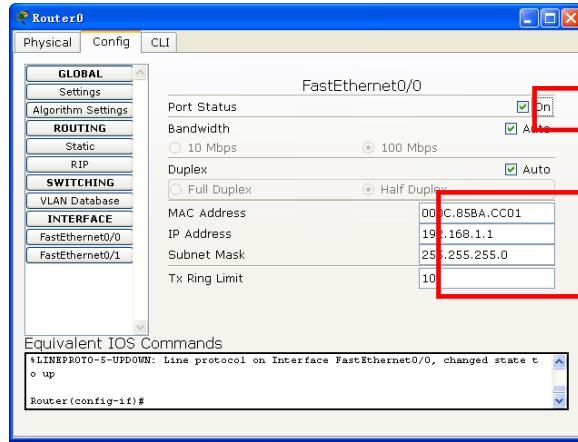


图 3-9 路由器配置

(4) 使用 Ping 命令，并在模拟模式下观察。如图 3-10 所示，进入模拟模式。双击 PC0 的图标，选择“Desktop”标签，选择“Command Prompt”，输入“ping 192.168.2.2”，如图 3-11 所示。同时，点击“Auto capture/play”按钮，运行模拟过程，观察事件列表“Event List”中的报文。

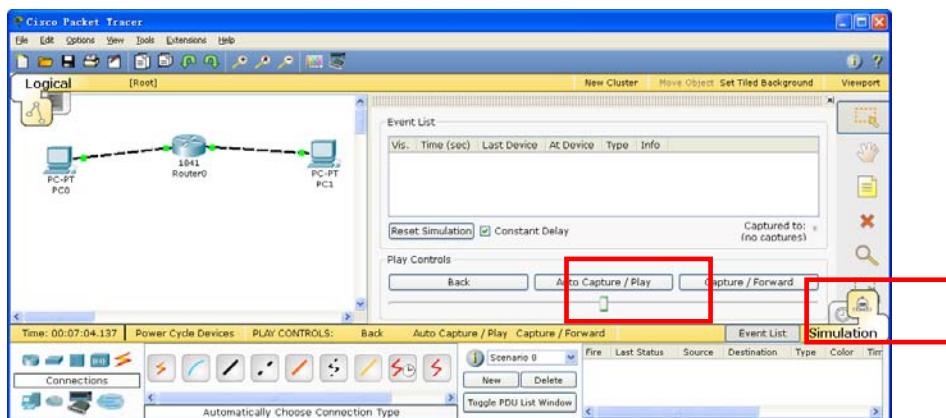


图 3-10 进入模拟模式

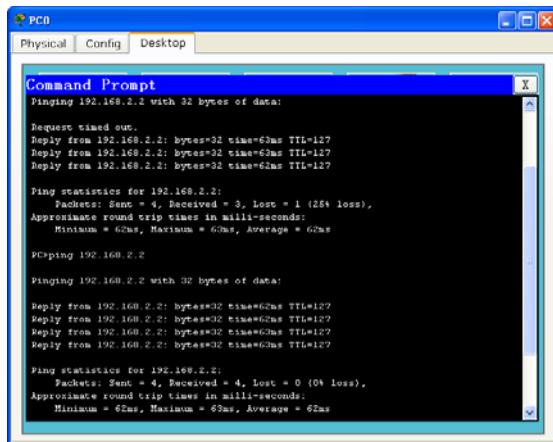


图 3-11 运行 ping 命令

(5) 使用 tracert 命令，并在模拟模式下观察。

## 5. 在 PacketTracer 模拟器中配置星型网络拓扑

在 PacketTracer 模拟器中配置如下图所示的网络拓扑，其中通用交换机连接 4 台普通 PC，通用集线器 hub 连接 2 台普通 PC。

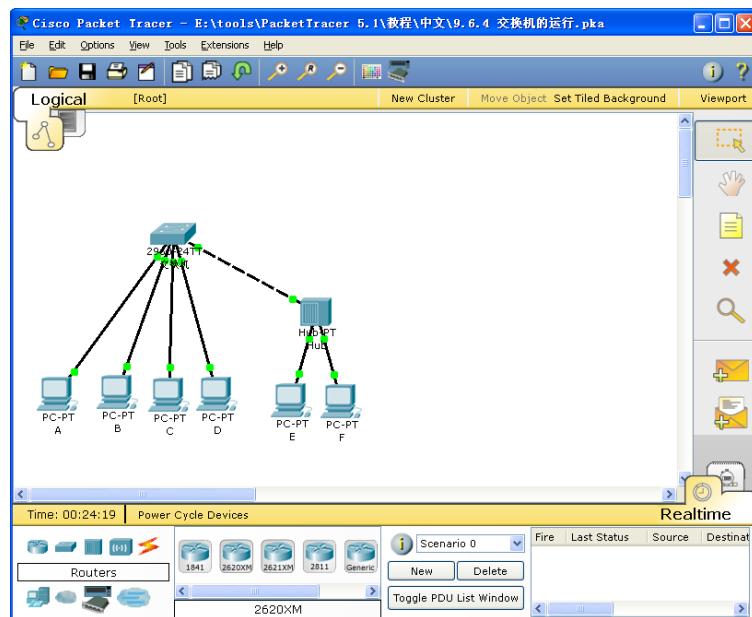


图 3-12 实验网络拓扑图

单击 PC，在每台 PC 的配置窗口中配置合理的 IP 地址和子网掩码，无需为交换机和集线器配置 IP 地址（为什么？）。

## 6. 观察交换机如何处理广播和已知单播

(1) 在实时与模拟模式之间切换 4 次，完成生成树协议。所有链路指示灯应变为绿色。最后停留在模拟模式中。

(2) 使用 Inspect（检查）工具（放大镜）打开 PC A 和 PC B 的 ARP 表以及交换机的 MAC 表。本练习不关注交换机的 ARP 表。将选择箭头移到交换机上，查看交换机端口及其接口 MAC 地址的摘要。请注意，这不是交换机获取的地址表。将窗口排列在拓扑上方。

(3) 添加简单 PDU 以从 PC A 发送 ping 到 PC B（也可在 PC A 的 DeskTop 窗口中打开模拟命令行“Command Prompt”，运行 PING 命令）。

使用 Add Simple PDU（添加简单 PDU）（闭合的信封）从 PC A 发送一个 ping 到 PC B。单击 PC A（源），然后单击 PC B（目的）。Event List（事件列表）中将会显示两个事件：一个 ICMP 回应请求和一个 ARP 请求，用以获取 PC B 的 MAC 地址。单击 Info（信息）列中的彩色框以检查这些事件。

(4) 逐步运行模拟。

使用 Capture/Forward（捕获/转发）按钮跟踪数据包的最终顺序。由于 PC A ARP 表中没有 PC B 的相应条目，因此为了完成 ping，它必须发出 ARP 请求。交换机从 ARP 请求获取 PC A 的 MAC 地址及其连接的端口，从 ARP 回复获取 PC B 的 MAC 地址及其连接的端口，交换机会将 ARP 请求从所有端口泛洪出去，因为 ARP 请求始终是广播。PC A 收到 ARP 回复之后，便可完成 ping。从交换机的角度来看，ping 是已知单播。完成对数据包的跟踪之后，单击 Reset Simulation（重置模拟）按钮。

## 7. 观察交换机如何处理未知单播

(1) 清除交换机的 MAC 地址表。

单击交换机。单击 CLI 选项卡。在出现命令提示符时，按几次 Enter 键，将会显示 Switch> 提示。键入 enable 并按 Enter 键。提示应会变为 Switch#。键入命令 clear mac-address-table dynamic 并按 Enter 键。请注意，早先显示的交换机 MAC 表重新为空。但仍会填充 PC ARP 表。关闭交换机配置窗口。

(2) 重新发送数据包。

您应该还是处于模拟模式。用户创建的 PDU（在任务 1 中创建的从 PC A 到 PC B 的 ping）仍然在 Event List（事件列表）中。使用 Capture/Forward（捕获/转发）按钮跟踪数据包的最终顺序。由于 ARP 表已经填充，因此无需 ARP 请求。但是，当回应请求数据包到

达 MAC 地址表为空的交换机时，将被视为未知单播。在这种情况下，交换机就会像处理广播一样，将数据包从除接收端口以外的所有其它端口泛洪出去。完成之后，单击 Delete（删除）按钮删除场景 0。

## 8. 在 PacketTracer 模拟器中配置复杂园区网络拓扑

在本实验中，根据各个校区的地理位置，我们设计校区间间的网络互联拓扑结构。需要注意的是，本实验中所选择的 1841 路由器标配仅带有两个 10/100Mbps 的以太端口，而根据我们设计的网络拓扑。B 校区和 A 校区的路由器需要 3 个端口，因此要为它们各增加一个以太网端口。可在路由器的物理设备视图中增加 WIC-1ENET 模块，从而增加一个 10 Mbps 以太网接口，添加后如图 3-13 所示。

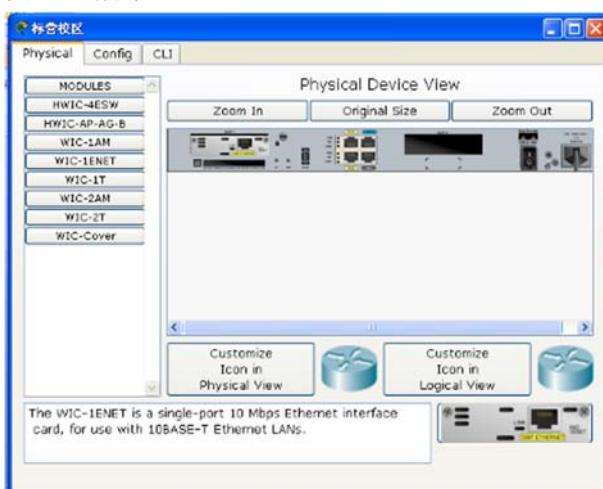


图 3-13 添加路由器接口模块

最后，分别利用直连铜线和交叉铜线互联各个网络设备，形成如图 3-14 所示的物理网络拓扑结构。

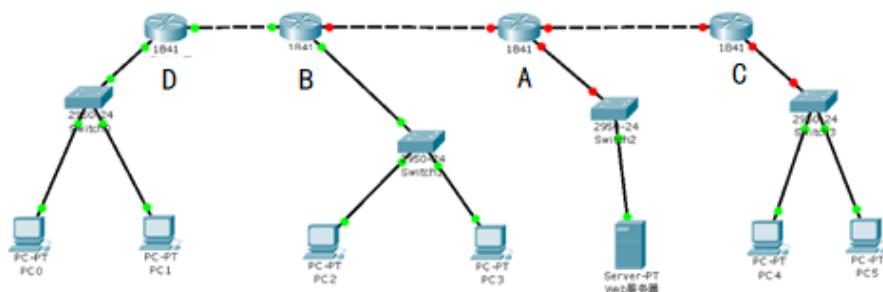


图 3-14 校区间网络互联拓扑结构

## 3. 利用 CIDR 技术划分 IP 地址块

完成拓扑规划后，下一步工作即如何为每个校区划分 IP 地址块，以满足各个校区的接

入设备数量要求。由于总部为学校分配的 IP 地址块 26.28.0.0/16，其网络前缀为 16bit，因此后面可变的主机地址位数为  $32-16=16$ bit，总的可用地址数量为  $2^{16} = 65536$ ，因此地址数量能够满足学校总的地址需求。每个校区网络接入设备约 4000 台， $2^{12} = 4096$ ，因此每个校区需要的主机地址位数为 12bit。

此外，不同校区的接入路由器互联也需要划分一个子网空间，因此，一种可行的 IP 地址划分机制如表 3-1 所示。需要指出的是，这里地址空间划分方式并不唯一，只要能够满足接入主机数目要求的划分机制均可。

表 3-1 一种可行的地址空间划分方案

校区	地址空间前缀	IP 地址数目	子网掩码
D 校区	26.28.0.0/20	4096	255.255.240.0
D-B	26.28.16.0/20	4096	255.255.240.0
B 校区	26.28.32.0/20	4096	255.255.240.0
B-A	26.28.48.0/20	4096	255.255.240.0
A 校区	26.28.64.0/20	4096	255.255.240.0
A-C	26.28.80.0/20	4096	255.255.240.0
C 校区	26.28.96.0/20	4096	255.255.240.0

#### 4. 配置主机 IP 地址

以配置 D 校区子网内主机 PC0 为例，双击“PC0”，在弹出窗口内选择“Config”属性页，配置 PC0 的 IP 地址，子网掩码。（我们假定 D 校区子网内，主机 PC0 分配的 IP 为 26.28.0.2）

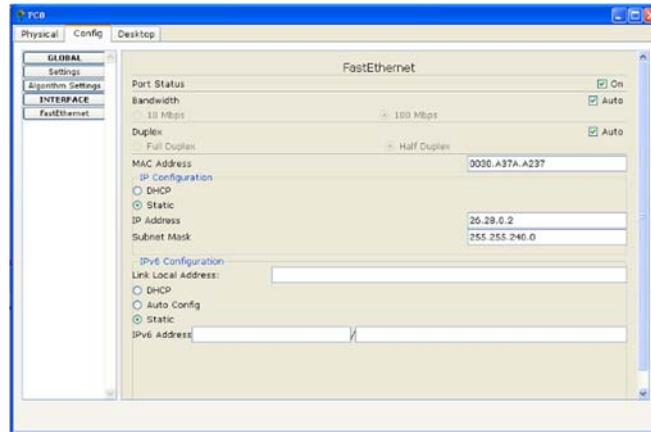


图 3-15 主机 IP 地址配置

配置 PC0 的默认路由器（网关）。拓扑图上，D 校区路由器显然是该校区子网的网关，其与 D 校区子网相连的接口的 IP 地址为 26.28.0.1：

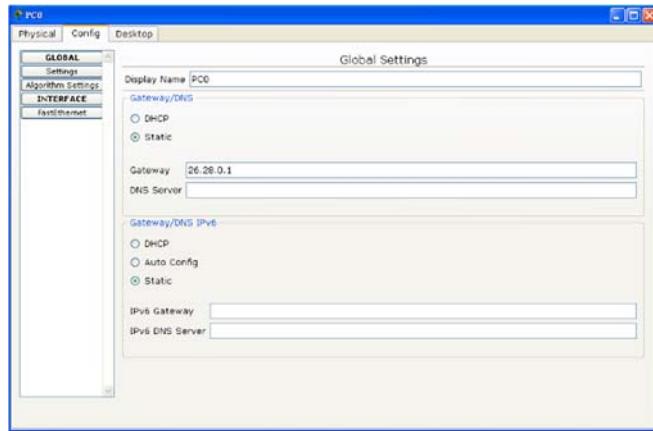


图 3-16 主机网关配置

类似的可以对其他主机的 IP 地址进行配置：

配置 PC1 的 IP 地址为 26.28.0.3，网关为 26.28.0.1；

配置 B 校区子网内的主机 PC2,PC3 的 IP 地址为 26.28.32.2 和 26.28.32.3，网关为 26.28.32.1。

配置 A 校区子网内的主机 Web 服务器的 IP 地址为 26.28.64.2，网关为 26.28.64.1。

配置 C 校区子网内的主机 PC6, PC7 的 IP 地址为 26.28.96.2 和 26.28.96.3，网关为 26.28.96.1。

## 5. 配置路由器的接口 IP 地址

每一个路由器均有多个网络接口，因此需要多个 IP 地址。在利用 Packet Tracer 进行路由器 IP 地址配置时，有图形界面和命令行两种配置模式，而实际工作中对路由器进行配置通常需要通过串口以命令行的模式进行配置，因此本部分我们首先以图形界面方式配置路由器，随后给出相应的命令行配置模式，供同学们参考。

以配置 D 校区接入路由器为例，双击 D 校区路由器图标，点击“Config”选项卡。先配置 FastEthernet0/0 端口，根据图示的 IP 地址规划将其配置为 26.28.0.1，子网掩码为 255.255.240.0；点击 FastEthernet0/1 端口将其配置为 26.28.16.1，子网掩码为 255.255.240.0。其中 On 选项表示启动该接口。图 3-17 下面部分显示了用户每次在图形界面上的操作所对应的思科 IOS 命令。

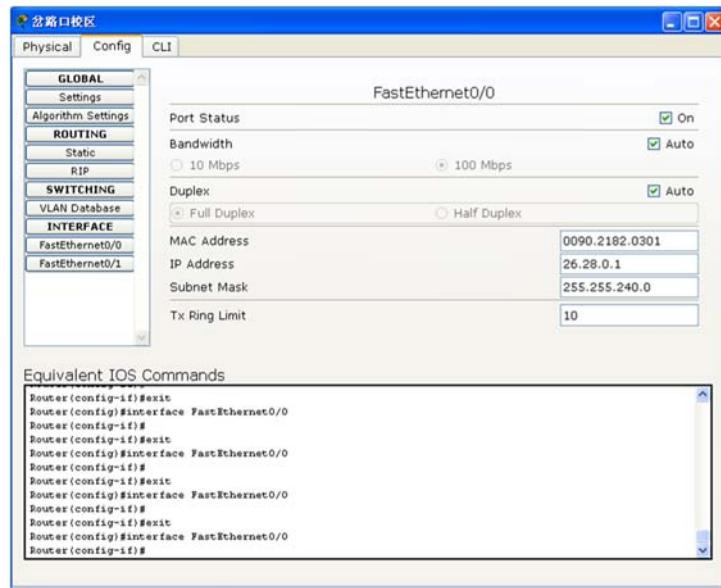


图 3-17 配置路由器接口 IP 地址

配置成功以后，把鼠标放在 D 校区路由器上一段时间后，可以在提示信息里看到两个接口的 IP 地址都已经配置成功了。并且，如图 3-18 所示，该路由器和 D 校区子网之间的线路也变成绿色了，表明物理线路已经畅通。

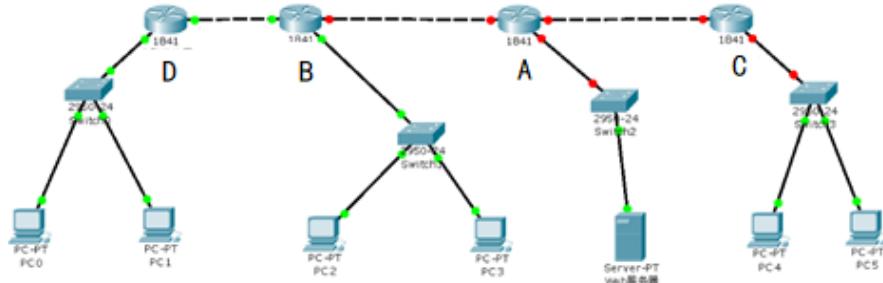


图 3-18 D 校区路由器接口配置结果

实际工作中，在配置路由器参数时，大多是以命令行界面的方式进行配置，注意看 Config 页面下方的 Equivalent IOS Commands（等价的命令）。例如，刚才我们对 D 校区路由器的配置操作，也可以用如下的命令实现：

```

Press RETURN to get started!
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 26.28.0.1 255.255.240.0
Router(config-if)#no shutdown

```

```

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#ip address 26.28.16.1 255.255.240.0
Router(config-if)#no shutdown

```

对于其他路由器的配置与上述过程类似，这里不再赘述。配置全部完成后，观察网络拓扑，所有的链路已经变成绿色（畅通），如图 3-19 所示。但此时仅是物理联通，路由表并没有配置。无法跨在子网之间进行 IP 数据报的传递。

