## 3.2.1 Protocols

**Protocol** – a set of rules governing the way that devices communicate with each other

With networks and the internet, we need to allow computers to talk to each other. The computers must talk to each other in a way that the receiving end can understand the message. In order for that to happen, there are sets of rules governing modes of communication. These rules are called *protocols*. There are many different protocols out there, each defining rules for specific communication types.

Examples include: **FTP**, **HTTP**, **POP3**, and **SMTP**

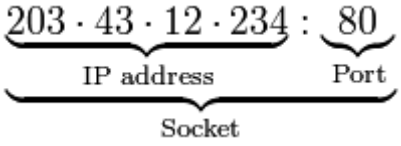
We will look into all of these protocols.

**Port number** – an application endpoint or process specific communication endpoint attached to an IP address

When you send and receive data from a client or server, you will be sending lots of different types of data. To make sure that the data is dealt with by the correct program, for example a website request is dealt with by the web server; sending data over Skype will be received at the other end on Skype and not anywhere else.

In order to ensure this proper activity, you need to add a port number. Each application will have a port number associated with it. For example, a web server is “port 80” and a Counter-Strike game port is “666”

Combining an IP address with a port gives us a **“socket”.** This is a direct connection to a process or application on a machine. The following example is connection to a webserver on address 203.43.12.234



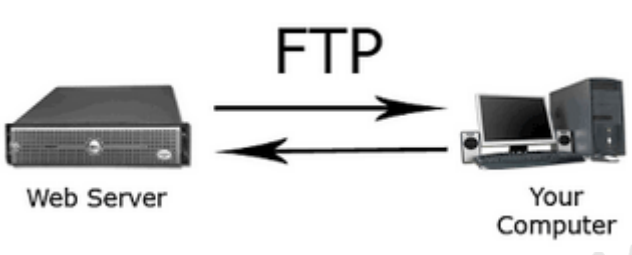

There are many well-known port numbers; here are a few you might want to keep in mind:

| Port number | Protocol that uses it |
|---|---|
| 21 | File Transfer Protocol (FTP) |
| 25 | Simple Mail Transfer Protocol (SMTP) |
| 80 & 8080 | HyperText Transfer Protocol (HTTP) |
| 110 | Post Office Protocol v3 (POP3) |

## 3.2.1 Protocols

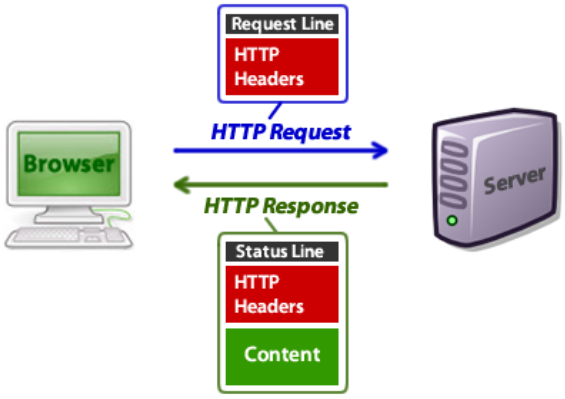
### FTP

File transfer Protocol (FTP) is a standard network protocol used to copy a file from one host to another over a TCP/IP-based network, such as **The Internet**. By copying from one host to another, you could say for example uploading a website with pictures from your PC to your website will require use of FTP. Or the transfer of files over Skype will require FTP. FTP works on port 21




