

****

信息学院软件工程系

《计算机网络》实验报告

**题　　目 实验五**CISCO IOS 路由器基本配置

**班　　级 软件工程2018级2班**

**姓　　名 李成洋**

**学　　号 24320182203220**

**实验时间 2020年4月8日**

**2020 年 4 月 8 日**

# 实验目的

使用 Router eSIM v1.1 模拟器来模拟路由器的配置环境；使用 CCNA Network Visualizer 6.0 配置静态路由、动态路由和交换机端口的 VLAN（虚拟局域网）。

# 实验环境

Windows10

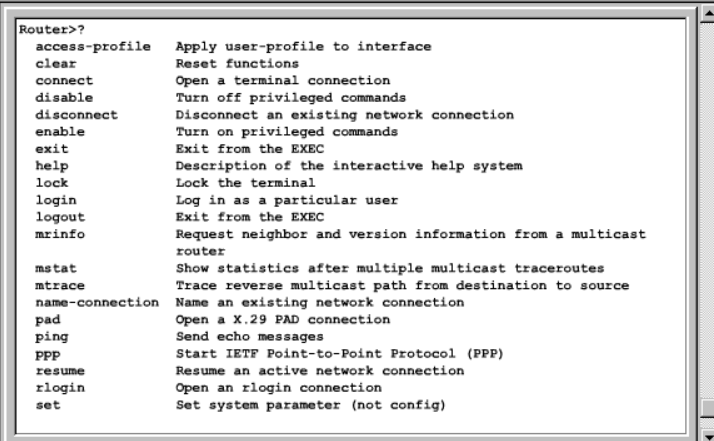
工具：Router\_eSIM\_v11

CCNA Network Visualizer 6.0

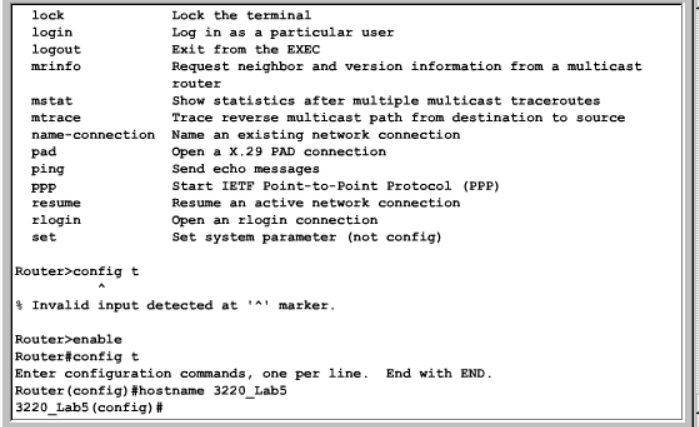
# 实验结果

1. 使用 Router eSIM v1.1 模拟器来模拟路由器的配置环境

首先查看相关指令：

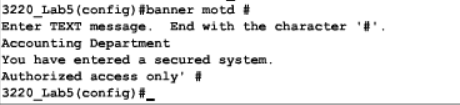


进入全局配置模式和超级用户模式更改用户名：

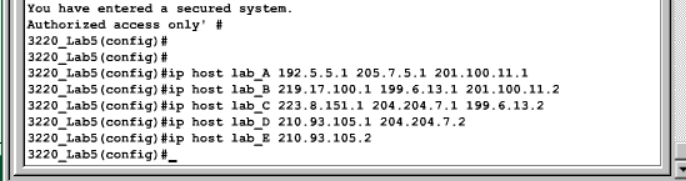


开始对路由器进行一些常规配置：

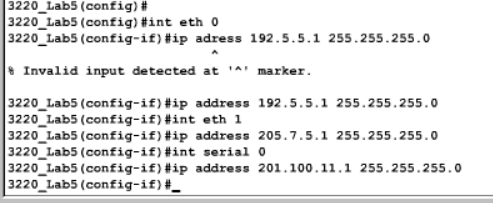
设置当日消息标题：



