网络互联实验

实验内容

- 在主机上安装arptables, iptables，用于禁止每个节点的相应功能
- 运行给定网络拓扑(router_topo.py)

  路由器节点r1上执行脚本(disable_arp.sh, disable_icmp.sh, disable_ip_forward.sh)，禁止协议栈的相应功能

  终端节点h1-h3上执行脚本disable_offloading.sh
- 在r1上执行路由器程序
- 在h1上进行ping实验
  - Ping 10.0.1.1 (r1)，能够ping通
  - Ping 10.0.2.22 (h2)，能够ping通
  - Ping 10.0.3.33 (h3)，能够ping通
  - Ping 10.0.3.11，返回ICMP Destination Host UnreachablePing 10.0.4.1，返回ICMP Destination Net Unreachable
- 构造一个包含多个路由器节点组成的网络
  - 手动配置每个路由器节点的路由表
  - 有两个终端节点，通过路由器节点相连，两节点之间的跳数不少于3跳，手动配置其默认路由表
- 连通性测试
  - 终端节点ping每个路由器节点的入端口IP地址，能够ping通
- 路径测试
  - 在一个终端节点上traceroute另一节点，能够正确输出路径上每个节点的IP信息

实验流程

代码见附件

make编译所有

```
nowcoder@nowcoder:~/ucas_network/4-router$ make
gcc -c -g -Wall -Iinclude arp.c -o arp.o
gcc -c -g -Wall -Iinclude arpcache.c -o arpcache.o
gcc -c -g -Wall -Iinclude icmp.c -o icmp.o
gcc -c -g -Wall -Iinclude ip_base.c -o ip_base.o
gcc -c -g -Wall -Iinclude rtable.c -o rtable.o
gcc -c -g -Wall -Iinclude rtable_internal.c -o rtable_internal.o
gcc -c -g -Wall -Iinclude device_internal.c -o device_internal.o
ar rcs libipstack.a arp.o arpcache.o icmp.o ip_base.o rtable.o rtable_internal.o device_internal.o
gcc -c -g -Wall -Iinclude main.c -o main.o
gcc -c -g -Wall -Iinclude ip.c -o ip.o
gcc -L. main.o ip.o -o router -lipstack -lpthread
```

运行py文件

```
nowcoder@nowcoder:~/ucas_network/4-router$ sudo python router_topo.py
[sudo] nowcoder 的密码：
mininet> net
h1 h1-eth0:r1-eth0
h2 h2-eth0:r1-eth1
h3 h3-eth0:r1-eth2
r1 r1-eth0:h1-eth0 r1-eth1:h2-eth0 r1-eth2:h3-eth0
```

展示网络结构，打开h1

```
mininet> net
h1 h1-eth0:r1-eth0
h2 h2-eth0:r1-eth1
h3 h3-eth0:r1-eth2
r1 r1-eth0:h1-eth0 r1-eth1:h2-eth0 r1-eth2:h3-eth0
mininet> xterm h1
```

运行路由器r1

```
mininet> r1 ./router
DEBUG: find the following interfaces:  r1-eth0 r1-eth1 r1-eth2.
Routing table of 3 entries has been loaded.
```

h1 分别ping r1 h2 h3 10.0.3.11 10.0.4.1

```
"Node: h1"@nowcoder
root@nowcoder:~/ucas_network/4-router# ping 10.0.1.1 -c 4
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=0.101 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=0.493 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=0.280 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=64 time=0.218 ms

--- 10.0.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.101/0.273/0.493/0.142 ms
root@nowcoder:~/ucas_network/4-router# ping 10.0.2.22 -c 4
PING 10.0.2.22 (10.0.2.22) 56(84) bytes of data.
64 bytes from 10.0.2.22: icmp_seq=1 ttl=63 time=0.120 ms
64 bytes from 10.0.2.22: icmp_seq=2 ttl=63 time=0.368 ms
64 bytes from 10.0.2.22: icmp_seq=3 ttl=63 time=0.224 ms
64 bytes from 10.0.2.22: icmp_seq=4 ttl=63 time=0.302 ms

--- 10.0.2.22 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.120/0.253/0.368/0.093 ms
root@nowcoder:~/ucas_network/4-router# ping 10.0.3.33 -c 4
PING 10.0.3.33 (10.0.3.33) 56(84) bytes of data.
64 bytes from 10.0.3.33: icmp_seq=1 ttl=63 time=0.110 ms
64 bytes from 10.0.3.33: icmp_seq=2 ttl=63 time=0.344 ms
64 bytes from 10.0.3.33: icmp_seq=3 ttl=63 time=0.307 ms
64 bytes from 10.0.3.33: icmp_seq=4 ttl=63 time=0.261 ms

--- 10.0.3.33 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3083ms
rtt min/avg/max/mdev = 0.110/0.255/0.344/0.090 ms
root@nowcoder:~/ucas_network/4-router# ping 10.0.3.11 -c 4
PING 10.0.3.11 (10.0.3.11) 56(84) bytes of data.
From 10.0.1.1 icmp_seq=1 Destination Host Unreachable
From 10.0.1.1 icmp_seq=2 Destination Host Unreachable
From 10.0.1.1 icmp_seq=3 Destination Host Unreachable
From 10.0.1.1 icmp_seq=4 Destination Host Unreachable

--- 10.0.3.11 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3065ms
pipe 4
root@nowcoder:~/ucas_network/4-router# ping 10.0.4.1 -c 4
PING 10.0.4.1 (10.0.4.1) 56(84) bytes of data.
From 10.0.1.1 icmp_seq=1 Destination Net Unreachable
From 10.0.1.1 icmp_seq=2 Destination Net Unreachable
From 10.0.1.1 icmp_seq=3 Destination Net Unreachable
From 10.0.1.1 icmp_seq=4 Destination Net Unreachable
```

