实验 8： 运输层端口、TCP 的运输连接管理、动态主机配置协议 DHCP 的作用

课程名称： 计算机网络实验　　实验日期： 2022.11.04

班　　级：计科 5 班　　姓名： 刘洋　　学　　号： 20202619

一、实验目的

1　验证运输层 TCP/IP 端口号的作用

2　验证 TCP 使用三报文握手建立连接

3　验证 TCP 使用四报文挥手释放连接

二、实验环境

Cisco Packet Tracer 模拟器

三、实验内容

1　运输层端口

（1）第一步：构建网络拓扑：在逻辑工作空间上，分别拖动一台主机及两台服务器，使用一台交换机连接，并将一些基本信息标注在设备旁边。如图所 1 示。

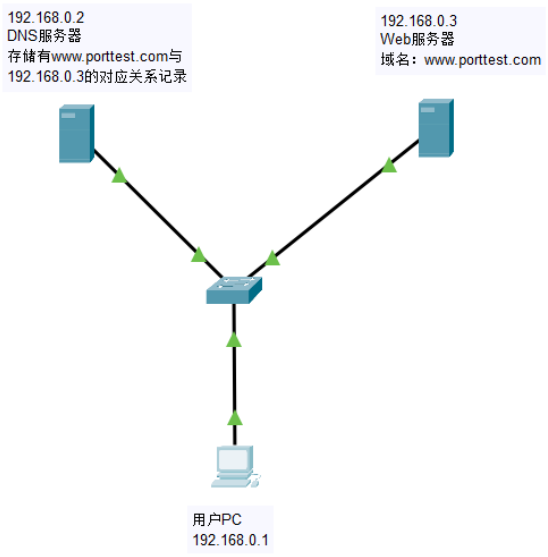

图 1　构建网络拓扑

（2）第二步：设置设备 IP 地址：鼠标左键单击设置的设备，选择桌面，选择 IP 设置，将主机 IP 地址设置为”192.168.0.1”，同时将 DNS 服务器设置为”192.168.0.2”。如图 2 所示。鼠标选择 DNS 服务器，选择桌面，选择配置，将设置 IP 地址为”192.168.0.2”，如图 3 所示。对于 WEB 服务器也进行相应的设置。由于本拓扑属于同一网络，因此无须进行子网的配置。

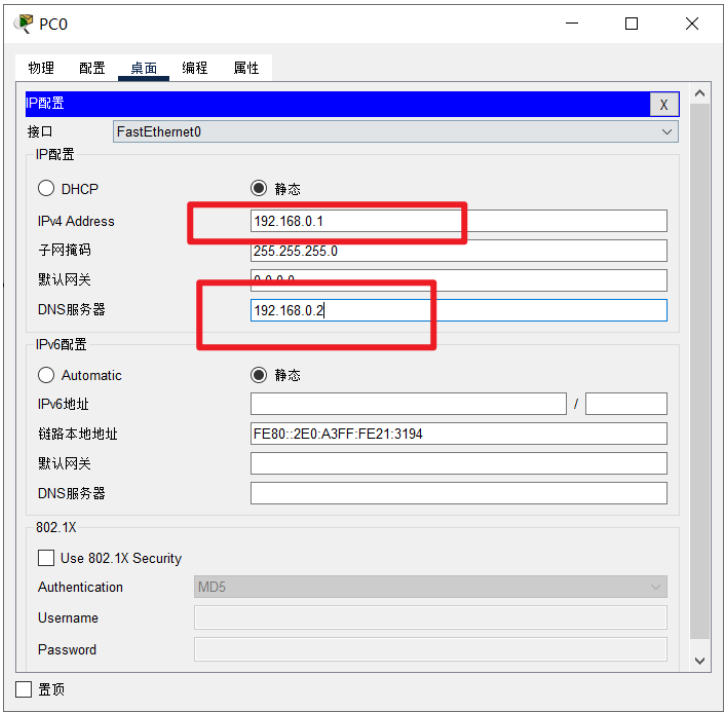

图 2　配置主机的 IP 地址

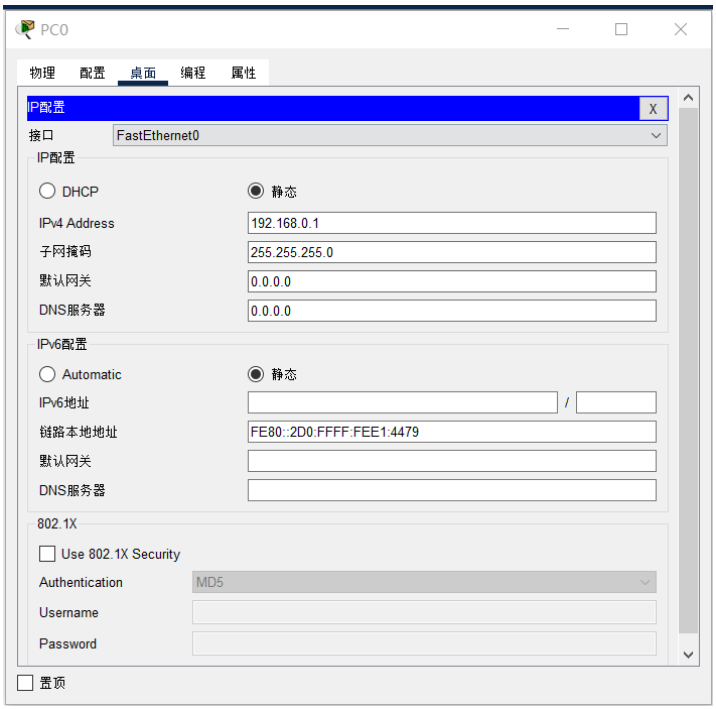

图 3　设置 IP 地址

（3）第三步：配置 DNS 服务器。鼠标选择 DNS 服务器，选择服务，选择 DNS，

添加 www.porttest.com 对应的 IP 地址记录，如图 4 所示。

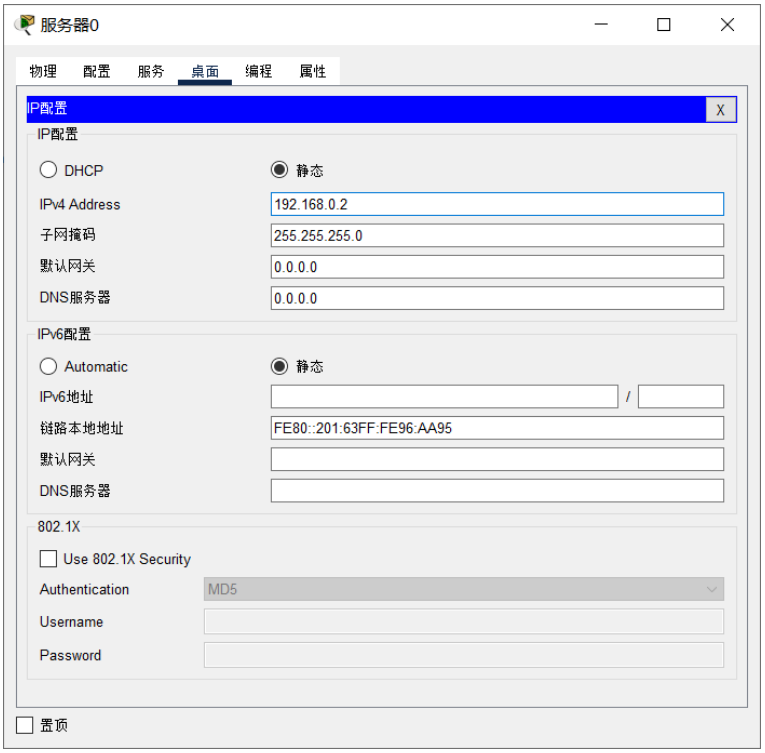

图 4　配置 DNS 服务器

（4）第四步：验证主机与设备之间的连通性。鼠标选择主机，选择桌面，选择命令提示符，输入命令"ping 192.168.0.2"，结果如图 5 所示。输入命令"192.168.0.3"，结果如图 6 所示。收到回复表示主机与设备之间的连通性完好。

