实验说明

一、Mininet实验环境安装

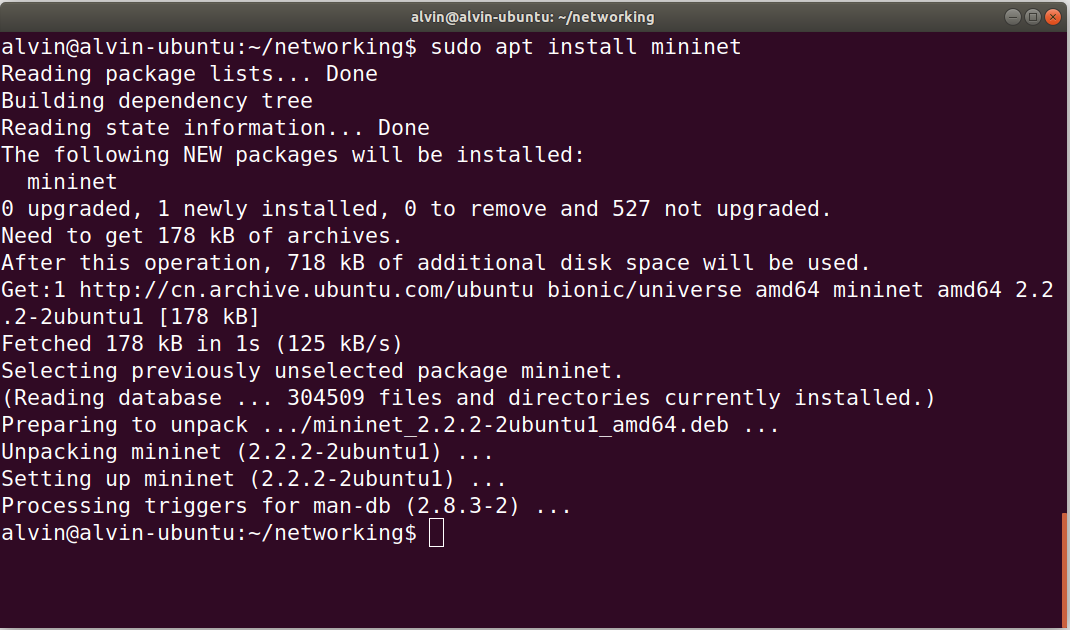
1. 实验基于的Mininet环境只能安装在Linux系统下，推荐使用Ubuntu发行版，版本号从14.04到18.10都可以，64位或32位都行，推荐使用64位系统。如果物理机为Windows系统，可以使用虚拟机方式安装Linux系统，推荐使用VirtualBox虚拟机。运行Mininet环境时需要root权限。
2. 创建networking目录，网络实验相关文件都放在该目录下：

~$ mkdir -p ~/networking

~$ cd ~/networking

1. 安装Mininet，有两种方式：一种是直接使用apt安装，该种方式的好处是简单，缺点是可能存在若干Bug（例如，sudo mn --link=tc,bw=10 会报“RTNETLINK answers: No such file or directory”），但这些Bug并不影响mininet的使用；二是从源码安装，该种安装方式可避免Ubuntu中的Bug，但可能会由于github网站被屏蔽而安装失败。推荐使用第二种方式安装。
   1. 使用apt安装mininet

