TCP/IP Ports and Sockets Explained

steve

Updated:July 6, 2018 By

On a TCP/IP network every device must have an IP address.

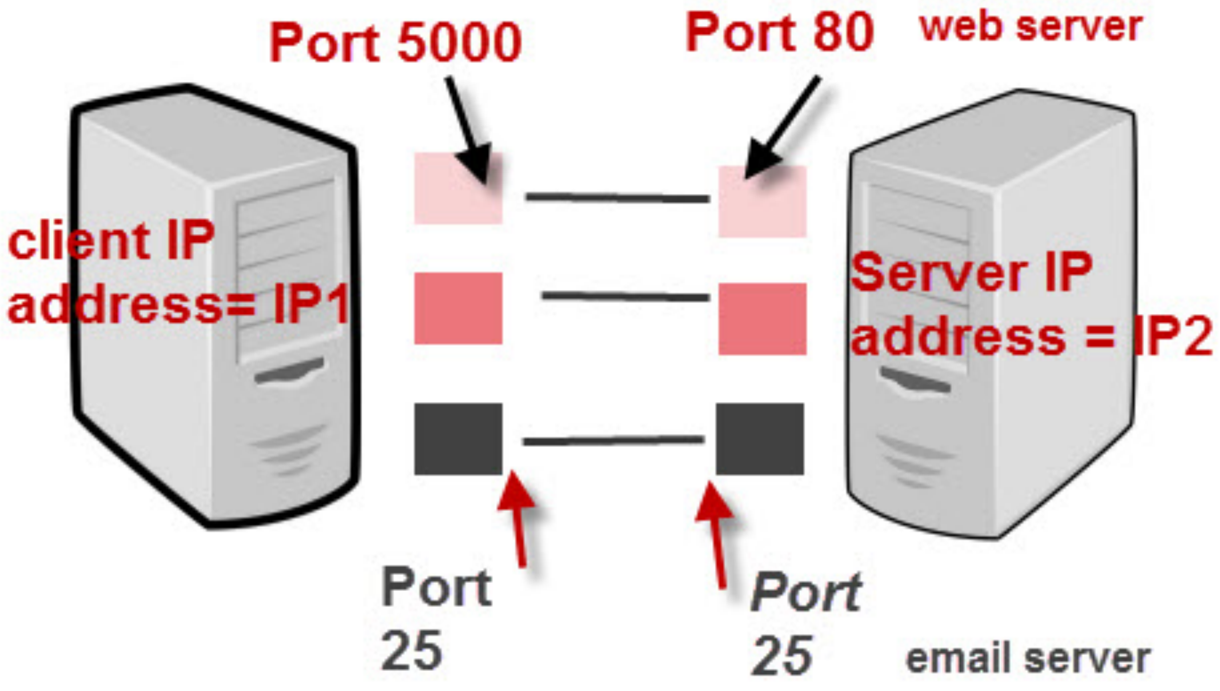
The
IP address identifies the device e.g. computer.

However an IP address alone is not sufficient for running network applications, as a computer can run **multiple applications** and/or **services**.

Just as the IP address identifies the computer, The network **port** identifies the **application or service** running on the computer.

> ***The use of ports allow computers/devices to run multiple services/applications***.

The diagram below shows a computer to computer connection and identifies the IP addresses and ports.



IP Address + Port number = Socket

TCP/IP Ports And Sockets

Analogy

If you use a house or apartment block analogy the IP address corresponds to the street address.

All of the apartments share the same street address.

However each apartment also has an apartment number which corresponds to the Port number.

Port Number Ranges and Well Known Ports

A port number uses 16 bits and so can therefore have a value from **0** to **65535** decimal

Port numbers are divided into ranges as follows:

Port numbers 0-1023 – Well known ports. These are allocated to **server services** by the **Internet Assigned Numbers Authority** (IANA). e.g Web servers normally use **port 80** and SMTP servers use **port 25** (see diagram above).

Ports 1024-49151- Registered Port -These can be registered for services with the **IANA** and should be treated as **semi-reserved.** User written programs should not use these ports.

Ports 49152-65535– These are used by **client programs** and you are free to use these in client programs. When a Web browser connects to a web server the browser will allocate itself a port in this range. Also known as **ephemeral ports**.

TCP Sockets

A connection between two computers uses a **socket.**

> *A socket is the combination of **IP address plus port***

{outline]Each end of the connection will have a socket.{/outline]

Imagine sitting on your PC at home, and you have two browser windows open.

One looking at the Google website, and the other at the Yahoo website.

The connection to Google would be:

Your PC – **IP1**+port 60200 ——— Google IP2 +port **80** (standard port)

The combination IP1+60200 = the socket on the client computer and **IP2 + port 80** = destination socket on the Google server.

The connection to Yahoo would be:

your PC – **IP1**+port 60401 ———Yahoo IP3 +port **80** (standard port)

The combination IP1+60401 = the socket on the client computer and **IP3 + port 80** = destination socket on the Yahoo server.

