ARP缓存表

- 为了实现ip地址与mac地址的查询和转换，arp协议引入了arp缓存表的概念
- 这个表包含了**ip地址**到**mac地址**的**映射关系**，表中记录了<IP地址，MAC地址>对，称之为arp表项
- 当需要发送数据时，主机会根据数据报中的**目标IP地址**信息，然后arp缓存表中查找**对应的MAC地址**，最后通过网卡将数据发送出去
- 注意：arp缓存表中每一项都被设置了生存时间，一般是20分钟，从被创建开始计时，到时则清除(arp -a：可查看arp缓存表)

icmp协议

icmp介绍

- 它是ipv4协议簇中的一个子协议，用于在ip主机，路由器之间传递**控制消息**
- 控制消息是指**网络通不通，主机是否可达，路由是否可用**等网络本身信息
- 这些控制消息虽然并**不传输用户数据**，但是对于用户数据的传递起着重要作用

交换机工作原理

当交换机收到一个数据帧后：

- 首先学习帧中的源MAC地址来形成MAC地址表
- 检查帧中的目标MAC地址，并匹配MAC地址表：如果有**匹配项**则**单播**转发；如果**无匹配项**，则除接收端口外，**广播**转发
- MAC地址表的老化时间默认为300s

◆ **交换机的基本工作模式**

1. 用户模式：

   switch>

   可以查看交换机的基本信息，但不能修改配置

```
Switch>
Switch>
```

用户模式

2. 特权模式：

   switch>enable

   switch#

   可以查看所有配置，但不能修改配置，可以做测试、保存、初始化等操作

```
Switch>enable
Switch#
Switch#
```

特权模式

3. 全局配置模式：

   switch# configure terminal

   switch(config)#

   可以修改配置，且全局生效，提供影响整个系统运行的命令

```
Switch#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#
Switch(config)#
```

全局配置模式

◆ **交换机的基本配置**

➢ 配置主机名

```
Switch(config)#hostname icq
```

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

➢ 设置登录密码

```
icq(config)#line console 0 #进入控制台状态
icq(config-line)#password icq #设置登录口令为icq
icq(config-line)#login  #允许登录
icq(config-line)#exit
```

```
User Access Verification

Password:

icq>
```

➢ 保存配置：

```
icq>en
icq#write
Building configuration...
[OK]
```

```
icq>
icq>en
icq#write
Building configuration...
[OK]
icq#
```

```
设置用户特权密码
icq> en
icq# conf t
icq(config)# enable password 密码
icq(config)# enable secret 密码
icq# write

查看缓存表
icq# show mac-address-table
```

◆ **交换机的基本配置**

➢ 删除配置

1. 在哪配置的，就在哪里删除
2. 命令前加：no 空格
3. 原命令中有参数，并且具有唯一性，则删除时不需要加参数

```
icq#conf t
icq(config)# hostname test1

icq#conf t
icq(config)# no hostname
Switch(config)#
```

```
test1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
test1(config)#no hostname
Switch(config)#
```

➢ 清除/初始化配置

```
Switch>en
Switch#
Switch#erase startup-config
```

```
Switch#
Switch#erase startup-config
Erasing the nvram filesystem will remove all configuration
files! Continue? [confirm]y[OK]
Erase of nvram: complete
%SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
Switch#
Switch#
```