~/networking$ sudo apt install mininet



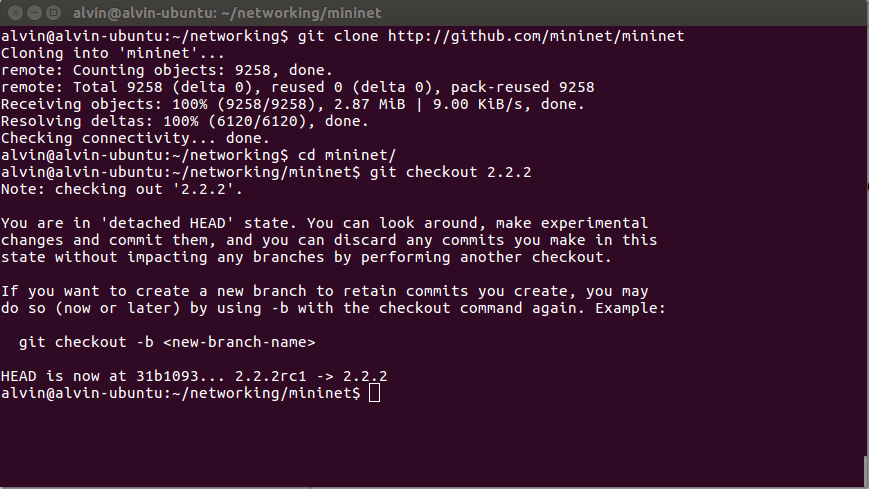
* 1. 从源代码安装

从github上下载mininet项目，切换到2.2.2版本的分支：

~/networking$ git clone http://github.com/mininet/mininet

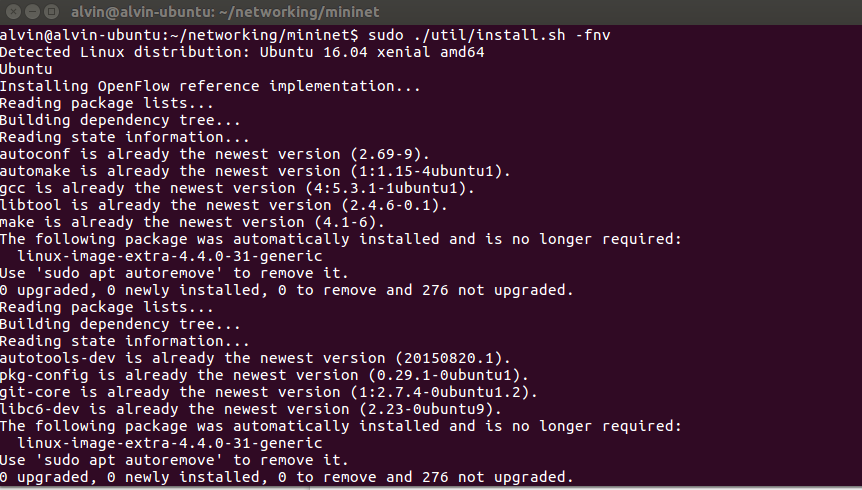
~/networking$ cd mininet

~/networking/mininet$ git checkout 2.2.2



安装mininet，该过程中会下载安装一些依赖程序，依据网络状况，大约需要几分钟。

~/networking/mininet$ sudo ./util/install.sh -fnv

