

网络实验手册

V1.0

南京大学软件学院

2018

目录

第一章 交换机	5
1.1 交换机概述.....	5
1.1.1 交换机的工作原理	5
1.1.2 第二层交换技术	6
1.1.3 转发和过滤数据包	7
1.2 交换机内部结构.....	8
1.2.1 构造及主要功能	8
1.2.2 内部结构.....	8
1.3. 交换机配置.....	9
1.3.1 切换命令行界面模式	9
1.3.2 基本交换机配置	10
1.3.3 验证交换机配置	11
1.3.4 基本交换机管理	12
第二章 路由器	13
2.1 路由器概述.....	13
2.1.1 路由器的功能	13
2.1.2 路由器的任务	13
2.2 路由器的内部结构	14
2.2.1 路由器的功能结构	14
2.2.2 路由器的系统组成	15
2.3 路由器配置.....	16
2.3.1 路由器配置途径	16
2.3.2 路由器状态以及配置模式	17
2.3.3 路由器常用配置	17
第三章 路由器基本命令	22
3.1 实验前准备.....	22
3.2 实验要求.....	22
3.3 实验拓扑.....	22
3.4 实验过程.....	23
3.5 实验命令列表.....	24
3.6 实验问题.....	24
第四章 路由器密码恢复	25
第五章 路由器 IOS 备份	26
5.1 实验前准备.....	26
5.2 实验要求.....	26
5.3 实验拓扑.....	26
5.4 实验过程.....	26
5.5 实验命令列表.....	29
5.6 实验问题.....	29
第六章 交换机基本命令	30

6.1 实验前准备.....	30
6.2 实验要求.....	30
6.3 实验拓扑.....	30
6.4 实验过程.....	31
6.5 实验命令列表.....	32
6.6 实验问题.....	32
第七章 交换机端口安全	33
7.1 实验前准备.....	33
7.2 实验要求.....	33
7.3 实验拓扑.....	33
7.4 实验过程.....	33
7.5 实验命令列表.....	34
7.6 实验问题.....	35
第八章 静态路由和简单组网	36
8.1 实验前准备.....	36
8.2 实验要求.....	36
8.3 实验拓扑.....	36
8.4 实验过程(需要区分 DTE/DCE)	36
8.5 实验命令列表.....	37
8.6 实验问题.....	38
第九章 动态 RIP.....	39
9.1 实验前准备.....	39
9.2 实验要求.....	39
9.3 实验拓扑.....	39
9.4 实验过程.....	39
9.5 实验命令列表.....	40
9.6 实验问题.....	40
第十章 配置单域 OSPF	41
10.1 实验前准备.....	41
10.2 实验要求.....	41
10.3 实验拓扑.....	41
10.4 实验过程.....	42
10.5 实验命令列表.....	44
10.6 实验问题.....	44
第十一章 VLAN 间路由	45
11.1 实验前准备.....	45
11.2 实验要求.....	45
11.3 实验拓扑.....	45
11.4 实验过程.....	46
11.5 实验命令列表.....	47
11.6 实验问题.....	47
第十二章 NAT 网络地址转换.....	48
12.1 实验前准备.....	48
12.2 实验要求.....	48

12.3 实验拓扑.....	48
12.4 实验过程.....	49
12.5 实验命令列表.....	55
12.6 实验问题.....	56
第十三章 ACL 实验.....	57
13.1 实验前准备.....	57
13.2 实验要求.....	57
13.3 实验拓扑.....	57
13.4 实验过程.....	57
13.5 实验命令列表.....	61
13.6 实验问题.....	61
第十四章 PPP 验证实验.....	62
14.1 实验前准备.....	62
14.2 实验要求.....	62
14.3 实验拓扑.....	62
14.4 实验过程.....	62
14.4.1 PAP 验证	62
14.4.2 CHAP 验证	63
14.5 实验命令列表.....	65
14.6 实验问题.....	65
第十五章 帧中继实验	66
15.1 实验前准备.....	66
15.2 实验要求.....	66
15.3 实验拓扑.....	66
15.4 实验过程.....	66
15.5 实验命令列表.....	68
15.6 实验问题.....	69
第十六章 DHCP 欺诈保护	70
16.1 实验前准备.....	70
16.2 实验要求.....	70
16.3 实验拓扑.....	70
16.4 实验过程.....	70
16.5 实验命令列表.....	72
16.6 实验问题.....	73
第十七章 IPv6 静态路由与默认路由实验.....	74
17.1 实验前准备.....	74
17.2 实验要求.....	74
17.3 实验拓扑.....	74
17.4 实验过程.....	74
17.5 实验命令列表.....	78
17.6 实验问题.....	78
第十八章 在 IPv6 环境中的 RIPng 的配置	79
18.1 实验前的准备	79
18.2 实验要求.....	79

18.3 实验拓扑图	79
18.4 实验过程.....	79
18.5 实验命令列表	84
18.6 实验问题.....	84
第十九章 基于 IPv6 环境的 OSPFv3 实验	85
19.1 实验前准备.....	85
19.2 实验要求.....	85
19.3 实验拓扑.....	85
19.4 实验过程.....	85
19.5 实验命令列表.....	88
19.6 实验问题.....	88

第一章 交换机

1.1 交换机概述

交换机是一种基于 MAC（网卡的硬件地址）识别，能完成封装转发数据包功能的网络设备，交换机正如它的名字一样采用的是交换的工作模式，它可以“学习”网络中各个终端的 MAC 地址，并把其存放在内部的 MAC 地址表中，通过在数据帧的始发者和目标接收者之间建立临时的交换路径，使数据帧直接由源地址到达目的地址。

交换机拥有一条高性能的背部总线和内部交换矩阵。交换机的所有端口均挂接在这条背部总线上，当控制电路接收到数据包后，处理端口会查找内存中的 MAC 地址对照表以确定目的 MAC 地址的网卡接在哪个端口上，通过内部交换矩阵直接将数据包传送到目的端口，而不是所有端口，交换机的这种工作方式较于集线器来说效率高，不浪费网络资源，因为它只是对目的地址传输数据，发送数据是其他节点很难侦听到所发送的信息。这也是交换机能很快取代集线器的重要原因之一。

交换机的另一个重要特点是它不像集线器一样每个端口共享带宽，它的每一个端口都是共享一部分交换机的总带宽，这样在速率上就对每个端口有个根本的保障。这样交换机就可以在同一时刻进行多个端口之间数据传输，每个端口都视为独立的网段，享有独立固定的带宽。无需同其他设备竞争使用。

交换机的目的是使得传输效率更高，它根据 MAC 地址来进行判断，决定数据帧该送到目的地址的连接端口，而不打扰其他不相干的连接端口，如果内存中的地址表中不包含目的 MAC 地址，交换机则会向所有端口广播这个数据包，找到后再将这个 MAC 地址加入到自己的 MAC 地址表中，这样下次发送到这个地址时便不会发错。

1.1.1 交换机的工作原理

交换机的主要工作原理有以下几条：

1. 地址表：端口地址表记录了端口下包含主机的 MAC 地址。端口地址表是交换机上电后自动建立的，保存在 RAM 中，并且自动维护。交换机隔离冲突域的原理是根据其端口地址表和转发决策决定的。
2. 转发决策：交换机的转发决策有三种操作：丢弃、转发和扩散。丢弃：当本端口下的主机访问已知本端口下的主机时丢弃。转发：当某端口下的主机访问已知某端口下的主机时转发。扩散：当某端口下的主机访问未知端口下的主机时要扩散。每个操作都要记录下发包端的 MAC 地址，以备其它主机的访问。
3. 生存期：生存期是端口地址列表中表项的寿命。每个表项在建立后开始进行倒计时，每次发送数据都要刷新计时。对于长期不发送数据主机，其 MAC 地址的表项在生存期结束时删除。所以端口地址表记录的总是最活动的主机的 MAC 地址。
4. 三层路由：通常，普通的交换机只工作在数据链路层上，路由器则工作在网络层。而功能强大的三层交换机可同时工作在数据链路层和网络层，并根据 MAC 地址或 IP 地址转发数据包。但是要注意到三层交换机并不能完全取代路由器，因为它主要是为了实现处于两个不同子网的 Vlan 进行通讯，而不是用来作数据传输的复杂路径选择。
5. 网管功能：一台交换机所支持的管理程度反映了该设备的可管理性与可操作性。带网管功能的交换机可对每个端口的流量进行监测，设置每个端口的速率，关闭/打开端口

连接。通过对交换机端口进行监测，便于对网络业务流量的区分和迅速进行网络故障定义，提高了网络的可管理性。

6. 端口聚合：这是一种封装技术，它是一条点到点的链路，链路的两端可以都是交换机，也可以是交换机和路由器，还可以是主机和交换机或路由器。基于端口汇聚（Trunk）功能，允许交换机与交换机、交换机与路由器、主机与交换机或路由器之间通过两个或多个端口并行连接同时传输以提供更高带宽、更大吞吐量，大幅度提高整个网络能力。

1.1.2 第二层交换技术

地址学习

以太网交换机通过学习地址来进行数据的转发操作。交换机开机启动后，会自动生成一张表，即 MAC 地址表，交换机关机后，MAC 地址表中的内容会自动清空。

交换机 MAC 地址表用于记录连到交换机的所有设备的位置。交换机的目标是分割网上通信量，使发送到给定冲突域中主机的数据包不至于传播到另一个网段。这是由交换机的“学习”功能完成的，“学习”功能使交换机了解到主机位于哪里。交换机的学习和转发过程如下：

- 当一个交换机首次初始化时，交换机地址表是空的。
- 用一个空 MAC 地址表，基于地址的源过滤或转发决策是不可能的，因此交换机将每一帧转发给所有连接的端口，而不只是接受的端口。
- 转发一个帧到所有连接端口，称为“泛洪”。泛洪是一种通过交换机传输数据的低效方法，因为它将数据帧传输到了不需要的网段，浪费了带宽。
- 因为交换机能同时处理多个网段的通信量，交换机执行内存缓冲以致能独立接受、传输每个端口或网段的数据帧。

MAC 地址表中的内容主要包括交换机端口、与交换机端口相连的主机 MAC 地址、VLAN 标识。图 1.1 显示了不同网段间两个工作站之间的事务。MAC 地址为 0260.8c01.1111 的站点 A 准备发送数据到 MAC 地址为 0260.8c01.2222 的站点 C，交换机接受该帧，执行以下几个动作：

1. 从物理以太网接收该帧并且存储到临时缓冲区。
2. 因为交换机不知道哪个接口连到目的站点，它将帧泛洪给所有端口。
3. 当站点 A 泛洪该帧时，交换机在 MAC 地址表中记录发送数据包站点的源地址及与之相连的端口 F0/1。
4. 如果该记录在一定时间内没有新的帧传到交换机来刷新，这个记录被废弃。

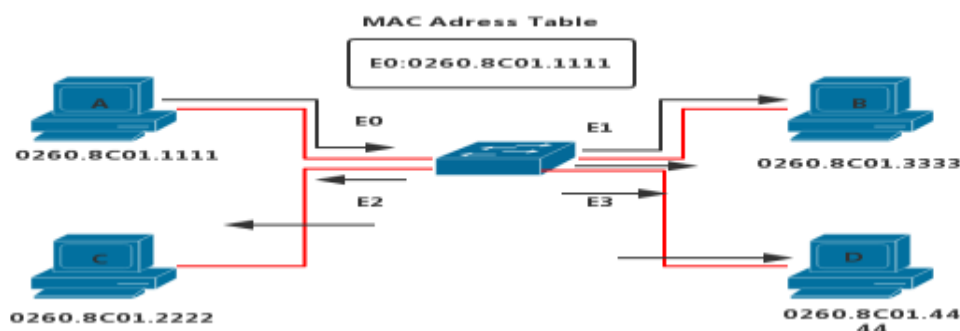


图 1.1 泛洪数据包

当站点继续发送帧到另一个站点时，学习过程继续。MAC 地址为 0260.8c01.4444 的站点 D 给 MAC 地址为 0260.8c01.2222 的站点 C 发送数据包，交换机采取以下几个动作：

1. 源地址 0260.8c01.4444 被加到 MAC 地址表中。
2. 将传输帧的目的 MAC 地址站点 C 与 MAC 地址表记录进行比较。
3. 当软件决定对这个目的地来说至此未见端口到 MAC 地址映射时，该帧被泛洪到交换机中所有的端口。

当站点 A 发送一帧到站点 C 时，交换机查询到站点 C 的 MAC 地址和端口 F0/2，这里交换机将站点 A 发送的数据包直接转发给站点 C，而不发送给 B 和 D 站点，如图 1.2 所示。只要在 MAC 地址表中记录生命周期内所有站点发送数据帧，就可以建立起完整的 MAC 地址表。

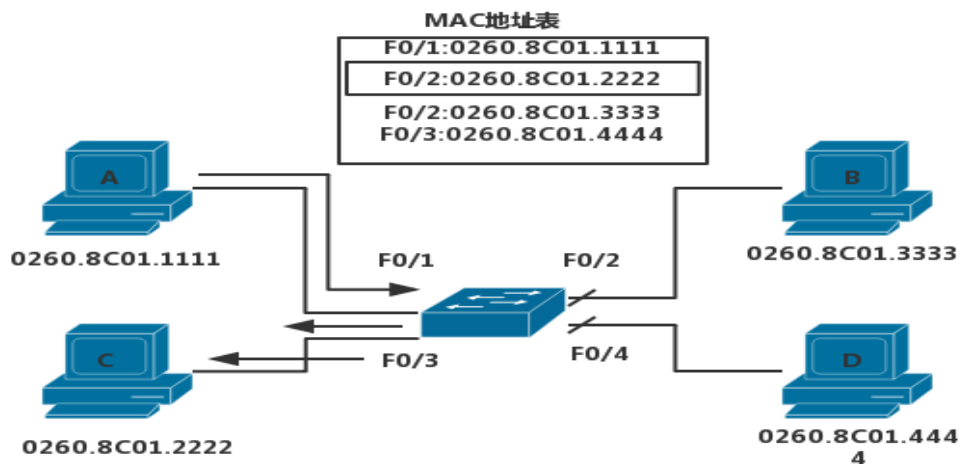


图 1.2 站点应答

1.1.3 转发和过滤数据包

当一个帧带有一个已知目的地址到达时，它被转发到连接该站点而不是所有站点的端口。在图 1.3 中，站点 A 给站点 C 发送一帧。当目的 MAC 地址（站点 C 的 MAC 地址）已在 MAC 地址表中时，交换机只将帧传输到表中所列的这个端口。

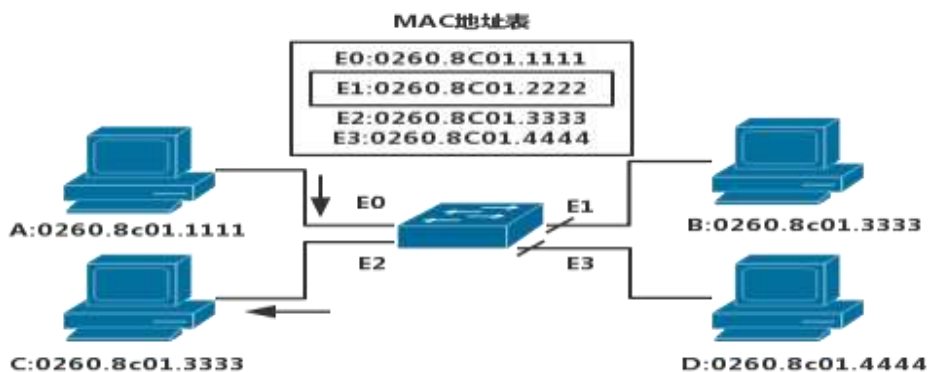


图 1.3 交换机过滤决策

站点 A 发送帧给站点 C 的步骤如下：

1. 传输帧的目的 MAC 地址 0260.8c01.2222 与 MAC 地址表中项进行比较。
2. 当交换机决定目的地址经由 E2 端口可到达时，它将该帧传到该端口。
3. 交换机为了保护链路上的带宽没有将帧传到 E1 和 E3 口，这个动作称为“帧过滤”。

广播和组播是一个特殊情况。因为广播帧和组播帧可能所有站点都关心，交换机通常将广播帧和组播帧泛洪给发起端口外的所有端口。交换机从来不学习广播或组播地址，因为广播地址和组播地址不出现在帧的源地址中。所有站点接受广播帧的事实意味着所有交换网络中网段是在同一广播域中。

1.2 交换机内部结构

1.2.1 构造及主要功能

交换机是一种基于 MAC 地址识别，能完成封装转发数据包功能的网络设备。交换机可以“学习”MAC 地址，并将其存放在内部地址表中，通过在数据帧的始发者和目标接收者之间建立临时的交换路径，使数据帧直接由源地址到达目的地址。

传统的交换机工作在 OSI 模型中的第二层，可以将其看作为一台专用的特殊计算机，主要包括中央处理器(CPU)、随机存储器(RAM)和操作系统。它利用专门设计的芯片 ASIC(Application Specific Integrated Circuits)使交换机以线路速率在所有的端口并行进行转发，因此，它比同在二层利用软件进行转发的网桥速度快的多。

交换机使用一种虚拟连接技术来连接通信的双方。所谓虚拟连接，就是指通信时通信双方建立一个逻辑上的专用连接，这个连接直到数据传送至目的节点后结束。虚拟连接是通过交换机的端口-地址表来实现的：交换机在工作过程中不断地建立和维护它本身的一个地址表，这个地址表标明了节点的 MAC 地址和交换机端口的对应关系。当交换机收到一个数据包，它便会去查看自身的地址表以验明数据包中的目的 MAC 地址究竟对应于哪个端口。一旦验证完毕，就将发送节点与该端口建立一个专用连接，发送方的数据仅发送到目的 MAC 地址所对应的交换机端口。

1.2.2 内部结构

局域网交换机卓越的性能表现，来源于其内部独特的技术结构。而不同的交换模式或不同的交换类型，也跟局域网交换机内部结构密不可分。目前局域网交换机采用的内部技术结构主要有以下几种。

1. 共享内存式结构

该结构依赖于中心局域网交换机引擎所提供的全端口的高性能连接，并由核心引擎完成检查每个输入包来决定连接路由。这种方式需要很大的内存带宽和很高的管理费用，尤其是随着局域网交换机端口的增加，需要内存容量更大，速度也更快，中央内存的价格就变得很高，从而使得局域网交换机内存成为性能实现的主要瓶颈。

2. 交叉总线式结构

交叉总线式结构可在端口间建立直接的点对点连接，这种结构对于简单的单点式 (Unicast) 信息传输来讲性能很好，但并不适合点对多点的广播式传输。由于实际网络应用环境中，广播和多播传输方式很常见，所以这种标准的交叉总线方式会带来一些传输问题。例如，当端口 A 向端口 D 传输数据时，端口 B 和端口 C 就只能等待。而当端口 A 向

所有端口广播消息时，就可能会引起目标端口的排队等候。这样将会消耗掉系统大量带宽，从而影响局域网交换机传输性能。而且要连接 N 个端口，就需要 $N \times (N+1)$ 条交叉总线，因而实现成本也会随着端口数量的增加而急剧上升。

3. 混合交叉总线式结构

鉴于标准交叉总线存在的缺陷，一种混合交叉总线实现方式被提了出来。该方式的设计思路是将一体的交叉总线矩阵划分成小的交叉矩阵，中间通过一条高性能总线连接。该结构的优点是减少了交叉总线数，降低了成本，还减少了总线争用。但连接交叉矩阵的总线成为新的性能瓶颈。

