Network Security I

CSE 565: Fall 2024

Computer Security

Xiangyu Guo (xiangyug@buffalo.edu)

University at Buffalo

Acknowledgement

- We don't claim any originality of the slides. The content is developed heavily based on
  - Slides from Prof. Dan Boneh and Prof. Zakir Durumeric's lecture on Computer Security (https://cs155.stanford.edu/syllabus.html)
  - Slides from Prof Nick McKeown's lecture on Computer Network (https://vixbob.github.io/cs144-web-page/)
  - Slides from Prof Ziming Zhao's past offering of CSE565 (https://zzm7000.github.io/teaching/2023springcse410565/index.html)
  - Slides from Prof Hongxin Hu's past offering of CSE565

Where we are now

- Basic security objectives / principles
- Cryptography basics
- Authentication & Authorization
- Web Security basics

What's next

- Network Security
- Software Security
- AI Security (probably?)

What's next

- **Network Security**
  - Computer Network Basics; OSI 7-Layer (5-Layer) Model.
  - Protocols: Ethernet, ARP, TCP/IP, DNS. Attacks & Defenses.
  - Firewalls, Tunnels, Network Intrusion Detection
  - ...
- Software Security
- AI Security (probably?)

The Internet

The Internet

*"The vast **collection of computer networks** which form and **act as a single huge network** for transport of data and messages across distances which can be anywhere from the same office to anywhere in the world."* — William F. Slater, III, 1996

The Internet: History

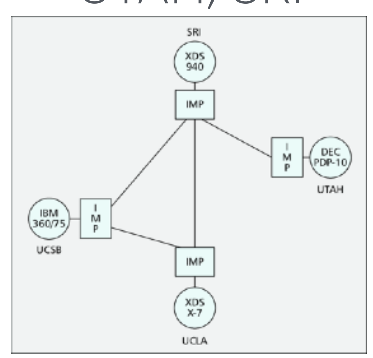
Communication in 1964

"A network to survive nuclear attack."

Paul Baran

DARPA starts "ARPANET"

First 4 nodes: UCLA, UCSB, UTAH, SRI



TCP spec. by Vint Cerf

TCP / IP

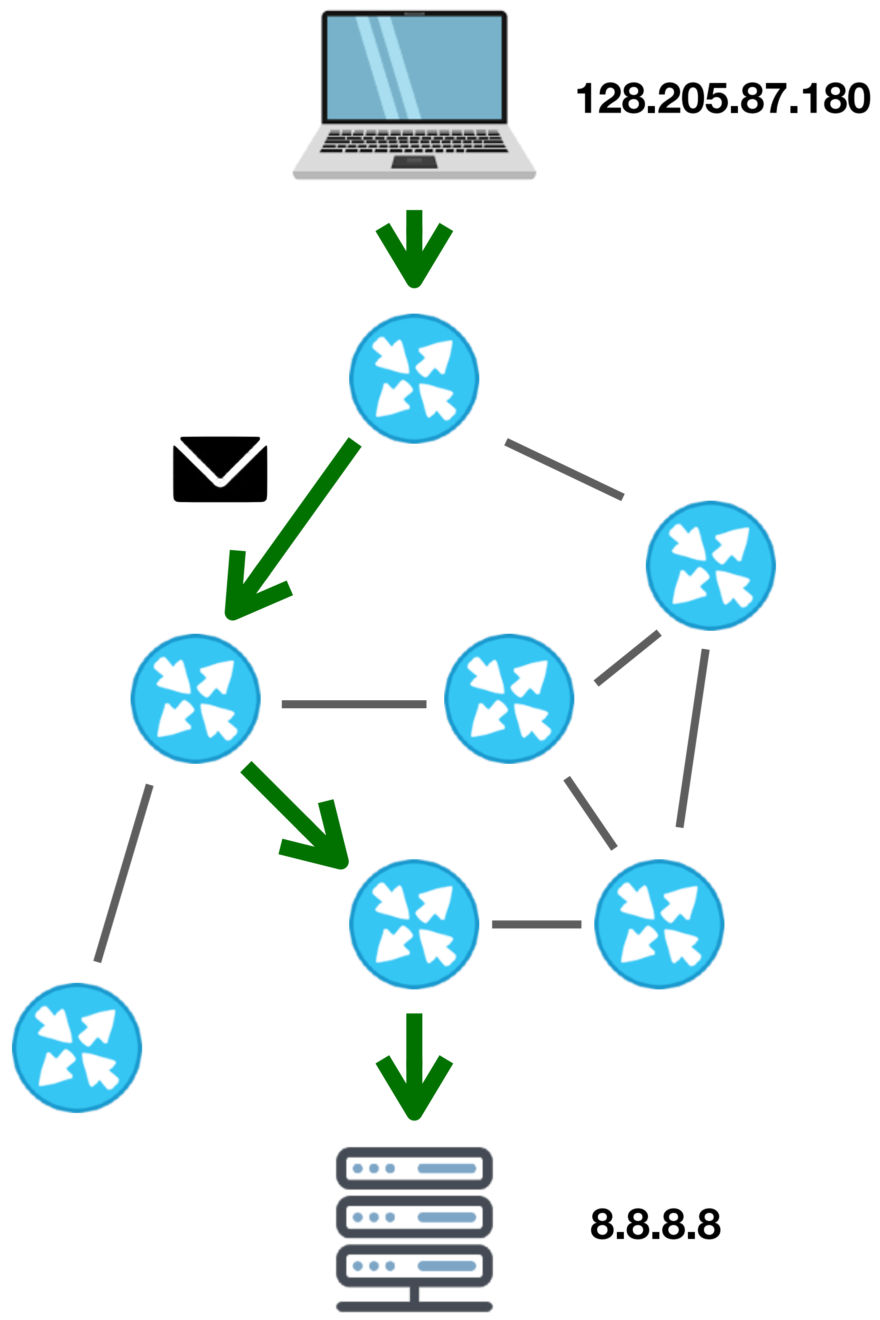
TCP split to TCP/IP

40 billion devices connected

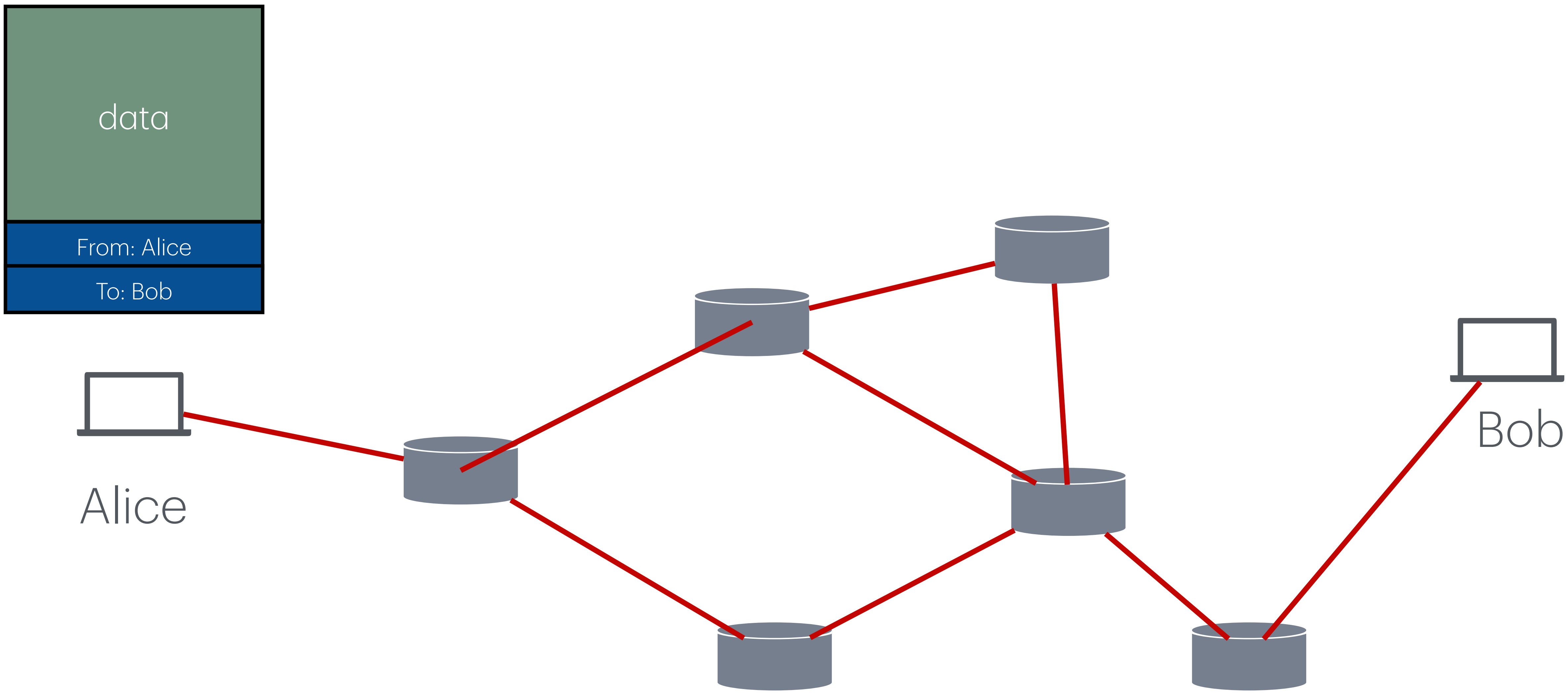
1964 | 1968 | 1970 | 1974 | 1978 | … | 2020

The Internet: Packets Delivery

- Global network that provides **best-effort** delivery of **packets** between connected hosts
- **Packet**: a structured sequence of bytes
  - Header: metadata used by network
  - Payload: user data to be transported
- Every host has a unique identifier — **IP address**
- Series of routers receive packets, look at destination address on the header and send it one hop towards the destination IP address

