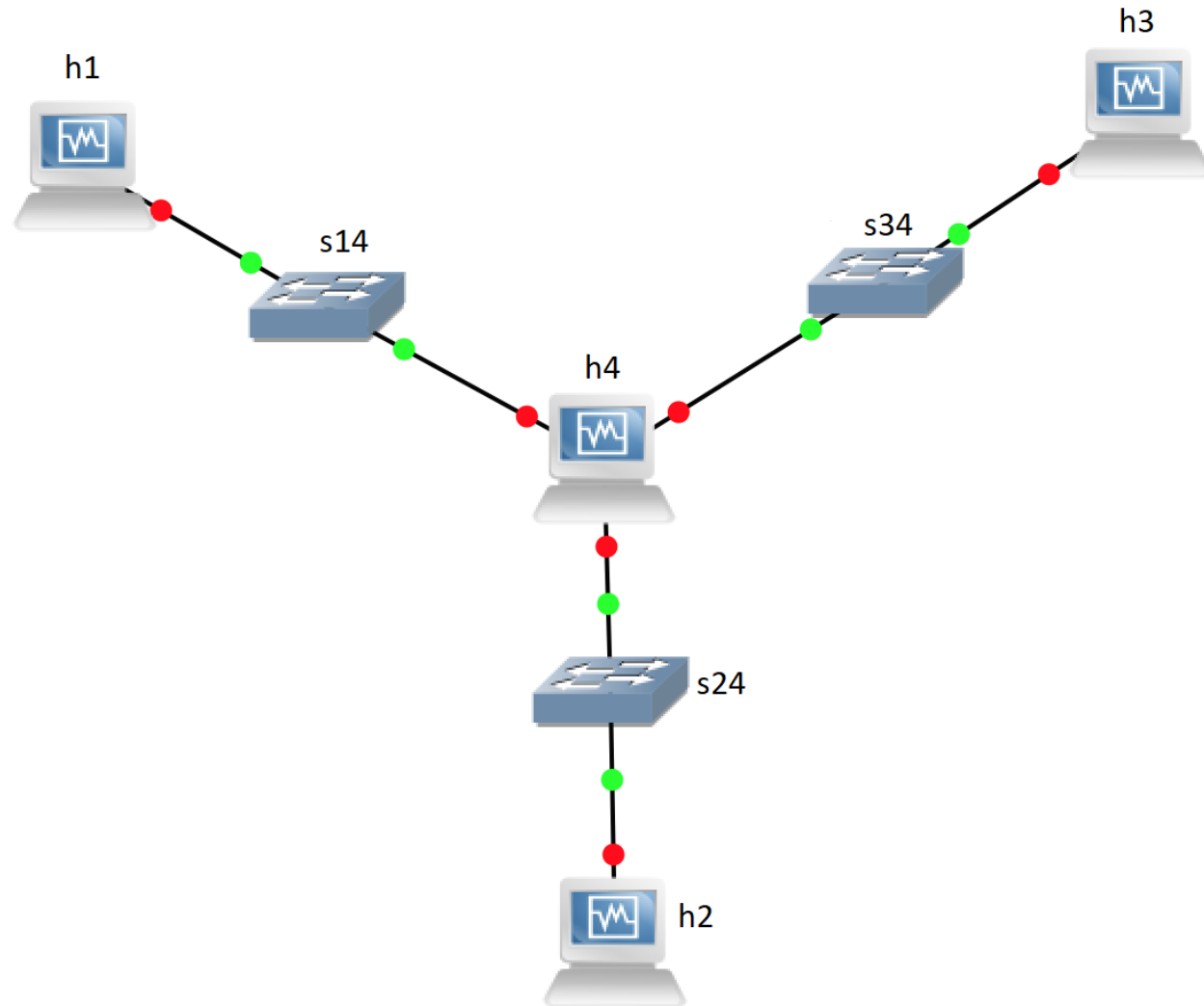


LAN Configuration

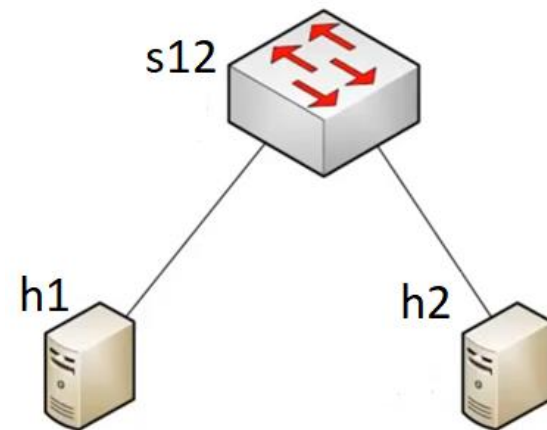
Custom topology



```

1 #!/usr/bin/python
2 """
3 This example shows how to create a Mininet object and add nodes to it
4 """
5 #Importing Libraries
6 from mininet.net import Mininet
7 from mininet.node import Controller
8 from mininet.cli import CLI
9 from mininet.log import setLogLevel, info
10
11 #Function definition: This is called from the main function
12 def firstNetwork():
13     #Create an empty network and add nodes to it.
14     net = Mininet()
15     info( '*** Adding controller\n' )
16     net.addController( 'c0' )
17
18     info( '*** Adding hosts\n' )
19     h1 = net.addHost( 'h1', ip='10.0.0.1' )
20     h2 = net.addHost( 'h2' )
21
22     info( '*** Adding switch\n' )
23     s12 = net.addSwitch( 's12' )
24
25     info( '*** Creating links\n' )
26     net.addLink( h1, s12 )
27     net.addLink( h2, s12 )
28
29     info( '*** Starting network\n' )
30     net.start()
31
32     #This is used to run commands on the hosts
33
34     info( '*** Starting xterm on hosts\n' )
35     h1.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h1 &')
36     h2.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h2 &')
37
38     info( '*** Running the command line interface\n' )
39     CLI( net )
40
41     info( '*** Closing the terminals on the hosts\n' )
42     h1.cmd("killall xterm")
43     h2.cmd("killall xterm")
44
45     info( '*** Stopping network' )
46     net.stop()