4. 环形总线式结构

这种结构方式在一个环内最多可支持四个交换引擎，并且允许不同速度的交换矩阵互连，以及环与环间通过交换引擎连接。由于采用环形结构，所以很容易聚集带宽。当端口数增加的时候，带宽就相应增加了。与前述几种结构不同的是，该结构方式有独立的一条控制总线，用于搜集总线状态、处理路由、流量控制和清理数据总线。另外，在环形总线上可以加入管理模块，提供完整的 SNMP 管理特性。同时还可以根据需要选用第三层交换功能。这种结构的最大优点就是扩展能力强，实现成本低，而且有效地避免了系统扩展时造成的总线瓶颈。

1.3. 交换机配置

以 Cisco 交换机为例，说明交换机的一些常规配置。

1.3.1 切换命令行界面模式

作为一项安全功能，Cisco IOS 软件将 EXEC（执行）会话分成以下两种访问级别。

用户执行：只允许用户访问有限量的基本监视命令。用户执行模式是在从 CLI 登录到 Cisco 交换机后所进入的默认模式。用户执行模式由 ">" 提示符标识。

特权执行：允许用户访问所有设备命令，如用于配置和管理的命令，特权执行模式可采用口令加以保护，使得只有获得授权的用户才能访问设备。特权执行模式由 # 提示符标识。

要从用户执行模式切换到特权执行模式，输入 enable 命令。要从特权执行模式切换到用户执行模式，输入 disable 命令。在实际网络中，交换机将提示输入口令。默认情况下未配置口令。表 1.1 中显示了用于在用户执行模式和特权执行模式之间来回切换的 Cisco IOS 命令。

表 1.1 执行模式切换

说 明	CLI
从用户执行模式切换到特权执行模式	Switch>enable
如果已为特权执行模式设置口令，则系统将提示您现在输入口令	Password:password
#提示符表示已处于特权执行模式	Switch#
从特权执行模式切换到用户执行模式	Switch#disable
>提示符表示已处于用户执行模式	Switch>

在 Cisco 交换机上进入特权执行模式之后，就可以访问其他配置模式。Cisco IOS 软件的命令模式结构采用分层的命令结构。

1.3.2 基本交换机配置

局域网中的第 2 层交换机的一些关键配置序列通常是在实施过程中进行的。这些配置序列包括配置交换机管理界面，默认网关，全双工和活动接口上的网速设置，对 HTTP 访问的支持和对 MAC 地址表的管理。

1. 管理接口

接入层交换机需要配置 IP 地址、子网掩码和默认网关。要使用 TCP/IP 来远程管理交换机，就需要为交换机分配 IP 地址。此 IP 地址将分配给称为虚拟 LAN（VLAN）的虚拟接口，然后必须确保 VLAN 分配到计算机上的一个或多个特定端口。

要在交换机的管理 VLAN 上配置 IP 地址和子网掩码，必须处在 VLAN 接口配置模式下。先使用命令 `interface vlan 99`，再输入 `ip address` 配置命令。必须使用 `no shutdown` 接口配置命令来使此第 3 层接口正常工作。当看到“interface VLAN x”时，这是指与 VLAN x 关联的第 3 层接口。只有管理 VLAN 才有与之关联的“interface VLAN”。表 1.2 列出了 Catalyst 2960 交换机上的管理接口配置。

表 1.2 管理接口配置

说 明	命 令
进入全局配置模式	S1#configure terminal
进入 VLAN99 接口的接口配置模式	S1(config)#interface vlan 99
配置接口 IP 地址	S1(config-if)#ip address 172.11.99.11 255.255.255.0
启动接口	S1(config-if)#no shutdown
返回特权执行模式	S1(config-if)#end
进入全局配置模式	S1#configure terminal
输入要分配 VLAN 的接口	S1(config)#interface fastethernet 0/18
定义端口的 VLAN 成员模式	S1(config-if)#switchport mode access
将端口分配给 VLAN	S1(config-if)#switchport access vlan 99
返回特权执行模式	S1(config-if)#end
将运行配置保存为交换机启动配置	S1#copy running-config startup-config

2. 默认网关

默认网关是用于将 IP 数据包转发到远程网络的机制。交换机将目的 IP 地址位于本地网络之外的 IP 数据包转发到默认网关。

使用 `ip default-gateway` 命令，为交换机配置默认网关。输入与需要配置默认网关的交换机直接相连的下一跳路由器接口的 IP 地址。

3. 双工和速度

可以使用 `duplex` 接口配置命令指定交换机端口的双工操作模式。可以手动设置交换机端口的双工模式和速度，以避免厂商间的自动协商问题。在将交换机端口双工设置配置为 `auto` 时可能出现问题，在图 1.5 中，S1 和 S2 交换机有着相同的双工设置和速度。

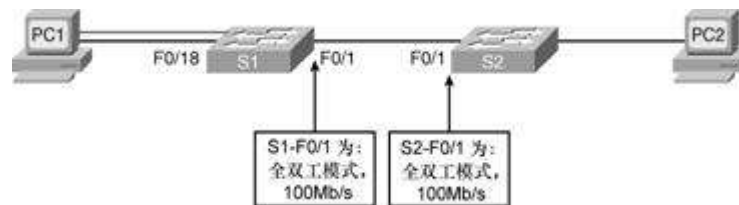


图 1.5 双工和速度

在交换机 S1 上配置端口 F0/1 的具体步骤描述如图 1.6 所示。

```
S1# configure terminal
S1(config)# interface fastethernet 0/1
S1(config-if)# duplex auto
S1(config-if)# speed auto
S1(config-if)# end
```

图 1.6 duplex 和 speed 命令

4. HTTP 访问

如图 1.7 是启用 HTTP 访问的基本配置，ip http authentication enable 是全局配置命令模式。

```
S1# configure terminal
S1(config)# ip http authentication enable
S1(config)# ip http server
```

图 1.7 启用 HTTP 访问的基本配置

5. 管理 MAC 地址表

交换机使用 MAC 地址表来确定如何在端口间转发流量。这些 MAC 表包含动态地址和静态地址。用 show mac-address-table 命令显示 MAC 地址表，其输出包含静态和动态 MAC 地址。

使用 mac-address-table static MAC-address vlan vlan-id interface interface-id 命令可在 MAC 地址表中创建静态映射。使用 no mac-address-table static MAC-address vlan vlan-id interface interface-id 命令可移除 MAC 地址表中的静态映射。

1.3.3 验证交换机配置

执行初始交换机配置之后，可使用不同的 show 命令来验证交换机是否已正确配置。

show 命令从特权执行模式下执行。表 1.3 列出了 show 命令的一些关键选项，它们可用于验证几乎所有可配置的交换机功能。

表 1.3 show 命令

说明	命令
显示交换机上单个或全部可用接口的接口状态和配置	show interface {interface-id cr}
显示启动配置的内容	show startup-config
显示当前运行配置	show running-config
显示关于 flash: 文件系统的信息	show flash:
显示系统硬件和软件状态	show version
显示会话命令历史记录	show history

显示 IP 信息 Interface 选项显示 IP 接口状态和配置 arp 选项显示 IP ARP 表	show ip {interface arp }
显示 MAC 转发表	show mac-address-table

1.3.4 基本交换机管理

交换机启动并运行之后，网络技术人员必须对交换机进行维护，这就包括备份和恢复交换机的配置文件，清除配置信息和删除配置文件。

1. 备份和恢复交换机配置文件

使用 `copy running-config startup-config` 特权执行命令备份了目前创建的配置。如果想在设备上保留多个不同的 `startup-config` 文件，则可以使用 `copy startup-config flash:filename` 命令将配置复制到不同文件名的多个文件中。存储多个 `startup-config` 版本可用于在配置出现问题时回滚到某个时间点。

恢复配置是一个简单的过程。只需用已存配置覆盖当前配置即可。例如，如果有名为 `config.bak1` 的已存配置，则输入 Cisco IOS 命令 `copy flash:config.bak1 startup-config` 即可覆盖现有 `start-config` 并恢复 `config.bak1` 的配置。当配置恢复到 `startup-config` 中后，可在特权执行模式下使用 `reload` 命令重新启动交换机，如表 1.4 所示，使计算机重新加载新的启动配置。`reload` 命令将使系统停止。应在配置信息已输入到文件并保存到启动配置之后再使用 `reload` 命令。

表 1.4 恢复备份文件

说 明	CLI
将存储在闪存中的 <code>config.bak1</code> 文件复制到存储在闪存中的启动配置中。按 Enter 键接受，使用 Ctrl+C 组合键取消	S1#copy flash:config.bak1 startup-config Destination filename [startup-config]?
使 Cisco IOS 执行重新启动交换机。如果修改了运行配置文件，系统将询问是否保存。请按“y”或“n”确认。要确认重新装入，请按 Enter 键接受，使用 Ctrl+C 组合键取消	S1#reload System configuration has been modified. Save?[yes/no]:n Proceed with reload?[confirm]

2. 清除配置信息

交换机配置的清除使用 `erase nvram:`或 `erase startup-config` 特权执行命令实现。当网络技术人员可能执行了一项很复杂的配置任务，并在闪存中存储了文件的很多备份副本，此时要从闪存中删除文件，要使用 `delete flash:filename` 特权执行命令。根据 `file prompt` 全局配置命令的设置，系统可能在技术人员删除文件之前提示确认。默认情况下，在删除文件时，交换机都会提示确认。抹除或删除配置之后，即可重新加载交换机以启动交换机的新配置。

第二章 路由器

2.1 路由器概述

2.1.1 路由器的功能

路由器是在网络层实现互联的设备。路由器实现网络层上数据包的存储转发，它具有路径选择功能，可依据网络当前的拓扑结构，选择“最佳”路径，把接收的数据包转发出去，从而实现网络负载平衡，减少网络拥塞。路由器工作在网络层，用于连接不同的局域网和广域网，故称为“LAN 网间互联设备”。一个路由器可以连接两个局域网、一个局域网和一个广域网，或两个广域网。

路由器的具体功能如下：

1. 路由功能（寻径功能）——寻找并记录到达目的网段的最佳路径，体现在路由器上则包括路由表的建立、维护和查找
2. 交换功能——路由器的交换功能与以太网交换机执行的交换功能不同，路由器的交换功能是指在网络之间转发分组数据的过程，涉及到从接收接口收到数据帧，解封装，对数据包做相应处理，根据目的网络查找路由表，决定转发接口，做新的数据链路层封装等过程
3. 隔离广播、指定访问规则——路由器阻止广播的通过，并且可以设置访问控制列表 (ACL) 对流量进行控制
4. 异种网络互连——支持不同的数据链路层协议，可以连接异种网络
5. 子网间的速率匹配——路由器有多个接口，不同接口具有不同的速率，路由器需要利用缓存及流控协议进行速率适配

2.1.2 路由器的任务

路由器的主要任务是把通信引导到目的地网络，然后到达特定的节点站地址。后一个功能是通过网络地址分解完成的。例如，把网络地址部分的分配指定成网络、子网和区域的一组节点，其余的用来指明子网中的特别站。分层寻址允许路由器对有很多个节点的网络存储寻址信息。在广域网范围内的路由器按其转发报文的性能可以分为两种类型，即中间节点路由器和边界路由器。尽管在不断改进的各种路由协议中，对这两类路由器所使用的名称可能有很大的差别，但所发挥的作用却是一样的。中间节点路由器在网络中传输时，提供报文的存储和转发。同时根据当前的路由表所保持的路由信息情况，选择最好的路径传送报文。由多个互连的 LAN 组成的公司或企业网络一侧和外界广域网相连接的路由器，就是这个企业网络的连界路由器。它从外部广域网收集向本企业网络寻址的信息，转发到企业网络中有关的网络段；另一方面集中企业网络中各个 LAN 段向外部广域网发送的报文，对相关的报文确定最好的传输路径。

2.2 路由器的内部结构

2.2.1 路由器的功能结构

路由器结构从功能上可以分成两个部分：分组转发部分和路由选择部分。分组转发主要由三个部分组成：输入端口，输出端口，交换结构。路由选择部分也可以称作控制部分，其核心是路由选择处理机。

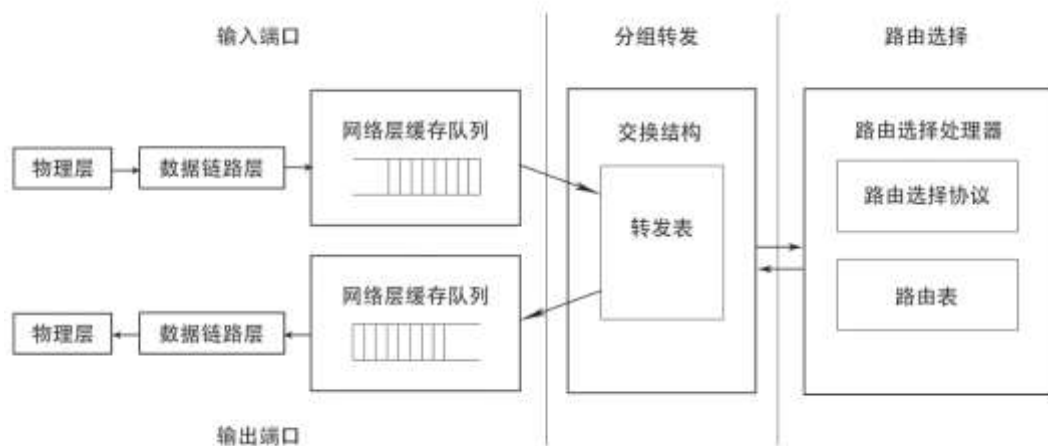


图 2.1 路由器的功能结构示意图

1. 输入端口：输入端口是物理链路和输入包的进口处。端口通常由线卡提供，一块线卡一般支持 4、8 或 16 个端口，一个输入端口具有许多功能。第一个功能是进行数据链路层的封装和解封装。第二个功能是在转发表中查找输入包目的地址从而决定目的端口（称为路由查找），路由查找可以使用一般的硬件来实现，或者通过在每块线卡上嵌入一个微处理器来完成。第三，为了提供 QoS（服务质量），端口要对收到的包分成几个预定义的服务级别。第四，端口可能需要运行诸如 SLIP（串行线网际协议）和 PPP（点对点协议）这样的数据链路级协议或者诸如 PPTP（点对点隧道协议）这样的网络级协议。一旦路由查找完成，必须用交换开关将包送到其输出端口。如果路由器是输入端加队列的，则有几个输入端共享同一个交换开关。这样输入端口的最后一项功能是参加对公共资源（如交换开关）的仲裁协议。

2. 交换结构：交换结构可以使用多种不同的技术来实现。迄今为止使用最多的交换结构技术是总线、交叉开关和共享存储器。最简单的开关使用一条总线来连接所有输入和输出端口，交换结构的缺点是其交换容量受限于总线的容量以及为共享总线仲裁所带来的额外开销。交叉开关通过开关提供多条数据通路，具有 $N \times N$ 个交叉点的交叉开关可以被认为具有 $2N$ 条总线。如果一个交叉是闭合，输入总线上的数据在输出总线上可用，否则不可用。交叉点的闭合与打开由调度器来控制，因此，调度器限制了交换开关的速度。在共享存储器路由器中，进来的包被存储在共享存储器中，所交换的仅是包的指针，这提高了交换容量，但是，开关的速度受限于存储器的存取速度。尽管存储器容量每 18 个月能够翻一番，但存储器的存取时间每年仅降低 5%，这是共享存储器交换开关的一个固有限制。

3. 输出端口：输出端口在包被发送到输出链路之前对包存贮，可以实现复杂的调度算法以支持优先级等要求。与输入端口一样，输出端口同样要能支持数据链路层的封装和解封装，以及许多较高级协议。

4. 路由处理器：路由处理器计算转发表实现路由协议，并运行对路由器进行配置和管理的软件。同时，它还处理那些目的地址不在线卡转发表中的包。

2.2.2 路由器的系统组成

以 cisco 路由器系统组成为例：

$$\text{CPU} + \text{存储器} \left\{ \begin{array}{l} \text{ROM} \\ \text{RAM} \\ \text{FLASH} \\ \text{NVRAM} \end{array} \right. + \text{接口} \left\{ \begin{array}{l} \text{局域网接口} \\ \text{广域网接口} + \text{控制电路} + \text{IOS} \\ \text{调试接口} \end{array} \right.$$

1. CPU

与计算机一样，路由器也包含了一个中央处理器（CPU）。不同系列和型号的路由器，其中的 CPU 也不尽相同。Cisco 路由器一般采用 Motorola 68030 和 Orion/R4600 两种处理器。

路由器的 CPU 负责路由器的配置管理和数据包的转发工作，如维护路由器所需的各种表格以及路由运算等。路由器对数据包的处理速度很大程度上取决于 CPU 的类型和性能。

2. 存储器

ROM:存储开机诊断程序，用于引导操作系统，类似于计算机的 BIOS

RAM:路由器的主存储器，存放 Running-config，路由器，ARP 表，类似于计算机的内存。

FLASH:路由器的快闪存储器，用于存放路由器的 IOS，类似于计算机硬盘。

NVRAM:非易失存储器，用于放置启动配置文件 Startup-Config 文件

3. 接口

$$\left\{ \begin{array}{l} \text{物理接口} \left\{ \begin{array}{l} \text{局域网接口: RJ-45、BNC、FDDI} \\ \text{广域网接口} \left\{ \begin{array}{l} \text{ISDN BRI: RJ45} \\ \text{异步串口: RS-232C、V.24} \\ \text{同步串口: V.35、RS 449} \end{array} \right. \\ \text{调试接口: CON, AUX} \end{array} \right. \\ \text{逻辑接口} \left\{ \begin{array}{l} \text{子接口: 绑定于物理接口作为独立接口使用} \\ \text{Loopback: 反馈接口用于外部网关协议} \\ \text{Null: 清零接口, 用于过滤网络数据包} \\ \text{Tunnel: 隧道接口, 用于建立 VPN 隧道} \\ \text{拨号接口: 用于配置网络拨号及 DDR 的拨号环境} \end{array} \right. \end{array} \right.$$

所有路由器都有接口（Interface），每个接口都有自己的名字和编号。一个接口的全名称由它的类型标志与数字编号构成，编号自 0 开始。

对于接口固定的路由器（如 Cisco 2500 系列）或采用模块化接口的路由器（如 Cisco 4700 系列），在接口的全名称中，只采用一个数字，并根据它们在路由器的物理顺序进行编号，例如 Ethernet0 表示第 1 个以太网接口，Serial1 表示第 2 个串口。

对于支持“在线插拔和删除”或具有动态更改物理接口配置的路由器，其接口全名称中至少包含两个数字，中间用斜杠“/”分割。其中，第 1 个数字代表插槽编号，第 2 个数字代表接口卡内的端口编号。如 Cisco 3600 路由器中，serial3/0 代表位于 3 号插槽上的第 1 个串口。

对于支持“万用接口处理器（VIP）”的路由器，其接口编号形式为“插槽/端口适配器/端口号”，如 Cisco 7500 系列路由器中，Ethernet4/0/1 是指 4 号插槽上第 1 个端口适配器的第 2 个以太网接口。

1) 控制台端口

几乎所有路由器都在路由器背后安装了一个控制台端口。控制台端口提供了一个 EIA/TIA—232(以前叫作 RS—232)异步串行接口、使能与路由器通信。至于同控制台端口建立哪种形式的物理连接，则取决于路由器的型号。有些路由器采用一个 DB25 母连接(DB25F)，有些则用 RJ45 连接器。通常，较小的路由器采用 RJ45 控制台连接器，而较大路由器采用 DB25 控制台连接器。

2) 辅助端口

大多数 Cisco 路由器都配备了一个“辅助端口”(Auxiliary Port)。它和控制台端口类似，提供了一个 EIA / TIA—232 异步串行连接，能与路由器通信。辅助端口通常用来连接 Modem，以实现对路由器的远程管理。远程通信链路通常并不用来传输平时的路由数据包，它的主要的作用是在网络路径或回路失效后访问一个路由器。

