Network devices

OSI model

7. Application layer

    Consists of application programs that use the network.

6. Presentation layer

    Standardizes data presentation to the applications that use the network.

5. Session layer

    Manages sessions between applications.

4. Transport layer

    Provides end-to-end error detection and correction.

3. Network layer

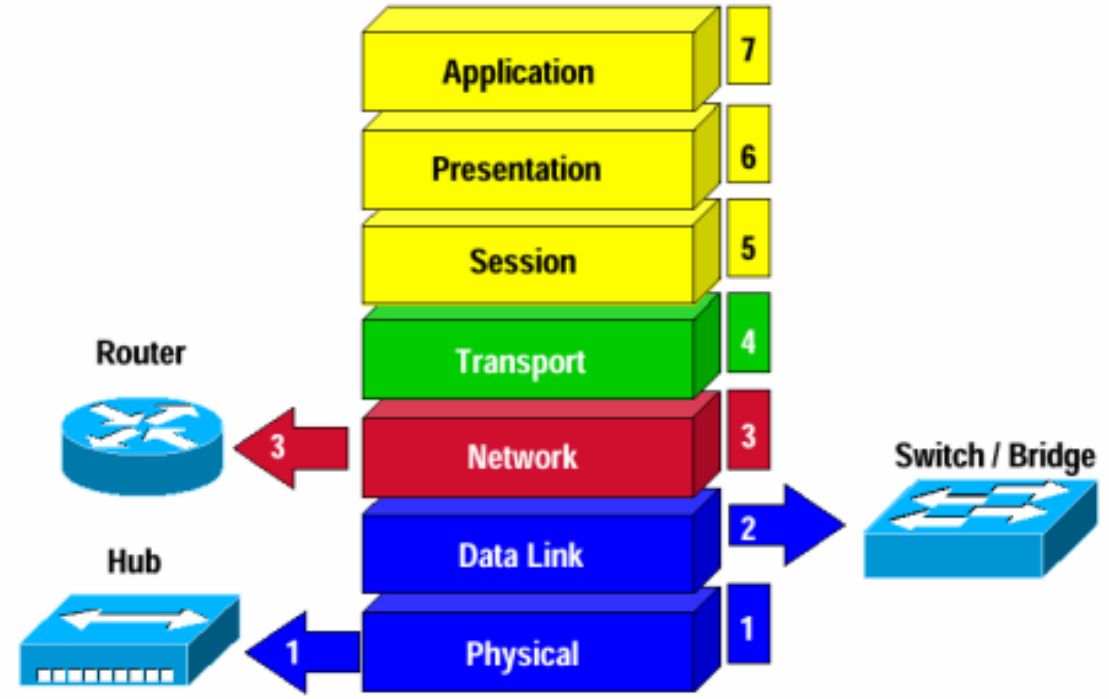
    Decides which physical path the data will take.

2. Data link layer

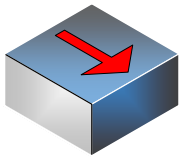
    Provides reliable data delivery across the physical link.

1. Physical layer

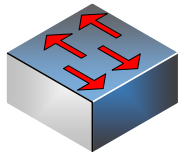
    Defines the physical characteristics of the network media.



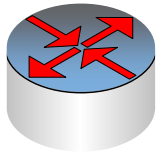
Network devices



- Hub
  - Broadcast packets

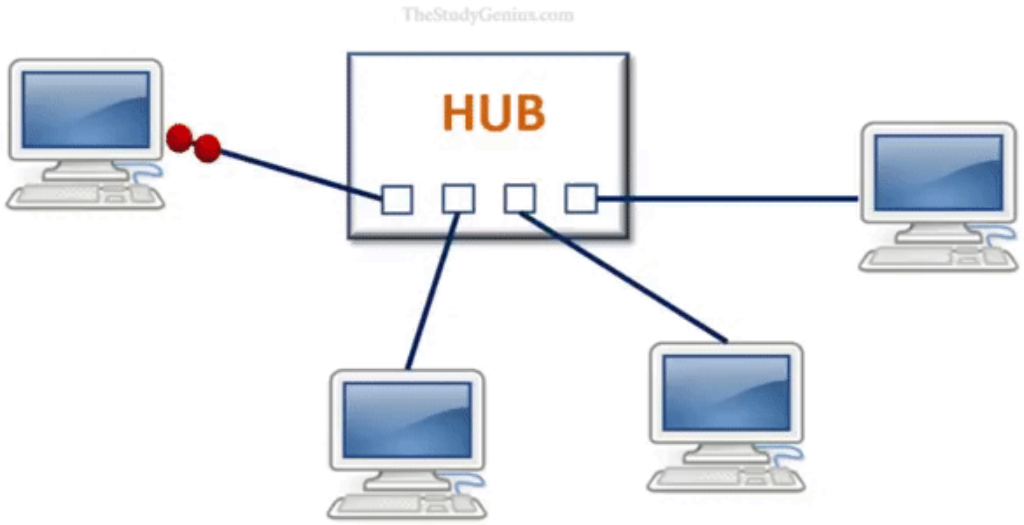


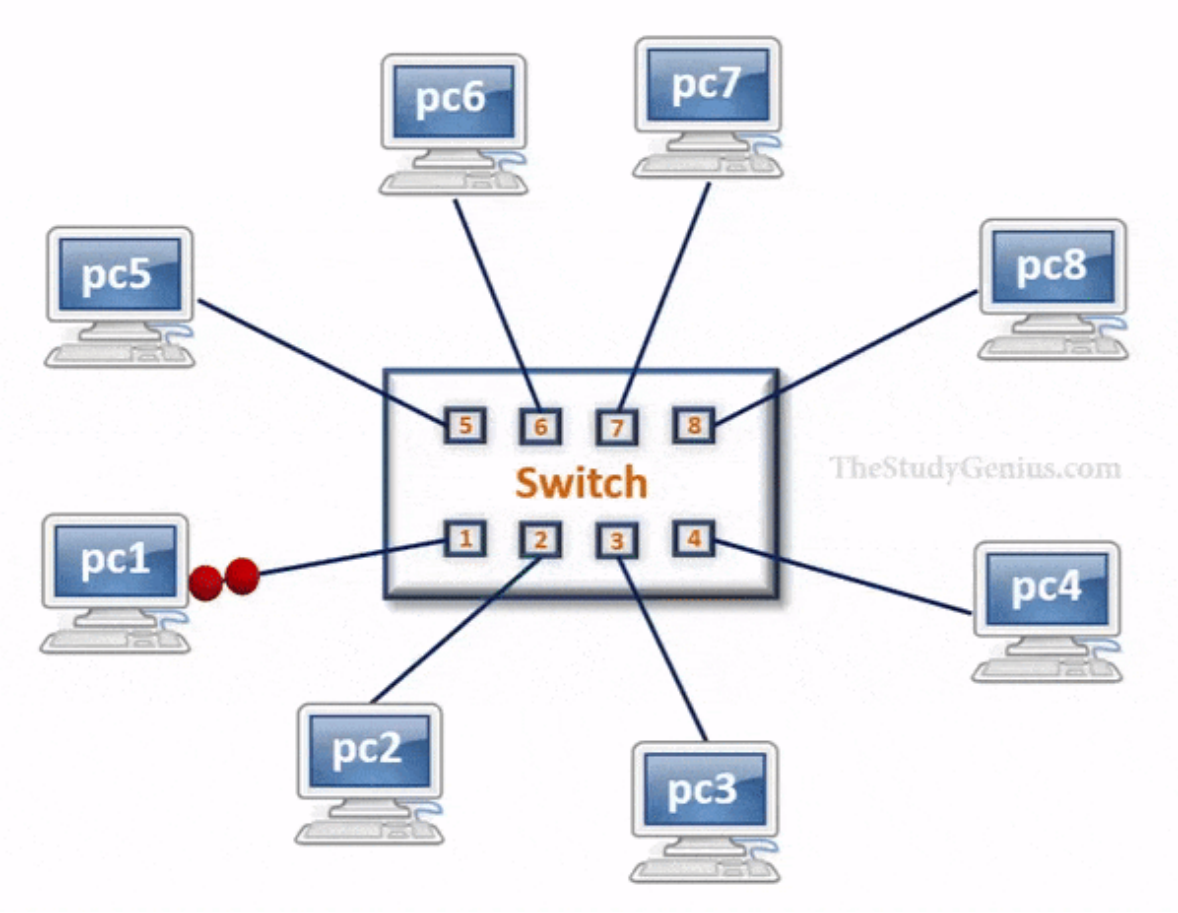
- Switch
  - Forward packet to the destination interface



