CCNA 200-301

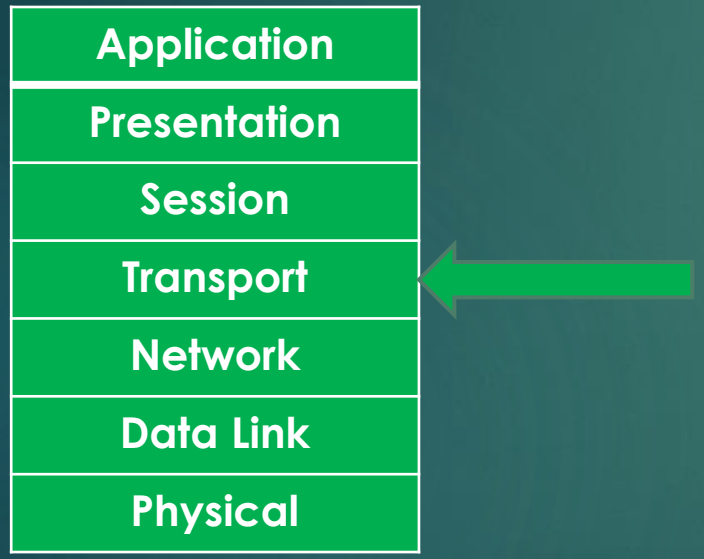
Lesson 3

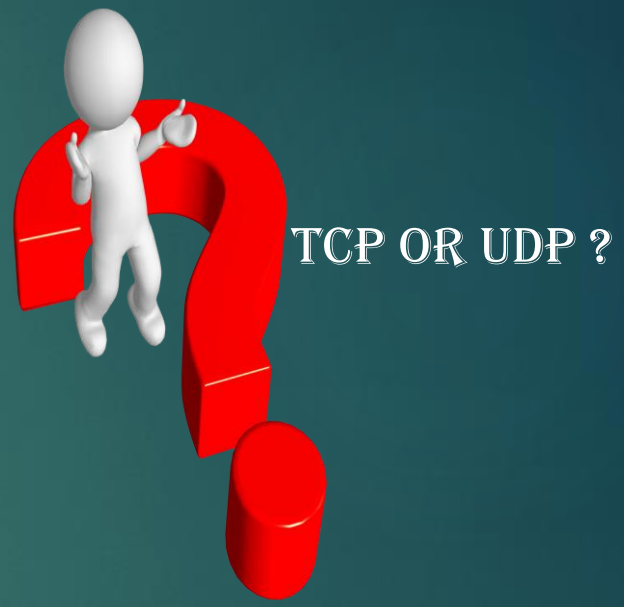
- TCP/UDP
- MAC ADDRESS
- ICMP
- DNS

TCP/UDP

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)

TCP and UDP are Transport Layer protocols.





- TCP is a **reliable** protocol.
- UDP is an **unreliable** or **best-effort** protocol.

TCP vs UDP

TCP sets up connection using "**3-way handshake**" between nodes before sending data.

UDP sends data and does not care whether it arrives.

TCP add sequence number to segments if any segment does not arrive to receiver node it is sent again.

UDP does not use sequence number.

	TCP	**UDP**
Connection Type:	Connection-oriented	Connectionless
Sequencing:	Yes	No
Usage:	Downloads File Sharing Printing	VoIP Video (streaming)

TCP “3-way Handshake”