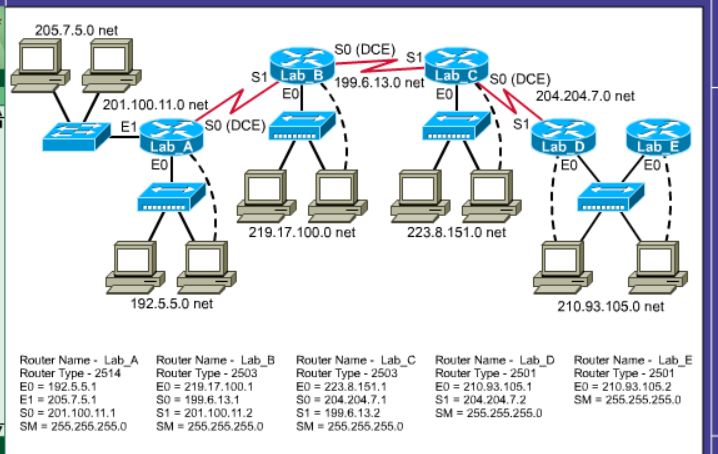
在路由器内建立一个IP地址的映射表：



对路由器的各个接口进行配置：



为S0 设置Clock Rate：

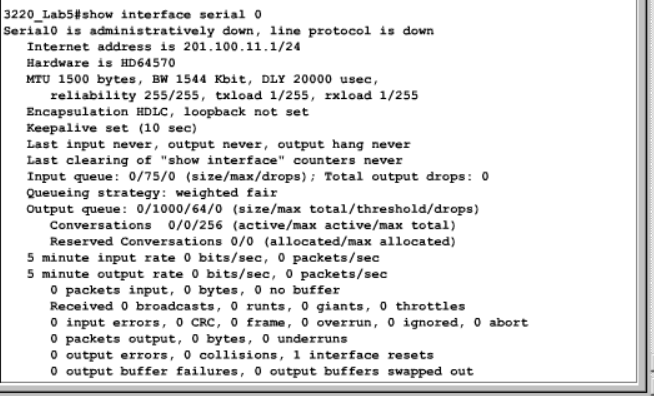


由拓扑图可知，S0端口是DCE的

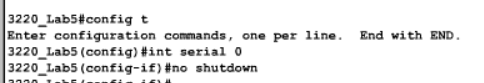
设置Clock Rate：



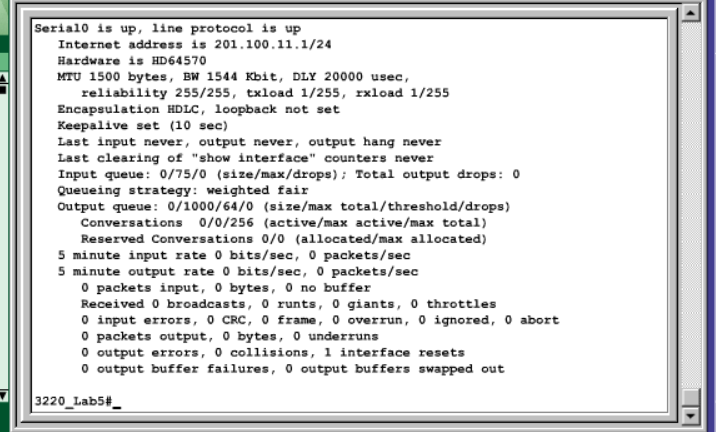
用show命令查看串口的配置情况：



上图第一行管理员手动关闭了该端口，现在把该端口打开：

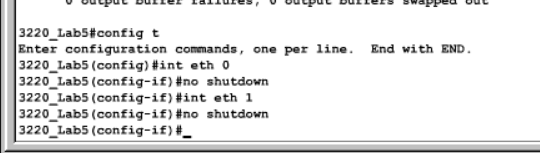


现在再次查看S0，看是否有打开：



可以看见现在接口的工作情况正常了

将E0，E1端口也打开：

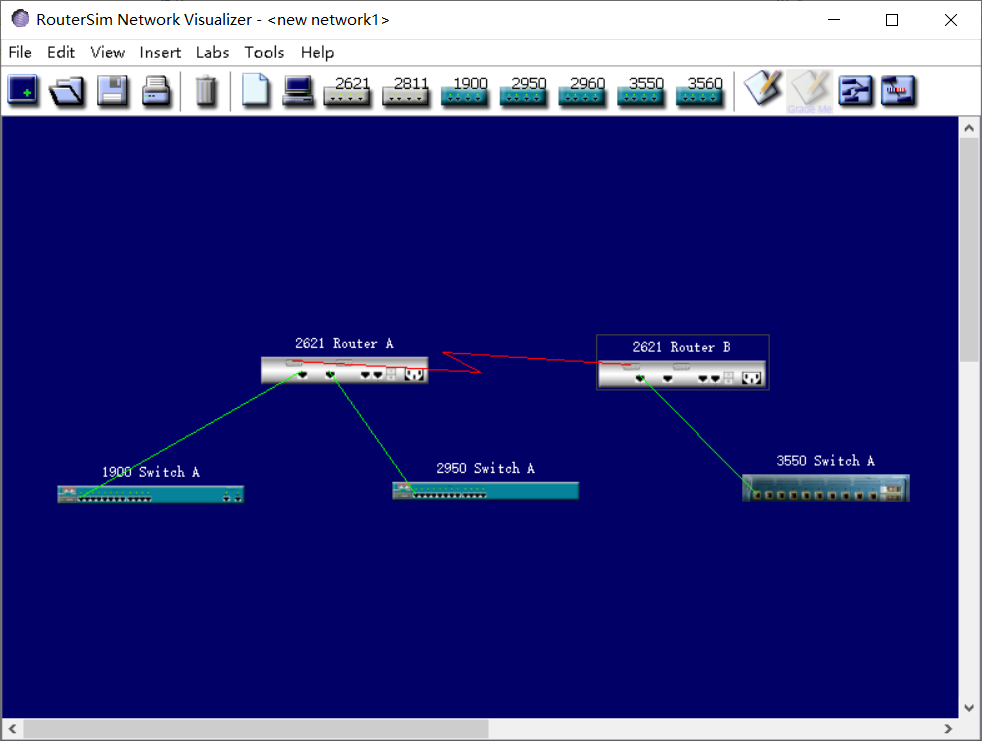


至此，用Router eSIM v1.1 模拟器来模拟路由器的配置环境的实验任务完成

1. 使用 CCNA Network Visualizer 6.0 配置静态路由、动态路由和交换机端口的 VLAN（虚拟局域网）

配置静态路由：

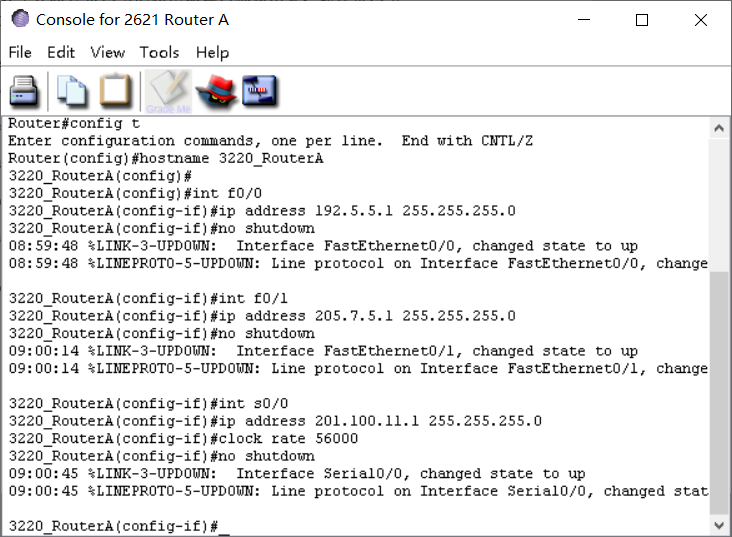
首先将实验设备在模拟器的设计界面上按拓扑图链接完成：



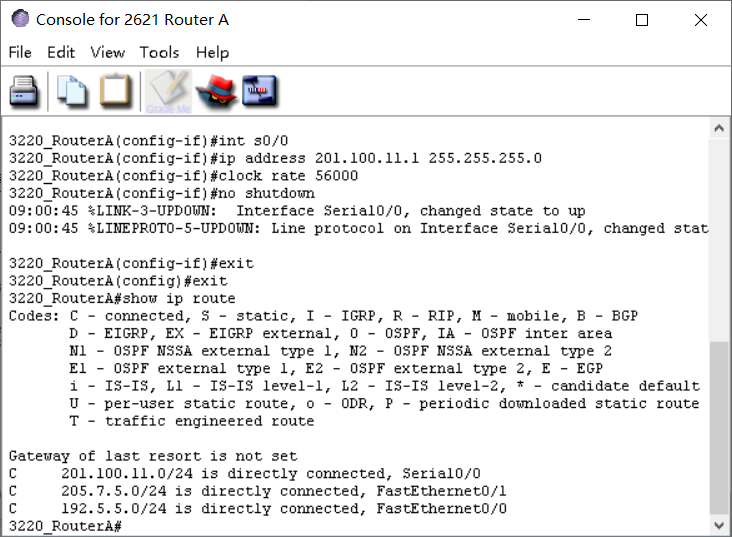
