CSC4200/5200 – COMPUTER NETWORKING

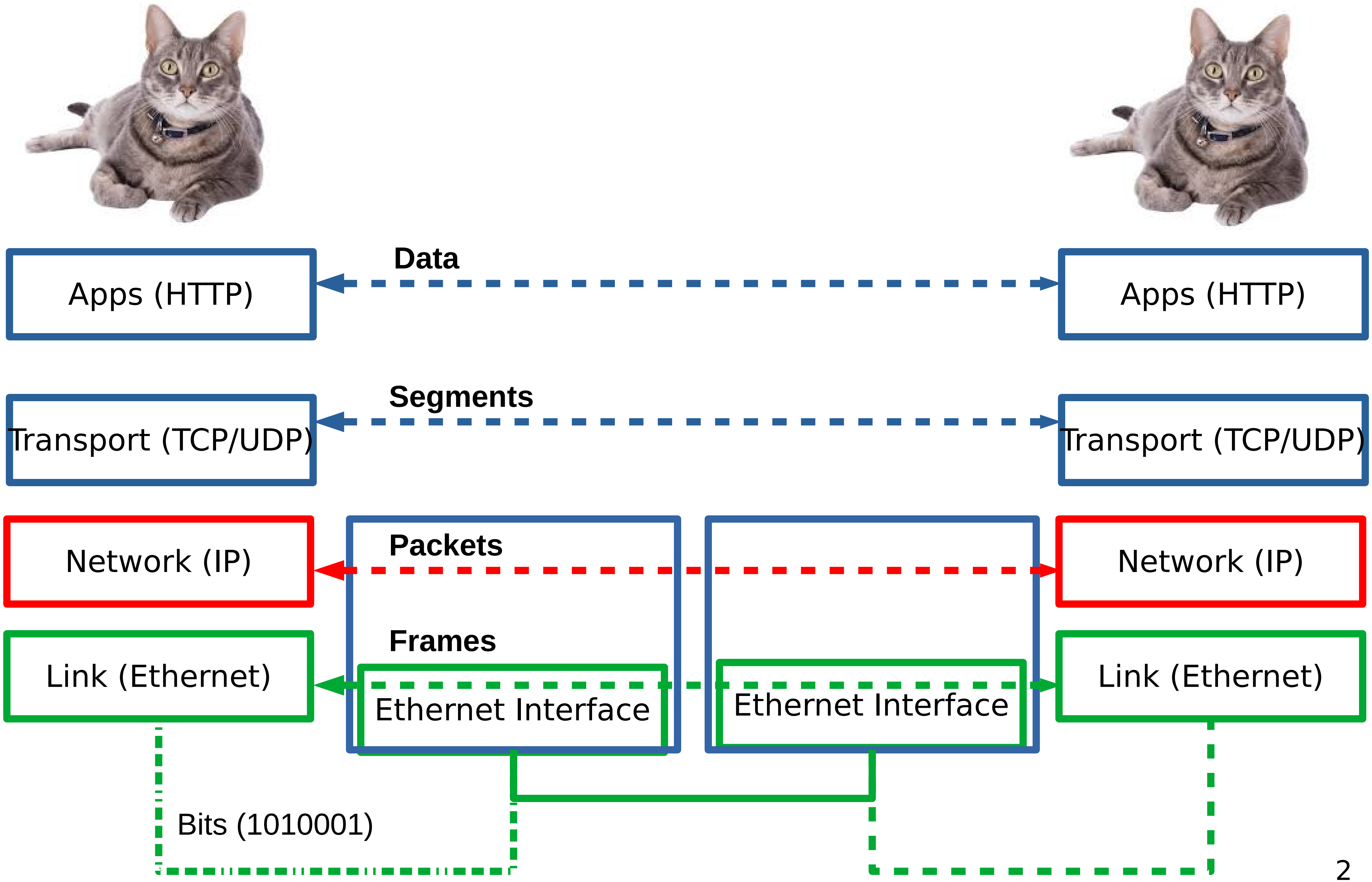
Instructor: Susmit Shannigrahi

ARP AND DHCP

sshannigrahi@tntech.edu

GTA: dereddick42@students.tntech.edu

Tennessee TECH

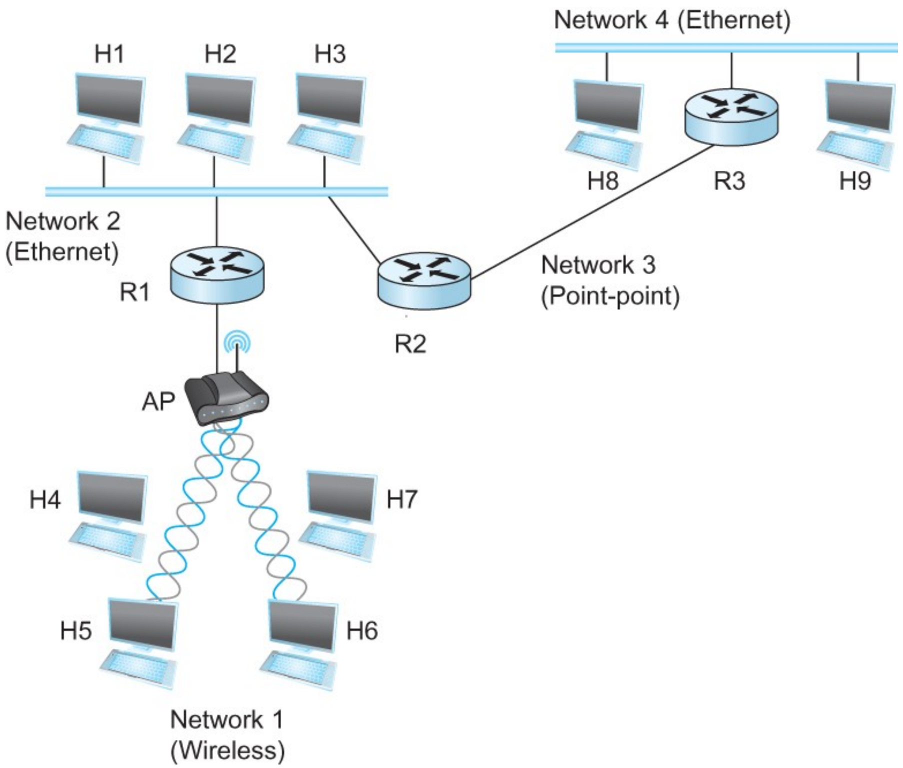

Apps (HTTP)
Data
Apps (HTTP)
Transport (TCP/UDP)
Segments
Transport (TCP/UDP)
Network (IP)
Packets
Network (IP)
Link (Ethernet)
Frames
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

So far...

- We now know how to address hosts and networks!
- Subnetting for scale

Internetworking Protocol (IP)

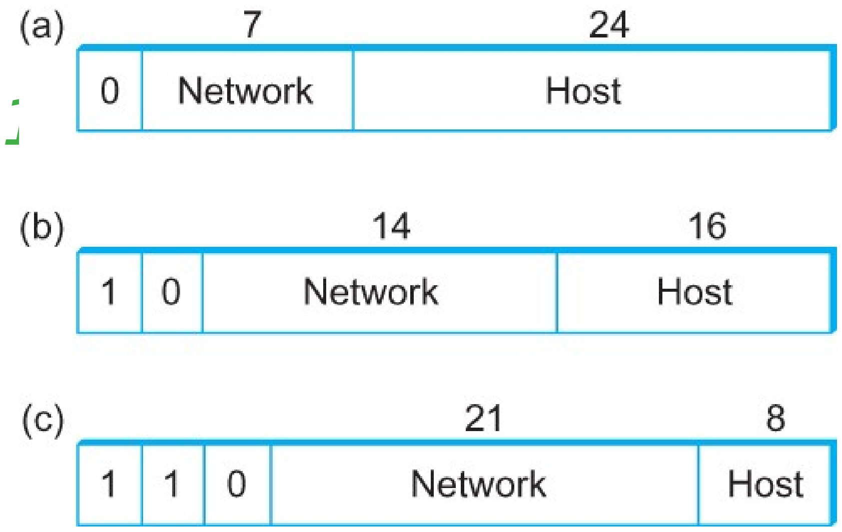
- What is an internetwork?
  - An arbitrary collection of networks
  - provide some sort of host-host to packet delivery service



Global Address in IP – Each node has an unique address

- A 32 bit number in quad-dot notation
- Identifies an ***Interface***
  - ***A host might have several interfaces!!!***
- *129.82.138.254*

  10000001.01010010.10001010.11



IP addresses are in Network + Host

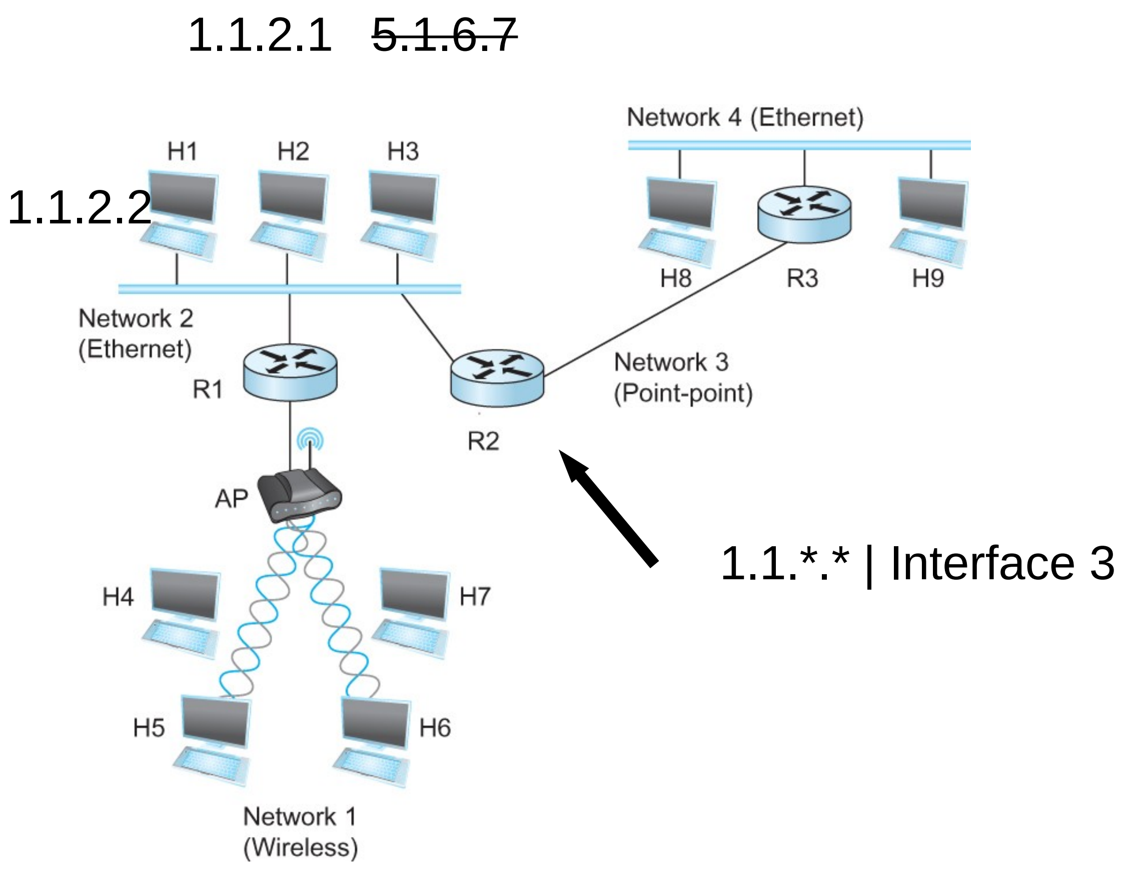
- 1.1.2.1 →
  - 1.1 → Network part
  - 2.1 → host part
- Each octet can range from 1- 255

- Hierarchical address

129.82.138.254

10000001.01010010.10001010.11111110

Network part (24 bits). Host part(8 bits)