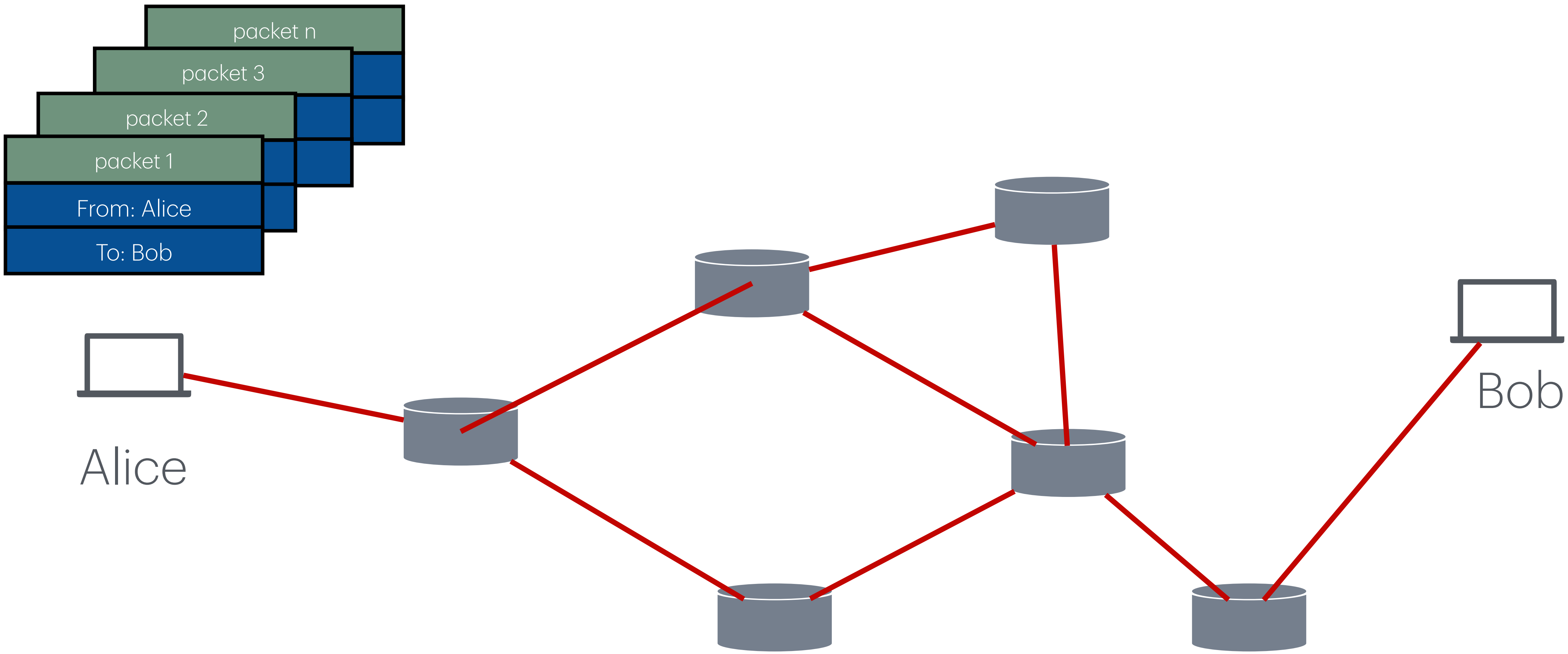


The Internet: Packets Delivery



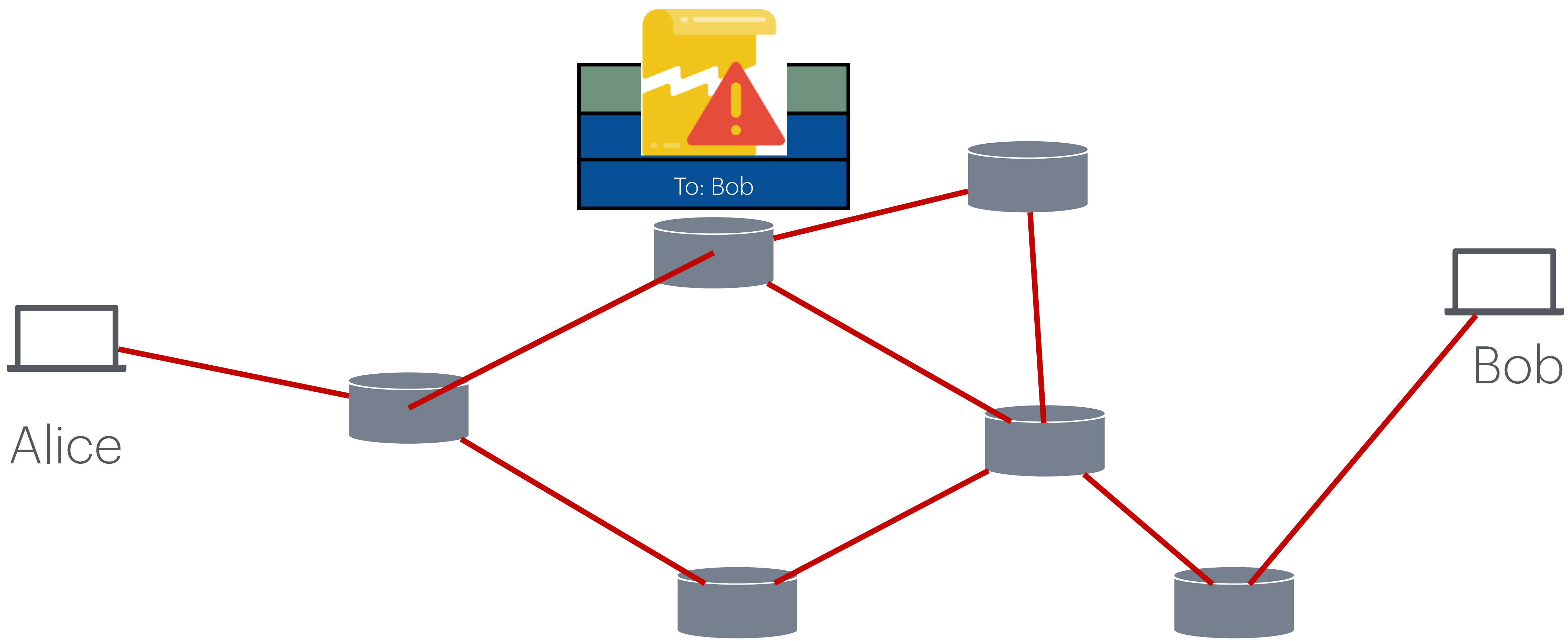
The Internet: Packets Delivery

Multiple packets in pipeline at same time



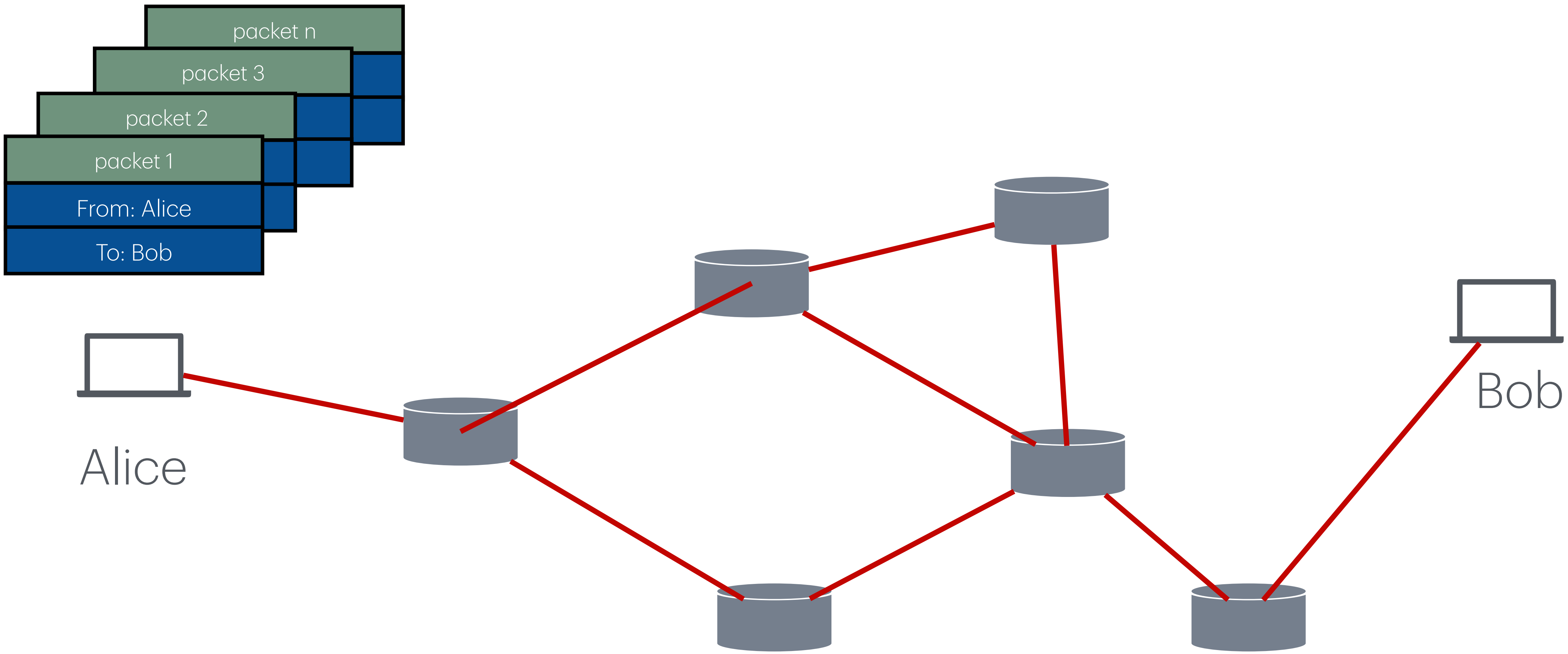
The Internet: Packets Delivery

Packets may be damaged



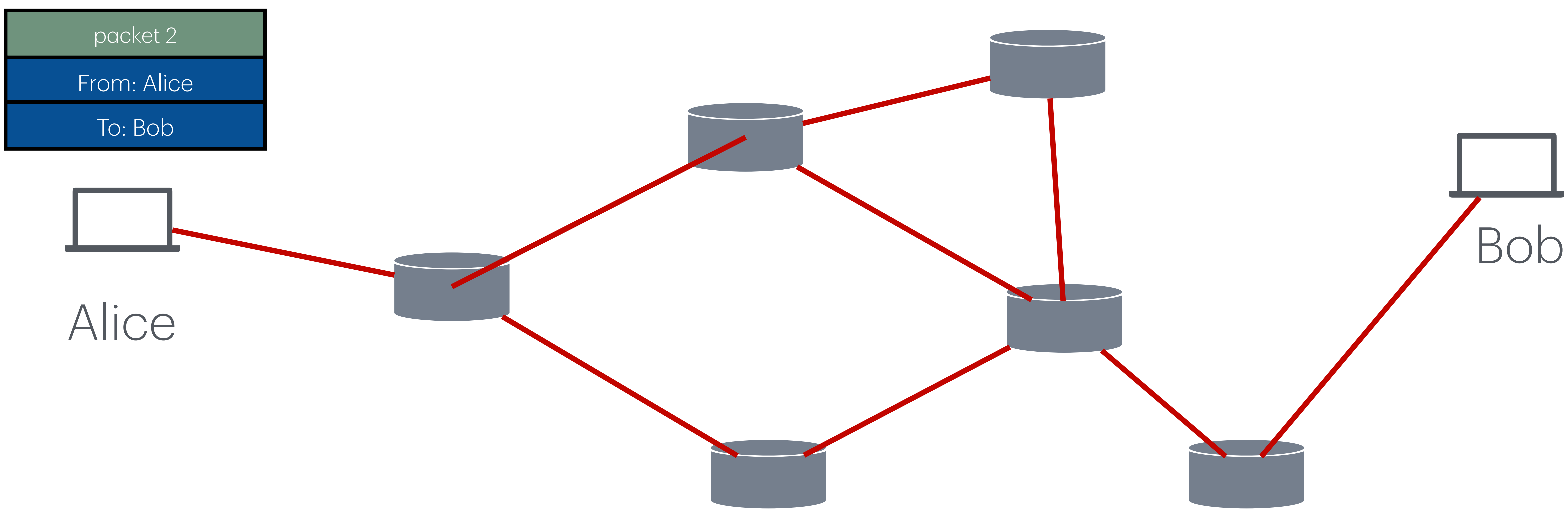
The Internet: Packets Delivery

Packets may arrive out of order



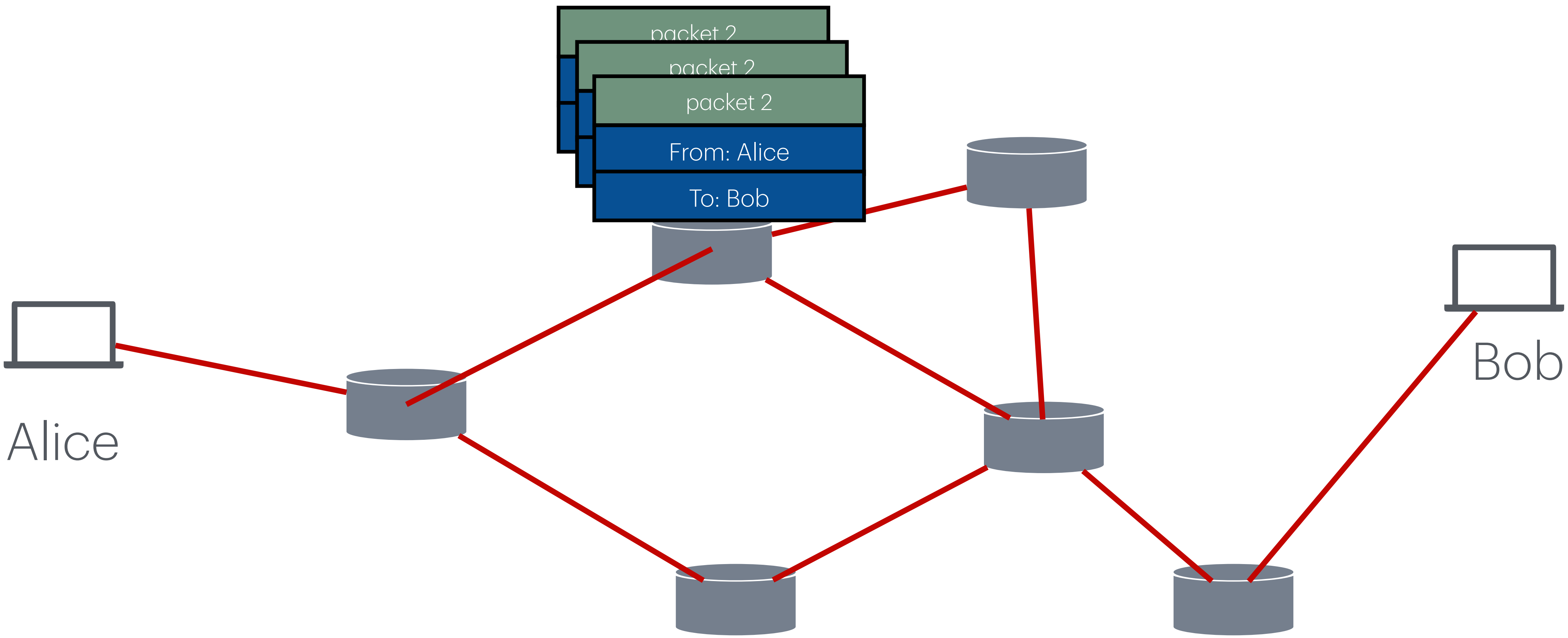
The Internet: Packets Delivery

Packets may by duplicated



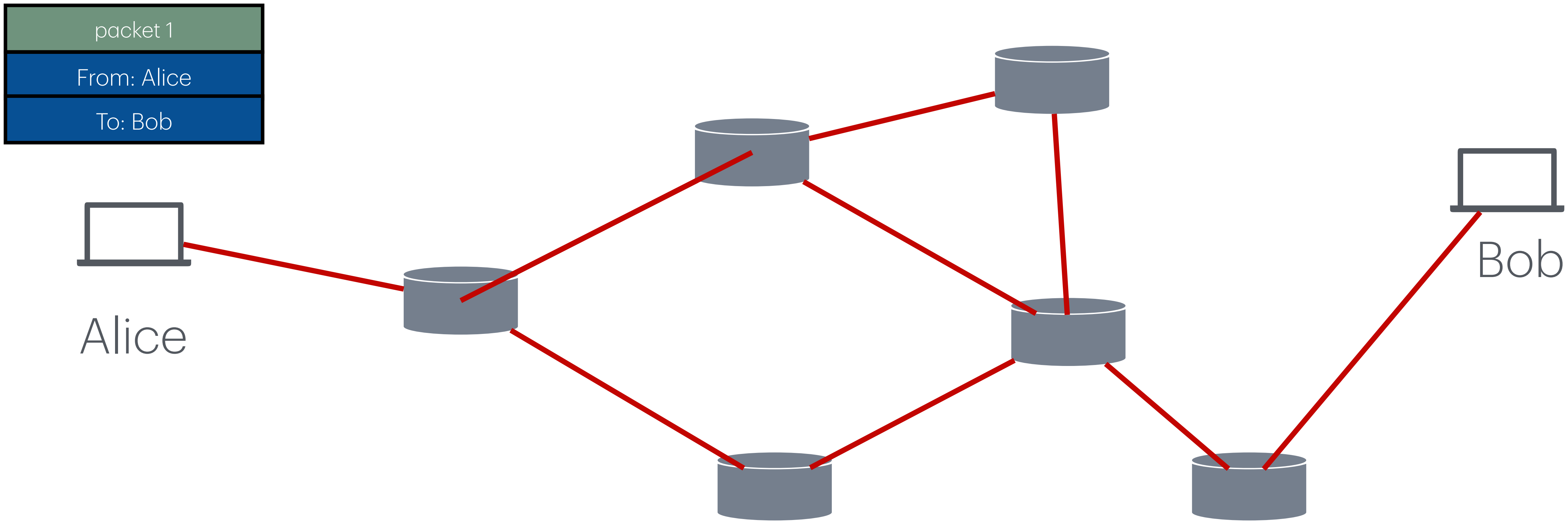
The Internet: Packets Delivery

Packets may by duplicated



The Internet: Packets Delivery

Some packets may not arrive at all



The Internet: Packets Delivery

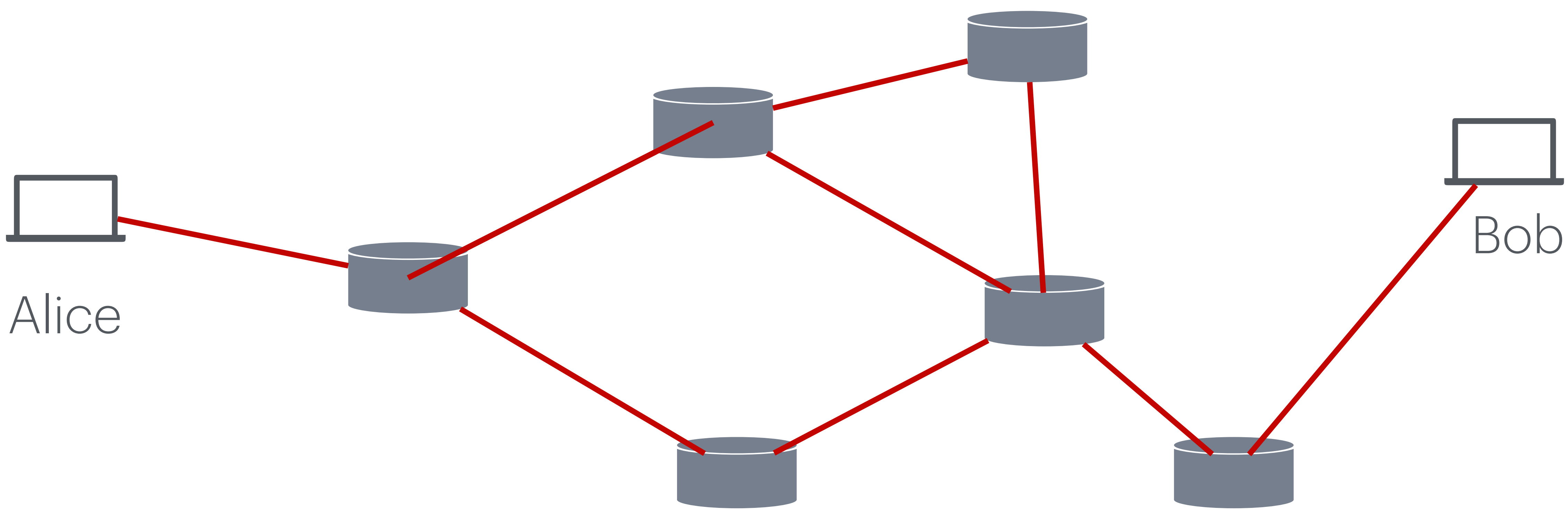
The Internet doesn't promise to deliver packets *in order*.

It doesn't promise to deliver packets quickly, or *on time*.

It doesn't even promise to deliver them at all!

It just makes a "best-effort" attempt.