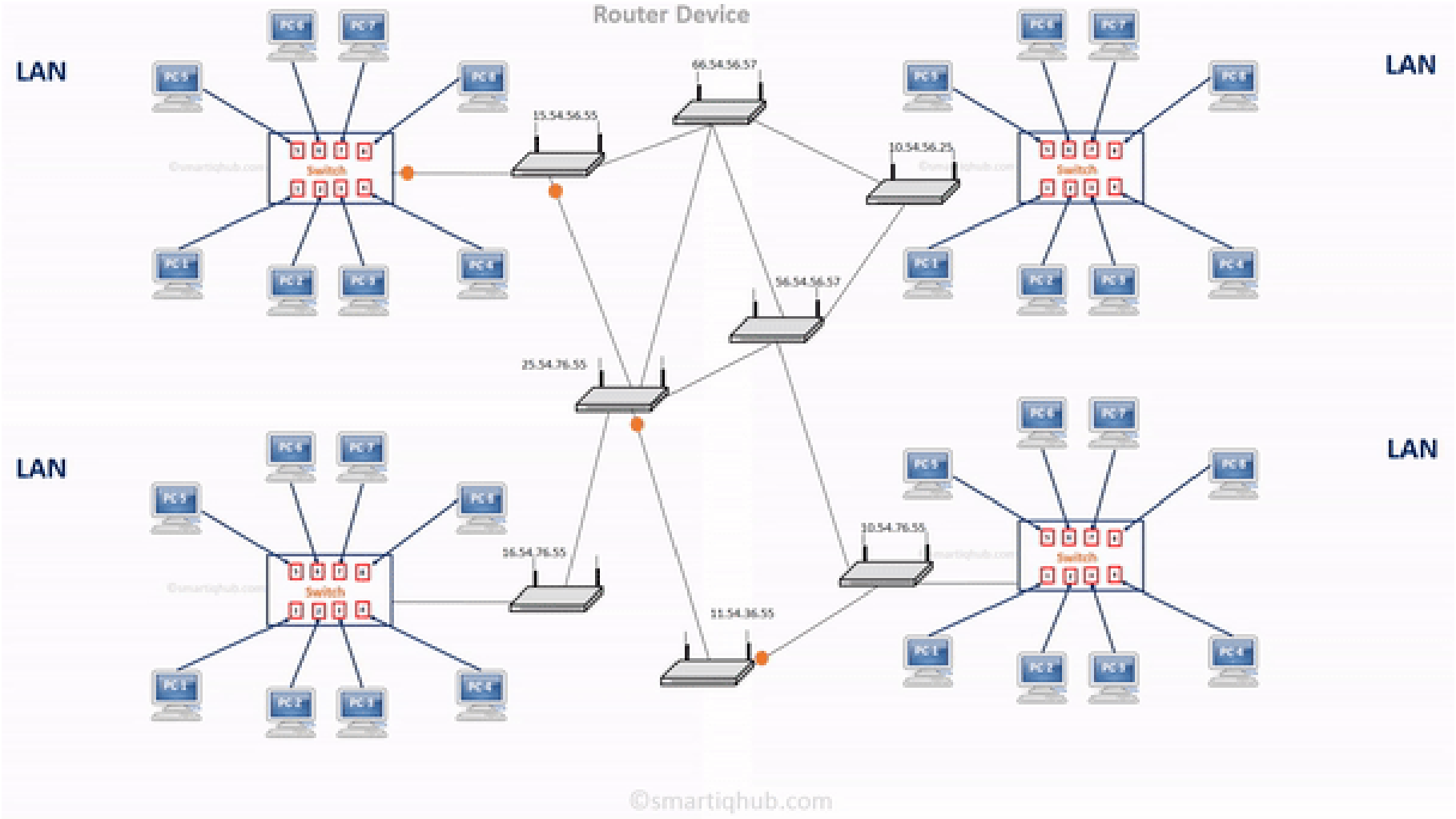
- Router
  - Change source MAC address and then forward packet to the destination interface

