NetSec MP Checkpoint 1

CS 461 / ECE 422

Agenda

- Wireshark
- A first look at the MP network and traffic
- dig and curl
- Port scanning
- TCP SYN scan
- nmap and scapy

Reminder

Follow the NetSec handout VM setup instructions closely!

(netid.txt content needs to end with a newline char)

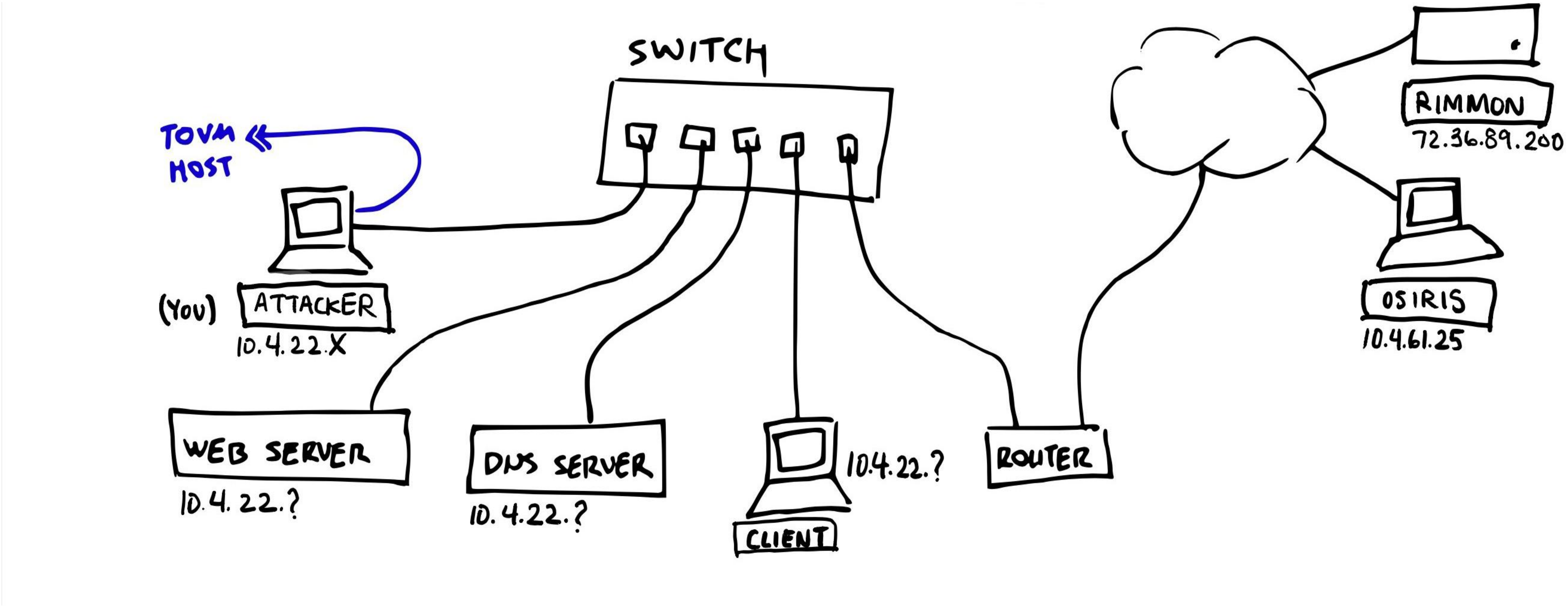
Wireshark

- A tool for passively capturing and viewing packets on a host network interface
  - `wireshark` in terminal to open

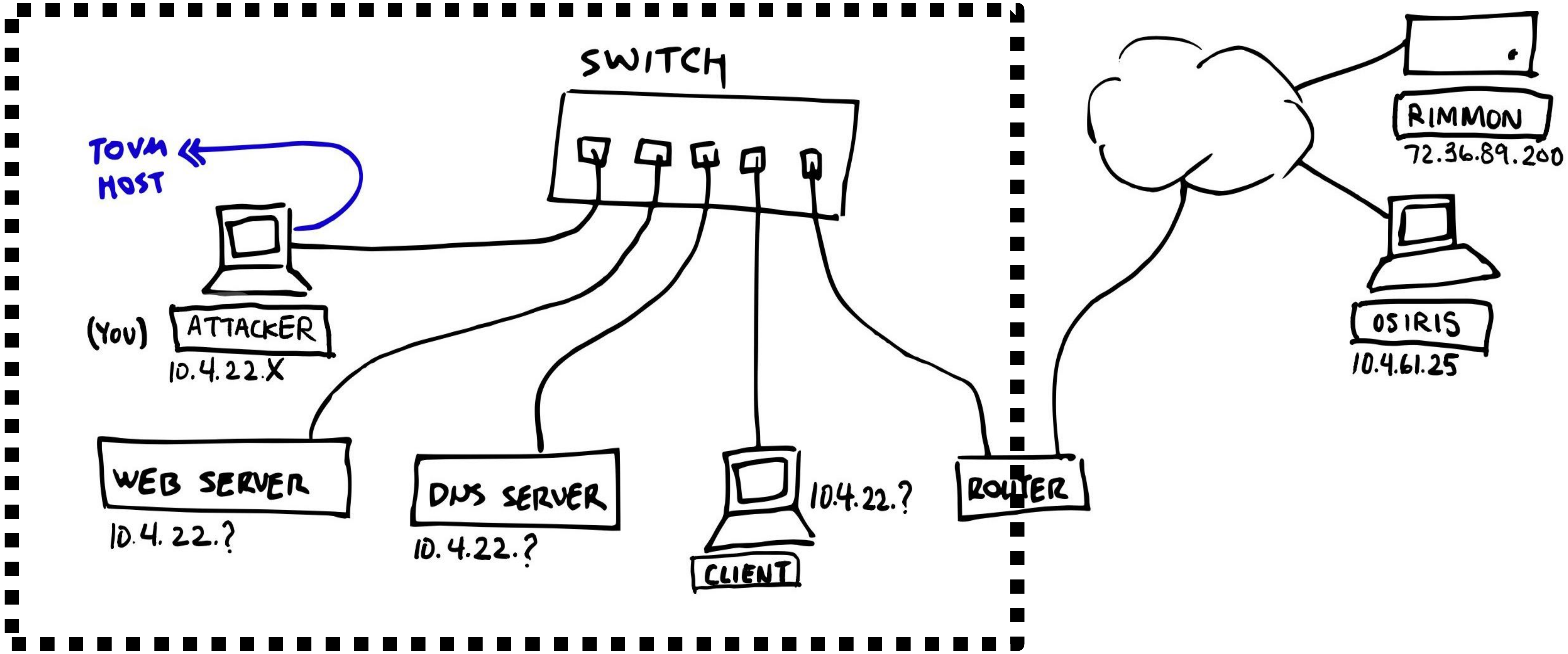
- Provide packet parsing and filtering capability
  - Give parse outcome via GUI
  - Powerful display filters (with && || support)
    - `ip.addr == a.b.c.d` (show pkts with the IP addr)
    - `eth.addr == a:b:c:d:e:f` (show pkts with the Ethernet MAC addr)
    - `tcp` / `udp` (show pkts related to a transport protocol)
    - tcp.port == X / udp.port == X (show pkts with the transport port number)

