Networking Basics

- CEG 6430/4430 Cyber Network Security
- Junjie Zhang
- junjie.zhang@wright.edu
- Wright State University

Network

A network is a set of protocols, software, and hardware used to exchange information among different systems.

Various Types of Networks

- Telephone Networks
- Cellular Networks
- Satellite Networks
- The Internet
- And more...

The Internet

- The Internet is a global-scale network that
  - uses the Internet protocol suite (TCP/IP) to communicate between networks and devices
  - offers general-purpose service
  - has billions of users worldwide
  - consists of millions of "smaller" networks
  - supports numerous types of services (Web, Email, VoIP……)

Rooted in the Internet protocol suite (TCP/IP)

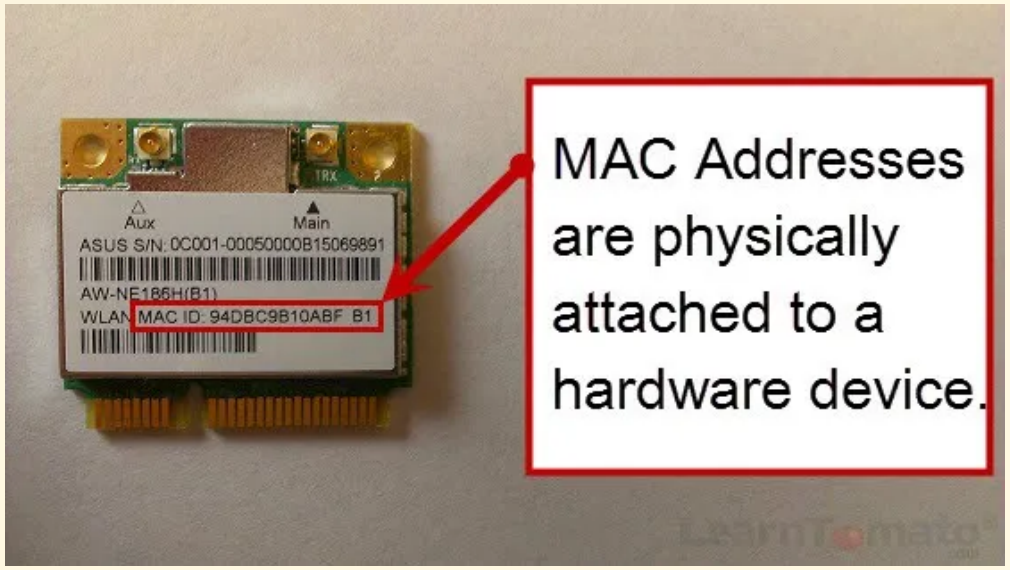
- The Internet protocol suite is commonly known as TCP/IP
  - Provide end-to-end communications
  - The Advanced Research Projects Agency Network (ARPANET) was the first wide-area packet-switched network with distributed control and one of the first networks to implement the TCP/IP protocol suite.

Internet Addresses

- Media Access Control (MAC) Address
- The Internet Protocol (IP) Address

MAC Address

- Physical Address
- Assigned to a NIC by the manufacturer
- Not expected to be changed
- Used for communication within one hop
- 48 bits



IP Address

- Logical Address
- Assigned to a NIC by the network administrator manually or automatically
- Expected to be changed
- Used for communication over multiple hops
- 32 bits

Your Addresses

- Use `ifconfig` in Linux or `ipconfig` in Windows.

```
[jzhang@DESKTOP-DSVPHPI CEG-4430-6430-Cyber-Network-Security]$ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.31.104.107  netmask 255.255.240.0  broadcast 172.31.111.255
        inet6 fe80::215:5dff:fee5:1691  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:e5:16:91
        ...
```

- IP address (in octal format): 172.31.104.107
- MAC Address (in hex format): 00:15:5d:e5:16:91

IP Address

- Usually in octal format (more human-readable)
  - 172.31.104.107
- Essentially a sequence of 32 bits

```
>>> import ipaddress
>>> print(bin(int(ipaddress.IPv4Address('172.31.104.107'))))
0b10101100000111110110100001101011
```

Packet

[Header|Payload]

- Packet Header
  - Address and other control information
- Packet Payload or Data
  - Data to be carried by this packet
  - Can be [Header|Payload] of another layer.

[Ether_Header|IP_Header|TCP_Header|HTTP_Header|HTTP_Data]

Hop-to-Hop Communication - Only Using MAC

- Correct MAC addresses are necessary and sufficient
- Host A and B are directly connected (i.e., using an Ethernet cable)
  - Host A with MAC: AAAA
  - Host B with MAC: BBBB
- A Packet from A to B
  - [Src_MAC: AAAA][Dst_MAC: BBBB][Data]

Hop-to-Hop Communication - With IP Involved

- It is inflexible to *only* use MAC addresses
  - If you have a new box for host B, and thereby a new NIC and new MAC of B, you will reconfigure Host A.
- It is more reasonable to use IP to represent Host B.
  - Host B uses the same IP address.
  - But now you will need a protocol to translate B's IP address into B's new MAC address.

ARP

The Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address.

A Running Example of ARP

Host A attempts to find B's MAC given B's IP.

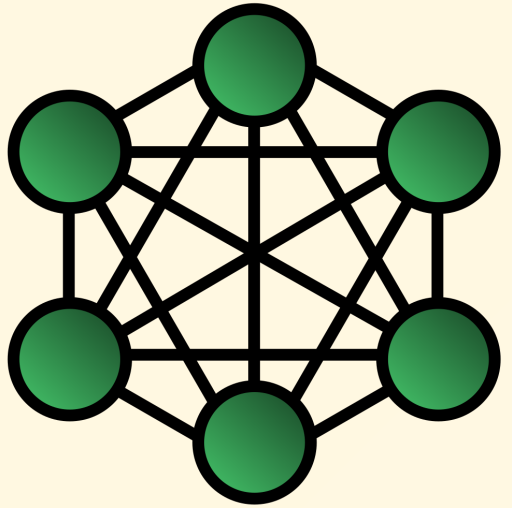
- Host A: 192.168.1.5 (IP Addr), AAAA (MAC Addr)
- Host B: 192.168.1.6 (IP Addr), BBBB (MAC Addr)

Packets Exchanged

- AAAA -> FF:FF:FF:FF:FF:FF: who has the IP address 192.168.1.6?
- BBBB -> AAAA: BBBB has the IP address 192.168.1.6.

Forming The "Smallest" Network

A Network Segment or a Local Area Network (LAN)



- What MAC Addrs will you use? - No Choice.
- How about IPs? You will have the flexibility (strictly speaking, some flexibility) to decide IP addresses to form your network.
- How do you interconnect these hosts? It does not scale up to directly interconnect each pair of hosts using Ethernet Cables.

Forming The "Smallest" Network

Then what IP addresses should you use?

- A subnetwork or subnet is a logical subdivision of an IP network.
  - A subnetwork is defined by a network prefix.
    - Example 1: 200.100.10.0/24
    - Example 2: 200.100.10.0/25
- Hosts in the same subnetwork should have the same network prefix.

