实验 24_ARP 消息分析实验

学生姓名:林觉凯　　合作同学:无

实验地点:济事楼 330　　实验时间:2024.5.20

【实验目的】

ARP(地址解析协议)是一种用于将 IP 地址映射到 MAC 地址的协议。通过本次 ARP 消息分析实验，我们可以具体地了解到 ARP 协议的作用，同时理解 ARP 协议解决逻辑地址到物理地址的映射问题，以及 ARP 协议在局域网中的作用。

【实验原理】

ARP 协议

1. ARP 出现原因

ARP 协议是“Address Resolution Protocol”(地址解析协议)的缩写。其作用是在以太网环境中，数据的传输所依懒的是 MAC 地址而非 IP 地址，而将已知 IP 地址转换为 MAC 地址的工作是由 ARP 协议来完成的。在局域网中，网络中实际传输的是“帧”，帧里面是有目标主机的 MAC 地址的。在以太网中，一个主机和另一个主机进行直接通信，必须要知道目标主机的 MAC 地址。但这个目标 MAC 地址是如何获得的呢？它就是通过地址解析协议获得的。

所谓“地址解析”就是主机在发送帧前将目标 IP 地址转换成目标 MAC 地址的过程。ARP 协议的基本功能就是通过目标设备的 IP 地址，查询目标设备的 MAC 地址，以保证通信的顺利进行。

2. ARP 映射方式

2.1. 静态映射

静态映射的意思是要手动创建一张 ARP 表，把逻辑(IP)地址和物理地址关联起来。这个 ARP 表储存在网络中的每一台机器上。

例如，知道其机器的 IP 地址但不知道其物理地址的机器就可以通过查 ARP 表找出对应的物理地址。这样做有一定的局限性，因为物理地址可能发生变化：

(1)机器可能更换 NIC(网络适配器)，结果变成一个新的物理地址。

(2)在某些局域网中，每当计算机加电时，他的物理地址都要改变一次。

(3)移动电脑可以从一个物理网络转移到另一个物理网络，这样会时物理地址改变。要避免这些问题出现，必须定期维护更新 ARP 表，此类比较麻烦而且会

影响网络性能。

2.2. 动态映射

动态映射时，每次只要机器知道另一台机器的逻辑(IP)地址，就可以使用协议找出相对应的物理地址。已经设计出的实现了动态映射协议的有 ARP 和 RARP 两种。ARP 把逻辑(IP)地址映射为物理地址。RARP 把物理地址映射为逻辑(IP)地址。

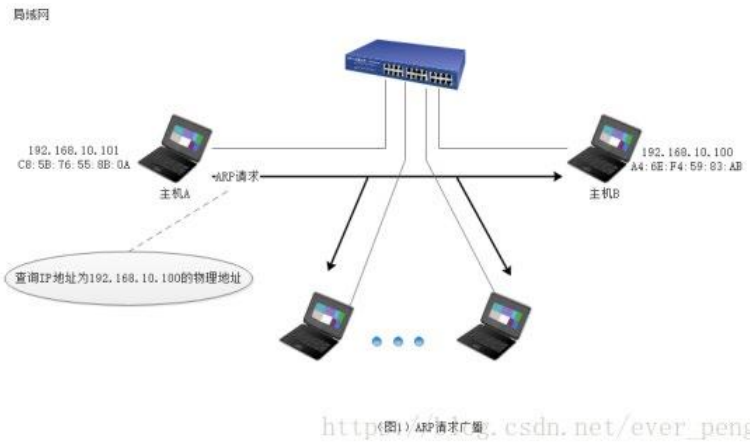
3. ARP 原理及流程

在任何时候，一台主机有 IP 数据报文发送给另一台主机，它都要知道接收方的逻辑(IP)地址。但是 IP 地址必须封装成帧才能通过物理网络。

这就意味着发送方必须有接收方的物理(MAC)地址，因此需要完成逻辑地址到物理地址的映射。而 ARP 协议可以接收来自 IP 协议的逻辑地址，将其映射为相应的物理地址，然后把物理地址递交给数据链路层。

3.1.ARP 请求

任何时候，当主机需要找出这个网络中的另一个主机的物理地址时，它就可以发送一个 ARP 请求报文，这个报文包好了发送方的 MAC 地址和 IP 地址以及接收方的 IP 地址。因为发送方不知道接收方的物理地址，所以这个查询分组会在网络层中进行广播。(见图 1)