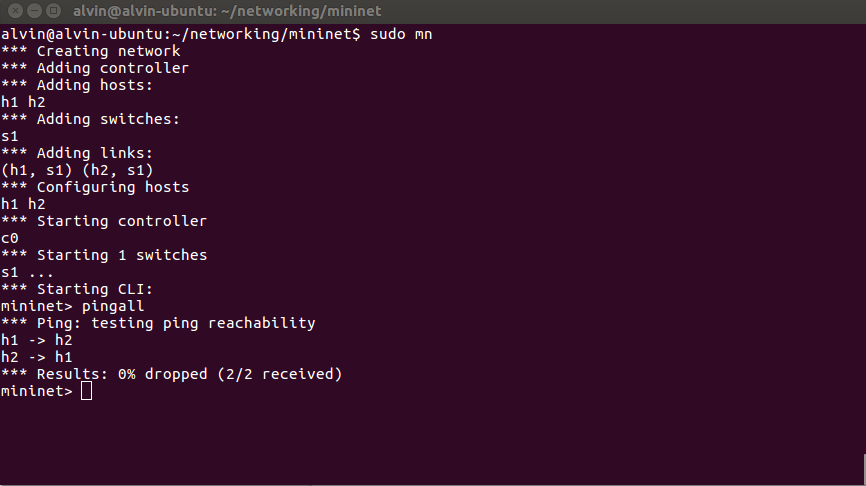


1. 运行mininet，验证mininet是否成功安装：执行pingall命令，如果所有节点都互通，说明安装成功。

~/networking$ sudo mn

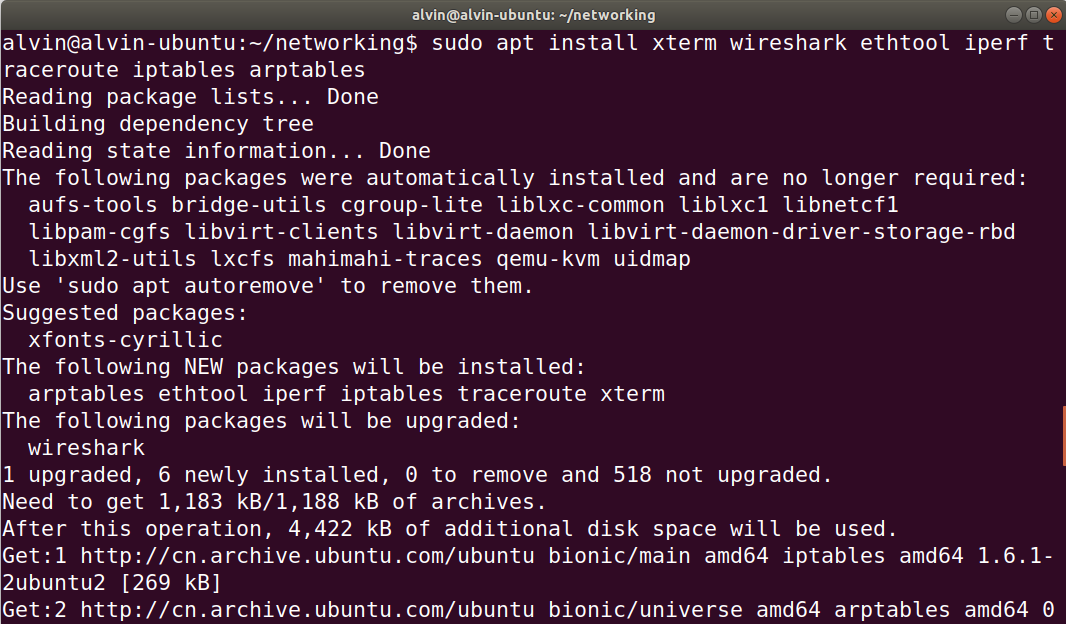
mininet> pingall

mininet> quit



1. 安装实验中用到的相关工具

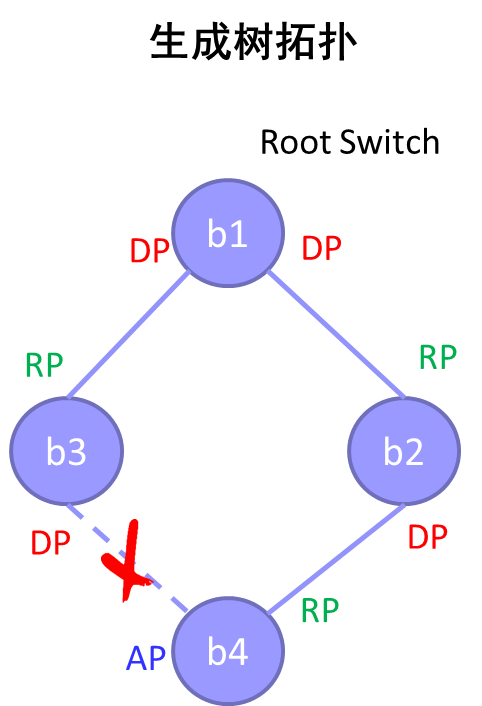
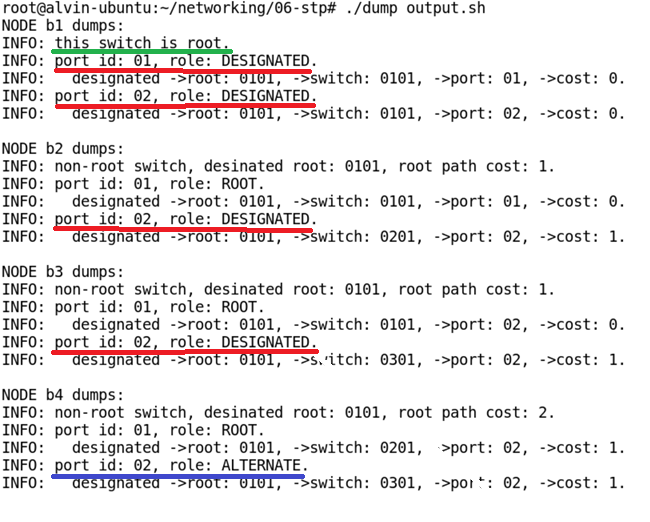
~/networking$ sudo apt install build-essential xterm wireshark ethtool iperf traceroute iptables arptables



