Security goals vs attacker's model

Interception
Modification
Fabrication
Interruption

Let us consider **confidentiality, integrity and availability**

Network (in)security

Thierry Sans

How many of you …

- have programmed with **sockets** ?
- have taken a **networking course** ?
- have used tools like ?
  `ping, traceroute, ipconfig/ifconfig, nslookup`
  `netstat, netcat, nmap, wireshark`
- know what is :
  IP address, port, a canonical hostname
  client, server, router
  switch (or hub), gateway
- can explain with a fair amount of details :
  Ethernet, WiFi
  IP, TCP
  ARP, BGP, DNS

The Internet

1980's - few hosts connected : government institutions and universities

➡ Trustworthy environment

2019 - ~ 4.2 billion internet users: network of networks

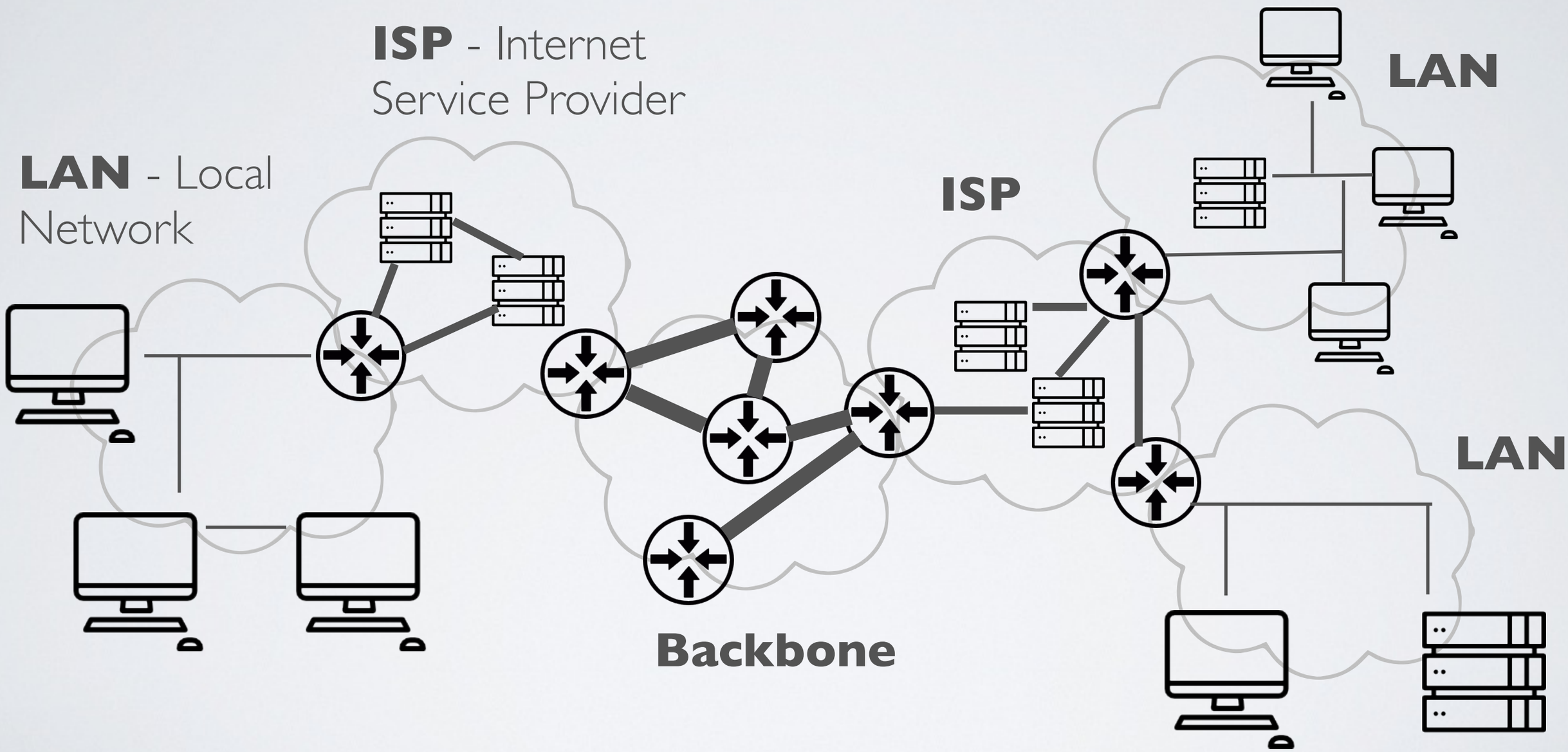
➡ Untrustworthy environment

➡ Internet (and its protocols) was
not designed for untrustworthy environment

A network of networks

ISP - Internet Service Provider

LAN - Local Network



A network of networks

```
traceroute: Warning: any.run has multiple addresses; using 172.67.20.89
traceroute to any.run (172.67.20.89), 64 hops max, 52 byte packets
 1  100.101.0.2 (100.101.0.2)  4.114 ms  4.712 ms  3.296 ms
 2  192.168.0.30 (192.168.0.30)  3.288 ms  2.711 ms  2.629 ms
 3  192.168.0.42 (192.168.0.42)  2.770 ms  2.788 ms
    192.168.0.40 (192.168.0.40)  2.758 ms
 4  bdr-ccbr-01-utsc-bell.gw.utoronto.ca (128.100.96.101)  3.612 ms
    bdr-dcb-01-utsc-cogeco.gw.utoronto.ca (128.100.96.97)  3.409 ms
    bdr-ccbr-01-utsc-bell.gw.utoronto.ca (128.100.96.101)  3.389 ms
 5  10.16.128.10 (10.16.128.10)  4.192 ms  5.445 ms
    10.17.128.10 (10.17.128.10)  4.312 ms
 6  10.96.16.20 (10.96.16.20)  6.174 ms  7.078 ms *
 7  10.96.16.34 (10.96.16.34)  7.118 ms  4.629 ms *
 8  ut-hub-utoronto1-if-internet.gtanet.ca (205.211.94.241)  7.701 ms  11.358 ms  7.556 ms
 9  te0-0-0-1.rcr13.b011027-3.yyz02.atlas.cogentco.com (38.104.251.81)  7.685 ms  7.280 ms  9.325 ms
10  te0-0-0-10.ccr32.yyz02.atlas.cogentco.com (154.54.0.121)  8.574 ms  7.414 ms  7.552 ms
11  be2993.ccr21.cle04.atlas.cogentco.com (154.54.31.225)  13.952 ms
    be2994.ccr22.cle04.atlas.cogentco.com (154.54.31.233)  16.504 ms
    be2993.ccr21.cle04.atlas.cogentco.com (154.54.31.225)  14.499 ms
12  be2717.ccr41.ord01.atlas.cogentco.com (154.54.6.221)  21.736 ms
    be2718.ccr42.ord01.atlas.cogentco.com (154.54.7.129)  21.165 ms
    be2717.ccr41.ord01.atlas.cogentco.com (154.54.6.221)  22.819 ms
13  be2766.ccr41.ord03.atlas.cogentco.com (154.54.46.178)  21.200 ms  22.776 ms
    be2765.ccr41.ord03.atlas.cogentco.com (154.54.45.18)  20.249 ms
14  38.122.181.134 (38.122.181.134)  26.333 ms  34.496 ms *
15  172.70.124.2 (172.70.124.2)  20.229 ms
    172.70.128.2 (172.70.128.2)  18.312 ms
    172.70.124.2 (172.70.124.2)  21.780 ms
16  172.67.20.89 (172.67.20.89)  17.568 ms  16.488 ms  17.244 ms
```