Forming The "Smallest" Network

On your host, you can derive the network prefix using your `IP` and the `netmask`, which is a sequence of bits (e.g., 32 bits for IPv4).

```
[jzhang@DESKTOP-DSVPHPI CEG-4430-6430-Cyber-Network-Security]$ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.31.104.107  netmask 255.255.240.0  broadcast 172.31.111.255
        ...
```

- 255.255.240.0 -> /20
- 172.31.104.107 AND 255.255.240.0 -> 172.31.96.0
- Prefix: 172.31.96.0/20

Forming The "Smallest" Network

- Practically, Host A can use its own IP and its own netmask to check whether another IP (e.g., `anotherIP`) is inside the same network segment (or A and B are *directly* connected).

```
def is_same_network(anotherIP):
    if (self.ip & self.netmask) == (anotherIP & self.netmask):
        return True
    else:
        return False
```

Forming The "Smallest" Network

IP addresses are solved.

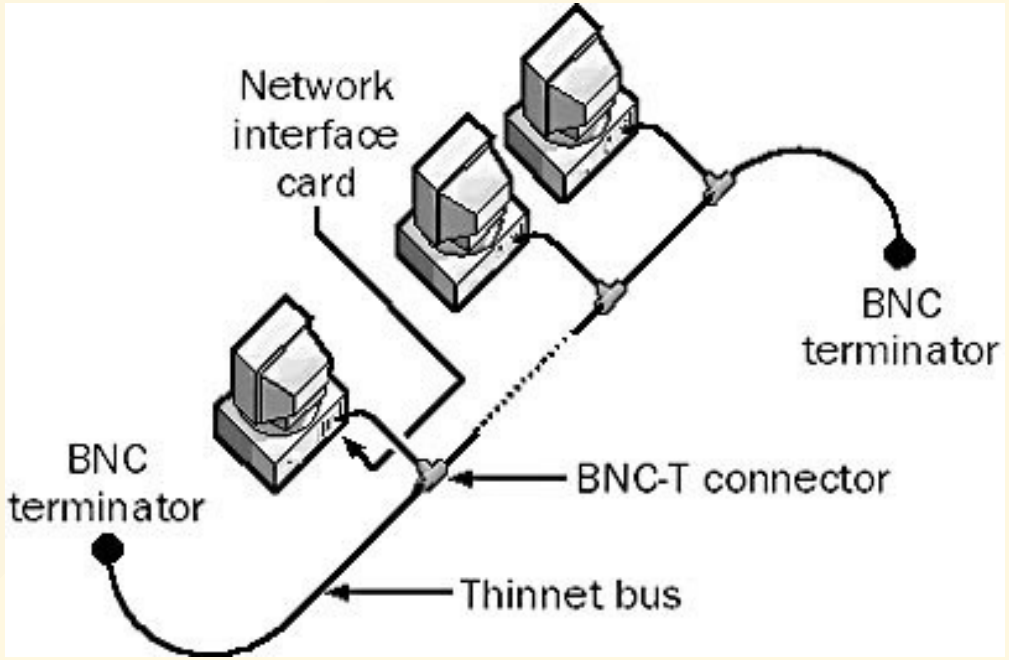
Host	MAC	IP	MASK
Host A	AAAA	192.168.1.5	255.255.255.0
Host B	BBBB	192.168.1.6	255.255.255.0
Host C	CCCC	192.168.1.7	255.255.255.0
Host D	DDDD	192.168.1.8	255.255.255.0

Question: How can we interconnect them?

Forming The "Smallest" Network

Option-1: 10Base2

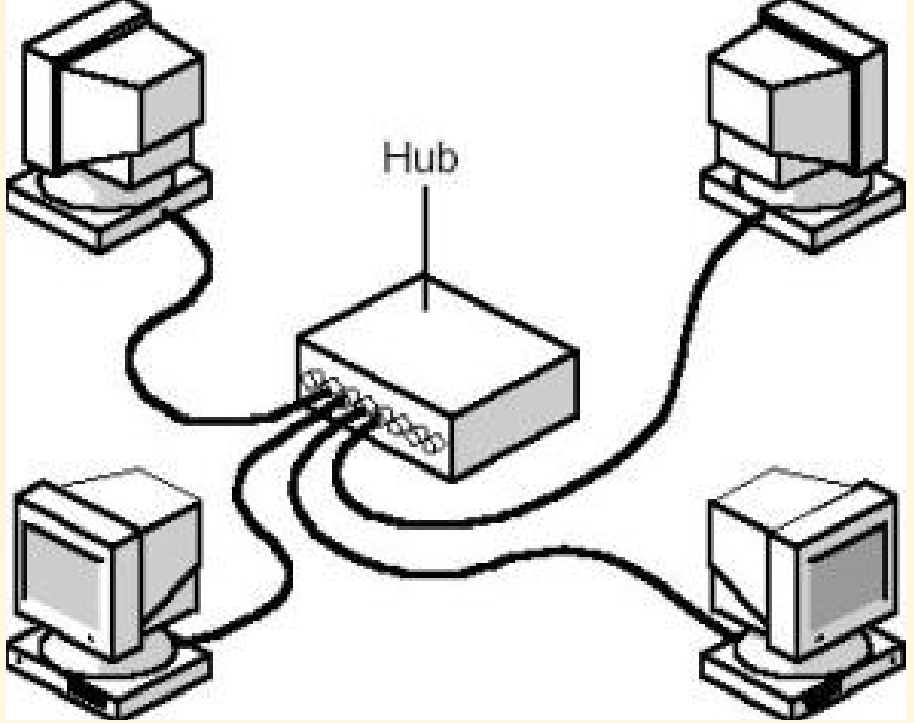
- Each host is attached to the cable using a T-connector.



Forming The "Smallest" Network

Option-2: Hub

- Each host is attached to the hub through an input/output port.