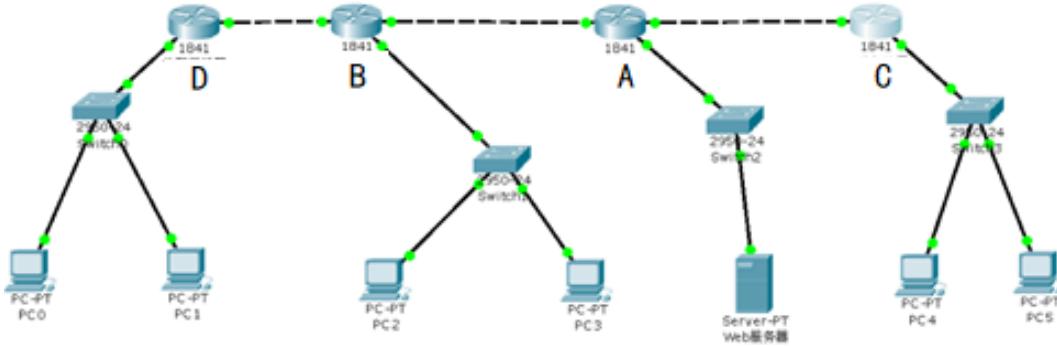


图 3-19 路由器接口地址配置成功

完成上述配置工作后，在 Packet Tracer 主窗口的右侧，找到按钮“Inspect”，即放大镜图标，点击后，在 D 校区路由器上单击，选择“Routing Table”，查看其路由表。

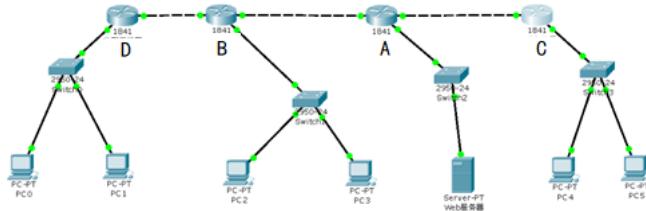


图 3-20 查看路由器路由表

此时可以发现该路由器对应的路由表只有两项，都是与其直接相连的子网。

Routing Table for 岔路口校区				
Type	Network	Port	Next Hop IP	Metric
C	26.28.0.0/20	FastEthernet0/0	---	0/0
C	26.28.16.0/20	FastEthernet0/1	---	0/0

图 3-21 路由器路由表项

随后双击 PC0 的图标，在弹出窗口中，选择“Desktop”页面，上面放置了该主机可以提供的应用程序。



图 3-22 主机功能选项

选择“command prompt”图标，弹出类似于 Windows 的 CMD 命令行窗口，在该窗口内输入 ping 26.28.64.2，发现尽管整个网络物理上已经连通，但位于不同子网内的主机仍无法通信（因为没有配置路由）。

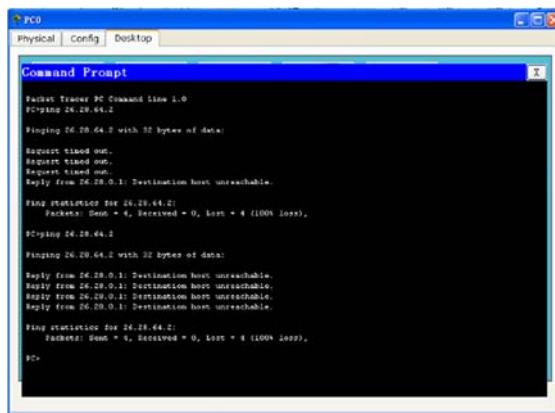


图 3-23 主机连通性测试

## 6. 配置静态路由

D 校区路由器的静态路由配置：

(a) 双击 D 校区路由器，在 Config 窗口里，选择 route -> static。为每一个子网增加静态路由。在弹出的窗口内输入到达其他子网的静态路由，然后单击“add”加入路由表。

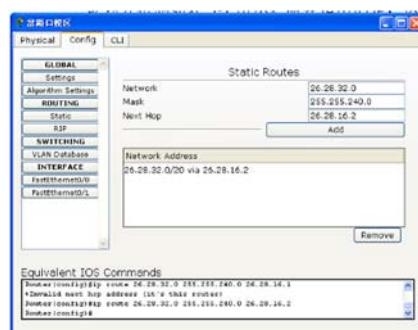


图 3-24 添加静态路由

注意：这里 Next Hop 指的是下一跳路由器接口的 IP 地址。

(b) 按照类似的方式，配置到达子网 26.28.32.0, 26.28.48.0, 26.28.64.0, 26.28.80.0 的静态路由。配置的结果如下：(用 Inspect 查看 D 校区路由器的路由表)

Routing Table for 岔路口校区				
Type	Network	Port	Next Hop IP	Metric
C	26.28.0.0/20	FastEthernet0/0	---	0/0
C	26.28.16.0/20	FastEthernet0/1	---	0/0
S	26.28.32.0/20	---	26.28.16.2	1/0
S	26.28.48.0/20	---	26.28.16.2	1/0
S	26.28.64.0/20	---	26.28.16.2	1/0
S	26.28.80.0/20	---	26.28.16.2	1/0
S	26.28.96.0/20	---	26.28.16.2	1/0

图 3-25 查看路由表

上述配置过程也可以通过命令行的方式进行，其基本指令如下。

```
Router>enable          //进入特权模式  
Router#conf t          //进入全局配置模式  
  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)# ip route 26.28.32.0 255.255.240.0 26.28.16.2  
Router(config)# ip route 26.28.48.0 255.255.240.0 26.28.16.2  
Router(config)# ip route 26.28.64.0 255.255.240.0 26.28.16.2  
Router(config)# ip route 26.28.80.0 255.255.240.0 26.28.16.2  
Router(config)# ip route 26.28.96.0 255.255.240.0 26.28.16.2
```

配置完毕后，也可以通过 show ip route 命令查看路由器现在的路由表。

(c) 按照类似的方式，配置 B 校区路由器的静态路由如图 3-26 所示。

Routing Table for 标营校区				
Type	Network	Port	Next Hop IP	Metric
C	26.28.16.0/20	FastEthernet0/0	---	0/0
C	26.28.32.0/20	Ethernet0/1/0	---	0/0
C	26.28.48.0/20	FastEthernet0/1	---	0/0
S	26.28.0.0/20	---	26.28.16.1	1/0
S	26.28.64.0/20	---	26.28.48.2	1/0
S	26.28.80.0/20	---	26.28.48.2	1/0
S	26.28.96.0/20	---	26.28.48.2	1/0

图 3-26 B 校区路由器路由表

(d) 按照类似的方式，配置 A 校区路由器的静态路由如图 3-27 所示。

Routing Table for 中心校区				
Type	Network	Port	Next Hop IP	Metric
C	26.28.48.0/20	FastEthernet0/0	---	0/0
C	26.28.64.0/20	Ethernet0/1/0	---	0/0
C	26.28.80.0/20	FastEthernet0/1	---	0/0
S	26.28.0.0/20	---	26.28.48.1	1/0
S	26.28.16.0/20	---	26.28.48.1	1/0
S	26.28.32.0/20	---	26.28.48.1	1/0
S	26.28.96.0/20	---	26.28.80.2	1/0

图 3-27A 校区路由器路由表

(e) 按照类似的方式，配置 C 校区路由器的静态路由如图 3-28 所示。

Routing Table for 双龙街校区				
Type	Network	Port	Next Hop IP	Metric
C	26.28.80.0/20	FastEthernet0/0	---	0/0
C	26.28.96.0/20	FastEthernet0/1	---	0/0
S	26.28.0.0/20	---	26.28.80.1	1/0
S	26.28.16.0/20	---	26.28.80.1	1/0
S	26.28.32.0/20	---	26.28.80.1	1/0
S	26.28.48.0/20	---	26.28.80.1	1/0
S	26.28.64.0/20	---	26.28.80.1	1/0

图 3-28C 校区路由器路由表

## 7. 测试主机之间的连通性

(a) 首先，采用 ping 命令测试任意两台计算机之间的连通性，在位于 D 校区子网的 PC0 上向位于 C 校区子网的 PC5 发起 ping 测量，图 3-29 显示了测量结果，可见经过在各个路由器上配置静态路由，位于不同子网内的主机之间已经能够正常通信。

```
PC0
Physical Config Desktop
Command Prompt
IP Address.....: 26.28.0.2
Subnet Mask.....: 255.255.240.0
Default Gateway.: 26.28.0.1

PC>ipconfig

IP Address.....: 26.28.0.2
Subnet Mask.....: 255.255.240.0
Default Gateway.: 26.28.0.1

PC>ping 26.28.96.2

Pinging 26.28.96.2 with 32 bytes of data:
Reply from 26.28.96.2: bytes=32 time=21ms TTL=124
Reply from 26.28.96.2: bytes=32 time=14ms TTL=124
Reply from 26.28.96.2: bytes=32 time=157ms TTL=124
Reply from 26.28.96.2: bytes=32 time=203ms TTL=124

Ping statistics for 26.28.96.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 14ms, Maximum = 219ms, Average = 180ms

PC>
```

图 3-29 不同子网内主机间 ping 测量结果

(b) 其次，通过浏览器测试主机到 A 校区子网内 Web 服务器的连通性。双击 Web 服务器主机，在 Config 窗口内，查看 http 服务器的配置：

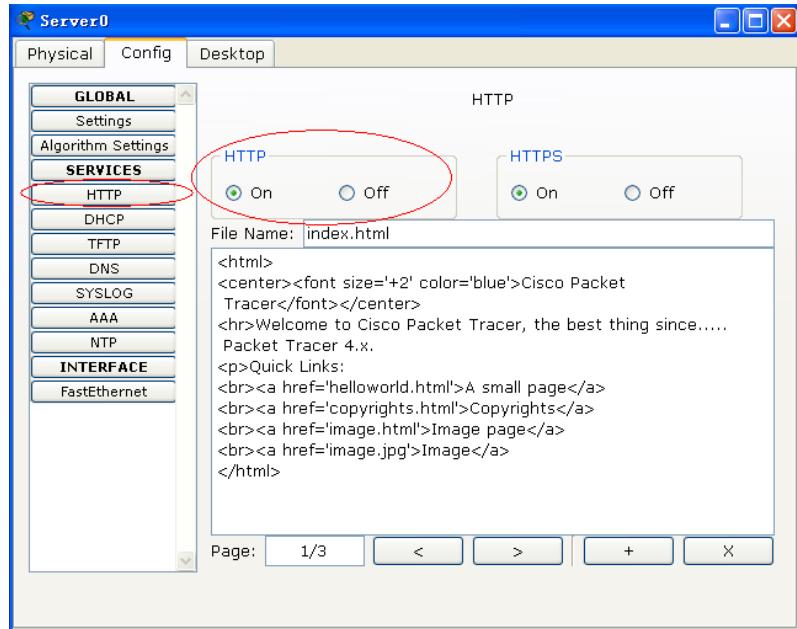


图 3-30 Web 服务器配置

双击客户主机 PC0，在 Desktop 窗口内，点击“Web Browser”虚拟应用，在浏览器的地址栏内，输入 `http://26.28.64.2`，以便连接 http 服务器，图 3-31 显示了 Web 访问情况。



图 3-31 Web 服务器访问结果

结果表明，D 校区 PC0 和 A 校区 Web 服务器之间可以通信。利用其它主机访问 Web 服务器也可以得到类似结果。

## 8. 配置动态路由协议 RIP

动态路由协议采用自适应路由算法，能够根据网络拓扑的变化而重新计算最佳路由。RIP 协议是一种广泛使用的域内选路协议，其全称是 Routing Information Protocol，采用 Bellman-Ford 算法。RFC1058 是 RIP version 1 标准文件，RFC 2453 是 RIP Version 2 的标准文档。

针对步骤 1) ~7) 划分的子网, 首先删除步骤 8) 中配置的静态路由信息, 随后配置路由器执行 RIPv2 算法, 动态产生路由表。为了模拟实际路由器的配置过程, 本部分配置过程中我们全部采用命令行模式进行。以对 D 校区路由器进行配置为例进行分析, 单击 D 校区路由器的图标, 点击 CLI 窗口。在命令行模式下首先设置 IP 地址。

```
Router>enable          //进入特权模式
Router#conf t          //进入全局配置模式
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet 0/0    //进入路由器接口配置模式
Router(config-if)#ip address 26.28.0.1 255.255.240.0 //配置接口 0/0 的 IP 地址和掩码
Router(config-if)#no shutdown      //开启接口
Router(config-if)#interface fastEthernet 0/1    //进入接口 0/1 配置
Router(config-if)#ip address 26.28.16.1 255.255.240.0 //配置接口 0/1 的 IP 地址和掩码
Router(config-if)#no shutdown      //激活接口
Router(config-if)#exit
```

上述命令为路由器两个端口分别设置了 IP 地址 26.28.0.1 以及 26.28.16.1, 子网掩码均为 255.255.240.0。

随后配置 RIP 协议, 输入 exit 进入 config 状态, version 2 命令代表使用 RIP 版本 2, 随后将路由器直接相连的两个网络地址 26.28.0.0 以及 26.28.16.0 向邻居路由器发布。

```
Router(config-if)#exit      //后退进入全局配置模式
Router(config)#router rip    //配置 RIP 协议
Router(config-router)#version 2 //指定 RIP 的版本为 RIP V2
Router(config-router)#network 26.28.0.0 //通告 26.28.0.0 子网
Router(config-router)#network 26.28.16.0 //通告 26.28.16.0 子网
Router(config-router)#no auto-summary //不允许自动路由聚合
```

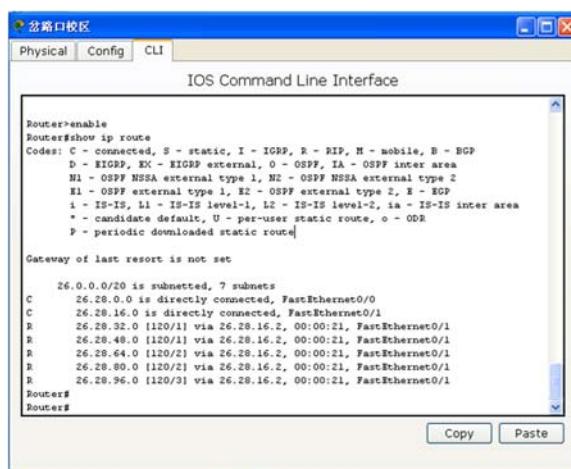


图 3-32 查看路由器路由表

其他路由器配置与此类似, 同样首先配置 IP 地址, 随后配置 RIP 协议, 所有路由器配置

完成后，点击 D 校区路由器，进入 CLI 命令行模式，执行下述命令，可以查看路由器当前路由表信息如图 3-23 所示。

```
Router>enable
```

```
Router#show ip route
```

由路由表可见，26.28.0.0 和 26.28.16.0 两个子网与该路由器直接相连，其他子网均需要经过路由器 26.28.16.2 转发，且到不同的子网距离分别为 1~3 不等。

配置完成后，可以同样采用步骤 9) 中的方法测试主机间的连通性。

思考： 路由器如何通过相互交换信息获得（更新）自己的路由表？

## 9. 查看路由器交换 RIP 报文的过程

(a) 选中 Packet Tracer 主窗口右下角的“Simulation”按钮，进入模拟模式，在该模式下，能够查看报文的交换过程。

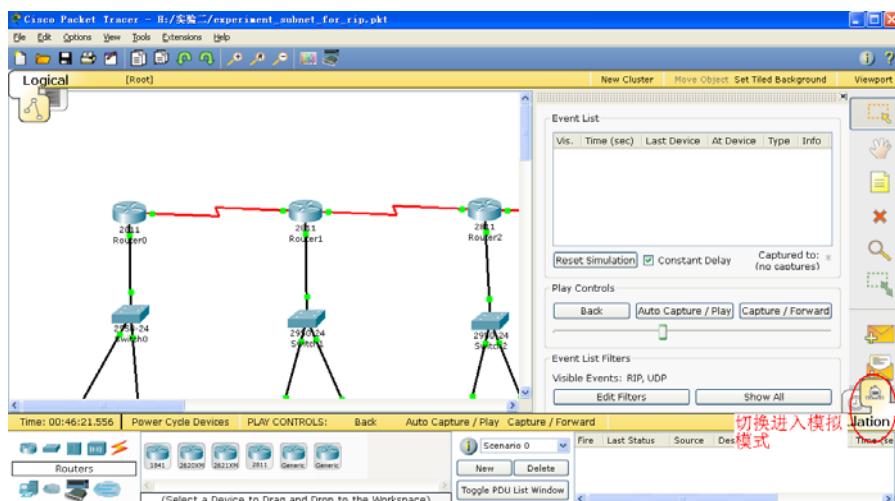


图 3-33 选择分组通信模式

(b) 选中“Edit Filter”按钮，将除 RIP 和 UDP 之外的选项全部去除（只捕获符合 RIP 和 UDP 的报文）

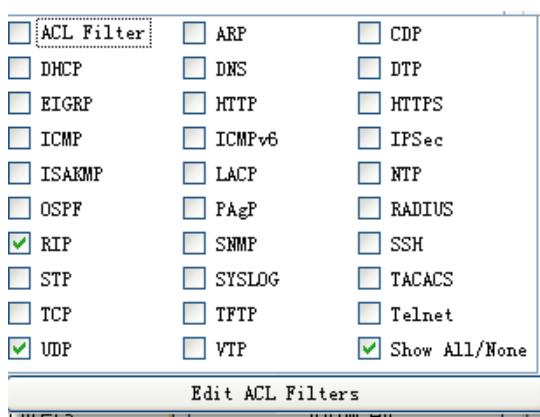


图 3-34 过滤显示分组类型

(c) 连续点击“capture/forwad”按钮，观察网络上产生的 RIP v2 报文传递的方向。

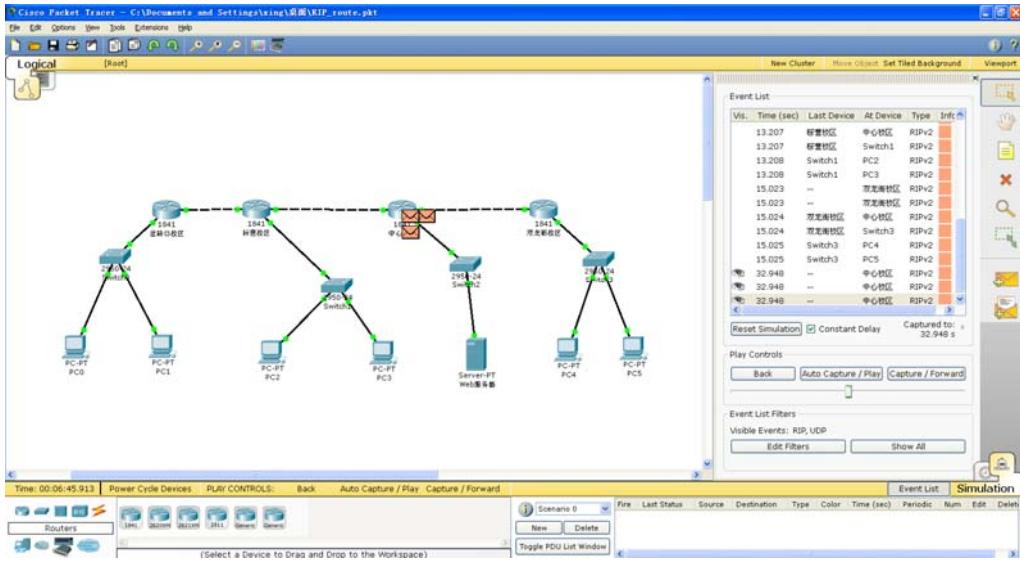


图 3-35 模拟模式下查看 RIPv2 报文

(d) 双击任何一个 RIPv2 报文，可以查看报文的首部信息。分析发现该报文的目的地址为 224.0.0.9，为一个多播地址，表明 RIPv2 协议在向邻居路由器进行路由通告时采用了多播技术。