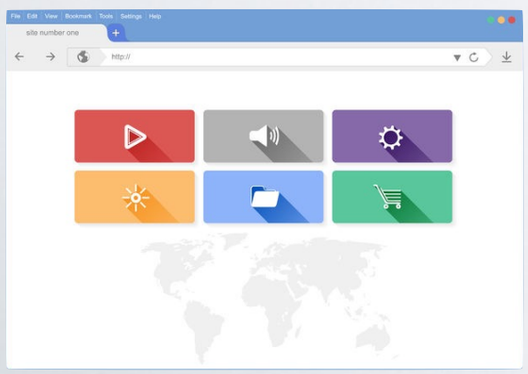
What is a protocol

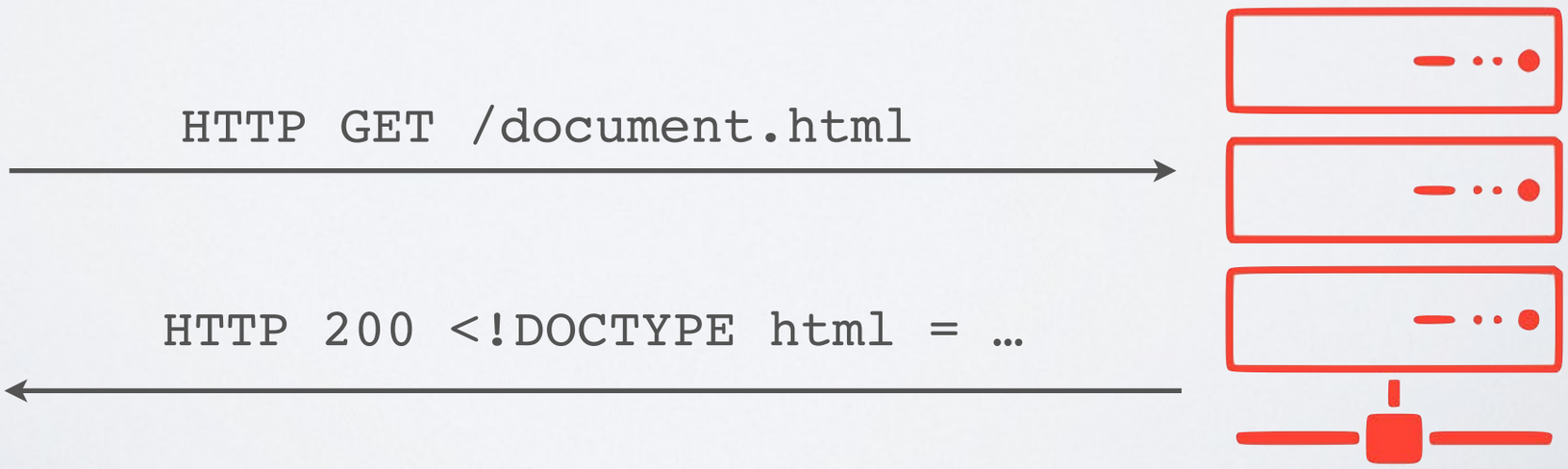
Communication protocol

is an agreement on how communication should take place

- defines the data encoding and/or format
- defines the message sequence

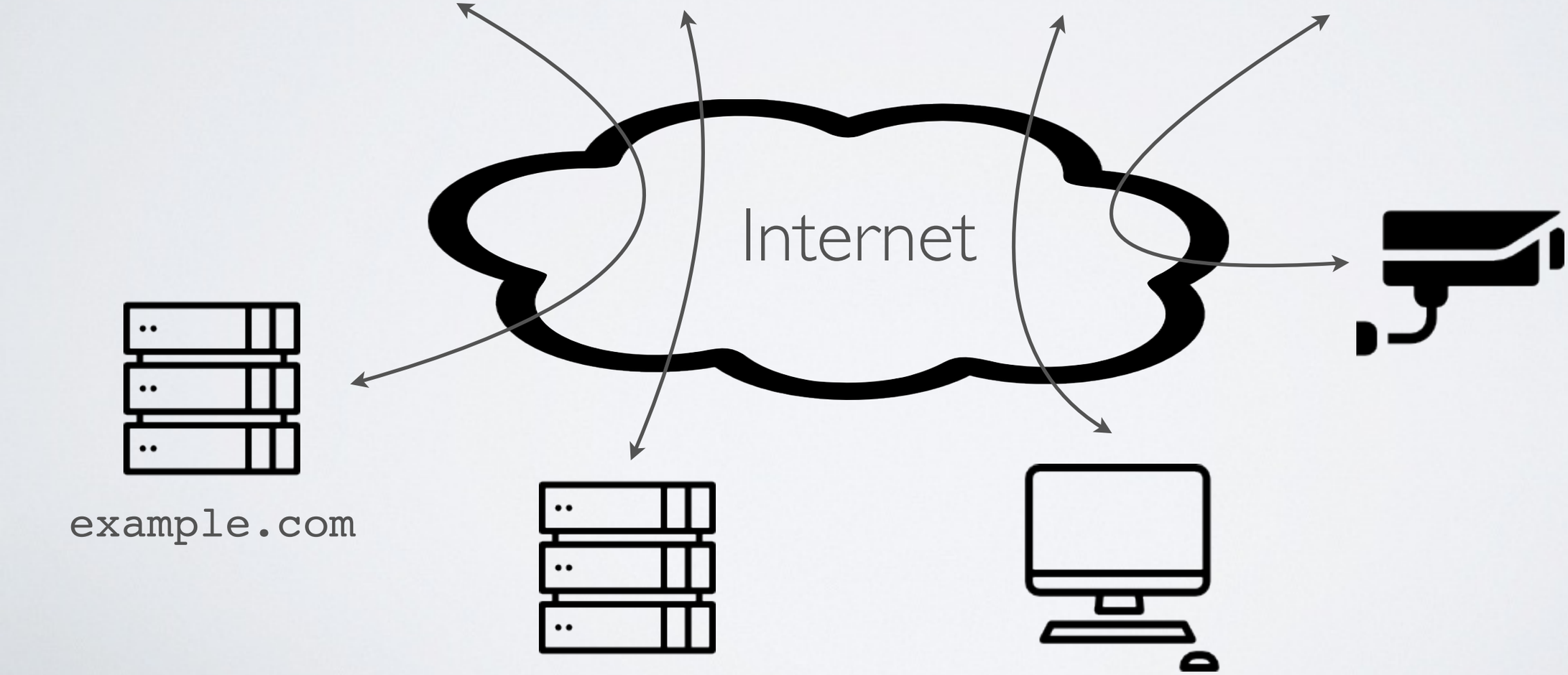
➡ (most) protocols are standards defined by the IETF - The Internet Engineering Task Force





Internet Applications

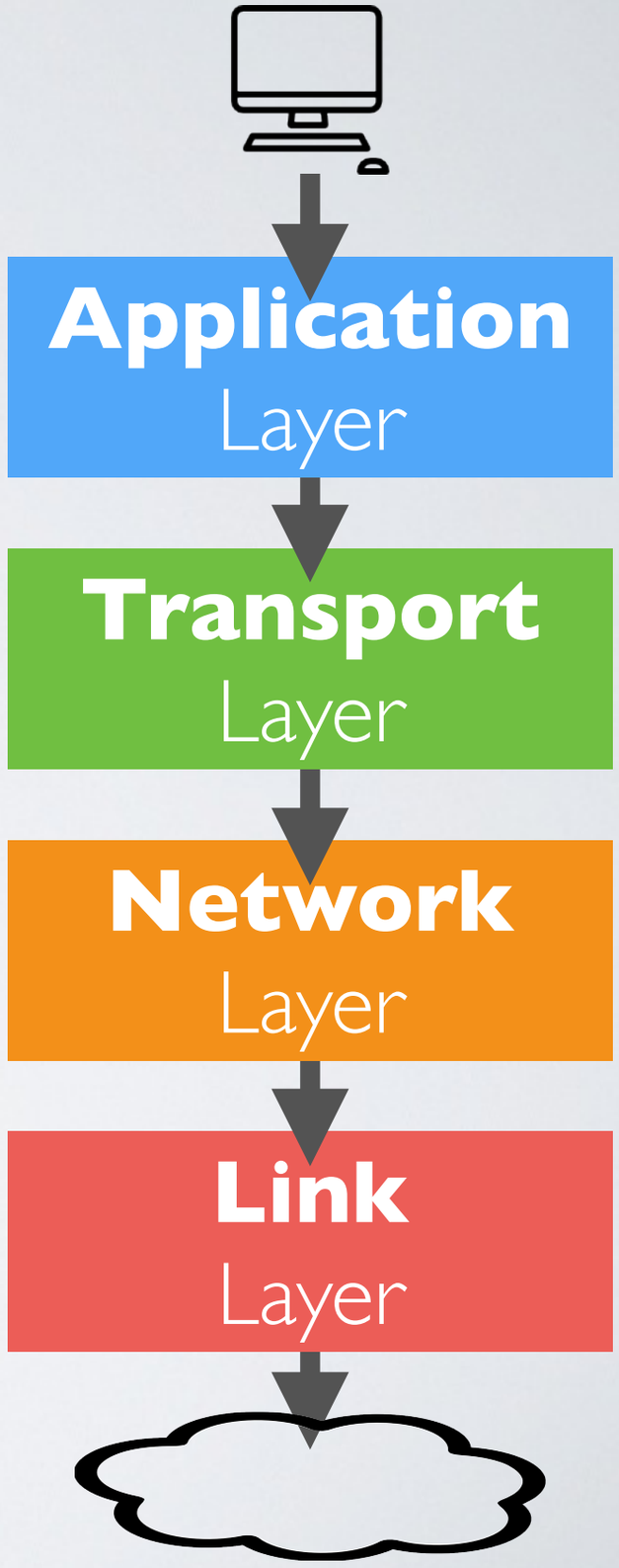
Web	Mail	File Exchange	Remote Shell	...
`http`	`smtp imap pop3 exchange`	`BiTorrent`	`ssh`	



The Internet Protocol Suite (a.k.a the network stack)

Protocols are built on top of each other as layers (modularity and encapsulation)

- How can two programs send messages to each other ?
- How to make sure that messages have been well transmitted ?
- How to route messages through the network ?
- How to encode messages to go through copper, fiber or air ?



The attacker is capable of …

confidentiality
integrity
availability

Scanning - survey the network and its hosts

Eavesdropping - read messages

Spoofing - forge illegitimate messages

DOS (Denial of Service) - disrupt the communications

➡ The attacker can target any layer in the network stack

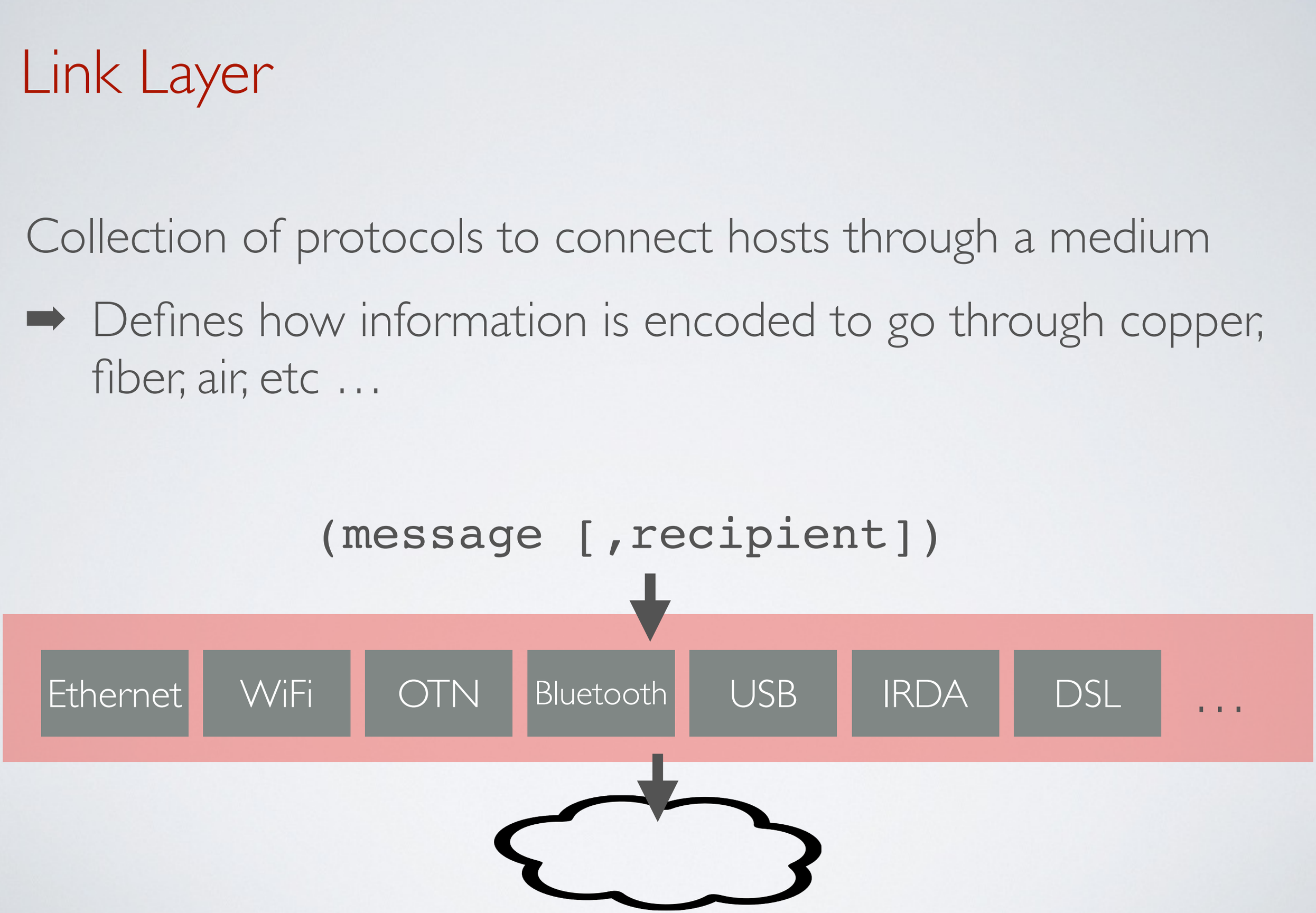
Link Layer

connecting machines together

Link Layer

Collection of protocols to connect hosts through a medium

➡ Defines how information is encoded to go through copper, fiber, air, etc …



Multiple Interfaces

A host can be connected to several hosts or networks through **multiple interfaces**

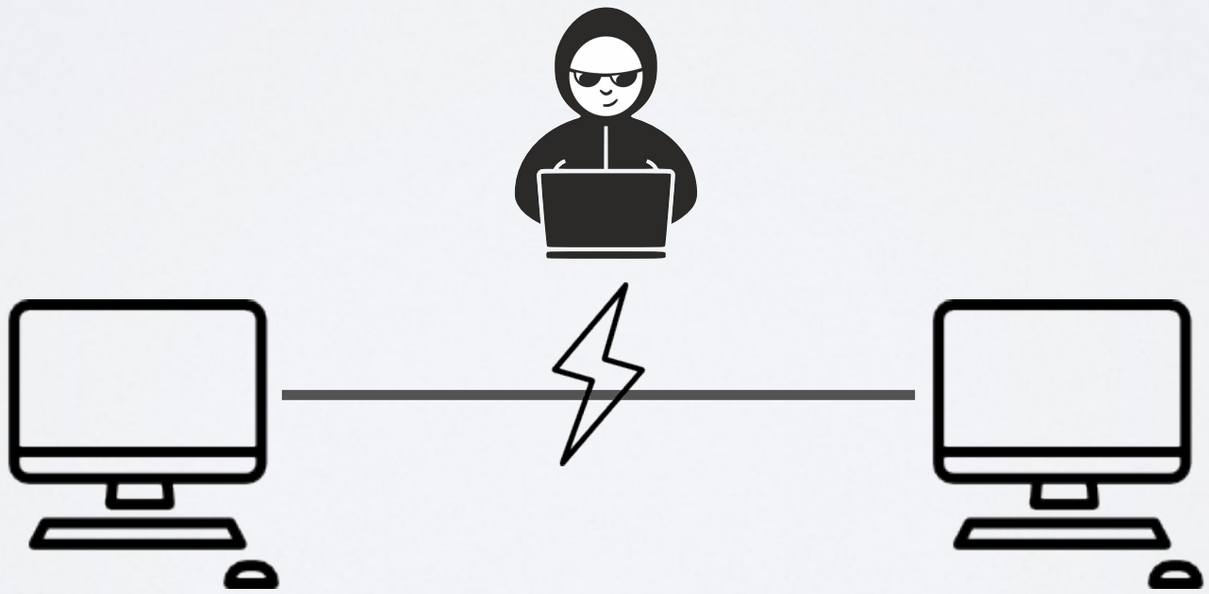
- Some are connected to a single host only (Point-to-Point)
- Others are connected to a entire network (BUS)

