IP Associated Services - NAT, ARP and ICMP

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Rajat Subhra Chakraborty
rschakraborty@cse.iitkgp.ac.in

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

Private IP Addresses

- Problem of IPv4 -- number of IP addresses are limited, number of network devices are growing over time
  - IPv6 was introduced to solve many such problems associated with IPv4
  - However, the major limitation of IPv6 is its direct backward compatibility with IPv4 -- therefore, we never seen a global deployment of IPv6
- Remedy that has been accepted widely -- reuse “private IPv4 addresses”
- Private IPv4 addresses
  - 10.0.0.0/8

Private IP Addresses

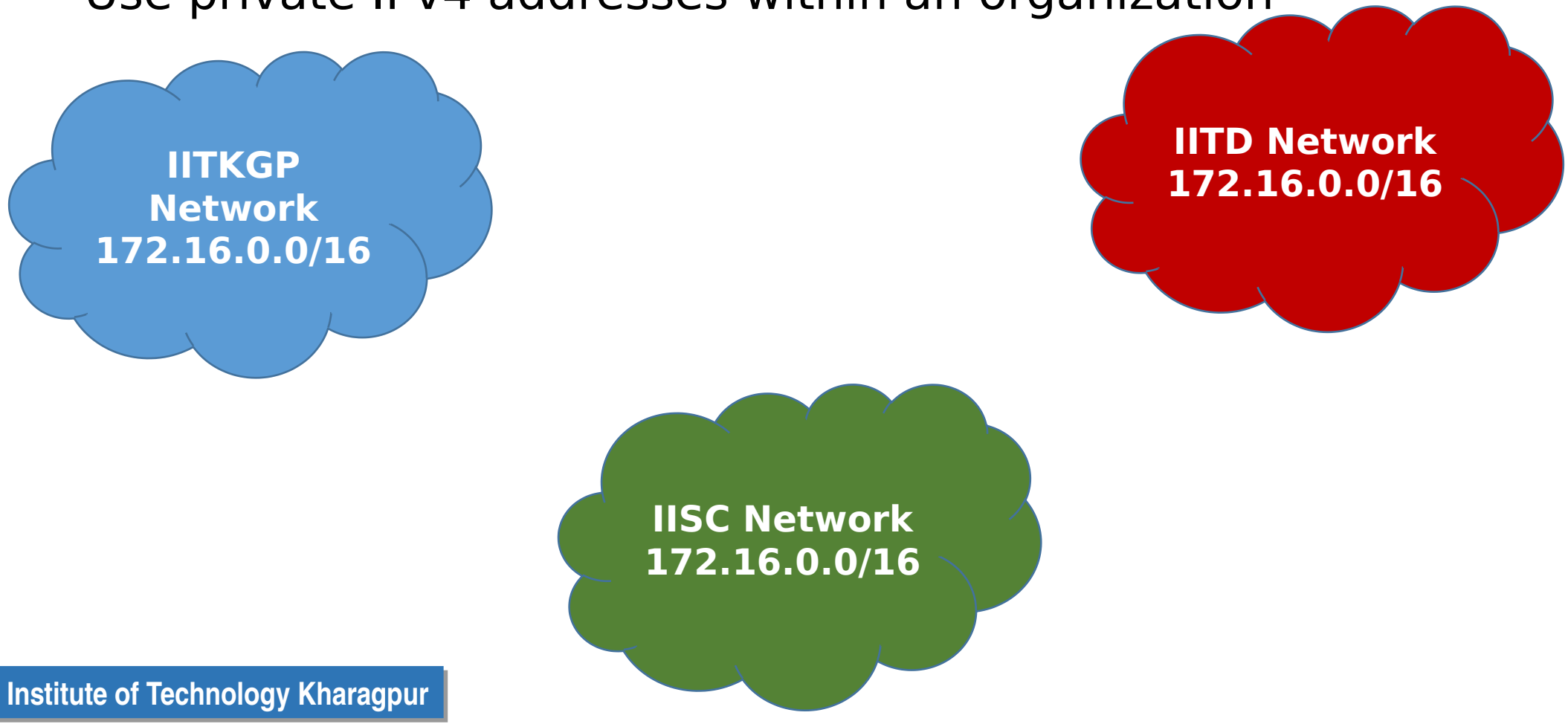
- Private IPv4 addresses are meant for “private” networks
  - Example: You develop a small network which does not have any connectivity with the global Internet
  - As there is no connectivity with the global Internet, these block of IPv4 addresses are reusable -- you can have two private network with the same block of IPv4 addresses assigned to them

- A network interface **must** have a public IPv4 address to communicate with the global Internet
  - Why? -- to enable the routing of a data packet towards the proper destination

- How can we use the private IPv4 addresses to solve the problem of IPv4 address scarcity?

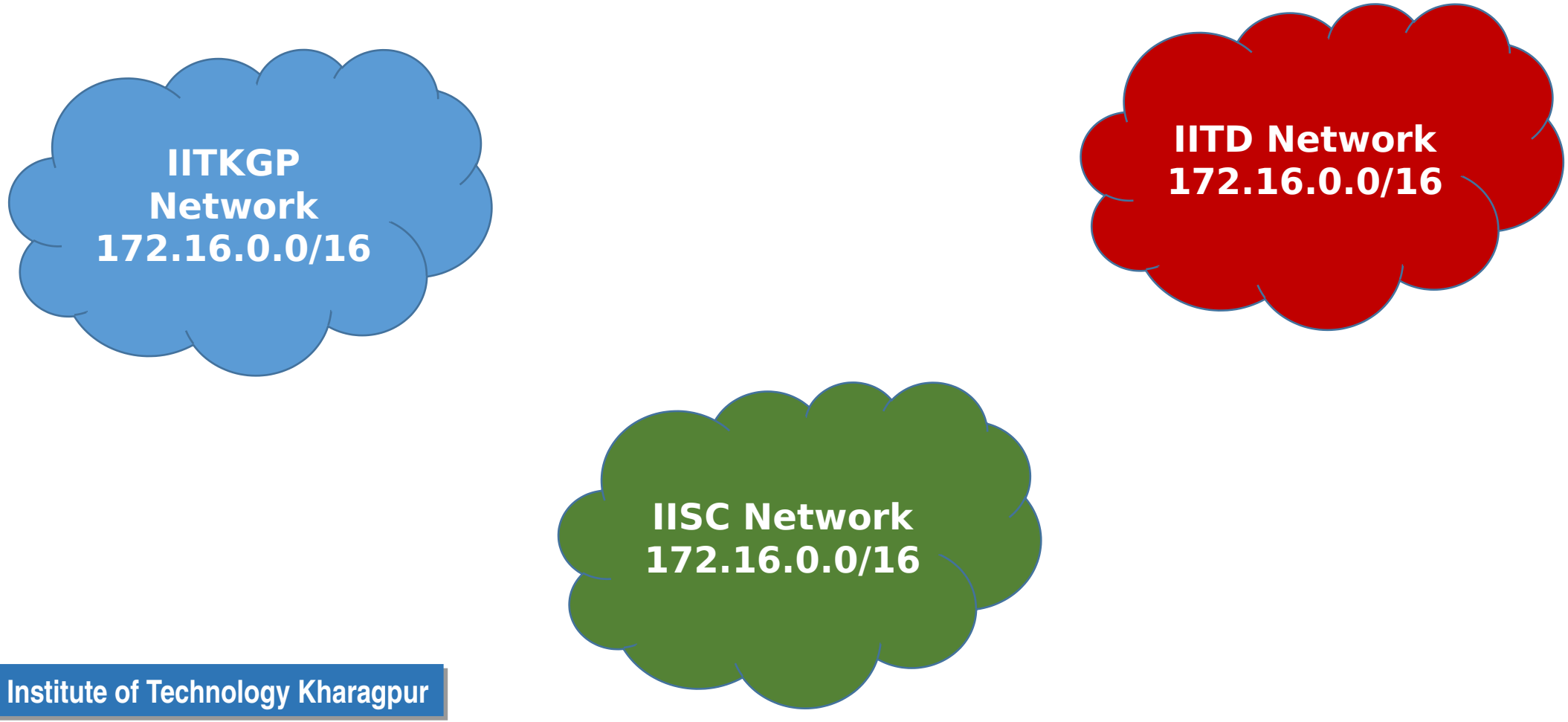
Interface Private IPv4 Addresses with Public Networks

- Create pockets of subnets with “reusable” private IPv4 addresses
  - Use private IPv4 addresses within an organization



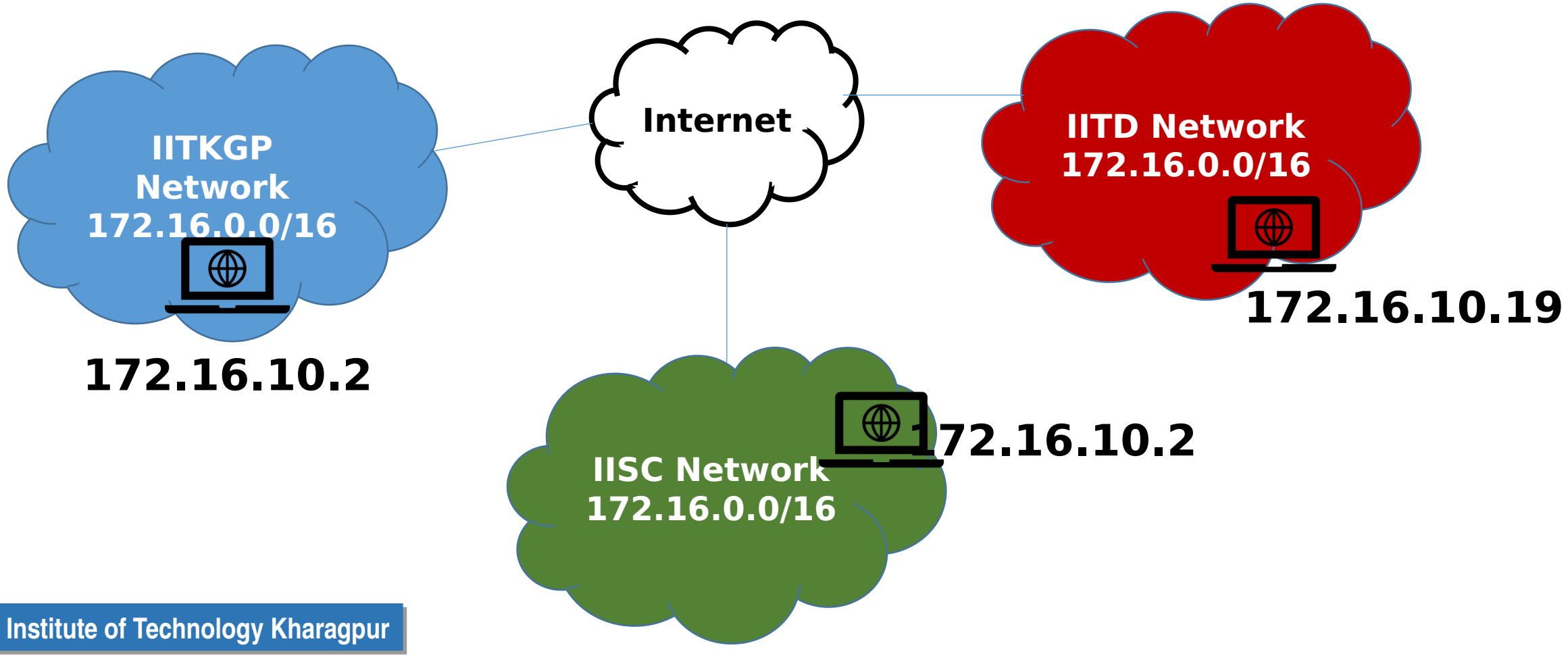
Interface Private IPv4 Addresses with Public Networks

- **Observations**: Not all the users in an organization are active simultaneously



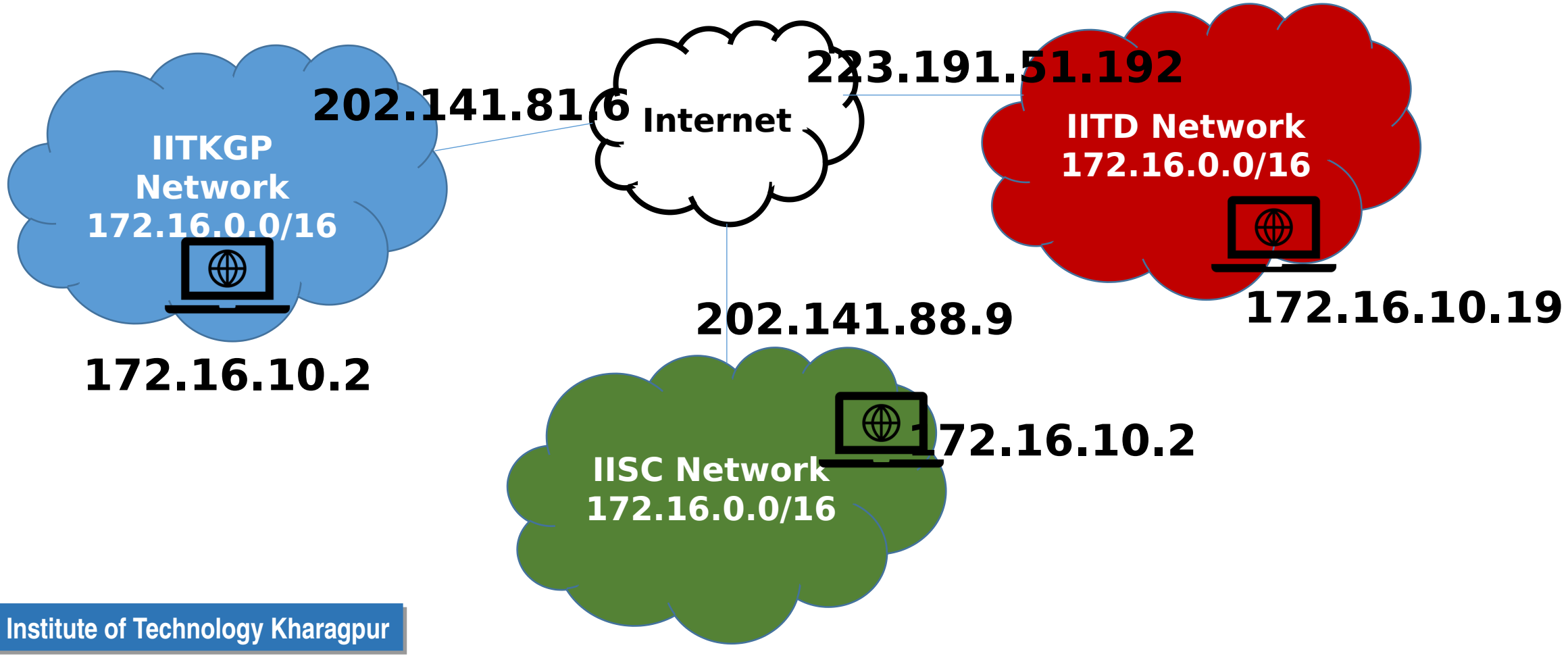
Interface Private IPv4 Addresses with Public Networks

- **Solution**: Map private IP addresses of the "**active**" users to corresponding public IP addresses



Interface Private IPv4 Addresses with Public Networks

- **Solution**: Map private IP addresses of the "**active**" users to corresponding public IP addresses

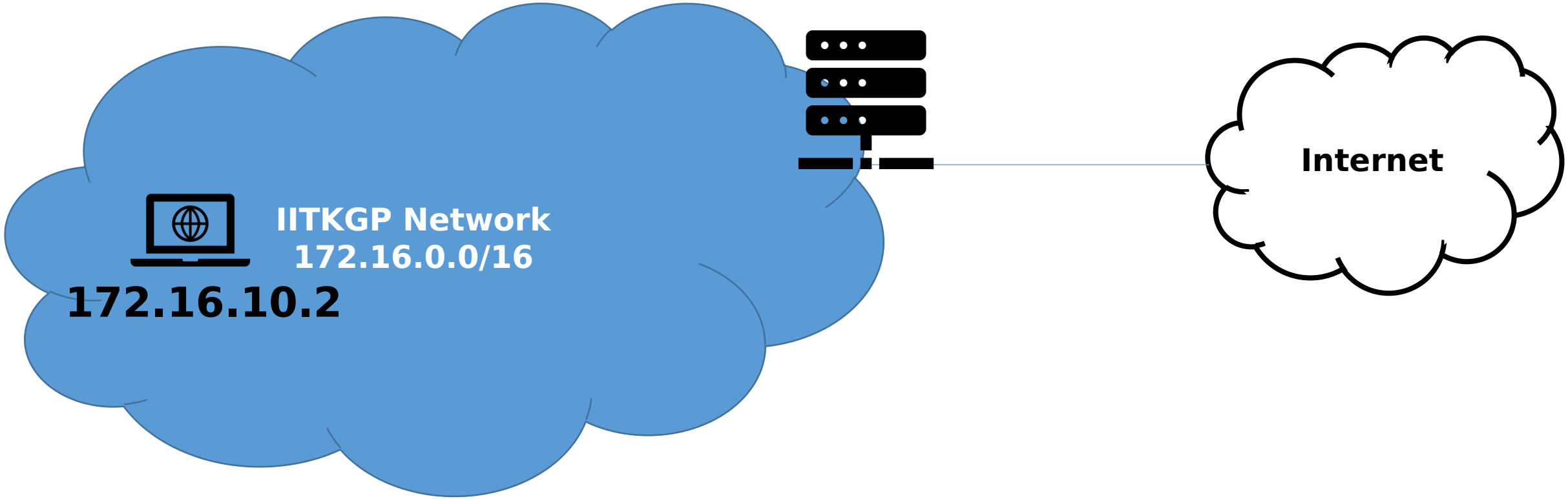


Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses**
  to public IP addresses and vice-versa

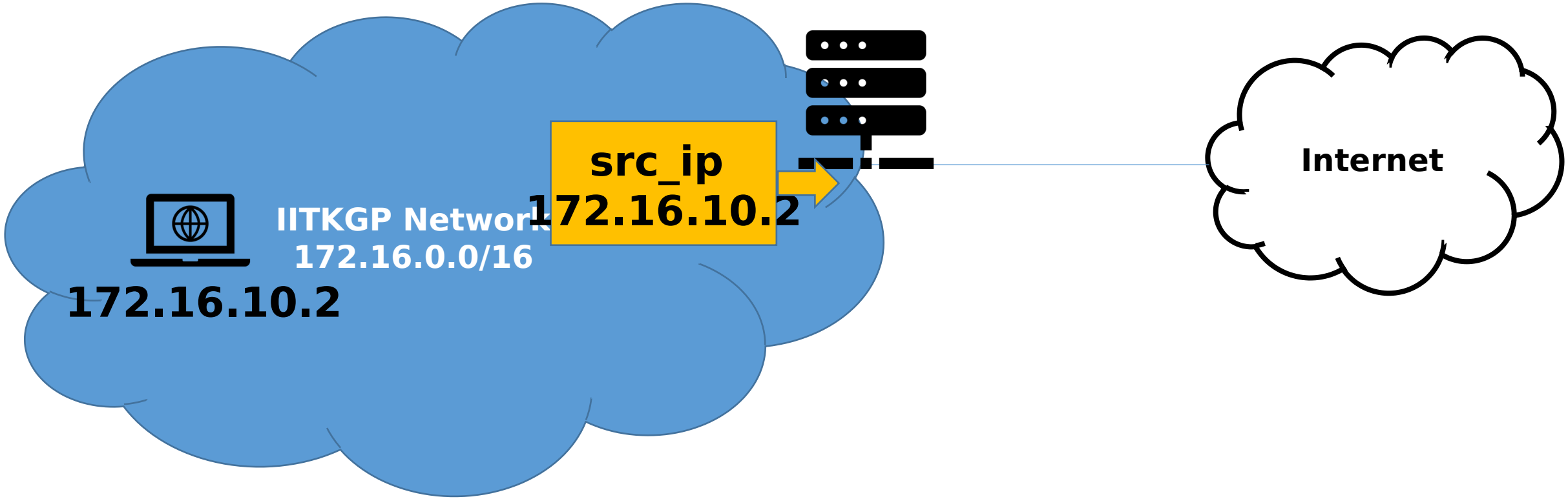
Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**