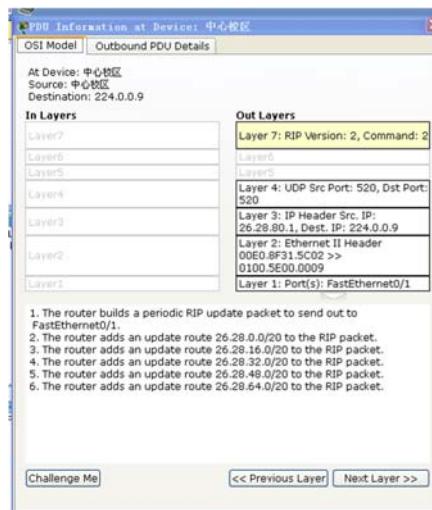


图 3-36 RIPv2 报文基本结构

随后，在 outbound PDU detail 窗口内，可以查看报文的详细信息，包括 RIP 报文的具体内容（请大家结合 RIP 算法的运行过程对报文分析）。

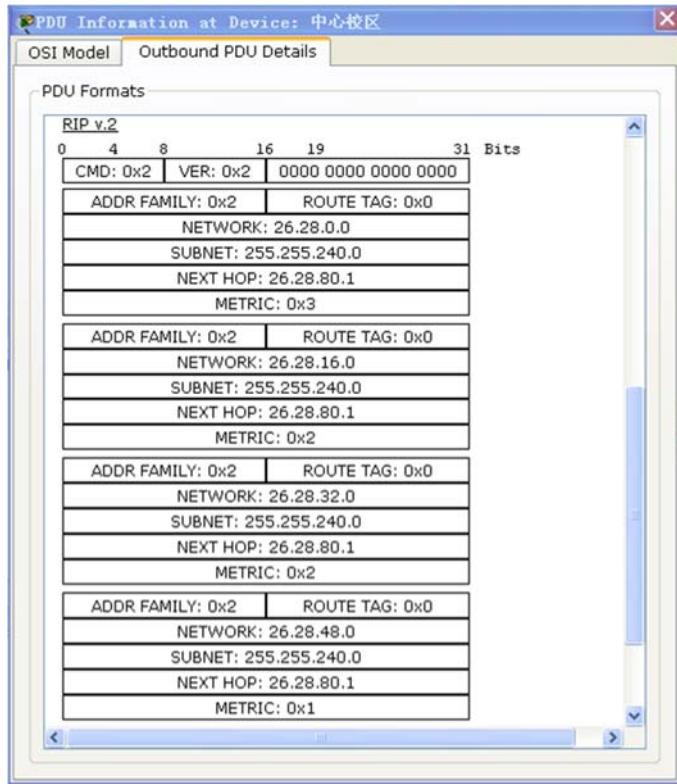


图 3-37 RIPv2 报文数据信息

## 相关知识

(1) 选路算法。当分组从发送方流向接收方时，网络层必须决定这些分组所采用的路由或路径。计算这些路径的算法被称为选路算法 (routing algorithm)。

(2) 选路信息协议。选路信息协议 (Routing Information Protocol, RIP) 是一种最早的用于自治系统内部选路协议，用于确定在一个自治系统内执行选路的方式，目前仍在广泛使用。它的产生与命名源于 Xerox 网络系统 (XNS) 体系结构。RIP 的广泛应用主要是由于它被包含在支持 TCP/IP 的 1982 年 UNIX 伯克利软件分布 (BSD) 版本中。在[RFC 1058]中定义了 RIP 版本 1，在 RFC 2453 中定义了它的向后兼容的版本 2。RIP 是一种距离向量协议，在 RFC 1058 中定义的 RIP 版本使用跳数作为其费用测度，即每条链路的费用为 1。

## 思考及注意事项

(1) 在网络规划过程中我们将两个直接相连的路由器也作为一个子网处理，并为其分配了一个 4096 个地址的子网地址空间，显然造成了 IP 地址资源的很大浪费，有无更好的解决办法？

(2) 本实验设计的网络拓扑中任意校区之间物理链路的失效都将导致网络不再连通，对此你有无好的解决办法？请提出你的设计思路并分析其优劣。

(3) 在增减网络设备接口卡要关闭电源，配置之前则要打开电源。

(4) 如果网络设备之间物理连线错误，则网络设备之间连接点显示红色，显示绿色则表示物理连线正确。

# 实验四 TCP 文件传输的设计与实现

## 【实验目的】

通过文件传输程序的设计，加深学生对 TCP 原理的理解，使同学们初步掌握 Windows 环境下使用 Socket 开发的方法，培养学生综合思考与设计能力。

## 【实验要求】

- (1) 要求学生预习 C 语言的基础知识，基本 WinSock 函数的使用。
- (2) 了解 TCP 传输的基本原理。

## 【思考题】

- (1) 为什么不能一次性地使用 send()和 recv()函数收发整个文件，而是要放在循环中完成？
- (2) 为什么要设置发送和接收缓冲区？
- (3) 发送和接收缓冲区大小设置得不一致会不会有问题？TCP 能否保证每一次 send()发送的数据和 recv()接收到的是一一对应的？

## 【实验内容】

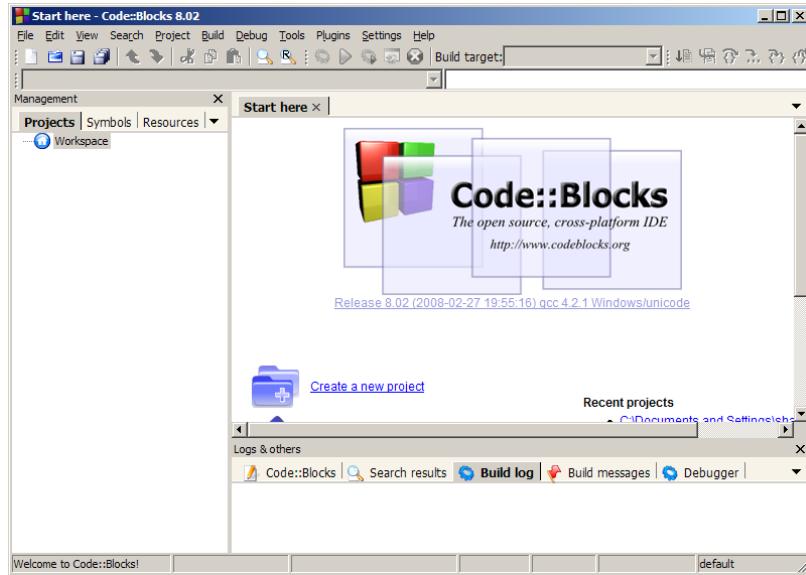
- (1) 熟悉 Code::Blocks 开发环境
- (2) 文件传输服务器端和客户端程序
- (3) 编译运行文件传输服务器端和客户端源代码
- (4) 改进程序

## 【实验步骤】

### 1. Code::Blocks 开发环境简介

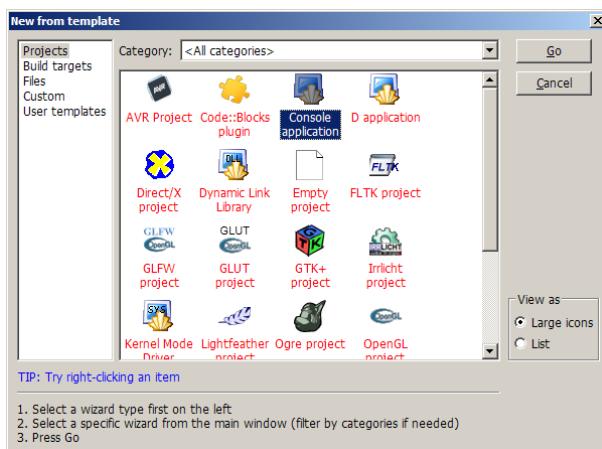
Code::Blocks 是一个开源、免费、跨平台（支持 Windows、GNU/Linux、Mac OS X 以及其他类 UNIX）、支持插件扩展的 C/C++集成开发环境。本课程实验所使用的是 Windows 环境下打包好的绿色版，包含了 GCC 编译器和常用帮助文件。

把压缩包解压缩到某一个目录(注意目录中不能含有中文)，双击其中的“codeblocks.exe”即可打开集成开发环境。

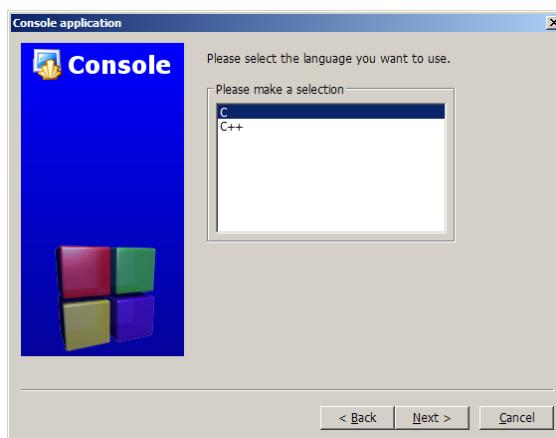


## 创建项目

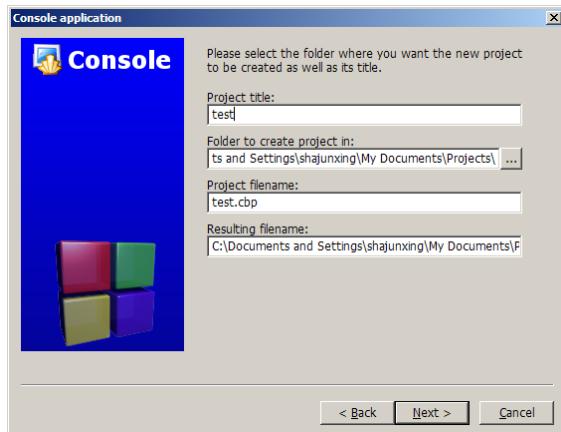
Code::Blocks 的使用方法和 VC 是类似的，单击菜单 “File” - “New” - “Project”，弹出对话框新建一个项目，此处选 “Console Application”，单击 “Go” 按钮进入到下一步。



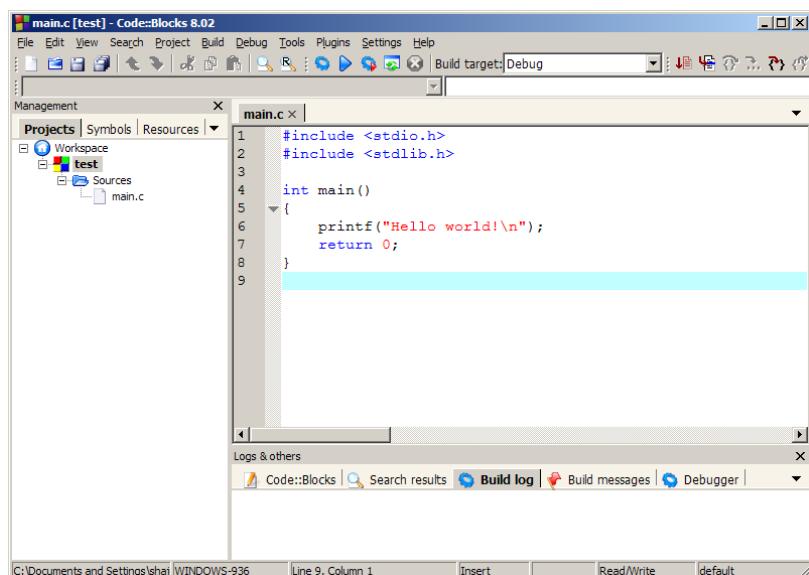
在语言选项中选择 “C”，注意 C 和 C++ 在语法上是有一定区别的，不同的选择将影响到编译器的参数设置和默认代码的生成。



项目的名字和目录注意不能包含中文，然后继续单击 “Next” 按钮，直到创建好项目。



项目创建好后，展开左侧“Projects”目录树，可以看到，已经默认创建了一个“main.c”文件，该文件很简单，即打印一行“Hello world”信息，后续可以在此文件基础上进行开发，也可以在项目中新建其它的.c 文件和.h 文件。



### 编译运行

单击菜单“Build” - “Build”可以编译项目，如果有错误将会在下方“Build log”窗口中显示。

单击菜单“Build” - “Select target”中的子项可以切换编译目标，默认有“Debug”和“Release”两种选择。前者编译出来的可执行程序中带有调试符号，可以进行调试，缺点是文件比较大；后者编译出来的文件小，适合调试通过后正式发布用。

编译后的可执行程序位于项目对应目录中。单击菜单“Build” - “Run”即可执行编译好的程序。

### 调试

修改程序：

```
#include <stdio.h>
#include <stdlib.h>
```

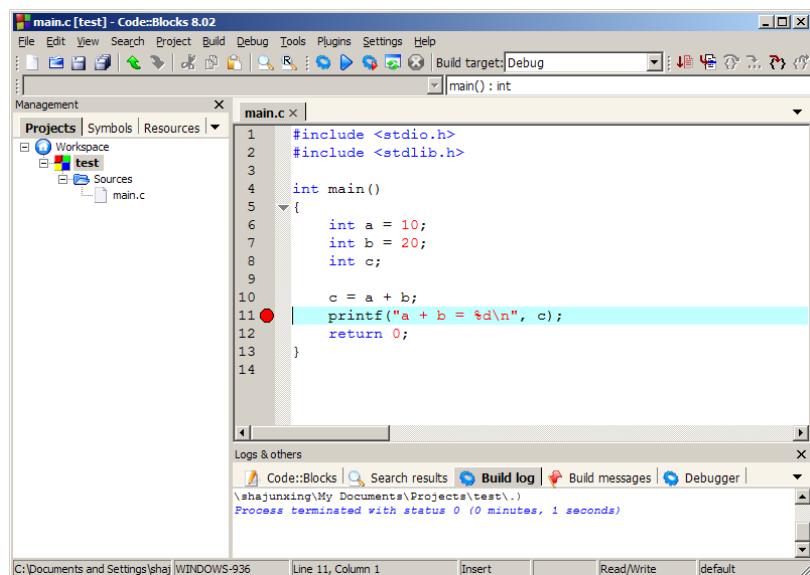
```

int main()
{
    int a = 10;
    int b = 20;
    int c;

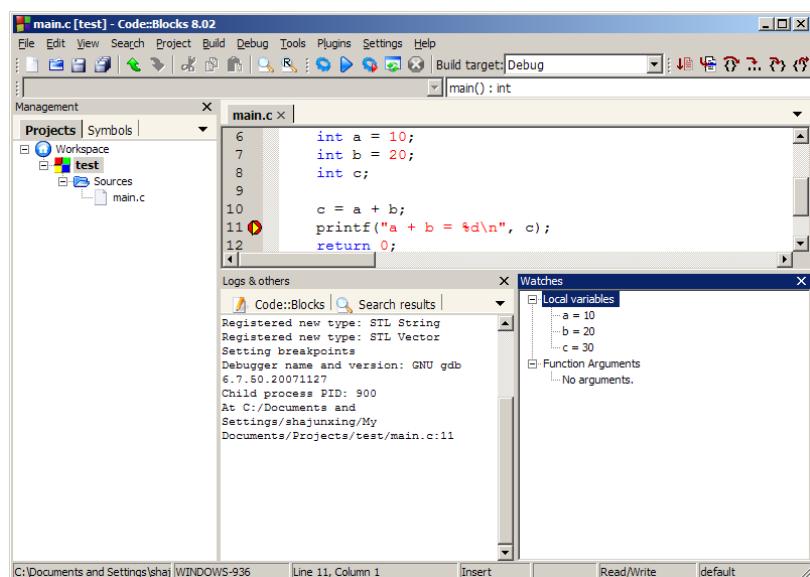
    c = a + b;
    printf("a + b = %d\n", c);
    return 0;
}

```

然后光标定位到“printf("a + b = %d\n", c);”这一行，按下 F5 键或者单击行号右侧的空白区，会看到出现一个红点。这个红点学名叫“断点”，是程序调试的重要手段之一。

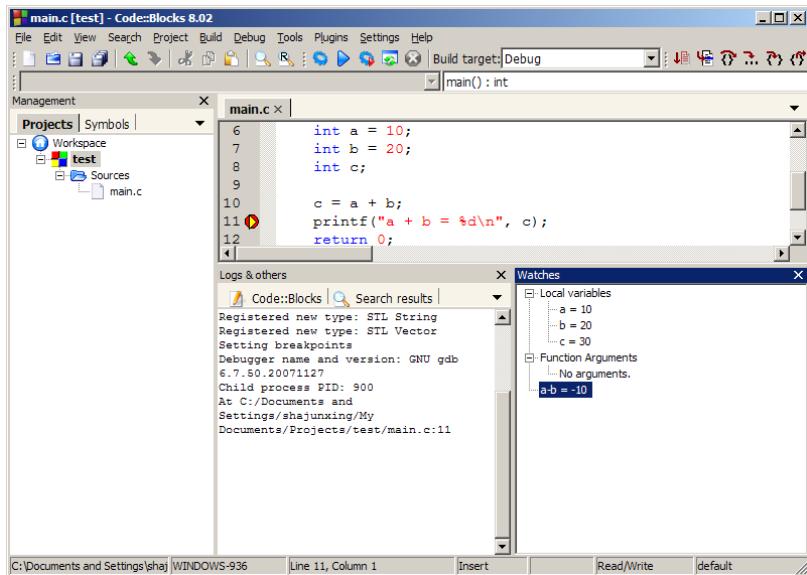


将编译目标设置为“Debug”，重新编译，然后单击菜单“Debug” - “Start”，即可开始调试程序。此时程序开始执行，并停在了断点的地方，如图所示黄色三角形指示的行。



在右下方“Watches”窗口中可以看到所有本地变量的值，甚至可以自行添加 watch，例

如在“Watches”窗口中右键菜单选择“Add watch”，然后在弹出对话框中输入“a-b”，确定，可以看到 Watch 甚至把值计算出来了。



如果要继续执行程序，那么单击菜单“Debug” - “Continue”，如果程序设置有多个断点，那么将会停到下一个断点。如果要强行终止程序，那么那么单击菜单“Debug” - “Stop debugger”即可。

断点和 Watch 是程序调试的两项必备技术，必须掌握。

查看帮助

将光标置于“printf”函数上，单击菜单“Help” - “C Runtime Library Reference”，可以查看该函数的详细帮助说明。注意，不同的函数按照分类位于“Help”下的不同帮助中。

## 2. Windows Socket API 入门

头文件和库文件

Windows Socket 函数的定义包含在 winsock2.h 头文件中，而具体实现位于 ws2\_32.dll 中，程序链接的时候必须包含 ws2\_32.lib 库文件。

初始化和资源释放

在所有基于 Windows Sockets 的程序里，初始化函数 WSAStartup() 和终止函数 WSACleanup() 是必须使用的，通常放在程序代码的开始和结束，用法是：

```
WSADATA wsaData;
// Windows 特有的初始化动作
WSAStartup(0x202, &wsaData);

...
...
...

// Windows 特有的关闭动作
```

```
WSACleanup();
```

### 创建和释放套接字

套接字（Socket）Socket 是一个抽象的概念，类似于文件句柄，它把复杂的 TCP/IP 协议族隐藏在 Socket 接口后面，对前台用户来说，程序开发的时候只需要面对一个套接字描述符和一组简单的函数，而在后台，Socket 负责组织数据，使用指定的协议进行通信。