由结果可知，对于 r1 h2 h3 都可以连通，不存在的节点显示主机不可达，不存在的路由器显示网络不可达，实验成功。

按照实验要求创建了两个主机两个路由器的网络，实现见代码new_router.py

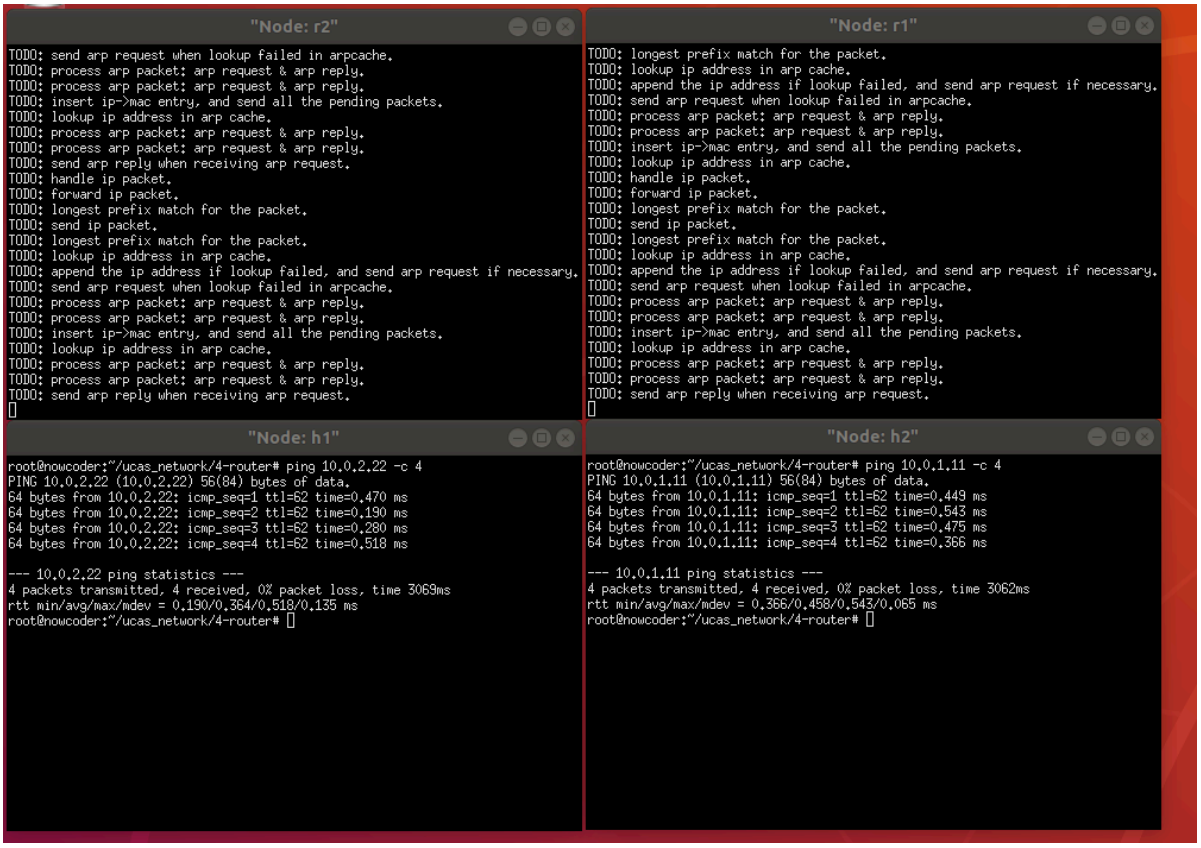
运行py文件

```
nowcoder@nowcoder:~/ucas_network/4-router$ sudo python new_router.py
[sudo] nowcoder 的密码:
mininet> net
h1 h1-eth0:r1-eth0
h2 h2-eth0:r2-eth0
r1 r1-eth0:h1-eth0 r1-eth1:r2-eth1
r2 r2-eth0:h2-eth0 r2-eth1:r1-eth1
```