- When running Wireshark: similar to a **passive attacker/eavesdropper**

NetSec MP Network

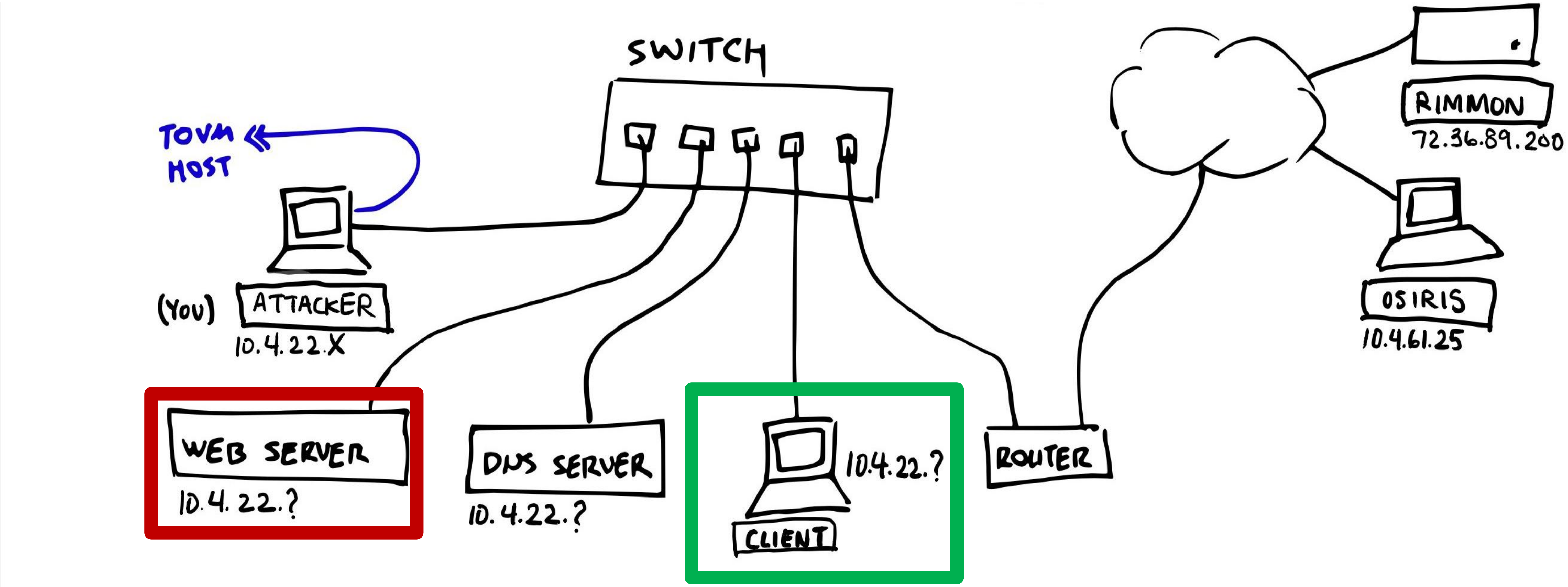


NetSec MP Network



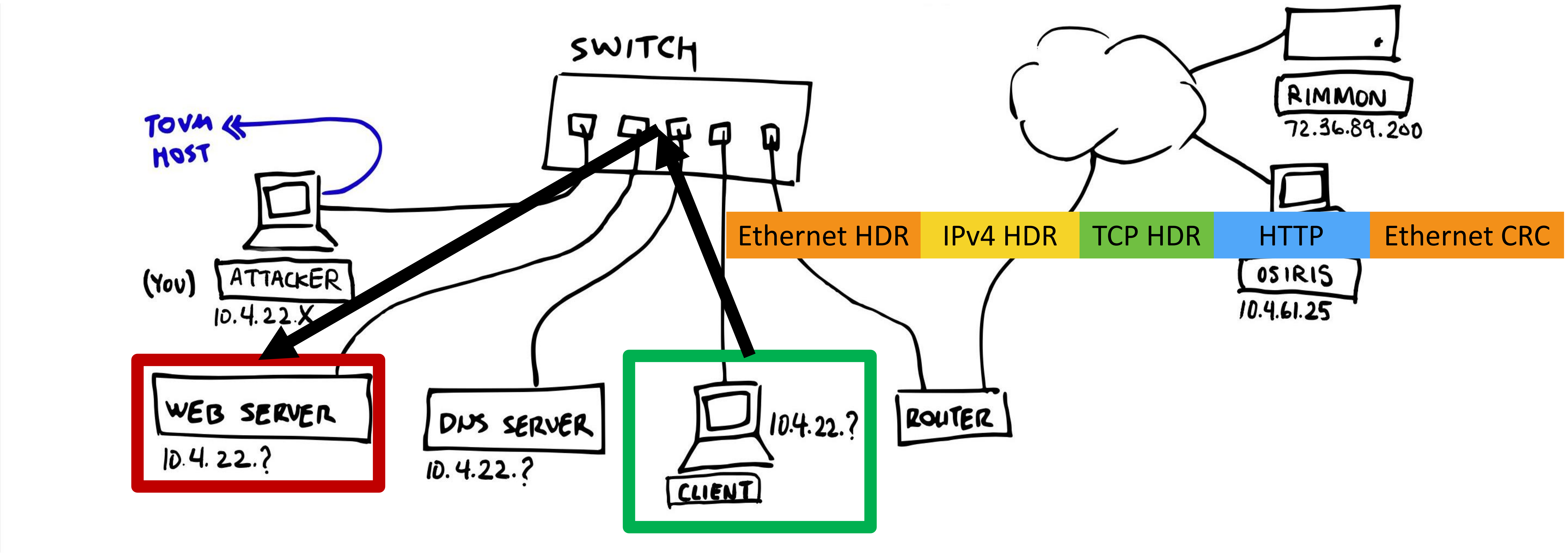
Local (area) network under 10.4.22.0/24 subnet
(all 10.4.22.X hosts are in this subnet)

