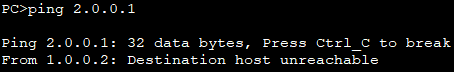
姓名：付若轩

年级：2019

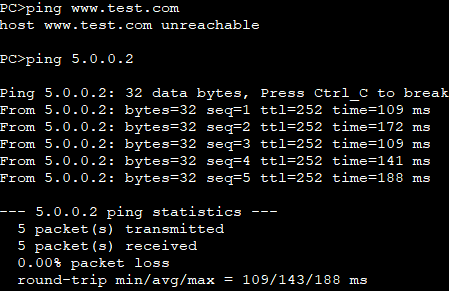
班级：软件工程2班

问题一

如果不配置网关，三台PC的报文只能在1.0.0.0网段的局域网中传输，无法访问外部网络，结果如下图。



因为网络是用IP地址标识主机的，如果不配置DNS服务器，在ping www.test.com的时候不能正确地将域名转换为IP地址，也就无法访问到域名的web服务器。结果如下图。



问题二

使用到的协议：ICMP, ARP, Ethernet

原因：Ping命令是工作在TCP/IP网络结构中应用层的命令，主要是向目的主机发送ICMP请求报文，从目的主机接收ICMP响应报文。在以太网环境中，数据传输依赖MAC地址，ARP用于MAC地址和IP地址的转换。网络中的帧使用Ethernet协议进行封装。

使用到的技术：子网划分，交换机表的转发和过滤

原因：不同子网的流程表现会有所不同，如果在同一子网中则不涉及路由，反之则涉及。因为局域网中存在多个主机，需要交换机来进行转发和过滤帧。

PC1 ping PC2 的主要流程如下：

1. Ping命令使PC1建立ICMP请求报文包含数据包和PC2的IP地址转交给网络层，网络层将PC2的IP地址作为目的地址，本机IP地址作为源地址，加上头部控制信息构建IP数据报
2. PC1通过自己的子网掩码与IP地址5.0.0.2检查发现目标主机与自己在同一网段。PC1检查自己的ARP缓存中是否存在PC2的IP地址对应的MAC地址，如果有则将将数据报封装入帧中并将PC2的MAC地址写入帧中，PC1发送Ethernet帧到局域网中这个MAC地址对应的主机。如果没有，则进入第3步
3. PC1在局域网中广播包含PC1的IP地址，MAC地址以及PC2的IP地址的ARP请求报文
4. 交换机接收到来自PC1的入帧，在自己的交换机表中存储该帧源地址字段中的MAC地址，该帧到达的接口以及该帧到达的时间
5. ARP请求报文被所有运行在局域网中主机的ARP模块接收。ARP模块检查自己的IP地址和ARP请求报文中的地址是否一致，若一致则接收进行处理并向PC1单播发送写入了自己MAC地址的ARP响应报文，将PC1的IP地址与MAC地址映射放入自己的ARP缓存；若不一致则丢求这个包
6. 交换机接收到来自PC2的入帧，在自己的交换机表中存储该帧源地址字段中的MAC地址，该帧到达的接口以及该帧到达的时间
7. PC1接收到来自PC2的ARP响应报文，将PC2的IP地址与MAC地址映射放入自己的ARP缓存
8. PC1经过交换机发送ICMP请求报文到PC2
9. PC2接收到ICMP请求报文后发送ICMP响应报文

问题三

协议：ICMP, ARP, Ethernet，静态路由，缺省路由

原因：选择ICMP, ARP, Ethernet的理由同问题二。因为涉及不同网段之间的互联互通，所以需要使用路由协议来路由转发来自不同网段的数据包。路由协议中包含直连路由，静态路由，动态路由；本次拓扑示例采用了静态路由。使用缺省路由可以将静态路由的配置变得容易，但需要注意路由环路问题