Calculate the first and the last IP address of a subnet

129.82.138.254/27

First host - host bits 0

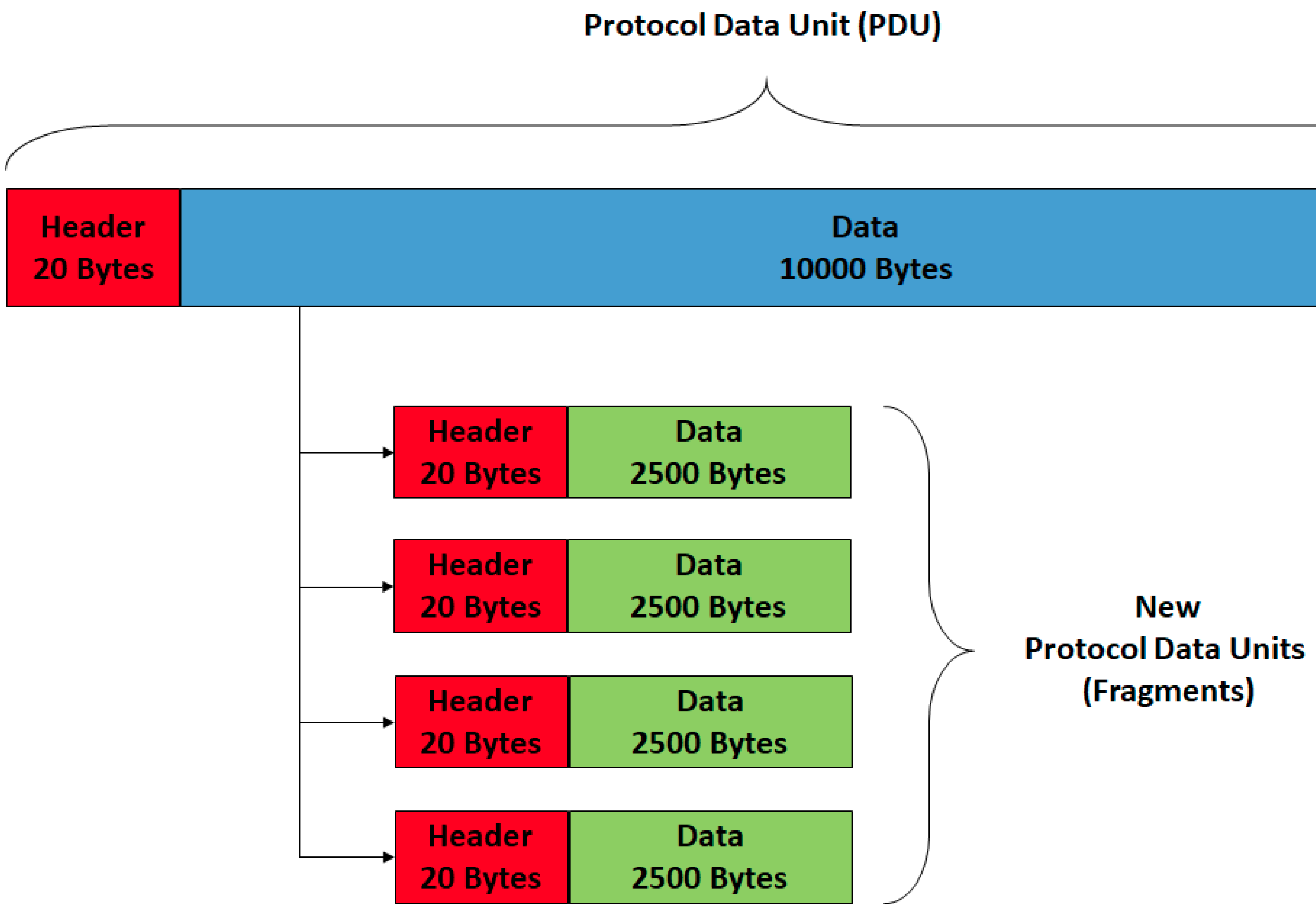
10000001.01010010.10001010.11111110
11111111.11111111.11111111.11100000 (LOGICAL AND)
___
10000001.01010010.10001010.11100000 → 129.82.138.224

Last host – host bits 1

10000001.01010010.10001010.11111110
11111111.11111111.11111111.11111111 (LOGICAL AND)
___
10000001.01010010.10001010.11111110 → 129.82.138.255

Perform logical AND to get the network part = 129.82.138.224
Available addresses – 129.82.138.225-129.82.138.254
Broadcast address – 129.82.138.255

IP Fragmentation and Reassembly



wikipedia

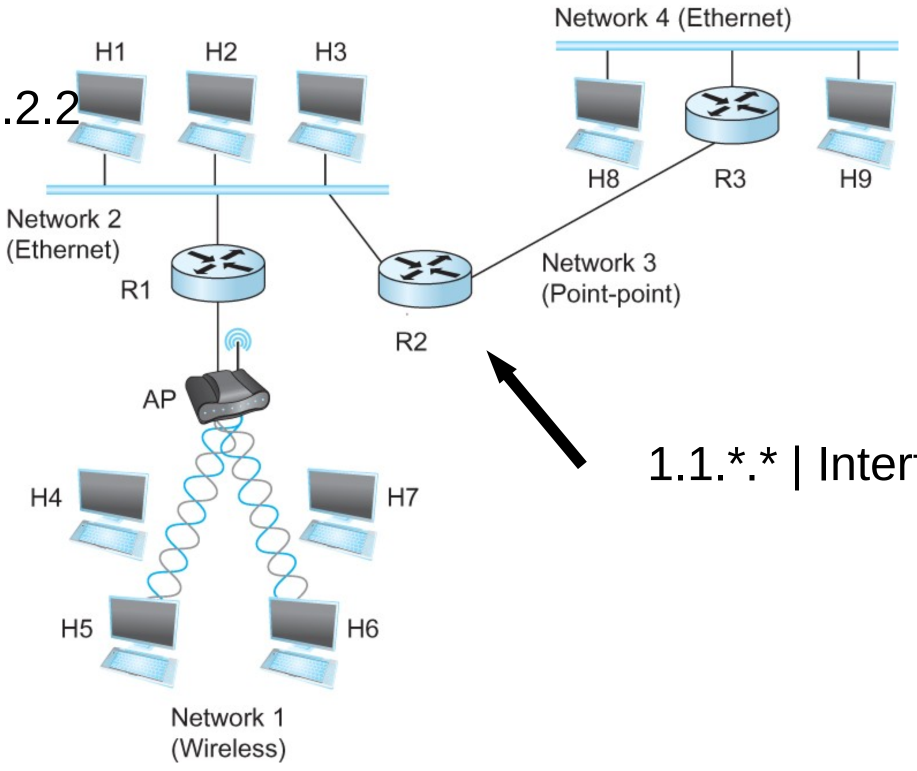
IP addresses are in Network + Host

- 1.1.2.1 →
  - 1.1 → Network part
  - 2.1 → host part
- Each octet can range from 1- 255

- Hierarchical address

1.1.2.1 ~~5.1.6.7~~

1.1.2.2



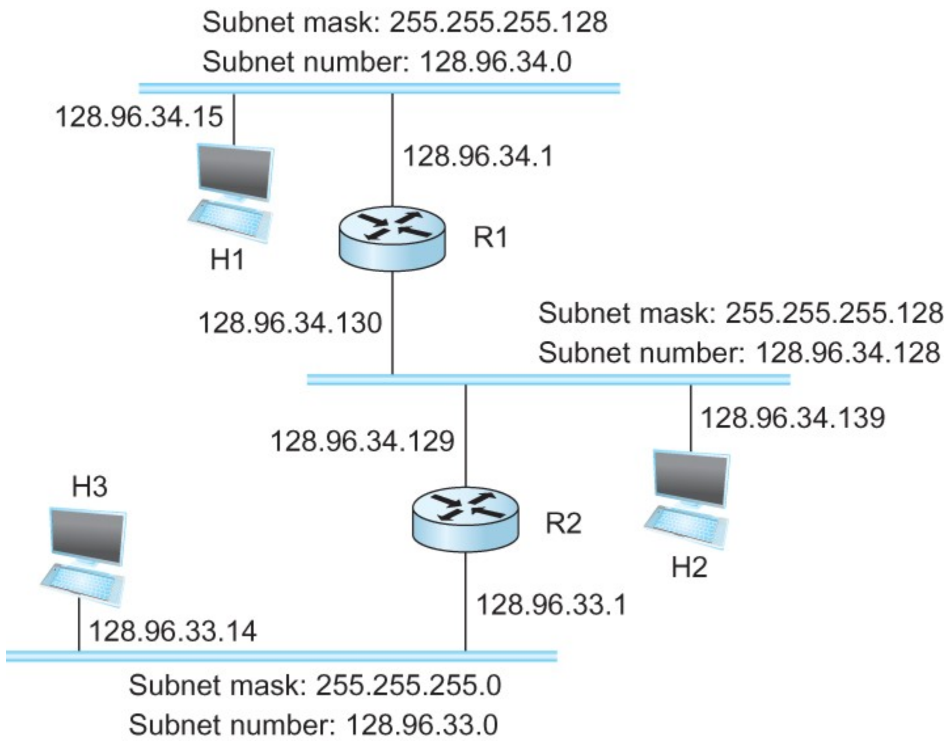
1.1.*.* | Interface 3

129.82.138.254

10000001.01010010.10001010.11111110

Network part (24 bits). Host part(8 bits)

Subnetting



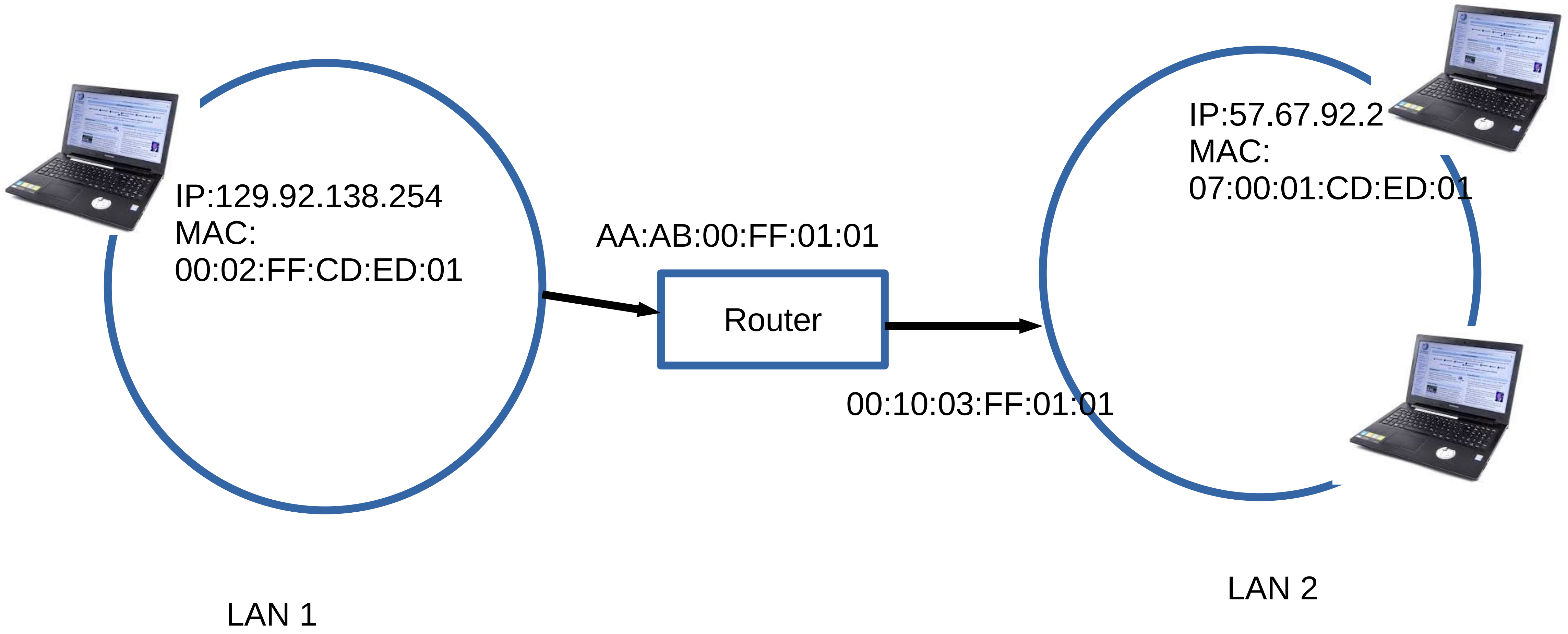
Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Now let’s map that to MAC address

- Adaptors only understand MAC addresses
- Source: 129.82.138.254, Destination: 129.82.138.5
- You machine does not know what that means:
  - Routers for getting you to the room
  - In the room, you still need to use the MAC address
- Put IP packet in a frame → **Encapsulation**

IP ↔ MAC mapping: Address Resolution Protocol (ARP)

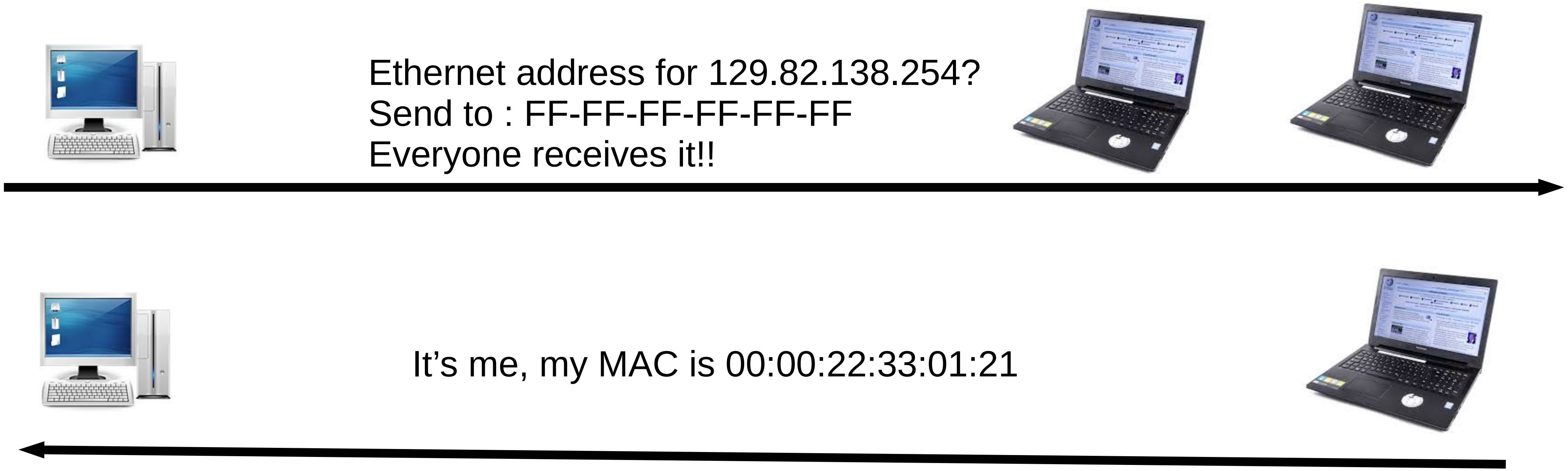


IP ↔ MAC mapping: Address Resolution Protocol (ARP)

- Important concept → Broadcast
  - Shout in the room → Who here is Rachel?