路由表的形成

路由表：

- 路由器中维护的**路由条目的集合**
- 路由器根据路由表做**路径选择**

路由表的形成：

- 直连网段：配置IP地址>端口处于up状态>形成直连路由
- 非直连网段：对于非直连网段，需要**静态路由**或**动态路由**，将网段**添加到路由表**中

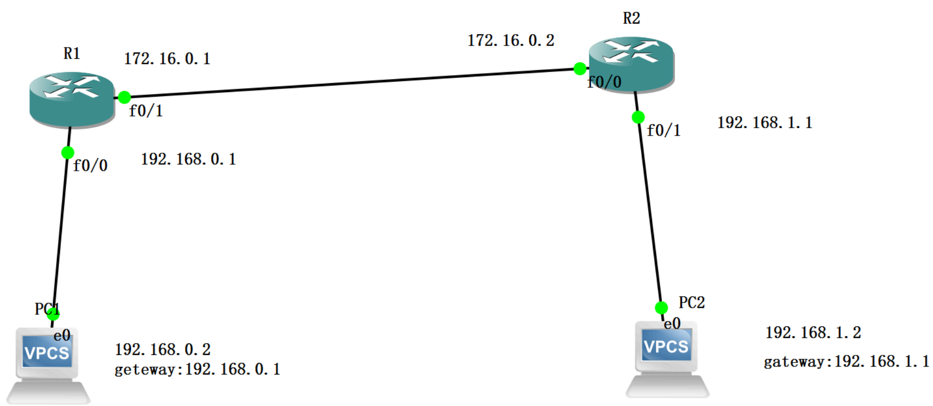
静态路由：

- 由管理员手工配置的，是**单向**的，缺乏灵活性

- 数据包如果要到达**非直连网络**需要在路由表中**添加条目**，静态路由需要**手动添加**路由条目
- 静态路由的配置：

```
Router(config)# ip route 目标网段 子网掩码 下一跳IP
```

下面配置静态路由：




```
正常情况下，PC1是没有办法ping通PC2的，我们需要配置静态路由

连线，配置如图所示的PC1，PC2的IP和网关
PC1> ip 192.168.0.2/24 192.168.0.1
PC2> ip 192.168.1.2/24 192.168.1.1
```

```
PC1> ip 192.168.0.2/24 192.168.0.1
Checking for duplicate address...
PC1 : 192.168.0.2 255.255.255.0 gateway 192.168.0.1
```

```
PC2> ip 192.168.1.2/24 192.168.1.1
Checking for duplicate address...
PC1 : 192.168.1.2 255.255.255.0 gateway 192.168.1.1
```

```
然后配置交换机（这里用路由器代替）

配置R1：
Router> enable
Router# conf t
Router(config)# int f0/0
Router(config-if)# ip add 192.168.0.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# exit
Router(config)# int f0/1
Router(config-if)#  ip add 172.16.0.1 255.255.255.0
Router(config-if)#  no shut
Router(config-if)#  exit

配置R1静态路由：
Router(config)# ip route 192.168.1.0 255.255.255.0 172.16.0.2
Router(config)# end
Router# write memory
```

```
配置R2:
Router> enable
Router# conf t
Router (config)# int f0/0
Router(config-if)# ip add 172.16.0.2 255.255.255.0
Router (config-if)# no shut
Router (config-if)# exit
Router(config)# int f0/1
Router(config-if)#  ip add 192.168.1.1 255.255.255.0
Router (config-if)# no shut
Router(config-if)#  exit

配置R2静态路由:
Router(config)# ip route 192.168.0.0 255.255.255.0 172.16.0.1
Router(config)# end
Router# write memory
```

配置R1

```
R1#enable
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.0.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Mar  1 00:07:54.175: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*Mar  1 00:07:55.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R1(config)#int f0/1
R1(config-if)#ip add 172.16.0.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
*Mar  1 00:08:24.959: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state t
o up
*Mar  1 00:08:25.959: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/1, changed state to up
R1(config)#
```

配置R1静态路由

```
R1(config)#ip route 192.168.1.0 255.255.255.0 172.16.0.2
R1(config)#end
R1#
*Mar  1 00:09:31.415: %SYS-5-CONFIG_I: Configured from console by console
R1#write memory
Building configuration...
[OK]
```

配置R2

```
R2#enable
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 172.16.0.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Mar  1 00:09:46.815: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state t
o up
*Mar  1 00:09:47.815: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/0, changed state to up
R2(config)#int f0/1
R2(config-if)#ip add 192.168.1.1 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#
*Mar  1 00:10:56.919: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state t
o up
*Mar  1 00:10:57.919: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthern
et0/1, changed state to up
```