做静态路由配置之前的工作：

配置路由器各个端口的IP地址，并激活端口：

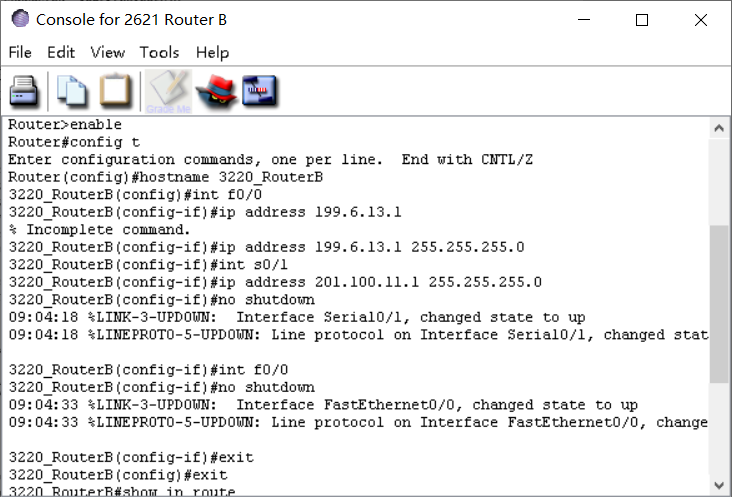
先将routerA配置好：



查看配置好的路由表：

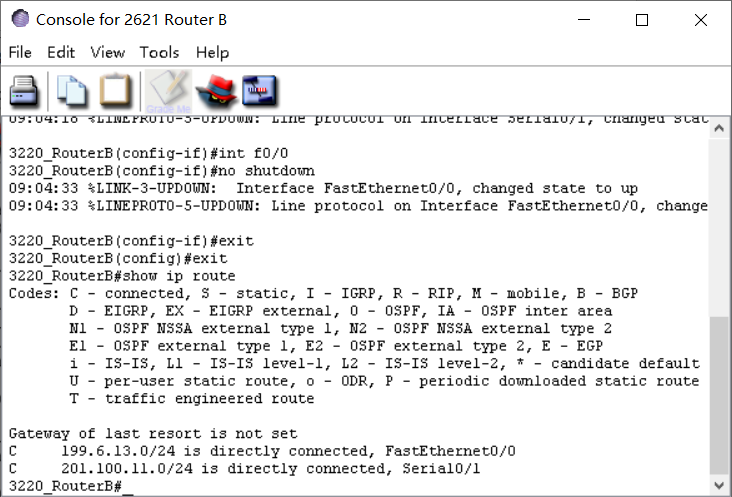


再用同样的方法将B配置好：

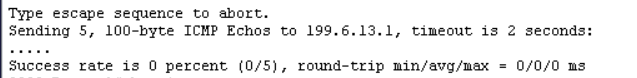


上图S0的ip地址设置错误，应该为201.100.11.2，已在后续操作中更改

查看B的路由表：

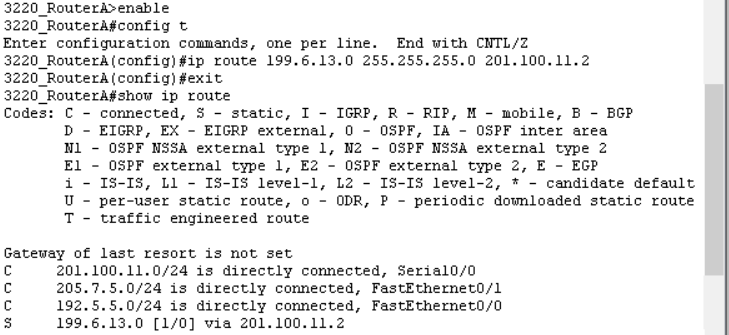


在Router A上查看与B是否连通：

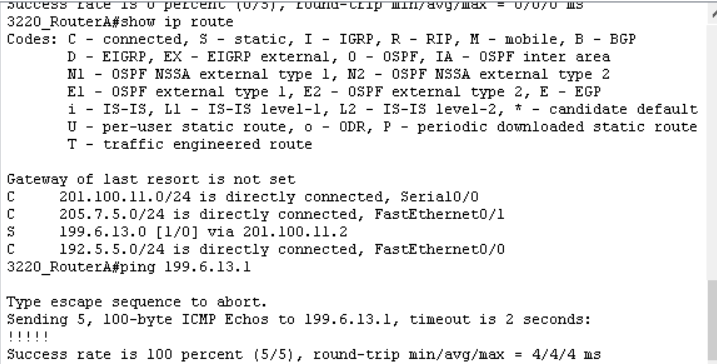


Ping不同，则需要配置静态路由

开始配置静态路由：

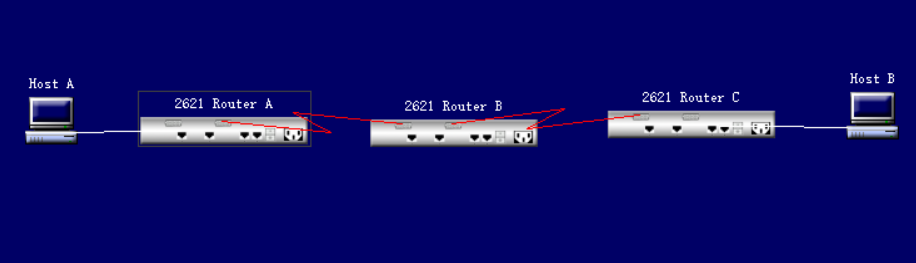


配置好静态路由之后发现ping仍然不同，经过检查指导B的s0/1端口的ip地址配置错误，应该配置为201.100.11.2 更改好配置后，ping通：



配置动态路由：

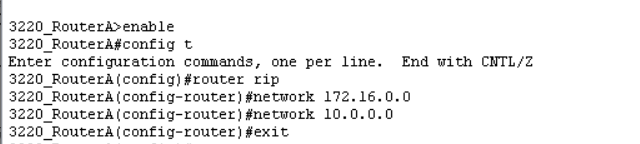
首先建立如下图所示的拓扑图：



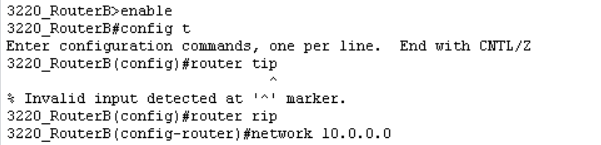
对A B C三台路由器设置的各个端口设置IP地址，具体操作与静态路由的配置相同，故这里不再截图

