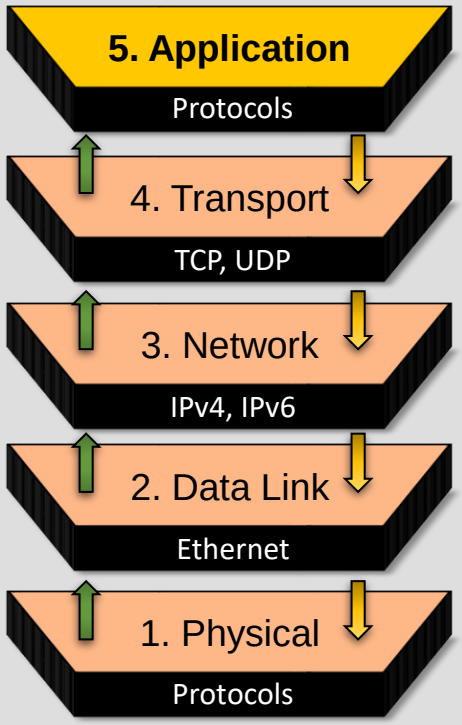
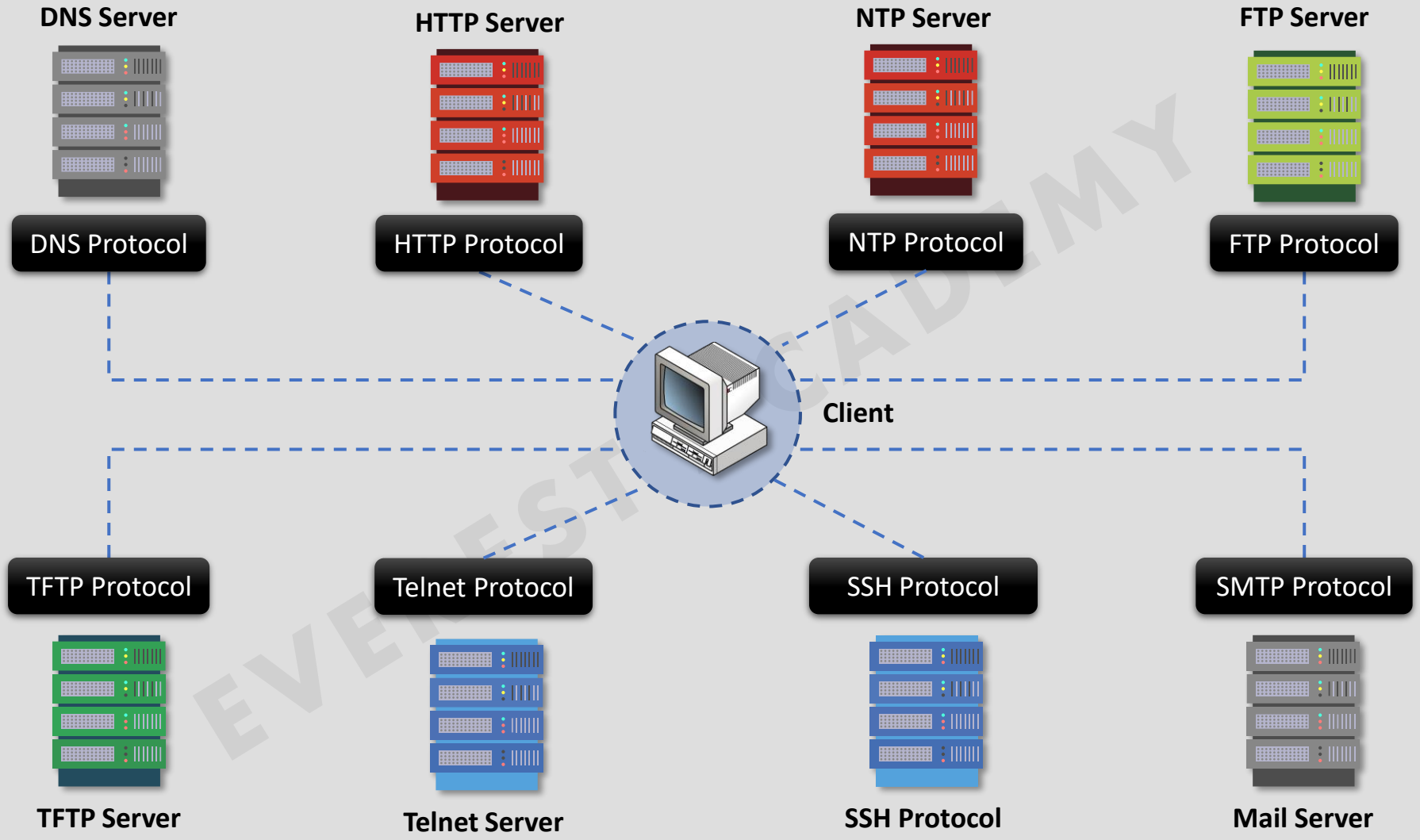
TCP/IP Applications

1. **DNS :** Domain Name System. — TCP, UDP (53)
2. **HTTP:** Hypertext Transfer Protocol . — TCP (80)
3. **HTTPs :** Hypertext Transfer Protocol Secure . — TCP (443)
4. **FTP :** File Transfer Protocol . — TCP (20,21)
5. **TFTP :** Trivial File Transfer Protocol . — UDP (69)
6. **Telnet :** Teletype Network. — TCP (23)
7. **SSH :** Secure Shell . — TCP (22)
8. **SMTP :** Simple Mail Transfer Protocol. — TCP (25)
9. **POP3 :** Post Office Protocol. — TCP (110)
10. **IMAP :** Simple Mail Transfer Protocol. — TCP (143)
11. **NTP :** Network Time Protocol. — UDP (123)
12. **DHCP :** Dynamic Host Configuration Protocol. — UDP (67,68)