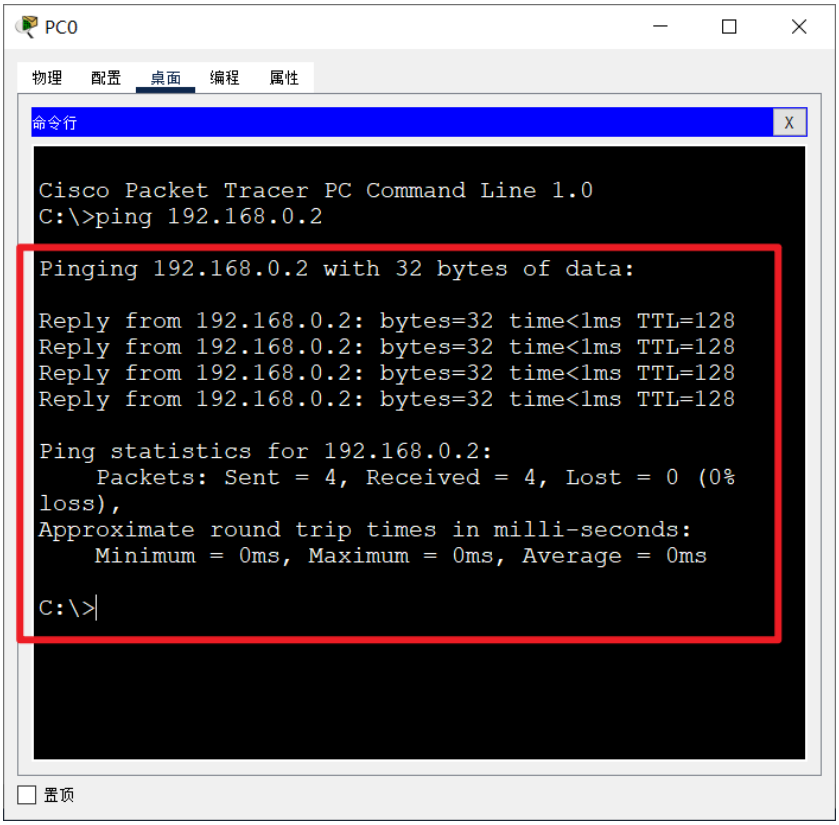

图 5　测试主机与 **DNS** 服务器的连通性

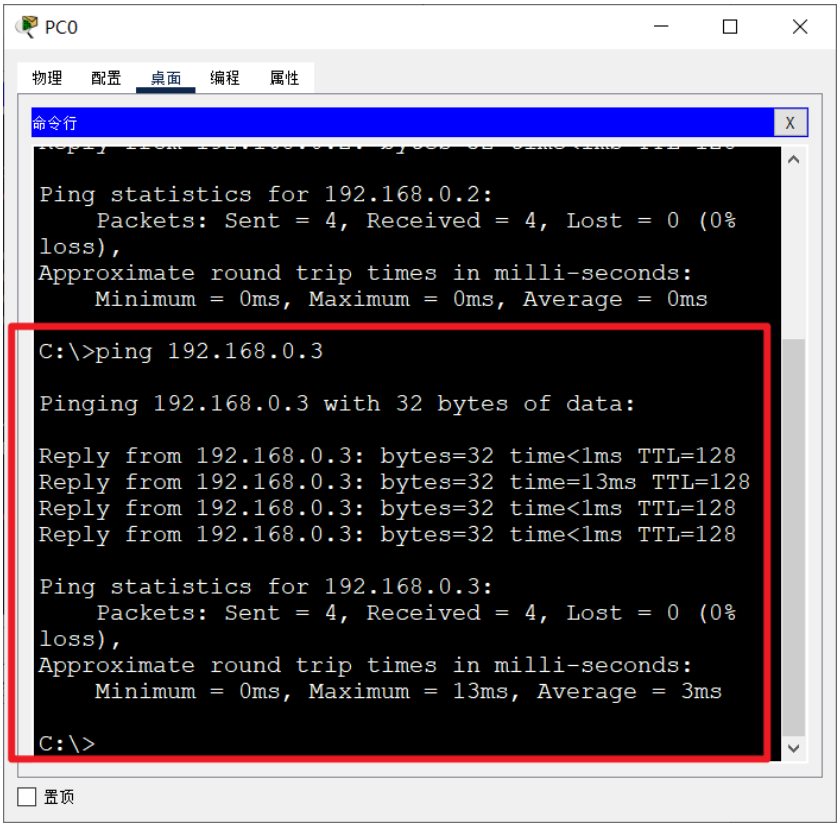

图 6　测试主机与 **WEB** 服务器的连通性

（**5**）第五步：设置监听的协议。切换到仿真模式，鼠标点击控制面板中的”全显/隐藏”按钮，选择”编辑过滤器”，在 **IPv4** 栏选择 **DNS**，如图 **7** 所示。在 **Misc** 栏勾选 **HTTP**，如图 **8** 所示。

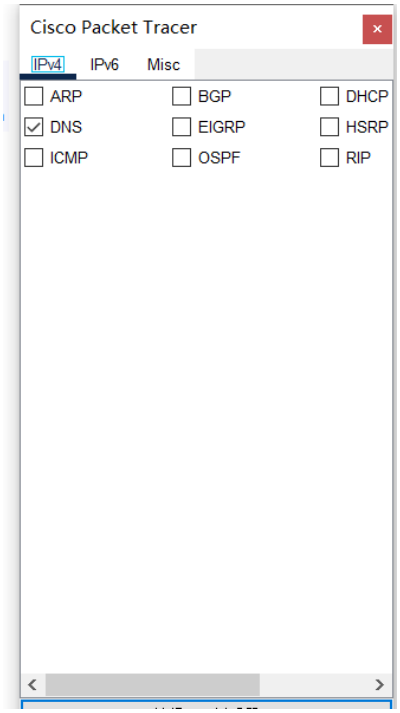

图 **7**　选择 **DNS** 协议

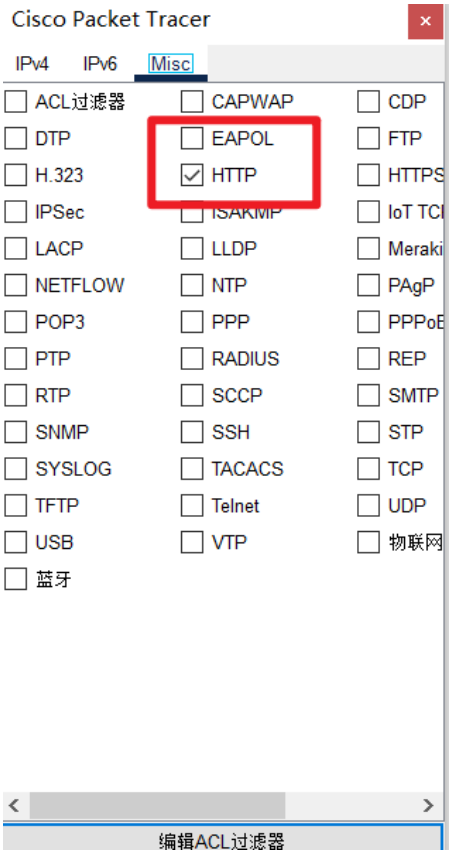

图 8　选择 HTTP 协议

（6）第六步：主机浏览器向 DNS 服务器发送请求。鼠标选择主机，选择桌面，选择浏览器，如图所示。在浏览器窗口输入 www.porttest.com，可以看到主机封装了一个报文，查看该报文，如图 9 所示。可以看到，报文在应用层使用了 DNS 协议，并在运输层封装了 UDP，成为 UDP 用户数据报。UDP 协议中的目的端口为 53，为 DNS 的熟知端口号，源端口号为 1025，为非熟知端口号，标识发送该请求报文的用户进程。UDP 数据报在网络层被封装为 IP 数据报，IP 数据报在数据链路层被封装为以太网帧。