(图1) ARP请求广播

https://blog.csdn.net/ever_peng

3.2.ARP 响应

局域网中的每一台主机都会接受并处理这个 ARP 请求报文，然后进行验证，查看接收方的 IP 地址是不是自己的地址，只有验证成功的主机才会返回一个

ARP 响应报文，这个响应报文包含接收方的 IP 地址和物理地址。这个报文利用收到的 ARP 请求报文中的请求方物理地址以单播的方式直接发送给 ARP 请求报文的请求方。(见图 2)

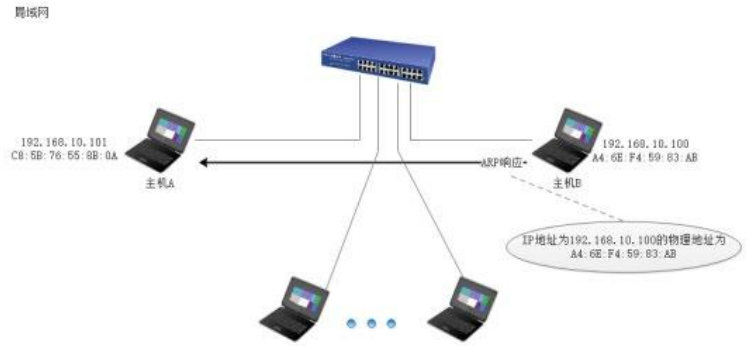


图2：ARP响应单播

http://blog.csdn.net/ever_peng

4.1.报文格式

硬件类型：16 位字段，用来定义运行 ARP 的网络类型。每个局域网基于其类型被指派一个整数。例如：以太网的类型为 1。ARP 可用在任何物理网络上。

协议类型：16 位字段，用来定义使用的协议。

例如：对 IPv4 协议这个字段是 0800。ARP 可用于任何高层协议

硬件长度：8 位字段，用来定义物理地址的长度，以字节为单位。例如：对于以太网的值为 6。

协议长度：8 位字段，用来定义逻辑地址的长度，以字节为单位。例如：对于 IPv4 协议的值为 4。

操作码：16 位字段，用来定义报文的类型。已定义的分组类型有两种：ARP 请求(1)，ARP 响应(2)。

源硬件地址：这是一个可变长度字段，用来定义发送方的物理地址。例如：对于以太网这个字段的长度是 6 字节。

源逻辑地址：这是一个可变长度字段，用来定义发送方的逻辑(IP)地址。例如：对于 IP 协议这个字段的长度是 4 字节。目的硬件地址：这是一个可变长度字段，用来定义目标的物理地址，例如，对以太网来说这个字段位 6 字节。对于 ARP 请求报文，这个字段为全 0，因为发送方并不知道目标的硬件地址。目的逻辑地址：这是一个可变长度字段，用来定义目标的逻辑(IP)地址，对于 IPv4 协议

这个字段的长度为 4 个字节。

4. ARP 协议报文字段抓包解析

4.1. 报文格式(见图 3)

（图3）ARP报文格式 http://blog.csdn.net/ever_peng

4.2. ARP 报文总长度

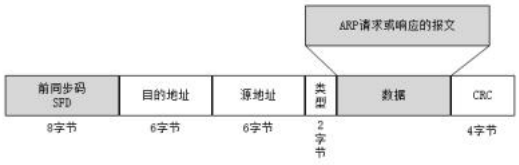
ARP 报文的总长度为 64 字节。首先要知道帧的概念 帧是在数据链路层传输的数据格式，比如以太网 v2，以太网 IEEE802.3 和 PPP 等。所以 Wireshark 抓到的帧是包含帧头的，即包含以太网 v2 的帧头，长 14 bytes；而 ARP 数据包的长度固定为 28 bytes；帧总长度=帧头+网络层包头+传输层报文头+应用数据；而 ARP 请求中 ARP 包已经是最高层，之上没有传输层和应用层，所以总长度为：帧总长度=帧头+ARP 包头=14+28=42 bytes；而真正发包的时为了保证以太网帧的最小帧长为 64 bytes，会在报文里添加一个 padding 字段，用来填充数据包大小。使用 wireshark 抓包时，抓到的包为 60 bytes。比以太网帧的最小帧长扫了 4 bytes，原因是因为 wireshark 抓包时不能抓到数据包最后的 CRC 字段。