TCP/IP Updated TCP/IP Model



TCP/IP Applications

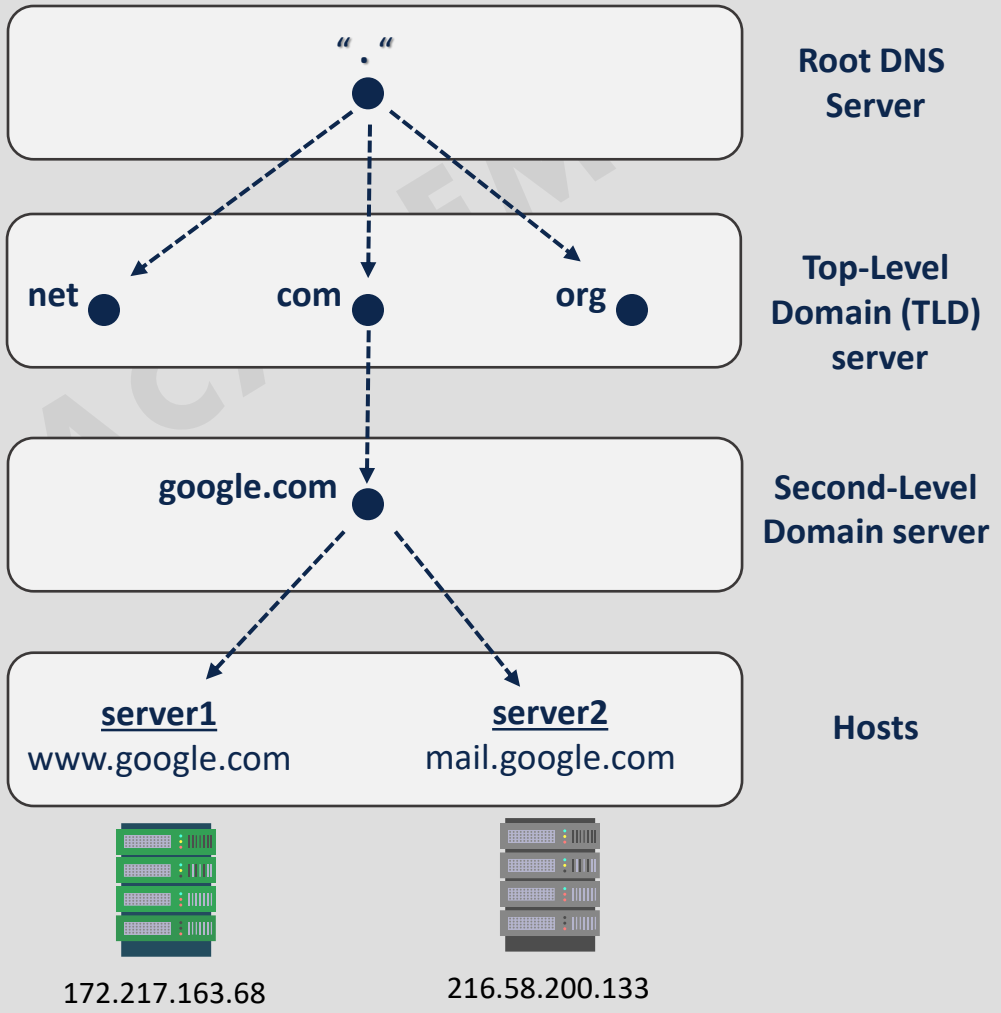


Domain Name System Architecture

The Domain Name is a symbolic string associated with an IP address such as youtube.com

The Domain Name Space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels with a root at the top.

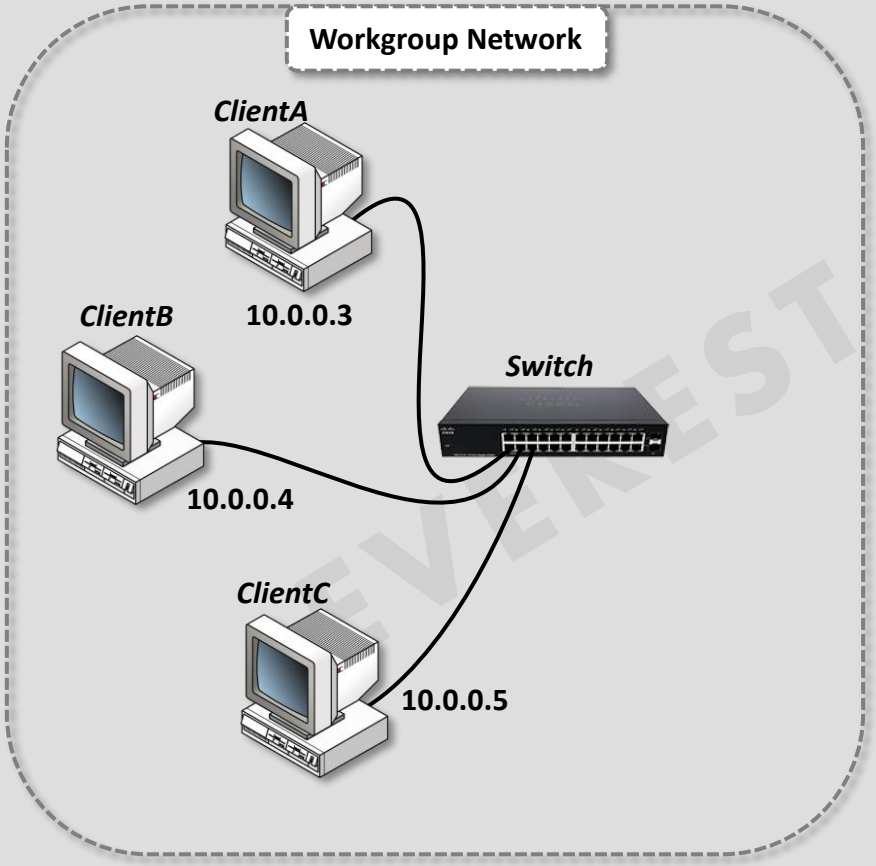
The Name Server contains the DNS database. This database comprises of various names and their corresponding IP addresses.

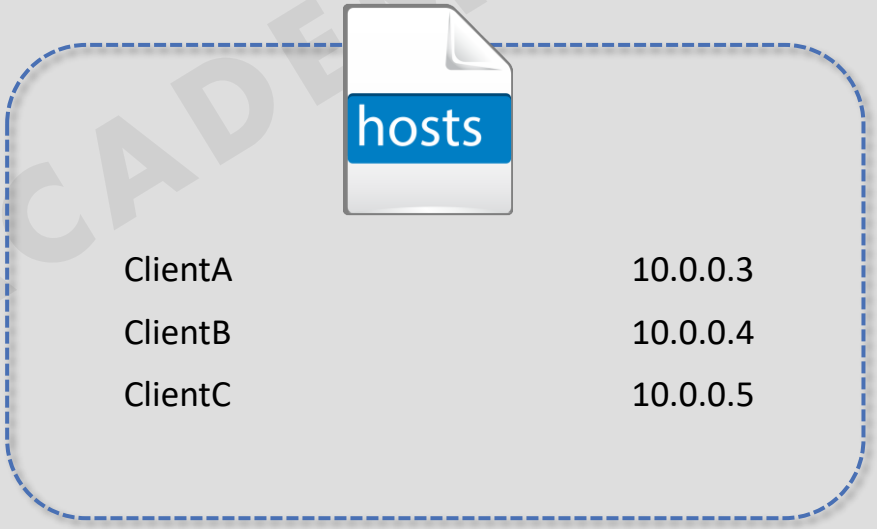


Domain Network System (DNS) – Workgroup Network

Domain Name System helps to resolve the host name to an address**.**

It uses a hierarchical naming scheme and distributed database of IP addresses and associated names

