一. 实验环境

本实验基于软件仿真的方式完成，通过在本地主机上运行网络仿真平台，模拟真实 Cisco 路由器的工作环境，以验证 RIP 与 OSPF 动态路由协议的配置方法及其协议运行过程。

1. 宿主机环境

- **操作系统**：macOS 15.7.2
- **运行方式**：本地运行网络仿真服务（gns3server 直接运行于宿主机）

本实验未使用虚拟机或远程服务器，因为本机环境可以直接方便运行server，不必要再添加虚拟机增加负担，也方便查看输出，由于安装时出现了一些配置问题，所以gns3server是单独安装的，并不是一体化的，但是整体使用差距不大。

2. 网络仿真平台与软件环境

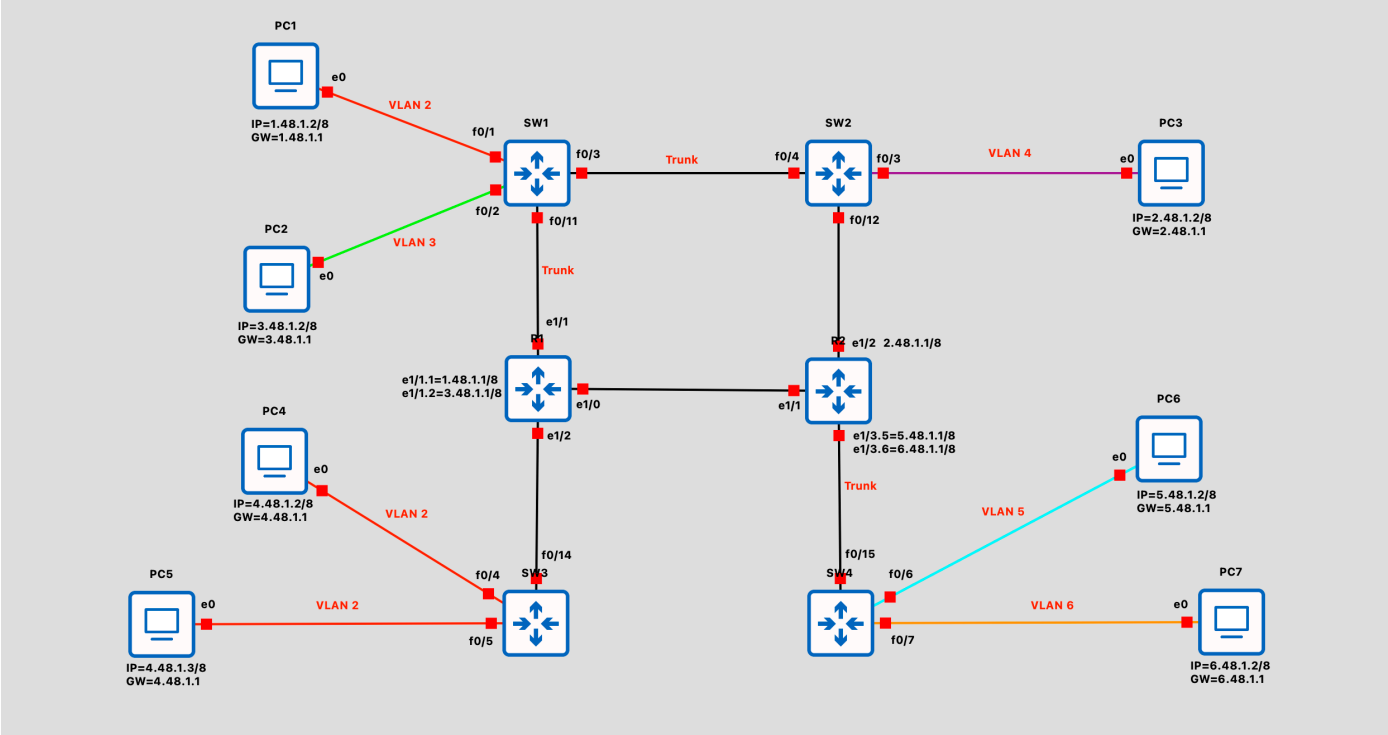
实验采用 GNS3 网络仿真平台进行拓扑搭建与设备仿真。本实验使用 **GNS3 3.0.2**，并采用 **前端 GUI + 手动安装后端 server** 的方式在本机运行，以便于本地调试与查看日志输出。具体软件环境如下：