Within Network Communication



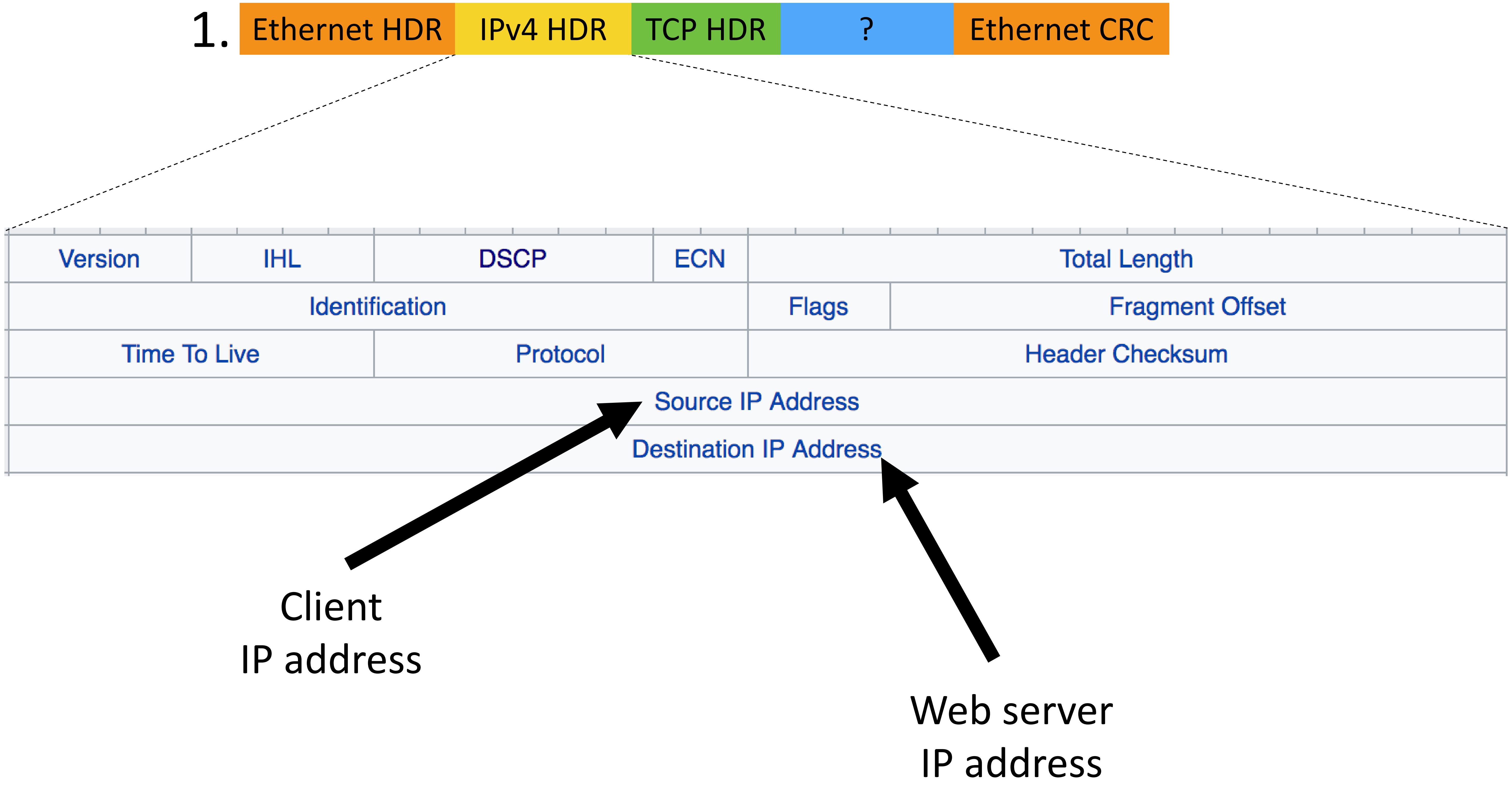
Client is sending HTTP traffic to the web server. What does it look like?