CRC 字段是为了校验以太网帧的正确性。在数据包填充完成后，回去通过算法计算一个值放到数据包的 CRC 字段中。当接受端收到数据包后，会同样使用算法计算一个值，然后和 CRC 字段的值进行对比，查看是否相同。如果不同则证明数据包被更改，如果相同则证明数据包并未被更改。

4.3.报文封装

ARP 报文直接封装在数据链路帧中，例如，图 4 中，ARP 分组被封装在以

太网的帧中。注意，帧中的类型字段指出此帧所携带的数据是 ARP 报文。



https://blog.csdn.net/ever_peng

（图4）ARP报文封装

【实验设备】

硬件设备：济事楼 330 机房电脑和本人的笔记本电脑

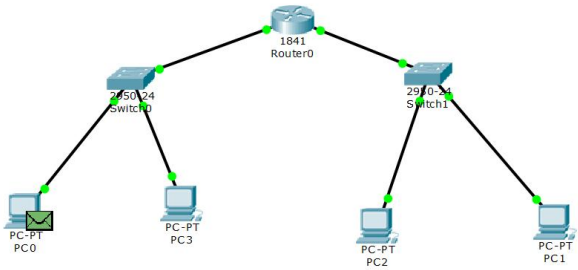
软件设备：Windows 操作系统和 Cisco Packet Tracer 网络仿真软件

【实验步骤】

1.按照实验的要求放置网络设备并完成连线(如下图);

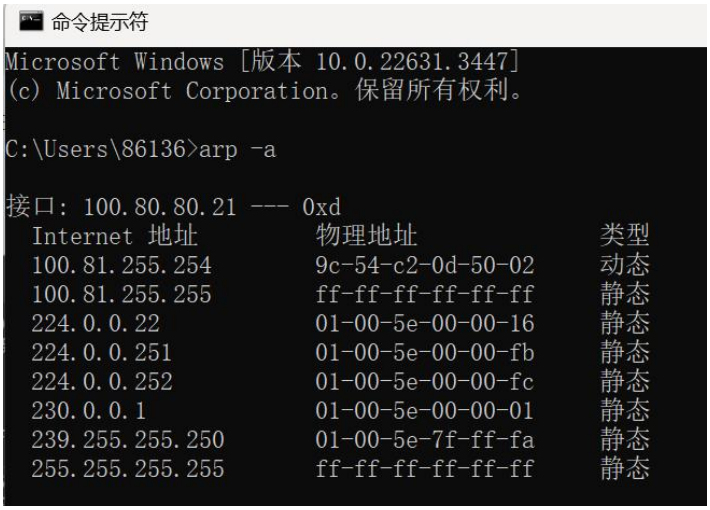
2.在仿真模式下发送请求，并抓取 ARP 报文并且分析报文;