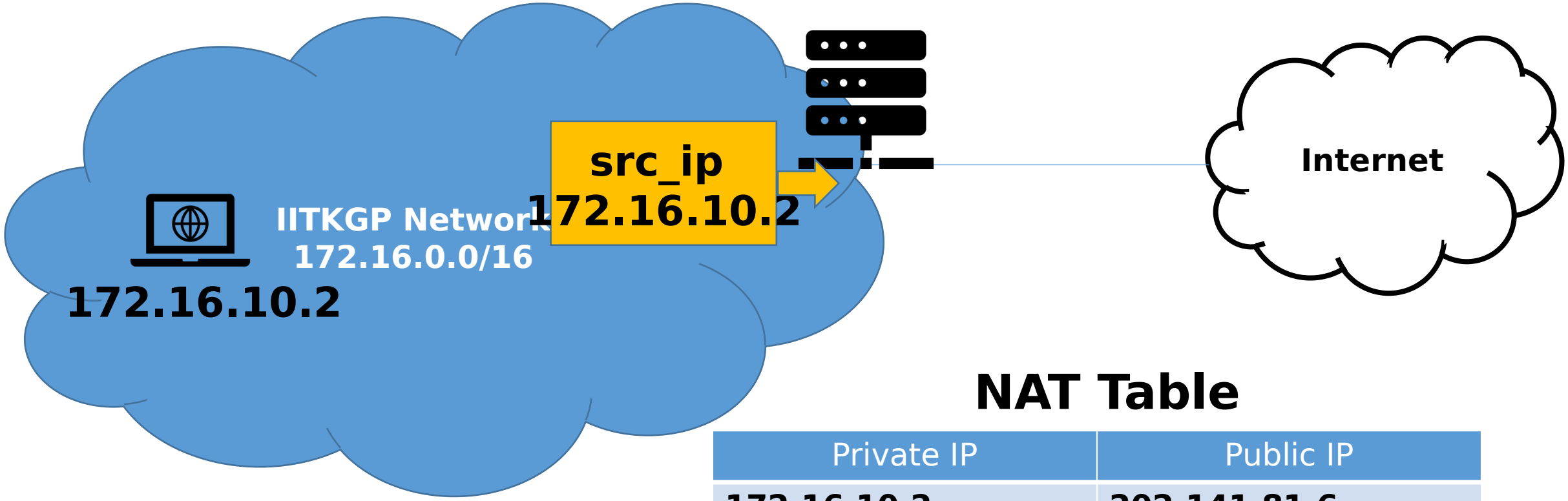
Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**



Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**

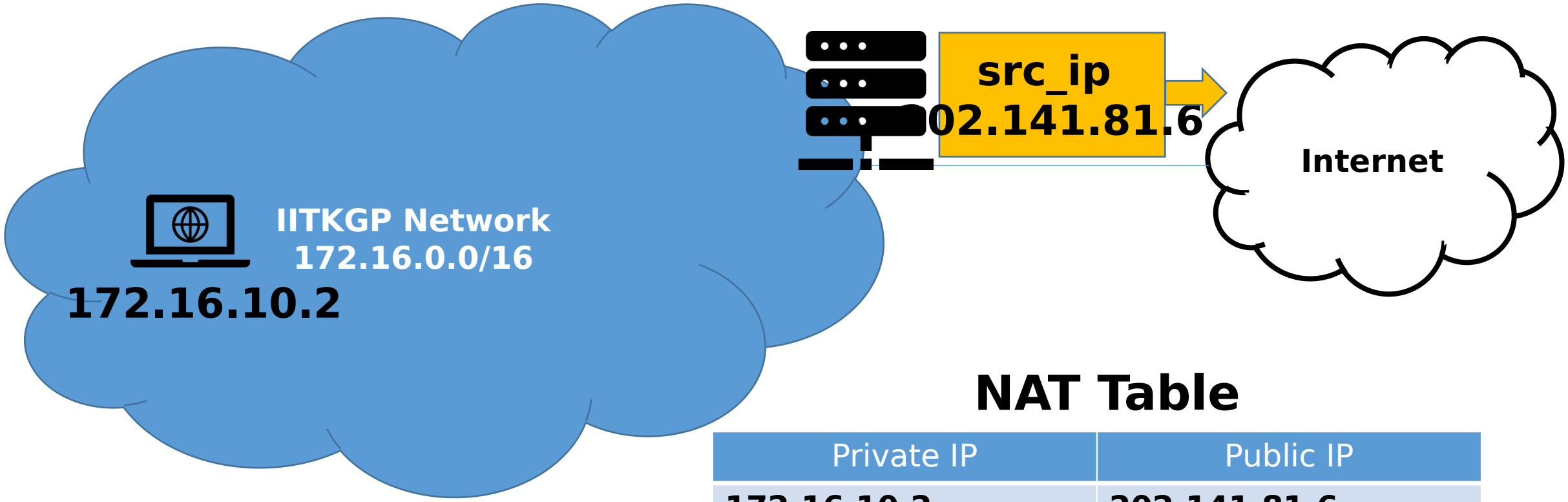


NAT Table

Private IP	Public IP
172.16.10.2	**202.141.81.6**

Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**

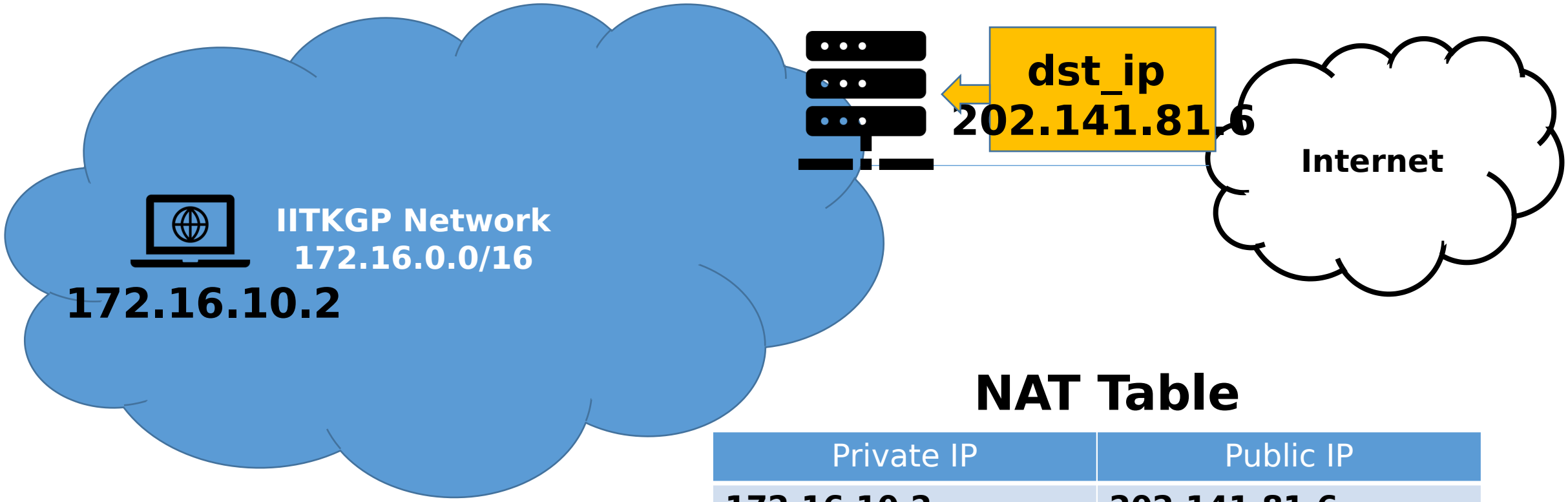


NAT Table

Private IP	Public IP
172.16.10.2	**202.141.81.6**

Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**

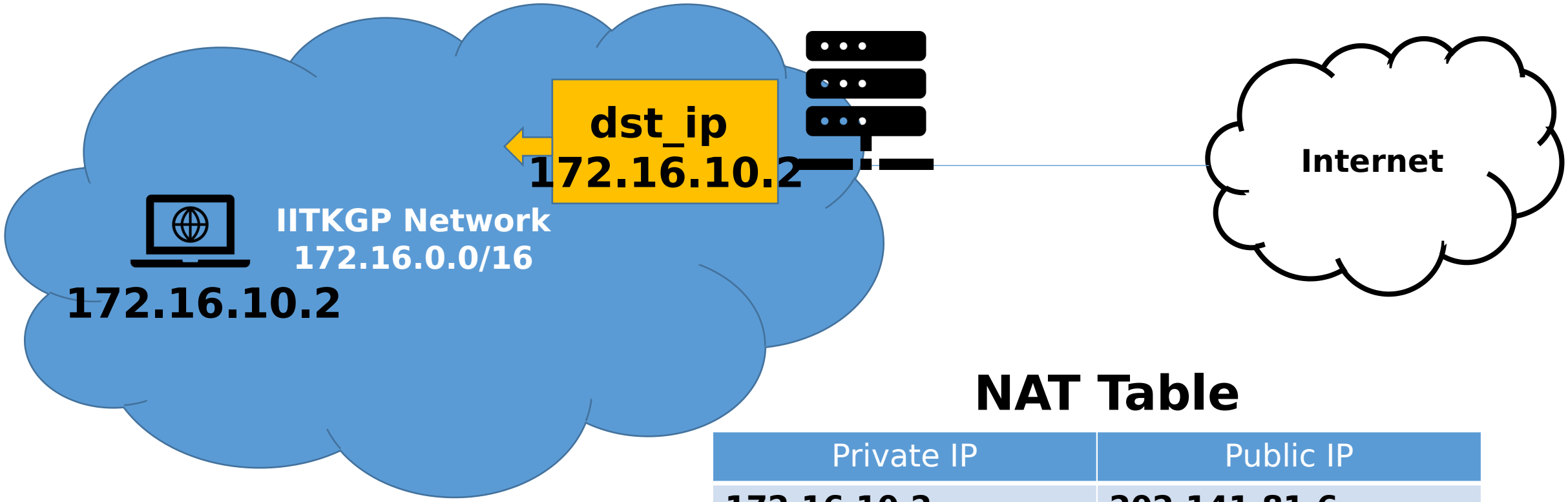


NAT Table

Private IP	Public IP
172.16.10.2	202.141.81.6

Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**

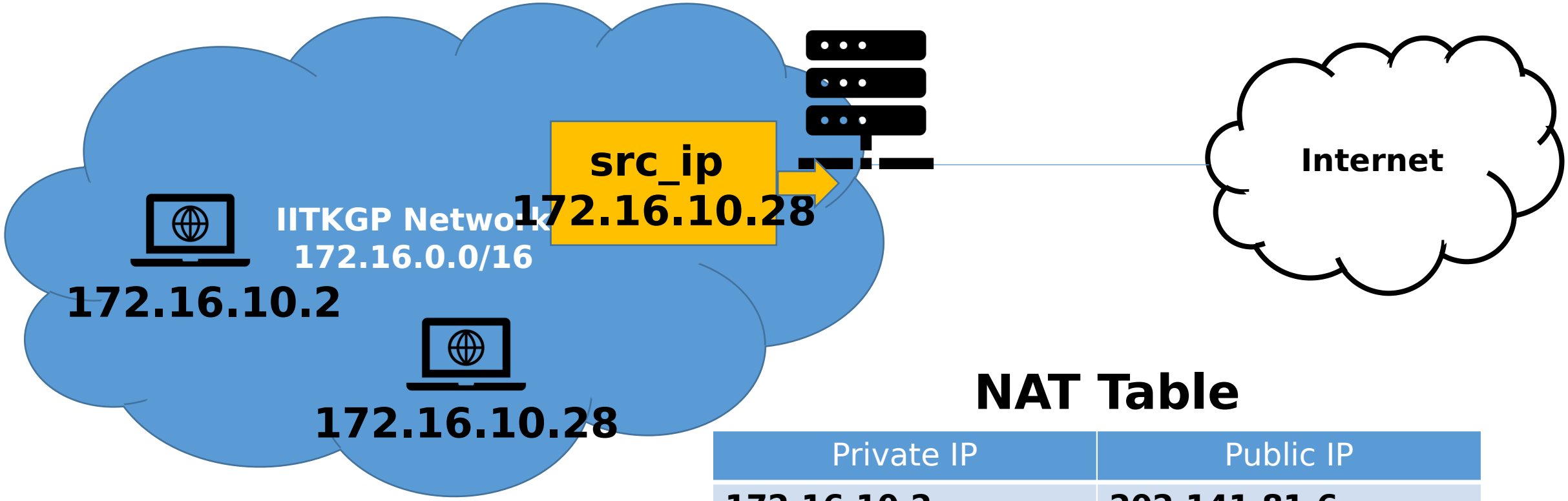


NAT Table

Private IP	Public IP
172.16.10.2	**202.141.81.6**

Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**

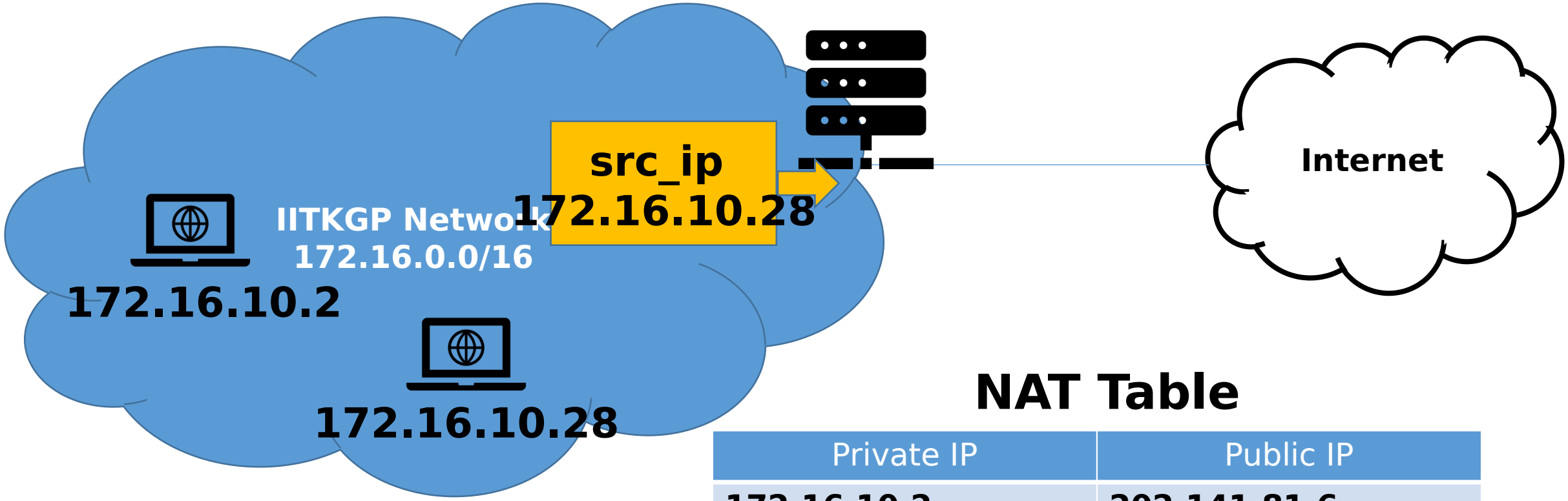


NAT Table

Private IP	Public IP
172.16.10.2	202.141.81.6

Network Address Translation (NAT)

- **An internet "middlebox" that translates private IP addresses to public IP addresses and vice-versa**

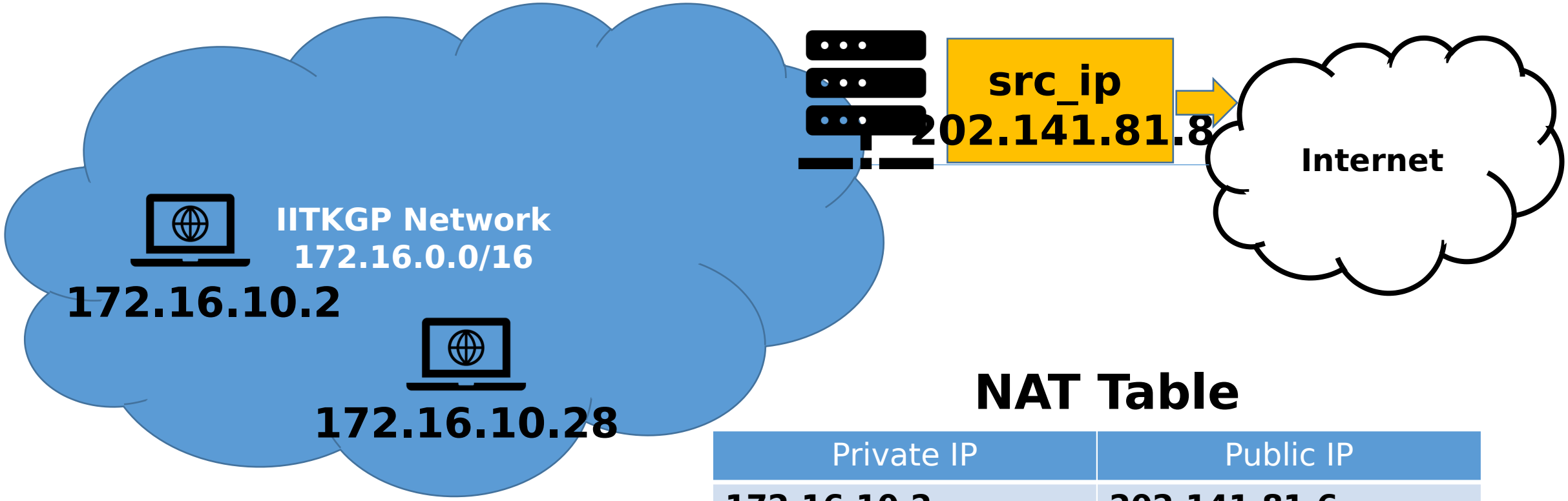


NAT Table

Private IP	Public IP
172.16.10.2	202.141.81.6
172.16.10.28	202.141.81.8

Network Address Translation (NAT)

- **An internet “middlebox” that translates private IP addresses to public IP addresses and vice-versa**

