计算机网络实验

实验报告

**1951725 邓若涛**

# **进程运行原理实验**

**进程运行原理实验**

**实验目的**

计算机网络交互主体是计算机进程,了解进程运行的基本原理,对于理解端与进程关系十分重要。本实验利用操作系统的进程管理软件来展示进程的生命周期。

（1）了解进程的基本概念。

（2）了解进程的基本运行原理,掌握基本进程管理技能。

**实验原理**

计算机程序是指用计算机语言开发的代码文件,一般需要编译成可执行代码。程序以文件方式存储在操作系统的文件系统中。进程指程序装载到内存中进行运行的代码段。

现代操作系统可以对进程实施管理。

（1）进程创建。程序一旦运行就成为进程,每个进程都分配一个唯一的进程编号,但其进程编号并不固定。

（2）进程运行。现代操作系统允许同时运行多个进程,每个进程运行时需要使用多种计算机部件资源,其中CPU资源必不可少,当CPU核数量小于并发进程数量时,操作系统会对CPU实施共享,将CDII“名出问.M八RA人no比八给仅个进积提供运行,使所有进程都能同时运行,进程的并发运行并不是直正意义上的并发。

（3）进程终止。一船通过程良想什的山地知然止要思.和可以通过进程管理命令或软件结束进程。

**实验环境**

实验由一台安装 Windows 10企业版操作系统的计算机担当,使用其任务管理器软件实施进程管理实验,也可以使用其他操作系统作为操作平台。

**实验内容**

实验将围绕进程的管理内容展开,使用任务管理器软件进行进程管理,通过鼠标进行操作,从屏幕观看操作结果。

（1）多次运行同一个特定应用程序,看到该程序可以以多个不同的进程同时运行,每次运行,进程号会不一样。

（2）通过应用程序界面关闭程序,看到其进程被结束。（3）通过任务管理器关闭进程,终止应用程序的运行。

实验步骤

（1）打开任务管理器。同时按“Ctrl"+“Alt”+"Delete”→“任务管理器”→“详细信息”，

显示进程运行状态信息。

右边第一列是程序名，第二列PID列出的就是当前所有已运行程序的进程号，没有看到命令行窗口程序（cmd）进程。

1. 产生两个命令行窗口新进程。
2. 打开两个命令行窗口。连续两次执行命令行窗口程序。
3. 查看命令行窗口程序的进程信息。切换到任务管理器窗口。

可以看到任务管理器窗口中新增了两行命令行程序“cmd.exe”进程信息，其进程号分别为5600和7084

1. 通过应用程序界面关闭进程。
2. 关闭一个命令行窗口程序

点击窗口关闭标签。

1. 查看命令行窗口程序的进程消息。切换到任务管理器。

可以看到进程号为5600的命令行窗口进程消失。

1. 通过任务管理器关闭进程。

右击7084号进程->”结束进程”，余下命令行窗口将被关闭。

**实验分析**

本实验主要了解了进程这一概念，学会了如何在windows操作系统上查看、关闭对应进程，属于计算机网络实验中的基础，同时也能在现实生活中加以运用，需要掌握。

# **网络端地址实验**

**网络端地址实验**

**实验目标**

网络端地址用于标识计算机网络进程,网络进程是计算机网络传输主体,由于语言表达上同题,容易误将计算机作为计算机网络的传输主体。明确了网络传输主体,就容易理解计算机网络各项具体功能处理的基本原理。本实验利用浏览器上网这个最为熟悉的应用,呈现网络端地址作用。

(1) 明确计算机网络交互的主体是进程。计算机网络两台计算机之间的交互,实质上是两个进程之间的交互。

(2) 了解网络端地址构成及使用。端地址是用于标识网络上任意一台计算机上的任意一个网络进程,具有唯一性和不变性,只有通过访问网络端地址才能通过网络同该进程交互。但在日常使用应用协议时，常常忽略端口地址,自动采用该应用协议缺省端口地址作为网络端地址。

**实验原理**

计算机网络正如其名字所揭示的一样,就是用于计算机之间的交互,但只是宽泛的表述现代计算机中真正执行主体实际上是各个进程。所谓两台计算机交互,按照计算机网络通信交互模型,必然是某两个进程之间的交互。网络进程的识别是通过网络端地址来实施。

端地址格式如下:

IP:PORT

用于标识互联网任意一个节点上的任意一个网络进程,IP 指主机 IP 地址,也可以使用:名,域名会被域名系统自动转换成 IP 地址;PORT 指该进程的网络端口号,进程运行时会被册并建立与进程号的映射关系,通过端口号,可以访问到对应进程,进行网络交互。

**实验环境**

实验环境由一台计算机来担当实验设备,计算机必须连接互联网。使用浏览器访问互联网任意一个网站, 其目标网站的端地址为“www.×××.com: 80”，“×××”代表任意域名。

**实验内容**

任何开放的互联网 Web 网站按照业界习惯,都使用 80 端口地址作为 HTTP 协议的端地址。浏览器地址栏支持端地址使用,当没有输入端 J 时,就会自动采用应用协议的缺省端地址,HTTP 协议的缺省端地址是 80。实验使用 Chrome 浏览器访问互联网上任意一个网站,通过设置网络端地址,来验证其作用。

主要实验内容:

（1）端地址设置成 81 端口时,浏览器无法浏览网页。

（2）端地址设置成 80 端口时,浏览器可以正常浏览网页。

**实验步骤**

启动浏览器,通过访问同一个网址的不同端口,实验中使用了同济大学官网,实际可使用任何一个网址。

（1）访问非80端口。地址栏中输入“http://www.tongji.edu.cn:81”

若浏览器显示无法获得该 URL 地址网页，Web 服务器端 П 地址不是 81。

1. 访问80端口，在地址栏中输入“httt://www.tongji.edu.cn:80”

浏览器能正常显示网站主页内容。

**实验分析**

本次实验中主要进行了对于网络端地址的理解，明确了网络的传输主题。属于计算机网络实验中的基础，但也有其重要性。

# **网络线的制作和测试实验**

**网络线的制作和测试实验**

**实验目标：**

学会利用TJ45水晶头和五类双绞线制作以太网直通线，并能够使用网络电缆测试仪对该以太网直通线功能进行测试

**实验材料：**

两个TJ45水晶头、一段五类双绞线、压线钳、网络电缆测试仪

**实验步骤：**

1. 用压线钳将网线一端的套管皮剪掉2cm
2. 按照白橙、橙、白绿、蓝、白蓝、绿、白棕、棕线序把网线排列好
3. 把网线摆平拉直，剪齐留下1.5cm
4. 将水晶头有塑料弹簧片的一面向下，有金属针脚的一面向上，将线插入水晶头，并使其紧紧地顶在顶端
5. 把水晶头插入压线钳套住水晶头用力压，使得网线和水晶头卡在一起
6. 同样的方法制作另一个插头
7. 使用以太网测试工具，将网线线两端头插入网线测试仪，灯亮表示测试通过

**实验分析：**

本实验目的在于能够利用水晶头、网线以及相关工具制作以太网直通线，并且能够对所制作的以太网直通线利用网线测试仪对其功能进行测试。

在实验过程中，主要注意该五类双绞线具有8条网线，且直通线两端的网线排列顺序需要一致，本实验中采用的是白橙、橙、白绿、蓝、白蓝、绿、白棕、棕线顺序，但是除了该种接法，也还有568A种接法，既白绿、绿、白橙、蓝、白蓝、橙、白、棕。同时所有网线尽量保持整齐以确保能够与水晶头之间的接触良好，在网线插入水晶头之后需要用压线钳将水晶头压紧，确保网线和水晶头卡在一起。

在使用网线测试仪对以太网直通线进行测试的时候，需要观察网线测试仪上的灯闪烁情况，主测试器和远程测试端上都具有1，2，3，4，5，6，7，8，G这9个小灯，其中1至8小灯表示8根网线的连接情况，若主测试器和远程测试端上8根网线的连接情况，若远程测试端和主测试器的闪烁情况保持一致，并且以1，2，3，4，5，6，7，8的顺序闪烁，则表示该以太网直通线能够正常工作，若网线少于2根线联通时，灯都不亮，当网线有2根短路时，则主测试器显示不变，而远程测试端显示短路的两根线灯都微亮，若有3根以上（含3根）短路时，则所有短路的几条线号的灯都不亮。

本实验的主要难点在于制作以太网直通线部分，需要保证8根网线尽量持平，且两端网线顺序必须一致，但是在实验过程中，对于网线测试仪的所有情况并没有了解清除，所以可以进一步进行相关的测试

# **UDP协议网络编程实验**

**UDP协议网络编程实验**

**实验目的：**

1.熟悉OSI七层协议体系结构

2.了解各层完成的工作

3.掌握协议的工作过程及特点

**实验原理：**

UDP协议特点：

（1）UDP无需建立连接。因此UDP不会引入建立连接的时延。

（2）无连接状态。UDP不维护连接状态，也不跟踪参数，因此某些专用应用服务器使用UDP时，一般都能支持更多的活动客户机。

（3）分组首部开销更小。UDP只有8个字节的首部开销。

（4）应用层能够更好地控制要发送的数据和发送时间。UDP没有拥塞控制，因此网络中的拥塞也不会影响主机的发送效率。某些实时应用（如直播）要求以稳定的速度发送，能容忍一些数据的丢失，但不允许有较大的时延，而UDP正好可以满足这些应用的需求。

（5）UDP提供尽最大努力的交付，即不保证可靠交付，但并不意味着应用对数据的要求是不可靠的，因此需要维护传输可靠性的工作需要用户在应用层来完成。应用实体可以根据应用需求来灵活设计自己的可靠性机制。

（6）UDP是面向报文的。发送方UDP对应用层交下来的报文，在添加首部后就交付给IP层，既不合并，也不拆分，而是保留这些报文的边界；接受方UDP对IP层交上来的用户数据报，在去除首部后就原封不动的交付给上层的应用进程，一次交付一个完整的报文，因此报文不可分割，是UDP数据处理的最小单位。

（7）UDP常用于一次性传输比较小数据的网络应用，如DNS、SNMP等，因为对于这些应用，若采用TCP，则将为创建连接、维护和拆除而带来不小的开销。UDP也常用于多媒体应用（如IP电话、实时视频会议、流媒体等），显然，可靠数据传输对于这些应用来说并不是最重要的，但TCP的拥塞控制会导致数据出现较大的延迟，这是它们不可容忍的。

UDP报文：

UDP首部有8个字节，由4个字段构成，每个字段都是两个字节

1.源端口： 源端口号，需要对方回信时选用，不需要时全部置0.

2.目的端口：目的端口号，在终点交付报文的时候需要用到。

3.长度：UDP的数据报的长度（包括首部和数据）其最小值为8（只有首部）

4.校验和：检测UDP数据报在传输中是否有错，有错则丢弃。

该字段是可选的，当源主机不想计算校验和，则直接令该字段全为0。

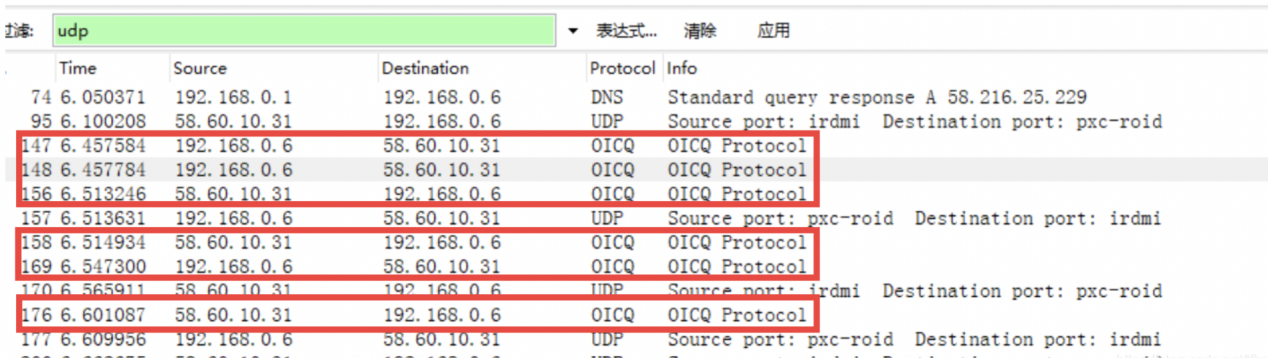
当传输层从IP层收到UDP数据报时，就根据首部中的目的端口，把UDP数据报通过相应的端口，上交给应用进程。

如果接收方UDP发现收到的报文中的目的端口号不正确（不存在对应端口号的应用进程0），就丢弃该报文，并由ICMP发送“端口不可达”差错报文给对方。

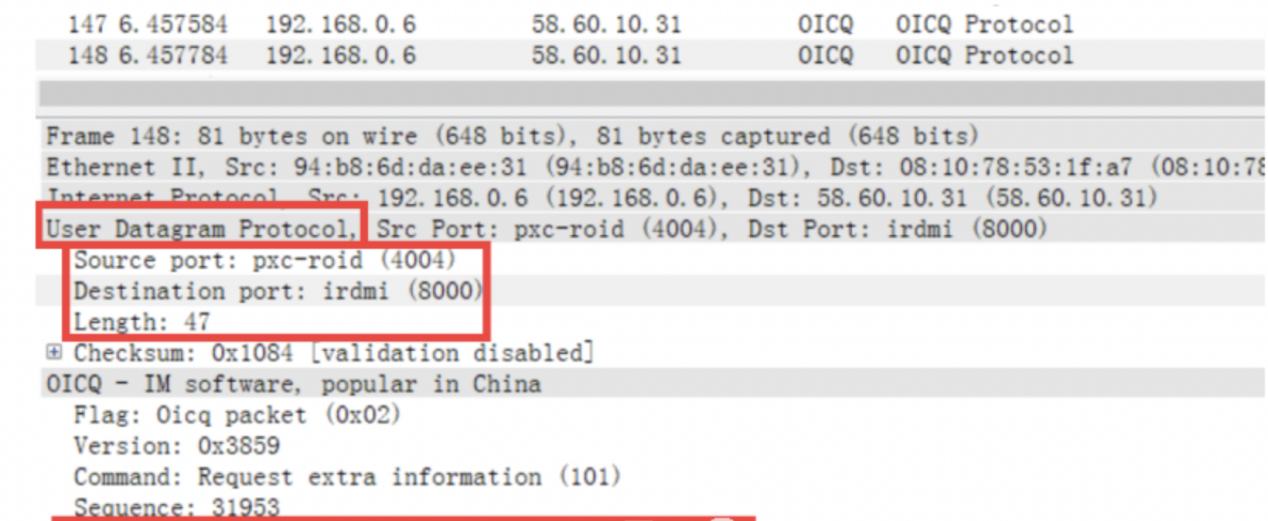
**实验过程：**

1.过滤UDP数据包

启动wireshark抓包软件，输入过滤条件UDP，开始对腾讯QQ进行抓包



2.QICQ数据包分析

****

由图中可以看到传输层协议应用的是UDP协议。其中，源端口为：pxc-roid（4004），目的端口为：irdmi(8000)，传输的数据段的长度为：47byte，校验和为：0x1084。

由图中的应用层协议可以得到各种信息。其中，数据段（QICQ number）打码部分，正符合实验者的QQ帐号。

**实验分析**：

通过UDP抓包实验，体会到UDP协议与TCP协议的不同之处：

1.UDP所花费的头部开销较少，而TCP所花费的头部开销较多；

2.从报文的长度来看，UDP传输的报文长度与TCP不同。这是因为UDP是面向报文的，它不对报文做任何处理，即既不拆分报文也不合并报文，因此UDP传输的报文需要选择合适的报文长度。

在实验过程中，由于网络原因，产生丢包现象。

通过本实验，我更好的理解了UDP协议传输原理、UDP与TCP协议的区别等，收获颇丰。

# **TCP应用协议编程实验**

**TCP应用协议编程实验**

**实验目标：**

本实验是使用Socket来编写一个基于TCP简易通讯程序，且具有并发能力。

1. 了解基于TCP网络应用服务器的基本编程架构。
2. 了解面向连接和无连接的区别，了解TCP编程基本步骤。
3. 了解并发服务原理及编程方式。

**实验原理：**

1. 并发服务

TCP采用面向连接方式，服务器一旦同一个客户机建立了会话，就无法同其他客户再建立会话，严重影响用户体验。并发服务能力是指服务器具有同时建立多个会话，允许多个客户访问的能力，日常使用的TCP服务器均是并发服务器。但并发服务不属于网络技术而属于编程技术研究范畴，并发服务器程序有着特殊的编程架构，典型架构有两部分代码组成，第一部分是主进程，负责请求调度，接受客户请求申请，建立会话并为每个请求创建服务处理线程；第二部分是服务处理线程，负责具体会话处理，为客户请求提供服务，可以被主进程不断创建。

假定有两个客户访问同一个TCP服务器，TCP服务器启动后，只有主进程在运行，开启服务端口PORT，等待客户端访问。第一个客户访问PORT端口，主进程接收到请求1，创建一个服务线程1负责处理请求1，随后的会话发生在客户机1和服务线程1之间，主进程等待下一次请求到来；第二个客户访问PORT端口，主进程接收到请求2，再创建另一个服务线程2负责处理请求，随后的会话发生在客户机2和服务线程2之间，主进程继续等待下一个请求。如果同时又N个客户访问服务器，这时候就有N+1个线程（进程）在运行；主进程永远等待客户访问。并发处理的数量理论上没限制，主要取决于计算机处理能力，设计CPU核数量和内存大小。

2.TCP实验程序编程

TCP服务器编程除了要遵循客户机/服务器模型，还要让服务器具备并发能力，所以雋要创建三个类，客户端仍一个，服务器两个：主进程和服务线程。基本功能同UDP编程实验类似，客户机从发送窗口发送任意一段字符串给服务器，服务器接收来自客户机的字符串，然后将字符串返回给客户机，客户机在另一个窗口显示。

客户机由TCPClient类担当，通过端口号5588访问服务器；服务器主进程由MainServer类担当，服务端口号为5588,主要功能是接受客户机访问并创建服务线程来处理，实验时主进程必须首先启动，才能让客户机能访问服务器；服务线程由ServiceServer担当，承担实际服务处理功能。

Socket接口中，已实现了丁CP协议若干对象，主要有ServerSocket类和Socket类。

1.ServerSocket类。代表TCP协议对象，包含了代表TCP端口的属性，主要方法有accept( )。accept( )用于接受客户会话请求创建Socket会话对象，然后，通过会话对象进行  
数据通信。

2.Socke类。代表TCP 会话对象，主要方法 getInputStream()和 getOutputStream()分别用于应用数据流的接收和发送，进行双向通信。应用数据大小不限，TCP发送前会自行分割，收到后会自行拼接，其功能已封装在接口内。当不再需要通信，close()方法用于关闭会话。

客户机同服务器主迸程交互主要是建立TCP会话，客户机同服务线程交互是通过会话进行数据收发。

⑴MainServer类，作为服务器主进程，负责接受客户会话请求，创建服务线程，涉及三个  
环节。

①创建TCP服务Socket, Java提供了 ServerSocket专门作为TCP服务Socket，已封装  
了传统Socket的多种功能，创建端口并实施监听，核心代码如下：

ServerSocket server = new ServerSocket(5555)；

②等待来自客户端的会话请求，一旦成功就会创建Socket会话对象，该对象代表同客户  
端的TCP会话，通过该对象才能进行数据通信，核心代码如下：

Socket client = server, accept ( ) ;

③创建ServiceServer祕线程，将Socket会话对象传递过去启动线程运行负责同客  
户灌的数据传输及处理，核心代码如下：

ServiceServer service = new ServiceServer(client),

Thread thread = new Thread(service)；  
thread. start()；

⑵ServiceServer程序，作为服务线程负责实际数据处理和服务，需要实现Runnable线  
程接口，核心处理代码包含在run()方法中，由MainServer启动线程运行，涉及三个环节。  
①从会话对象Socket获得数据流，发送是输出流，接收是输人流，核心代码在构造方  
法中：

public ServiceServerCSocket c)throws IOException{  
this. client = c；

is = new ObjectInputStream(client. getlnputStream())；  
os=new ObjectOutputStreamCclient. getOutputStream())；

}

②利用输人输出流同客户端进行数据交互，核心代码片段如下：

//发送

Object obj = is. readObject() ;

BufferedReader in = new BufferedReaderCnew InputStreamReaderCclient.

getlnputStream() )) ;  
//接收

out = new PrintWriterCclient. getOutputStream());

③交互结束,关闭会话对象，核心代码片段如下：

in. close();  
out. close();  
client. close();

⑶客户端TcpClient程序，作为客户端程序使用面向连接方式进行通信，包含三个步骤

1. 请求同TCP服务器建立会话，指定服务器地址和服务端口，该对象代表同服务器的网络连接，通过该会话对象可以进行数据传输，核心代码如下：

//1.Open a Socket

serviceSocket = new Socket(this.txtFieldIP.getText(),5555);

1. 从会话对象Socket获得数据流，发送是输出流，接收是输入流，核心代码在构造方法中：

out = new PrintWriter(serviceSocket.getOutputStream());

in = new BufferedReader(new InputStreamReader(serviceSocket.getInpuStream());

1. 交互结束，关闭会话对象，核心代码片段如下：

serviceSocket.close();

(4)测试控制代码。

测试控制代码包含在TePciient程序中，主要嵌人到main()中，目的是创建两个测试客户

以检验并发访问，核心代码片段如下：  
public static void main(String[ ] args) {

// TODO Auto-generated method stub  
TcpClient clientl = new TcpClient(wJinw) j  
clientl. pack()；  
clientl. setSize(400, 360)；  
clientl. setVisible(true)；

TcpClient client2 = new TcpClient("zhang");

client2. pack()；  
client2. setSize(400, 360)；  
client2. setVisible(true)；

**实验内容：**

在Java项目Socket下，开发服务器程序和客户机程序

1. 开发服务器程序，创建并实现MainServer类和ServiceServer类，将附录3的MainServer类和ServiceServer类全部源代码复制进去。
2. 开发客户机程序TCPClient类，为方便实验，TCPClient包含图形化界面，代码较长，将课本附录3的TCPClient全部源代码复制进去。
3. 测试交互。先运行MainServer,后运行TCPClient，将连续创建两个TCPClient进程，并同时访问服务器，然后再TCPClient界面中输入字符串，测试并发通信服务。

zazaiz**实验步骤：**

运行Eclipse开发平台，并选择已创建Socket项目。

1. 创建java包”socket”，用于存放实验代码

右击项目Socket/src->”New”->”Package”。

输入包名”Name=socket”->”Finish”。

1. 开发数据类User。User代表用户对象，是辅助类。
2. 创建 User类。在包”Socket”下创建User类，右击”src/Socket”->”New”->”Class”。Name=User，取消”public static void main(String[] args)”->”Finish”。
3. 开发。输入源代码，用课本附录3中User源代码完全覆盖初始代码。
4. 保存及编译。键人Ctrl+S或菜单“File”—“Save”。
5. 开发客户端TcpClient类。

1.创建 TcpClient 类。在包“Socket”下，创建 TcpChent 类，右击“src/Socket’’ —“New —  
“Class”—Name=TcpClient—取消“public static void main(Sring[ ] args)” —“Finish”。

2.输入实验代码。用附录3中TcpClient源代码完全覆盖初始代码。

3.保存。保存包含了编译，键人Ctrl+S或“File”—“Save”。

1. 开发基于线程的服务处理类ServiceServer类。
2. 创建ServiceServer类，在包”Socket”下，创建service Server类，右击”src/Socket” —”New” —”Class” —输入类名;Name=ServiceServer —取消”public static void main(String[] args)” —”Finish”。
3. 输入实验代码，用课本附录3中ServiceServer源代码完全覆盖初始代码。
4. 保存。保存包含编译，键入”Ctrl+S”或”File” —”Save”。
5. 开发服务调度类MainServer类。
6. 创建MainServer类，右击”Socket/src/Socket” —”New” —”Class” —Name=MainServer,选择”public static void main(String[] args)” —”Finish”。
7. 输入源代码，用课本附录3中MainServer源代码完全覆盖初始代码。
8. 保存。保存包含编译，键入”Ctrl+S”或”File” —”Save”。
9. 运行实验。
10. 运行MainServer。
11. 运行客户端TCPClient，发送文本，用鼠标拖动zhang窗口，出现两个窗口。

点击“连接”，在下面MainServer控制台上，会显示用户连接信息。在发送区域，分别输入文本并发送。

TCPClient上的接受曲线是，在数据前面加上用户标识”Jin”和”zhang”；左下角是MainServer控制台，显示处理请求过程，实验成功。

**实验分析：**

本实验主要目的是了解Socket编程，学会利用Socket编程实现一个基于TCP协议的简易通信装置，并且使用编程的并发性解决TCP协议中服务器一旦同一个客户机建立了会话，就无法同其他客户再建立会话的问题。

在实验过程中需要对Eclipse编译器进行环境搭建，本次实验使用的是java 1.8版本的jdk，否则会有部分package无法编译，所以在进行实验前需要构建好Eclipse环境。在进行实验过程中时，需要先运行MainServer类进行服务器搭建，然后才能运行TCPClient类，否则用户无法与服务器产生通信，且运行TCPClient类之后用户需要在界面点击”连接“按钮与服务器进行连接，若用户在连接之前尝试与服务器断开连接，则系统会显示连接出错。