设置好各个端口后，开始进行RIP配置：

对A进行配置：



对B进行配置：

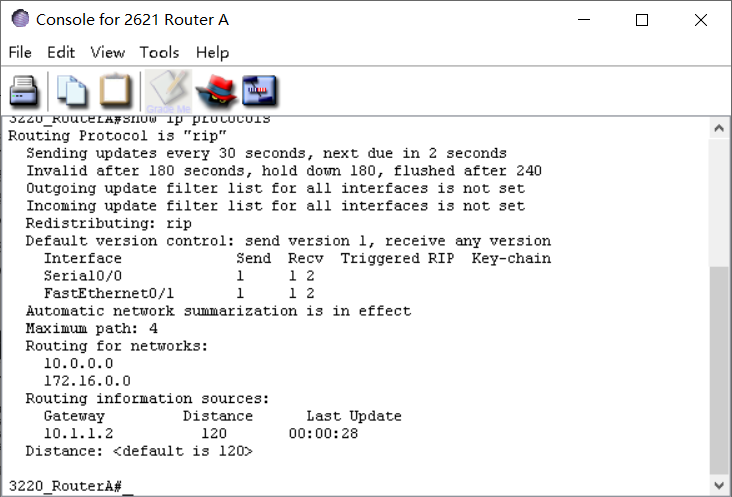


对C进行配置

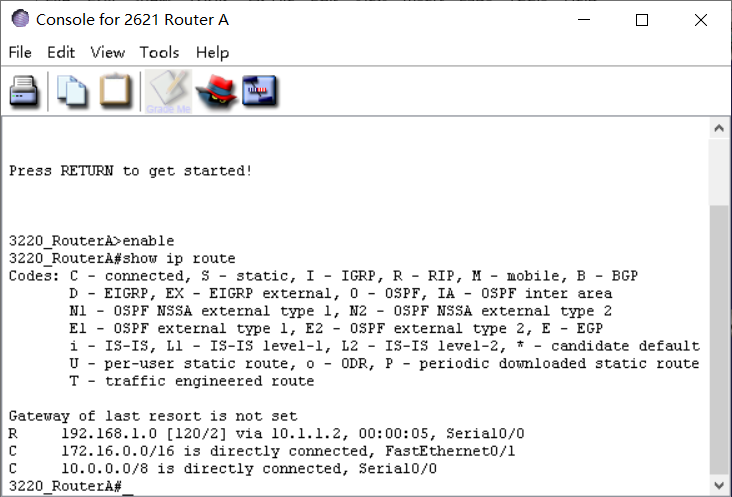


设置完成后：

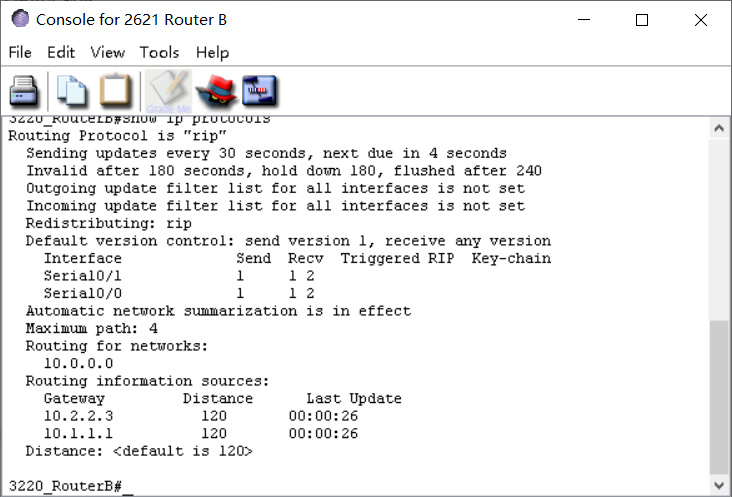
查看A的路由协议RIP的工作情况以及路由表：



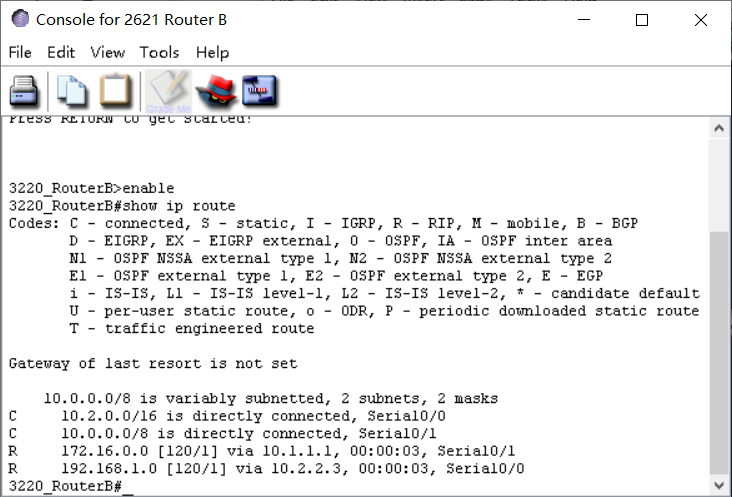
路由表：



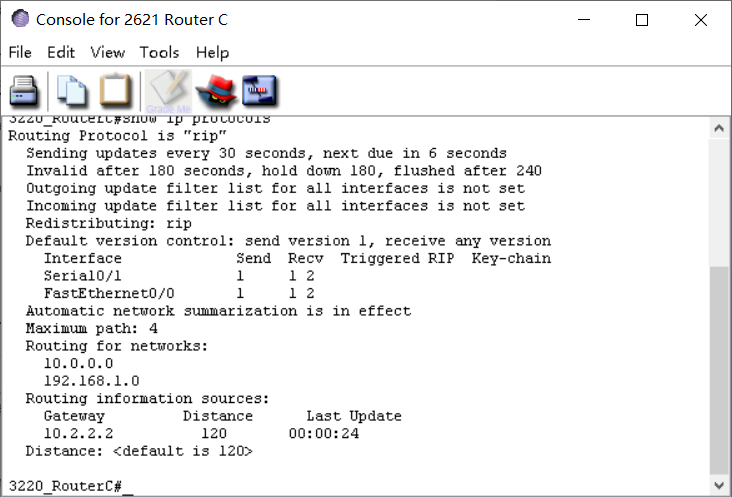
查看B：



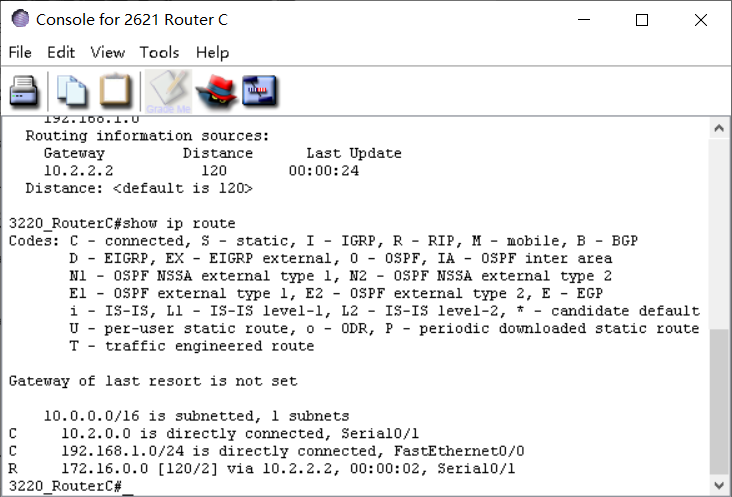
路由表：



查看C：



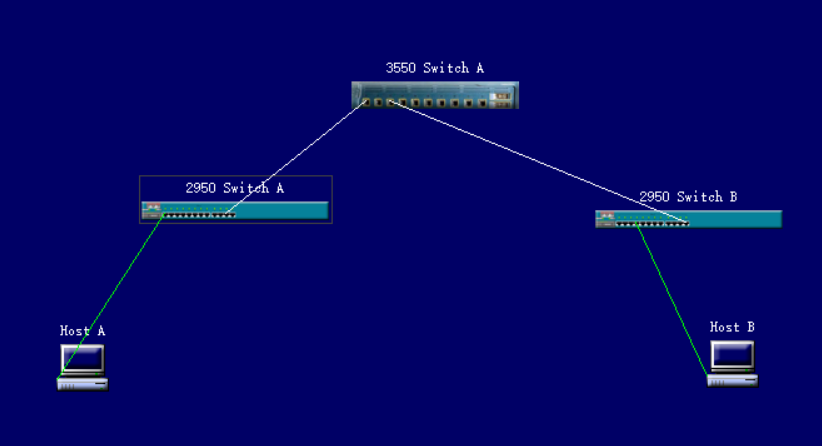
路由表：



配置换机端口的 VLAN（虚拟局域网）

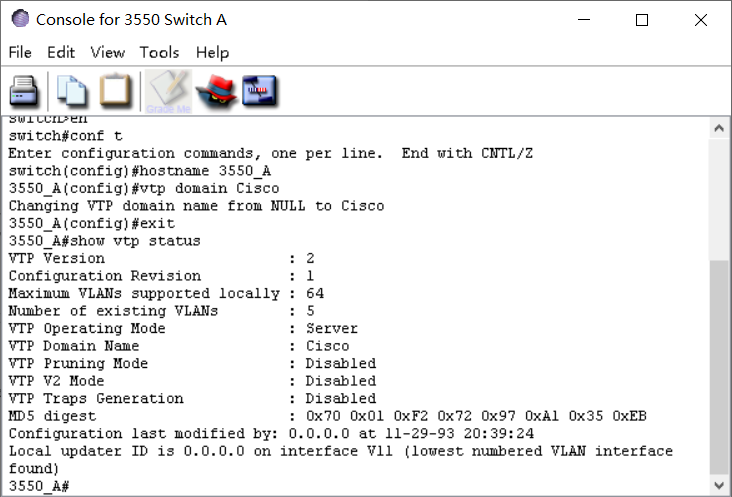
在一个典型的快速以太局域网中实现VLAN：

