

网 络 工 程

本 科 实 验 报 告

实验名称: VPN 实例配置

学 员 姓 名	王李烜	学 号	202202001046
培 养 类 型	无军籍	年 级	2022
专 业	网络工程	所 属 学 院	计算机学院
指 导 教 员	张军	职 称	工程师
实 验 室	306-707	实 验 时 间	2024.11.26

《本科实验报告》填写说明

实验报告内容编排应符合以下要求：

- (1) 采用 A4 (21cm×29.7cm) 白色复印纸，单面黑字。上下左右各侧的页边距均为 3cm；缺省文档网格：字号为小 4 号，中文为宋体，英文和阿拉伯数字为 Times New Roman，每页 30 行，每行 36 字；页脚距边界为 2.5cm，页码置于页脚、居中，采用小 5 号阿拉伯数字从 1 开始连续编排，封面不编页码。
- (2) 报告正文最多可设四级标题，字体均为黑体，第一级标题字号为 4 号，其余各级标题为小 4 号；标题序号第一级用“一、”、“二、”……，第二级用“（一）”、“（二）”……，第三级用“1.”、“2.”……，第四级用“（1）”、“（2）”……，分别按序连续编排。
- (3) 正文插图、表格中的文字字号均为 5 号。

目录

1 实验目的	4
2 实验原理	4
2.1 VPN	4
2.2 VRF	4
2.2.1 概述	4
2.2.2 工作原理	4
3 实验环境	5
3.1 实验背景	5
3.2 实验设备	5
4 实验步骤及结果	5
4.1 实验拓扑	5
4.2 按照拓扑图接线	6
4.3 配置基本网络	7
4.3.1 配置 PC	7
4.3.2 配置路由器 IP 地址	7
4.3.3 配置 RIP 协议使互通	8
4.4 配置 VPN 实例	10
4.4.1 创建 VPN 实例	10
4.4.2 配置 VPN 实例的路由	11
4.5 结果检验	12
5 实验总结	12

图目录

Figure 1: 实验拓扑图	5
Figure 2: 机柜正面接线图	6
Figure 3: 机柜背面接线图	6
Figure 4: 配置 AR1 的 IP 地址	7
Figure 5: 配置 AR2 的 IP 地址	7
Figure 6: 配置 AR3 的 IP 地址	8
Figure 7: 在 AR1 上配置 RIP 协议	8
Figure 8: 在 AR2 上配置 RIP 协议	9
Figure 9: 在 AR3 上配置 RIP 协议	9
Figure 10: PC2 ping 其他主机	10
Figure 11: 创建 VPN 实例并设置 RD 值	10
Figure 12: 绑定接口	11
Figure 13: 重新设置接口 IP 地址	11
Figure 14: 配置 RIP 进程	11
Figure 15: AR3 ping PC1	12
Figure 16: PC1 分别 ping VPN 内外的主机	12

1 实验目的

1. 了解 VPN 的基本概念及应用场景；
2. 掌握 VPN 在隔离数据流、构建虚拟网络中的关键作用；
3. 熟悉 VPN 的配置方法与工作机制。

通过本实验，我将学习 VPN 的基本原理及其在实现多租户隔离、提升网络灵活性方面的重要应用。同时，通过实践掌握 VPN 实例的创建、接口绑定及路由表配置等操作，深入理解 VPN 实例在数据转发和隔离中的实现机制，并了解其与根实例的区别及优点。实验将进一步验证 VPN 实例在多接口场景下的流量隔离效果，帮助我建立对虚拟网络技术的系统性认识，为日后设计复杂网络拓扑和实现网络隔离奠定基础。

2 实验原理

2.1 VPN

虚拟专用网络（VPN，Virtual Private Network）是一种通过公共网络（如 Internet）建立安全连接的技术，用于实现远程用户接入、分支机构互联、数据加密传输等应用场景。VPN 技术通过加密隧道、身份认证、数据加密等手段，保障数据传输的安全性和隐私性，提高网络通信的可靠性和保密性。常见的 VPN 类型包括 IPSec VPN、SSL VPN、PPTP VPN 等，广泛应用于企业网络、远程办公、移动通信等领域。

2.2 VRF

2.2.1 概述

虚拟路由转发（VRF，Virtual Routing and Forwarding）技术通过在同一台三层设备上创建多张独立的路由表，实现数据流的隔离和独立转发。VRF 广泛应用于 MPLS VPN、防火墙等需要隔离的场景。在 VRF 环境中，每个实例拥有独立的接口、路由表和路由协议进程等，使流量彻底隔离。没有 VPN 实例时，设备的所有接口都属于同一个根实例；而创建 VPN 实例后，可以将特定接口绑定到该实例，使其服务于特定业务，从而实现多租户数据的独立管理。