创建套接字使用 socket() 函数，例如如果要创建 TCP 套接字，那么：

```
SOCKET s = socket(AF_INET, SOCK_STREAM, 0);
```

若无错误发生，返回新套接字的描述符，否则返回 INVALID\_SOCKET。

和文件操作 fopen() 不一样的是，socket() 创建的套接字并没有和具体的 IP 地址/端口相关联，这需要其它函数实现。

类似于 fclose() 函数，closesocket() 函数负责关闭一个套接字，例如：

```
closesocket(s);
```

### 绑定地址，建立连接

在 TCP 协议的服务器端，bind() 函数可将套接字同 IP 地址/端口绑定到一起，例如以下程序将套接字 s 绑定到地址 127.0.0.1:80：

```
struct sockaddr_in addr;

memset((void *)&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
addr.sin_port = htons(80);

bind(ls, (struct sockaddr *)&addr, sizeof(addr));
```

一旦绑定之后，就可以监听并等待客户端的连接了：

```
listen(s, SOMAXCONN);
SOCKET ss = accept(s, NULL, NULL);
```

accept() 函数是阻塞等待的，一旦有客户端连接上来，该函数就返回一个新套接字 ss，后续可使用该套接字和客户端进行数据通信。而原有的套接字 s 可以继续进行 accept()，等待其它客户端的连接。

在 TCP 协议的客户端，connect() 函数同时起到了连接服务器以及绑定本地地址的作用，例如：

```
struct sockaddr_in server_addr;

memset((void *)&server_addr, 0, sizeof(server_addr));
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");
server_addr.sin_port = htons(80);
```

```
connect(s, (struct sockaddr *)&server_addr, sizeof(server_addr));
```

一旦连接成功，`connect()`函数返回 0，失败返回 `SOCKET_ERROR`。连接成功后，操作系统会给套接字 `s` 自动分配本地端口（一般大于 1024），然后就可以使用该套接字和服务器端进行数据通信了。

### 数据收发

TCP 连接是全双工的，两端可以同时进行数据收发。

发送数据使用 `send()` 函数，定义如下：

```
int send (
    SOCKET s,
    const char FAR * buf,
    int len,
    int flags
);
```

参数 `s` 是由 `accept()` 函数返回、或者经过 `connect()` 函数调用的已经建立连接的套接字，该函数将在这个套接字上发送数据。第二个参数 `buf` 是包含待发送数据的缓冲区。第三个参数 `len` 指定 `buf` 的长度。最后一个参数 `flags` 一般填 0。

如果成功调用，`send()` 返回实际发送的字节数；若发生错误，返回 `SOCKET_ERROR`。

接受数据使用 `recv()` 函数。定义如下：

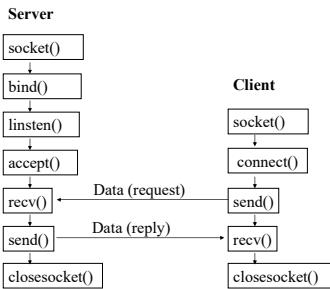
```
int recv (
    SOCKET s,
    char FAR* buf,
    int len,
    int flags
);
```

参数 `s` 是由 `accept()` 函数返回、或者经过 `connect()` 函数调用的已经建立连接的套接字，该函数将在这个套接字上接收数据。第二个参数 `buf`，是用来存放收到数据的缓冲区，第三个参数 `len` 指定缓冲区 `buf` 的长度。最后一个参数 `flags` 一般填 0。

如果没有错误发生，`recv()` 返回接收到的字节数；如果连接已中止，返回 0；若发生错误，返回 `SOCKET_ERROR`。

### 总结

在 Windows 环境下 TCP 开发的一般流程是：



### 3. 编译运行文件传输服务器端和客户端源代码

服务器端和客户端的源代码如下。

服务器端：

```

///////////
// 文件传送服务器端
/////////
#include <stdio.h>
#include <stdlib.h>
#include <winsock2.h>

// 定义接收缓冲区大小
#define MAX_DATA_BLOCK_SIZE 8192

void error_exit(const char * msg, int val);
void serve_at(u_short port);
void serve_client(SOCKET s);

// 主函数
int main(int argc, char ** argv) {
    u_short port;

    if (argc == 1) {
        // 如果不指定监听端口，那么默认为 8888
        serve_at(8888);
    } else if (argc == 2) {
        // 用户指定了监听端口
        port = (u_short) atoi(argv[1]);
        if (port == 0) {
            error_exit("非法的监听端口", -1);
        } else {
            serve_at(port);
        }
    } else {

```

```
        error_exit("参数错误", -1);
    }

    return 0;
}

// 打印错误、使用说明并退出程序
void error_exit(const char * msg, int val) {
    if (msg) {
        printf("%s\n\n", msg);
    }
    printf("使用方法: ft_server [监听端口]\n");
    printf("监听端口是可选参数, 默认为 8888\n\n");
    exit(val);
}

// 在指定端口监听并等待客户端连接
void serve_at(u_short port) {
    WSADATA wsaData;
    SOCKET ls; // 监听套接字
    SOCKET as; // 处理客户端连接的套接字
    struct sockaddr_in addr;
    struct sockaddr_in cli_addr;
    int cli_addr_len;

    // Windows 特有的初始化动作
    WSAStartup(0x202, &wsaData);

    // 创建监听套接字
    ls = socket(AF_INET, SOCK_STREAM, 0);

    // 填写地址结构
    memset((void *)&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = inet_addr("0.0.0.0"); // 在所有 IP 地址上监听
    addr.sin_port = htons(port);

    // 绑定监听套接字和地址结构，并做好监听准备
    bind(ls, (struct sockaddr *)&addr, sizeof(addr));
    listen(ls, SOMAXCONN);

    printf("服务器已启动, 监听于端口%d...\n", port);
}
```

```
for (;;) {
    cli_addr_len = sizeof(cli_addr);
    memset((void *)&cli_addr, 0, cli_addr_len);
    // 等待客户端连接，返回标志该客户端连接的套接字
    // 该函数是阻塞的
    as = accept(ls, (struct sockaddr *)&cli_addr,
&cli_addr_len);
    printf("客户端%s:%d 已连接\n", inet_ntoa(cli_addr.sin_addr),
ntohs(cli_addr.sin_port));
    // 处理该客户端的连接
    serve_client(as);
}

// 关闭套接字
closesocket(ls);
// Windows 特有的关闭动作
WSACleanup();
}

// 处理某一个客户端的连接
void serve_client(SOCKET s) {
    char file_name[MAX_PATH];
    char data[MAX_DATA_BLOCK_SIZE]; // 接收缓冲区
    int i;
    char c;
    FILE * fp;

    // 首先接收文件名，文件名以\0结尾
    printf("接收文件名...\n");
    memset((void *)file_name, 0, sizeof(file_name));
    for (i = 0; i < sizeof(file_name); i++) {
        if (recv(s, &c, 1, 0) != 1) {
            printf("接收失败或客户端已关闭连接\n");
            closesocket(s);
            return;
        }
        if (c == 0) {
            break;
        }
        file_name[i] = c;
    }
    if (i == sizeof(file_name)) {
        printf("文件名过长\n");
        closesocket(s);
    }
}
```

```

        return;
    }

printf("文件名为%s\n", file_name);

fp = fopen(file_name, "wb");
if (fp == NULL) {
    printf("无法以写方式打开文件\n");
    closesocket(s);
    return;
}

// 然后接收文件内容
// 注意观察缓冲区 data 是如何使用的，以及 recv 函数的返回值是如何处理的
printf("接收文件内容");
for (;;) {
    memset((void *)data, 0, sizeof(data));
    i = recv(s, data, sizeof(data), 0);
    putchar('.');
    if (i == SOCKET_ERROR) {
        printf("\n接收失败，文件可能不完整\n");
        break;
    } else if (i == 0) {
        printf("\n接收成功\n");
        break;
    } else {
        fwrite((void *)data, 1, i, fp);
    }
}

// 关闭文件句柄和套接字
fclose(fp);
closesocket(s);
}

```

客户端：

```

///////////
// 文件传送客户端
/////////
#include <stdio.h>
#include <stdlib.h>
#include <winsock2.h>

// 定义发送缓冲区大小

```

```
#define MAX_DATA_BLOCK_SIZE 8192

void error_exit(const char * msg, int val);
void send_file(const char * file_name, const char * ip, u_short port);

// 主函数
int main(int argc, char ** argv) {
    u_short port;

    if (argc == 3) {
        // 如果不指定服务器端口，那么默认为 8888
        send_file(argv[1], argv[2], 8888);
    } else if (argc == 4) {
        // 用户指定了服务器端口
        port = (u_short) atoi(argv[1]);
        if (port == 0) {
            error_exit("非法的服务器端口", -1);
        }
    } else {
        send_file(argv[1], argv[2], port);
    }
    return 0;
}

// 打印错误、使用说明并退出程序
void error_exit(const char * msg, int val) {
    if (msg) {
        printf("%s\n\n", msg);
    }
    printf("使用方法: ft_client <文件名> <服务器 IP 地址> [服务器端口]\n");
    printf("服务器端口是可选参数，默认为 8888\n\n");
    exit(val);
}

// 发送文件到服务器
void send_file(const char * file_name, const char * ip, u_short port) {
    WSADATA wsaData;
    SOCKET s;
```

```
FILE * fp;
struct sockaddr_in server_addr;
char data[MAX_DATA_BLOCK_SIZE];
int i;
int ret;

fp = fopen(file_name, "rb");
if (fp == NULL) {
    printf("无法打开文件\n");
    return;
}
```

```
WSAStartup(0x202, &wsaData);
```

```
// 创建套接字
s = socket(AF_INET, SOCK_STREAM, 0);

// 填写服务器的地址结构
memset((void *)&server_addr, 0, sizeof(server_addr));
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = inet_addr(ip);
server_addr.sin_port = htons(port);

// 连接到服务器，注意观察是如何处理连接失败的
if (connect(s, (struct sockaddr *)&server_addr,
sizeof(server_addr)) == SOCKET_ERROR) {
    printf("连接服务器失败\n");
    fclose(fp);
    closesocket(s);
    WSACleanup();
    return;
}

// 首先发送文件名以及标志文件名结束的\0
printf("发送文件名...\n");
send(s, file_name, strlen(file_name), 0);
send(s, "\0", 1, 0);

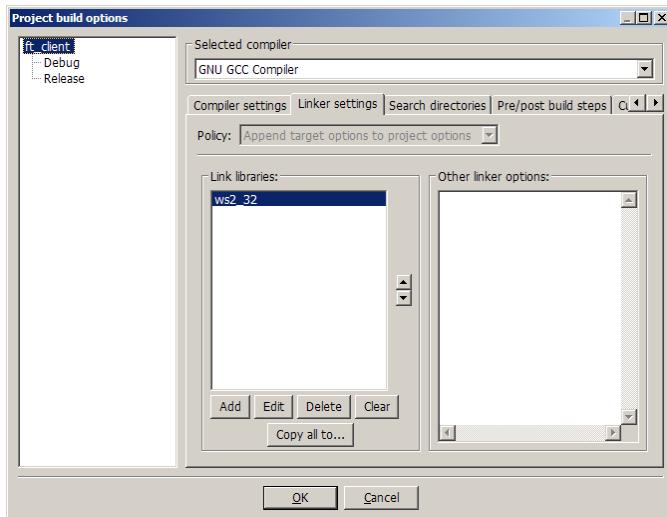
// 然后发送文件内容
// 注意观察缓冲区 data 是如何使用的，以及 fread、send 函数的返回值是如何处理的
printf("发送文件内容");
for (;;) {
    memset((void *)data, 0, sizeof(data));
```

```

        i = fread(data, 1, sizeof(data), fp);
        if (i == 0) {
            printf("\n发送成功\n");
            break;
        }
        ret = send(s, data, i, 0);
        putchar('.');
        if (ret == SOCKET_ERROR) {
            printf("\n发送失败，文件可能不完整\n");
            break;
        }
    }
    fclose(fp);
    closesocket(s);
    WSACleanup();
}

```

在 Code::Blocks 中创建服务器端和客户端的项目，把源代码导入，注意项目的“Build options”中，需要添加库文件“ws2\_32”，然后进行编译。



服务器端的使用方法是：

**ft\_server [监听端口]**

监听端口是可选参数，默认为 8888。

客户端的使用方法是：

**ft\_client <文件名> <服务器 IP 地址> [服务器端口]**

服务器端口是可选参数，默认为 8888。

一条 TCP 连接必须包含本地 IP 地址、本地端口、对端 IP 地址、对端端口四要素，那么在 Socket 中也必须包含这四个要素，使用以下函数在程序各个阶段打印 Socket 的四要素，并进行分析，结果写到实验报告中。

```
void print_socket_detail(SOCKET s)
{
    struct sockaddr_in name;
    int namelen;

    namelen = sizeof(name);
    memset(&name, 0, namelen);
    getsockname(s, (struct sockaddr *)&name, &namelen);
    printf("local: %s:%d\n", inet_ntoa(name.sin_addr),
        ntohs(name.sin_port));

    namelen = sizeof(name);
    memset(&name, 0, namelen);
    getpeername(s, (struct sockaddr *)&name, &namelen);
    printf("peer: %s:%d\n", inet_ntoa(name.sin_addr),
        ntohs(name.sin_port));
}
```

传送一个较大的大文件，传送文件过程中，以及传送结束一段时间内，在命令提示符中使用“netstat -ano”命令观察客户端和服务器端连接状态，记入实验报告中。

#### 4. 程序改进

阅读源代码，总结客户端和服务器端的文件传送协议，并写到实验报告中。

服务器启动后会一直循环等待客户端的连接以及文件发送，当一个客户端正在传输数据的时候，其它客户端是无法连接的，如果要能够允许多个客户端同时连接，必须在程序中引入线程，也就是 accept()返回的套接字应该交由新创建的线程处理，而主线程继续 accept()其它客户端。试对其进行改进，加入线程机制。

Windows 环境下创建线程可以用 \_beginthread() 或者 CreateThread() 函数，具体使用方法参见帮助文件“C Runtime Library Reference”和“Win32 Programmer's Reference”。

# 实验五 网络协议分析

## 一、实验目的

- 1) 能够正确安装配置网络协议分析仪软件 Wireshark。
- 2) 熟悉使用 Wireshark 分析网络协议的基本方法，加深对协议格式、协议层次和协议交互过程的理解。
- 3) 通过在以太网中分析 Web 应用中的报文交互，深入分析以太网的帧结构、IP 报文结构、TCP 报文结构、UDP 报文结构和 HTTP 协议报文结构；通过分析深入理解以太网帧格式、ARP 协议工作原理、IP 工作原理、TCP 工作原理、UDP 工作原理和 HTTP 协议工作原理。
- 4) 掌握无线局域网的基本组成和设备连接关系
- 5) 学习使用无线路由器配置无线局域网的基本技能

## 二、实验环境

- 1) 运行 Windows 2008 Server/XP/7 操作系统的 PC 一台。
- 2) 每台 PC 具有以太网卡一块，通过局域网与校园网相连。
- 3) Wireshark 程序（可以从 <http://www.wireshark.org/> 下载）和 WinPcap 程序（可以从 <http://www.winpcap.org/> 下载，如果 Wireshark 为版本 1.2.10 或更高版本，其中包含了 WinPcap 版本 4.1.2）。
- 4) 每台 PC 运行程序 CISCO 公司提供的 PacketTracer 版本 5.3.0。

## 三、实验内容

- 1) 安装配置网络协议分析仪软件 Wireshark。
- 2) 使用 Wireshark 分析网络协议。
- 3) 分析 Web 应用中的报文交互。
- 4) 构建虚拟 Internet 路由器及互联网 Web 服务器
- 5) 部署实验网络并对网络设备进行配置
- 6) 验证无线连接并对实验网络进行分析
- 7) 学习使用无线路由器配置无线局域网的基本技能

## 四、实验步骤

### 1. Wireshark 的使用

#### 1) 安装网络协议分析仪

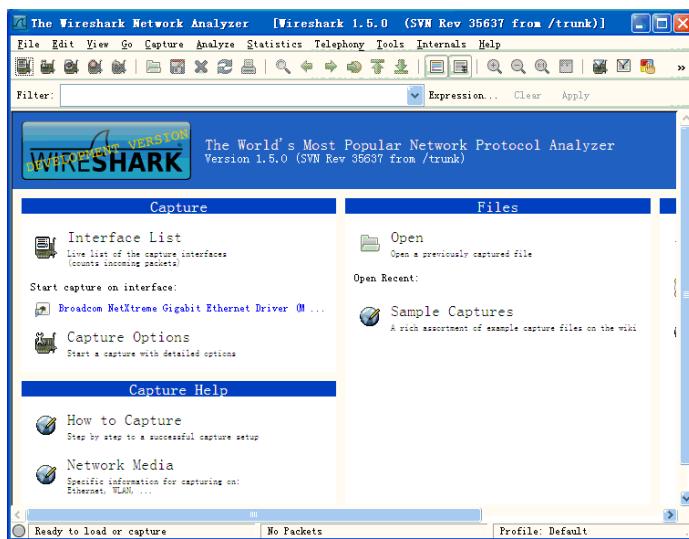


图 1 Wireshark 系统界面

安装 Wireshark 版本 1.2.10。双击 Wireshark 安装程序图标，进入安装过程。根据提示进行选择确认，可顺利安装系统。当提示“Install WinPcap 4.1.2”时，选择安装；此后进入安装 WinPcap 版本 4.1.2，并选择让 WinPcap 在系统启动时运行。此后，Wireshark 将能安装好并运行。

#### 2) Wireshark 基本操作

(1) 启动系统。点击“Wireshark”图标，将会出现如图 1 所示的系统界面。

其中“俘获(Capture)”和“分析(Analyze)”是 Wireshark 最重要的功能。

(2) 分组俘获。点击“Capture/Interface”菜单，出现图 2 的界面。

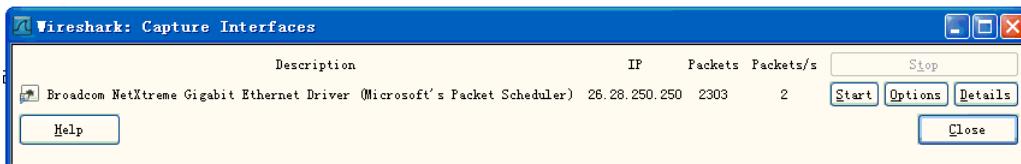


图 2 俘获/接口界面

如果该机具有多个接口卡，则需要指定你希望在哪个接口卡俘获分组。点击“Options”，则出现图 3 所示界面

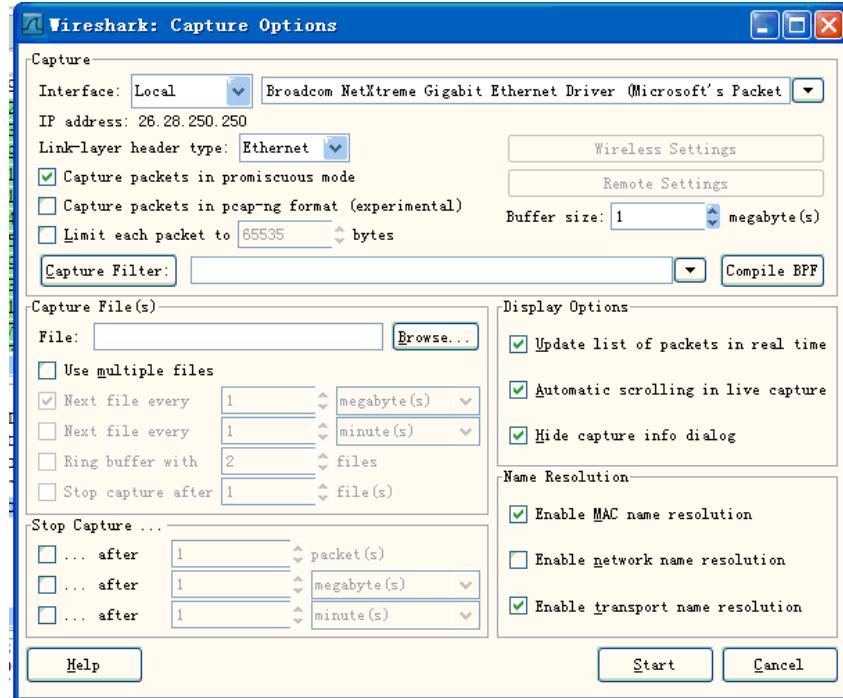


图 3 俘获/接口/选项界面

在该界面上方的下拉框中将列出本机发现的所有接口；选择一个你需要的接口；也能够在此改变俘获或显示分组的选项。

此后，在图 2 界面中，点击“Start(开始)”，Wireshark 开始在指定接口上俘获分组，并显示类似于图 4 的界面。

当需要时，可点击“Capture/Stop”停止俘获分组，并将俘获的分组信息存入踪迹(trace)文件中。当需要再次俘获分组时，可点击“Captuer/Start”开始俘获分组。