二、计算机专业同学相关实验

1. STP生成树机制实验

本实验要求学生在已有代码基础上，完善其中的TODO部分，实现交换机之间通过交换STP消息，能够选举出该网络中的根节点，以及确定每个端口的状态。附件中给出了一个包含4个节点带环路的拓扑。基于这个拓扑，在节点b1-b4上分别运行stp程序，每个stp确定本节点是否是根节点，以及每个端口的角色，并把这些信息输出到屏幕或者文件，样式如下图所示。注意，与课堂上讲的不太一样，本实验只要求实现生成树拓扑的构建，不要求实现数据包转发。

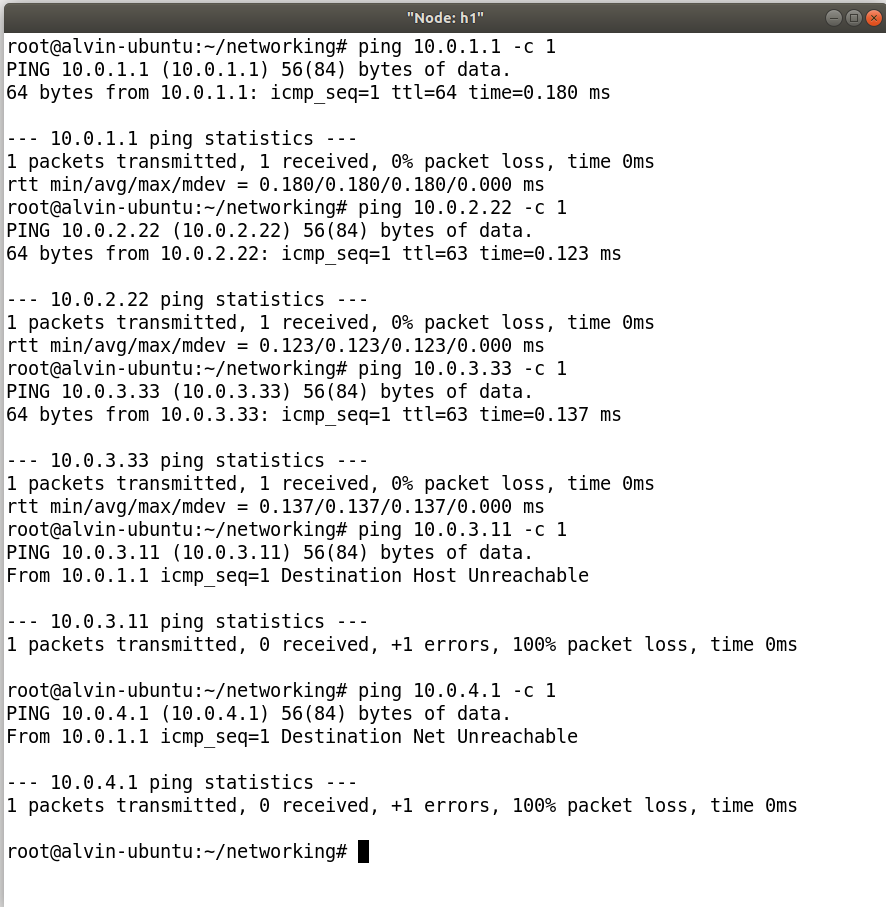
 

进一步，学生需要根据前面的拓扑样例，自己编写一个包含不少于7个交换机节点，不少于2条冗余边的拓扑，每个交换机分别运行stp程序，输出节点和端口角色等信息，样式如上。