在华为设备上 VRF 又称为 VPN 实例。

2.2.2 工作原理

1. 创建实例：在物理设备中创建 VPN 实例，用于定义独立的转发域；
2. 接口绑定：将三层接口（如物理接口、子接口或 VLANIF 接口）分配给实例，使其成为实例的一部分；
3. 路由表建立：基于实例绑定的接口和配置的路由协议建立独立的路由表，并按照这些路由表转发数据，从而实现实例间的数据隔离。

3 实验环境

3.1 实验背景

网工系的学生正在学习 VPN 技术，需要通过实验验证 VPN 实例在网络隔离中的应用。为此，需要搭建一个简单的网络拓扑，包括三台路由器和四台 PC，通过配置 VPN 实例实现不同网络之间的数据隔离。

3.2 实验设备

设备名称	设备型号	设备数量
路由器	华为 AR6120-S	3
PC	联想启天 M410 Windows 10	4

另有网线若干，控制线 3 条。

4 实验步骤及结果

4.1 实验拓扑

按实验背景，绘制拓扑图如下：

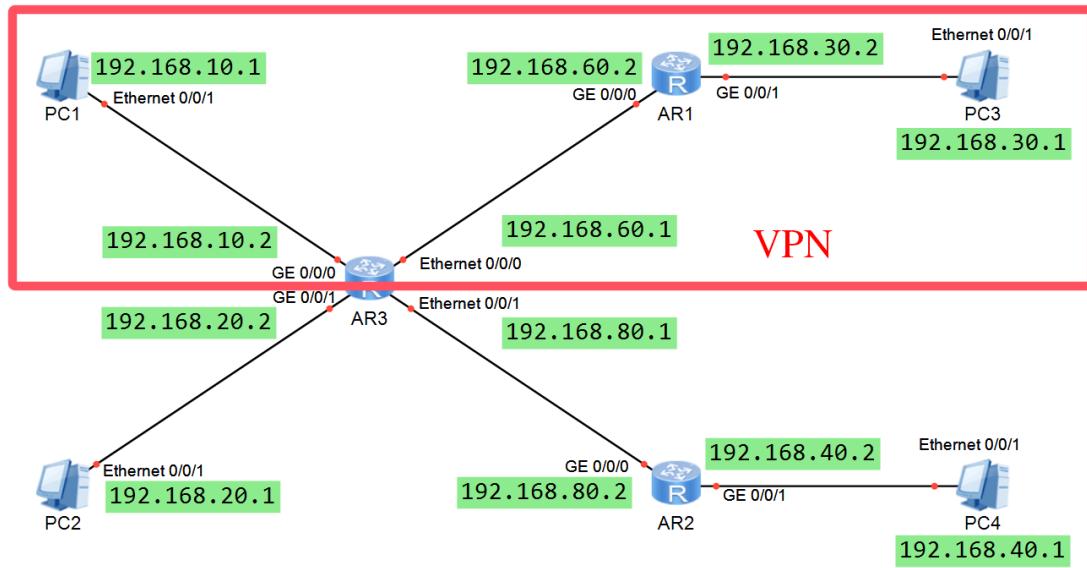


Figure 1: 实验拓扑图

要求将红框内的 PC1、PC3 划分到一个 VPN 实例中。

4.2 按照拓扑图接线

按照拓扑图接线。



Figure 2: 机柜正面接线图



Figure 3: 机柜背面接线图

4.3 配置基本网络

4.3.1 配置 PC

- 配置 PC1 的 IP 地址为 192.168.10.1/24，网关为 192.168.10.2；
- 配置 PC2 的 IP 地址为 192.168.20.1/24，网关为 192.168.20.2；
- 配置 PC3 的 IP 地址为 192.168.30.1/24，网关为 192.168.30.2；
- 配置 PC4 的 IP 地址为 192.168.40.1/24，网关为 192.168.40.2。