4. IOS

IOS 为 CISCO 的专有操作系统，功能有连接多种网络，用于不同协议的路由和转换，实现流量控制、QoS 服务质量控制、网络安全服务，网络拨号及 VPN 等。

IOS { 基本特征集：为 CISCO 路由器使用的基本功能操作系统 c2600 – is
增强特征集：集成了硬件平台特性的增强功能操作系统 c2600 – io3
加密特征集：使用了加密技术的 CISCO 操作系统 c2600 – ipvoice

有两种类型的 IOS 配置：

1)运行配置：有时也称作“活动配置”，驻留于 RAM，包含了目前在路由器中“活动”的 IOS 配置命令。配置 IOS 时，就相当于更改路由器的运行配置。

2)启动配置：启动配置驻留在 NVRAM 中，包含了希望在路由器启动时执行的配置命令。有时也把启动配置称作“备份配置”。这是由于修改并认可了运行配置后，通常应将运行配置复制到 NVRAM 里，将作出的改动“备份”下来，以便路由器下次启动时调用。启动完成后，启动配置中的命令就变成了“运行配置”。

两者均以 ASCII 文本格式显示。所以，能够很方便地阅读与操作。一个路由器只能从这两种类型中选择一种。

2.3 路由器配置

2.3.1 路由器配置途径

1.控制台

将 PC 机的串口直接通过 Rollover 线与路由器控制台端口 Console 相连，在 PC 计算机上运行终端仿真软件，与路由器进行通信，完成路由器的配置。也可将 PC 与路由器辅助端口 AUX 直接相连，进行路由器的配置。

2.虚拟终端(Telnet)

如果路由器已有一些基本配置，至少有一个端口有效(如 Ethernet 口)，就可通过运行 Telnet 程序的计算机作为路由器的虚拟终端与路由器建立通信，完成路由器的配置。

3.网络管理工作站

路由器可通过运行网络管理软件的工作站配置，如 Cisco 的 CiscoWorks、HP 的 OpenView 等。

4.CISCO ConfigMaker

ConfigMaker 是一个由 CISCO 开发的免费的路由器配置工具。ConfigMaker 采用图形化的方式对路由器进行配置，然后将所做的配置通过网络下载到路由器上。ConfigMaker 要求路由器运行在 IOS 11.2 以上版本，可用 Show Version 命令查看路由器的版本信息。

5.TFTP(Trivial File Transfer Protocol)服务器

TFTP 是一个 TCP/IP 简单文件传输协议，可将配置文件从路由器传送到 TFTP 服务器上，也可将配置文件从 TFTP 服务器传送到路由器上。TFTP 不需要用户名和口令，使用非常简单。

注意：路由器的第一次设置必须通过第一种方式进行；这时终端的硬件设置为波特率：9600，数据位：8，停止位：1，无校验。

2.3.2 路由器状态以及配置模式

路由器的配置模式是通过控制台连接路由器进入的模式，该模式下路由器有以下几个状态。

1. 用户命令状态

前置符类似“Router>”，此时路由器处于用户命令状态，这时用户可以看路由器的连接状态，访问其它网络和主机，但不能看到和更改路由器的设置内容。

2. 特权命令状态

前置符类似“Router#”，用户命令状态下输入“enable”即可进入，此时路由器处于特权命令状态，这时不但可以执行所有的用户命令，还可以看到和更改路由器的设置内容。

3. 全局设置状态

前置符类似“Router(config)#”，特权命令状态下输入“configure terminal”即可进入，此时路由器处于全局设置状态，这时可以进行路由器端口以外的一些设置，如：路由协议，nat 等。

4. 局部设置状态

从全局设置状态进入，对某个功能的详细设置，这时可以设置路由器某个局部的参数。

5. RXBOOT 状态

前置符为“>”，在开机后 60 秒内按 ctrl-break 可进入此状态，这时路由器不能完成正常的功能，只能进行软件升级和手工引导。

6. 设置对话状态

这是一台新路由器开机时自动进入的状态，在特权命令状态使用 SETUP 命令也可进入此状态。这时可通过对话方式对路由器进行设置。

2.3.3 路由器常用配置

[路由器使用注意事项]

1. 须确认线路连接正确后才能打开路由器电源。
2. 绝对不允许热插拔 flash 卡（用于装载 IOS），否则易造成 flash 卡烧毁。
3. 不允许频繁开关路由器。

1. 连接路由器

- 1) 用 console 线（反转线，注意与网线的比较）把计算机的串口（com1, i.e. RS232）与路由器的 console 口直接相连。
- 2) 在 win2000 中打开“附件/通讯/超级终端”建立连接，在连接设置的波特率选择 9600，其余为默认选项。
- 3) 开机，通常进入用户模式，使用 Enable 命令，进入特权模式。

2. 状态命令

show version 这个命令可以查看 IOS 版本号，已启动时间，flash 中的 IOS 的文件名，router 里面共有什么的端口，寄存器的值等等。

show protocol 显示与 IP 有关的路由协议信息。各个端口的情况。

show flash 查看 flash 中的内容，IOS 的长度，文件名，剩余空间，总共空间。

show running-config 查看路由器当前的配置信息。

show startup-config 查看 nvram 中的路由器配置信息。

show interface 查看路由器上的各个端口的状态信息。（很多重要信息）

show controller 查看接口控制器的状态，可看到连接的是 DTE 还是 DCE

show history 查看 history buffer 里面的命令列表

show controller s0 查看 s0 是 DCE 口还是 DTE 口

show ip route 查看路由器的路由配置情况

show hosts 查看 IP host 表

terminal history size<size> 设置 history buffer 里面保存命令的个数，最大允许为 256

3. 修改系统时钟（按步骤体验一下？的作用）可以顺带提一提 tab 键的功能

- 1) clock
- 2) clock ?
- 3) clock set ?
- 4) clock set 10:30:30 ?
- 5) clock set 10:30:30 20 oct ?
- 6) clock set 10:30:30 20 oct 2001 ?
- 7) enter
- 8) show clock

4. 使用组合键编辑

输入一行命令（不执行它），然后操作下列组合键：

Ctrl+A: 光标回到命令行的最开头

Ctrl+E: 光标回到命令行的最后

Ctrl+B: 光标向左移一字符位置

Ctrl+F: 光标向右移一字符位置

执行刚刚输入的命令，然后操作下列组合键：

Ctrl+P (or 上箭头): 使用上一条用过的命令

Ctrl+N (or 下箭头): 使用下一条用过的命令

Ctrl+Z (在其他模式下): 保存设置并退出到特权模式

可以使用 terminal no editing 命令来使组合键失效，要使组合键重新生效，可用 terminal editing 命令。

5. 路由器中各种配置模式的转换

路由器的几种配置模式：

- 1) 用户模式 (user mode) router>
- 2) 特权模式 (privileged mode) router#
- 3) 全局配置模式 (global configuration mode) : router (config)#
- 4) Setup 模式 (setup mode):
- 5) ROM Monitor 模式(ROM Monitor Mode): > 或 rommon>。
- 6) RXBoot 模式(RXBoot mode): Router<boot>
(注：前 3 种模式是该实验需要用到)
- 7) 用户模式 (user mode)：该模式下只能查看路由器基本状态和普通命令，不能更改路由器配置。此时路由器名字后跟一个“>”符号，表明是在用户模式下。如：router>
- 8) 特权模式 (privileged mode)：该模式下可查看各种路由器信息及修改路由器配置。在用户模式下以 enable 命令登陆，此时“>”将变成“#”，表明是在 privileged mode。如：
router#
- 9) 全局配置模式 (global configuration mode)：该模式下可进行更高级的配置，并可由此模式进入各种配置子模式。其提示符如：router (config)#
- 10) Setup 模式 (setup mode)：该模式通常是在配置文件(configuration file)丢失的情况下进入的，以进行手动配置。在此模式下只保存着配置文件的最小子集，再以问答的形式由管理员选择配置。
- 11) ROM Monitor 模式(ROM Monitor Mode)：当路由器启动时没有找到 IOS 时，自动进入该模式。提示符为> 或 rommon>。
- 12) RXBoot 模式(RXBoot mode)：该模式通常用于密码丢失时，要进行破密时进入。其提示符如：Router<boot>

```
Router>
Router>enable
Router#
Router#configure terminal
Router(config)#
Router(config)#int f0/0
Router(config-if)#
输入 Ctrl+Z
Router#
```

6. 给路由器命名

进入全局配置模式，用 hostname <name>命令来设定路由器的名称。

7. 编辑路由器登录信息

```
banner motd <message>
例： banner motd #
You are logging in C1600 Router
#
```

8. 给端口配 IP 地址

在全局配置模式下，进入各端口配置模式配置 IP 地址。

1) 以太网口的配置

```
Router(config) # int f0/0
```

```
Router (config-if) # ip address <ipaddress><subnet marsk>
```

```
Router(config-if) # no shutdown
```

2) 串行线，根据串口是 DTE 还是 DCE 选择下面的配置

DTE:

```
Router(config)# int s0/1
```

```
Router(config-if) # ip address <ip address><subnet marsk>
```

```
Router(config-if)#bandwidth 64
```

```
Router(config-if)# no shutdown
```

DCE:

```
Router(config)# int s0/0
```

```
Router(config-if) # ip address <ip address><subnet marsk>
```

```
Router(config-if)#bandwidth 64
```

```
Router(config-if)#clock rate 56000
```

```
Router(config-if)# no shutdown
```

9. Ping 命令

```
ping <ip address>
```

命令: Router# ping <ip address>

```
Router# ping <hostname>
```

分别从路由器和主机上使用 ping 命令

路由器间互 ping:

```
Router1 <—> Router2
```

```
Router2 <—> Router3
```

```
Router1 <—> Router3
```

10. CDP 配置及查看

sh cdp 注意输出的信息

```
conf t
```

```
cdp timer ?
```

```
cdp timer 90
```

sh cdp 注意输出的信息与上一次有何不同

```
sh cdp ?
```

```
sh cdp entry ?
```

```
sh cdp entry *
```

```
sh cdp neighbors
```

11. 配置文件的复制与保存

1) copy running-config startup-config

2) copy startup-config running-config

- 3) erase startup-config
- 4) show startup-config

12. 设置 Telnet 登陆用密码

能进行 telnet 的前提:

- 1) 主机能 ping 通路由器;
- 2) 路由器设置了 telnet 密码;
- 3) 路由器允许通过 telnet 登录;
- 4) 如果需要进入特权模式, 还需要配置 enable 密码。

配置命令: Router# telnet <ip address>

Router# telnet <hostname> // 要先配置 IP host 表, 见 11.<选做实验>

启动 telnet:

Router# config t

Router(config)# line vty 0 4 // 同时允许 0-4 共 5 个连接

Router(config-line)# login //登录

Router(config-line)# password cisco // 设置登录密码为 cisco

设置 enable 密码:

Router(config)#enable password cisco

Router(config-line)#password cisco

第三章 路由器基本命令

3.1 实验前准备

本次实验只涉及基础命令，暂无实验前准备

3.2 实验要求

本次实验,主要完成以下几个基本命令的操作:

1. 设置路由器系统时间:

系统时间是一个非常重要的参数，设备在运行过程中产生的每个日志信息都会有产生的时间作为参考，如果系统时间设置不正确，对于判断网络设备在某个时刻的状态是非常不利的，因此，设备在加电运行的时候,都会设置一个特定的时间,便于随时掌握设备的运行情况。

2. 启动光标跟随服务:

网络管理员在对设备进行配置的时候，设备会不断的弹出控制台信息，告诉网络管理员设备的运行状态，但频繁的控制台信息会打断网络管理员正在输入的命令，给配置带来很大的不便，因此，可以打开光标跟随的功能。这样，即使弹出控制台信息,命令也不会被打断，该服务默认是关闭的。

3. 设置路由器登陆界面:

对于一些商业机构，或者公众网络来说，部分网络设备必须暴露在互联网上，这样一来，除了合法的网络管理员，任何能接入互联网的用户都可以登陆到设备上来，因此，必须设置一个登陆界面，告知用户设备的归属方，该设备所起到的作用以及非法用户登陆所应该承担的法律风险，切忌在登陆界面上出现欢迎字样(内网设备除外)，以防止给入侵的黑客找到入侵的借口。

4. 配置端口描述:

网络管理员在部署大规模的复杂网络时必须规划清楚各个端口连到对端的什么设备，什么端口，以及端口的作用。所以，在初始化配置的时候,端口描述成为必不可少的配置，尤其对于运营商来说,端口描述已经成为配置规范中不可缺少的一部分。

3.3 实验拓扑

一台 PC 通过 Console 线接入设备，本次实验的所有配置都在这样的拓扑下完成。



图 3.1 拓扑图

3.4 实验过程

1. 设置路由器时间，将设备的系统时间设置为 2016 年 1 月 1 日 8 点整

```
Router#clock set 08:00:00 1 jan 2016
```

2. 启动光标跟随功能

```
Router(config)#line con 0
```

```
Router(config-line)#logging synchronous
```

3. 设置路由器登陆界面

```
R1(config)#banner motd "Welcome to NJU"
```

测试结果如图 3.2 所示。

```
Welcome to NJU
User Access Verification
Password:
Router>
```

图 3.2 设置路由器登陆界面

4. 配置端口描述

```
R1(config)#int fa 0/0
```

```
R1(config-if)#description To ISP
```

5. 关闭思科设备的域名解析功能

对于思科的设备，如果在特权模式下，网络管理员不小心输入了错误的命令，那么思科设备会认为这条错误的命令是一个域名，它会做域名解析，如图 3.3 所示：

```
Router#fsdafasdf
Translation "fsdafasdf"...domain server (255.255.255.255)
(255.255.255.255)
Translating "fsdafasdf"...domain server (255.255.255.255)
Translating "fsdafasdf"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
#Router#
```

图 3.3 域名解析

在这个情况下，设备会卡在这里一段时间，这里千万不要按回车键，多按一次回车，域名就多解析一次。这里正确的做法是按 **Ctrl+Shift+6**，打断设备的域名解析，等设备退回到正常的情况下后，再输入下面的命令，关闭设备的域名解析。

```
R1(config)#no ip domain-lookup
```

测试结果如图 3.4 所示。

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z
Router(config)#no ip domain-lookup
Router(config)#fsdafkjs
% Invalid input detected at '^' marker
```

图 3.4 关闭域名解析

6. 将实验端口恢复到默认设置

对端口进行错误的设置之后，需要将其恢复为默认设置，这步操作会清空所有端口下做的配置。所以，在实际工作中，将端口恢复默认设置是一个风险操作，一定要小心谨慎。


```
Router(config)#default interface f0/0
Building configuration...

Interface FastEthernet0/0 set to default configuration
```

图 3.5 恢复默认设置

3.5 实验命令列表

表 3.1 实验命令表

设置系统时间	clock set 时间 日期 月份 年份
设置登陆界面	banner motd 欢迎语
配置端口描述	description 描述信息
关闭域名解析	no ip domain-lookup
端口恢复默认设置	default interface 端口

3.6 实验问题

第四章 路由器密码恢复

第五章 路由器 IOS 备份

5.1 实验前准备

1. 交换机根据收到数据帧中的源 MAC 地址建立该地址同交换机端口的映射，并将其写入 MAC 地址表中。
2. 交换机将数据帧中的目的 MAC 地址同已建立的 MAC 地址表进行比较，以决定由哪个端口进行转发。
3. 如数据帧中的目的 MAC 地址不在 MAC 地址表中，则向所有端口转发。这一过程称为泛洪（flood）。
4. 广播帧和组播帧会被转发到所有端口。

5.2 实验要求

本次实验主要完成以下几个基本命令的操作：

1. 配置 PC 的 IP 地址为 192.168.1.1/24
2. 配置路由器的以太网口的 IP 地址为 192.168.1.254/24
3. 在 PC 上开启 TFTP SERVER
4. 把路由器的 IOS 导入到 PC 上

5.3 实验拓扑

拓扑如图所示：

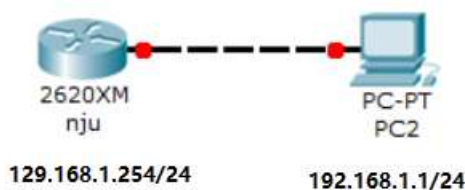


图 5.1 拓扑图

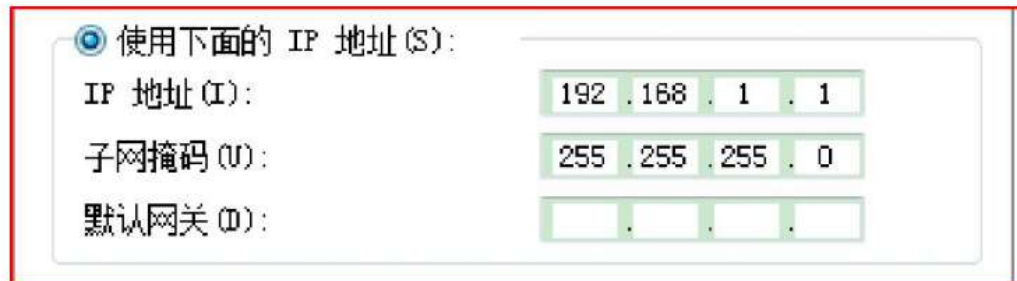
5.4 实验过程

nju 上的配置

```
nju(config)#interface fastEthernet 0/1
nju(config-if)#ip address 192.168.1.254 255.255.255.0
nju(config-if)#no shutdown
```

图 5.2 配置路由器 IP

配置 PC 端的 IP 地址为 192.168.1.1，子网掩码为 255.255.255.0。



The screenshot shows a configuration window with a red border. At the top, there is a radio button labeled "使用下面的 IP 地址(S):" which is selected. Below this, there are three rows of input fields:

- IP 地址(I): 192 . 168 . 1 . 1
- 子网掩码(O): 255 . 255 . 255 . 0
- 默认网关(O): (empty fields)

图 5.3 配置电脑 IP

用路由器 ping 主机，验证 3 层连通性。注意，此时要将主机的防火墙关闭，否则 ping 不通主机。

```
nju#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 second:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

图 5.4 Ping 主机

在 PC 端开启 TFTP 软件，在桌面上点击以下图标。



修改 upload/download 的目录为 D:\IOS。

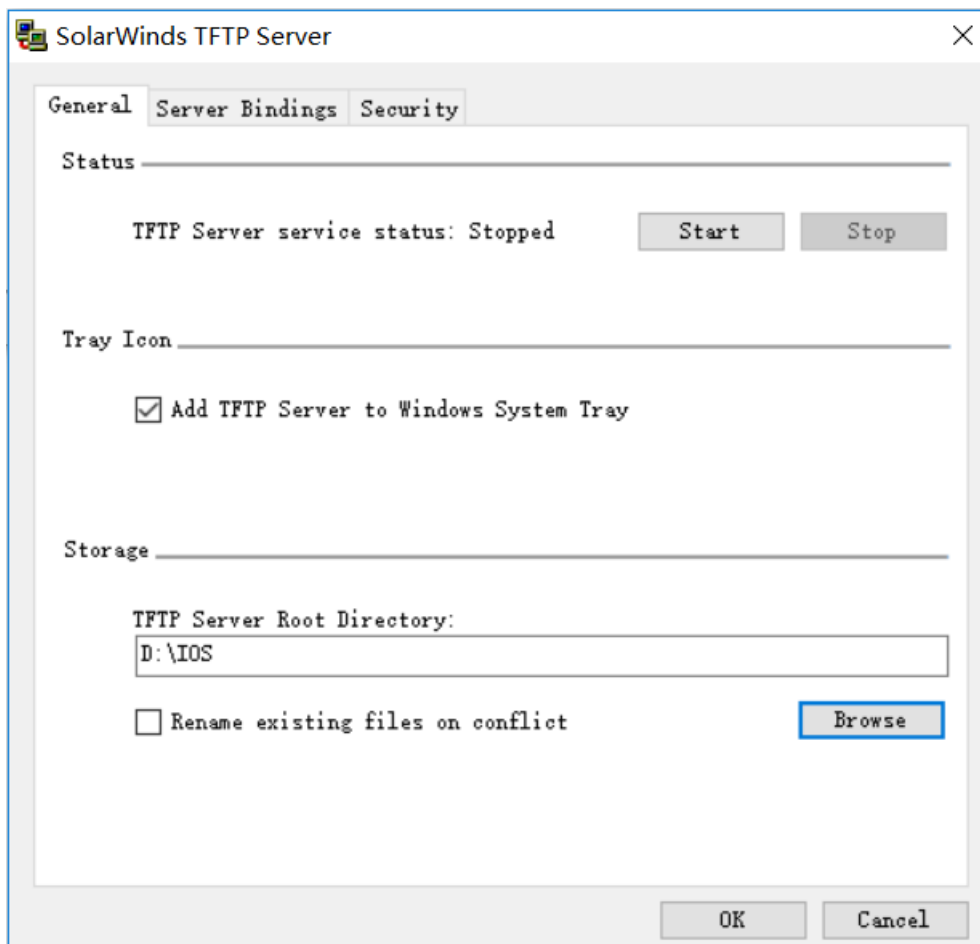


图 5.5 设置 TFTP 保存路径

查看路由器的 IOS 名称。M7.bin 此文件就是路由器的 IOS 名称

```
nju#show flash:
-#- --length-- -----date/time----- path
1      58136556 Feb 22 1907 17:31:44 c2800nm.bin

70598656 byte available (58138624 bytes used)
```