本实验算是对于Socket编程的一次基础了解，以及对并发性、TCP协议等基础概念做了一定的了解，了解如何通过程序的编写来完善网络通信的功能，属于基础实验类型之一。

# **异步串联通信收发实验**

**异步串联通信收发实验**

**实验目标：**

数据通信是物理层的核心功能，其基本原理属于通信学范畴。实验利用计算机的串行口进行两台计算机间的字符收发，展示通信基本原理，加深了解物理层作用。

1. 理解异步串行通信基本原理。
2. 熟悉掌握RS-232通信标准以及RS-232帧格式。
3. 了解波特率等主要通信参数的作用和使用。

**实验原理：**

1. **串行传输模式**

串行传输模式是指任何通信用户单独只能占用一条通信信道，对二进制数据按照顺序依次发送，每次只能传输一个数位，0或1。串行传输模式被广泛应用在远程通信中。

1. **ASCII编码**

计算机内部所有的数据都使用二进制数字。当需要标识非数字信息时，就需要用数字组成码来标识。编码标准是指规定各个字符使用什么样的二进制数字来表示。

ASCII，美国信息交换标准编码，是基于拉丁字母的一套电脑编码标准，主要用于显示现代英语和其他西欧语言，使用一个字来表示英语字母。

1. **RS-232接口**

RS-232接口是1970年由美国电子工业协会联合贝尔实验室，调制解调器厂家及计算机终端生产厂家共同制定的用于异步串行传输的标准。主要内容就是定义数据终端设备DTE和数据通信设备DCE之间的接口标准。RS-232传输对象是字符，每次传输一个字符，可持续传输多个字符。RS-232异步串行传输模式，其物理信号采用电压作为传输信号，电压处于±15V之间，其信号状态标准：

1. 空闲状态信号，用负电压来表示。
2. 数位0，用正点压表示。
3. 数位1，用负电压表示。
4. **波特率**

波特率也称调制速率，指单位时间内通信信号变化的次数。波特率决定着每个数位的传输速率，显然，设计揭发两端的两个设备配置，为保证两者在发送信号和检测信号时的时间同步，就必须设置成相同的波特率。发送设备和接收设备采用相同波特率成为波特率匹配。如果接发双方波特率不匹配，就会发生传输错误，称为帧错误。

**实验环境：**

实验环境主要由两台带串口的计算机组成，一根串行交叉线用于连接两个计算记得串口。使用超级终端工具软件作为异步串行通信的实验操作平台。

**实验内容：**

进行两台计算机之间的异步串行通信，一端从键盘输入字符发送，另一端观看接收到的内容。使用超级终端工具软件作为实验操作平台。

1. 异步串行通信实验。相同通信参数下的通信，进行相互间的字符接发送。
2. 波特率通信实验。不同波特率条件下的通信，其他参数保持相同，进行相互间的字符接发送。

zazaiz**实验步骤：**

按照实验拓扑结构要求，将两台计算机的COM口用串口反接连接线连接起来，实验拓扑结构如下图所示：



**实验一：异步串行通信实验**

1. 建立异步串行链接。

①Host1运行超级终端创建连接。输入连接名:”testCOMA”->”确定”

②设置通信参数。接受缺省通信参数。点击”确定“。

③Host2建立串行传输连接，操作完全同Host1一样。新建超级终端连接testCOMB，接受缺省通信参数。

1. Host2至Host1字符发送实验。在两端建立超级终端连接后，进行字符收发，由Host2发送，Host1接收。

①Host2发送字符。

从键盘键入”Hello From Host2”，请注意该工具软件并不会屏幕上回显输入内容。

②Host1接收字符。Host1超级终端会自动接收，并显示接收到的内容。可以看到接收到的字符串”Hello From Host2”。

（3）Host1至Host2字符发送实验。由Host1发送，Host2接收。

①Host1发送字符。

从键盘键入”Hello From Host1”，请注意该工具软件并不会屏幕上回显输入内容。

②Host2接收字符。Host2超级终端会自动接收，并显示接收到的内容。可以看到接收到的字符串”Hello From Host1”；结果完全正确。

**实验二：波特率通信实验**

重新设置Host1通信参数，保留Host2原有通信参数，使用不同的波特率。

（1）重新设置Host1通信参数。将Host1通信参数设置为：波特率（4800）、校验位（无）、数据位（8）、停止位（1）。

①断开连接。点击断开连接图标，中断通信连接。

②修改通信参数。

“文件”->“属性”。点击”配置“。修改参数，端口设置：波特率（4800）、校验位（无）、数据位（8）、停止位（1）->”确定“->”确定“

③重新建立通信连接。点击链接图标重新建立连接。

1. 通信测试。

①Host2发送字符。从键盘键入”12 35 46 77“，屏幕上并不会显示输入内容。

②Host1接收字符。Host1超级终端将显示接收到的内容。屏幕上显示乱码。当通信双方采用不同波特率，将导致通信错误，显示乱码。

**实验分析：**

本实验主要在于掌握异步串行通信的基本原理，采用的是RS-232通信标准，并且了解超级终端软件的操作基础。在此标准下，两个用户之间需要建立通信需要保证两台客户机之间用串行交叉线进行连接，但是通信双方需要保证采用的波特率一样，否则双方数据之间的传输会出现错误。

本次实验目前只考虑了波特率作为通信参数的影响和作用，但是波特率不是在进行异步串行传输通信时影响通信的唯一标准，在接下来的实验中会研究校验方式等因素对于异步串行通信的影响。

在使用超级终端软件进行异步串行通信的时候，需要注意的是在客户机上不会出现本客户机发送的信息，只能接收到另一台客户机所发送的信息，而且在键入信息的时候，超级终端软件只支持以行为单位输入、撤销字符，而无法对已经输入的上一行字符进行撤销操作。

# **以太网组网实验**

**以太网组网实验**

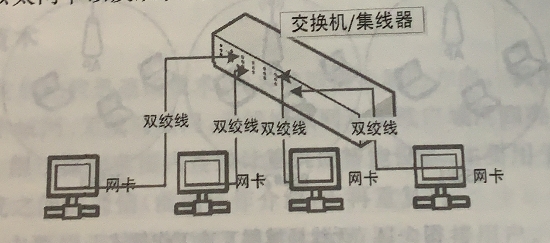
**实验目标：**

物理网络是计算机网络的基本组织单元，其各个节点之间可以进行数据通信。物理网络是互联网的基础架构，无论对于理解网络基本原理还是网际互联原理都非常关键。实验利用以太网交换机组成一个独立的双绞线以太网物理网络，实现网络节点之间的互通。

1. 理解局域网组网原理
2. 理解掌握以太网组网步骤
3. 了解以太网网络地址格式

**实验原理：**

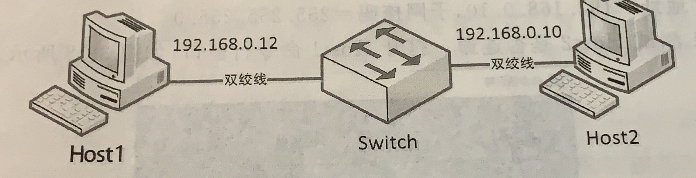
以太网是当今有线局域网主体，目前家庭和企业安装的有线局域网全部采用以太网，即使使用无线网络连接主机，无线网络最终还是要借助以太网连接互联网才能上网。以太网主要由以太网网络设备、以太网卡以及双绞线组成。双绞线以太网组网如下图所示：



加入以太网的主机，必须拥有一块以太网卡，现在的计算机一般在主板上都已集成了以太网网卡。集线器和交换机都是基于双绞线的以太网网络设备，一般会提供多个以太网端口，使用双绞线网线可以将主机网卡同交换机端口连接起来，构成一个独立的物理网络，加入网络的节点可以相互通信。

**实验环境：**

以太网组网实验拓扑结构如下图所示，反映了典型的以太网组网方式：



两台计算机和一台交换机担当实验设备，使用两根双绞线网络，将两台计算机以太网网卡同交换机连接起来。计算机Host1作为操作平台和测试操作平台，另一台计算机Host2作为测试平台。也可以使用家用无线路由器代替交换机，开展本实验。

**实验内容：**

以太网组网实验

1. 将两台计算机和交换机连接成一个局域网。需要用双绞线网线将网卡和交换机端口连接起来。现有操作系统都使用TCP/IP协议为网络协议，不直接支持物理网络通信，需要通过TCP/IP协议使用网卡，必须为网卡配置IP地址，为处理方便，所有子网掩码设置为255.255.255.0，计算机Host1网卡IP地址设置为192.168.0.12；计算机Host2网卡IP地址设置为192.168.0.10。
2. 利用ping命令对两台计算机间的连通进行测试。

以太网网卡地址查看实验。使用ipconfig命令查看网卡物理地址，了解以太网地址格式。

**实验步骤：**

**实验一：以太网组网实验**

1. 用两根双绞线网线分别将两台计算机网卡同交换机端口连接起来，这样就形成了局域网。
2. 为主机配置IP地址，主机网卡IP地址设置如下：

Host1：IP地址=192.168.0.12，子网掩码=255.255.255.0

Host2：IP地址=192.168.0.10，子网掩码=255.255.255.0

1. Host1测试Host2是否连通。打开Host1命令行窗口，输入ping命令“ping 192.168.0.10”，可以看到联通，由于交换机处理有个延缓过程，命令要多打几次，才能实验成功。

**实验二：以太网网卡地址查看实验**

Host1查看自身以太网物理地址，打开命令行窗口，输入命令“ipconfig：ipconfig/all“，可以看到Host1以太网卡的IP地址就是192.168.0.12，物理地址是40-2C-F4-ED-74-74。

**实验分析：**

本实验同样是实现两台客户机之间的互通实验，但是实现使用的方法是利用双绞线将两台客户机连接到一台交换机，形成独立的局域网络，从而使在该局域网中的两台计算机实现数据互通。该实验成功的基础在于当今计算机大部分在主板上已经集成了以太网卡，因此可以直接通过双绞线将以太网卡与交换机连接从而构成一个局域网。

在实验过程中我们可以发现当今大多数计算机采用的是TCP/IP协议，在本实验中通过更改其IPv4地址进行试验，其IP地址由四个3位二进制数组成，在本实验中需要保证两台客户机的IP地址不同，且以太网中不能存在其他网络的IP地址与对方的IP地址相同，否则会影响实验结果。

实验中存在一些注意事项，在更改计算机的IP地址后立刻使用ping命令进行连通测试可能会失败，因为交换机处理存在延缓过程，可能无法立刻进行互通，所以需要多尝试几次ping命令直至成功，若仍无法互通则是实验步骤出现问题。与此同时在实验过程中存在未能解决的疑问，如实验中出现了Host1能够接收到Host2的信息，但是Host2在接收Host1发送的信息时却会存在超时的现象，这个问题并未得到解决，希望在今后能够尝试发现问题。

# **VLAN配置实验**

**VLAN配置实验**

**实验目标：**

对于企业而言，可能含有许多部门，为便于管理，常常以部门为单位，构建多个物理子网。传统网络工程，只有相近的办公室才可以组成同一个物理子网，鉴于种种原因，很可能同个部门的两个办公室位于不同楼层，甚至不同大楼。

虚拟局域网，标准编号为IEEE802.1Q，可以实现将两个相距较远的办公室组成同一个物理子网。实验利用交换机提供虚拟局域网功能，实现VLAN划分。

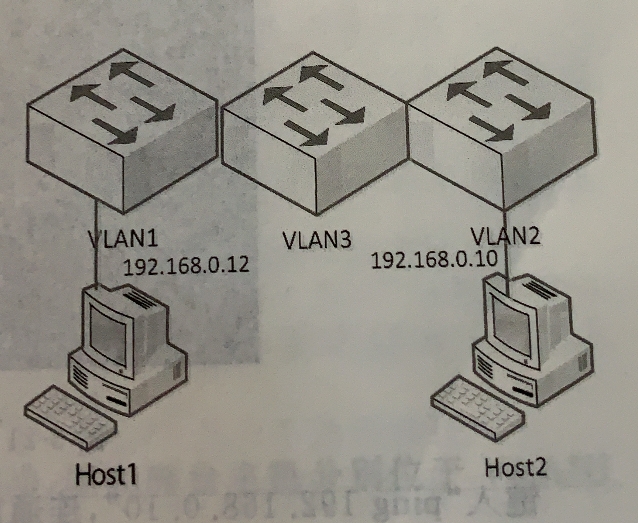
1. 了解虚拟局域网基本概念。
2. 掌握交换机实施虚拟局域网技术。

**实验原理：**

虚拟局域网可以把处于不同地理位置的主机组成同个局域网，以太网交换机是网桥的一种产品形态，原本用于扩展物理网络覆盖区域，利用交换机内部软件，可以将交换机网络上的端口进行逻辑分组，一个逻辑端口组构成虚拟网络成为一个独立的物理网络。一个VLAN就是一个广播域，不同VLAN之间的通信是隔离的，必须通过第3层的路由功能才能进行通信。VLAN设置比较灵活，具有多种划分方法。

**实验环境：**

交换机VLAN配置实验拓扑结构如图3-20所示：



由一台交换机和两台计算机担当实验设备，是用两根双绞线网线，将两台计算机以太网网卡同交换机连接起来；通过串行线将计算机Host1串口com1同交换机Console口连接起来，使用超级终端作为交换机的操作平台。

**实验内容：**

1. **实验概况**

使用交换机作为实验设备，将交换机划分成三个VLAN，VLAN1包含1~8端口，VLAN2包含9~16端口，VLAN3包含其余端口；两台计算机作为测试平台，设置成同一个子网IP地址。

1. 使用Host1超级终端为交换机配置VLAN
2. 为Host1和Host2以太网卡配置同子网IP地址，分别是192.168.0.12和192.168.0.10。
3. 然后用双绞线将两个网卡连接到交换机端口上，不断变换交换机的端口，测试联通状况。位于同一个VLAN时，两个主机将会连通；位于不同VLAN时，两个主机无法连通。
4. **VLAN主要配置命令**

交换机的VLAN配置涉及一下命令：

1. 进入VLAN配置模式：vlan database，只有在此模式下，才能管理VLAN。
2. 创建vlan:vlan NO name NAME

其中，NO代表BLAN组编号数字，可以任意，但必须保持唯一，NAME表示BLAN别名。每个交换机都缺省包含一个编号为1的VLAN，该VLAN不能删除，缺省情况下所有交换机端口都属于该VLAN。

1. 物理端口划归指定VLAN：switchport access vlan NO。NO代表VLAN组编号。
2. 显示VLAN配置情况：show Vlan。
3. 删除VLAN：no vlan NO，NO代表VLAN组编号。

**实验步骤：**

按照实验环境要求，完成实验拓扑结构连接，并打开相关设备电源。

1. 设置计算机Host1和Host2网卡IP地址。主机网卡IP地址设置如下：

Host1：IP地址=192.168.0.12，子网掩码=255.255.255.0

Host2：IP地址=192.168.0.10，子网掩码=255.255.255.0

1. 测试初始阶段同子网主机之间连通性。用双绞线将Host1和Host2连接到交换机任意端口。从Host1打开命令行窗口，测试连通性，键入”ping 192.168.0.10“，连通就表示试验成功。
2. 交换机配置。启用超级终端，创建VLAN2，并将端口9等划分给VLAN2。

①进入特权模式：switch>en,Enable Secret Password=cisco.

②建立VLAN2：

进入vlan配置模式：switch#vlan database；

添加vlan2：switch（vlan）#vlan 2 name vlan；

退出配置和生效：switch（vlan）#exit。

③为VLAN2分配端口：

进入配置模式：switch#config t；

进入f0/9端口：switch(config)#int f0/9;

将端口f0/9端口分配给vlan2：switch(config -if)#switchport access vlan2；

推出配置和生效：switch(config -if)#exit;

仿照操作，将f0/10-f/16端口分配给vlan2；

查看VLAN2配置：switch# sh vlan name vlan2。

（4）重新测试不同VLAN之间主机连通性。将Host1插到端口1，Host2插到端口9.Host1打开命令行窗口，测试连通性，键入”ping 192.168.0.10“，无法连通就表示实验成功，因为两台主机分别位于VLAN1和VLAN2。

**实验分析：**

本实验在以太网(组网)实验的基础上，将交换机划分为了三个VLAN口，通过VLAN形成虚拟广播域，成为一个独立网络，也正因为如此，只有位于同一个VLAN的两台计算机才能够相互连通。

本实验需要掌握如何对VLAN进行配置，其余的实验步骤如计算机与交换机相连，更改计算机IP地址的步骤则与以太网(组网)实验相同，但是需要注意的是在本实验中需要保证两台客户机的IP地址相同，否则即使处于同一VLAN也无法进行正常互通。

在进行本实验的过程中，可以分别尝试两台客户机处于同一VLAN使用同一IP地址、处于同一VLAN使用不同IP地址、处于不同VLAN使用同一IP地址、处于不同VLAN使用不同IP地址，实验结果发现只有在两台客户机处于同一VLAN使用同一IP地址的情况下才能够实现正常的数据互通。但是在实验过后查阅资料的过程中发现可以通过将两台计算机IP地址设置为不同的独立子网来实现互通，但这一方法还未经过实验验证，需要进行进一步的尝试。

# **物理地址解析**

**物理地址解析实验**

**实验目标**

IP 网络不具有实际通信能力，需要将 IP 数据包封装在物理帧中进行传输，封装前需要

对下一跳 IP 地址进行物理地址解析，物理地址解析式理解 IP 封装的关键知识点，但是物理地址解析行为非常隐蔽，难以察觉。实验利用 ARP 协议的缓存机制来简介证明物理地址解析行为的发生

**实验原理**

物理地址解析指由 IP 地址获取物理地址的过程。IP 网络是虚拟网络,需要将 IP 数据包

封装在物理帧中进行传输,在构造物理帧的过程中,需要使用物理地址作为源节点和目的节

点 ARP 协议是 IP 协议的辅助协议,负责完成 IP 地址解析。在 IP 协议传输前,将调用 ARP

协议获取下一跳节点物理地址。 ARP 数据包称为 ARP 消息。ARP 是底层协议,ARP 消息本身也是直接封装在物理帧中进行传输的。ARP 协议采用消息交互方式来实施。假如第二台主机需要解析第三台主机的网卡物理地址,知道其 IP 地址为 IP3。协议实施分两次通信:①第二台主机对地址解析请求消息进行广播。构建 ARP 解析请求消息,ARP REQ:IP3 表示请求解析 IP3 主机物理地址的 ARP 请求消息,请求消息被直接封装到物理帧中,并设置目标物理地址为以太网广播地址 FF:FF:FF:FF:FF:FF,然后发送该物理帧。2 网络广播域内的所有主机,即第一、三和四主机,均将接收到该广播的 ARP 请求消息,并检查其解析请求消息中存放的目标 IP 地址;最终处理结果: 其他主机因 IP 地址不是 IP3 则丢弃 ARP 请求消息,只有第三台主机的 IP 地址才是 IP3，负责解析其物理地址,构建 ARP 响应消息,ARP RESP:IP3/MAC3 表示 ARP 响应消息填 写了自身的物理地址 MAC3,并依据请求中源物理地址 MAC2 直接给该地址返回 ARP 响应消息;第二台主机接收 ARP 响应消息﹐从ARP 响应消息中取出 MAC3,就得到了解析的物理地址。两台主机往往需要连续传输多个IP 数据包,为避免重复进行地址解析,ARP 协议软件引进了级存机制,一旦收到响应消息,解析地址取出使用后还要缓存起来。每当 ARP 需要解析寸,首先查询 ARP 缓存表;一旦在缓存中找到解析地址,就不再发送解析请求,而是直接使用夏存中保存的解析地址,只有在找不到解析地址时,才利用网络进行地址解析。缓存表填满时,则将最长久未被使用的解析项替代出来;另外,还改足」友行 o 时 Ip 不放便用的项,都将会被自动删除,以消除长时间缓存带来的副作用。

**实验环境**

两台计算机和一台交换机担当实验设备,使用两根以太网络线,将两台计算机网卡都用 F 线直接连接交换机。主机 Hostl 作为地址解析源节点,另一台主机 Host2 作为地址解析目未使用 Windows 操作系统自带的 ARP 工具软件作为实验工具。

**实验内容**

1. 实验概况

在一个子网内,地址解析源节点先删除 ARP 缓存表,然后向地址解析目标节点发出 pi 测试命

令,就会触发ARP 协议去解析目标节点的网卡物理地址,解析后,解析地址会存放其 ARP 协

议缓存表内。通过前后对比 ARP 缓存表的解析地址变化,就可以间接证明目标点的物理地址得到了解析。

(1）配置主机 IP 地址,使得地址解析源节点和目标节点处于相同子网。

(2) 地址解析源节点删除 ARP 缓存表。

(3）地址解析源节点用 ping 命令测试目标节点是否连通。

2.ARP 软件工具

ARP 软件工具用来管理 ARP 缓存表,在命令行窗口下使用,常用命令:,9。(1）显示 ARP 缓

冲区解析条目:ARP-a。

**实验步骤**

按照实验环境要求,完成实验拓扑结构连接,并打开相关设备电源。

(1)为 Hostl 和 Host2 设置 IP 地址。设置 IP 地址,可不设网关地址,限于篇幅,摸见 1.4.1 小节。主机网卡 IP 地址设置如下: Host1:IP 地址=192.168.0.12,子网掩码=255.255.255.0,

Host2:IP 地址=192.168.0.10,子网掩码一 255.255.255.0,网关空缺Host2 触发对 Hostl 的地址解析。

**实验分析**

本次实验使用ARP地址协议来通过IP地址获取物理地址，其中最重要的地方在于理解在这一过程。而实验中的主要难点在于对于两个主机的配置。 只要完成了两个主机的配置就能够很好地展现物理地址解析的过程。

经过这一实验之后，让我更进一步理解了 IP 数据报在不同网络之间的传输过程，理解

到在网络中实际传输的并不是按照 IP 地址进行的，而是在每一段使用物理地址进行传输。

# **主机路由实验**

**主机路由实验**

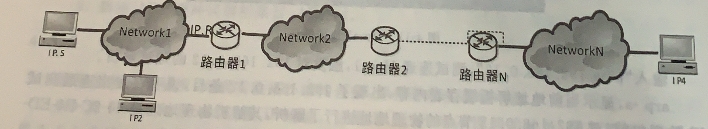
**实验目标：**

按照网际网原理，IP网络是个多跳网络，两个节点之间的传输将穿越多个IP子网，经过多个路由器，才能到达目标主机，而这一切均有赖于路由机制完成。一般认为，路由是路由器的专利。实际上，主机上也设置了路由表，只是较为隐蔽。主机路由表是理解主机和路由器建立转发关系的关键所在。实验构建两个IP子网，利用单路由器实现子网互联，验证主机缺省网关作用。

1. 深入了解主机路由机制。
2. 了解和掌握主机路由配置方法。

**实验原理：**

IP网络模型中，不存在孤立的IP子网络，所有子网都应通过路由器连在一起，因此每个IP子网对外均有一个进出口，称之为网关，网关相连就是路由器。根据源主机节点和目标节点所在子网关系，主机存在两种传输目标：子网内或子网外，其传输目标策略模型如下图所示：



假设IP.R是路由器1接入节点地址，IP.S，PI2和IP.R同属一个IP子网，其网络地址为Network1；IP4在子网外，其网络地址为NetworkN。

为表述方便，用变量IP.T表示当前目标主机地址，变量IP.NP表示下一跳地址；变量M.S表示源节点的网络掩码。

1. 源节点所在网络地址为N.S=Network1.目标节点所在网络地址N.T=IP.T&&&M.S
2. 路由策略：

如果N.S=N.T，则IP.NP=IP.T；否则IP.NP=IP.R

对于目标主机IP2，N.T=Network1，即N.S=N.T，则下一跳地址IP.NP=IP2。

对于目标主机是IP4，N.T=NetworkN,即N.S<>N.T，则下一跳地址IP.NP=IP.R，应先把IP数据包发给路由器1，再通过路由器1转发出去，最终由路由器N发送给节点IP4.