### HTTP

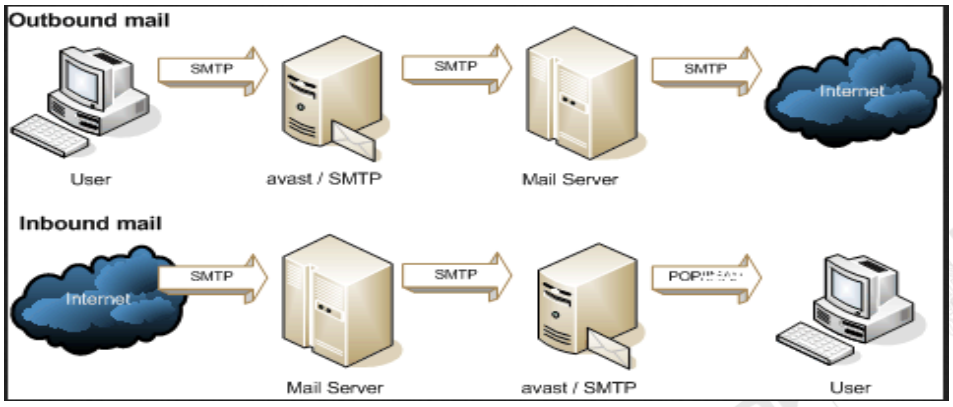
The Hypertext Transfer Protocol (HTTP) is a networking protocol behind the World Wide Web. It allows for users on the web to exchange information found on web pages. HTTP works on ports 80 & 8080



## 3.2.1 Protocols

**POP3 & SMTP**

**Simple Mail Transfer Protocol (SMTP)** is an internet standard for sending e-mail across networks. SMTP is specified for "outgoing" mail transport and uses port 25. The protocol for receiving mail is called **Post Office Protocol 3 (POP3)** and uses port 110.




The main task that each of these protocols do is simple. However, the protocol itself is made up of several modules each of which is responsible for a certain subtask. All the layers together complete the main task and form the protocol. This idea of dividing a protocol into subtasks can be viewed as a stack structure where each subtask is an individual block. We will further elaborate this concept on a protocol called "**TCP/IP**"

It stands for **T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol. It defines how electronic devices (like computers) should be connected over the Internet, and how data should be transmitted between computers.

**TCP – Transmission Control Protocol**

TCP is responsible for breaking data down into small packets before they can be sent over a network, and for assembling the packets again when they arrive.

**IP – Internet Protocol**

IP takes care of the communication between computers. It is responsible for addressing, sending and receiving the data packets over the Internet.


03-111-222-ZAK | AlevelComputer | @zakonweb | zak@zakonweb.com | www.zakonweb.com

### 3.2.1 Protocols

Communication between computers on a network is done through **protocol suits**. The most widely used and available suite is **TCP/IP**. A protocol suit consists of a layered architecture where each layer depicts some functionality which can be carried out by a protocol.

Each layer usually has more than one protocol option to carry out the responsibility that the layer adheres to. TCP/IP is normally considered to be a 4 layer system. The 4 layers are as follows:

1. **Application layer**
2. **Transport layer**
3. **Network layer**
4. **Data link layer**

**Application Layer**:

This is the top layer of TCP/IP. It includes applications or processes that use transport layer protocols to deliver the data to destination computers.

As mentioned above, at each layer there are certain protocol options to carry out the task designated to that particular layer. So, application layer also has various protocols that applications use to communicate with the transport layer. Some of the popular application layer protocols are:

- HTTP (Hypertext transfer protocol)
- FTP (File transfer protocol)
- SMTP (Simple mail transfer protocol)
- SNMP (Simple network management protocol)

## 3.2.1 Protocols

**Transport Layer**:

This layer provides backbone to data flow between two hosts. This layer receives data from the application layer above it. There are many protocols that work at this layer but the 2 most commonly used protocols here are **TCP** and **UDP**

TCP is used where a reliable connection is required while UDP is used in case of unreliable connections.

**TCP** divides the data (coming from the application layer) into proper sized chunks and then passes these chunks onto the network. It acknowledges received packets, waits for the acknowledgment of sent packets and sets timeout to resend the packets if acknowledgements are not received in time. The term **'reliable connection'** is used where it is not desired to lose any information that is being transferred over the network through this connection. So, the protocol used for this type of connection must provide the mechanism to achieve this desired characteristic. For example, while downloading a file, it is not desired to lose any information (bytes) as it may lead to corruption of the downloaded file.