Point-to-Point Link

Only two hosts are connected at each end of the medium

e.g. OTN, IRDA, DSL …

➡ Harder for an attacker to intercept messages

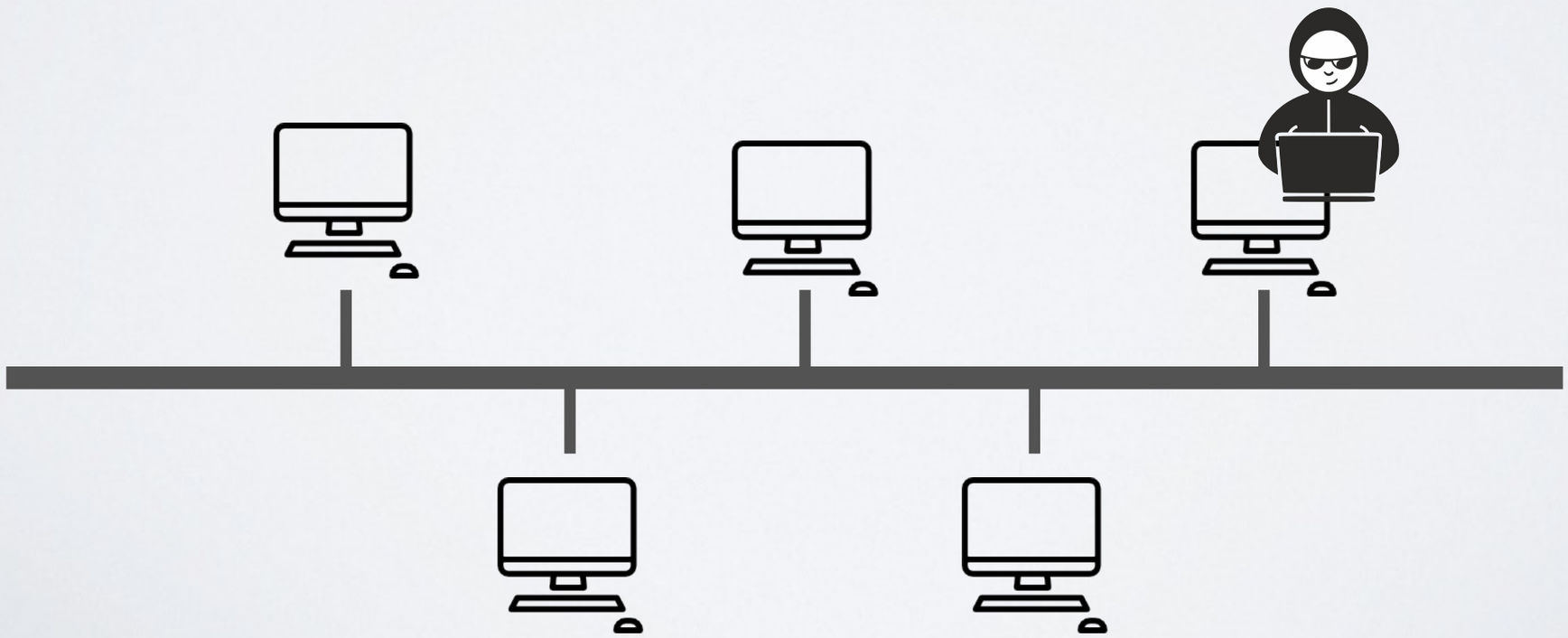


Bus Link (a.k.a LAN - Local Area Network)

Several hosts are connected to the same medium with a unique physical address called

e.g. Ethernet and WiFi uses MAC Media Access Control addresses

➡ Easier for the attacker to intercept messages since they are all broadcasted to the same medium



```
▾ Ethernet II, Src: VMware_30:da:bf (00:0c:29:30:da:bf), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
  ▾ Destination: VMware_c0:00:08 (00:50:56:c0:00:08)
      Address: VMware_c0:00:08 (00:50:56:c0:00:08)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: VMware_30:da:bf (00:0c:29:30:da:bf)
      Address: VMware_30:da:bf (00:0c:29:30:da:bf)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 192.168.22.128, Dst: 192.168.22.1

0000  00 50 56 c0 00 08 00 0c  29 30 da bf 08 00 45 08   ·PV·····  )0····E·

Source Hardware Address (eth.src), 6 bytes                Packets: 32 · Displayed: 32 (100.0%) · Dropped: 0 (

student@d27-vm:~/labs-review/packet-sniffing-starter$ cat /sys/class/net/ens33/address
00:0c:29:30:da:bf
```

confidentiality

Packet Sniffing over Ethernet or WiFi

- All messages are transmitted on the medium with the MAC address of the recipient

- Each network interface only picks messages that correspond to its MAC address

➡ An attacker can set its network interface in ***promiscuous mode*** to capture (sniff) all traffic

e.g. Wireshark

The WiFi Cactus @DefCon'19

source: theoutline.com

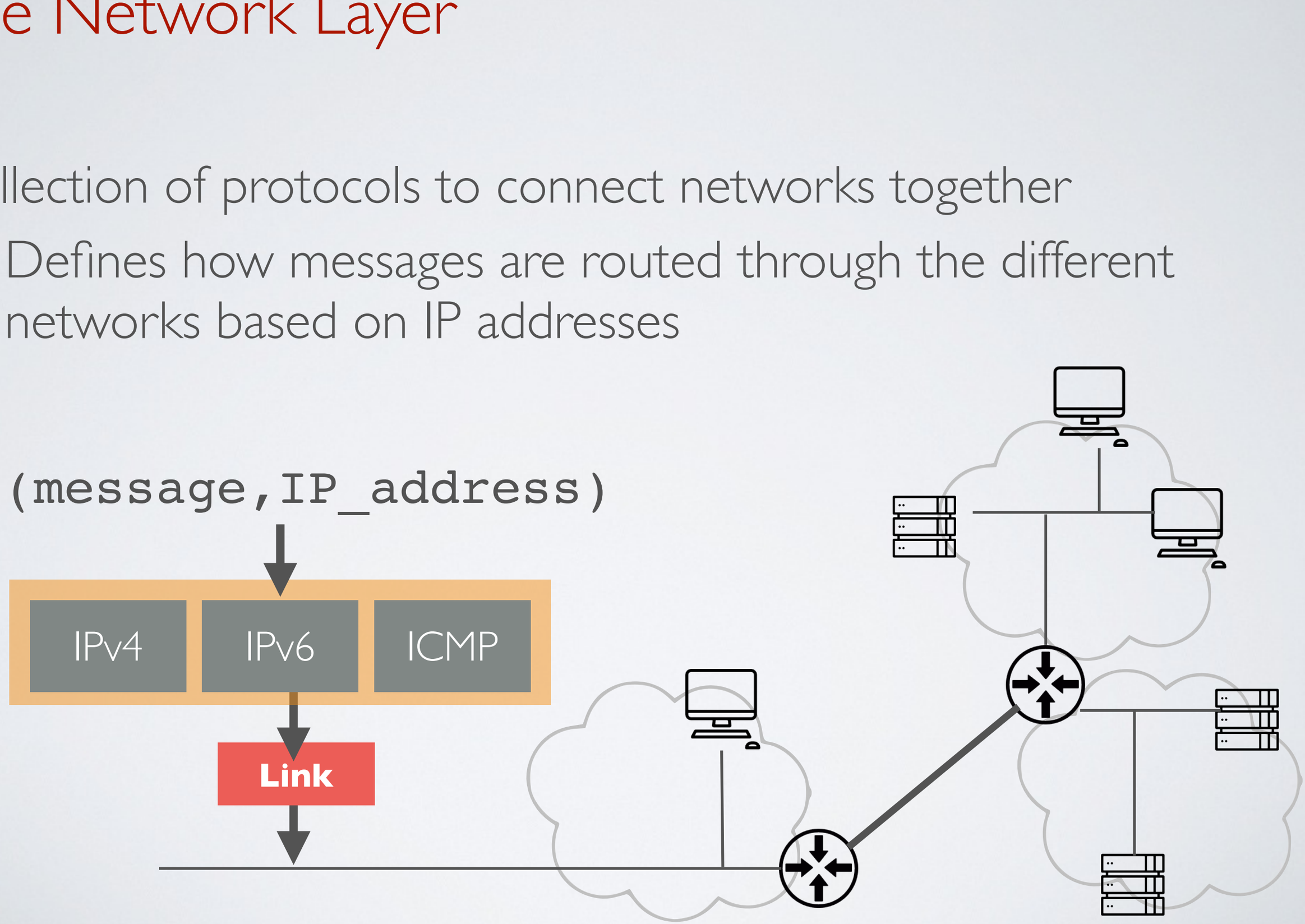
Network Layer

connecting networks together

The Network Layer

Collection of protocols to connect networks together

➡ Defines how messages are routed through the different networks based on IP addresses



IP - Internet Protocol

- Each message has the IP address of the issuer and recipient
- Routers route packet based on their routing table and a default route

➡ Best effort protocol

ICMP - Internet Control Message Protocol

Exchange information about the network

e.g. error reporting, congestion control, network reachability

➡ `ping, traceroute`

Host Discovery

~ confidentiality

By default, hosts answer to ICMP echo request messages

➡ An attacker scan an entire network to find IP addresses of active hosts

e.g. **`nmap`** (does that among other things)

IP Spoofing

integrity
availability

- Routers do not validate the source
- Receiver cannot tell that the source has been spoofed

➡ An attacker can generate raw IP packets with custom IP source fields

e.g. DOS (blackhole) and MITM attacks

ICMP ping of death (before 1997)

availability

Any host receiving a 64K ICMP payload would crash or reboot

- 64K bytes payload were assumed to be invalid by programmers
- An attacker could split a 64K payload, transmit it and would be reassembled by the receiver overflowing a buffer

Security Bulletin

Microsoft Security Bulletin MS10-009 - Critical

Vulnerabilities in Windows TCP/IP Could Allow Remote Code Execution (974145)