(3) 协议分析。系统能够对 Wireshark 俘获的或打开的踪迹文件中的分组信息(用 File/Open 功能)进行分析。如图 4 所示，在上部“俘获分组的列表”窗口中，有编号(No)、时间(Time)、源地址(Source)、目的地址(Destination)、协议(Protocol)、长度(Length)和信息(Info)等列，各列下方依次排列着俘获分组。中部“所选分组首部的细节信息”窗口给出选中帧的首部详细内容。下部“分组内容”窗口中是对应所选分组以十六进制数和 ASCII 形式的内容。

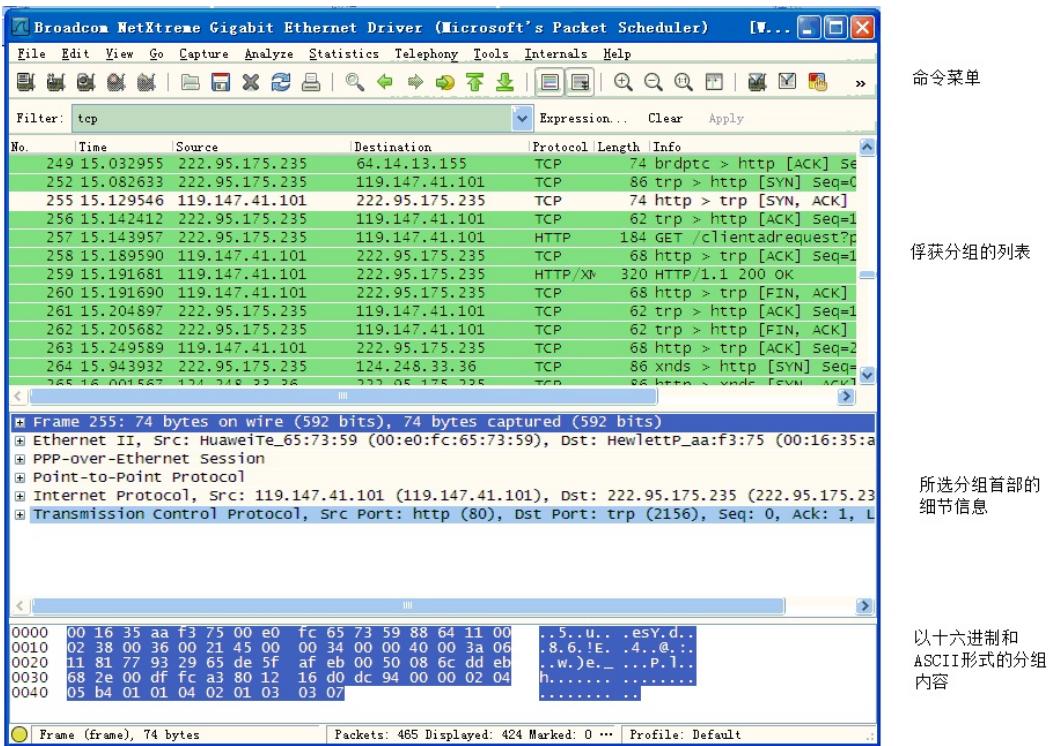


图 4 Wireshark 的俘获分组界面

若选择其中某个分组如第 255 号帧进行分析。从图 7 中的信息可见，该帧传输时间为俘获后的 15.129546 秒；从源 IP 地址 119.147.41.101，MAC 地址是 00.e0.fc.65.73.59(从中部分组首部信息窗口中可看到)；) 传输到目的地 IP 地址 222.95.175.235，MAC 地址是 00.16.35.aa.f3.75；分组长度 74 字节；是 TCP 协议携带的 HTTP 报文。

从分组首部信息窗口，我们可以看到各个层次协议及其对应的内容。例如，对应图 8 的例子，包括了 Ethernet II 帧及其对应数据链路层信息(参见图 5)。你可以对应 Ethernet II 帧协议来解释对应下方协议字段的内容。接下来，我们发现了 Ethernet II 协议上面还有 PPP-over-Ethernet 协议、Point-to-Point 协议、IP 协议和 TCP 协议，我们同样可以对照网络教科书中对应各种协议标准，分析解释相应字段的含义。

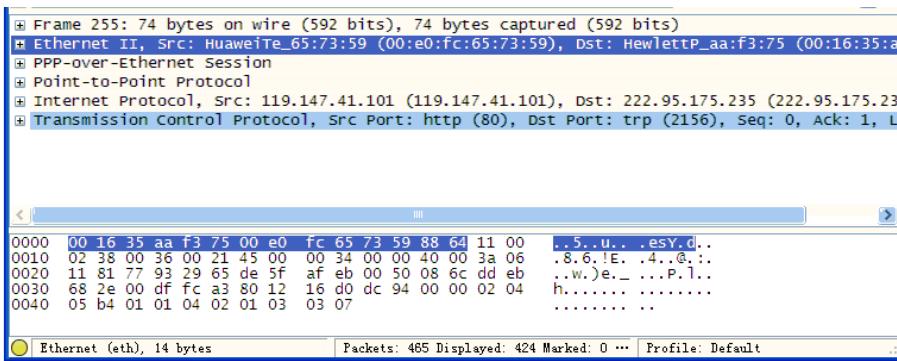


图 5 Ethernet 帧及其对应数据链路层信息

### 3) Wireshark 过滤功能

在利用 wireshark 捕获报文的过程中，可能会捕获到大量无关的报文，人工从这些报文中挑选出所需的报文是一项枯燥且单调的任务，wireshark 的过滤功能能够为我们解决这方面的问题。wireshark 的过滤功能分为两种：捕获过滤（capture filter）和显示过滤（display

filter)。

捕获过滤的作用是在捕获的过程中过滤不符合条件的报文，在选项界面中可输入相关的捕获过滤条件，如图 6 所示。



图 6 选项界面中的捕获过滤

捕获过滤的语法如下：[not] primitive [and|or [not] primitive ...]，其中常用的 primitive 包括[src|dst] host <host>，例如 src host 192.168.0.1 指定源地址为 192.168.0.1、host 192.168.0.1 指定 IP 地址为 192.168.0.1（源或目的）；[tcp|udp] [src|dst] port <port>，例如 tcp dst port 80 目的端口为 80 的 TCP 报文、tcp port 80 源端口或者目的端口是 80 的 TCP 报文；<protocol>，例如 ip 捕获 IP 报文、tcp 捕获 TCP 报文。

如果是初次使用捕获过滤，可点击选项界面中的“capture filter”按钮，弹出如图 7 所示对话框。在这个对话框里已经有不少预先设置的过滤条件，可在这些条件的基础上进行修改，获得符合要求的过滤条件。

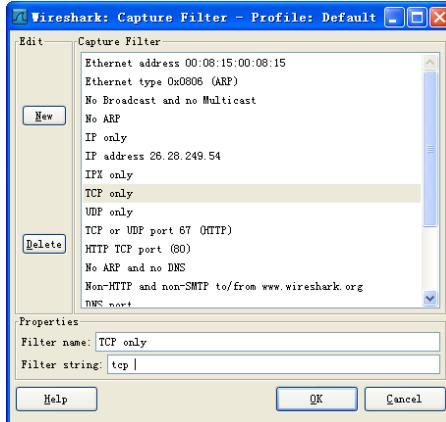


图 7 默认捕获过滤

显示过滤的作用是在已经捕获的报文中进行过滤，即此时只显示捕获报文中符合过滤条件的报文。如图 8 所示，在捕获分组界面，可输入过滤条件。注意，显示过滤的过滤条件的语法与捕获过滤的语法不一样。可以点击“Expression”按钮，选择合适的过滤条件，如图 9 所示。

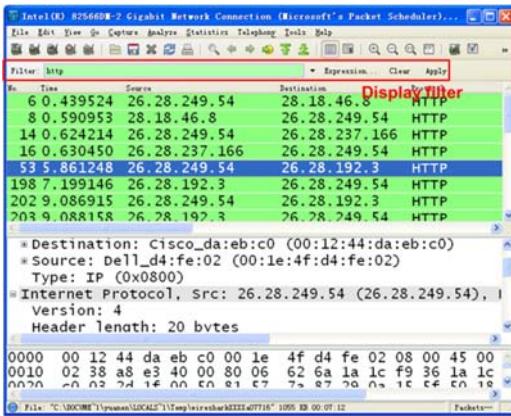


图 8 捕获分组界面中的显示过滤

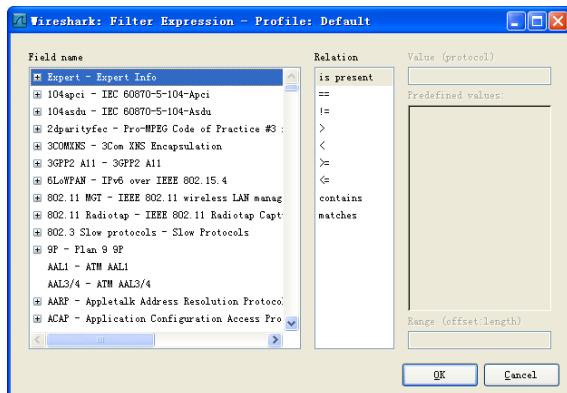


图 9 显示过滤输入对话框

## 2. 分析 Web 应用中的协议交互

### 1) 清空本机 ARP 表和 DNS 缓存

打开“命令提示符”界面，分别输入“arp -d”命令和“ipconfig /flushdns”清空本机 ARP 表和 DNS 缓存。

### 2) 启动 wireshark 开始捕获报文

启动 wireshark，在捕获过滤器中输入合适的条件，开始捕获报文。

### 3) 访问 web 网站

打开常用的浏览器，输入某个网站的 URL，访问网站。当网站页面展现后，关闭浏览器。然后停止 wireshark 捕获报文。

### 4) 以太网帧及 arp 协议分析

在 wireshark 显示过滤器中输入合适的过滤条件，过滤出 ARP 解析过程，回答下列问题：

- (1) 从 arp 请求报文来看，本机的 48 比特以太网 MAC 地址是什么？该报文的目的 MAC 地址是什么，代表什么意思？
- (2) 给出 2 字节以太类型字段的十六进制的值。它表示该以太帧包含了什么样的协议？
- (3) 此时通过 arp 命令查看 ARP 表，此时 ARP 表有什么变化？
- (4) 分析 ARP 协议执行的全过程，并画出或写出 ARP 协议报文的交互过程。

## 5) DNS 协议分析

在 wireshark 显示过滤器中输入合适的过滤条件，过滤出 DNS 解析过程，回答下列问题：

(1)从捕获的报文中分析 DNS 解析的过程，并画出报文交互过程。你所看到的过程与理论有什么不同？为什么？

(2)默认域名服务器的 IP 地址是多少？本机的 IP 地址是多少？

(3)IP 报文的协议字段的值是多少？该字段表明 DNS 的运输层采用的是什么协议？默认端口号是多少？

## 6) TCP 协议分析

在 wireshark 显示过滤器中输入合适的过滤条件，过滤出 TCP 报文。对于 TCP 的分析可借助于 wireshark 的统计工具，点击“Statistics/Flow Graph”，弹出如图所示选项对话框，点击确定，弹出 TCP 的流图。根据实际情况回答下列问题：

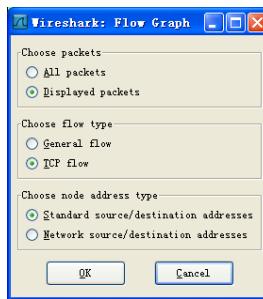


图 Flow Graph 选项对话框

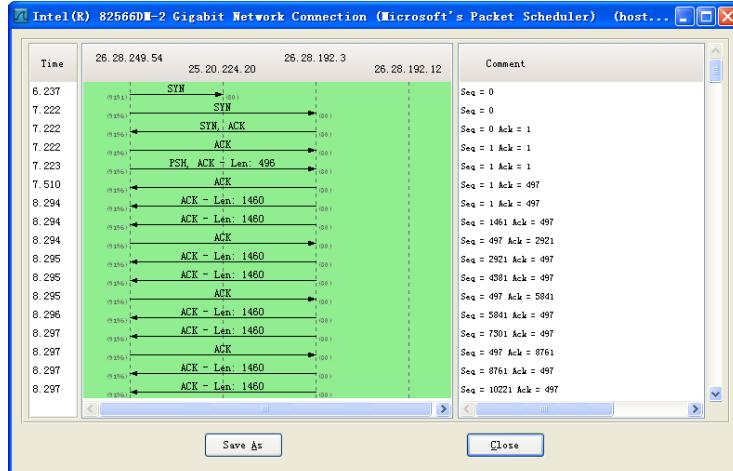


图 Flow Graph

(1) TCP 的连接建立在 HTTP 协议请求之前还是之后？

(2)分析 TCP 的连接建立过程，并画图说明。

(3)从连接建立的第一个报文看，web 服务器的 IP 地址是多少？该报文中以太网帧的目的 MAC 地址是多少？这个 MAC 地址是哪个接口的 MAC 地址？是否是 web 服务器接口的 MAC 地址？

(4)用于发起与服务器 TCP 连接的 TCP SYN 报文段的序号是多少？在该报文段中标识其为 SYN 报文段的标志是什么？

(5)服务器应答上述 TCP SYN 报文段的 SYN ACK 报文段的序号是什么？在该 SYN ACK 报文段的 ACK 应答字段中的值是多少？服务器是怎样确定这个 ACK 值的？在该报文

段中标识其作为 SYN ACK 报文段的标志是什么？

(6) 客户端对服务器的 ACK 报文应答的数据一般为多长？你如何确定接收方是对哪个报文段进行应答的？

(7) 观察 TCP SYN 报文段达到的时间以及 SYN ACK 报文段回复的时间。它们与后继请求和应答报文对之间的时间差一样吗？

(8) 客户端接收缓存通常的可用缓存的量是一样的吗？最小量是多少？出现了为抑制发送方而减少接收缓存空间的情况吗？

(9) 能够观察到 TCP 连接的关闭过程吗？如果观察到关闭过程请根据报文画图分析。

## 7) HTTP 协议分析

在 wireshark 显示过滤器中输入合适的过滤条件，过滤出 TCP 报文，也可以根据需要过滤出 HTTP 报文。根据实际情况回答下列问题：

(1) HTTP 协议的运输层采用什么协议？HTTP 协议的默认端口是多少？这个端口是服务器的端口还是客户端的？

(2) 分析利用 HTTP 协议请求基本 HTML 文件的过程，画图说明。

(3) 在请求包含多个被引用对象的网页的过程中，采用的是持续连接还是非持续连接？你是如何做出判断的？

(4) 在传输网页的过程中，TCP 连接建立了一次还是多次？你是如何做出判断的？

(5) 从报文分析看，HTTP 使用的方法是什么？HTTP 协议的版本是多少？

## 3、无线组网

通过 PacketTracer 搭建无线接入实验网络，网络拓扑如图 10。

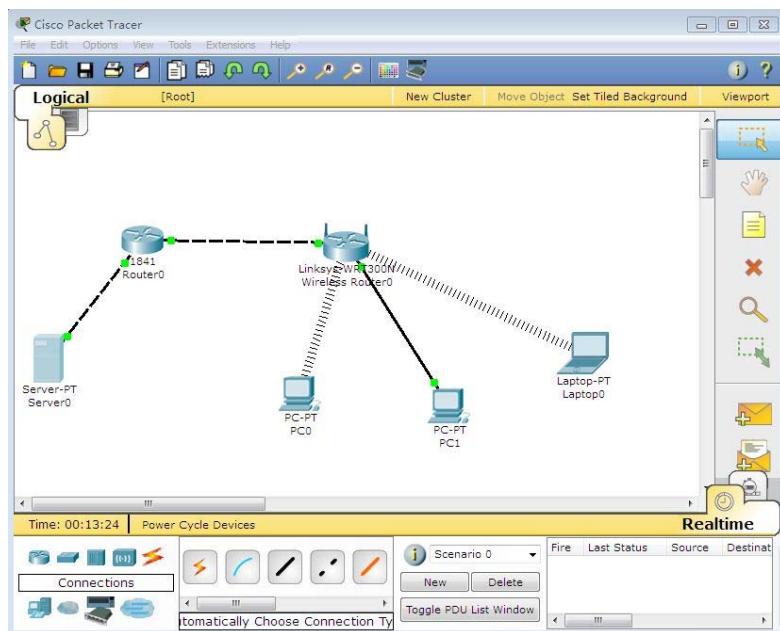


图 10

### 1) 构建虚拟 Internet 路由器及互联网 Web 服务器

在 PacketTracer 主界面中，添加 1841 路由器 Router0 和通用服务器 Server0。用自动选择端口方式连接 Router0 和 Server0。

配置 Router0，激活 FastEthernet0，并配置静态 IP 地址 12.0.0.254/24，如图 11 所示

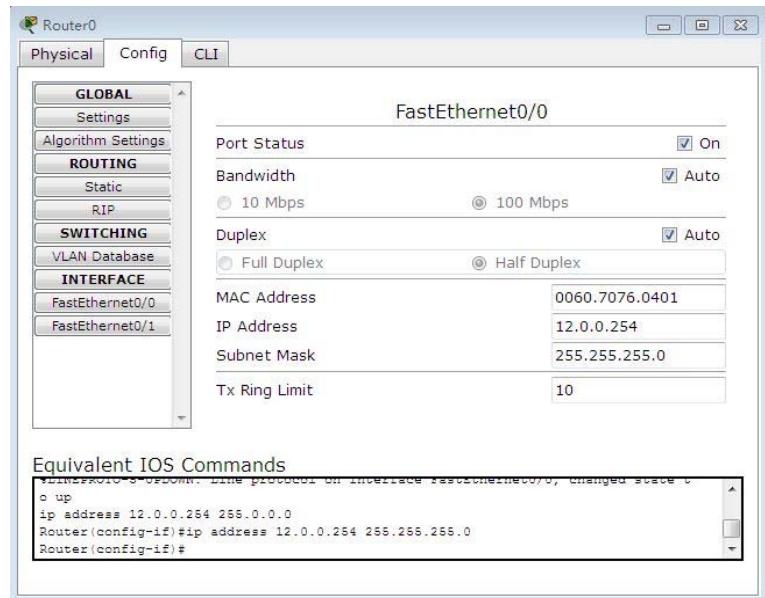


图 11

类似的，激活端口 FastEthernet1，并配置静态 IP 地址 11.0.0.254/24。

配置服务器 Server0。在 FastEthernet 配置页，设置静态 IP 地址 12.0.0.1/24。在全局设置页面(Global→Settings)配置默认网关为 12.0.0.254。