图 5.6 查看 IOS 名称

使用下图命令来备份 IOS。

```
nju#copy flash:c2800nm.bin tftp:
Address or name of remote host []? Ping 192.168.1.1
?Invalid host address or name
%Error parsing filename (Invalid IP address or hostname)
nju#copy flash:c2800nm.bin tftp:
Address or name of remote host []? 192.168.1.1
Destination filename [c2800nm.bin]?
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
58136556 bytes copied in 215.252 secs (270086 bytes/sec)
```

图 5.7 备份 IOS

TFTP Server 显示正在备份的过程。约过 10 分钟后，IOS 备份成功，在 PC 的 D 盘的 IOS 文件夹是否存在备份好的 IOS 文件。

5.5 实验命令列表

表 5.1 实验命令列表

拷贝 IOS	copy flash:[IOS 名称]
--------	---------------------

5.6 实验问题

第六章 交换机基本命令

6.1 实验前准备

交换机根据收到数据帧中的源 MAC 地址建立该地址同交换机端口的映射，并将其写入 MAC 地址表中。交换机将数据帧中的目的 MAC 地址同已建立的 MAC 地址表进行比较，以决定由哪个端口进行转发。如数据帧中的目的 MAC 地址不在 MAC 地址表中，则向所有端口转发。这一过程称为泛洪（flood）。广播帧和组播帧向所有的端口转发。

6.2 实验要求

本次实验主要完成以下几个基本命令的操作：

将一台交换机的 hostname 改成 njc。

1. 将交换机的特权密码设置为 ccna。网管人员连接进入网络设备之后，首先进入的是用户模式，在这个模式下，能使用的命令很少，也无法对网络设备进行配置操作，因此，需要在用户模式下，输入 enable 命令，进入特权模式，在这步操作时，可以设置密码，验证用户身份。增加设备的安全性。
2. 将交换机的 vty 线路密码设置为 ccnp。大多数情况下，网络设备并不在网络管理人员可以接触的地方，因此，有时需要远程登陆到网络设备上进行操作，远程登陆使用的是 VTY 线路，因此，对 VTY 线路设置密码，使得网络管理人员在远程登陆网络设备时需要被验证身份。增加设备的安全性。
3. 给交换机设置管理 IP 地址和网关。路由器属于三层设备，可以通过接口设置 IP 地址，进行远程登录管理设备。交换机需要通过设置管理 IP 地址，使得网络管理人员通过这个地址远程登录管理交换机。
4. 给交换机静态绑定 MAC 地址。交换机在转发数据帧时，通过查找 MAC 地址表进行转发，通过静态绑定 MAC 地址，减少交换机的泛洪的反应时间。

6.3 实验拓扑

拓扑如图所示：



图 6.1 交换机配置拓扑图

6.4 实验过程

1. 将 hostname 改为 nju

```
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname nju
nju(config)#
```

图 6.2 修改 hostname 命令及结果图

2. 设置特权密码和 vty 线路密码

```
nju(config)#enable password ccna
nju(config)#line vty 0 4
nju(config-line)#password ccnp
```

图 6.3 设置密码及结果图

3. 设置管理 ip 地址

```
nju(config)#inter vlan1
nju(config-if)#ip add 192.168.1.1 255.255.255.0
nju(config-if)#no shut
nju(config-if)#exit
nju(config)#ip default-gateway 192.168.1.100
```

图 6.4 设置 ip 地址及结果图

4. 验证实验

vlan1 默认关闭，需要手动打开，设置了管理 ip 地址后，就可以通过远程登录来管理这台 IP 地址了，将 PC 机的 IP 地址设置为 192.168.1.2，然后与交换机的 fa0/1 相连，并打开“开始”菜单——“运行”——“cmd”，接着按照下面给出的 DOS 所显示的信息来验实验。

telnet 登陆后，分别输入相应的 vty 密码和特权密码，即可管理交换机。

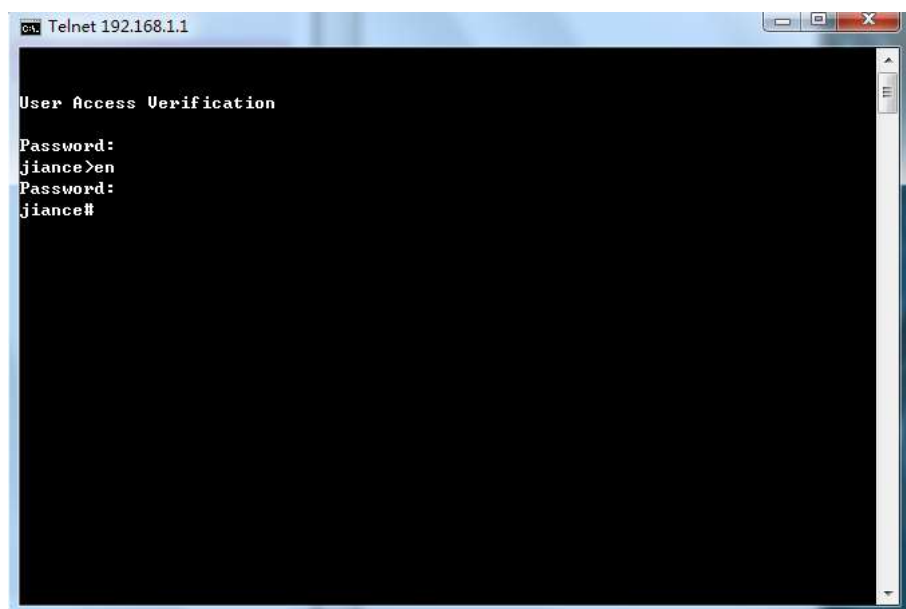


图 6.5 cmd 运行图

6.5 实验命令列表

表 6.1 实验命令列表

进入特权模式	Enable
配置主机名	hostname [hostname]
设置登陆台密码	password [password]
配置 IP 地址	ip address [address]
配置交换机网关	switch(config)#ip default-gateway [address]

6.6 实验问题

第七章 交换机端口安全

7.1 实验前准备

现实生活中，交换机的使用数量远远多于路由器的使用，交换机因为接口数目多，可以连接多个节点，为了保护交换机的安全性，实行了交换机的端口安全，将 MAC 地址进行绑定，提高安全性。

7.2 实验要求

本次实验主要完成以下几项操作：

1. 启用端口安全措施
必须先开启端口安全功能，才能开始制定端口安全策略。
2. 限制 fa0/23 口最大允许访问量为 1
通过限制访问量来保护设备安全。
3. 采用的安全措施为保护，限制或关闭
端口安全检测到问题使用三种惩罚措施。

7.3 实验拓扑

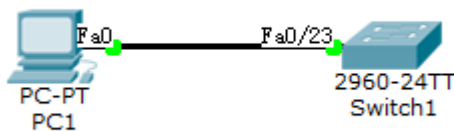


图 7.1

7.4 实验过程

1. 惩罚措施为关闭

```
nju(config)#interface fastEthernet 0/23
nju(config-if)#switchport mode access
nju(config-if)#switchport port-security
nju(config-if)#switchport port-security mac-address aaaa.aaaa.aaaa
nju(config-if)#switchport port-security maximum 1
nju(config-if)#switchport port-security violation shutdown
```

注：惩罚措施有保护、限制和关闭。关闭：当新的计算机接入时，如果该接口的 MAC 地址条目超过了最大数目，则该接口将会被关闭，则这个新的计算机和原来的计算机都无法接入。

试验检测：用一根直通线将 pc 和交换机的 0/23 口相连，查看 0/23 接口的指示灯的变化情况。如果有橙色经过大约 50 秒的时间变为绿色再关闭，说明试验成功。

nju#:sh interface fastEthernet 0/23

在交换机上显示如图：

```
FastEthernet 0/23 is down, line protocol is down (err-disabled)
Hardware is Fast Ethernet, address is ec44.767a.d519 (bia ec44.767a.d519)
```

图 7.2 端口被强制关闭

2. 其他端口测试另外两种惩罚措施的现象

```
nju(config)#interface fastEthernet 0/22
nju(config-if)#switchport mode access
nju(config-if)#switchport port-security
nju(config-if)#switchport port-security mac-address aaaa.aaaa.aaab
nju(config-if)#switchport port-security maximum 1
nju(config-if)#switchport port-security violation protect
```

当新的计算机接入时，如果该接口的 MAC 地址条目超过了最大数目，则该端口将允许已知 MAC 地址发送的数据流但将抛弃未知 MAC 地址发送的数据流；

试验检测：用一根直通线将 pc 和交换机的 0/23 口相连，使用 cmd 发送 ping 命令发送报文至交换机端口 ip: 192.168.1.1，若 mac 地址不是之前设置的 aaaa.aaaa.aaab，则无法成功。结果如图：

```
C: \Users\Administrator>ping 192.168.1.1
正在 Ping 192.168.1.1 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.1.1 的 Ping 统计信息:
    数据包: 已发送=4, 已接收=0, 丢失=4 (100% 丢失)
```

图 7.3 ping 被限制导致失败

当新的计算机接入时，如果该接口的 MAC 地址条目超过了最大数目，则该端口将允许已知 MAC 地址发送的数据流但将抛弃未知 MAC 地址发送的数据流，但同时会发送一条讯息通知违规发生，大致过程与保护类似，不再赘述。

7.5 实验命令列表

表 7.1 实验命令列表

选择交换机端口	interface fastEthernet [端口号]
将端口定义为主机端口	switchport mode access
开启交换机端口安全功能	switchport port-security
绑定 mac 地址到端口上	switchport port-security mac-address [地址]
设置安全访问的最大用户数	switchport port-security maximum [数目]
设置端口安全惩罚措施	switchport port-security violation [措施]

7.6 实验问题

第八章 静态路由和简单组网

8.1 实验前准备

本次实验需要准备三台路由器，并且熟悉路由器的相关配置命令。

8.2 实验要求

本次实验主要完成以下几项操作：

1. 在 Cisco26XX 系列路由器上进行静态路由配置，通过使用静态路由将三台 Cisco26XX 路由器连接起来,组成一个小网络；
2. 练习在简单网络中查看网络和设备状态的各种指令。

8.3 实验拓扑

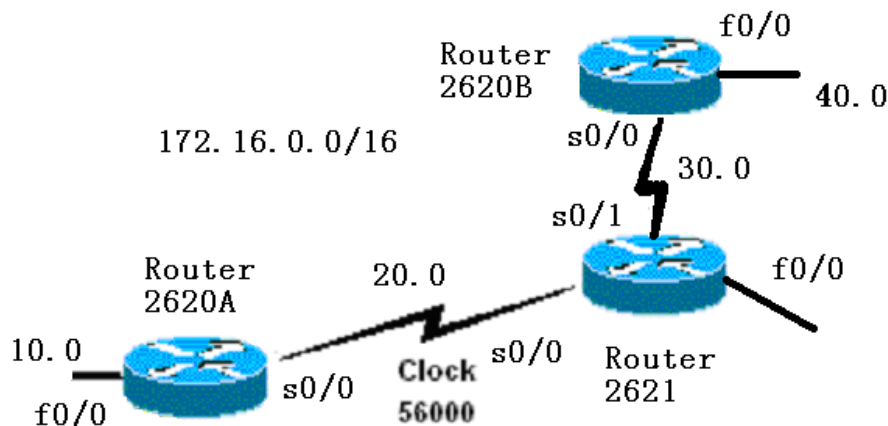


图 8.1

8.4 实验过程(需要区分 DTE/DCE)

1. 端口 IP 地址配置

配置 Router2620A: (f0/0:192.168.10.1,s0/0:192.168.20.1)

```
Router>enable
```

```
Router#config terminal
```

```
Router#hostname Router2620A
```

```
Router2620A(config)#int f0/0
```

```
Router2620A(config-if)#ip address 192.168.10.1 255.255.255.0
```

```
Router2620A(config-if)#no shut
```

```
Router2620A(config-if)#int s0/0
```

```
Router2620A(config-if)#ip address 192.168.20.1 255.255.255.0
```

```
Router2620A(config-if)#no shut
```

配置 Router2621: (s0/0:192.168.20.2,s0/1:192.168.30.1)

```
Router>enable
Router#config terminal
Router#hostname Router2621
Router2621(config)#int s0/1
Router2621(config-if)#ip address 192.168.30.1 255.255.255.0
Router2621(config-if)#no shut
Router2621(config-if)#int s0/0
Router2621(config-if)#ip address 192.168.20.2 255.255.255.0
Router2621(config-if)#clock rate 56000
Router2621(config-if)#no shut
```

配置 Router2620B: (s0/0:192.168.30.2,f0/0:192.168.40.1)

```
Router>enable
Router#config terminal
Router#hostname Router2620B
Router2620B(config)#int s0/0
Router2620B(config-if)#ip address 192.168.30.2 255.255.255.0
Router2620B(config-if)#no shut
Router2620B(config-if)#int f0/0
Router2620B(config-if)#ip address 192.168.40.1 255.255.255.0
Router2620B(config-if)#no shut
```

用 Ping 命令测试各网段的连通性

2. 路由表配置

格式: ip route <目标网段> <子网掩码> <下一跳路由器地址(IP 地址)>

例如:

```
Router2620A(config)#ip route 192.168.40.0 255.255.255.0 192.168.20.2
```

将路由表配置完备后, 用 ping 命令检查各个端口间是否已顺利接通

3. 配置默认路由

对于该实验的拓扑结构来说, 只有 Router1 和 Router3 允许配置默认路由。

首先应该删除静态路由的配置, 才配置默认路由。

以 Router2620A 为例:

```
Router2620A(config)# no ip route 192.168.40.0 255.255.255.0 192.168.20.2
Router2620A(config)# ip route 0.0.0.0 0.0.0.0 192.168.20.2
```

查看路由表 (命令: Router# show ip route)

注: 有*号表示默认路由

8.5 实验命令列表

表 8.1 实验命令列表

路由表配置	ip route [目标网段] [子网掩码] [下一跳路由器地址(IP 地址)]
删除静态路由的配置	no ip route 192.168.40.0 255.255.255.0 192.168.20.2
配置默认路由	ip route 0.0.0.0 0.0.0.0 192.168.20.2
查看路由表	show ip route
停止查看路由表	no debug all

8.6 实验问题

假如只分配了一个网段：192.168.10.0/24，你该如何搭建上述拓扑？请设计并加以实现。

第九章 动态 RIP

9.1 实验前准备

本次实验需要准备三台路由器，并且熟悉路由器的相关配置命令。

9.2 实验要求

本次实验主要完成以下操作：

在 Cisco26XX 系列路由器上，通过使用动态 RIP 路由协议将三台 Cisco26XX 路由器组成一个小网络；

9.3 实验拓扑

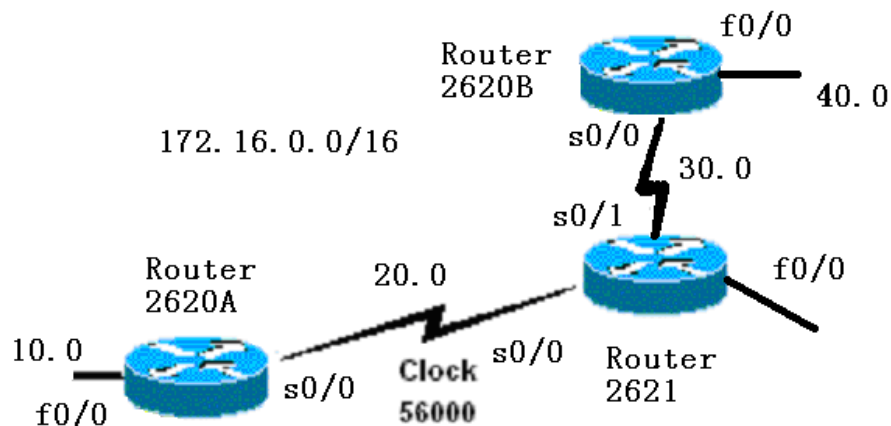


图 9.1

9.4 实验过程

1. 使用 show ip route 查看路由表
检查现在能否从 Router2620A 上 ping 192.168.40.1。

2. 配置动态路由

Router2620A:

```
Router2620A(config)#router rip
Router2620A(config)#network 192.168.10.0
Router2620A(config)#network 192.168.20.0
```

Router2621:

```
Router2621(config)# router rip
Router2621(config)#network 192.168.20.0
Router2621(config)#network 192.168.30.0
```


Router2620B:

Router2620B(config)#router rip

Router2620B(config)#network 192.168.30.0

Router2620B(config)#network 192.168.40.0

3. 使用 show ip route 查看路由表

检查现在能否从 Router2620B 上 ping 192.168.40.1。

4. 使用如下指令查看路由表更新(每 30 秒更新一次)

debug ip rip//开始查看

no debug all//停止查看

9.5 实验命令列表

表 9.1 实验命令列表

路由表配置	ip route [目标网段] [子网掩码] [下一跳路由器地址(IP 地址)]
查看路由表	show ip route
查看路由表更新	debug ip rip
停止查看路由表	no debug all

9.6 实验问题

1. 在配置结束后用什么命令来查看具体的设置,请显示具体内容。

2. 在路由器的全局模式下用“show ip protocol”检查当前时间参数设置，所显示的时间值分别代表什么？

3. 观察网络路由路径的选择

4. 在路由器的全局模式下，“traceroute”命令可用来追踪数据包在网络上所经过的路由。可选择若干条有代表性的路径进行路由选择的跟踪，并将由源到目标的路径的结果记录下来。下表可作为参考格式：

路径编号	源 IP	中间节点 1	中间节点 2	中间节点 3	中间节点 4	目的 IP

第十章 配置单域 OSPF

10.1 实验前准备

OSPF 是一种典型的链路状态路由协议。采用 OSPF 的路由器彼此交换并保存整个网络的链路信息，从而掌握全网的拓扑结构，独立计算路由。因为 RIP 路由协议不能服务于大型网络，所以，IETF 的 IGP 工作组特别开发出链路状态协议——OSPF。目前广为使用的是 OSPF 第二版，最新标准为 RFC2328。

OSPF 作为一种内部网关协议，用于在同一个自治域（AS）中的路由器之间发布路由信息。区别于距离矢量协议（RIP），OSPF 具有支持大型网络、路由收敛快、占用网络资源少等优点，在目前应用的路由协议中占有相当重要的地位。