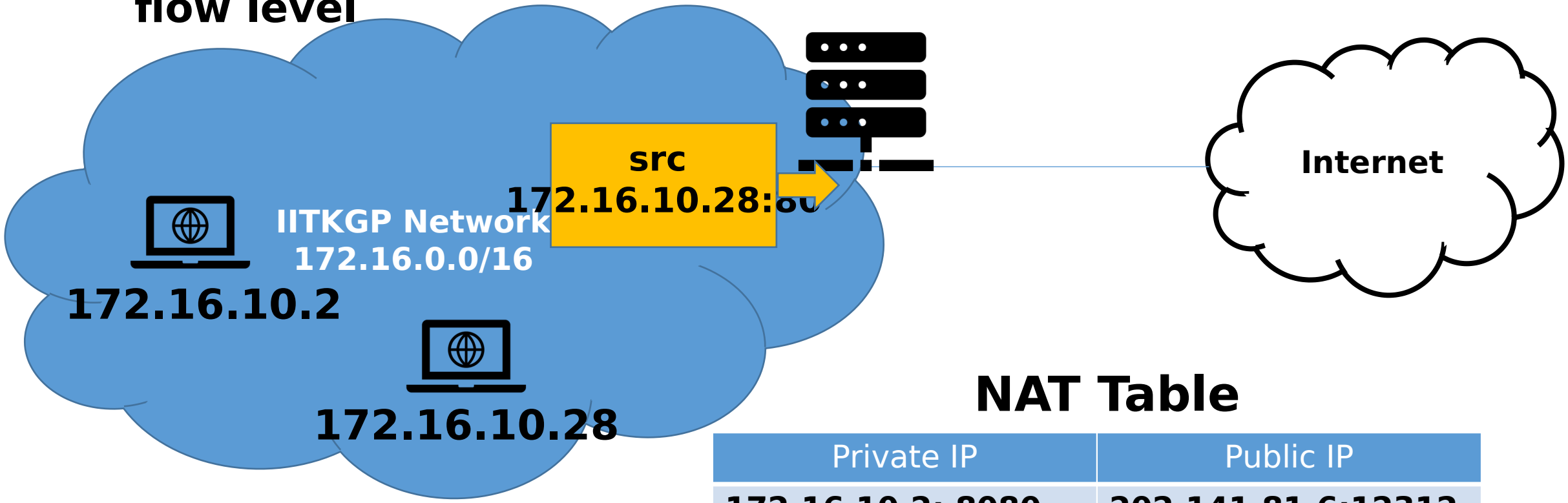


NAT Table

Private IP	Public IP
172.16.10.2	202.141.81.6
172.16.10.28	202.141.81.8

Port Based NAT (P-NAT)

- **Use a combination of IP address and the port number for mapping**
  - **The <IP, Port> pair defines an end-to-end flow -- so NAT at flow level**

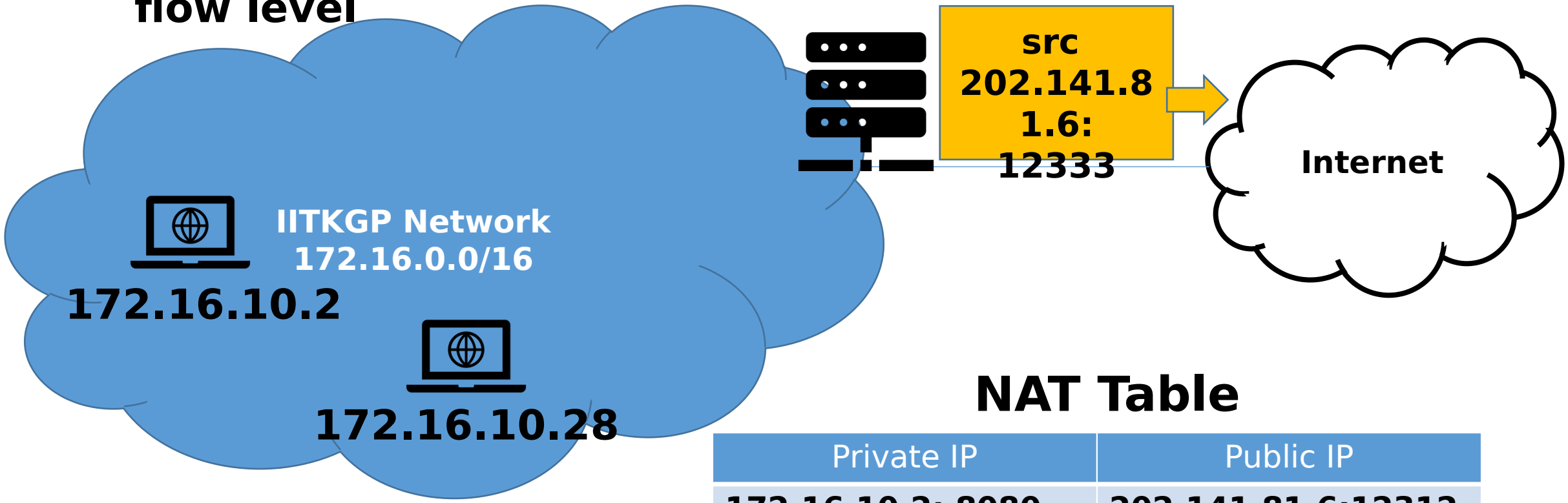


NAT Table

Private IP	Public IP
172.16.10.2: 8080	202.141.81.6:12312
172.16.10.28:80	202.141.81.6:12333

Port Based NAT (P-NAT)

- **Use a combination of IP address and the port number for mapping**
  - **The <IP, Port> pair defines an end-to-end flow -- so NAT at flow level**



NAT Table

Private IP	Public IP
172.16.10.2: 8080	202.141.81.6:12312
172.16.10.28:80	202.141.81.6:12333

Overheads/Problems associated with NAT

- Many of the Internet protocols and services (H.323, Rshell, IRC, PPTP, ICMP, IPSec, etc.) do not work over such Internet middleboxes -- particularly when the outside machine needs to initiate a connection
  - Sometime, NAT is combined with DNS to provide this support, DNS provides the IP address of the NAT server which helps in address translation

- The processing of the NAT table needs to be very fast -- many a times, the NAT works as the capacity bottleneck for a network

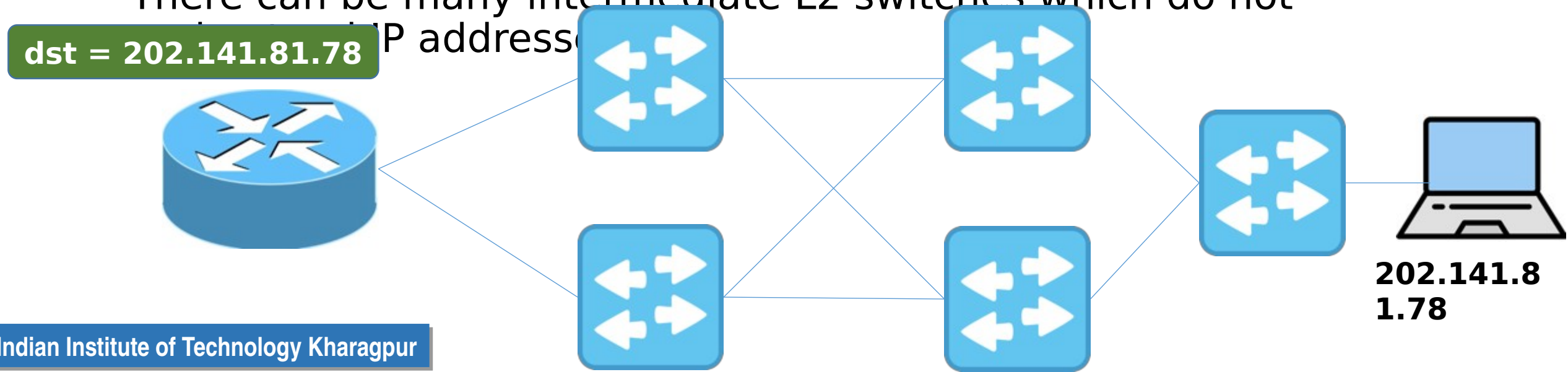
IP Masquerading

- Is a process where
  - one computer acts as an IP gateway for a network;
  - all other computers forward IP packets to that gateway,
  - the gateway replaces the source IP address with another IP address (its own IP, sometimes) and forwards the packet to the outside network
  - Mostly used to hide the IP addresses of the internal computers of a network
- NAT works like an IP masquerader
- Check `ipmasq` Linux tool (https://man.cx/ipmasq(8) to set IP masquerading over Linux

Address Resolution Protocol (ARP)

- Routing is done based on the IP addresses by making a lookup on the routing table.

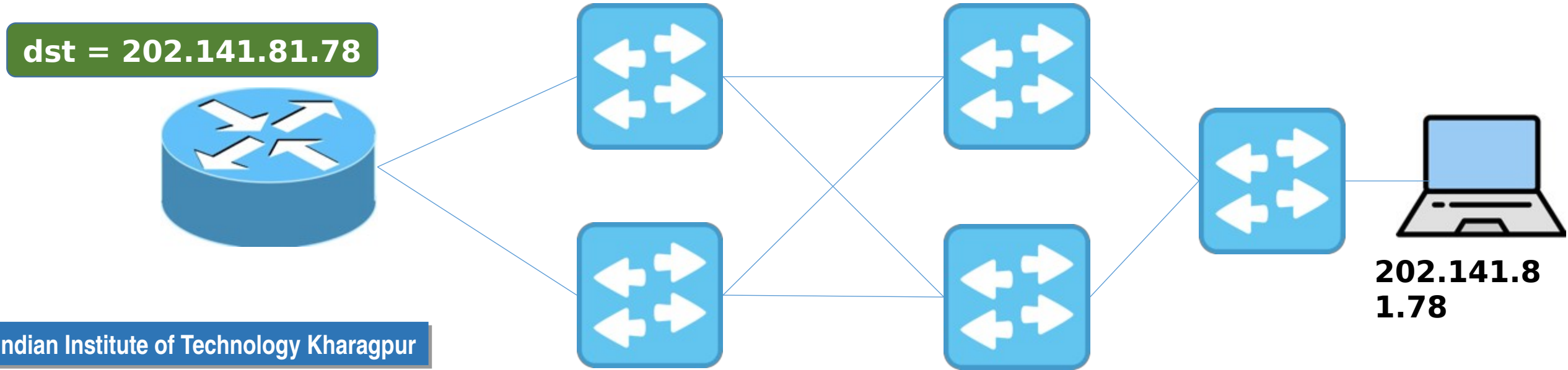
- Say, an IP packet has reached to the gateway router of a subnet, now the packet needs to be delivered to the final destination machine
  - There can be many intermediate L2 switches which do not



Address Resolution Protocol (ARP)

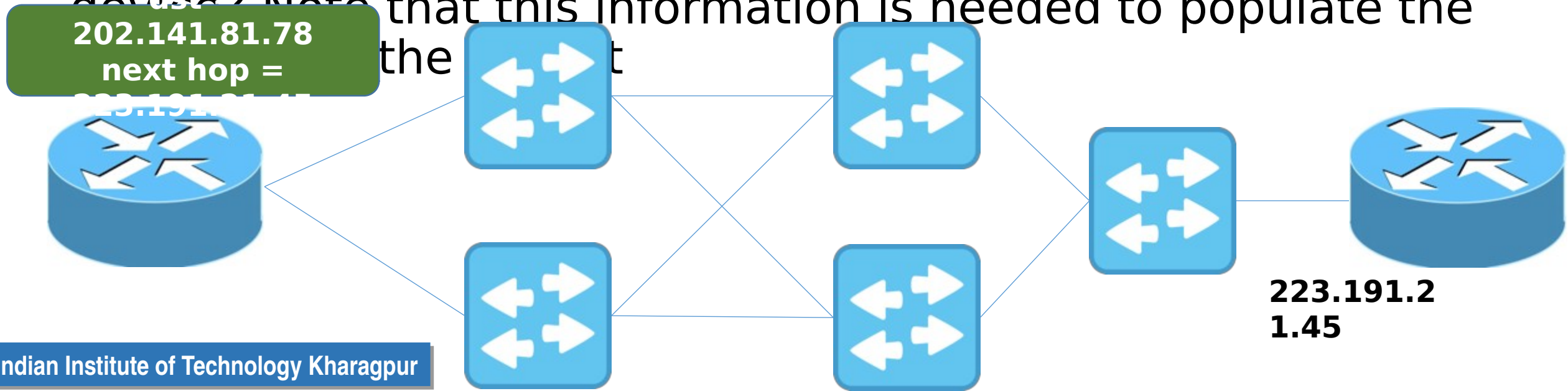
- The last hop packet delivery through the L2 switches is done based on the MAC addresses

- However, the routers do not know what is the MAC address of the final destination -- the IP header of the packet does not contain that



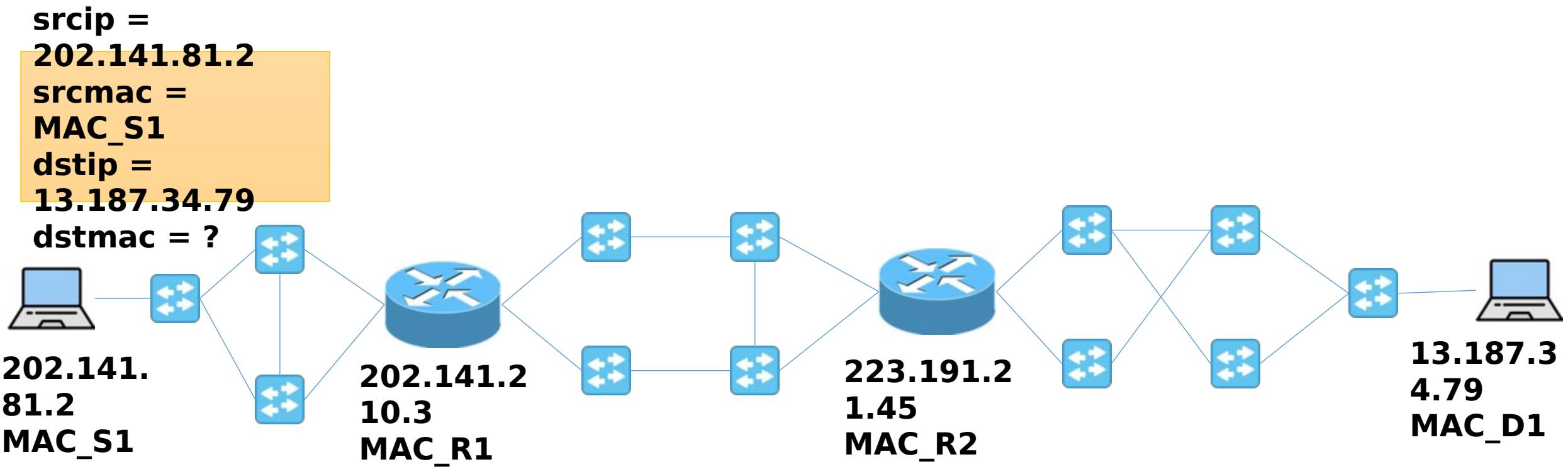
Address Resolution Protocol (ARP)

- The above example is true for every L3 hop in the Internet
- The routing table gives the IP of the next hop L3 device and the interface through which that L3 devices can be reached, however, it does not give the MAC address of the next hop device
- How will the router know the MAC address of the next hop device? Note that this information is needed to populate the [illegible] the [illegible]t



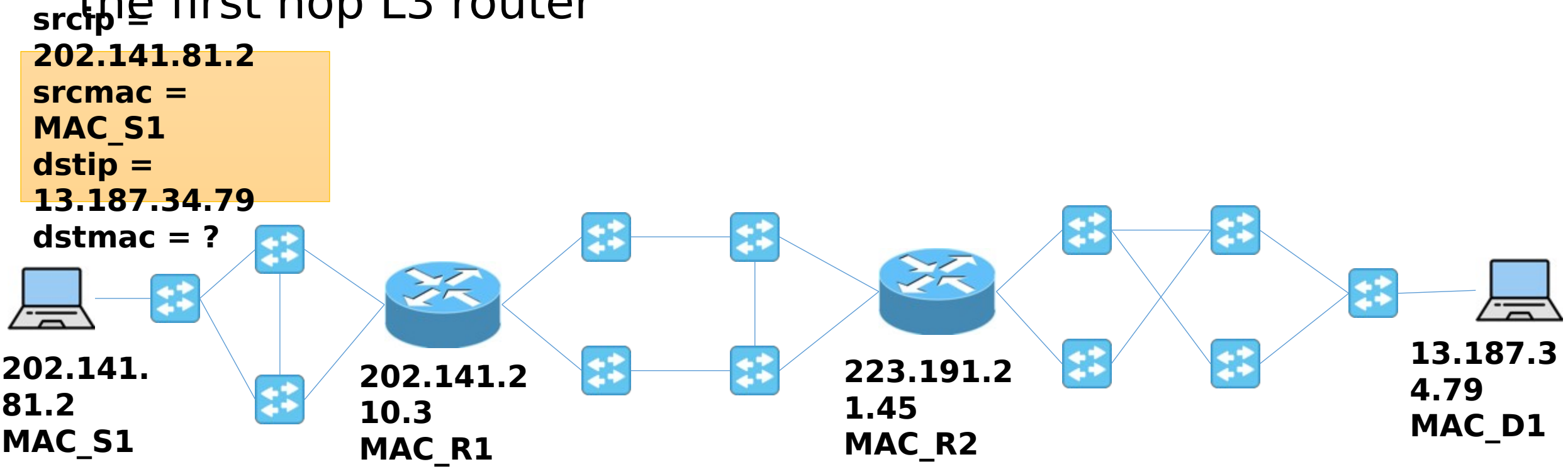
Construction of L2 Header during Routing