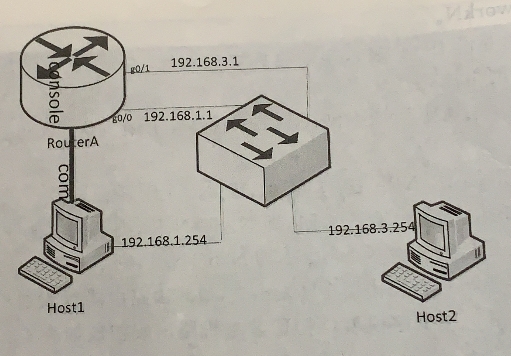
主机路由表贯彻了上述策略模型，则IR.S主机路由表构成如下：

|  |  |  |
| --- | --- | --- |
| 目标网路 | 掩码 | 下一跳 |
| Network1 | M.S | Deliver direct |
| 0.0.0.0 | 0.0.0.0 | IP.R |

主机路由表比较简单，只有两条条目，第一条以本机网络地址为目标，就直接访问该目标节点，用Deliver direct表示；第二条是缺省路由，即除本机网络外的其他网络地址，就转发到路由器接口地址。

**实验环境：**

主机路由实验拓扑结构如下图所示：



实验环境由一台路由器、两台计算机和一台交换机组成。由交换机担当网络连接设备，将路由器两个以太网端口和两台计算机网卡都用网线直接连接到交换机；通过串行线将主机Host1串口com同路由器Console口连接起来，启用其超级终端作为路由器管理的操作平台，主机Host2则使用命令行窗口用于连通测试。为方便处理，所有IP子网掩码都设置成255.255.255.0，配置参数如下：

1. 路由器A设置。以太网接入端口g0/0，其IP地址设置为192.168.1.1；以太网接入端口g0/1，IP地址设置为192.168.3.1.
2. 主机Host1网卡地址设置为192.168.1.254，网关地址设置成192.168.1.1。
3. 主机Host2网卡地址设置成192.168.3.254，网关地址设置成192.168.3.1。

**实验内容：**

设置两个子网，子网间无法联通。用路由器直接连接这两个子网，为子网联通进行路由，但前提是主机设置IP地址时，必须将缺省网关设置为路由器以太网端口地址，以便通过路由器实现路由。

1. 主机路由实验
2. 配置主机IP地址，并将缺省网关地址设置为路由器以太网端口地址。
3. 配置路由器哥哥以太网端口地址，并启用路由功能。
4. 验证子网互通。测试两台主机是否连通。
5. 主机网关缺失实验

在上述连通基础上，取消主机缺省网关，再测试两台主机是否联通。

1. 涉及路由器配置命令
2. 端口IP地址配置命令：ip address<address> <subnet mask>其中，address指IP地址，subnet mask指地址掩码
3. 启用端口功能命令：no shutdown
4. 开启路由功能命令：ip routing

zazaiz**实验步骤：**

按照实验环境要求，完成实验拓扑结构连接，并打开相关设备电源。

**实验一：不同校验方式通信实验**

1. 配置主机Host1和Host2地址，并测试子网连通性。

①配置Host1和Host2地址，主机网卡IP地址设置如下：  
Host1：IP地址=192.168.1.254，子网掩码=255.255.255.0，网关=192.168.1.1

Host2：IP地址=192.168.3.254，子网掩码=255.255.255.0，网关=192.168.3.1

②测试子网连通性。Host1打开命令行窗口。

输入“ping 192.168.3.254”，没有连通，原因是Host1和Host2的网络地址不同，网关地址虽设置了，但网关节点并不存在，设置无法起作用。

③查看主机Host1主机路由表。Host1使用命令行窗口。

输入“Route print”，可以看到缺省路由（0.0.0.0 0.0.0.0）是192.168.1.1。

1. 配置RouterA以设置子网网关。启用超级终端，配置路由器两个以太网端口分别成为两个子网的网关。

①进入配置模式

进入特权模式：routerA>en,Enable Secret Password=cisco

进入配置模式：routerA#config t

②192.168.1.0/24网关配置。

进入g0/0以太网端口配置模式：routerA(config)#int g0/0

设置IP地址：routerA(config-if)#ip address192.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置失效：routerA(config-if)#exit

③192.168.3.0/24网关配置。

进入g0/1以太网端口配置模式：routerA(config)#int g0/1

设置IP地址：routerA(config-if)#ip address 192.168.3.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置失效：routerA(config-if)#exit

④启用路由功能。

启用路由功能：routerA(config)#ip routing

退出配置模式，使配置失效：routerA(config)#exit

（3）测试子网连通性。测试主机Host1是否连通主机Host2，Host1中打开命令行窗口，输入“ping 192.168.3.254”，表示Host1已连通主机Host2，主机路由表和主机网关发生作用，发挥了路由器路由转发功能。

**实验二：主机网关缺失实验**

取消主机Host1缺省网关地址，并测试同Host2连通性。

1. 重置主机Host1的缺省网关地址，删除默认网关地址->“确定”。
2. 测试子网连通性。Host1打开命令行窗口，输入“ping 192.168.3.254”，表示不能连通Host2.网关节点存在，但主机的网关地址没有设置，将不能发挥网关节点作用，无法访问其他子网。

**实验分析：**

本实验是本课程第一次真正意义上使用互联网进行多个客户机连接通信的实验，主要需要掌握的是主机路由机制以及配置方法。在本实验中初步接触了子网、掩码、网关等概念，了解其所代表的具体内容和作用。

本实验使用单路由器实现了子网互联，需要使用一台交换机担当网络连接设备，实验当中将路由器两个以太网端口和两个计算机网卡使用网线直接连接到交换机来实现数据交换功能，作为主机的客户机需要用串行线将串口com同路由器Console口连接起来，从而才能够更改路由器的ip地址。

实验中必须保证网关节点存在并且主机的网关地址已经设置才能够完成基础的主机与客户机交互工作，如果主机与客户机网络地址不同，那么网关节点并不存在，即使设置了网关节点但仍然无法起作用；相反，如果取消了主机的缺省网关地址，则主机的默认网关地址处于未设置状态，在该种情况下主机无法发挥网关节点作用，因此无法访问其他子网。

本实验属于使用网络进行通信的基础实验，但是却涉及到许多重要的概念与过程，需要能够熟练掌握实验要求掌握的内容，对以后的学习很有帮助。

# **静态路由配置实验**

**静态路由配置实验**

**实验目标：**

静态路由，是指由人工根据网络拓扑结构创建路由表。路由器需要依靠路由表来转发IP数据包，该实验是路由器实验中最基础的实验，后续的几个实验都以静态路由为基础。静态路由也是理解路由原理最直观的途径。实验模仿两个远程子网的互联，两个子网在本地各界一个路由器，路由器之间用远程网络相连，使用静态路由实现远程子网互联。

1. 深入了解IP路由基本原理。
2. 了解和掌握配置静态路由配置方法。

**实验原理：**

静态路由是指通过人工编辑方法，在路由器中直接设置路由表。静态路由表可以由多条路由条目组成，下表是静态路由表典型结构：

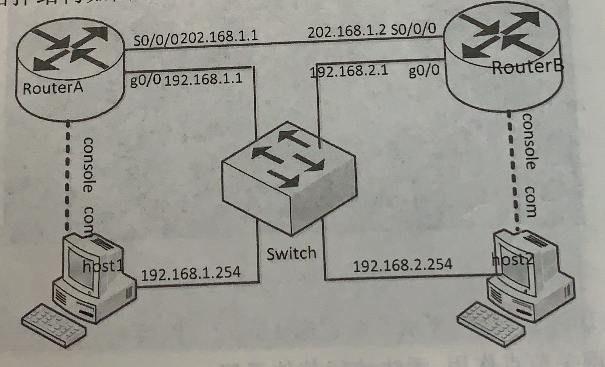
|  |  |  |
| --- | --- | --- |
| 目标网络 | 掩码 | 下一跳 |
| Local Network Address | M.S | Deliver direct |
| Network Address | M.S | IP.R |

表中代表两类路由条目，第一类目标网络是本地网络地址，指直接跟路由器端口相连的IP子网网络地址，M.S是网络掩码，路由器就会直接发送IP数据包给该目标节点；第二类目标网络是非本地网络，路由器会转发到IP.R地址,该地址往往是通往目标节点的邻居路由器接入地址.

静态路由最大的优点是能解释路由的基本原理,因为路由由手工配置出来,一般适用于比较简单的网络环境,工程实践采用动态路由为主.

**实验环境：**

静态路由实验拓扑结构如下图所示：



实验环境由两台路由器、两台计算机和一台交换机组成，模拟两个远程子网互联。使用单根串行交叉线将两个路由器的串口对接起来，代表路由器之间的远程网络；将路由器以太网端口和两台计算机网卡都用网线直接连接到交换机，由交换机担当网络连接；通过串行线将计算机串口com同路由器console口连接起来，两台计算机超级终端将作为路由器管理的操作平台。为方便处理，所有IP子网掩码都设置成255.255.255.0，其他配置参数如下：

1. 路由器A配置。以太网端口g0/0，IP地址设置为192.168.1.1；串口s0/0/0，IP地址设置为202.168.1.1。
2. 路由器B配置。以太网端口g0/0，IP地址设置为192.168.2.1；串口s0/0/0，IP地址设置为202.168.1.2。
3. 主机host1网卡地址设置为192.168.1.254，网关地址设置成192.168.1.1。
4. 主机host2网卡地址设置为192.168.2.254，网关地址设置成192.168.2.1。

**实验内容：**

1. **实验概况**

两个以太网子网各自在本地连接一个路由器，然后将两个路由器的串口连接起来配置成远程连接子网，是的两个以太网子网经过该连接子网实现互通。

1. 以太网子网配置。使得主机和所连接路由器的以太网端口节点同处一个子网，RouterA和Host1为一组，RouteB和Host2为一组。主机的缺省网关设置成路由器以太网端口节点地址。
2. 配置静态路由。为两个路由器各自配置静态路由表，开启路由功能。
3. 验证子网互通。测试两台主机Host1到Host2连通。
4. **涉及配置命令**
5. 端口IP地址配置命令：ip address<address><subnet mask>

其中，address代表IP地址，subnet mask代表地址掩码。

1. 静态路由表添加命令：ip route<Dest.address><subnet mask><NextHop.address>

其中，Dest.address代表目标网络地址，subnet mask代表目标网络地址掩码，NextHop.address代表下一跳地址。

1. 查看静态路由表命令：show ip route

**实验步骤：**

按照实验环境要求，完成实验拓扑结构连接，并打开相关设备电源。

1. 配置主机Host1和Host2网卡地址，测试连通性。

①配置主机网卡地址，主机网卡IP地址设置如下：  
Host1：IP地址=192.168.1.254，子网掩码=255.255.255.0，网关=192.168.1.1

Host2：IP地址=192.168.2.254，子网掩码=255.255.255.0，网关=192.168.2.1

②测试子网联通。Host1打开命令行窗口，测试Host2是否连通。

输入“ping 192.168.2.1”，没有联通，因为两个节点在不同两个子网中，同关节点还不存在，无法通过网关连通。

1. 路由器RouterA配置。启用Host1超级终端，进行网关设置、远程连接子网设置和静态路由设置。

①进入配置模式。

进入特权模式：routerA>en,Enable Secret Password=cisco

进入配置模式:routerA#config t

②网关配置，以太网端口作为192.168.1.0/24子网网关。

进入以太网端口配置模式：routerA(config)#int g0/0

设置IP地址：routerA(config-if)#ip address 192.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置生效:routerA(config-if)#exit

③远程连接子网配置。

进入串口配置模式：routerA(config)#int s0/0

设置IP地址：routerA(config-if)#ip address 202.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置生效：routerA(config-if)#exit

④配置静态路由表。

添加对端路由：routerA(config)#ip route 192.168.2.0 255.255.255.0 202.168.1.2

退出配置模式，使配置生效：routerA(config)#exit

查看路由表：routerA#sh ip route

1. 路由器RouterB配置。启用Host2超级终端，进行网关设置、远程连接子网设置和静态路由配置，操作方式同RouterA，具体配置如下：

①进入配置模式。

进入特权模式：routerB>en,Enable Secret Password=cisco

进入配置模式:routerB#config t

②网关配置，以太网端口作为192.168.2.0/24子网网关。

进入以太网端口配置模式：routerB(config)#int g0/0

设置IP地址：routerB(config-if)#ip address 192.168.2.1 255.255.255.0

开启端口：routerB(config-if)#no shut

退出端口配置模式，使端口配置生效:routerB(config-if)#exit

③远程连接子网配置。

进入串口配置模式：routerB(config)#int s0/0

设置IP地址：routerB(config-if)#ip address 202.168.1.2 255.255.255.0

开启端口：routerB(config-if)#no shut

退出端口配置模式，使端口配置生效：routerB(config-if)#exit

④配置静态路由表。

添加对端路由：routerB(config)#ip route 192.168.1.0 255.255.255.0 202.168.1.1

退出配置模式，使配置生效：routerB(config)#exit

查看路由表：routerB#sh ip route

1. 测试子网连通。测试Host1是否连通Host2，列出传输路径，Host1打开命令行窗口。

测试连通性：ping 192.168.2.254，连通就表示实验成功，网关、连接子网和静态路由均发挥作用。

跟踪路由：tracert 192.168.2.254，经过了192.168.1.1和202.168.1.2两个中间节点，恰好是网关地址（路由器A的以太网端口地址）和路由器B的串口地址。

**实验分析：**

本实验是路由器实验的基础实验，在往后的路由器实验中都是采用静态路由的方法，故本实验需要好好掌握。本实验的需要两台路由器、两台计算机和一台交换机，两台计算机通过网线直连路由器，并且使用单根串行交叉线将两个路由器的串口对接起来，用于模拟路由器之间的远程网络，另外，两台计算机同样需要连接到交换机。

静态路由是指通过人工编辑方法，在路由器中直接设置路由表，拥有许多优点，例如[网络安全](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%AE%89%E5%85%A8" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)保密性高。[动态路由](https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E8%B7%AF%E7%94%B1" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)因为需要[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)之间频繁地交换各自的[路由表](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E8%A1%A8" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)，而对路由表的分析可以揭示网络的[拓扑结构](https://baike.baidu.com/item/%E6%8B%93%E6%89%91%E7%BB%93%E6%9E%84" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)和[网络地址](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)等信息、不占用网络带宽，因为静态路由不会产生更新流量，但最大的优点是能解释路由的基本原理,因为路由由手工配置出来,一般适用于比较简单的网络环境，但是在工程环境中，因为[网络管理员](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%AE%A1%E7%90%86%E5%91%98" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)难以全面地了解整个网络的[拓扑结构](https://baike.baidu.com/item/%E6%8B%93%E6%89%91%E7%BB%93%E6%9E%84" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)，而且当网络的拓扑结构和链路状态发生变化时，[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "https://baike.baidu.com/item/%E9%9D%99%E6%80%81%E8%B7%AF%E7%94%B1/_blank)中的静态路由信息需要大范围地调整，这一工作的难度和复杂程度非常高，所以不使用静态路由。

本实验只是利用交换机模拟两个远程子网互联的情况，在连入用于模拟子网互联的交换机的计算机中，两台计算机如果处于不同子网中，则需要保证同关节点存在，网关、连接子网和静态路由正常发挥作用。

对于静态路由其他作用的体现和适用领域，需要在今后的实验中了解，但我们需要很好地掌握静态路由，为今后的实验打下基础。

# **DHCP实验**

**动态IP地址分配DHCP实验**

**实验目的**

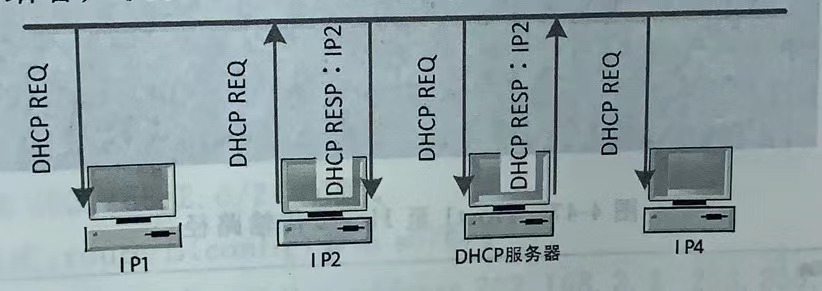
IP地址在人工配置中极易造成重复，从而发生冲突，动态主机配置协议（Dynamic Host Configuration Protocol,DHCP),将自动为子网内的节点网卡分配IP地址。DHCP可以简化网络地址配置，适合非专业人员使用，目前很多网络设备都具备DHCP服务功能，如家庭无线猫也是用DHCP。实验通过开启路由器的DHCP功能，对连接同一个交换机的主机开展动态地址分配。

1. 深入了解动态主机配置协议原理。
2. 了解和掌握DHCP服务的配置步骤。

**实验原理**

IP地址是网络节点网卡在IP网络中的唯一标识，任何网络节点网卡都必须配置IP地址才能使用TCP/IP协议。人工配置IP地址，必须实现分配好子网地址和主机地址，企业中往往设立网管人员负责地址分配工作。尽管如此，因为地址易于遗忘，地址配置时常发生重复冲突。

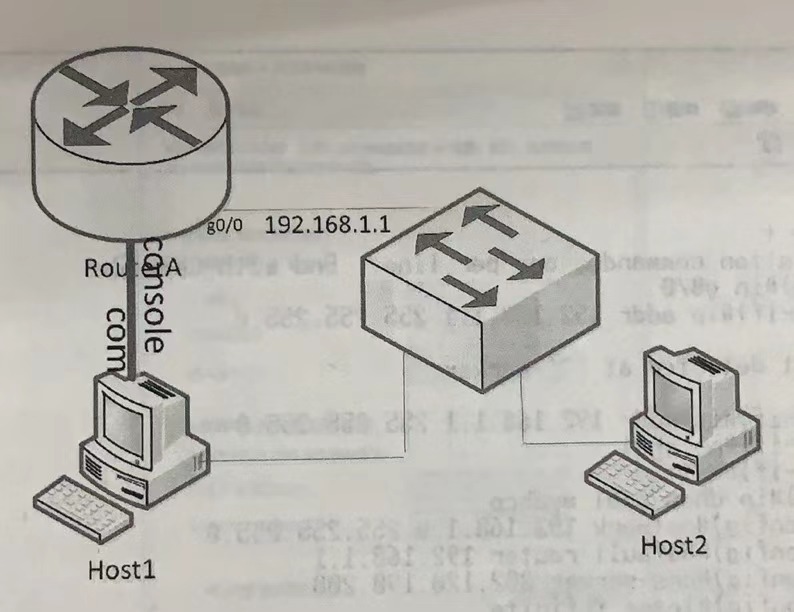
DHCP将为每个新接入交换机的主机节点网卡自动分配一个未使用的IP地址。DHCP属于应用层协议，采用客户机/服务模式。但客户机并不需要知道DHCP服务器IP地址，DHCP客户机会利用物理广播地址发送IP地址分配请求信息，DHCP服务器管理着一段IP地址区间作为分配地址，收到IP地址分配请求消息后，会将一个IP地址分配响应消息返回给客户机。DHCP协议原理如图所示：



1. 假定第二胎主机网卡需要分配IP地址。其DHCP客户机就会广播一个DHCP地址分配请求消息，图中用DHCP REQ来表示，DHCP地址分配请求消息被封装在一个物理帧内，源地址是MAC2，目标地址为以太网广播地址FF:FF:FF:FF:FF:FF。
2. 网络内其他三台计算机均受到了该DHCP REQ消息的广播，但经过判断，只有DHCP服务器才会相应地址分配请求消息。
3. DHCP服务器会按照请求中包含的源物理地址MAC2，在已分配地址数据库中查询；如果查询到，就返回已分配过的IP地址；如果未查询到，服务器则从未分配地址池中按顺序选择一个IP地址分配给MAC2，并在数据库增加这一项。然后，以MAC2为目标地址，返回DHCP地址分配响应消息，DHCP RESP:IP2表示分配地址为IP2。
4. 第二台主机收到DHCP地址分配响应消息，取出地址IP2，对网卡进行设置。

**实验设备**

动态IP地址分配实验拓扑结构如图所示：



实验环境主要由一台路由器、两台计算机和一台交换机组成。路由器将担当DHCP服务器；将路由器以太网端口和两台计算机网卡都用网线直接连接到交换机，由交换机担当网络连接；通过串行线将计算机Host1串口com同路由器console口连接起来，使用超级终端作为路由器管理的操作平台，两台计算机都作为地址分配实验主机。假设没有路由器和交换机设备，也可以用家用无线路由器来代替。为处理方便，所有IP子网掩码都设置成255.255.255.0，配置参数如下：

1. RouterA：以太网端口f0/0设置为192.168.1.1。
2. 主机Host1和Host2网卡地址设置成自动地址分配。

**实验内容**

1. **实验概况**
2. 配置DHCP服务器。在路由器上配置DHCP服务。
3. 设置主机地址配置方式为自动获取。重启网卡，查看获取的IP地址。
4. **路由器主要涉及命令**
5. 进入DHCP服务配置模式：ip dhcp pool<NAME>

其中，NAME表示地址池名。

1. 设置待分配网络地址和掩码：network <NETWORK\_ADDRESS><SUBMASK>

其中，NETWORK\_ADDRESS表示待分配网络地址；SUBMASK表示待分配网络地址掩码。

1. 域名设置：domain-name<DOMAIN\_NAME>

DOMAIN\_NAME指域名。

1. 默认网关地址分配设置：default-router<IP\_ADDRESS>

IP\_ADDRESS指缺省网关地址

1. 本地DNS服务器地址分配设置：dns-server<DNS\_ADDRESS>

DNS\_ADDRESS指本地DNS服务器地址。

1. 设置租期：lease infinite
2. 启动DHCP服务：service dhcp

**实验步骤：**

按照实验环境要求，完成实验拓扑结构连接，并打开相关设备电源。

1. 配置RouterA的DHCP服务。使用超级终端，进行网关设置和DHCP服务配置。

①进入配置模式：

进入特权模式：routerA>en,Enable Secret Password=cisco

进入配置模式：routerA#config t

②网关设置。

进入以太网端口配置模式：routerA(config)#int f0/0

设置IP地址：routerA(config-if)#ip address 192.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置生效：routerA(config-if)#exit

③配置DHCP服务

进入DHCP服务配置模式：routerA(config)#ip dhcp pool mydhcp

设置待可分配网络地址：routerA(dhcp-config)#network 192.168.1.0 255.255.255.0

域名设置：routerA(dhcp-config)#domain-name networklab.con

默认网关地址分配设置：routerA(dhcp-config)#default-router 192.168.1.1

本地DNS服务器地址分配设置：routerA(dhcp-config)#dns-server 202.120.190.208

设置租期：routerA(dhcp-config)#lease infinite

设置不参与分配地址：routerA(dhcp-config)#ip dhcp excluded-address 192.168.1.1 192.168.1.7

关闭DHCP冲突日志，以免显示影响操作：routerA(config)#no ip dhcp conflict logging\*5

启动DHCP服务：routerA(config)#service dhcp

退出配置模式，使配置生效：routerA(config)#exit

1. 配置Host1网卡地址自动分配设置。

①设置主机Host1网卡IP地址自动获取，选择“Internet协议版本4”->“属性”。

选择“自动获得IP地址”和“自动获得DNS服务地址”->“确定”。

②查看主机Host1的IP地址分配状态。打开命令行窗口。

释放分配地址：ipconfig /release

重新申请分配：ipconfig /renew

查看已分配的主机Host1网卡IP地址：ipconfig /all，IP地址得到了分配，地址为192.168.1.21，子网掩码为255.255.255.0，默认网关地址为192.168.1.1，本地DNS服务器地址为202.120.190.208。

**实验分析：**

本实验为IP地址在人工分配的情况下容易造成重复的问题提供了一个解决方案，解决了这种情况造成的冲突。使用动态主机配置协议可以为子网内的节点网卡分配IP地址，保证每一个子网的节点网卡拥有不同的IP地址。DHCP理解不算困难，且实现起来较为容易，非常适合非专业人员使用，所以目前很多网络设备都具备DHCP服务功能，包括家庭无线猫。

DHCP需要一个IP池，每当有一个主机节点网卡新接入交换机，就为其分配一个未使用的IP，未使用的IP的分配工作是按照顺序的方式将未使用的IP地址分配给新接入的节点，且DHCP客户机会利用物理广播地址发送IP地址分配请求信息，DHCP服务器管理者一段地址区间作为分配地址，收到IP地址分配请求消息后，会将一个IP地址分配响应消息返回给客户机。

本实验相对而言比较简单，是解决人工分配IP地址重复冲突的问题，但是仍存在一些问题，例如如果开启了DHCP功能的路由器无法提供足够的IP地址会发生什么情况，需要进行额外的实验来分析和证明。

# **组播实验**

**组播实验**

**实验目的**

组播是一对多的传输模式，即一个节点向组播地址发送一个IP数据包，所有该组播的成员都将接收这个IP数据包，具有较高传输效率。组播应用非常广泛，视频会议就是组播的典型应用，正确使用组播对于提高网络传输效率非常有价值。但组播不会自动跨域子网，需要对路由器进行适当设置。实验模仿两个远程子网的互联，两个子网各界一个路由器，路由器之间用远程网络相连，安排一个组播发送节点和两个组播接收节点，配置路由器使得组播能跨越子网。

1. 了解组播协议基本原理
2. 掌握路由器组播配置技能

**实验原理**

IP组播地址指位于224.0.0.0到239.255.255.255之间的IP地址段。组播在一个IP子网内部使用没任何问题，只要直接使用以太网组播地址，其成员均能接收IP数据包。担当组播跨越IP子网时，路由器并不会自动支持组播路由。组播地址使用始终处于动态变化中，不断有成员加入，也不断有成员离开，无法像单播地址那样实现在路由器中设置组播路由。

组播协议包括组成员管理协议喝组播路由协议。组成员管理协议用于管理组播组成员的加入和离开。互联网组播协议（Internet Group Multicast Protocol IGMP）是组成员管理协议，运行在主机和组播路由之间。路由器为建立组播路由必须了解每个组员在网络中的分布，当新成员加入某个组播时，通知路由器，由路由器设置该组播地址，一旦接收到该组播地址数据包就予以转发；日常会定时主动查询组播成员，以维护组播成员构成状况，适应组播成员的动态变化。IGMP协议运行原理如下：