检查服务器的 HTTP 服务是否已开启（默认开启）。

此时可在服务器桌面标签下，打开命令行窗口并使用 ping 命令，测试服务器到路由器 Router0 的可达性。

## 2) 部署实验网络并对网络设备进行配置

在 PacketTracer 主界面中，添加型号为 Linksys-WRT300N 的无线路由器 Wireless Router0。此外，添加两台普通台式机 PC0、PC1 和普通笔记本电脑 Laptop0。在这里，我们的目标是，把位于本地网络的三台电脑(PC0、PC1 和 Laptop0)，通过无线路由器 Wireless Router0 联入 Internet。

首先给 PC0、Laptop0 换上无线网卡：关闭电源→拖出原网卡→拖入无线网卡→打开电源。如图 12 所示。

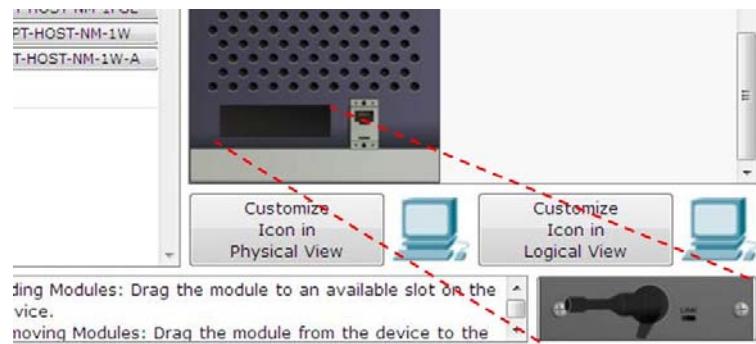


图 12

用自动选择端口方式连接 Router0 和 Wireless Router0、Wireless Router0 和 PC1。

配置无线路由器 Wireless Router0。在网络接口(Interface)配置中，首先配置互联网(Internet)接口。这里我们使用静态 IP 地址方式，需要配置默认网关(11.0.0.254)、本地互联网接口 IP 地址(11.0.0.1/24)，如图 13 所示。

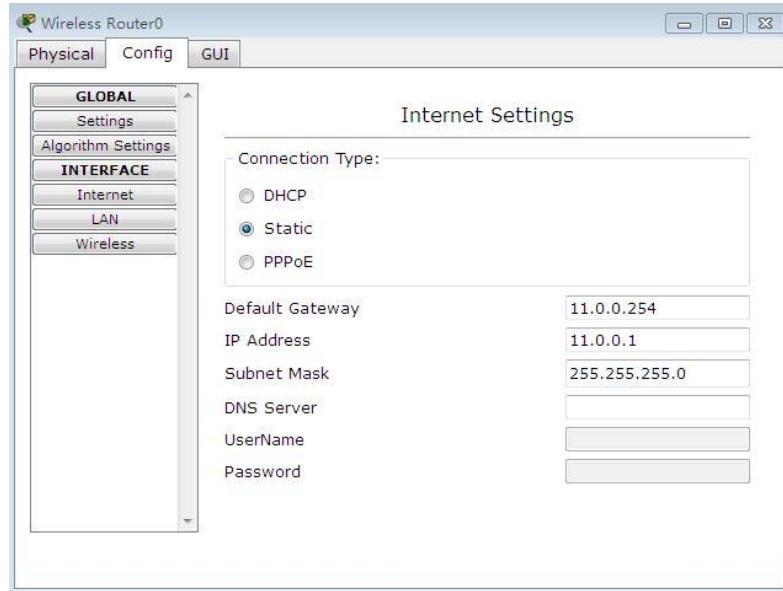


图 13

接下来配置无线路由器本地网络。无线路由器的 LAN，包括有线连接和无线连接两种方式，通过两种方式的任何一种接入的主机都在一个以太网下。检查 Interface 下的 LAN 配置，使用默认的设置：192.168.0.1/24。

下面配置无线路由器的无线接入端口。首先修改 SSID 为 wr。SSID 为无线路由器对外提供服务的名称。所有无线网卡要接入此无线接入点，都需要和 SSID 名称以及所指定的认证方式进行匹配并通过认证，方可接入。选择认证方式 WPA2-PSK，输入密码 12345678。如图 14 所示。

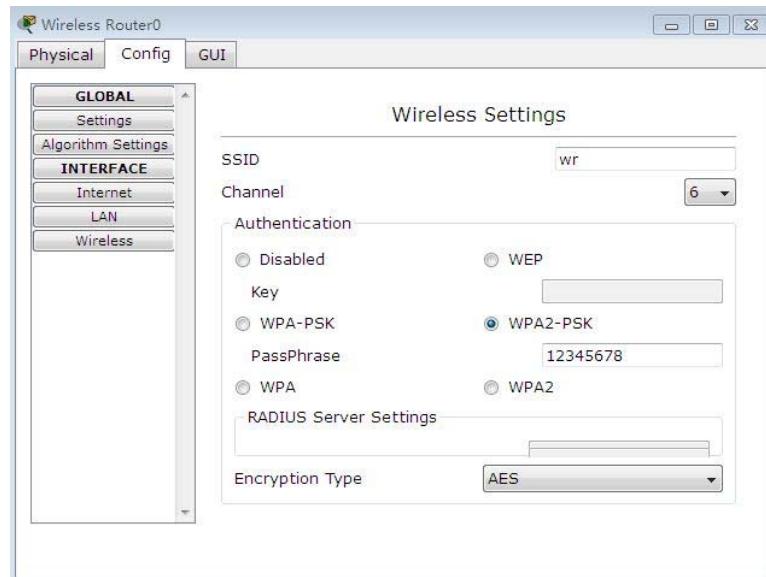


图 14

配置主机 PC0、Laptop0 无线网卡。输入需要连接的无线接入点 SSID(在这里是我们前面已经配置好的无线路由器服务 ID): wr，选择 WPA2-PSK 认证方式并输入密码 12345678。如图 15 所示。



图 15

### 3) 验证无线连接并对实验网络进行分析

在主机 PC0、PC1 和 Laptop 上使用浏览器访问服务器 Server0(12.0.0.1)的 Web 服务。应能看到如图 16 所示页面。

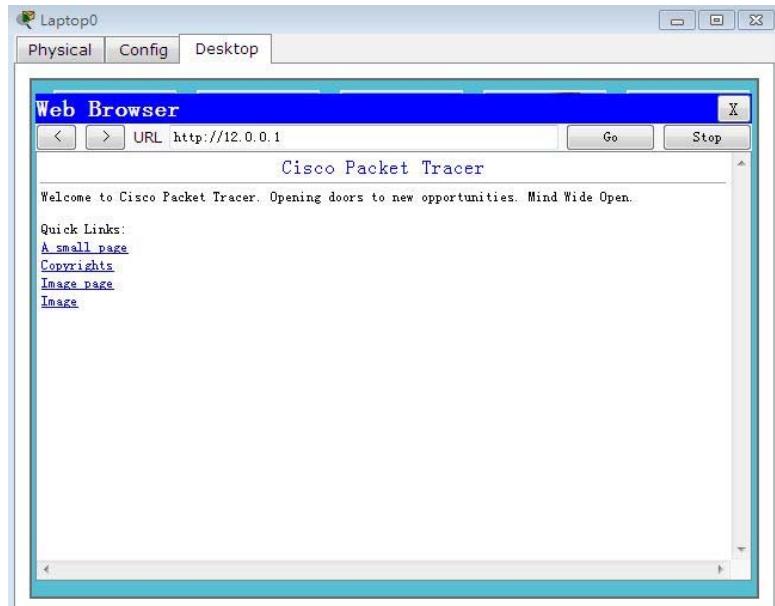


图 16

用虚拟主机命令行，查看 IP 地址，利用 ping 命令回答下面的问题：

- (1) PC0 与 Laptop0 是否能互相 ping 通？
- (2) Laptop0 是否能 ping 通 Server0？请分析分组传输的过程（经过哪些网络设备、分组的源 IP 地址、目的 IP 地址是什么）
- (3) Server0 能 ping 通 Laptop0 吗？为什么？

## 五、相关知识

### 1. Wireshark 相关知识

#### 1) Wireshark 简介

Wireshark 是一种具有图形用户界面的网络协议分析仪，可用于从实际运行的网络俘获分组或从以前保存的踪迹文件中交互地浏览、分析处理分组数据。Wireshark 是一个免费软件，因商标原因从 Ethereal 改名而得，是运行在 Windows、Linux/Unix 和 Mac 计算机上的免费分组嗅探器(packet sniffer)。Wireshark 能够读取 libpcap 俘获文件，也能够读取包括用 Tcpdump 俘获的文件，以及 snoop, atmsnoop, LanAlyzer, Sniffer(压缩和非压缩的), Microsoft Network Monitor, AIX 的 iptrace, NetXray, Sniffer Pro, Etherpeek, RADCOM 的 WAN/LAN analyzer, Lucent/Ascend router debug output, HP-UX 的 nettl, Cisco 的安全入侵检测系统以 IPLLog 格式输出的 pppd 日志文件。它自行决定文件类型，即使用 gzip 进行压缩也是如此。

Wireshark 对于实践中分析和调试网络协议特别是对初学者理解网络协议都是十分有用的工具。当你将在家中或在实验室中使用桌面计算机在各种情况下运行网络应用程序，你将可以用 Wireshark 观察网络协议与在因特网别处执行的协议实体交互和交换报文。因此，Wireshark 使你的计算机成为真实动态实验的有机组成部分，通过动手实验来观察网络的奥秘，进而深入理解和学习。你的网络概念和实验技能能够得到极大的深化：观察网络协议的动作和经常摆弄它们，即观察两个协议实体之间交换的报文序列，钻研协议运行的细节，使协议执行某些动作，观察这些动作及其后果。

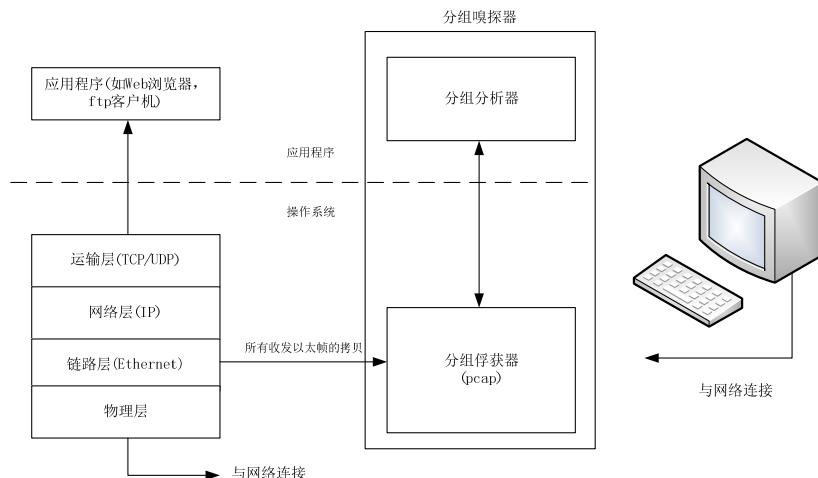


图 17 分组嗅探器结构

#### 2) Wireshark 的结构

作为分组嗅探器，Wireshark 俘获从计算机发送接收的报文，通常也能够存储和显示这些俘获的报文中各个协议字段的内容。分组嗅探器自身是被动的，观测着运行在计算机应用程序和协议所发送及接收的报文，但自身不发送分组。类似地，接收到的分组决不会显式地以分组嗅探器为目的地址，它们仅是在机器上运行的应用程序和协议收发分组的拷贝。图 10 显示了分组嗅探器的结构。图中的计算机通常运行着应用程序及其协议，显示在图中方框内的分组嗅探器是计算机中附加的一个普通软件，它由两部分组成。分组俘获器接收计算机收发的每个链路层帧的拷贝。我们知道较高层协议如 HTTP、FTP、TCP、UDP、DNS 或 IP 之间交换的报文全都逐个封装在链路层帧中，并在物理介质如以太电缆上传输。如果图中的物

理介质是以太网，所有高层协议则将封装在以太帧中。俘获所有链路层帧从而为你提供在计算机中执行的所有应用程序和协议收发的报文。

嗅探器的第二部分是分组分析器，它显示协议报文的所有字段的内容。为了实现该功能，分组分析器必须要能理解协议交换的所有报文结构。例如，我们想要显示图中由 FTP 交换报文的各个字段，则该分组分析器理解以太帧格式，这样才能识别以太帧中的 IP 数据报，才能从数据报中提取 TCP 段。只有理解了 TCP 段结构，才能提取包含在 TCP 段中的 FTP 报文。最终，只有理解了 FTP 协议，才能正确显示“USER”、“PASS”或“LIST”等命令。

## 2. 相关命令简介

### 1) arp 命令简介

arp 命令主要用于查看和修改 ARP 表，该命令有多种可选参数，配合不同的参数实现对 ARP 表的操作。

(1) arp 命令的帮助。arp 命令提供了帮助选项，通过帮助可对 arp 命令有一个全面的了解。打开“命令提示符”界面，输入“arp -?”，结果如图 18 所示。

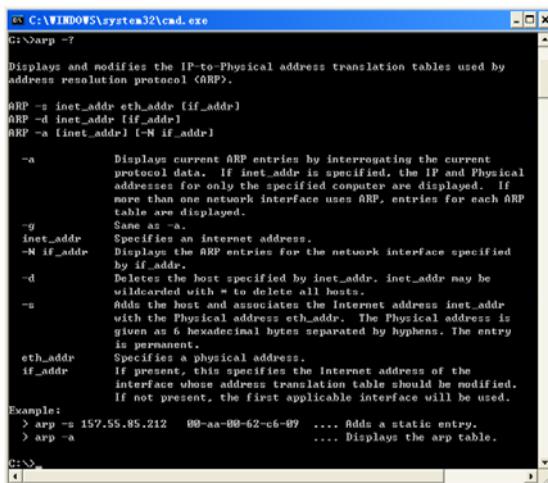


图 18 ARP 命令帮助

图中描述了 arp 命令的各种参数和使用方法，可根据需求使用具体的参数。

(2) 查看 ARP 表的内容。在“命令提示符”界面中输入“arp -a”查看本机 ARP 表的内容，结果如图 19 所示。

Interface: 26.28.249.54 --- 0x2			
Internet Address	Physical Address	Type	
26.28.249.37	78-e3-b5-a1-78-ef	dynamic	
26.28.249.252	00-e0-4c-08-31-44	dynamic	
26.28.249.254	00-12-44-da-eb-c0	dynamic	

图 19 查看 ARP 表

在 ARP 表中，主机的 IP 地址与 MAC 地址一一对应。主机之间进行通信前，首先需要查找 ARP 表，如果有对应的表项，则获得 IP 地址对应的 MAC 地址。“Type”栏下的“dynamic”字段表明该表项是动态更新的。如果 20 分钟内表项中的主机没有访问网络，表项就会被清空。

(3) 清除 ARP 表的内容。可以通过手工的方式清除 ARP 表中的内容。在“命令提示符”界面中输入“arp -d”清除本机 ARP 表的内容，结果如图 20 所示。

```
C:\>arp -d
C:\>arp -a
No ARP Entries Found
C:\>
```

图 20 清除 ARP 表

## 2) ipconfig 命令简介

ipconfig 命令可用来查看本机当前网络的配置状态，该命令有多种可选参数，通过不同的参数可查看本机所有当前 TCP/IP 网络配置值、刷新 DHCP 和 DNS 设置。

(1) ipconfig 命令的帮助。ipconfig 命令提供了帮助选项，通过帮助可对 ipconfig 命令有一个全面的了解。打开“命令提示符”界面，输入“ipconfig /?”，结果如图 21 所示。

```
C:\>ipconfig /?

USAGE:
    ipconfig [/? | /all | /renew [adapter] | /release [adapter] |
              /flushdns | /displaydns | /registerdns |
              /showclassid adapter |
              /setclassid adapter [classid] ]

where
    adapter      Connection name
                <wildcard characters * and ? allowed, see examples>

Options:
    /?
        Display this help message
    /all
        Display full configuration information.
    /release
        Release the IP address for the specified adapter.
    /renew
        Renew the IP address for the specified adapter.
    /flushdns
        Purges the DNS Resolver cache.
    /registerdns
        Refreshes all DHCP leases and re-registers DNS names.
    /displaydns
        Display the contents of the DNS Resolver Cache.
    /showclassid
        Displays all the dhcp class IDs allowed for adapter.
    /setclassid
        Modifies the dhcp class id.
```

图 21 ipconfig 命令帮助

(2) 利用 ipconfig 命令查看 TCP/IP 网络配置值。在“命令提示符”界面中输入“ipconfig /all”查看本机 TCP/IP 网络配置值，结果如图 22 所示。

```
C:\>ipconfig /all

Windows IP Configuration

Host Name . . . . . : ym
Primary Dns Suffix . . . . . :
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter 本地连接:

    Connection-specific DNS Suffix . :
    Description . . . . . : Intel(R) 82566DM-2 Gigabit Network Connection
    Physical Address . . . . . : 00-1E-4F-D4-FE-02
    DHCP Enabled. . . . . : No
    IP Address . . . . . : 26.28.249.34
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 26.28.249.54
    DNS Servers . . . . . : 26.28.192.10
```

图 22 查看 TCP/IP 网络配置值

(3) 利用 ipconfig 命令操作 DNS 缓存。在“命令提示符”界面中输入“ipconfig /displaydns”查看本机 DNS 缓存，结果如图 23 所示。

```
C:\>ipconfig /displaydns

Windows IP Configuration

www.lgdx.mtn
-----
Record Name . . . . . : www.lgdx.mtn
Record Type . . . . . : A
Time To Live . . . . . : 7765
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . : 26.28.192.10

dns.lgdx.mtn
-----
Record Name . . . . . : dns.lgdx.mtn
Record Type . . . . . : A
Time To Live . . . . . : 7765
Data Length . . . . . : 4
Section . . . . . : Additional
A (Host) Record . . . . : 26.28.192.10
```

图 23 查看本机 DNS 缓存

通过上述命令可查看本机当前的 DNS 缓存，进一步还可对本机 DNS 缓存进行操作，在

“命令提示符”界面中输入“ipconfig /flushdns”清除本机 DNS 缓存，结果如图 24 所示。

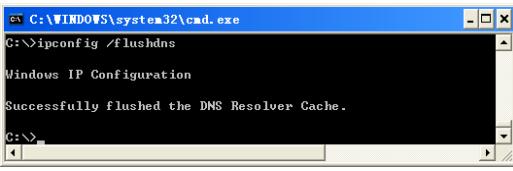


图 24 清除本机 DNS 缓存

无线局域网是一种重要的网络接入技术，目前已经得到广泛的应用。众多无线局域网协议中，最重要的标准是 IEEE 802.11，我们常说的无线局域网 WiFi 就是用的这套标准。802.11 中规定的无线接入点(AP)是用来把无线主机接入网络的基础设施。无线终端通过无线的方式连接到 AP，而 AP 通常以有线的方式接入本地网络或互联网。实验中用到的无线路由器，是无线 AP 和 NAT 网关“二合一”的设备。