步骤简单，展示图略。

4.3.2 配置路由器 IP 地址

按照拓扑图配置路由器的 IP 地址。配置 AR1 的 G0/0/0 口 IP 地址为

1. 配置 AR1 路由器：

- 接口 GE 0/0/0 连接到 AR3，IP 地址为 192.168.60.2。
- 接口 GE 0/0/1 连接到 PC3，IP 地址为 192.168.30.2。

```
[Huawei] int g0/0/0
[Huawei-GigabitEthernet0/0/0]undo portswitch
[Huawei-GigabitEthernet0/0/0]
[Huawei-GigabitEthernet0/0/0] ip add 192.168.60.2 24
[Huawei-GigabitEthernet0/0/0] int g0/0/1
[Huawei-GigabitEthernet0/0/1]undo portswitch
[Huawei-GigabitEthernet0/0/1] ip add 192.168.30.2
^
Error:Incomplete command found at '^' position.
[Huawei-GigabitEthernet0/0/1] ip add 192.168.30.2 24
[Huawei-GigabitEthernet0/0/1]
```

Figure 4: 配置 AR1 的 IP 地址

2. 配置 AR2 路由器：

- 接口 GE 0/0/0 连接到 AR3，IP 地址为 192.168.80.2。
- 接口 GE 0/0/1 连接到 PC4，IP 地址为 192.168.40.2。

```
[AR2] int g0/0/1
[AR2-GigabitEthernet0/0/1] ip add 192.168.40.2 24
[AR2-GigabitEthernet0/0/1]
[AR2-GigabitEthernet0/0/1]
[AR2-GigabitEthernet0/0/1] int g0/0/0
[AR2-GigabitEthernet0/0/0] undo portswitch
[AR2-GigabitEthernet0/0/0]
[AR2-GigabitEthernet0/0/0] ip add 192.168.80.2
^
Error:Incomplete command found at '^' position.
[AR2-GigabitEthernet0/0/0] ip add 192.168.80.2 24
[AR2-GigabitEthernet0/0/0]
```

Figure 5: 配置 AR2 的 IP 地址

3. 配置 AR3 路由器：

- 接口 GE 0/0/0 连接到 PC1，IP 地址为 192.168.10.2。
- 接口 GE 0/0/1 连接到 PC2，IP 地址为 192.168.20.2。

注意拓扑图中的 Ethernet0/0/0 和 Ethernet0/0/1 分别对应 GE 0/0/3 和 GE 0/0/2：

- 接口 GE 0/0/2 连接到 AR2，IP 地址为 192.168.80.1。
- 接口 GE 0/0/3 连接到 AR1，IP 地址为 192.168.60.1。

```
[Huawei]int g0/0/0
[Huawei-GigabitEthernet0/0/0]ip addr 192.168
^
Error: Unrecognized command found at '^' position.
[Huawei-GigabitEthernet0/0/0]undo ports
[Huawei-GigabitEthernet0/0/0]undo portswitch
[Huawei-GigabitEthernet0/0/0]ip addr 192.168.10.2 24
[Huawei-GigabitEthernet0/0/0]int g0/0/1
[Huawei-GigabitEthernet0/0/1]ip addr 192.168.20.2 24
^
Error: Unrecognized command found at '^' position.
[Huawei-GigabitEthernet0/0/1]undo portswitch
[Huawei-GigabitEthernet0/0/1]ip addr 192.168.20.2 24
[Huawei-GigabitEthernet0/0/1]int g0/0/3
[Huawei-GigabitEthernet0/0/3]undo portswitch
[Huawei-GigabitEthernet0/0/3]ip addr 192.168.60.2 24
[Huawei-GigabitEthernet0/0/3]int g0/0/2
[Huawei-GigabitEthernet0/0/2]undo portswitch
[Huawei-GigabitEthernet0/0/2]ip addr 192.168.80.2 24
[Huawei-GigabitEthernet0/0/2]q
[Huawei]
```

Figure 6: 配置 AR3 的 IP 地址

4.3.3 配置 RIP 协议使互通

本次实验使用 RIP 协议，配置路由器使得各个网络互通：

```
[AR1]rip 2
[AR1-rip-2]net
[AR1-rip-2]network 192.168.10.0
[AR1-rip-2]network 192.168.20.0
[AR1-rip-2]network 192.168.30.0
[AR1-rip-2]network 192.168.40.0
[AR1-rip-2]network 192.168.60.0
[AR1-rip-2]network 192.168.80.0
[AR1-rip-2]ver 2
[AR1-rip-2]q
```

Figure 7: 在 AR1 上配置 RIP 协议

```
[AR2]rip 2
[AR2-rip-2]network 192.168.10.0
[AR2-rip-2]network 192.168.20.0
[AR2-rip-2]network 192.168.30.0
[AR2-rip-2]network 192.168.40.0
[AR2-rip-2]network 192.168.60.0
[AR2-rip-2]network 192.168.80.0
[AR2-rip-2]
[AR2-rip-2]version 2
[AR2-rip-2]
```