- **网络仿真控制台（前端）**：GNS3 GUI 3.0.2
  负责提供图形化拓扑构建界面，并将启动/停止设备、连线与配置等控制指令下发至后端仿真服务。
- **网络仿真服务器（后端）**：GNS3 Server（gns3server）3.0.2（手动安装，运行于宿主机）
  负责承载各类仿真节点的创建与运行，并统一管理项目所需的镜像、配置文件与运行日志。
- **仿真核心引擎**：Dynamips
  用于模拟 Cisco 路由器的硬件架构，使路由器能够运行真实 IOS 系统，从而支持 RIP、OSPF 等动态路由协议的完整配置与调试。
- **协议分析工具**：Wireshark
  通过 GNS3 的链路抓包功能调用 Wireshark，对链路上的 IP 数据包进行捕获与解析，可以用于配置时进行调试，查找一些问题。

3. 虚拟网络设备

本实验共使用2台路由器，7台PC，4台交换机（由路由器模拟），除了VLAN 1外一共划分了5个VLAN，使用的路由器版本还是Cisco C3640

实验拓扑如下：



可以看到左上和右下部分都使用了单臂路由。

二. 实验目的

- 熟练掌握交换机 VLAN 的创建、端口划分，以及 Access 与 Trunk 链路的配置区别与应用场景。
- 深入理解并配置基于 802.1Q 封装的路由器子接口，实现单臂路由，解决单物理链路下的 VLAN 间互通问题。
- 在包含单臂路由、物理直连路由的混合拓扑中，部署 RIP动态路由协议，实现不同 VLAN 网段的互联互通。
- 通过 show ip route、show vlan 等命令及 Ping 测试，分析数据包在不同网络设备间的转发路径及 VLAN 标签的封装与剥离过程。

三. 实验内容及步骤

1. 实验拓扑的构建

这部分直接可以按照设计出的拓扑图进行构建，同时按照分配的ip地址等信息，在图中标注一些相关信息方便后续查看与展示

2. 网络设备的配置

2.1 PC的配置

图中已经给出了各个PC及对应的网关，这里不单独标出，相关指令：

```
ip 1.48.1.2 255.0.0.0 1.48.1.1
save
```

注意写入配置后保存，否则配置重启后会丢失，可能会在后续实验中出现不必要的麻烦

2.2 交换机的配置

这里我们必须使用带有VLAN功能的交换机，而GNS3自带的交换机无法使用telnet连接，也就没法配置VLAN了，所以我们采用路由器进行模拟，相关接口使用FastEthernet即可。

先以SW1为例，创建 VLAN 并配置主机接口：

```
SW1(config)# vlan 2
SW1(config)# vlan 3
```

将连接 PC1 的端口加入 VLAN2：

```
SW1(config)# interface f0/1
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 2
```

将连接 PC2 的端口加入 VLAN3：

```
SW1(config)# interface f0/2
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 3
```

同时SW1和SW2之间还需要配置Trunk模式，以两者之间的链路为例：

```
SW1(config)# interface f0/3
SW1(config-if)# switchport trunk encapsulation dot1q
SW1(config-if)# switchport mode trunk
```

```
SW2(config)# interface f0/4
SW2(config-if)# switchport trunk encapsulation dot1q
SW2(config-if)# switchport mode trunk
```

对于SW4的单臂路由做法也是类似的，我们可以用show interf trunk查看trunk接口状态和允许vlan列表

```
SW4#show interf trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa0/15    on           802.1q         trunking      1

Port      Vlans allowed on trunk
Fa0/15    1-1005

Port      Vlans allowed and active in management domain
Fa0/15    1,5-6

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/15    1,5-6
```