该 DNS 请求帧被发往交换机，接着被转发给 DNS 服务器，服务器对请求层层解封，发现这是一个查询请求，于是将域名对应的 IP 地址封装在数据报中，如图所示。发送给交换机，交换机将该响应报文转发给主机。

主机对收到的数据报层层解封，如图 11 所示。在运输层发现目的端口号为

1025，且数据报包含域名"www.porttest.com"对应的 IP 地址，于是主机的浏览器进程就可以通过所获取到的 IP 地址访问目的域名了。

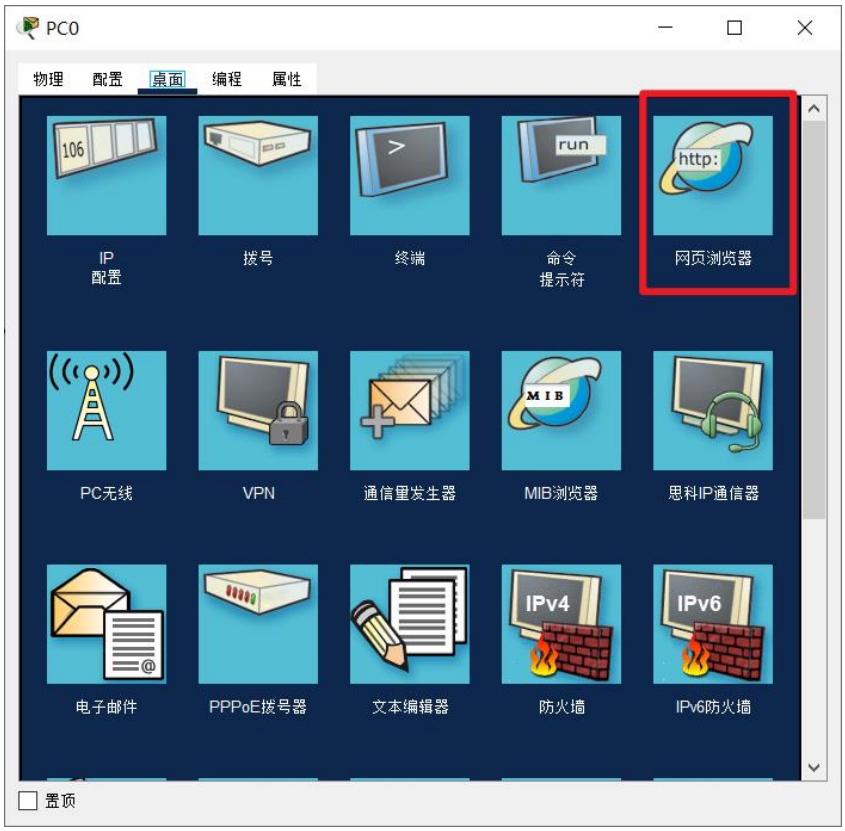

图 9　进入网页浏览器

图 10　查看报文

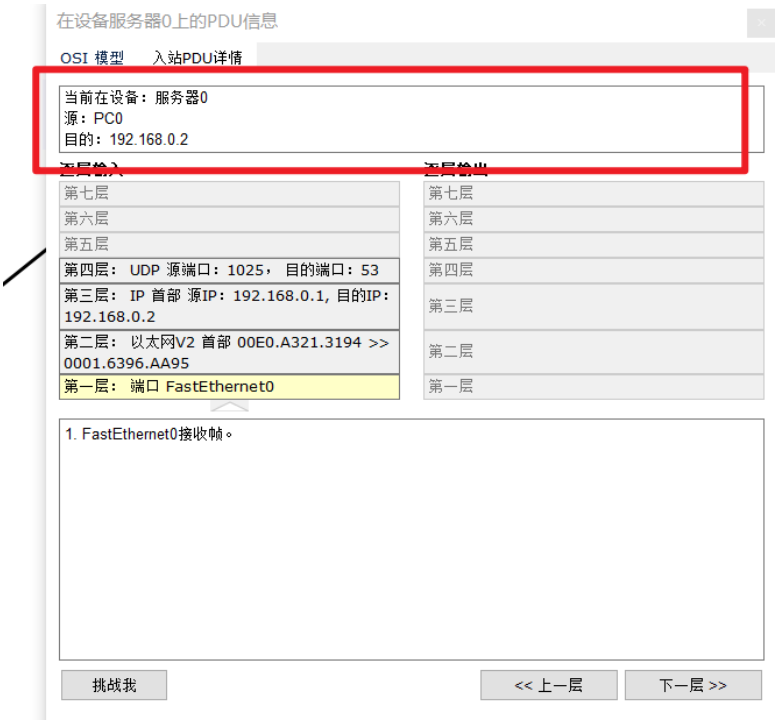

图 11　服务器解析请求

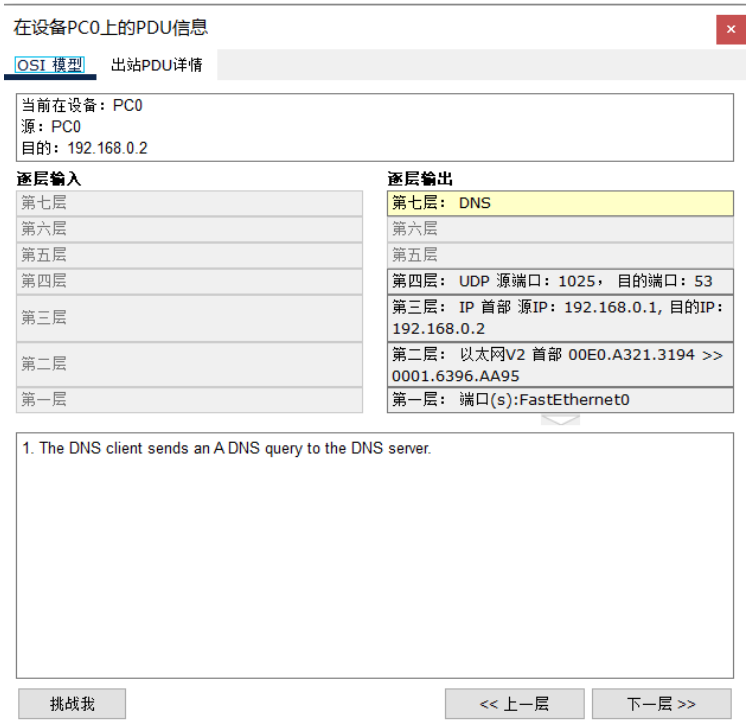

图 12　主机解析 DNS 数据报

（7）第七步：主机浏览器进程向 WEB 服务器发送请求。鼠标选择下一步，可以看到主机构建了一个 HTTP 请求报文，如图 13 所示。可以看到，应用层使用 HTTP 协议，运输层使用 TCP 协议，成为了 TCP 报文。TCP 报文中，目的端口为 80，这是 HTTP 服务器端进程所使用的熟知端口；源端口号为 1025，这是非熟知端口，用来标识发送请求的客户端 HTTP 进程。TCP 报文在网际层使用 IP 协议，被封装为 IP 报文段。

图 13　主机封装 HTTP 请求

（8）第八步：主机浏览器进程向 WEB 服务器发送请求。首先主机与服务器通过 TCP 三次握手建立可靠连接。点击”捕获/前进”按钮，报文被发往交换机，交换机将其转发给 WEB 服务器。点击报文查看细节，如图 14 所示。WEB 服务器会对报文进行层层解封，在 TCP 报文中，目的端口为 80，这是一个熟知端口，表示 WEB 服务器端进程 HTTP 进程；源端口为 1025，这是一个非熟知端口，用来表示主机中发出 HTTP 请求的客户端进程。

WEB 服务器查找该请求对应的内容，将其封装在 HTTP 响应报文中，该报文在运输层使用 TCP 协议进行封装，成为 TCP 报文段，该报文段的源端口、目的端口与之前对应。