Hub vs. Switch

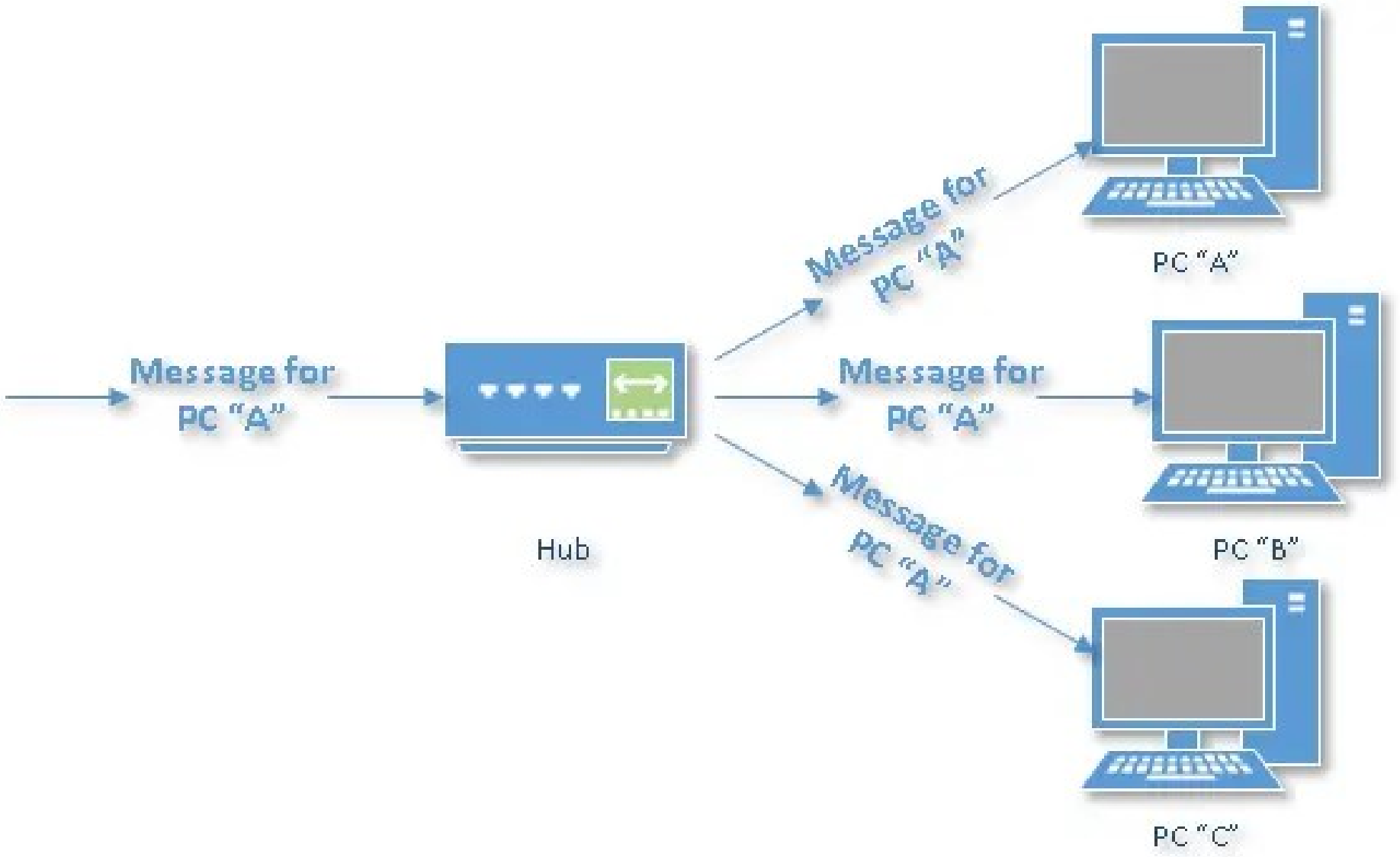




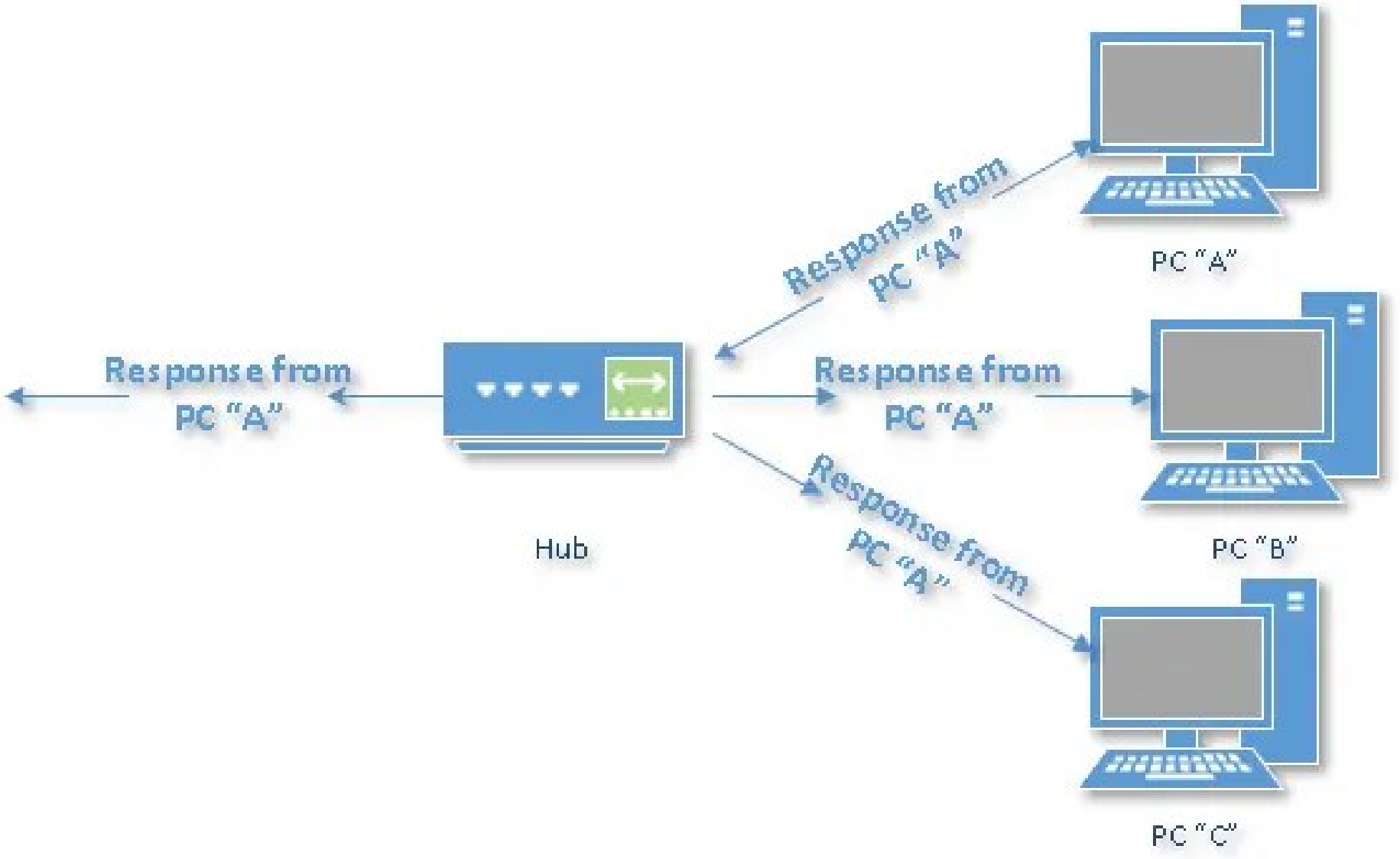
Routers


Router Device
LAN
LAN
LAN
LAN
Switch
Switch
Switch
Switch
66.54.56.57
15.54.56.55
10.54.56.25
56.54.56.57
25.54.76.55
16.54.76.55
10.54.76.55
11.54.36.55
©smartiqhub.com