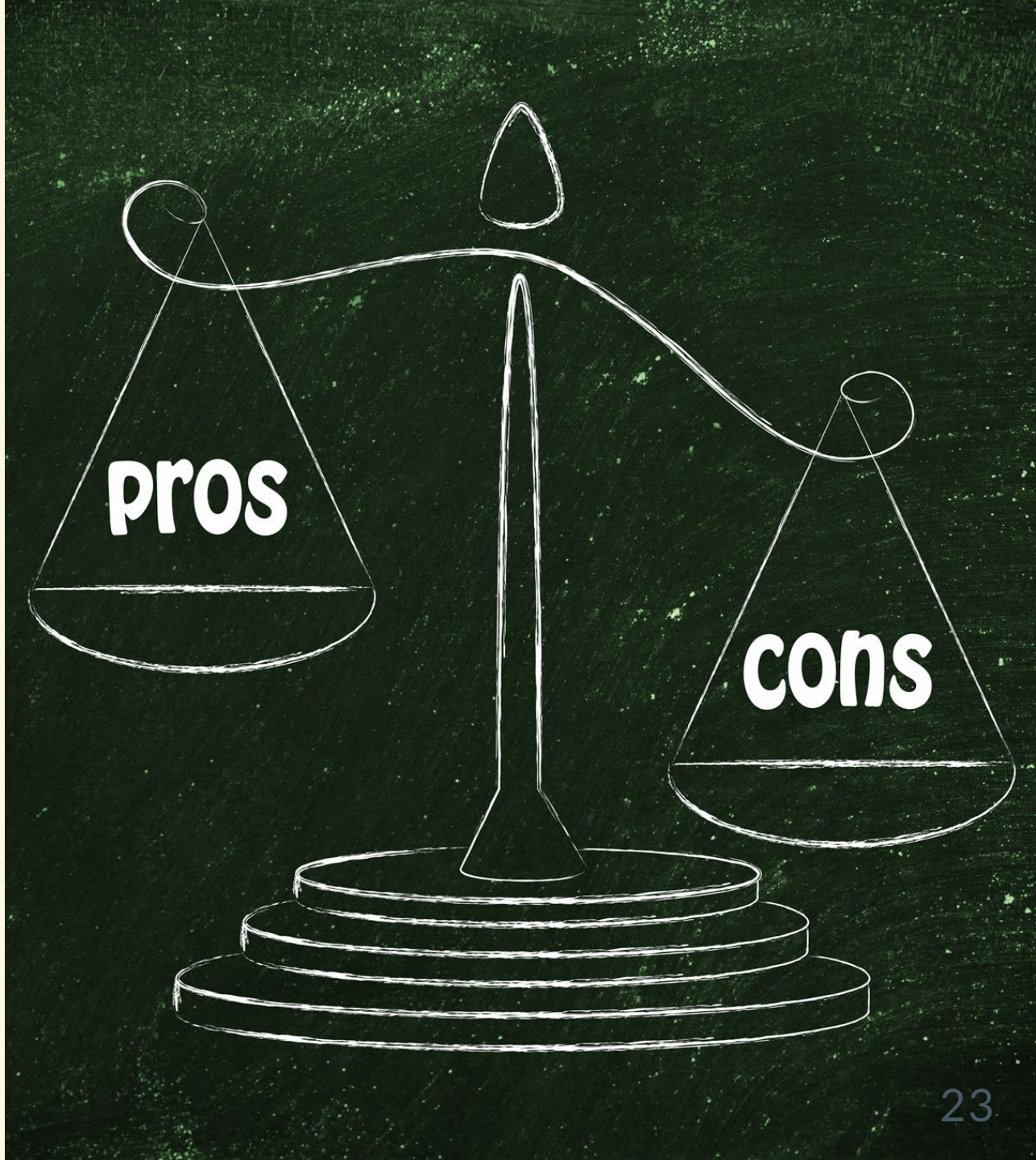
10Base2 and Hub

Pros

- Simple
- Cheap

Cons

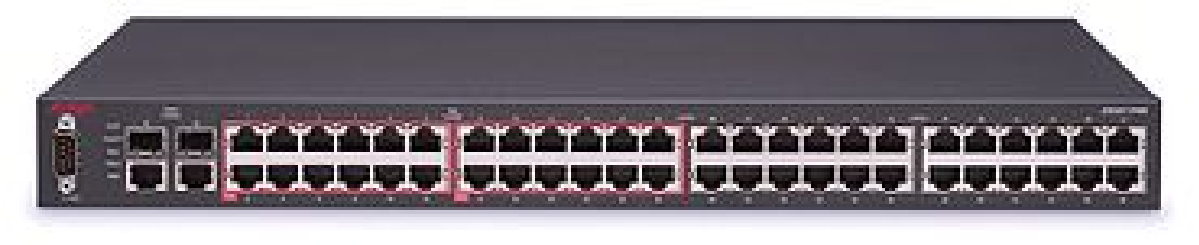
- Performance: coillison
- Privacy: A packet will be broadcasted to all ports



Forming The "Smallest" Network

Option-3: Switch

- Each host is attached to the switch through an input/output port (a.k.a, an interface).
- An switch forwards a packet to the appropriate destination instead of broadcasting it.



Switch

Maintains a MAC-to-Port Mapping Table.

- the table is intialized as empty
- entries are learnt by observing packets
- entries are used to forward packets

Switch

MAC	Port
AAAA	1
BBBB	2
CCCC	3
DDDD	4

Switch

Three Functions

- **Learning**: As the switch receives a packet, it reads the source MAC address and maps the interface to that MAC address.
- **Flooding**: The switch sends the incoming packet to all occupied and active interfaces except for the one from which the packet was received.
- **Forwarding**: Forward the packet according to its destination MAC address.

Communication In A Network Segment

An interaction among

- Hosts
- Switch
- ARP

Communication In A Network Segment

Host A sends a packet to Host B; A knows B's IP address

https://github.com/jzhang369/images/blob/main/switch.png

Communication In A Network Segment

- A uses its IP address, the netmask, and B's IP address to know that A and B are in the same network segment. Therefore, B is reachable via one-hop communication.
- Since A does not know B's MAC address, it will send a a packet to the switch:
  - [Dst_MAC:FF:FF:FF:FF:FF:FF|Src_MAC:AAAA|who has 192.168.1.6]

Communication In A Network Segment

- The switch receives this packet.
  - It knows the packet is coming from AAAA via Interface 1, adding an entry [1|AAAA] to the table.
  - The dest MAC address is FF...FF, a broadcast address. So it floods the packet to all occupied interfaces except for Interface 1.

Communication In A Network Segment

- Host C receives this packet but drops it after parsing.
- Host D receives this packet but drops it after parsing.
- Host B receives the packet, parses it, and finds that the destination IP address matches with its own IP address. It will send a packet to the switch:
  - [Dst_MAC:AAAA|Src_MAC:BBBB|I have 192.168.1.6]

Communication In A Network Segment

- The switch receives this packet
  - It knows the packet is coming from BBBB via Interface 2, adding an entry [2|BBBB] to the table.
  - The dest MAC address is AAAA, which is associated with Interface 1. So it forwards the packet to Interface 1.

Communication In A Network Segment

- Host A receives this packet
  - It knows 192.168.1.6 has MAC of BBBB and keeps this mapping in its cache.
  - It creates an IP packet
    - [Dst_MAC:BBBB|Src_MAC:AAAA|Dst_IP:192.168.1.6|Src_IP: 192.168.1.5|Data]
  - It sends this packet to the switch.

Communication In A Network Segment

- The switch receives this packet
  - The dest MAC address is BBBB, which is associated with Interface 2. So it forwards the packet to Interface 2.
- Host B recevies this IP packet.

Communication In A Network Segment

Additional Questions:

- What happens if a packet with non-broadcasting destination MAC address arrives at the switch but its switch table is empty?
  - flood
- You want to have a /24 network (e.g., with 255 hosts) and you have a number of switches, where each switch only has 48 ports. How do connect them?
  - [switch]-[switch]-[switch].....[switch]
  - It is still "flat".