Within Network Communication



Ethernet HDR	IPv4 HDR	TCP HDR	HTTP	Ethernet CRC

Within Network Communication



1. Ethernet HDR | IPv4 HDR | TCP HDR | ? | Ethernet CRC

Version | IHL | DSCP | ECN | Total Length

Identification | Flags | Fragment Offset

Time To Live | Protocol | Header Checksum

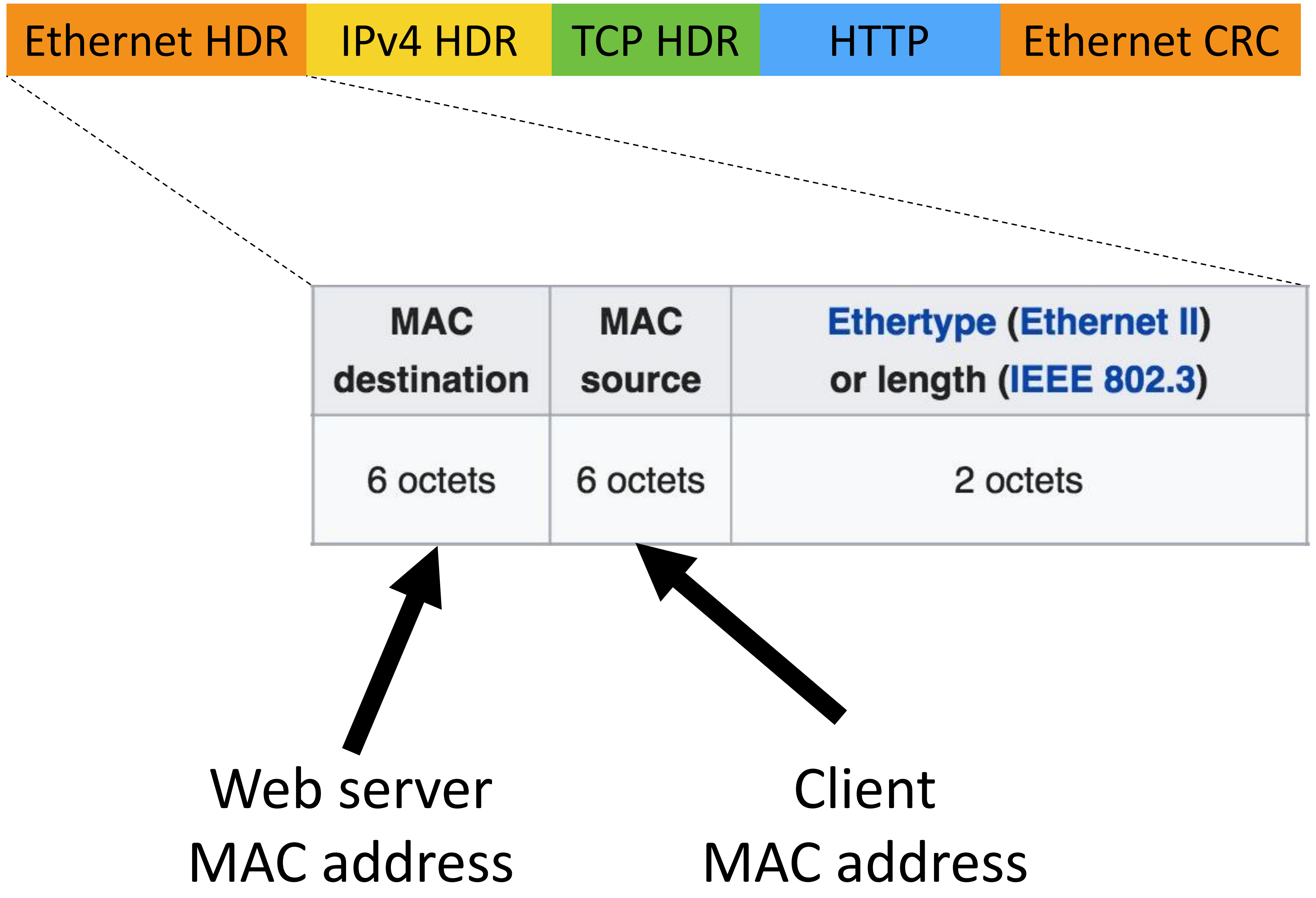
Source IP Address

Destination IP Address

Client
IP address

Web server
IP address

Within Network Communication



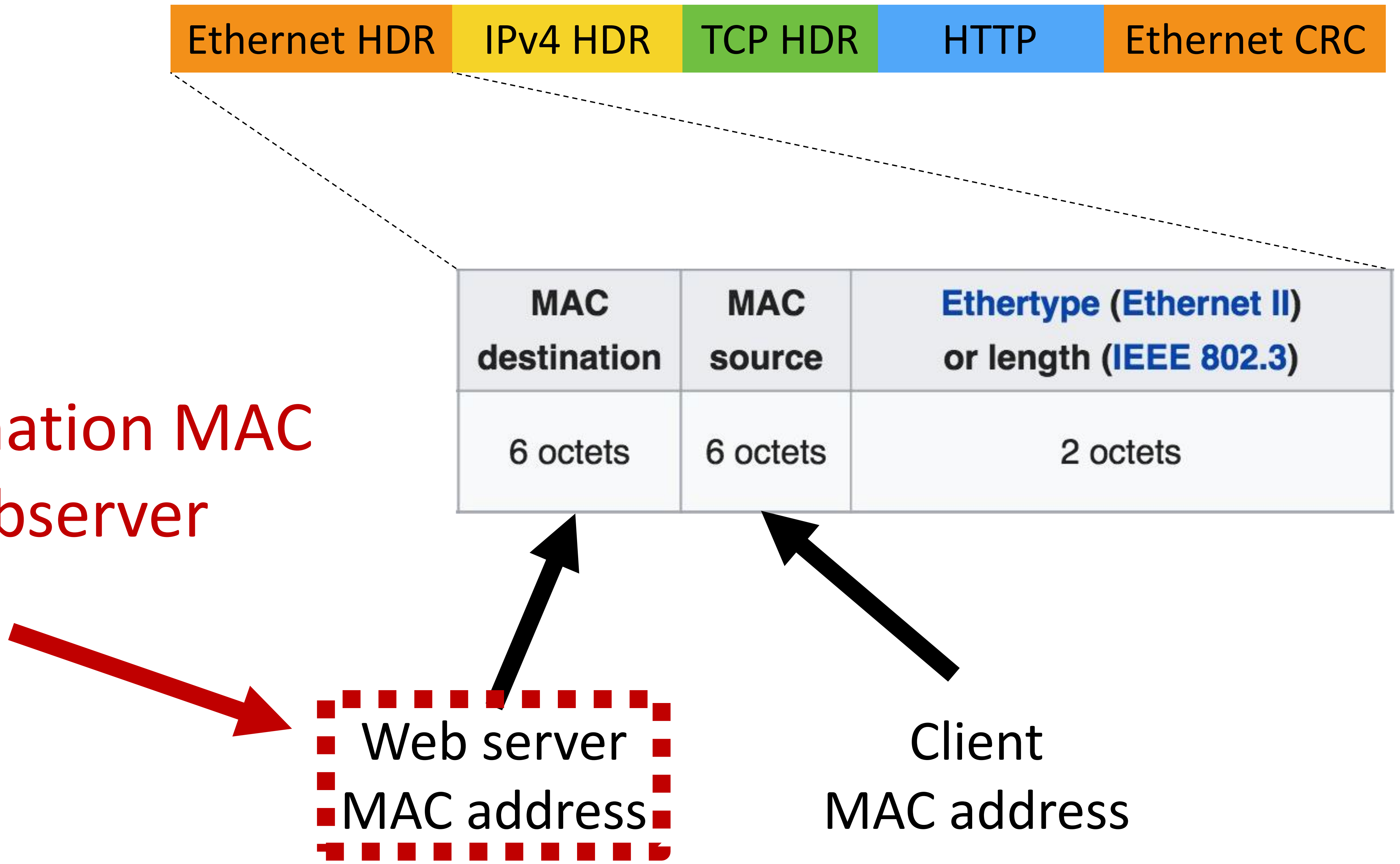
Ethernet HDR | IPv4 HDR | TCP HDR | HTTP | Ethernet CRC