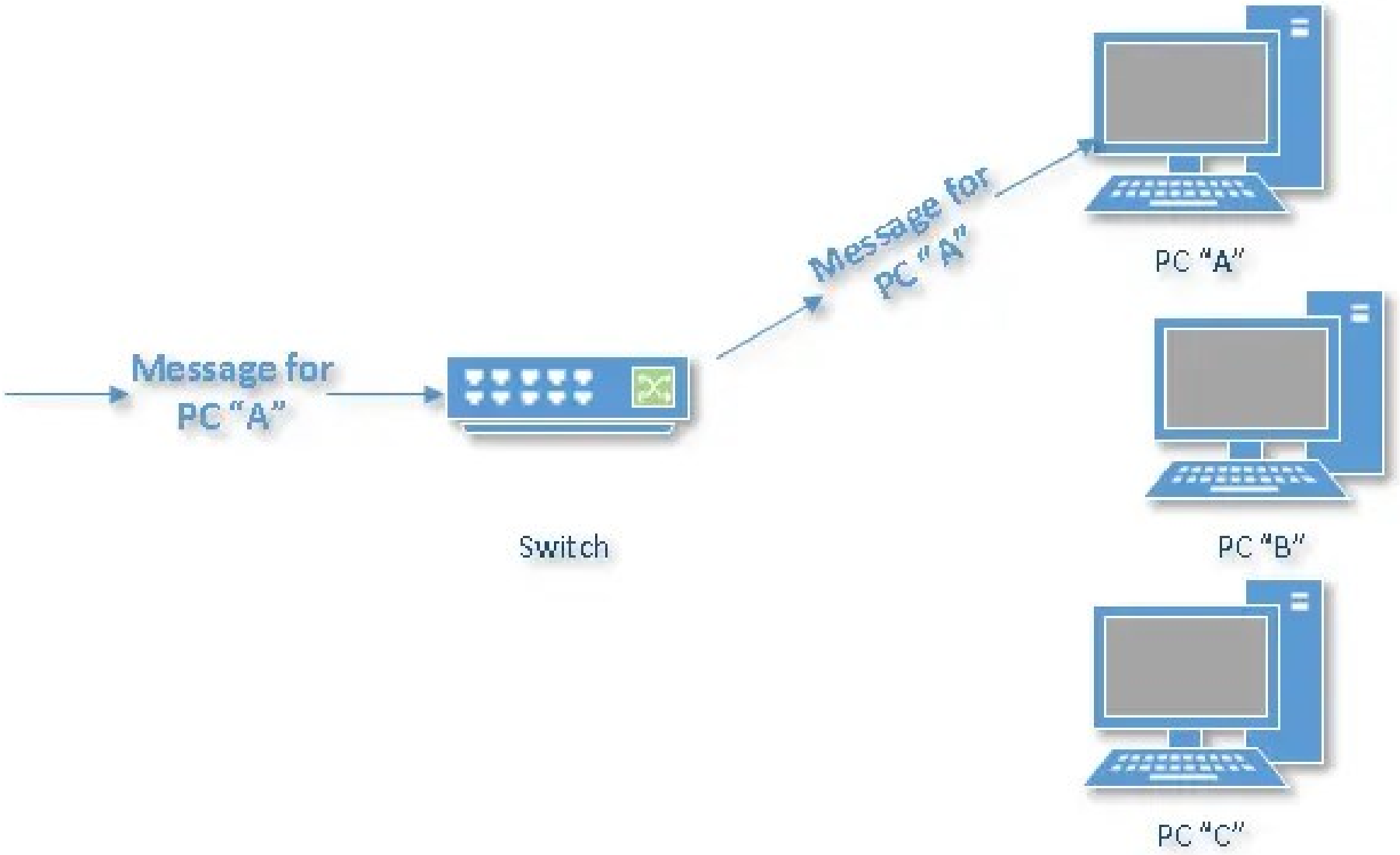
Hub


Message for PC "A"
Hub
Message for PC "A"
Message for PC "A"
Message for PC "A"
PC "A"
PC "B"
PC "C"