配置R2静态路由

```
R2(config)#ip route 192.168.0.0 255.255.255.0 172.16.0.1
R2(config)#end
R2#wir
*Mar  1 00:12:24.063: %SYS-5-CONFIG_I: Configured from console by console
R2#write memory
Building configuration...
[OK]
```

测试

```
PC2> ping 192.168.0.2
84 bytes from 192.168.0.2 icmp_seq=1 ttl=62 time=60.730 ms
84 bytes from 192.168.0.2 icmp_seq=2 ttl=62 time=61.135 ms
84 bytes from 192.168.0.2 icmp_seq=3 ttl=62 time=61.090 ms
84 bytes from 192.168.0.2 icmp_seq=4 ttl=62 time=61.252 ms
84 bytes from 192.168.0.2 icmp_seq=5 ttl=62 time=60.918 ms

PC2>
```

```
PC1> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=1 ttl=254 time=45.704 ms
84 bytes from 192.168.1.1 icmp_seq=2 ttl=254 time=44.899 ms
84 bytes from 192.168.1.1 icmp_seq=3 ttl=254 time=45.934 ms
84 bytes from 192.168.1.1 icmp_seq=4 ttl=254 time=44.696 ms
84 bytes from 192.168.1.1 icmp_seq=5 ttl=254 time=45.513 ms
```

通过以上静态路由的配置可以发现，原来无法ping通的两个ip能相互ping通了！

VLAN技术原理

- 在物理网络上划分出**逻辑网**，对应**OSI模型第二层**
- VLAN划分不受端口物理位置限制，VLAN和普通物理网络有同样属性
- 第二层数据单播，广播只在一个VLAN内转发，不会进入其它VLAN中
- 一个VLAN = 一个广播域 = 一个网段

作用：

- 1.安全性，减少保密信息遭到破坏的可能性
- 2.节约成本，无需昂贵的网络升级
- 3.提高性能，将二层网络划分成多个广播域，减少不必要的数据流
- 4.缩小广播域，减少一个广播域上的设备数量
- 5.提升管理效率

vlan Trunk：

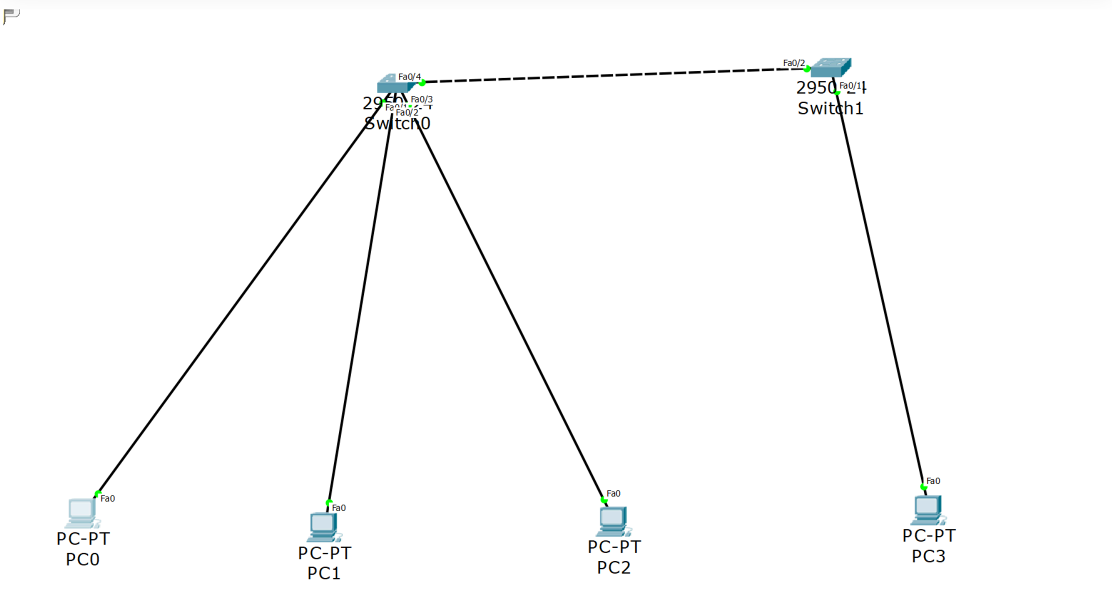
- Trunk是在两个**网络设备之间**，承载**多于一种VLAN**的端到端的连接，将VLAN延申至**整个网络**
- 作用：允许所有vlan数据通过trunk链路
- 方法：通过在数据帧上加标签，来区分不同的VLAN数据