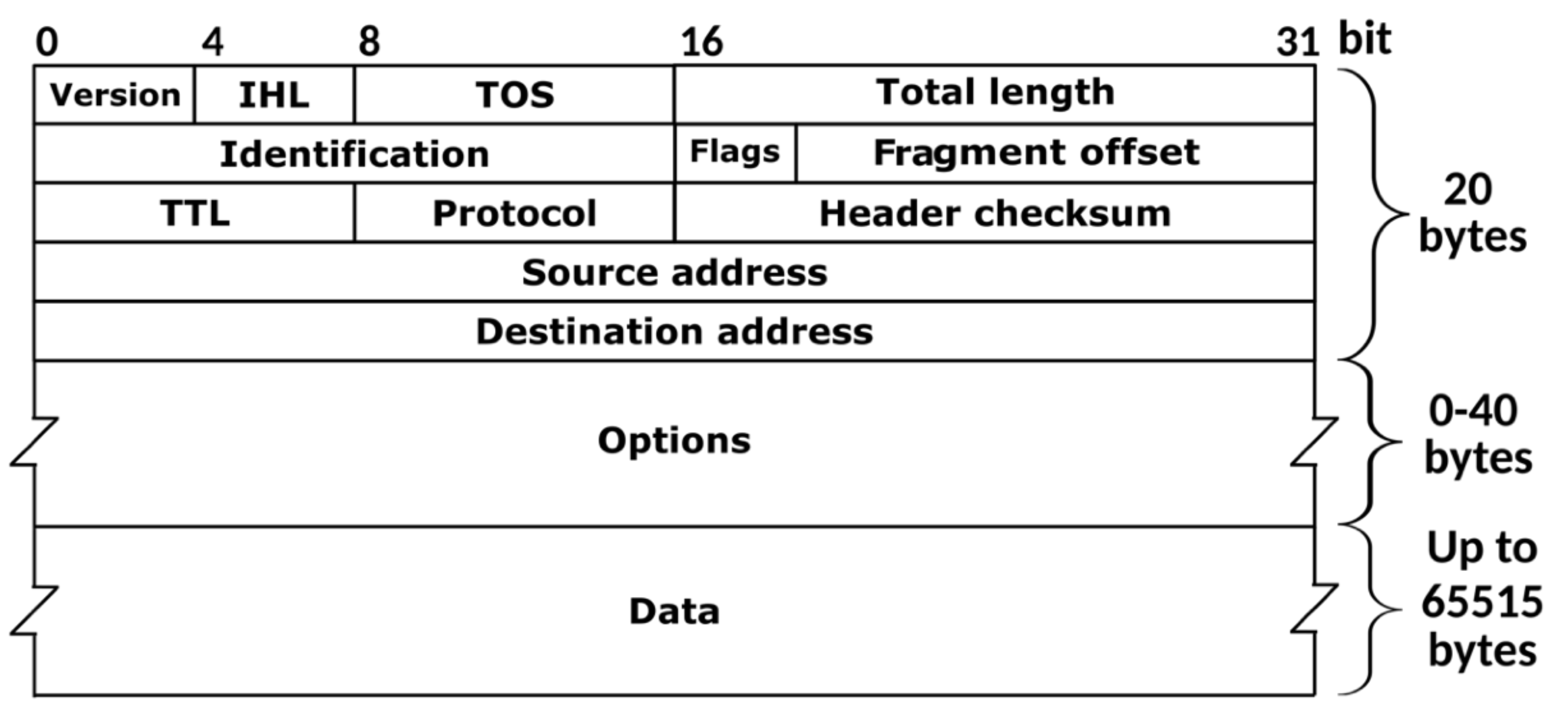
Computer Network:Sending data **reliably** over an Internet that is **unreliable**.



Network Protocols

Network Protocols

- Define how hosts communicate in published network protocols
  - **Syntax**: How communication is structured (e.g., format and order of messages)
  - **Semantics**: What communication means. Actions taken on transmit or receipt of message, or when a timer expires. What assumptions can be made.

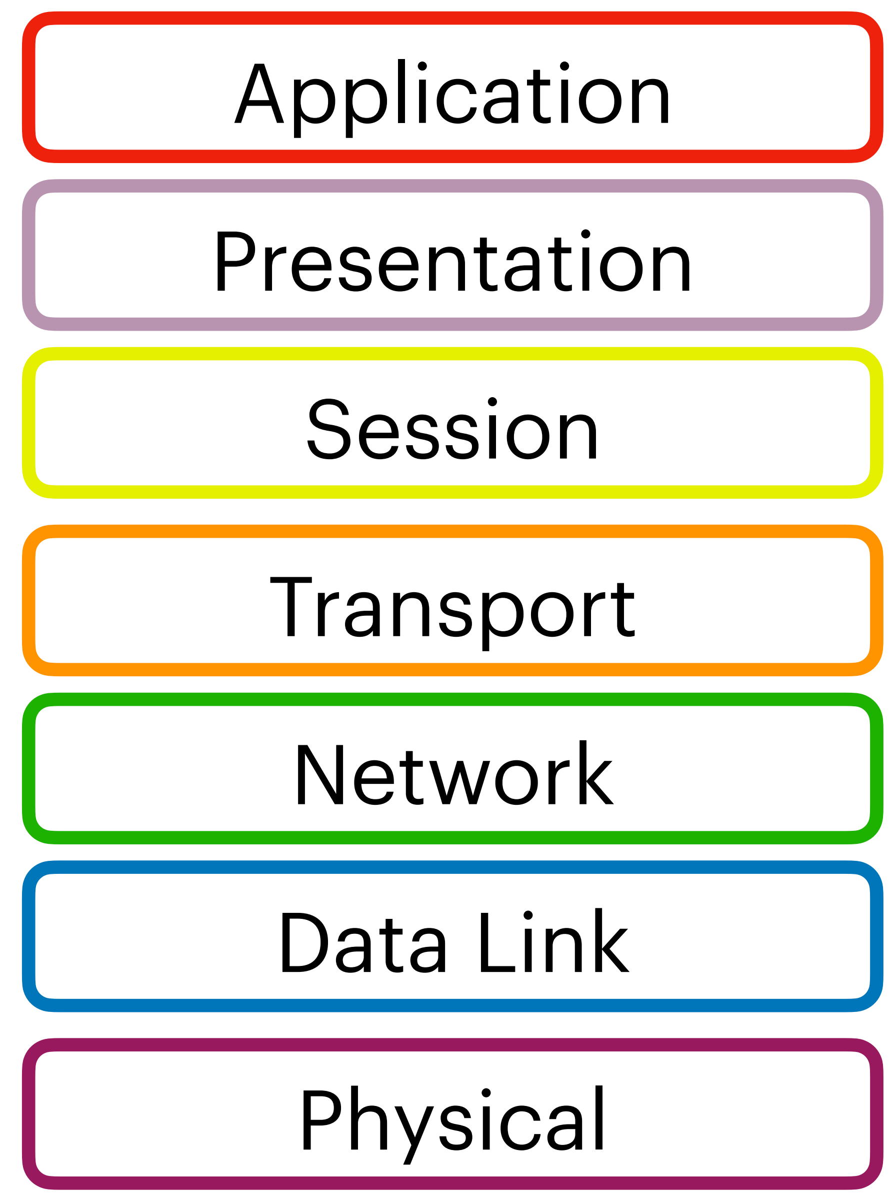


Example: Spec. of an IPv4 Packet

Protocol Layering

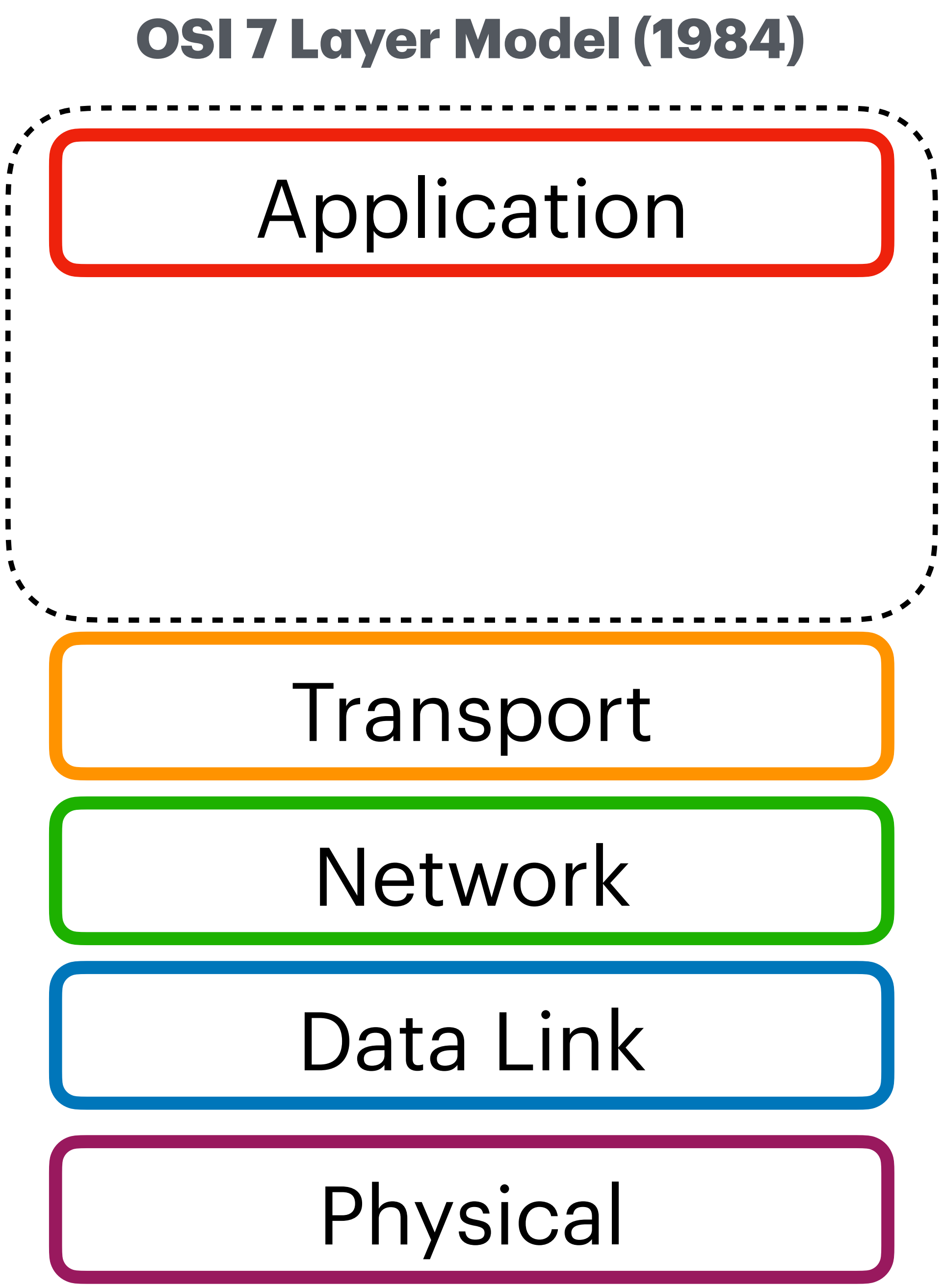
- Networks use a stack of protocol layers
  - Each layer has different responsibilities.
  - Layers define *abstraction boundaries*
- Lower layers provide services to layers above
  - Don't care what higher layers do
- Higher layers use services of layers below
  - Don't worry about how it works

OSI 7 Layer Model (1984)



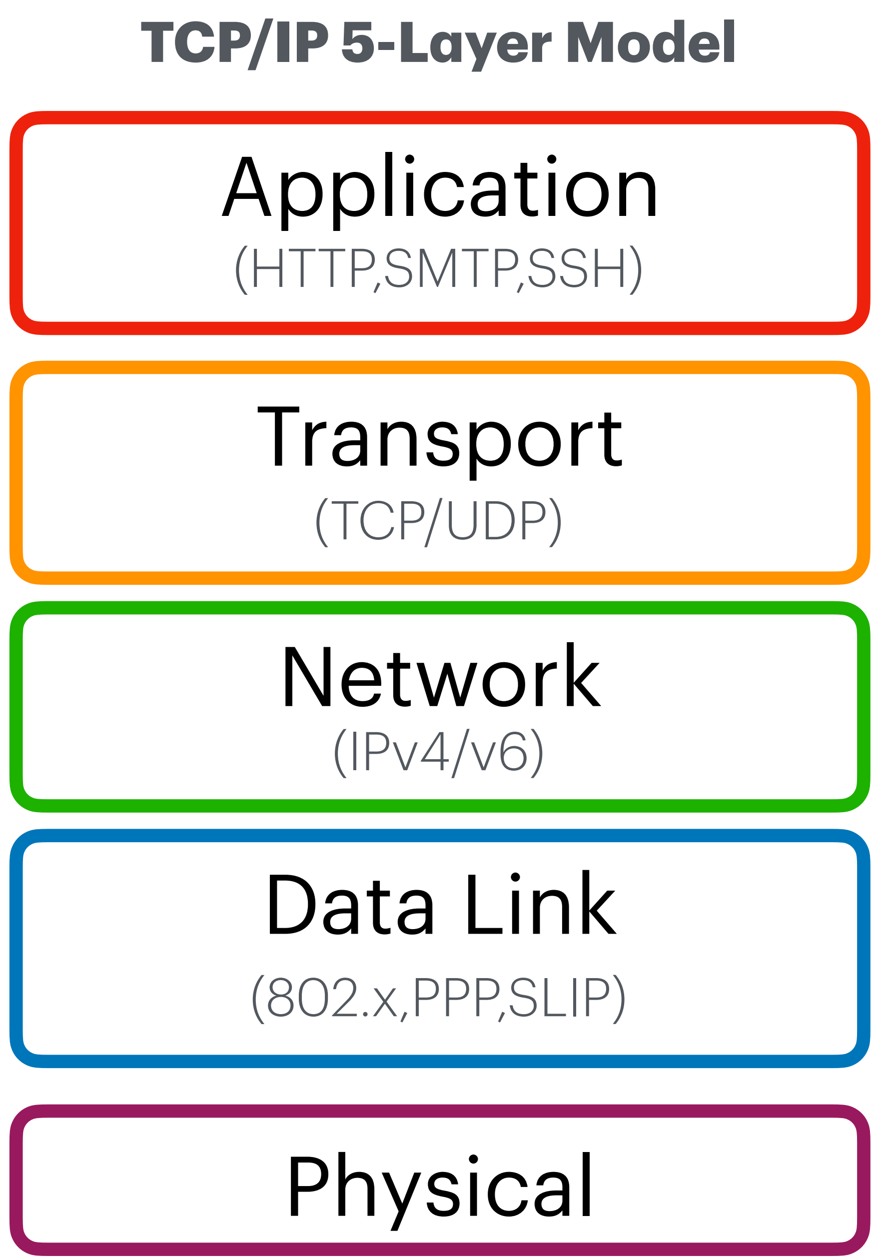
Protocol Layering

- Networks use a stack of protocol layers
  - Each layer has different responsibilities.
  - Layers define *abstraction boundaries*
- Lower layers provide services to layers above
  - Don’t care what higher layers do
- Higher layers use services of layers below
  - Don’t worry about how it works



Protocol Layering

- Networks use a stack of protocol layers
  - Each layer has different responsibilities.
  - Layers define *abstraction boundaries*
- Lower layers provide services to layers above
  - Don't care what higher layers do
- Higher layers use services of layers below
  - Don't worry about how it works



TCP/IP 5-Layer Model

How do bits get translated into electrical, optical, or radio signals

Physical

TCP/IP 5-Layer Model

How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link.

Data Link
(802.x,PPP,SLIP)

How do bits get translated into electrical, optical, or radio signals

Physical

TCP/IP 5-Layer Model

Packet forwarding. How to get a packet to the final destination when there are many hops along the way.

Network
(IPv4/v6)

How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link.

Data Link
(802.x,PPP,SLIP)

How do bits get translated into electrical, optical, or radio signals

Physical

TCP/IP 5-Layer Model

Allows a client to establish a connection to specific services (e.g., web server on port 80). Provides reliable communication.

Transport
(TCP/UDP)

Packet forwarding. How to get a packet to the final destination when there are many hops along the way.

Network
(IPv4/v6)

How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link.