Communication Across Network Segments

- Routers are needed.
- A router serves as the **gateway** of a network segment.
- Find your gateway's IP address using `ip route` or `route -n`

```
[jzhang@DESKTOP-DSVPHPI system32]$ip route
default via 172.31.96.1 dev eth0
172.31.96.0/20 dev eth0 proto kernel scope link src 172.31.104.107
[jzhang@DESKTOP-DSVPHPI system32]$route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.31.96.1     0.0.0.0         UG    0      0        0 eth0
172.31.96.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
```

Router

A router

- has an IP address for each active interface/port.
- has a MAC address for each interface/port.
- has an ARP table/cache.
- maintains routing table.

Router

A routing table decides where to forward a packet based on its destination IP address.

An entry of the routing table could be:

- Directly Connected (DC): it is for a host that is reachable from the router within one hop (considering switches transparent).
- Static Route: the ip address of another router, staticly configured.
- Dynamic Route: the ip address of another router, dynamically learnt (e.g., RIP, OSPF, and BGP).

Router

An example routing table

Type	Prefix	Destination
DC	192.168.1.0/24	1
DC	192.168.2.0/24	2
Static	192.168.3.0/24	192.168.2.2
Dynamic	192.168.4.0/24	192.168.2.2

Communication Across Network Segments

192.168.1.5 attempts to send a packet to 192.168.2.6.
https://github.com/jzhang369/images/blob/main/router.png

Communication Across Network Segments

Context:

- Each switch can build its MAC-to-Port table.
- Each interface of a router/host can discover the MAC address of an IP address using ARP.

For example:

- 192.168.1.5 can identify the MAC address of 192.168.1.1 using ARP and vice versa.

Communication Across Network Segments

192.168.1.5 wants to send an IP packet to 192.168.2.6.

- 192.168.1.5 checks 192.168.1.5 & 255.255.255.0 != 192.168.2.6 & 255.255.255.0. It knows that this IP address is in another network segment and it needs its gateway (192.168.1.1) to route this packet.
- 192.168.1.5 consults its ARP cache or uses ARP to know the gateway's MAC, BBBB. It then sends an IP packet to the gateway.
  - [Dst_MAC:BBBB|Src_MAC:AAAA|Dst_IP:192.168.2.6|Src_IP:192.168.1.5|Data]

Communication Across Network Segments

- The interface-1 of the router receives this packet. It checks the routing table and knows the packet should be sent to the interface-2, which connects to the network segment of 192.168.2.0/24.
- It consults its ARP cache, or performs ARP, to get MAC address of 192.168.2.6, which is DDDD.
- It sends the following packet to interface-2
  - [Dst_MAC:DDDD|Src_MAC:CCCC|Dst_IP:192.168.2.6|Src_IP:192.168.1.5|Data]
- 192.168.2.6 receives this packet.

Communication Across Network Segments

192.168.1.5 attempts to send a packet to 192.168.3.8.
https://github.com/jzhang369/images/blob/main/router2.png

Communication Across Network Segments

192.168.1.5 wants to send an IP packet to 192.168.3.8.

- 192.168.1.5 checks 192.168.1.5 & 255.255.255.0 != 192.168.3.8 & 255.255.255.0. It knows that this IP address is in another network segment and it needs its gateway (192.168.1.1) to route this packet.
- 192.168.1.5 consults its ARP cache or uses ARP to know the gateway's MAC, BBBB. It then sends an IP packet to the gateway.
  - [Dst_MAC:BBBB|Src_MAC:AAAA|Dst_IP:192.168.3.8|Src_IP:192.168.1.5|Data]

Communication Across Network Segments

- The interface-1 of the router-1 receives this packet. It checks the routing table and knows i) the packet should be routed via 192.168.2.2 and ii) this IP is in the segment of 192.168.2.0/24 with interface 2.
- It consults its ARP cache, or performs ARP, to get MAC address of 192.168.2.2, which is EEE1.
- It sends the following packet to interface-2
  - [Dst_MAC:EEE1|Src_MAC:CCCC|Dst_IP:192.168.3.8|Src_IP:192.168.1.5|Data]

Communication Across Network Segments

- The interface-1 of the router-2 receives this packet. It checks the routing table and knows the destination IP address 192.168.3.8 is in the segment of 192.168.3.0/24 with interface 2.
- It consults its ARP cache, or performs ARP, to get MAC address of 192.168.3.8, which is AAA3.
- It sends the following packet to its interface-2
  - [Dst_MAC:AAA3|Src_MAC:EEE2|Dst_IP:192.168.3.8|Src_IP:192.168.1.5|Data]

Communication Across Network Segments

- The host 192.168.3.8 now receives this packet.

Communication Across Network Segments

Additional Questions:

- Can you enumerate how 192.168.2.6 sends a packet to 192.168.3.8?
  - Which router gets this packet first?

Communication Across Network Segments

Additional Questions:

- If you remove 192.168.2.6, will the leftmost segment and the rightmost one mutually reachable?
  - Router-1 and -2 are now "directly" connected via the Ethernet (i.e., switch(es))
    - [Hosts]---[Switch-3]---[Router-1]--[Switch-2]--[Router-2]---[Switch-1]---[Hosts]

Communication Across Network Segments

Two routers can be connected via

- Ethernet (see the previous slide)
- Serial Ports (point-to-point connection)

Communication Across Network Segments

You can visually remove possible switches between routers so that the diagram is more similar to those ones you can find in the Internet.

- [Hosts]---[Switch-3]---[Router-1]---[Router-2]---[Switch-1]---[Hosts]

https://github.com/jzhang369/images/blob/main/router_simple.png

Communication Across Network Segments

The hierarchical organization of routers.