先构建好如下图所示的拓扑图：

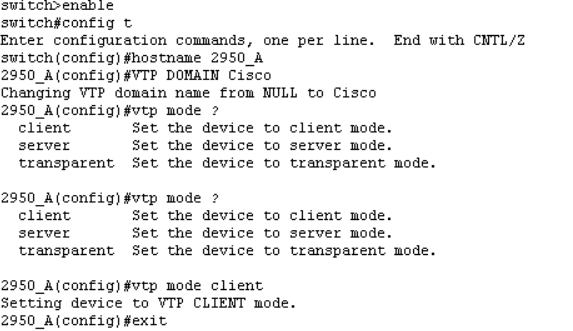


首先，设置VTP域：

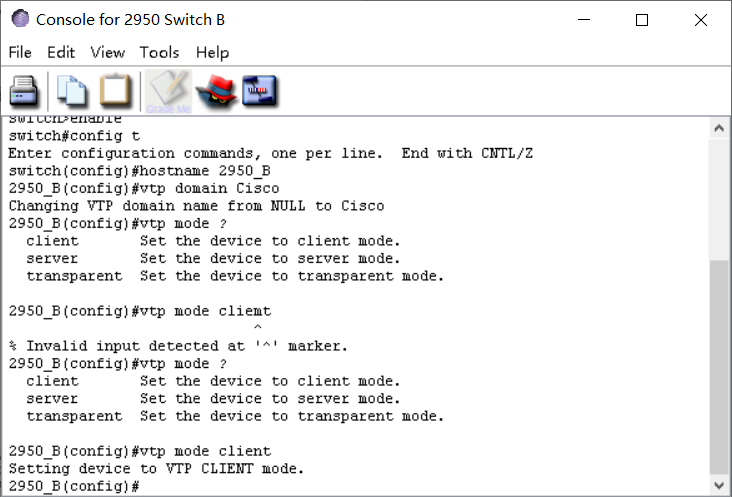
对3550交换机进行设置（默认为服务模式，所以不用设置vtp mode）：



设置2950 A（设置为客户模式）：

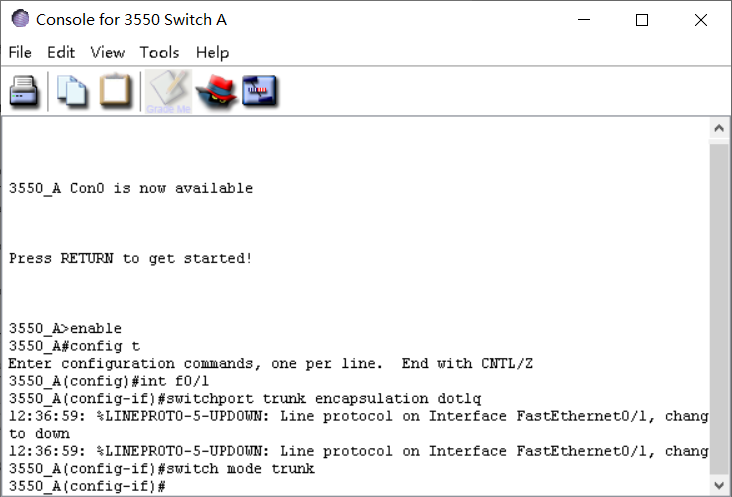


设置2950 B（设置为客户模式）：



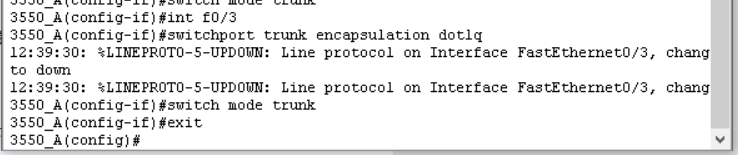
配置Trunk：

将服务端交换机的F0/1和F0/3端口配置为Trunk端口，并用802.1q封装（选择dot1p）：

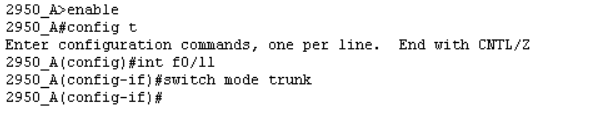


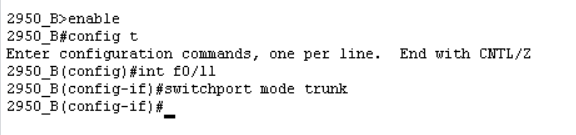
（这里犯了一个错误，之前几次输入dot1q时一直将字母1输入成了字母l，一直找不到无效的原因，直到在网上搜索了才知道错在哪里）