技术：子网划分，交换机表的转发和过滤，路由选择技术

原因：选择子网划分以及交换机表的转发和过滤的原因同问题二。因为涉及到路由器的互联，需要使用路由选择技术来确定最佳路由路径。

PC1 ping 5.0.0.2的主要流程如下

1. Ping命令使PC1建立ICMP请求报文包含数据包和目标主机的IP地址5.0.0.2转交给网络层，网络层将PC2的IP地址作为目的地址，本机IP地址作为源地址，加上头部控制信息构建IP数据报
2. PC1通过自己的子网掩码与IP地址5.0.0.2检查发现目标主机与自己不在同一网段。PC1检查自己的ARP缓存中是否存在网关的IP地址对应的MAC地址，如果有则将将数据报封装入帧中并将网关的MAC地址写入帧中，PC1发送Ethernet帧到网关。如果没有，则在局域网中广播PC1的IP地址，MAC地址以及包含网关IP地址ARP请求；得到网关的ARP响应之后，在自己的ARP缓存中存储网关的IP地址以及MAC地址对应关系。交换机更新交换机表
3. PC1封装网络层数据报成帧转交给网关，网关收到数据包，检查目的IP地址，发现不是给自己的，决定要进行路由，然后查询路由表；查询到路由表中有对应ip地址则查询mac地址表。如果有记录则从对应接口进行转发，如果没有记录则检查ARP缓存，更新。
4. IP地址为5.0.0.2的主机从他的网关处接收到ICMP请求报文，向PC1发送ICMP响应报文。

问题四

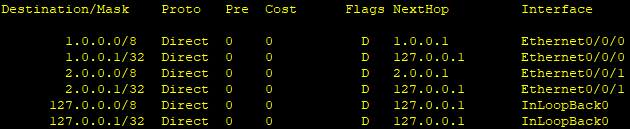
路由互通技术：直连路由，静态路由，动态路由

直连路由

基本原理

连接在各个路由器接口网段的路由，直连路由由链路层协议直接发现。直连路由不需要网络管理员维护，也不需要路由器通过路由算法计算获得。只要接口处于活动状态，且网络管理员在接口上绑定了IP地址，路由器就可以把通向该网段的路由信息加入路由表。

如图，在配置了接口的IP地址和子网掩码后，路由表自动更新到接口网段的路由。



优点

* 不需要网络管理员维护

缺点

* 直连路由只能使用于直接相连的路由器接口。如果只使用直连路由，网络拓扑会只限制在一个路由器内

静态路由

基本原理

网络管理员根据网络拓扑，手动地使用命令在路由器上配置的路由信息。静态路由不会因为网络拓扑改变而自身产生变化

优点：

* 因为不涉及到交换路由表，对路由器的CPU没有负担
* 不占用网络带宽
* 静态路由在默认情况下是私有的，安全性较高
* 网络管理员易于清楚地了解网络的拓扑结构，因为需要自己配置正确的路由信息

缺点：

* 网络管理员必须了解完整的网络拓扑
* 每次新增路由都需要手动配置，较繁琐
* 与缺省路由的混合使用不当会导致路由环路，解决方法见动态路由
* 对于大型网络，手动配置的工作量巨大

动态路由

基本原理

根据网络流量负载或拓扑发生变化时改变路由选择路径，自动建立自己的路由表。每一个路由器都存在一个动态路由选择算法，它可以周期性地运行或直接响应拓扑或链路开销的变化而运行。动态路由依赖于路由器之间动态的路由表交换，最终目的是通过路由表找到最佳路由，在路由表中存储到目的地的最佳下一跳。常用的动态路由算法由距离向量算法和链路状态算法。

距离向量算法

距离向量算法是一个迭代的，异步的，分布式的算法。该算法要求每个节点都要从一个或多个直接相连的邻居交换路由表进行计算，并且一直执行到邻居之间没有更多的信息可以交换

算法执行步骤如下

1. 对每个结点执行如下操作进行初始化
   1. 遍历所有网络中的目的地
      1. 估计到结点的距离向量。如果不是邻居，那么距离设为∞
   2. 遍历每个邻居
      1. 计算到邻居的距离向量
   3. 遍历每个邻居
      1. 发送本结点的距离向量到每个邻居
2. 循环
   1. 等待链路状态（开销）变化或接收到其他邻居结点的距离向量
   2. 遍历网络中的每个结点
      1. 根据bellman-ford方程更新本地x距离向量
   3. 如果距离向量存在变化
      1. 发送距离向量到每个邻居

链路状态算法

距离向量算法是一个全局式的算法。该算法要求在执行路由算法前，所有的网络拓扑和链路费用都是已知的，通过节点广播使所有节点都具备了等同的该网络的完整视图。获得视图后，通过LS算法可以计算出从源节点到网络任意节点的最低费用路径