```



```

47
48 #main Function: This is called when the Python file is run
49 if __name__ == '__main__':
50     setLogLevel( 'info' )
51     firstNetwork()
52

```

Custom topology

- Change directory to shared folder:

- `$ cd Desktop/shared`

- Edit a python file, e.g. `lab1.py`:

- `$ sudo leafpad lab1.py`

- Run topology:

- `$ sudo python lab1.py`

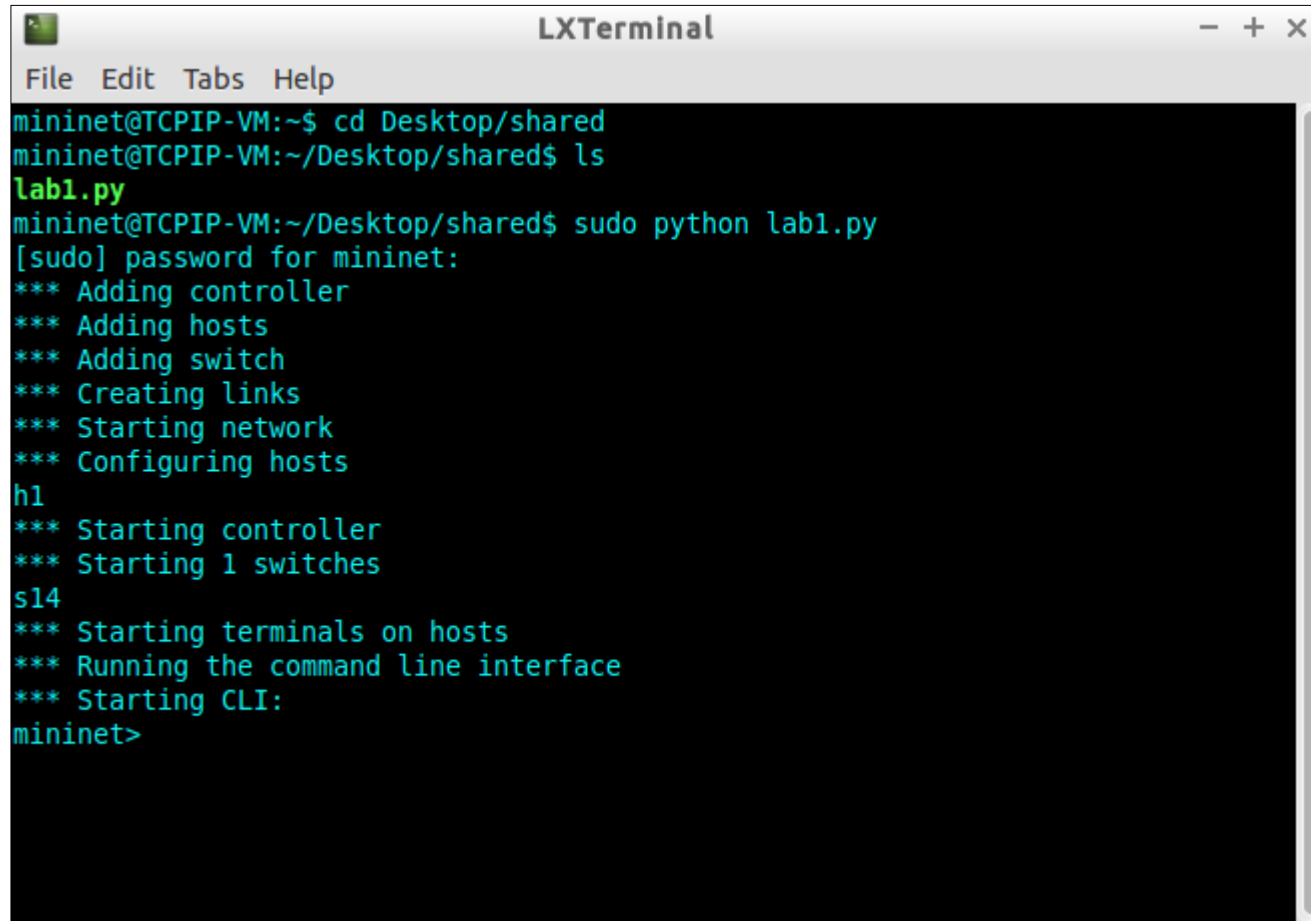
- Exit topology:

- `mininet> exit`

- Clean up:

- `$ sudo mn -c`

last slide codes

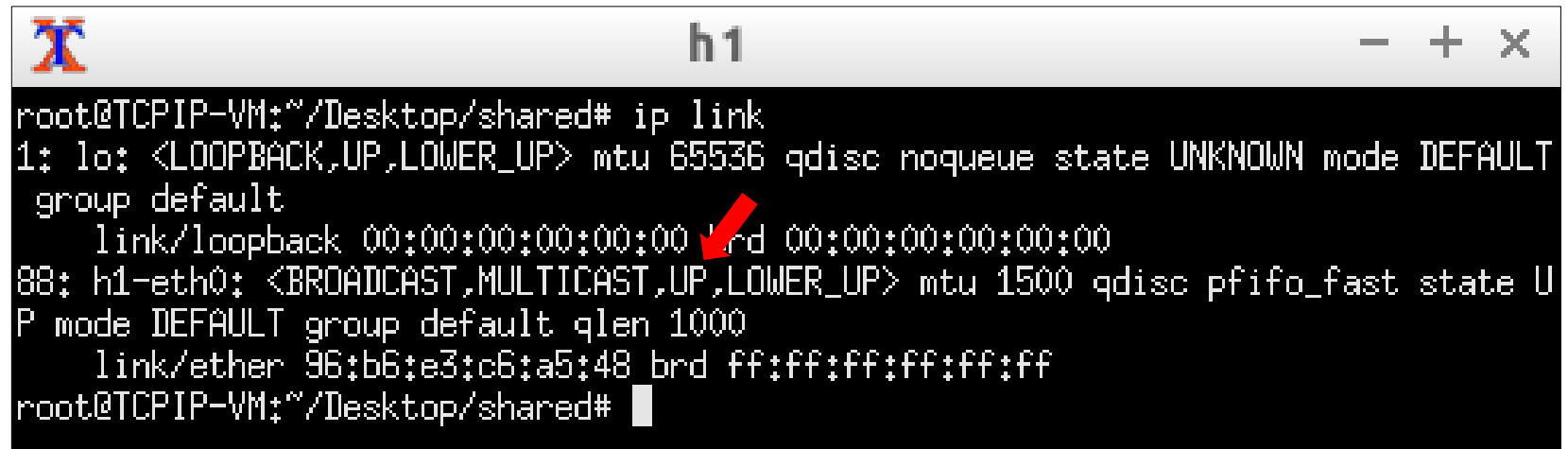


```
LXTerminal
File Edit Tabs Help
mininet@TCPIP-VM:~$ cd Desktop/shared
mininet@TCPIP-VM:~/Desktop/shared$ ls
lab1.py
mininet@TCPIP-VM:~/Desktop/shared$ sudo python lab1.py
[sudo] password for mininet:
*** Adding controller
*** Adding hosts
*** Adding switch
*** Creating links
*** Starting network
*** Configuring hosts
h1
*** Starting controller
*** Starting 1 switches
s14
*** Starting terminals on hosts
*** Running the command line interface
*** Starting CLI:
mininet>
```

Interfaces

- Show the mode of a host interfaces:

- # ip link



```
root@TCPIP-VM:~/Desktop/shared# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
88: h1-eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
    mode DEFAULT group default qlen 1000
    link/ether 96:b6:e3:c6:a5:48 brd ff:ff:ff:ff:ff:ff
root@TCPIP-VM:~/Desktop/shared#
```

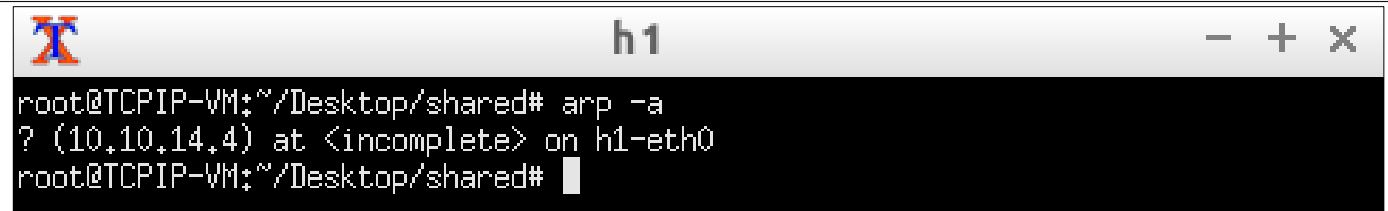
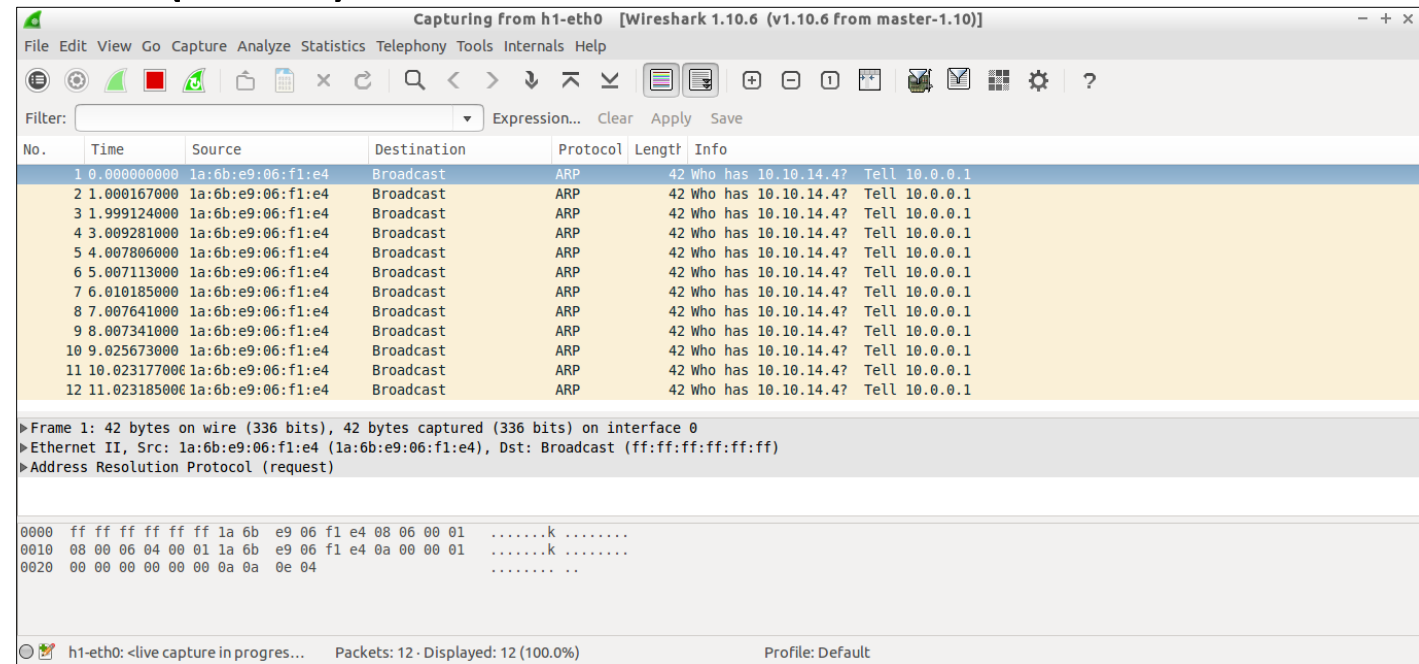
- If an interface mode is DOWN, change it to UP, e.g. h1-eth0:

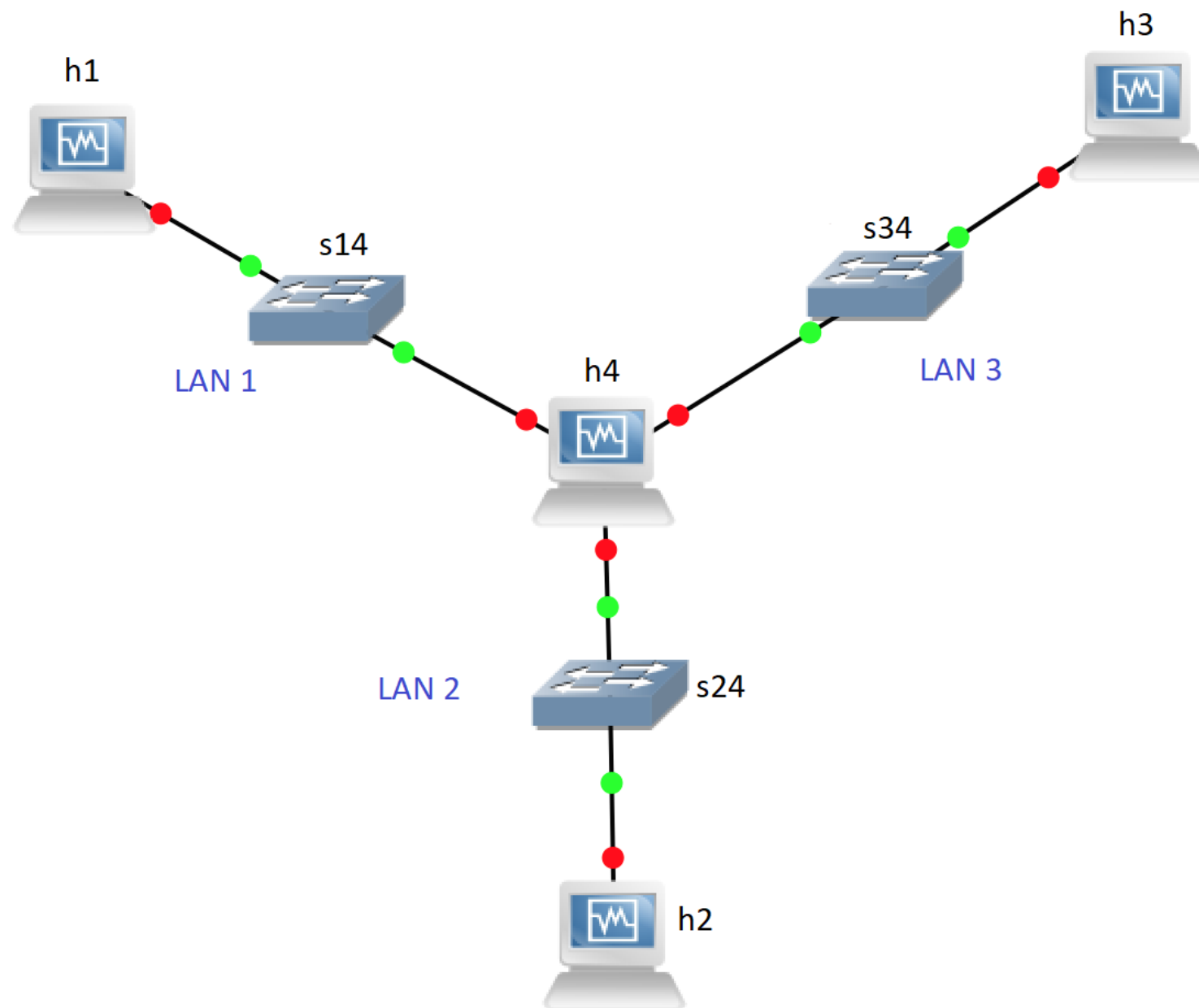
- # ip link set h1-eth0 up

- ping ✗

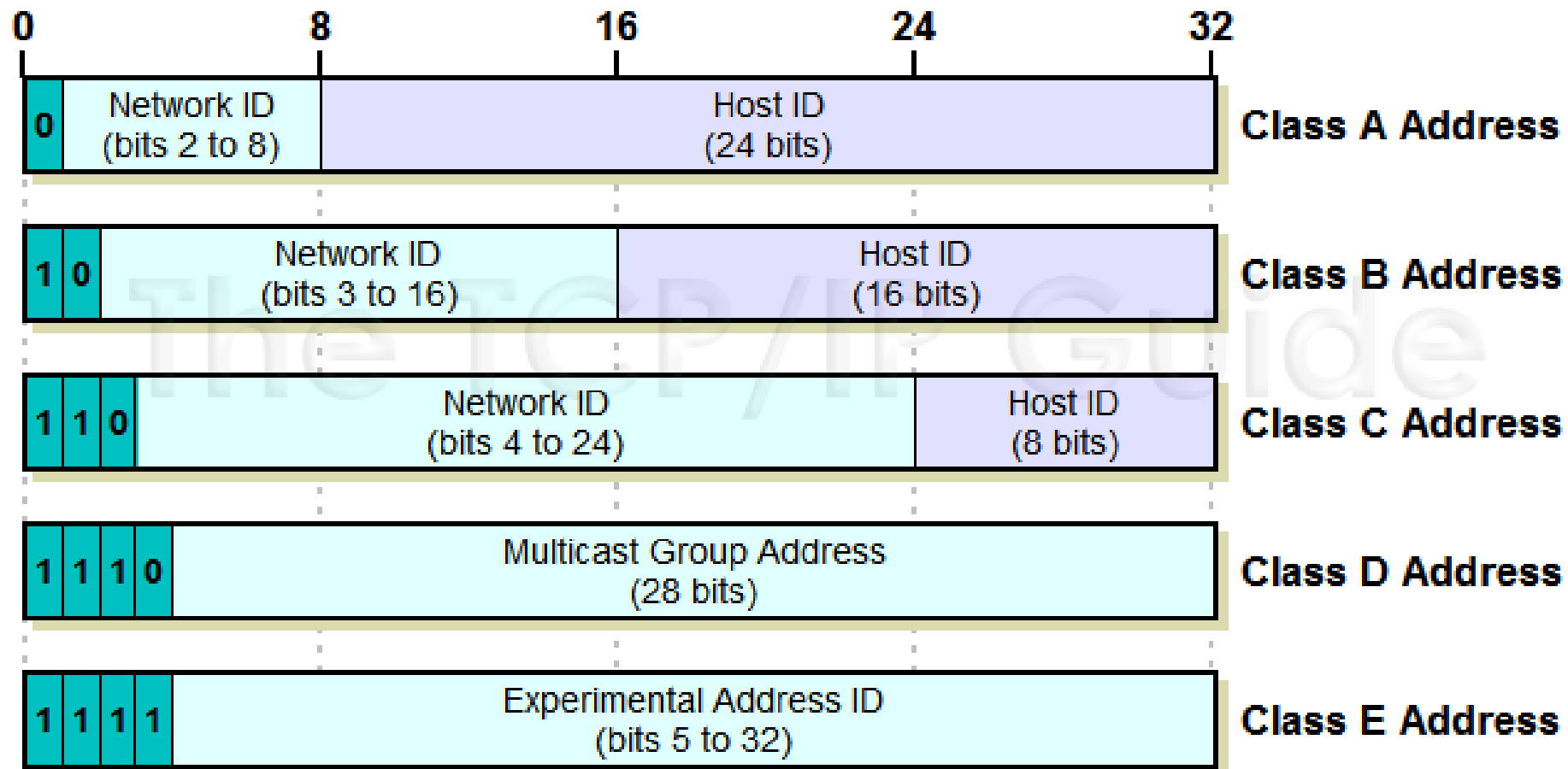
ARP (Address Resolution Protocol)

- A procedure for mapping a dynamic IP address to a physical address, known as a media access control (MAC) address.
 - ARP request
 - ARP reply
- Open Wireshark on a host:
 - # wireshark &
- mapping to physical address
- Show ARP table of a host:
 - # arp -a