OSPF 路由器收集其所在网络区域上各路由器的连接状态信息，即链路状态信息（Link-State），生成链路状态数据库（Link-State Database）。路由器掌握了该区域上所有路由器的链路状态信息，也就等于了解了整个网络的拓扑状况。OSPF 路由器利用“最短路径优先算法（ShortestPath First, SPF）”，独立地计算出到达任意目的地的路由。同时，OSPF 协议还引入了“分层路由”的概念，将网络分割成一个“主干”连接的一组相互独立的部分，这些相互独立的部分被称为“区域”（Area），“主干”的部分称为“主干区域”。每个区域就如同一个独立的网络，该区域的 OSPF 路由器只保存该区域的链路状态。每个路由器的链路状态数据库都可以保持合理的大小，路由计算的时间、报文数量都不会过大。

10.2 实验要求

本次实验主要完成以下几个基本命令的操作：

1. 根据拓扑组建和配置网络。配置好网络后，先不要配置 OSPF，先用“ping”命令来核验工作，并测试以太网接口之间的连通性。
2. 为每台路由器配置一个环回接口。将环回接口（而不是物理接口）的地址用作路由器 ID 时，OSPF 将更稳定，因为不同于物理接口，这种接口总是处于活动状态，为不会出现故障，因此再所有重要路由上，都应使用环回接口。
3. 配置 OSPF。可结合使用命令 `router ospf` 和 `network area` 命令。
4. 查看 OSPF 运行情况。使用 `show ip protocols` 命令显示 IP 路由协议参数，包括定时器、过滤器、度量值、网络及路由器的其他信息。使用 `show ip ospf interface` 命令查看接口是否被加入到正确的区域中；该命令还显示各种定时器和邻接关系。
5. 调节 OSPF 的计时器。调节 OSPF 的计时器，以使这些核心路由器能更快地检测出失效的情况，但这会导致额外的数据流量增加。
6. 设置 OSPF 认证。使用接口配置命令 `ip ospf message-digest-key key-id md5 key` 给采用 OSPF MD5 身份验证的路由器指定要使用的密钥 ID 和密钥。

10.3 实验拓扑

拓扑如图所示，此外需要为各台路由器配置环回地址 RTA: 10.0.0.1/32, RTB: 10.0.0.2/32, RTC:10.0.0.3/32。以此为基础配置单区域的 OSPF 网络，即 Area 0 里 OSPF 的配置。

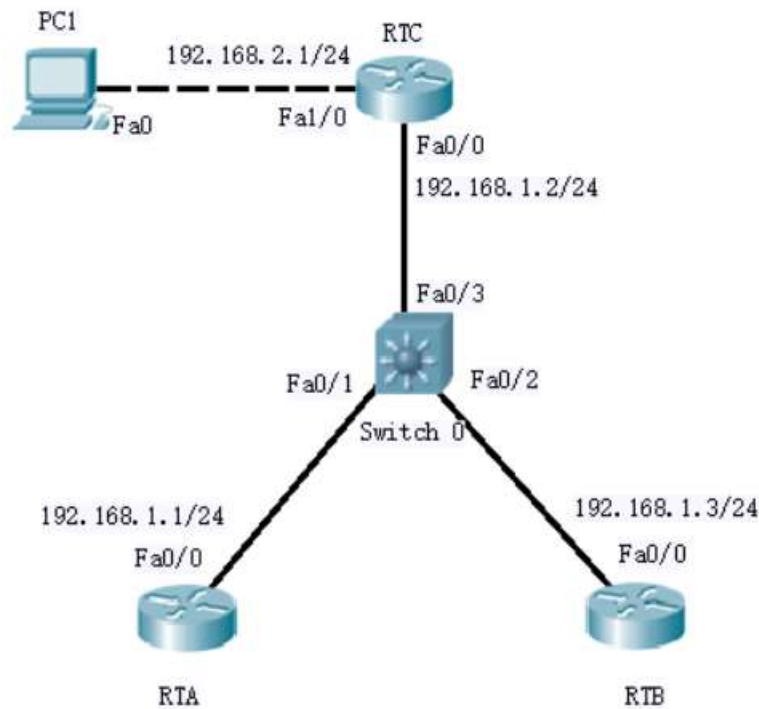


图 10.1 OSPF 配置实验拓扑图

10.4 实验过程

1. 请按照前面几次实验练习的配置方法，根据给出的图示组建和配置网络
2. 在每台路由器上，用一个唯一的 IP 地址配置一个环回接口

RTA(config)#interface lo0

RTA(config-if)#ip address 10.0.0.1 255.255.255.255

RTB(config)#interface lo0

RTB(config-if)#ip address 10.0.0.2 255.255.255.255

RTC(config)#interface lo0

RTC(config-if)#ip address 10.0.0.3 255.255.255.255

3. 在配置了环回接口之后，可以开始配置 OSPF 了：

RTA(config)#router ospf 1

RTA(config-router)#network 192.168.1.0 0.0.0.255 area 0

RTB(config)#router ospf 1

RTB(config-router)#network 192.168.1.0 0.0.0.255 area 0

RTC(config)#router ospf 1

RTC(config-router)#network 192.168.1.0 0.0.0.255 area 0

RTC(config-router)#network 192.168.2.0 0.0.0.255 area 0

4. 用“show”命令来核验它的操作运行。

RTC#show ip protocols

注意，更新计时器被设置为 0。路由更新不是在固定时间间隔上被发送的，它们是事件驱动的。下一步，用“show ip ospf”命令来获得有关 OSPF 进程的消息信息。

```

Routing protocol is "ospf 1"
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Router ID 10.0.0.3
  Number of areas in this router is 1.1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    192.168.1.0 .0.0.0.255 area 0
    192.168.2.0 .0.0.0.255 area 0
  Reference bandwidth unit is 100 mbps
  Routing Information Sources:
Gateway      Distance      Last Update
  Distance: (default is 110)

```

图 10.2 OSPF 进程消息信息

查看 DR/BDR:

```

RTB#show ip ospf interface
FastEthernet0/0 is uop, line protocol is up
  Internet Address 192.168.1.3/24,Area 0
  Process ID 1, Router ID 10.0.0.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.0.0.2, Interface address 192.168.1.2
  Backup Designated router (ID) 10.0.0.2, Interface address 192.168.1.3
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
oob-resync timeout 40
Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 ,sec, maximum is 0 msec
  Neighbor Count is 2,Adjust neighbor count is 2
Adjust with neighbor 10.0.0.1
Adjust with neighbor 10.0.0.3 (Designated Router)

```

图 10.3 DR/BDR 信息

可以看到 DR 为 RTC， BDR 为 RTB

RTA(config)#interface f0/0

RTA(config-if)#ip ospf hello-interval 5

5. 调节 OSPF 的计时器

RTA(config-if)#ip ospf dead-interval 20

6. 设置 OSPF 认证

RTA(config-if)#ip ospf message-digest-key 1 md5 7 itsasecret

RTA(config-if)#router ospf 1
RTA(config-router)#area 0 authentication message-digest

10.5 实验命令列表

表 10.1 实验命令列表

全局配置命令	router ospf [router-id]
接口配置命令	network [ipaddress] [wildcard-mask] area [area-id]
显示 ip 路由协议参数	show ip protocols
显示接口的 ospf 状态	show ip ospf interface
修改 hello 间隔	ip ospf hello-interval [time]
修改 dead 时间	ip ospf dead-interval [time]
配置 ospf MD5 身份	ip ospf message-digest-key [key-id] md5 [key]

10.6 实验问题

哪个路由器成为了 DR? 哪个路由器成为了 BDR?为什么?

第十一章 VLAN 间路由

11.1 实验前准备

目前，很多中小型企业内部网络都是通过交换机互联而成，为了实现广播域的分割和广播包传播范围的控制，划分 Vlan 已成为网络架构中不可缺少的操作，通过划分 Vlan，可以使得同一台交换机下的不同 Vlan 里的端口下连接的设备不能直接互相访问，这样有效的隔离了网络。虽然划分 Vlan 有效的地控制了广播包的传播范围，但是对于某些既希望隔离网络，也希望有些不同的 Vlan 能够通信的企业来讲，Vlan 间路由就成为必要的技术，常常在中小型企业网中部署。为了完成 Vlan 间路由的实验，必须事先掌握 Vlan 的划分，VTP 同步，把接口划分进相应的 Vlan 等交换机的基本操作，以及一个相对比较新的概念——子接口。

11.2 实验要求

首先需要明确一点，不同的 Vlan 相互隔离广播域，因此，传统的以太网 ARP 方式的通信机制在这里是不可用的，需要在网络中添加三层设备，这里的三层设备可以是路由器，也可以是 Cisco 三层交换机（例如 Cisco 3550，Cisco3560）。

本次实验的目的，是让处于不同 Vlan 下的主机能够通信，因此用路由器充当上述的三层设备，需要用到的知识点有：

1. Vlan 的划分
2. VTP 同步
3. 将接口划分进 Vlan
4. Trunk 链路的封装类型
5. 子接口的配置

11.3 实验拓扑

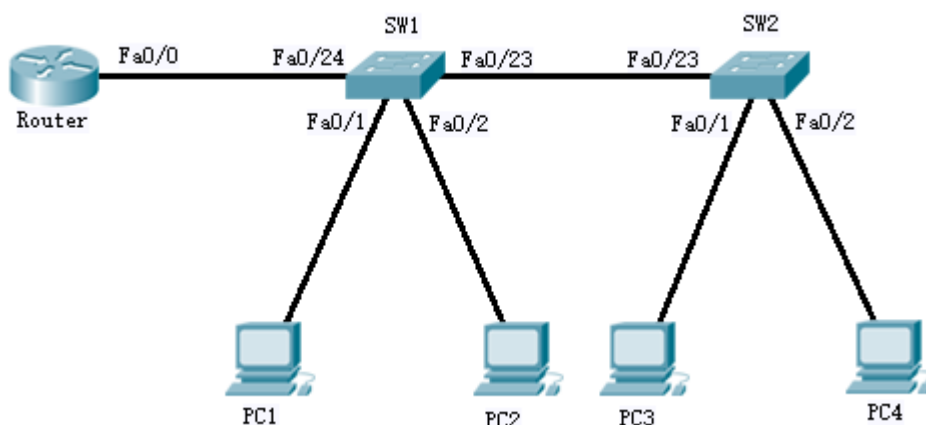


图 11.1 实验拓扑

PC1 和 PC3 属于 Vlan 10, PC2 和 PC4 属于 Vlan 20, 如果上述知识点能够配置正常, 期望的现象应该是 PC1 和 PC3 能够 ping 通 PC2 和 PC4, 同样, PC2 和 PC4 也能够 ping 通 PC1 和 PC3。

11.4 实验过程

首先, 将 SW1 和 SW2 之间的链路设置为 Trunk 链路。

```
sw1(config)#interface fa 0/23
sw1(config-if)#switchport trunk encapsulation dot1q
sw1(config-if)#switchport mode trunk
sw2(config)#interface fa 0/23
sw2(config-if)#switchport trunk encapsulation dot1q
sw2(config-if)#switchport mode trunk
```

图 11.2 设置 Trunk 链路

划分两个 Vlan, Vlan 10 和 Vlan 20

```
sw1(config)#vlan 10
sw2(config)#vlan 20
```

图 11.3 划分两个 Vlan

分别将 SW1 和 SW2 的 fa0/1 口划分入 Vlan 10, fa0/2 口划分入 Vlan 20。

```
sw1(config)#interface fa 0/1
sw1(config-if)#switchport mode access
sw1(config-if)#switchport access vlan 10
sw1(config-if)#exit
sw1(config)#interface fa 0/2
sw1(config-if)#switchport mode access
sw1(config-if)#switchport access vlan 20
```

图 11.4 划分接口

sw2 上进行同样的操作, 操作完成后, 在 sw1 和 sw2 上分别使用 show vlan brief 命令, 查看对应接口是否在正确的 vlan 中。

然后将 SW1 的 fa0/24 接口设置为 Trunk 接口, 与 Router 互联。

```
sw1(config)#interface fa 0/24
sw1(config)#switchport trunk encapsulation dot1q
sw1(config-if)#switchport mode trunk
```

图 11.5 将 SW1 的 fa0/24 接口设置为 Trunk 接口

Router 的 fa0/0 口需要划分两个子接口, 分别对应 Vlan10 和 Vlan20, 作为它们的网关。

```
Router(config)#interface fa 0/0
Router(config-if)#no ip address
Router(config-if)#no shutdown
Router(config)#int fa 0/0.10
Router(config-if)#encapsulation dot1q 10
Router(config-if)#ip address 192.168.10.1 255.255.255.0
Router(config)#int fa 0/0.20
Router(config-if)#encapsulation dot1q 20
```

Router(config-if)#ip address 192.168.20.1 255.255.255.0

图 11.6 划分 router 的两个子接口

测试：PC1 的 IP 地址为 192.168.10.2，网关为 192.168.10.1，PC2 的 IP 地址为 192.168.20.2，网关为 192.168.20.1。

配置正确，PC1 能够 ping 通 PC2。

Reply from 192.168.20.2: bytes=32 time=5ms TTL=127
Reply from 192.168.20.2: bytes=32 time=2ms TTL=127
Reply from 192.168.20.2: bytes=32 time=3ms TTL=127
Reply from 192.168.20.2: bytes=32 time=2ms TTL=127

图 11.7 PC1 ping 通 PC2

如果没有看到上述现象，证明 PC1 和 PC2 无法正常通信，请对照检查配置。

11.5 实验命令列表

表 11.1 实验命令列表

设置 Trunk 封装类型	switchport trunk encapsulation [type]
设置 Trunk 链路	switchport mode trunk
划分 vlan	vlan [vlan name]
将接口划分入 vlan	swichport access vlan [vlan name]
显示 vlan 简要信息	show vlan brief

11.6 实验问题

将主机移动至其他 VLAN 上并且尝试 ping命令， 观察 ping 运行的结果。

第十二章 NAT 网络地址转换

12.1 实验前准备

任何位于内部网络和外部网络之间的设备都可以使用 NAT（RFC3022 对 NAT 进行定义、讲解）。转换的地址不一定必须是私有地址，它可以是任何地址。

1. 需要使用地址转换常见的原因：

- 1) 由于 ISP 没有分配足够的共有 IPv4 地址，不得不使用私有地址；
- 2) 使用了公有地址，但是更换了 ISP，新的 ISP 不再支持这些公有地址；
- 3) 两家公司进行合并，他们使用了相同的地址空间；
- 4) 要将同一个 IP 地址分配给多台机器；

2. NAT 术语：

- 1) 内部本地地址：分配给位于内部网络主机的 IPv4 地址。内部本地地址可能不是由网络信息中心（NIC）或者服务提供商分配的 IPv4 地址。
- 2) 内部全局地址：由网络信息中心（NIC）或者服务提供商分配的合法 IPv4 地址，他对外代表着一个或者多个内部本地 IPv4 地址。
- 3) 外部本地地址：外部主机显示给内网的 IPv4 地址。外部本地地址不一定是合法的地址，它是从可路由地址空间分配到内部网络的地址。
- 4) 外部全局地址：主机所有者分配给外部网络上某一主机的 IPv4 地址。外部全局地址从全局可路由地址或者空间中分配。

3. 地址转换的类型：

- 1) 静态 NAT：将未注册的 IPv4 地址映射到注册的 IPv4 地址（一对一）。在必须从网络外部访问设备时静态 NAT 特别有用。
- 2) 动态 NAT：将未注册的 IPv4 地址与某个注册的 IPv4 地址组中的注册的 IPv4 进行映射。
- 3) 过载 NAT：使用不同的端口号将多个未注册 IPv4 地址映射到单个注册的 IPv4 地址（多对一）。过载也称 PAT，是动态 NAT 的一种形式。

4. NAT 具有以下优势：

- 1) 不需要重新分配所有需要访问外部网络的主机的地址，从而节约时间和金钱。
- 2) 通过应用端口级的多路复用节约了地址。
- 3) 保护网络安全。

12.2 实验要求

本次实验，希望通过地址转换，使拓扑图中左边内部网络中的内部本地地址分别通过三种方式转换成外部全局地址并成功的访问右边网络中的 R3。

12.3 实验拓扑

实验拓扑如图所示，R1 和 R2 之间是 192.168.1.0/24 网段，R2 和 R3 之间是 200.1.1.0/24 网段。

192.168.1.3/24 (辅助地址)
192.168.1.4/24 (辅助地址)

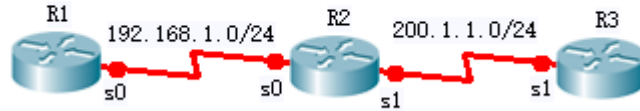


图 12.1 实验拓扑

12.4 实验过程

1. 配置每个设备的名称和接口 ip 地址，确保彼此之间的三层连通性。

```
R1(config)#interface s0/0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip address 192.168.1.3 255.255.255.0 secondary
R1(config-if)#ip address 192.168.1.4 255.255.255.0 secondary
R1(config-if)#no shutdown
```

```
R2(config)#interface s0/0/0
R2(config-if)#ip address 192.168.1.2 255.255.255.0
R2(config-if)#no shutdown
R2(config)#interface s0/0/1
R2(config-if)#ip address 200.1.1.1 255.255.255.0
R2(config-if)#no shutdown
```

```
R3(config)#interface s0/0/1
R3(config-if)#ip address 200.1.1.2 255.255.255.0
R3(config-if)#no shutdown
```

2. 在 R2 上完成静态 NAT 的配置。

```
R2(config)#ip nat inside source static 192.168.1.1 200.1.1.254
*Oct 27 09:04:33.599: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0,
changed state to
R2(config)#interface s0/0/0
R2(config-if)#ip nat inside
R2(config)#interface s0/0/1
R2(config-if)#ip nat outside
R2(config-if)#end
*Oct 27 09:05:32.947: %SYS-5-CONFIG_I: Configured from console by console
R2#debug ip nat
IP NAT debugging is on
R2#_
```

图 12.2 在 R2 上配置静态 NAT

然后在 R1 上用本地地址 192.168.1.1 Ping 200.1.1.2,结果没有 ping 通，为什么？

```
R1#ping 200.1.1.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1#_
```

图 12.3 在 R1 上用本地地址 192.168.1.1 Ping 200.1.1.2

查看 R1 上是否有地址转换的 NAT 表，转换表为空，说明没有发生地址转换，分析原因，R1 去往 200.1.1.0 网段，需要一条静态路由。

```
R1#show ip nat translations

R1#_
```

图 12.4 查看 R1 的 NAT 转换表

为 R1 加上去往 R3 的静态路由，现在 R1 可以 ping 通 R3。

```
R1(config)#ip route 200.1.1.0 255.255.255.0 serial 0/0/0
R1(config)#end
R1#ping 200.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/42/44 ms
R1#_
```

图 12.5 为 R1 加上去往 R3 的静态路由

在 R1 上使用扩展 ping，发送 50 个数据包，默认情况下为 5 个数据包。

```
R1#ping
Protocol [ip]:
Target IP address: 200.1.1.2
Repeat count [5]: 50
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (50/50), round-trip min/avg/max = 40/43/44 ms
R1#_
```

图 12.6 在 R1 上使用扩展 ping

快速切换到 R2 上，来查看具体的转换过程。

```
*Oct 27 09:11:34.791: NAT*: s=192.168.1.1->200.1.1.254, d=200.1.1.2 [5]
*Oct 27 09:11:34.819: NAT*: s=200.1.1.2, d=200.1.1.254->192.168.1.1 [5]
```

图 12.7 R2 终端信息

查看 R2 的 NAT 转换表，R2 建立 NAT 表，当有流量符合这个匹配规则时就会两个地址进行转换。

```
R2#show ip nat translations
```

Pro Inside global	Inside local	Outside local	Outside global
--- 200.1.1.254	192.168.1.1	---	---
R2#_			