报文会被转发到交换机，交换机再将其转发给主机。

主机收到报文后，鼠标点击报文查看细节，如图 15 所示。主机对报文层层解封，该运输层首部中提取出目的端口为 1025，于是将该报文的数据载荷部分

交付给应用层的 **HTTP** 客户端请求进程，该进程对 **HTTP** 响应报文中的内容进行解析，在网页浏览器中进行解析，显示内容如图 **16** 所示。

图 **14** **WEB** 服务器解析报文

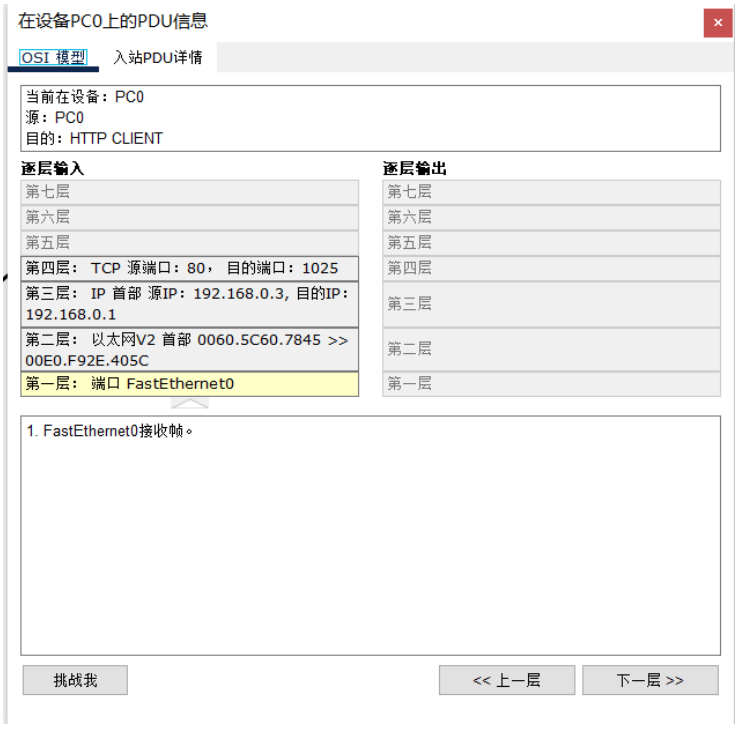

图 15　主机解析报文

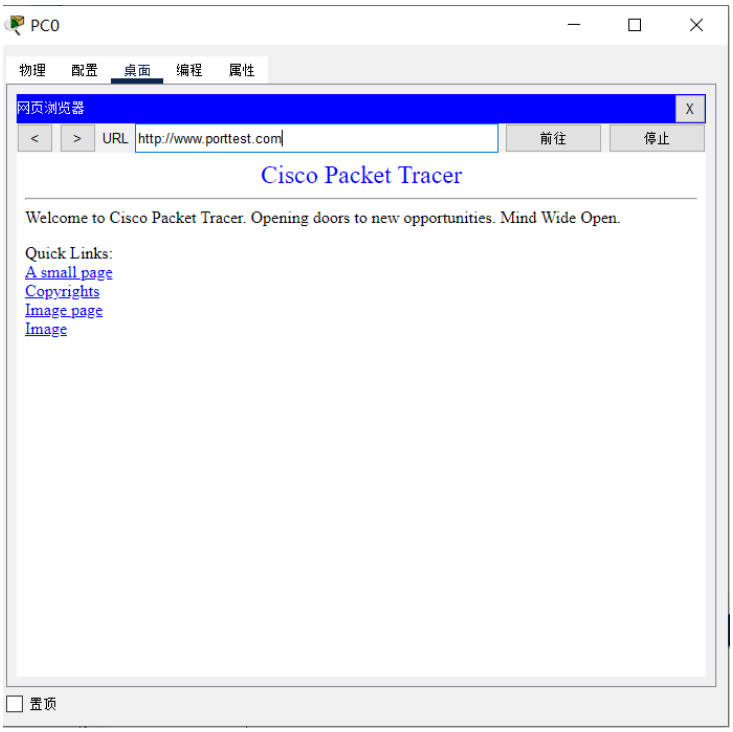

图 16　浏览器显示内容

2　TCP 的运输连接管理

（1）第一步：构建网络拓扑。拖动一台主机与一台服务器，通过自动连线连接，如图 17 所示。

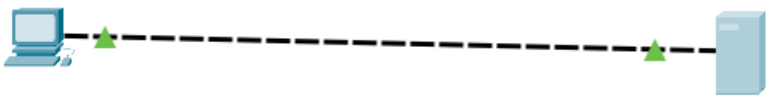

图 17　构建网络拓扑

（2）第二步：配置 IP 地址。鼠标选择主机，选择桌面，选择 IP 地址，输入”192.168.0.1”，按下回车，如图 18 所示。对于服务器进行类似的操作，

配置其 IP 地址为”192.168.0.2”。

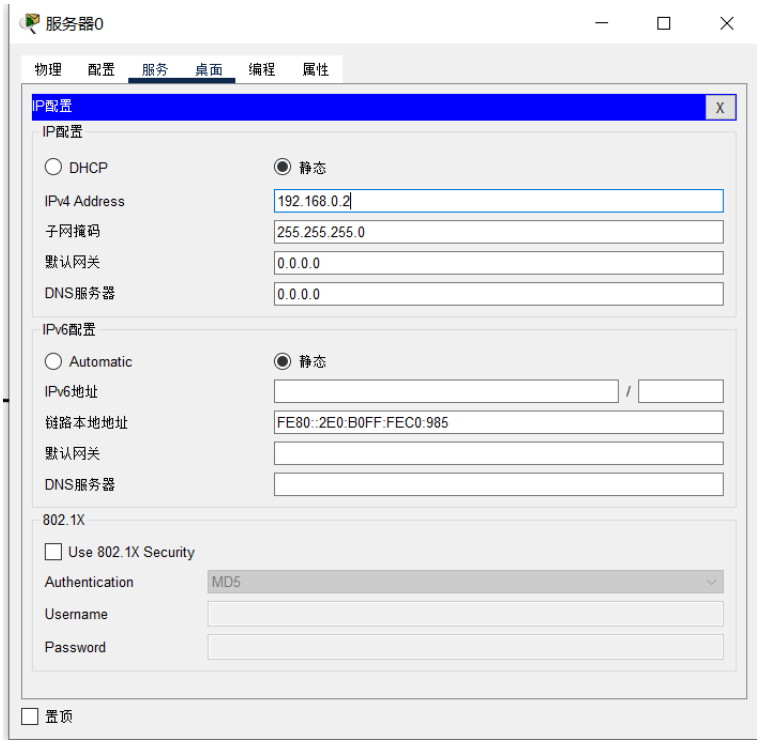

图 19　配置 IP 地址

（3）第三步：测试主机与服务器的连通性。鼠标选择主机，选择桌面，选择网页浏览器如图所示。在实时模式下输入”192.168.0.2”，按下回车后显示网页内容，如图所示。得到结果说明主机与服务器之间连通性完好。