ARP table

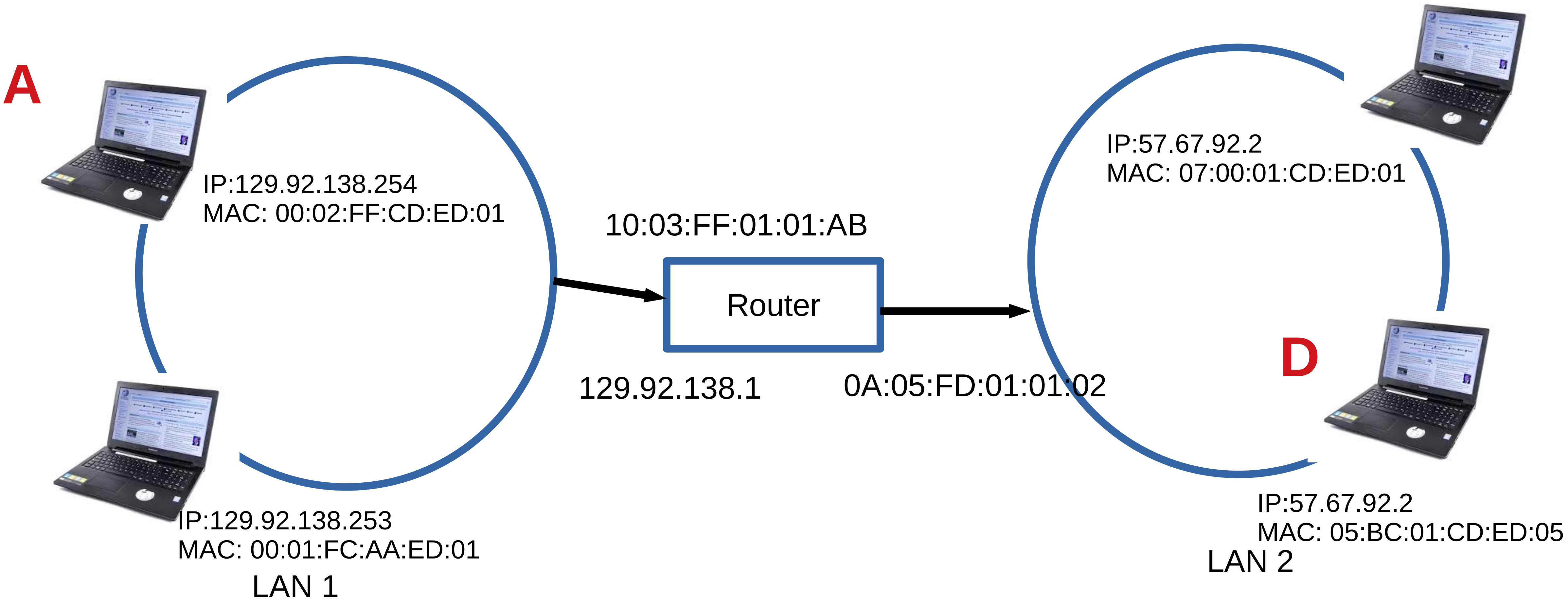
- Important concept → Broadcast
  - Shout in the room → Who here is Rachel?



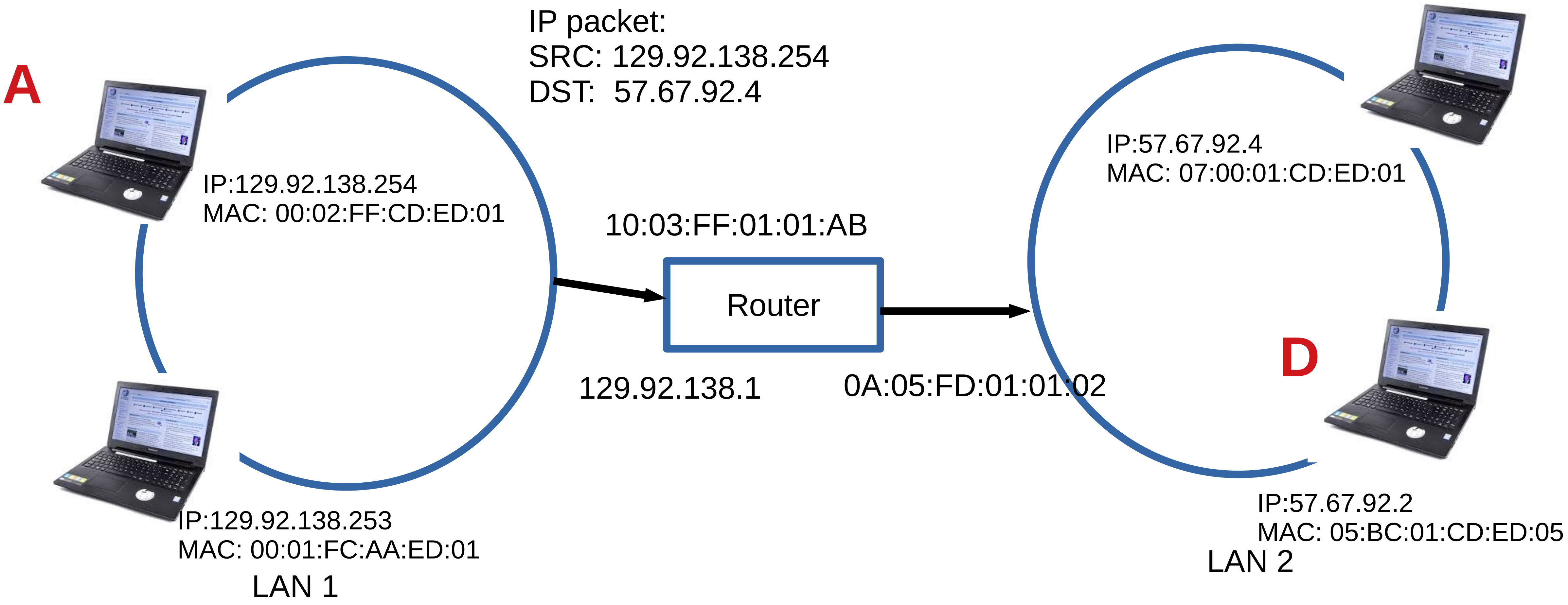
IP ↔ MAC mapping: Address Resolution Protocol (ARP)

- Every node maintains an ARP table
  - <MAC, IP> mapping
- Consult this table when sending IP packets
- Encapsulate with the MAC address, send it the address
- If address is not known, broadcast!
- Cache the response for some time, and eventually forget
  - **Why not broadcast the IP packet?**

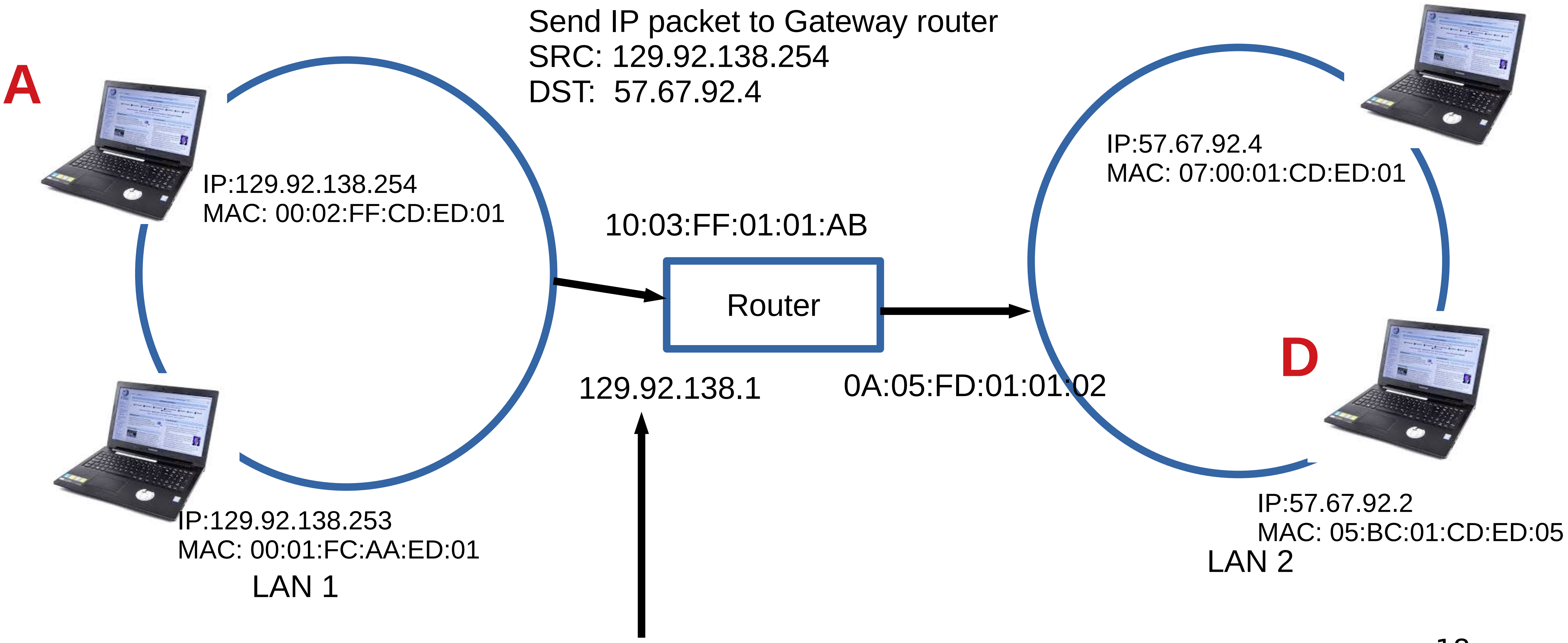
How does A talk to D?



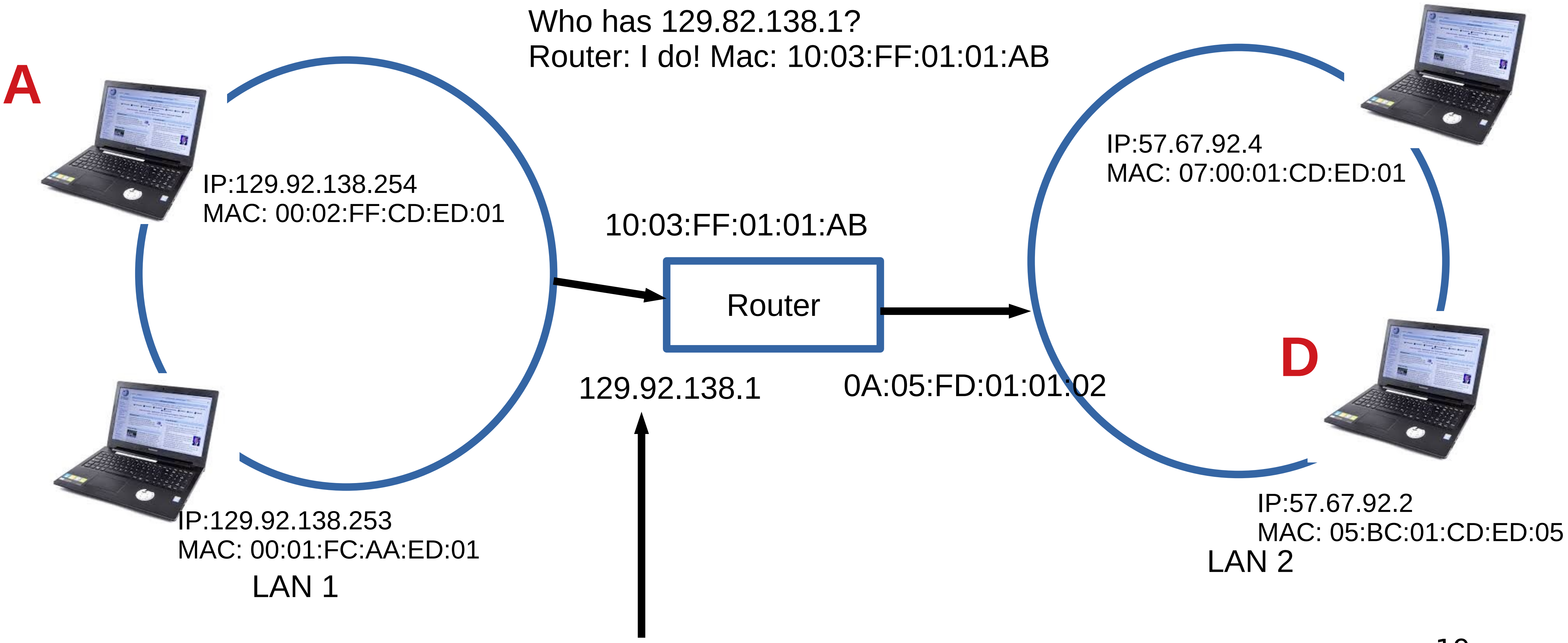
How does A talk to D?