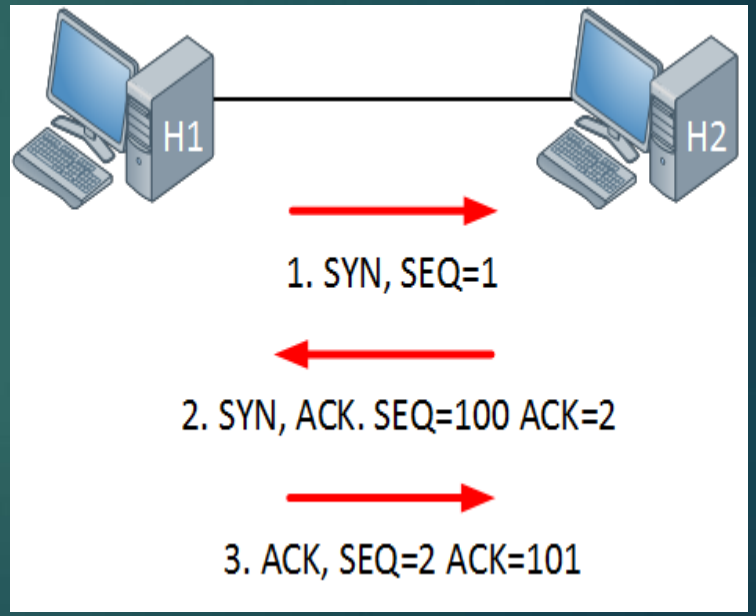
Now we have 2 hosts H1 and H2. H1 wants to send data to H2 In reliable way. That is why H1 use TCP.

Let’s follow how “3-way handshake” happens.

Step 1. H1 will send a **TCP SYN**, telling H2 that it wants to setup a connection.
There’s also a sequence number and to keep things simple I picked number 1.

Step 2. H2 will respond to H1 by sending a **SYN,ACK** message back.

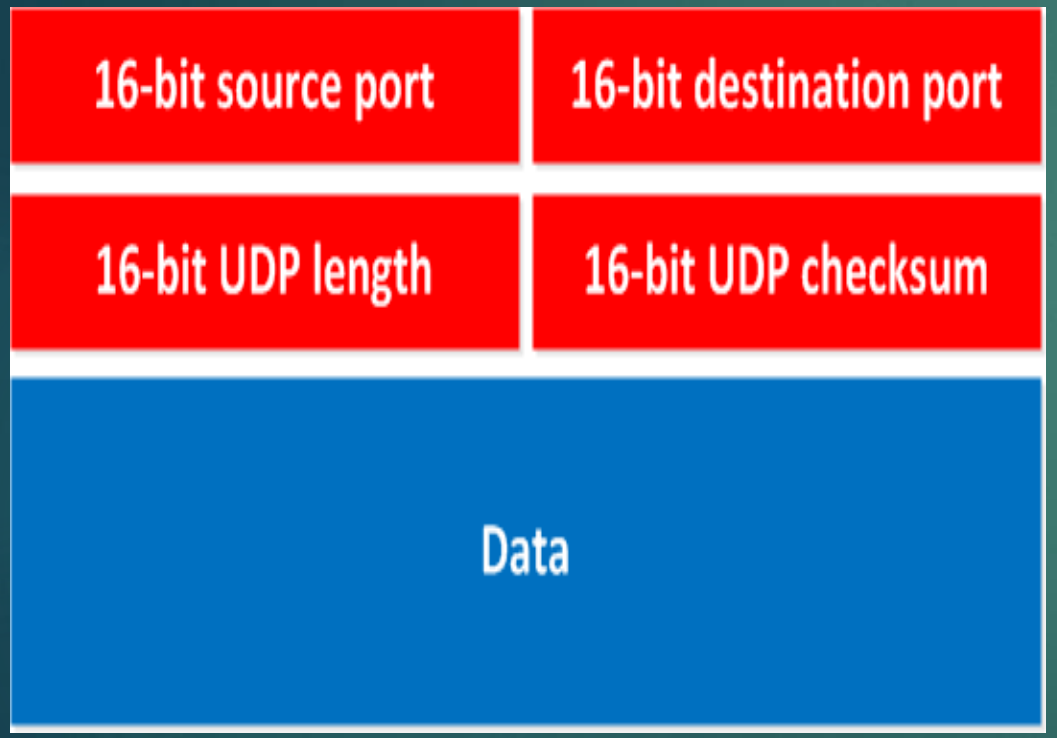
Step 3. The last step is that H1 will send an **acknowledgement** towards H2 in response of the SYN that H2 sent towards H1.
As a result, H1 and H2 sets up reliable TCP connection.