- The destination MAC needs to be found out at the source

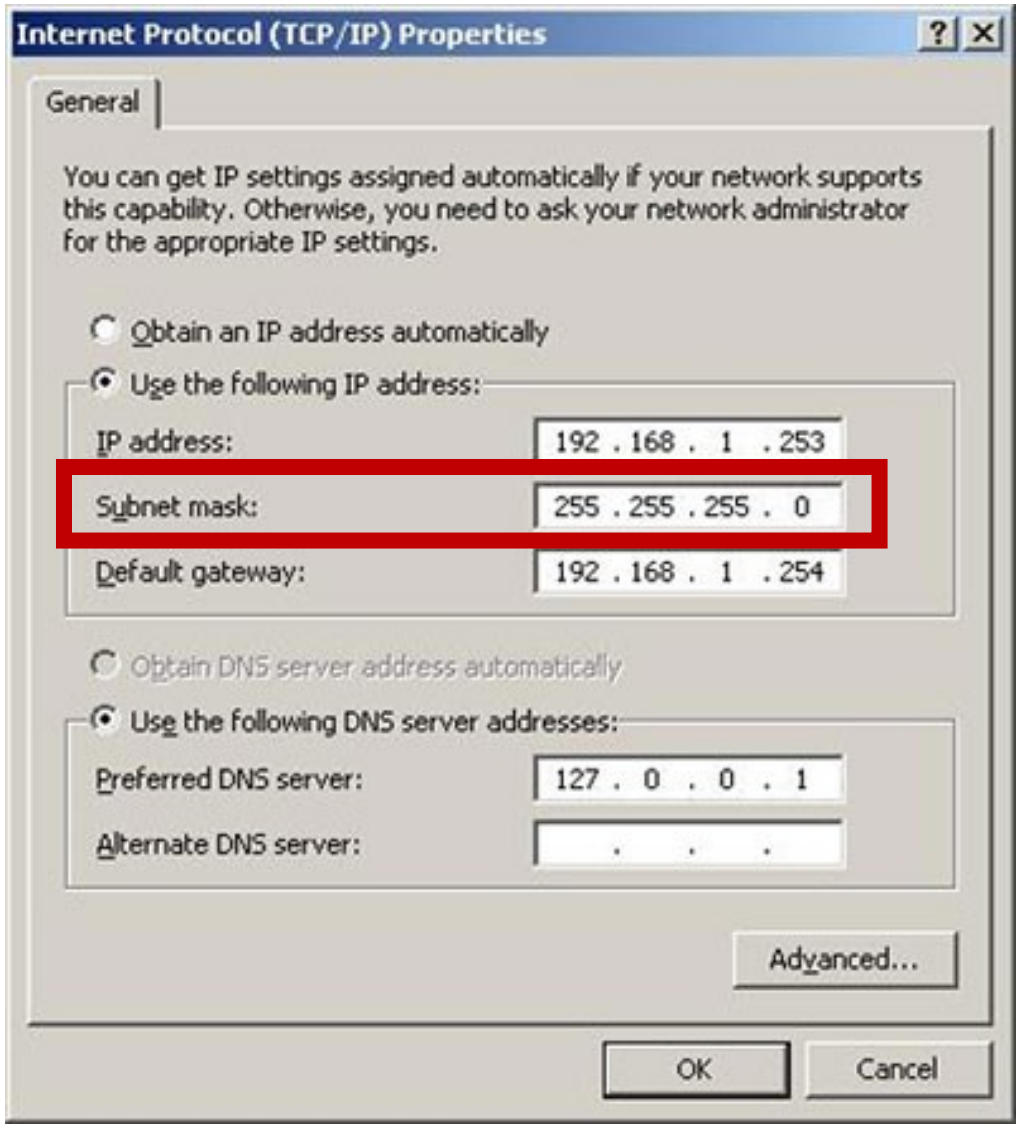


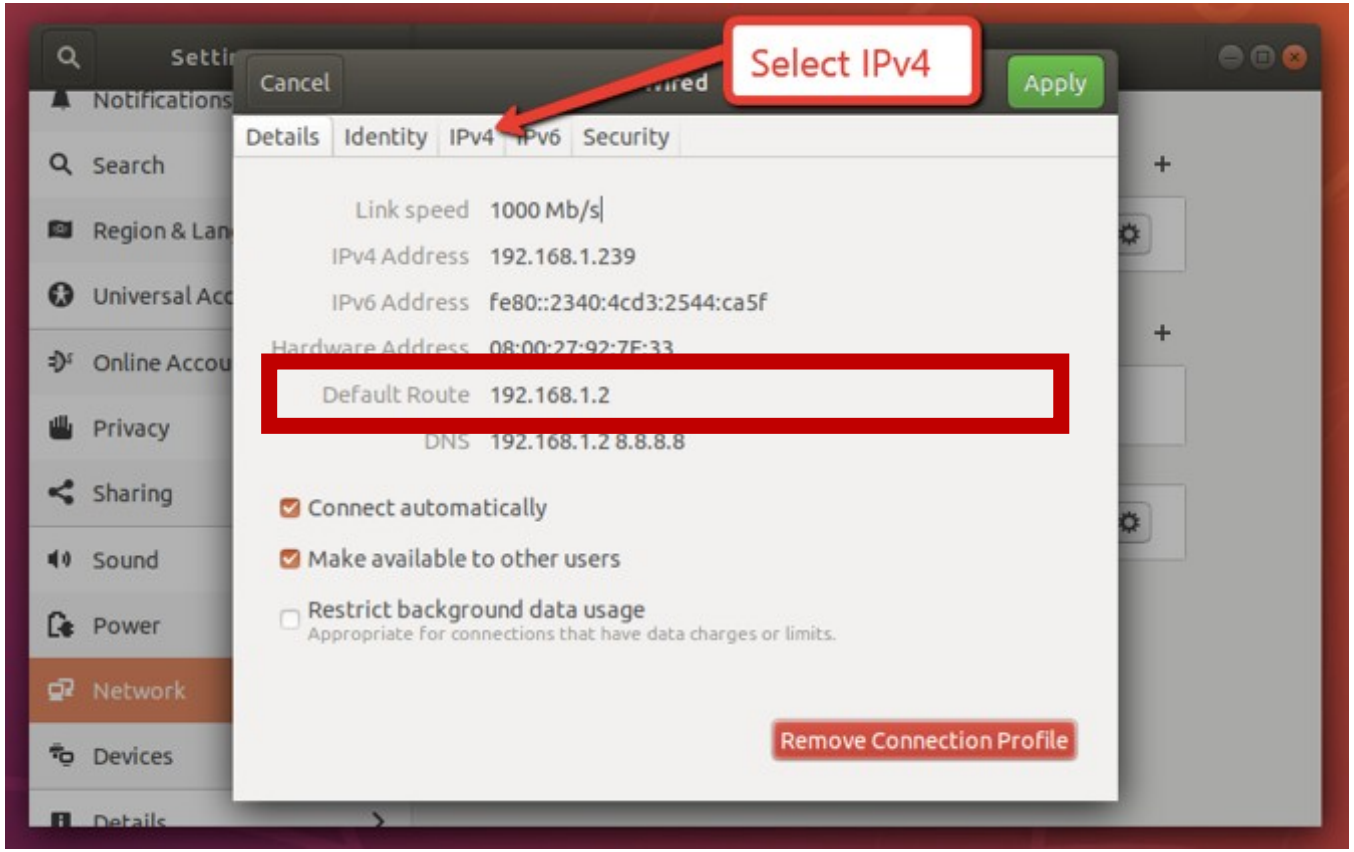
Construction of L2 Header during Routing

- The destination MAC needs to be found out at the source
- The source IP configuration contains a parameter called "default gateway" or "gateway", it contains the IP address of the first hop L3 router



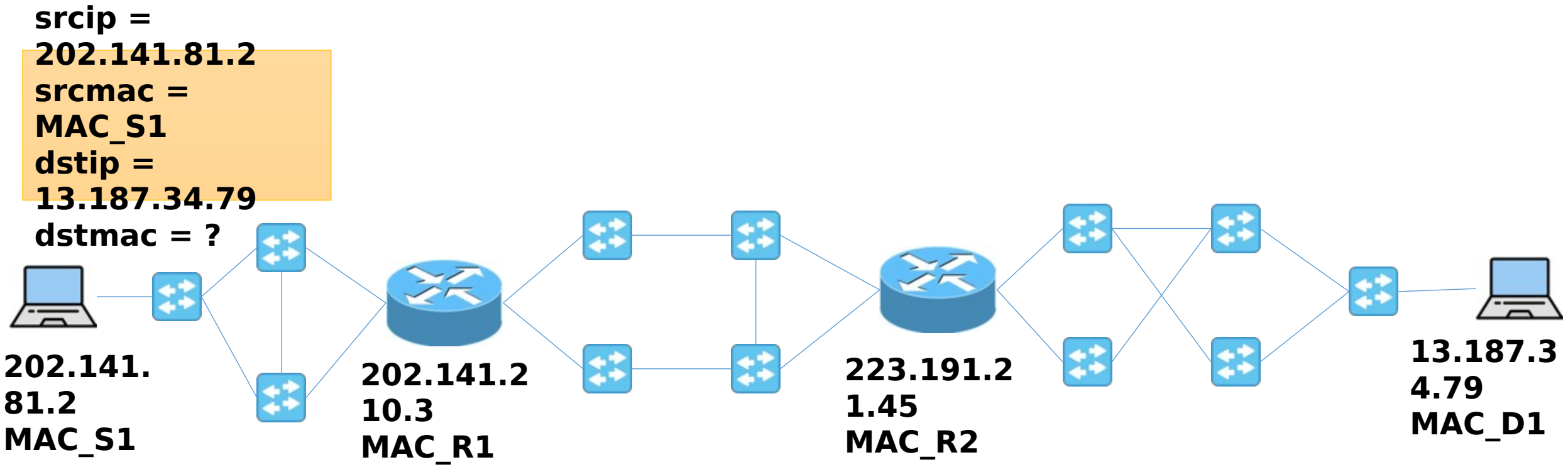
Construction of L2 Header during Routing





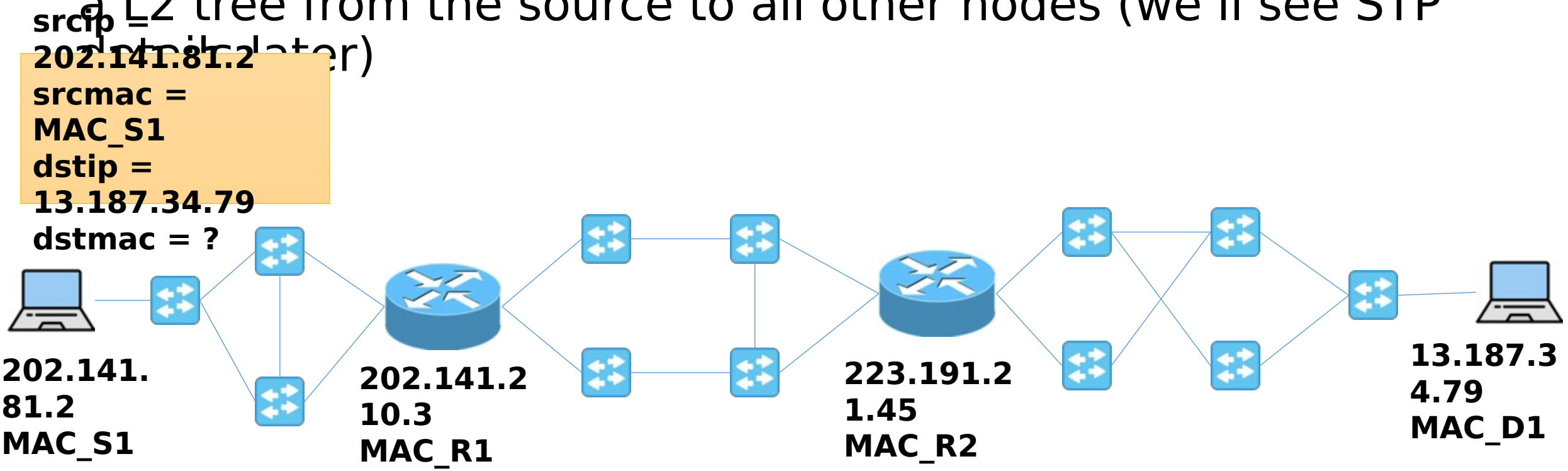
Construction of L2 Header during Routing

- So, here the IP address of the gateway is 202.141.210.3
- However, these two L3 devices are not directly connected. How do the network forward the packet to 202.141.210.3?



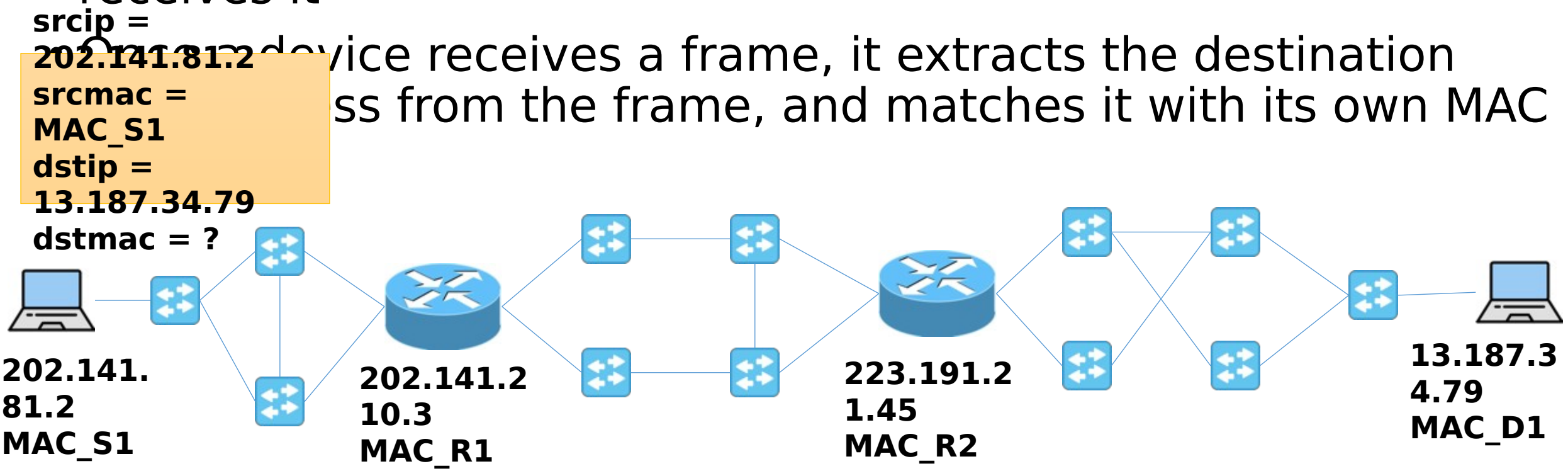
Construction of L2 Header during Routing

- The network need to do a L2 forwarding, forwarding based on the MAC addresses
- The protocol used: **Spanning Tree Protocol (STP)** - creates a L2 tree from the source to all other nodes (we'll see STP details later)



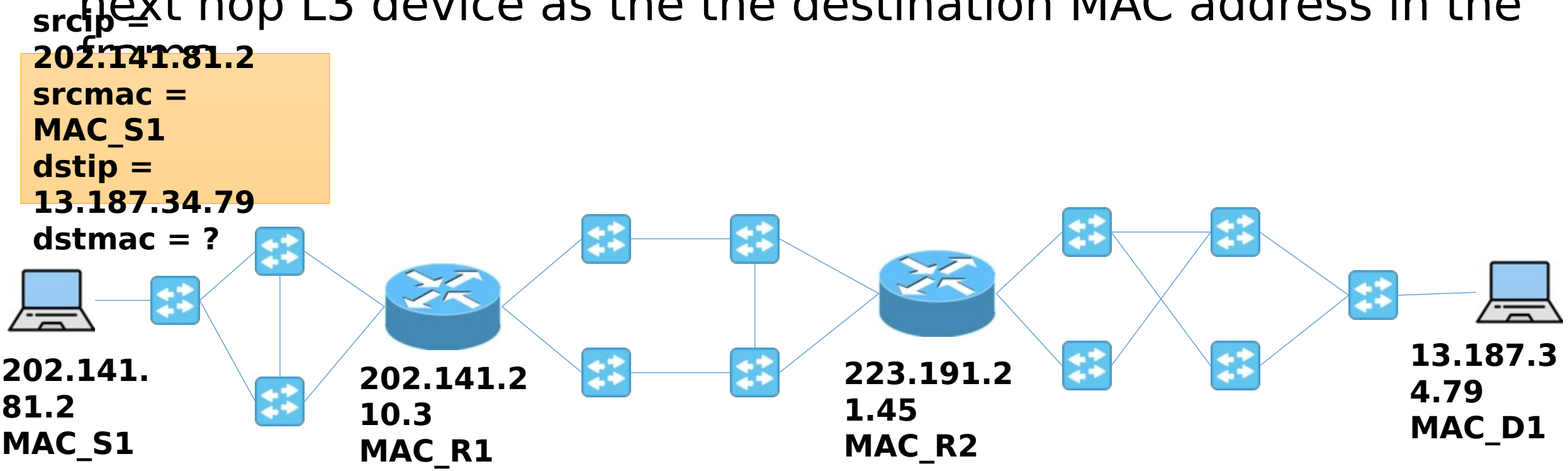
Construction of L2 Header during Routing

- The MAC frame is broadcast to all the L2 devices in the subnet
  using this spanning tree -- so, every device in the subnet receives it
- Each device receives a frame, it extracts the destination address from the frame, and matches it with its own MAC



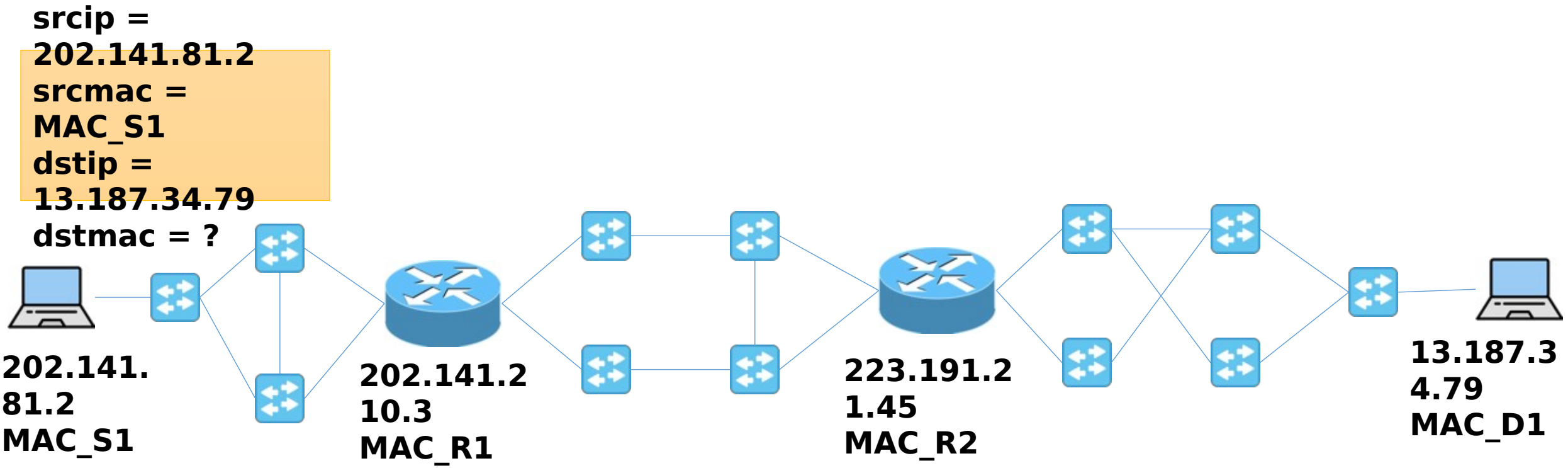
Construction of L2 Header during Routing

- If there is a match, it accepts the frame and forwards it to the upper layer, otherwise it drops the frame
- So, every L3 device needs to update the MAC address of the next hop L3 device as the the destination MAC address in the frame



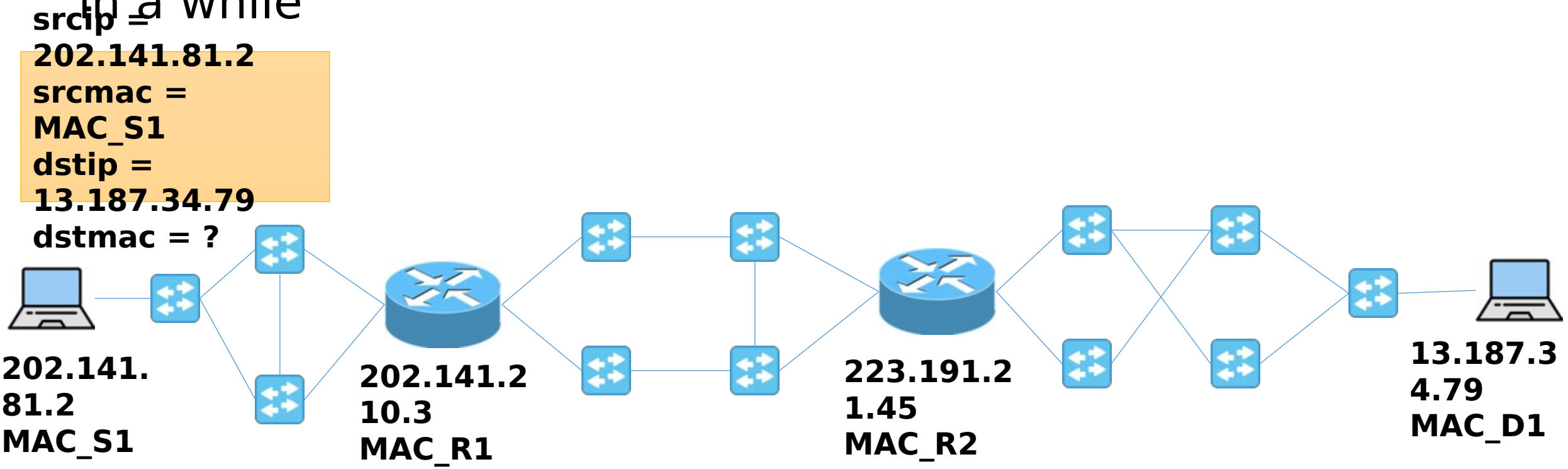
Construction of L2 Header during Routing

- But, how does a device find out the MAC address of the next hop L3 device?