3.通过抓包软件 Wireshark 抓取 ARP 报文并分析。

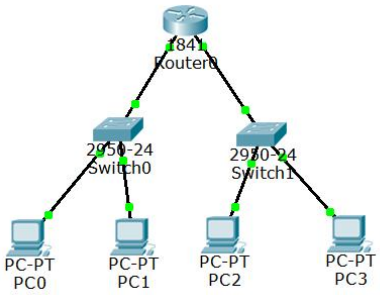


【实验现象】

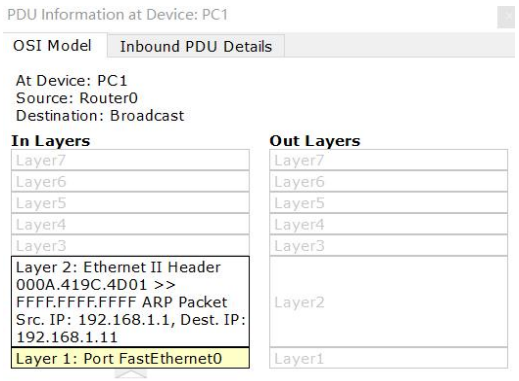
查看本人笔记本电脑的 ARP

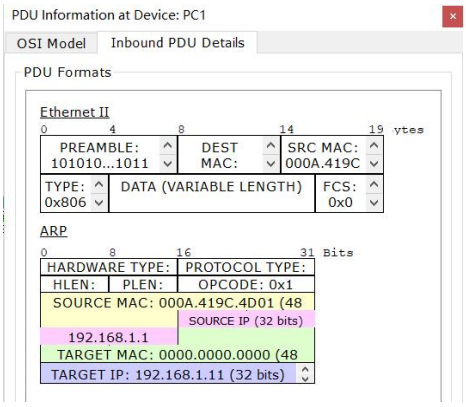


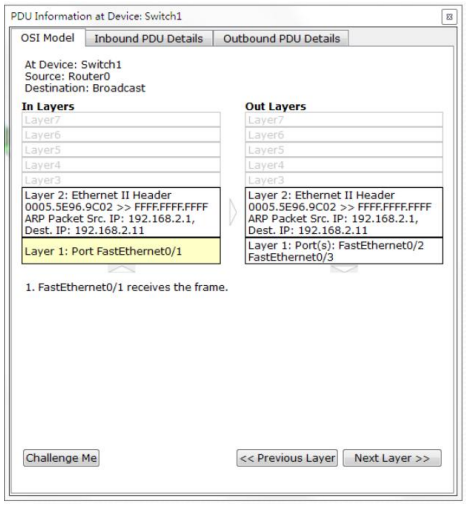
首先按照实验的要求完成拓扑图的连接与完善。

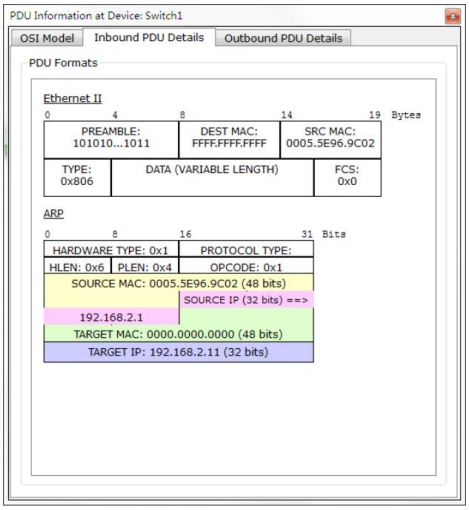


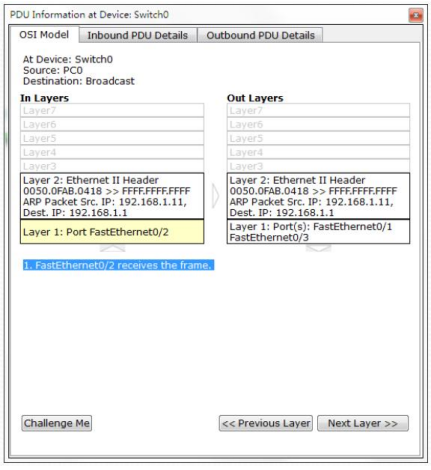
发送请求，并使用 Packet Tracer 抓取 ARP 报文并且分析报文;