MAC destination	MAC source	Ethertype (Ethernet II) or length (IEEE 802.3)
6 octets	6 octets	2 octets

Web server
MAC address

Client
MAC address

Within Network Communication



Ethernet HDR	IPv4 HDR	TCP HDR	HTTP	Ethernet CRC

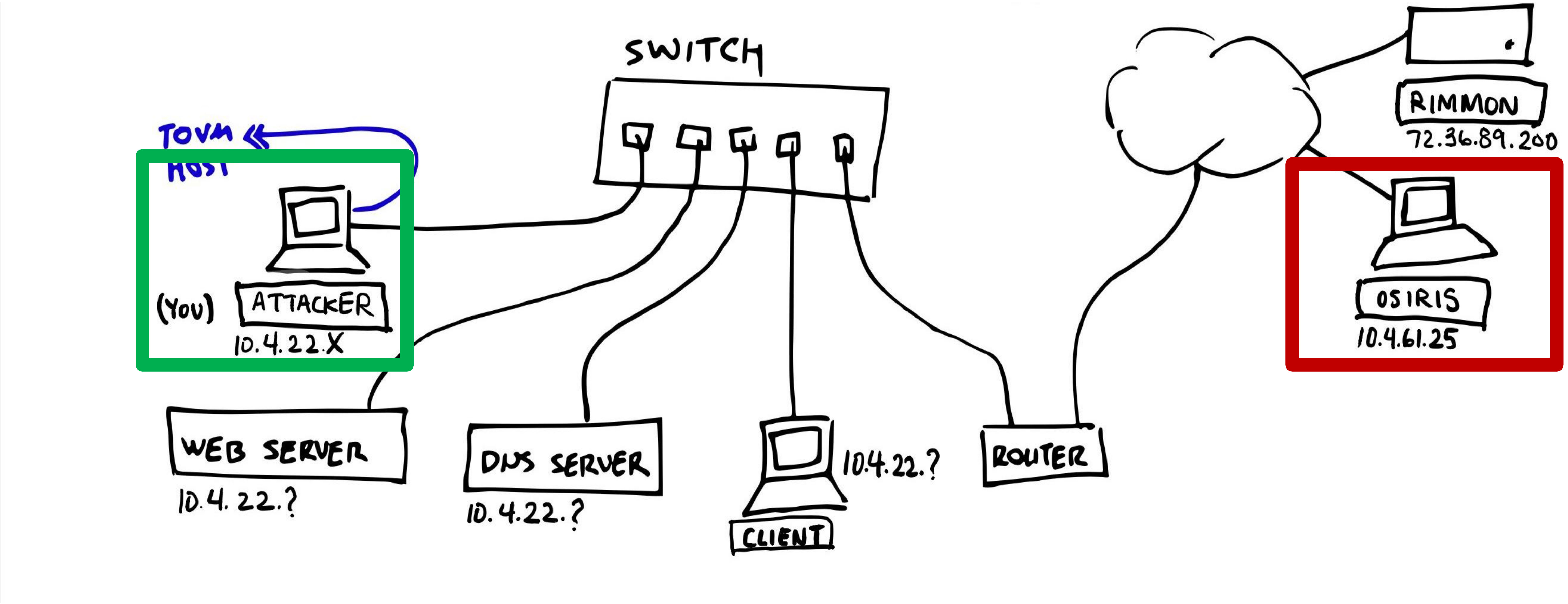
MAC destination	MAC source	Ethertype (Ethernet II) or length (IEEE 802.3)
6 octets	6 octets	2 octets

Switch use destination MAC to forward to webserver

Web server
MAC address

Client
MAC address

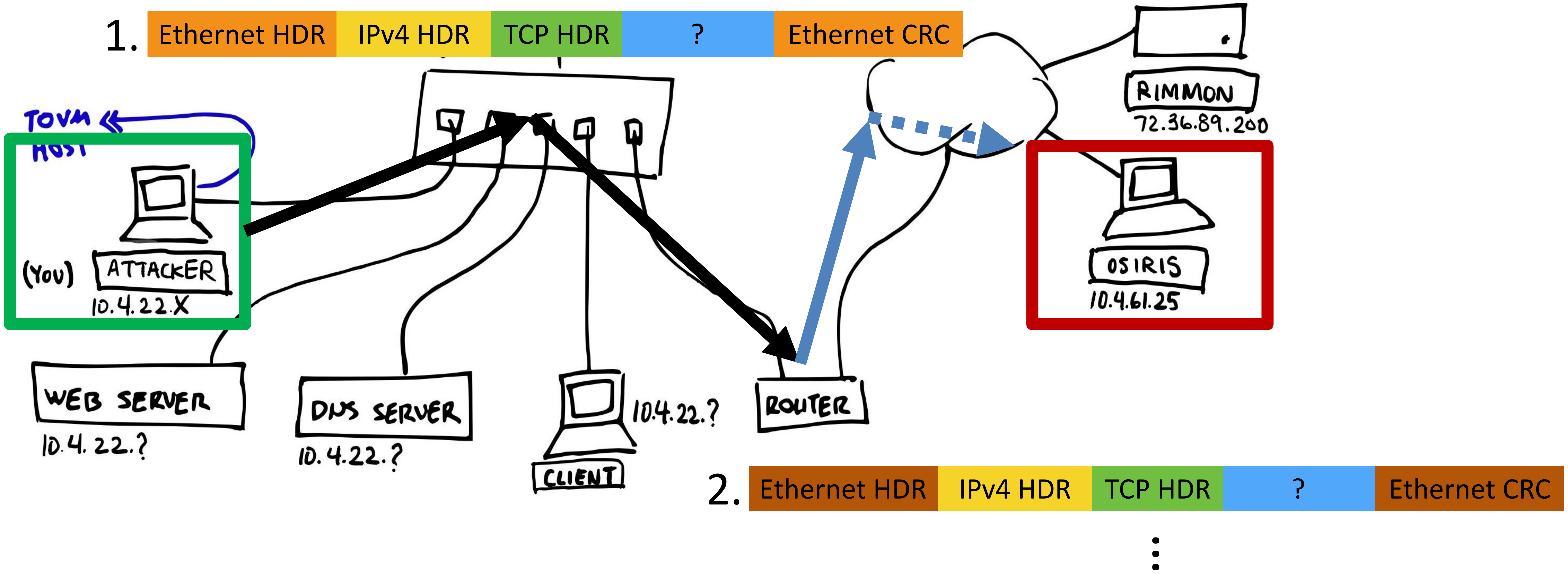
Among Networks Communication



Attacker is sending TCP traffic to Osiris.