Published: February 09, 2010 | Updated: February 10, 2010

Version: 1.1

General Information

Executive Summary

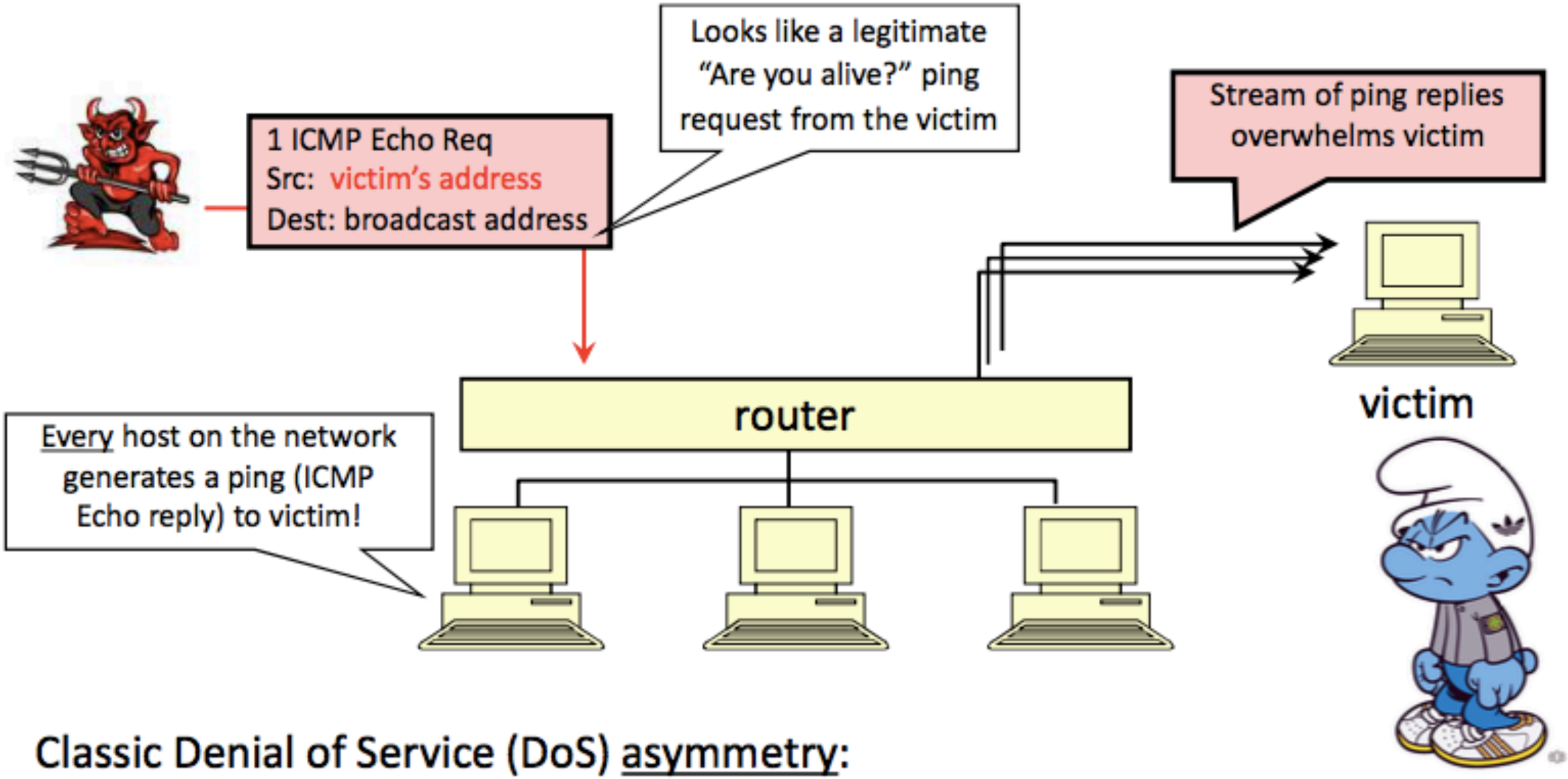
This security update resolves four privately reported vulnerabilities in Microsoft Windows. The most severe of these vulnerabilities could allow remote code execution if specially crafted packets are sent to a computer with IPv6 enabled. An attacker could try to exploit the vulnerability by creating specially crafted ICMPv6 packets and sending the packets to a system with IPv6 enabled. This vulnerability may only be exploited if the attacker is on-link.

ICMP Ping Flood

availability

- An attacker can overwhelm a host by sending multiples ICMP echo requests

ICMP Smurf Attack - an elaborated ping flood attack



Classic Denial of Service (DoS) asymmetry:
cheap for attacker, expensive for victim, due to protocol amplification

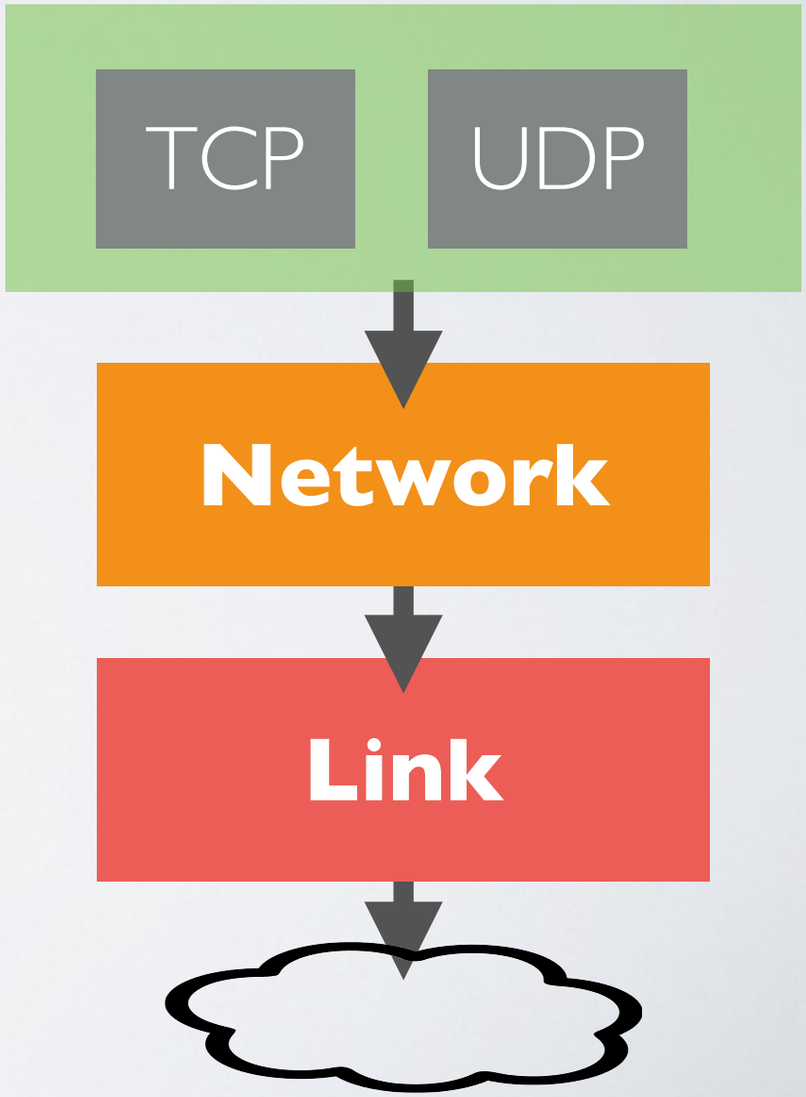
Transport Layer

end-to-end connection

The Transport Layer

Collection of protocols to ensure end-to-end connections

- ➡ Allows hosts to have multiple connections through **ports**
- ➡ Allows messages to be **fragmented** into small IP packets
- ➡ Make sure that all packets are received

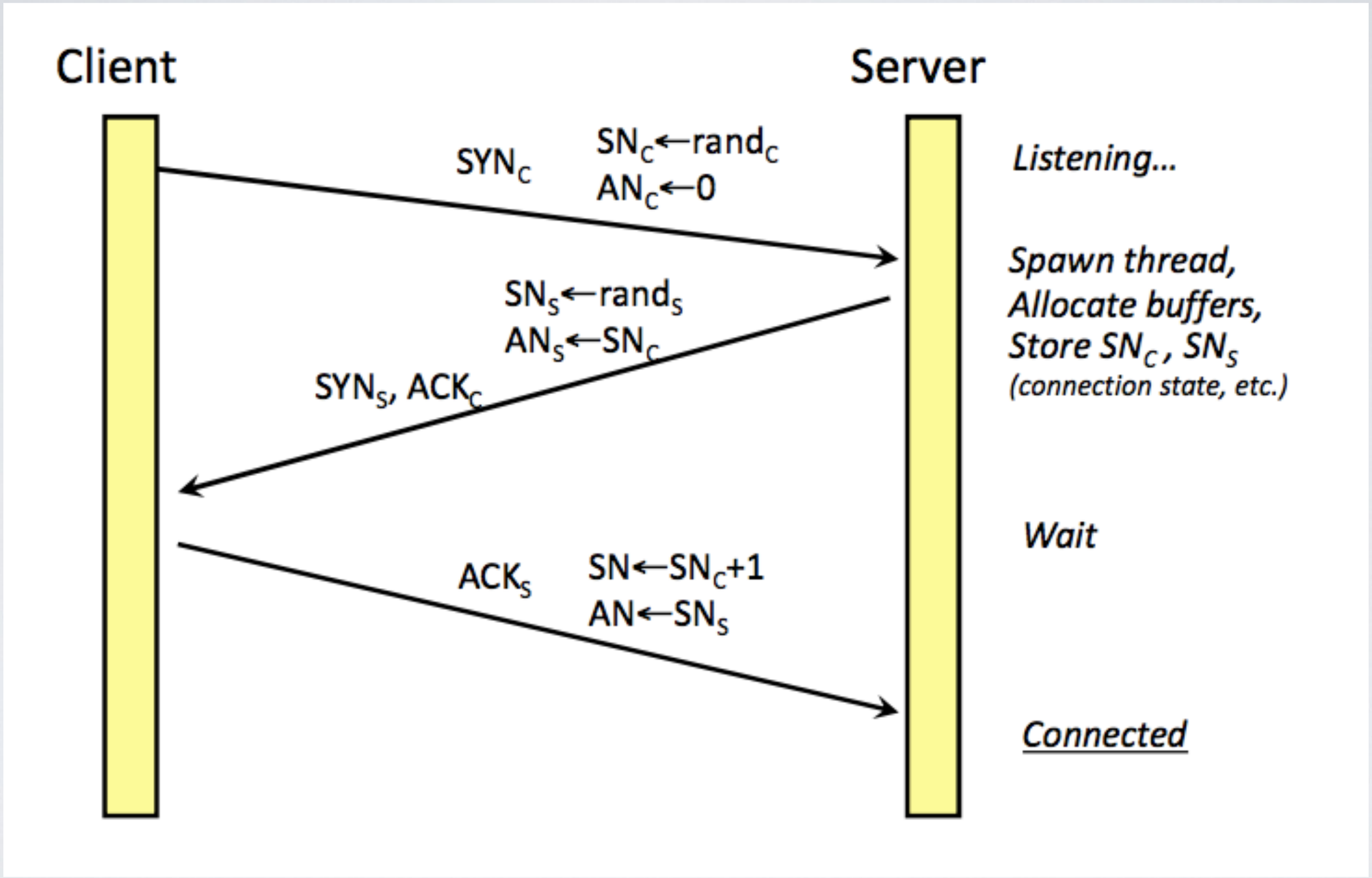


TCP - Transmission Control Protocol

- The sender divides data-stream into packets sequence number is attached to every packet
- The receiver checks for packets errors, reassembles packets in correct order to recreate stream
- ACK (acknowledgements) are sent when packets are well received and lost/corrupt packets are re-sent

➡ Connection state maintained on both ends

TCP "3-way" handshake




```
    1 0.000000000   192.168.23.1      192.168.23.128    TCP      66 60645 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
    2 0.000069486   192.168.23.128    192.168.23.1      TCP      66 8000 → 60645 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
    3 0.000758866   192.168.23.1      192.168.23.128    TCP      60 60645 → 8000 [ACK] Seq=1 Ack=1 Win=131328 Len=0
```

Port scanning

~ confidentiality

- Using the "3-way" handshake, an attacker can scan for all open ports for a given host

  e.g. **`nmap`**

```
… 549.4… 192.168.2… 192.168.2… TCP  66 51467 → 8001 [SYN] Seq=0 Win=64240 Len=0 MSS=…
… 549.4… 192.168.2… 192.168.2… TCP  54 8001 → 51467 [RST, ACK] Seq=1 Ack=1 Win=0 Len…
```

availability

TCP-syn flooding

Client

Server

SYN_{C1}

SYN_{C2}

SYN_{C3}

SYN_{C4}

SYN_{C5}

Listening…

Spawn a new thread, Allocate buffers

… and more

… and more

… and more

… and more

… and more

Hold on, won't 3-way handshake cause Client to receive it's own "syn flood" as server responds to all it's SYN's with SYN-ACK's?