对于其他交换机的配置也是类似的，配置一个Access口，根据拓扑图中的vlan进行配置即可

2.3 路由器的配置

我们的拓扑图中左上和右下都有单臂路由，都需要进行相应配置，我们需要给各个接口配置相应的网关以及R1和R2之间的接口也需要配置ip地址用于互联，我们采用RIP协议配置动态路由，这里以R1为例子，展示一下配置相关命令：

```
R1# conf t

R1(config)# interface e1/0
R1(config-if)# ip address 10.48.1.1 255.255.255.252
R1(config-if)# no shutdown

R1(config)# interface e1/1
R1(config-if)# no shutdown

R1(config)# interface e1/1.1
R1(config-subif)# encapsulation dot1Q 2
R1(config-subif)# ip address 1.48.1.1 255.0.0.0

R1(config)# interface e1/1.2
R1(config-subif)# encapsulation dot1Q 3
R1(config-subif)# ip address 3.48.1.1 255.0.0.0

R1(config)# interface e1/2
R1(config-if)# ip address 4.48.1.1 255.0.0.0
R1(config-if)# no shutdown

R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# no auto-summary
R1(config-router)# network 1.0.0.0
R1(config-router)# network 3.0.0.0
R1(config-router)# network 4.0.0.0
R1(config-router)# network 10.0.0.0
```

```
R1(config)# end
```

对于R2使用的相关命令和上面的类似，根据拓扑图修改一下相关信息即可

```
Ethernet1/0                10.48.1.1       YES NVRAM  up                    up
Ethernet1/1                unassigned      YES NVRAM  up                    up
Ethernet1/1.1              1.48.1.1        YES NVRAM  up                    up
Ethernet1/1.2              3.48.1.1        YES NVRAM  up                    up
Ethernet1/2                4.48.1.1        YES NVRAM  up                    up
```

验证一下路由表是否正常：

```
C    1.0.0.0/8 is directly connected, Ethernet1/1.1
R    2.0.0.0/8 [120/1] via 10.48.1.2, 00:00:12, Ethernet1/0
C    3.0.0.0/8 is directly connected, Ethernet1/1.2
C    4.0.0.0/8 is directly connected, Ethernet1/2
R    5.0.0.0/8 [120/1] via 10.48.1.2, 00:00:12, Ethernet1/0
R    6.0.0.0/8 [120/1] via 10.48.1.2, 00:00:12, Ethernet1/0
     10.0.0.0/30 is subnetted, 1 subnets
C       10.48.1.0 is directly connected, Ethernet1/0
```

四. 实验结果

我们已经完成了相关的所有配置，接下来验证一下配置的结果

4.1 VLAN内通信结果

我们对同一VLAN内的主机进行测试：

- 位于 VLAN2 中的 PC1 与 PC4、PC5 可相互通信
- 位于 VLAN5 的 PC6 可正常访问其默认网关
- 位于 VLAN6 的 PC7 可正常访问其默认网关

测试结果表明，各交换设备上的 VLAN 划分及 Access 接口配置正确，同一 VLAN 内的广播与单播通信均能够正常完成。下面是相关截图

```
PC4> ping 1.48.1.2

1.48.1.2 icmp_seq=1 timeout
84 bytes from 1.48.1.2 icmp_seq=2 ttl=63 time=58.715 ms
84 bytes from 1.48.1.2 icmp_seq=3 ttl=63 time=14.799 ms
84 bytes from 1.48.1.2 icmp_seq=4 ttl=63 time=43.872 ms
84 bytes from 1.48.1.2 icmp_seq=5 ttl=63 time=46.031 ms

PC7> ping 6.48.1.1

84 bytes from 6.48.1.1 icmp_seq=1 ttl=255 time=36.056 ms
84 bytes from 6.48.1.1 icmp_seq=2 ttl=255 time=7.556 ms
84 bytes from 6.48.1.1 icmp_seq=3 ttl=255 time=8.114 ms
84 bytes from 6.48.1.1 icmp_seq=4 ttl=255 time=7.378 ms
84 bytes from 6.48.1.1 icmp_seq=5 ttl=255 time=12.011 ms
```