建议组队人数不超过2人。

1. IP路由器转发实验

本实验要求学生在已有代码基础上，完善其中的TODO部分，实现路由器的IP查找转发、ARP请求和应答、ARP缓存管理、发送ICMP消息等功能。附件中给出了一个包含1个路由器节点和3个主机节点的拓扑。基于该拓扑，在路由器节点r1上运行router程序，在h1节点上ping如下地址：10.0.1.1(r1与h1相连端口的IP地址)，10.0.2.22（主机h2的IP地址），10.0.3.33（主机h3的IP地址），10.0.3.11（该IP地址不存在），10.0.4.1（该IP地址所在网络不存在），输出消息应如下图所示。

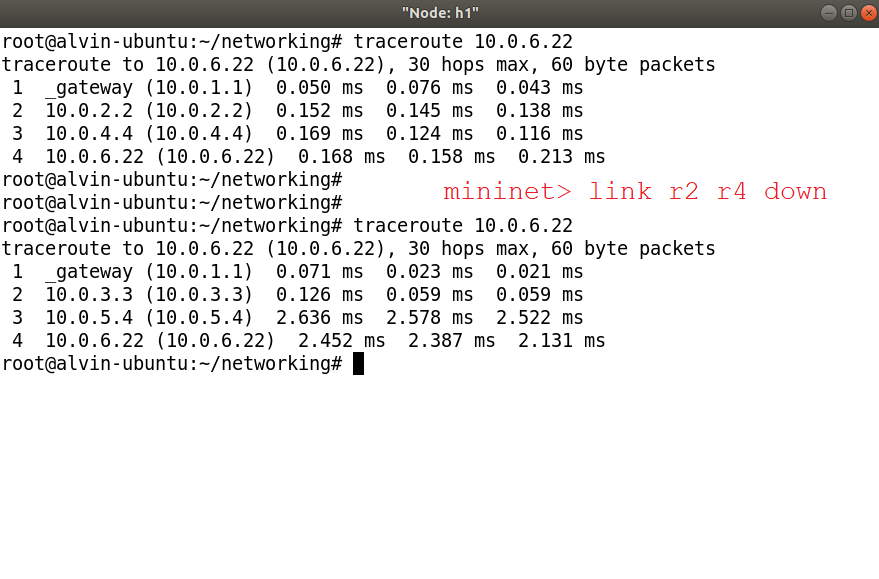


进一步，学生需要根据前面的拓扑样例，自己编写一个包含不少于2个路由器节点、2个主机节点的拓扑，并根据PPT中的路由表配置示例进行给每个节点进行路由器配置。每个路由器节点运行router程序，两个主机之间能够ping通，在一主机上traceroute另一主机，能够得到数据包经过每个节点的入端口IP地址。

建议组队人数不超过2人。

1. MOSPF动态路由实验

本实验要求学生在已有代码基础上，完善其中的TODO部分，实现路由器生成和处理mOSPF Hello/LSU消息等功能，构建一致性链路状态数据库，并计算路由表项。附件中给出了一个包含4个路由器节点和2个主机节点的拓扑。基于该拓扑，在路由器r1-r4上分别运行mospfd程序，每个节点生成、发送并处理mOSPF消息，构建自己的链路状态数据库，并根据该数据库独立计算生成自己的路由表项。大约等待半分钟，在主机h1上去ping h2，应该能够ping通，在主机h1上去traceroute h2，能够得到路径上每跳节点入端口的IP地址。在mininet命令行下，使用命令link <n1> <n2> down将该路径上的一条链路断开，但不影响该拓扑的连通性，等待一段时间（大概1分钟，链路状态数据库需要通过超时才能检测到过期的链路状态）后，在主机h1上去traceroute h2，能够得到路径上每跳节点入端口的IP地址，且该路径与断开链路前的路径不同。实验结果如下所示。