What does it look like?

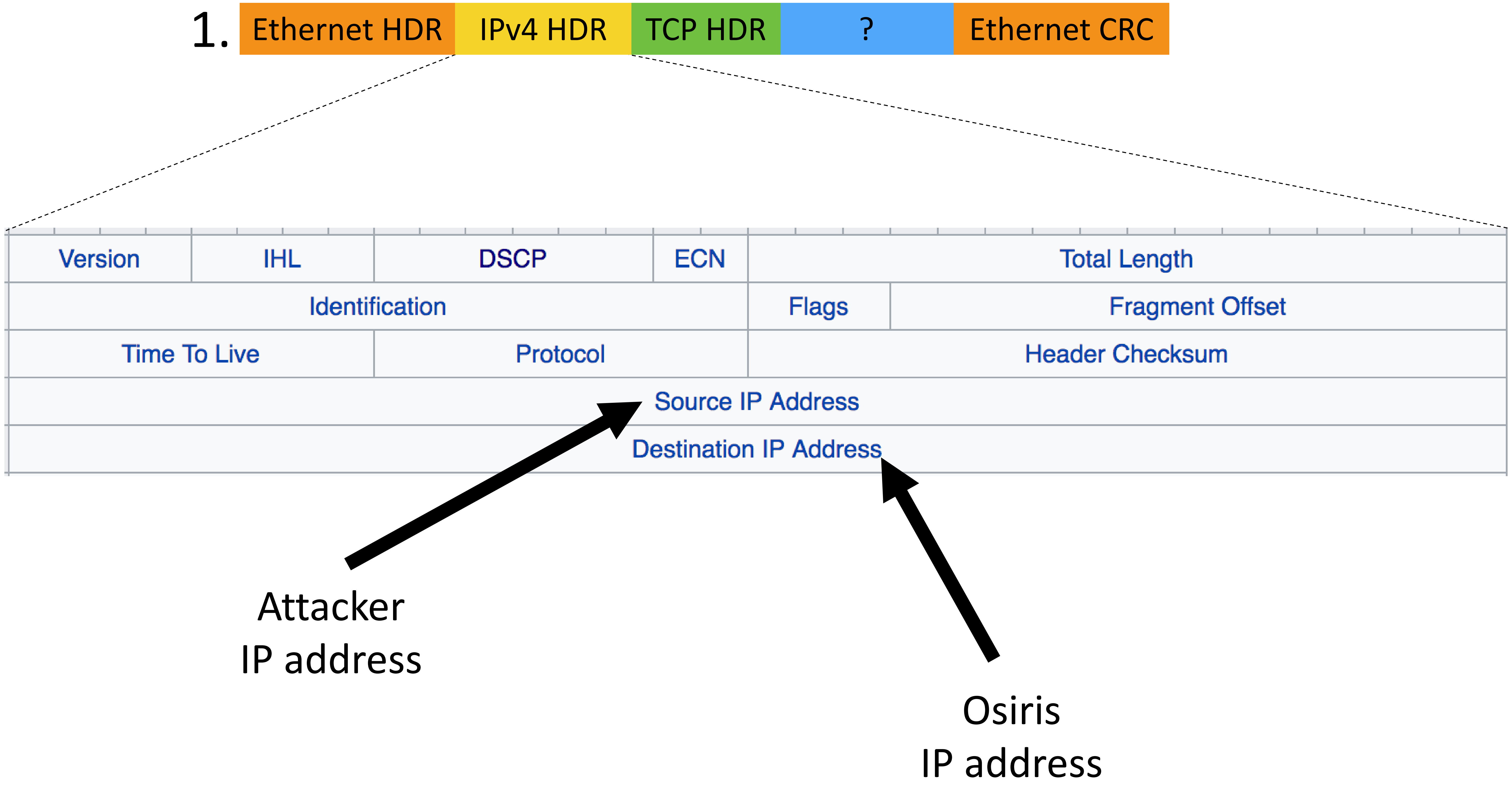
Among Networks Communication



1.	Ethernet HDR	IPv4 HDR	TCP HDR	?	Ethernet CRC
2.	Ethernet HDR	IPv4 HDR	TCP HDR	?	Ethernet CRC

⋮

Among Networks Communication



1. Ethernet HDR | IPv4 HDR | TCP HDR | ? | Ethernet CRC

Version | IHL | DSCP | ECN | Total Length

Identification | Flags | Fragment Offset

Time To Live | Protocol | Header Checksum

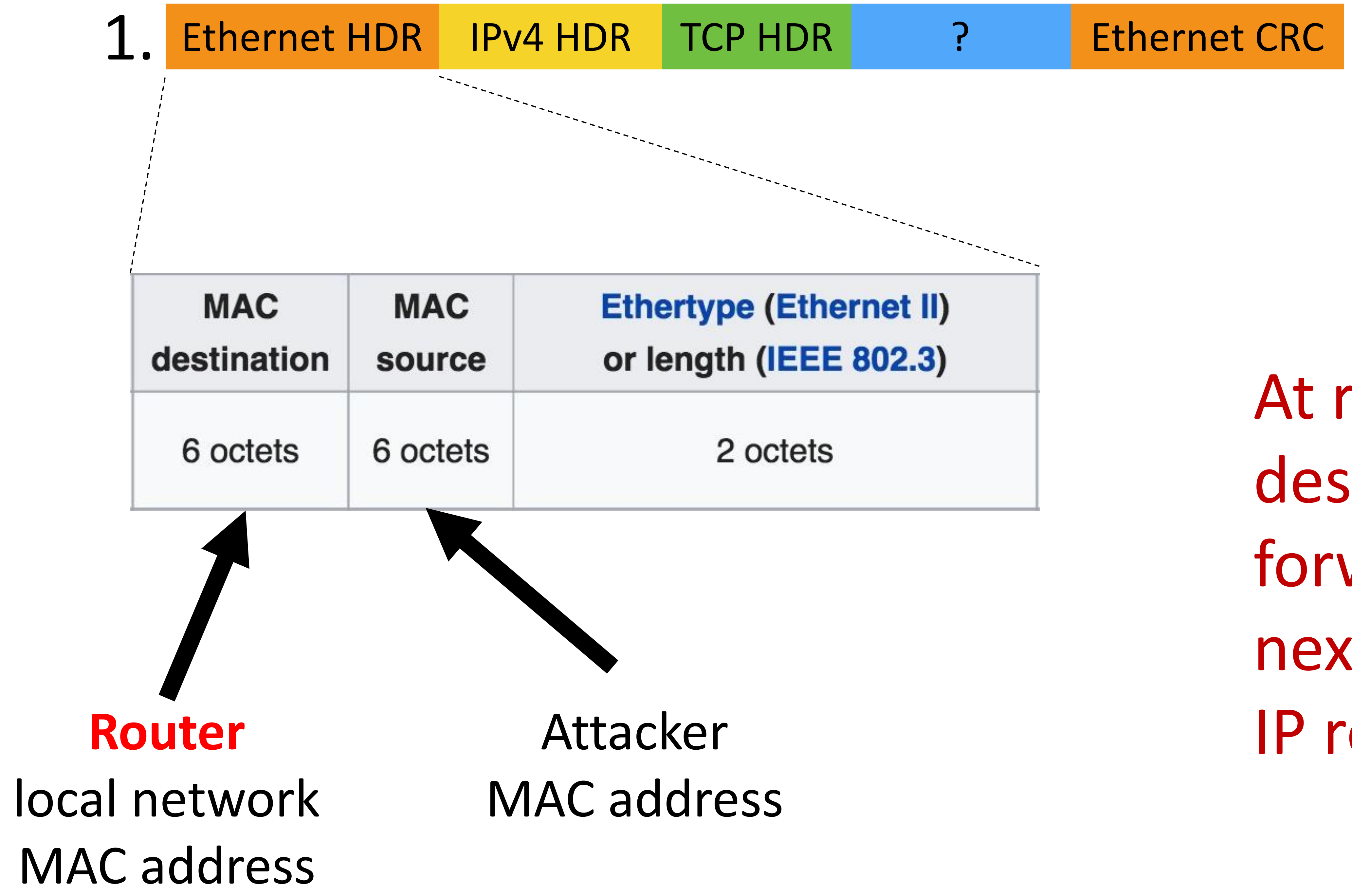
Source IP Address