1. 当主机上的一个进程加入组播组时，就必须发送一个IGMP请求消息给路由器，让该主机加入组播组。IGMP协议以主机身份方式而非进程方式加入组播组，多个进程加入同一组播，只需要发送一条IGMP请求消息。进程退出组播组时，则主机不需要发送IGMP请求消息。
2. 组播路由器收到IGMP请求消息。路由器为每个组播维护一个组播组成员表。将发送请求的主机加入组播组成员表中。
3. 组播数据包转发。当路由器收到组播数据包时，就会将该数据包转发到属于组播组成员的主机所连接的端口上。
4. 组播路由器定时发送IGMP查询消息来维护组播组成员表。

①组播路由器定时通过各个端口发送一个IGMP查询消息，了解主机是否还包含属于组播组的进程。

②尚处于组播组成员的主机通过发送IGMP响应消息来恢复IGMP查询，否则，就不用响应IGMP查询。

③组播路由器根据收到的响应消息维护当前组播组成员表。凡是没有相应的主机就表示已退出组播组，相应的主机继续保留在组播组内。

组播路由协议负责在路由器之间交互信息来建立组播树。协议无关组播PIM（Prutocal Independent Multicast）是一个组播路由协议，利用单播路由协议所产生的单播路由表为IP组播提供转发路径。

**实验设备**

两台路由器，使用串行线将两个0串口对接；两台计算机作为操作平台；一台交换机担当网络连接。

[实验网络拓扑]

Router01

Router02

Host1

Host2

202.168.1.0/24

1,s0/0 s0/0,2

f0/0,1 f0/0,1

192.168.x.0/24 192.168.y.0/24

192.168.x.254 192.168.y.254

**实验内容**

1.连接路由器

--打开路由器电源

--使用console线将计算机串口com1与路由器console口直接相连；

--建立HyperTerminal：开始🡪程序🡪附件🡪通讯🡪超级终端🡪名称=router🡪连接=com1🡪Baut Rate=9600,8,no parity, 1 stop bit；

--进入特权模式：router01>en(able) ，Enable Secret Password=cisco

2查看端口状态：

--记录以太0/0口IP地址：router01# sh interface fast 0/0

--记录串口0/0口IP地址：router01# sh interface ser 0/0

3配置快速以太f0/0

--进入配置模式：router01#config t

--进入以太口：router01(config)#in f0/0

--删除旧IP地址： router01(config-if)#no ip address <ipaddress><subnet mask>

--添加IP地址： router01(config-if)#ip address <ipaddress><subnet mask>

--开启端口功能：router01(config-if)#no shut

--开启端口功能：router01(config-if)#no shut

4配置串口s0/0

--退到配置模式：router01(config-if)#exit

--进入串口：router01(config)#in s0/0

--设置新IP地址

5 静态路由

5.1添加对端路由： router01(config)#ip route 192.168.y.0 255.255.255.0 202.168.1.z # 对端网络地址和广域端口地址；

5.2查看路由表：router01# sh ip route

5.3 测试组播不成功

--配置计算机IP地址：McastSend.exe

--测试连通(从计算机)：MCastReiver.exe# 对端计算机

6 配置组播

6.1开启组播功能： router01(config)#ip multicast-routing

6.2配置串口组播方式： router01(config)#int s0/0/0 # 对端网络地址和广域端口地址；

router01(config-if)#ip pim dense-mode #

6.3配置以他口组播方式：router01(config)#int f0/0 # 对端网络地址和广域端口地址；

router01(config-if)#ip pim dense-mode #

6.4查看组播：router01# sh ip mroute

router01# sh ip pim inyter

router01# sh ip pim nei

--测试连通(从计算机): 192.168.y.254# 对端计算机

7查看运行配置：router01# sh running config

可以看到Host1也开始接收组播数据，而且注明是从Host2主机获得，组播的到了路由器的转发，跨越了子网，组播实验。成功。

**实验分析：**

本实验的基础在于组播这一概念。组播技术的初衷是在IP网络中，以"尽力而为"的形式发送信息到某个目标组，这个目标组称为组播组，这样在有源主机向多点目标主机发送信息需求时，源主机只发送一份数据，数据的目的地址是组播组地址，这样，凡是属于该组的成员，都可以接收到一份原主机发送的数据的拷贝，此组播方式下，只有真正信息需要的成员会收到信息，其他主机不会收到。

组播技术解决了单播情况下数据的重复拷贝及带宽的重复占用，也解决了广播方式下带宽资源的浪费。在[组播](https://baike.baidu.com/item/%E7%BB%84%E6%92%AD" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)方式中，信息的发送者称为“组播源”，信息接收者称为该信息的“组播组”，支持组播[信息传输](https://baike.baidu.com/item/%E4%BF%A1%E6%81%AF%E4%BC%A0%E8%BE%93" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)的所有[路由器](https://baike.baidu.com/item/%E8%B7%AF%E7%94%B1%E5%99%A8" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)称为“组播路由器”。加入同一[组播](https://baike.baidu.com/item/%E7%BB%84%E6%92%AD" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)组的接收者成员可以广泛分布在网络中的任何地方，即“组播组”没有地域限制。需要注意的是，[组播](https://baike.baidu.com/item/%E7%BB%84%E6%92%AD" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)源不一定属于组播组，它向组播组发送数据，自己不一定是接收者。多个[组播](https://baike.baidu.com/item/%E7%BB%84%E6%92%AD" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)源可以同时向一个组播组发送[报文](https://baike.baidu.com/item/%E6%8A%A5%E6%96%87" \t "https://baike.baidu.com/item/%E7%BB%84%E6%92%AD/_blank)。

组播协议分为主机与路由器之间的组成员关系协议和路由器与路由器之间的组播路由协议，常用的组播协议除了本次实验用到的IGMP之外，还有CGMP，PIM-SM等。

组播路由实验是属于比较重要的一个实验，因此需要好好掌握。

# **ACL访问控制实验**

**动态IP地址分配DHCP实验**

**实验目的**

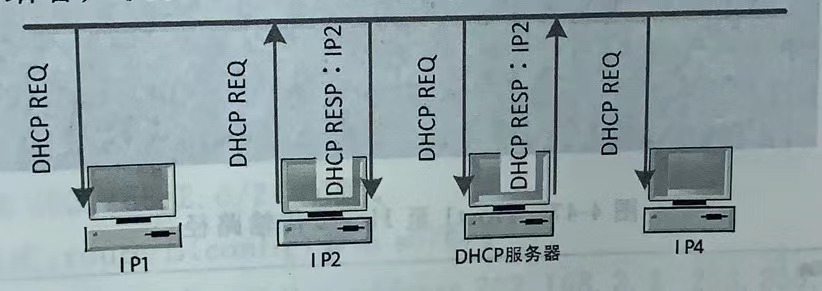
IP地址在人工配置中极易造成重复，从而发生冲突，动态主机配置协议（Dynamic Host Configuration Protocol,DHCP),将自动为子网内的节点网卡分配IP地址。DHCP可以简化网络地址配置，适合非专业人员使用，目前很多网络设备都具备DHCP服务功能，如家庭无线猫也是用DHCP。实验通过开启路由器的DHCP功能，对连接同一个交换机的主机开展动态地址分配。

1. 深入了解动态主机配置协议原理。
2. 了解和掌握DHCP服务的配置步骤。

**实验原理**

IP地址是网络节点网卡在IP网络中的唯一标识，任何网络节点网卡都必须配置IP地址才能使用TCP/IP协议。人工配置IP地址，必须实现分配好子网地址和主机地址，企业中往往设立网管人员负责地址分配工作。尽管如此，因为地址易于遗忘，地址配置时常发生重复冲突。

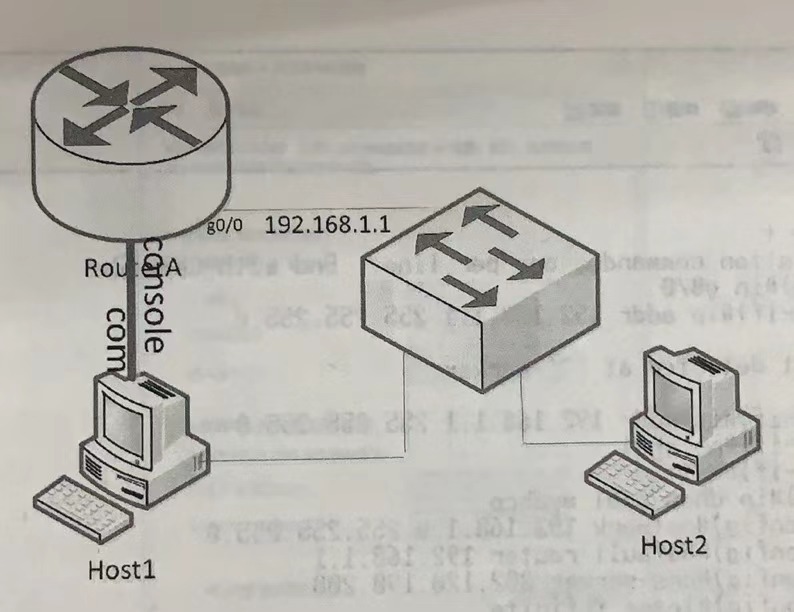
DHCP将为每个新接入交换机的主机节点网卡自动分配一个未使用的IP地址。DHCP属于应用层协议，采用客户机/服务模式。但客户机并不需要知道DHCP服务器IP地址，DHCP客户机会利用物理广播地址发送IP地址分配请求信息，DHCP服务器管理着一段IP地址区间作为分配地址，收到IP地址分配请求消息后，会将一个IP地址分配响应消息返回给客户机。DHCP协议原理如图所示：



1. 假定第二胎主机网卡需要分配IP地址。其DHCP客户机就会广播一个DHCP地址分配请求消息，图中用DHCP REQ来表示，DHCP地址分配请求消息被封装在一个物理帧内，源地址是MAC2，目标地址为以太网广播地址FF:FF:FF:FF:FF:FF。
2. 网络内其他三台计算机均受到了该DHCP REQ消息的广播，但经过判断，只有DHCP服务器才会相应地址分配请求消息。
3. DHCP服务器会按照请求中包含的源物理地址MAC2，在已分配地址数据库中查询；如果查询到，就返回已分配过的IP地址；如果未查询到，服务器则从未分配地址池中按顺序选择一个IP地址分配给MAC2，并在数据库增加这一项。然后，以MAC2为目标地址，返回DHCP地址分配响应消息，DHCP RESP:IP2表示分配地址为IP2。
4. 第二台主机收到DHCP地址分配响应消息，取出地址IP2，对网卡进行设置。

**实验设备**

动态IP地址分配实验拓扑结构如图所示：



实验环境主要由一台路由器、两台计算机和一台交换机组成。路由器将担当DHCP服务器；将路由器以太网端口和两台计算机网卡都用网线直接连接到交换机，由交换机担当网络连接；通过串行线将计算机Host1串口com同路由器console口连接起来，使用超级终端作为路由器管理的操作平台，两台计算机都作为地址分配实验主机。假设没有路由器和交换机设备，也可以用家用无线路由器来代替。为处理方便，所有IP子网掩码都设置成255.255.255.0，配置参数如下：

1. RouterA：以太网端口f0/0设置为192.168.1.1。
2. 主机Host1和Host2网卡地址设置成自动地址分配。

**实验内容**

1. **实验概况**
2. 配置DHCP服务器。在路由器上配置DHCP服务。
3. 设置主机地址配置方式为自动获取。重启网卡，查看获取的IP地址。
4. **路由器主要涉及命令**
5. 进入DHCP服务配置模式：ip dhcp pool<NAME>

其中，NAME表示地址池名。

1. 设置待分配网络地址和掩码：network <NETWORK\_ADDRESS><SUBMASK>

其中，NETWORK\_ADDRESS表示待分配网络地址；SUBMASK表示待分配网络地址掩码。

1. 域名设置：domain-name<DOMAIN\_NAME>

DOMAIN\_NAME指域名。

1. 默认网关地址分配设置：default-router<IP\_ADDRESS>

IP\_ADDRESS指缺省网关地址

1. 本地DNS服务器地址分配设置：dns-server<DNS\_ADDRESS>

DNS\_ADDRESS指本地DNS服务器地址。

1. 设置租期：lease infinite
2. 启动DHCP服务：service dhcp

**实验步骤：**

按照实验环境要求，完成实验拓扑结构连接，并打开相关设备电源。

1. 配置RouterA的DHCP服务。使用超级终端，进行网关设置和DHCP服务配置。

①进入配置模式：

进入特权模式：routerA>en,Enable Secret Password=cisco

进入配置模式：routerA#config t

②网关设置。

进入以太网端口配置模式：routerA(config)#int f0/0

设置IP地址：routerA(config-if)#ip address 192.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置生效：routerA(config-if)#exit

③配置DHCP服务

进入DHCP服务配置模式：routerA(config)#ip dhcp pool mydhcp

设置待可分配网络地址：routerA(dhcp-config)#network 192.168.1.0 255.255.255.0

域名设置：routerA(dhcp-config)#domain-name networklab.con

默认网关地址分配设置：routerA(dhcp-config)#default-router 192.168.1.1

本地DNS服务器地址分配设置：routerA(dhcp-config)#dns-server 202.120.190.208

设置租期：routerA(dhcp-config)#lease infinite

设置不参与分配地址：routerA(dhcp-config)#ip dhcp excluded-address 192.168.1.1 192.168.1.7

关闭DHCP冲突日志，以免显示影响操作：routerA(config)#no ip dhcp conflict logging\*5

启动DHCP服务：routerA(config)#service dhcp

退出配置模式，使配置生效：routerA(config)#exit

1. 配置Host1网卡地址自动分配设置。

①设置主机Host1网卡IP地址自动获取，选择“Internet协议版本4”->“属性”。

选择“自动获得IP地址”和“自动获得DNS服务地址”->“确定”。

②查看主机Host1的IP地址分配状态。打开命令行窗口。

释放分配地址：ipconfig /release

重新申请分配：ipconfig /renew

查看已分配的主机Host1网卡IP地址：ipconfig /all，IP地址得到了分配，地址为192.168.1.21，子网掩码为255.255.255.0，默认网关地址为192.168.1.1，本地DNS服务器地址为202.120.190.208。

**实验分析：**

本实验为IP地址在人工分配的情况下容易造成重复的问题提供了一个解决方案，解决了这种情况造成的冲突。使用动态主机配置协议可以为子网内的节点网卡分配IP地址，保证每一个子网的节点网卡拥有不同的IP地址。DHCP理解不算困难，且实现起来较为容易，非常适合非专业人员使用，所以目前很多网络设备都具备DHCP服务功能，包括家庭无线猫。

DHCP需要一个IP池，每当有一个主机节点网卡新接入交换机，就为其分配一个未使用的IP，未使用的IP的分配工作是按照顺序的方式将未使用的IP地址分配给新接入的节点，且DHCP客户机会利用物理广播地址发送IP地址分配请求信息，DHCP服务器管理者一段地址区间作为分配地址，收到IP地址分配请求消息后，会将一个IP地址分配响应消息返回给客户机。

本实验相对而言比较简单，是解决人工分配IP地址重复冲突的问题，但是仍存在一些问题，例如如果开启了DHCP功能的路由器无法提供足够的IP地址会发生什么情况，需要进行额外的实验来分析和证明。

# **RIP动态路由实验**

**RIP动态路由实验**

**实验目的**

1了解和掌握路由信息协议RIP概念；

2 配置RIP动态路由，实现网际通信。

**实验设备**

两台路由器，使用串行线将两个0串口对接；两台计算机作为操作平台；一台交换机担当网络连接。

**实验网络拓扑**

Router01

Router02

Host1

Host2

s0/0 202.168.1.0/24 s0/0

s0/1,1 s0/1,2

f0/0,1 202.168.2.0/24(BW 256) f0/0,1

192.168.1.0/24 192.168.2.0/24

192.168.1.250 192.168.2.250

**实验内容**

1.连接路由器

--打开路由器电源

--使用console线将计算机串口com1与路由器console口直接相连；

--建立HyperTerminal：开始🡪程序🡪附件🡪通讯🡪超级终端🡪名称=router🡪连接=com1🡪Baut Rate=9600,8,no parity, 1 stop bit；

--进入特权模式：router01>en(able) ，Enable Secret Password=cisco

2查看端口状态：router01# sh interface

记录IP地址；

3配置快速以太网f0/0

--进入配置模式：router01#config t

--进入以太口：router01(config)#in f0/0

--删除旧IP地址： router01(config-if)#no ip address <ipaddress><subnet mask>

--添加IP地址： router01(config-if)#ip address <ipaddress><subnet mask>

--开启端口功能：router01(config-if)#no shut

4配置串口s0/0

--退到配置模式：router01(config-if)#exit

--进入串口：router01(config)#in s0/0

--设置IP地址: router01(config-if)#ip addr 202.168.1.1 255.255.255.0

5配置串口s0/1

--退到配置模式：router01(config-if)#exit

--进入串口：router01(config)#in s0/1

--设置IP地址: router01(config-if)#ip addr 202.168.2.1 255.255.255.0

--设置带宽: router01(config-if)#band 256

6 配置RIP动态路由

--添加RIP： router01(config)#router rip #如果路由功能关闭，rip必须重新配置；

--指定邻居网络：router01(config-router)# network 192.168.1.0

router01(config-router)# network 202.168.1.0

router01(config-router)# network 202.168.2.0

--查看RIP路由表：router01# sh ip route rip

7 测试

--配置计算机IP地址：192.168.x.254

-- router01#no ip domain-lookup

-- router01#trace ip 192.168.2.250

8 跟踪调试

-- router01#debug ip rip#查看信息发送端口

9 被动接口设置

--进入RIP设置： router01(config)#router rip

--以太网端口配置成被动模式： router01(config-router)#passive-interface f0/0

--查看调试：以太口不再发送

**实验分析：**

本实验属于OSPF动态路由协议实验的对比实验，采用了另一种动态路由协议——即RIP动态路由协议，相比OSPF动态路由协议，RIP动态路由协议更加简单、更适用于中小型网络，是一种基于距离矢量（Distance-Vector）算法的协议，它通过UDP报文进行路由信息的交换，使用的端口号为520。

RIP使用跳数（Hop Count）来衡量到达目的地址的距离，称为度量值。在RIP中，缺省情况下，路由器到与它直接相连网络的跳数为0，通过一个路由器可达的网络的跳数为1，其余依此类推。也就是说，度量值等于从本网络到达目的网络间的路由器数量。为限制收敛时间，RIP规定度量值取0～15之间的整数，大于或等于16的跳数被定义为无穷大，即目的网络或主机不可达。由于这个限制，使得RIP不可能在大型网络中得到应用。

相比OSPF动态路由协议实验，OSPF占用的实际链路带宽比RIP少；OSPF使用的CPU时间比RIP少；OSPF适用的内存比RIP大；RIP在网络上达到平衡用的时间比OSPF多。但虽然OSPF看似相比远胜于RIP动态路由协议，但OSPF实现更加复杂，在中小型网路得情况下，使用RIP动态路由协议的情况却更多。

# **OSPF动态路由实验**

**OSPF动态路由实验**

**实验目标：**

动态路由是指由软件根据网络拓扑结构自动构建路由表，适合于较大规模网络的路由配置。最难能可贵的是动态路由能自动适应网络故障，一旦发生网络故障，会根据网络故障发生情况重新生成路由表，及时消除故障的影响。动态路由配置技能是路由器管理的主要工程技能，必须熟悉和掌握。实验模仿两个远程子网的互联，两个子网各接一个路由器，路由器之间用远程网络相连，使用开放式最短路径优先协议(OSPF)实现远程子网互联。

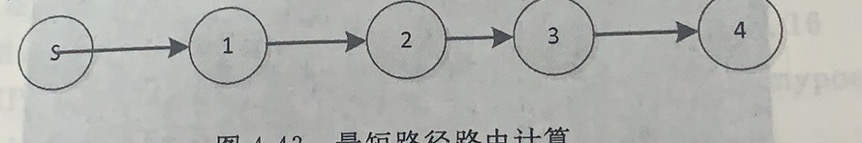
1. 了解动态路由表生成基本原理。
2. 了解最短路径优先算法基本思想。
3. 了解掌握OSPF动态路由技能。

**实验原理：**

开放式最短路径优先协议，是目前路由配置管理中应用最广泛的动态路由协议之一。

OSPF路由协议的核心算法称为最短路径优先，SPF算法以Dijkstra算法为基础。Dijkstra算法可以为图中任意两个节点，找出一条最短路径。SPF算法原理：用路由器相互连接的拓扑图构建一个图，在图中以所在路由器为源点，寻求其到其他路由器节点的所有最短路径，最多将找到若干条最短路径，网络中所有路由器节点都分布在这些最短路径上。

由最短路径很容易计算获得下一跳路由，下图是最短路径路由计算示意图。



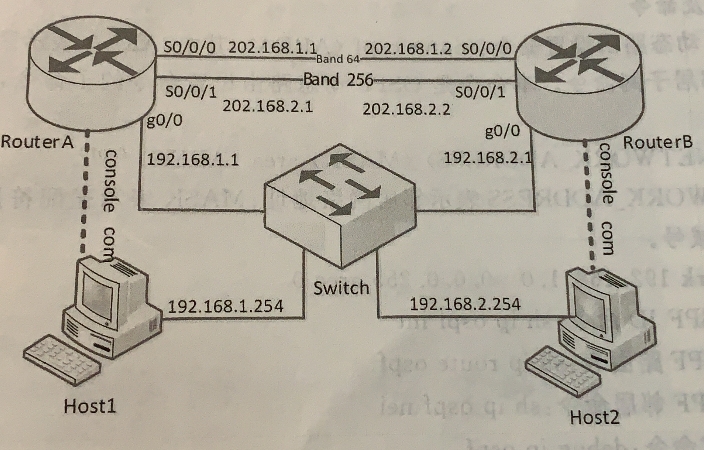
上图表示找到了一条以S为源节点到节点4的最短路径，用反证法就不难证明从S到路径上其他中间节点的路径也是最短路径，对于这条最短路径上所有节点而言，如果作为目标节点，则具有共同下一跳，即这条最短路径上的第二个节点，节点1.上图中节点1，2，3，4为目标节点的下一跳路由都是节点1.因此，由这些最短路径就能生成完整路由表。

链接状态信息是指两个路由器之间连通性，利用链接状态信息可以方便地构建网络拓扑图，以便进行最短路径计算。因此OSPF路由协议也称链路状态协议，其处理步骤如下：

1. 广播链接状态信息。每个路由器生成各自同相邻路由器之间的链接状态信息，然后通过广播方式将链接状态信息发送给所有其他路由器。
2. 每台路由器使用SPF算法计算路由表。路由器一旦受到从其他路由器发送过来地最新链接状态信息，就需要重新构建整个网络拓扑图，并用SPF重新计算各自的动态路由表。以自身为原点，寻找其他节点为目标节点的最短路径并生成路由表。
3. 路由器将定时检测连接状态。一旦发生改变如发生故障，就会重新生成链接状态信息并广播。

**实验环境：**

OSPF动态路由实验拓扑结构如下图所示：



实验环境主要由两台路由器、两台计算机和一台交换机组成。是用两根串行交叉线将两个路由器的串口对接起来，创建两个远程传输子网，便于动态路由选择；将路由器以太网端口和两台计算机网卡都用网线直接连接交换机，由交换机担当网络连接；通过串行线将计算机串口com同路由器console口连接起来，两台计算机超级终端作为路由器管理的操作平台。为处理方便，所有IP子网掩码都设置成255.255.255.0，配置参数如下：

1. RouterA：以太网端口g0/0，IP地址设置为192.168.1.1；串口s0/0/0，IP地址设置成202.168.1.1；串口s0/0/1，IP地址设置为202.168.2.1
2. RouterB：以太网端口g0/0，IP地址设置为192.168.2.1；串口s0/0/0，IP地址设置成202.168.1.2；串口s0/0/1，IP地址设置为202.168.2.2
3. 主机host1网卡地址设置为192.168.1.254，网关地址设置成192.168.1.1
4. 主机host2网卡地址设置为192.168.2.254，网关地址设置成192.168.2.1

**实验内容：**

1. **实验概况**

两个以太网子网，各自在本地链接一个路由器，然后将两个路由器的两个串口连接起来配置成两个远程连接子网，使得两个以太网子网经过这两个连接子网中任意一个实现互通。

1. 以太网子网配置。使得主机和所连接路由器的以太网端口节点同处一个子网，RouterA和Host1为一组，RouterB和Host2为另一组。主机的缺省网关设置成路由器以太网端口节点地址。
2. 配置OSPF动态路由。为两个路由器各自配置网络连接状态，设置两个连接子网的通道成不同地传输速率，OSPF动态路由协议将以传输速率作为最优路由度量标准，选择其中一条通道，自动建立起路由表。
3. 验证子网互通。测试两台主机Host1和Host2之间的连通。
4. **OSPF路由层次性**