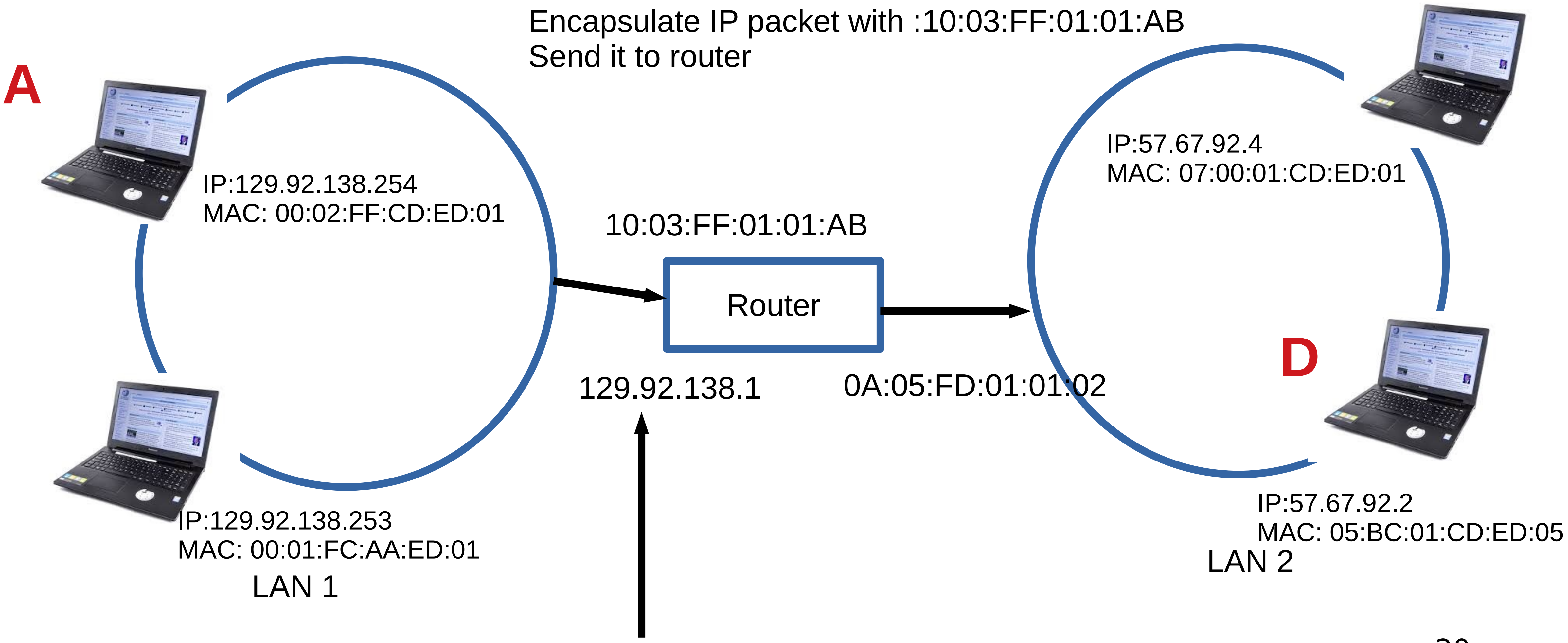
How does A talk to D?



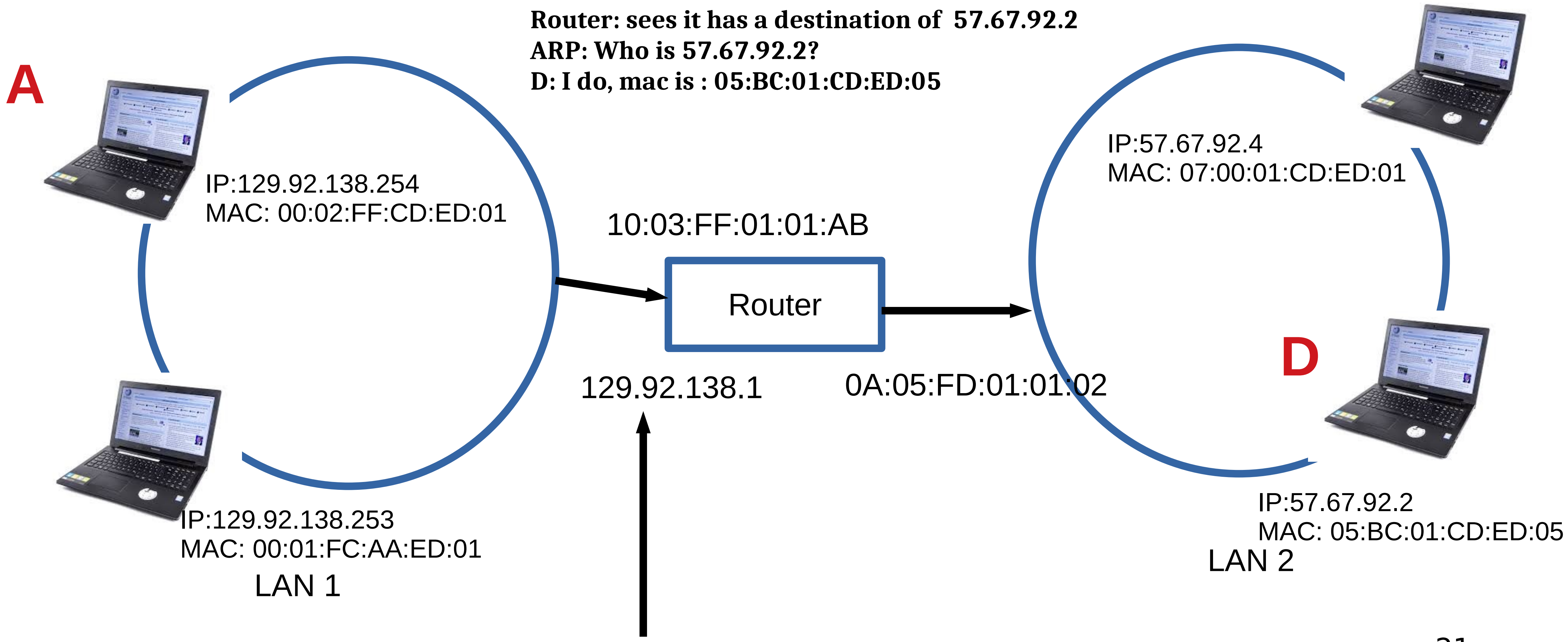
How does A talk to D?



How does A talk to D?

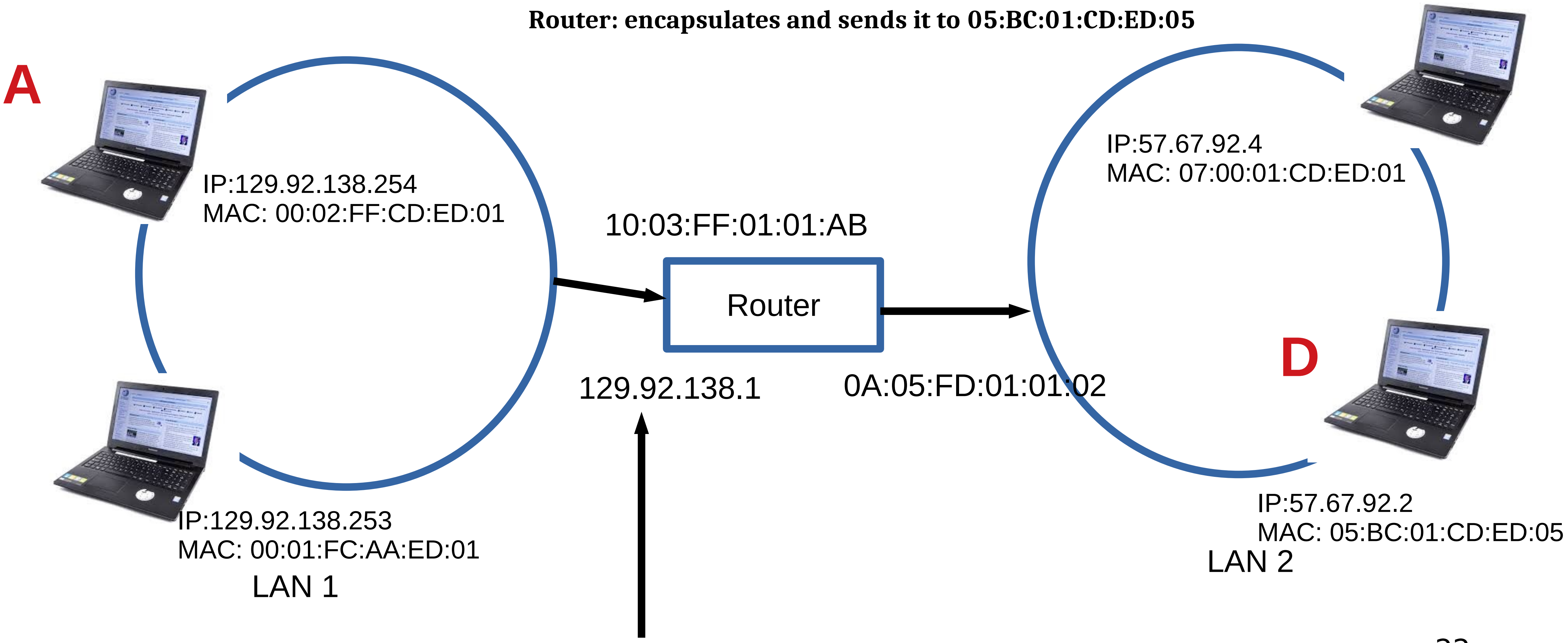


How does A talk to D?



How does A talk to D?

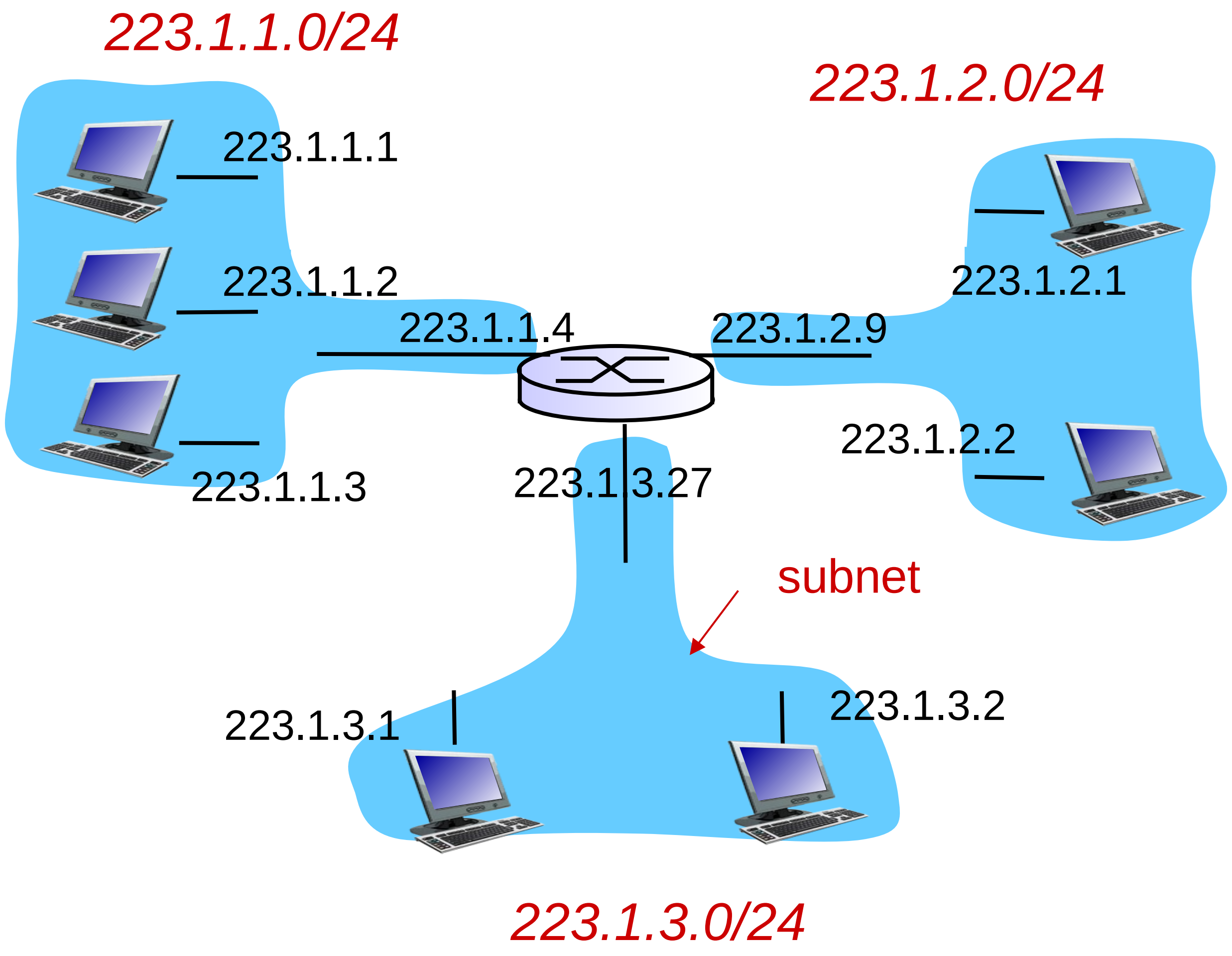
Router: encapsulates and sends it to 05:BC:01:CD:ED:05