配置f0/3：



将两台客户交换机的端口f0/11设置为trunk端口：

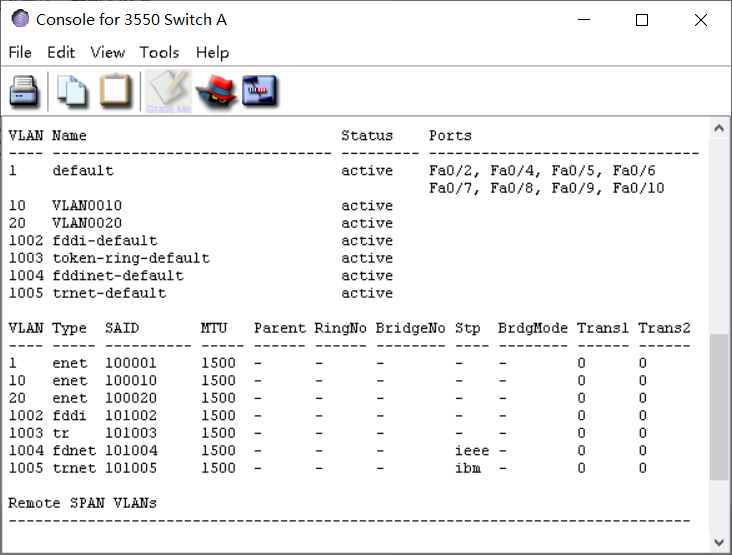




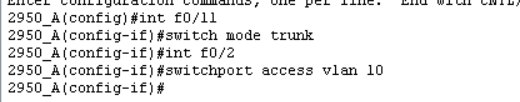
下一步就开始创建VLAN（2个，一个为VLAN 10一个为VLAN 20）：

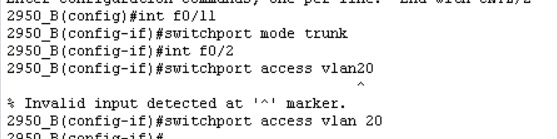


创建好了之后进行验证：



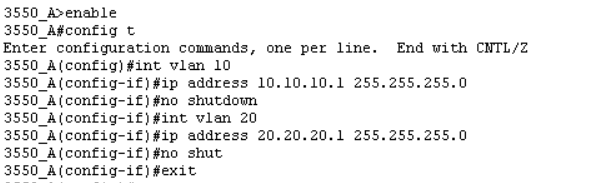
让两台客户机的端口F0/2分别加入VLAN 10和VLAN 20：





配置第三层交换机：

首先在服务交换机上分别设置各VLAN的接口IP地址：

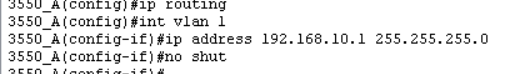


然后启用路由：

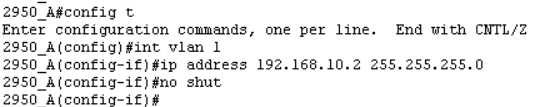


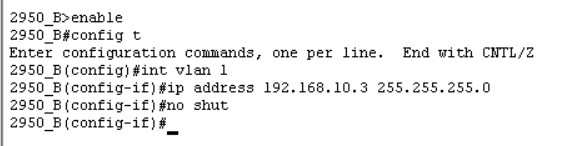
启用路由后，配置各交换机的管理地址：

配置服务机：

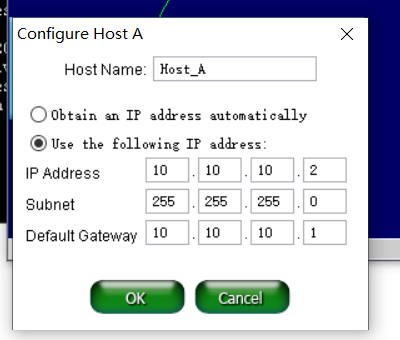


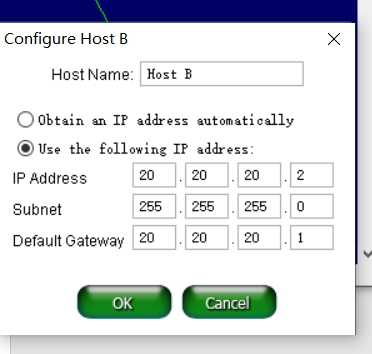
配置两客户机：



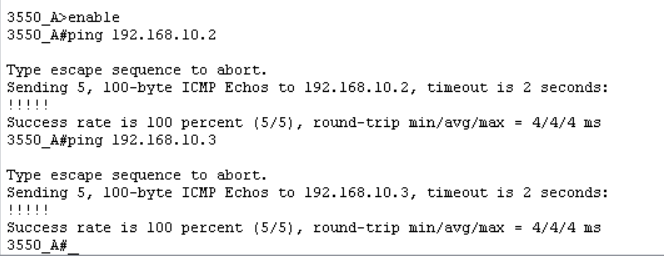


最后配置两台主机Host A和Host B，并进行检测：



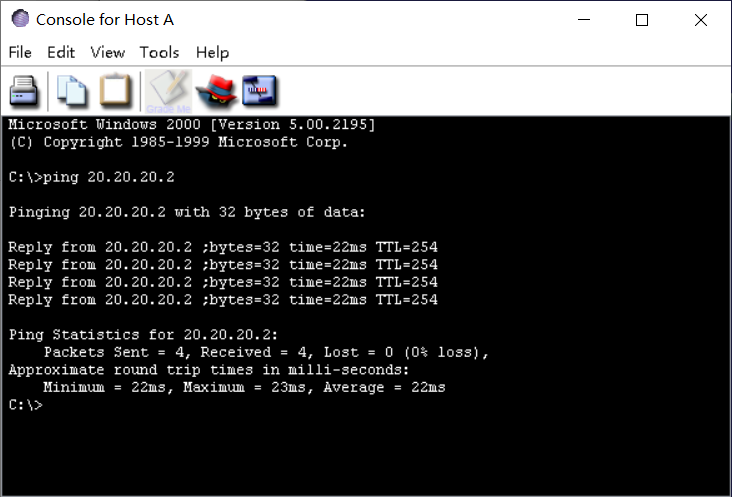


在服务机上分别ping两台客户机：



由上图可知已经连通

主机A ping主机B：



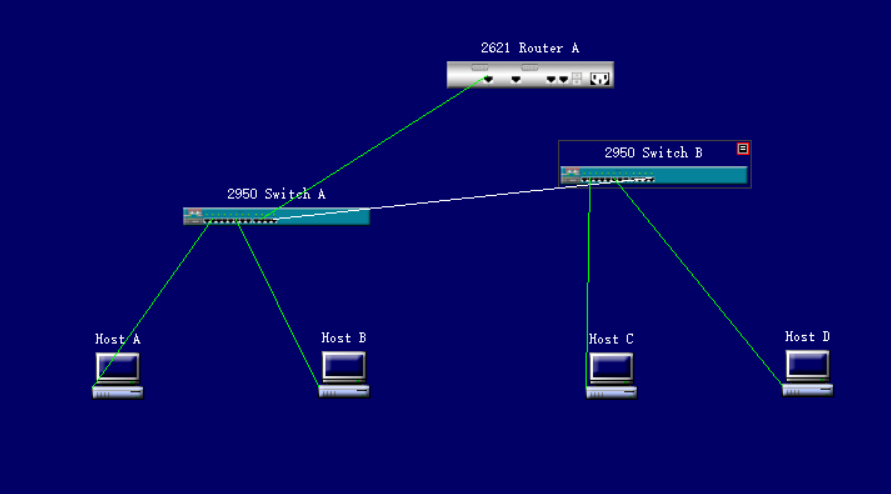
Lost为0，表示成功

配置VLAN实例二：

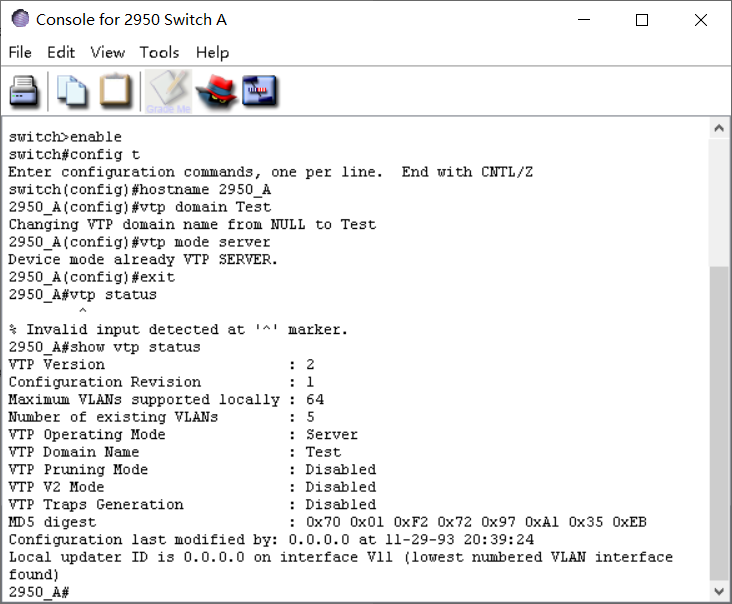
实现VLAN跨越多个交换机及不同VLAN之间的通信：