Note asymmetric effort between attacker client and victim server

availability

TCP Connection Reset (DOS)

Each TCP connection (i.e each port) has an associated state sequence number

- An attacker can guess (sniff) the current sequence number for an existing connection and send packet with reset flag set, which will close the connection

UDP - User Datagram Protocol

UDP is a connectionless transport-layer protocol

- ➡ No acknowledgement, no flow control, no message continuation, no reliability guarantees

e.g. media streaming (VoIP, video broadcasting)

e.g modern protocols (HTTP 3)

availability

UDP Flood

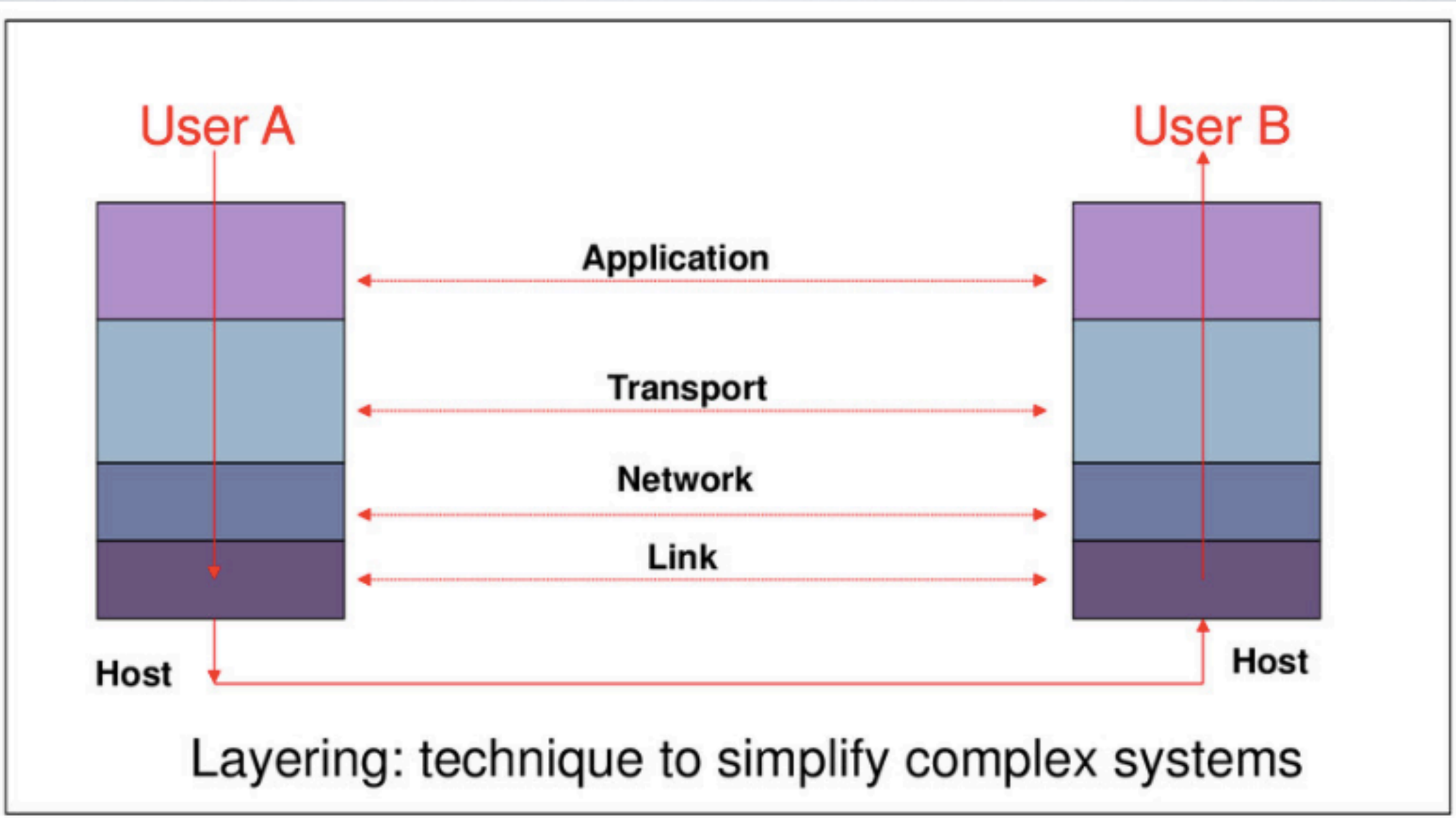
When a UDP packet is received on a non-opened port, the host replies with an `ICMP Destination Unreachable`

➡ An attacker can send a large number of UDP packets to all ports of a target host

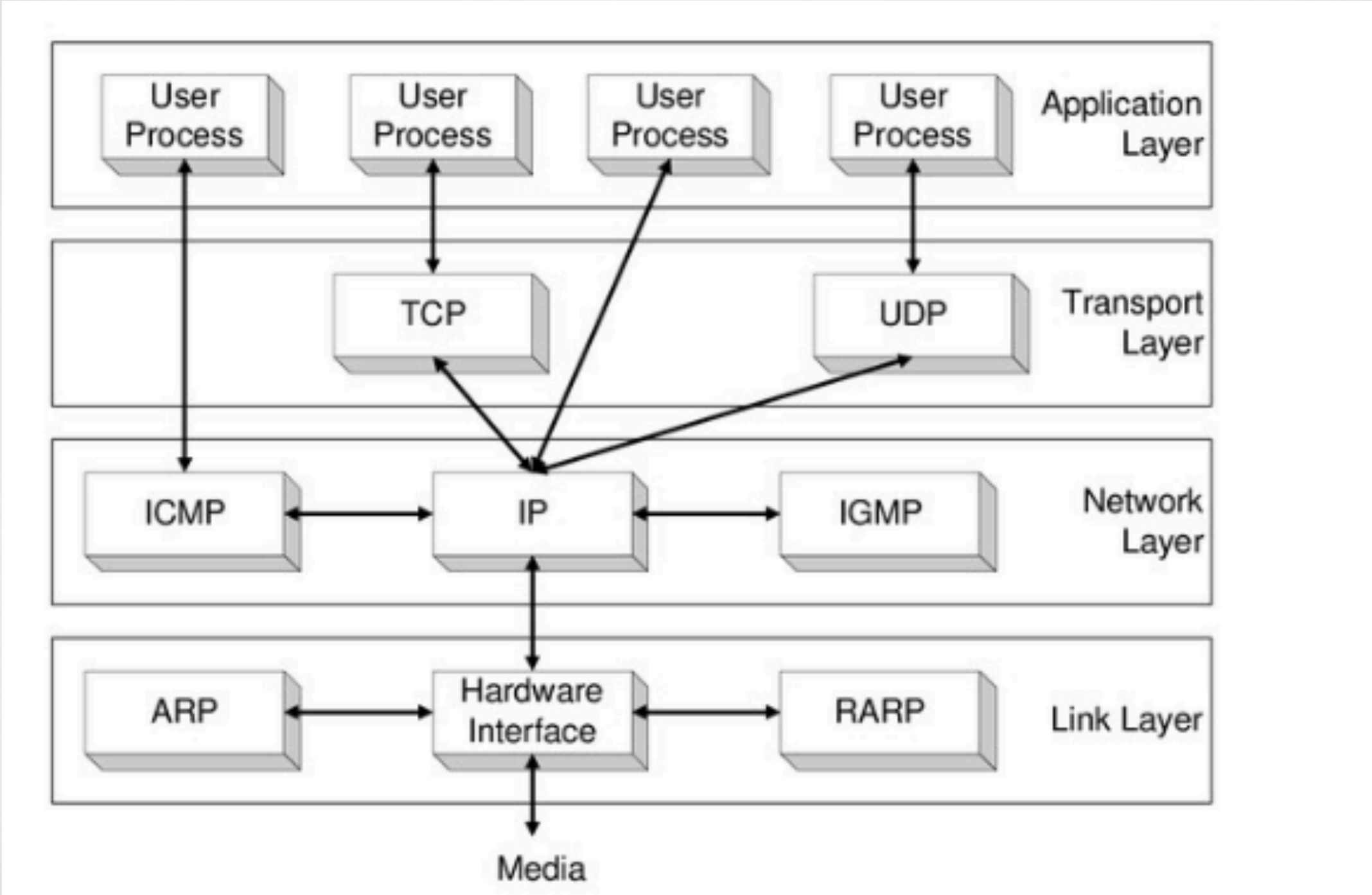
e.g *Low Orbit Ion Cannon*

The TCP/IP Stack

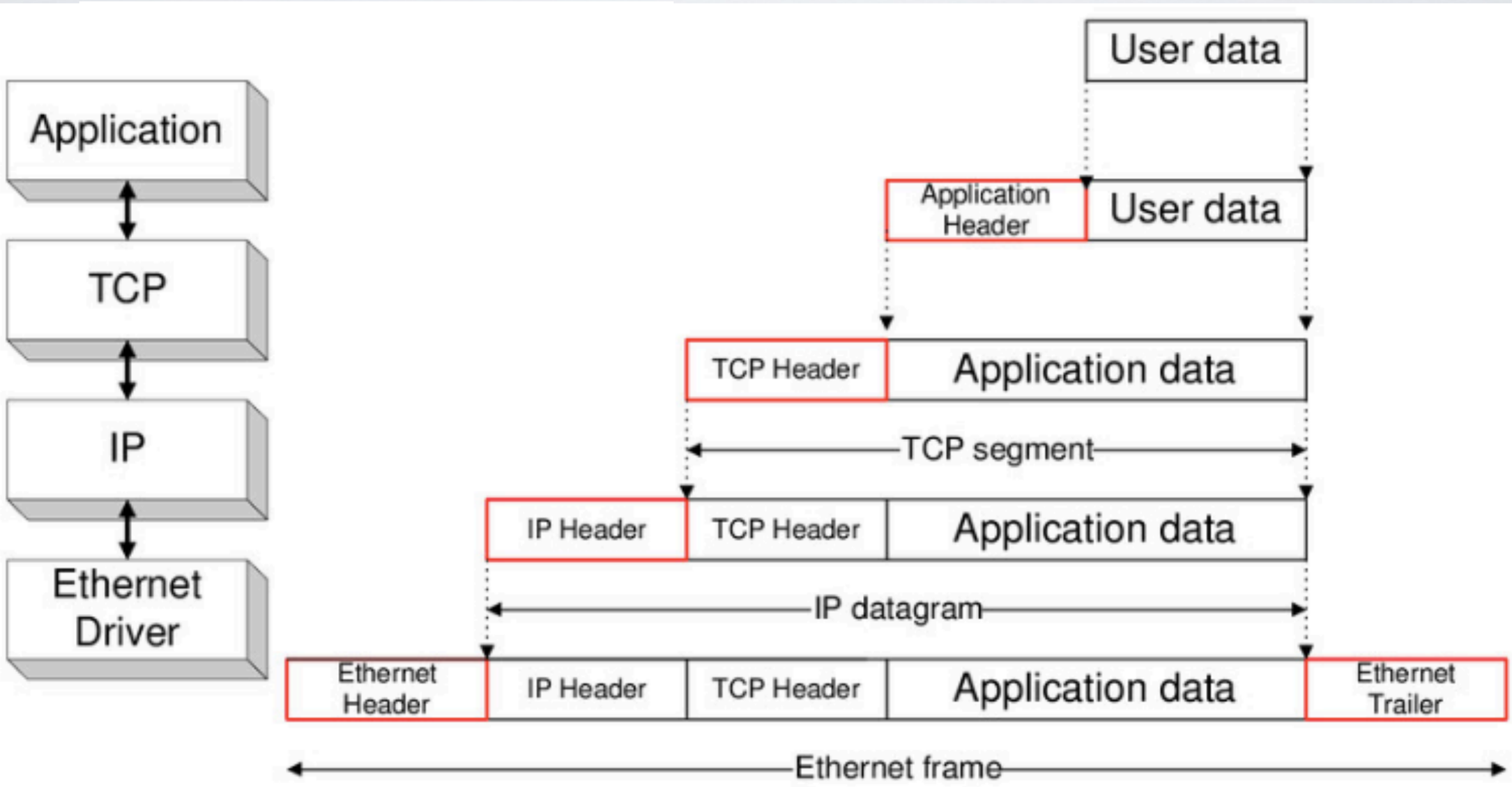
Layering


User A
User B
Application
Transport
Network
Link
Host
Host
Layering: technique to simplify complex systems