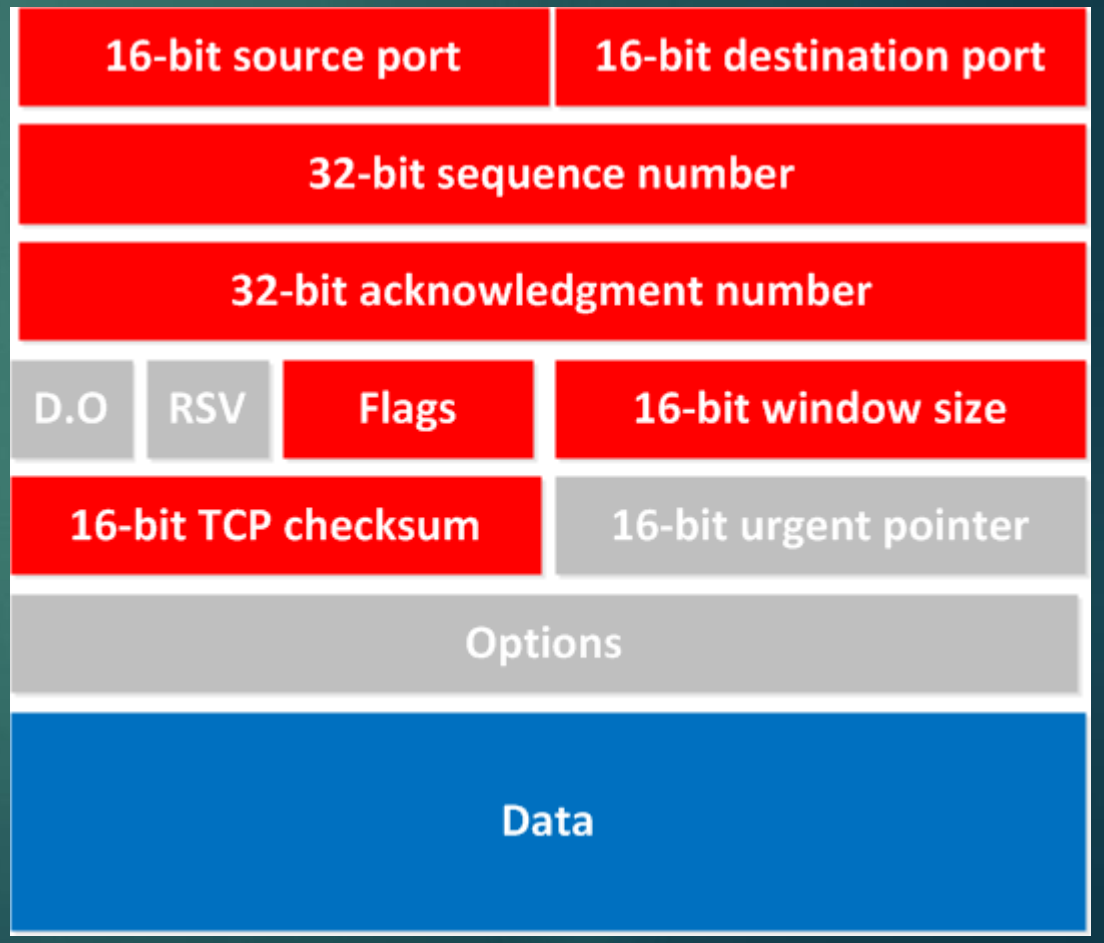
TCP/UDP header comparison

TCP is complex protocol than UDP. If we look at the headers we can see the difference.