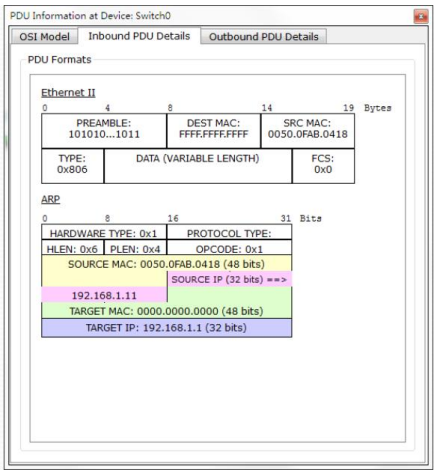








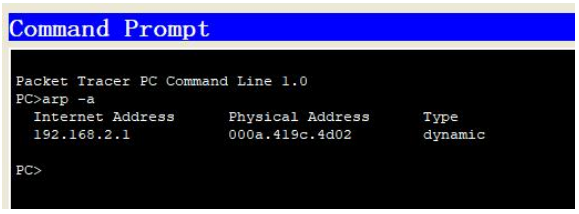




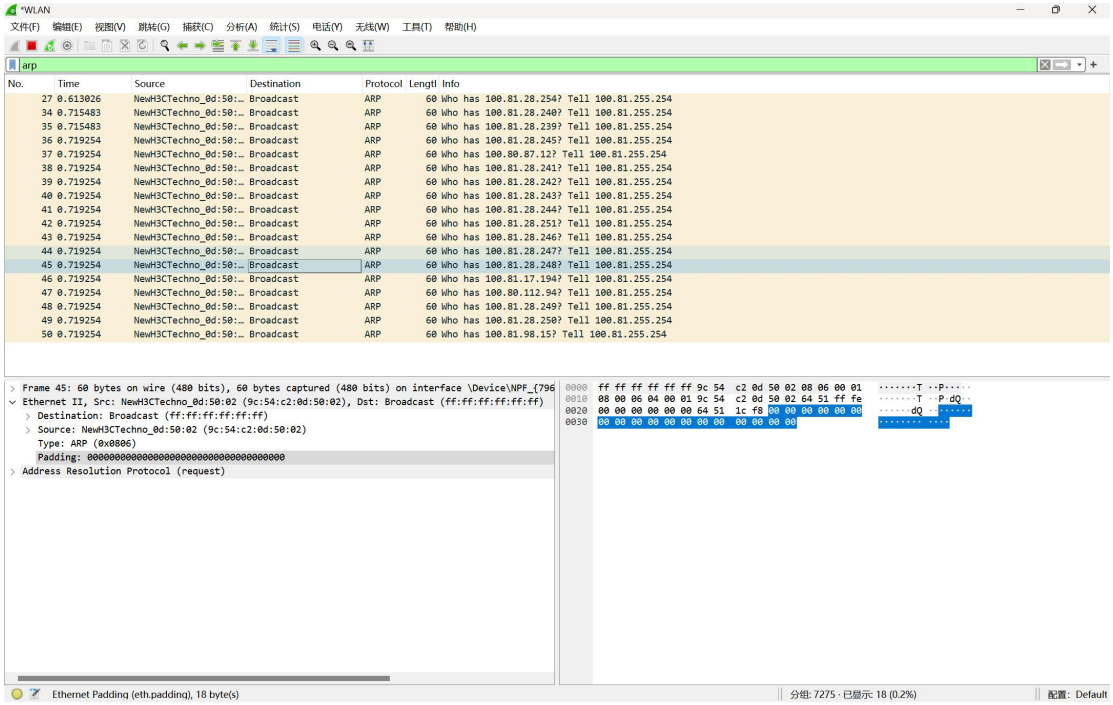
Packet Tracer 分析报文

查看终端的 ARP 命令

ARP -d



通过抓包软件 Wireshark 抓取 ARP 报文并分析。



00 01：目标硬件类型，这里是以太网。

08 00：目标协议类型，这里是 IPv4。

06：硬件地址长度，通常为以太网地址的长度，即 6 字节。

04：协议地址长度，IPv4 地址通常是 4 字节。

00 01：操作码，这里是请求(Request)。

9c 54 c2 0d 50 02：发送方以太网地址，即发送 ARP 请求的设备的 MAC 地址。

64 51 ff fe：发送方的 IPv4 地址，即发送 ARP 请求的设备的 IP 地址。

【分析讨论】

本次实验通过分析 ARP 请求和应答消息的数据包，我们深入理解地址解析协议的工作原理。实验中，我们学习了 ARP 消息的格式和含义，了解了如何通过 ARP 消息解析目标设备的 MAC 地址。这有助于我们更好地理解局域网中设备之间的通信过程，并学会利用 ARP 消息来诊断和解决网络通信中的问题。

ARP 的工作原理可以简要概括为以下步骤：

ARP 请求： 当主机 A 需要与目标主机 B 通信时，但不知道目标主机 B 的 MAC 地址时，主机 A 会在本地网络广播一个 ARP 请求，询问网络中是否有与目标 IP 地址相对应的 MAC 地址。

ARP 应答： 目标主机 B 收到 ARP 请求后，如果知道自己的 IP 地址，就会直接发送一个 ARP 应答给主机 A，包含自己的 MAC 地址。这个 ARP 应答是单播发送的，只有主机 A 会收到。