Figure 8: 在 AR2 上配置 RIP 协议

```
[Huawei]
[Huawei]
[Huawei]sysname AR3
[AR3]rip 2
[AR3-rip-2]net
[AR3-rip-2]network 192.168.10.0
[AR3-rip-2]network 192.168.20.0
[AR3-rip-2]network 192.168.30.0
[AR3-rip-2]network 192.168.40.0
[AR3-rip-2]network 192.168.60.0
[AR3-rip-2]network 192.168.80.0
[AR3-rip-2]q
[AR3]rip 2
[AR3-rip-2]vers
[AR3-rip-2]version 2
[AR3-rip-2]q
[AR3]■
```

Figure 9: 在 AR3 上配置 RIP 协议

此时在 PC4(192.168.40.1/24)上 ping 其他三台主机，发现可以互通：

```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.175]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping 192.168.30.1

正在 Ping 192.168.30.1 具有 32 字节的数据:
来自 192.168.30.1 的回复: 字节=32 时间<1ms TTL=125

192.168.30.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\Administrator>ping 192.168.10.1

正在 Ping 192.168.10.1 具有 32 字节的数据:
来自 192.168.10.1 的回复: 字节=32 时间<1ms TTL=126

192.168.10.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\Administrator>ping 192.168.20.1

正在 Ping 192.168.20.1 具有 32 字节的数据:
来自 192.168.20.1 的回复: 字节=32 时间<1ms TTL=126

192.168.20.1 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\Administrator>
```

Figure 10: PC2 ping 其他主机

4.4 配置 VPN 实例

4.4.1 创建 VPN 实例

在 AR1 上创建 VPN 实例 wlxvpn，并为该实例设置 RD 值为 100:1:

```
[AR3]ip vpn-
[AR3]ip vpn-instance wlxvpn
[AR3-vpn-instance-wlxvpn]route-dis
[AR3-vpn-instance-wlxvpn]route-distinguisher 100:1
[AR3-vpn-instance-wlxvpn-af-ipv4]q
[AR3-vpn-instance-wlxvpn]q
[AR3]int ■
```

Figure 11: 创建 VPN 实例并设置 RD 值

并配置接口 GE 0/0/0、GE 0/0/2 绑定到该实例:

```
[AR3-GigabitEthernet0/0/0]q
[AR3]int g0/0/0
[AR3-GigabitEthernet0/0/0]ip binding vpn-instance wlxvpn
Info: The interface is already associated to the same VPN instance.
[AR3-GigabitEthernet0/0/0]int g0/0/2
[AR3-GigabitEthernet0/0/2]ip binding vpn-instance wlxvpn
Info: All IPv4 related configurations on this interface are removed!
Info: All IPv6 related configurations on this interface are removed!
[AR3-GigabitEthernet0/0/2]q
```

Figure 12: 绑定接口

如下图 Figure 13，绑定之后接口的配置会被清空，需要重新设置接口的 IP 地址：

```
[AR3]int g0/0/2
[AR3-GigabitEthernet0/0/2]di th
[V300R019C10SPC200]
#
interface GigabitEthernet0/0/2
undo portswitch
ip binding vpn-instance wlxvpn
#
return
[AR3-GigabitEthernet0/0/2]ip addr 192.168.10.2 24
[AR3-GigabitEthernet0/0/2]ip addr 192.168.60.1 24
[AR3-GigabitEthernet0/0/2]int g0/0/0
[AR3-GigabitEthernet0/0/0]di th
[V300R019C10SPC200]
#
interface GigabitEthernet0/0/0
undo portswitch
ip binding vpn-instance wlxvpn
#
return
[AR3-GigabitEthernet0/0/0]ip addr 192.168.10.2
^
Error:Incomplete command found at '^' position.
[AR3-GigabitEthernet0/0/0]ip addr 192.168.10.2 24
[AR3-GigabitEthernet0/0/0]q
[AR3]■
```

Figure 13: 重新设置接口 IP 地址

4.4.2 配置 VPN 实例的路由

向 wlxvpn 的路由表中创建 RIP 进程，进程号为 100，并且该进程与 VPN 实例 wlxvpn 进行绑定。设备在该 RIP 进程中所学习到的路由会加载到 wlxvpn 的路由表中。

```
[AR3]rip 100 vpn-instance wlxvpn
[AR3-rip-100]network 192.168.10.0
[AR3-rip-100]network 192.168.60.0
[AR3-rip-100]network 192.168.30.0
```

