

# What's Network?

Network is the way how two devices are communicating, The data sent between the two devices are called **packets**

## Communication protocol:

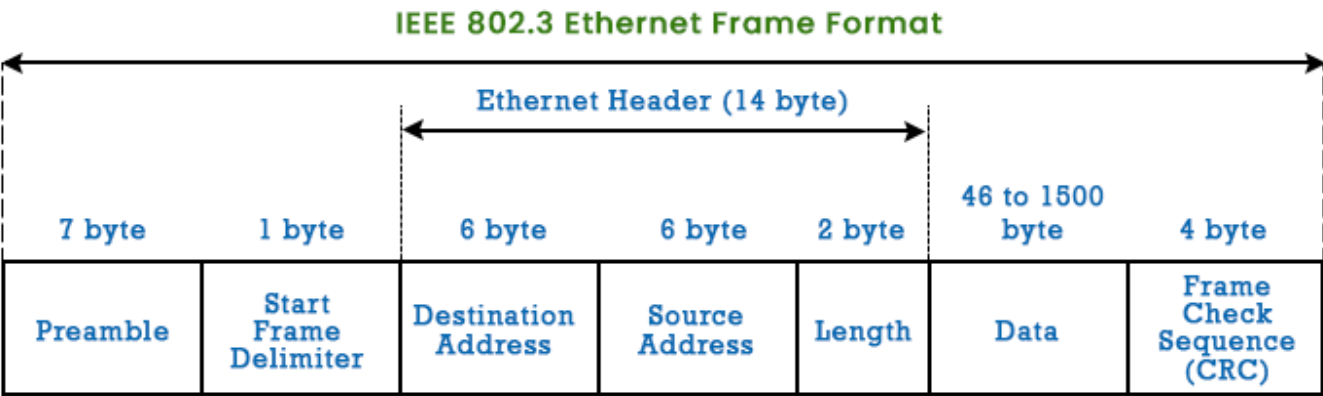
It's the way that the two devices are using to communicate with each others, we will use **SomeIp** it's a protocol used in the automotive industry.

- **Remote Process Communication:** Communication between two machines.
- **Inter-Process Communication:** Communication between two processes on the same machine.

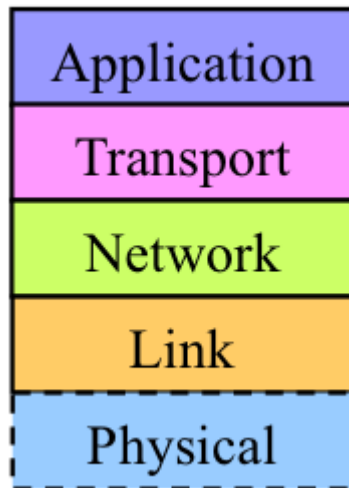
## TCP/IP Layers and Protocols

1. **Application Layer** Communication Protocol installed over that layer to be communicated with: **http**, **https**, **SOMEIP**, **ssh**.
2. **Transport Layer** Communication Protocol installed over that layer to be communicated with: **TCP**, **UDP**.
3. **Network Layer** Communication Protocol installed over that layer to be communicated with: **IP-protocol**.
4. **Network Access Layer** Communication Protocol installed over that layer to be communicated with: **Ethernet**, **Wifi**, **Bluetooth**.

## The Ethernet Frame stages:



1. When a two Network cards are communicating they send Ethernet frames to each others, The receiver one start By comparing the Destination address with it's own MacAdress,If matched it start analysing the data and take the subFrame to another stage
2. Then the subFrame goes to the network stack Layes as shown in fig



- 
- 3. In the Network We compare with the IP and port
- 4. Transport layer will deal with the sockets and ports

## Useful Commands and Tools

1. **ifconfig -a**: Displays all network interfaces and their MAC addresses.

- **NIC Information:**

- **MAC Address:** A unique identifier for each NIC, assigned at the factory.

```
enp4s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
  → ether 60:18:95:29:0c:fb txqueuelen 1000 (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
```

- **IP Address:** Can be dynamically assigned by the router.