## 六、注意事项

- 1) 安装 Wireshark 网络协议分析仪前应安装 WinPcap 网络监测驱动程序。
- 2) 俘获分组前应注意选择正确的网络接口。
- 3) 协议分组的俘获结果可保存在指定的文件中。
- 4) Wireshark 网络协议分析仪还具有丰富的其他功能，学生可参阅随软件的“Wireshark 帮助”文档。
- 5) 在实际的无线路由器配置中，可能还需要修改 NAT 相关配置、激活 DNS 配置、并根据网络运营商提供的 DNS 地址配置路由器的 DNS 服务器地址。

# 实验六 网络服务配置

## 一、实验目的

1. 掌握在 Windows Server 2003/2008 上配置域名系统 DNS 的方法，深入理解 DNS 系统的工作过程；
2. 掌握使用 MDaemon 邮件服务器软件配置并管理邮件服务器的方法，通过实验了解邮件协议 SMTP 及 POP3 的原理及功能。
3. 掌握在 Packet Tracer 模拟器上配置 DHCP (动态主机配置协议) 服务器的技能，深入理解 DHCP 服务的工作过程。

## 二、实验环境

该实验是在局域网环境下进行的，所需硬件及软件设备如下：

1. 运行 Windows Server 2003/2008 操作系统的 PC 机一台，运行 Windows XP 操作系统的 PC 机一台；
2. 每台 PC 机具有一块以太网卡，通过双绞线和局域网相连；
3. MDaemon 邮件服务器软件安装包一个，Foxmail 邮件客户端软件安装包一个。
4. Packer Tracer 模拟器软件安装包一个；

## 三、实验内容

1. 配置 DNS 服务器；
2. 测试 DNS 服务器；
3. 安装 MDaemon 邮件服务器软件；
4. 配置 MDaemon 邮件服务器；
5. 测试 MDaemon 邮件服务器；
6. DHCP 服务器的安装与配置；
7. DHCP 服务器的测试。

## 四、实验步骤

1. 配置 DNS 服务器  
1) 安装 DNS 服务器

安装 Windows Server 2003/2008 操作系统的主机都能够充当 DNS 服务器（一般由域控制器担当这一角色）。如果管理员程序组或计算机管理器中没有 DNS 项，则需要添加 DNS 服务。选择“控制面板/添加删除 Windows 组件/网络服务/详细信息/域名系统(DNS)”，安装

DNS 系统。重新引导 Windows Server2003/2008 之后，DNS 服务开始生效。

**注意：**前提是该主机已经安装了 TCP/IP 协议。

## 2) 启动 DNS 管理器

选择“开始/管理工具/DNS”，启动 DNS 管理器窗口，如图 1) 图 1 所示。

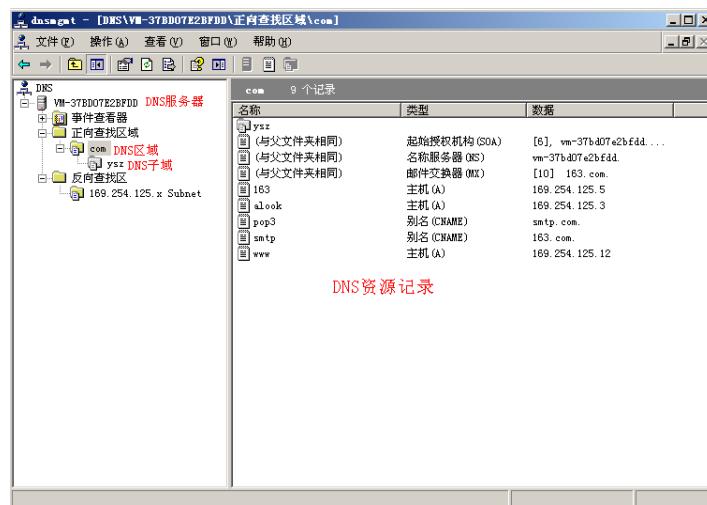


图1 DNS 管理器窗口

## 3) 创建查找区域

DNS 管理器的 DNS 服务器节点下的“正向查找区域”和“反向查找区域”两个子节点是 DNS 服务管理的基本单位。其中正向查找区域用于正向查找，它将域名解析为 IP 地址。一台 DNS 服务器上至少要有一个正向查找区域才能工作。反向查找区域用于反向查找，它将 IP 地址解析为域名。

**创建正向查找区域：**在 DNS 管理窗口中选择需要配置的服务器，默认为本地计算机。在 DNS 管理器中展开 DNS 服务器图标，选择“正向查找区域”，右键菜单选择“新建区域”，打开 DNS 新建区域向导，如图 2 所示。

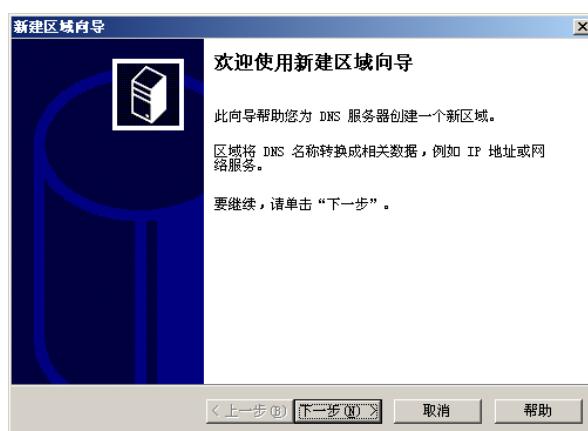


图2 DNS 区域创建向导

在“下一步”的区域类型页面，如图 3 所示，选择新建正向查找区域的类型：主要区

域、辅助区域或存根区域。主要区域是一个新区域的标准主拷贝，创建区域的主机负责维护主要区域。辅助区域是一个已存在区域的副本，辅助区域本身是只读的，它从主要区域拷贝数据。辅助区域的用途是产生冗余，一方面减少了主控服务器的流量负载，另一方面降低了主控区域关机造成的时间损失。存根区域是一个区域副本，只包含标识该区域的权威域名系统服务器所需的那些资源记录，用于使主持父区域的 DNS 服务器知道其子区域的权威 DNS 服务器，从而保持 DNS 名称解析效率。集成的 Active Directory(活动目录)区域是一个新区域的主拷贝，用 Active Directory 存储和复制区域文件。在本实验中选择“主要区域”。

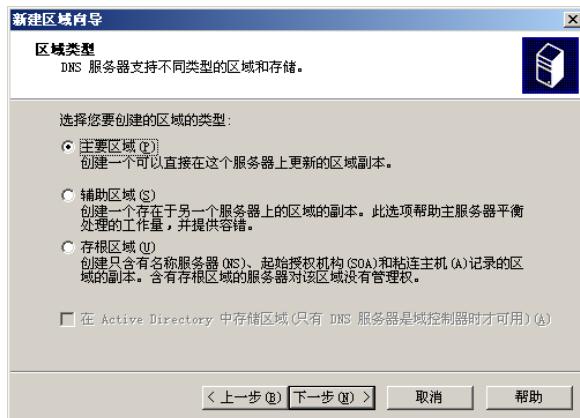


图3 区域类型选择

在区域名称页面指定区域名称，例如 com，如图 4 所示。



图4 输入区域名称

在“下一步”的区域文件页面，根据前面所选区域类型的不同，此时配置的信息亦不相同。如果选择创建主要区域，则在此指定区域映射文件名称，或者指定一个现有文件作为区域文件，如图 5 所示。如果选择创建辅助区域，则在此指定辅助区域所对应主区域的 DNS 服务器，如图 6 所示，在“IP 地址”栏中添入主控 DNS 服务器地址，单击“添加”加入列表，DNS 将按照列表中的主控服务器顺序逐一联系它们；单击“上移”或者“下移”可以更改主控服务器在列表中的顺序。本实验因选择“主要区域”，会自动创建新区域文件。

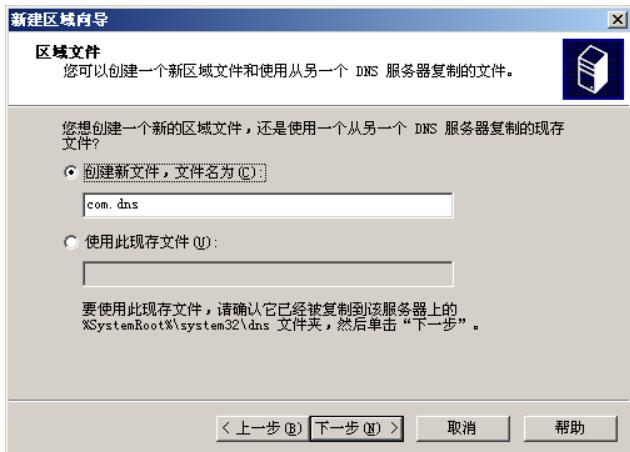


图5 创建主要区域区域文件

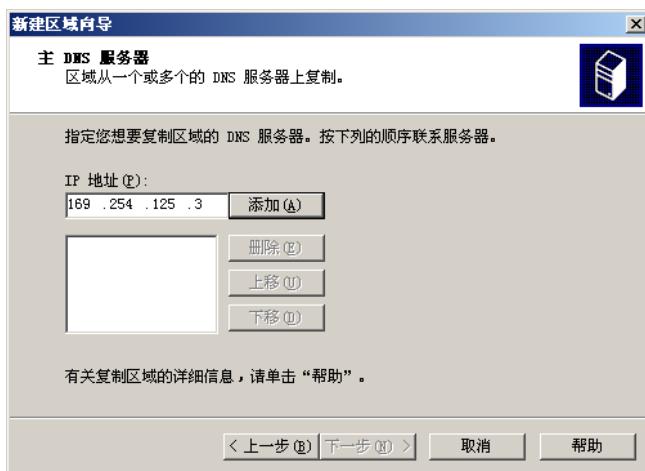


图6 输入辅助区域的主服务器

**创建反向查找区域:** 在 DNS 管理器中展开 DNS 服务器图标，选择“反向查找区域”，右键菜单选择“创建区域”，指定新建反向查找区域的类型，本实验中选择“主要区域”。输入反向查找 DNS 的 IP 地址，例如 169.254.125.，如图 7 所示，系统将会自动创建反向查找区域的新文件名，如图 8 所示。

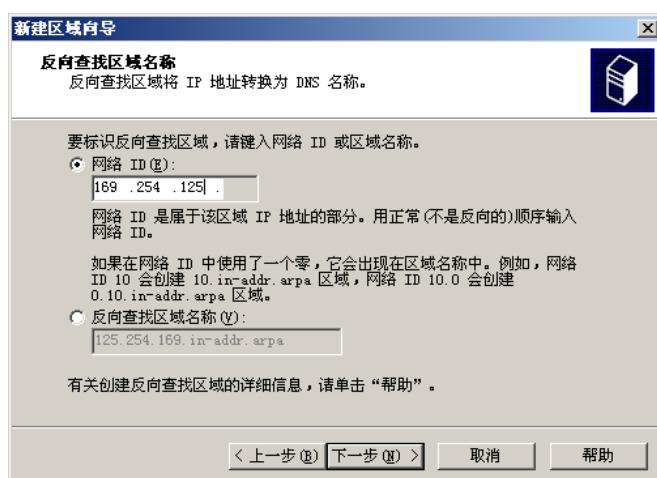


图7 输入反向查找区域的网络 ID



图8 生成反向查找区域新文件

#### 4) 添加资源记录

创建区域之后还需要向区域中添加资源记录才能使 DNS 服务器工作。例如 Web 站点域名 163.com 映射为站点的 IP 地址 169.254.125.5，这就是一条资源记录。

为正向查找的新区域添加要解析的主机名的方法有：

- ◆ 展开 DNS 管理器控制树中的相应节点，右击欲创建主机资源记录的正向查找的新区域(可以是标准主要区域或者其子区域)，如上所示 com 域，选择“新建主机”。
- ◆ 输入主机名称(主机名即可，不必输入全域名 FQDN)，指定其 IP 地址，单击“添加主机”。例如全名为 163.com，只输入 163 即可，IP 地址为 169.254.125.5，如图 9 所示。

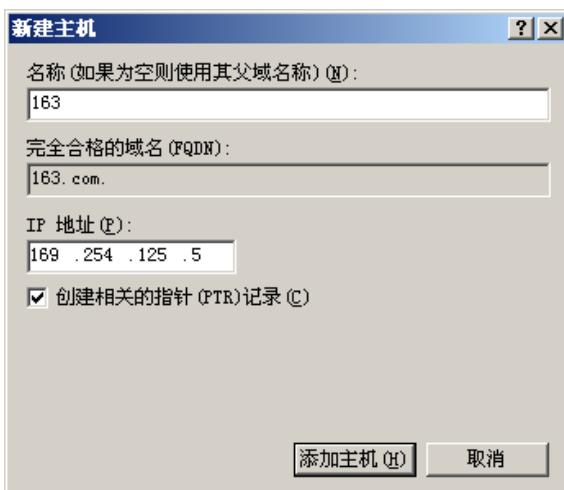


图9 新建主机

为反向查找的新区域新建指针的方法有：

- ◆ 展开 DNS 管理器控制树中的相应节点，右键反向查找的新区域，选择“新建指针”。
- ◆ 输入要创建的指针的主机 IP 地址和对应主机名，单击“确定”。至此，已经完成了 DNS 的配置，该域名服务器可对域名为 163.com 和 IP 地址为 169.254.125.5 的主机进行正反方向的解析，这样便使 163.com 和 169.254.125.5 对应起来了。

## 2. 测试 DNS 服务器

为测试配置的 DNS 服务器是否正常，可使用另一台机器作为 DNS 客户机。在配置 IP 地址时，将“首选 DNS 服务器”的 IP 地址设置为你刚配置好的 DNS 服务器的主机 IP 地址。即选择“网上邻居/属性/本地连接/属性/Internet 协议(TCP/IP)”，填写 DNS 服务器的主机 IP 地址及如图 10 所示 DNS 服务器的 IP 地址是 169.254.125.3。

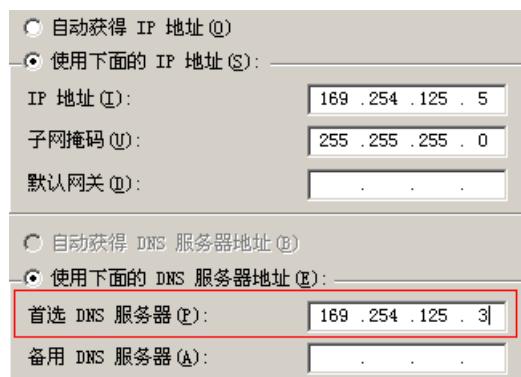


图10 填写 DNS 服务器的主机 IP

### 1) 用 ping 命令测试

正向解析操作：在命令提示符下执行“ping 163.com”，163.com 即为要解析的域名，经 DNS 服务器解析出它的 IP 地址如图 11 所示。

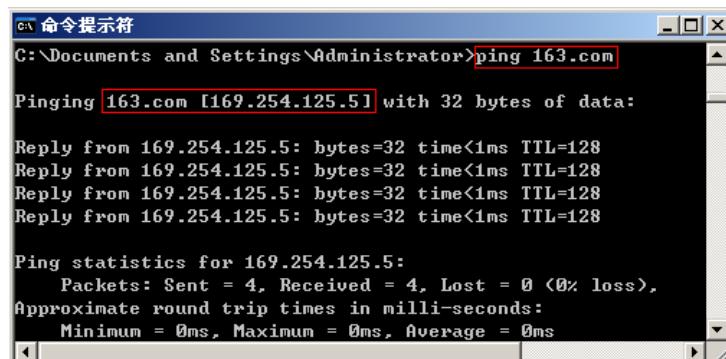


图11 Ping 命令正向解析出 IP 地址

反向解析操作：在命令提示符下执行“ping -a 10.65.19.162”，经 DNS 服务器解析出的它的域名如图 12 所示。

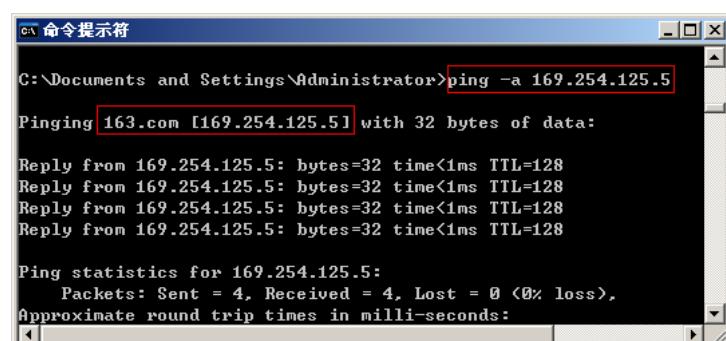


图12 Ping 命令反向解析出域名

2) 用 Nslookup 工具测试

用 Nslookup 工具测试正方向的解析，即从域名解析出 IP 地址，如图 13 所示。

```
C:\命令提示符 - nslookup
C:\Documents and Settings\Administrator>nslookup
Default Server: alook.com
Address: 169.254.125.3

> 163.com
Server: alook.com
Address: 169.254.125.3

Name: 163.com
Address: 169.254.125.5

> smtp.com
Server: alook.com
Address: 169.254.125.3

Name: 163.com
Address: 169.254.125.5
Aliases: smtp.com

> pop3.com
Server: alook.com
Address: 169.254.125.3

Name: 163.com
Address: 169.254.125.5
Aliases: pop3.com, smtp.com

>
```

图13 Nslookup 工具正向解析出 IP 地址

利用 Nslookup 工具测试反方向的解析，即从 IP 地址解析出域名，如图 14 所示。

```
C:\命令提示符 - nslookup
C:\Documents and Settings\Administrator>nslookup
Default Server: alook.com
Address: 169.254.125.3

> 169.254.125.5
Server: alook.com
Address: 169.254.125.3

Name: 163.com
Address: 169.254.125.5

> 169.254.125.3
Server: alook.com
Address: 169.254.125.3

Name: alook.com
Address: 169.254.125.3

>
```

图14 Nslookup 工具反向解析出域名

### 3. 添加邮件服务器所需资源记录

使用上述 DNS 服务器来获取邮件服务器的 DNS 服务，则需要在 DNS 服务器的对应区域中添加邮件服务需要的资源记录。该资源记录包括主机、别名、MX 邮件交换器。

- ◆ 添加主机记录

若以 169.254.125.5 的主机为邮件服务器，域名设置为 163.com，则需要将此记录信息添加到 DNS 域名服务器资源记录表中，如上图 1 所示。

- ◆ 添加别名记录

若 SMTP 和 POP3 协议也在该邮件服务器上，可为其设置别名。

展开 DNS 管理器控制树中的相应节点，右击欲创建别名资源记录的正向查找的新区域（可以是标准主要区域或者其子区域），如上所示 com 域，选择“新建别名”。输入别名（不必输入全域名 FQDN），指选择目标主机。例如全名为 smtp.com，只输入别名 smtp 即可，目标主机全名为 163.com，如图 15 所示。

依次建立 SMTP 和 POP3 两个别名记录。



图15 新建别名

- ◆ 添加邮件交换器（MX）记录

展开 DNS 管理器控制树中的相应节点，右击欲创建别名资源记录的正向查找的新区域（可以是标准主要区域或者其子区域），如上所示 com 域，选择“新建邮件交换器（MX）”。主机或子域默认为空，邮件服务器优先级使用默认值，邮件服务器选择为 SMTP 服务所在主机，如图 16 所示。



图16 新建邮件交换器 (MX)