实例二与实例一有很多相似之处，描述会相对简略一点，只配上必要的截图

首先构建拓扑图：

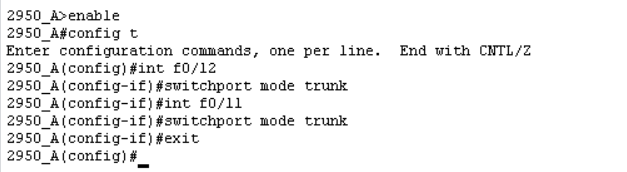


配置VTP：

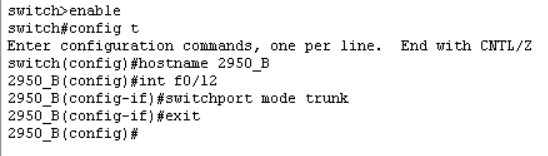


启动Trunk：

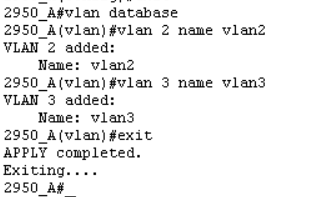
设置A：



设置B：

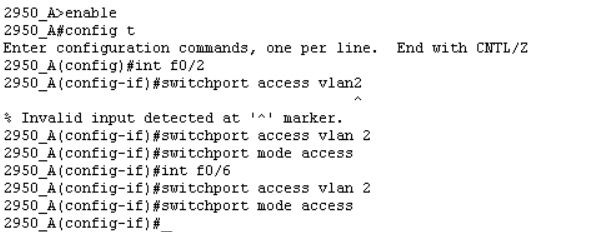


创建VLAN：

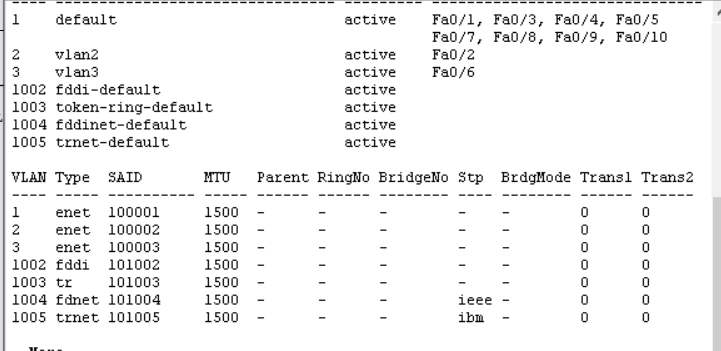


分配端口到VLAN：

分配A的端口：

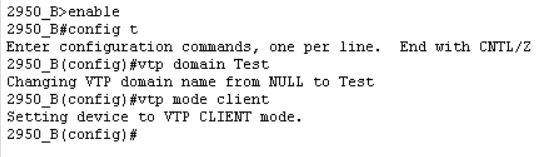


使用show vlan验证：

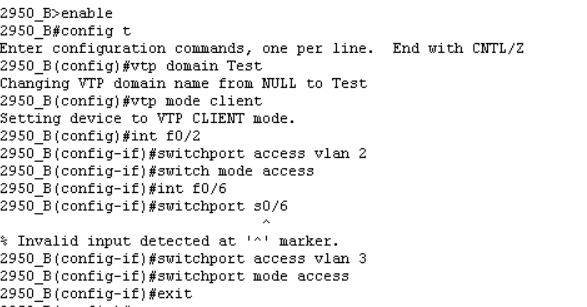


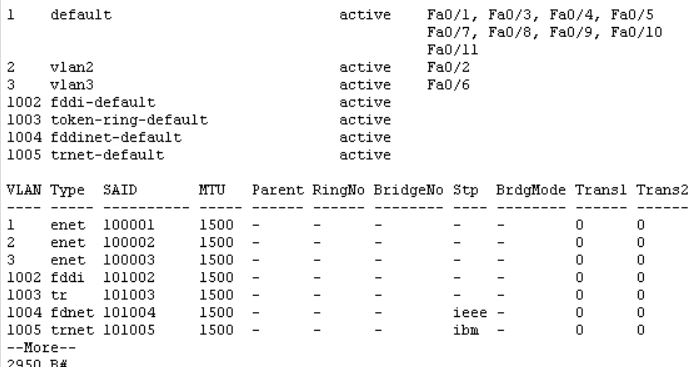
配置交换机B：

设置为客户模式：



分配端口并验证：

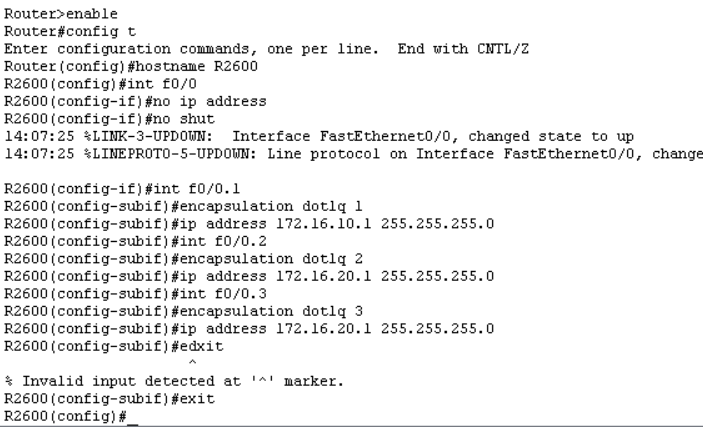




配置VLAN之间的路由：

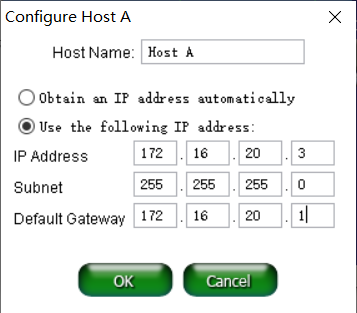
这里与实例一不同，这里是由路由器来实现VLAN之间的通信

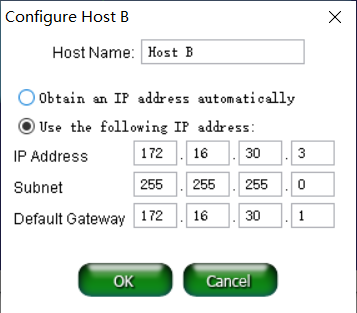
所以对该路由器进行配置：

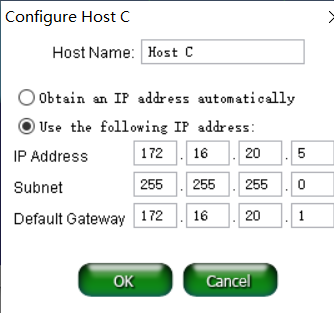


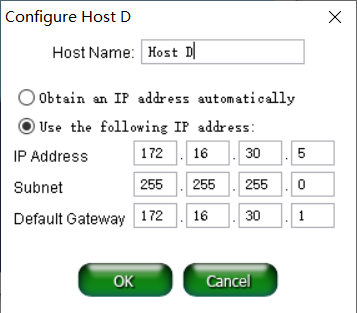
这里0.3接口的ip地址写成了172.16.20.1 导致后面ping VLAN 3时一直不成功，后面已经更改为172.16.30.1

配置四台主机：



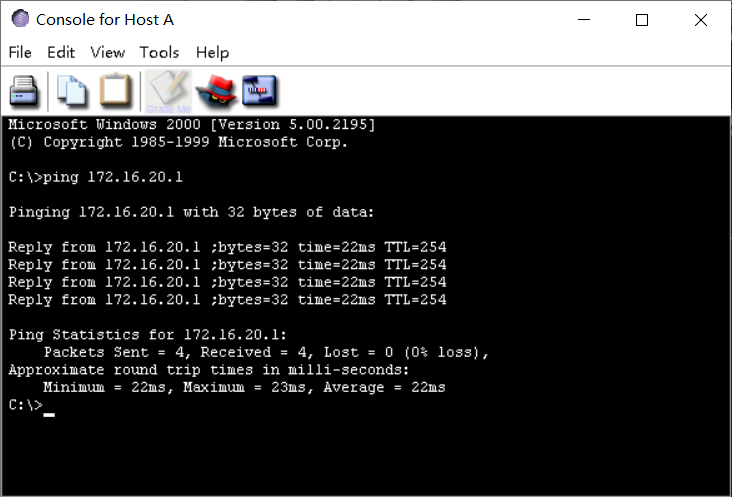