**UDP** provides a comparatively simpler but unreliable service by sending packets from one host to another. UDP does not take any extra measurements to ensure that the data sent is received by the target host or not. The term **'unreliable connection'** is used where minor data loss does not hamper the task being fulfilled through this connection. For example while streaming a video; loss of few bytes of information is acceptable as this does not harm the user experience much.

**Network Layer**:

This layer is also known as the **Internet Layer**. The main purpose of this layer is to organize or handle the movement of data on the network. By movement of data, we generally mean routing of data over the network. The main protocol used at this layer is **IP**. While ICMP and IGMP are also used at this layer.

**Data Link Layer**:

This layer is also known as the **Network Interface Layer**. This layer normally consists of device drivers in the OS and the network interface card attached to the system. Both the device drivers and the network interface card take cake of the communication details with the media being used to transfer the data over the network. In most of the cases, this media is in the form of **cables**. Some of the famous protocols that are used at this layer include **ARP** (Address resolution protocol) and **PPP** (Point to point protocol) etc.

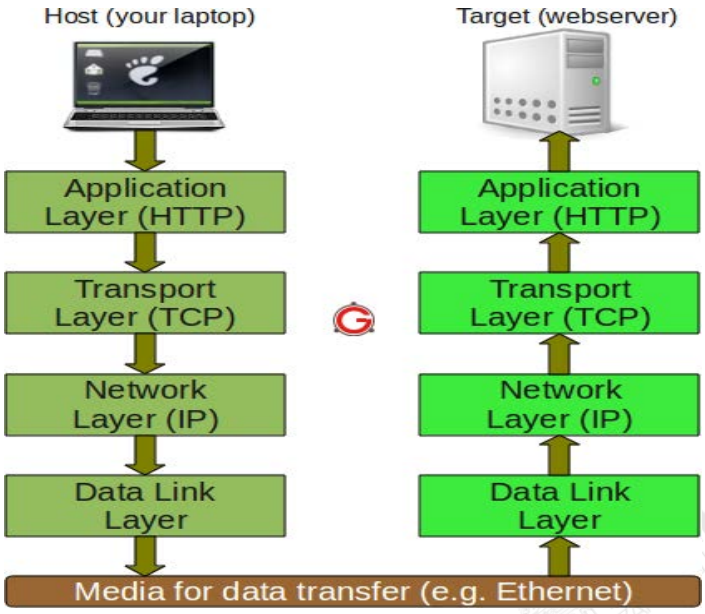
## 3.2.1 Protocols

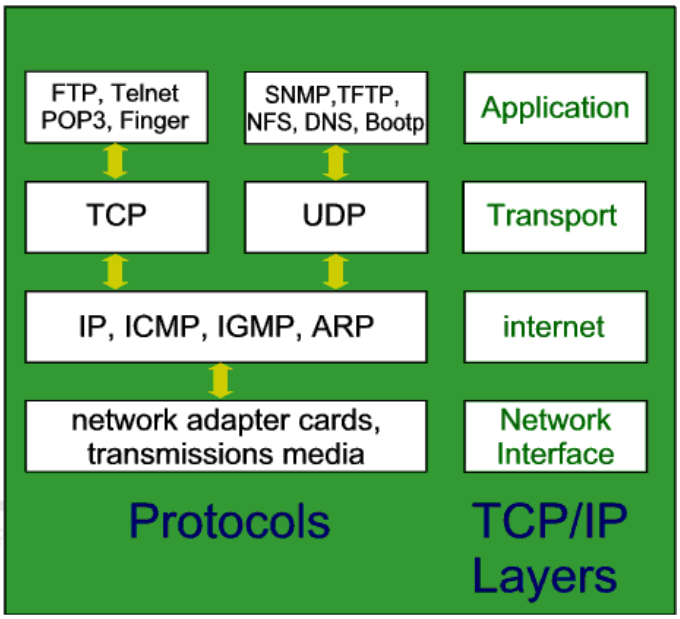
**TCP/IP CONCEPT EXAMPLE**

Now, since we have discussed the underlying layers which help the data flow from host to target over a network. Let's take a simple example to make the concept clearer.