Subnets Revisited

recipe

- to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- each isolated network is called a *subnet*



subnet mask: /24

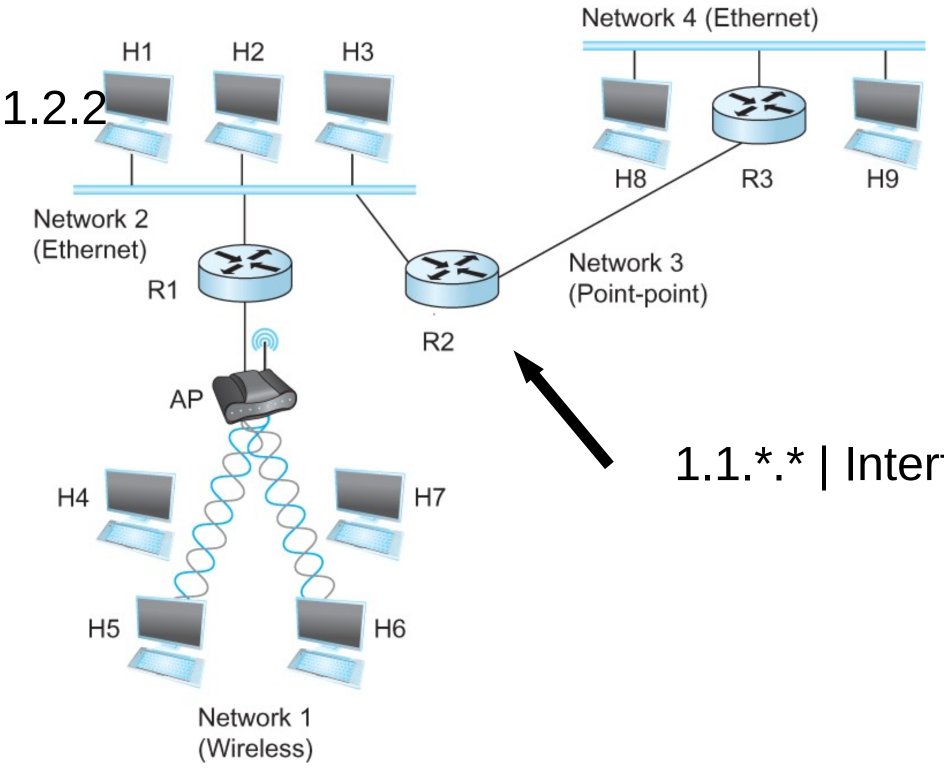
IP addresses are in Network + Host

- 1.1.2.1 →
  - 1.1 → Network part
  - 2.1 → host part
- Each octet can range from 1- 255

- Hierarchical address

1.1.2.1 ~~5.1.6.7~~

1.1.2.2



1.1.*.* | Interface 3

129.82.138.254

10000001.01010010.10001010.11111110

Network part (24 bits). Host part(8 bits)

Calculate the first and the last IP address of a subnet

129.82.138.254/27

First host - host bits 0

10000001.01010010.10001010.11111110
11111111.11111111.11111111.11100000 (LOGICAL AND)

10000001.01010010.10001010.11100000 → 129.82.138.224

Last host – host bits 1

10000001.01010010.10001010.11111110
11111111.11111111.11111111.11111111 (LOGICAL AND)

10000001.01010010.10001010.11111110 → 129.82.138.255

Perform logical AND to get the network part = 129.82.138.224
Available addresses – 129.82.138.225-129.82.138.254
Broadcast address – 129.82.138.255

Problem

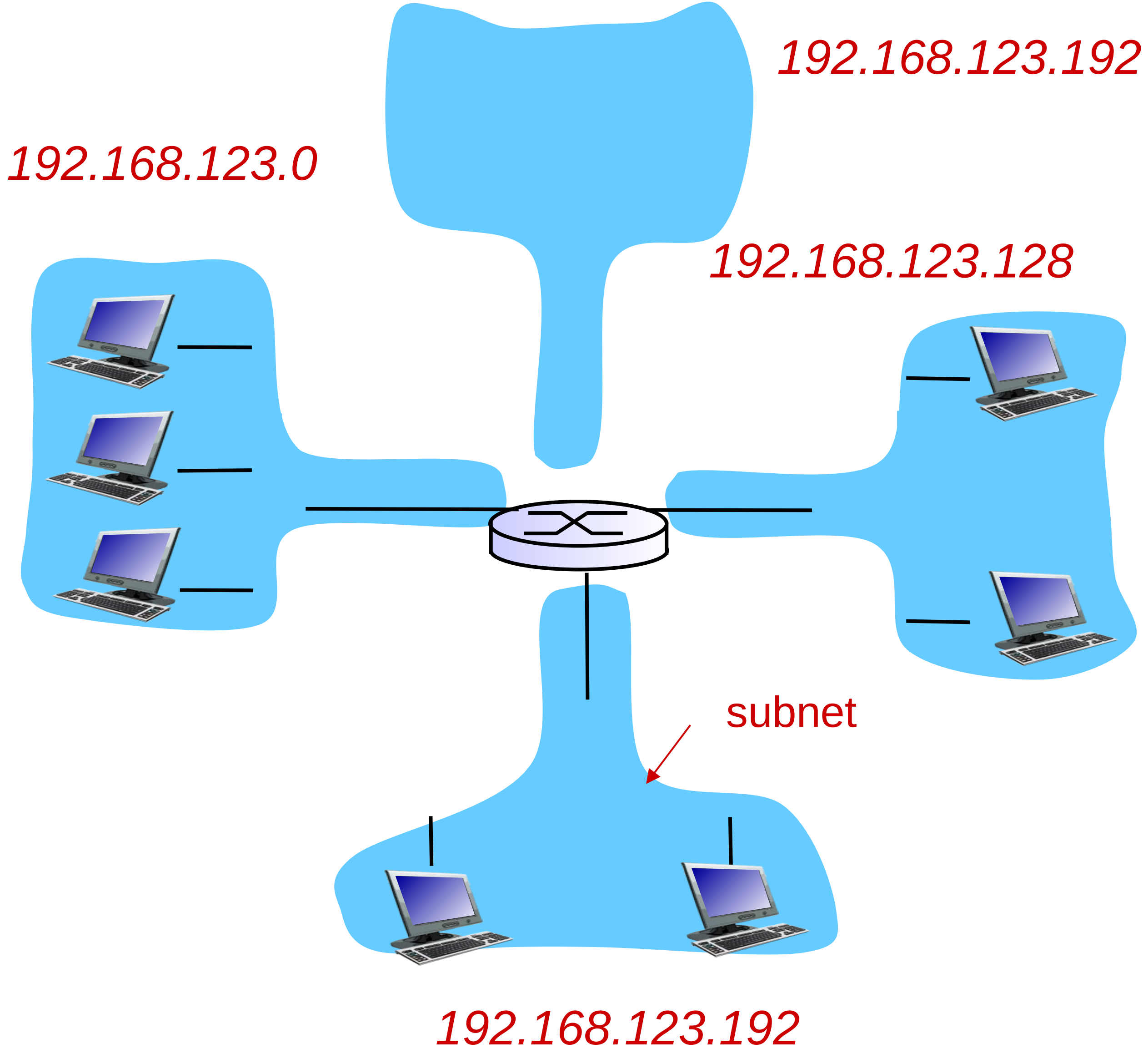
You have an address block:
192.168.123.0/24

- CSC needs 50 addresses
- Library needs 50
- Math needs 50
- ME needs 50

They can not overlap!
Borrow some bits from the host part.

24 bits - 1111111.11111111.1111111.00000000
2 bits for network –
1111111.11111111.1111111.11000000

- *How many networks?*
- *How many hosts in each of these networks?*

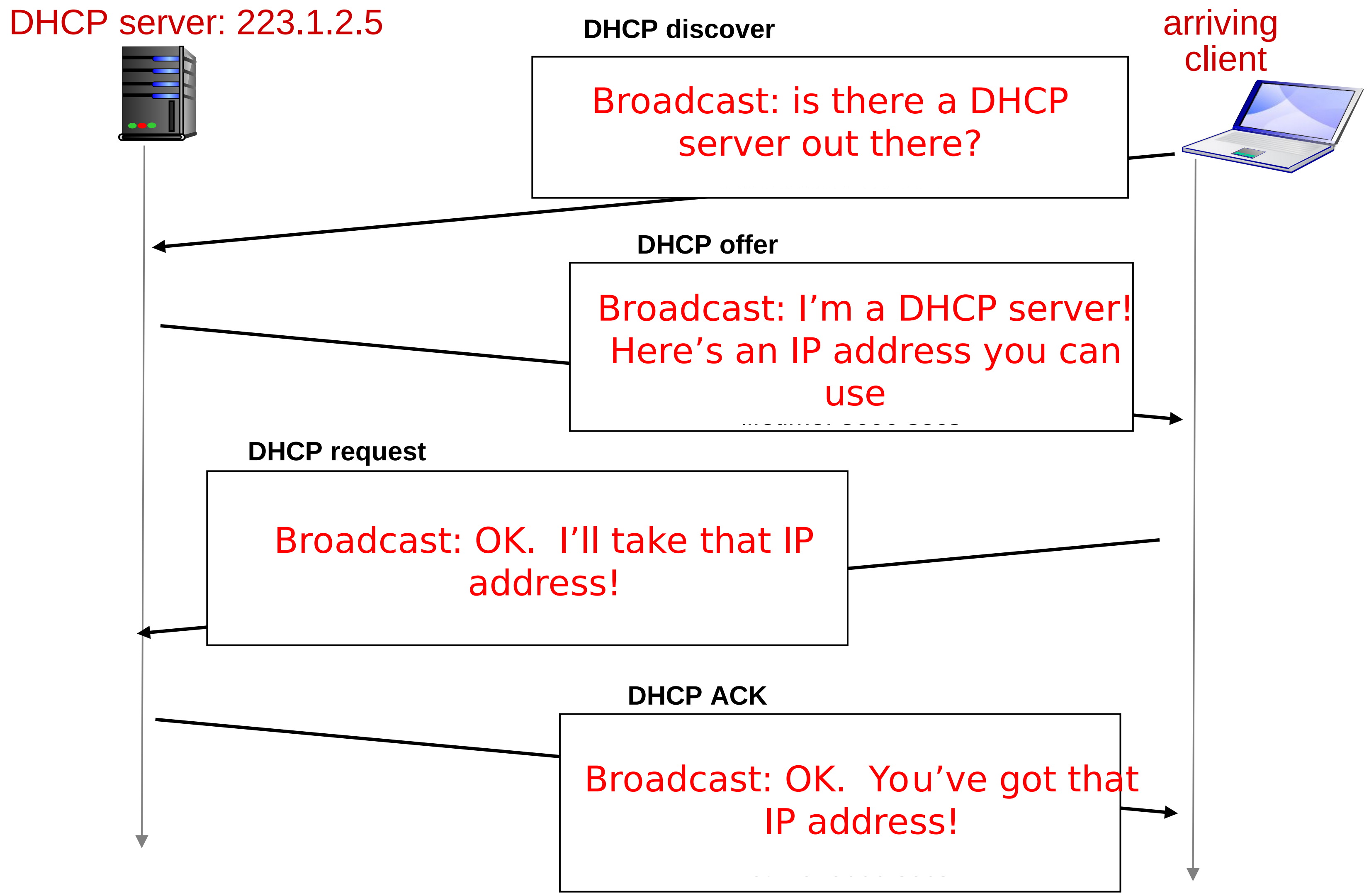


subnet mask: /26

DHCP

- **New laptop joins a network**
  - Does not have source address
  - Does not know who to ask
  - Does not know other network parameters like DNS or Gateway router information

DHCP client-server scenario



kurose/ross

DHCP Server

- A local central database with a list of IP addresses
  - 10.0.0.1/8
- Offers an available IP to a client for a period of time
  - Lease time – 24 hours, 1 hour, configurable ← ***Soft State***
- Multiple servers might coexist and offer IP to the same request
  - Broadcast medium
  - Client decides which one to accept

DHCP Client – Keep refreshing!