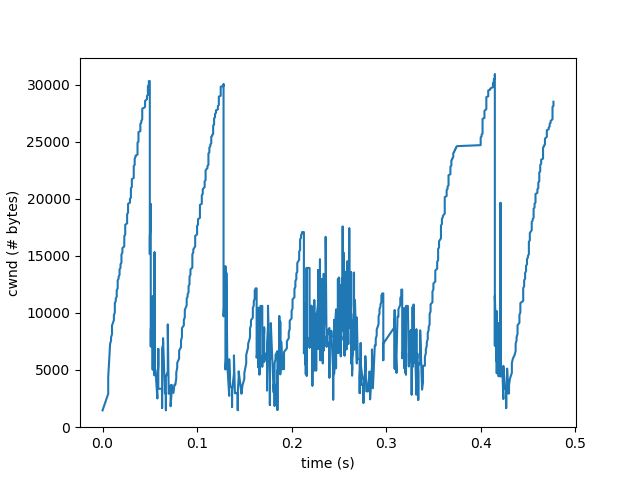
进一步，在Mininet中使用link <n1> <n2> up将该链路重新启用，等待一段时间，用traceroute确定h1与h2之间的路径。然后将该路径上中间节点（r2或r3）的mospfd程序关掉，模拟节点失效，等待一段时间，再使用traceroute确定h1与h2之间的路径。确保每次链路的断开/重连，以及节点的失效，mospfd都能按时更新链路状态数据库并计算出可达路径。

需要说明的是，该实验依赖IP查找转发的功能，对应静态路由转发实验中的C文件实现，这些功能以静态链接库的形式提供给大家，其API使用方法可以参考上述C文件，里面有更详细的注释，或者参考本实验include文件下的各个头文件。

建议组队人数不超过3人。

1. TCP网络传输实验

本实验要求学生在已有代码基础上，完善其中的TODO部分，实现TCP连接管理、数据传输、丢包恢复、拥塞控制等功能，使得两个协议栈之间能够进行可靠数据传输，并根据丢包情况按照TCP newReno规则调整自己的拥塞窗口大小。附件中给了两个拓扑，每个拓扑各包含2个主机节点，其中一个拓扑中链路无丢包，仅用于测试使用，另一个拓扑中链路有一定的丢包率。基于有丢包的拓扑，主机节点h1上运行TCP Server程序(./tcp\_stack server 10001)，主机h2上运行TCP Client程序(./tcp\_stack client 10.0.0.1 10001)，两个协议栈之间进行数据传输。使用md5sum程序验证，Server接收到的数据应与Client发送的数据完全相同。在Client程序（数据发送方）记录每次拥塞窗口变化的时间和大小，在有丢包环境下，拥塞窗口的变化图应类似于下图中的锯齿状。



需要说明的是，该实验依赖IP查找转发的功能，对应静态路由转发实验中的C文件实现，这些功能以静态链接库的形式提供给大家，其API使用方法可以参考上述C文件，里面有更详细的注释，或者参考本实验include文件下的各个头文件。

建议组队人数不超过4人。

三、非计算机专业同学相关实验

1. 广播网络实现和验证

本实验要求学生在已有代码基础上，完善其中的TODO部分，实现集线器节点广播数据的功能，使得不同主机能够通过集线器进行数据传输。附件中包含两个拓扑文件，第一个拓扑包含1个集线器(b1)和3个主机节点，其中主机h1与b1之间的带宽为20Mbps，h2，h3与b1之间的带宽为10Mbps。在该拓扑上使用iperf验证广播网络的效率问题，两种场景：h1做为iperf server，h2和h3做为iperf client同时给h1发送数据；h2和h3做为iperf server，h1做为iperf client同时给h1发送数据。获取两种场景下各条传输流的速率并解释这种现象的原因。（提示：链路带宽为双向的，a与b之间有一条10Mbps的链路，a在以10Mbps速率向b发送数据的同时，b也可以以10Mbps的速率给a发数据）。第二个拓扑包含3个集线器构成的环状网络和2个主机节点，基于该拓扑，h2主机上打开wireshark对h2-eth0端口捕获数据包，在h1上向h2发送一个ping数据包(ping -c 1 10.0.0.2)，如果在转发网络中，h2只能捕获到4个数据包（2个ARP包和2个ping包），但在广播网络中，应该能在h2抓到非常多的数据包（有些Ubuntu内核有Bug，只能抓到几百到几千个数据包）。