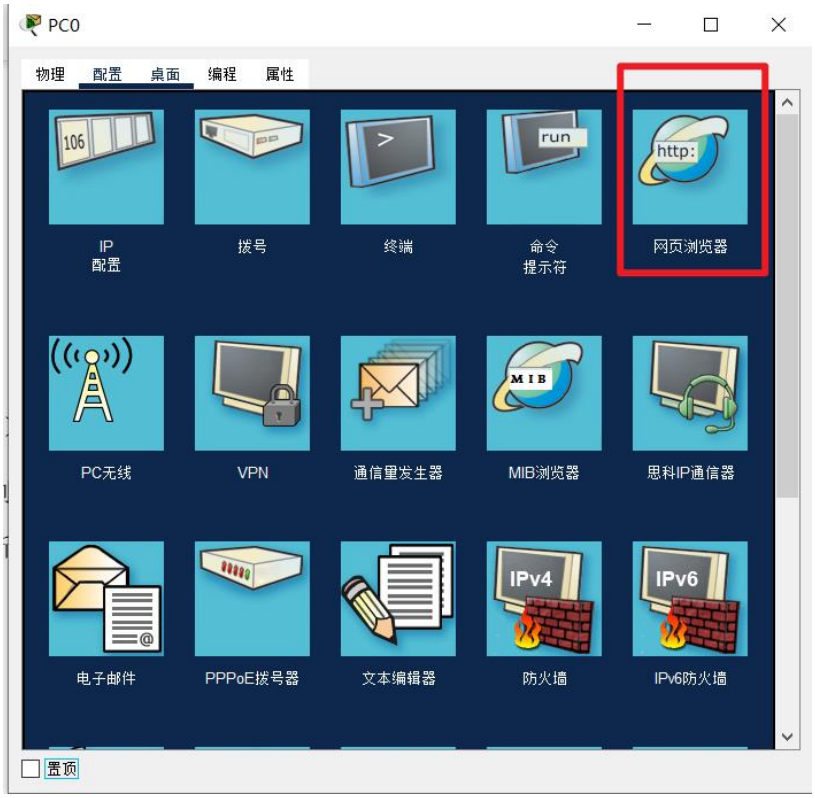

图 19　进入网页浏览器

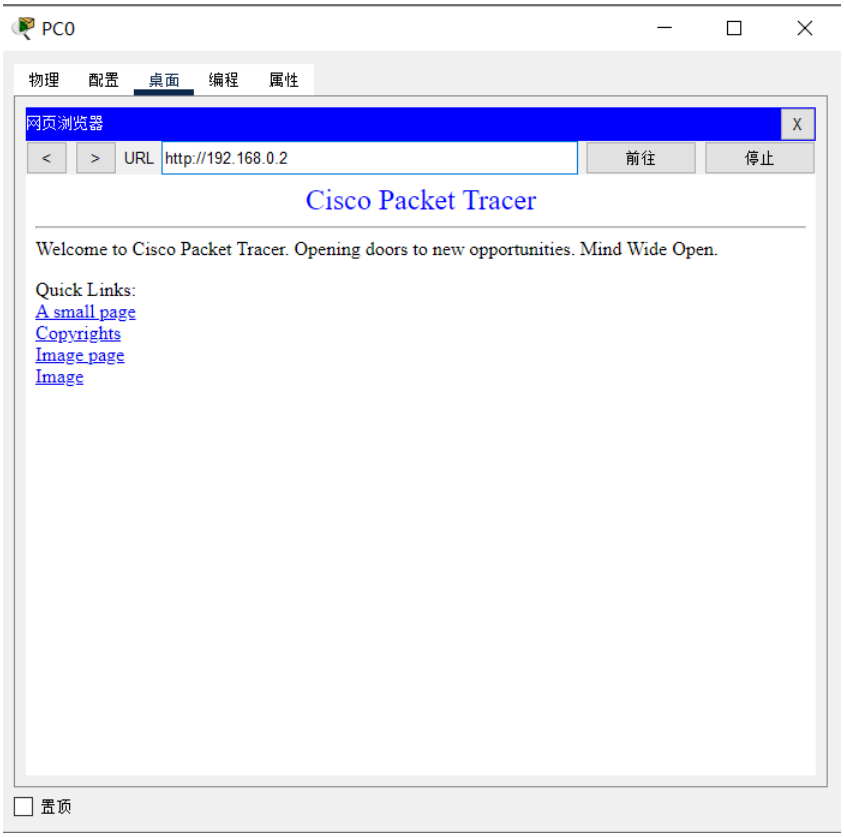

图 20　网页内容

（4）第四步：监听部分协议。鼠标点击控制面板”全显/隐藏”，隐藏全部协议，点击”编辑过滤器”，在 Misc 列表中勾选 HTTP、TCP，如图所示。

图 21　选择部分协议

（5）第五步：主机向浏览器发送 HTTP 请求。在浏览器中输入”192.168.0.2”，点击”前往”。在 HTTP 协议中，HTTP 请求在运输层是使用 TCP 协议，而 TCP 是面向连接的，因此主机此时会尝试与服务器建立 TCP 连接，如图所示。

在主机进程的运输层中，首先会是设置状态为同步已发送（SYN-SENT），之后会与服务器协商一些参数，例如最大窗口等等。，之后会发送一个 SYN 报文段，此报文段中，SYN=1，ACK=0，序号为主机自己选择的 1，数据长度为 24 表示这条 TCP 连接请求报文段的总长度为 24 字节，由于该报文不可以携带数据，所以该报文实际上就是 TCP 的首部（20 字节的固定部分及最大 4 字节的可变部分）。

服务器收到 **TCP** 连接请求后，进行层层解封，其细节如图所示。首先，服务器发现这是一个 **TCP SYN** 连接请求报文，于是接受请求，并设置状态为连接已接受状态（**SYN-RECEIVED**）。接着，服务器准备发送连接请求确认报文段（**TCP SYN+ACK**）。其中，**SYN=1**，**ACK=1**，序号被设置为 **0**（服务器自行决定）。

图 **22**　主机的 **TCP** 请求报文

图 23　服务器解析报文

（6）第六步：服务器向主机发送确认连接报文。、鼠标点击下一步，**TCP SYN-ACK** 报文被发往主机，点击该报文段查看细节，如图所示。

主机对报文层层解封，发现这是一个连接确认报文段，于是进入连接状态（**ESTABLISHED**）。并准备给服务器发挥确认报文段（**TCP-ACK**），其中，**ACK=1**，序号为 **1**（为最开始的序号+1）

主机给服务器发送确认报文段，服务器收到后层解封发现这是一个确认报文段，如图所示。于是设置自己的状态为连接状态（**ESTABLISHED**）。此时，主机与服务器之间可以基于建立好的连接进行数据传输了。

图 **24**　主机解析连接确认报文段

图 **25**　服务器解析确认报文段

（**7**）第七步：主机向服务器发送 **HTTP** 请求。服务器收到后进行层层解封，细

节如图所示。服务器发现这是一个 **HTTP** 请求报文，其中，**ACK=1** 表明该报文段还对此前收到连接请求确认报文段进行了重复确认；序号为 **1**，这是因为此前主机发送的普通确认报文段的序号为 **1**，其不携带数据，不消耗序号。

服务器此时向主机发送响应报文。其中，确认号 **ACK=101**，这是对之前收到的封装有 **HTTP** 请求的报文段的确认，其长度为 **100**，因此此时的确认号为 **101**。数据长度为 **471**。