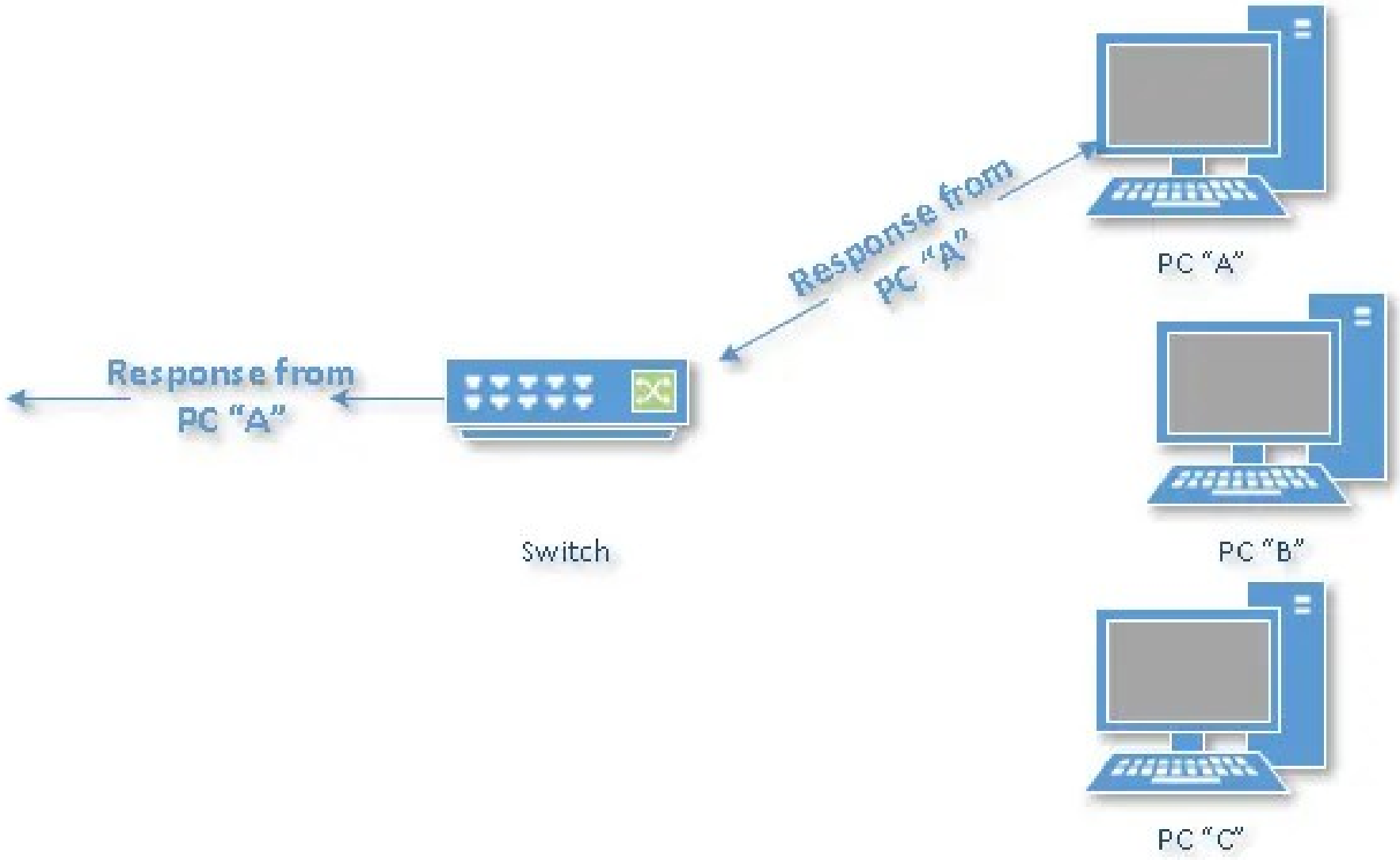
Hub


Response from
PC "A"
PC "A"
Response from
PC "A"
Response from
PC "A"
Hub
Response from
PC "A"
PC "B"
PC "C"

Switch



Switch

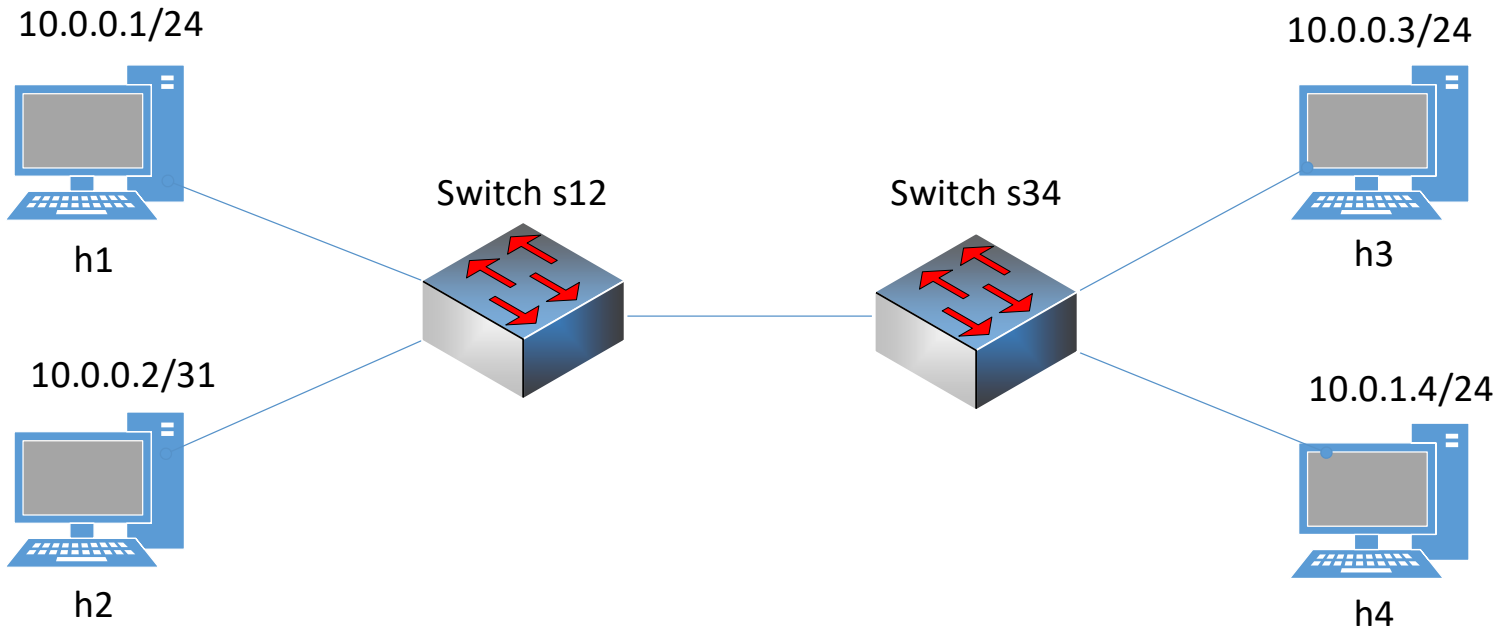


Comparison

Hub	Switch	Router
HUB work on Physical Layer of OSI Model	Switch work on Data Link Layer of OSI Model	Router work on Network Layer of OSI Model
HUB is Broadcast Device	Switch is Unicast/Multicast Device	Router is a routing device use to create route for transmitting data packets
Hus is use to connect device in the same network	Switch is use to connect devices in the same network	Router is use to connect two or more different network.
Hub sends data in the form of binary bits	Switch sends data in the form of frames	Router sends data in the form packets
Hub only works in half duplex	Switch works in full duplex	Router works in full duplex
Only one device can send data at a time	Multiple devices can send data at the same time	Multiple devices can send data at the same time
Hub does not store any mac address or IP address	Switch store MAC Address	Router stores IP address

Hub function

- $ cd Desktop/shared
- $ sudo python lab3-1.py



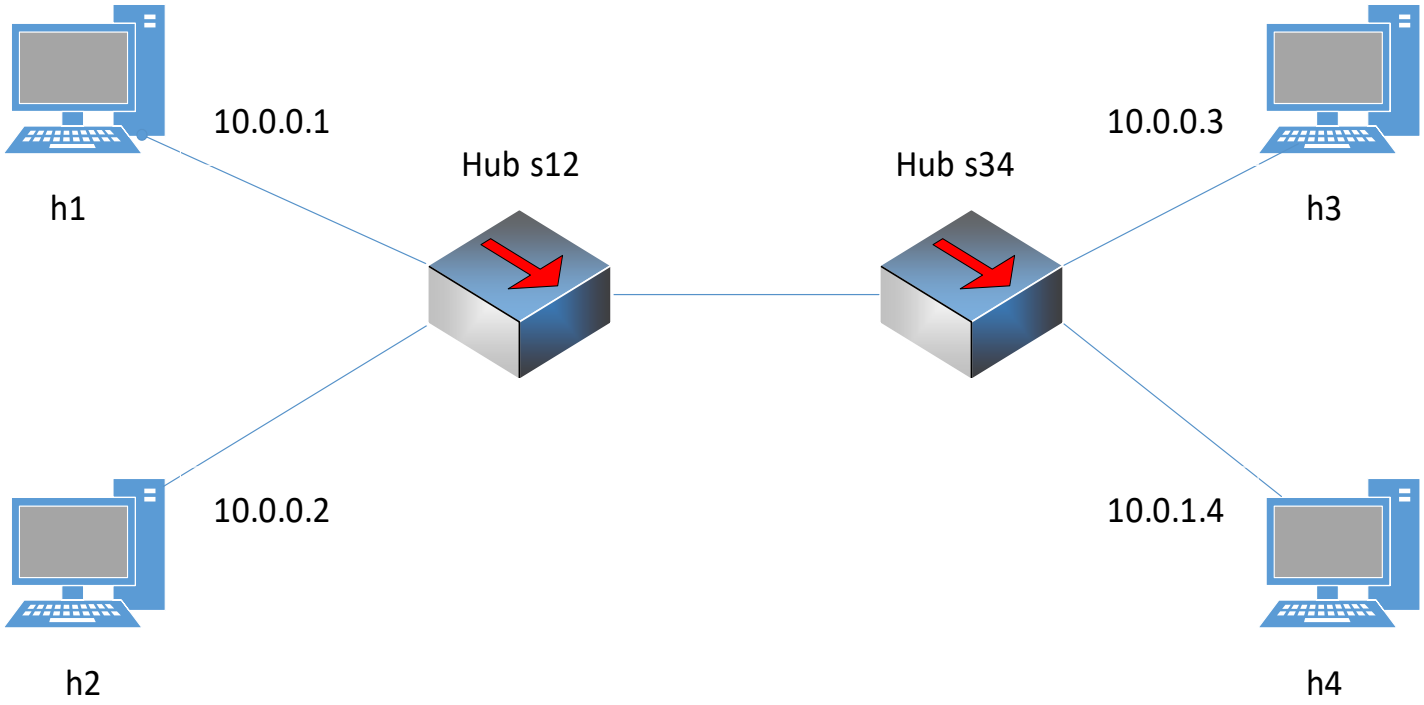
- # ifconfig
- Lan1: h1, h2, h3
- Lan2: h4