算法执行步骤如下

1. 对每个结点执行如下操作进行初始化
   1. N’集合中存储所有已知从源节点到自己的最小开销（D（n））的节点
   2. 遍历除源节点外所有网络中的目的地
      1. 如果目的地是源节点的邻居，则更新他们之间的开销
      2. 否则他们之间的开销为无穷大
2. 循环
   1. 找到不在N‘中且从源节点到w的开销最小的节点w
   2. 将w加入到N‘中
   3. 更新与w节点相邻的节点n的D（n），且n不能再N‘中
      1. 使用Dijstra方程更新
   4. 直到所有节点在N‘中结束

在实际中使用中RIP，BGP（本拓扑不需要使用）采用了距离矢量路由算法，OSPF采用了链路状态算法。

RIP

RIP是一种分布式的基于距离向量的系统内部路由选择协议，RIP适用于中小型网络并采用了水平分割、毒性逆转和触发更新等工作机制防止路由环路。RIP分为两个版本：RIPV1以及RIPV2，RIPV2支持了CIDR，组播和认证

优点：

* 实现简单，开销较小
* RIPV2实现了认证，可以确认受信任的距离向量更新，更加安全

缺点：

* 距离向量协议传输整个路由表，报文大小和子网节点数成正比，节点越多，报文越大，浪费网络资源
* 收敛速度较慢
* 有跳数限制，超过16跳的路由被认为不可达

OSPF

OSPF使用泛洪链路状态信息和Dijkstra算法。每台路由器都构建了一个关于整个自治系统的网络拓扑。使用OSPF时，在链路状态发生变化时，路由器向自治系统内所有其他路由器泛洪地发送与本路由器相邻的所有路由器的链路状态信息。

优点

* 收敛快速
* 采用了认证系统，能够确认受信任的链路状态更新，更加安全
* 允许存在有相同开销的路径，一定程度上实现负载均衡
* 同时支持单播和多播

缺点

* 配置相对复杂

问题五

应用层

应用层中主要使用的协议是HTTP,DNS，主要使用的技术是非持续连接

**HTTP**

HTTP（HyperText Transfer Protocol）也被称为超文本传输协议是web应用的核心。客户端与服务器段运行在不同的端系统上通过HTTP报文进行会话，HTTP定义了这些报文的结构以及客户与服务器进行报文交换的方式。基本流程是当用户请求一个web页面时，浏览器向服务器发出对该页面中所包含对象的HTTP请求报文。服务器可以根据HTTP报文的请求行了解到请求的方法（如GET,POST,HEAD,DELETE），URL以及HTTP版本；服务器通过读取首部行了解应该如何发送响应报文。服务器接收到请求后，向用户发送包含这些对象的HTTP响应报文。响应报文的状态行可以让用户了解到协议版本字段、状态码和相应状态信息。状态码通常代表不同的响应类别(如200，400，401，404，)；首部行包含对TCP连接的操作以及一些响应报文和对象的信息；实体体中包含请求的数据。

HTTP使用TCP作为它的支撑运输协议用于保证可靠数据传输。

HTTP是一个无状态协议，HTTP不会保存关于客户的任何信息，如果用户在短时间内申请同一个对象多次，那么服务器也会发送该对象多次。

HTTP默认使用80端口

**DNS**

DNS（Domain Name System）也被称为域名系统。主要任务是将主机名转换为IP地址的目录服务。DNS协议建立在UDP上，当UDP数据包过大时自动使用TCP。DNS默认使用53端口

DNS的基本工作原理是使用分布式，层次数据库存储所有主机名对应IP地址的记录。基本流程如下：

* 当用户想知道example.com的IP地址时，首先向本地的DNS服务器发送一个DNS查询报文
* 本地DNS服务器将请求报文转发给根服务器，根服务器将根据前缀（com）向本地服务器发回负责该前缀的顶级域服务器（TLD服务器）IP地址列表
* 本地DNS服务器再向TLD服务器发送查询报文，TLD收到请求后返回example.com权威服务器的IP地址
* 本地DNS服务器再向权威服务器发送查询请求，权威服务器返回主机名和IP的映射

递归查询与迭代查询对查询流量与时延的影响很大，DNS缓存可以让本地DNS服务器暂时保存主机名和IP的映射关系

**非持续连接**

非持续连接的意思就是每一次用户对服务器的请求都是经过一个单独的TCP连接进行发的。

传输层

传输层协议主要运用了UDP和TCP

**UDP**

UDP（UserDatagramProtocol）是一个面向消息的传输层。除了完整性验证（校验和）不再提供任何保证可靠性传输的功能。这个特性使得UDP简单且轻量化同时也让UDP无法支撑无法容忍差错的数据的传输。