图 12.8 查看 R2 的 NAT 转换表

3. 在 R2 上完成动态 NAT 的配置。

将原来的静态 NAT 的条目删除，通过使用用户访问控制列表来定义本地地址池。

```
R2(config)#no ip nat inside source static 192.168.1.1 200.1.1.254
*Oct 27 09:17:07.491: ipnat_remove_static_cfg: id 1, flag
R2(config)#access-list 1 permit 192.168.1.0 0.0.0.255
R2(config)#ip nat pool nju 200.1.1.253 200.1.1.254 p 24
R2(config)#ip nat inside source list 1 pool nju

R2(config)#
*Oct 27 09:19:45.703: ipnat_add_dynamic_cfg_common: id 1, flag 5, range 1
*Oct 27 09:19:45.707: id 1, flags 0, domain 0, lookup 0, aclnum 1, aclname1,mapn
ame idb 0x00000000
*Oct 27 09:19:45.707: poolstart 200.1.1.253 poolend 200.1.1.254
R2(config)#
```

图 12.9 在 R2 上完成动态 NAT 的配置

4. 用 192.168.1.1 ping 200.1.1.2

```
R1#ping
Protocol [ip]:
Target IP address: 200.1.1.2
Repeat count [5]: 50
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (50/50), round-trip min/avg/max = 40/43/44 ms
R1#_
```

图 12.10 ping 200.1.1.2

Ping 通说明路由添加正确，查看 R2 的终端信息。

```
*Oct 27 09:20:54.591: NAT*: s=192.168.1.1->200.1.1.253, d=200.1.1.2 [60]
*Oct 27 09:20:54.619: NAT*: s=200.1.1.2, d=200.1.1.253->192.168.1.1 [60]
```

图 12.11 R2 终端信息

5. 在 R1 上用 192.168.1.3 ping 200.1.1.2

```
R1#ping
Protocol [ip]:
Target IP address: 200.1.1.2
Repeat count [5]: 20
Datagram size [100]:
```

```

Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.1.3
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 20, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (20/20), round-trip min/avg/max = 40/43/44 ms
R1#_

```

图 12.12 ping 200.1.1.2

查看 R2 的终端信息以及 NAT 转换表，源地址 192.168.1.2 转换成 200.1.1.254，很明显调用了第 2 公有地址。

```

*Oct 27 09:26:10.339: NAT*: s=192.168.1.3->200.1.1.254, d=200.1.1.2 [139]
*Oct 27 09:26:10.367: NAT*: s=200.1.1.2, d=200.1.1.254->192.168.1.3 [139]

```

图 12.13 R2 终端信息

```

R2#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 200.1.1.253        192.168.1.1      ---                ---
--- 200.1.1.254        192.168.1.3      ---                ---
R2#_

```

图 12.14 查看 R2 的 NAT 转换表

6. 在 R1 上用 192.168.1.4 ping 200.1.1.2

```

R1#ping
Protocol [ip]:
Target IP address: 200.1.1.2
Repeat count [5]: 20
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.1.4
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 20, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:

```

```
Packet sent with a source address of 192.168.1.4
```

```
.....
```

```
Success rate is 0 percent (0/20)
```

```
R1#_
```

图 12.15 ping 200.1.1.2

结果发现不能 ping 通到目的。查看 R2 的 NAT 转换表，发现没有 192.168.1.4 的条目。

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
---	200.1.1.253	192.168.1.1	---	---
---	200.1.1.254	192.168.1.3	---	---

```
R2#_
```

图 12.16 查看 R2 的 NAT 转换表

解决的方法：清除 R2 的 NAT 表中的条目，将公有地址池中的公有地址释放出来。

```
R2#clear ip nat translation *
```

```
R2#show ip nat translations
```

```
R2#_
```

图 12.17 清除 R2 的 NAT 表中的条目

在 R1 上重试。

```
R1#ping
```

```
Protocol [ip]:
```

```
Target IP address: 200.1.1.2
```

```
Repeat count [5]:
```

```
Datagram size [100]:
```

```
Timeout in seconds [2]:
```

```
Extended commands [n]: y
```

```
Source address or interface: 192.168.1.4
```

```
Type of service [0]:
```

```
Set DF bit in IP header? [no]:
```

```
Validate reply data? [no]:
```

```
Data pattern [0xABCD]:
```

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

```
Sweep range of sizes [n]:
```

```
Type escape sequence to abort.
```

```
Sending 20, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
```

```
Packet sent with a source address of 192.168.1.4
```

```
!!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/44 ms
```

```
R1#_
```

图 12.18 在 R1 上重试

R2 终端上所显示的转换过程。

```
*Oct 27 09:37:24.699: NAT*: s=192.168.1.4->200.1.1.253, d=200.1.1.2 [170]
```

```
*Oct 27 09:37:24.727: NAT*: s=200.1.1.2, d=200.1.1.253->192.168.1.4 [170]
```

图 12.19 R2 终端信息

再查看 R2 的 NAT 转换表。

R2#show ip nat translations			
Pro	Inside global	Inside local	Outside local Outside global
icmp	200.1.1.253:7	192.168.1.4:7	200.1.1.2:7 200.1.1.2:7
---	200.1.1.253	192.168.1.4	--- ---
R2#_			

图 12.20 R2 的 NAT 转换表

7. 配置 PAT

先删除转换语句，再删除之前建立的 pool，注意删除的顺序。

```
R2(config)#no ip nat inside source list 1 pool nju

Dynamic mapping in use, do you want to delete all entires? [no]: yes
R2(config)#no ip nat pool nju 200.1.1.253 200.1.1.254 prefix-length 24
R2(config)#ip nat pool nju 200.1.1.253 200.1.1.253 prefix-length 24
R2(config)#ip nat inside source list 1 pool nju overload

R2(config)#
*Oct 27 09:42:38.571: ipnat_add_dynamic_cfg_common: id 2,flag 5, range 1
*Oct 27 09:42:38.571: id 2, flags 0, domain 0, lookup 0, aclnum 1, aclname 1, map
name idb 0x00000000
*Oct 27 09:42:38.571: poolstart 200.1.1.253 poolend 200.1.1.253 _
```

图 12.21 配置 PAT

8. 在 R1 用 192.168.1.1 上 ping 200.1.1.2

```
R1#ping 200.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =44/44/44 ms
R1#_
```

图 12.22 ping 200.1.1.2

查看 R2 的终端信息以及 NAT 转换表，随机产生端口号 6。

```
*Oct 27 09:44:05.283: NAT*: s=192.168.1.1->200.1.1.253, d=200.1.1.2 [175]
*Oct 27 09:44:05.311: NAT*: s=200.1.1.2, d=200.1.1.253->192.168.1.1 [175]
```

图 12.23 R2 终端信息

R2#show ip nat translations				
Pro	Inside global	Inside local	Outside local	Outside global
icmp	200.1.1.253:6	192.168.1.1:6	200.1.1.2:6	200.1.1.2:6
R2#_				

图 12.24 查看 R2 的 NAT 转换表

R2 约 1 分钟的时间释放地址转换的空间，此时查找 NAT 表中没有任何的转换条目。

```
R2#show ip nat translations

R2#_
```

图 12.25 查看 R2 的 NAT 转换表

9. 在 R1 用 192.168.1.3 ping 200.1.1.2

```
R1#ping
Protocol [ip]:
Target IP address: 200.1.1.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.1.3
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 20, 100-byte ICMP Echos to 200.1.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.3
.....
Success rate is 0 percent (5/5), round-trip min/avg/max = 40/43/44 ms
R1#_
```

图 12.26 ping 200.1.1.2

查看 R2 的终端信息。

```
*Oct 27 09:47:40.827: NAT*: s=192.168.1.1->200.1.1.253, d=200.1.1.2 [180]
*Oct 27 09:47:40.855: NAT*: s=200.1.1.2, d=200.1.1.253->192.168.1.3 [180]
```

图 12.27 R2 的终端信息

端口号已改为 9。

```
R2#show ip nat translations
Pro    Inside global    Inside local      Outside local     Outside global
icmp   200.1.1.253:9    192.168.1.1:9    200.1.1.2:9      200.1.1.2:9
R2#
*Oct 27 09:48:41.467:  NAT: expiring 200.1.1.253(192.168.1.3) icmp 9 (9)
R2#
```

图 12.28 查看 R2 的 NAT 转换表

12.5 实验命令列表

表 12.1 实验命令列表

配置静态 NAT	ip nat inside source static [inside local ip address] [inside global ip address]
删除静态 NAT 条目	no ip nat inside source static [inside local ip address] [inside global ip address]
指定内部 ip 地址接口	ip nat inside

指定外部 ip 地址接口	ip nat outside
查看 NAT 转换表	show ip nat translations
清空 NAT 转换表	clear ip nat translation *

12.6 实验问题

第十三章 ACL 实验

13.1 实验前准备

ACL 访问控制列表是路由器和交换机接口的指令列表，用来控制端口进出的数据包。ACL 适用于所有的被路由协议。配置 ACL 后，可以限制网络流量，允许特定设备访问，指定转发特定端口数据包等。目前有两种主要的 ACL：标准 ACL 和扩展 ACL。扩展 ACL 相比标准 ACL 提供了更广泛的控制范围。

13.2 实验要求

本次试验，两台路由器通过 s0/0/0 口互联，hostname 分别为 R1 和 R2，两设备采用 192.168.1.0/24 网段。实现标准 ACL 和扩展 ACL 的实验，并对两种 ACL 进行比较。

13.3 实验拓扑

拓扑如图 13.1 所示：

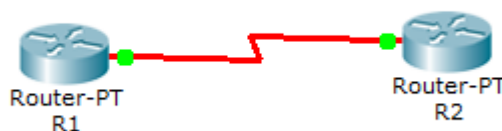


图 13.1 实验拓扑图

13.4 实验过程

首先为了区分两台路由器，分别将设备的 hostname 改为 R1，R2，在相应接口下配置 IP 地址以及时钟速率，以保证 3 层连通性，具体配置如下：

```
Router>enable
Router#conf terminal
Router(config)#hostname R1
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#clock rate 64000

Router(config)#hostname R2
R2(config)#interface serial 0/0/0
R2(config-if)#ip address 192.168.1.2 255.255.255.0
R2(config-if)#no shutdown
```

1. 使用扩展的 ACL 封杀 R1 到 R2 的 PING 命令

1) 验证 3 层连通性

```
R2#ping 192.168.1.1

Type escape sequence to abort.
Sending 5,100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent(5/5), round-trip min/avg/max = 28/28/28 ms
```

图 13.2 验证是否联通

2) 创建 ACL

```
R1(config)#access-list 100 deny icmp 192.168.1.1 0.0.0.0 192.168.1.2 0.0.0.0
R1(config)#access-list 100 permit ip any any
R1#show ip access-lists
Extended IP access list 100
deny icmp host 192.168.1.1 host 192.168.1.2
permit ip any any
```

3) 应用 ACL 到接口

```
R1(config)#interface serial 0/0/0
R1(config-if)#ip access-group 100 out
```

4) 验证效果

```
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 100 out
R1(config-if)#^Z
R1#p

*Oct 27 09:53:14.979:%SYS-5-CONFIG_I: Configured from console by consoleing
.168.1.2

Type escape sequence to abort.
Sending 5,100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent(5/5), round-trip min/avg/max = 28/28/32 ms
R1#ping 192.168.1.2

Type escape sequence to abort.
Sending 5,100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent(5/5), round-trip min/avg/max = 28/28/32 ms
```

图 13.3 实验效果

对于 ACL 的放置位置，有以下原则：扩展 ACL 放置在靠近源的位置，标准 ACL 放置在靠近目的位置。那按照上述的原则，创建一个扩展的 ACL，并放置在源端，并没有错误。

5) 排错

```
R1#show ip access-lists
```

```
Extended IP access list 100
deny icmp host 192.168.1.1 host 192.168.1.2
permit ip any any (15 matches)
```

问题分析：最后一条语句匹配到 15 个数据包。对于 ACL，有个非常重要的特性，他不能过滤本地数据流！也就是说，对于 R1 上发送的数据，设置在 R1 接口上的 ACL 并不能对它进行过滤。为了能对数据流进行过滤，需要把 ACL 设置在对端的 R2 上。

6) 在 R2 上设置并应用 ACL

```
R2(config)#access-l 100 deny icmp host 192.168.1.1 host 192.168.1.2
R2(config)#access-l 100 permit icmp any any
R2(config)#interface serial 0/0/0
R2(config-if)#ip access-group 100 in
```

7) 检测效果

```
R1#ping 192.168.1.2

Type escape sequence to abort.
Sending 5,100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
U.U.U
Success rate is 0 percent(0/5)
show ip access-lists
Extended IP access list 100
    10 deny icmp host 192.168.1.1 host 192.168.1.2 (15matches)
    20 permit icmp any any
```

图 13.4 ping 测试

实验成功，ICMP 包被拒绝，如图 13.4 所示。

2. 使用 ACL 禁止 R1 到 R2 的 TELNET 应用

注意：在进行第二部分实验前请将第一部分配置清除

路由器名为:R1、R2。R1 的 S0/0/0 端口与 R2 的 S0/0/0 端口相连，IP 地址和第一部分相同,在 R2 上设置特权密码为 nju，线路密码为 cisco。从 R1 使用 PING 命令测试到 R2 的连通性，结果可达，但却不可以 TELNET 到 R2。

有两种方法可以实现这样的操作。

方法一:使用扩展 ACL

1) 检测基本配置

```
R2(config)#enable secret nju
R2(config)#line vty 0 4
R2(config-line)#password cisco
R2(config-line)#login
```

```
R1#ping 192.168.1.2

Type escape sequence to abort.
Sending 5,100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent(5/5), round-trip min/avg/max = 28/28/28 ms
R1#telnet 192.168.1.2
```

```
Trying 192.168.1.2 ... Open
```

```
User Access Verification
```

```
Password:
```

```
R2>enable
```

```
Password:
```

```
R2#exit
```

图 13.5 检验基本配置

2) 创建 ACL

```
R2(config)#access-list 101 deny tcp host 192.168.1.1 any eq 23
```

```
R2(config)#access-list 101 permit ip any any
```

```
R2(config)#int s0/0/0
```

```
R2(config-if)#ip access-group 101 in
```

```
R1#telnet 192.168.1.2
```

```
Trying 192.168.1.2 ...
```

```
% Destination unreachable; gateway or host down
```

```
R1#ping 192.168.1.2
```

```
Type escape sequence to abort.
```

```
Sending 5,100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent(5/5), round-trip min/avg/max = 28/28/32 ms
```

图 13.6 测试结果

telnet 被拒绝但 ping 成功。

方法二:使用标准 ACL

1) 删除先前的配置

```
R2(config)#int s0/0/0
```

```
R2(config-if)#no ip access-group 101 in
```

2) 创建 ACL

```
R2(config)#access-list 1 deny host 192.168.1.1
```

```
R2(config)#access-list 1 permit any
```

3) 应用 ACL

```
R2(config)#line vty 0 4
```

```
R2(config-line)# access-class 1 in
```

实验结果如图 13.7 所示:

```
R1#telnet 192.168.1.2
```

```
Trying 192.168.1.2 ...
```

```
% Connection refused by remote host
```

图 13.7 使用标准 ACL

问题分析: 设置了 2 种 ACL, 但是得到的效果却不一样。如果是使用了扩展的 ACL, 那么它的提示是“% Destination unreachable; gateway or host down”, 说明 23 号端口根本不可达。如果是使用了标准的 ACL 放置在 VTY 线路中, 则提示“% Connection refused by

remote host”，说明的确是到达了 23 号端口，只不过被拒绝了。在实际使用中最好使用扩展的 ACL，减少 23 号端口的负担。

13.5 实验命令列表

表 13.1 实验命令列表

配置 access list	access-list [list number] [permit deny] [source address] [address] [wildcard mask] [log]
将指定访问列表应用到相关接口，并指定 ACL 作用的方向	ip access-group {[access-list-number]}{[name]} [int out]
显示已设置的访问控制列表内容	show ip access-lists

13.6 实验问题

第十四章 PPP 验证实验

14.1 实验前准备

在现实生活中，上网需要先向电信和联通这种运营商去缴费申请一个用户名和密码，然后通过登录连接到互联网。本章实验描述了如何通过 PPP 的验证使链路能够通畅。

14.2 实验要求

本次实验主要完成以下两项操作：

PAP 验证

1. 为路由器指定唯一主机名，为了辨别设备，需要设置不同的主机名。
2. 列出认证路由器时所使用的远程主机名称和口令，PAP 为单项验证，所以被验证方需要正确的主机名和口令，方能使链路通畅。
3. WAN 接口上完成 PPP 协议的封装。PAP 验证基于 PPP 协议，必须封装后才能启用。
4. nju1 为服务端，nju2 为客户端，客户端主动向服务端发出认证请求，密码设置为 ccna。

CHAP 验证

1. 为路由器指定唯一主机名，为了辨别设备，需要设置不同的主机名
2. 列出认证路由器时所使用的远端主机名称和口令，密码为 ccna。CHAP 为双向验证，通过将对方发送的用户名和口令与本地的用户列表来对比确认。一致通过，不一致链路阻塞。WAN 接口上完成 PPP 协议的封装和 CHAP 认证的配置 PAP 验证基于 PPP 协议，必须封装后才能启用。

14.3 实验拓扑

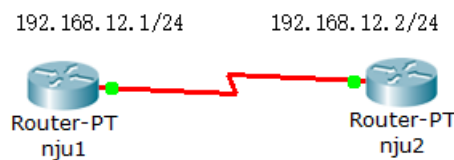


图 14.1 拓扑图

14.4 实验过程

14.4.1 PAP 验证

1. 配置 nju1 的服务端设置

```
Router(config)#hostname nju1
nju1(config)#username nju password ccna
nju1(config)#interface serial 0/0/0
nju1(config-if)#ip address 192.168.12.1 255.255.255.0
```

```
nju1(config-if)#encapsulation ppp
nju1(config-if)#ppp authentication pap
nju1(config-if)#no shutdown
```

2. 配置 nju2 的客户端设置

```
Router(config)#hostname nju2
nju2(config)#interface serial 0/0
nju2(config-if)#ip address 192.168.12.2 255.255.255.0
nju2(config-if)#clock rate 64000
nju2(config-if)#encapsulation ppp
nju2(config-if)#no shutdown
```

3. client 端发送用户名和密码

```
nju2(config-if)#ppp pap sent-username adsf password adsf
```

注：当用户名和口令中的任意一个和验证方的本地用户列表不同时，无法通信。

```
nju2#ping 192.168.12.1
```

测试结果如图：

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

图 14.2 ping 失败

4. 设置正确的用户名和密码

```
nju2(config-if)#ppp pap sent-username nju password ccna
nju2(config-if)#end
nju2#ping 192.168.12.1
```

测试结果如图：

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

图 14.3 ping 成功

14. 4. 2 CHAP 验证

1. 配置 nju1

```
Router(config)#hostname nju1
nju1(config)#username nju2 password ccna
nju1(config)#interface serial 0/0
nju1(config-if)#ip address 192.168.12.1 255.255.255.0
nju1(config-if)#encapsulation ppp
nju1(config-if)#ppp authentication chap
nju1(config-if)#no shutdown
```


2. 配置 nju2

```
Router(config)#hostname nju2
nju2(config)#inter serial 0/0
nju2(config-if)#ip address 192.168.12.2 255.255.255.0
nju2(config-if)#clock rate 64000
nju2(config-if)#encapsulation ppp
nju2(config-if)#ppp authentication chap
nju2(config-if)#no shutdown
```

注：对端需要配置相同，因为 chap 是双向认证，由于一端没有发送本地用户名和列表，导致链路不通。

3. 设置 nju2 上的用户名和密码

```
nju2(config)#username nju1 password ccnp
nju2#ping 192.168.12.1
```