TCP/IP


User Process
User Process
User Process
User Process
Application Layer
TCP
UDP
Transport Layer
ICMP
IP
IGMP
Network Layer
ARP
Hardware Interface
RARP
Link Layer
Media

Data encapsulation

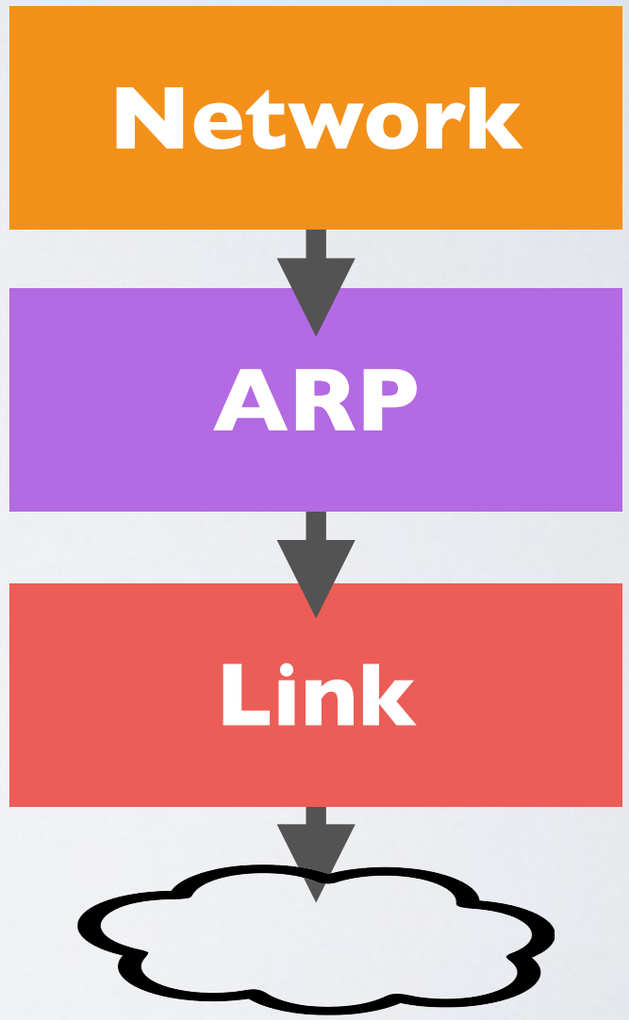


Special Protocols

ARP - Address Resolution Protocol

Each host has an ARP table that contains mapping between MAC and IP addresses

➡ Host broadcasts their own IP address and MAC address to others to build their ARP table



ARP - Address Resolution Protocol

```
    30 5.018678      02:42:e7:08:96:52    02:42:0a:00:00:02    ARP          42 Who has 10.0.0.2? Tell 10.0.0.1
    31 5.018686      02:42:0a:00:00:02    02:42:e7:08:96:52    ARP          42 10.0.0.2 is at 02:42:0a:00:00:02
```

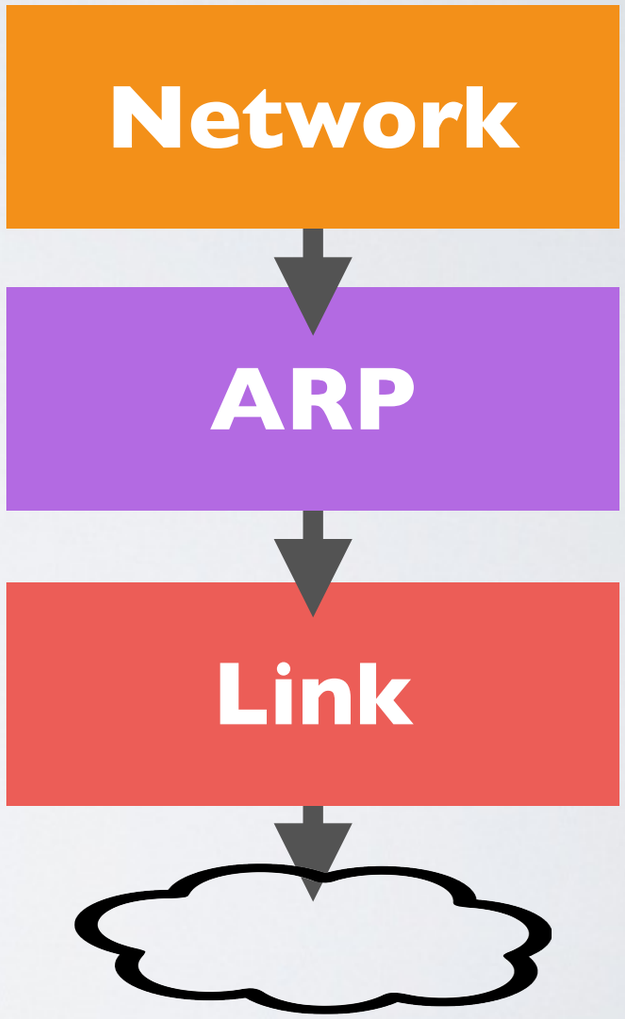
```
Frame 31: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
Ethernet II, Src: 02:42:0a:00:00:02 (02:42:0a:00:00:02), Dst: 02:42:e7:08:96:52 (02:42:e7:08:96:52)
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 02:42:0a:00:00:02 (02:42:0a:00:00:02)
    Sender IP address: 10.0.0.2
    Target MAC address: 02:42:e7:08:96:52 (02:42:e7:08:96:52)
    Target IP address: 10.0.0.1
```

```
Frame 82: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface ens33, id 0
Ethernet II, Src: VMware_30:da:bf (00:0c:29:30:da:bf), Dst: VMware_e7:52:23 (00:50:56:e7:52:23)
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: VMware_30:da:bf (00:0c:29:30:da:bf)
    Sender IP address: 192.168.23.128
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.23.2
```

```
0010  08 00 06 04 00 01 00 0c  29 30 da bf c0 a8 17 80   ........ )0......
```

Sender IP address (arp.src.proto_ipv4), 4 bytes Packets: 299 · Displayed: 299 (100.0%) Profile: Default

```
student@d27-vm:~/labs-review/packet-sniffing-starter$ ip --brief address show
lo               UNKNOWN        127.0.0.1/8 ::1/128
ens33            UP             192.168.23.128/24 fe80::7fc8:9a37:c4e:c01b/64
docker0          DOWN           172.17.0.1/16
student@d27-vm:~/labs-review/packet-sniffing-starter$ arp -i ens33
Address                  HWtype  HWaddress           Flags Mask            Iface
169.254.169.254                  (incomplete)                              ens33
192.168.23.254           ether   00:50:56:e5:4f:6c   C                     ens33
_gateway                 ether   00:50:56:e7:52:23   C                     ens33
192.168.23.1             ether   00:50:56:c0:00:08   C                     ens33
student@d27-vm:~/labs-review/packet-sniffing-starter$ ip neigh show
169.254.169.254 dev ens33  FAILED
192.168.23.254 dev ens33 lladdr 00:50:56:e5:4f:6c STALE
192.168.23.2 dev ens33 lladdr 00:50:56:e7:52:23 REACHABLE
192.168.23.1 dev ens33 lladdr 00:50:56:c0:00:08 REACHABLE
student@d27-vm:~/labs-review/packet-sniffing-starter$
```



ARP Cache Poisoning

integrity
availability

➡ An attacker can broadcast fake IP-MAC mappings to the other hosts on the network

e.g. DOS and MITM attacks

BGP - Border Gateway Protocol (a.k.a routing)

Each router has a routing table to IP messages

BGP is the protocol for establishing routes

➡ Routers advertise the best route to other nearby routers depending on the state of the network

Route hijacking

confidentiality

availability

➡ An attacker can advertise fake routes

e.g. DOS (blackhole) and MITM attacks

Pakistan's Accidental YouTube Re-Routing Exposes Trust Flaw in Net

A Pakistan ISP that was ordered to censor YouTube accidentally managed to take down the video site around the world for several hours Sunday.

Source: Wired

DNS - Domain Name Server

Internet applications relies on canonical hostname rather than IP addresses

DNS servers translates domain names into IP addresses

➡ DNS servers form a distributed directory service by exchanging information about domains and other DNS servers