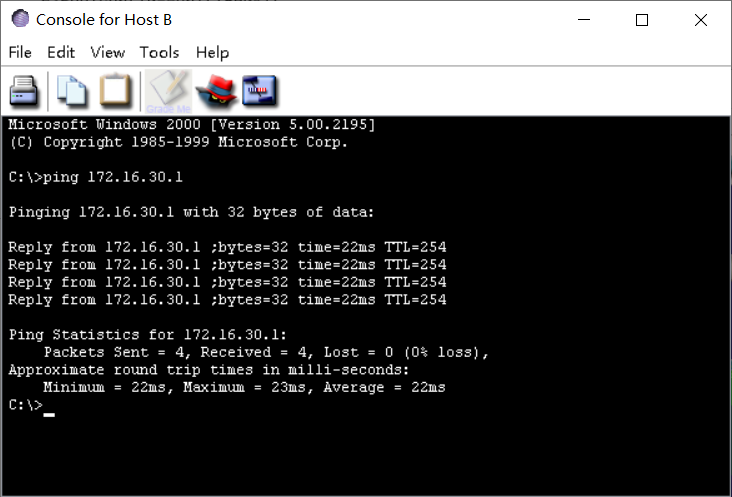


检验连通性：

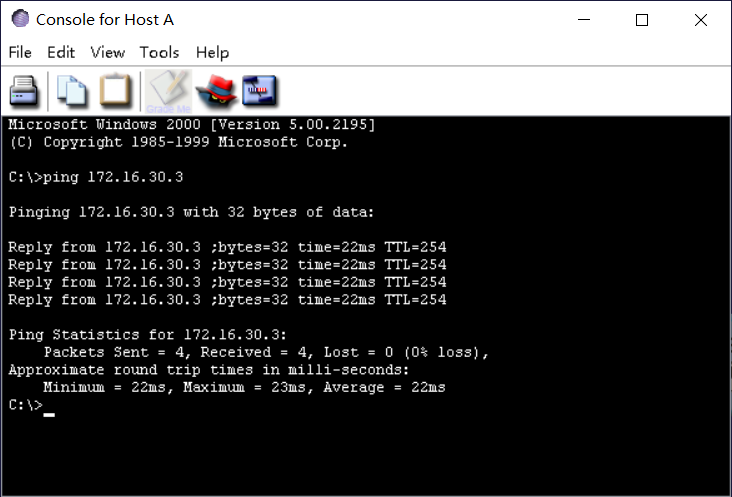
在属于VLAN 2的Host上ping172.16.20.1



在属于VALN 3的Host C上ping172.16.30.1：



在A上PING B：



上面三个ping的丢失都为零（lost=0），表明连接成功

# 实验总结

本次实验理所当然的收获是知道了路由器的配置方法：进入到各个端口，设置其对应的IP地址，如果为DCE还应该设置它的clock rate

在配置路由器的过程中，需要对端口和拓扑图有一定的理解，所以经过这次实验，对拓扑图更加熟悉了，之前学习的时候，也只是知道有拓扑图这样一个东西，但是具体实际使用，并不是很明确。同时，经过本次实验，我知道路由器有两种端口，一种只能用于连接路由器，而另一种则能连接主机和交换机，也是之前学习的时候寄出不好，导致刚开始连接拓扑图的时候怎么都连接不上，闹了笑话

在配置静态路由和动态路由的过程中，对路由表有了更深的理解，之前学习之后，让我画路由表我感觉很迷茫，若是现在，对目的IP和下一站已经拓扑图有了更深的理解后，再来画路由表应该更加简单。不过虽说是跟着实验手册把整个配置流程走了一遍，但有些地方还是没有自己的理解，如果后期不去回忆加固，还是会遗忘很多，所以还需要后期的学习

配置VLAN时，有了之前配置静态路由和动态路由的基础，感觉没有一开始看实验手册那么迷茫了，整个流程除了一些细节上的错误，大致上还行

总的来说，经过本次实验，对路由器的配置有了一个大致的了解，但对于其中的原理，例如VLAN的配置，理解都还是在很浅的层次，后面还需要不断学习理解。还有就是因为在配置过程中用到了ping来检测是否连通，所以对ping的理解加深了，在之前的课程中虽然学习了ping的原理，但说实话对它具体的作用并不清楚，这次实验之后，至少知道了它可以用来检测是否连通成功，以及如果不连通，可以通过它来检测是哪里出了问题。