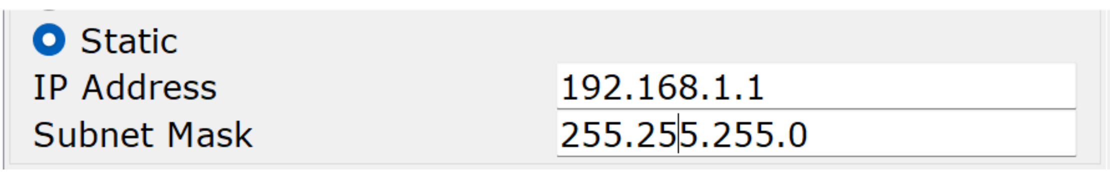
交换机端口链路类型：

- 接入端口：也称为access端口，一般用于**连接pc**，只能属于某一个vlan，也只能传输一个vlan数据
- 中继接口：也成为trunk接口，一般用于**连接其他交换机**，属于**公共端口**，允许所有vlan数据通过

配置VLAN

```
设置静态ip，如图(只显示一个，其它pc机类似，分别为192.168.1.x x=1,2,3,4)
```






```
配置好了以后用PC1 ping其他pc机发现能ping通
```

```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=1ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128
Reply from 192.168.1.3: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:

Reply from 192.168.1.4: bytes=32 time=0ms TTL=128
Reply from 192.168.1.4: bytes=32 time=0ms TTL=128
Reply from 192.168.1.4: bytes=32 time=0ms TTL=128
Reply from 192.168.1.4: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

创建vlan

```
Switch(config-if)#vlan 10
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#exit
Switch(config)#vlan 30
Switch(config-vlan)#exit
```

查看vlan

```
Switch#sh vlan b

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
10   VLAN0010                         active
20   VLAN0020                         active
30   VLAN0030                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#
```

配置

```
将端口加入到vlan中 :
int f0/x
switchport access vlan ID
exit
```

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#int f0/1
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#int f0/2
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#int f0/3
Switch(config-if)#switchport access vlan 30
Switch(config-if)#exit
```

查看vlan表(可以发现多了 Fa0/1/2/3)

```
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/4, Fa0/5, Fa0/6, Fa0/7
                                                Fa0/8, Fa0/9, Fa0/10, Fa0/11
                                                Fa0/12, Fa0/13, Fa0/14, Fa0/15
                                                Fa0/16, Fa0/17, Fa0/18, Fa0/19
                                                Fa0/20, Fa0/21, Fa0/22, Fa0/23
                                                Fa0/24
10   VLAN0010                         active    Fa0/1
20   VLAN0020                         active    Fa0/2
30   VLAN0030                         active    Fa0/3
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

效果(可以发现均无法ping通了,被隔离了)

```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

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

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

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

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

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

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:

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

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

配置trunk端口

```
先打开0/1和0/2端口
```

```
Switch(config)#int f0/2
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to up
```

```
Switch(config-if)#int f0/1
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

> 此时可以发现pc0和pc1能相互ping了

```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=9ms TTL=128
Reply from 192.168.1.2: bytes=32 time=10ms TTL=128
Reply from 192.168.1.2: bytes=32 time=4ms TTL=128
Reply from 192.168.1.2: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 5ms
```

总结

VLAN就类似一堵墙阻止通信的,和Trunk就是用于给通信开后门的!!!