就像个缩小规模的外部网关协议，OSPF允许自治系统划分成多个区域，进一步限制路由信息广播，以控制网络上的广播；OSPF还允许区域之间进行通信，以保证自治系统的通信畅通；区域中的一个路由器必须配置成同其他区域的路由器进行通信。

1. **主要涉及命令**
2. OSPF动态路由设置命令：router ospf<AREA>，其中AREA表示区域号
3. 指定邻居子网命令。本命令是OSPF动态路由设置命令的子命令，用于设置OSPF网络连接状态：

network<NETWORK\_ADDRESS><MASK>area<MASK>

其中，NETWORK\_ADDRESS表示邻居网络地址；MASK表示通配符掩码；AREA是区域号。

1. 查看OSPF ID命令：sh ip ospf int
2. 查看OSPF路由表：sh ip route ospf
3. 查看OSPF邻居命令：sh ip ospf nei
4. 跟踪调试命令：debug ip ospf

**实验步骤**

按照实验环境要求，完成实验拓扑结构链接，并打开相关设备电源。

1. 配置主机Host1和Host2网卡地址，测试连通性。

①配置主机网卡地址，主机网卡IP地址设置如下：  
Host1：IP地址=192.168.1.254，子网掩码=255.255.255.0，网关=192.168.1.1

Host2：IP地址=192.168.2.254，子网掩码=255.255.255.0，网关=192.168.2.1

②测试子网联通。Host1打开命令行窗口，测试Host2是否连通。

输入“ping 192.168.2.1”，没有联通，因为两个节点在不同两个子网中，同关节点还不存在，无法通过网关连通。

（2）路由器RouterA配置。启用Host1超级终端，进行网关设置、远程连接子网设置和静态路由设置。

①进入配置模式。

进入特权模式：routerA>en,Enable Secret Password=cisco

进入配置模式:routerA#config t

②网关配置

进入以太网端口配置模式：routerA(config)#int g0/0

设置IP地址：routerA(config-if)#ip address 192.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置生效:routerA(config-if)#exit

③远程连接子网配置。

进入串口配置模式：routerA(config)#int s0/0/0

设置IP地址：routerA(config-if)#ip address 202.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置模式，使端口配置生效：routerA(config-if)#exit

④远程连接子网202.168.2.0/24配置。

进入串口配置模式：routerA(config)# int s0/0/1

设置IP地址：routerA(config-if)#ip address 202.168.2.1 255.255.255.0

开启端口：routerA(config-if)#no shut

设置传输速率：256kbps:routerA(config-if)#band 256

退出配置模式，使配置生效：routerA(config-if)#end

⑤跟踪调试：routerA#debug ip ospf#查看信息发送端口

⑥配置OSPF动态路由。

进入OSPF动态路由配置模式：routerA(config)# router ospf 100

设置网络链路状态：

routerA(config-router)# network 192.168.1.0 0.0.0.255 area 0

routerA(config-router)# network 202.168.1.0 0.0.0.255 area 0

routerA(config-router)# network 202.168.2.0 0.0.0.255 area 0

退出配置模式，使配置生效：routerA(config-router)#end

（3）路由器RouterB配置。启用Host2超级终端，进行网关设置、远程连接子网设置和静态路由设置。

①进入配置模式。

进入特权模式：routerB>en,Enable Secret Password=cisco

进入配置模式:routerB#config t

②网关配置

进入以太网端口配置模式：routerB(config)#int g0/0

设置IP地址：routerB(config-if)#ip address 192.168.2.1 255.255.255.0

开启端口：routerB(config-if)#no shut

退出端口配置模式，使端口配置生效:routerB(config-if)#exit

③远程连接子网配置。

进入串口配置模式：routerB(config)#int s0/0/0

设置IP地址：routerB(config-if)#ip address 202.168.2.1 255.255.255.0

开启端口：routerB(config-if)#no shut

退出端口配置模式，使端口配置生效：routerB(config-if)#exit

④远程连接子网202.168.2.0/24配置。

进入串口配置模式：routerB(config)# int s0/0/1

设置IP地址：routerB(config-if)#ip address 202.168.2.2 255.255.255.0

开启端口：routerB(config-if)#no shut

设置传输速率：256kbps:routerB(config-if)#band 256

退出配置模式，使配置生效：routerB(config-if)#end

⑤跟踪调试：routerB#debug ip ospf#查看信息发送端口

⑥配置OSPF动态路由。

进入OSPF动态路由配置模式：routerB(config)# router ospf 100

设置网络链路状态：

routerB(config-router)# network 192.168.2.0 0.0.0.255 area 0

routerB(config-router)# network 202.168.1.0 0.0.0.255 area 0

routerB(config-router)# network 202.168.2.0 0.0.0.255 area 0

退出配置模式，使配置生效：routerB(config-router)#end

（4）测试子网连通。

测试Host1至Host2连通性及传输路径。Host1打开命令行窗口

测试联通：ping 192.168.2.1，联通就表示网关、远程连接子网和动态路由均发挥作用。

跟踪传输路径：tracert 192.168.2.254，经过了网关、远程连接子网202.168.1.0/24地202.168.1.2端口，没经过远程连接子网202.168.2.0/24

（5）RouterA查看OSPF动态路由配置。切换到Host1超级终端，RouterA查看OSPF动态路由配置

查看OSPF路由表：routerA#sh ip route ospf

查看OSPF邻居：router01# sh ip ospf nei

可以看到，路由器之间有两条通信线路，选择了远程连接子网202.168.1.0/24，该子网经过了s0/0/0端口，其缺省传输速率为1544Kbps，比远程连接子网2地传输速率256K要高，OSPF以传输速率作为路由优化度量标准，因此选择了连接子网202.168.1.0/24.

**实验分析：**

本实验是在静态路由的基础之上进行的实验，虽然采用了动态路由的方法，但是在进行设备相关参数的配置的过程中却有很多相似之处，也凸显出对于超级终端的使用是本实验课程的基础之一。

相比静态路由，动态路由的一大优点在于动态路由能自动适应网络故障，一旦发生网络故障，会根据网络故障发生情况重新生成路由表，及时消除故障的影响，也因此动态路由适用于工程项目中，但也有着更高的复杂度，远程子网之间的互联需要通过各种协议来选择合适的线路。

本实验中采用了OPFS协议，即开放式最短路径优先协议，是目前路由配置管理中应用最广泛的动态路由协议之一。OPFS协议以传输速率作为路由优化度量标准，会选择传输速率更快的端口进行连接，因此在本实验中，需要保证子网之间不同的连接方式拥有着不同的传输速率。

除了本实验中采用的OPFS协议，动态路由协议还包括许多其他协议，比如RIP、IS-IS、IGRP、EIGRP、BGP等。其中我们还需要掌握RIP路由协议。RIP是Routing Information Protocol（路由信息协议）的简称。它是一种较为简单的内部网关协议IGP（Interior Gateway Protocol），主要用于规模较小的网络中，比如校园网以及结构较简单的地区性网络。对于更为复杂的环境和大型网络，一般不使用RIP。

# **帧中继配置实验**

**帧中继配置实验**

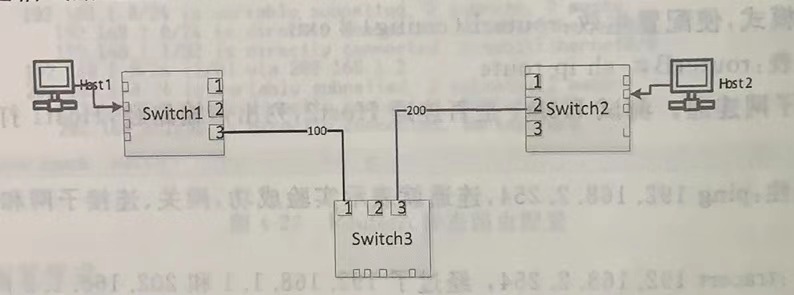
**实验目标：**

广域网是另一类主要有限物理网络，可以实现跨地域的网络连接，承担着骨干传输网络作用，只有ISP(Internet Service Provider，互联网服务提供商)才会拥有，如中国电信、移动等，普通企业很难见到此类设备。本实验利用路由器模拟帧中继交换机，用于远程连接两个以太网，实现网络互连。

1. 了解广域网基本概念
2. 区分二层路由和三层路由基本概念
3. 熟悉帧中继交换机永久虚电路及配置步骤

**实验原理：**

广域网跨越地域，由多个包交换机组成，包交换机分布在各个地域上，相互之间用告诉通信线路直接相连，具有远距离、高性能，承担骨干网络作用。其结构如下图：



包交换机是广域网组成核心，控制着数据的传输。物理网络分一跳网络和多跳网络。一跳网络就是数据从源节点直接到达目标节点，以太网就是典型的一跳网络。多跳网是指数据从源节点发送到目标节点的途中，需要经过其他网络节点才能到达，中间节点称为路由节点，广域网是典型的多跳网络。上图中Switch3就是Switch1和Switch2之间传输的路由节点。广域网路由算法和IP路由算法是完全一样，但两者实现细节不同，基于所属网络协议层层次不同，业界将物理网络路由称为二层路由，IP网络路由则称为三层路由。

广域网连接到互联网，也需要进行类似以太网的地址解析，但方式不同，广域网采用面向连接通信，使用电路号作为通信标识，因此需要将电路号同IP地址建立映射，IP数据包封装时就可以通过映射表进行转换。

帧中继是一种广域网技术，在功能上只实现了物理层和链路层两层功能。帧中继利用数字电话进行传输，不进行差错纠正，依靠高层协议纠错，如果帧包含误码，或者输入缓冲区满，就简单地丢弃到来的帧。帧中继采用面向连接服务，在两个节点之间进行数据通信之前，必须先建立数据链路，然后利用连接表示进行数据交换。

1. **DLCI(数据链路连接标识)**

DLCI是一种电路号，用于标识帧中继源和目的设备间的逻辑链路。DLCI是帧中继网络中的第2层地址。路由器将IP数据包通过帧中继网络转发到下一跳路由器时，必须知道IP地址和DLCI地映射关系才能进行帧地封装。有两种方法可以获得该映射，一种是静态映射，由管理员手动输入；另一种是动态映射。默认时，路由器帧中继接口开启动态映射功能。

1. **PVC(永久虚电路)**

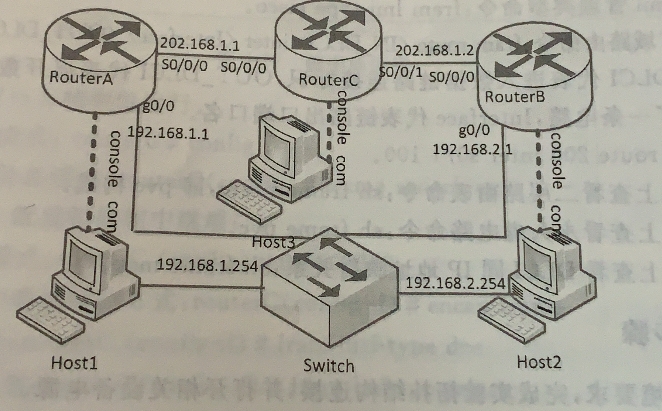
永久虚电路是指人工建立的链路，通过设置帧中继交换机静态路由表实现。一对进出的链路可以配置成一条永久虚电路。

1. **LMI(本地管理接口)**

一种信令标志，负责设备管理，实现帧中继设备之间的连接管理，本地管理接口类型分cisco，ansi和q939a三种，配置时需要将帧中继交换机和路由器相连端口设置成同一种接口类型。

**实验环境：**

帧中继网络实验拓扑结构如下图：



实验环境主要由三台路由器、三台计算机和一台交换机组成，模拟一个由局域网和广域网组成的网际网。路由器（模拟帧中继交换机）使用单根串行交叉线将路由器A和路由器C的广域网串口连接起来，路由器B和路由器C的广域网串口连接起来（构建帧中继网络专线以连接两个远程子网），并使得同RouterC相连的电缆连接线类型均为DCE，将路由器A和luyouqiB以太网端口和两台计算机串口com同路由器console口连接起来，使用各自超级终端作为路由器管理的操作平台。为方便处理，所有子网掩码设置成255.255.255.0，配置参数如下：

1. 路由器A，以太网端口地址：192.168.1.1，串口地址202.168.1.1。
2. 路由器B，以太网端口地址：192.168.2.1，串口地址202.168.1.2。
3. 路由器C和host3，分别充当帧中继设备和配置平台，不配置IP地址。
4. host1网卡配置：192.168.1.254，网关地址设置成192.168.1.1。
5. host2网卡配置：192.168.2.254，网关地址设置成192.168.2.1。

**实验内容：**

两个以太网子网各自在本地分别连接一个路由器，然后用一个帧中继交换机将两个路由器连接起来，是两个以太网子网经过帧中继交换机实现互通。

1. 以太网子网配置。使得主机和所连接路由器的以太网端口节点同处一个子网，RouterA和Host1为一组，RouterB和Host2为另一组。主机的缺省网关设置成立尤其以太网端口节点地址。
2. 配置帧中继交换机。将中间一台路由器C作为帧中继交换机使用，分别用串行线同另两个路由器串口对接，配置成两条帧中继通信链路，形成单挑永久虚电路。
3. 配置静态路由。路由器A和路由器B使用帧中继永久虚电路作为连接网络，实现网际互连，需要将连接端口设置成帧中继类型，以便正确封装；路由表配置还是按照静态路由设置。
4. 启用作为帧中继交换机命令：frame switching.
5. 设置端口为帧中继封装方式命令：encap frame.
6. 设置帧封装类型命令：fram intf-type dce.
7. 设置lmi管理类型命令：fram lmi-type dce.
8. 配置广域路由命令：fram route <IN\_DLCI>inter <Interface><OUT\_DLCI>。

其中，IN\_DLCI代表进入数据链路连接标识，OUT\_DLCI代表离开数据链路连接标识，两个标识构成了一条电路，Interface代表链路出口端口名。

1. 帧中继上查看二层路由表命令：sh frame route，即pvc构成。
2. 路由器上查看永久虚电路命令：sh frame pvc。
3. 路由器上查看DLCI同IP地址映射关系：sh frame map。

**实验步骤**

按照实验环境要求，完成实验拓扑结构链接，并打开相关设备电源。

1. RouterC配置成为帧中继交换机。启用超级终端，将路由器配置成一台帧中继交换机，并建立一跳PVC电路用于连接连个以太网子网。

①确认串口线是否连接正确。必须使得同RouterC相连的电缆连接线类型均为DCE。

进入特权模式：routerC>en,Enable Scret Password=cisco

查看端口s0/0物理线路连接状态及接口类型，必须是DCE：routerC#sh control s0/0

查看端口s0/1物理线路连接状态及接口类型，必须是DCE：routerC#sh control s0/1

看到DCE表示连接正确，否则，要重新连线。

②帧中继配置。配置帧中继永久电路以连接路由器RouterA和RouterB。

1. 启用路由器作为帧中继运行。

进入全局配置模式：routerC# config t

设置成帧中继仿真模式：routerC(config)#frame switching

1. 串口s0/0/0配置模拟帧中继端口。

进入端口配置模式：routerC(config)#int s0/0/0

设置端口为帧中继封装方式：routerC(config)#encap frame

设置帧封装类型：routerC(config-if)#fram intf-type dce

设置lmi管理类型：routerC(config-if)#fram lmi-type cisco

时钟频率设置：routerC(config-if)#clock rate 56000

配置广域路由(pvc)：routerC(config-if)#fram router 200 inter s0/0/1 100

启动端口：routerC(config-if)#no shut

退出端口配置，使端口配置生效：routerC(config-if)#exit

1. 串口s0/0/1配置模拟帧中继端口。

进入端口配置模式：routerC(config)#int s0/0/1

设置端口为帧中继封装方式：routerC(config)#encap frame

设置帧封装类型：routerC(config-if)#fram intf-type dce

设置lmi管理类型：routerC(config-if)#fram lmi-type cisco

时钟频率设置：routerC(config-if)#clock rate 56000

配置广域路由(pvc)：routerC(config-if)#fram router 200 inter s0/0/0 100

启动端口：routerC(config-if)#no shut

退出端口配置，使端口配置生效：routerC(config-if)#end

1. 查看帧中继交换机永久电路。

查看二层路由表：routerC#sh frame route，即pvc构成表，如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| IN\_Port | IN\_DLCI | OUT\_Port | OUT\_DLCI |
| S0/0 | 200 | S0/1 | 100 |
| S0/1 | 100 | S0/0 | 200 |

1. 主机Host1和Host2网卡地址配置，主机网卡IP地址设置如下：

Host1：IP地址=192.168.1.254，子网掩码=255.255.255.0，网关=192.168.1.1

Host2：IP地址=192.168.2.254，子网掩码=255.255.255.0，网关=192.168.2.1

1. 路由器Router A和RouterB配置。

①路由器RouterA端口配置。启用host1超级终端，进行网关设置和帧中继连接端口设置。

1. 进入配置模式。

进入特权模式：routerA>en,Enable Secret Password=cisco

进入配置模式：routerA#config t

1. 192.168.1.0/24网关配置

进入以太网端口配置模式：routerA(config)#int f0/0

设置IP地址：routerA(config-if)#ip address 192.168.1.1 255.255.255.0

开启端口：routerA(config-if)#no shut

退出端口配置，使端口配置生效：routerA(config-it)#exit

1. 帧中继连接端口配置

进入串口端口配置模式：routerA(config)#int s0/0/0

设置IP地址：routerA(config)#ip addr 202.168.1.1 255.255.255.0

设置端口为帧中继封装方式：routerA(config-it)#encap frame

设置lmi管理类型：routerA(config-if)#fram lmi-type cisco

启动端口功能：routerA(config-if)#no shut

退出配置模式，使配置生效：routerA(config-it)#end

②路由器RouterB端口配置。启用host2超级终端，仿照RouterA配置，除网关设置和帧中继连接端口配置，还进行路由配置，具体配置：

a.进入配置模式。

进入特权模式：routerB>en,Enable Secret Password=cisco

进入配置模式：routerB#config t

b.192.168.1.0/24网关配置

进入以太网端口配置模式：routerB(config)#int f0/0

设置IP地址：routerB(config-if)#ip address 192.168.2.1 255.255.255.0

开启端口：routerB(config-if)#no shut

退出端口配置，使端口配置生效：routerB(config-it)#exit

c.帧中继连接端口配置

进入串口端口配置模式：routerB(config)#int s0/0/0

设置IP地址：routerB(config)#ip addr 202.168.1.2 255.255.255.0

设置端口为帧中继封装方式：routerB(config-it)#encap frame

设置lmi管理类型：routerB(config-if)#fram lmi-type cisco

启动端口功能：routerB(config-if)#no shut

退出配置模式，使配置生效：routerB(config-it)#end

1. 路由器RouterB路由配置。添加静态路由。

添加缺省路由：routerB(config)#ip router 0.0.0.0 0.0.0.0 202.168.1.1

启动IP路由功能：routerB(config)#ip routing

退出配置模式，使配置生效：routerB(config)#exit

③路由器RouterA路由配置。查看帧中继线路使用情况，添加静态路由

1. 查看帧中继永久电路使用状况。

查看DLCI同IP地址映射状态，已建立映射：routerA#sh frame map

查看pvc的DLCI编号，routerC配置才能看到：routerA#sh frame pvc

1. 静态路由添加。

添加缺省路由：routerA(config)#ip route 0.0.0.0 0.0.0.0 202.168.1.2

查看静态IP路由表：routerA# sh ip route

1. 测试子网连通。测试主机Host1是否连通主机Host2

Host1打开命令窗口

键入“ping 192.168.2.254”，连通就表示网关、帧中继交换机永久电路和路由均起作用，实验成功。

**实验分析：**

本实验是关于广域网的第一个实验，让我们在了解广域网的概念的基础上第一次了解到广域网的连接方式。广域网可以实现跨地域的网络连接，拥有以下优点：支持用户使用计算机进行远距离的信息交换、覆盖范围广,通信的距离远。但也正因为广域网覆盖范围广的特点，需要考虑的点更多，并且有线路的冗余、媒体带宽的利用和差错处理问题，因此一般由电信部门或公司负责组建、管理和维护，并向全社会提供面向通信的有偿服务，流量统计和计费问题，普通企业很难见到此类设备。

在本实验中采用了利用帧中继交换机进行广域网连接，处于条件原因，本实验中利用路由器模拟帧中继交换机，用于远程连接两个以太网，实现网络互连。在实验中主要注意的是，与静态路由等实验不同，RouterA和RouterB都需要用单根串行交叉线和第三台路由器C连接起来，并且需要保证相连的电缆连接线类型均为DCE。

本实验相对而言属于理解基础概念的实验，因此实验操作相比之下比较简单，但是因为模拟的是广域网这一复杂网络，需要的步骤较多，在实验过程中需要集中注意力，保证设备的连接方式正常，路由器A和路由器B需要处于不同子网等。此外，对于本实验反应的广域网连接方式和概念需要好好掌握。

# **虚拟无线隐藏节点实验**

**虚拟无线隐藏节点实验**

**实验目的：**

1.学习无线网络中隐藏和暴露节点问题

2.利用NS2分析隐藏和暴露节点问题

3.了解OTcl脚本语言的基础知识

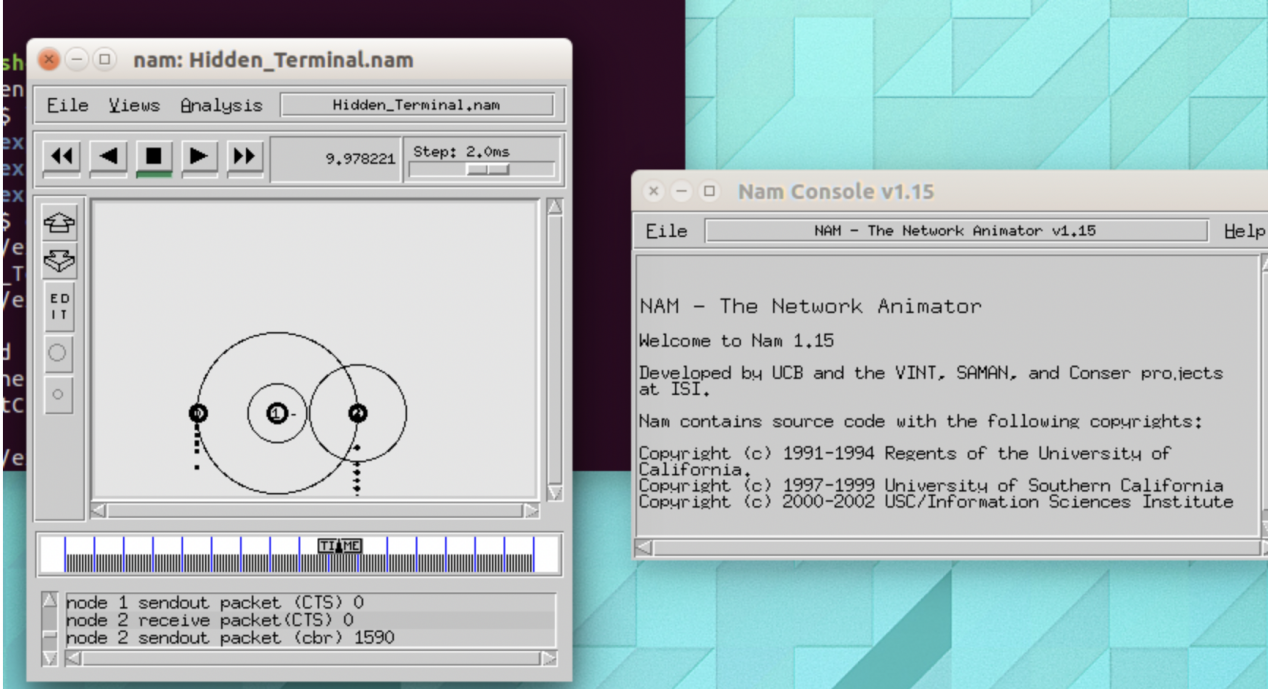
**实验设备：**

树莓派套件、wnt-ns2仿真环境、实验代码。

实验过程：

1.执行实验代码

2.输入“ns Hidden\_Terminal.tcl”命令，运行本实验，截图是运行到一半的时间，注意步长调整为2ms即可，不宜过大。



仿真开始时（0~1s），节点间相互广播路由（DSDV）信息，建立路由表。从1.5s开始，节点0开始向节点1发送数据，需要注意，此时节点0需要先发送RTS报文（实验中设置802.11的RTSThreshold=0，满足数据包大于RTSThreshold=3000，表示开启TRS/CTS，解决暴露终端问题：实验暴露终端请设置RTSThreshold=3000，满足数据包小于RTSThreshold，因为数据包不会大于3000），以保证信道（,1→0）的顺利占用。在2s时，节点2也开始向节点3发送数据，同样，节点2也发送RTS报文，以获得信道（2→3）的顺利占用，并正确传输数据。仿真时间到达15s时，两条流传输同时结束，此后无数据传输，偶尔有链路保活报文。在20s时，仿真实验结束。

**结果分析：**

隐藏节点和暴露节点问题对无线网络传输影响较大，极大降低了网络吞吐量，增加了传输时延，而CTS/RTS机制可较好的解决隐藏和暴露节点问题。

仿真结果可从两方面来分析，即仿真动画和仿真数据。在NS2中演示仿真动画采用NAM，将拓扑结构、数据包传输情况及队列丢包等以动画形式展现，仿真结束后，可通过分析产生的trace数据对仿真过程中的延迟、丢包、吞吐量等进行定量分析。