Destination IP Address

Attacker
IP address

Osiris
IP address

Among Networks Communication



1.	Ethernet HDR	IPv4 HDR	TCP HDR	?	Ethernet CRC

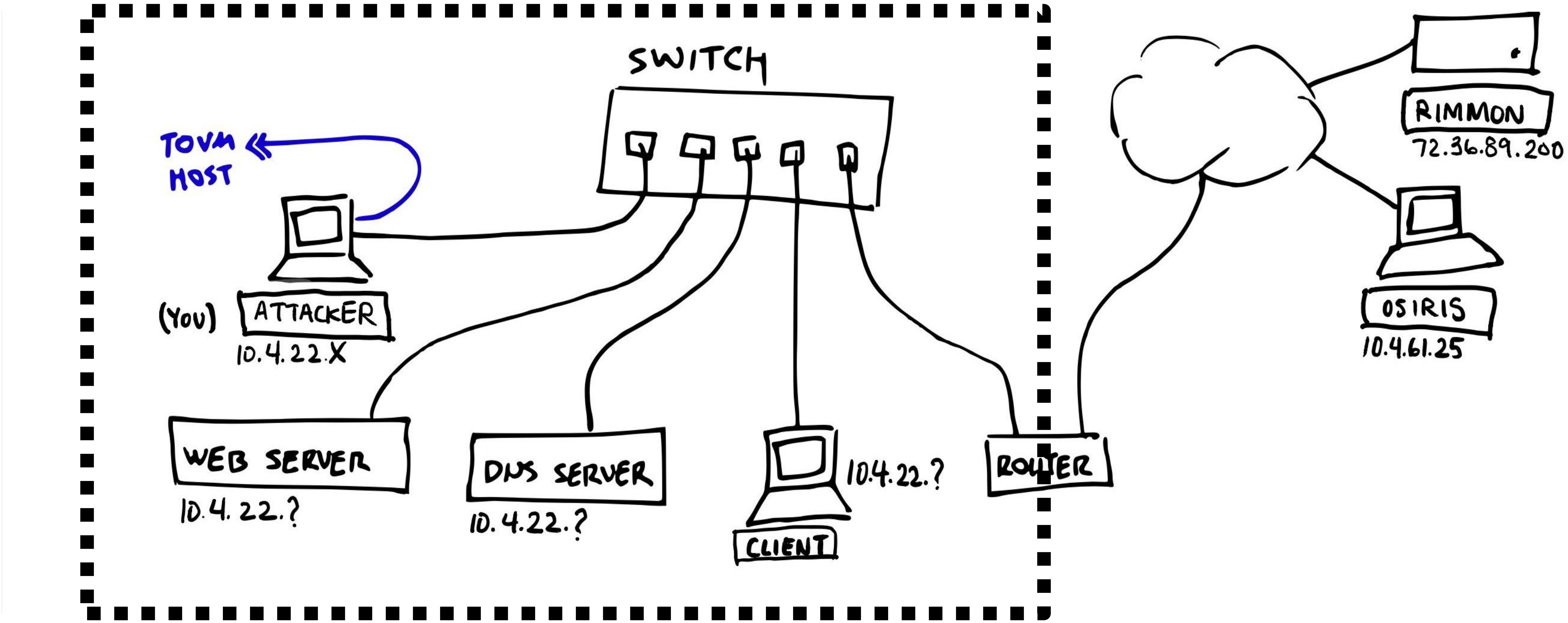
MAC destination	MAC source	Ethertype (Ethernet II) or length (IEEE 802.3)
6 octets	6 octets	2 octets

Router local network MAC address

Attacker MAC address

At router, src MAC and dest MAC are rewritten to forward the packet to the next hop. Src IP and dest IP remain the same.