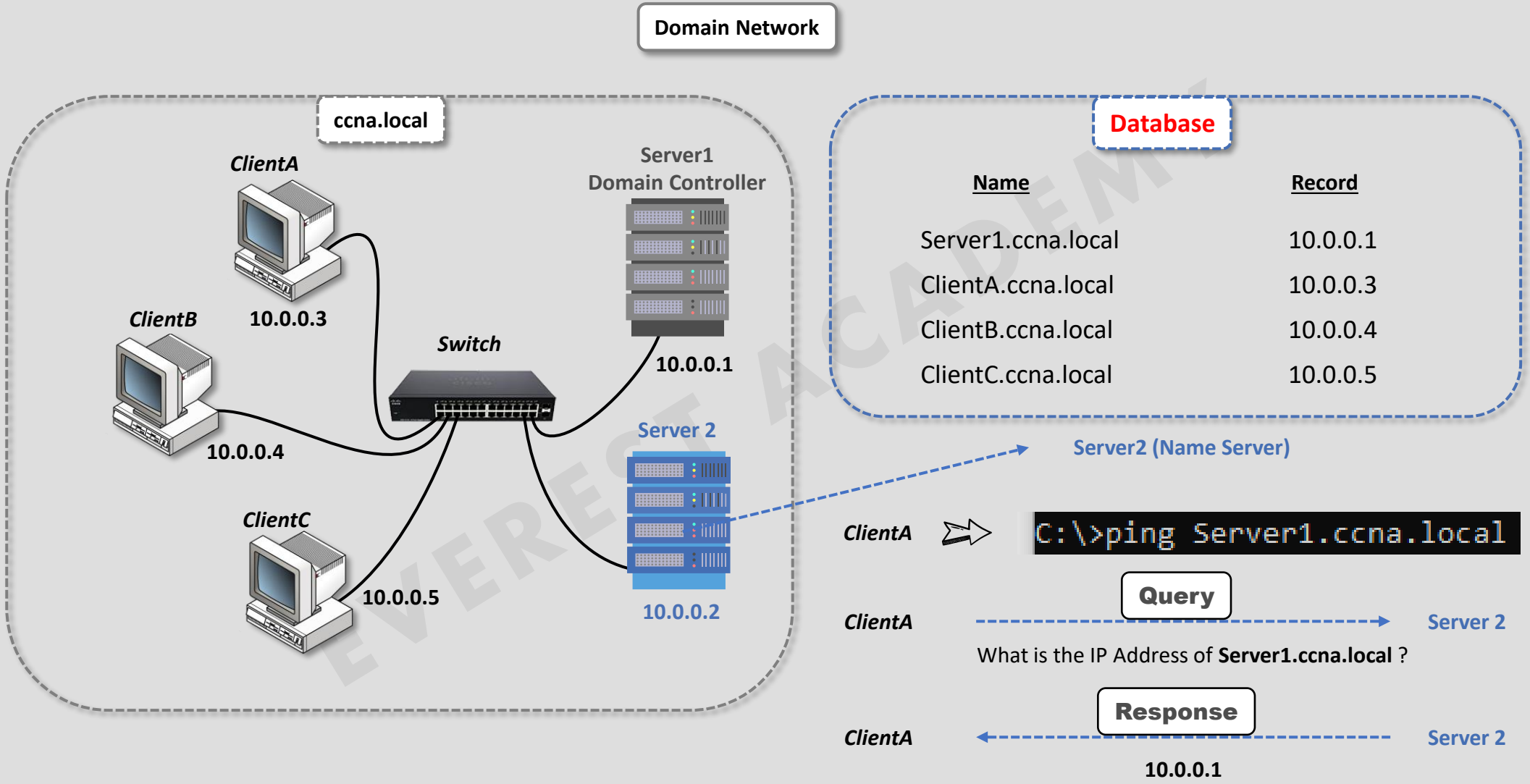




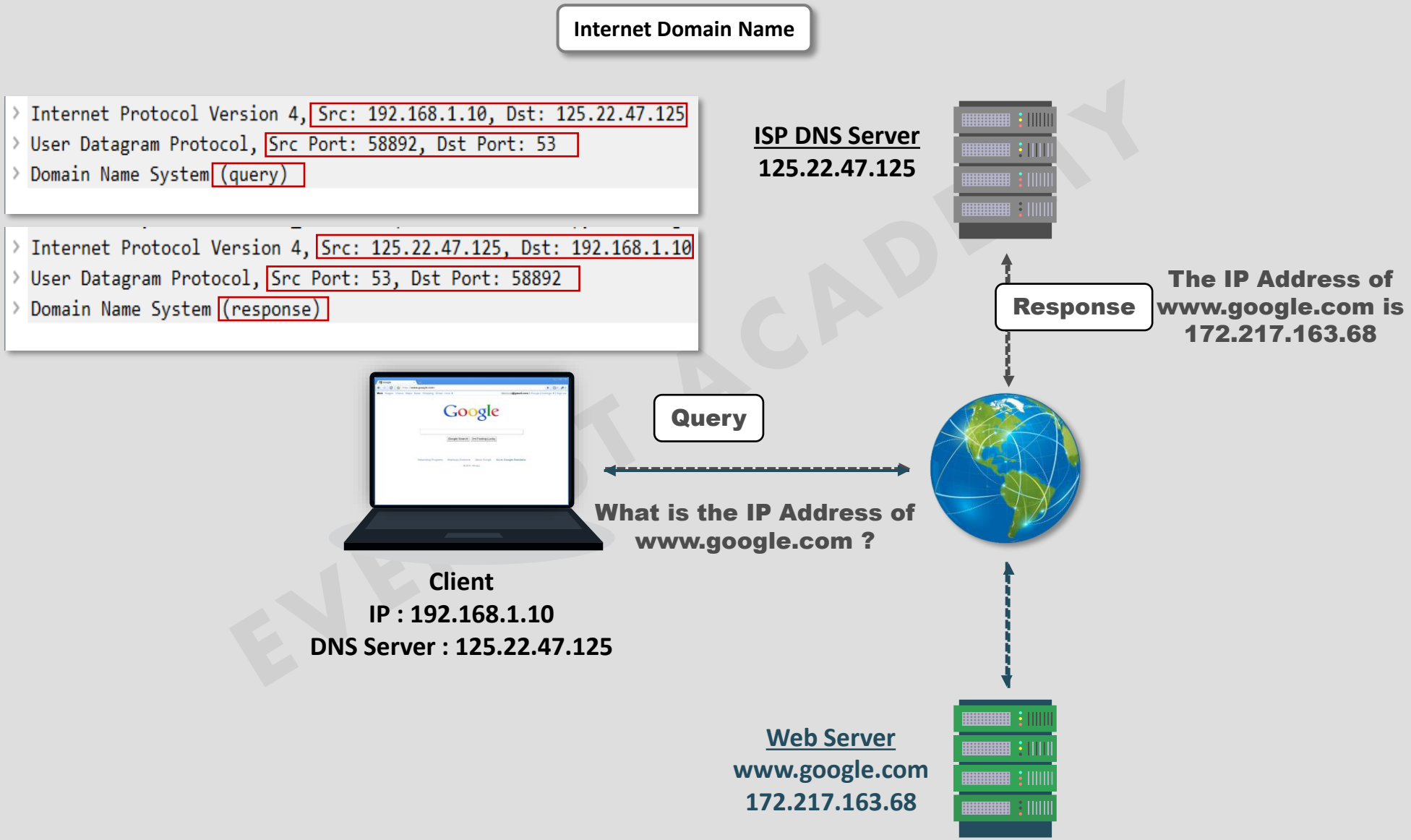
Local Disk (C:) › Windows › System32 › drivers › etc

```
C:\>ping ClientA
C:\>ping ClientB
C:\>ping ClientC
```