Data Link
(802.x,PPP,SLIP)

How do bits get translated into electrical, optical, or radio signals

Physical

TCP/IP 5-Layer Model

Defines how individual applications communicate. For example, HTTP defines how browsers send requests to web servers.	**Application** (HTTP,SMTP,SSH)
Allows a client to establish a connection to specific services (e.g., web server on port 80). Provides reliable communication.	**Transport** (TCP/UDP)
Packet forwarding. How to get a packet to the final destination when there are many hops along the way.	**Network** (IPv4/v6)
How to get packet to the next hop. Transmission of data frames between two nodes connected by a physical link.	**Data Link** (802.x,PPP,SLIP)
How do bits get translated into electrical, optical, or radio signals	**Physical**

IP: The Narrow Waist

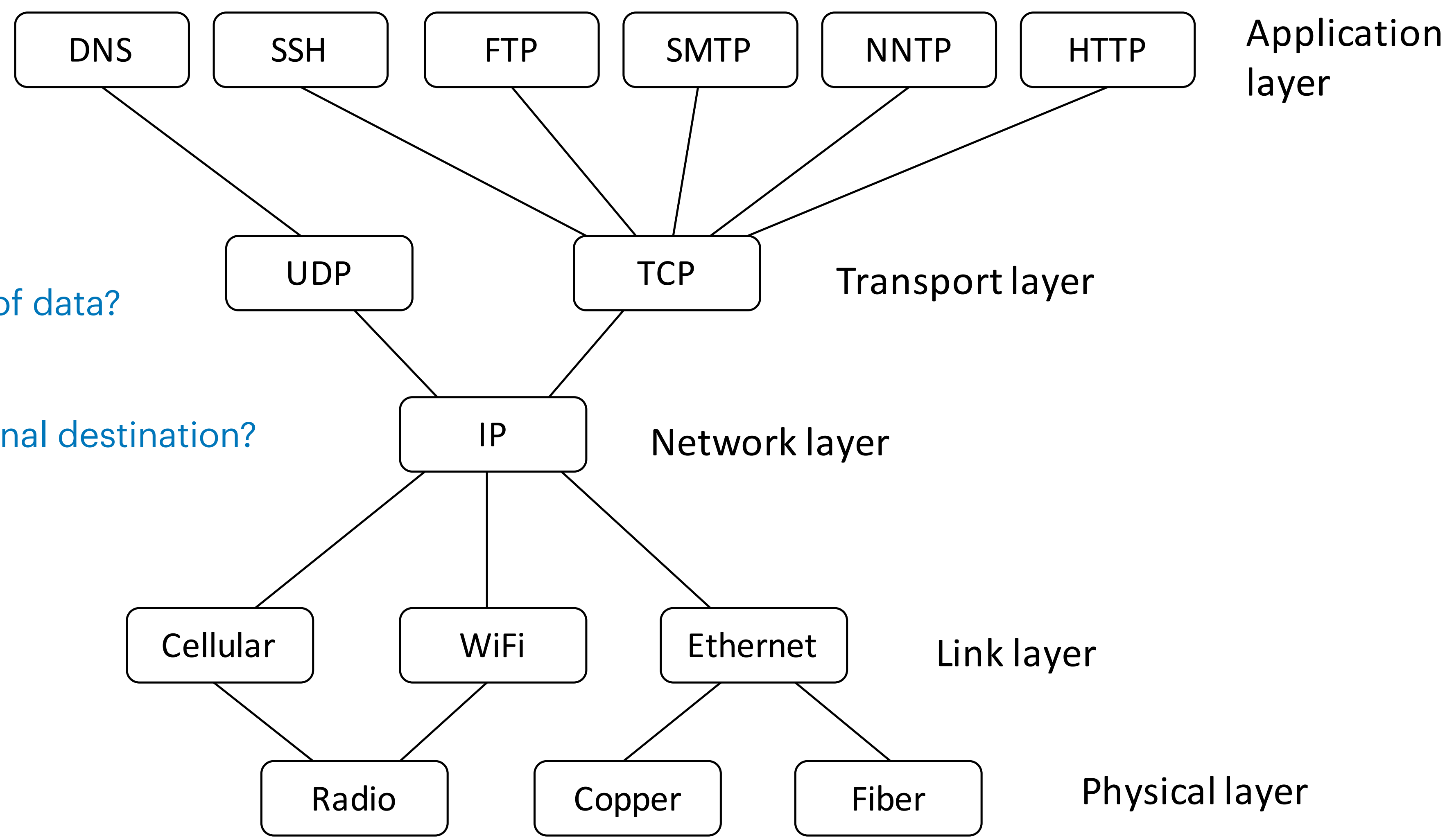
How does Application structure data?

How do I get to the right service?

How do I have a reliable “stream” of data?

How a does packet final destination?

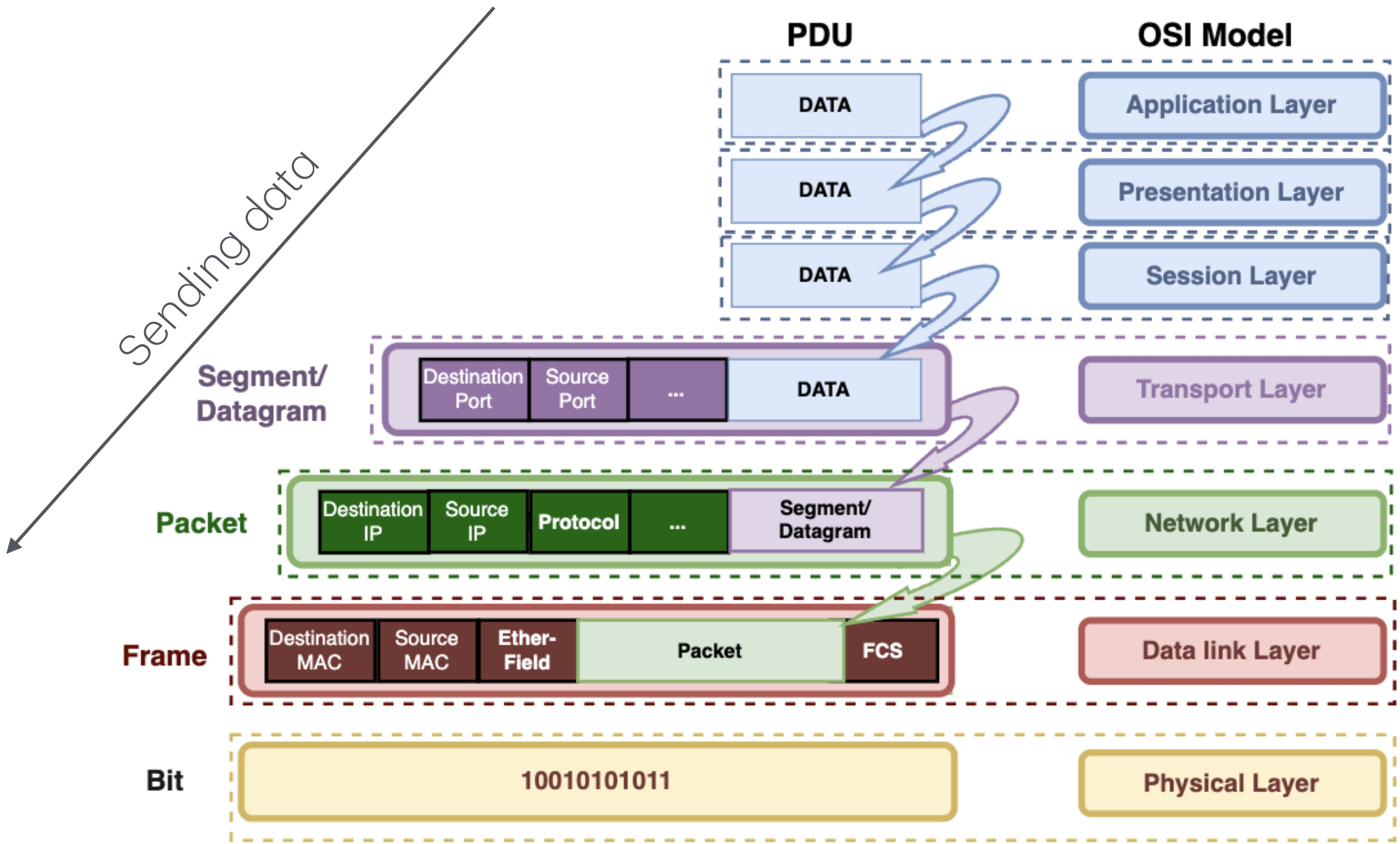
How do I get to next hop?



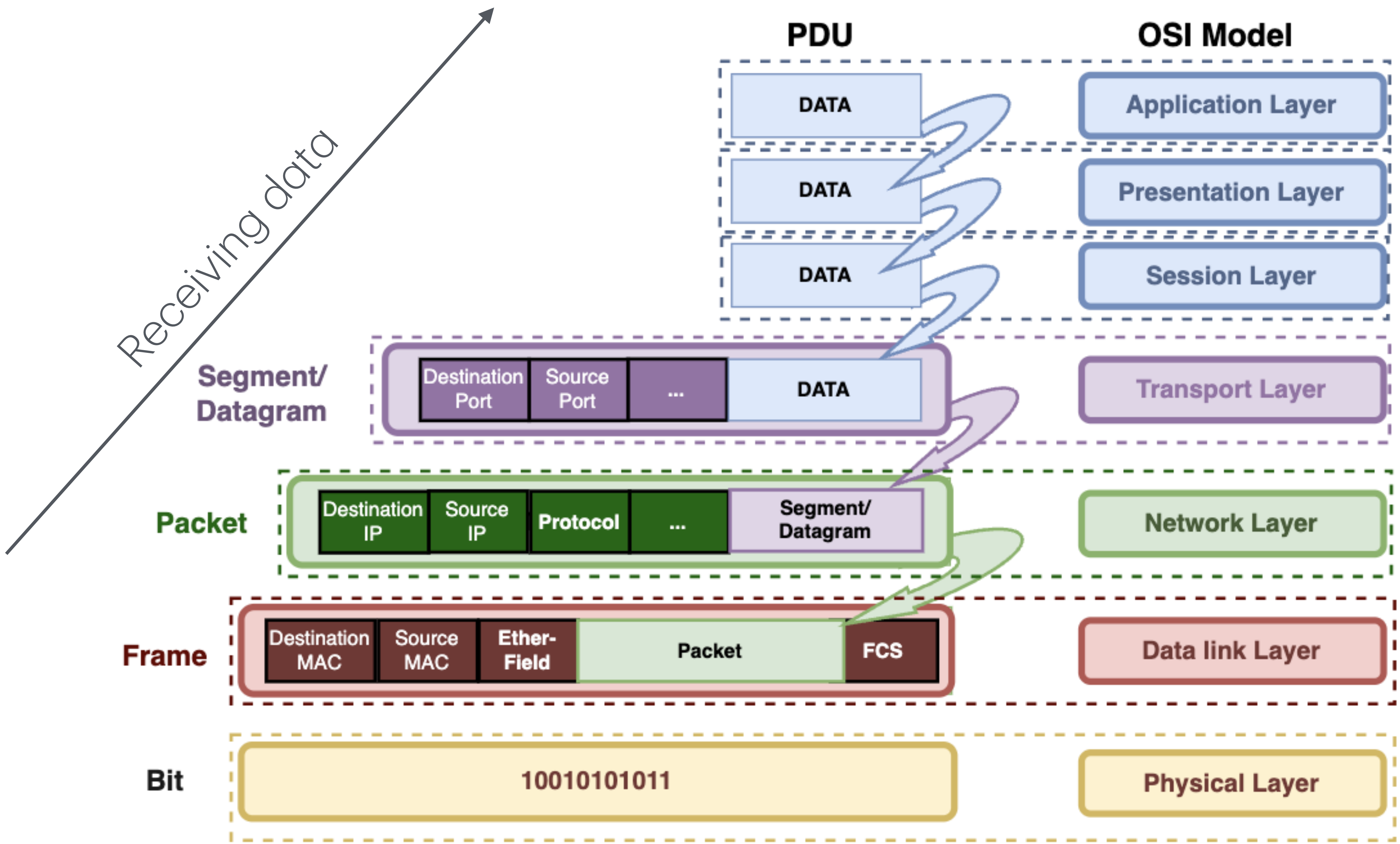
Protocol Data Unit (PDU)

- A **unit of data** specified **in the protocol of a given layer**, which consists of protocol control information and user data.
  - Application layer: PDU is referred to as **data**
  - Transport layer: PDU is a **segment** (TCP segment)
  - Network layer: PDU is a **packet** or diagram
  - Data link layer: PDU is a **frame**
  - Physical layer: PDU is just **bit**

Protocol Data Unit (PDU)

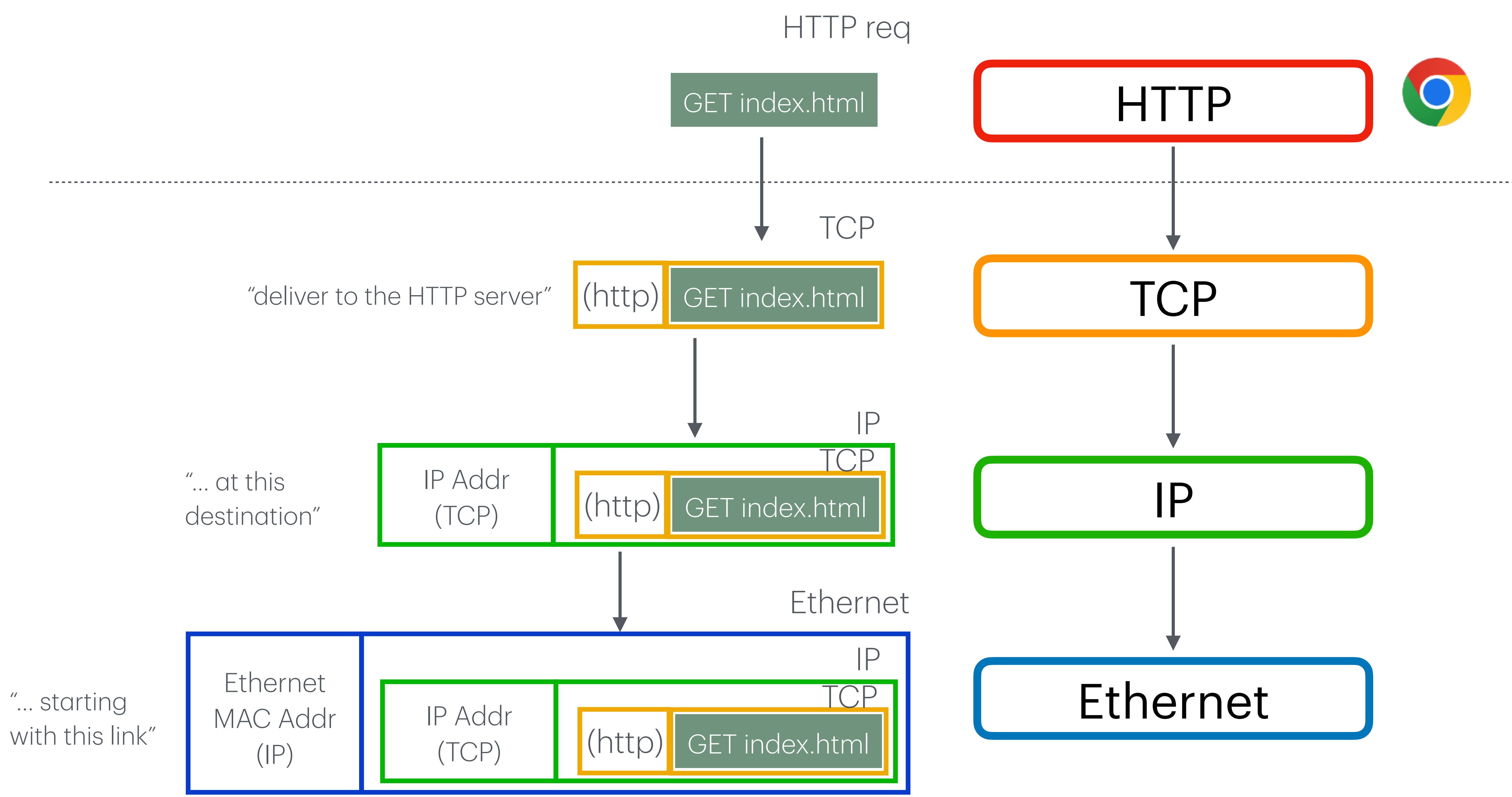


Protocol Data Unit (PDU)