DNS Cache Poisoning

confidentiality

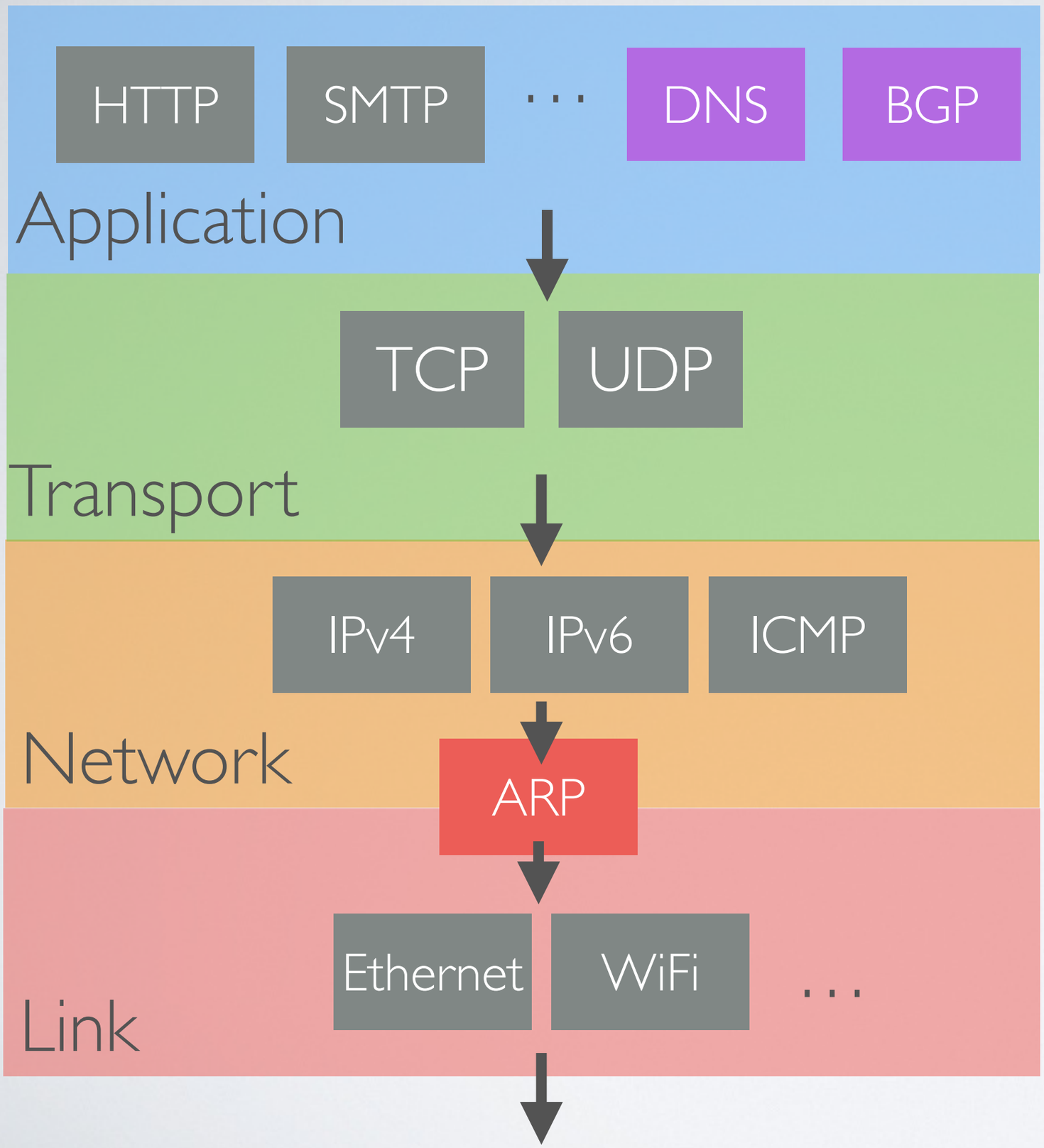
availability

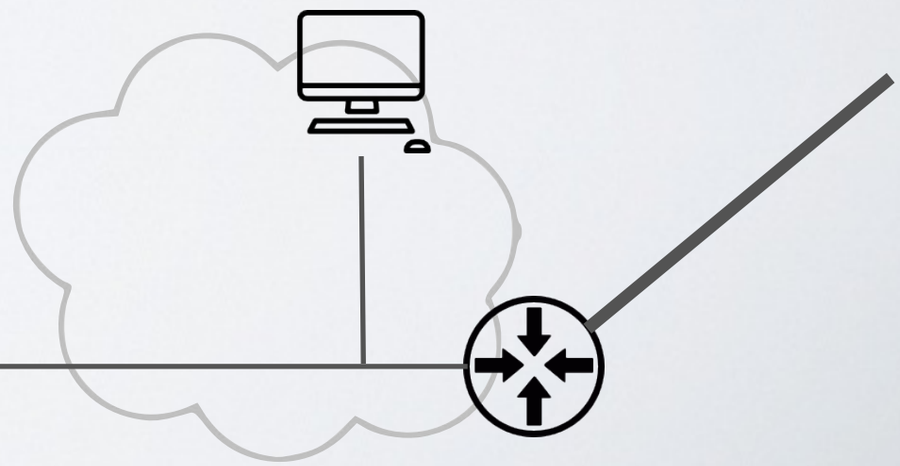
➡ An attacker can advertise fake DNS information

e.g. DOS and MITM attacks

Summary

The Protocol Stack





The attacker is capable of …

confidentiality
integrity
availability

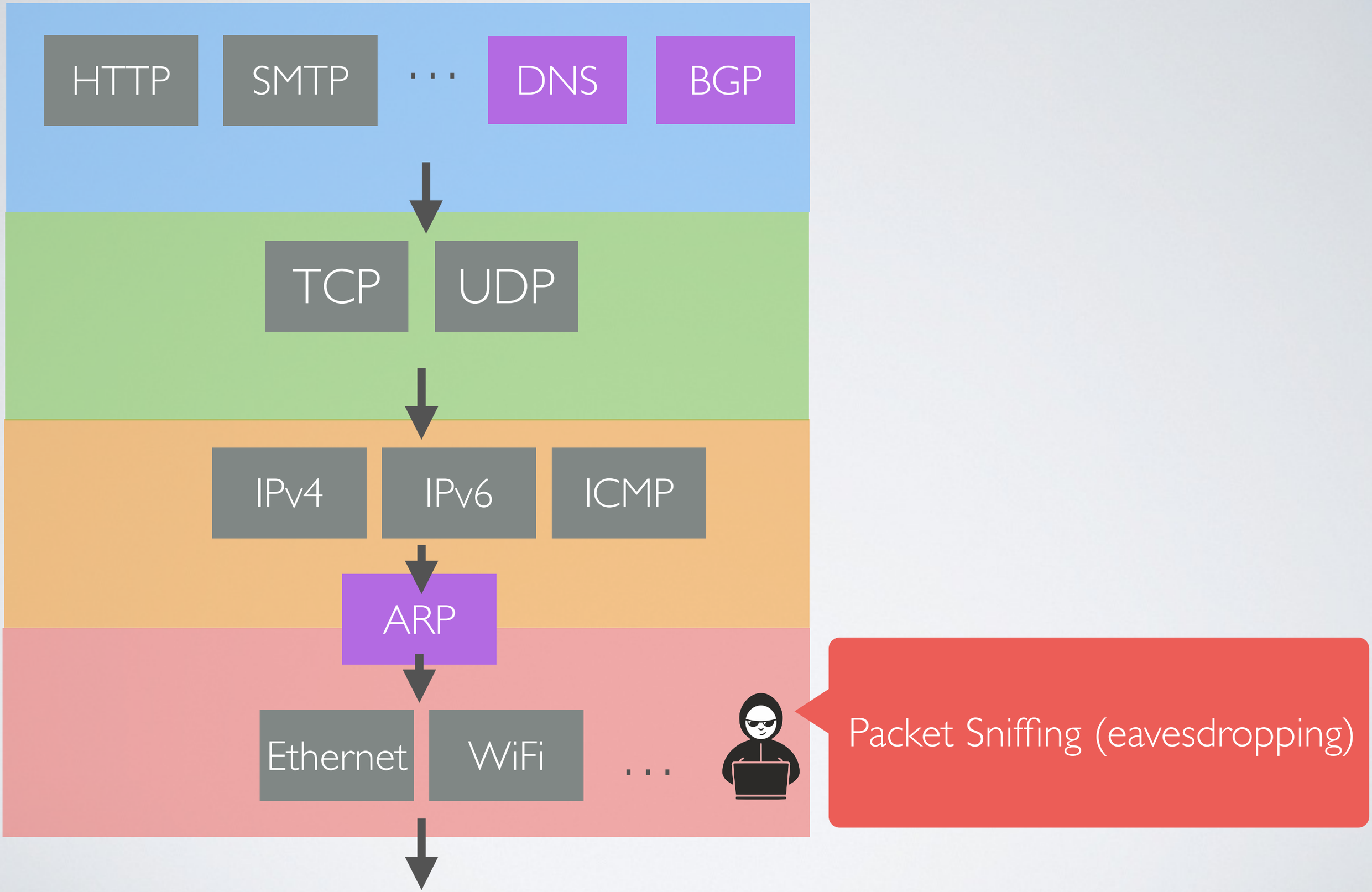
Scanning - survey the network and its hosts