#### 4. 安装 MDaemon 邮件服务器软件

MDaemon 软件有多个版本，在此以 MDaemon7 中文版为例进行实验。

在 MDaemon 软件安装包中选择 `md723_sc.exe` 双击，打开软件安装向导，按照向导提示进行安装。默认情况下，注册码为空，MDaemon Server 可以试用 30 天。若要使用更长时间，需要使用安装包中的 `keygen723.exe` 软件生成注册码，生成时可选择该注册码的使用时长，1 年或者 2 年。

#### 5. 配置 MDaemon 邮件服务器

MDaemon 软件安装结束后，默认弹出 MDaemon Server 域设置窗口。输入上述 DNS 域名服务器中设置的域名，如 `163.com` 即 E-mail 地址的@符号的右边部分，如图 17 所示。



图17 MDaemon Server 域设置

点击“下一步”按钮进入管理员帐户设置窗口，在这个窗口设置管理员邮箱及密码，如图 18 所示。



图18 管理员帐户设置

点击“下一步”按钮进入在 DNS 地址设置窗口，在此可设置该邮件服务器使用的 DNS 服务器。可选择 Windows 的 DNS 设置，如图 10 所示的 DNS 服务器设置，也可以选择其他 DNS 服务器的 IP 地址，如图 19 所示，包括主 DNS IP 地址和备 DNS IP 地址。



图19 指定 DNS 服务地址

点击“下一步”按钮进入操作模式设置页面，如图 20 所示。在操作模式设置页面，可选择使用“简易”或“高级”两种方式。



图20 设置操作模式

点击“下一步”按钮可将 MDaemon 设置为系统服务启动，如图 21 所示。



图21 设置 MDaemon 设置为系统服务启动

若将 MDaemon 设置为系统服务启动，则当服务器启动时就自动启动，显示启动页面，如图 22 所示。

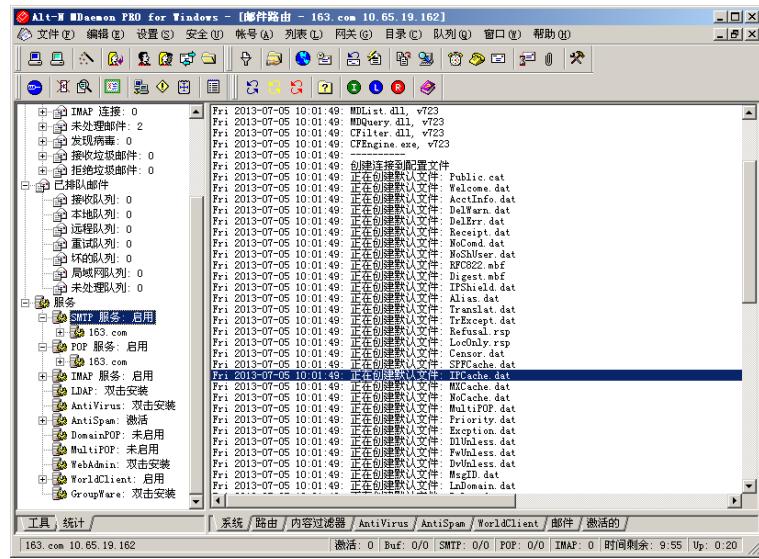


图22 MDaemon 服务器启动页面

在 MDaemon 启动页面中左侧会看到邮件服务 SMTP 及 POP3 的状态为启用，那么该服务器即启动完成，既可以发送邮件也可以接收邮件。

**注：**有时会发现 SMTP 服务启动状态为失败。此时需要检查 Server 2003 自带的 IIS 中的 SMTP 是否启用。两者默认都是使用 25 端口号，所以会产生冲突。

## 6. 测试 MDaemon 邮件服务器

### 1) 新建邮件帐户

在邮件服务页面的“帐号”菜单中新建帐号，页面如图 23 所示。



图23 新建帐户

初始建立用户帐户时，用户密码要求为强密码，要求如图 24 所示。

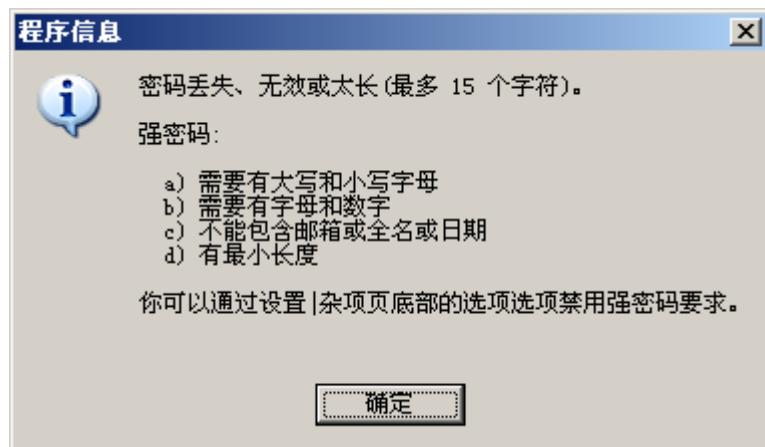


图24 强密码要求

若不想使用强密码，则可以选择设置/其他选项中的“杂项”，将“需要强密码”之前的勾选去掉。其他的设置也可以在该页面中进行。

## 2) 配置邮件客户端

邮件客户端软件可以选择 Windows 自带的 Outlook Express，也可以选择其他常用的邮件客户端软件，如 FoxMail 等。

在开始/程序中选择 Outlook Express，打开新建帐户连接向导，添加帐户显示名如图 25 所示。



图25 添加帐户显示名

点击“下一步”按钮，进入电子邮件地址填写页面，添加邮件地址，如图 26 所示。



图26 添加邮件地址

点击“下一步”按钮，进入邮件服务器地址填写页面，如图 27 所示，选择邮件接收服务器的协议类型：pop3、IMAP、HTTP，默认为 POP3 服务器。本实验接收邮件服务器是 POP3 服务器。

输入 POP3 服务器和 SMTP 服务器地址，可以输入域名，如 pop3.com 和 smtp.com，也可以输入服务器所在主机的 IP 地址，如 169.254.125.5，本实验中 POP3 服务和 SMTP 服务器地址一样。



图27 配置邮件服务器地址

点击“下一步”按钮，进入邮件帐户输入页面，如图 28 所示，输入帐户名（邮箱名中 @前面部分）、密码。若希望记住密码，则可在“记住密码”前勾选。



图28 添加邮件帐户

点击“下一步”按钮，完成在邮件客户端中添加帐户信息的设置。

若需要在邮件客户端中再添加用户，则可以在 Outlook Express 客户端选择工具/帐户/添加邮件，按照添加用户向导进行。

若需要在邮件客户端中修改用户帐户信息，则可在 Outlook Express 客户端选择工具/帐户/邮件,选中要修改的邮件，点击右侧的“属性”，进入属性页面修改。

### 3) 测试邮件服务器

使用步骤 2) 中添加的用户帐户测试邮件服务器。在 Outlook Express 邮件客户端工具栏中选择“创建邮件”，则弹出创建新邮件页面，如图 29 所示

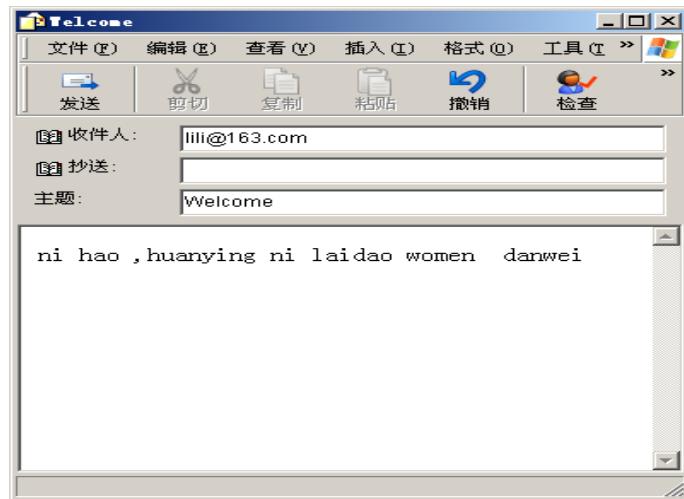


图29 创建新邮件

输入收件人邮件地址、主题信息以及邮件内容，点击工具栏中的“发送”按钮。

再回到 Outlook Express 邮件客户端主页面，在工具栏中点击“发送/接收”按钮，则可以看到收件箱中有一封未读邮件，如图 30 所示。

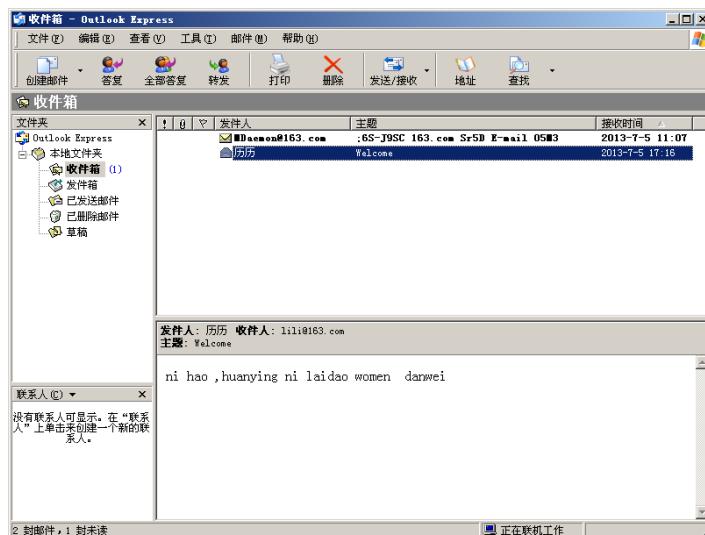


图30 接收邮件

双击，打开邮件信息，如图 31 所示。



图31 查看邮件

点击工具栏中的“答复”按钮，可在此页面上回复邮件，如图 32 所示。



图32 回复邮件

## 7.安装 Packet Tracer 模拟器软件

在 Packet Tracer 5.1 软件安装包中选择 `PacketTracer51_setup.exe` 双击，进入软件安装向导。按照软件安装向导进行安装。安装完成后打开 `PacketTracer` 窗口页面，如 33 所示。

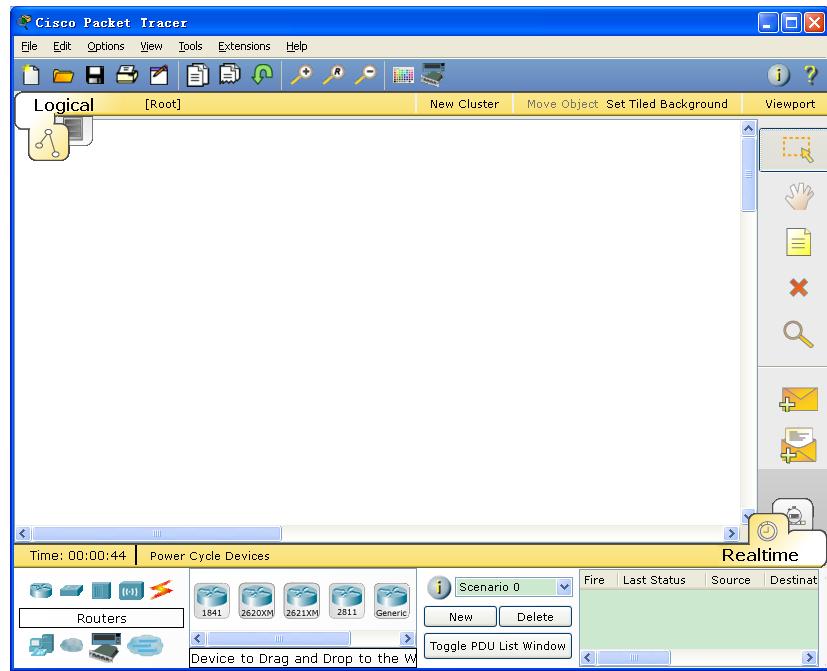


图 33 Packet Tracer 主页面

### 1) 搭建网络环境

使用 Packet Tracer 模拟器自带的设备构建网络拓扑，结构如 34 所示。

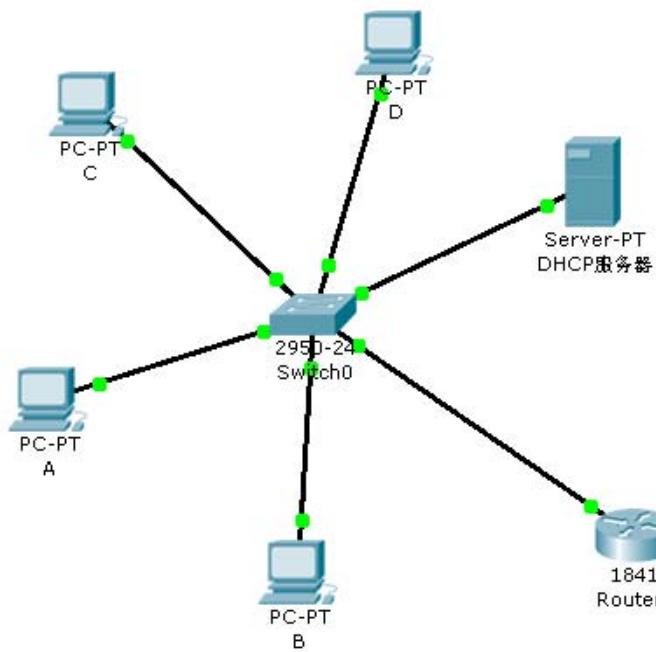


图 34 网络拓扑图

### 2) 配置 DHCP 服务器

点击拓扑图中的 DHCP 服务器，打开服务器配置页面，如 04 所示，在左侧配置栏中选择 DHCP，则显示 DHCP 配置页面。在 Service 栏选择 “on”，打开服务；Default Gateway 和 DNS Server 可以不填；Start IP Address 中输入 DHCP IP 地址池中的开始地址，如 192.168.1.2；Maximum number of users 中输入可以分配的最大地址数个数，如 20，可以自己确定这个数。

目，最大值为 253 个。

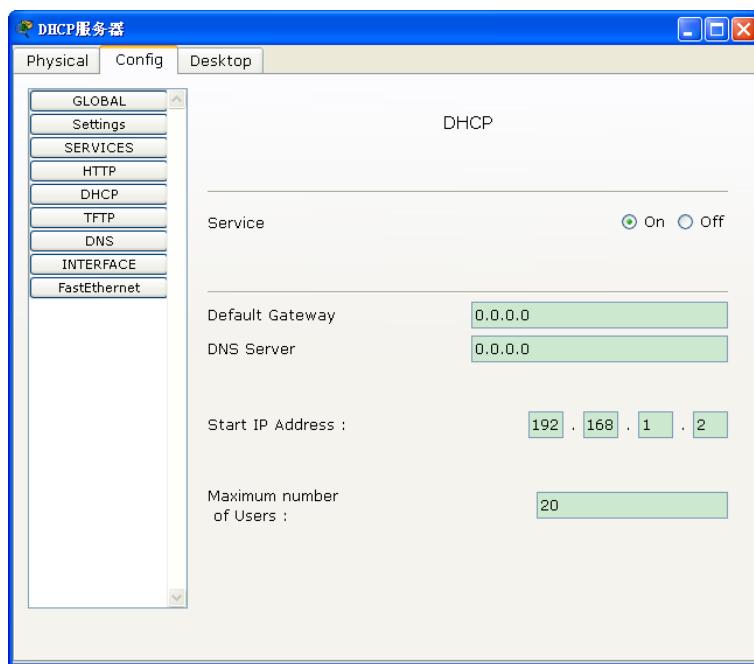


图 35 DHCP 服务器配置页面

### 1. 配置主机

在该局域网中的主机目前有 A、B、C、D 四台，打开任意一台主机的 Config 栏，在 IP Configuration 中选择 DHCP，稍等会发现 IP Address 自动获取了 IP 地址和子网掩码。如图 36 所示。

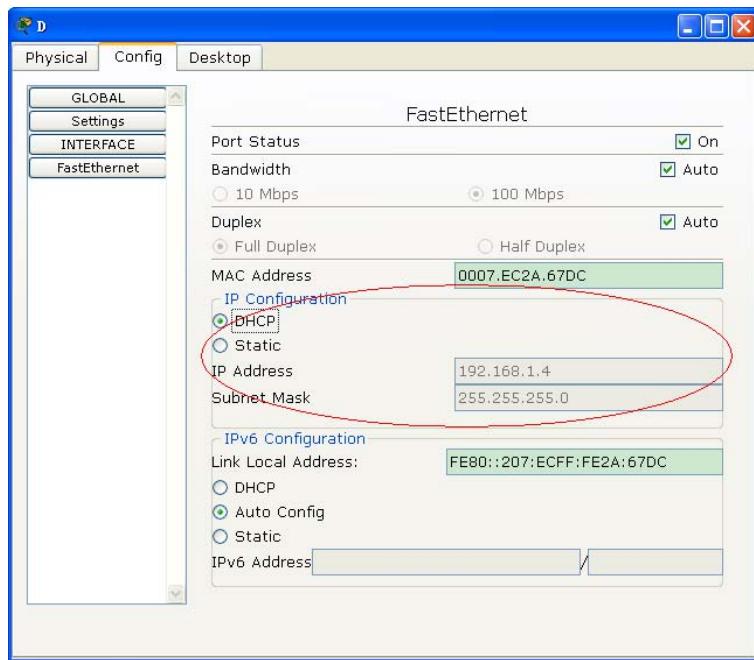


图 36 自动获取网络配置

或者在 Desktop 栏中的 IP Configuration 中选择 DHCP，则稍等也可获取 IP 地址和子网掩码，如图 36 所示。

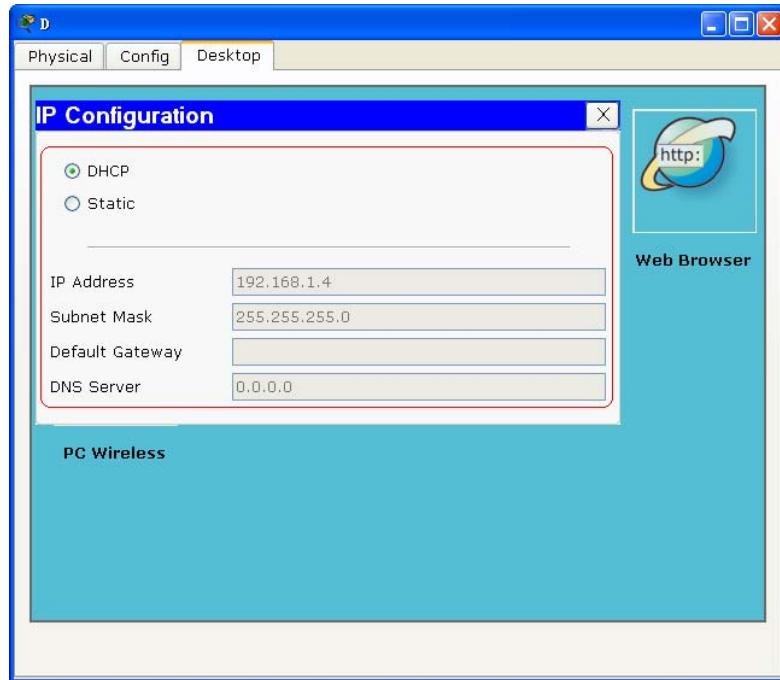


图 37 查看自动获取网络配置

## 五、思考与实践

1. 如果用 nslookup 已经验证了 DNS 的配置是正确的，而用 ping 命令却发现不能解析该域名，则问题出在哪里？若此时 ping 命令能够解析该域名，但是请求超时（ping 的输出为 timeout），这又说明什么？
2. 在 MDaemon 邮件服务器软件中怎样设置通过浏览器方式访问邮件服务器，怎样访问？