UDP常被用于支撑DNS与流媒体运用

UDP的工作原理非常简单，只是简单地将上层数据进行封装后向下层传输

**TCP**

TCP（Transport Control Protocol）是一种面向连接的，可靠的传输层协议。在应用进程可以向另一个应用进程发送数据之前，这两个进程必须先互相“握手”（建立TCP连接）即他们必须相互发送某些预备报文段，以建立确保数据传输的参数。

TCP的建立（三次握手），关闭（四次挥手）的基本工作流程如下

1. 客户端发送给服务器一个请求连接数据包，即发送了一个指向服务器目标端口的一个 SYN 位为 1 的TCP 报文，表示此刻正在建立TCP连接。
2. 服务器接收到客户端的连接请求之后，会回应一个 SYN 位为 1 的TCP 报文，表示同意连接。并且，会把 ACK 位也置 1 表示确认收到上次消息。
3. 客户端接收到服务器的同意连接的数据包之后，还要回复一个 ACK 为 1 的 TCP 报文，表示确认收到。
4. 至此TCP连接建立完成
5. 客户端发送给服务器一个请求释放连接的数据包，即发送了一个指向服务器目标端口的一个 FIN 位为 1 的TCP 报文，表示客户端没有数据要发送了，但是仍然可以接收数据；并且 ACK 位也为 1，表示对上次传输数据结果的确认。并且之后处去等待状态，等待服务器的两次回应。
6. 服务器接收到来自客户端的释放连接请求之后，回应一个ACK位为1的报文，表示确认收到
7. 服务器发送一个FIN为1的TCP报文，表示我没有需要发送的数据了
8. 客户端接收到服务器的同意释放连接的数据包之后，回复一个 ACK 为 1 的 TCP 报文，表示确认收到。
9. 至此，TCP连接结束完成

使用三次握手的原因是，三次通信时保证全双工通信的最小通信次数，防止失效的连接请求报文段又传输到了服务器端。

使用四次挥手的原因是终止两台主机上的数据传输与用于该连接的所有资源。

TCP是一个可靠数据传输协议，具体使用以下技术

* 序号，ACK：TCP给发送的每一个包进行编号，接收方对数据包进行排序，把有序的数据传输给应用层。ACK可以使接收方告诉发送方我期望收到的下一个字节的序号应该是哪一个。通过序号和ACK，TCP实现了累积确认（回退N步）即对发送方来说只要在超时时间内成功接到了接收方返回的一串应答中的最后一个，则包含了前面全部的确认信息。累积确认避免了某个应答的丢失造成的不必要重传。同时还采用了选择重传技术，
* 检验和：TCP保持首部和数据的检验和。如果收到段的检验和有差错（不全为1），TCP 将丢弃这个报文段和不确认收到此报文段。
* TCP接收端会丢弃重复的数据
* 流量控制：TCP连接的每一方存在一个接收缓存，TCP的接收端只允许发送端发送接收端缓冲区能接纳的数据。当接收方来不及处理发送方的数据，能提示发送方降低发送的速率，防止包丢失。TCP 使用的流量控制协议是可变大小的滑动窗口协议。发送方维护一个变量提示接收方还有多少缓存。本质上是一个速度匹配的服务。流量控制由接收方控制，发送方被迫与接收方同步速度
* 拥塞控制：网络变得拥塞时由于路由器缓存溢出导致了丢包，这时需要控制网络中大量的报文传输。为了解决网络拥塞或保证不出现拥塞，需要约束发送方的速率。拥塞控制要求发送方维护一个变量称为拥塞窗口，通过约束窗口的大小进行速率控制，拥塞控制算法内容包括慢启动，拥塞避免，以及快速重传

网络层

网络层使用到的协议是IP协议，路由转发协议；使用到的技术是子网划分

**IP协议**

IP协议包含IP编址方案，分组封装格式以及分组转发规则

IP协议是TCP/IP协议族的动力，它为上层协议提供无状态、无连接、不可靠的服务。IP协议拥有两个版本V4以及V6，大作业拓扑采用的是V4。IP协议的主要功能如下