IP Address Class Bit Assignments and Network/Host ID Sizes

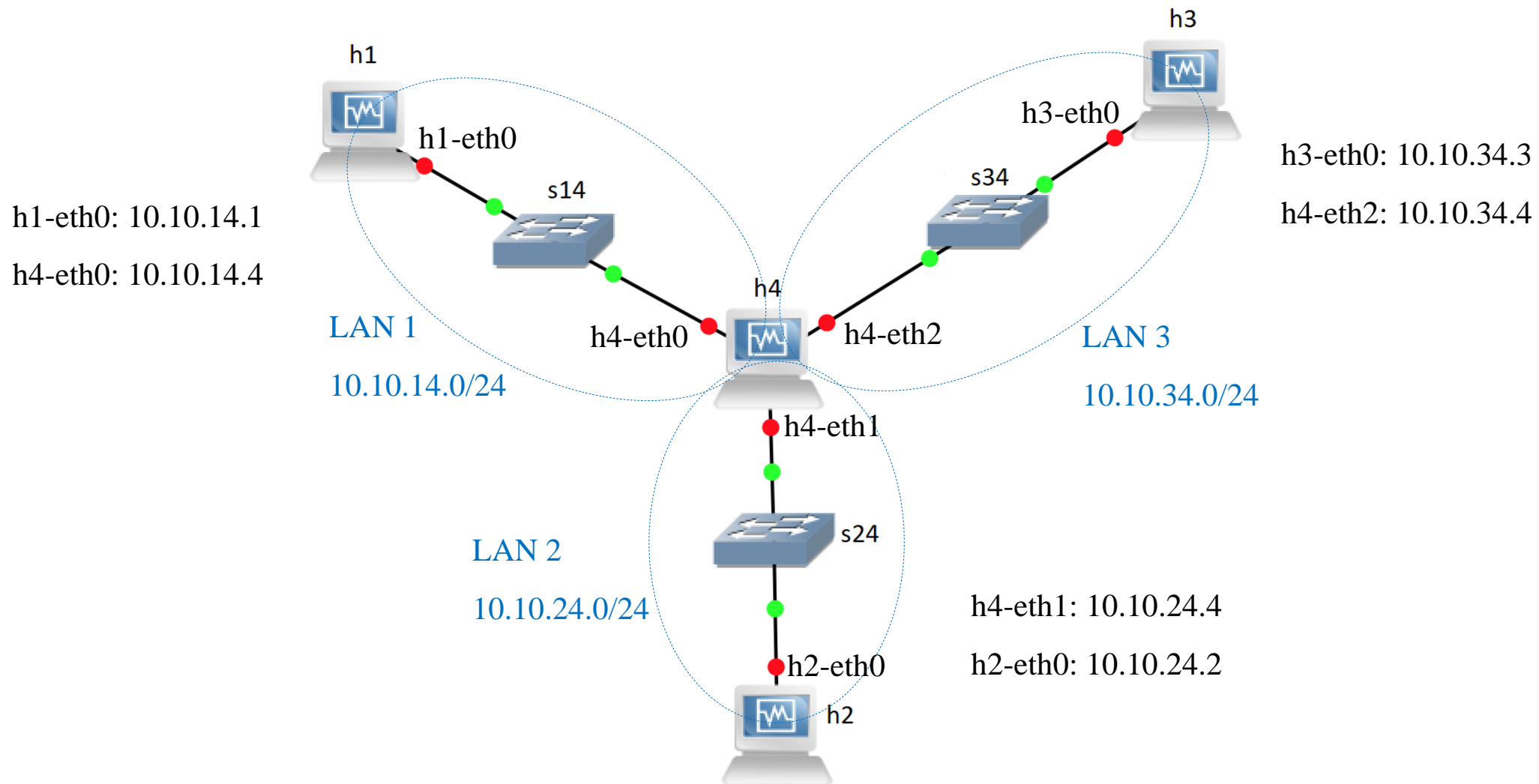


IP Address classes: Chart Representation

Address Classes	Range	Bit Pattern of 1 st byte	Decimal Range	Default Subnet Mask	Reserved for
A	1.0.0.0 to 127.255.255.255	0xxxxxxx	1 to 127	255.0.0.0	Governments
B	128.0.0.0 to 191.255.255.255	10xxxxxx	128-191	255.255.0.0	Medium Companies
C	192.0.0.0 to 223.255.255.255	110xxxxx	192-223	255.255.255.0	Small Companies
D	224.0.0.0 to 239.255.255.255	1110xxxx	224-239	Not Applicable	Reserved for Multicasting
E	240.0.0.0 to 255.255.255.255	11110xxx	240-255	Not Applicable	Experimental or future use