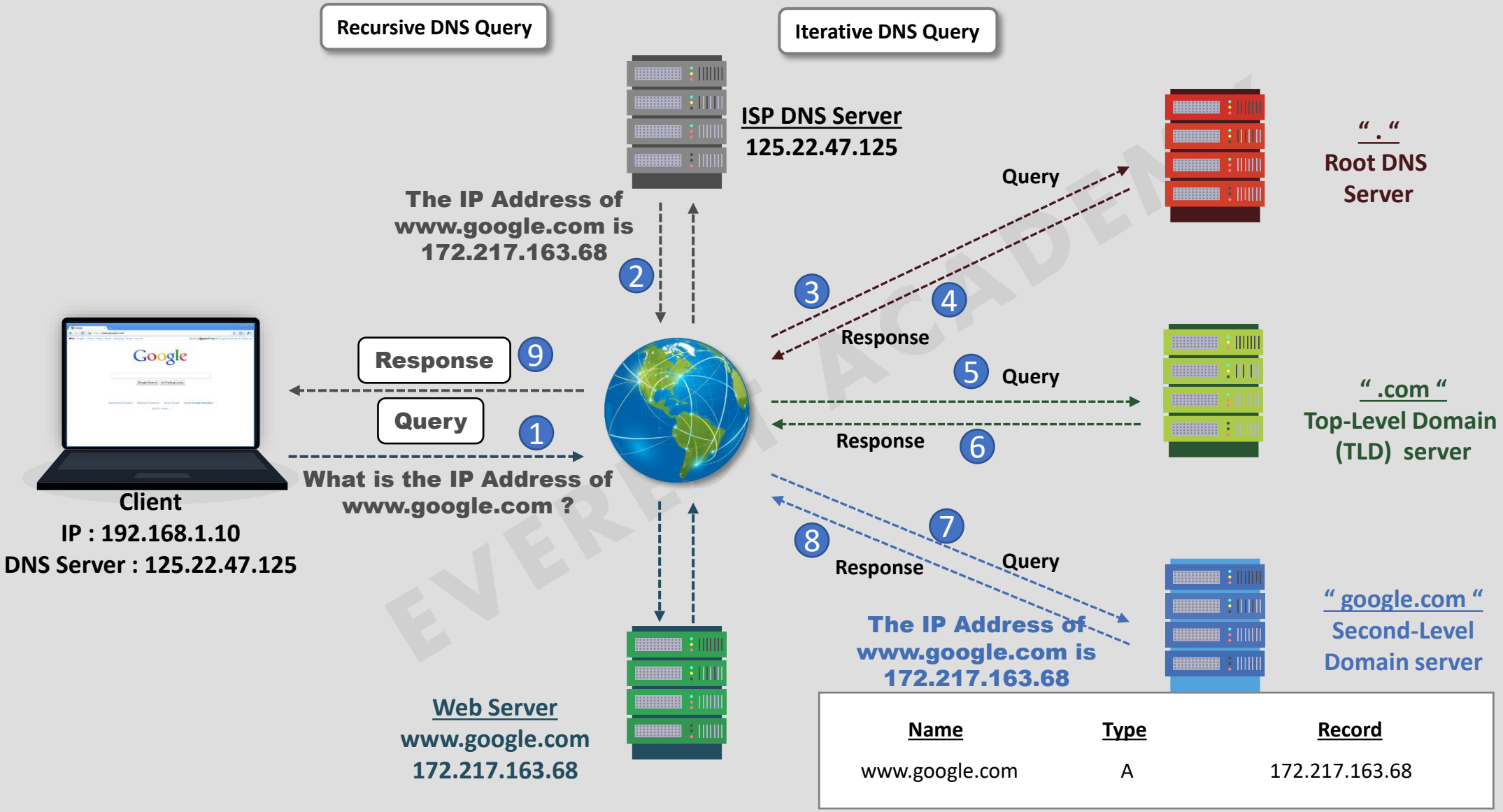
Domain Network System (DNS) – Private Network



Domain Network System (DNS) – Messages



Iterative And Recursive DNS Query



DNS Cache

A DNS Cache is a temporary database, maintained by a computer's operating system, that contains records of all the recent visited websites and other internet domains.

The DNS Cache attempts to speed up the process even more by handling the name resolution of recently visited addresses before the request is sent out to the internet.

The contents of a local DNS cache can be viewed on Windows using the command ipconfig /displaydns.

The contents of a local DNS cache can be flushed on Windows using the command ipconfig /flushdns.

```
C:\>ping www.google.com

Pinging www.google.com [172.217.163.68] with 32 bytes of
Reply from 172.217.163.68: bytes=32 time=12ms TTL=119
Reply from 172.217.163.68: bytes=32 time=11ms TTL=119
Reply from 172.217.163.68: bytes=32 time=12ms TTL=119
Reply from 172.217.163.68: bytes=32 time=12ms TTL=119

Ping statistics for 172.217.163.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 12ms, Average = 11ms
```

```
C:\>ipconfig /displaydns


    www.google.com
    ----------------------------------------
    Record Name . . . . . : www.google.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 218
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 172.217.163.68
```

```
C:\>ipconfig /flushdns
```

The Most Common Types Of DNS Record

1	**"A" Records:** These records map domain names to traditional (IPv4) addresses.	**Example:** www.google.com. 172.217.163.68
2	**"CNAME" Records:** A Canonical Name or CNAME record is a type of DNS record that maps an alias name to a true or canonical domain name.	**Example:** www.google.in. www.google.com.
3	**"TXT" Records:** TXT records are used for Sender Policy Framework (SPF) codes , and for ownership verification of a domain.	**Example:** everestacademy.in google-site-verification=GhBc3Ty5GfaF
4	**"MX" Records:** MX records define the Mail exchange records for a domain, or where inbound mail for a domain should get directed.	**Example:** @ or everestacademy.in mail.everestacademy.in.
5	**"NS" Records:** NS records specify the name servers used by a domain.	**Example:** google.com. ns1.google.com
6	**"PTR" Records:** PTR records map IP addresses into domain names.	**Example:** 172.217.163.68.in-addr.arpa. www.google.com

Hypertext Transfer Protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is an application layer protocol used by clients and web servers to transfer the data in the form of plain text, audio, video, image and so on.

The messages sent by the client are called requests and the messages sent by the server as an answer are called responses.