图 **26**　服务器解析 **HTTP** 请求报文

（**8**）第八步：服务器向主机发送 **HTTP** 响应报文。主机收到后对其中的内容进行解析，将结果显示在浏览器中，如图所示。

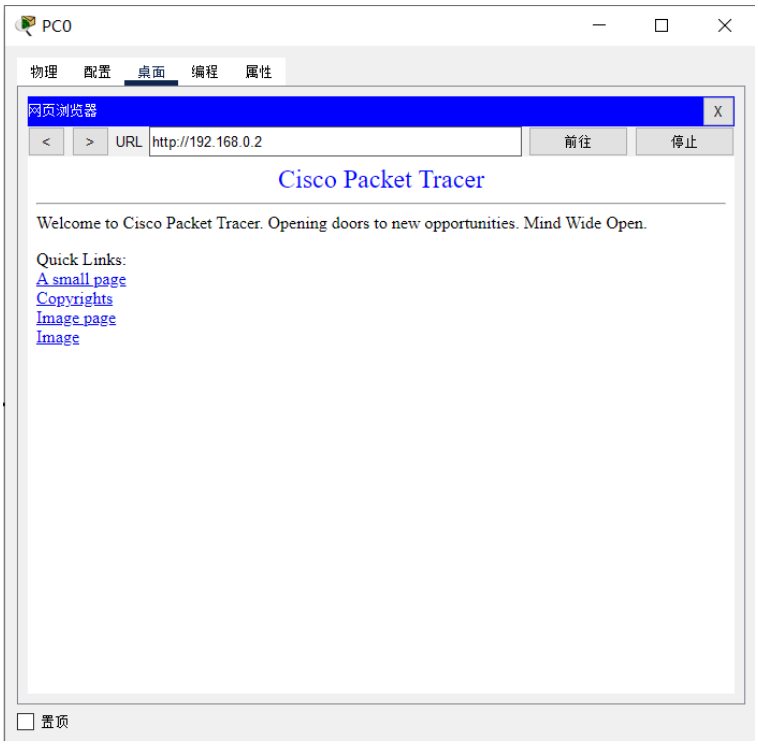

图 27　浏览器解析结果

（9）第九步：收到 HTTP 响应后，主机会向服务器解除 TCP 连接。此时主机会向服务器发送一个报文段，其细节如图所示。首先主机进行终止等待阶段（FIN-WAIT），并发送连接释放报文段（TCP FIN-ACK），其中 ACK=472，这是对封装 HTTP 响应报文段的确认，因为其序号为 1，且长度为 471。序号为 1，这是因为之前发送的 HTTP 请求报文的序号为 0，且长度为 100。

图 28　主机发送断开连接报文段

（10）第十步：主机发送 TCP 连接释放报文段给服务器，服务器层层解封，其细节如图所示。发现这是一个连接释放报文段，于是进入关闭等待（CLOSE-WAIT）状态，此时服务器没有其他数据发送给主机，于是接着进入最后确认状态（LAST-ACK）

图 **29**　服务器解析报文段

（**11**）第十一步：服务器给主机发送连接释放报文段（**FIN-ACK**）。其中，序号字段为 **472**，这是因为此前发送的 **HTTP** 响应报文序号为 **1**，长度为 **471**。**ACK** 为 **102**，这是对收到的 **TCP** 连接释放报文段的确认。点击”捕获前进按钮”。主机收到报文段，进行层层解封，发现这是一个连接释放报文段，于是进行关闭状态（**CLOSING**）。

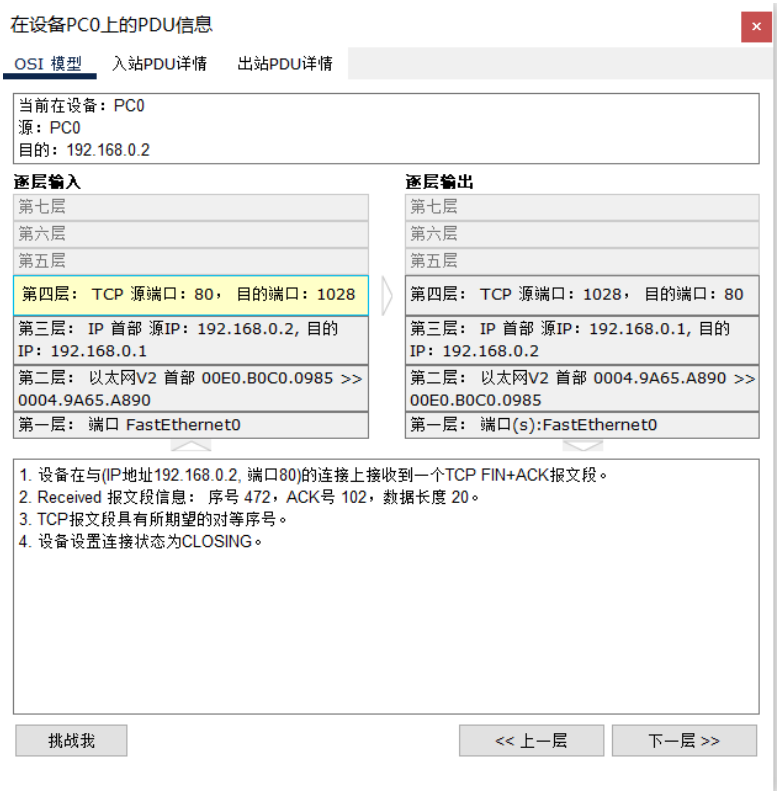

图 30　主机收到服务器的连接释放报文段

（12）第十二步：至此，主机与服务器之间的连接已彻底关闭。

3　动态主机配置协议 DHCP 的作用

（1）配置网路拓扑。在逻辑结构中，配置如图所示的网络拓扑，并在各设备旁边标注对应的信息。

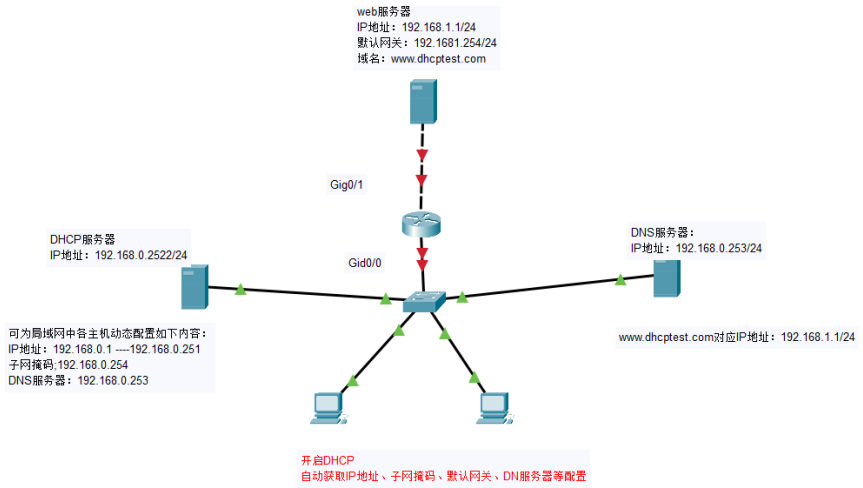

图 31　网络拓扑

（2）第二步：配置 web 服务器。鼠标选中 web 服务器，选中桌面，选择 IP 配置，将 IP 地址、默认网关设置为一开始标注的信息，如图所示。

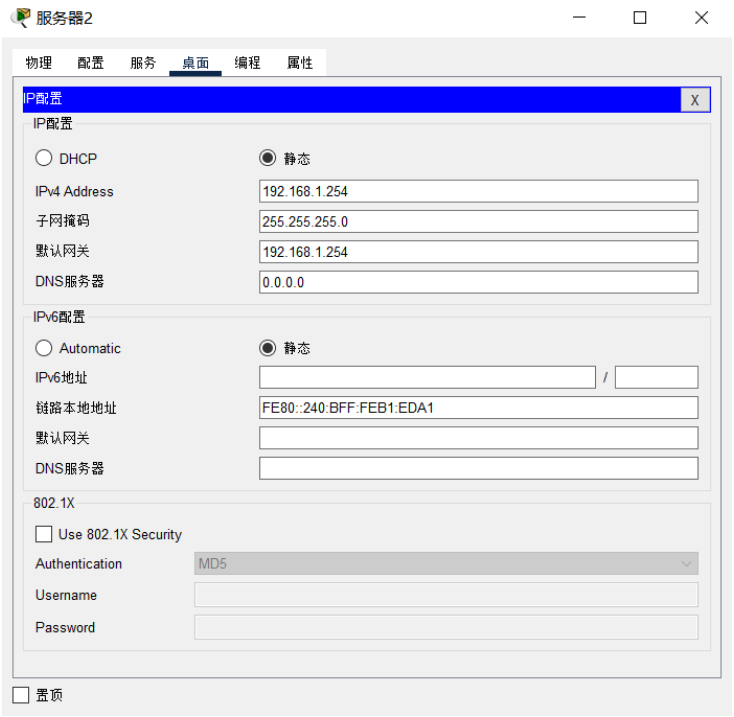

图 32　路由器配置

（3）配置路由器。鼠标选中路由器，点击配置，选择 0/0 端口，将 IP 地址设置为一开始标注的内容，并将端口状态设置为开。，如图所示。对于 1/0 端口也进行相同的配置。