建议该题目1人独立完成。

1. 交换机MAC学习实现

本实验要求学生在已有代码基础上，完善其中的TODO部分，实现交换机学习MAC到端口的映射关系，以及数据包转发，使得主机的大部分数据包通过转发而不是广播到达目的节点。附件中的拓扑包含1个交换机节点和3个主机节点，在主机h2和h3上分别打开wireshark程序，监听各自主机的eth0端口(h2-eth0和h3-eth0)，在h1主机上分别ping h2和h3两个主机，在两个wireshark中捕获的应该只包含自己节点和h1产生的数据包。

建议组队人数不超过2人。

四、实验补充说明

1. Python版本问题

Mininet环境只能使用Python2来解释运行，如果Linux系统中同时安装有Python2/3时，使用python2 topo.py来启动Mininet环境。

1. 脚本和reference程序的可执行权限问题

整个实验的源文件、可执行文件、脚本等都用.tar.gz格式压缩打包，压缩文件应该在Linux系统下解压缩才能保留每个文件的读写执行权限。如果在Windows下解压再拷贝到Linux系统下，或者把文件解压缩到Windows与Linux共享的文件夹下，那么可执行文件和脚本可能会失去可执行权限而不能运行。这种情况下，使用chmod命令给这些文件添加可执行权限：chmod +x \*.sh \*.py \*reference\*

另外，Bash脚本依赖的程序（例如，ethtool, arptables等）一定要提前安装，否则执行Bash脚本时不会成功，也没有任何错误提示。

1. 32位/64位操作系统

本实验提供的reference包括64位版本和32位版本(\*-reference.32)，通过uname -a命令查看输出中是否含有x86\_64字段，如果有则为64位系统，没有则为32位系统，根据系统版本执行相应的reference文件。如果Linux系统为32位，且选择的是MOSPF或TCP实验，需要使用libipstack32.a覆盖libipstack.a，否则会编译失败。

1. Linux发行版

如果使用的不是Ubuntu发行版，可能会遇到相应安装包找不到，或者脚本不能正常执行等问题，例如Arch发行版中IP转发相关配置文件不是/proc/sys/net/ipv4/ip\_forward，导致disable\_ip\_forward.sh脚本运行失败。这些问题都可以通过网上搜索找到相应解决方案。

1. 添加新的源文件

当实验中需要添加新的源文件时，如果添加头文件，则直接将该文件放到include目录下，不需要做其他改动；如果添加C文件，则需要修改Makefile，将该文件的名字放到该文件SRCS = …一行结尾。

五、提交实验说明

提交形式：

1. 实验报告：模板不限，内容包括但不限于实验题目、项目组员、实验内容、实验流程、实验结果及分析；
2. 项目完整代码：以tar.gz格式打包，编译时不能有错误或警告，编译出的程序能正常运行，完成实验要求；
3. 每组队伍只需一人提交内容即可，一定要在报告正文中写清楚成员姓名和学号。

起止时间：

开始时间：4月12日；

截止时间：6月9日（8周），在此之前提交，可获得的最高分为100%；

缓冲时间：6月23日（10周，硬截止），在9-10周内提交，可获得的最高分为90%，之后不再接收任何提交。

反抄袭说明：

禁止抄袭实验代码或报告内容，如检测出代码或报告内容雷同，所有队员的实验分数按不及格处理。建议不要直接拷贝实验说明PPT中的文字。