# **虚拟LTE蜂窝网络组网实验**

**实验目的：**

本实验子项是LTE-EPC蜂窝网络信元切换实验，构建一个多信元场景，每个基站相隔足够远，以致于信元间的重叠区非常小，通过模拟移动终端的移动，根据基站的信号强度，自动进行信元切换，让学生了解LTE-EPC网络的信元的构成原理。

1）理解无线局域网的信元的构成及机制；

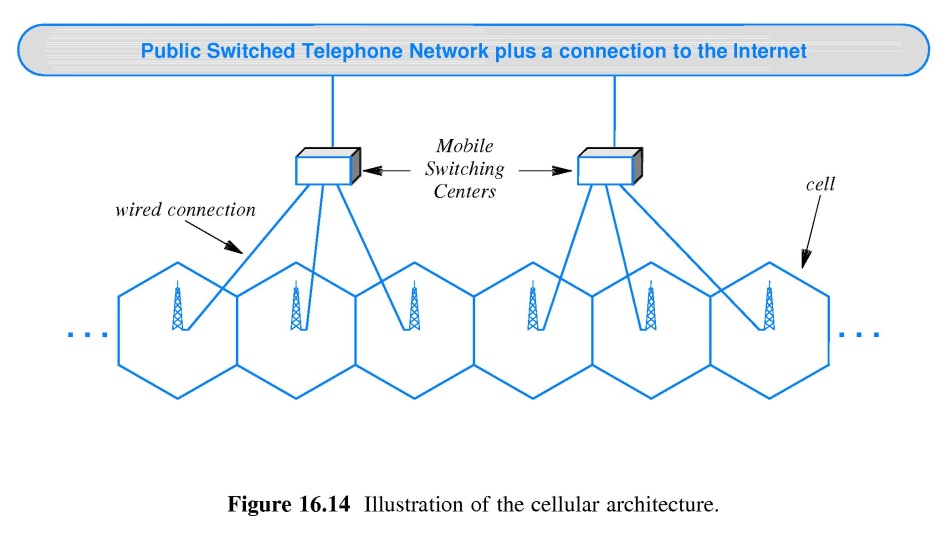
2）理解LTE-EPC网络X2切换原理和机制；

3）理解基于X2接口的内部切换仿真流程。

**实验原理：**

LTE蜂窝网络信元

信元，特指无线广域网无线信号覆盖区，区内都包含一个信号塔作为基站，负责同进入区域内智能终端的无线信号发送和接收，理想的信元是一个正六边形



智能终端通过所在信元的基站接入LTE网络，智能终端在移动过程中，不断进出不同信元，就需要在不同的信元之间进行切换，信元切换是改变用户服务小区的连接方式的过程。

LTE蜂窝网络 X2接口

X2接口是LTE基站切换的接口标准。X2接口主要功能是在LTE系统内，UE在连接状态下从一个基站切换到另一个基站的移动性管理，技术上，前后两个基站分别称为源基站和目标基站。X2接口控制面还可以对各eNodeB之间的资源状态、负责状态进行监测，用于eNodeB负载均衡、负荷控制或者准入控制的判断依据。此外还负责X2连接的建立、复位、eNodeB配置更新等接口管理工作。

**实验内容**

主要实验内容了解在移动终端UE移动过程中，进入不同信元之间而进行信号切换处理情况，模拟控制智能手机移动方向，进行向上、下、左、右或随机方向，引起不同的信元之间的切换，还选择不同的切换算法，充分反映信元切换状况。实验结束后会生成日志文件，验证相邻信元切换情况

**实验步骤**

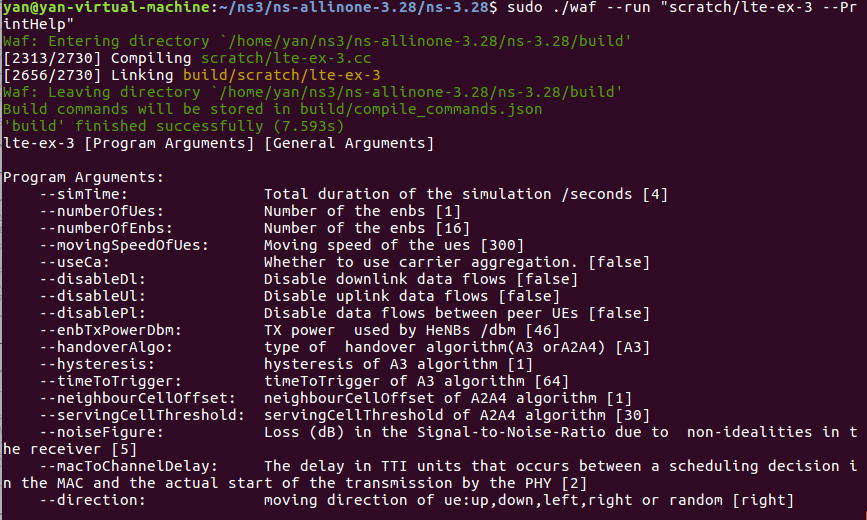
虚拟实验子项目由lab-lte-handover.cc程序担当，其文件目录，/home/yan/ns3/ns-allinone-3.28/ns-3.28/scratch/lab-lte-handover.cc。该程序包含多个参数，便于实验人员调整，核心参数如下：

numberOfUes，

numberOfEnbs，

movingSpeedOfUes等参数，修改一个或多个参数观察对比结果变化。

命令行参数查看命令“sudo ./waf --run "scratch/lte-ex-1 --PrintHelp”命令，如图所示。



LTE蜂窝网络基站切换实验项目子项查看参数

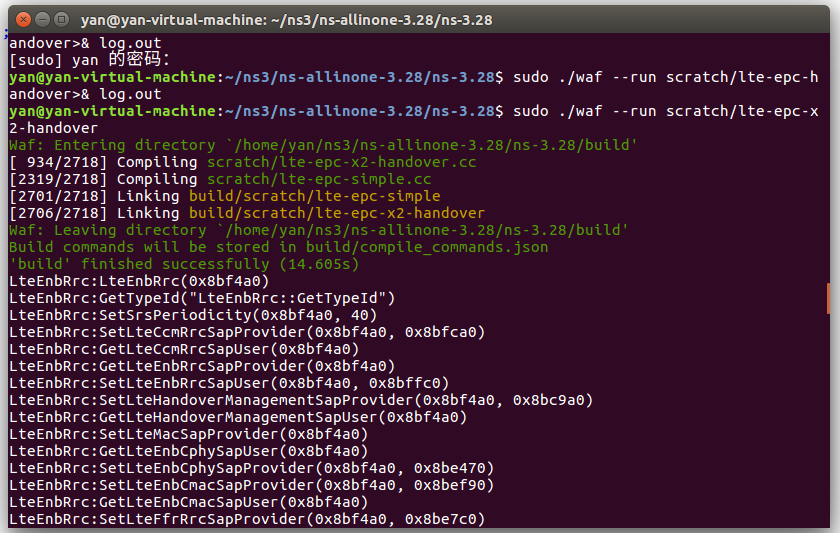
实验步骤如下：

1、运行虚拟实验子项目。

1）打开一个终端窗口。

2）切换到执行目录，/home/yan/ns3/ns-allinone-3.28/ns-3.28/。

3）运行lte-ex-3.cc。sudo ./waf --run "scratch/lab-lte-handover"，输入口令，运行如图所示。



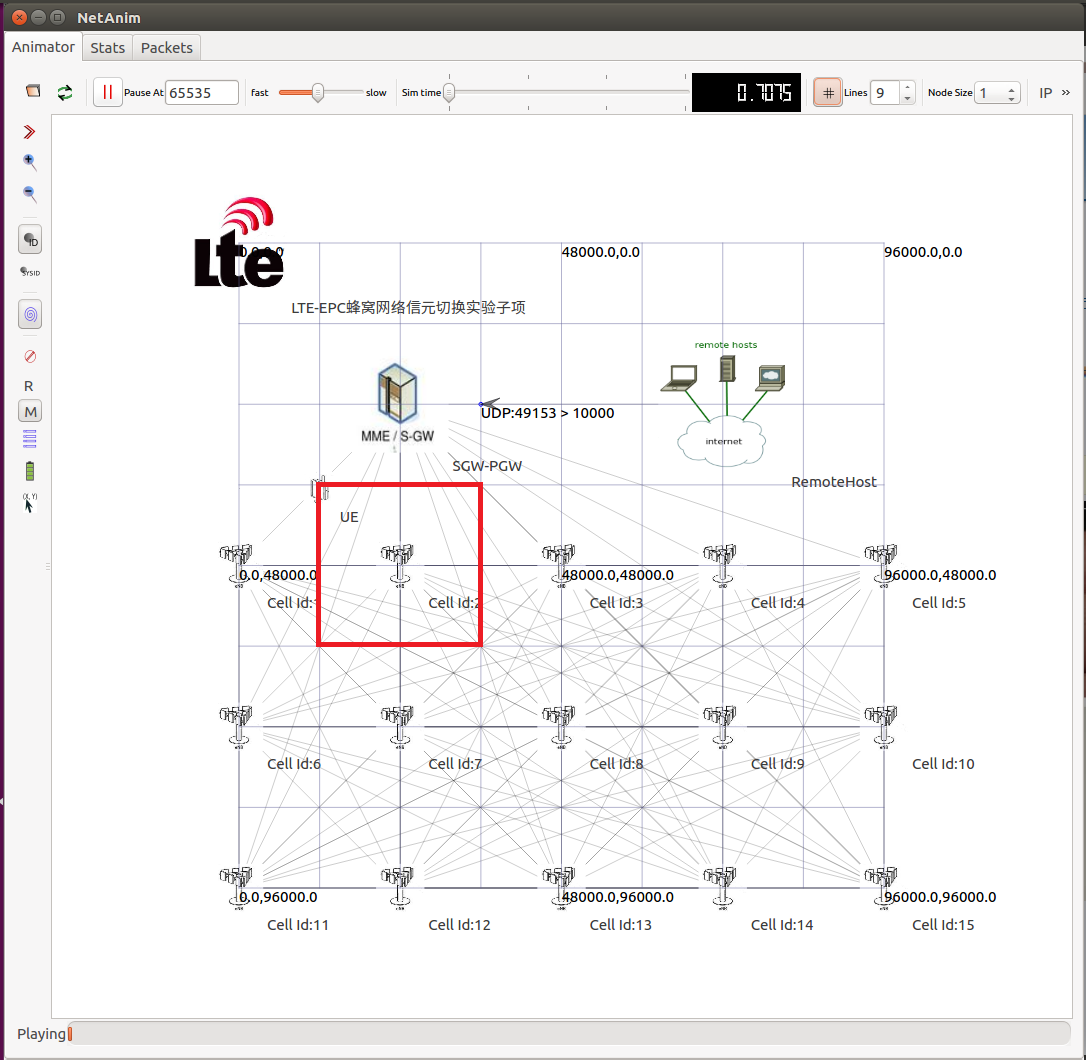
运行NetAnim。

1）打开一个新终端窗口。

2）切换到执行目录，/home/yan/ns3/ns-allinone-3.28/netanim-3.108/。

3）运行NetAnim。./NetAnim

4）打开实验跟踪文件。File🡪lte-ex-3.xml，将以动画方式呈现实验场景



验证日志文件数据验证。打开生成的日志文件log.out，分析在移动过程中移动终端和信元之间的切换，信元的区域可以使用红框所示，如图所示。在日志中，可以看到Service表示切换处理过程。



**实验分析：**

在本次的虚拟实验当中，真结果可从两方面来分析，即仿真动画和仿真数据。在NS3中演示仿真动画采用NAM，将拓扑结构、数据包传输情况及队列丢包等以动画形式展现，仿真结束后，可通过分析产生的trace数据对仿真过程中的延迟、丢包、吞吐量等进行定量分析。

# **NAT网络地址转换实验**

**NAT网络地址转换实验**

**实验目的**

了解和掌握路由器地址转换功能，作为网络安全内容。

**实验设备**

两台路由器，使用串行线将两个0串口对接；两台计算机作为操作平台；一台交换机担当网络连接。

**实验网络拓扑**

Router01

Router02

Host1

Host2

202.168.1.0/24

1,s0/0 s0/0,2

f0/0,1 f0/0,1

192.168.1.0/28 192.168.2.0/28

192.168.1.15/17/254 192.168.2.15/17/254

**实验内容**

1.连接路由器

--打开路由器电源

--使用console线将计算机串口com1与路由器console口直接相连；

--建立HyperTerminal：开始🡪程序🡪附件🡪通讯🡪超级终端🡪名称=router🡪连接=com1🡪Baut Rate=9600,8,no parity, 1 stop bit；

--进入特权模式：router01>en(able) ，Enable Secret Password=cisco

3内部地址转换

3.1 转换方案192.168.1.16—31🡺202.168.1.4--6

3.2 建立NAT

--建立访问控制组：router01(config)# access-list 1 permit 192.168.1.16 0.0.0.15

--建立转换地址池mypool：router01(config)# ip nat pool mypool 202.168.1.4 202.168.1.6 netmask 255.255.255.0

--设置动态NAT关系：router05(config)# ip nat inside source list 1 pool mypool

3.3 端口设置

--设置端口：router01 (config)#in s0/0#端口s0/0

--设置地址：router01 (config-if)#ip address 202.168.1.1 255.255.255.0

--设置转换方向：router01 (config-if)#ip nat outside#出口

--设置端口：router01 (config)#in f0/0#端口f0/0

--设置地址：router01 (config-if)#ip address 192.168.1.1 255.255.255.0

--设置转换方向：router01 (config-if)#ip nat inside#进口

--设置单向路由：router01 (config)#ip route 0.0.0.0 0.0.0.0 202.168.1.2

3.4 测试

--检测：router01# debug ip nat

--配置计算机IP地址：192.168.1.15

--测试连通(从计算机)：ping 192.168.2.1

--配置计算机IP地址：192.168.1.17

--测试连通(从计算机)：ping 192.168.2.1

--查看转换地址表：router01# sh ip nat translation#动态转换只有才完成时才会显示。

**实验分析：**

NAT允许一个整体机构以一个公用IP（Internet Protocol）地址出现在Internet上，是一种把内部私有网络地址（IP地址）翻译成合法网络IP地址的技术，因此，NAT的一大作用是解决公网地址不足的问题。

NAT有三种类型：静态NAT(Static NAT)、动态地址NAT(Pooled NAT)、网络地址端口转换NAPT（Port-Level NAT）。

其中，网络地址端口转换NAPT（Network Address Port Translation）则是把内部地址映射到外部网络的一个IP地址的不同端口上。它可以将中小型的网络隐藏在一个合法的IP地址后面。NAPT与 动态地址NAT不同，它将内部连接映射到外部网络中的一个单独的IP地址上，同时在该地址上加上一个由NAT设备选定的端口号。

NAPT是使用最普遍的一种转换方式，在HomeGW中也主要使用该方式。它又包含两种转换方式：SNAT和DNAT。在本实验中，为了实现方便，使用的是动态地址NAT类型的NAT。

本实验需要了解NAT地址转换的基本原理，了解其是如何解决公网地址不足的问题，即当私有网主机和公共网主机通信的IP包经过NAT网关时，将IP包中的源IP或目的IP在私有IP和NAT的公共IP之间进行转换。这个实验算是较为进阶的实验，加深我们对NAT地址转换方法的认识，需要通过实验对NAT地址转换的原理有一定的掌握。

# **邮件收发实验**

**邮件收发实验**

**实验目的**

电子邮件是目前人们工作和个人进行沟通交流的主要渠道,已成为人们生活和工作的重要组成部分,也是互联网应用最广的服务之一。了解电子邮件原理,可以更有效地开展邮件应用。实验通过安装邮件服务器,创建一个邮件应用环境,然后使用Telnet 工具,使用命令行操作进行简单邮件的收发。

（1）了解应用层数据规范和交互命令基本组成。

（2）了解电子邮件运行的基本模型。

（3）了解邮件服务器基本安装和管理步骤

（4）了解邮件发送交互方式，熟悉SMTP协议消息。

（5）了解邮件接收交互方式，熟悉POP协议消息。

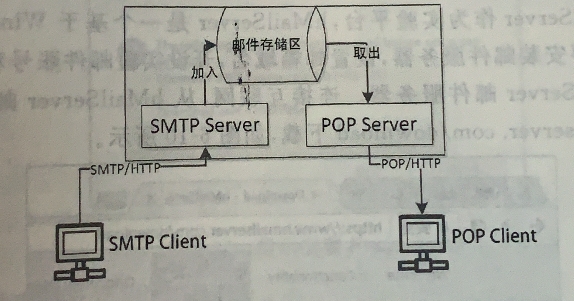
**实验原理**

电子邮件具音传统邮箱相同特性，任何人不需要得到收件人许可,就可以通过邮箱软件往某个邮箱中以八后I什,但以后件只有所有者才能处理。电子邮件系统较其他应用系统复杂,它具有两类客尸,奇信人各尸和收信人客户,却具有共同的邮箱。电子邮箱采用域名方式定义，电子邮箱格式如下:

mailbox@computer

其中,mailbox表示电子邮箱名,标识收信用户; computer表示电子邮箱所在主机域名,标识邮箱所在的主机。

邮件服务为寄信人和收信人制订了不同的应用协议。简单邮件传输协议(Simple MailTransfer Protoco, SMTP),用于发送邮件;邮局协议(Post Office Protocol,POP),用于接收邮件。两个协议配合使用才构成完整的电子邮件系统,电子邮件运行模型如图下图所示。



邮件存储区担当个人邮箱,是一个被动存储区域,通常用数据文件来担当,用来存放邮件，每个收件人均拥有独立的邮件存储区域。发件人使用SMTP客户机把电子邮件发送给SMTP服务器,SMTP服务器的主要功能是把接收到的邮件存放到个人邮箱邮件存储区;收件人使用POP 客户机从 POP服务器接收邮件,POP服务器的主要功能是对个人邮箱邮件存储区进行管理,只有邮箱所有者才被允许管理,必须通过身份认证。SMTP 协议和POP协议均有专门客户端软件,比如 Outlook软件就包含了SMTP客户机和POP 客户机SMTP客户机和POP 客户机需要安装设置,操作较为繁复,目前主流是基于 HTTP协议的邮件客户机,一个特殊网页担当SMTP客户机和POP客户机处理邮件收发,使用浏览器访问SMTP服务器和POP服务器。

1. **SMTP协议**

SMTP 协议作为一个应用层协议,使用TCP 协议作为传输层协议,以保证邮件发送的可靠性,SMTP 服务器缺省端口号是25。STMP 只负责邮件发送,SMTP 客户机同SMTP 服务器建立一条TCP 连接会话后,通过一系列SMTP交互消息,可以将邮件内容发送过去。

1. **POP协议**

POP协议,也采用TCP 协议作为传输层通信协议，PUF h版力06I走信件 BP1 10·该h议允许邮箱用户远程存取邮箱,完全仿照手工取信力，从杰T端自,才能对邮f尸机向POP服务器建立一条TCP 连接会话后,通过同一系列PUP 交土nl, I即相中的电作实施管理,查看邮件或者删除邮件。

SMTP协议和POP协议就像其他应用层协议一样,其眇以捐恩御不I于N恰式,非常客易理解。

邮件服务器是邮件协议的产品形态,均包含了 SMTP服务器和POP服务器。操作系按自身一般不包含邮件服务器,需要额外安装。

**实验环境**

使用一台计算机作为实验运行平台,实验环境主要由软件环境担当。

1. 启动包含SMTP 和 POP 服务器的邮件服务器,并开设实验邮件账号供实验使用;
2. 创建networklab.com 作为邮箱服务器伪域名;
3. 使用Telnet 工具软件作为客户机,分别访问SMTP服务器和 POP 服务器,进行邮件的发送和接收。
4. **安装和设置邮件服务器**

本文选用hMailServer作为实验平台,hMailServer 是一个基于Windows的开源邮件服务器。实验环境需要安装邮件服务器,设置邮箱域名,开设实验邮件账号和口令。

（1）下载hMailServer邮件服务器。连接互联网,从、 hMailServer 邮件服务器官网网址https:// [www.hmailserver.com/download下载。](http://www.hmailserver.com/download下载。)

点击“Download hMailServer 5.6.6-Build 2383”,将下载得到安装文件“hMailServer5.6.6-B2383.exe”。

（2）安装hMailServer邮件服务器。

1. 运行安装文件“hMailServer-5.6.4-B2283,exe”，点击“Next”→“I accept theagreement”“Next"→安装目录,缺省“C:\Program Files(X86)\hMailServer"“Next"→邮件组件,缺省Full installation→“Next"数据库类型,缺省内置数据库→“Next”→快捷方式，缺省hMailServer-→"Next”。

服务器管理员口令=password,确认口令=password→“Next”一"Install”"Finish”。

1. 安装Framework。hMailServer需要Framework3.5作为运行环境。

“是”->“下载并安装此功能”

1. hMailServer邮件服务器设置
2. 启动邮件服务器管理平台。启动hMailServer管理器。

左击“开始”->所有程序->hMailServer/hMailServer Aministrator->”connect”

输入邮箱服务器管理员口令：password->”OK”

1. 设置邮箱服务器域名为networklab.com。

选择Domain->”Add”->Domain=networklab.com->”Save”

1. 创建实验邮件账号，创建邮件账户student1@networklab.com

选择Domain/networklab.com/Accouts->”Add”->Address=stuent1,Password=password,Maxiam size(MB)=1->”Save”->”Exit”

1. 检查是否设置伪域名。必须设置networklab.com为127.0.0.1。

**2.环境设置**

（1）设置networklab.com伪域名

（2）操作系统启用Telnet工具软件。

**实验内容**

**1.SMTP邮件发送实验**

(1)启动邮件服务器服务,然后利用Telnet作为客户机,连接TCP端口25,访问 SMTP服务器,输人一系列SMTP请求消息,目的是向studentl 邮件账号发送一封不带附件的电子

邮件。

(2) SMTP消息。SMTP消息分请求消息和响应消息。

1 SMTP请求消息,由邮件客户发送给邮件服务器,采用字符串格式。HELLO:<SENDER>,标识发送人身份;

MAIL FROM:<MAILBOX>,指明发件人邮箱地址,MAILBOX是发件人邮箱名;PCPT TO:MAILBOX>,指明收件人邮箱地址,MAILBOX是收件人邮箱名;

DATA要求发送信件主体,以<cr><LF>.<cr<LF>结束;

QUIT,结束TCP连接退出。

2 SMTP响应消息,由邮件服务器返回给邮件客户,使用数字表示:250,回复客户请求消息成功接收;

221,确认结束连接;

220,正常。

**2. POP邮件接收实验**

(1)启动邮件服务器服务﹐然后利用Telnet作为客户机,连接110端口访问POP服务器，输入一系列POP请求消息,目的是从studentl邮箱中获取已接收的邮件,并查看邮箱中的电

子邮件内容。

(2) POP 消息。主要POP请求消息。

USER<MAILBOX>,指明邮件接收者的邮相账了,MALDUA代表邯箱名;

PASS (PASSWORD>,指明邮件按收有的即相0相账号口4LIST,列出邮箱内邮件编号列表;

RETR<NO>,显示指定邮件编号内容,NO代表邮件编号;QUIT,结束POP服务器连接退出。

**实验步骤**

【实验一:SMTP邮件发送实验】

启动hMailServer服务。打开系统服务管理，开炬尔现只他官理工具”“管理工具”→“服务”,参见1.4.1节,启动操作如图6-17所示。

如果hMailServer还没启动,右击hMailServer→“启动”。

使用Telnet 发送邮件。

1.进入Telnet命令。打开命令行窗口。

输入“telnet”,进入 Telnet 交互模式。

2.连接SMTP服务器，端口号是25。

输入“open networklab. com 25”。

3.向邮箱studentl@networklab.com发送邮件消息。使用SMTP消息向邮箱studentl@networklab.com 发送邮件,具体消息含义参见 SMTP消息介绍,教字开头是服务器的响应状态码。

标识发件人身份,HELO networklab.com

指定发件人邮箱,MAIL FROM:jinweizu@163.com

指定收件人邮箱,RCPT TO:student1@networklab.com

输入信件主体:

DATA

Subject:test SMTP

结束信件输入:<cr<LF>.Kcr<LF>,使用<回车>.<回车>

结束连接:QUIT。

【实验二:POP邮件接收实验】

1. 使用 Telnet 接收邮件。
2. 连接POP服务器,端口号是110。

open networklab.com 110

1. 接收实验一发送的邮件。通过发送POP消息,接收实验一发送的邮件,具体消息含叉参见 POP 消息介绍,数字开头是服务器的响应状态码。

输入邮件账号和口令:

USER [studentl@networklab.com](mailto:studentl@networklab.com)

PASS password

查看当前信件列表：LIST

阅读编号为1的信件内容：RETR1，可以看到实验一发送的邮件内容，邮件成功接收。

结束链接：QUIT

**实验分析**

电子邮件已经成为目前人们工作和个人进行沟通交流的一种主要手段。本实验主要了解了电子邮件原理让我们更加了解邮件应用。实验主要通过安装邮件服务器，创建一个邮件应用环境，然后使用Telnet工具进行简单的邮件收发。