MAC 地址映射： 主机 A 收到 ARP 应答后，会将目标 IP 地址和 MAC 地址的映射关系存储在本地的 ARP 缓存中，以便将来通信时直接使用。

ARP 缓存： 主机 A 在 ARP 缓存中保存了目标 IP 地址和 MAC 地址的映射关系，这样在以后与目标主机 B 通信时，就不需要再次发送 ARP 请求，直接从 ARP 缓存中获取目标主机 B 的 MAC 地址即可。

ARP 的工作原理简单而直接，它解决了 IP 地址到 MAC 地址的映射问题，使得设备能够在局域网上相互通信。

实验 28_组网技术

学生姓名:林觉凯　　合作同学:无

实验地点:济事楼 330　　实验时间:2024.5.20

【实验目的】

单臂路由(Single Arm Routing)是一种网络设计模式，它只有一个物理接口用于连接到局域网或广域网，而该接口同时扮演着路由器和网络设备的角色。通过本次组网技术实验，我们可以初步了解单臂路由的工作原理和优缺点，以及它在网络设计和部署中的实际应用场景。

【实验原理】

单臂路由

1.单臂路由的概述

“单臂路由(router-on-a-stick)是指在路由器的一个接口上通过配置子接口(或“逻辑接口”，并不存在真正物理接口)的方式，实现原来相互隔离的不同 VLAN(虚拟局域网)之间的互联互通。”交换机连接主机的端口为 access 链路交换机连接路由器的端口为 Trunk 链路。

2.单臂路由应用

VLAN 能有效分割局域网，实现各网络区域之间的访问控制。但现实中，往往需要配置某些 VLAN 之间的互联互通。比如，某个公司划分为管理层、销售部、财务部、人力部、研发部、审计部，并为不同部门配置了不同的 VLAN，部门之间不能相互访问，有效保证了各部门的信息安全。但经常出现管理部门需要跨越 VLAN 访问其他各个部门，这个功能就由单臂路由来实现。另外，单臂路由也具有较高的性价比！

3.单臂路由主要配置过程

(1)配置 VLAN10 的网关

Router(config)#interface fa0/0.1 //进入第 1 个子接口，进行配置

Router(config-subif)#encapsulation dot1q 10 // 为这个接口配置 802.1Q 协议封装，最后面的 10 是 vlan 号;

Router(config-subif)#ip address 192.168.1.254 255.255.255.0 //为该接口划分网关地址和掩码。

Router(config-subif)#exit

(2)配置 VLAN20 的网关

Router(config)#interface fa0/0.2 //进入第 1 个子接口，进行配置

Router(config-subif)#encapsulation dot1q 20 //为这个接口配置 802.1Q 协议封装，最后面的 20 是 vlan 号;

Router(config-subif)#ip address 192.168.2.254 255.255.255.0 //为该接口划分网关地址和掩码。

Router(config-subif)#exit

(3)配置 VLAN30 的网关

Router(config)#interface fa0/0.3 //进入第 1 个子接口，进行配置

Router(config-subif)#encapsulation dot1q 30 //为这个接口配置 802.1Q 协议封装，最后面的 30 是 vlan 号;

Router(config-subif)#ip address 192.168.3.254 255.255.255.0 //为该接口划分网关地址和掩码。

Router(config-subif)#exit

【实验设备】

硬件设备：济事楼 330 机房电脑和本人的笔记本电脑

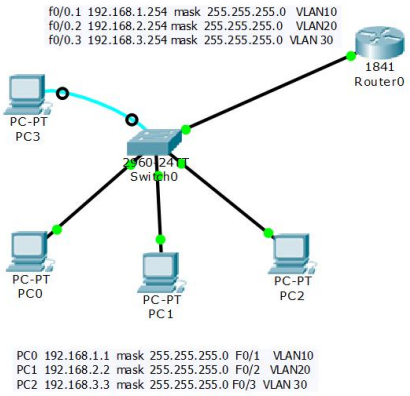
软件设备：Windows 操作系统和 Cisco Packet Tracer 网络仿真软件