* 用于指定IP通信的源端IP地址、目的端IP地址，指导IP分片和重组，以及指定部分通信行为。
* IP数据报的路由和转发。IP数据报的路由和转发发生在除目标机器之外的所有主机和路由器上。它们决定数据报是否应该转发以及如何转发。

**路由转发协议**

详情见第四题

**子网划分**

用来标识IP地址中哪部分为网络标识。子网掩码是一个长32bit值，连续的1表示lP地址的相应部分为网络地址，连续的0表示lP地址的相应部分为主机地址，大作业拓扑采用了CIDR（无类域间路由），形式如：1.0.0.2/8。当需要求得网段时，需要将IP地址与子网掩码做异或运算，结果就为网段

数据链路层

数据链路层使用到的协议主要是Ethernet，ARP协议，技术是CRC检测编码，交换机的转发和过滤

**Ethernet**

以太网是当今TCP/IP采用的主要的局域网技术，主要采用了CSMA/CD的媒体介入方法，也可以允许点对点链路，大作业拓扑使用的是点对点链路。

以太网将上层数据封装为以太网帧，加入目的MAC地址以及源地址并使用CRC检测帧中是否引入了差错，前同步码用于同步发送方与接收方的时钟。

以太网协议实现了数据链路层的数据传输和地址封装

**ARP协议**

ARP协议是地址解析协议（Address Resolution Protocol），是通过解析IP地址得到MAC地址的协议。ARP协议在主机或路由器中维护一个ARP缓存表，这个表包含IP地址到MAC地址的映射关系。当需要发送数据的时候，主机就会根据数据报中的目标IP地址信息，然后在ARP缓存表中进行查找对应的MAC地址，最后通过网卡将数据发送出去。ARP缓存表包含一个生存周期（TTL），它将记录每个ARP表项的生存时间，生存时间到了就会从缓存表中删除。

**CRC检测编码**

循环冗余检测（Cyclic Redundancy）也被称为多项式编码（Polynomial code），因为该编码能够将要发送的比特串看作系数是0或1的多项式。CRC是计算机网络中广泛应用的差错检测技术，是一种特殊的检验和算法（与位置相关）。CRC计算的具体流程如下：

* 发送方和接收方协商r+1的比特模式
* 接收方生成多项式G，G的最高有效比特位为1
* 发送方对于给定数据段D，发送方要选择r个附加比特R得到d+r的比特模式
* 接收方用G除以（模2算术，二进制除法）接收到的d+r比特。如果余数为非零，接收方知道出了差错，否则认为数据正确而被接收

**交换机的转发和过滤**

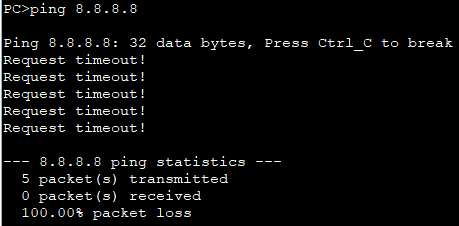
交换机的转发是决定将包发向什么接口，交换机的过滤是判断一个帧是应该转发到某个接口还是应该丢弃。交换机的转发和过滤借助于交换机表，交换机表包括局域网上某些主机和路由器的MAC地址，帧到达的接口以及当前时间。交换机是自学习的设备，如果有帧通过接口，那么交换机就会存储表项。