HTTP is a "**stateless**" protocol which means each time a client connects to the server the client opens a separate connection to the Web server .

HTTP relies on **TCP** to send and receive data. The default port is **TCP 80**,

Hypertext Transfer Protocol (HTTP)

Web Client
192.168.1.6

SYN

SYN + ACK

ACK

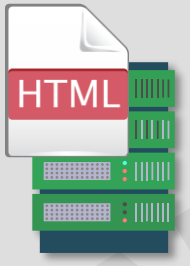
GET : /index.html

Data : Segment 1

Data : Segment 2

HTTP Status Code : 200 ok

ACK



Web Server
192.168.1.10
Port = 80

```
index.html
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <title>Document</title>
6   </head>
7   <body>···
10  </body>
11  </html>
```

Source	Destination	Protocol	Len	Info
192.168.1.6	192.168.1.10	TCP	0	63853 → 80 [SYN]
192.168.1.10	192.168.1.6	TCP	0	80 → 63853 [SYN, ACK]
192.168.1.6	192.168.1.10	TCP	0	63853 → 80 [ACK]
192.168.1.6	192.168.1.10	HTTP	343	GET /index.html HTTP/1.1
192.168.1.10	192.168.1.6	TCP	1460	80 → 63853 [ACK]
192.168.1.10	192.168.1.6	TCP	1460	80 → 63853 [ACK]
192.168.1.10	192.168.1.6	HTTP	284	HTTP/1.1 200 OK (text/html)
192.168.1.6	192.168.1.10	TCP	0	63853 → 80 [ACK]

TCP 3-way Handshake

Http version
URL
Method

The request has succeeded

Hypertext Transfer Protocol (HTTP)