- IP address provided expires after time ***t***

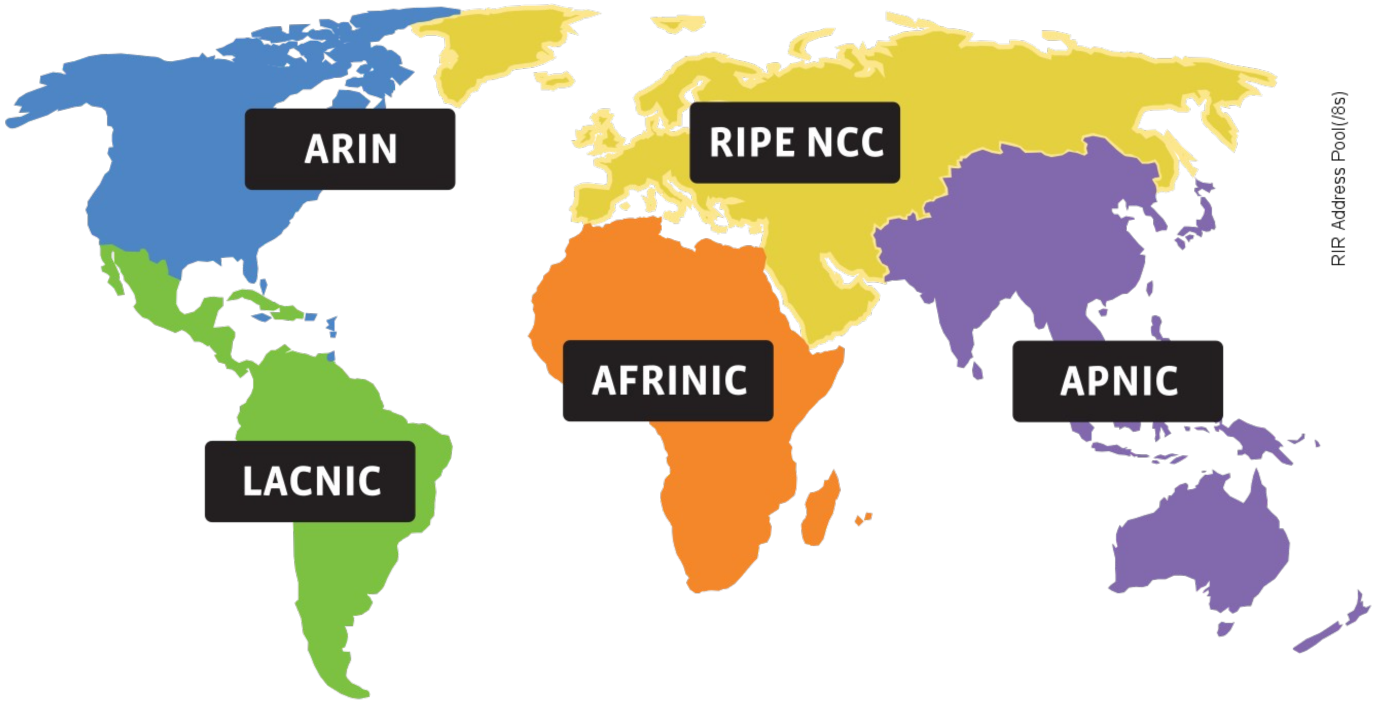
- Client can release DHCP lease
  - Shutdown the laptop

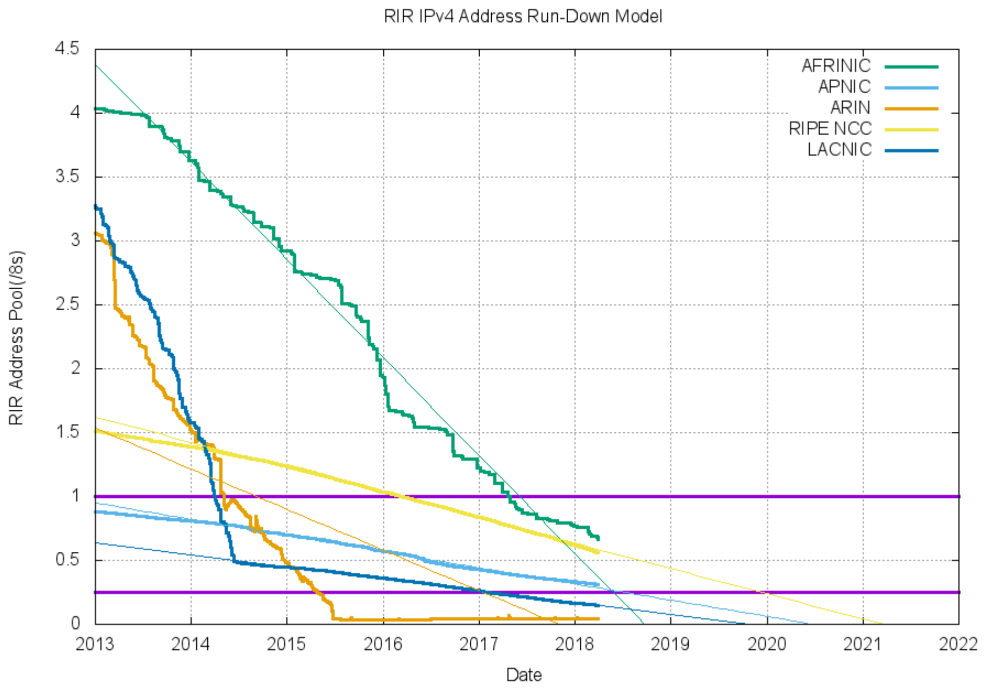
- If you walk away from the building
  - Crash

- Performance trade off
  - Short time – too many broadcasts, quick recovery of addresses
  - Long time – less network traffic, longer recovery of addresses

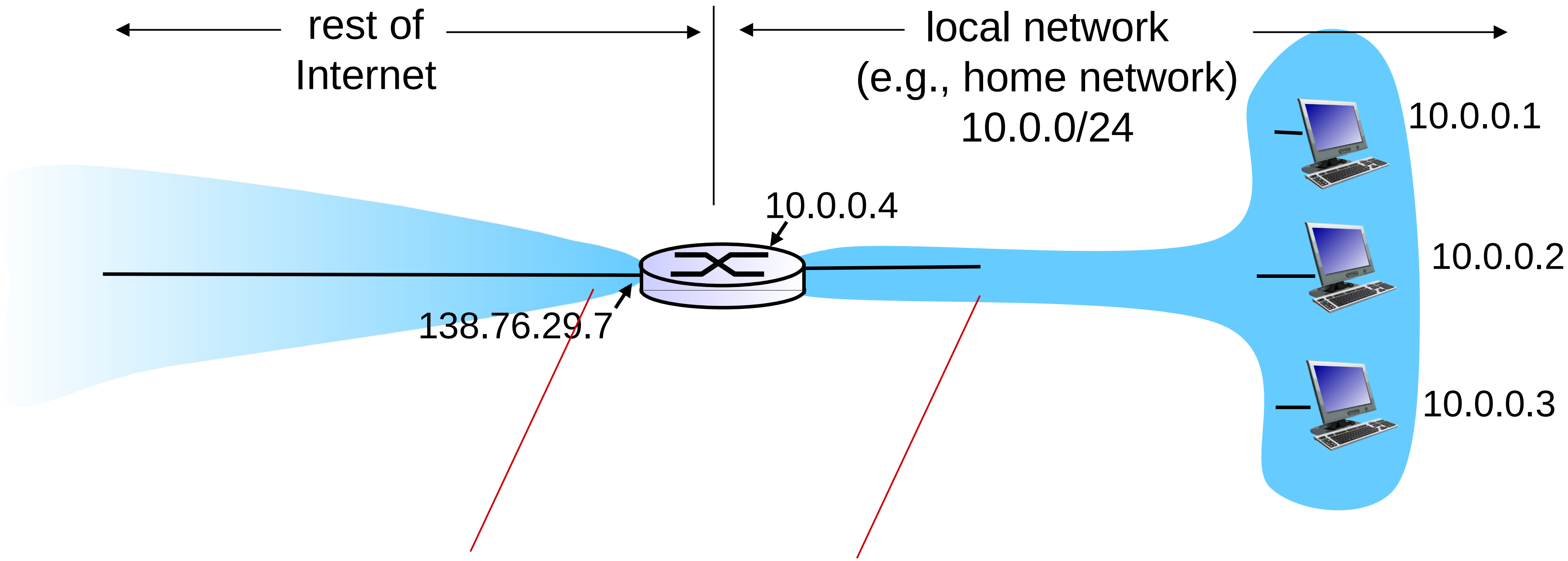
Address shortage

- IPv4 – 32 bits – Around 4 billion



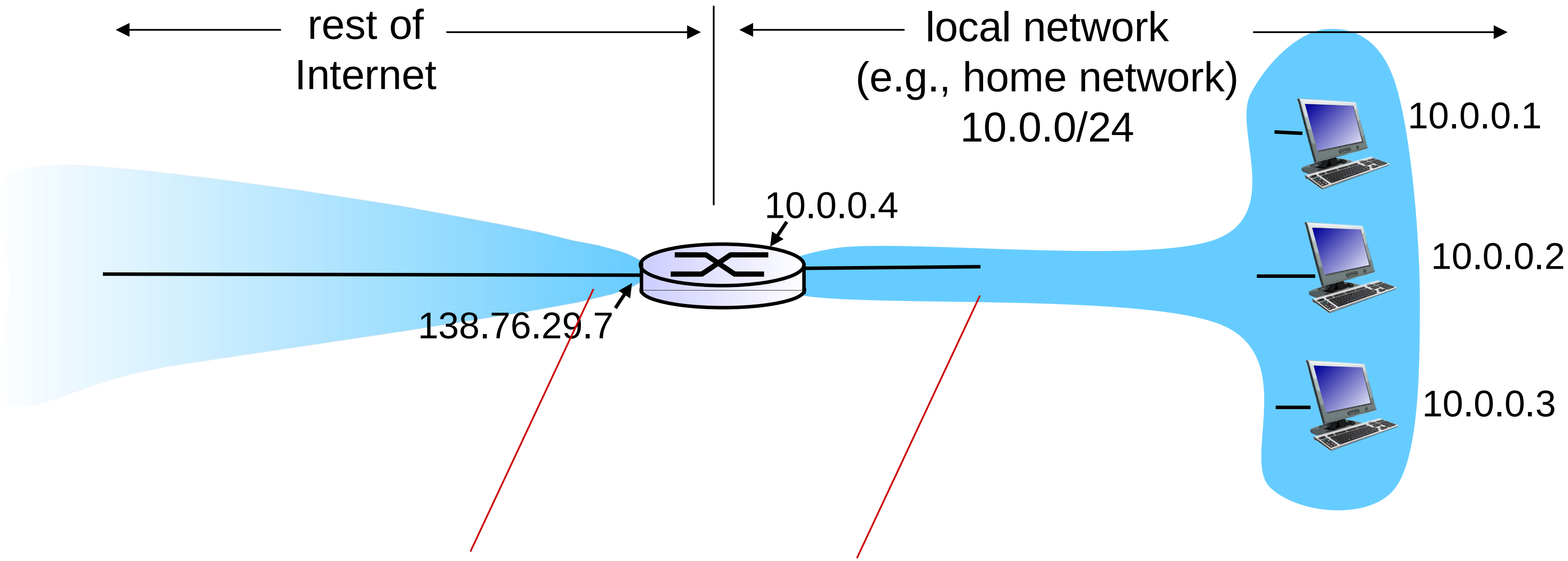


NAT: network address translation



datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: Network Address Translation



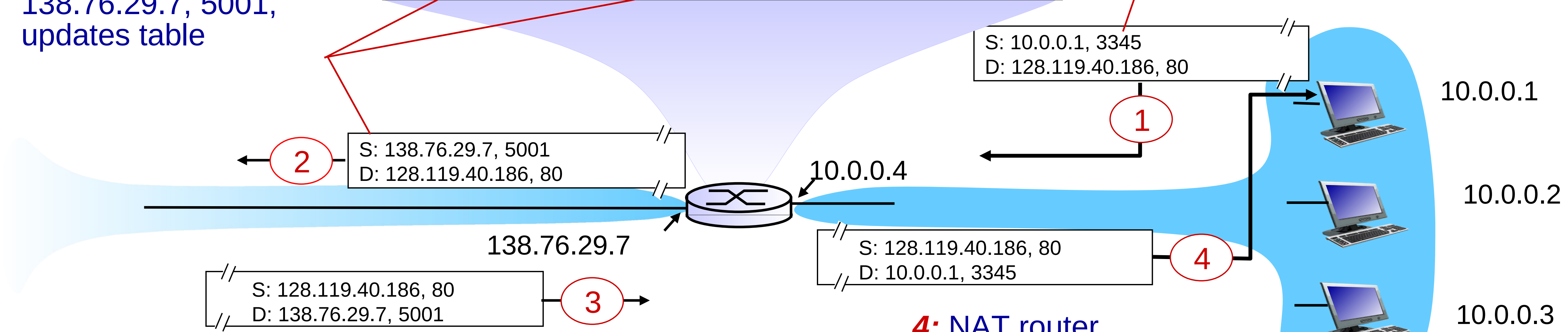
datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