Gateway



Gateway: a network device connecting multiple networks. Which node is the gateway here?

dig

- Tool to query DNS name servers

- `dig [options] [domains …]`
  - E.g. `dig www.bankofbailey.com` => Resolve www.bankofbailey.com by querying the OS configured DNS name server(s)

curl

- Tool to transfer data from or to a server
  - In this MP, we mainly use it for HTTP requests

- `curl [options] [URL …]`
  - E.g. `curl http://www.bankofbailey.com` => Get the content of index.html from the HTTP server hosted at www.bankofbailey.com

- CP1: find an option that includes unmodified HTTP response headers in output

Port Scanning

- The process of determining what (transport layer) ports on a host are open to traffic

- Common mapping between open port and service
  - 22 <=> SSH; 53 <=> DNS; 80 <=> HTTP

- Knowing a port is open => likely the host is running the corresponding service
  - Port scanning usually targets *interesting* ports that corresponds to well-known services
  - Can be over TCP or UDP

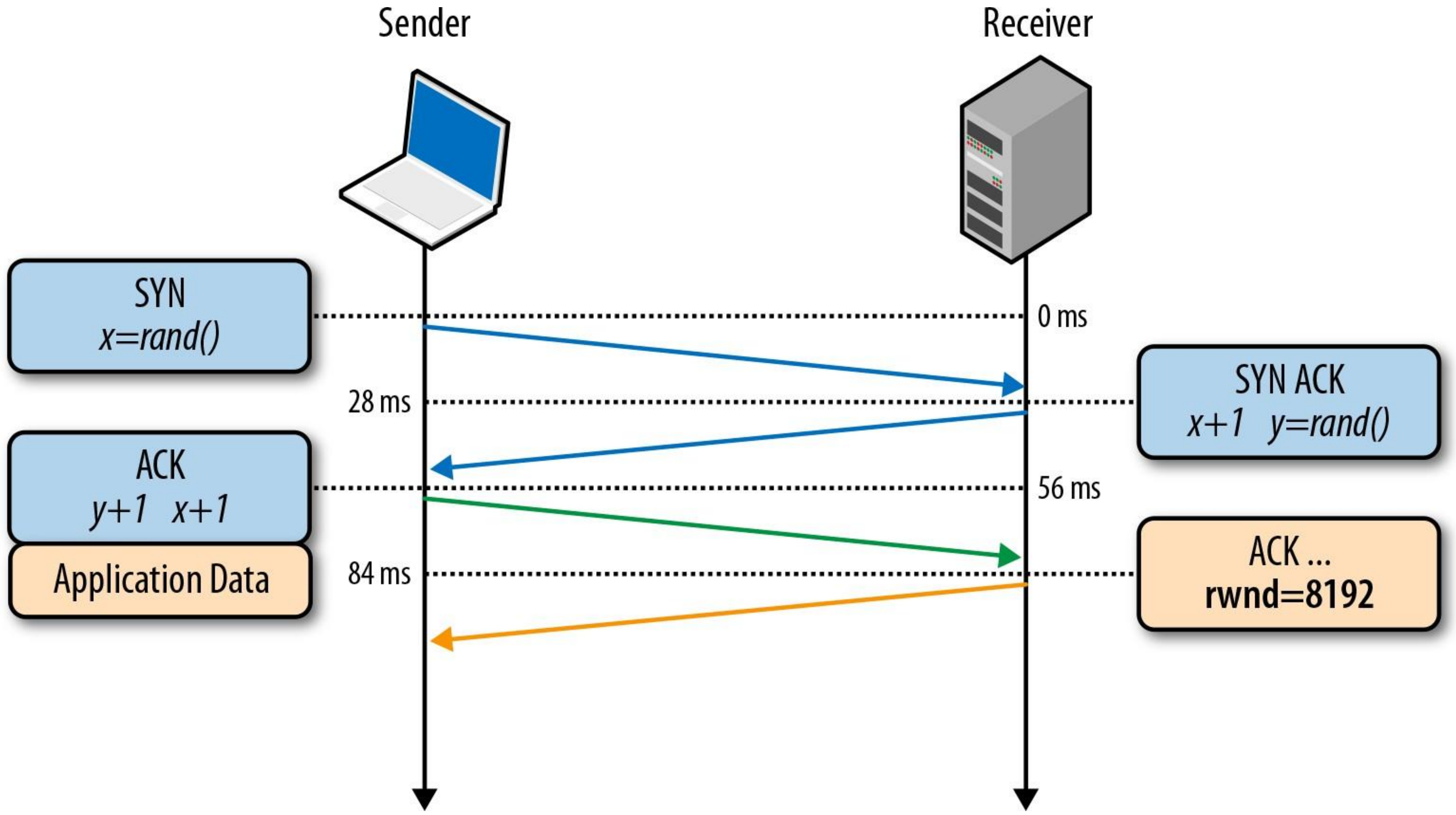
- Network attack reconnaissance often involve port scanning
  - Do not arbitrary scan!

TCP SYN Scan

- A port scanning method based on TCP 3-way handshake behavior

- TCP uses 3-way handshake to a establish connection
  - **SYN – SYNACK** – ACK

- Just send TCP SYN to target ports and observe if there is any SYNACK back
  - Port open if there is
  - RST to clean up



Nmap

- Tool to perform network discovery/port scanning

- `nmap [options] {target_spec}`
  - E.g. `nmap www.bankofbailey.com` => Perform TCP SYN scan on www.bankofbailey.com w.r.t. default scan ports
  - Ports may not be the ones you want

- Only installed on the attacker!

Scapy

- Library for packet manipulation
  - Sniffing packets and also crafting/sending packets
  - In this MP, used as a Python module

- Crafting packets can be done by stacking protocol layers
  - An IP packet to 10.4.22.1, encapsulating a TCP segment with destination port 5566:

```
IP(dst="10.4.22.1")/TCP(dport=5566)
```

  - Supported protocols: IP, TCP, UDP, Ether, ARP, DNS etc.
  - Default values for non-specified parameters (useful for checksum)

- Send packets: `send` (network layer or ARP), `send`**`p`** (link layer)
  - send and receive : `sr`/`sr1` (network layer or ARP), `sr`**`p`** (link layer)

Scapy Demo

```
from scapy.all import *

# 10.4.22.209 is attacker IP for demo machine
p = IP(dst="10.4.22.209")/TCP(dport=22, flags="A")
p.show()
rsp = sr1(p)
rsp.show()
```

Wireshark Slow?

- Campus VM farm VM can be slow for GUI, what should I do?
  - `tcpdump` (CLI) to pcap file, copy file to your laptop (e.g., `scp`), then view file on your laptop with local Wireshark
  - Or use `tshark` (CLI)

Thank You