TCP
Connection Termination

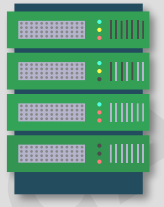
Web Client
192.168.1.6

FIN + ACK

ACK

FIN + ACK

ACK



Web Server
192.168.1.10
Port = 80

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>···
</body>
</html>
```

Source	Destination	Protocol	Len	Info
192.168.1.10	192.168.1.6	TCP	0	80 → 63853 [FIN, ACK]
192.168.1.6	192.168.1.10	TCP	0	63853 → 80 [ACK]
192.168.1.6	192.168.1.10	TCP	0	63853 → 80 [FIN, ACK]
192.168.1.10	192.168.1.6	TCP	0	80 → 63853 [ACK]

Hypertext Transfer Protocol Secure (HTTPs)

Hypertext Transfer Protocol Secure (HTTPs) is a secure version of HTTP. This protocol enables secure communication between the web client and web server.

HTTPS uses Transport Layer Security (**TLS**) protocol or its predecessor Secure Sockets Layer (**SSL**) for encryption.

HTTPS URLs begin with https instead of http and the web client uses the default port **443** when sends a request to the web server.

https://www.google.com:*443*/index.html

URL (Uniform Resource Locator)

Source	Destination	Protocol
151.101.157.108	192.168.1.10	TLSv1.2
172.217.166.110	192.168.1.10	TLSv1.3

URL and URI

URL (Uniform Resource Locator) is used to uniquely identify a resource over the web.

URL has the following syntax:
protocol://hostname:port/*path-and-file-name*.

Protocol: HTTP, FTP, telnet and so on.

Hostname: The DNS domain name.

Port: The TCP port number that the server is listening for incoming requests from the clients.

Path-and-file-name: The name and location of the requested resource.

http://www.google.com/index.html
ftp://ftp.hp.com/pub/docs/soar_en_cust_contact.html
mailto:user@gmail.com

URI (Uniform Resource Identifier) is more general than **URL**, which can even locate a fragment within a resource.

URI has the following syntax for HTTP:
http://hostnamet:port/path?*request-parameters#nameAnchor*

The request parameters, in the form of name=value pairs, are separated from the URL by a '**?**'. The name=value pairs are separated by a '**&**'.

The #nameAnchor identifies a fragment within the HTML document, defined via the **anchor tag <a** name="anchorName"**>....**

http://info.my.org/AboutUs/Phonebook
https://www.google.com/search**?**client=firefox-b-d**&**q=abcd
http://www.myu.edu/org/admin/people**#**andy

File Transfer Protocol (FTP)

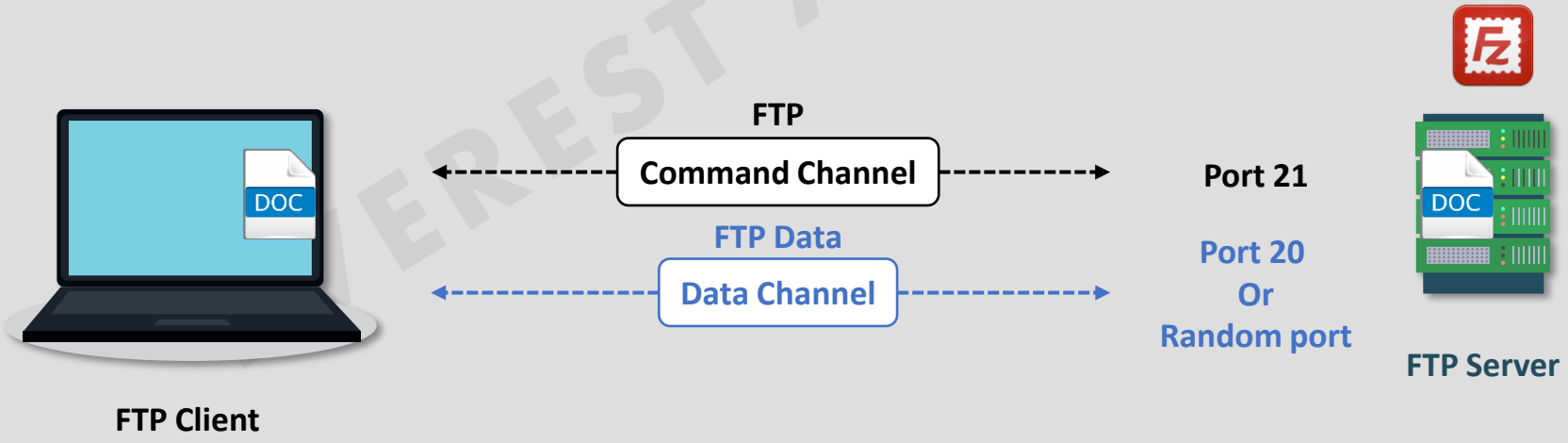
The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP Client works in two different modes:

1. **Active Mode :** uses TCP port numbers 20 and 21.
2. **Passive Mode :** uses TCP port number 21 with a random number.

FTP is built on a **client-server** model architecture using separate control and data connections between the client and the server.

Passive Mode is used if there is a Firewall device in the network, In this Mode the server uses a random port number for Data Channel.



File Transfer Protocol (FTP)