UDP header

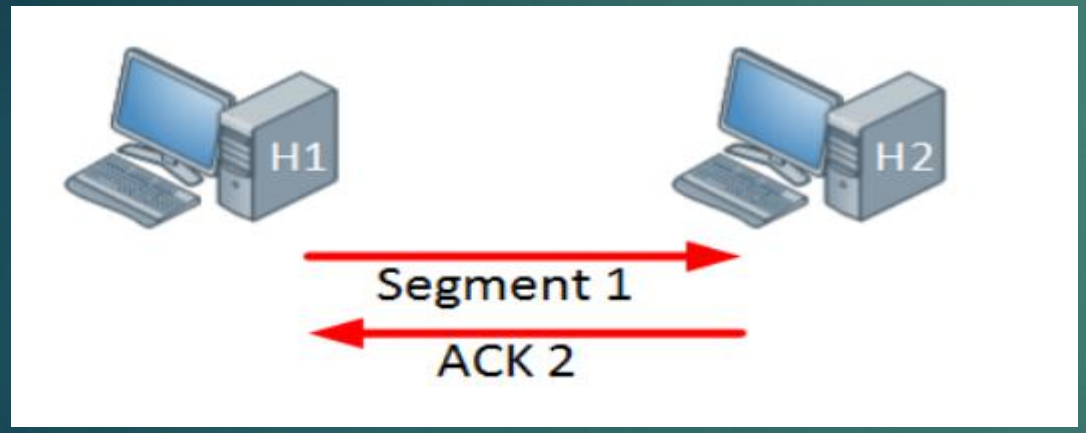


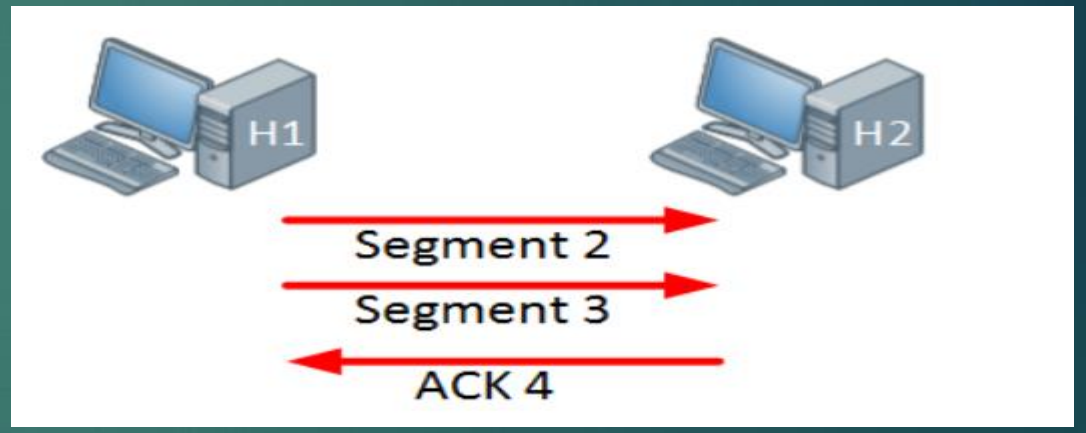
TCP header



TCP Window Size Scaling

Between two hosts, the transmitted and received data is under control of TCP. The sender sends and receiver acknowledges. In transport layer an original data is divided into segments. The process starts with the small segment. At the end of process the sender and receiver makes and agreement about "WINDOW SIZE" or "Windowing"





In the left image H1 starts to send segment one and H2 replies with the ACK2.

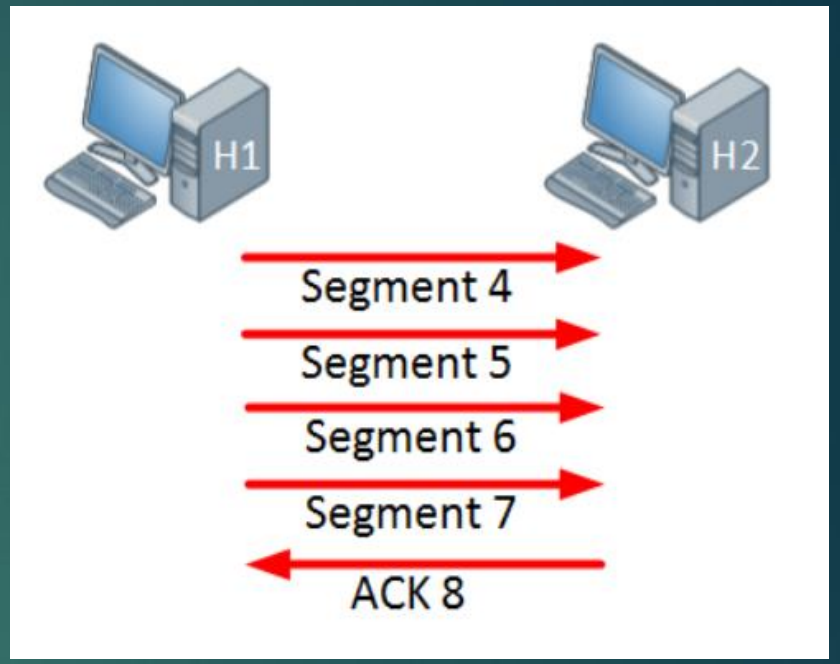
After receiving ACK2 H1 increases double size of sending segment and H2 replies with the ACK 4 and process continues.