测试结果如图：

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

图 14.4 ping 失败

4. 设置正确的用户名和密码

```
nju2(config)#username nju1 password ccna
nju2#ping 192.168.12.1
```

测试结果如图：

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

图 14.5 ping 成功

在验证通过的情况下，将任意一边的口令随意设置成一个非 ccna 的口令，再测试连通性。

```
nju2(config)#username nju1 password ccnp
nju2#ping 192.168.12.1
```

测试结果如图：

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

图 14.6 ping 仍然成功

注：因为当验证通过后会一直保存已经建立好的连接，解决方法是将接口关闭在启动。

14.5 实验命令列表

表 14.1 实验命令列表

启用 PPP 封装协议	encapsulation ppp
启用 PAP 身份验证	ppp authentication pap
设置被验证发送的用户名和 口令	ppp pap sent-username [用户名] password [密码]
启用 CHAP 认证协议	ppp authentication chap

14.6 实验问题

第十五章 帧中继实验

15.1 实验前准备

帧中继是由 ITU-T 标准化的高性能 WAN 协议，并在美国广泛应用。帧中继是一种面向连接的数据链路层技术，它定义了路由器与服务提供商的本地接入交换设备之间的互联过程。

连接到帧中继 WAN 的设备分为以下两类：

DTE： DTE 设备通常位于客户所在地并且可能为客户所有，帧中继接入设备、路由器、网桥都属于 DTE 设备。

DCE： 运营商所拥有的网间设备，DCE 设备的作用是在网络中提供时钟服务和交换服务，并通过 WAN 传输数据。

15.2 实验要求

本次实验主要完成以下几个要求：

1. 配置帧中继交换机。帧中继是一种面向连接的数据链路层技术，它定义了路由器与服务提供商的本地接入交换设备之间的互联过程。
2. 按照拓扑组建和配置路由器。
3. 进行验证实验。通过命令查看在帧中继交换机上虚电路交换的过程；通过手动配置 DLCI 号与 IP 地址的映射，在路由器上分别禁用逆向 ARP 查询。

15.3 实验拓扑

拓扑如图所示：

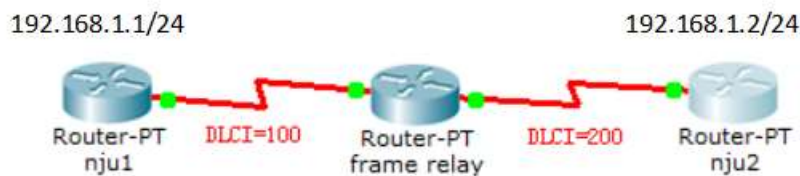


图 15.1 帧中继实验拓扑图

15.4 实验过程

1. 配置中间的帧中继交换机

```
Fr-sw(config)#frame-relay switching
Fr-sw(config)#interface serial 0/0/0
Fr-sw(config-if)#encapsulation frame-relay
Fr-sw(config-if)#frame-relay intf-type dce
Fr-sw(config-if)#clock rate 64000
```

```
Fr-sw(config-if)#frame-relay route 100 interface serial 0/0/1 200
Fr-sw(config-if)#no shutdown
Fr-sw(config-if)#exit
```

```
Fr-sw(config)#interface serial 0/0/1
Fr-sw(config-if)#encapsulaiton frame-relay
Fr-sw(config-if)#frame-relay intf-type dce
Fr-sw(config-if)#clock rate 64000
Fr-sw(config-if)#frame-relay route 200 interface serial 0/0/0 100
Fr-sw(config-if)#no shutdown
```

2. 配置 nju1

```
nju1(config)#interface serial 0/0/1
nju1(config-if)#ip address 192.168.1.1 255.255.255.0
nju1(config-if)#encapsulation frame-relay
nju1(config-if)#no shutdown
```

3. 配置 nju2

```
nju2(config)#interface serial 0/0/0
nju2(config-if)#ip address 192.168.1.2 255.255.255.0
nju2(config-if)#encapsulation frame-relay
nju2(config-if)#no shutdown
```

4. 验证实验

```
nju2#ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100 type percent (5/5), round-trip min/avg/max=56/56/60 ms
```

图 15.2 检测连通性

```
nju2#show frame-relay map
Serial0/0/1 (up): ip 192.168.1.1 dlci 100(0*64,0*1840), dynamic,
```

图 15.3 终端显示信息

通过命令可以查看在帧中继交换机上虚电路交换的过程。从接口 s0/0/1 的 200 虚电路交换到 s0/0/0 的 100 的虚电路。

```
Fr-sw#show frame-relay route
```

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial0/0/0	100	Serial0/0/1	200	active
Serial0/0/1	200	Serial0/0/0	100	active

图 15.4 终端显示信息

路由器的虚电路 200 在交换机 s0/0/1 上，这样从交换机 s0/0/1 过来的数据就会发送给路由器的 s0/0/0 上。

```
nju1#show frame-relay pvc
```

PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)

	Active	Inactive	Deleted	Static
Local	1	0	0	0
Switched	0	0	0	0
Unused	0	0	0	0

DLCI = 200, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0

```
input pkts 16      output pkts 16      in bytes 1594
out bytes 1594     dropped pkts 0      in pkts dropped 0
out pkts dropped 0 out bytes dropped 0
in FECN pkts 0    in BECN pkts 0      out FECN pkts 0
out BECN pkts 0   in DE pkts 0      out DE pkts 0
out va=casj=t okts 1 out bcast bytes 34
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:08:30, last time pvc status changed 00:08:20
```

图 15.5 终端显示信息

也可以在 njul 和 nju2 上分别禁用逆向 ARP 查询，手动配置 DLCI 号与 IP 地址的映射。

nju1 的配置：

```
nju1(config-if)#no frame-relay inverse-arp
```

```
nju1(config-if)#frame-relay map ip 192.168.1.1 100 broadcast
```

nju2 的配置：

```
nju2(config-if)#no frame-relay inverse-arp
```

```
nju2(config-if)#frame-relay map ip 192.168.1.2 200 broadcast
```

实验结果：

```
nju1#show frame-relay map
```

```
Serial0/0/0 (up) : ip 192.168.1.2 dlci 200 (0xC8,0*2080), static,
                    broadcast,
                    CISCO, status defined, active
```

图 15.6 终端显示信息

15.5 实验命令列表

表 15.1 实验命令列表

把路由器当成帧中继交换机	frame-relay switching
接口封装成帧中继	encapsulation frame-relay
配置接口是帧中继的 DCE 还是 DTE	frame-relay intf-type dce
配置帧中继交换表	frame-relay route
显示帧中继交换表	show frame-relay route
显示帧中继 PVC 状态	show frame pvc
查看帧中继映射	show frame-relay map
关闭帧中继自动映射	no frame-relay inverse-arp

15.6 实验问题

第十六章 DHCP 欺诈保护

16.1 实验前准备

DHCP 的主要作用是给网络中的其他设备动态分配 IP 地址，从而节约 IP 资源。

DHCP 欺骗：攻击者可以通过伪造大量的 IP 请求包，消耗掉现有 DHCP 服务器的 IP 资源。当有计算机请求 IP 时，DHCP 服务器就无法分配 IP。此时，攻击者可以伪造一个 DHCP 服务器为计算机分配 IP，并指定一个虚假的 DNS 服务器地址。这时，当用户访问网站的时候，就被虚假 DNS 服务器引导到错误的网站。

在交换机上开启 DHCP snooping 功能，绑定并过滤不信任的 DHCP 信息可以防止 DHCP 欺骗。对于信任端口收到的 DHCP 服务器报文，交换机不会丢弃而直接转发，来自非信任端口的 DHCP 报文则无法通过，从而有效的防止了 DHCP 欺骗。

16.2 实验要求

本次实验要求在路由器上启用 DHCP 服务为两台计算机动态分配 IP。此外，还需配置交换机的 DHCP snooping 功能防止 DHCP 欺骗。

16.3 实验拓扑

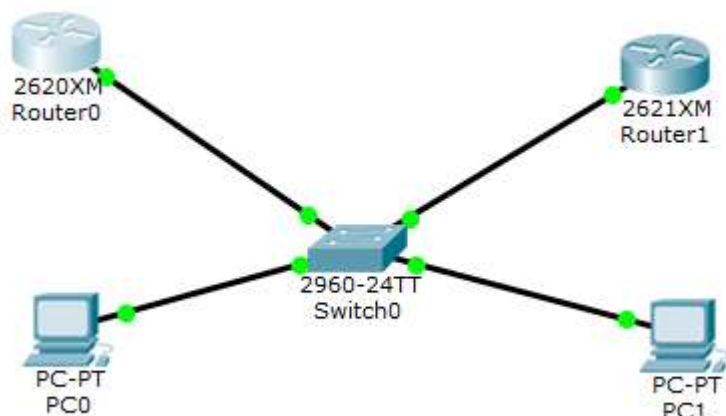


图 16.1 实验拓扑

16.4 实验过程

1. 配置路由器 Router1 的 DHCP 功能。

```
Router1(config)#service dhcp
Router1(config)#ip dhcp pool nju1
Router1(dhcp-config)#network 10.1.1.0 255.255.255.0
Router1(dhcp-config)#default-router 10.1.1.1
Router1(dhcp-config)#dns-server 10.1.1.1
```

Router1(config)#ip dhcp excluded-address 10.1.1.1 10.1.1.10

2. 设置计算机 ip 获取为 DHCP

检查计算机 IP 并在 Router1 上查看地址分配，如图 16.2 所示。

Router#show ip dhcp binding			
IP address	Client-ID/ Hardware address	Lease expiration	Type
10.1.1.21	00E0.A3A5.4929	- -	Automatic
10.1.1.22	00D0.BAD4.D490	- -	Automatic

图 16.2 dhcp 地址分配信息

3. 防止 DHCP 欺骗

按照步骤 1 配置 Router0，将 Router0 的 DHCP 地址池设置为 20.1.1.0

配置交换机 snooping 功能，将与 Router1 相连的 f0/2 端口设置为信任端口

Switch(config)#ip dhcp snooping

Switch(config)#ip dhcp snooping vlan 1

Switch(config)#int f0/2

Switch(config-if)#ip dhcp snooping trust

配置路由器 Router1

Router1(config)#ip dhcp relay information trust-all

Switch#show ip dhcp snooping	
Switch DHCP snooping is enabled	
DHCP snooping is configured on following VLANs:	
1	
Insertion of option 82 is enabled	
Option 82 on untrusted port is not allowed	
Verification of hwaddr field is enabled	
Interface Trusted Rate limit (pps)	

FastEthernet0/1 no unlimited	
FastEthernet0/2 yes unlimited	
FastEthernet0/3 no unlimited	
FastEthernet0/4 no unlimited	

图 16.3 信任 f0/2 端口

此时，两台计算机 IP 地址均由 Router1 分配，IP 地址如图 16.4 和 16.5 所示。

C:\>ipconfig	
FastEthernet0 Connection:(default port)	
Link-local IPv6 Address.....	FE80::2D0:BAFF:FED4:D490
IP Address.....	10.1.1.22
Subnet Mask.....	255.255.255.0
Default Gateway.....	10.1.1.1

图 16.4 计算机 0IP 地址


```
C:\>ipconfig

FastEthernet0 Connection:(default port)

    Link-local IPv6 Address.....: FE80::2E0:A3FF:FEA5:4929
    IP Address.....: 10.1.1.21
    Subnet Mask.....: 255.255.255.0
    Default Gateway.....: 10.1.1.1
```

图 16.5 计算机 1IP 地址

将交换机 f0/1 设置为信任端口，f0/2 设置为非信任端口。

```
Switch(config)#int f0/2
Switch(config-if)#no ip dhcp sn
Switch(config-if)#no ip dhcp snooping trust
Switch(config)#int f0/1
Switch(config-if)#ip dhcp snooping trust
Switch(config-if)#end
```

再次检查两台计算机的 IP 地址，如图 16.6 和 16.7 所示。

```
C:\>ipconfig

FastEthernet0 Connection:(default port)

    Link-local IPv6 Address.....: FE80::2E0:A3FF:FEA5:4929
    IP Address.....: 20.1.1.17
    Subnet Mask.....: 255.255.255.0
    Default Gateway.....: 20.1.1.1
```

图 16.6 计算机 0IP 地址

```
C:\>ipconfig

FastEthernet0 Connection:(default port)

    Link-local IPv6 Address.....: FE80::2D0:BAFF:FED4:D490
    IP Address.....: 20.1.1.18
    Subnet Mask.....: 255.255.255.0
    Default Gateway.....: 20.1.1.1
```

图 16.7 计算机 1IP 地址

可以看到，通过交换机 snooping 的配置，能够阻止非信任端口的 DHCP 报文传输，从而避免 DHCP 欺骗。

16.5 实验命令列表

表 16.1 实验命令列表

打开 dhcp 功能	service dhcp
配置 dhcp 地址池名称	dhcp dhcp pool [pool name]
配置要分配的网段	network [address] netmask
配置默认网关	default-router [address]
配置 dns 服务器	dns-server [address]
配置不分配地址	ip dhcp excluded-address [address1] [address2]
打开 dhcp snooping 功能	ip dhcp snooping
设置作用的 vlan	ip dhcp snooping vlan <i>n</i>
配置信任端口	ip dhcp snooping trust
配置 dhcp 中继代理的所有接口都作为 dhcp 中继信息选项的信任源	ip dhcp relay information trust-all

16.6 实验问题

第十七章 IPv6 静态路由与默认路由实验

17.1 实验前准备

不论是 IPv4 或者是 IPv6 的网络环境都完整的支持静态路由，静态路由是指由网络管理员手工配置的路由信息。当网络的拓扑结构或者链路的状态发生变化时，网络管理员需要手工去修改路由表中相关的静态路由信息。静态路由信息在缺省情况下是私有的，不会传递给其他路由器。

17.2 实验要求

1. 在路由器 R2 上配置 3 个环回接口 IPv6 地址，分别模拟三个不同的 IPv6 前缀，作为 IPv6 的目标网络。
2. 在路由器 R1 上为三个 IPv6 前缀配置静态路由，并检测其连通性。
3. 使用 IPv6 的默认路由去替代具体的静态路由条目。

17.3 实验拓扑



图 17.1 实验拓扑图

17.4 实验过程

1. 为路由器 R1 和 R2 完成基础配置，包括启动 IPv6 和 IPv6 的地址配置，并激活相关接口，具体配置如下所示：

在路由器 R1 上的配置：

```
R1(config)#ipv6 unicast-routing //启动 IPv6 的路由功能，否则静态路由无法完成。  
R1(config)#interface serial 0/0/0 //进入 serial 0/0/0 接口模式。  
R1(config-if)#ipv6 address 2001:10::1/64 //为该接口配置 IPv6 地址  
R1(config-if)#no shutdown  
R1(config-if)#exit
```

在路由器 R2 上的配置：

```
R2(config)#ipv6 unicast-routing
R2(config)#interface serial 0/0/1
R2(config-if)#ipv6 address 2001:10::2/64
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface loopback1
R2(config-if)#ipv6 address 2001:2::1/64
R2(config-if)#exit
R2(config)#interface loopback2
R2(config-if)#ipv6 address 2001:3::1/64
R2(config-if)#exit
R2(config)#interface loopback3
R2(config-if)#ipv6 address 2001:4::1/64
R2(config-if)#exit
```

2. 在路由器 R1 上去 ping 路由器 R2 上的那几个环回 IPv6 地址，结果应该是 ping 不通，因为在路由器 R1 上暂时没有到目标地址的路由，关于这一技术知识点与 IPv4 的环境相同，要配置 IPv6 静态路由和默认路由功能类似于 IPv4 静态路由和默认路由，但是书写形式上还是存在一定区别，而默认路由是一种特殊的静态路由。

建议的 IPv6 静态路由输入的格式

ipv6 route <目标 IPv6 前缀><出站接口><下一跳 IPv6 地址>

目标 IPv6 前缀：指示目标的 IPv6 网络，这与 IPv4 的目标子网的意义相同。

出站接口：当前路由器转发数据包的出站接口，如果使用了邻接路由器的 IPv6 本地链路地址来作为下一跳地址，那么在静态路由的语法中必须包含出站接口关键字。

下一跳 IPv6 地址：要到达目标网络所要历经的下一跳路由器的 IPv6 的地址，这与 IPv4 的环境相同，注意：根据 RFC2461 规定，路由器必须能够确定下一跳路由器的本地链路地址，所以，在配置 IPv6 静态路由时，下一跳地址建议配置为邻接路由器的本地链路 IPv6 地址。在该实验环境中可以在路由器 R2 上使用 show ipv6 interface serial 0/0/1 来查看路由器 R2 的本地链路 IPv6 地址，如图 17.2 所示。

```
Router#show ipv6 interface serial 0/0/1
Serial0/0/1 is up, line protocol is up
  Ipv6 is enabled, link-local address is FE80::6FE:7FFF:FEEB:7E8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:10::2, subnet is 2001:10::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:2
    FF02::1:FEEB:7E8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 34331)
  Hosts use stateless autoconfig for addresses.
Router#
```

图 17.2 查看路由器 R2 的本地链路 IPv6 地址

在路由器 R1 上配置 IPv6 的静态路由：

```
R1(config)#ipv6 route 2001:2::/64 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
R1(config)#ipv6 route 2001:3::/64 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
R1(config)#ipv6 route 2001:4::/64 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
```

当完成上述配置后，可以在路由器 R1 上通过指令 `show ipv6 route` 查看 Ipv6 的路由表，如图 17.3 所示，可清晰地看见被添加的三条静态路由。然后在路由器 R1 上再次测试与目标 Ipv6 的通信，如果没有故障，应该成功通信，如图 17.4 所示。

```

R1#show ipv6 route
IPv6 Routing Table – Default – 6 entries
Codes: C – Connected, L – Local, S – Static, U – Per-user Static route
        B – BGP, M – MIPv6, R – RIP, I1 – ISIS L1
        I2 – ISIS L2, IA – ISIS interarea, IS – ISIS summary, D – EIGRP
        EX – EIGRP external
        O – OSPF Intra, OI – OSPF Inter, OE1 – OSPF ext 1, OE2 – OSPF ext 2
        ON1 – OSPF NSSA ext 1, ON2 – OSPF NSSA ext 2
S  2001:2::/64  [1/0]
    via FE80::6FE:7FFF:FEEB:7E8, Serial0/0/0
S  2001:3::/64  [1/0]
    via FE80::6FE:7FFF:FEEB:7E8, Serial0/0/0
S  2001:4::/64  [1/0]
    via FE80::6FE:7FFF:FEEB:7E8, Serial0/0/0
C  2001:10::/64 [0/0]
    via Serial0/0/0, directly connected
L  2001:10::1/128 [0/0]
    via Serial0/0/0, receive
L  FF00::/8  [0/0]
    via Null0, receive
R1#

```

图 17.3 查看路由器 R1 的 IPv6 的路由表

```

R1#ping 2001:2::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:2::1, timeout is 2 seconds:
!!!!
Success rate is 100 rate percent(5/5), round-trip min/avg/max = 16/16/16 ms
R1#ping 2001:3:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:3::1, timeout is 2 seconds:
!!!!
Success rate is 100 rate percent(5/5), round-trip min/avg/max = 12/13/16 ms
R1#ping 2001:4:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:4::1, timeout is 2 seconds:
!!!!
Success rate is 100 rate percent(5/5), round-trip min/avg/max = 16/16/16 ms
R1#

```

图 17.4 成功与目标 IPv6 地址通信

3. 在完成上述实验后，请删除三条静态路由，然后配置 IPv6 的默认路由来完成与目标网络通信，关于删除三条静态路由和添加默认路由的配置如下所示，当完成配置后，可以通过再次查看 IPv6 的路由表，如图 17.5 所示，可清晰地看到被添加的 IPv6 的默认路由，此时，路由器 R1 应该能成功的 ping 通三条目标 IPv6 地址。