Consider the data flow when you open a website.

**Client requests for a service while the server processes the request for the client.**



### 3.2.1 Protocols

As shown in the figure above, the data flows downward through each layer on the host machine.

- At the first layer, since **'http'** protocol is being used, so an HTTP request is formed and sent to the transport layer.
- Here the protocol **TCP** assigns some more information (sequence number, source port number, destination port number, etc.) to the data coming from the upper layer so that the communication remains reliable
  (I.e. a track of sent and received data could be maintained)
- At the next lower layer, IP adds its own information over the data coming from the transport layer. This information would help data travelling over the network.
- Lastly, the Data Link Layer makes sure that the data transfer to/from the physical media is done properly. Here again the communication done at the data link layer can be reliable or unreliable.
- This information travels on the physical media (like Ethernet) and reaches the target machine.

**Now at the target machine (web server) the same series of interactions happen, but in reverse order.**

- The packet is first received at the data link layer. At this layer, the information (which was stuffed by the data link layer of the host machine) is read and the rest of the data is passed to the upper layer.
- Similarly at the Network layer, the data sent by the Network layer of the host machine is read and the rest of the data is passed on the next upper layer. Same happens at the transport layer and finally the HTTP request sent by the host application (your browser) is received by the target application (web server).
- At the target machine, when data reaches this layer, the TCP makes note of the sequence number of the packet and sends an acknowledgement.
- If the host TCP does not receive the acknowledgement within some specified time, it re sends the same packet. So this way TCP makes sure that no packet gets lost. So we see that protocol at every layer reads the information sent by its counterparts to achieve the functionality of the layer it represents.

## 3.2.1 Protocols

**What are Peer-to-Peer Networks?**

In a nutshell, a peer-to-peer (P2P) network is created when 2 or more PCs are connected and share resources without going through a separate server computer. A P2P network can be an ad hoc connection or a P2P network can be a network on a much greater scale in which applications set up direct relationships among users over the internet. Most P2P programs are focused on media sharing and hence P2P is often associated with software privacy and copyright violation. A famous example of P2P application is BitTorrent and so we are going to discuss how BitTorrent actually carries out its job.

Over the past decade, the demand for movies and TV shows has greatly increased and the public wishes to attain this content for free. So when a video is uploaded to a website for the public to download, there is a high probability that you will experience slower than normal download speeds because many users are trying to download that video from the same source as you (i.e. too much website traffic).

Thankfully, we have options that make sharing big files over the internet fast and easy. Unlike typical downloads files downloaded using BitTorrent actually download **faster** when more people are involved.

The BitTorrent application is somewhat similar to a web browser, just as a web browser needs websites to be useful, BitTorrent needs special files called “torrents” to work. When you locate a specific torrent file and set it to download, it will immediately start downloading.

### 3.2.1 Protocols

Aside from BitTorrent when someone downloads a file, it comes to their computer in a stream from a single source. When multiple people want that file, that stream gets divided and the source can get overworked and even shut down.

BitTorrent solves this problem by making each downloader **a source!** This way they get pieces of the file, but also share pieces with each other. Together, the downloaders become a network of multiple sources all working to provide pieces to one another which makes downloads fast and reliable.

When a torrent file is downloaded, it contains all the information about the video file. Like what pieces are needed to complete it. So how do you connect with the other users which have the pieces you need to complete your video?

BitTorrent uses a computer called a **Tracker** that helps your computer find other computers called **Peers**. The Tracker keeps a track of the computers which are downloading or already have the whole file and introduces your computer to them. With the connections in place, he receives what he needs and also distributes pieces to other users who are also downloading.

The reason why BitTorrent is so popular is because it turns downloaders into sources and more downloaders mean more sources and faster downloads