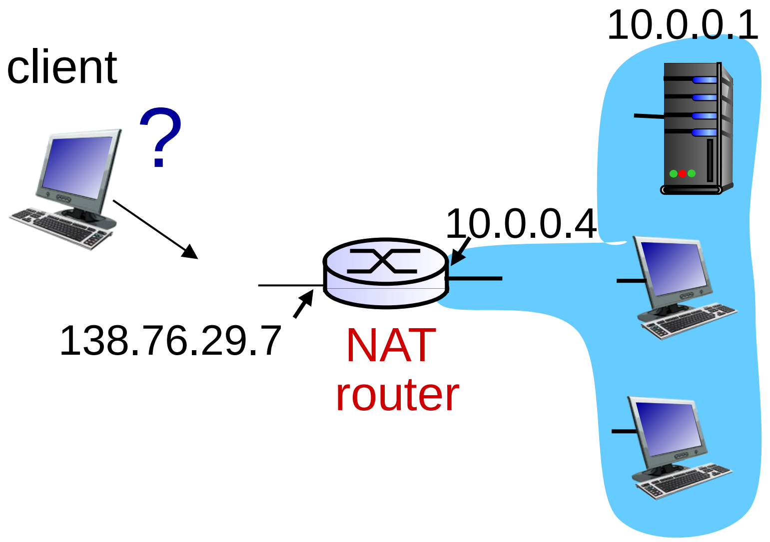


3: reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

NAT

- One IP address for all devices
  - Addresses the address space problem
- Can change local addresses without involving the ISP
- NAT traversal problem
  - Is a server is behind NAT, how does the client talk to it?



https://samsclass.info/ipv6/exhaustion-2016.htm

Address shortage – Better solution? IPv6

- IPv4 – 128 bits

There are only this many IPv6 addresses left:

340,282,366,920,938,463,463,374,607,430,530,552,200

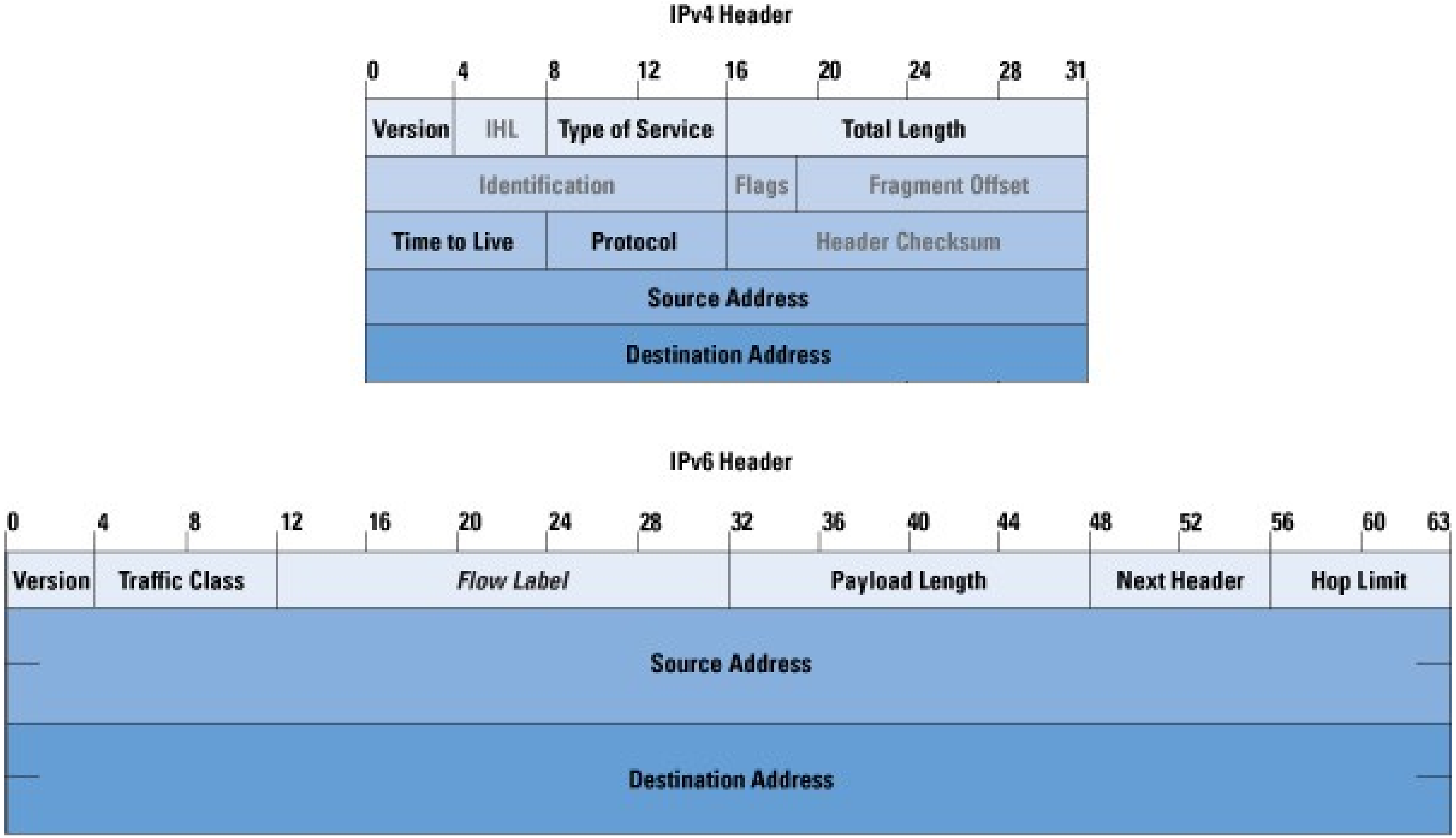
Projected IPv6 Exhaustion Date

9,000,000 AD

https://samsclass.info/ipv6/exhaustion-2016.htm

Address shortage – Better solution? IPv6

- IPv4 – 128 bits

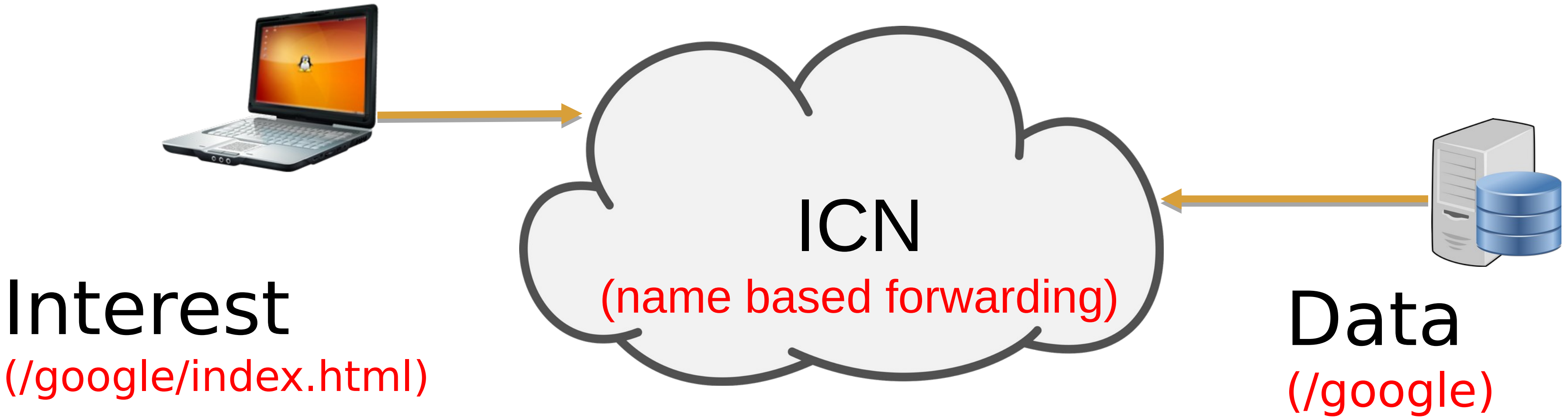


Address shortage – Better solution? Get rid of the Addresses!

- Next generation of the Internet

- You don’t care about the hosts anyway
  - For most part

- Why not ask for content directly?
  - Information Centric Networking (ICN)



ICMP: Internet Control Message Protocol

- Errors in network:
  - Router does not know how to forward a packet
  - Packet is broken

- IP is best effort
  - Can silently drop packets

- How would be ever know something is wrong?
  - Feedback about the problem
  - ICMP

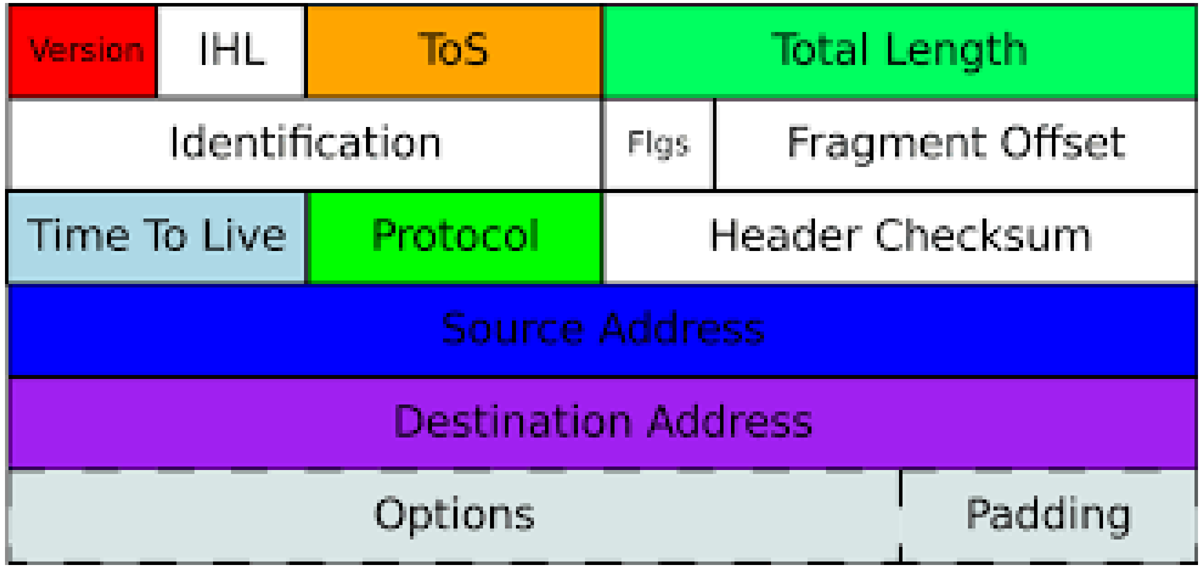
ICMP: Internet Control Message Protocol

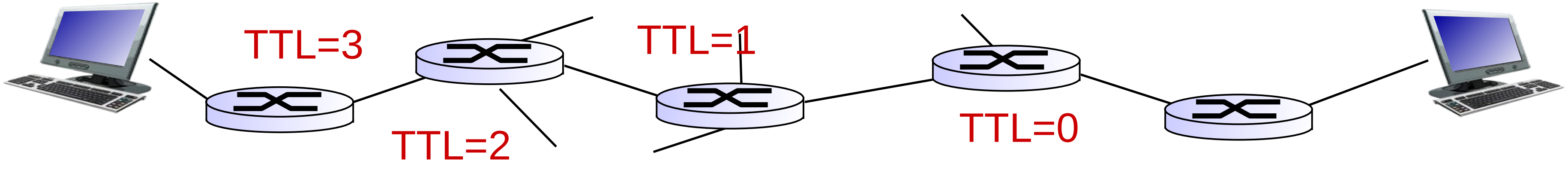
- Used for
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- Application at network-layer
  - ICMP msgs carried in IP datagrams
  - Essentially at application layer
  - Considered part of IP

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

ICMP and Time to Live

- Each time a host sends a packet it sets the TTL field
- Each router that forwards it decrements the number
- When TTL reaches 0, send a time exceeded message



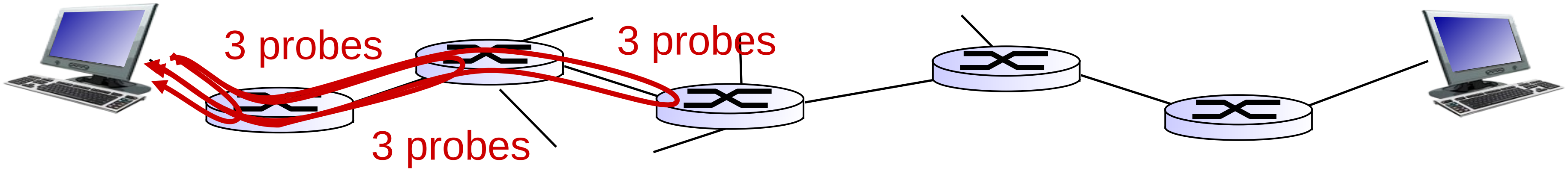


Traceroute and ICMP

- source sends series of UDP segments to dest
  - first set has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when *n*th set of datagrams arrives to nth router:
  - router discards datagrams
  - and sends source ICMP messages (type 11, code 0)
  - ICMP messages includes name of router & IP address
- when ICMP messages arrives, source records RTTs

stopping criteria:

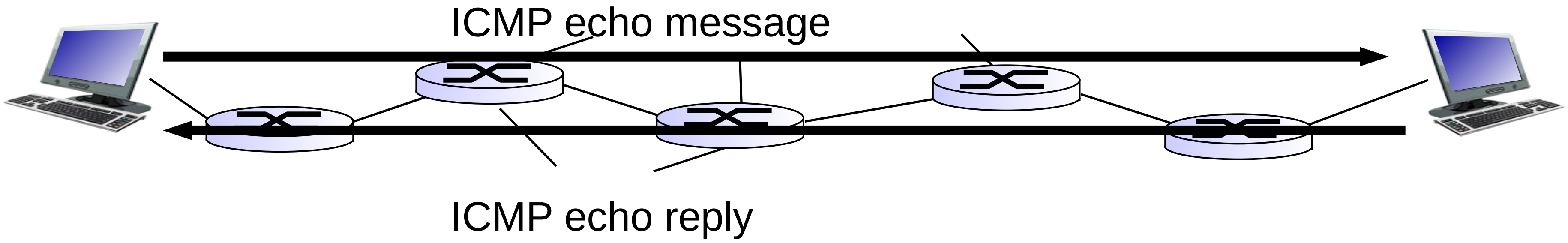
- UDP segment eventually arrives at destination host
- destination returns ICMP “port unreachable” message (type 3, code 3)
- source stops

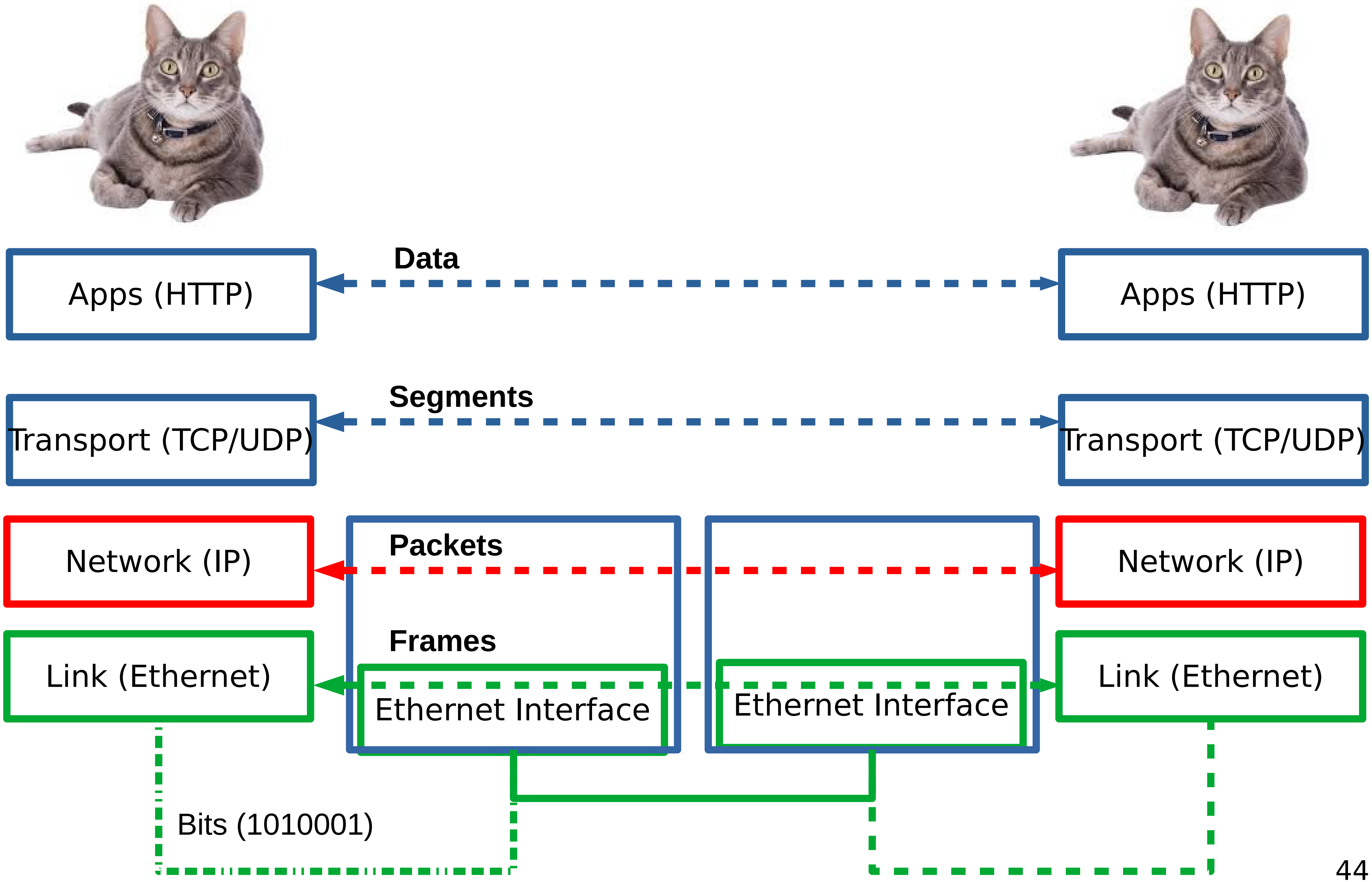


Ping and ICMP

- source sends an ICMP echo message

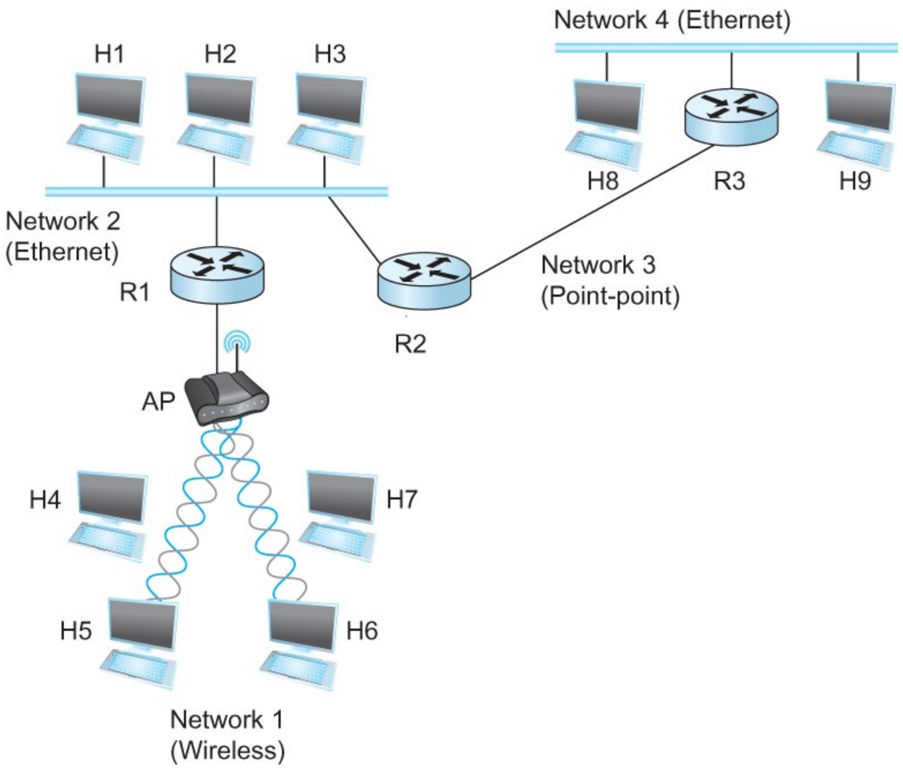
- Destination sends an ICMP echo reply




Apps (HTTP)
Data
Apps (HTTP)
Transport (TCP/UDP)
Segments
Transport (TCP/UDP)
Network (IP)
Packets
Network (IP)
Frames
Link (Ethernet)
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

Tying it all together in the network layer

Internetworking Protocol (IP)



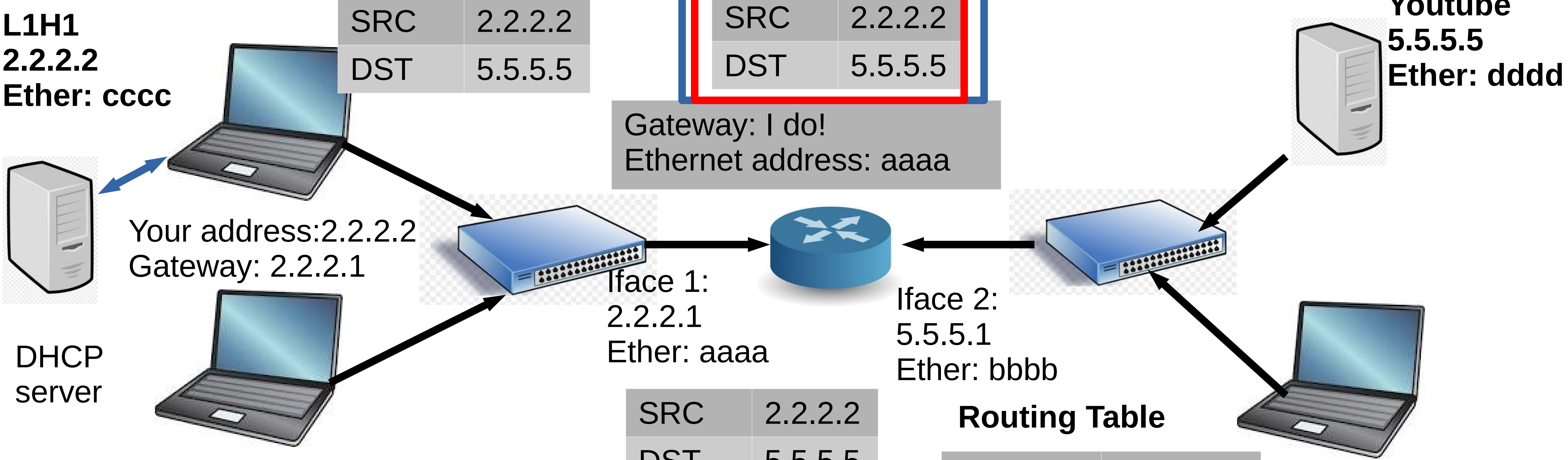
Tying it all together in the network layer

SRC	cccc
DST	aaaa

SRC	2.2.2.2
DST	5.5.5.5

Encapsulate IP packet in an Ethernet frame!

Gateway: I do!
Ethernet address: aaaa



SRC	2.2.2.2
DST	5.5.5.5

Routing Table

5.5.5.0/8	IF: 2
2.2.2.0/8	IF: 1

Decapsulate IP packet

DST in another network.
send it to the gateway.
gateway address 2.2.2.1
ARP: WHO HAS 2.2.2.1?

Tying it all together in the network layer

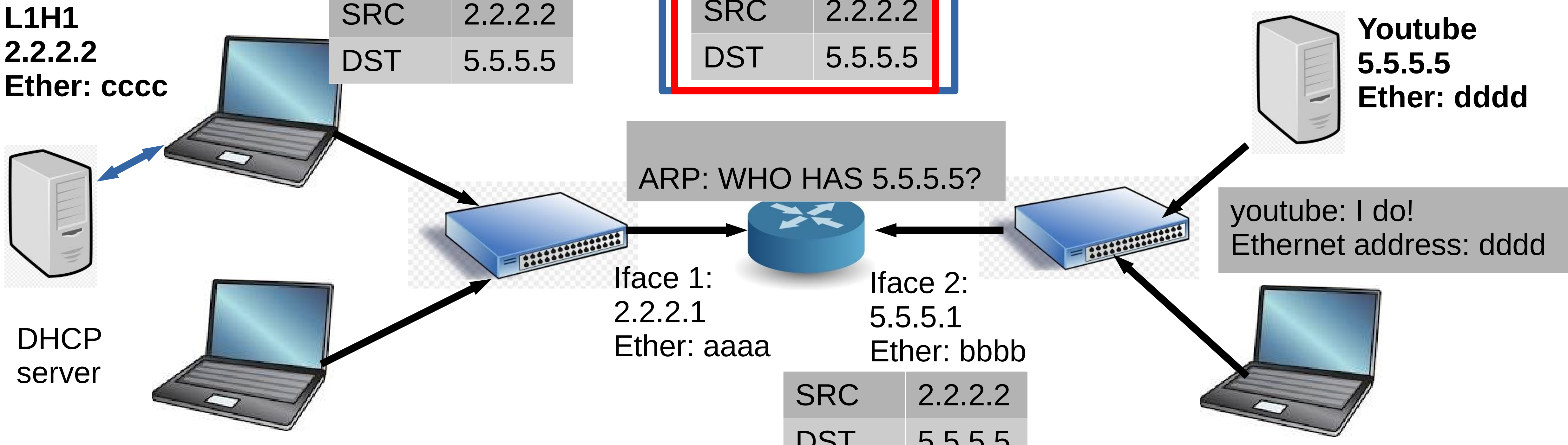
SRC	2.2.2.2
DST	5.5.5.5

Decapsulate IP packet

SRC	bbbb
DST	dddd

SRC	2.2.2.2
DST	5.5.5.5

SRC	2.2.2.2
DST	5.5.5.5



SRC	2.2.2.2
DST	5.5.5.5

Routing Table

5.5.5.0/8	IF: 2
2.2.2.0/8	IF: 1

Next Steps

Wait - how are the routing tables populated?
Read through chapter 3.2.

Very useful video: https://www.youtube.com/watch?v=rYodcvhh7b8