具体流程

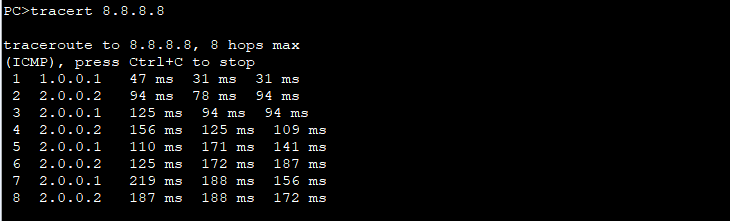
根据大作业的拓扑，具体流程如下

1. PC1用户在浏览器中输入http://www.test.com/1.html
2. 操作系统生成DNS查询报文，将http://www.test.com/1.html放入DNS报文的问题段中。DNS报文被传递给传输层封装入一个具有目的端口为53号的UDP报文段中。UDP报文段被传递给网络层封装入一个目的地IP地址为6.0.0.2，源IP地址为1.0.0.2的IP数据报。IP数据报被传输给数据链路层封装入以太网帧。
3. PC1检查自己的ARP缓存表，发现没有目的IP地址对应的MAC地址，利用ARP协议生成一个具有目的IP地址（网关）为1.0.0.1的ARP查询报文，将该ARP报文放置在一个具有广播目的地址（FF:FF:FF:FF:FF:FF）的以太网帧中，并向交换机发送该帧，交换机将该帧交付给所有连接的设备，交换机学习到PC1的MAC地址
4. 网关接收到包含ARP查询报文的帧，通过路由表将帧转交给R2
5. R2接收到帧，转交给ARP查询报文中的目的IP地址的主机也就是DNS Server。DNS Server存储关于PC1的ARP缓存，创建一个ARP应答报文包含自己的MAC地址，目的IP地址为1.0.0.2进行转发
6. PC1经过交换机的转发得到ARP应答，抽取并存储网关的MAC地址与IP地址的映射，此时PC1得到。交换机学习到网关的MAC地址
7. PC1发送帧到网关，网关路由器接收该帧并抽取包含DNS查询的IP数据报，根据目的IP地址和路由转发表转发给R2，R2接收该帧并抽取包含DNS查询的IP数据报，根据目的IP地址转发给DNS server
8. DNS Server接收帧，从中抽取出DNS查询报文，查询自己的DNS记录，准备包含http://www.test.com/1.html对应的IP地址5.0.0.2的DNS应答报文，封装并经过路由器，交换机转发给PC1
9. PC1生成TCP套接字，与5.0.0.2的主机进行TCP三次握手（具体见TCP原理），PC1生成包含要获取的URL的HTTP GET报文，逐层封装进行转发
10. Web Server接收到帧，逐层解封读取HTTP GET报文，生成一个HTTP相应报文，将请求页面内容放入HTTP报文中，逐层封装进行转发
11. PC1接收到帧，逐层解封，得到HTML页面

问题六



当Ping 8.8.8.8时会造成路由环路，在ping命令的结果中表现为超时丢包，在tracert命令的结果中可以看到路由环路的过程



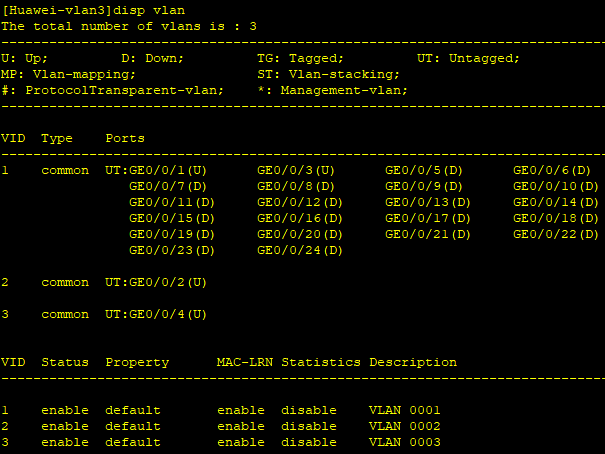
产生的原因

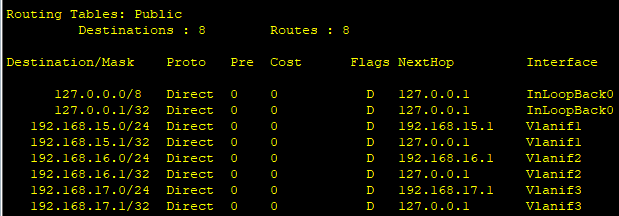
8.8.8.8是一个不存在于拓扑中的IP地址，而在R1配置了指向R2缺省路由且R1路由表中没有关于8.0.0.0网段路由的情况下，请求报文从R1转发给R2，R2没有关于8.0.0.0网段的路由，因此R2将依据指向R1的缺省路由将报文转发给R1。R1又再次把报文传给了R2，如此重复，形成了路由环路。

问题七

将连接到同一交换机的主机划分子网需要使用到VLAN，具体实现步骤如下

1. 更改三个主机的IP地址，分别为192.168.15.2，192.168.16.2，192.168.17.2，网关分别为192.168.15.1，192.168.16.2，192.168.17.2，子网掩码均为255.255.255.0
2. 添加VLAN2，3，为VLAN接口设置IP地址
3. 更改三个接口类型为access，并加入相应的VLAN
4. 显示VLAN设置与IP路由表





从表中可以看到VLAN划分成功

1. 测试连通性