TCP Window Size Scaling cont.

H1 goes on to send segments. Each time the number of segments is increased double.

When the receiver doesn't send an acknowledgment within a certain time period (called the round-trip time) then the window size will be reduced one segment.

Let's support senders have 8 segments in the case of congestion. When sender does not get ACK message it will reduce 1 of sending segments and in our case it will be 7 segments.



Example with Wireshark capture

PC1 and PC2 starts with 2070 bytes window size agreement.

St.1 = 2070

St.2 = 4140

St.3 = 8280

……

St.7 = 132480

As a result, the amount of sending data agreement equals to 132480 and is greater 64 times from stater size.

Layer1
Layer2
Layer3
Layer4

```
Frame 639: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: Raspberr_68:1e:36 (b8:27:eb:68:1e:36), Dst: AsustekC_7d:22:8c (74:d0:2b:7d:22:8c)
Internet Protocol Version 4, Src: 10.56.100.164 (10.56.100.164), Dst: 10.56.100.1 (10.56.100.1)
Transmission Control Protocol, Src Port: 22 (22), Dst Port: 56748 (56748), Seq: 2520, Ack: 51956, Len: 0
    Source Port: 22 (22)
    Destination Port: 56748 (56748)
    [Stream index: 16]
    [TCP Segment Len: 0]
    Sequence number: 2520    (relative sequence number)
    Acknowledgment number: 51956    (relative ack number)
    Header Length: 20 bytes
    .... 0000 0001 0000 = Flags: 0x010 (ACK)
        000. .... .... = Reserved: Not set
        ...0 .... .... = Nonce: Not set
        .... 0... .... = Congestion Window Reduced (CWR): Not set
        .... .0.. .... = ECN-Echo: Not set
        .... ..0. .... = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
    Window size value: 2070
    [Calculated window size: 132480]
    [Window size scaling factor: 64]
    Checksum: 0x102f [validation disabled]
    Urgent pointer: 0
    [SEQ/ACK analysis]
```

MAC (Media Access Control)

MAC address is assigned to NICs by the vendors and equals to 48-bit. MAC address is stored in CAM (Content Address Memory) table.

Other reference names:

- Burned-in-address,
- Hardware address,
- Ethernet address.

MAC address format: each octet =8 bit.