https://github.com/jzhang369/images/blob/main/router_hier.png

MISC

VLAN

https://github.com/jzhang369/images/blob/main/VLAN1.png

MISC

VLAN

https://github.com/jzhang369/images/blob/main/VLAN2.png

Packet Formats

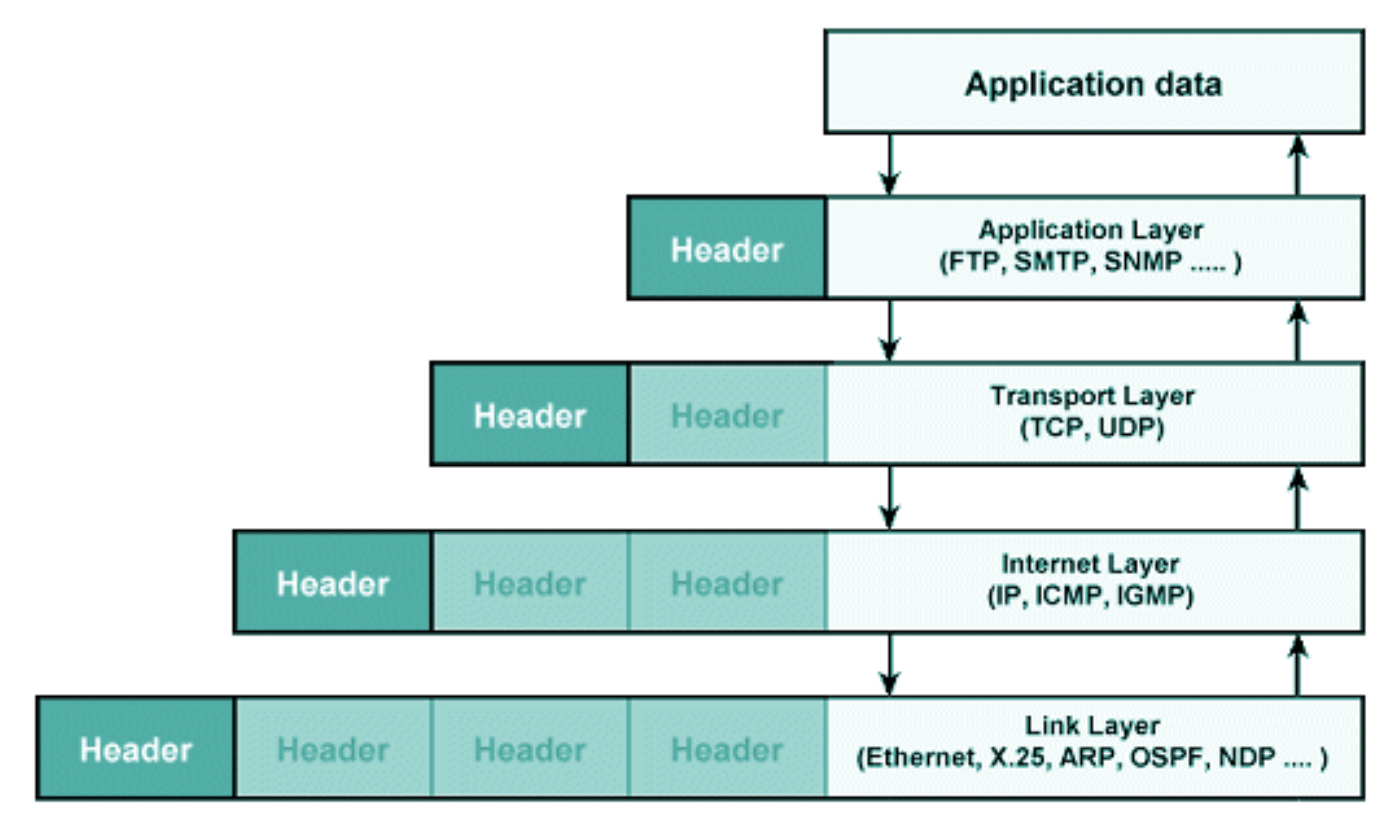
Revisit the structure of a packet - [Header|Payload]

- Packet Header
  - Address and other control information
- Packet Payload
  - Can be [Header|Payload] of another layer

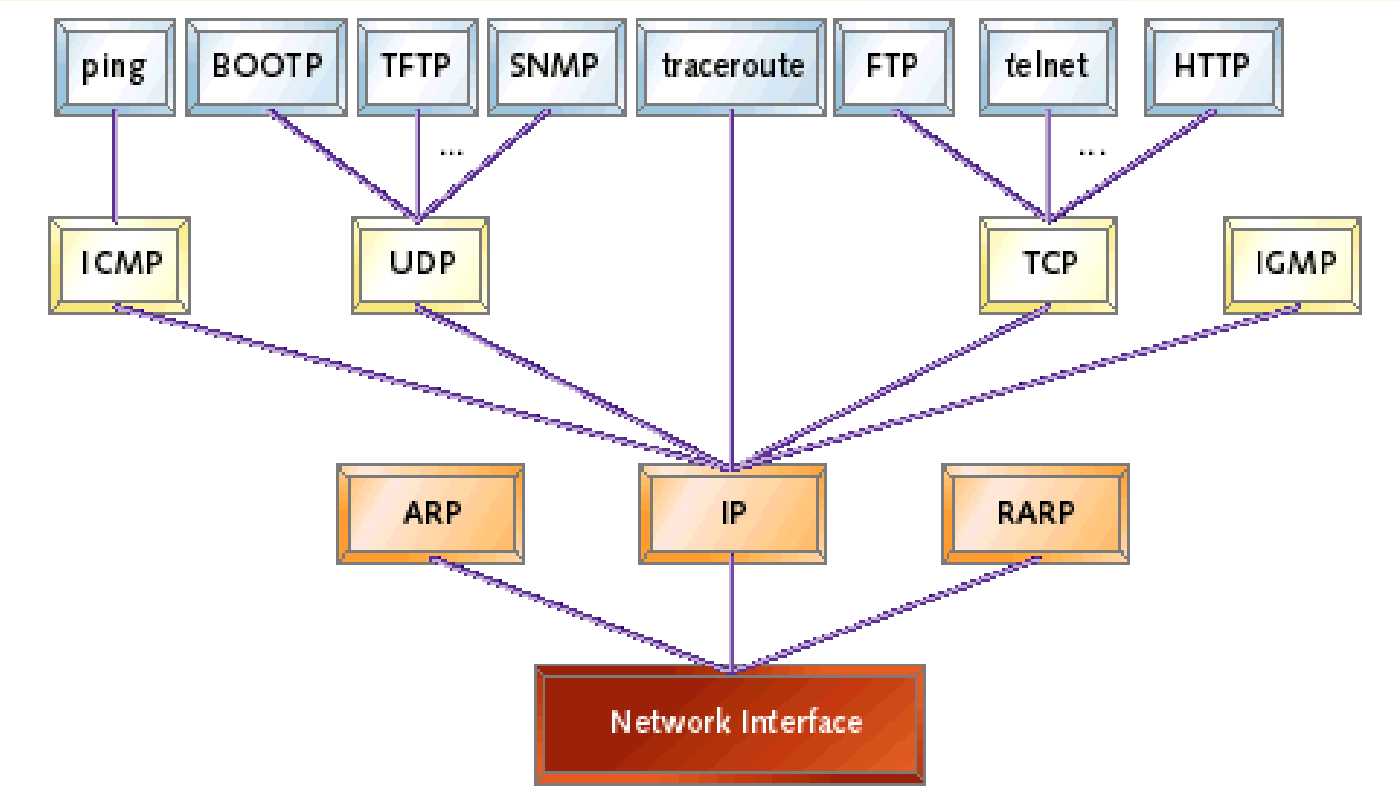
An Example:

- [Ether_Header|IP_Header|TCP_Header|HTTP_Header|HTTP_Data]