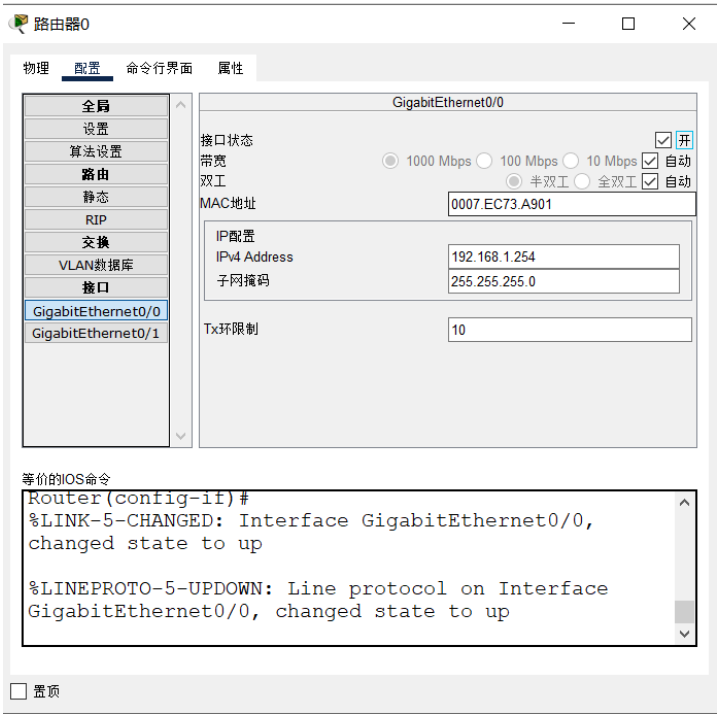

图 33　配置路由器

（4）配置 DNS 服务器，鼠标选择 DNS 服务器，选择桌面，选择 IP 配置，将 IP 地址配置为一开始标注的内容，如图所示。点击服务，选择 DNS，将 www.dhcptest.com 对应 IP 地址：192.168.1.1 进行添加，如图所示。

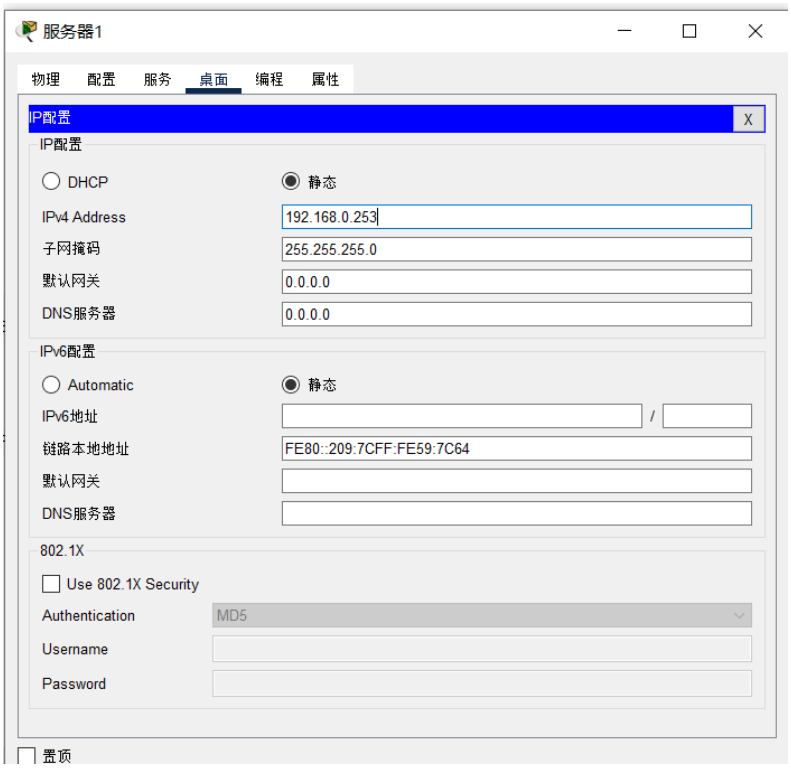

图 **44**　配置 **DNS** 服务器

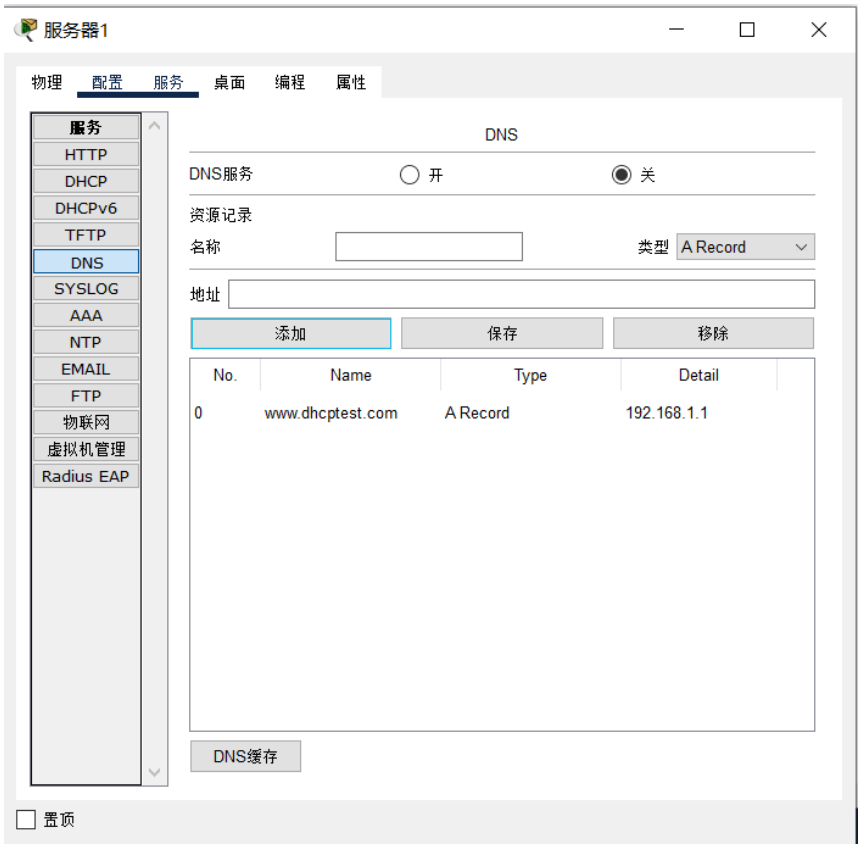

图 **45**　添加 **DNS** 记录

（**5**）配置 **DHCP** 服务器。鼠标选中 **DHCP** 服务器，选中桌面，选中 **IP** 配置，将 **IP** 地址、默认网关配置为一开始标注的信息，如图所示。选中服务，选中 **DHCP**，将默认网关、**DNS** 服务器配置为一开始标注的信息，将 **DHCP** 状态设置为开，如图所示。