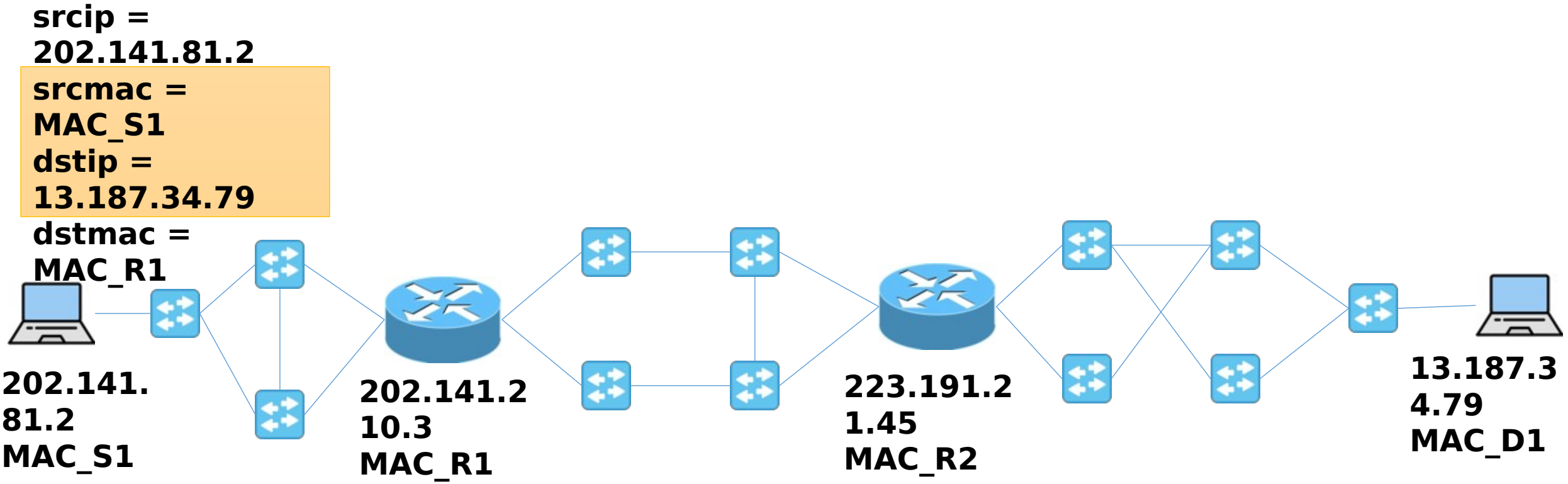
Construction of L2 Header during Routing

- But, how does a device find out the MAC address of the next hop L3 device?
- **Solution: Address Resolution Protocol (ARP)** - we'll see in a while



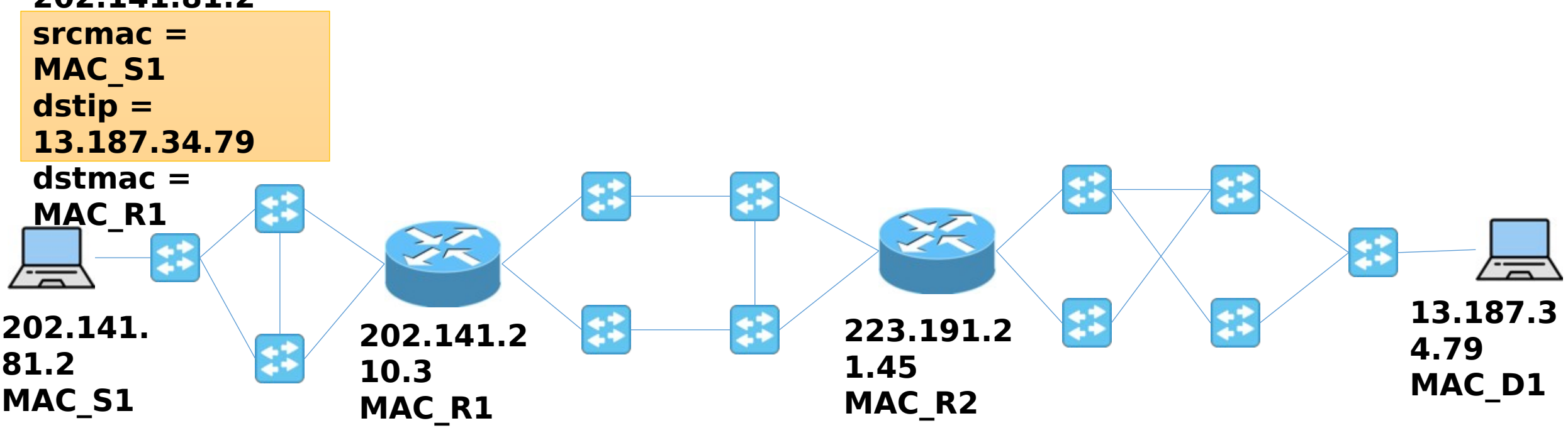
Construction of L2 Header during Routing

- So, the device uses ARP to find out the MAC address of R1 and
  populates it in the MAC frame.



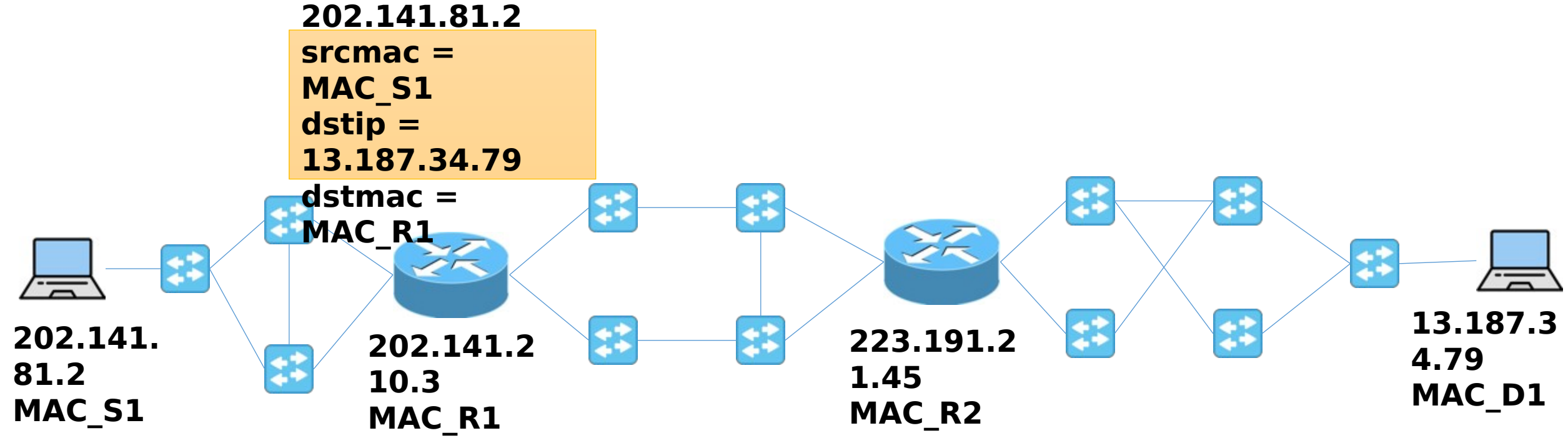
Construction of L2 Header during Routing

- So, the device uses ARP to find out the MAC address of R1 and
  populates it in the MAC frame.
- Now, STP is used to forward the frame to R1.



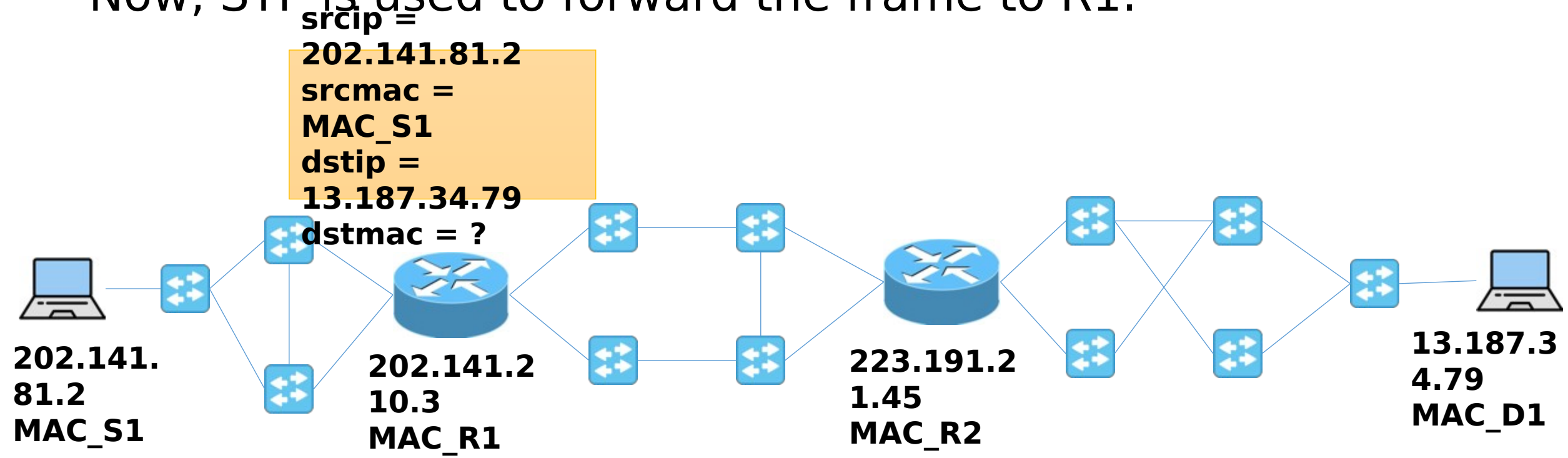
Construction of L2 Header during Routing

- So, the device uses ARP to find out the MAC address of R1 and
  populates it in the MAC frame.
- Now, STP is used to forward the frame to R1.



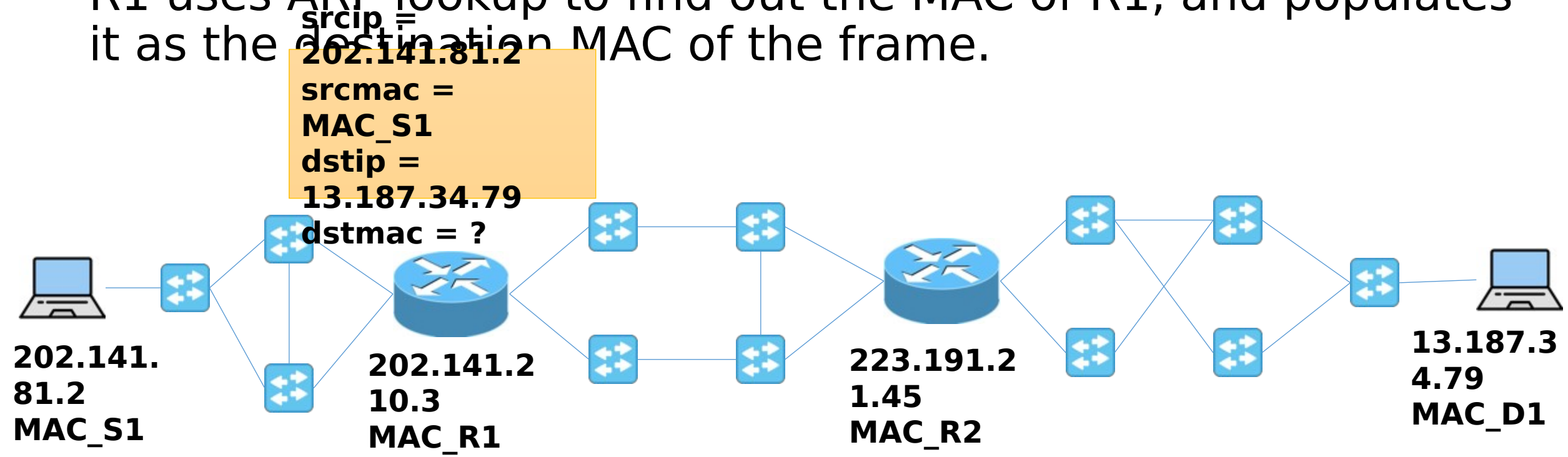
Construction of L2 Header during Routing

- So, the device uses ARP to find out the MAC address of R1 and
  populates it in the MAC frame.
- Now, STP is used to forward the frame to R1.



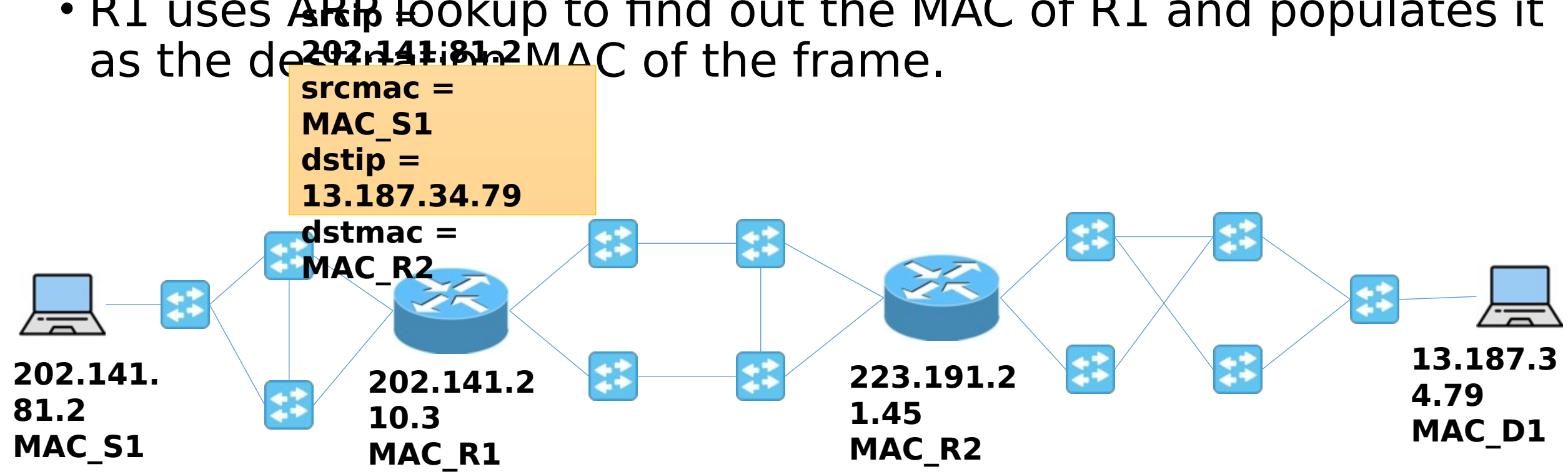
Construction of L2 Header during Routing

- Now, R1 uses the route lookup, and see that R2 is the next hop.
  So, next hop IP is 223.191.21.45
- R1 uses ARP lookup to find out the MAC of R1, and populates it as the destination MAC of the frame.



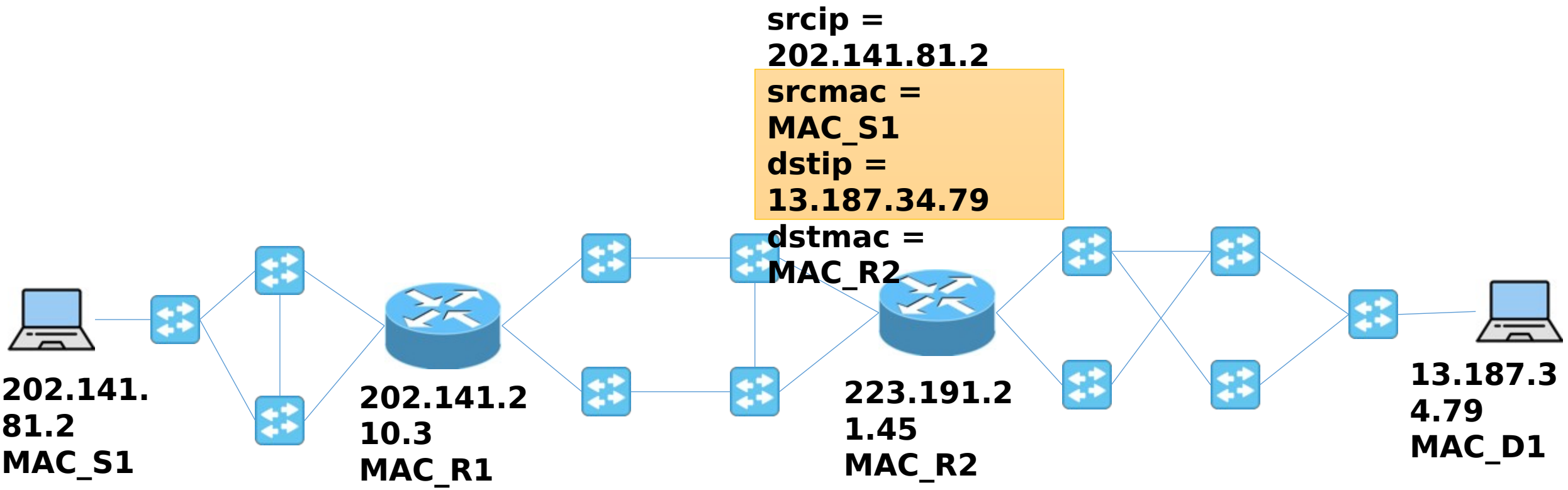
Construction of L2 Header during Routing

- Now, R1 uses the route lookup, and see that R2 is the next hop.
  So, next hop IP is 223.191.21.45
- R1 uses ARP lookup to find out the MAC of R1 and populates it as the destination MAC of the frame.



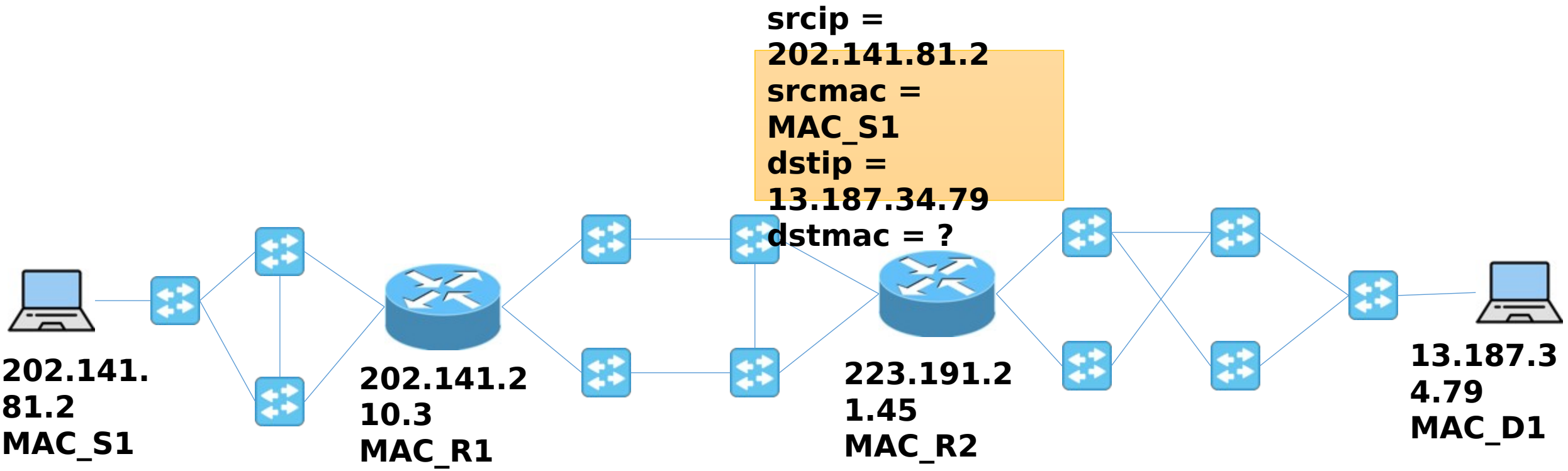
Construction of L2 Header during Routing

- Now, R2 uses the same method to do an ARP lookup, and uses
  the MAC address of D1 to forward it at the final destination.



