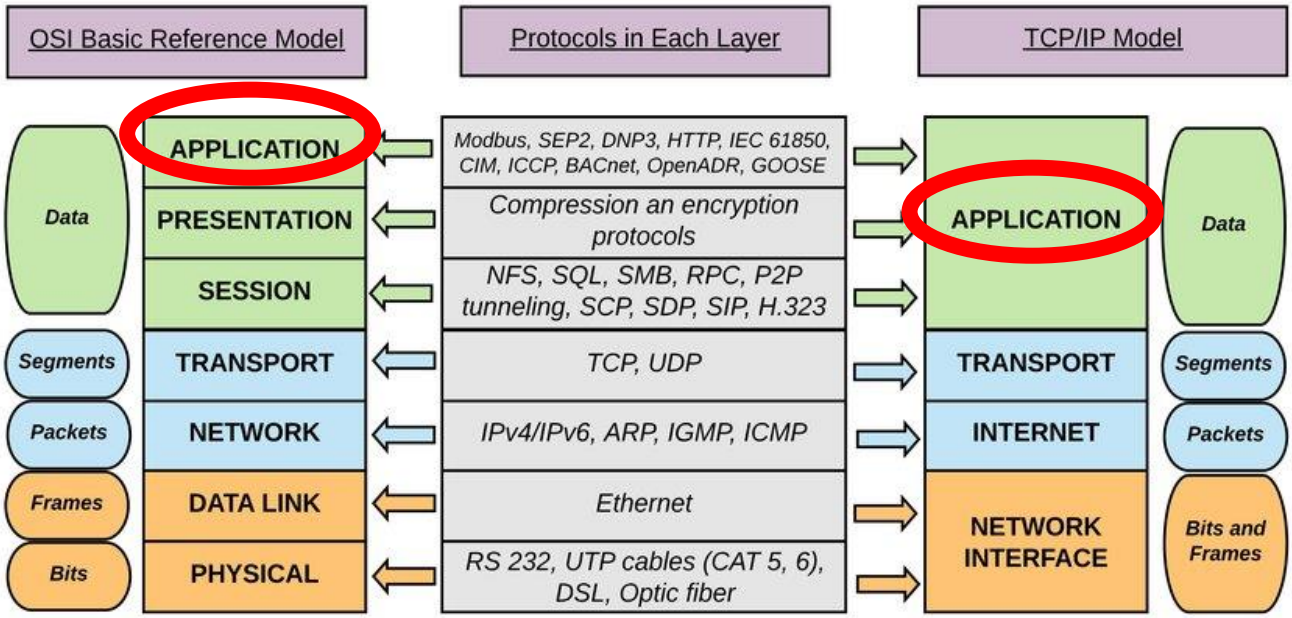
1. The main difference between OSI model and TCP/IP model

OSI Model	TCP/IP Model
It stands for **Open System Interconnection.**	It stands for **Transmission Control Protocol.**
It is an independent standard and generic protocol used as a communication gateway between the network and the end user.	It consists of standard protocols that lead to the development of an internet. It is a communication protocol that provides the connection among the hosts.
In the OSI model, the transport layer provides a guarantee for the delivery of the packets.	The transport layer does not provide the surety for the delivery of packets.we can say that it is a reliable model.
This model is based on a vertical approach.	This model is based on a horizontal approach.
The session and presentation layers are separated, i.e., both the layers are different.	Both session and presentation layers are included in the application layer.
It is also known as a reference model through which various networks are built. For example, the TCP/IP model is built from the OSI model	It is an implemented model of an OSI model.
In this model, the network layer provides both connection-oriented and connectionless service.	The network layer provides only connectionless service.
Protocols in the OSI model are hidden and can be easily replaced when the technology changes.	In this model, the protocol cannot be easily replaced.
It consists of 7 layers.	It consists of 4 layers.
OSI model defines the services, protocols, and interfaces as well as provides a proper distinction between them. It is protocol independent.	In the TCP/IP model, services, protocols, and interfaces are not properly separated. It is protocol dependent.
The usage of this model is very low.	This model is highly used.
It provides standardization to the devices like router, motherboard, switches, other hardware devices.	It does not provide the standardization to the devices. It provides a connection between various computers.

2. The logical mapping between OSI basic reference model and the TCP/IP stack



The Application Layer is the seventh layer of the seven-layers OSI model. Application layer interface directly interacts with the application and provides common web application services. The application layer also makes a request to the presentation layer.