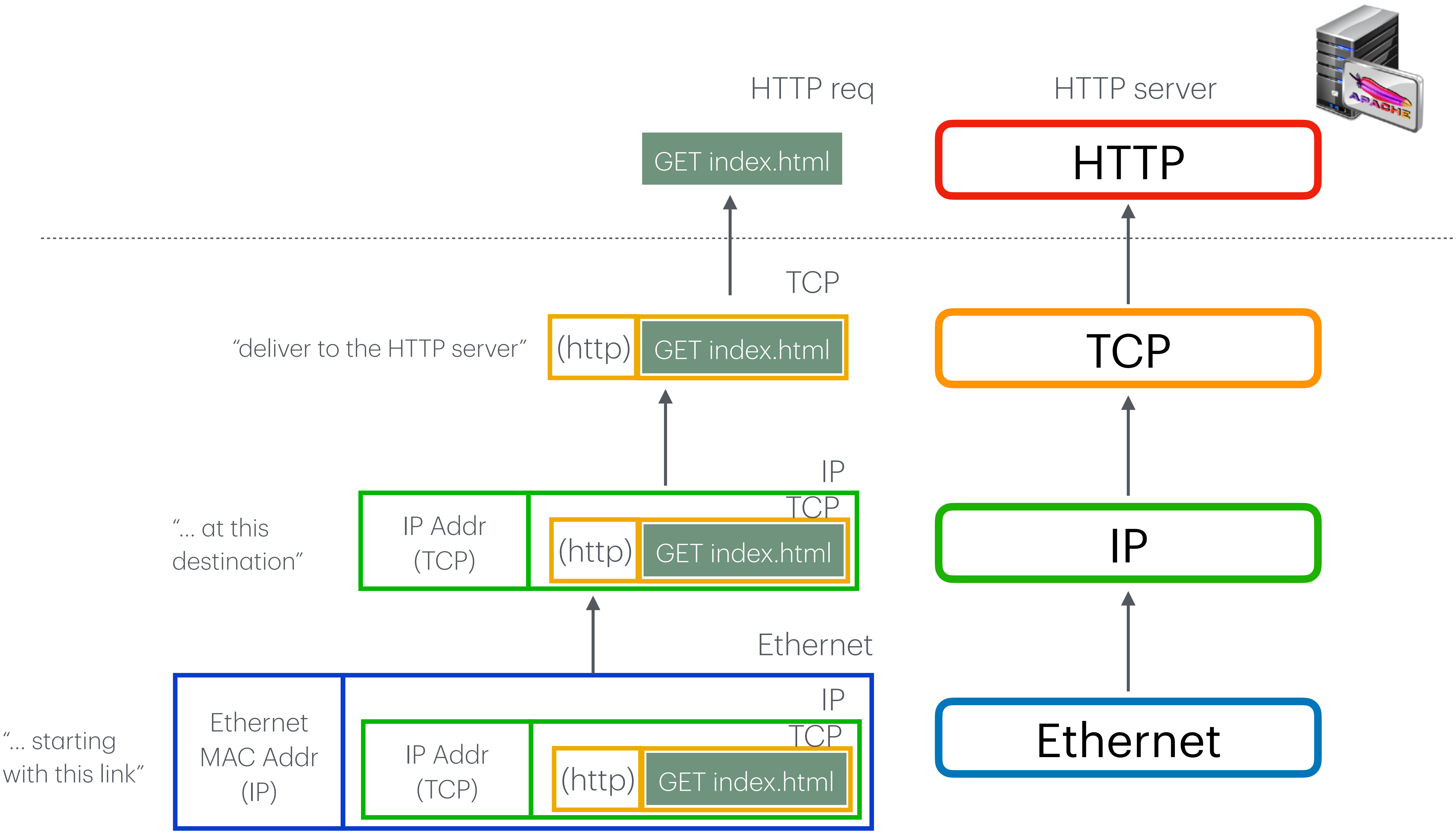
PDU Encapsulation Example

HTTP Client (e.g. Chrome)



PDU Encapsulation Example

HTTP Server (e.g. Apache)

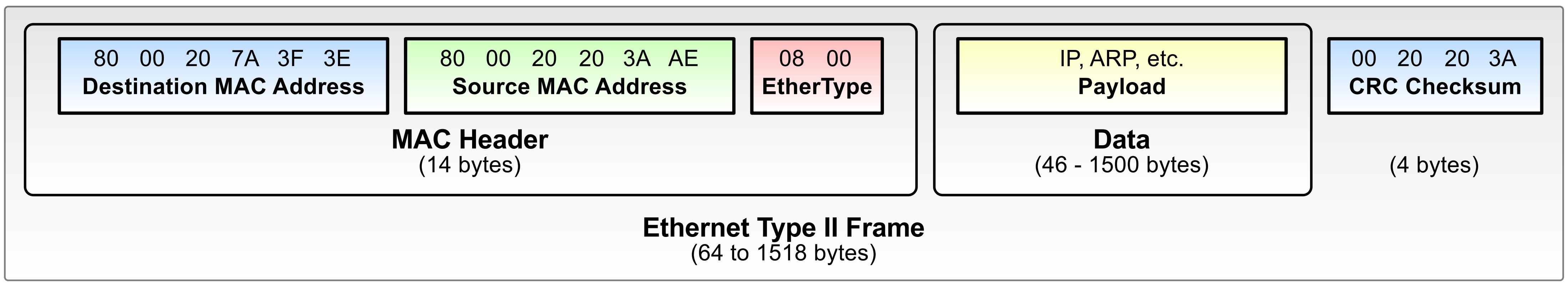


Link Layer

- **Assumes**: Local nodes are physically connected
- **Task**: Transfer bytes between two hosts on the physically connected network

Ethernet: IEEE 802.3

Most common Link Layer Protocol. Let's you send packets to other local hosts.



At layer 2 (link layer) packets are called *frames*

MAC addresses: 6 bytes, universally unique

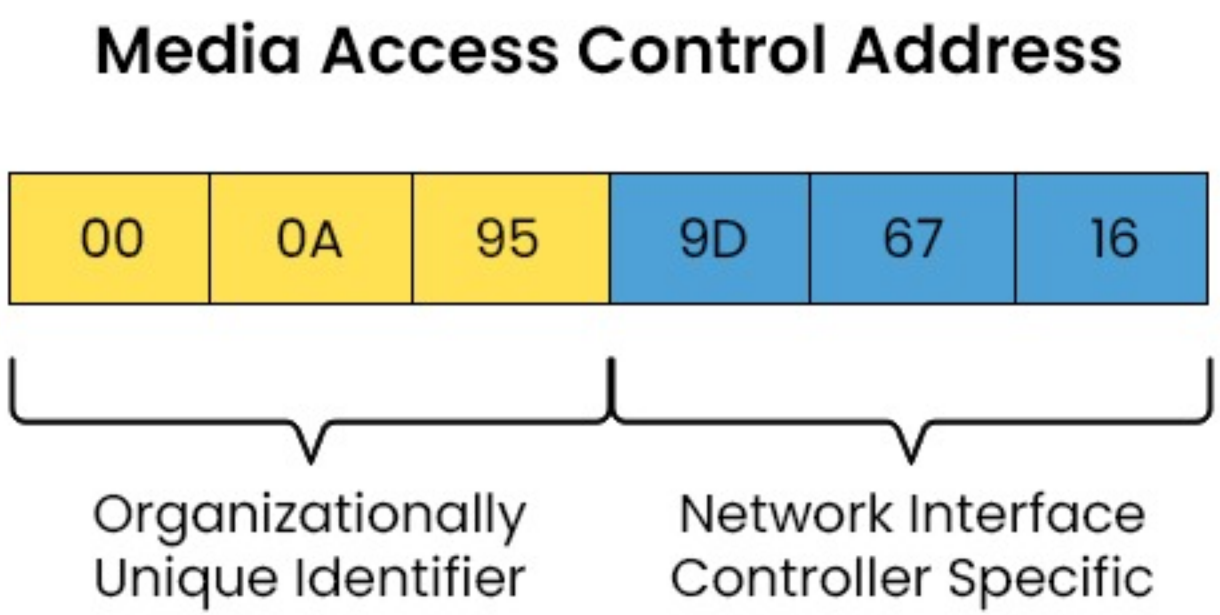
EtherType gives layer 3 protocol in payload

- 0x0800: IPv4
- 0x0806: ARP
- 0x86DD: IPv6

MAC Address

- Media Access Control **address**
- The MAC address is a unique value associated with a **network adapter** (network interface controller, NIC).
- Also known as hardware addresses or physical addresses.
- MAC addresses are 12-digit hexadecimal numbers (48 bits in length). Usually written in: `MM : MM : MM : SS : SS : SS`

MAC Address



- *Theoretically*, every single NIC in the world should have a totally unique MAC address.
  - 1st half: Organizationally Unique Identifier (OUI), the ID number of the adapter manufacturer
  - 2nd half: Device ID that represents the serial number assigned to the adapter by the manufacturer.

MAC Address

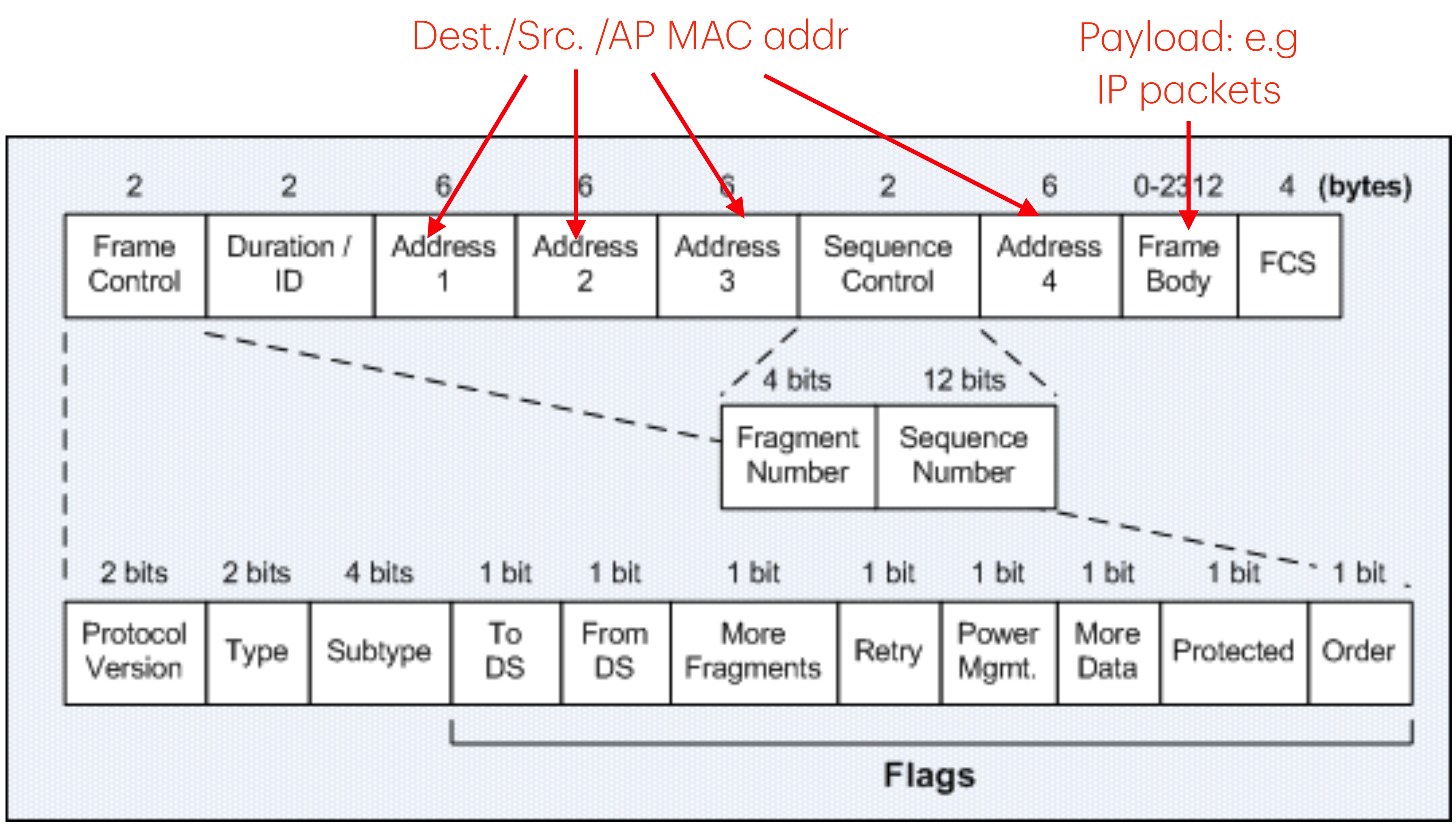
```
seed@seed-vm:~$ ifconfig
br-6d868408adde: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        inet6 fe80::42:abff:fec8:596d  prefixlen 64  scopeid 0x20<link>
        ether 02:42:ab:c8:59:6d  txqueuelen 0  (Ethernet)
        RX packets 100  bytes 41825 (41.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 768  bytes 83205 (83.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:31:ce:d5:50  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.83.3.133  netmask 255.255.224.0  broadcast 10.83.31.255
        inet6 fe80::edbd:c321:f9f8:890f  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:8f:d2:4f  txqueuelen 1000  (Ethernet)
        RX packets 748811  bytes 742579768 (742.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 171950  bytes 20268758 (20.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 44  memory 0x3fe00000-3fe20000
```

Wi-Fi: IEEE 802.11

Another common Link Layer Protocol.
Let's you send packets to a wireless LAN

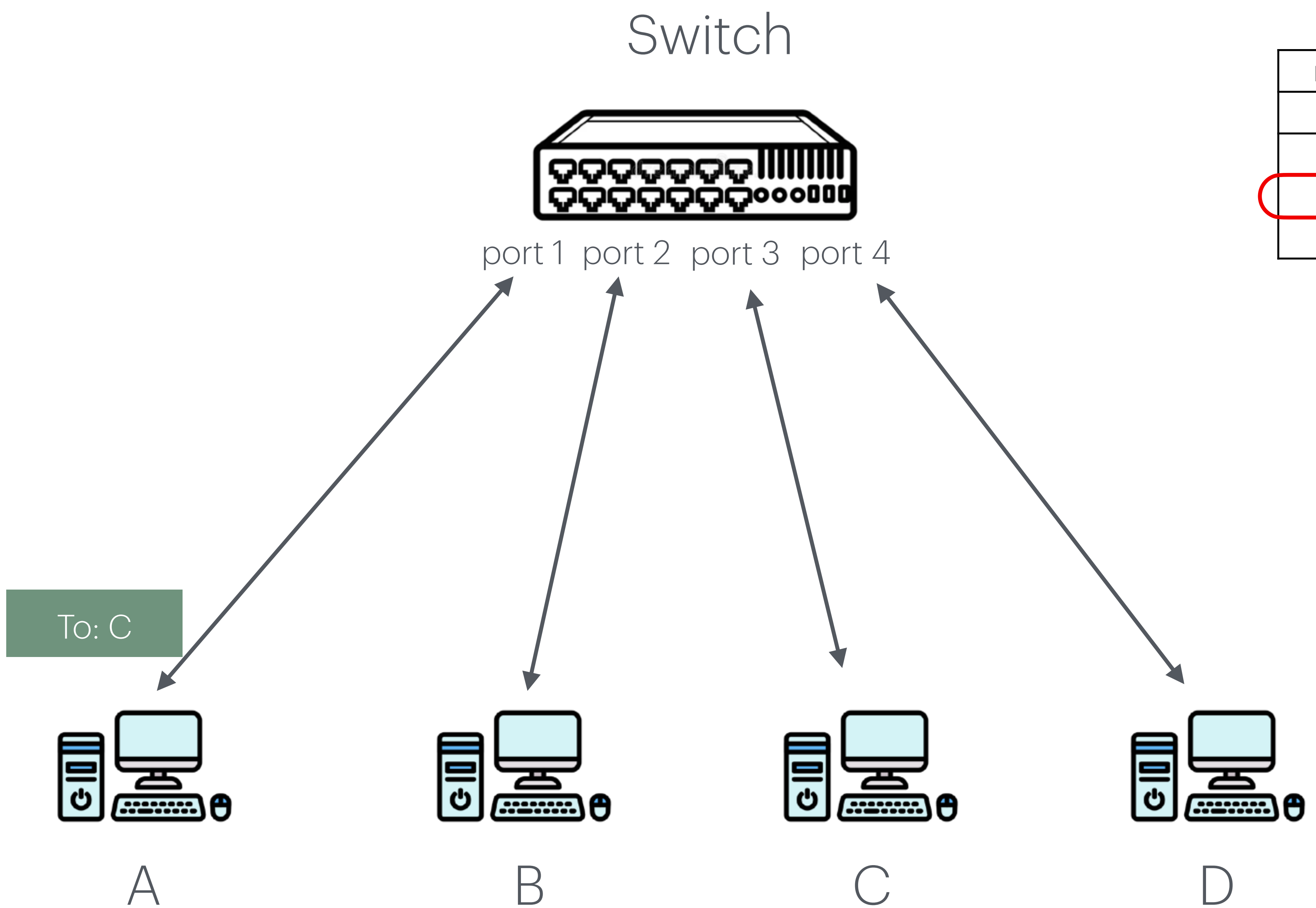


Switched Ethernet (LAN)

- (Network) **Switches** forward frames selectively
- With *switched* Ethernet, the switch *learns* at which physical port each MAC address lives based on MAC source addresses
  - If switch knows MAC address `M` is at port `P`, it will only send a packet for `M` out port `P`
  - If switch does not know which port MAC address `M` lives at, will broadcast to all ports

not this!