在路由器 R1 上去删除 IPv6 的静态路由：

```
R1(config)#no ipv6 route 2001:4::/64 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
```

```
R1(config)#no ipv6 route 2001:3::/64 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
```

```
R1(config)#no ipv6 route 2001:2::/64 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
```

在路由器 R1 上配置 Ipv6 的默认路由：

```
R1(config)#ipv6 route ::/0 serial 0/0/0 fe80::6ef:7fff:feeb:7e8
```

```
R1#show ipv6 route
IPv6 Routing Table – Default – 4 entries
Codes: C – Connected, L – Local, S – Static, U – Per-user Static route
       B – BGP, M – MIPv6, R – RIP, I1 – ISIS L1
       I2 – ISIS L2, IA – ISIS interarea, IS – ISIS summary, D – EIGRP
       EX – EIGRP external
       O – OSPF Intra, OI – OSPF Inter, OE1 – OSPF ext 1, OE2 – OSPF ext 2
       ON1 – OSPF NSSA ext 1, ON2 – OSPF NSSA ext 2
S    ::/0    [1/0]
      via FE80::6FE:7FFF:FEEB:7E8, Serial0/0/0
C    2001:10::/64    [0/0]
      via Serial0/0/0, directly connected
L    2001:10::1/128    [0/0]
      via Serial0/0/0, receive
L    FF00::/8    [0/0]
      via Null0, receive
R1#
```

图 17.5 路由器 R1 上的 IPv6 默认路由

17.5 实验命令列表

表 17.1 实验命令列表

启动 IPv6 的路由功能	ipv6 unicast-routing
配置 IPv6 地址	ipv6 address <地址>

17.6 实验问题

第十八章 在 IPv6 环境中的 RIPng 的配置

18.1 实验前的准备

IETF 在 1997 年为了解决 RIP 协议与 IPv6 的兼容性问题 RIP 协议进行了改进，制定了基于 IPv6 的 RIPng(RIP next generation)标准。

RIPng 是一种距离向量 (Distance Vector) 算法。此协议所用的算法早在 1969 年，ARPANET 就用其来计算路由。然而该协议最初属于 XEROX 网络协议。PUP 协议通过网关信息协议交换路由选择信息，而 XNS 则采用该协议的更新版本，命名为路由选择信息协议 (RIP) 实现路由选择信息交换。Berkeley 的路由协议很大程度上与 RIP 相同，即能够处理 IPV4 及其它地址类型的通用地址格式取代了 XNS 地址，同时路由选择每隔 30 秒更新一次。正是因为这种相似性，RIP 既适用于 XNS 协议，也适用于路由类协议。

18.2 实验要求

- 1.配置每台路由器的 IPv6 地址
- 2.在 R1 和 R3 路由器之间配置 RIPng，实现两个 IPv6 网络互相通信

18.3 实验拓扑图

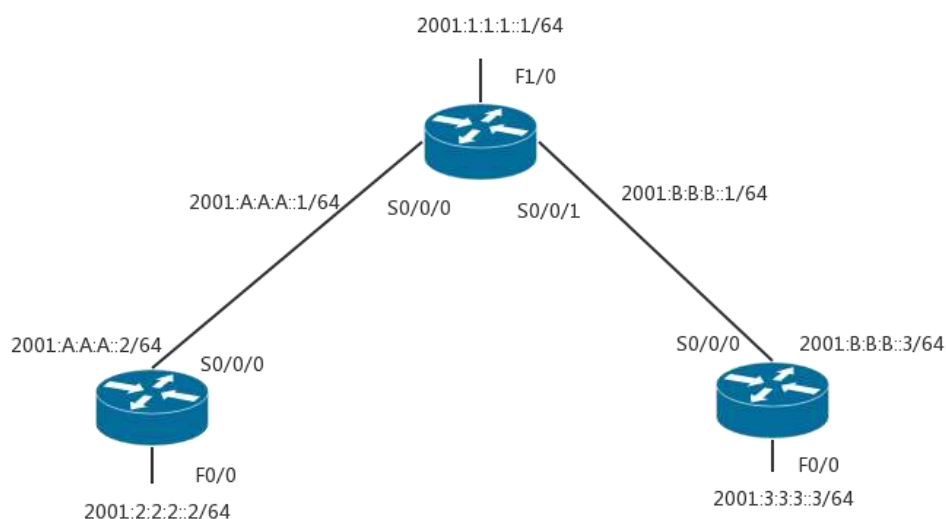


图 18.1 实验拓扑图

18.4 实验过程

1. 首先完成路由器 R1、R2、R3 的 IPv6 的基础配置，其中包括启动 IPv6 和配置 IPv6 的接口地址，激活接口，具体配置如下：

路由器 R1 的基础配置

```
R1(config)#ipv6 unicast-routing
R1(config)#interface f 0/0
R1(config-if)#ipv6 address 2001:2:2:2::2/64
R1(config-if)#ipv6 rip cisco enable
R1(config-if)#no keepalive

R1(config)#interface Serial 0/0/0
R1(config-if)#ipv6 address 2001:A:A:A::2/64
R1(config-if)#ipv6 rip cisco enable
R1(config)#ipv6 router rip cisco
```

路由器 R2 的基础配置

```
R2(config)#ipv6 unicast-routing
R2(config)#interface f 0/0
R2(config-if)#ipv6 address 2001:1:1:1::1/64
R2(config-if)#ipv6 rip cisco enable
R2(config-if)#no keepalive
R2(config)#interface Serial 0/0/0
R2(config-if)#ipv6 address 2001:A:A:A::1/64
R2(config-if)#ipv6 rip cisco enable
R2(config)#interface Serial 0/0/1
R2(config-if)#ipv6 address 2001:B:B:B::1/64
R2(config-if)#ipv6 rip cisco enable
R2(config)#ipv6 router rip cisco
```

路由器 R3 的基础配置

```
R3(config)#ipv6 unicast-routing
R3(config)#interface f 0/0
R3(config-if)#ipv6 address 2001:3:3:1::3/64
R3(config-if)#ipv6 address 2001:3:3:2::3/64
R3(config-if)#ipv6 address 2001:3:3:3::3/64
R3(config-if)#ipv6 address 2001:3:3:4::3/64
R3(config-if)#ipv6 address 2001:3:3:5::3/64
R3(config-if)#ipv6 address 2001:3:3:6::3/64
R3(config-if)#ipv6 rip cisco enable
R3(config)#no keepalive
R3(config)#interface Serial 0/0/0
R3(config-if)#ipv6 address 2001:B:B:B::3/64
R3(config-if)#ipv6 rip cisco enable
R3(config)#ipv6 router rip cisco
```

2.现在开始在完成基础配置的基础上，验证配置
在 R1 验证配置,如下图 18.2 所示

```

R1#show ipv6 rip
RIP process "cisco", port 521, multicast-group FF02::9, pid 168
    Administrative distance is 120. Maximum paths is 16
    Updates every 30 seconds, expire after 180
    Holddown lasts 0 seconds, garbage collect after 120
    Split horizon is on; poison reverse is off
    Default routes are not generated
    Periodic updates 92, trigger updates 16
Interfaces:
    FastEthernet 0/0
    Serial 0/0/0
Redistribution:
    None
R1#show ipv6 route
IPv6 Routing Table – Default – 5 entries
Codes: C – Connected, L – Local, S – Static, U – Per-user Static route
        B – BGP, M – MIPv6, R – RIP, I1 – ISIS L1
        I2 – ISIS L2, IA – ISIS interarea, IS – ISIS summary, D – EIGRP
        EX – EIGRP external
        O – OSPF Intra, OI – OSPF Inter, OE1 – OSPF ext 1, OE2 – OSPF ext 2
        ON1 – OSPF NSSA ext 1, ON2 – OSPF NSSA ext 2
R  2001:1:1:1::/64  [120/2]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
C  2001:2:2:2::/64  [0/0]
    via ::, FastEthernet 0/0
L  2001:2:2:2::2/128  [0/0]
    via ::, FastEthernet 0/0
R  2001:3:3:1::/64  [120/3]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
R  2001:3:3:2::/64  [120/3]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
R  2001:3:3:3::/64  [120/3]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
R  2001:3:3:4::/64  [120/3]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
R  2001:3:3:5::/64  [120/3]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
R  2001:3:3:6::/64  [120/3]
    via FE80::CE00:3FF:FE68:0, Serial 0/0/0
C  2001:A:A:A::/64  [0/0]
    via ::, Serial 0/0/0
L  2001:A:A:A::2/128[0/0]
    via ::, Serial 0/0/0
R  2001:B:B:B::/64  [120/2]

```

```

        via FE80::CE00:3FF:FE68:0, Serial 0/0/0
L   FE80::/10   [0/0]

R1#show ipv6 router rip
IPv6 Routing Table - 14 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
        U - Per-user Static route
        I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2001:1:1:1::/64   [120/2]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3:1::/64   [120/3]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3:2::/64   [120/3]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3:3::/64   [120/3]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3:4::/64   [120/3]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3:5::/64   [120/3]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3:6::/64   [120/3]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:B:B:B::/64   [120/2]
        via FE80::CE00:3FF:FE68:0, Serial0/0/0
R1#show ipv6 rip database
RIP process "cisco", local RIB
    2001:1:1:1::/64, metric 2, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:3:3:1::/64, metric 3, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:3:3:2::/64, metric 3, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:3:3:3::/64, metric 3, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:3:3:4::/64, metric 3, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:3:3:5::/64, metric 3, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:3:3:6::/64, metric 3, installed
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs
    2001:A:A:A::/64, metric 2
        Serial0/0/0/ FE80::CE00:3FF:FE68:0, expires in 157 secs

```

```

2001:B:B:B::/64, metric 2, installed
Serial0/0/0 FE80::CE00:3FF:FE68:0, expires in 157 secs
R1#show ipv6 rip next-hops
RIP process "cisco", Next Hops
FE80::CE00:3FF:FE68:0/Serial0/0/0 [9 paths]
R1#

```

2.在 R3 上实现聚合路由

```
R3(config)#interface serial 0/0/0
```

```
R3(config-if)ipv6 rip cisco summary-address 2001:3:3::/48
```

在 R1 上查看路由表(聚合后的路由)，如下图 18.3 所示

```

R1#show ipv6 route rip
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
        U - Per-user Static route
        I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   2001:1:1:1::/64 [120/2]
    via FE80::CE00:3FF:FE68:0, Serial1/0
R   2001:3:3::/48 [120/3]
    via FE80::CE00:3FF:FE68:0, Serial1/0
R   2001:B:B:B::/64 [120/2]
    via FE80::CE00:3FF:FE68:0, Serial1/0

```

图 18.3 聚合后的路由

3.在 RIPng 中分发默认路由

```
R3(config)#interface Serial 0/0/0
```

```
R3(config-if)ipv6 rip cisco default-information originate metric 5
```

在 R1 上查看默认路由，如下图 18.4 所示

```

R1#show ipv6 route rip
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
        U - Per-user Static route
        I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   ::0 [120/7] //ipv6 里默认路由表示
    via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:1:1:1::/64 [120/2]
    via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:3:3::/48 [120/3]
    via FE80::CE00:3FF:FE68:0, Serial0/0/0
R   2001:B:B:B::/64 [120/2]

```

```
via FE80::CE00:3FF:FE68:0, Serial0/0/0
R1#
```

图 18.4 在 R1 上查看默认路由

18.5 实验命令列表

表 18.1 实验命令列表

启动 RIPng	ipv6 rip cisco enable
标识 RIPng 进程	ipv6 router rip cisco
实现路由聚合	ipv6 rip cisco summary-address [address]

18.6 实验问题

第十九章 基于 IPv6 环境的 OSPFv3 实验

19.1 实验前准备

OSPFv3 主要用于在 IPv6 网络中提供路由功能，OSPFv3 是基于 OSPFv2 上开发用于 IPv6 网络的路由协议。而无论是 OSPFv2 还是 OSPFv3 在工作机制上基本相同；但为了支持 IPv6 地址格式，OSPFv3 对 OSPFv2 做了一些改动。

OSPFv3 是基于链路运行的，一个链路可以划分为多个 IPv6 前缀（类似于子网的概念），节点即使不在同一个前缀范围，只要在同一链路上也可以形成邻居关系，这与 OSPFv2 完全不同，因为在 IPv6 中一条链路可以属于多个子网。

OSPFv3 将使用本地链路地址作为报文发送的源地址。一台路由器可以学习到同一链路上相连的所有路由器的本地链路地址，并使用这些本地链路地址作为下一跳来转发报文。但是在虚拟链路连接上，必须使用全球范围地址或者本地站点地址作为 OSPFv3 协议报文发送的源地址。本地链路地址只在本地链路上有意义且只能在本地链路上泛洪。

19.2 实验要求

1. 分别在路由器 R1、R2、R3 上配置三个环回接口，分别配置三个全球单播范围内的 IPv6 地址，模拟三个不同的 IPv6 前缀（类似于 IPv4 的子网）
2. 在三台路由器上启动 OSPFv3，最后来观察 IPv6 的路由学习结果，查看 OSPFv3 的邻居关系等。

19.3 实验拓扑

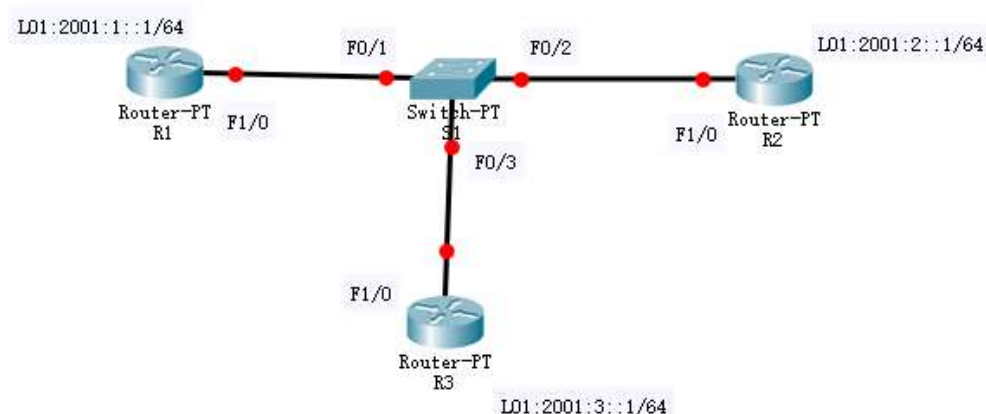


图 19.1 实验拓扑图

19.4 实验过程

1. 首先完成路由器 R1、R2、R3 的 Ipv6 的基础配置，其中包括启动 IPv6 和配置 IPv6 的接口地址，激活接口，具体配置如下
路由器 R1 的基础配置：

```
R1(config)#ipv6 unicast-routing //启动 IPv6 路由功能
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 enable //在接口下启动 IPv6，将自动生成本地链路地址
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface loopback1
R1(config-if)#ipv6 address 2001:1::1/64
```

路由器 R2 的基础配置：

```
R2(config)#ipv6 unicast-routing
R2(config)#interface serial 0/0/0
R2(config-if)#ipv6 enable
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#interface loopback1
R2(config-if)#ipv6 address 2001:2::1/64
```

路由器 R3 的基础配置：

```
R3(config)#ipv6 unicast-routing
R3(config)#interface serial 0/0/0
R3(config-if)#ipv6 enable
R3(config-if)#no shutdown
R3(config-if)#exit
R3(config)#interface loopback1
R3(config-if)#ipv6 address 2001:3::1/64
```

2. 现在开始在完成基础配置的基础上，在各个路由器上启动 OSPFv3 路由协议，具体配置如下所示：

在路由器 R3 的 OSPFv3 配置：

```
R3(config)#ipv6 router ospf 1 //启动 OSPFv3 的路由进程 1
R3(config-rtr)#router-id 3.3.3.3 //为 OSPFv3 配置路由器 ID (RID)
R3(config-rtr)#exit
R3(config)#interface serial 0/0/0
R3(config-if)#ipv6 ospf 1 area0 //使该接口加入到 OSPFv3 进程 1 并申明区域为 0
R3(config-if)#exit
R3(config)#interface loopback1
R3(config-if)#ipv6 ospf 1 area0
R3(config-if)#exit
```

注意：在配置 OSPFv3 时，必须为路由器进程配置路由器 ID(RID)这与 OSPFv2 完全不同，在 OSPFv2 的环境中，RID 是一个可选项配置，但是在 OSPFv3 的环境中 RID 是必须配置，否则 OSPFv3 将无法启动。OSPFv3 的 RID 将仍然以点分十进制的方法显示，比如：1.1.1.1 这很像 IPv4 地址的表达方式。

在路由器 R2 的 OSPFv3 配置：

```

R2(config)#ipv6 router ospf 1
R2(config-rtr)#router-id 2.2.2.2
R2(config)#interface serial 0/0/0
R2(config-if)#ipv6 ospf 1 area0
R2(config-if)#exit
R2(config)#interface loopback1
R2(config-if)#ipv6 ospf 1 area0
R2(config-if)#exit

```

在路由器 R1 的 OSPFv3 配置：

```

R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#exit
R1(config)#interface loopback1
R1(config-if)#ipv6 ospf 1 area0
R1(config-if)#exit

```

3. 现在可以检查 OSPFv3 邻居关系的状态、路由学习的情况，以及连通性检测。可以使用 `show ipv6 ospf neighbor` 来查看 OSPFv3 的邻居关系正常，如图 19.2 所示，并且可知路由器 R3 是 DR 路由器，R2 是 BDR 路由器，关于为什么这样选举，在 OSPFv2 中有详细描述，这里不再重复描述。然后可以通过 `show ipv6 route` 查看路由器 R1 的 IPv6 路由表，如图 19.3 所示，可看出 R1 成功的学习到了路由器 R2 和 R3 公告出来的 OSPF 路由，其中的“O”就表示通过 OSPFv3 所学到的路由。最后在路由器 R1 上通过 `ping` 指令检测与路由器 R2 和 R3 上相关 IPv6 前缀的连通性，如图 19.4 所示，一切正常。

```
R1#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
3.3.3.3	1	FULL/DROTHER	00:00:36	6	FastEthernet0/1
2.2.2.2	1	FULL/DR	00:00:33	6	FastEthernet0/1

图 19.2 查看 OSPF 的邻居关系

```
R1#show ipv6 route
```

IPv6 Routing Table – Default – 5 entries

Codes: C – Connected, L – Local, S – Static, U – Per-user Static route

B – BGP, M – MIPv6, R – RIP, I1 – ISIS L1

I2 – ISIS L2, IA – ISIS interarea, IS – ISIS summary, D – EIGRP

EX – EIGRP external

O – OSPF Intra, OI – OSPF Inter, OE1 – OSPF ext 1, OE2 – OSPF ext 2

ON1 – OSPF NSSA ext 1, ON2 – OSPF NSSA ext 2

```
C 2001:1::/64 [0/0]
```



```
via Loopback1, directly connected
L 2001:1::1/128 [0/0]
via Loopback1 receive
O 2001:2::1/128 [110/1]
via FE80::6FE:7FFF:FEEB:7E9, FastEthernet0/1
O 2001:3::1/128 [110/1]
via FE80::6FE:7FFF:FEEB:7E9, FastEthernet0/1
L FF00::/8 [0/0]
via Null0, receive
R1#
```

图 19.3 查看 IPv6 路由表

```
R1#ping 2001:2::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:2::1, timeout is 2 seconds:
!!!!
Success rate is 100 rate percent(5/5), round-trip min/avg/max = 0/0/4 ms
R1#
```

图 19.4 在路由器 R1 上检测连通性

19.5 实验命令列表

表 19.1 实验命令列表

启动 IPv6	ipv6 enable
为 OSPFv3 配置路由器 ID（RID）	router-id <ID>
查看 OSPF 邻居关系	show ipv6 ospf neighbor
查看 IPv6 路由表	show ipv6 route

19.6 实验问题