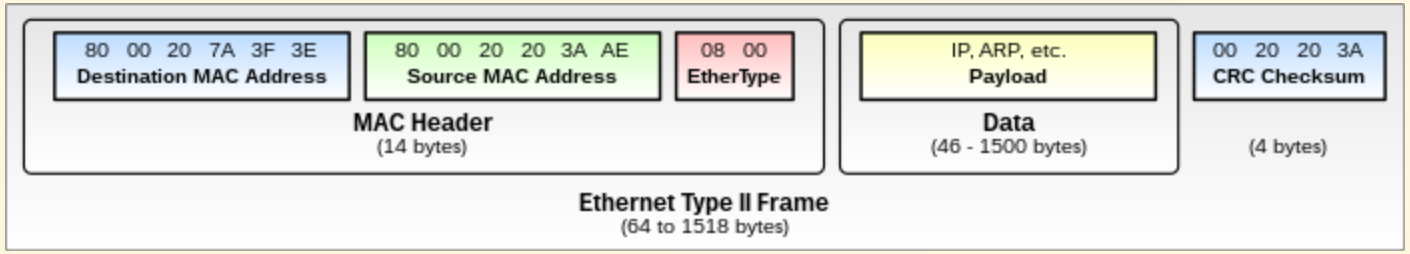
Packet Formats



Packet Formats

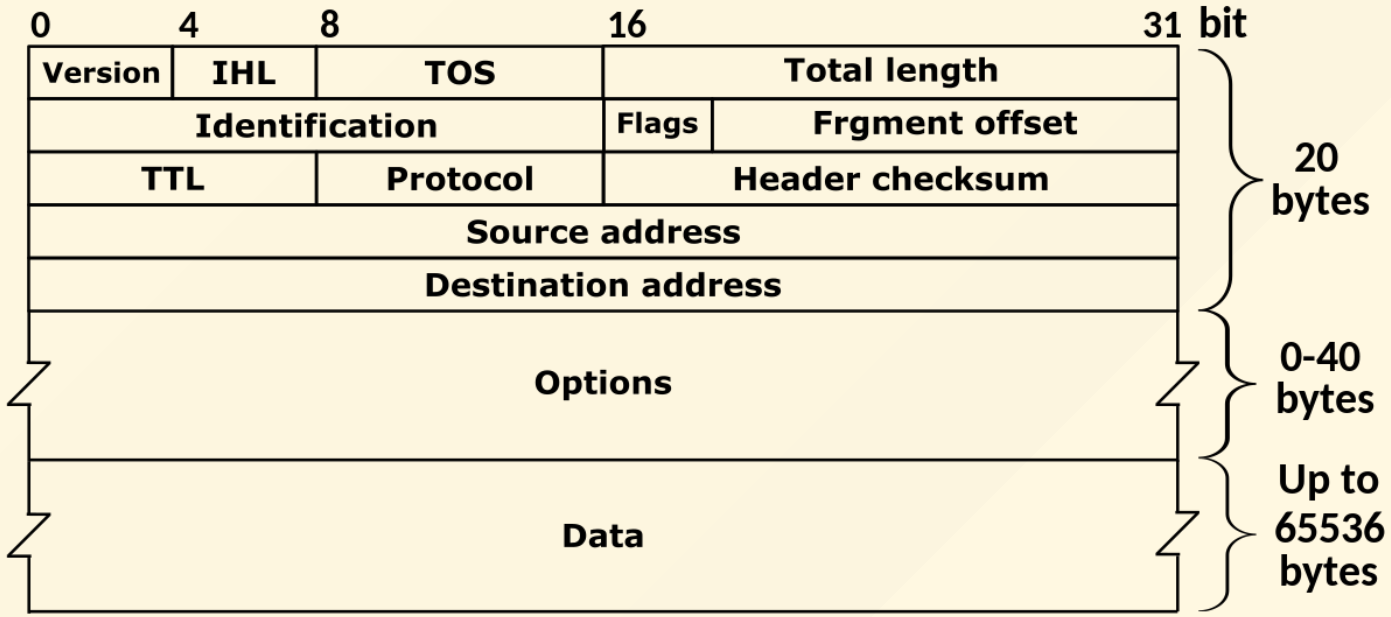


Packet Formats - Ethernet Type II Frame



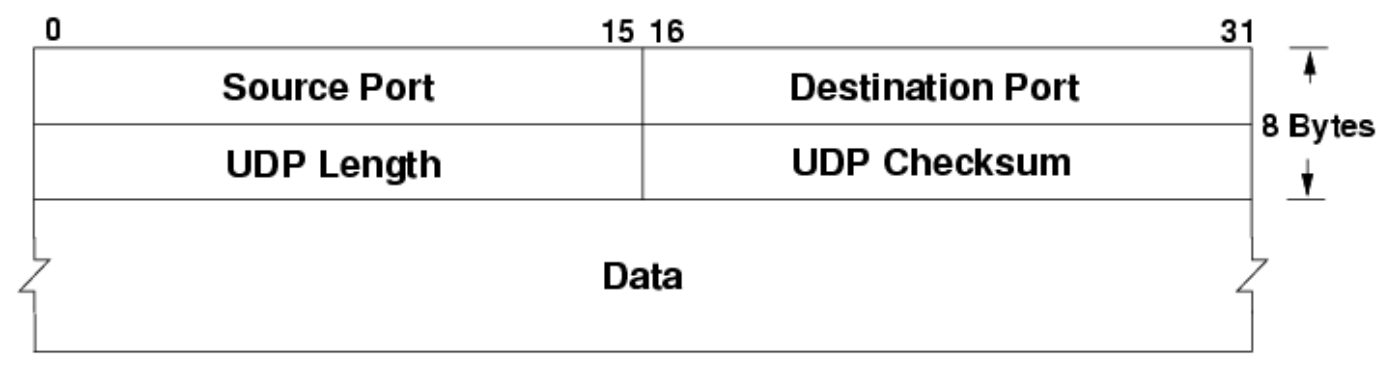
EtherType: IPv4-0x0800, ARP-0x0806, ...

Packet Formats - IP Header



Protocol: ICMP - 0x01, IGMP - 0x02, TCP - 0x06, UDP - 0x11

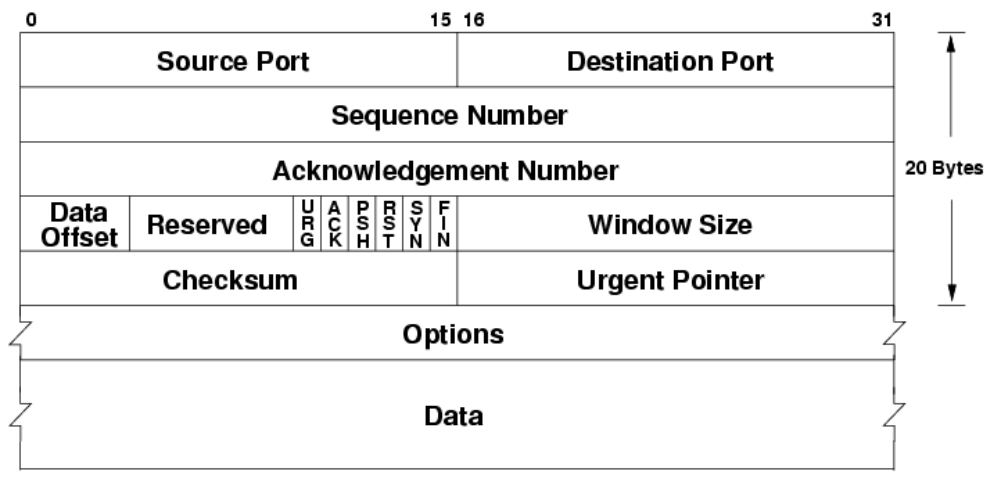
Packet Formats - UDP Header



Comments:

- Major differences compared to IP: source port and destination port added
- Same relability compared to IP

Packet Formats - TCP Header



Comments:

- Major differences compared to IP: source port and destination port added
- Provids fields to enable reliable communication

Packet Formats - Why Ports?