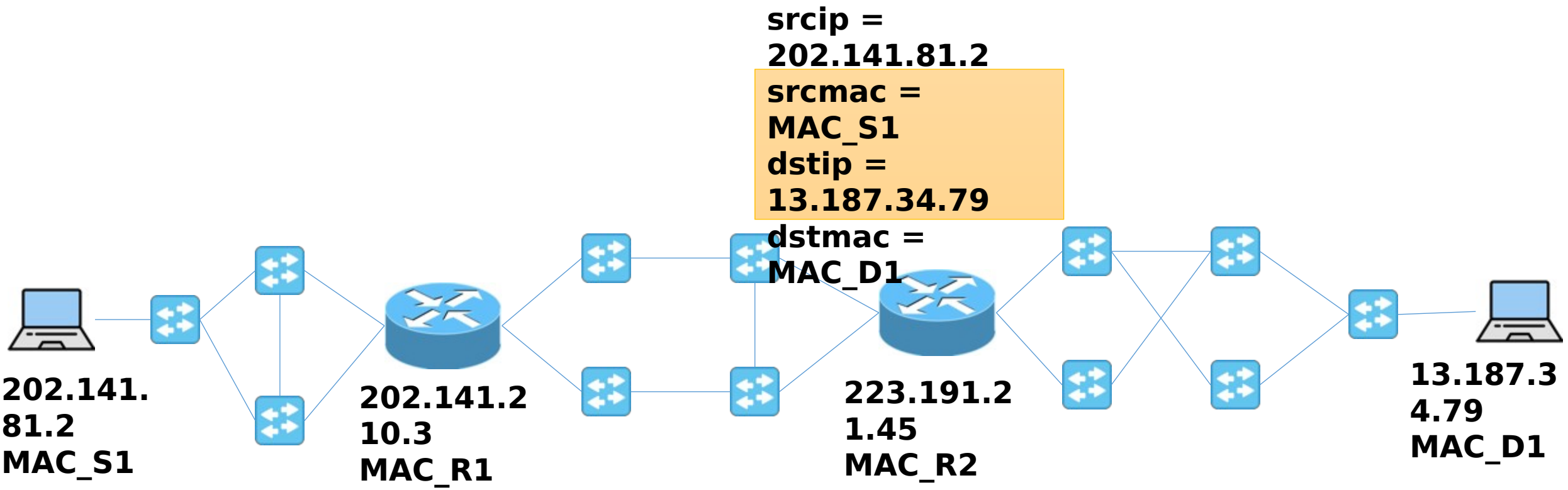
Construction of L2 Header during Routing

- Now, R2 uses the same method to do an ARP lookup, and uses
  the MAC address of D1 to forward it at the final destination.



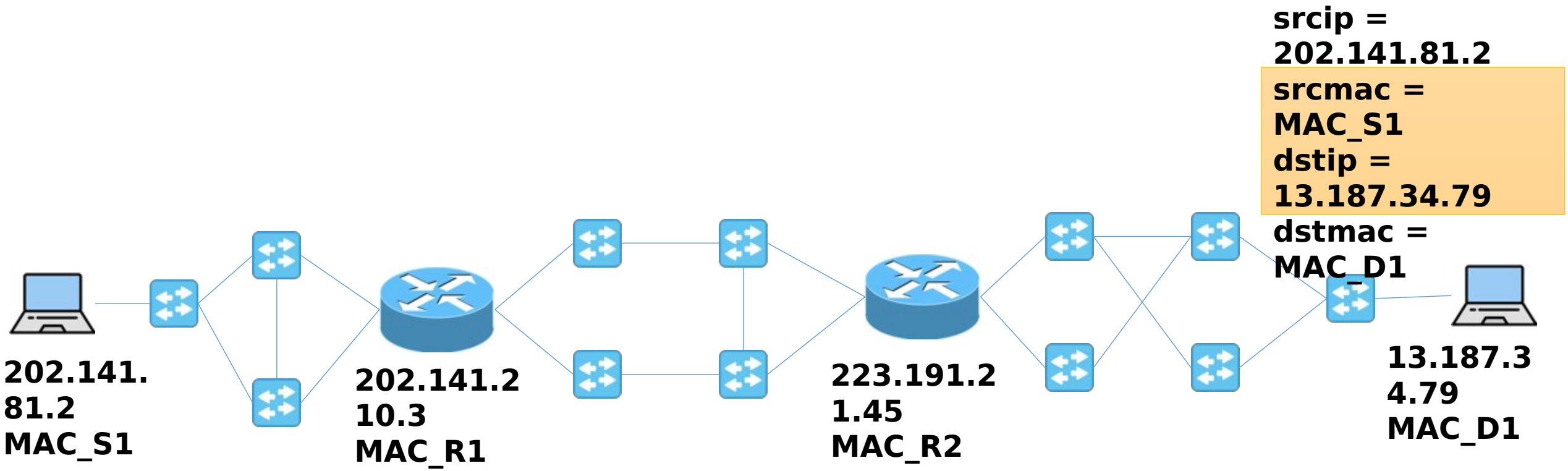
Construction of L2 Header during Routing

- Now, R2 uses the same method to do an ARP lookup, and uses
  the MAC address of D1 to forward it at the final destination.



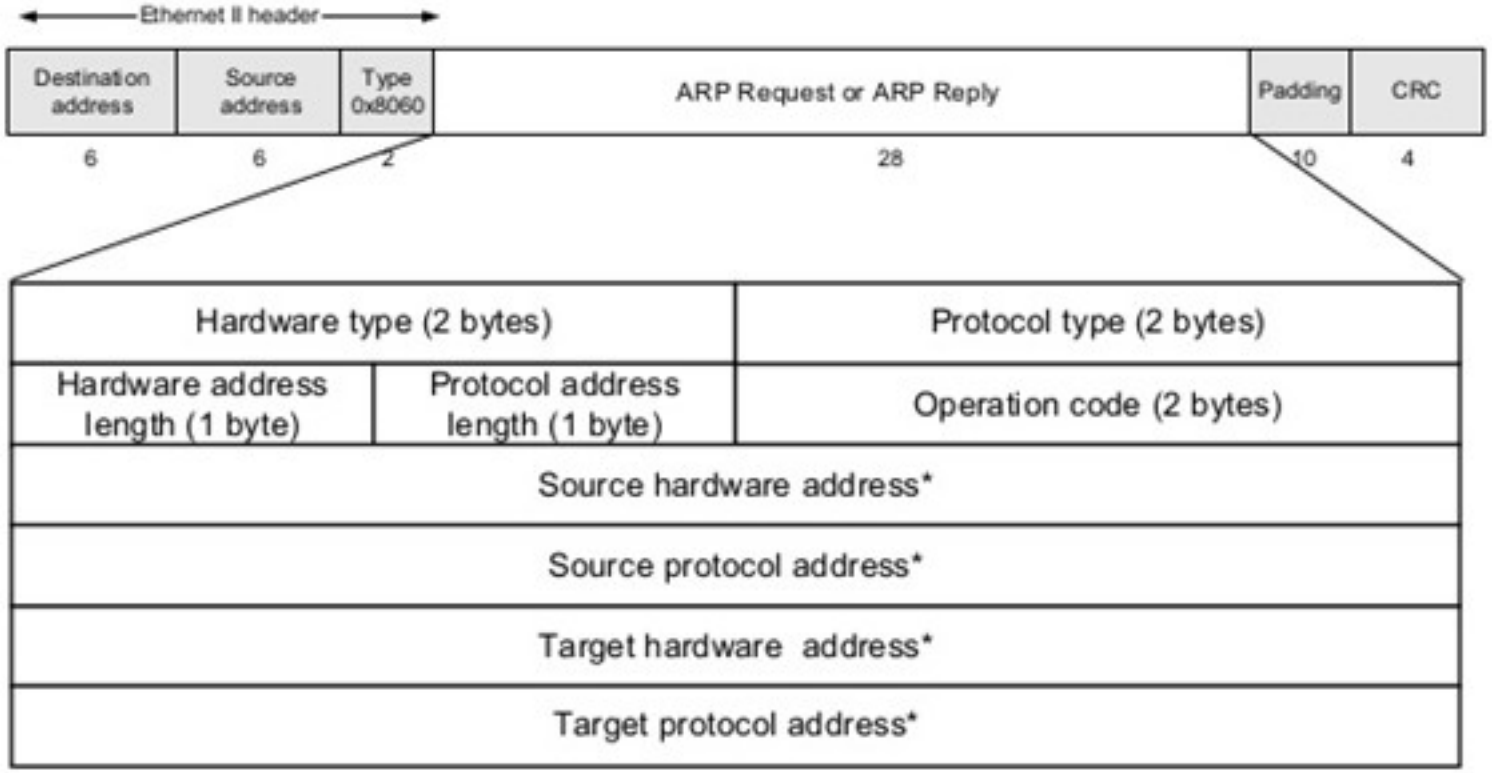
Construction of L2 Header during Routing

- Now, R2 uses the same method to do an ARP lookup, and uses
  the MAC address of D1 to forward it at the final destination.



Address Resolution Protocol (ARP)

- Two types of messages -- ARP Request and ARP Reply



Address Resolution Protocol (ARP)

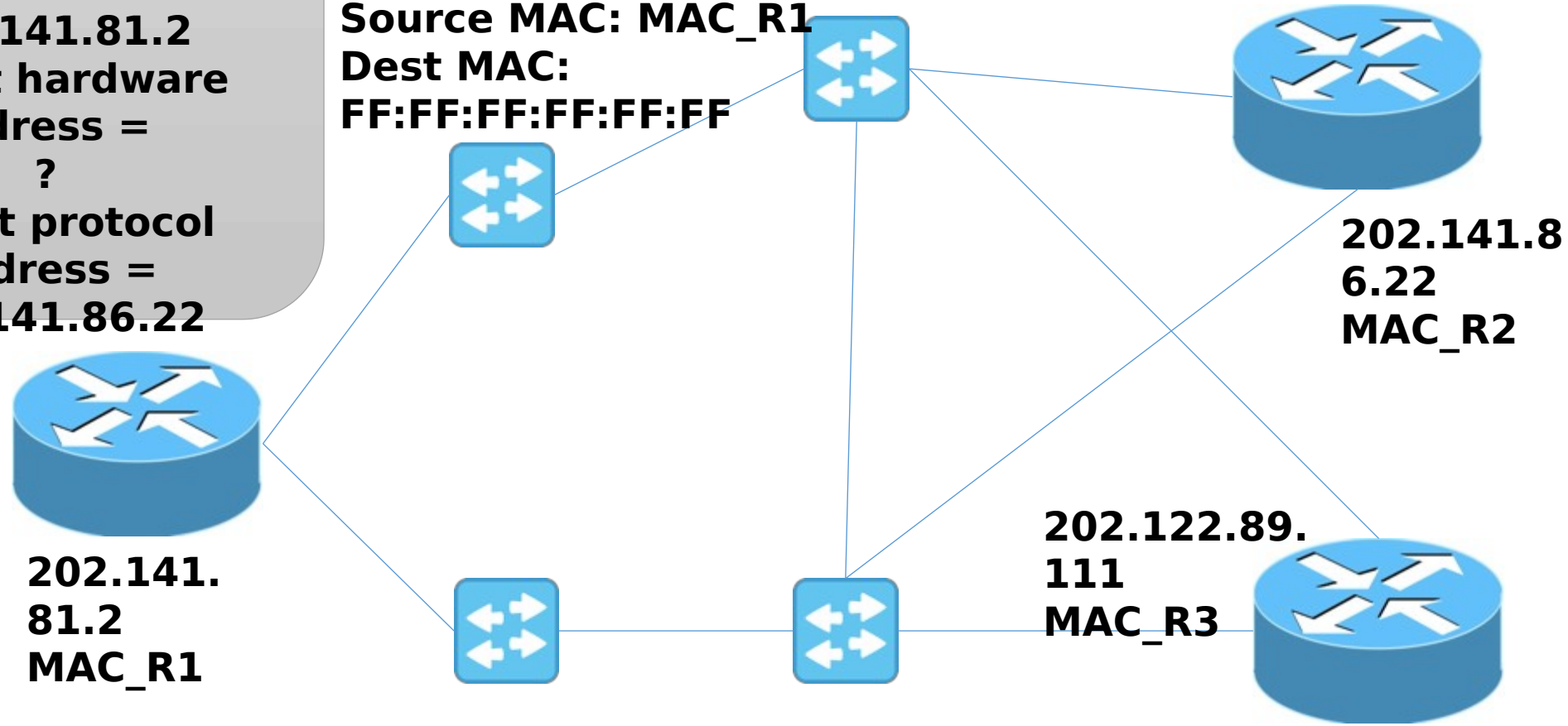
- ARP Request is a broadcast in the network with
  - broadcast IP address 255.255.255.255 and
  - broadcast MAC address FF:FF:FF:FF:FF:FF

- Once a device receives an ARP Request message where the *target protocol address* matches with its own IP address, it sends back an ARP reply with its MAC address at the *target hardware address*.

Address Resolution Protocol (ARP)

source hardware address = MAC_R1
source protocol address = 202.141.81.2
target hardware address = ?
target protocol address = 202.141.86.22

Source IP: 202.141.81.2
Dest IP: 255.255.255.255
Source MAC: MAC_R1
Dest MAC: FF:FF:FF:FF:FF:FF

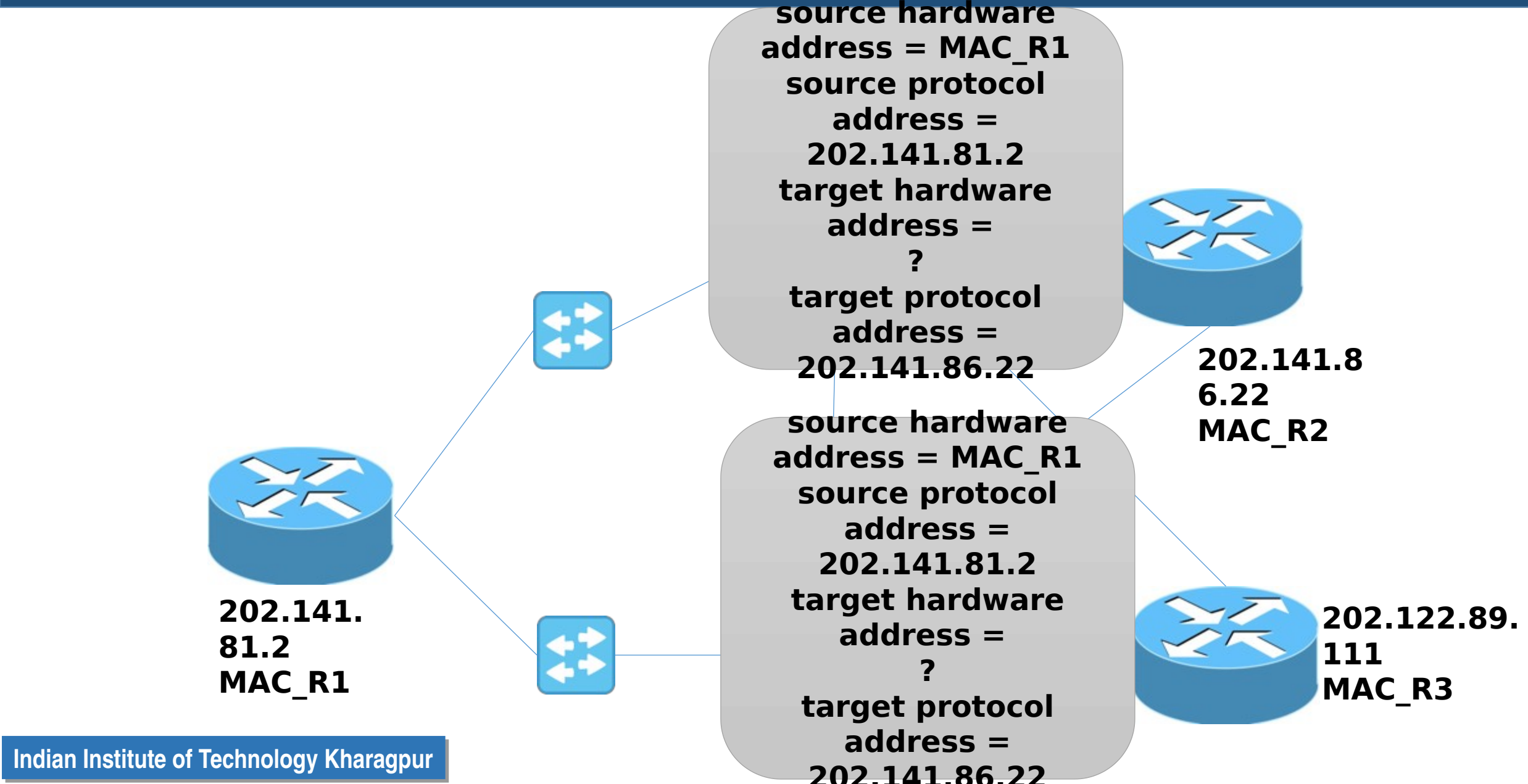


202.141.81.2 MAC_R1

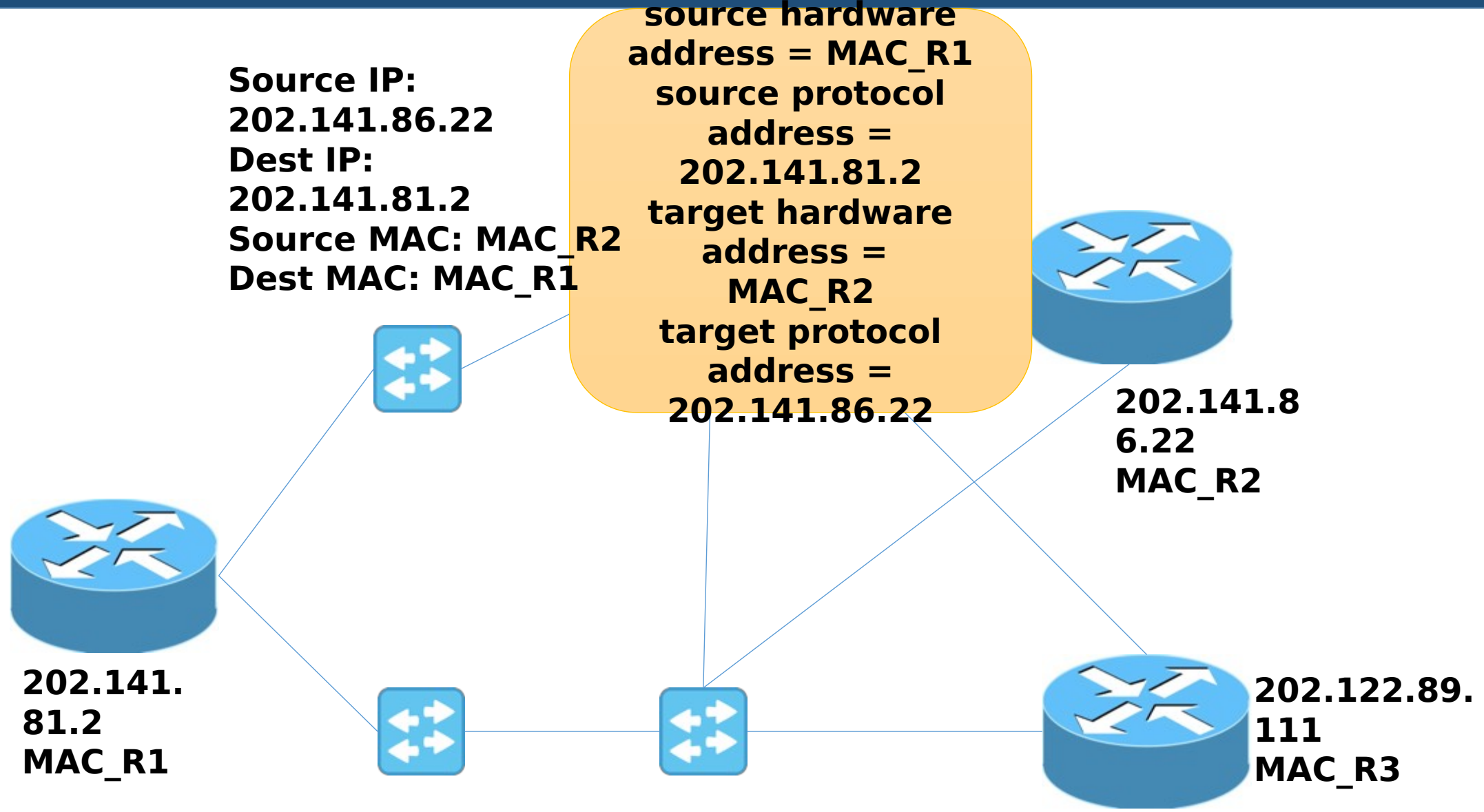
202.141.86.22 MAC_R2

202.122.89.111 MAC_R3

Address Resolution Protocol (ARP)