Switched Ethernet (LAN)



(A knows C's MAC)

Port	MAC
1	00:00:00:aa:aa:aa
2	00:00:00:bb:bb:bb
3	00:00:00:cc:cc:cc
4	00:00:00:dd:dd:dd

Internet Protocol (IP)

- **Internet Protocol (IP)** defines what packets that cross the Internet need to look like to be processed by routers
- Every host is *assigned* a unique identifier ("IP Address")
  - Who assigns IP? This is a bit tricky, see later.
- Every packet has an IP header that indicates its sender and receiver
- Routers forward packet along to try to get it to the destination host
- Rest of the packet should be ignored by the router

IP Addresses

- **IPv4**: 32-bit host addresses
  - Written as 4 bytes in form `A.B.C.D` where `A,...,D` are 8 bit integers in decimal (called dotted quad)
  - e.g. `192.168.1.1`
  - CIDR notation `A.B.C.D/X`: first `X` bits are subnet (LAN) prefix
- **IPv6**: 128 bit host addresses
  - Written as 16 bytes in form `AA:BB::XX:YY:ZZ` where `AA,...,ZZ` are 16 bit integers in hexadecimal and `::` implies zero bytes
  - e.g. `2620:0:e00:b::53` = `2620:0:e00:b:0:0:0:53`

Private vs Public IP Address

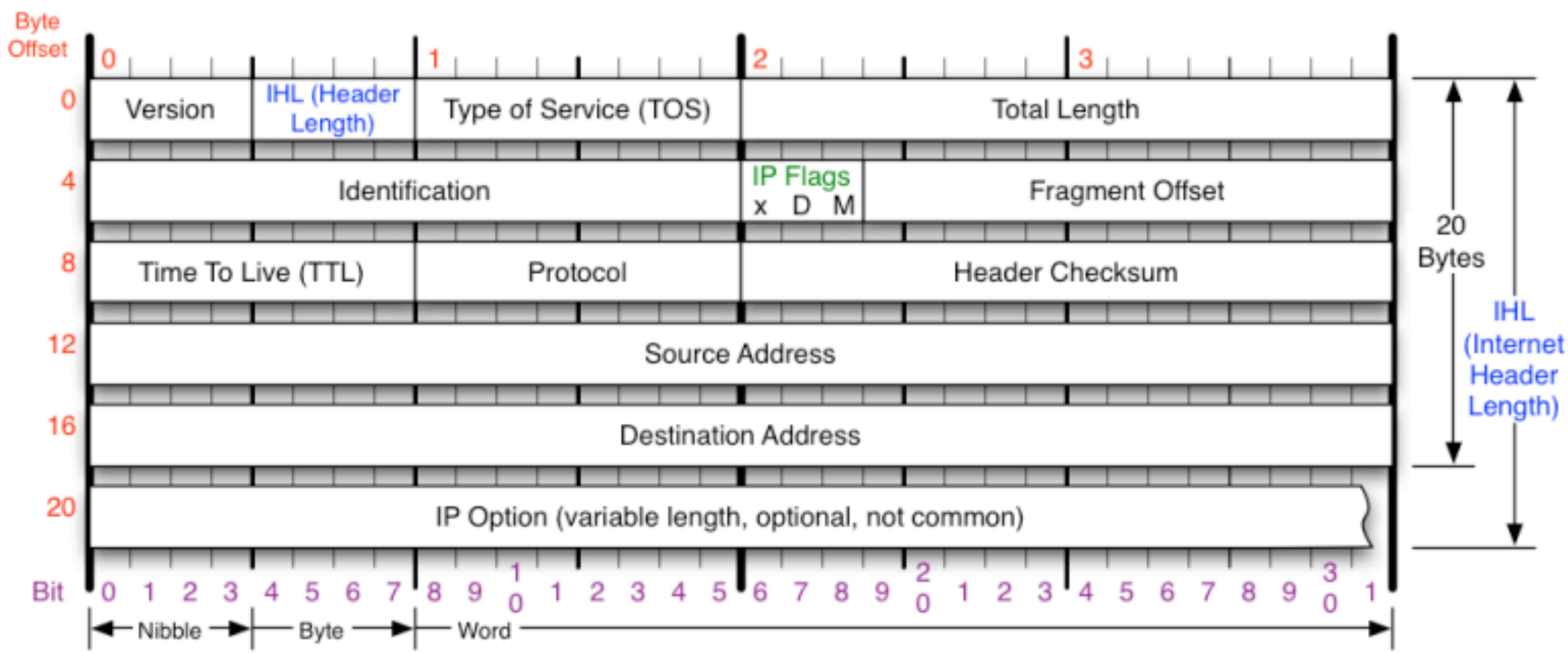
- **Private IP addresses** are reserved for use within **Local Area Networks (LANs)** and are *not* routable on the internet. The following IP ranges are *reserved* for internal networks:
  - Class A: `10.0.0.0` to `10.255.255.255` (CIDR: `10.0.0.0/8`)
  - Class B: `172.16.0.0` to `172.31.255.255` (CIDR: `172.16.0.0/12`)
  - Class C: `192.168.0.0` to `192.168.255.255` (CIDR: `192.168.0.0/16`)
- **Public IP Addresses (External IPs)**:
  - Public IP addresses are used on the internet and are globally unique. Any IP address not in the above private ranges is considered public.

How are IPs assigned?

- Where does a device get its IP?
  - **Inside a LAN**
    - A **router** or a **DHCP server** assigns a *private* IP to each device
    - If no router/DHCP server exists, most OS will *assign itself* an IP from Automatic Private IP Addressing (APIPA)
      - Avoid conflict by broadcasting ARP requests (see later)
  - The external-facing device (usually the router) of a LAN gets its *public* IP from the ISP.

IPv4 Header

- Instruct **routers** and **hosts** what to do with a packet
- All values are filled in by the sending host



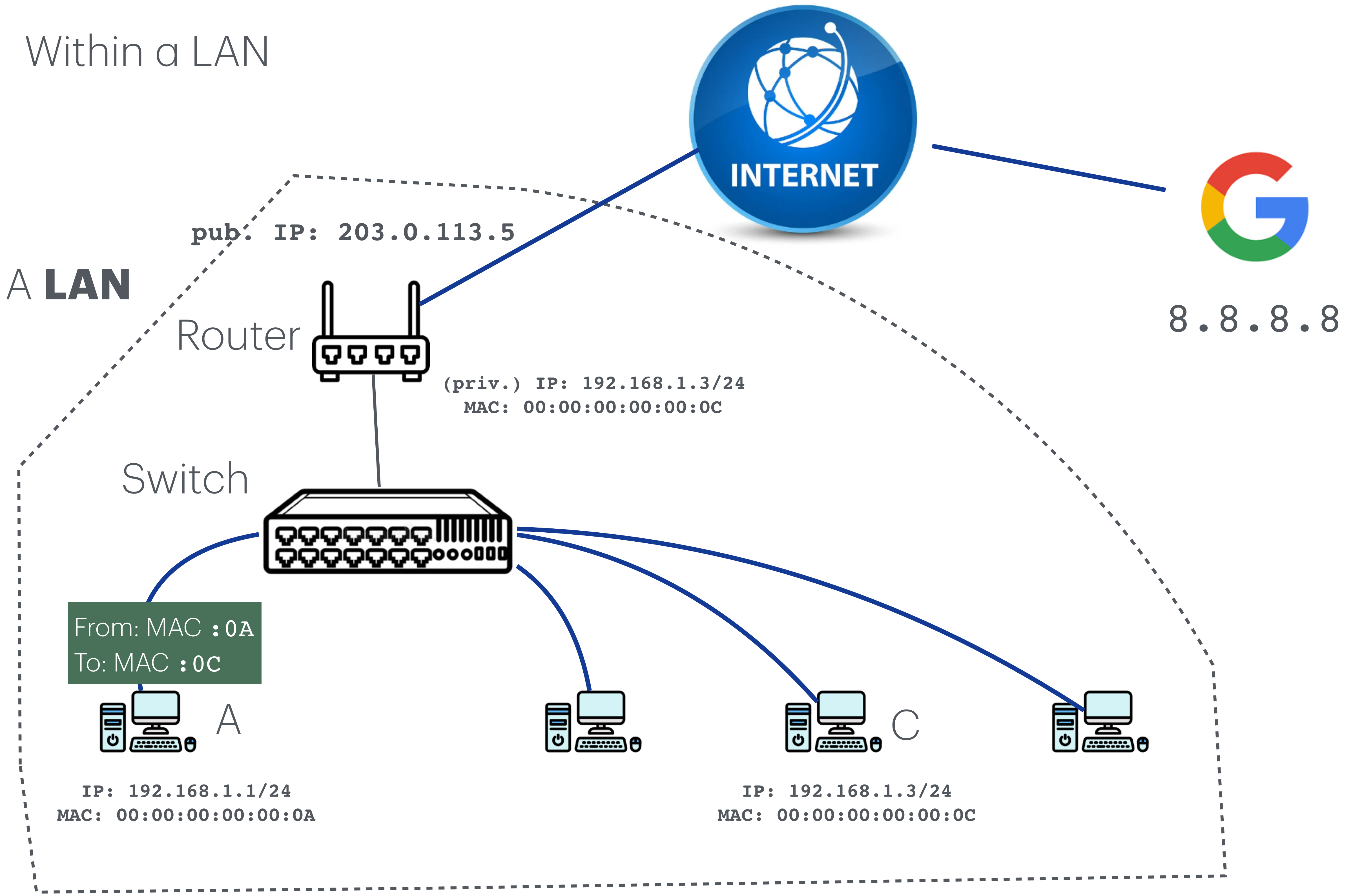
Internet Protocol (IP)

- **Yes:**
  - Routing. If host knows IP of destination host, route packet to it.
  - Fragmentation and reassembly: Split data into packets and reassemble
  - Error Reporting: (maybe, if you're lucky) tell source it dropped your packet
- **No:**
  - Everything else. No ordering. No retransmission. No (real) error checking. No acknowledgement of receipt. No "connections". No security. Just packets.

Communicating at the Link & Network Layer

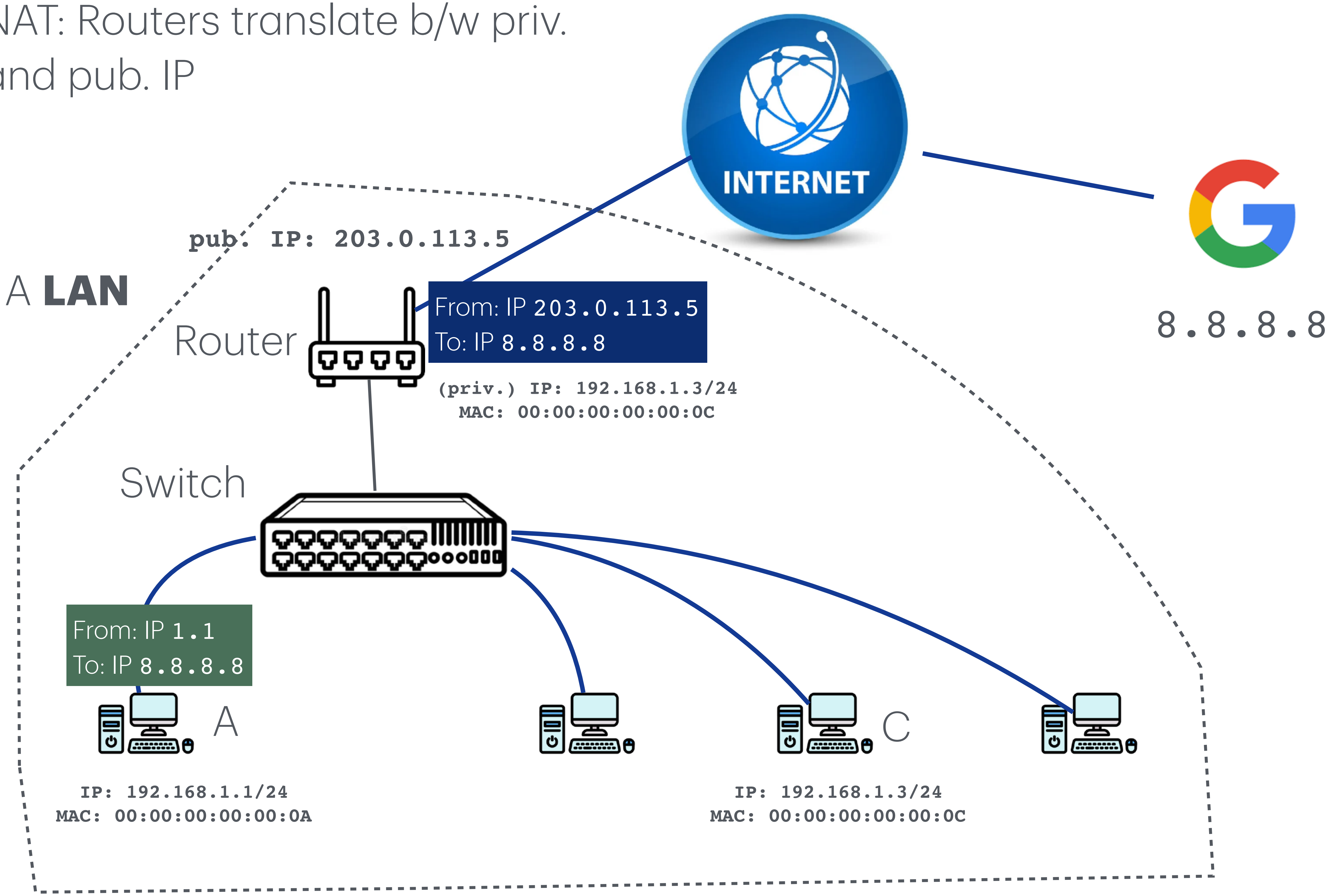
- Summary
  - **Within a same LAN**
    - Only need Link Layer support: MAC address
    - Peer-to-peer or centralized (forwarded by a switch)
  - **Between different LANs**
    - Need Network Layer support: IP address
    - Go through routers