To enable isolation on end host.

- With port number, each UDP/TCP packet can be sent to its corresponding process, which is supposed to be isolated from other processes.

```
[jzhang@DESKTOP-DSVPHPI system32]$netstat -lp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 0.0.0.0:8888            0.0.0.0:*                           989/nc
```

Packet Formats - Application Level

Application Protocol	Transportation Layer	Popular Service Port
HTTP	TCP	80
HTTPS	TCP	443
SSH	TCP	22
DNS	UDP	53

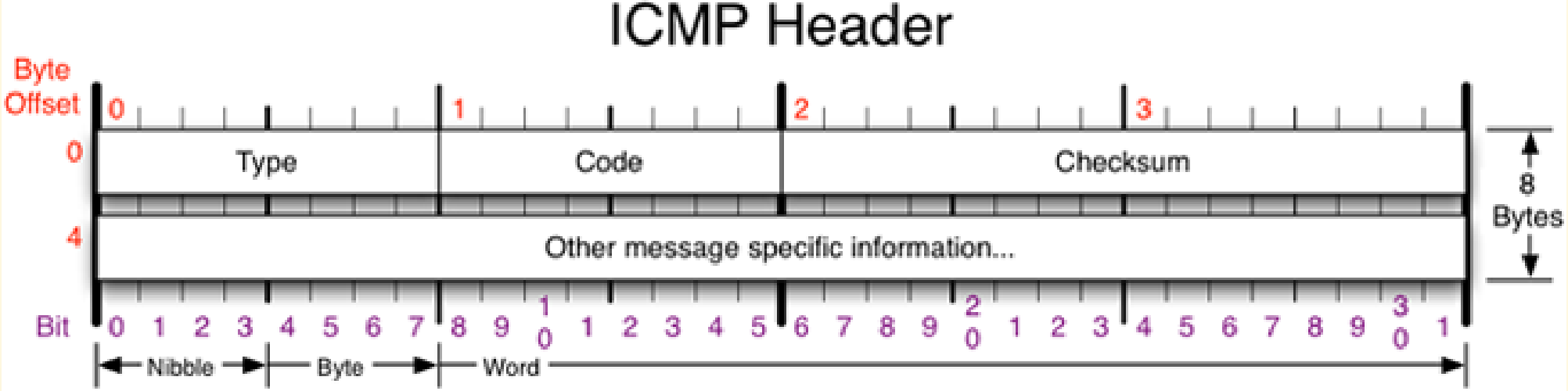
Popular services usually use their corresponding default ports **but they do not have to**.

Packet Formats - ICMP

The Internet Control Message Protocol (ICMP) is a protocol that devices within a network use to communicate problems with data transmission.

```
[jzhang@DESKTOP-DSVPHPI system32]$ping www.cnn.com
PING cnn-tls.map.fastly.net (146.75.83.5) 56(84) bytes of data.
64 bytes from 146.75.83.5 (146.75.83.5): icmp_seq=1 ttl=54 time=26.7 ms
64 bytes from 146.75.83.5 (146.75.83.5): icmp_seq=2 ttl=54 time=26.1 ms
64 bytes from 146.75.83.5 (146.75.83.5): icmp_seq=3 ttl=54 time=26.2 ms
```