FTP Client

FTP

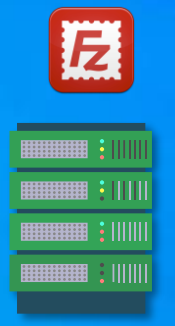
Command Channel

FTP Data

Data Channel

Port 21

Port 20
Or
Random port



FTP Server

11:06 AM
2/11/2021

Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (TFTP) is a network protocol used to transfer files between hosts such as configuration files.

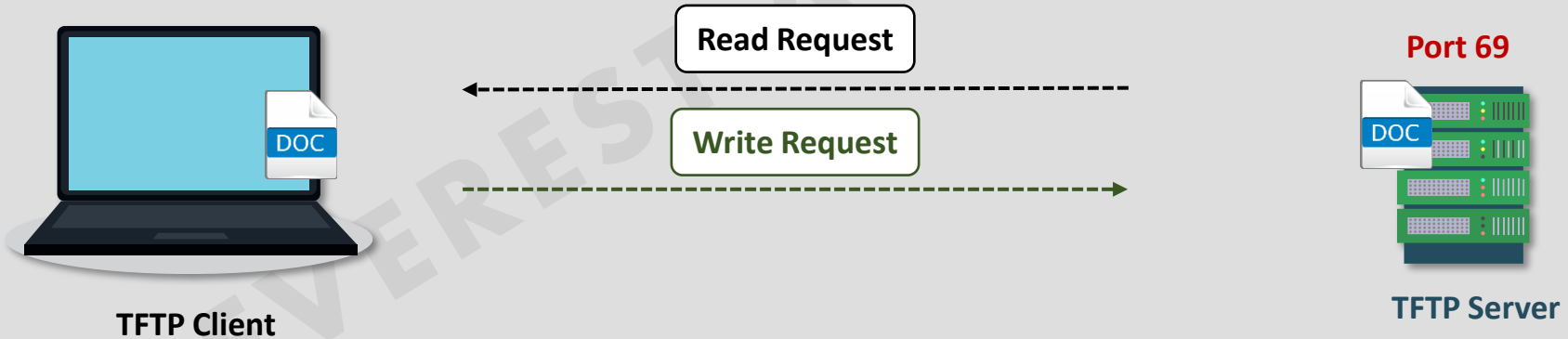
TFTP can not list, delete, or rename files or directories on a remote server.

TFTP doesn't support user authentication and encryption.

TFTP itself takes care of reliability by requiring the peer to acknowledge each successfully received block.

Cisco does still use it on its devices for backing up router **configuration files** and its **IOS images.**

TFTP uses **UDP** port **69.**



Differences Between FTP and TFTP

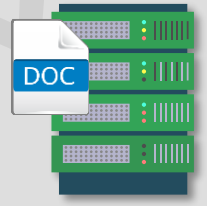
FTP	TFTP
FTP uses TCP connection to transfer data.	TFTP uses UDP connection to transfer data.
TCP connection uses ports 20 and 21 .	UDP connection uses port 69 .
FTP needs authentication for communication.	TFTP does not need authentication for communication.
FTP allows bidirectional transfer of files .	TFTP allows only unidirectional transfer of files.
TCP provides the acknowledgement and retransmission of data.	TFTP provides the acknowledgement and retransmission of data since it uses UDP.
FTP used to upload and download files by remote users.	TFTP used to transfer configuration files and its IOS images.

TFTP Messages

FTP Client
192.168.1.2

Read Request
Option Acknowledgment
Acknowledgment
Data Packet
Acknowledgment
Write Request
Option Acknowledgment
Data Packet
Acknowledgment
Read Request
Error Code

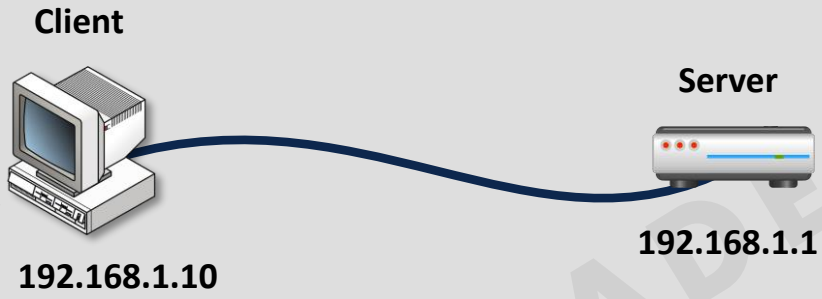
Port 69

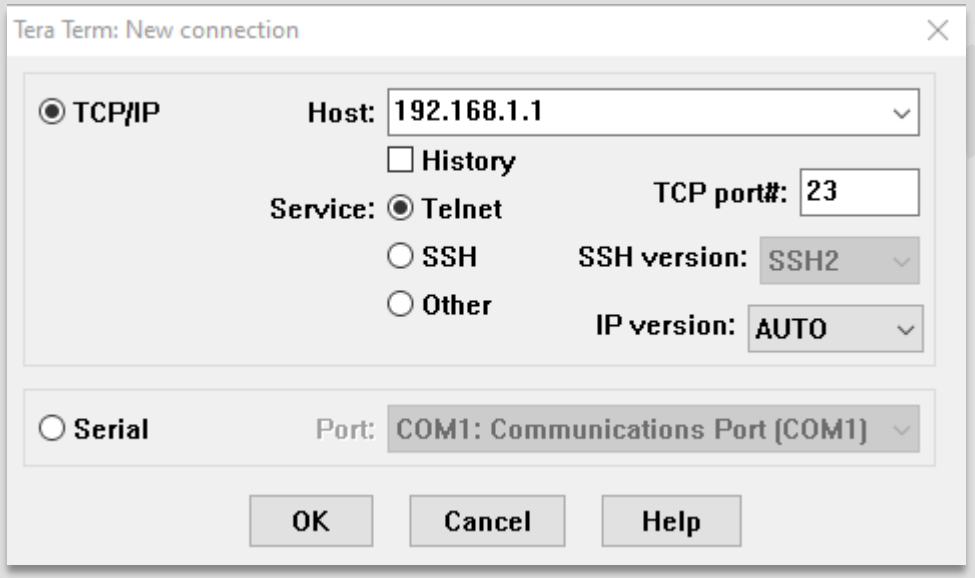


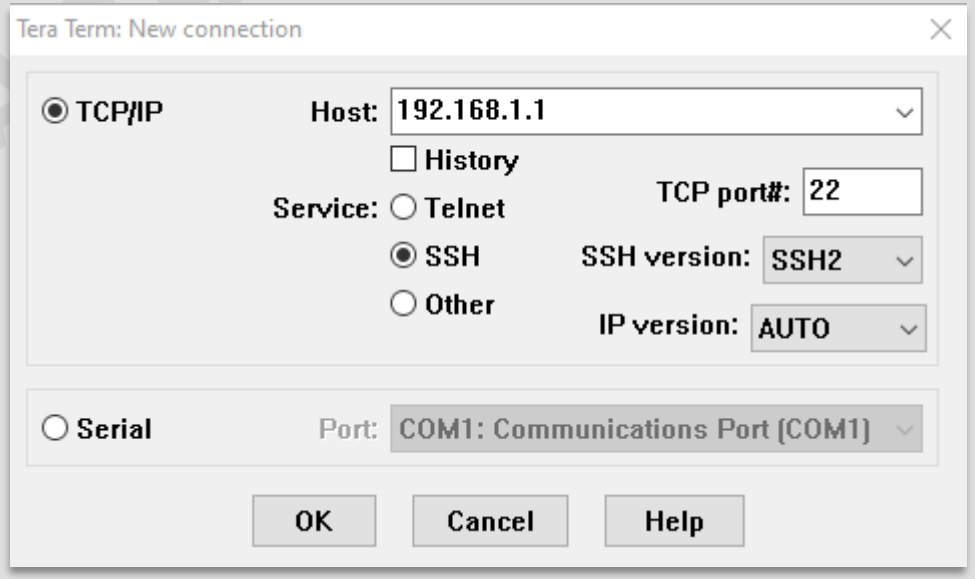
FTP Server
192.168.1.10

Source	Destination	Protocol	Info
192.168.1.2	192.168.1.10	TFTP	Read Request, File: ServerFile.txt,
192.168.1.10	192.168.1.2	TFTP	Option Acknowledgement,
192.168.1.2	192.168.1.10	TFTP	Acknowledgement,
192.168.1.10	192.168.1.2	TFTP	Data Packet,
192.168.1.2	192.168.1.10	TFTP	Acknowledgement,
192.168.1.2	192.168.1.10	TFTP	Write Request, File: ClientFile.txt,
192.168.1.10	192.168.1.2	TFTP	Option Acknowledgement,
192.168.1.2	192.168.1.10	TFTP	Data Packet,
192.168.1.10	192.168.1.2	TFTP	Acknowledgement, Block: 1
192.168.1.2	192.168.1.10	TFTP	Read Request, File: MyFile.txt,
192.168.1.10	192.168.1.2	TFTP	Error Code, Code: File not found

Telnet and SSH





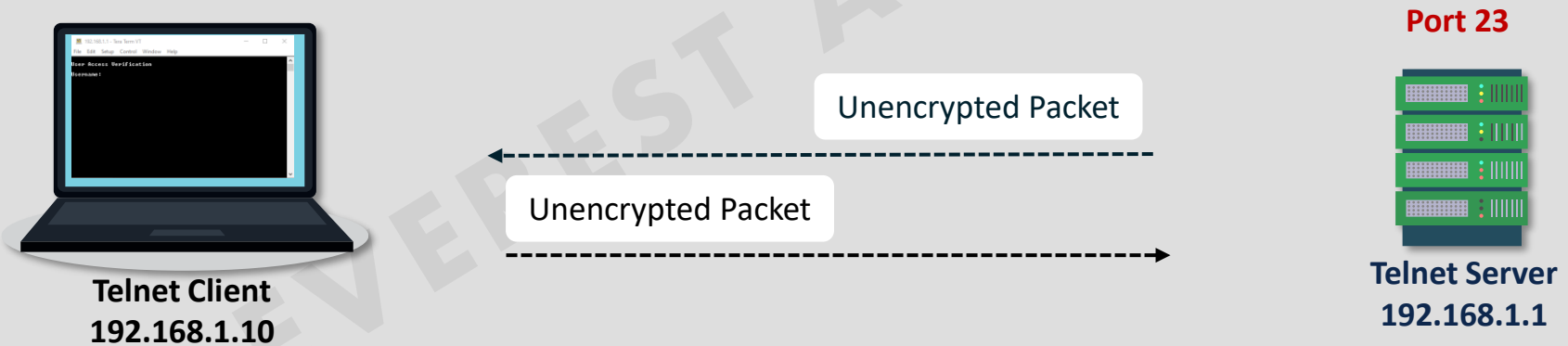


Telnet Protocol

Telnet is an application protocol that allows a user to communicate with a remote device using **command-line interface (CLI).**

Telnet Client connect to **Telnet server** using **TCP** port **23**.

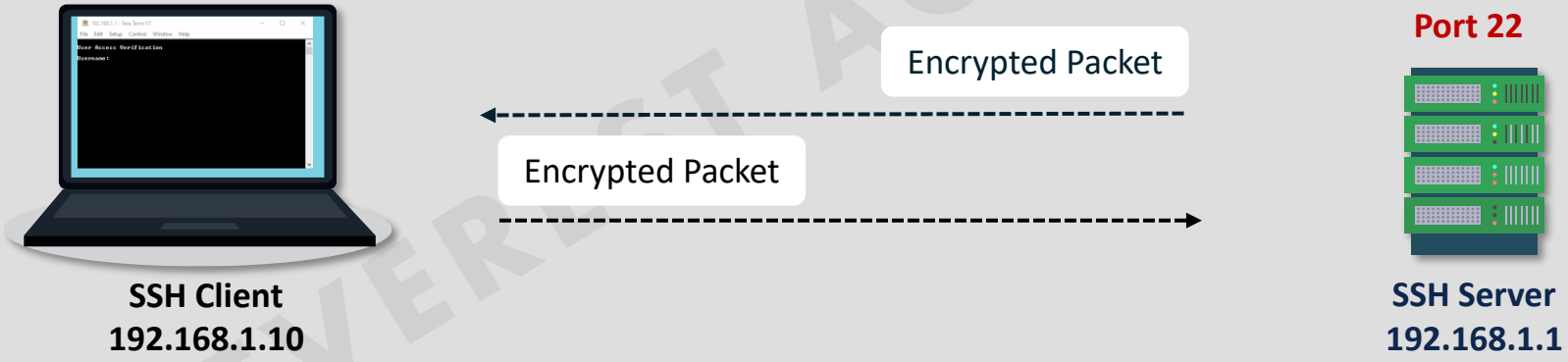
Telnet protocol is unsecure because it sends and receive data in **plain text**.



Secure Shell (SSH)