展示网络构建情况

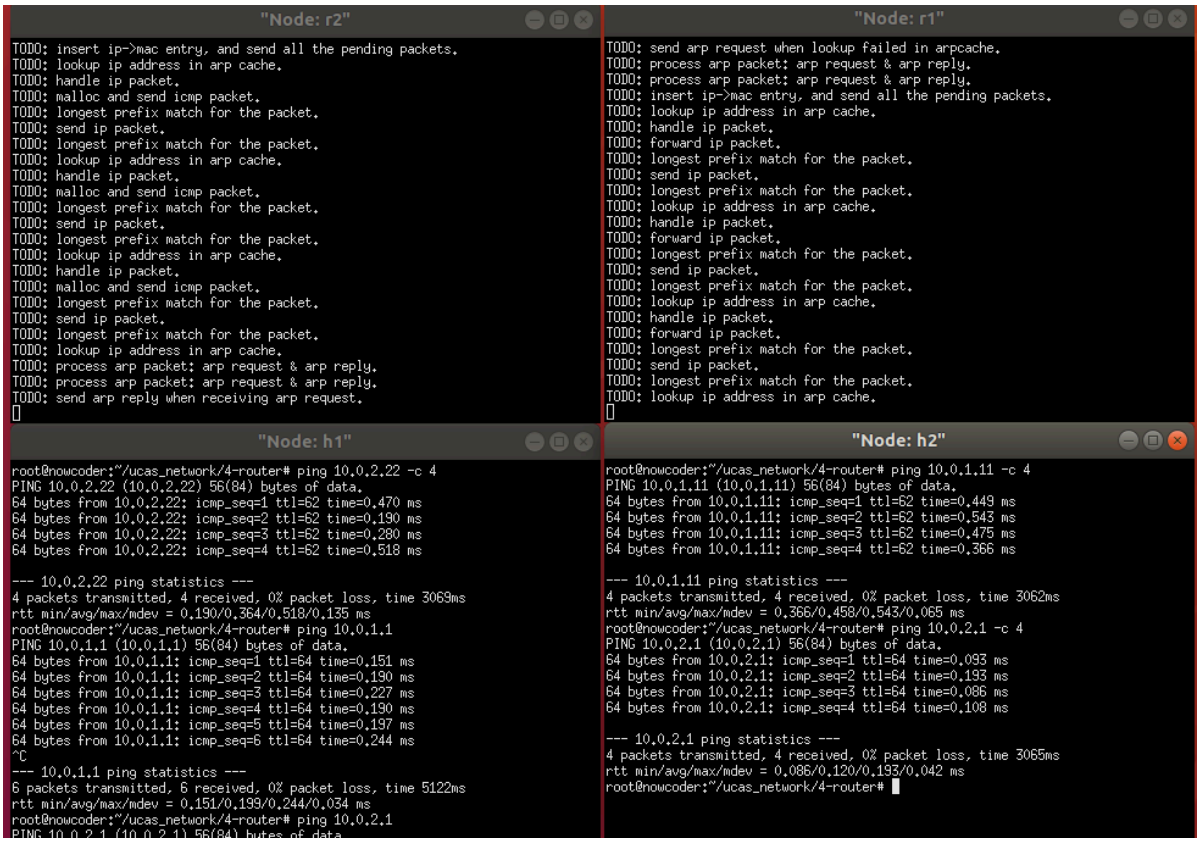
```
mininet> net
h1 h1-eth0:r1-eth0
h2 h2-eth0:r2-eth0
r1 r1-eth0:h1-eth0 r1-eth1:r2-eth1
r2 r2-eth0:h2-eth0 r2-eth1:r1-eth1
mininet> xterm h1 h2 r1 r2
mininet>
```

在r1 r2分别启动路由器后 h1 h2互相ping



发现都可以ping通

分别ping对应路由器的接口



也可以ping通

分别查看traceroute另一个终端节点的信息