Application layer is the highest level of open systems, providing services directly for the application process.

Application layer functions

- Transport access and management :it allows a user to access, retrieve and manage files in a remote computer.
- Mail services : It provides the basis for email forwarding and storage facilities.
- Virtual terminal : One of The OSI is to define a virtual terminal that is really just an abstract data structure that takes the abstract state of the actual terminal.
- Other functions : In addition to the three functions above, there are some other functions : directory services, remote job entry, graphics, information communication and so on.

Application layer examples

- DNS (Domain Name System)
- HTTP (Hypertext Transfer Protocol)
- FTP (File Transfer Protocol)

The application service layer

The application service layer establishes the ground level foundation that exists to express technology-specific functionality. Services that reside within this layer can be referred to simply as application services. Their purpose is to provide reusable functions related to processing data within new or legacy application environments.

3. What happen when writing the url and click enter ?

We traced a URL request from the browser all the way to the server hosting it and it's response back to the browser to be rendered. We covered the relationship between websites, servers, IP addresses and stepped through each of the steps that the browser goes through when we type a URL into the browser and press enter. For review, here are those six steps:

1. We type a URL in the browser and press Enter (e : www.google.com)
2. Browser looks up IP address for the domain (e : 8.8.8.8)
3. Browser initiates TCP connection with the server
4. Browser sends the HTTP request to the server
5. Server processes request and sends back a response
6. Browser renders the content

DNS (The Domain Name System)

The Domain Name System (DNS) protocol is an important part of the web's infrastructure, serving as the Internet's phone book: every time you visit a website, your computer performs a DNS lookup. Complex pages often require multiple DNS lookups before they start loading, so your computer may be performing hundreds of lookups a day. The DNS maps human-readable domain names (in URLs or in email address) to IP addresses. For example, DNS translates and maps the domain www.google.com to the IP address 8.8.8.8.

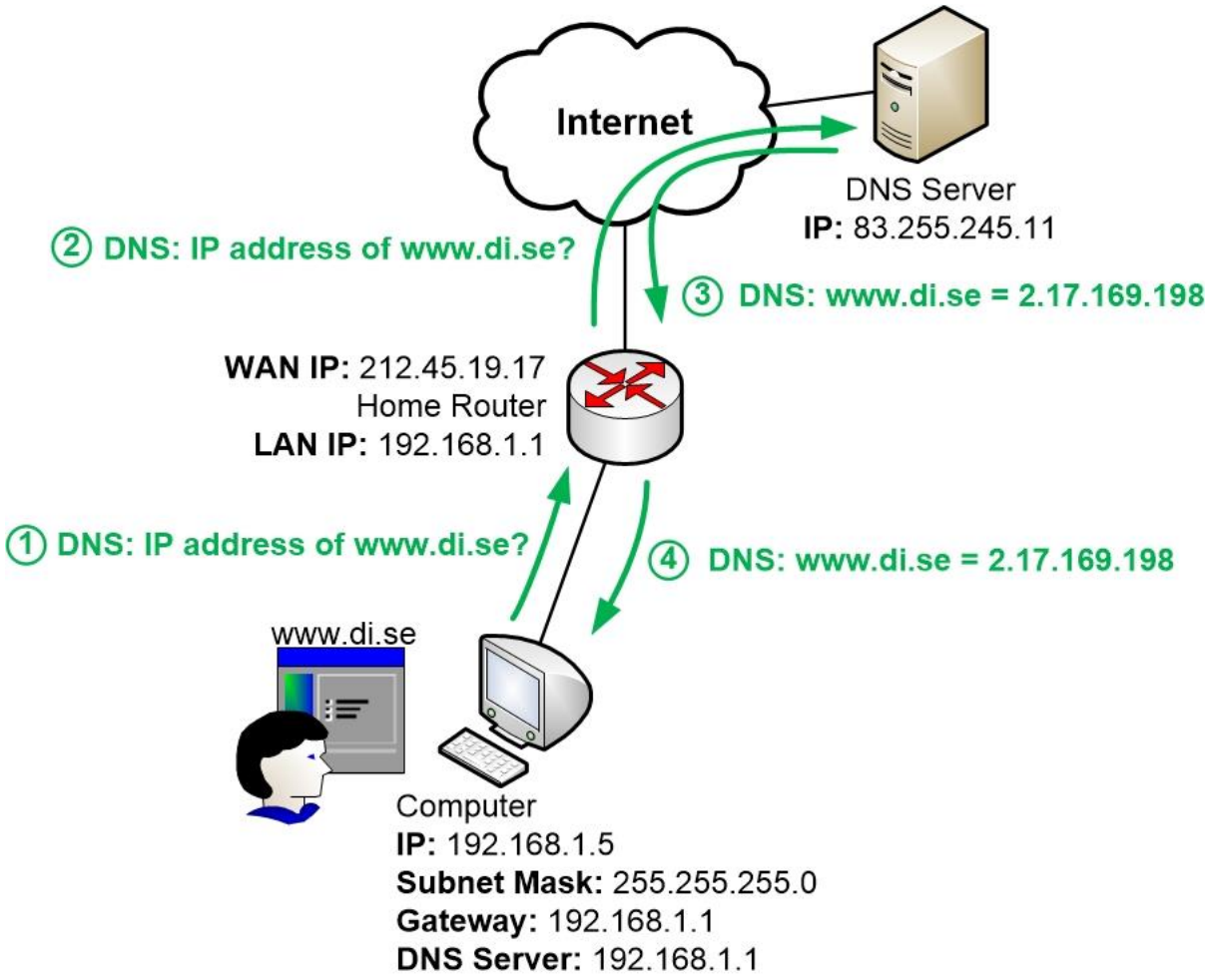
The relation between the DNS and the @IP address:

For a computer to be able to look up which IP address that a particular domain name has got, the computer must first find its way to a DNS server. The computer can obtain this information from the home router, then the router also passes along information about which DNS server that the computer should use.

When a computer wants to browse to a domain name it queries the DNS server for what IP address that domain name has got. Once it gets a DNS response

back containing the IP address of the domain name it can use that IP address as its destination for the traffic.

There are also home routers that simply tell the computers on the LAN to send their DNS queries directly to the DNS servers of the ISP.

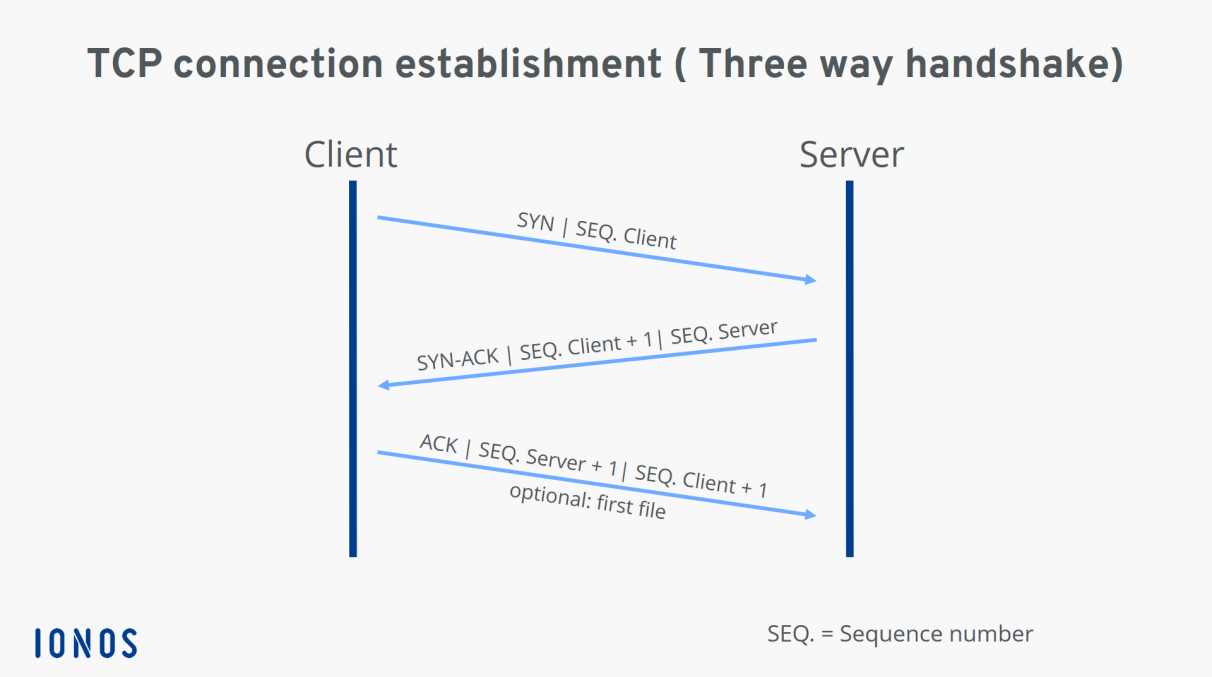


4. TCP Interconnection Client/Server

For establishing a valid TCP connection: Both endpoints (our Localhost and the Server) must already have a **unique IP address** (IPv4 or IPv6) and have assigned and enabled the **desiredport** for data transfer. The IP address serves as an identifier, whereas the port allows the operating system to assign connections to the specific

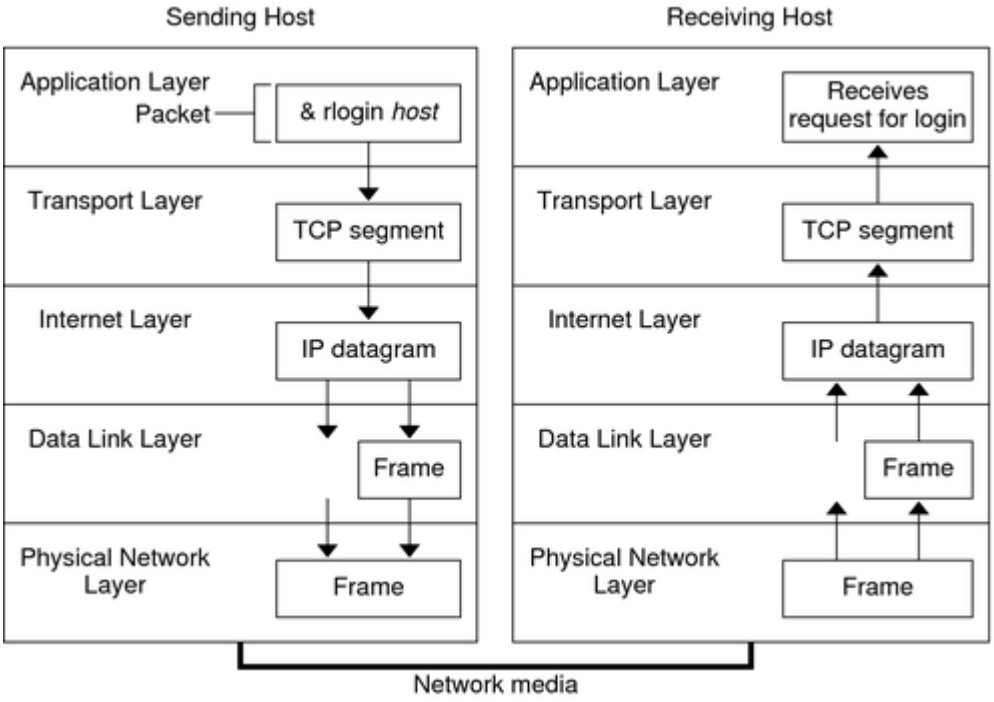
client and server applications. The **process** for establishing a connection with the TCP protocol is as follows:

1. First, the localhost sends the server a **SYN packet** or segment (SYN stands for synchronize) with a unique, random number. This number ensures full transmission in the correct order (without duplicates).
2. If the server has received the segment, it agrees to the connection by returning a **SYN-ACK packet** (ACK stands for acknowledgment) including the client's sequence number plus 1. It also transmits its own sequence number to the client.
3. Finally, the client (the localhost) acknowledges the receipt of the SYN-ACK segment by sending its own ACK packet, which in this case contains the server's sequence number plus 1. At the same time, the client can already begin transferring data to the server.



How Data transfer over Internet (TCP Packet)

The packet is the basic unit of information that is transferred across a network. The basic packet consists of a header with the sending and receiving systems' addresses, and a body, or payload, with the data to be transferred. As the packet travels through the TCP/IP protocol stack, the protocols at each layer either add or remove fields from the basic header. When a protocol on the sending system adds data to the packet header, the process is called data encapsulation. Moreover, each layer has a different term for the altered packet, as shown in the following figure.



The life cycle starts when you issue a command or send a message. The life cycle finishes when the appropriate application on the receiving system receives the packet.

Presented By:

Oussama Bessaad