Figure 14: 配置 RIP 进程

此时可以测试一下 VPN 的连通性：

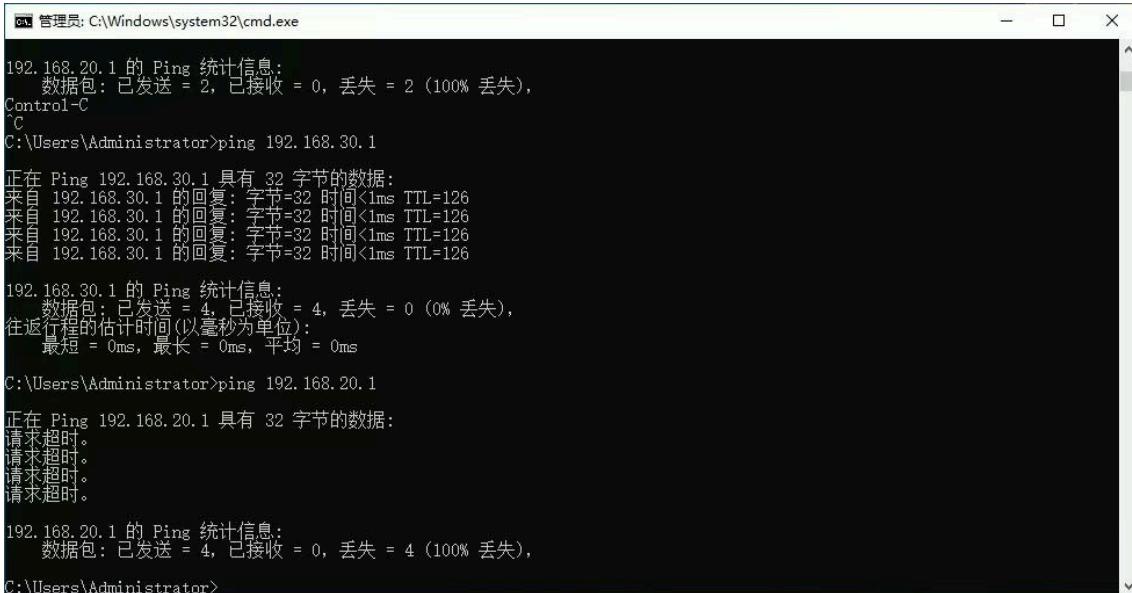
```
[AR3]ping -vpn-instance wlxvpn 192.168.10.1
PING 192.168.10.1: 56 data bytes, press CTRL-C to break
Reply from 192.168.10.1: bytes=56 Sequence=1 ttl=128 time=1 ms
Reply from 192.168.10.1: bytes=56 Sequence=2 ttl=128 time=1 ms
Reply from 192.168.10.1: bytes=56 Sequence=3 ttl=128 time=1 ms
Reply from 192.168.10.1: bytes=56 Sequence=4 ttl=128 time=1 ms
Reply from 192.168.10.1: bytes=56 Sequence=5 ttl=128 time=1 ms

--- 192.168.10.1 ping statistics ---
5 packet(s) transmitted
5 packet(s) received
0.00% packet loss
round-trip min/avg/max = 1/1/1 ms
```

Figure 15: AR3 ping PC1

4.5 结果检验

在 PC1 上分别 ping PC2 与 PC3，发现 PC1 与 PC2 无法通信，PC1 与 PC3 可以通信：



```
管理员: C:\Windows\system32\cmd.exe
192.168.20.1 的 Ping 统计信息:
  数据包: 已发送 = 2, 已接收 = 0, 丢失 = 2 (100% 丢失),
Control-C
^C
C:\Users\Administrator>ping 192.168.30.1

正在 Ping 192.168.30.1 具有 32 字节的数据:
来自 192.168.30.1 的回复: 字节=32 时间<1ms TTL=126

192.168.30.1 的 Ping 统计信息:
  数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
  最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\Administrator>ping 192.168.20.1

正在 Ping 192.168.20.1 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.20.1 的 Ping 统计信息:
  数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
C:\Users\Administrator>
```

Figure 16: PC1 分别 ping VPN 内外的主机

说明 VPN 实例 wlxvpn 的配置成功，实现了 PC1 与 PC2 等的数据隔离。

5 实验总结

本次实验通过配置 VPN 实例，实现了不同网络之间的数据隔阂，验证了 VPN 实例在网络隔离中的重要作用。通过实验，我掌握了 VPN 实例的创建、接口绑定、路由配置等操作方法，深入理解了 VPN 实例在数据转发和隔离中的实现机制。同时，通过实验证明了 VPN 实例在多接口场景下的流量隔离效果，为我今后设计复杂网络拓扑和实现网络隔离提供了重要参考。