【实验步骤】

1.按照下图所示完成网络拓扑关系图的连接。

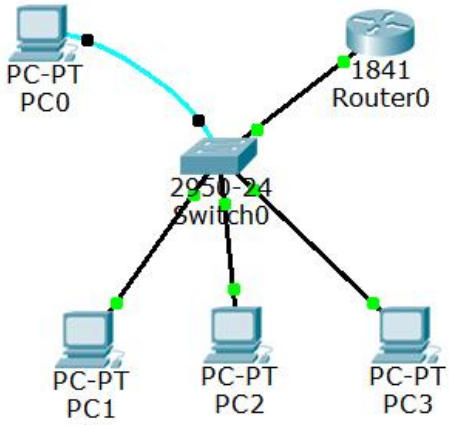
2.输入相应的命令(串)新建 VLAN10、VLAN20、VLAN30，并且分配相应的端口，完成相应的 VLAN 配置，并且观察是否可以 ping 连通。(即模拟每个部门在同一网段，部门内机器彼此可以互访，不同部门之间平时网络相互隔离)

3.配置单臂路由，完成相应的任务，最后再观察是否可以 ping 连通。(即模拟年末，公司这三个部门之间网络需保持互通，以便网络互通“联欢”)

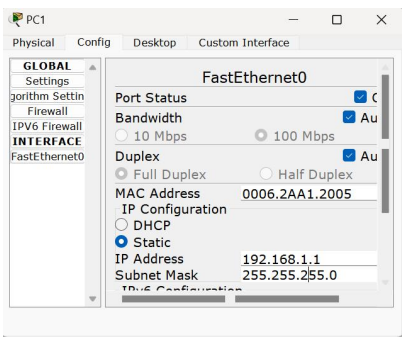


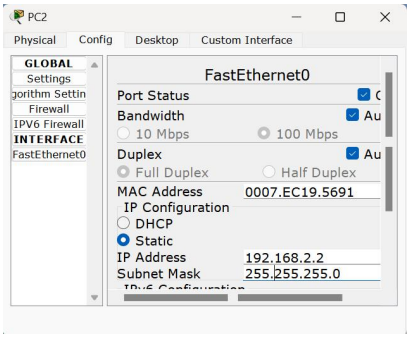
【实验现象】

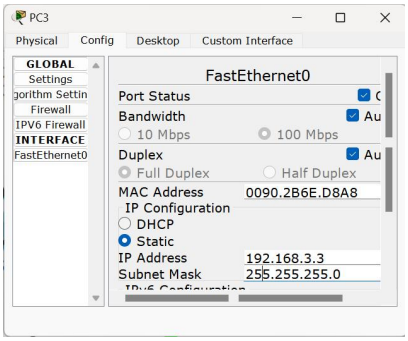
首先打开模拟仿真软件，完成网络拓扑关系图的连接。



完成各个 PC 机的 IP 地址、子网掩码和网关。







PC1:IP 地址：192.168.1.1；子网掩码：255.255.255.0；网关：192.168.1.254

PC2:IP 地址：192.168.2.2；子网掩码：255.255.255.0；网关：192.168.2.254

PC3:IP 地址：192.168.3.3；子网掩码：255.255.255.0；网关：192.168.3.254

接下来按照先前的实验内容配置 vlan10、vlan20、vlan30。

```
Switch(vlan)#vlan 10
VLAN 10 added:
    Name: VLAN0010
Switch(vlan)#exit
APPLY completed.
Exiting....
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTI
Switch(config)#interface f0/1
Switch(config-if)#switchport a vlan 10
Switch(config-if)#exit
```

```
Switch#en
Switch#vlan database
% Warning: It is recommended to configure VLAN from config
  as VLAN database mode is being deprecated. Please consul
  documentation for configuring VTP/VLAN in config mode.

Switch(vlan)#vlan 20
VLAN 20 added:
    Name: VLAN0020
Switch(vlan)#exit
APPLY completed.
Exiting....
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTI
Switch(config)#interface f0/2
```

```
Switch(vlan)#vlan 30
VLAN 30 added:
    Name: VLAN0030
Switch(vlan)#exit
APPLY completed.
Exiting....
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTI
Switch(config)#interface f0/3
Switch(config-if)#switchport a vlan 30
Switch(config-if)#exit
Switch(config)#exit
```

以 PC1 为例，ping 其余两台 PC 机。

```
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

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

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
PC>ping 192.168.3.3

Pinging 192.168.3.3 with 32 bytes of data:

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

Ping statistics for 192.168.3.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

我们发现，此时是 ping 不通的，这是因为，当配置了不同的 VLAN 后，同一个交换机下的 PC 处于不同的 VLAN 中，而 VLAN 之间默认是隔离的。这样就实现了模拟每个部门在同一网段，部门内机器彼此可以互访，不同部门之间平时网络相互隔离这种情况。

接着我们利用路由器配置对应的单臂路由。