Secure Shell (SSH) is a cryptographic network protocol that provides secure **remote control** and secure **file transferer** over an unsecured network.

Secure Shell (SSH) uses TCP port 22.



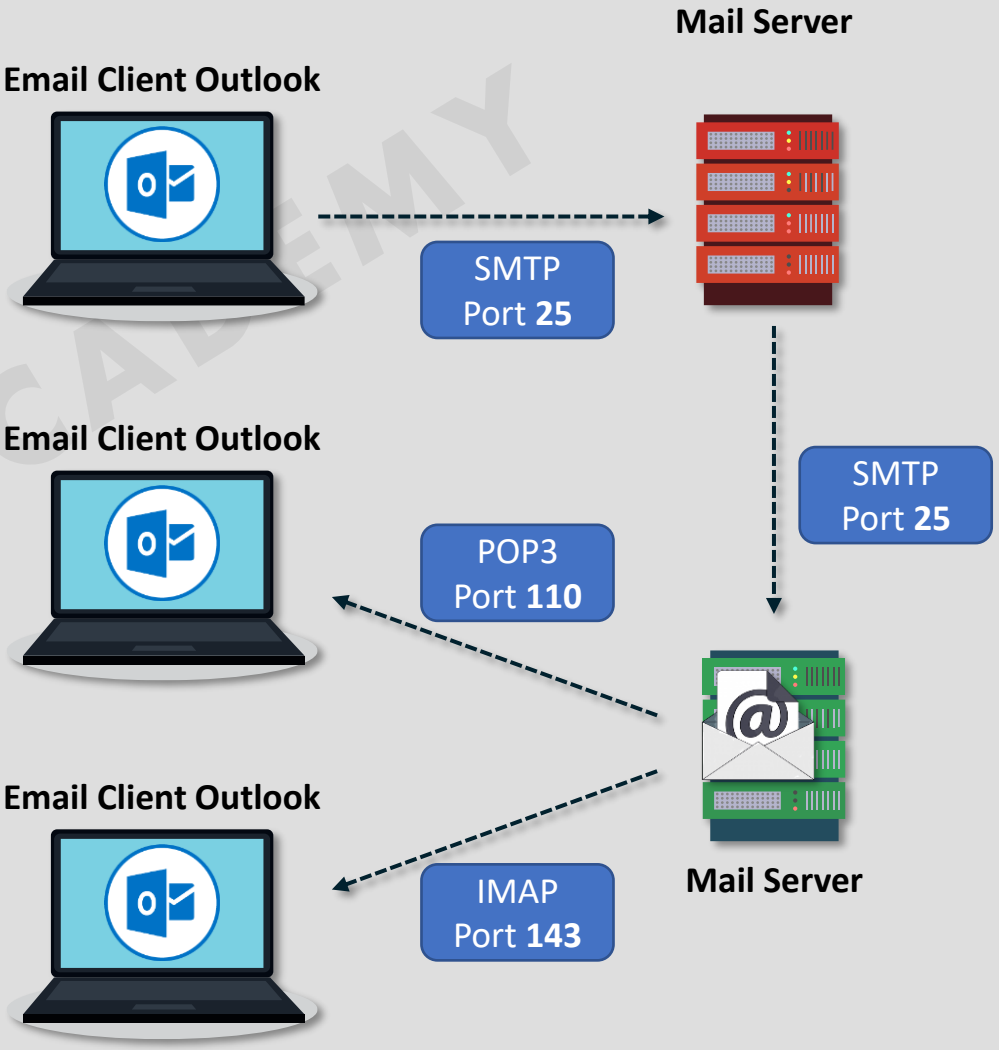
SMTP, POP3 and IMAP

- **Simple Mail Transfer Protocol (SMTP)** is an application protocol used to deliver an email from a client to an email server or from one email server to another. It uses **port 25.**

- **Post Office Protocol (POP3) version 3** is an application protocol that allows user to download an email from an email server and delete it from the server, It uses **port 110**.

- **Internet Message Access Protocol (IMAP)** is a protocol that allows user to download an email from an email server without deleting it from the server, It uses **port 143**,

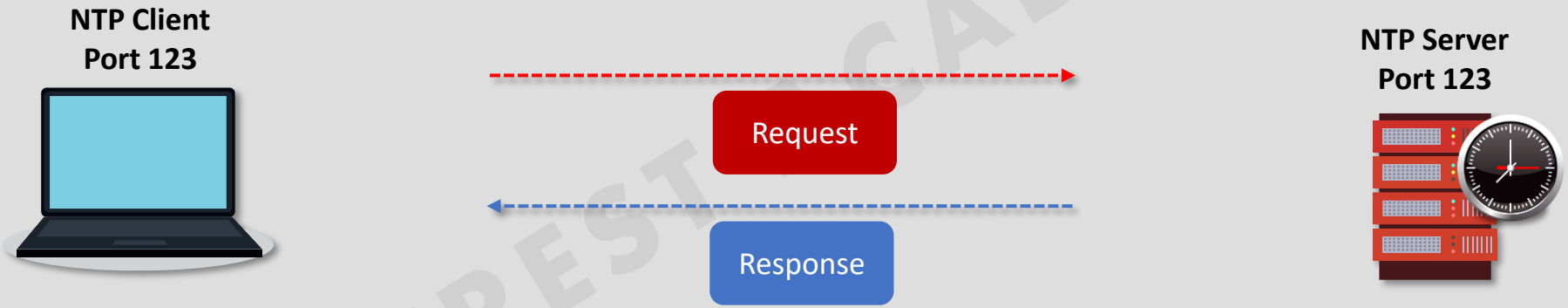
- **hMailServer** is a free email server for Windows. It includes administration tools for management and backup. It has support for **IMAP**, **POP3**, and **SMTP**.



Network Time Protocol (NTP)

- **Network Time Protocol (NTP)** is an application layer protocol used for clock synchronization between network devices.

- **NTP** uses a **client-server architecture**; one host is configured as the **NTP server** and all other hosts are configured as **NTP clients**.
- It uses **UDP port 123** for the **client** and the **server**.



Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is an application layer protocol used to distribute network configuration parameters, such as **IP address**, **subnet mask**, **default gateway**, **DNS servers**, etc.

DHCP Protocol has four different messages, **Discover**, **Offer**, **Request** and **Acknowledgement**.

DHCP Protocol uses **UDP** port **68** for the client and **UDP** port **67** or the server.

Source	Destination Broadcast	Protocol	Source Port	Destination Port	Info
0.0.0.0	255.255.255.255	DHCP	68	67	DHCP Discover -
192.168.1.1	255.255.255.255	DHCP	67	68	DHCP Offer -
0.0.0.0	255.255.255.255	DHCP	68	67	DHCP Request -
192.168.1.1	255.255.255.255	DHCP	67	68	DHCP ACK -

Dynamic Host Configuration Protocol (DHCP)

```
Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x00002455
    Seconds elapsed: 0
  > Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.1.2
```

Source	Destination Broadcast	Protocol	Source Port	Destination Port	Info
0.0.0.0	255.255.255.255	DHCP	68	67	DHCP Discover -
192.168.1.1	255.255.255.255	DHCP	67	68	DHCP Offer -
0.0.0.0	255.255.255.255	DHCP	68	67	DHCP Request -
192.168.1.1	255.255.255.255	DHCP	67	68	DHCP ACK -

Source	Destination	Protocol	Source Port	Destination Port	Info
0.0.0.0	255.255.255.255	DHCP	68	67	DHCP Discover -
192.168.1.1	192.168.1.4	DHCP	67	68	DHCP Offer -
0.0.0.0	255.255.255.255	DHCP	68	67	DHCP Request -
192.168.1.1	192.168.1.4	DHCP	67	68	DHCP ACK -

Dynamic Host Configuration Protocol (DHCP)