- Correct subnet mask of h2-eth0:
  - # ifconfig h2-eth0 10.0.0.2 netmask 255.255.255.0

Hub function

- Configure s12 and s34 as hubs:
  - mininet> sh ovs-ofctl add-flow s12 action=flood
  - mininet> sh ovs-ofctl add-flow s34 action=flood

- h1 ping h2

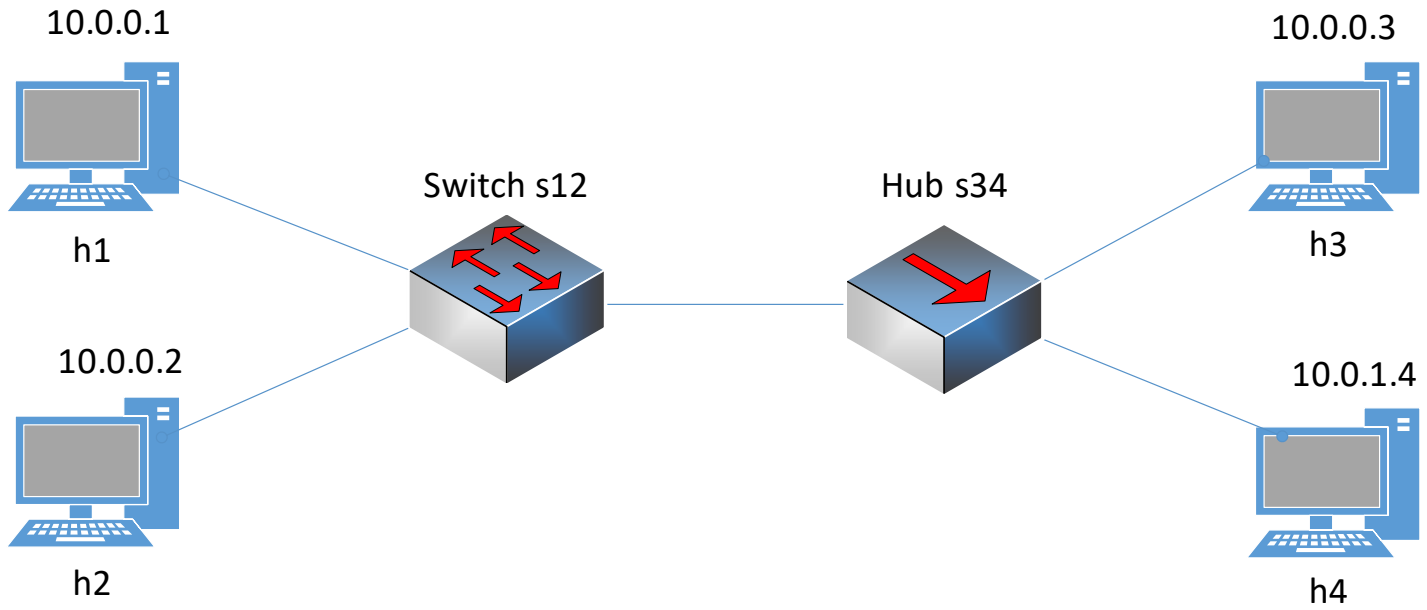


Switch function

- Return s12 to a switch:
  - mininet> sh ovs-ofctl add-flow s12 action=normal

- Delete a specified entry of ARP cache:
  - # arp -d 10.0.0.2

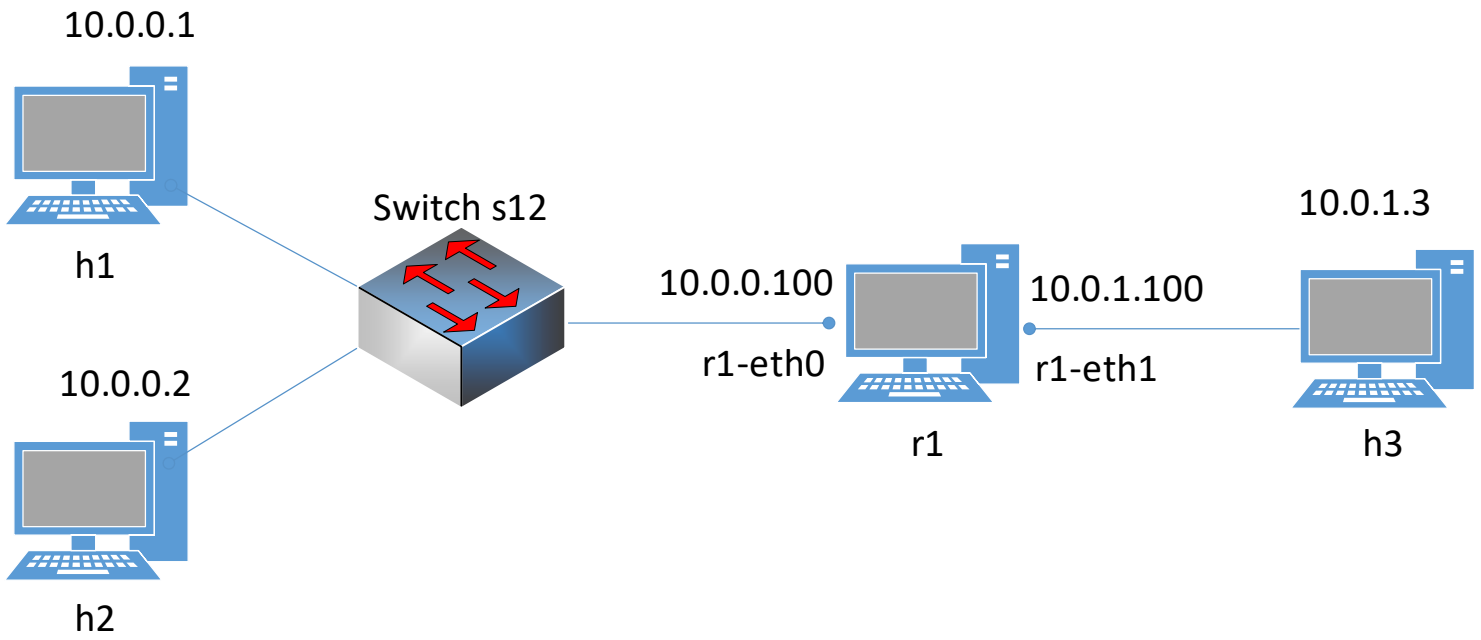