4.2 VLAN间通信结果

在三层设备配置完成后，对不同 VLAN 间的通信情况进行测试。

测试结果如下：

- VLAN2 与 VLAN3 之间主机可正常通信
- VLAN4 与 VLAN2、VLAN3 之间主机可正常通信
- VLAN5 与 VLAN6 之间主机可通过 R2 的单臂路由正常通信

说明路由器三层接口配置正确，交换机 Trunk 链路能够正确转发带有 VLAN 标签的数据帧，VLAN 间隔离与互通机制均正常工作。

```
PC1> ping 3.48.1.2

84 bytes from 3.48.1.2 icmp_seq=1 ttl=63 time=109.318 ms
84 bytes from 3.48.1.2 icmp_seq=2 ttl=63 time=19.993 ms
84 bytes from 3.48.1.2 icmp_seq=3 ttl=63 time=74.349 ms
84 bytes from 3.48.1.2 icmp_seq=4 ttl=63 time=30.736 ms
84 bytes from 3.48.1.2 icmp_seq=5 ttl=63 time=48.351 ms
```

```
PC6> ping 6.48.1.2

84 bytes from 6.48.1.2 icmp_seq=1 ttl=63 time=88.765 ms
84 bytes from 6.48.1.2 icmp_seq=2 ttl=63 time=69.888 ms
84 bytes from 6.48.1.2 icmp_seq=3 ttl=63 time=42.310 ms
84 bytes from 6.48.1.2 icmp_seq=4 ttl=63 time=19.285 ms
84 bytes from 6.48.1.2 icmp_seq=5 ttl=63 time=63.217 ms
```

4.3 跨路由器通信结果

为验证多跳路由环境下的通信情况，对跨越 R1 与 R2 的主机通信进行测试。

测试结果表明：

- PC7（VLAN6）可以成功 ping 通左侧 VLAN 中的主机
- 数据包能够经过 R2、R1 等多台路由器正确转发并返回

该结果说明 RIP 动态路由协议在各路由器之间正确建立了路由信息，回程路径完整，路由收敛正常。

```
PC7> ping 1.48.1.2

84 bytes from 1.48.1.2 icmp_seq=1 ttl=62 time=67.863 ms
84 bytes from 1.48.1.2 icmp_seq=2 ttl=62 time=143.133 ms
84 bytes from 1.48.1.2 icmp_seq=3 ttl=62 time=82.926 ms
84 bytes from 1.48.1.2 icmp_seq=4 ttl=62 time=116.104 ms
84 bytes from 1.48.1.2 icmp_seq=5 ttl=62 time=91.348 ms
```

4.4 单臂路由的路由表

我们使用show ip route connected查看直连路由，可以看到单一接口维护了多个网段。

```
R2#show ip route connected
C    2.0.0.0/8 is directly connected, Ethernet1/2
C    5.0.0.0/8 is directly connected, Ethernet1/3.5
C    6.0.0.0/8 is directly connected, Ethernet1/3.6
     10.0.0.0/30 is subnetted, 1 subnets
C       10.48.1.0 is directly connected, Ethernet1/1
```

4.5 路由跟踪分析

我们选取同一交换机下但属于不同VLAN的PC6和PC7：

```
PC7> trace 5.48.1.2
trace to 5.48.1.2, 8 hops max, press Ctrl+C to stop
 1   6.48.1.1   33.976 ms  11.522 ms  24.124 ms
 2   *5.48.1.2   82.530 ms (ICMP type:3, code:3, Destination port unreachable)
```

可以看到发出的数据包还是会经过网关，没有在交换机内部转发，也体现了VLAN的作用

接下来我们选取同一交换机下并属于同一VLAN的PC4和PC5

```
PC4> trace 4.48.1.3
trace to 4.48.1.3, 8 hops max, press Ctrl+C to stop
 1   *4.48.1.3   1.016 ms (ICMP type:3, code:3, Destination port unreachable)
```