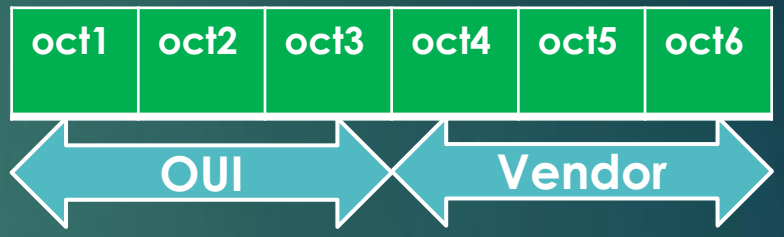
oct1	oct2	oct3	oct4	oct5	oct6

0050.7966.6802

00:50:79:66:68:02

00-50-79-66-68-02

0050-7966-6802



24 - bit 24 – bit

OUI – Organizationally Unique Identifier

Vendor – Cisco, Huawei, Alcatel, HP …

EVE-NG practical view

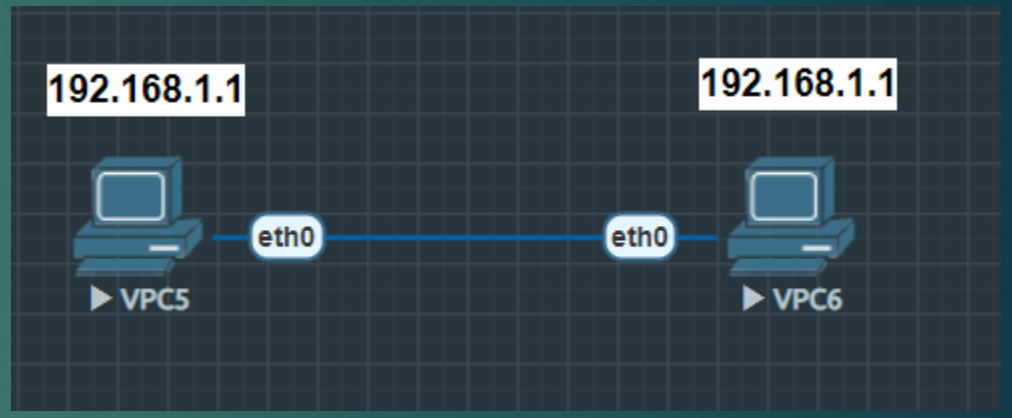
ICMP – Internet Control Message Protocol

ICMP is a network protocol that is used for network diagnostic. An example for ICMP is a PING utility. Ping uses ICMP for checking communication between network components.

Ping uses two types of ICMP message.

- ICMP Request
- ICMP Reply

Another example of ICMP is Traceroute utility.



VPCS> ping 192.168.1.2

84 bytes from 192.168.1.2 icmp_seq=1 ttl=64 time=0.539 ms

84 bytes from 192.168.1.2 icmp_seq=2 ttl=64 time=0.539 ms

84 bytes from 192.168.1.2 icmp_seq=3 ttl=64 time=0.493 ms

84 bytes from 192.168.1.2 icmp_seq=4 ttl=64 time=0.506 ms

84 bytes from 192.168.1.2 icmp_seq=5 ttl=64 time=0.479 ms

ICMP cont. – TTL value

TTL – Time - to - Live

To ensure IP packets have a limited lifetime on the network all IP packets have an 8-bit **Time to Live** (IPv4) or **Hop Limit** (IPv6) header field and value which specifies the maximum number of layer three hops (typically routers) that can be traversed on the path to their destination. For each traversal on the Layer 3 device the TTL is decreased by one. The max TTL value = 255.

This value is specified for each OS. In the below the default TTL value is shown for OSs.

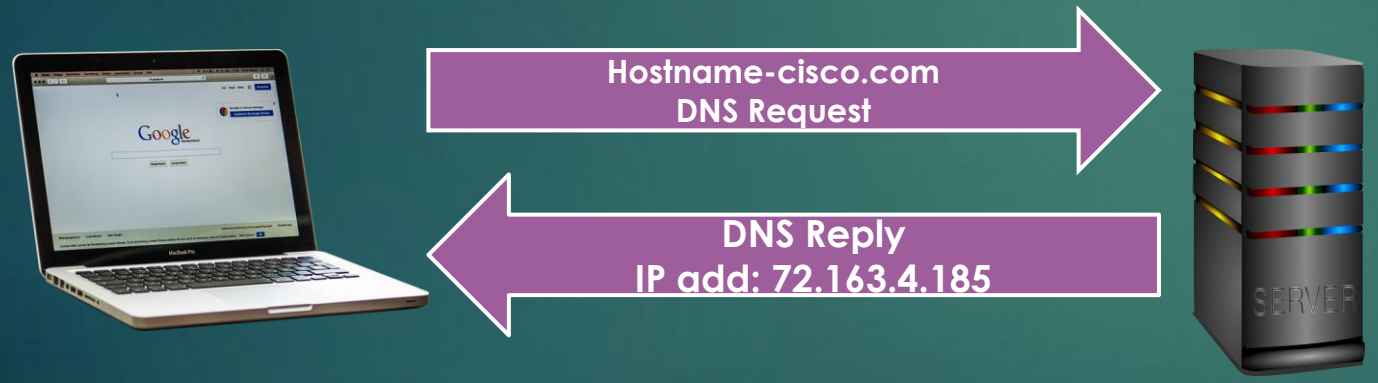
- Linux kernel 2.4 (circa 2001): **255** for TCP, UDP and ICMP
- Linux kernel 4.10 (2015): **64** for TCP, UDP and ICMP
- Windows XP (2001): **128** for TCP, UDP and ICMP
- Windows 10 (2015): **128** for TCP, UDP and ICMP
- Windows Server 2008: **128** for TCP, UDP and ICMP
- Windows Server 2019 (2018): **128** for TCP, UDP and ICMP
- MacOS (2001): **64** for TCP, UDP and ICMP