- h1 ping h2
- h1 ping h3
- h1 ping h4



Router function

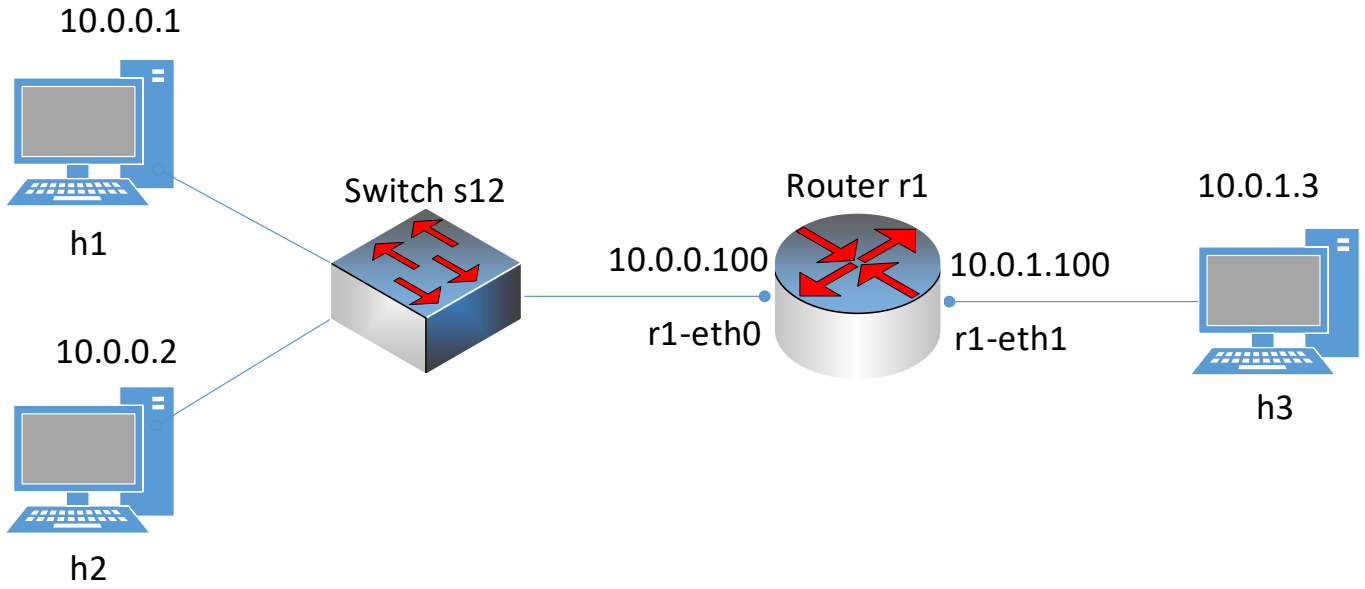
- mininet> exit
- $ sudo mn -c
- $ sudo python lab3-2.py

- mininet> pingall



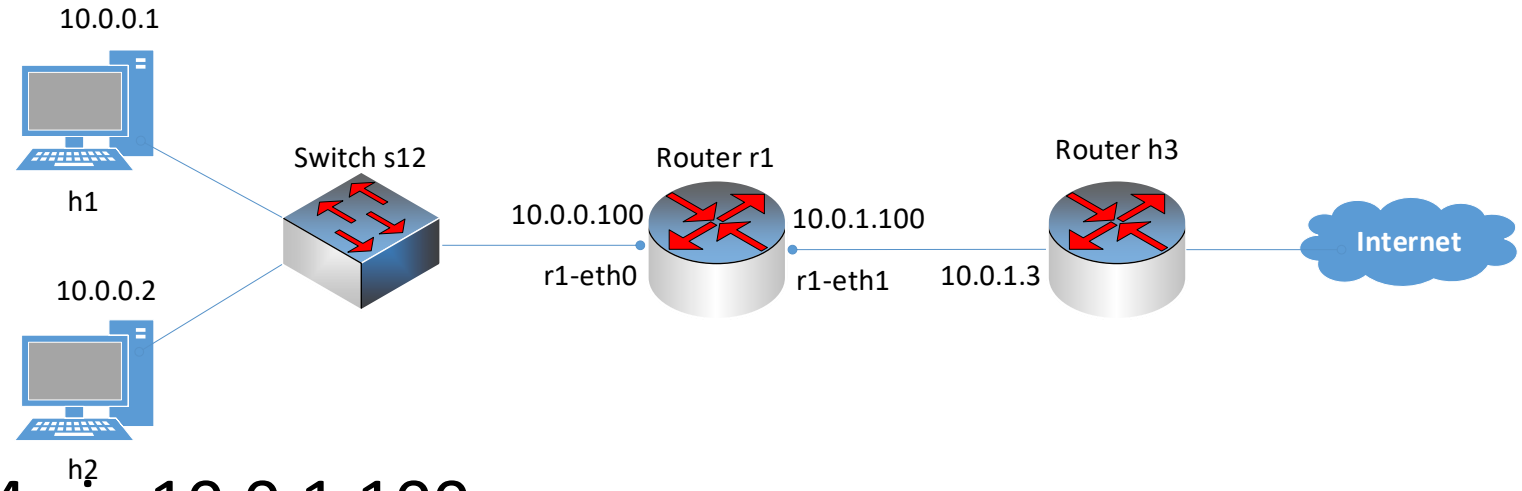
Router function

- Enable IP forwarding on r1:
  - # echo 1 > /proc/sys/net/ipv4/ip_forward
- mininet> pingall
- Show gateway of each host:
  - # ip route
- Correct gateway:
  - # ip route del default via 10.0.0.101
  - # ip route add default via 10.0.0.100