这里看到下一跳直接就是目标主机，直接在交换机内部完成了转发

4.6 相关的抓包分析

这里在SW4到R1的链路上捕获了由PC7发往PC1的ICMP包：

```
> Frame 76: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface -, id 0
v Ethernet II, Src: Private_66:68:06 (00:50:79:66:68:06), Dst: cc:04:01:dc:00:13 (cc:04:01:dc
  > Destination: cc:04:01:dc:00:13 (cc:04:01:dc:00:13)
  > Source: Private_66:68:06 (00:50:79:66:68:06)
    Type: 802.1Q Virtual LAN (0x8100)
    [Stream index: 5]
v 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 6
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0000 0110 = ID: 6
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 6.48.1.2, Dst: 1.48.1.2
> Internet Control Message Protocol
```

抓包结果显示，当数据帧通过 Trunk 接口传输时，帧头中包含 **IEEE 802.1Q** 标签，其主要字段包括：

- **TPID（Tag Protocol Identifier）**：0x8100，用于标识该帧为带 VLAN 标签的以太网帧
- **TCI（Tag Control Information）**：
  - **VLAN ID**：指明数据帧所属 VLAN
  - **PRI**：优先级字段（本实验未涉及 QoS）
  - **CFI/DEI**：用于兼容性和丢弃指示

我们再抓取一个Access接口的，选取SW1至PC1的链路，抓取PC3到PC1的ICMP包：

```
> Frame 66: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
v Ethernet II, Src: cc:03:01:da:00:11 (cc:03:01:da:00:11), Dst: Private_66:68:00 (00:50:79:66
  > Destination: Private_66:68:00 (00:50:79:66:68:00)
  > Source: cc:03:01:da:00:11 (cc:03:01:da:00:11)
    Type: IPv4 (0x0800)
    [Stream index: 3]
v Internet Protocol Version 4, Src: 2.48.1.2, Dst: 1.48.1.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x26f5 (9973)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 62
    Protocol: ICMP (1)
    Header Checksum: 0x5051 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 2.48.1.2
    Destination Address: 1.48.1.2
    [Stream index: 1]
> Internet Control Message Protocol
```

这里可以看到该包没有携带VLAN标签。

根据这个抓包信息可以更加清晰地看到VLAN的隔离特性与Trunk的封装机制。

五. 实验中的问题与心得

一些细节注意

本次实验个人的问题大部分都在于一些小细节。

配置操作时需要注意把配置写回，有时候需要重启时可能会丢失一些配置，导致配置时容易陷入困惑

```
PC7> ping 6.48.1.1

host (6.48.1.1) not reachable

PC7> ping 6.48.1.1

host (6.48.1.1) not reachable

PC7> sh

NAME   IP/MASK              GATEWAY           MAC                LPORT  RHOST:PORT
PC7    0.0.0.0/0            0.0.0.0           00:50:79:66:68:06  10076  127.0.0.1:10077
       fe80::250:79ff:fe66:6806/64
```

可能会出现如上的问题，这时候排查许久链路问题一直无果也让实验进行的十分困难。

同时也可以多结合抓包工具，查看一下数据包在什么地方出现了问题，例如我在R2上添加单臂路由后，忘记在R2上配置RIP的直连网络（network 6.0.0.0），这时候R1的路由表会缺少6.0.0.0网段的信息，导致无法正常转发数据包，出现了PC7 ping PC1不通的现象，并且此时其他PC都是正常的，而且PC7 不跨路由器ping 其他PC都是正常的。这时候可以根据这个现象，在R1到R2之间的链路上进行抓包，可以看到缺少reply包。这时候就可以尝试定位问题。

还有一个问题就是最开始连接拓扑时没有仔细查看，在GNS3中接线时不小心接错了一些接口。后面配置时发现有些PC没法ping通自己的网关，排查后发现接口连错了，重新配置接口后会奇怪的导致重新连接的接口没法正常no shutdown，必须先shutdown错配的接口，这时候卡了我很久，一直没法让PC ping通自己的网关，抓包也只能看到发出的ARP包没有回应，还是让人摸不着头脑。