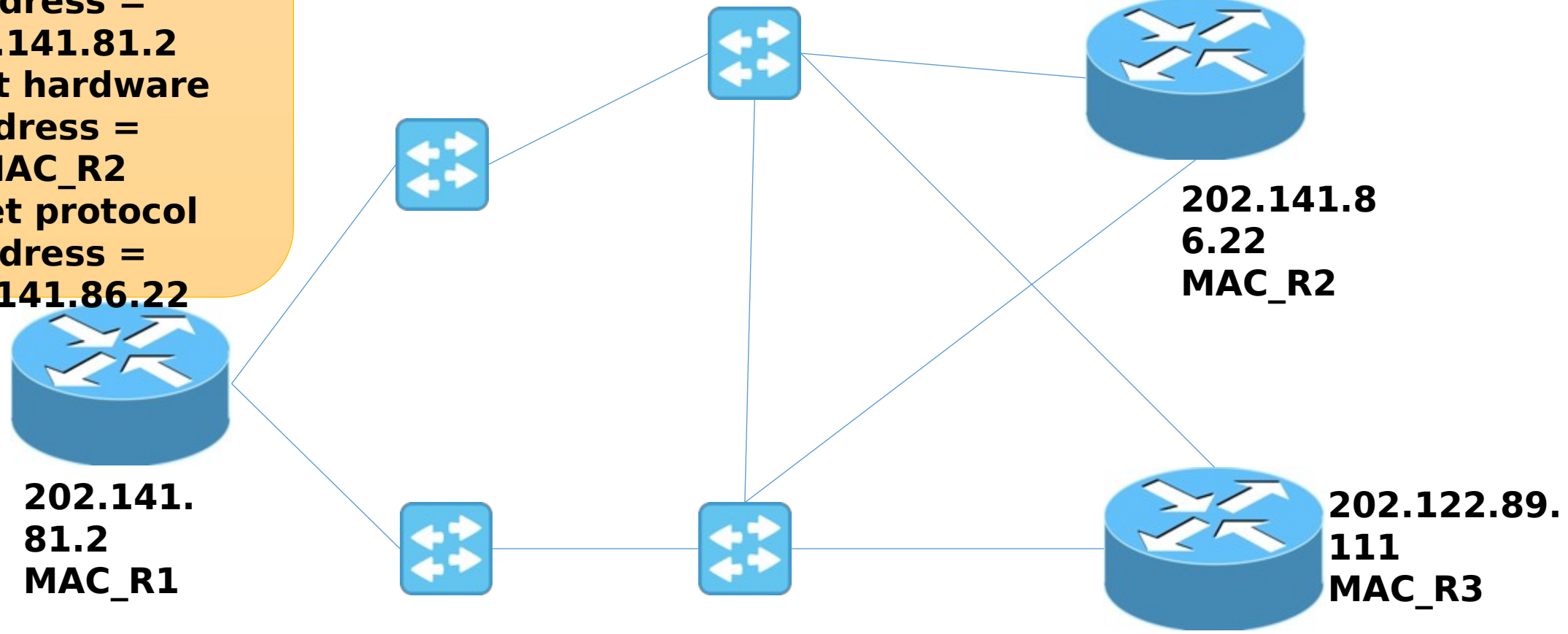


Address Resolution Protocol (ARP)



Address Resolution Protocol (ARP)

source hardware address = MAC_R1
source protocol address = 202.141.81.2
target hardware address = MAC_R2
target protocol address = 202.141.86.22



Internet Control Message Protocol (ICMP)

- A set of messaging protocol for Internet management, error messaging and network configuration verification.

- Typically, provides a feedback when an IP message is sent --
  - Inform the source when an IP packet is dropped from an intermediate L3 device
  - Check whether a L3 host is reachable
  - Redirect messages
  - Router advertisement

- Works on top of IP, ICMPv4 for IPv4 and ICMPv6 for IPv6

ICMP Message Types

- ICMP messages are divided into two classes -- **Error messages** and **Information (or query) messages**

- **Error messages:** Used to provide feedback to a source L3 device about an error that has occurred
  - **Example: Destination unreachable** -- the router cannot find out a path towards the destination

- **Information (or query) messages:** Used by L3 devices to exchange information, implement certain IP related features and perform testing
  - **Example:** Echo request and Echo reply

Echo Request and Echo Reply

- Used in ping application to check whether a L3 device is alive and to calculate the round-trip time to that device.

```
Sandip@Sandips-MacBook-Air ~ % ping 172.217.163.164 -c 10
PING 172.217.163.164 (172.217.163.164): 56 data bytes
64 bytes from 172.217.163.164: icmp_seq=0 ttl=53 time=104.321 ms
64 bytes from 172.217.163.164: icmp_seq=1 ttl=53 time=69.001 ms
64 bytes from 172.217.163.164: icmp_seq=2 ttl=53 time=58.311 ms
64 bytes from 172.217.163.164: icmp_seq=3 ttl=53 time=66.821 ms
64 bytes from 172.217.163.164: icmp_seq=4 ttl=53 time=62.243 ms
64 bytes from 172.217.163.164: icmp_seq=5 ttl=53 time=61.716 ms
64 bytes from 172.217.163.164: icmp_seq=6 ttl=53 time=65.805 ms
64 bytes from 172.217.163.164: icmp_seq=7 ttl=53 time=62.454 ms
64 bytes from 172.217.163.164: icmp_seq=8 ttl=53 time=68.606 ms
64 bytes from 172.217.163.164: icmp_seq=9 ttl=53 time=56.426 ms

--- 172.217.163.164 ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 56.426/67.570/104.321/12.869 ms
```

Echo Request and Echo Reply

- The ping application generates an ICMP Echo Request towards the targeted L3 device.
- Once the targeted L3 device receives the Echo Request, it replies back with an Echo Reply
- The time is calculated based on the time difference between the Echo Request and the receipt of Echo Reply

RTT = T2 - T1

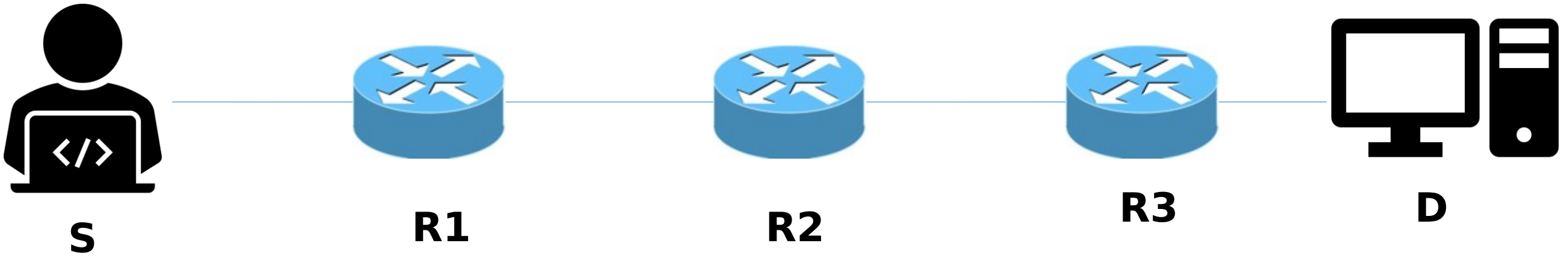
Traceroute

- Find out the intermediate L3 devices at every hop between a source and a destination

```
Sandip@Sandips-MacBook-Air ~ % traceroute 172.217.163.164
traceroute to 172.217.163.164 (172.217.163.164), 64 hops max, 52 byte packets
 1  www.huaweimobilewifi.com (192.168.1.1)  3.581 ms  3.082 ms  3.401 ms
 2  10.50.102.4 (10.50.102.4)  66.293 ms  78.664 ms  79.660 ms
 3  10.50.102.103 (10.50.102.103)  71.001 ms  88.091 ms  79.898 ms
 4  * * *
 5  aes-static-013.78.22.125.airtel.in (125.22.78.13)  34.004 ms  71.720 ms
    aes-static-054.78.22.125.airtel.in (125.22.78.54)  90.092 ms
 6  182.79.198.24 (182.79.198.24)  103.061 ms
    182.79.142.218 (182.79.142.218)  117.620 ms  98.619 ms
 7  72.14.211.198 (72.14.211.198)  80.188 ms  84.665 ms  87.808 ms
 8  * * *
 9  * * *
10  maa05s05-in-f4.1e100.net (172.217.163.164)  130.027 ms  130.337 ms  80.199 ms
```

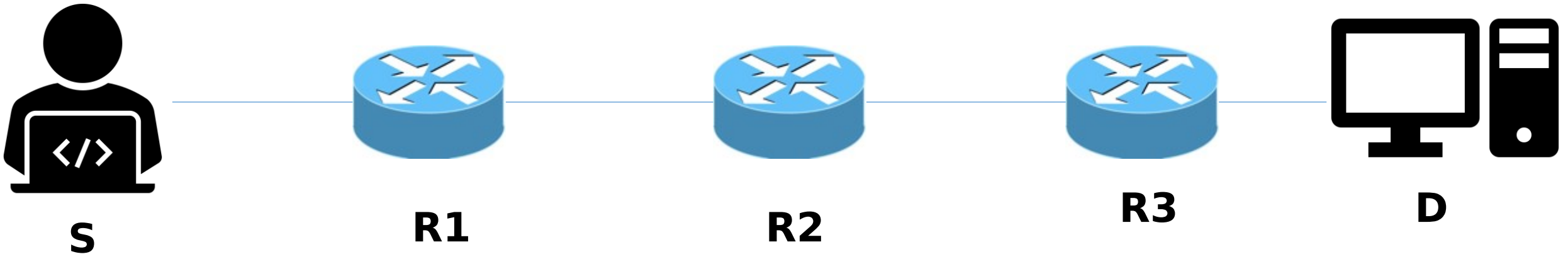
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Traceroute

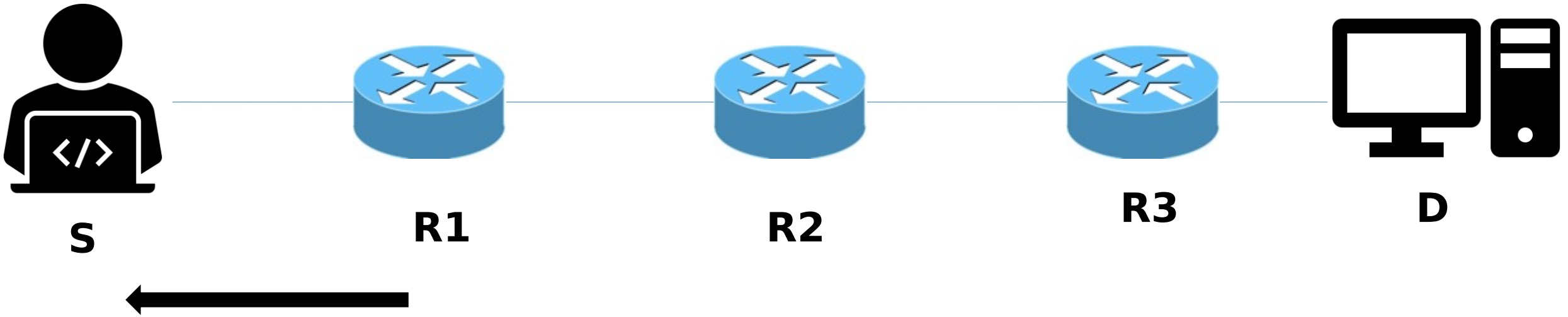
- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Protocol = UDP
Destination = D
Port = 33434
TTL = 1

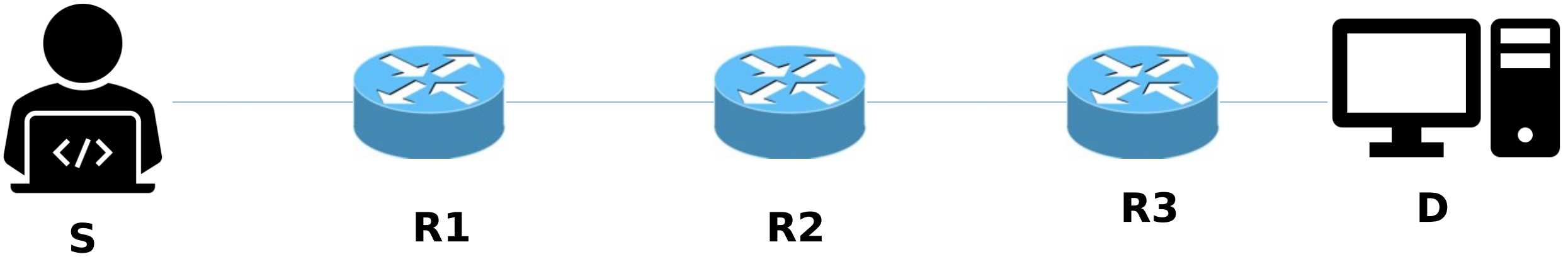
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Traceroute

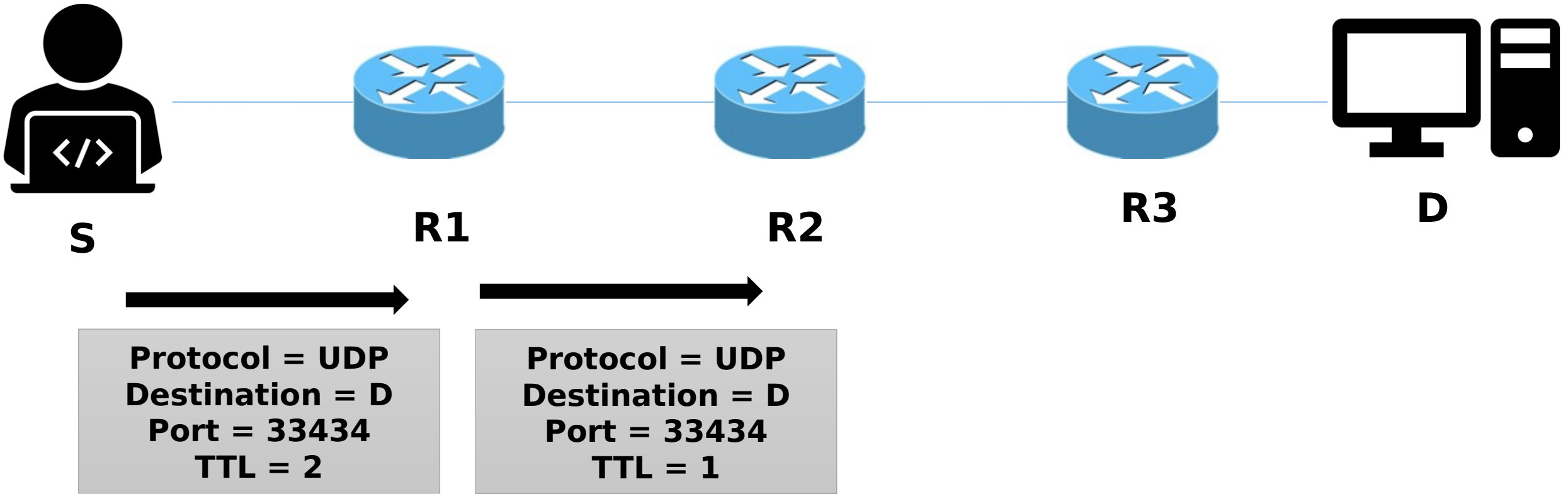
- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Protocol = UDP
Destination = D
Port = 33434
TTL = 2

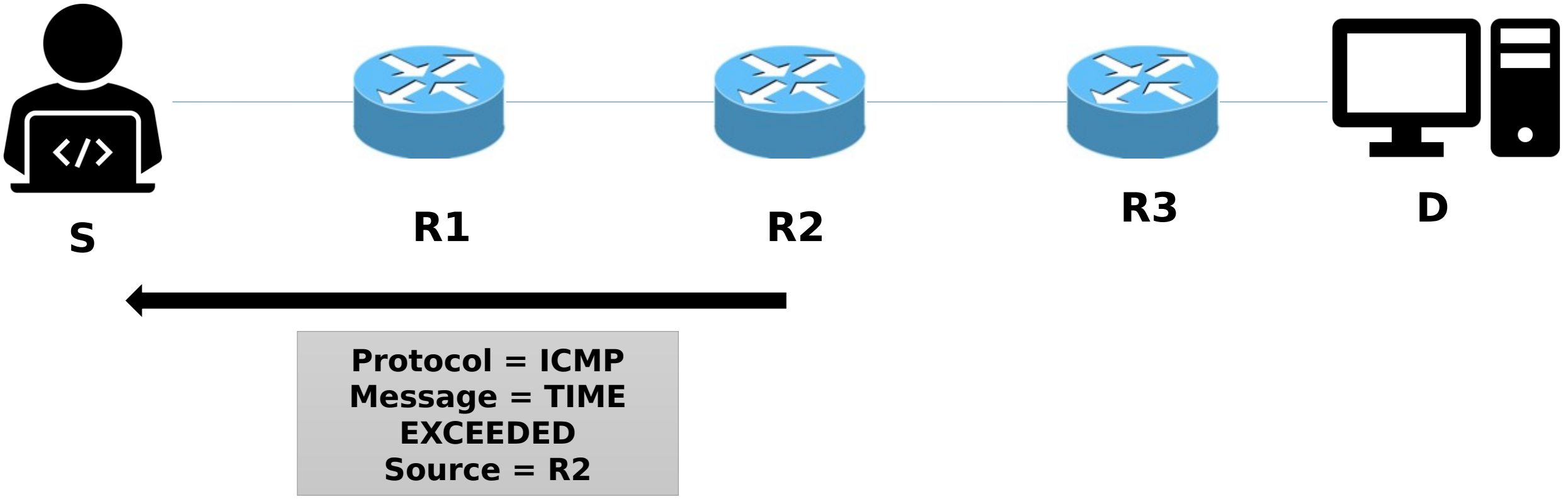
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