DNS (Domain Name System)

DNS is Application Layer protocol and used to find IP addresses according to the hostnames. Computers use IP addresses to reach servers. But it is not convenient for humans. Because it is difficult to learn all web servers IP address by heart. In this case DNS comes to help us. We write domain name and DNS resolve it to suitable IP address.

DNS uses Port Number 53 and is UDP based protocol.

How DNS works ?



DNS Practical

PC IPv4 add: 102.168.1/24

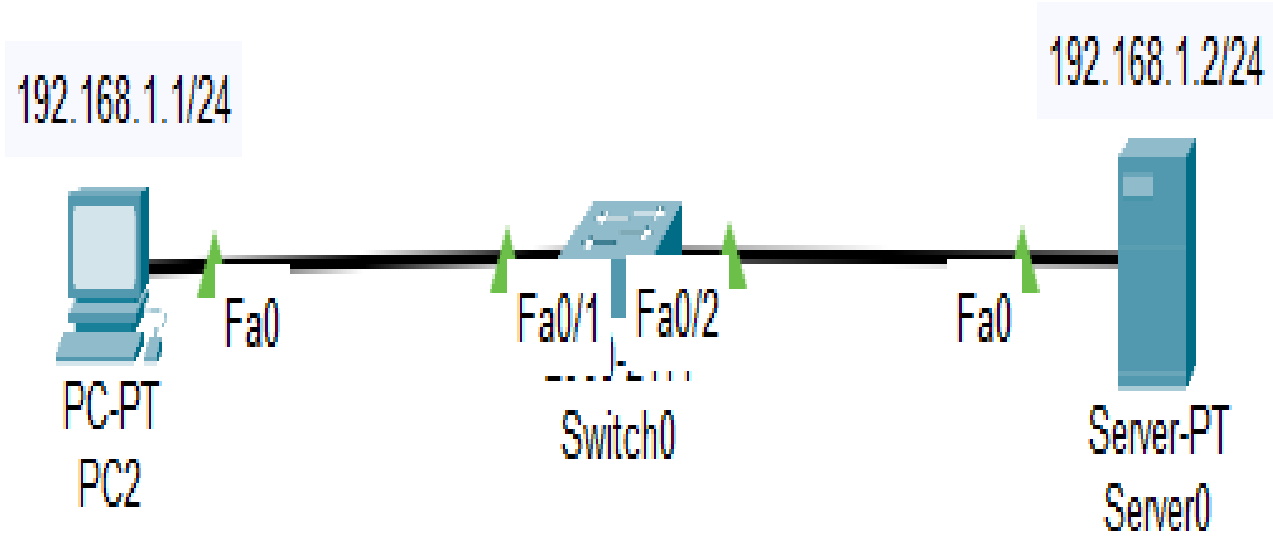
Server IPv4 add: 192.168.1.2/24

Domain name: cisco.com

PC2 sends: DNS Request or Query,

DNS server sends: DNS Reply.

Packet Tracer practical view…



That is all for Lesson 3

The key is :

Learn

Repeat

Practice

You will be able to reach your goals.

GOOD LUCK !!!!!...