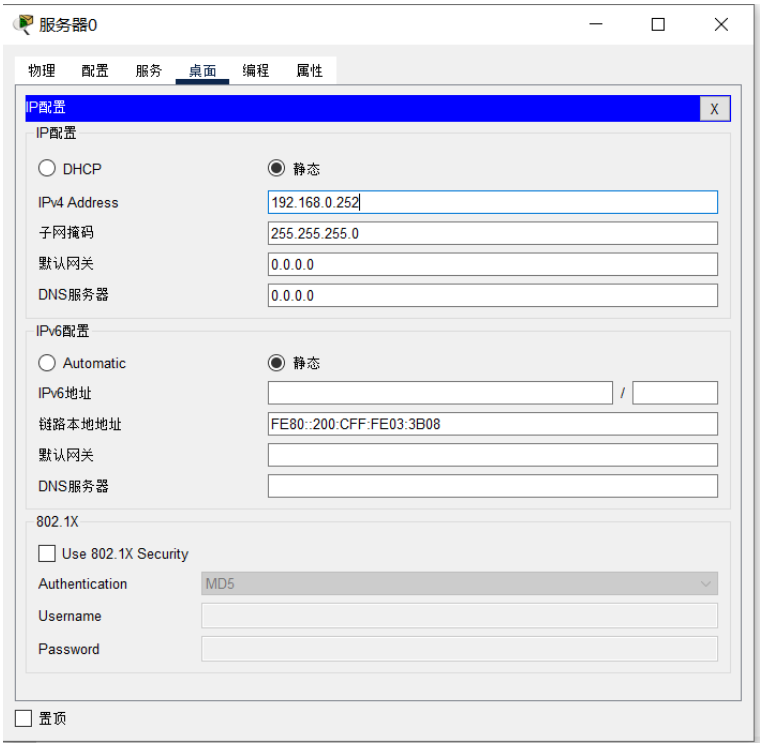

图 **46**　配置 **DHCP** 服务器

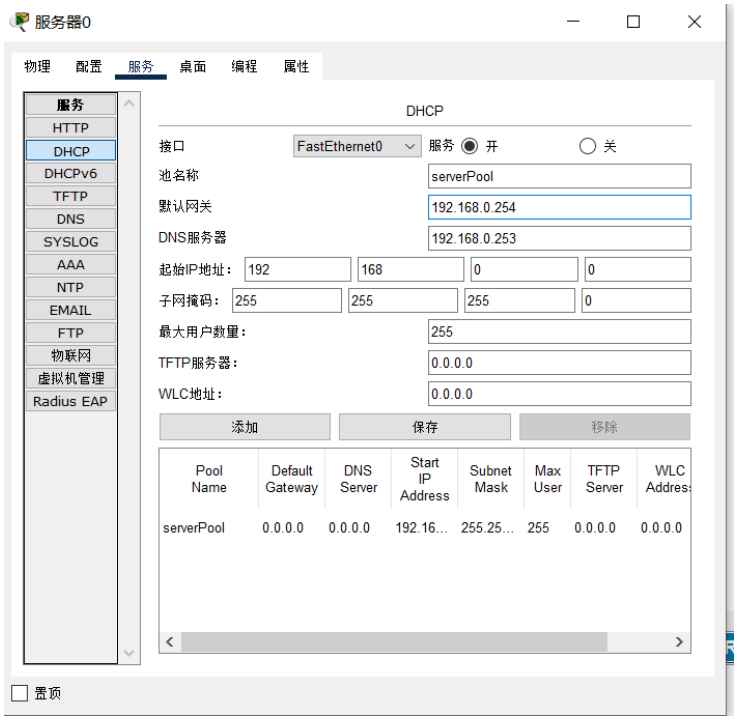

图 **47**　配置 **DHCP** 服务器

（**6**）配置主机。鼠标选中主机，选中桌面，选择 **IP** 配置，将 **IP** 配置切换到 **DHCP**，这样，主机会自动通过 **DHCP** 获取到一个 **IP** 地址，如图所示。

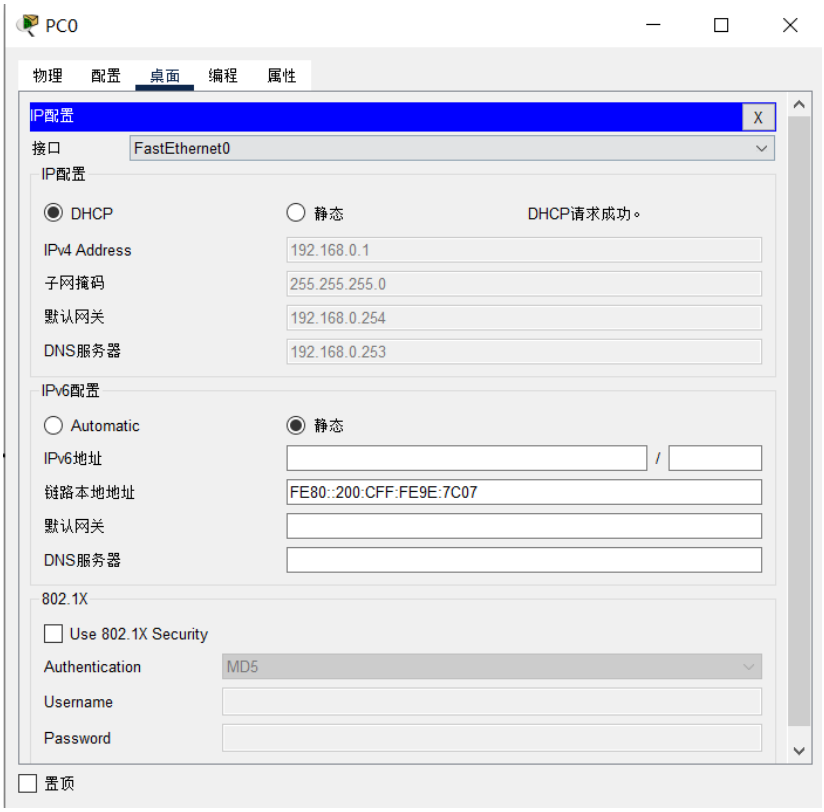

图 48　配置主机

（7）第七步：访问网页。点击主机，选择桌面，选择网页浏览器，输入 www.dhcptest.com，显示界面如图所示。

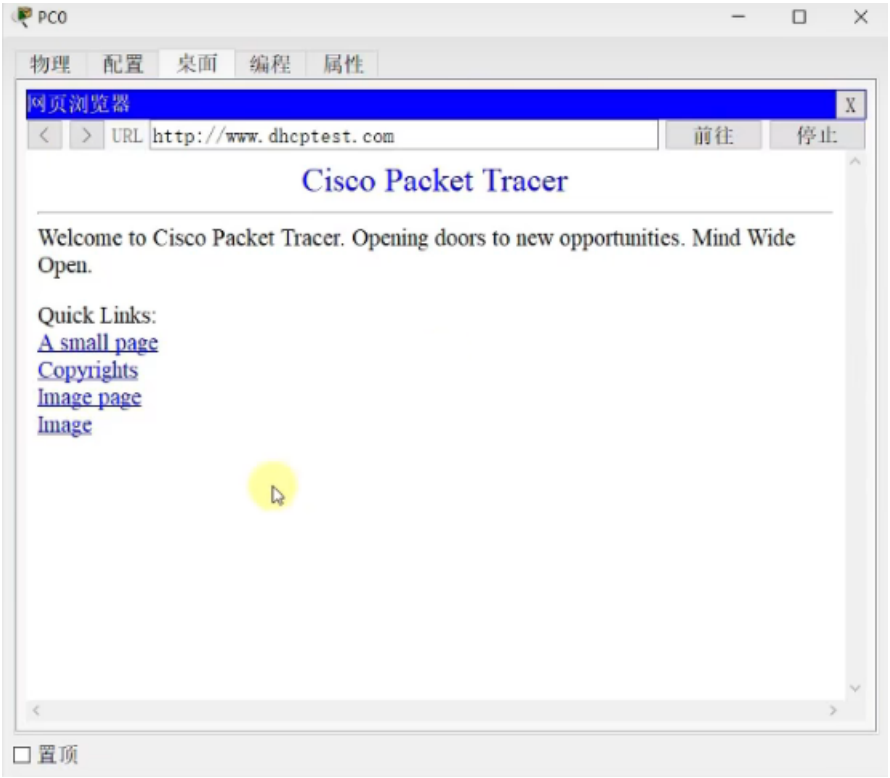

图 **49**　显示网页

四、实验体会

1　**MAC** 地址、**IP** 地址在计算机网络通信中占有极其重要的低位。它们唯一标识了计算机的地址。

2　总线型网络的结构较为简单，但是不可避免的会发生消息的碰撞，这是它的缺点。

3　动态主机配置协议 **DHCP** 对于大面积的网络设备来说是必须的，省去了很多麻烦。