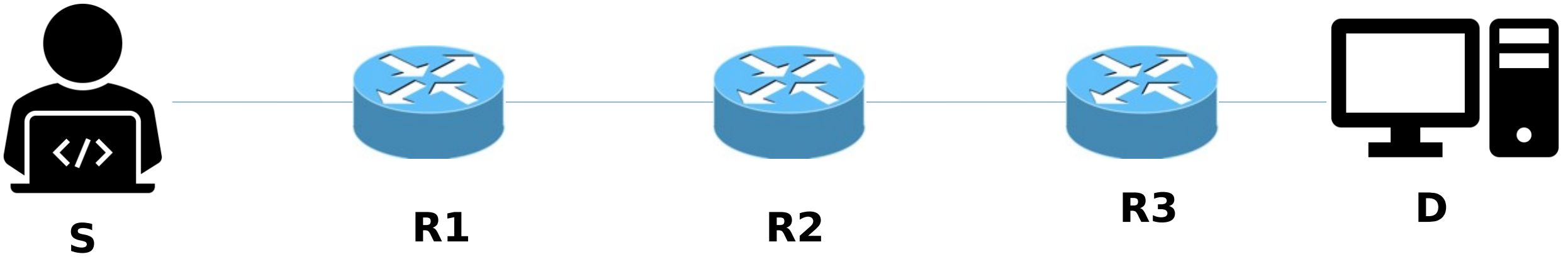
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Traceroute

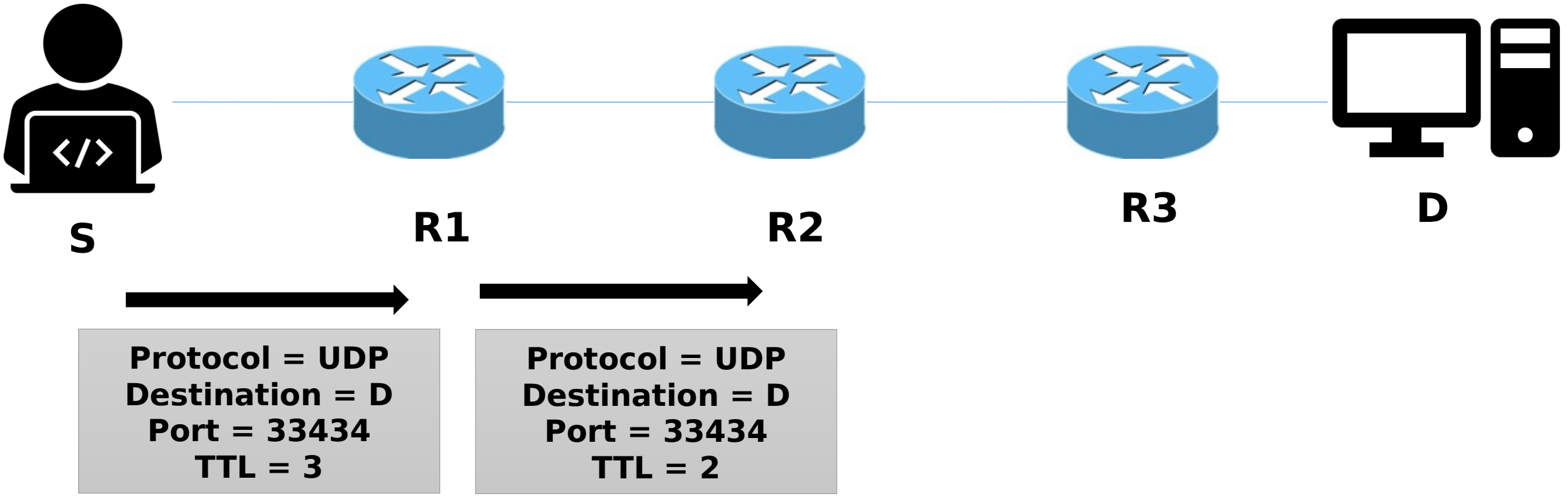
- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Protocol = UDP
Destination = D
Port = 33434
TTL = 3

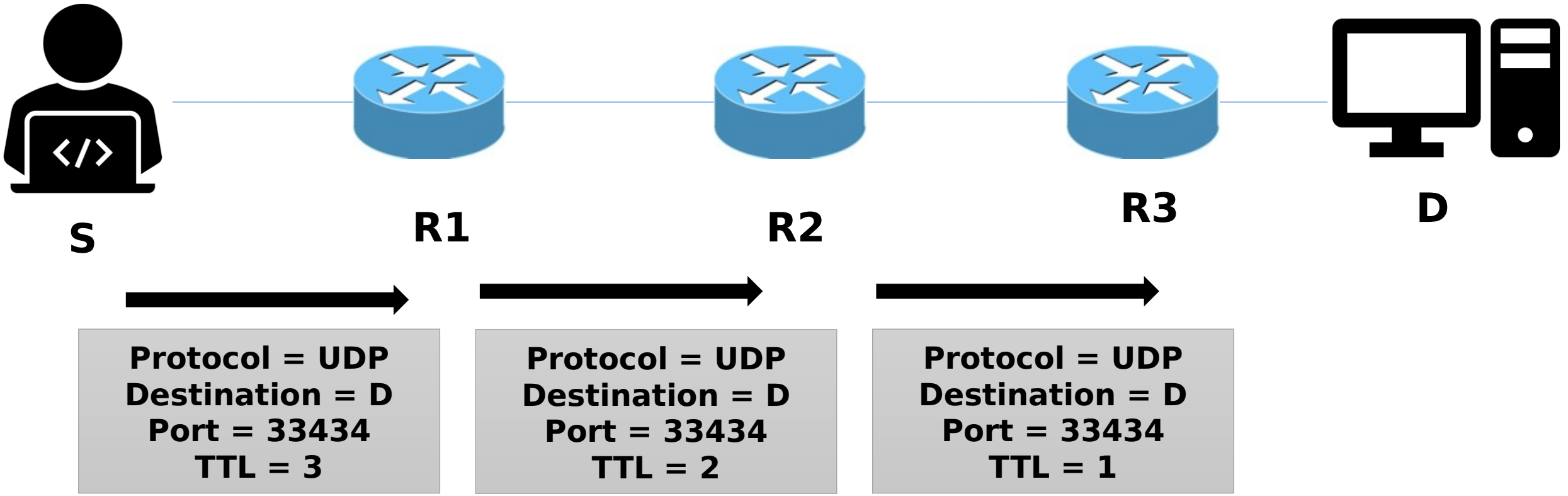
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



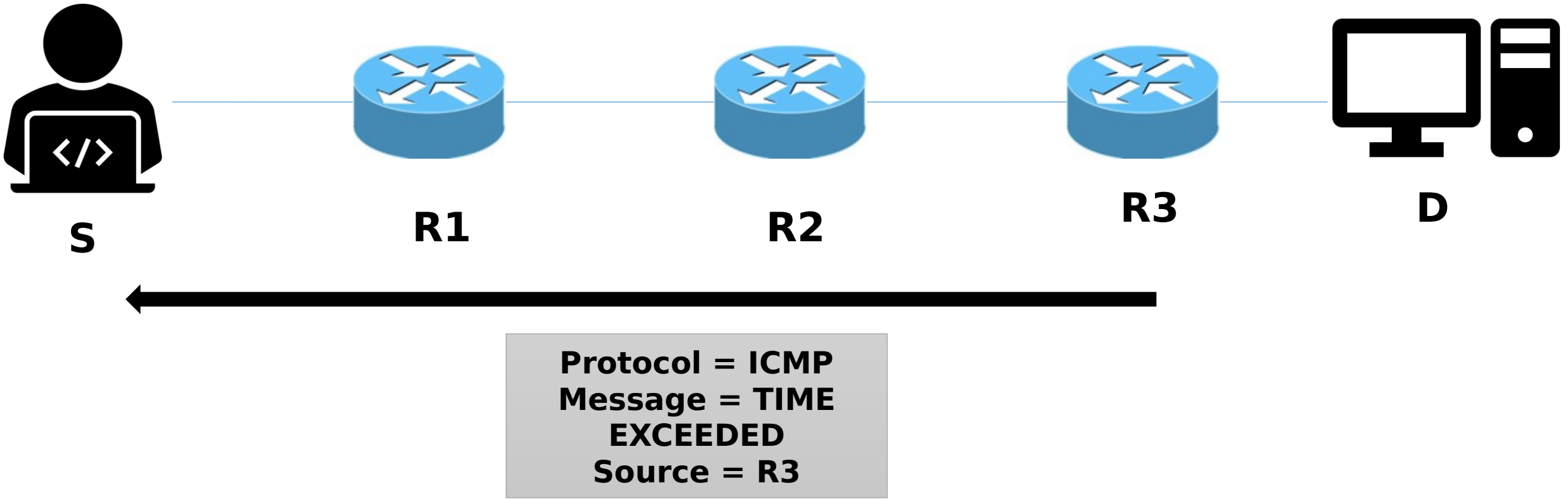
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



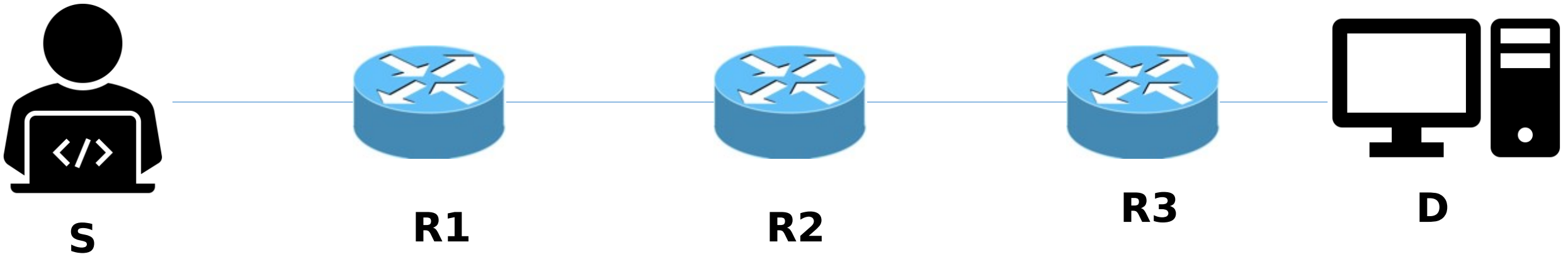
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Traceroute

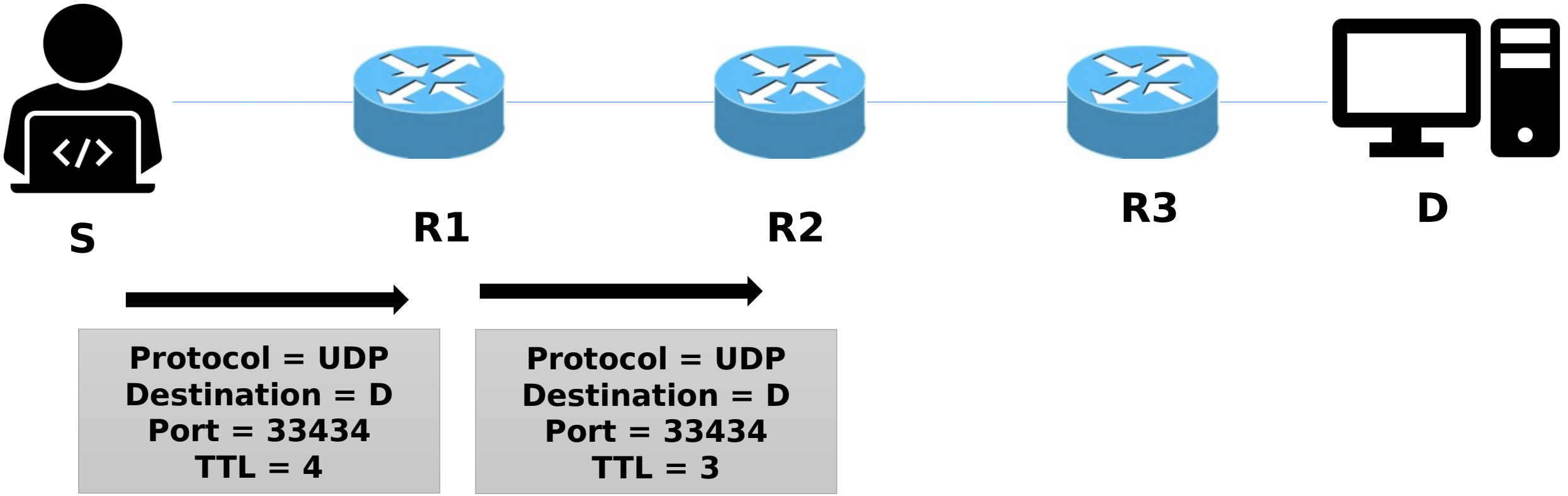
- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Protocol = UDP
Destination = D
Port = 33434
TTL = 4

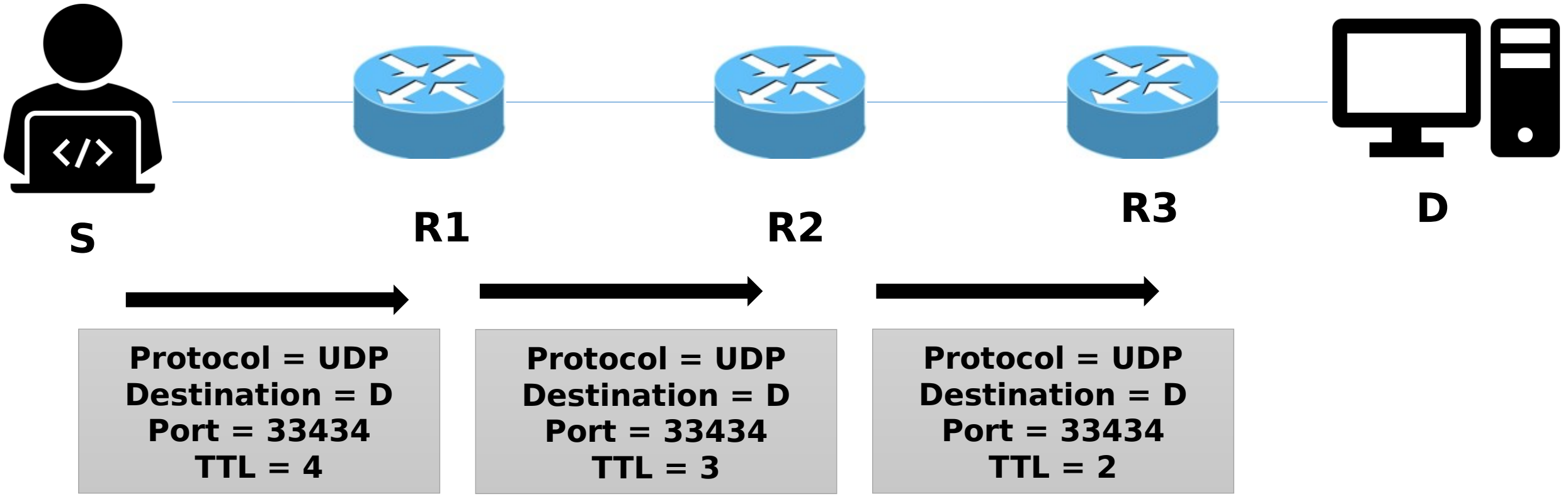
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



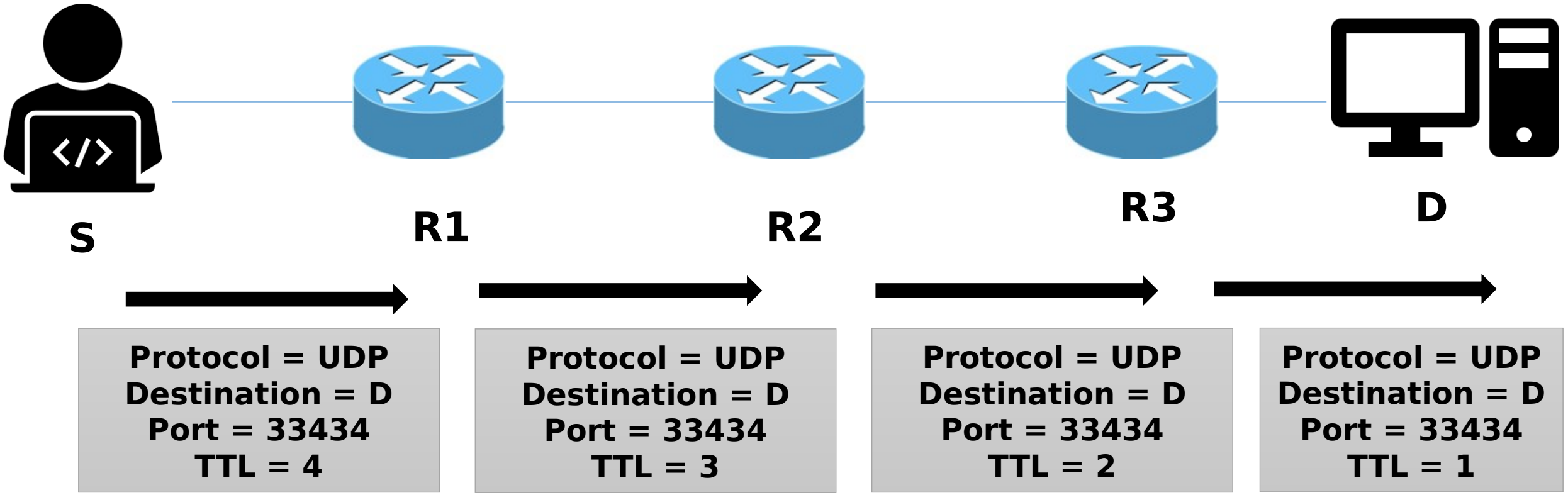
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



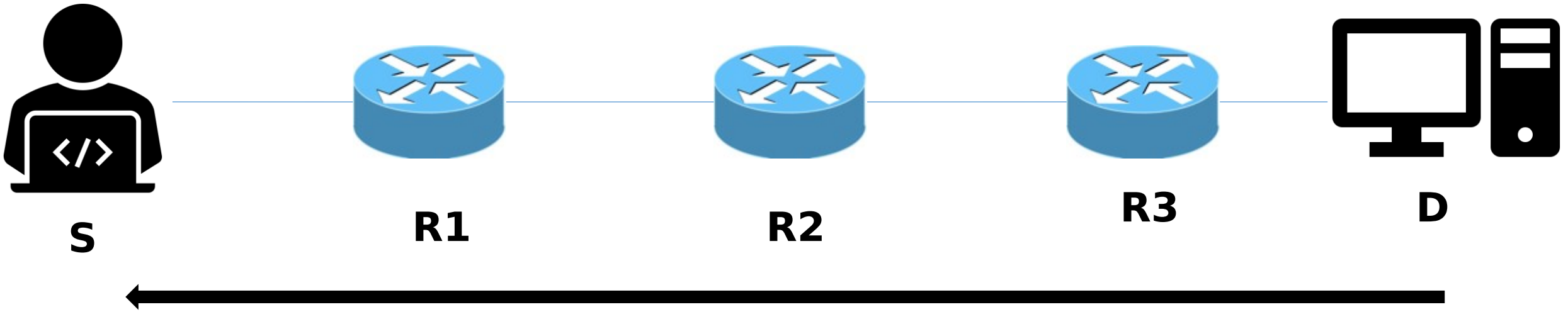
Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Traceroute

- Traceroute works using two ICMP messages, ICMP Time Exceeded
  and ICMP Destination Unreachable



Protocol = ICMP
Message = DESTINATION UNREACHABLE
Source = D


thank you!