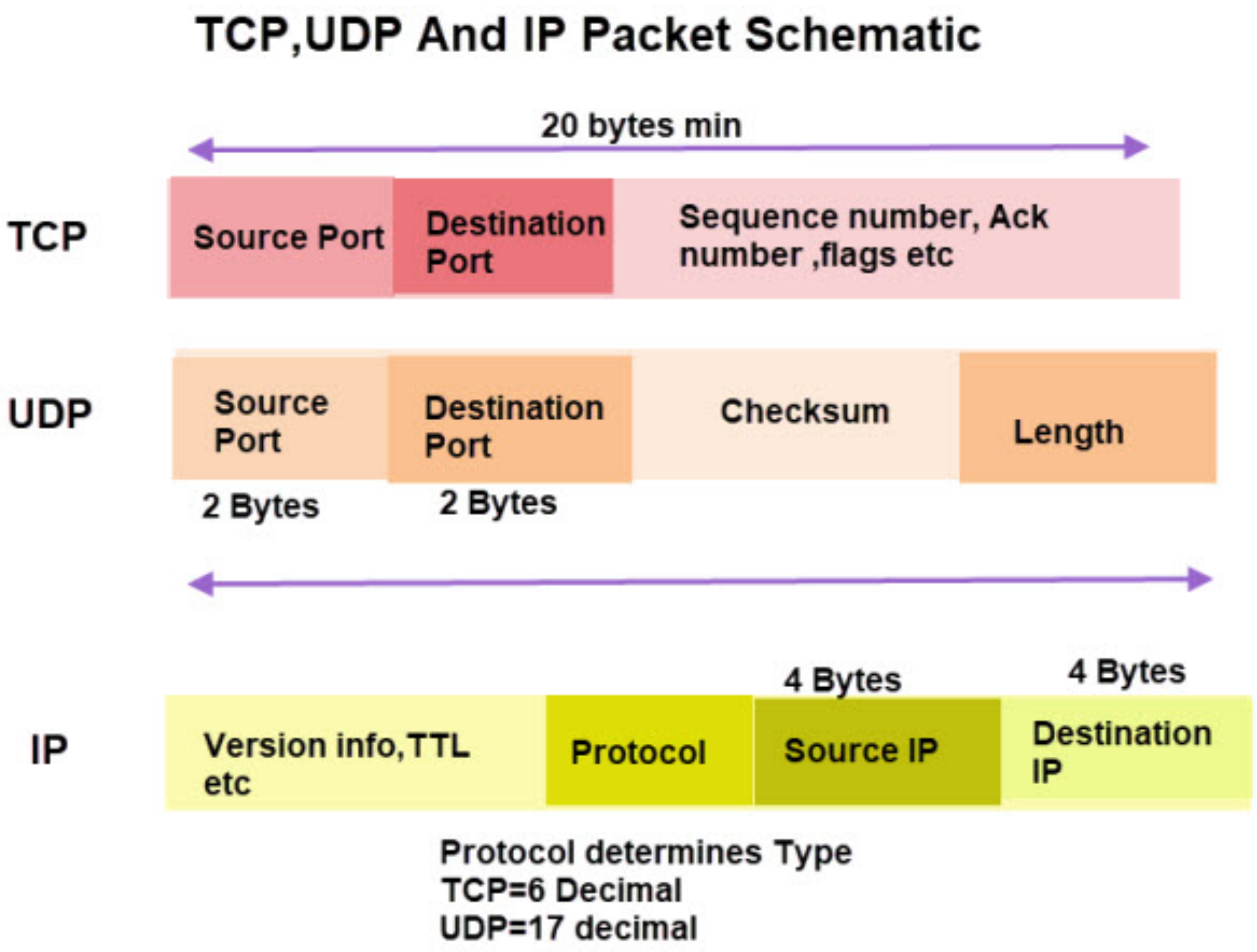
Notes: **IP1** is the IP address of your PC. Client port numbers are dynamically assigned, and can be reused once the session is closed.

TCP and UDP -The Transport Layer

Note: You may find reading the article on the TCP/IP protocol suite useful to understand the following

IP addresses are implemented at the networking layer which is the **IP layer.**

Ports are implemented at the transport layer as part of the **TCP or UDP header** as shown in the schematic below:



The TCP/IP protocol supports two types of port- **TCP Port** and **UDP Port**.

TCP – is for connection orientated applications. It has built in error checking and will re transmit missing packets.

UDP – is for connection less applications. It has no has built in error checking and **will not** re transmit missing packets.

Applications are designed to use either the UDP or TCP transport layer protocol depending on the type of connection they require.

For example a web server normally uses **TCP port 80**.

It can use any port, but the web server application is **designed to use a TCP connection. See** **TCP vs UDP**

Here is a very good video that explains ports and sockets really well

Checking For Open Ports

Windows and Linux systems have a utility called **netstat** which will give you a list of open ports on your computer.

These articles show you how to use **netstat** on windows and on linux.

You can check the **port status** of remote machines using a port scanner line nmap.

You can install NMAP on windows,Linux and Apple. It can be used with a graphical user interface of as a command line tool.

Here is a useful article on using NMAP from the command line.

Here is a good video on using **Nmap** and also covers TCP/IP connection procedures which is useful for understanding ports.

References and resources:

TCP and UDP basics -Connecting to a website- This is for programmers but there is no coding just an explanation of ports and sockets.

Connection states– if you are wondering what established and listening and the other state descriptions mean. here is a good state diagram that it refers to.

Online port tester Collection of tools for port scanning and web server testing.

Related Articles:

- TCP/IP protocol suite explained
- Port Forwarding Explained
- Basic networking course for Beginners
- IPv4 Basics