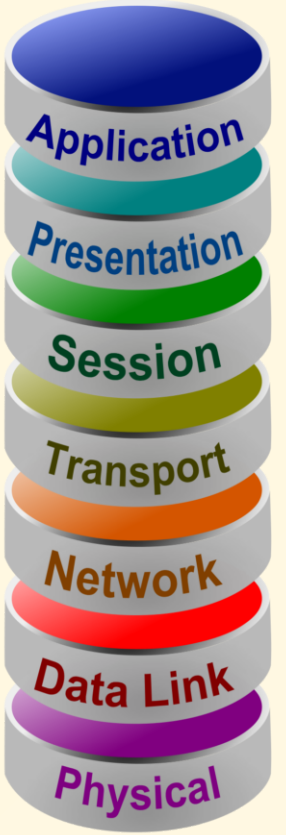
Packet Formats - ICMP



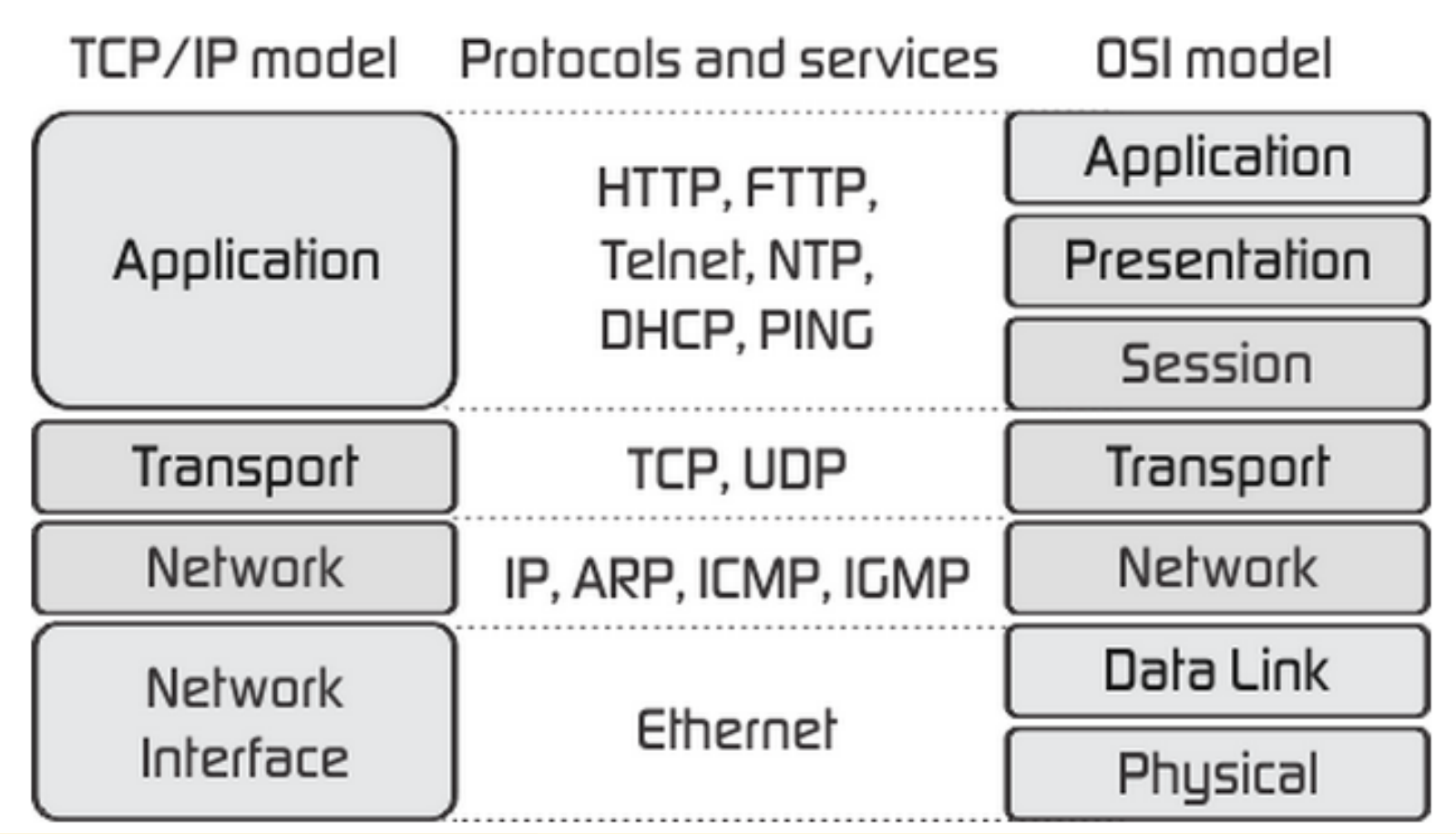
OSI & TCP/IP Model

The Open Systems Interconnection model (OSI model) is a conceptual model that describes the universal standard of communication functions of a telecommunication system or computing system.

The TCP/IP model is implements the OSI model using the TCP/IP protocol suites.



OSI & TCP/IP Model



Tools

- tcpdump
- wireshark
- netcat
- scapy
- whois and Team Cymru
- Argus

tcpdump

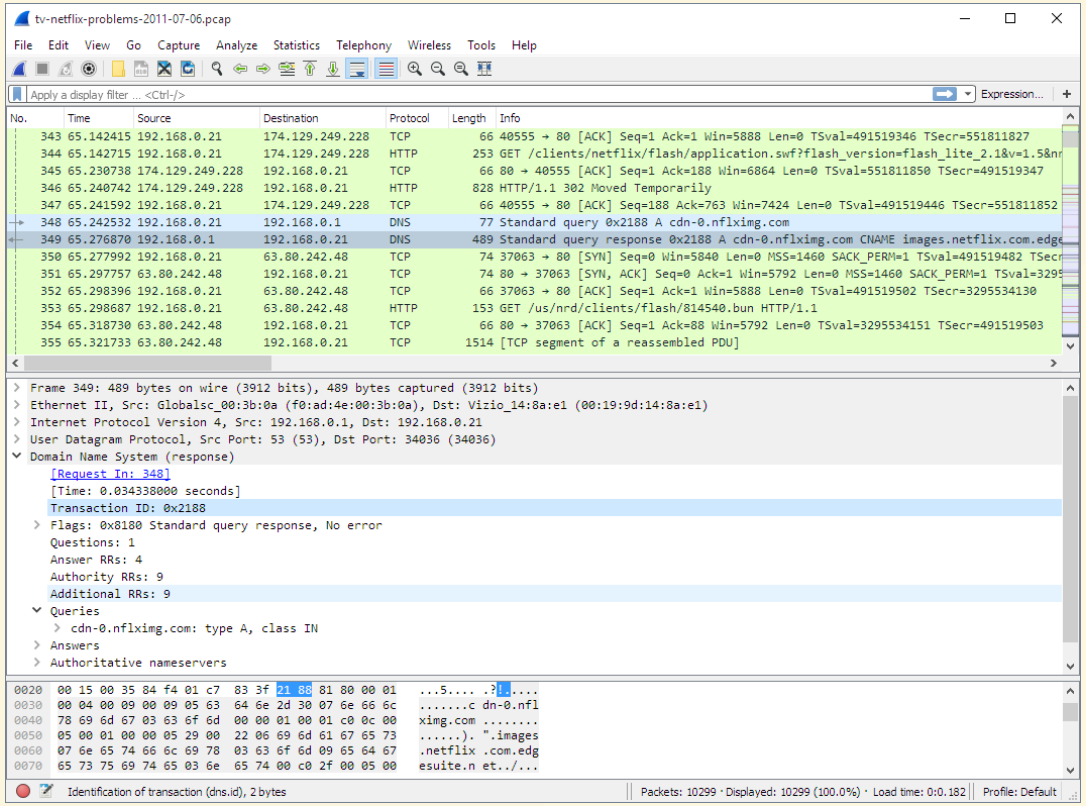
tcpdump is a data-network packet analyzer computer program that runs under a command line interface.

```
[jzhang@DESKTOP-DSVPHPI Temp]$sudo tcpdump -i eth0 -s0 -w trace.pcap -c 100
```

```
[jzhang@DESKTOP-DSVPHPI Temp]$tcpdump -r trace.pcap -an -c 2
reading from file trace.pcap, link-type EN10MB (Ethernet)
12:01:09.862663 IP 172.31.96.1.55505 > 239.255.255.250.1900: UDP, length 175
12:01:10.868764 IP 172.31.96.1.55505 > 239.255.255.250.1900: UDP, length 175
```

Wireshark

Wireshark is a GUI-enabled network packet analyzer.



netcat

Netcat is a networking utility which reads and writes data across network connections, using the TCP/IP protocol.

```
(networkenv) [jzhang@DESKTOP-DSVPHPI Temp]$nc -luv 8888
Bound on DESKTOP-DSVPHPI 8888
Connection received on localhost 56268
hello
```

```
[jzhang@DESKTOP-DSVPHPI system32]$nc -u localhost 8888
hello
```

scapy

Scapy is a python package used to sniff, analyze, and send and receive arbitrary network packets.

```
from scapy.all import *
print("packet sniffing starts.....")
def print_pkt(pkt):
    print("Src IP: ", pkt[IP].src)
    print("Dst IP: ", pkt[IP].dst)
    print("Protocol: ", pkt[IP].proto)
    print("\n")
pkt = sniff(filter="udp", prn=print_pkt)
```

whois and Team Cymru

Both whois and Team Cymru offers services to map IPs to BGP prefixes and ASNs. However, Team Cymru is more parsing- and automation-friendly.

```
[jzhang@DESKTOP-DSVPHPI system32]$whois 34.200.30.249
```

```
[jzhang@DESKTOP-DSVPHPI system32]$whois -h whois.cymru.com " -v  34.200.30.249"
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
14618   | 34.200.30.249    | 34.192.0.0/12       | US | arin     | 2016-09-12 | AMAZON-AES, US
```

Argus

Argus monitoring networks by generating **network flows**, where each network flow is a compact representation of a network session.

Storing packets is both expensive and sensitive.

A network flow record has fields of Timestamp, Duration, Src_IP, Src_Port, Dst_IP, Dst_Port, Protocol, Packets_Sent, Bytes_Sent, Packets_Received, Bytes_Received.

Argus

To generate network flows from pcap file, you can also use `statistics/conservations` in Wireshark.