- **NIC Status:**

- **Up:** The NIC is ready to transmit data.
- **Down:** The NIC is disabled and cannot transmit data.

```

enp4s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether 60:18:95:29:0c:fb txqueuelen 1000 (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

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4538 bytes 2037132 (2.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4538 bytes 2037132 (2.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lxcbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 10.0.3.1 netmask 255.255.255.0 broadcast 10.0.3.255
    ether 00:16:3e:00:00:00 txqueuelen 1000 (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

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.10.7 netmask 255.255.255.240 broadcast 172.20.10.15
    inet6 fe80::914d:267b:5e18:834f prefixlen 64 scopeid 0x20<link>
    ether 04:56:e5:ef:24:a0 txqueuelen 1000 (Ethernet)
    RX packets 127126 bytes 154954506 (154.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 43917 bytes 15654168 (15.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- The `enp4s0`, `lo`, `wlp0s20f3` all of them are network interfaces cards.
- The `mtu` (Maximum Transmission Unit) defines the largest packet size that can be transmitted over the NIC.

2. **ethtool**: Displays current NIC settings.

3. **ping**: Verifies connectivity between two IP addresses, utilizing the **ICMP** protocol.

4. **tcpdump**: Captures network traffic and saves it for analysis.

## Wireshark

We are going to use the wireshark to analyse the data, The layers in the wireshark are ordered from the physical layer up to the application layer

1	0.000000000	172.20.10.7	142.251.37.164	TLSv1.2	728 Application Data
2	0.002198636	172.20.10.7	142.251.37.164	TLSv1.2	105 Application Data
3	0.205155476	172.20.10.7	142.251.37.164	TCP	105 [TCP Retransmission] 57528
4	0.230553531	142.251.37.164	172.20.10.7	TCP	66 443 → 57528 [ACK] Seq=1 Ac
5	0.230553776	142.251.37.164	172.20.10.7	TCP	66 443 → 57528 [ACK] Seq=1 Ac
6	0.242303650	142.251.37.164	172.20.10.7	TLSv1.2	105 Application Data
7	0.285149246	172.20.10.7	142.251.37.164	TCP	66 57528 → 443 [ACK] Seq=702
8	0.308097604	142.251.37.164	172.20.10.7	TLSv1.2	751 Application Data
9	0.308175950	172.20.10.7	142.251.37.164	TCP	66 57528 → 443 [ACK] Seq=702
10	0.313696962	142.251.37.164	172.20.10.7	TLSv1.2	353 Application Data
11	0.313757126	172.20.10.7	142.251.37.164	TCP	66 57528 → 443 [ACK] Seq=702
12	0.313697220	142.251.37.164	172.20.10.7	TLSv1.2	127 Application Data
13	0.313801718	172.20.10.7	142.251.37.164	TCP	66 57528 → 443 [ACK] Seq=702
14	0.314058808	142.251.37.164	172.20.10.7	TLSv1.2	139 Application Data
15	0.314083983	172.20.10.7	142.251.37.164	TCP	66 57528 → 443 [ACK] Seq=702
16	0.314058878	142.251.37.164	172.20.10.7	TLSv1.2	105 Application Data

▶ Frame 1: 728 bytes on wire (5824 bits), 728 bytes captured (5824 bits) on interface wlp0s20f3, id 0  
 ▶ Ethernet II, Src: IntelCor\_ef:24:a0 (04:56:e5:ef:24:a0), Dst: 06:68:65:b0:de:64 (06:68:65:b0:de:64)  
 ▶ Internet Protocol Version 4, Src: 172.20.10.7, Dst: 142.251.37.164  
 ▶ Transmission Control Protocol, Src Port: 57528, Dst Port: 443, Seq: 1, Ack: 1, Len: 662  
 ▶ Transport Layer Security

## Ping and ICMP Protocol

The **ping** command uses the **ICMP** protocol to check if two machines can communicate. It sends an echo request and listens for an echo reply, verifying connectivity.

## Capturing and Analyzing Traffic

For systems without GUI tools like Wireshark, **tcpdump** can capture network traffic. The captured data can be transferred to another system for analysis with Wireshark.

[!TIP] The wireshark is a gui tool we can't use it with the embedded systems like raspberryPi, so we need to use the tcpdump tool to capture the traffic and save it to file called pcap, then take the pcap file and analyse it on the wireshark

we will use tcpdump tool to capture the traffic and save it to file called pcap, then take the pcap file and analyse it on the wireshark