10.10.0.0/16

کل شبکه



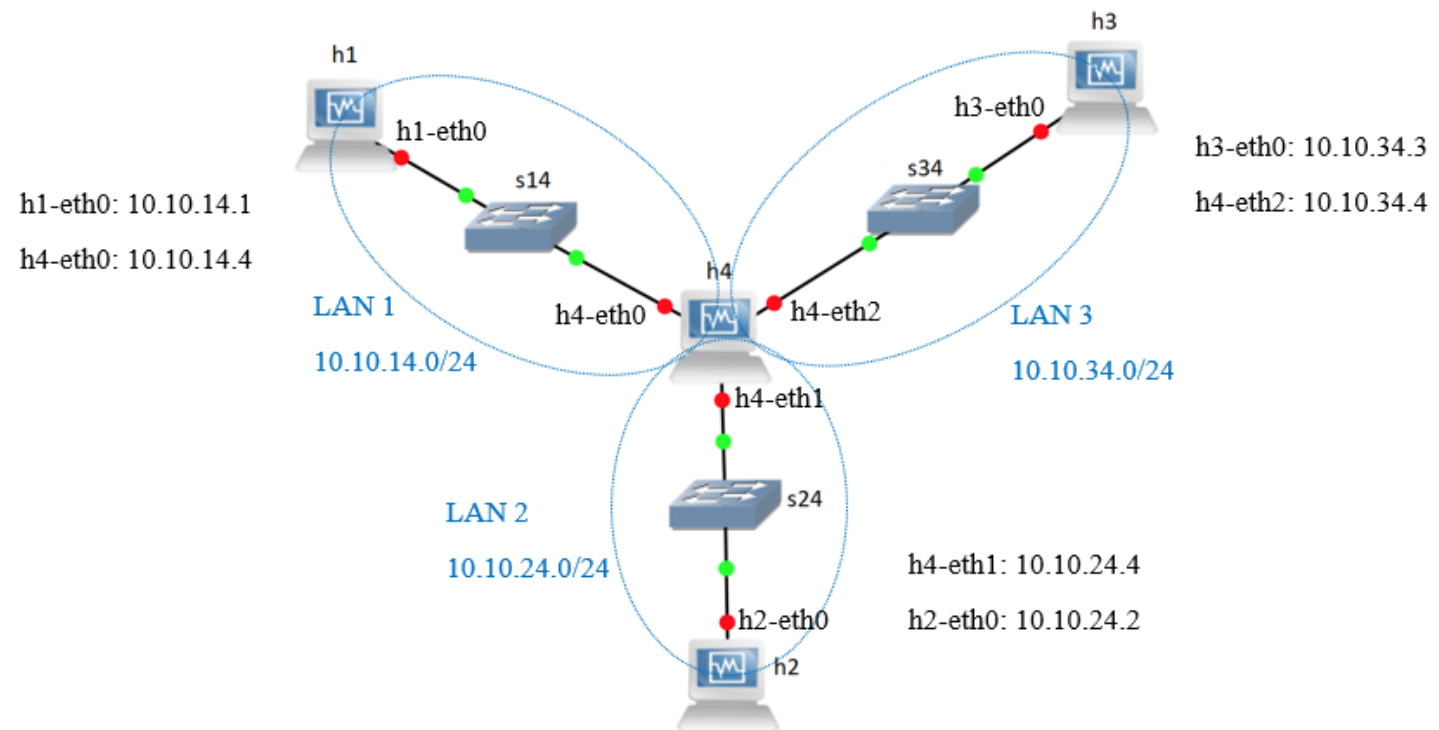
Assign an IP address to an interface

- `# ip addr flush dev h1-eth0` clear device ip addr flush instructs the ip utility to flush or clear all IP addresses. dev eth0 specifies the network device (eth0) to perform the operation on.
- `# ip addr add 10.10.14.1/24 dev h1-eth0` mask
- `# ifconfig -a`

```
root@TCPIP-VM:~/Desktop/shared# ip addr flush dev h1-eth0
root@TCPIP-VM:~/Desktop/shared# ip addr add 10.10.14.1/24 dev h1-eth0
root@TCPIP-VM:~/Desktop/shared# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr 76:42:fd:85:98:43
        inet addr:10.10.14.1 Bcast:0.0.0.0 Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:18 errors:0 dropped:0 overruns:0 frame:0
        TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1756 (1.7 KB)  TX bytes:828 (828.0 B)

lo       Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

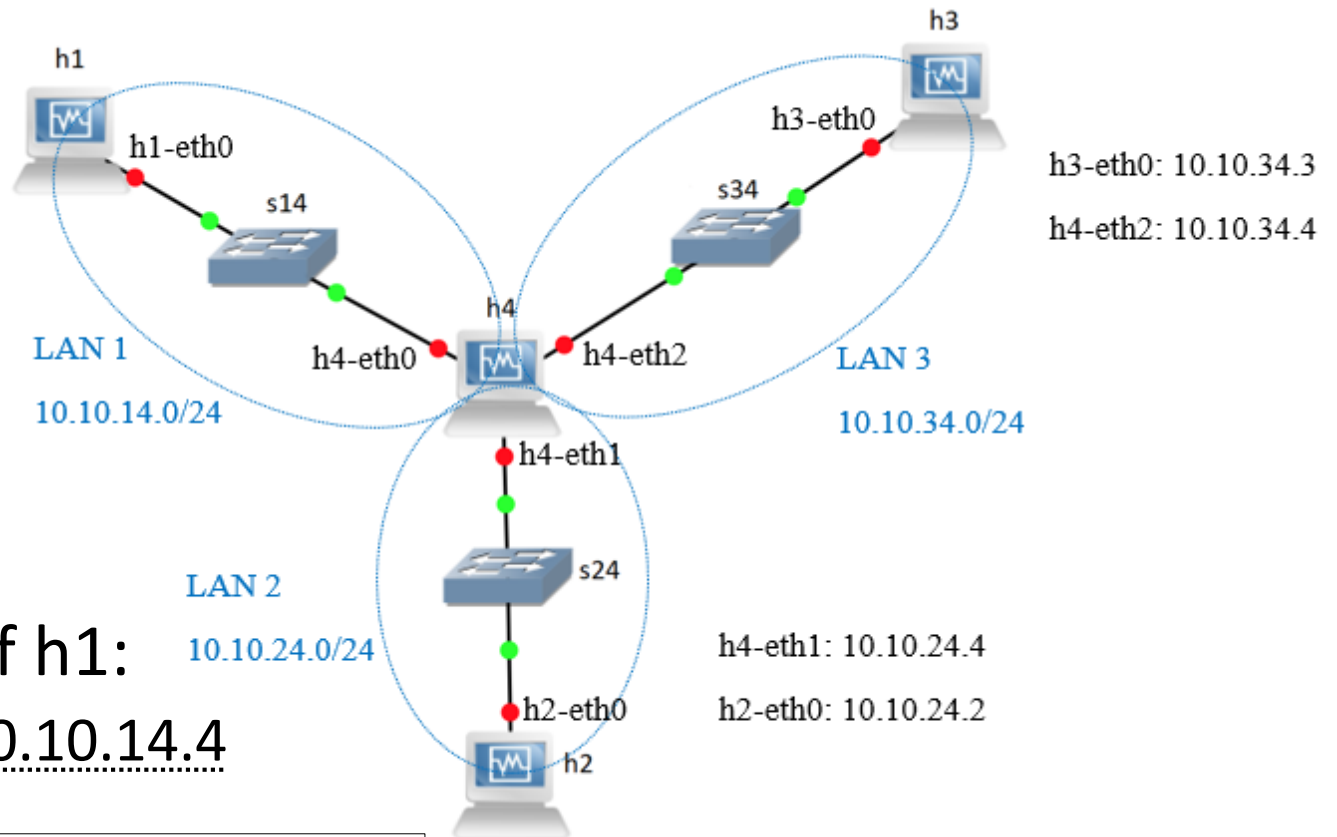
root@TCPIP-VM:~/Desktop/shared#
```