```
Router>
Router>en
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fa0/0.1
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed state
 to up

Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip address 192.168.1.254 255.255.255.0
Router(config-subif)#exit
```

```
Router(config)#interface fa0/0.2
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.2, changed state
 to up

Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.2.254 255.255.255.0
Router(config-subif)#exit
```

```
Router(config)#interface fa0/0.3
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.3, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.3, changed state
 to up

Router(config-subif)#encapsulation dot1q 30
Router(config-subif)#ip address 192.168.3.254
% Incomplete command.
Router(config-subif)#exit
```

我们这时候再次以 PC1 为例，ping 其余两台 PC 机。

```
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=5ms TTL=128
Reply from 192.168.2.2: bytes=32 time=1ms TTL=128
Reply from 192.168.2.2: bytes=32 time=5ms TTL=128
Reply from 192.168.2.2: bytes=32 time=8ms TTL=128

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 8ms, Average = 4ms
```

```
PC>ping 192.168.3.3

Pinging 192.168.3.3 with 32 bytes of data:

Reply from 192.168.3.3: bytes=32 time=5ms TTL=128
Reply from 192.168.3.3: bytes=32 time=4ms TTL=128
Reply from 192.168.3.3: bytes=32 time=1ms TTL=128
Reply from 192.168.3.3: bytes=32 time=5ms TTL=128

Ping statistics for 192.168.3.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 5ms, Average = 3ms
```

我们发现，配置好单臂路由后，可以互相 ping 通。这样就实现了模拟在年末，公司这三个部门之间网络需保持互通，以便网络互通“联欢”。这是因为当设备位于不同部门时，单臂路由器会根据其路由表进行路由处理，并将流量转发到正确的部门。单臂路由器具有路由不同部门之间的流量的能力，因此可以实现不同部门之间的连通。

如果有更多的终端，也可以依照上文的方法配置相应的单臂路由，实现更多部门终端之间的互通。

【分析讨论】

本次实验的内容与学期初的一个实验相联系，这也让我复习了先前了解到的知识。通过本次实验，我学会了如何配置和管理单臂路由器，以及如何处理本地流量和路由外部流量。我不仅提升了对单臂路由技术的理解，也提高了网络配置和管理的能力。同时，知识的互相交互，前后应用，也让我受益匪浅。

单臂路由是一种网络设计模式，通常用于实现特定的网络功能或优化网络架构。在单臂路由中，只有一个物理接口用于连接到网络，同时该接口充当了路由器和网络设备的角色。它的工作原理如下：

接收流量： 单臂路由器接收来自网络的所有流量，包括本地流量和路由外部流量。处理本地流量： 如果接收到的流量是本地网络之间的通信，则单臂路由器直接转发流量，无需进行路由处理。路由外部流量： 如果接收到的流量需要通过路由器转发到另一个网络，则单臂路由器会根据其路由表进行路由处理，并将流量转发到正确的网络。NAT 转换： 在某些情况下，单臂路由器可能会执行网络地址转换（NAT），以确保流量离开局域网时具有正确的源地址。

单臂路由具有以下优点：

节省成本： 单臂路由器通常只需要一个物理接口，因此在硬件和部署成本上相对较低。简化配置： 单臂路由器只需要管理一个物理接口的配置，而不必担心多个接口的配置和管理。灵活性： 单臂路由器可以灵活地部署于各种网络环境中，并且可以根据需求进行快速调整和扩展。

单臂路由的应用场景：

虚拟专用网(VPN)：单臂路由可用于建立 VPN 连接，实现远程访问和安全通信。访问控制列表(ACL)：单臂路由可用于配置和管理网络访问控制列表，控制流量的访问权限。网络监控和流量分析：单臂路由可用于监控网络流量，并进行分析和管理。

在实际应用中，单臂路由可以作为组网技术的一部分，用于实现特定的网络架构和功能。例如，在建立虚拟专用网(VPN)或设置访问控制列表(ACL)时，可以使用单臂路由来简化网络设计，并且通过组网技术将其与其他设备整合在一起，以实现完整的网络解决方案。