Communicating at the Link & Network Layer



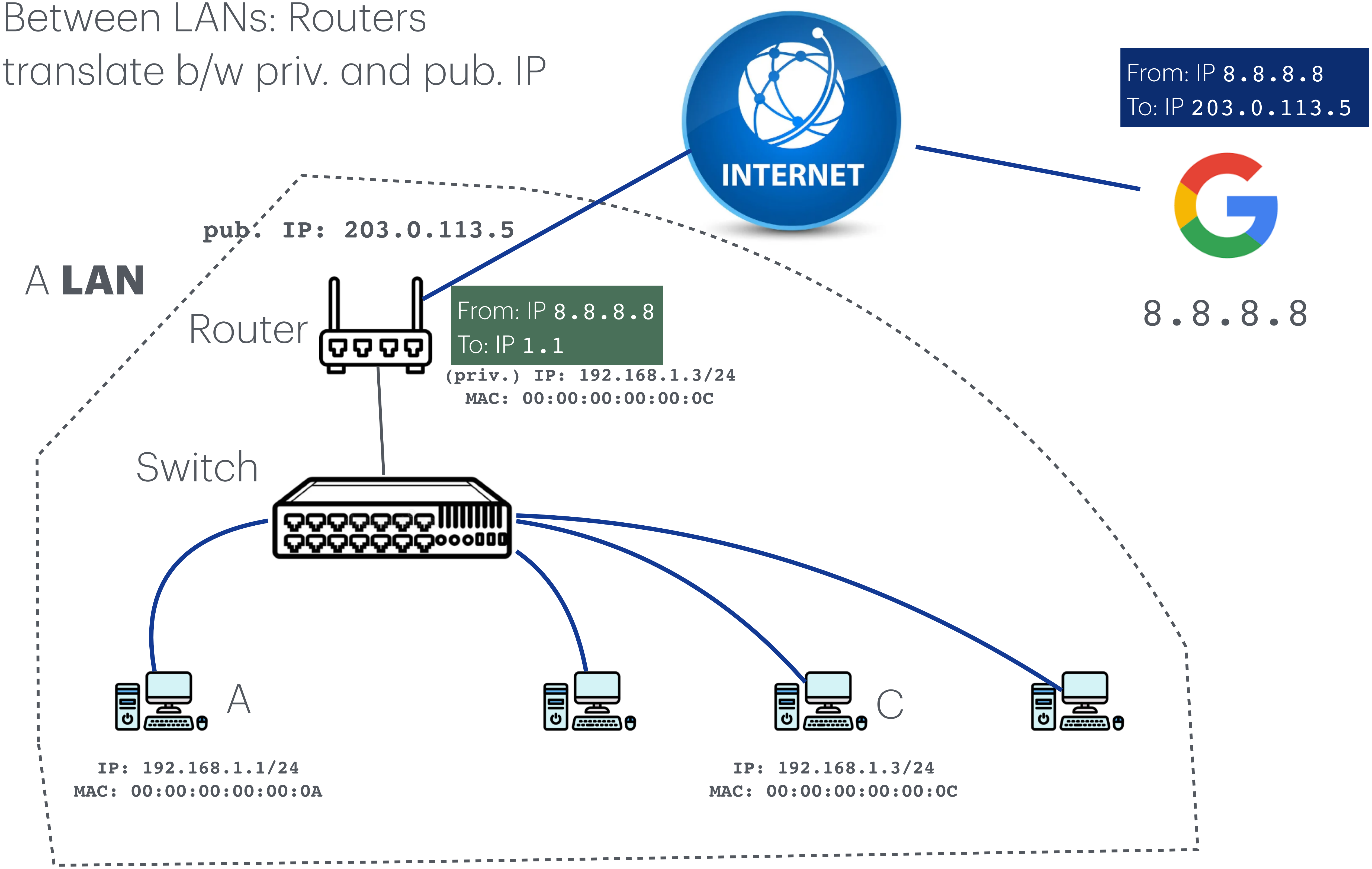
Communicating at the Link & Network Layer

NAT: Routers translate b/w priv. and pub. IP



Communicating at the Link & Network Layer

Between LANs: Routers translate b/w priv. and pub. IP



Two Problems

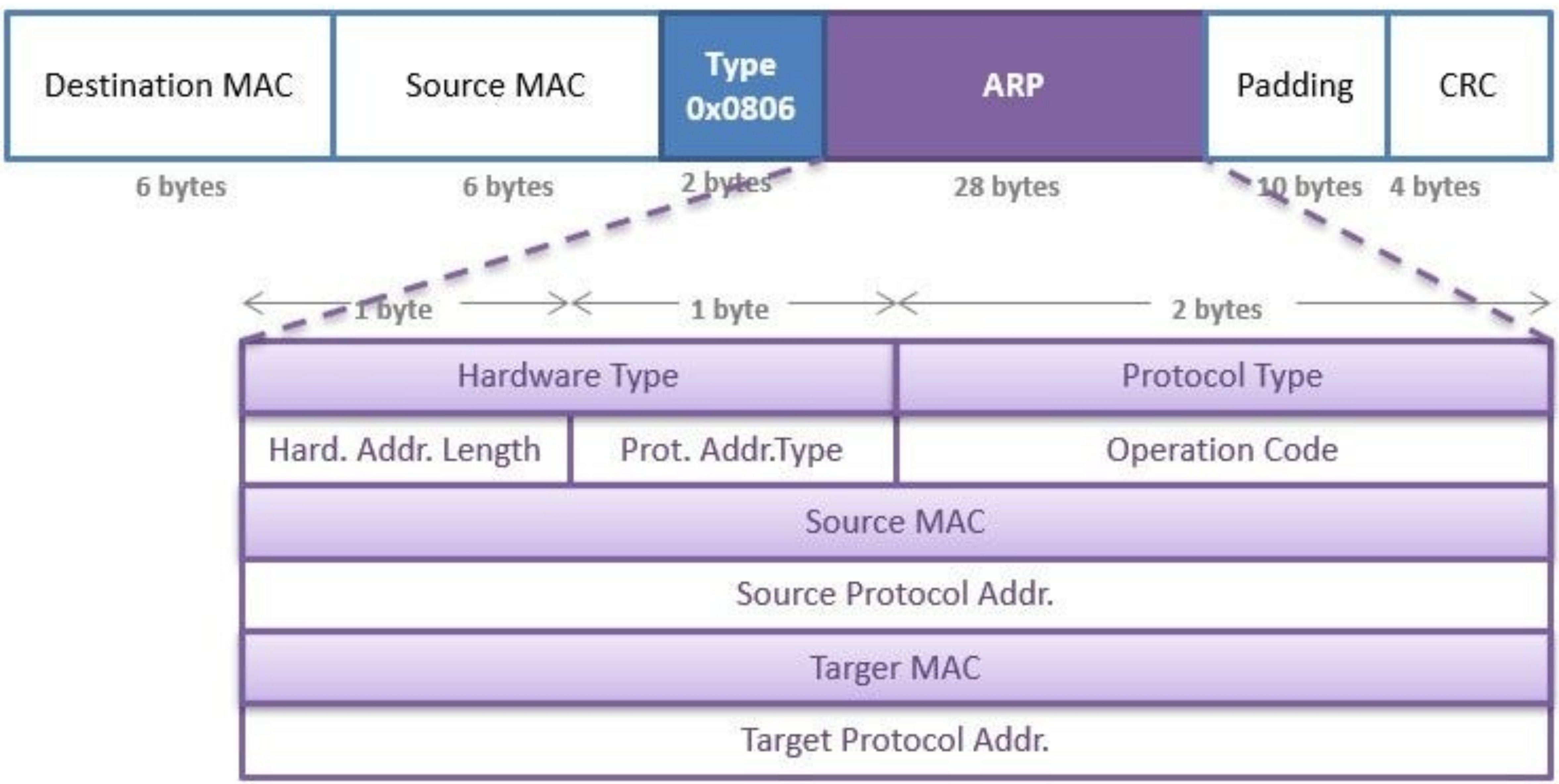
- **Local**: How does a host know what MAC address their destination has, given an IP address?
- **Internet**: How does each router know where to send each packet next?

Routing 101

ARP: Address Resolution Protocol

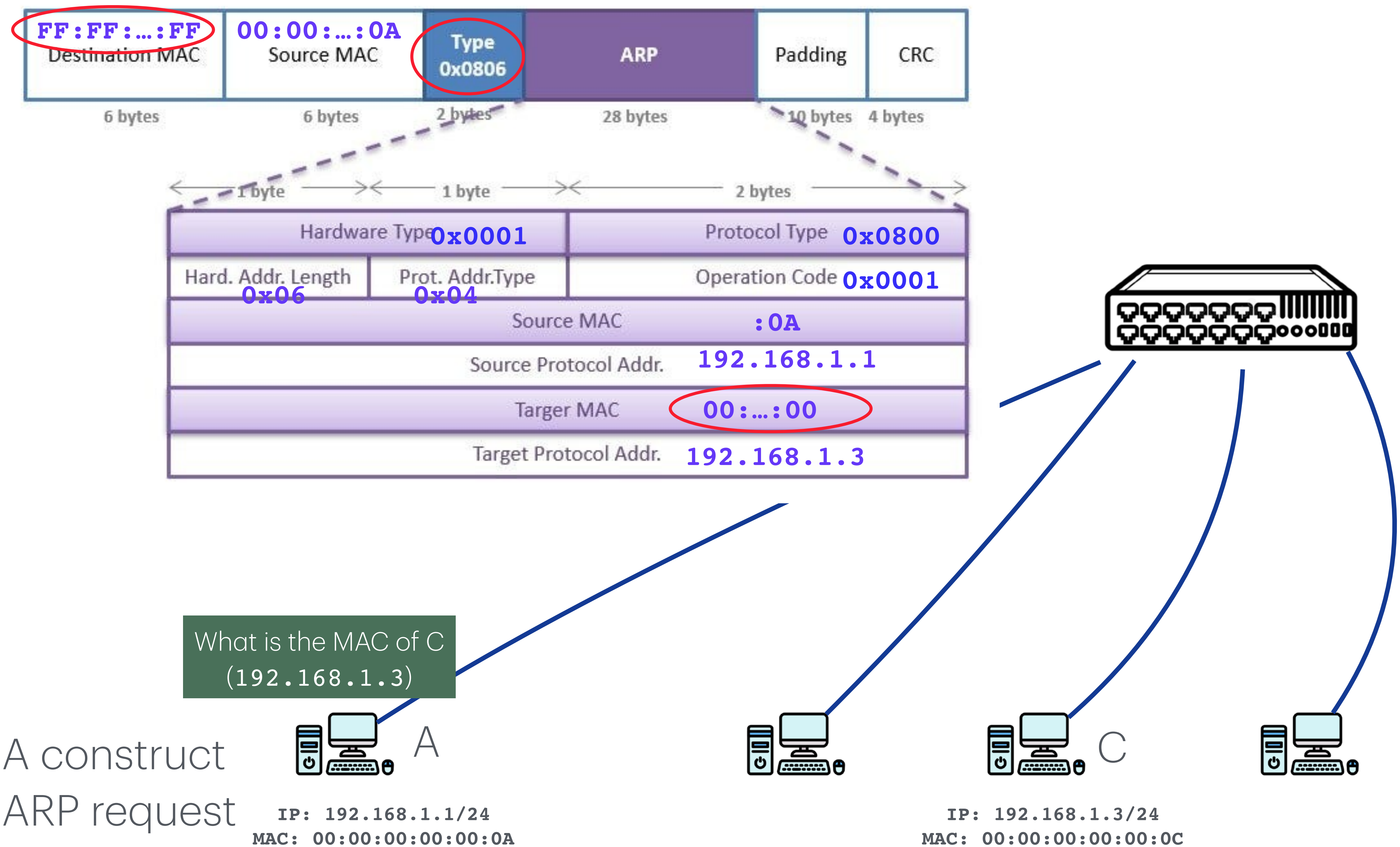
- ARP is a Network protocol that lets hosts **map IP addresses to MAC addresses**
    - Works at the boundary of Layer 2 (Link) and Layer 3 (Network)
- Host who needs MAC address `M` corresponding to IP address `N` broadcasts an ARP packet to LAN asking, "who has IP address `N`?"
- Host that has IP address `N` will reply, *"IP `N` is at MAC address `M`."*
- The updated ARP entries will be *cached* on hosts.

ARP Packet



Destination MAC
Source MAC
Type
0x0806
ARP
Padding
CRC
6 bytes
6 bytes
2 bytes
28 bytes
10 bytes
4 bytes
1 byte
1 byte
2 bytes
Hardware Type
Protocol Type
Hard. Addr. Length
Prot. Addr.Type
Operation Code
Source MAC
Source Protocol Addr.
Targer MAC
Target Protocol Addr.