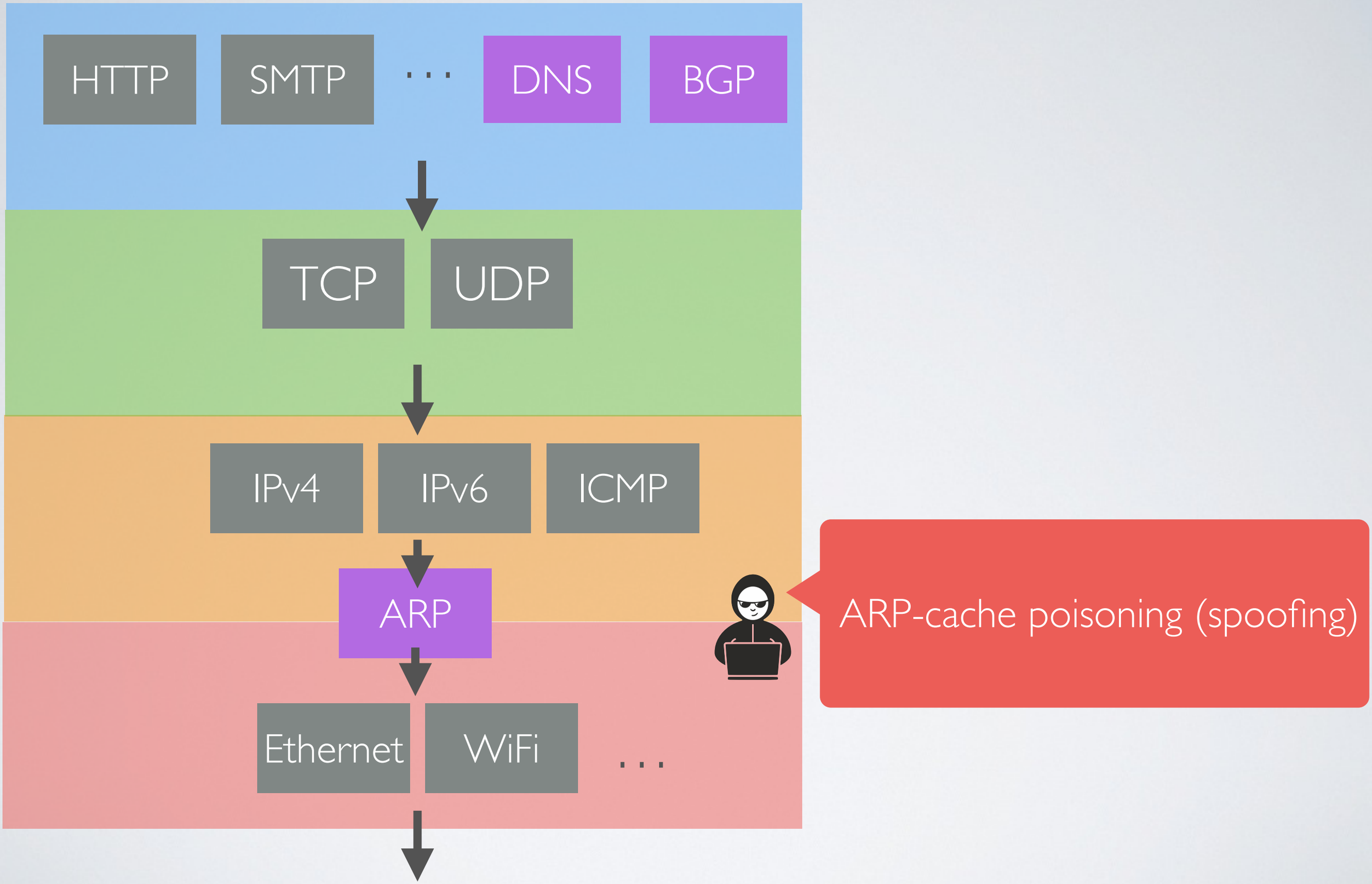
Eavesdropping - read messages

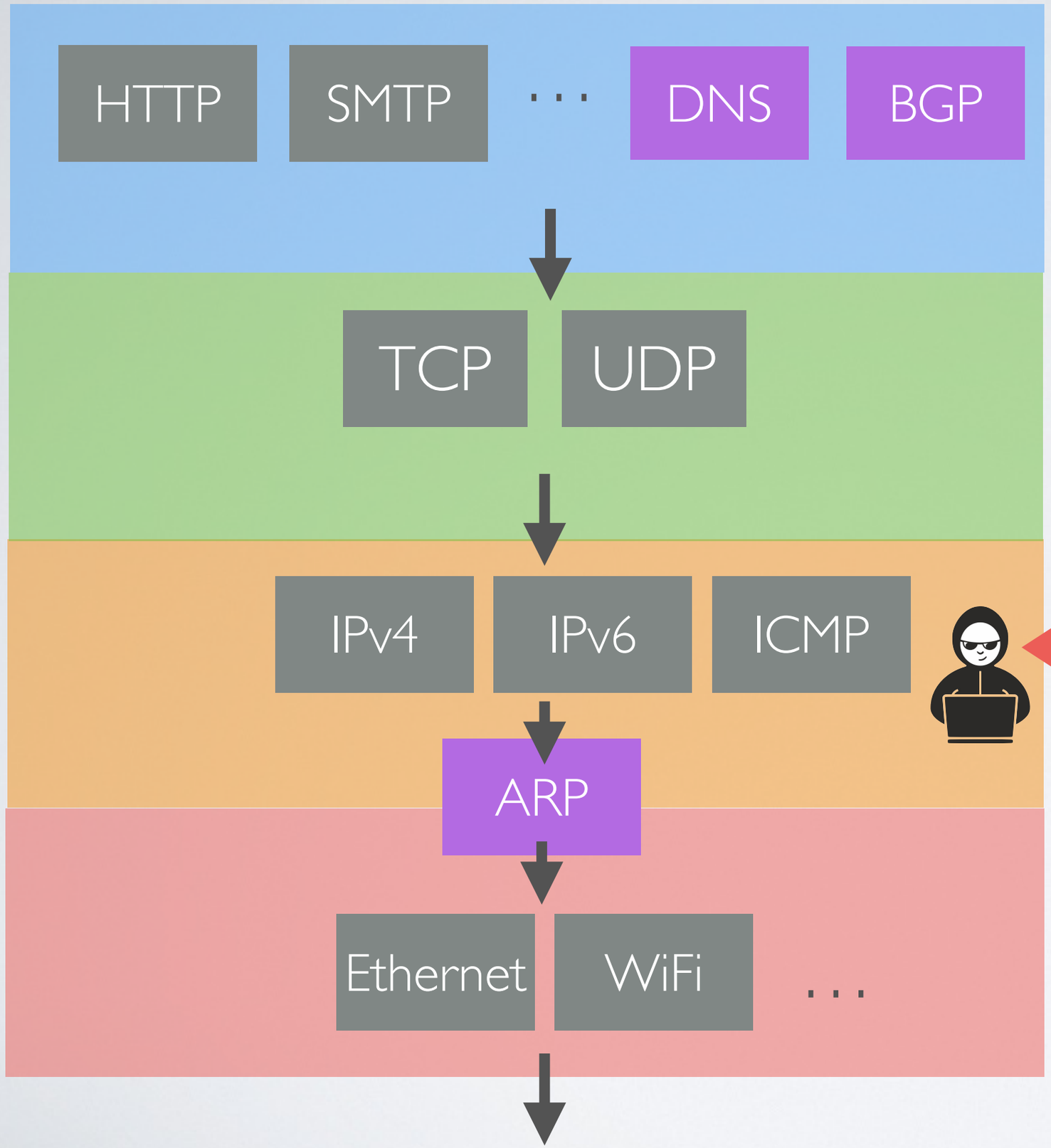
Spoofing - forge illegitimate messages

DOS (Denial of Service) - disrupt the communications

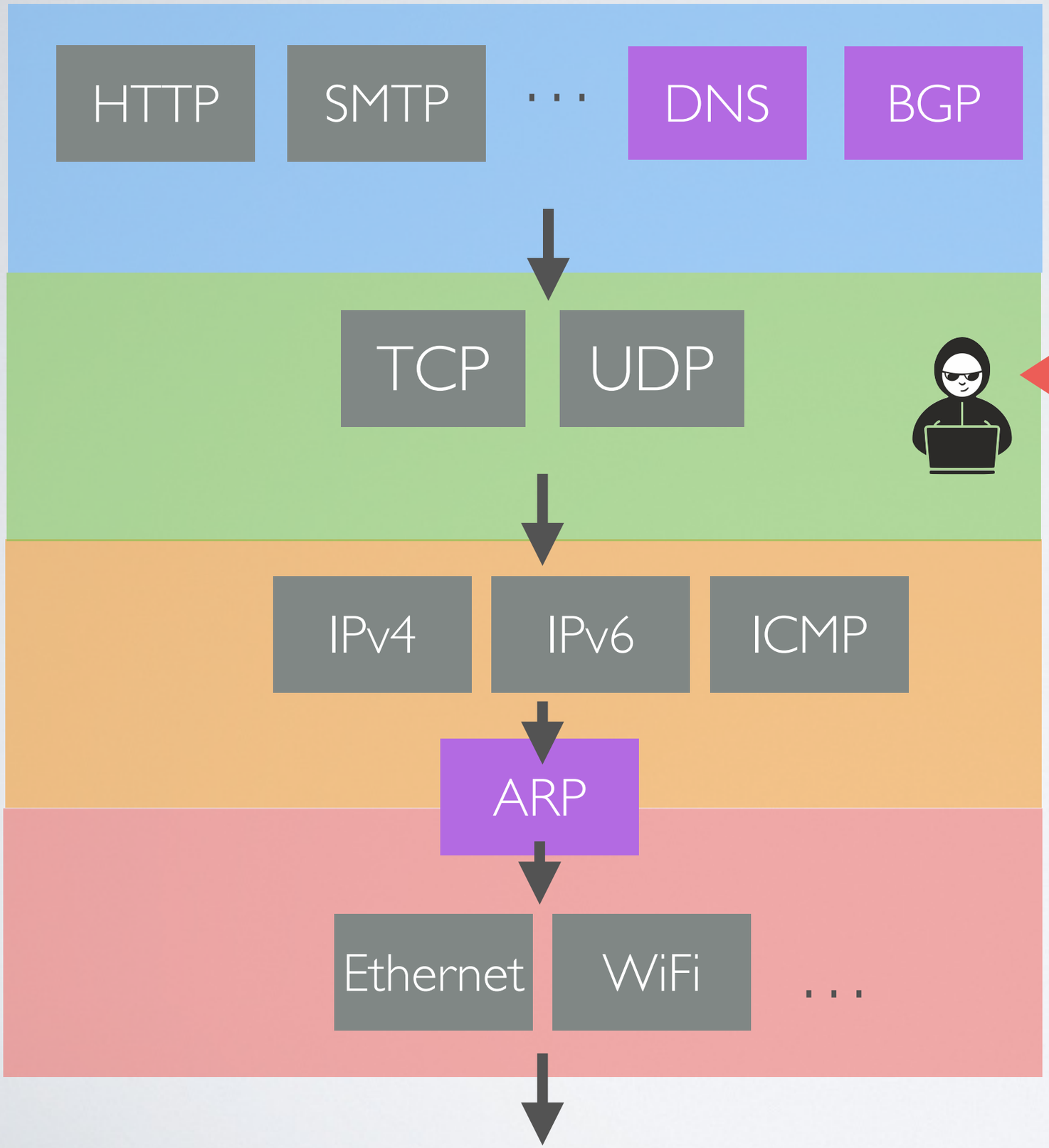
➡ The attacker can target any layer in the network stack


HTTP
SMTP
. . .
DNS
BGP
TCP
UDP
IPv4
IPv6
ICMP
ARP
Ethernet
WiFi
. . .
Packet Sniffing (eavesdropping)

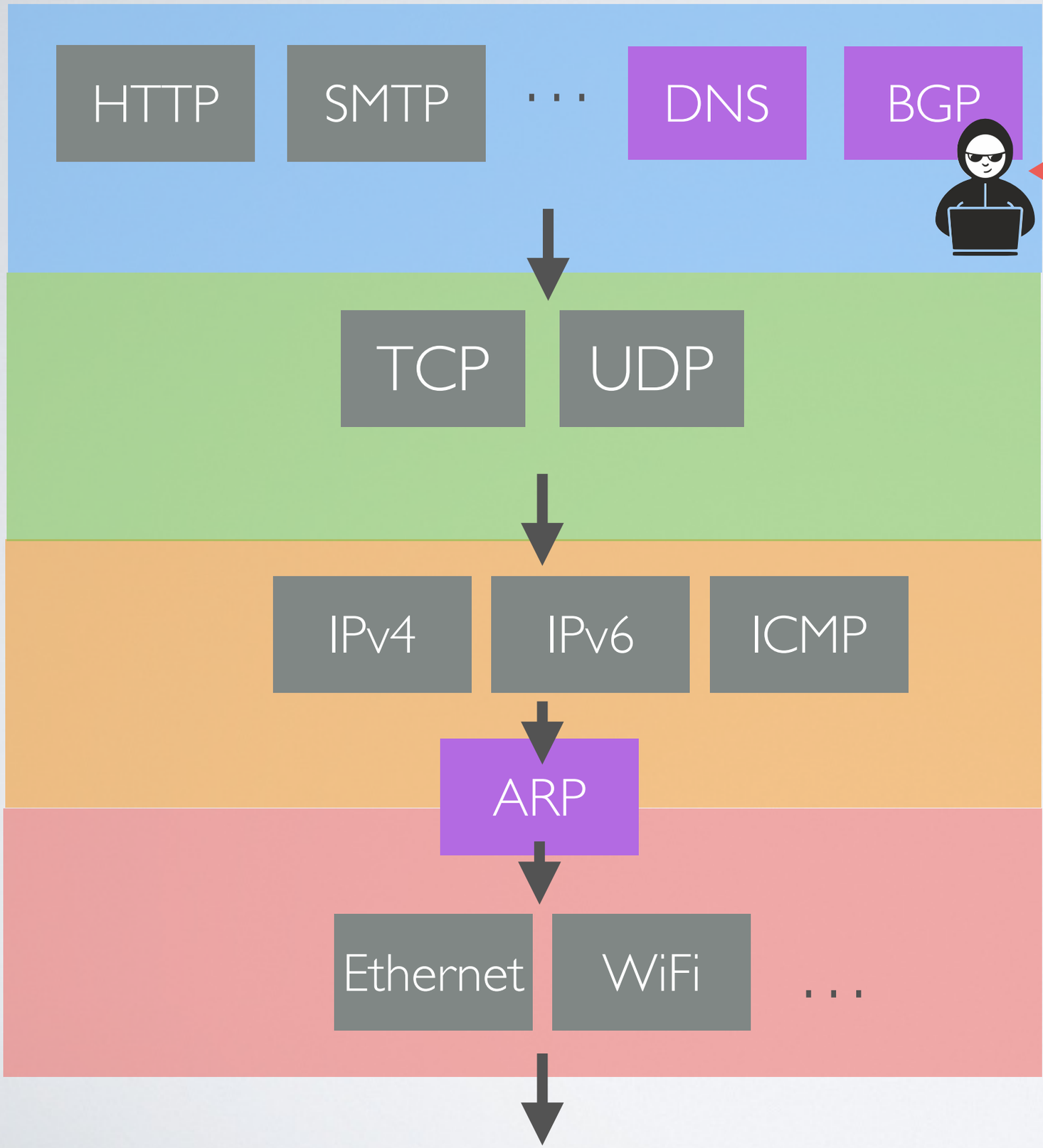

HTTP
SMTP
. . .
DNS
BGP
TCP
UDP
IPv4
IPv6
ICMP
ARP
Ethernet
WiFi
. . .
ARP-cache poisoning (spoofing)



- Host discovery (scanning)
- IP forgery (spoofing)
- ICMP Ping flooding (DOS)



- Port scanning (scanning)
- TCP forgery (spoofing, DOS)
- TCP-syn flooding (DOS)
- UDP flooding (DOS)



- Route Hijacking (spoofing, DOS)
- DNS-cache poisoning (spoofing, DOS)