first hub router: default gateway

Set gateway

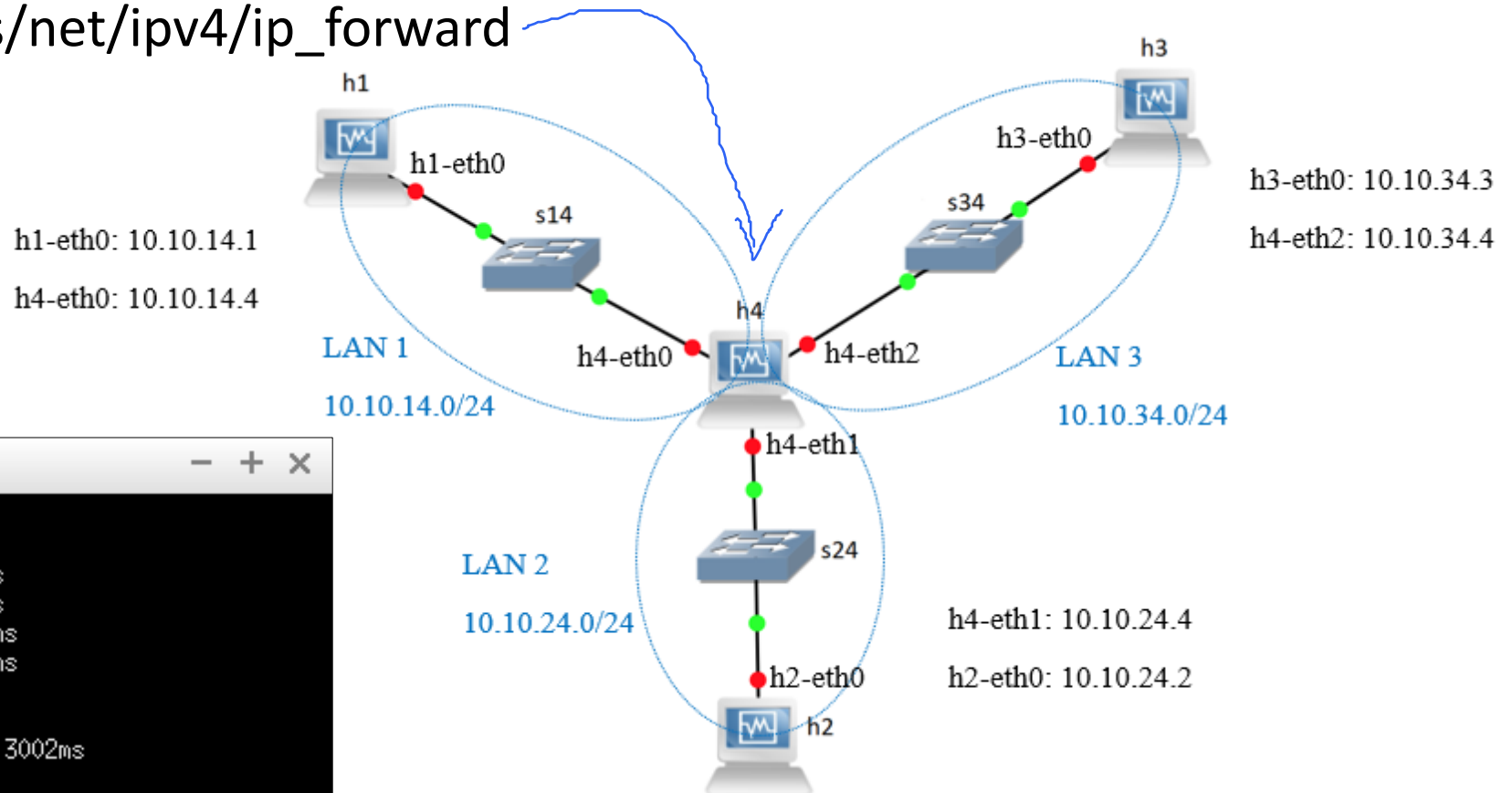
- Show routing table:
 - # ip route
- Add(Delete) default gateway of h1:
 - # ip route add(del) default via 10.10.14.4



```
h1
root@TCPIP-VM:~/Desktop/shared# ip route
10.10.14.0/24 dev h1-eth0 proto kernel scope link src 10.10.14.1
root@TCPIP-VM:~/Desktop/shared# ip route add default via 10.10.14.4
root@TCPIP-VM:~/Desktop/shared# ip route
default via 10.10.14.4 dev h1-eth0
10.10.14.0/24 dev h1-eth0 proto kernel scope link src 10.10.14.1
root@TCPIP-VM:~/Desktop/shared#
```

Convert into router

- Convert h4 into a router:
 - # echo 1 > /proc/sys/net/ipv4/ip_forward



```
h1
root@TCPIP-VM:~/Desktop/shared# ping 10.10.24.2
PING 10.10.24.2 (10.10.24.2) 56(84) bytes of data:
64 bytes from 10.10.24.2: icmp_seq=1 ttl=63 time=7.85 ms
64 bytes from 10.10.24.2: icmp_seq=2 ttl=63 time=1.16 ms
64 bytes from 10.10.24.2: icmp_seq=3 ttl=63 time=0.116 ms
64 bytes from 10.10.24.2: icmp_seq=4 ttl=63 time=0.108 ms
^C
--- 10.10.24.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.108/2.310/7.850/3.227 ms
root@TCPIP-VM:~/Desktop/shared#
```