ARP Example

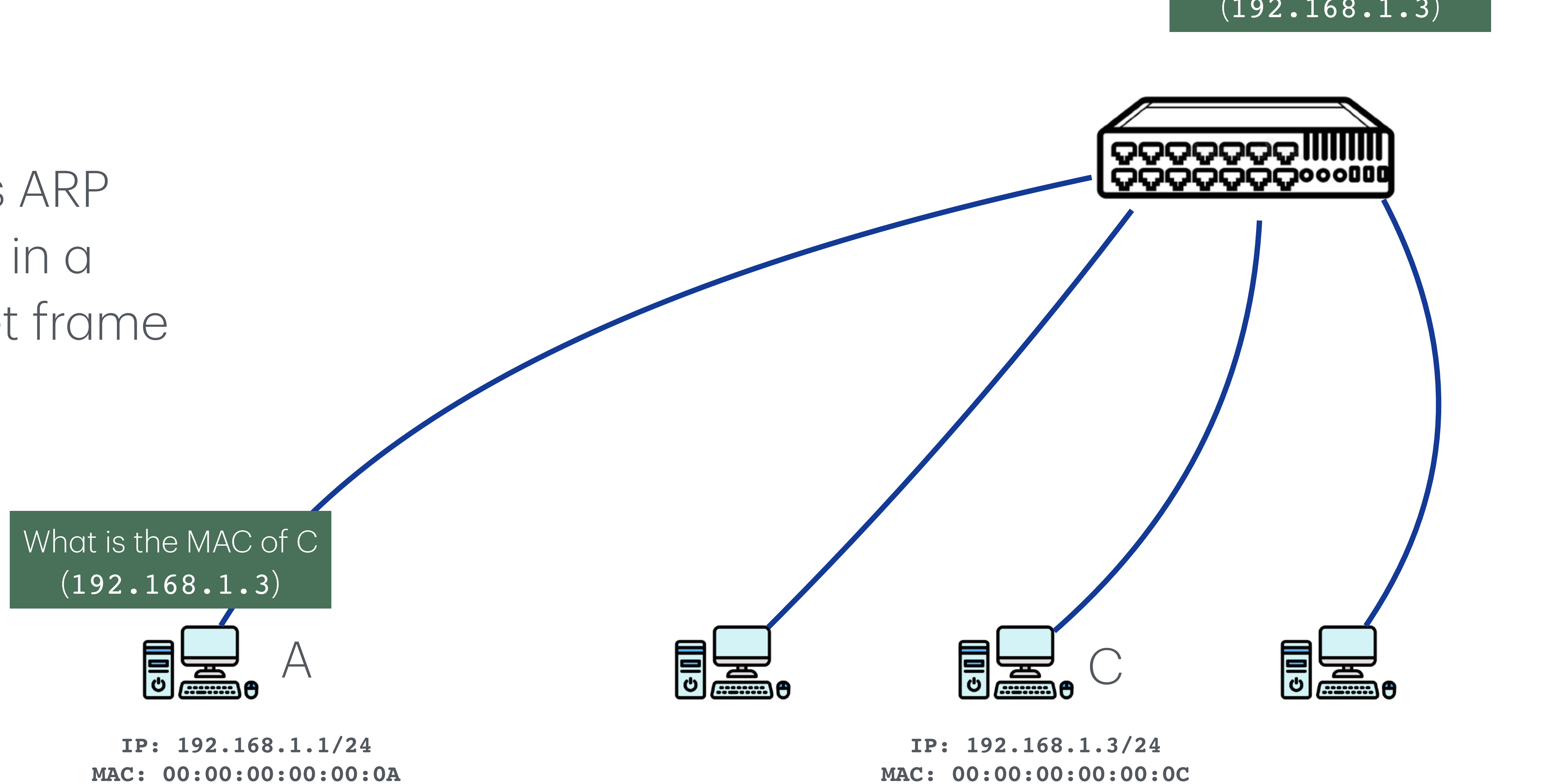


A construct ARP request

ARP Example

FF:FF:…:FF Destination MAC	00:00:…:0A Source MAC	Type 0x0806	ARP	Padding	CRC

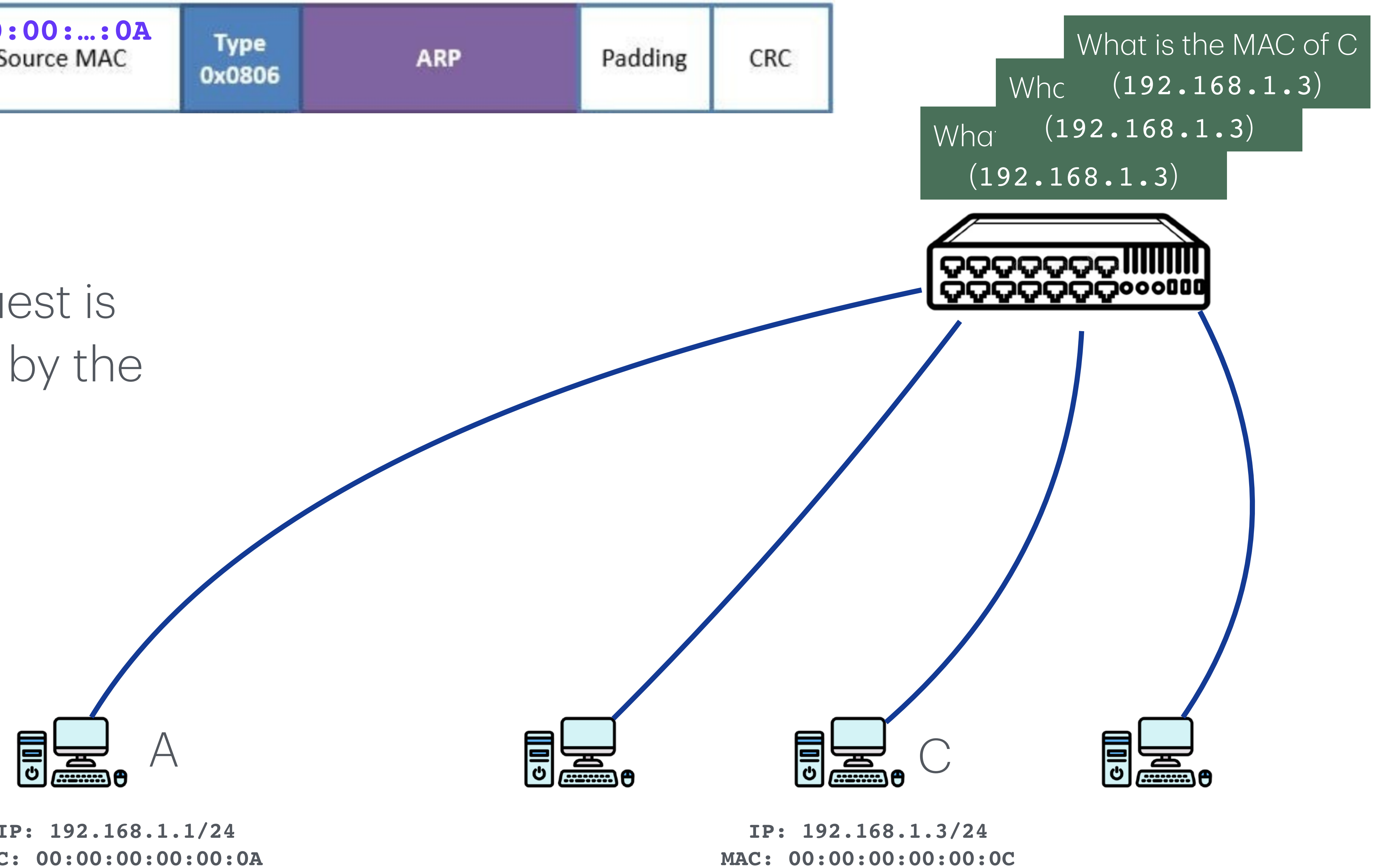
A sends ARP request in a Ethernet frame



ARP Example

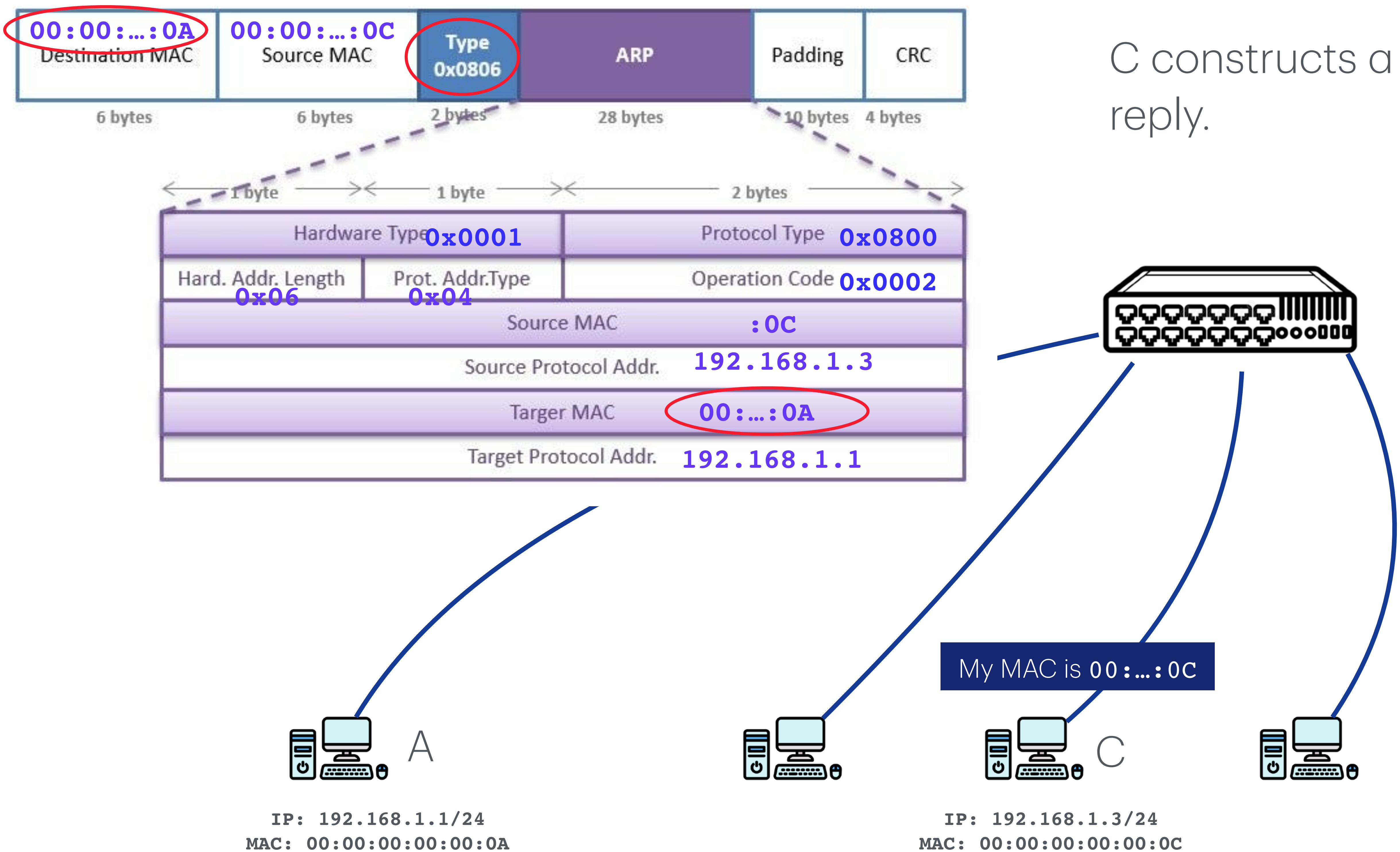
FF:FF:…:FF Destination MAC	00:00:…:0A Source MAC	Type 0x0806	ARP	Padding	CRC

The ARP request is broadcasted by the switch.



ARP Example

C constructs a ARP reply.



Gratuitous ARP

- When the host's IP address or MAC address has changed, the host can use ARP as a simple announcement protocol
  - ARP Replies can be broadcasted *even if there is no ARP Request*. All other hosts in the network may accept this reply
  - Host C can send the *fake ARP even if Host A did not send an ARP Request*. And, Host A may accept the fake ARP and update its cache.

ARP Security

- Any host on the LAN can send ARP requests and replies: *any host can claim to be another host on the local network!*
  - This is called **ARP spoofing** (a.k.a. ARP poisoning)

- This allows any host $\mathbf{X}$ to force IP traffic between any two other hosts $\mathbf{A}$ and $\mathbf{B}$ to flow through $\mathbf{X}$ (*MitM!*)
  - Claim N_A is at attacker's MAC address M_X
  - Claim N_B is at attacker's MAC address M_X
  - Re-send traffic addressed to N_A to M_A, and vice versa

ARP Security

- **Sniffing**
  - By using ARP spoofing, all the traffic can be directed to the hackers. It is possible to perform sniffing on a switched network now.
- **DoS**
  - Updating ARP caches with *non-existent* MAC addresses will cause frames to be dropped.
  - These could be sent out in a sweeping fashion to all clients on the network in order to cause a Denial of Service attack (DoS).

ARP Attack Model

- **Attack Model:**
  - Attacker should reside in *the same local network* as the victims
- **Root Cause:**
  - There is no method in the ARP protocol by which a host can *authenticate* the peer from which the packet originated

Defenses against ARP Spoofing

- Use static ARP entries
  - Cannot be updated (spoofed)
  - ARP replies are ignored
  - ARP table needs a static entry for each machine on the network
- Large overhead
  - Deploying these tables
  - Keep the table up-to-date

Defenses against ARP Spoofing

- **S-ARP Protocol**
  - S-ARP provides message authentication only
  - S-ARP uses asymmetric cryptography
  - Any S-ARP enabled host is identified by its own IP address and has a public/private key pair
    - A simple certificate provides the binding between the host identity and its public key <IP, PubKey>
  - A host that wants to connect to the LAN must first generate a public/private key pair and send its certificate to a Authoritative Key Distributor.

Routing (BGP)

- BGP (Border Gateway Protocol): protocol that allows routers to exchange information about their **routing tables**
- Each router announces what it can route to all of its neighbors.
- Every router maintains a global table of routes

Routing tables

Example routing table contents

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.100	10
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.255.0	192.168.0.100	192.168.0.100	10
192.168.0.100	255.255.255.255	127.0.0.1	127.0.0.1	10
192.168.0.1	255.255.255.255	192.168.0.100	192.168.0.100	10

- **Net. dest**. & **Netmask**: identifies the destination's subnet
- **Gateway**: the *next hop* through which the dest. can be reached
- **Interface**: what locally available interface (e.g. the NIC card) is responsible for reaching the gateway
- **Metric**: the associated cost of using the indicated route

Routing tables

```
$ netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags               Netif Expire
default            link#34            UCSg                utun4
default            10.83.31.254       UGScIg                en0
default            link#21            UCSIg           bridge100      !
default            link#23            UCSIg           bridge101      !
3.14.34.68         link#34            UHWIig              utun4
3.233.158.24       link#34            UHWIig              utun4
8.8.8.8            link#34            UHW3Ig              utun4      8
10.14/16           10.14.0.2          UGSc                utun4
10.14.0.2          10.14.0.2          UH                  utun4
10.83/19           link#11            UCS                   en0      !
10.83.3.21         10:9f:41:b9:61:93  UHLWI                 en0    500
10.83.3.133        0:c:29:8f:d2:4f    UHLWI                 en0      !
10.83.6.113        60:3e:5f:33:c7:7a  UHLWI                 en0    591
10.83.7.199        b2:4b:52:cc:85:a9  UHLWI                 en0    784
10.83.10.20/32     link#11            UCS                   en0      !
```

Interface

fallback route

conn. to
interface #34

tunnel if.
usually a VPN

A Specific host

A subnet

ethernet if.

BGP Protocol

- Working at the Application Layer. Very complex.
  - Enabling routing info exchange between Autonomous Systems (AS). (think of it as large networks managed by ISPs)
  - Key step: **Route Advertisement**.
    - Each AS advertises which IP prefixes (ranges of IP addresses) it is responsible for.
    - Other ASes thus know which IP addresses they can reach through a particular AS.

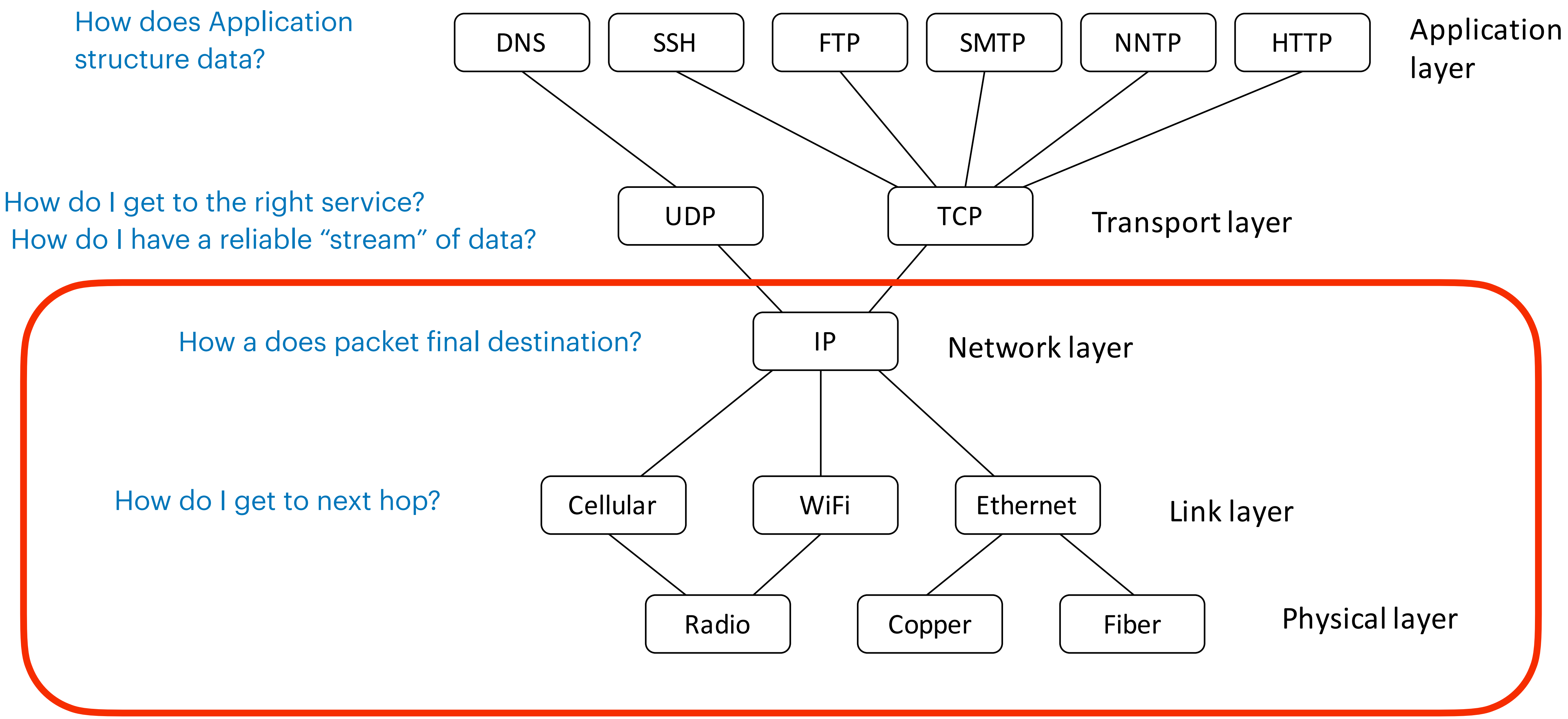
BGP Security

- Like ARP, no built-in security
  - No authentication of a BGP advertisement.
    - Again, can be mitigated by PKI
  - Slow convergence
    - When major routing change occurs, BGP can take days to converge.
    - Slowdown recovery from attacks.

Pakistan hijacks YouTube

- On 24 February 2008, Pakistan Telecom (AS 17557) began advertising a small part of YouTube's (AS 36561) assigned network
- PCCW (3491) did not validate Pakistan Telecom's (17557) advertisement for `208.65.153.0/24`
- Youtube offline.

Where we are now



Questions?