本实验主要采用了两种协议消息，分别是SMTP协议消息和POP协议消息。

SMTP称为简单Mail传输协议（Simple Mail Transfer Protocal）,目标bai是向用户提供高效、可靠的邮件传输。SMTP的一个重要特点是它能够在传送中接力传送邮件，即邮 件可以通过不同网络上的主机接力式传送。工作在两种情况下：一是电子邮件从客户机传输到服务器；二是从某一个服务器传输到另一个服务器。 .SMTP是个请求/响应协议，它监听25号端口，用于接收用户的Mail请求，并与远端Mail服务器建立SMTP连接。

POP3仍采用Client/Server工作模式，Client被称为客户端，一般我们日常使用电脑都是作为客户端，而Server（服务器）则是[网管](https://baike.baidu.com/item/%E7%BD%91%E7%AE%A1" \t "https://baike.baidu.com/item/pop%E5%8D%8F%E8%AE%AE/_blank)人员进行管理的。举个形象的例子，Server（服务器）是许多小信箱的集合，就像我们所居住楼房的信箱结构，而客户端就好比是一个人拿着钥匙去信箱开锁取信一样的道理。

# **IP数据包分析实验**

**IP数据包分析实验**

**实验目的**

IP协议是 TCP/IP模型的核心协议,IP数据包是网际网上进行传输的网络数据包,所有的传输与处埋都围统P数据包展开,因此,了解IP数据包格式,理解每一个数据域含义,对于网际网的传输机制理解至天重要。本实验将借助Tcpdump工具,捕获和分析IP数据包。

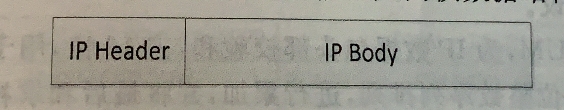
（1）了解IP数据包格式以及各个数据域含义。

（2）了解IP数据包传输原理。

**实验原理**

**IP数据包**

IP网络数据报实行无连接传输,十分类似以太网帧传输方式,只要有目标IP地址,就可以直接发送。IP数据包(IP datagram)结构也类似以太网帧数据结构,如下图所示。



前面是个头部,后跟数据体。IP数据包头部,其功能同以太网帧十分类似,用于传输控制,为标识接发双方,包含了源IP地址和目的IP地址,还设计了其他多个数据域,以帮助实IP数据包的虚拟传输;一般而言,IP数据包头部尺寸固定,常见是20个字节。IP数据包数据体区,用于存放传输层数据,常见如TCP段和 UDP用户数据包,数据区尺寸完全由封装层决定,大小并不固定,最小为一个字节,最大可以达到64K字节。

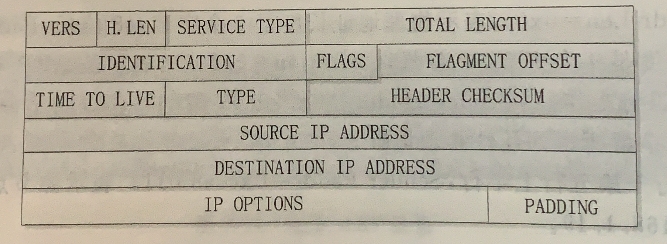
IP数据包在传输过程中可能遭遇分段。IP数据包分段是指IP数据包实施进一步分割。物理网络最大传输单元(Maximum Transmission Unit,MTU),是指物理帧最大传输尺寸。

显然,封装时IP数据包必须小于或等于MTU,否则IP包将无法封装到物理帧中。在由异构物理网络组成的网际网上,当一个路由器连接具有不同MTU的网络时,如果后续网络MTU

小于前面网络MTU,路由器就有可能面临由于受到MTU限制而不能完整地向下一个网络转发IP数据包,路由器就必须将IP数据包分割成更小尺寸的IP分段,才能继续传输。鉴于

篇幅,IP分段不再深入讨论。

IP数据包头部整合了标准IP数据包和IP数据包分段处理要求。IP数据包头部格式如下图所示，格式每行占32位，共4个字节，头部的各个域的尺寸都是固定的。



其中,VERS,占4位,标识版本号,当前IP版本号是4,即值为0x4。

H.LEN,标识IP数据包头部长度,占4位,头部尺寸可以不同,以四字节作为一个计量单位;一般值为0x5,即5行,计20字节。

SERVICE TYPE,标识服务类型,占8位。

TOTAL LENGTH,标识IP数据包总长度,占16位,以字节为计量单位,包括IP数据包头部和IP数据包数据区。

IDENTIFICATION,标识IP分段,数据包标识,占16位。FLAGS,标识段处理标记,占3位,

FLAGMENT OFFSET,标识段偏移,占13位。

TIME TO LIVE,标识生存时间,占8位,主要用来消除迷失的IP数据包。发送方为每个IP数据包设置一个生存时间初值,8数位,在0和255两个数之间选择一个中间值;每次经过一个路由器时,IP数据包生存时间就必须减去1,如果生存时间降到0,该IP数据包将被丢弃,并将发送一个错误消息给源主机,可以有效消除迷路的IP数据包。

TYPE,标识被封装的上层协议类型,占8位,其中,0x6表示 TCP协议,0x1l表示UDP协议,Ox01表示ICMP 协议。

HEADER CHECKSUM,为IP数据包头部校验和,占16位,用于验证头部数据是否发生错误,对头部数据作为16位整数序列排列,进行累加,并将最后和求补。

SOURCEIP ADDRESS ,标识源主机(发送方)IP地址,占32位。

DESTINATION IP ADDRESS,标识目的主机(接收者)IP地址,占32位。IP OPTIONS,可选项。

PADDING,填充项,当实际可选项长度达不到32位,后面就填入0,保证头部长度是32的倍数。

**实验内容**

(1)一个终端,启用Tcpdump工具软件,并设置相应过滤条件,从指定网卡捕获IP数据包,并进行分析。

(2)另一个终端触发产生IP数据包。输入 ping命令,向 Windows节点发出ICMP 消息,该消息封装在IP数据包中进行传输。

**实验步骤**

1. 启动捕获IP数据包监听。
2. 创建终端1,进入捕获IP数据包监听状态

输入捕获IP数据包监听命令:sudo tecpdump-ex -i enp0s3 ip

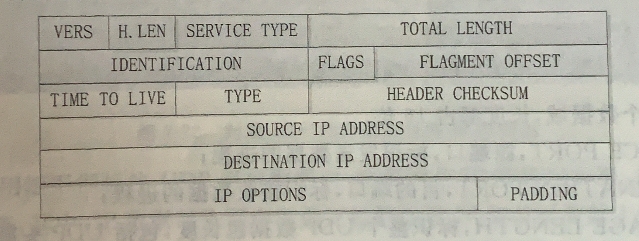
口令:123456789,处于监听状态。

（2）触发产生IP 数据包。创建终端2,向 Windows 节点发出ICMP 消息

输入“ping 192.168.1.254”。

(3)分析捕获IP数据包。终端1,捕获了一个IP 数据包。

IP数据包头部格式,如下图所示。



分成五行,每行对应两组数字,共计20个字节,按照IP数据包格式对IP数据包进行分析。图7-17中1个数字代表半个字节,每组8个数字刚好对应IP数据包格式中一行,特意用矩形框分隔,下面按照格式中行顺序来分析数据含义。

第一行, version=Ox4,标识IP版本号,表示第四版IP;h. len=0x5,表示IP头部长度为5行,每行长32位,总计20字节 ;Service Type=0x00,空缺;Total Length=0x54,表示IP数据

包总长度,当前为84个字节。

第二行,分段数据,不讨论。

第三行,TTL=0x80,表示生存时间,折合成十进制为128;type=Ox01,表示封装的上层协议,1表示 ICMP 协议;Header checksum=Ox4e99,表示IP数据包头部的校验和补码。

第四行，源地址=0xc0a8 01fe，表示源节点IP地址，折合成十进制是192.168.1.254.

第五行，目标地址=0xc0a8 0113，表示目标节点IP地址，折合成十进制是192.168.1.19

**实验分析**

本实验主要了解了IP数据包这一样一个在TCP/IP协议中的核心结构，所有的传输和处理都围绕IP数据报展开，因此了解IP数据包格式与每一部分的含义是十分重要的。

IP协议控制传输的协议单元称为IP数据报（IP Datagram，IP数据报、IP包或IP分组）。IP协议屏蔽了下层各种物理子网的差异，能够向上层提供统一格式的IP数据报。lP数据报采用数据报分组传输的方式，提供的服务是无连接方式。IP数据报的格式能够说明lP协议具有什么功能。IPv4数据报由报头和数据两部分组成，其中，数据是高层需要传输的数据，报头是为了正确传输高层数据而增加的控制信息。报头的前一部分长度固定，共20字节，是所有IP数据报必须具有。在首部固定部分的后面是可选字段，长度可变。

IP数据包在很多地方都起到重要作用，因此需要完全掌握IP数据报格式和内容。

# **UDP用户数据报分析实验**

**UDP用户数据报分析实验**

**实验目的**

UDP 协议是非常重要的传输层协议,人们常用的即时工具软件）是使用UDP作为传输层协议,UDP 用户数据包格式非常简单,了解 UDP用户数据包有利于帘助理解传输层作用,

本实验将借助 Tcpdump,捕获和分析UDP用户数据包。

（1）理解传输层作用。

（2）了解 UDP用户数据包结构。

**实验原理**

**UDP用户数据包**

UDP 协议层使用的数据包称为用户数据报。UDP为应用层提供了面向消息的接口,但UDP不提供分割机制,只能将整个应用数据作为UDP用户数据包负载,封装到单个 UDP用户数据包中。UDP 用户数据包需要封装到IP数据包中才能进行传输,因此应用程序的数据尺寸受到极大的限制,不能超过IP数据包尺寸限制。像视频和音频应用的 UDP 大数据应用就需要应用层实施分割控制。

UDP用户数据包采用无连接通信,因此 UDP用户数据包头部数据结构十分类似IP数据包格式:前面是个头部,用于标识接发双方应用程序,通过设置16位端口号来作为应用程序标识;UDP数据包数据区,直接存放用户数据,数据区尺寸由应用层决定,大小并不固定。数据区尺寸可变,最小为一个字节,最大为64K字节。UDP用户数据包头部格式如下图所示。

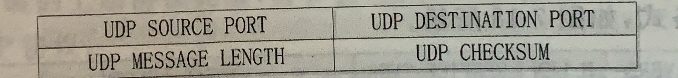


图7-19

头部包含四个数据域,长度都占16位。

UDP SOURCE PORT,源端口,标识发送数据的进程;

UDP DESTINATION PORT,目的端口,标识接收数据的进程;

UDP MESSAGE LENGTH,标识整个UDP数据包长度,包括UDP头部和数据;

UDP CHECKSUM,校验和,用于检验传输后的信息正确性,但校验范围超出了UDP 用户数据包头部范围,实际还包括了IP数据包头部地址部分。

**实验内容**

（1）一个终端,启用Tcpdump工具软件,并设置相应过滤条件,从指定网卡捕获UDP用户数据包,对 UDP 用户数据包进行分析。

（2）另一个终端,使用echo命令,发送一串字符串重定向 Windows 节点一个UDP瑞H

**实验步骤**

（1）创进终端1，启动抓包监听，以便在网络上铺货UDP用户数据报，“应用程序”->“附件”->“终端”。

监听UDP端口5555 ;sudo tcpdump-xp-i enp0s3 udp and port 5555口令:123456789,处于监听状态。

(2)产生 UDP 用尸数据包。创建终端2,使用 Linux命令方式向 Windows节点[DP5555端口及达子付串,以触发UDP 用户数据包

输人“echo 'networklab')/dev/udp/192.168.1.254/5555”,将字符串重定向到 Windows主机的UDP5555端口。

(3)分析捕获 UDP数据包。终端1,捕获了封装UDP用户数据包的IP数据包图7-22中用矩形框标注UDP用户数据包,框前面是IP数据包头部,共20个字节,具体含义参见7.4节,其中,IP数据包头部中的第三组数标识IP包被封装的上层协议,对应IP数

据包头部格式第三行前半行，TTL=Ox40,表示生存时间,折算成十进制是64;type=0xl1,标识封装类型,折算成十进制17,表示上层协议是UDP协议。

图7-19中 UDP 用户数据包头部分成两行,其余是UDP用户数据包数据体﹐即应用程序数据。图7-22中1个数字代表半个字节,每组8个数字刚好对应UDP用户数据包格式中一行,特意用矩形框分隔,下面按照格式中行顺序来分析数据含义。

第一行,UDP SOURCE PORT=0xa109,表示源端口号;UDP DESTINATION PORT=OX15b3,表示目标端口号,折算成十进制为5555。

第二行,UDP MESSAGE LENGTH=0x13,表示 UDP用户数据包长度,19个字节长,8个字节头部,11个字节数据;UDP CHECKSUM=0x8486,表示 UDP 用户数据包校验和。

第三行＋第四行＋第五行,代表UDP 数据体,0xbe65 7477 6f72 6b6c 6162,按照ASCII编码就是发送的字符串”networklab”;0x0a，则代表ASCII编码换行，刚好是11个字节。

**实验分析**

UDP协议是一个端到端的协议，也是常用的即使工具软件的传输层协议。所以需要了解UDP用户数据包帮助理解传输层作用。

UDP协议与[TCP](https://baike.baidu.com/item/TCP" \t "https://baike.baidu.com/item/UDP/_blank)协议一样用于处理数据包，在[OSI](https://baike.baidu.com/item/OSI" \t "https://baike.baidu.com/item/UDP/_blank)模型中，两者都位于[传输层](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82/4329536" \t "https://baike.baidu.com/item/UDP/_blank)，处于IP协议的上一层。UDP有不提供数据包分组、组装和不能对数据包进行排序的缺点，也就是说，当报文发送之后，是无法得知其是否安全完整到达的。UDP用来支持那些需要在[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA/140338" \t "https://baike.baidu.com/item/UDP/_blank)之间传输数据的网络应用。包括[网络视频会议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E8%A7%86%E9%A2%91%E4%BC%9A%E8%AE%AE" \t "https://baike.baidu.com/item/UDP/_blank)系统在内的众多的客户/服务器模式的网络应用都需要使用UDP协议。UDP协议从问世至今已经被使用了很多年，虽然其最初的光彩已经被一些类似协议所掩盖，但即使在今天UDP仍然不失为一项非常实用和可行的网络传输层协议。

许多应用只支持UDP，如：多媒体数据流，不产生任何额外的数据，即使知道有破坏的包也不进行重发。当强调传输性能而不是传输的完整性时，如：音频和多媒体应用，UDP是最好的选择。在数据传输时间很短，以至于此前的连接过程成为整个流量主体的情况下，UDP也是一个好的选择。

# **自主虚拟实验团队项目**

## 24.1 个人文献阅读

在寻找论文进行NS3小组实验的过程中，我找到了一篇与移动自组织网络相关的论文。也就是Ad-Hoc的相关内容。自20世纪90年代中期以来，随着笔记本电脑和WiFi无线网络的发展，移动自组织网络（已经成为一个热门的研究课题。MANET是一个由无线连接的移动设备组成的无需自配置基础设施的网络，其中每个设备可以在任何方向上独立移动。它具有动态拓扑结构、带宽受限、可变容量链路、能量受限操作和有限的物理安全性。路由协议在任何网络中都扮演着重要的角色。它指定路由如何相互通信，传播信息以选择网络上任意两个节点之间的路由。Ad-hoc的网络种类繁多可分为三种：平面层次结构、主动式和反应路由式协议、混合式路由协议。

论文第二节介绍了路由协议。第三、四、五和六节讨论了OLSR、AODV、DSDV和DSR路由协议的概述。第七节给出了仿真结果。第八节描述了绩效指标，第九节涉及基于上述指标的绩效分析。最后第十节对全文进行了总结。

AODV路由协议是一种反应式协议，意味着在需要时建立路由。它基于路由发现和路由维护的按需机制，加上逐跳路由和序列号的使用。路由表由下一跳到目的地的信息和从目的地接收到的序列号组成。该协议支持两个阶段：路由发现、路由维护和数据转发。路由发现是通过用请求的目的地序列号将RREQ消息广播给它的邻居来完成的，这可以防止循环问题。邻居用RREP包应答，而相应的路由将RREQ包转发给他们的邻居。当注意到路由中断时，节点向邻居发送RERR消息。它定期使用HELLO消息通知邻居到主机的链接处于活动状态。在接收HELLO消息时，节点更新路由表中节点信息的生存期。AODV协议是一种平面路由协议，不需要任何中央管理系统来处理路由过程。它还降低了控制流量消息开销，但代价是在查找新路由时增加了延迟。

  DSDV路由协议[3]是一种基于Bellman-Ford路由算法的主动式路由协议，它提供了两个节点之间最短路径的解决方案。此外，它还引入了新的特征，即为整个网络的每个路由表条目提供序列号，以避免形成路由环。路由表在整个网络中定期更新，以保持表中的一致性。为了保持网络的最新视图，表是按固定的时间间隔交换的。为了减少路由信息包在广播过程中携带的信息量，定义了两种类型的消息。

一个包含所有可用路由信息的类型称为完全转储，其他类型（即增量转储）包含自上次完全转储以来已更改的信息。完全转储需要多个网络协议数据单元（NPDU），而增量转储只需要一个就可以容纳所有信息。当从另一个节点接收信息包时，节点将序列号与条目的可用序列号进行比较，如果序列号较大或较小，则用新的序列号更新条目。如果到达的信息具有相同的序列号，则需要输入度量。

在该协议中，由于断开的链路，在高移动性期间更新导致高控制开销。另一个缺点是节点必须等待由同一目的地节点发起的表更新消息，以便获得关于特定目的地节点的信息。

DSR是一种基于源路由概念的反应式协议，它要求每个包携带从源到目的地的完整地址（路由中的每个跃点）。它基于路由发现和路由维护的按需机制。DSR协议的一个优点是节点可以在其路由缓存中存储多条路由。源节点可以在启动路由发现之前检查其路由缓存中的有效路由，如果找到有效路由，则不需要进行路由发现。另一方面，如果节点没有这样的路由，则它通过广播RREQ分组来发起路由发现。RREQ包包含目的地地址、源地址、路由记录字段和唯一标识号。一旦RREQ到达目的地或知道到目的地的路由的节点，它就用RREP以及RREQ收集的路由的反向响应。通过主动监视确认或被动地在混杂模式下运行来检测失败的链路，从而偷听到相邻节点转发的数据包。失败的链路用RERR包通知源节点。源节点可以使用到目的节点的其他已知路由，或者再次启动路由发现过程以查找到目的节点的新路由。另一点需要注意的是，它不需要hello消息交换，因此节点可以进入sleep node以节省电力。同时也节省了相当大的网络带宽。

通过模拟以上几种路由协议进行发包，作者比较了不同协议在数据包投递率、端对端时延、丢包率、抖动呃吞吐量不同的条件下进行了性能比较，AODV在吞吐量上表现优秀，而OSLR在其他情况下表现更好。

由于这篇论文相对来说比较定性，缺少一定的定量分析，我们最后没有采用这篇论文的实验，而是选用了另一位同学的BBR’相关论文进行了NS3的模拟实验。

## 24.2 团队实验报告

**基于NS-3 的TCP BBR拥塞控制算法的实现**

**[实验背景]**

传统TCP拥塞控制算法都是基于丢包的算法，例如收包加法增，丢包乘法减，但是基于丢包的拥塞控制算法存在缺陷：

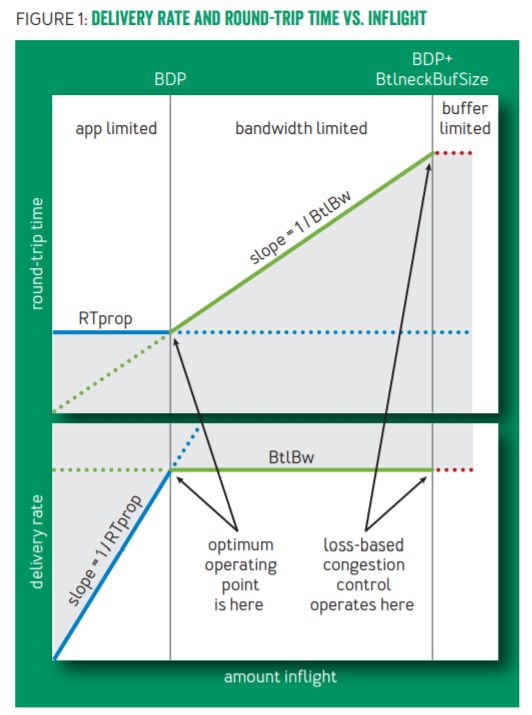
1. 无法区分拥塞导致的丢包和错误丢包

基于丢包的拥塞控制算法把数据包的丢失解释为网络发生了拥塞，而认为假定链路错误造成的分组丢失几率太低而可忽略不计。然而在高速网络中，当数据传输速率比较高时，链路错误是不能忽略的，无线网络中链路的误码率则更高，因此，将分组丢失认为是拥塞所引起从而降低一半的速率，将极大浪费网络资源。

1. 引起缓冲区膨胀

网络中设有用于吸收网络中的流量波动的缓冲区，在连接的开始阶段，基于丢包的拥塞控制方法倾向于填满缓冲区。当瓶颈链路的缓冲区很大时，需要很长时间才能将缓冲区中的数据包排空，造成很大的网络延时，这种情况称之为缓冲区膨胀。在一个先进先出队列管理方式的网络中，过大的buffer以及过长的等待队列，在某些情况下不仅不能提高系统的吞吐量甚至可能导致系统的吞吐量近乎于零。并且当所有缓冲区都被填满时，会出现丢包。

1. 无法达到理论的时延、带宽最优解



图表 1 TCP 投递率和往返时延图

当网络中数据包不多，还没有填满瓶颈链路的管道时，随着投递率的增加，往返时延（RTT）不发生变化。当数据包刚好填满管道，达到网络工作的最优点（满足最大带宽BtlBw和最小时延RTprop），定义带宽时延积BDP=BtlBw × RTprop，则在最优点网络中的数据包数量等于BDP。继续增加网络中的数据包，超出BDP的数据包会占用buffer，达到瓶颈带宽的网络的投递率不再发射变化，往返时延（RTT）会增加。继续增加数据包，buffer会被填满从而发生丢包。所以在BDP线的右侧，网络拥塞持续作用。基于丢包的拥塞控制算法工作在bandwidth-limited区域的右侧边界，将瓶颈链路管道填满后继续填充buffer，直到buffer填满发生丢包，拥塞控制算法发现丢包，将发送窗口减半后再线性增加，会使得系统稳定在右界，虽然达到了最大带宽，却以巨大的延迟为代价。

谷歌在2016年提出的基于拥塞的BBR拥塞控制算法，通过观察带宽变化和RTT变化来决定速率的变化，在保证最大瓶颈带宽（btlBw）的同时，获得最小延迟（RTprop），可以减少 TCP 连接的延迟、改善存在丢包的网络环境速度。

**[实验设备]**

Windows 10 现行版本客户端

VMware Workstation 16虚拟机

Linux 操作系统下Ubuntu 16.04 NS-3 3.33

**[实验内容]**

1. **算法介绍**

TCP BBR（Bottleneck Bandwidth and Round-trip propagation time）是由Google设计，于2016年发布的拥塞算法。以往大部分拥塞算法是基于丢包来作为降低传输速率的信号，而BBR则基于模型主动探测。

该算法使用网络最近出站数据分组当时的最大带宽和往返时间来创建网络的显式模型。数据包传输的每个累积或选择性确认用于生成记录在数据包传输过程和确认返回期间的时间内所传送数据量的采样率。该算法认为随着网络接口控制器逐渐进入千兆速度时，与缓冲膨胀相关的延迟相比丢包更应该被认为是识别拥塞的主要决定因素，所以基于延迟模型的拥塞控制算法会有更高的吞吐量和更低的延迟。BBR算法在实现上由5部分组成：