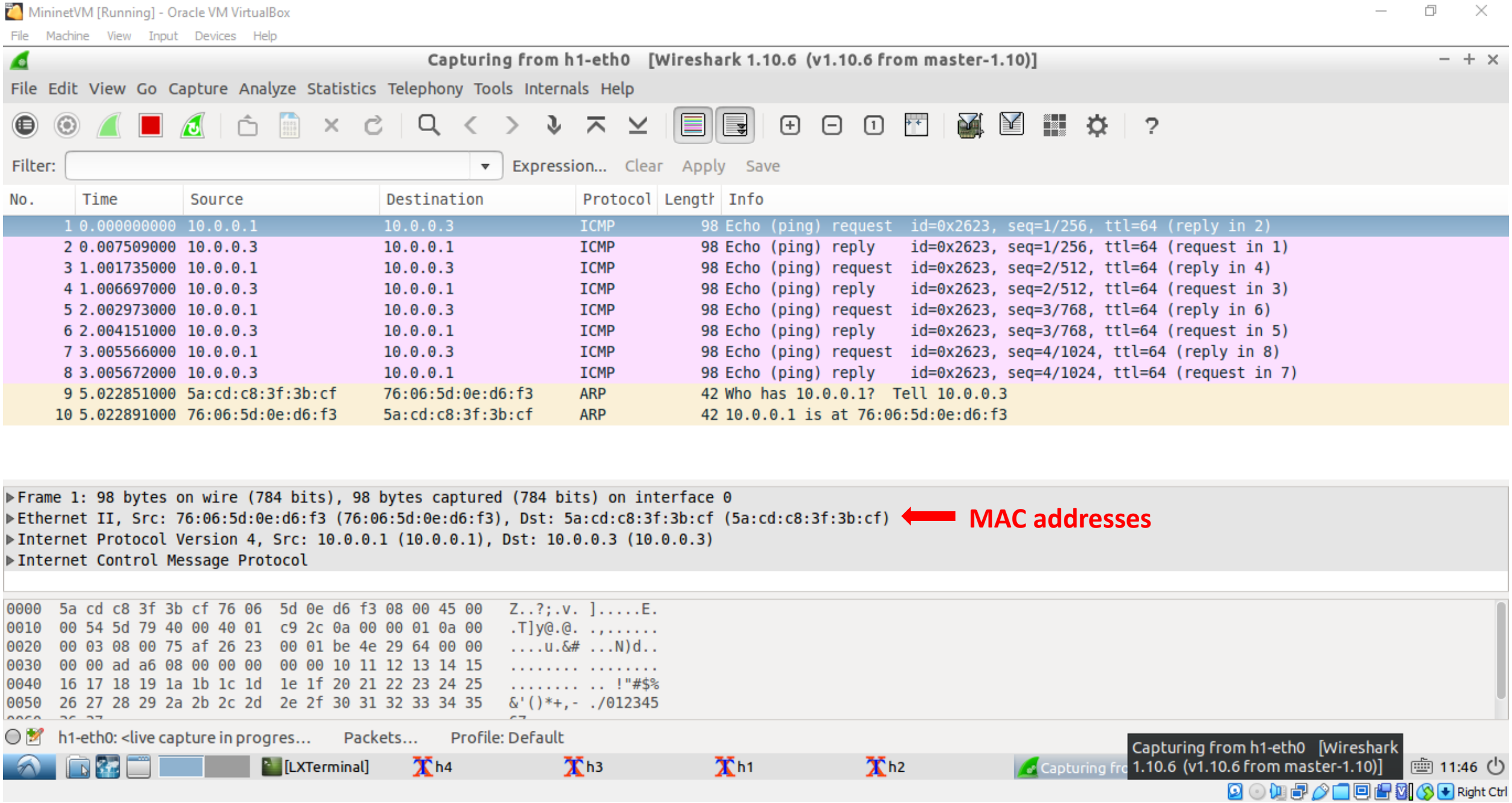


Routing with multiple hops

- For h3:
  - # echo 1 > /proc/sys/net/ipv4/ip_forward
  - # ip route del default via 10.0.1.100
- For r1:
  - # ip route add default via 10.0.1.3



- h1 ping h3
- h2 ping h3
- # ip route add 10.0.0.0/24 via 10.0.1.100


MininetVM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Capturing from h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.0.1 10.0.0.3 ICMP 98 Echo (ping) request id=0x2623, seq=1/256, ttl=64 (reply in 2)
2 0.007509000 10.0.0.3 10.0.0.1 ICMP 98 Echo (ping) reply id=0x2623, seq=1/256, ttl=64 (request in 1)
3 1.001735000 10.0.0.1 10.0.0.3 ICMP 98 Echo (ping) request id=0x2623, seq=2/512, ttl=64 (reply in 4)
4 1.006697000 10.0.0.3 10.0.0.1 ICMP 98 Echo (ping) reply id=0x2623, seq=2/512, ttl=64 (request in 3)
5 2.002973000 10.0.0.1 10.0.0.3 ICMP 98 Echo (ping) request id=0x2623, seq=3/768, ttl=64 (reply in 6)
6 2.004151000 10.0.0.3 10.0.0.1 ICMP 98 Echo (ping) reply id=0x2623, seq=3/768, ttl=64 (request in 5)
7 3.005566000 10.0.0.1 10.0.0.3 ICMP 98 Echo (ping) request id=0x2623, seq=4/1024, ttl=64 (reply in 8)
8 3.005672000 10.0.0.3 10.0.0.1 ICMP 98 Echo (ping) reply id=0x2623, seq=4/1024, ttl=64 (request in 7)
9 5.022851000 5a:cd:c8:3f:3b:cf 76:06:5d:0e:d6:f3 ARP 42 Who has 10.0.0.1? Tell 10.0.0.3
10 5.022891000 76:06:5d:0e:d6:f3 5a:cd:c8:3f:3b:cf ARP 42 10.0.0.1 is at 76:06:5d:0e:d6:f3
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: 76:06:5d:0e:d6:f3 (76:06:5d:0e:d6:f3), Dst: 5a:cd:c8:3f:3b:cf (5a:cd:c8:3f:3b:cf)
Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.0.3 (10.0.0.3)
Internet Control Message Protocol
MAC addresses
0000 5a cd c8 3f 3b cf 76 06 5d 0e d6 f3 08 00 45 00 Z..?;.v.].....E.
0010 00 54 5d 79 40 00 40 01 c9 2c 0a 00 00 01 0a 00 .T]y@.@. .,......
0020 00 03 08 00 75 af 26 23 00 01 be 4e 29 64 00 00 ....u.&# ...N)d..
0030 00 00 ad a6 08 00 00 00 00 00 10 11 12 13 14 15 ....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 .... !"#$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,- ./012345
h1-eth0: <live capture in progres... Packets... Profile: Default
[LXTerminal] h4 h3 h1 h2
Capturing from h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
11:46
Right Ctrl