所以做实验时还是要注意细节，最好连好拓扑后也在图上标一下课件上要求的接口，这样也方便参考和检查。

六. 实验思考

1. 如何在同一个局域网中，配置两个IP网段（要求这两个网段的设备可以互相ping通，采用两种以上的配置方法）

方法一：使用三层交换机或路由器实现多网段路由

- 配置思路：

  在交换机或路由器上为不同IP网段的设备配置不同的接口或子接口，形成多个逻辑接口，并通过路由功能实现不同网段之间的数据转发。

- 具体实现：
  - 物理接口或 VLAN 子接口绑定不同IP网段地址。
  - 启用路由功能，使不同接口间相互路由。
- 示例：
  - 单臂路由
  - 多接口三层交换机

方法二：使用多层交换机的SVI（Switched Virtual Interface）

- 配置思路：

  在多层交换机上配置多个 VLAN，每个 VLAN 绑定一个虚拟接口（SVI），为不同 VLAN 分配不同的 IP 网段，交换机内部进行 VLAN 间路由。

- 具体实现：
  - 配置多个 VLAN，分别划分给不同设备。
  - 在交换机上启用路由功能，配置 SVIs 并分配 IP。

2. 选择两个不同VLAN中的PC机，中间要经过trunk链路连接的路由器，阐述互相ping时的完整传输流程。（包括交换机和路由器的简单处理过程，并且要指出VLAN标签的变化）

我们选取位于VLAN5和VLAN6的PC6和PC7进行分析，完整过程如下：

1. 发送方 PC6 生成 ICMP Echo Request

- PC6 根据目标 IP（PC7）判断目标不在本地网段，决定将数据包发送到默认网关（R2 的子接口 IP）。
- PC6 查询本地 ARP 表获取默认网关的 MAC 地址，如果没有则广播 ARP 请求。
- ARP 请求得到响应后，PC6 封装数据帧，发送到交换机端口。

2. 交换机（VLAN5 Access 端口和 Trunk 链路）

- 交换机收到带有 VLAN5 标签的数据帧，从 Access 端口接收时无 VLAN 标签。
- 在发送至 Trunk 接口时，交换机插入 IEEE 802.1Q VLAN 标签，标记 VLAN ID = 5。
- 帧通过 Trunk 传输至路由器。

3. 路由器（单臂路由子接口处理）

- 路由器子接口识别 VLAN5 的标签，剥离 VLAN 标签，提取 IP 数据包。
- 路由器根据路由表查找目的地址，确定下一跳为 VLAN6。
- 路由器封装数据包，给数据帧加上 VLAN6 标签，并通过 Trunk 端口发送回交换机。

4. 交换机（VLAN6 Access 端口）

- 交换机接收带有 VLAN6 标签的数据帧，从 Trunk 端口进入。
- 交换机根据 VLAN ID 去除 VLAN 标签，转发到对应的 Access 端口。
- 数据帧到达目标主机 PC7。

5. 目标 PC7 处理响应

- PC7 处理接收的 ICMP Echo Request，生成 Echo Reply。
- 通过相同路径，经过 VLAN6 和 VLAN5 的相反流程返回 PC6。

3. 请阐述VLAN、物理网络及IP网段的关系

VLAN（虚拟局域网）

- VLAN 是在二层交换机上通过逻辑划分实现的网络分段技术。
- 同一物理网络中的端口可划分到不同 VLAN，实现广播域的隔离。
- 不同 VLAN 间需要三层设备（路由器或三层交换机）实现通信。

物理网络

- 指实际的物理连线和设备，包括交换机端口、链路和物理拓扑结构。
- 一个物理网络可以承载多个 VLAN，利用 Trunk 端口和 VLAN 标签进行逻辑隔离和多路复用。

IP 网段

- IP 网段是三层网络的逻辑划分，依据 IP 地址和子网掩码定义。
- 通常情况下，一个 VLAN 对应一个 IP 网段，实现地址的层次化管理和便捷路由。
- 不同 VLAN 可对应不同的 IP 网段，实现物理分隔和逻辑隔离的配合。

总的来说，VLAN 用于在二层对物理网络进行逻辑划分以隔离广播域，而 IP 网段是在三层对这些逻辑网络进行地址划分与路由管理，二者依托同一物理网络协同工作，实现网络的逻辑隔离与互联。