1. **即时速率的计算**

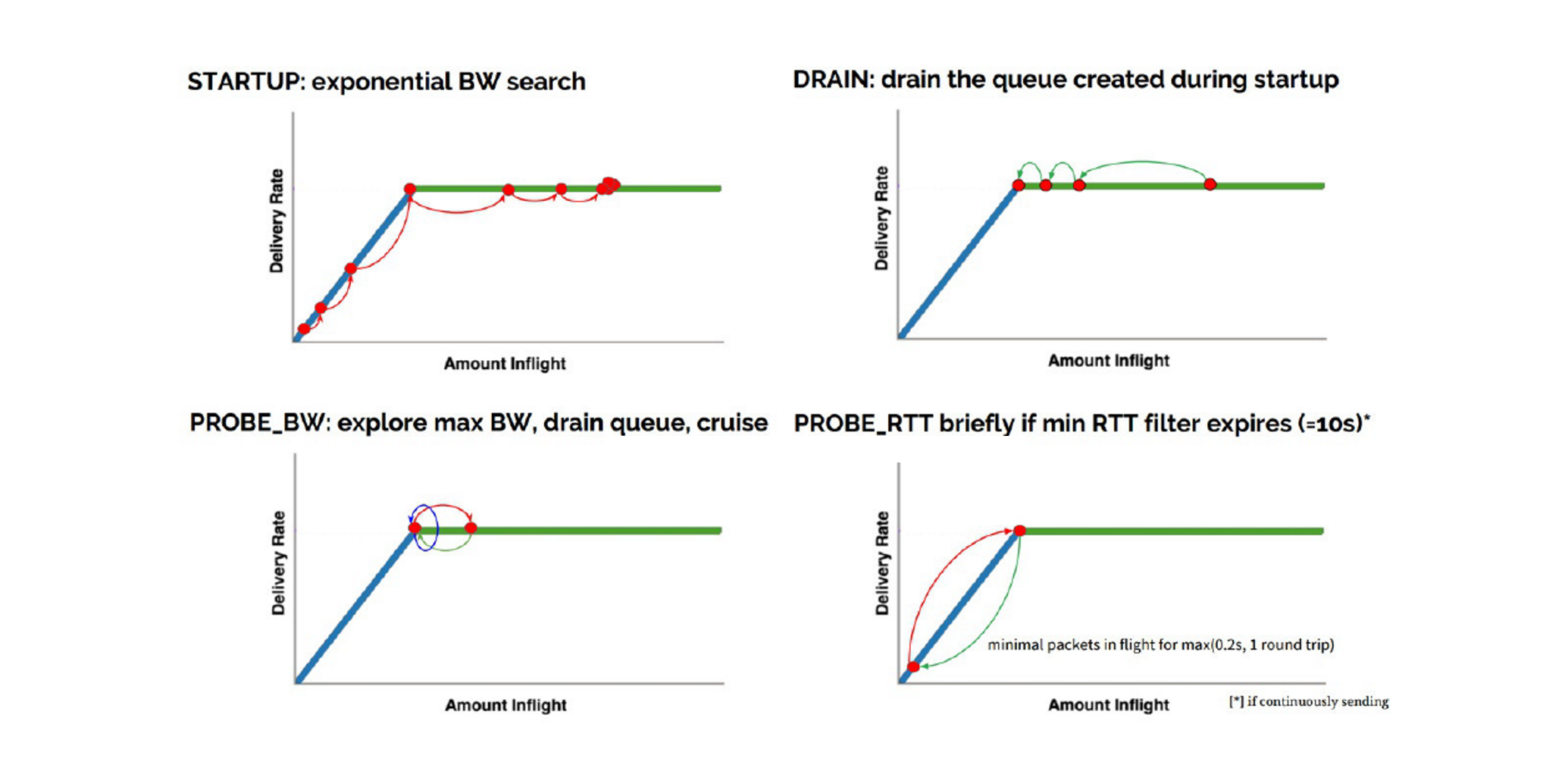
计算一个即时的带宽bw，该带宽是BBR一切计算的基准，BBR将会根据当前的即时带宽以及其所处的pipe状态来计算pacing rate以及cwnd(见下文)，后面我们会看到，这个即时带宽计算方法的突破式改进是BBR之所以简单且高效的根源。计算方案按照标量计算，不再关注数据的含义。在BBR运行过程中，系统会跟踪当前为止最大的即时带宽。

1. **RTT的跟踪**

BBR之所以可以获取非常高的带宽利用率，是因为它可以探测到带宽的最大值以及RTT的最小值，这样计算出来的BDP就是目前为止TCP管道的最大容量。BBR的目标就是达到这个最大的容量。这个目标最终驱动了cwnd的计算。在BBR运行过程中，系统会跟踪当前为止最小RTT。

1. **BBR pipe状态机的维持**

BBR算法根据互联网的拥塞行为有针对性地定义了4中状态，即STARTUP，DRAIN，PROBE\_BW，PROBE\_RTT。BBR通过对上述计算的即时带宽BW以及RTT的持续观察，在这4个状态之间自由切换，相比之前的所有拥塞控制算法，其革命性的改进在于BBR拥塞算法不再跟踪系统的TCP拥塞状态机，而旨在用统一的方式来应对pacing rate和cwnd的计算，不管当前TCP是处在Open状态还是处在Disorder状态，抑或已经在Recovery状态。



图表 2 BBR 状态图

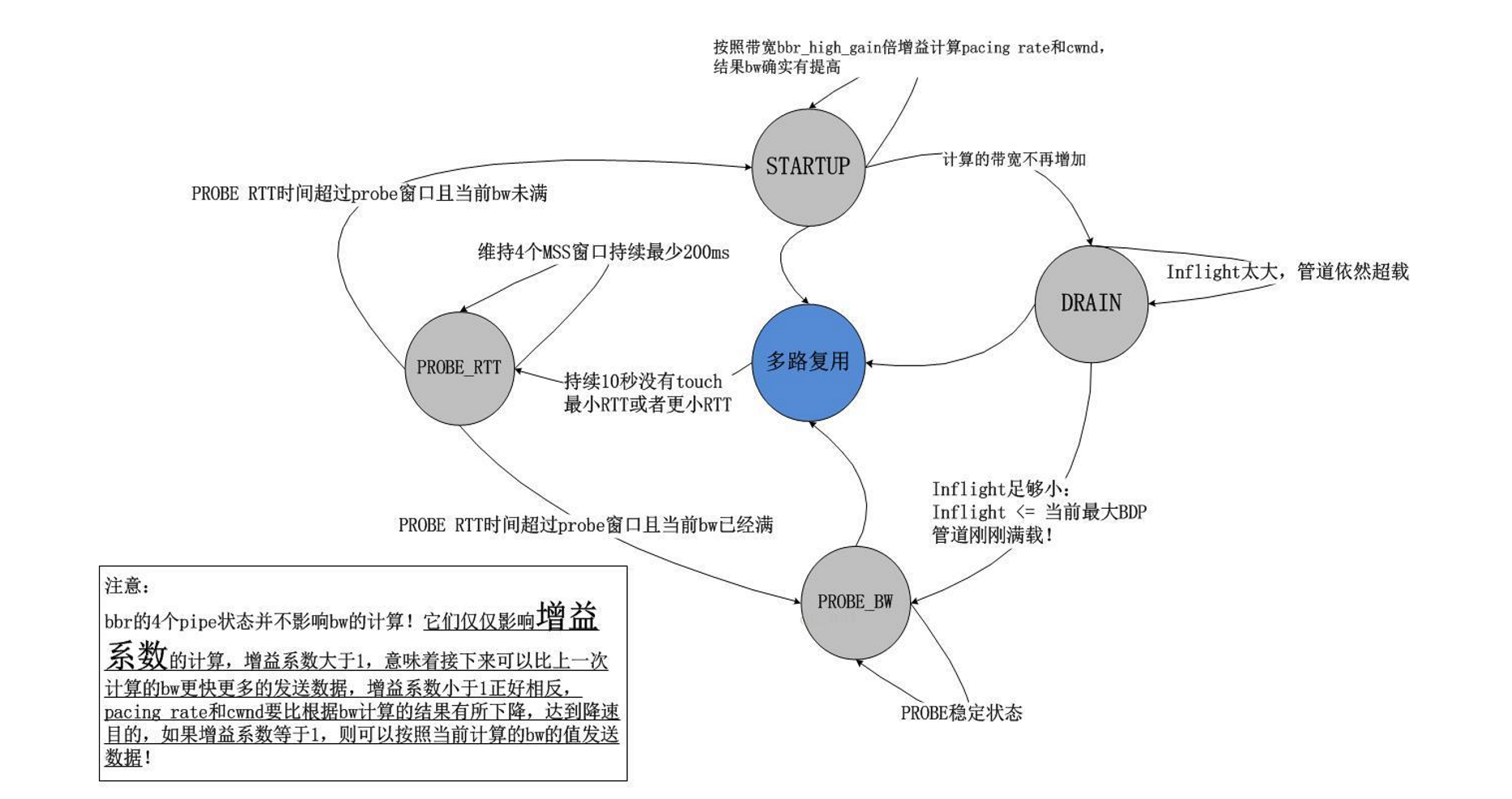
1. **结果输出-pacing rate和cwnd**

在传统意义上，cwnd是TCP拥塞控制算法的唯一输出，但是它仅仅规定了当前的TCP最多可以发送多少数据，它并没有规定怎么把这么多数据发出去，在Linux的实现中，Linux会把cwnd数据全部突发出去，而这往往会造成路由器的排队，在深队列的情况下，会测量出RTT剧烈地抖动。BBR在计算cwnd的同时，还计算了一个与之适配的pacing rate，该pacing rate规定cwnd指示的数据的数据包之间，以多大的时间间隔发送出去。

1. **其它外部机制的利用-fq，rack等**

BBR利用了许多已有的运行机制，通过对多方机制的整合，使BBR算法将整体重心放在数据计算之上。

1. **TCP BBR的状态机及分析**



图表 3 TCP BBR状态机

通过上述的状态机以及带宽计算方式，我们知道了BBR的工作方式：不断地基于当前带宽以及当前的增益系数计算pacing rate以及cwnd，以此2个结果作为拥塞控制算法的输出，在TCP连接的持续过程中，每收到一个ACK，都会计算即时的带宽，然后将结果反馈给BBR的pipe状态机，不断地调节增益系数,与TCP当前处于什么拥塞状态完全无关。

本次实验我们将通过构建简单的通信框架实践BBR算法，并通过参数和图表展示TCP BBR的优越之处。

1. **实验步骤**
   1. **搭建NS-3 3.33仿真工具并测试编译成功**
   2. **安装导入** TCP-CLIENT 模块

将对应模块添加至ns3.3/src/目录下，并执行以下语句:

1. $ ./ns3 configure --enable-examples --enable-tests
2. $ ./ns3 build
   1. **编写测试demo代码**

初始化组件及常量

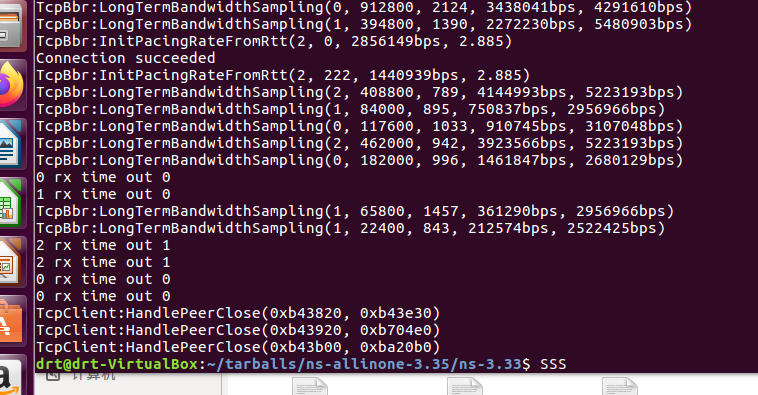
|  |
| --- |
| *//两个节点*      nodes.**Create** (2);  *// pointtopoint channel*      PointToPointHelper pointToPoint;  *//设置速率*      pointToPoint.**SetDeviceAttribute** ("DataRate", **DataRateValue**  (**DataRate** (bps)));  *//设置延迟*    pointToPoint.**SetChannelAttribute** ("Delay", **TimeValue** (**MilliSeconds** (msDelay)));      auto bufSize = **std**::**max**<uint32\_t> (DEFAULT\_PACKET\_SIZE, bps \* msQdelay / 8000);  *//包的个数*   int packets=bufSize/DEFAULT\_PACKET\_SIZE;  **NS\_LOG\_INFO**("buffer packet "<<packets);  *//传输队列*      pointToPoint.**SetQueue** ("ns3::DropTailQueue", "MaxSize", **StringValue** (**std**::**to\_string**(1)+"p"));  *// 用于管理网络设备*      NetDeviceContainer devices = pointToPoint.**Install** (nodes);  *//此帮助程序启用 pcap 和 ascii 跟踪与节点关联的 Internet 堆栈中的事件*      InternetStackHelper stack;      stack.**Install** (nodes);  *//流量控制的helper*      TrafficControlHelper pfifoHelper;  *//实现 FIFO（先进先出）策略的简单队列磁盘。*      uint16\_t handle = pfifoHelper.**SetRootQueueDisc** ("ns3::FifoQueueDisc", "MaxSize", **StringValue** (**std**::**to\_string**(packets)+"p"));  *//增加到网络队列上*      pfifoHelper.**AddInternalQueues** (handle, 1, "ns3::DropTailQueue", "MaxSize",**StringValue** (**std**::**to\_string**(packets)+"p"));  *//添加到服务上*      pfifoHelper.**Install**(devices);  *//Ipv4地址分配*      Ipv4AddressHelper address;  *////设置了一个ip网络号和子网掩码，意味着这个网络的ip地址从10.1.1.0~10.1.1.255.*  **std**::string nodeip="10.1.1.0";      address.**SetBase** (nodeip.**c\_str**(), "255.255.255.0");  *//实现ip地址的分配，devices中的device们，将按递增顺序获得ip地址和子网掩码，从10.1.1.1到10.1.1.2*      address.**Assign** (devices); |

创建Client并根据算法种类运行。

|  |
| --- |
| uint64\_t totalTxBytes = 40000\*1500;  *//构建了TCP的client并根据cc算法的种类来进行运输，终止时间一定，开始时间不同*      {          Ptr<TcpClient>  client= **CreateObject**<**TcpClient**> (totalTxBytes,**TcpClient**::E\_TRACE\_RTT|**TcpClient**::E\_TRACE\_INFLIGHT|**TcpClient**::E\_TRACE\_RATE);          h1->**AddApplication**(client);          client->**ConfigurePeer**(tcp\_sink\_addr);          client->**SetCongestionAlgo**(cc);          client->**SetStartTime** (**Seconds** (startTime));  *//停止时间在开头有定义*          client->**SetStopTime** (**Seconds** (simDuration));      }      {          Ptr<TcpClient>  client= **CreateObject**<**TcpClient**> (totalTxBytes,**TcpClient**::E\_TRACE\_RTT|**TcpClient**::E\_TRACE\_INFLIGHT|**TcpClient**::E\_TRACE\_RATE);          h1->**AddApplication**(client);          client->**ConfigurePeer**(tcp\_sink\_addr);          client->**SetCongestionAlgo**(cc);          client->**SetStartTime** (**Seconds** (startTime+20));          client->**SetStopTime** (**Seconds** (simDuration));      }      {          Ptr<TcpClient>  client= **CreateObject**<**TcpClient**> (totalTxBytes,**TcpClient**::E\_TRACE\_RTT|**TcpClient**::E\_TRACE\_INFLIGHT|**TcpClient**::E\_TRACE\_RATE);          h1->**AddApplication**(client);          client->**ConfigurePeer**(tcp\_sink\_addr);          client->**SetCongestionAlgo**(cc);          client->**SetStartTime** (**Seconds** (startTime+50));          client->**SetStopTime** (**Seconds** (simDuration));      }  *//模拟器操作*  **Simulator**::**Stop** (**Seconds** (simDuration+10.0));  **Simulator**::**Run** ();  **Simulator**::**Destroy** ();      return 0; |

* 1. **输出代码运行结果及数据构图**

根据运行demo，我们得到了一下运行截图：



图表 4 TCP BBR编写实例运行截图

其中，**TcpBbr:InitPacingRateFromRtt(2,222,1440939bps,2.88s)**(同类型log记录以本条为示例),其目的是为了实现输出初始化pacing rate.

模块运行根据为:

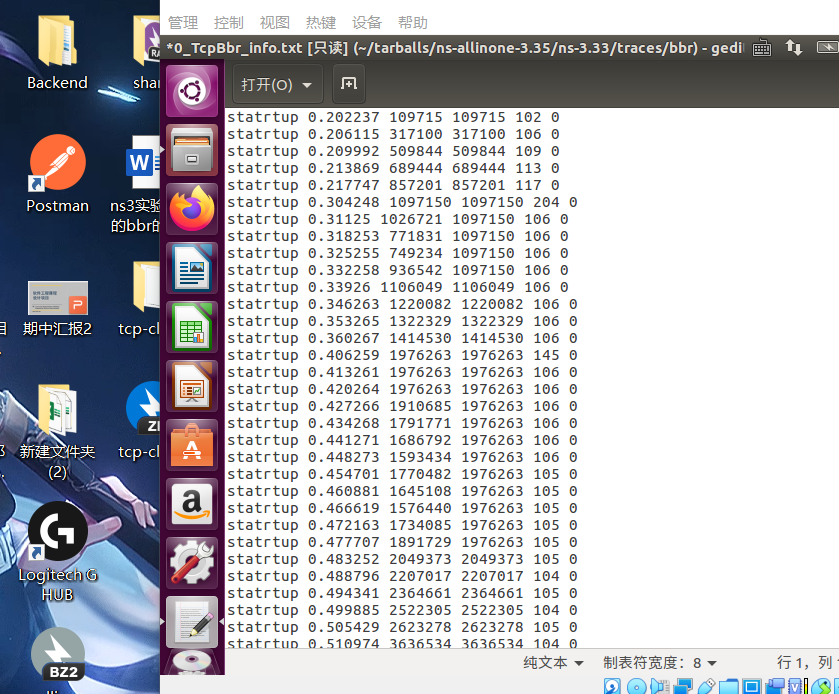
|  |
| --- |
| *//初始化pacing\_rate*  void **TcpBbr**::**InitPacingRateFromRtt**(**Ptr**<**TcpSocketState**> tcb){      uint32\_t mss=tcb->m\_segmentSize;      uint32\_t congestion\_window=tcb->m\_cWnd;      if(congestion\_window<tcb->m\_initialCWnd\*mss){          congestion\_window=tcb->m\_initialCWnd\*mss;      }      Time rtt=tcb->m\_lastRtt;      DataRate **bw**(1000000);      DataRate pacing\_rate=bw;      if(**Time**(0)==rtt){          pacing\_rate=**BbrBandwidthToPacingRate**(tcb,bw,m\_highGain);      }else{          m\_hasSeenRtt=1;          double bps=1.0\*congestion\_window\*8000/rtt.**GetMilliSeconds**();          bw=**DataRate**(bps);          pacing\_rate=**BbrBandwidthToPacingRate**(tcb,bw,m\_highGain);      }  #if (**TCP\_BBR\_DEGUG**)  **NS\_LOG\_FUNCTION**(m\_debug->**GetUuid**()<<rtt.**GetMilliSeconds**()<<pacing\_rate<<m\_highGain);  #endif      if(pacing\_rate>tcb->m\_maxPacingRate){          pacing\_rate=tcb->m\_maxPacingRate;      }      tcb->m\_pacingRate=pacing\_rate;  } |

**TcpBbr:LongTermBandwidthSampling(1,22400,843,212572bps,2522425bps)** (同类型log记录以本条为示例),代表: id为1的节点接受的数据，该段时间的采样间隔，与最大带宽。

模块运行根据为:

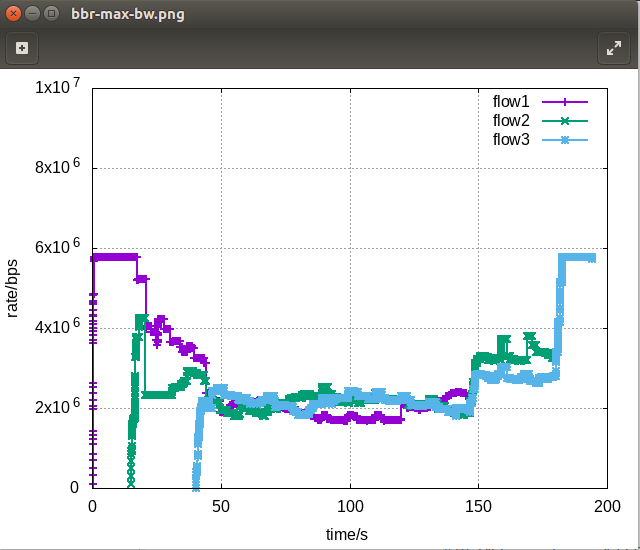
|  |
| --- |
| *//长期带宽采样*  void **TcpBbr**::**LongTermBandwidthSampling**(**Ptr**<**TcpSocketState**> tcb,const **TcpRateOps**::**TcpRateSample** &rs){      uint32\_t lost,delivered;      if(m\_ltUseBandwidth){          if(PROBE\_BW==m\_mode&&m\_roundStart&&(++m\_ltRttCount)>=bbr\_lt\_bw\_max\_rtts){  **ResetLongTermBandwidthSampling**();  **ResetProbeBandwidthMode**();}          return ;}  *//等待第一次丢失，估算监视器允许的启动速率*      if(!m\_ltIsSampling){          if(!rs.m\_bytesLoss){return ;}  **ResetLongTermBandwidthSamplingInterval**();          m\_ltIsSampling=true;}  *//避免数据不足，需要重新采样*      if(rs.m\_isAppLimited){  **ResetLongTermBandwidthSampling**();          return ;}      if(m\_roundStart){m\_ltRttCount++; }      if(m\_ltRttCount<bbr\_lt\_intvl\_min\_rtts){return ;  if(m\_ltRttCount>4 \* bbr\_lt\_intvl\_min\_rtts){**ResetLongTermBandwidthSampling**();return ;}  *//发生丢包停止采样*      if(!rs.m\_bytesLoss){          return ;      }      lost=m\_bytesLost-m\_ltLastLost;      delivered=m\_delivered-m\_ltLastDelivered;      if(!delivered||(lost\*bbr\_lt\_loss\_thresh\_den<delivered\*bbr\_lt\_loss\_thresh\_num)){          return ;      }  *//寻找在采样间隔的平均速率*      Time t=m\_deliveredTime-m\_ltLastStamp;      if(t<**MilliSeconds**(1)){return ;}      uint32\_t value=**std**::**numeric\_limits**<uint32\_t>::**max**()/1000;      if(t>=**MilliSeconds**(value)){  **ResetLongTermBandwidthSampling**();          #if (**TCP\_BBR\_DEGUG**)  **NS\_LOG\_FUNCTION**(m\_debug->**GetUuid**()<<"interval too long");          #endif          return ;      }      double bps=1.0\*delivered\*8000/t.**GetMilliSeconds**();  **DataRate** **bw**(**bps**);      #if (**TCP\_BBR\_DEGUG**)  **NS\_LOG\_FUNCTION**(m\_debug->**GetUuid**()<<delivered<<t.**GetMilliSeconds**()<<bw<<**BbrMaxBandwidth**());      #endif  **LongTermBandwidthIntervalDone**(tcb,bw);  } |

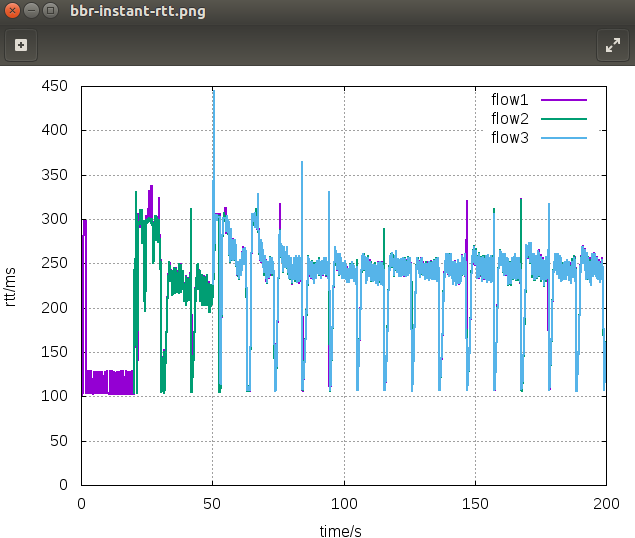
Demo运行数据图片展示:



记录了bbr的运行阶段，运行时间，最大带宽，rrt等重要数据

根据以上的三个数据，我们利用.sh脚本进行了绘图。图像如下：





***[实验结论]***

从上面两个图可以看出三个实验启动的时间段是不一样的，在max-bw中我们可以发现在启动阶段，bw上涨的很快，伴随着多个实验的并行，flow1逐渐bw宽度下降，但是可以注意到下降的比较平缓，对应的证明了bbr算法中对于减小带宽有滞后性，而增加是瞬间的。

在instant-bw中我们可以看到三条曲线的rtt以“平缓-剧烈变化-平缓”进行变化，证明了这是采样导致的，采样后会平缓运行一段时间。

算法的优势和可以解决的问题

BBR相对TCP的优点包括抗丢包能力强、延迟低、抢占能力强和平稳发送，在高带宽高延时的链路中有显著的性能优势。

由于BBR算法不区分普通数据和丢包重传数据，被重传的数据和普通数据一同进入发送缓存在同一个发送速率spacing rate和cwnd下送入网络中。因此BBR的拥塞窗口只有在发送端探测到当前的最大可用带宽减小时才会主动降低cwnd的值，而传统TCP拥塞控制算法在遇到丢包时就会立即将当前cwnd减小降低发送到网络中的数据总量。简而言之，在有一定丢包率的通信链路中BBR算法有着天然的优势。因此在一些误码率较高的网络上有积极意义。例如在卫星网络中，延时和丢包率都比较高，在这样的网络中部署BBR具有较高的效果。在实时音视频领域，BBR对带宽（特别是低带宽场景）探测准确，适合实时音视频传输的低延时需求，能够满足音视频传输码率平稳的需求以及快速响应带宽变化这四个要求。

## 24.3 实验代码

见文件夹文件