Chapter 38
Introduction to Sockets and Web Services

38.1 Introduction

In the following two chapters we will explore socket based and web service approaches to inter process communications. These processes may be running on the same computer or different computers on the same local area network or may be geographically far apart. In all cases information is sent by one program running in one process to another program running in a separate process via internet sockets. This chapter introduces the core concepts involved in network programming.

38.2 Sockets

Sockets, or rather Internet Protocol (IP) sockets provide a programming interface to the network protocol stack that is managed by the underlying operating system. Using such an API means that the programmer is abstracted away from the low level details of how data is exchanged between process on (potentially) different computers and can instead focus on the higher level aspects of their solution.

There are a number of different types of IP socket available, however the focus in this book is on *Stream Sockets*. A stream socket uses the Transmission Control Protocol (TCP) to send messages. Such a socket is often referred to as a TCP/IP socket.

TCP provides for ordered and reliable transmission of data across the connection between two devices (or hosts). This can be important as TCP guarantees that for every message sent; that every message will not only arrive at the receiving host but that the messages will arrive in the correct order.

A common alternative to the TCP is the User Datagram Protocol (or UDP). UDP does not provide any delivery guarantees (that is messages can be lost or may arrive out of order). However, UDP is a simpler protocol and can be particularly useful for

J. Hunt, *Advanced Guide to Python 3 Programming*,
Undergraduate Topics in Computer Science,
https://doi.org/10.1007/978-3-030-25943-3_38

broadcast systems, where multiple clients may need to receive the data published by a server host (particularly if data loss is not an issue).

38.3 Web Services

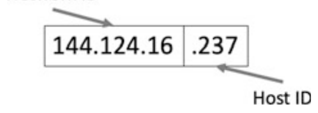
A Web Service is a service offered by a host computer that can be invoked by a remote client using the Hypertext Transfer Protocol (HTTP). HTTP can be run over any reliable stream transport protocol, although it is typically used over TCP/IP. It was originally designed to allow data to be transferred between a HTTP server and a web browser so that the data could be presented in a human readable form to a user. However, when used with a web service it is used to support program to program communication between a client and a server using machine-readable data formats. Currently this format is most typically JSON (Java Script Object Notation) although in the past XML (eXtensible Markup Language) was often used.

38.4 Addressing Services

Every device (host) connected to the internet has a unique identity (we are ignoring private networks here). This unique identity is represented as an IP address. Using an IP address we can connect a socket to a specific host anywhere on the internet. It is therefore possible to connect to a whole range of device types in this way from printers to cash tills to fridges as well as servers, mainframes and PCs etc.

IP addresses have a common format such as `144.124.16.237`. An IP version 4 address is always a set of four numbers separated by full stops. Each number can be in the range 0–255, so the full range of IP addresses is from 0.0.0.0 to 255.255.255.255.

An IP address can be divided up into two parts; the part indicating the network on which the host is connected and the host's ID, for example:



Thus:

- The Network ID elements of the IP address identifies the specific network on which the host is currently located.
- The Host ID is the part of the IP address that specifies a specificities device on the network (such as your computer).

On any given network there may be multiple hosts, each with their own host ID but with a shared network ID. For example, on a private home network there may be:

- `192.168.1.1` Jasmine's laptop.
- `192.168.1.2` Adam's PC
- `192.168.1.3` Home Printer
- `192.168.1.4` Smart TV

In many ways the network id and host id elements of an IP address are like the postal address for a house on a street. The street may have a name, for example Coleridge Avenue and there may be multiple houses on the street. Each house has a unique number; thus 10 Coleridge Avenue is uniquely differentiated from 20 Coleridge Avenue by the house number.

At this point you may be wondering where the URLs you see in your web browser come into play (such as www.bbc.co.uk). These are textual names that actually map to an IP address. The mapping is performed by something called a Domain Name System (or DNS) server. A DNS server acts as a lookup service to provide the actual IP address for a particular textual URL name. The presence of an english textual version of a host address is because humans are better at remembering (a hopefully) meaningful name rather than what might appear to be a random sequence of numbers.

There are several web sites that can be used to see these mappings (and one is given at the end of this chapter). Some examples of how the english textual name maps to an IP address are given below:

- www.aber.ac.uk maps to 144.124.16.237
- www.uwe.ac.uk maps to 164.11.132.96
- www.bbc.net.uk maps to 212.58.249.213
- www.gov.uk maps to 151.101.188.144

Note that these mappings were correct at the time of writing; they can change as new entries can be provided to the DNS servers causing a particular textual name to map to a different physical host.

38.5 Localhost

There is a special IP address which is usually available on a host computer and is very useful for developers and testers. This is the IP address:

```
127.0.0.1
```

It is also known as *localhost* which is often easier to remember.

Localhost (and `127.0.0.1`) is used to refer to the computer you are currently on when a program is run; that is it is your local host computer (hence the name localhost).

For example, if you start up a socket server on your local computer and want a client socket program, running on the same computer, to connect to the server program; you can tell it to do so by getting it to connect to localhost.

This is particularly useful when either you don't know the IP address of your local computer or because the code may be run on multiple different computers each of which will have their own IP address. This is particularly common if you are writing test code that will be used by developers when running their own tests on different developer (host) machines.

We will be using localhost in the next two chapters as a way of specifying where to look for a server program.

38.6 Port Numbers

Each internet device/host can typically support multiple processes. It is therefore necessary to ensure that each process has its own channel of communications. To do this each host has available to it multiple ports that a program can connect too. For example port 80 is often reserved for HTTP web servers, while port 25 is reserved for SMTP servers. This means that if a client wants to connect to a HTTP server on a particular computer then it must specify port 80 not port 25 on that host.

A port number is written after the IP address of the host and separated from the address by a colon, for example:

- www.aber.ac.uk:80 indicates port 80 on the host machine which will typically be running a HTTP server, in this case for Aberystwyth University.
- localhost:143 this indicates that you wish to connect to port 143 which is typically reserved for an IMAP (Internet Message Access Protocol) server on your local machine.
- www.uwe.ac.uk:25 this indicates port 25 on a host running at the University of the West of England, Bristol. Port 25 is usually reserved for SMTP (Simple Mail Transfer Protocol) servers.

Port numbers in the IP system are 16 bit numbers in the range 0–65 536. Generally, port numbers below 1024 are reserved for pre-defined services (which means that you should avoid using them unless you wish to communicate with one of those services such as telnet, SMTP mail, ftp etc.). Therefore it is typically to choose a port number above 1024 when setting up your won services.

38.7 IPv4 Versus IPv6

What we have described in this chapter in terms of IP addresses is in fact based on the Internet Protocol version 4 (aka IPv4). This version of the Internet Protocol was developed during the 1970s and published by the IETF (Internet Engineering Task Force) in September 1981 (replacing an earlier definition published in January 1980). This version of the standard uses 32 binary bits for each element of the host address (hence the range of 0 to 255 for each of there parts of the address). This provides a total of 4.29 billion possible unique addresses. This seemed a huge amount in 1981 and certainly enough for what was imagined at the time for the internet.

Since 1981 the internet has become the backbone to not only the World Wide Web itself, but also to the concept of the Internet of Things (in which every possible device might be connected to the internet from your fridge, to your central heating system to your toaster). This potential explosion in internet addressable devices/hosts lead in the mid 1990as to concerns about the potential lack of internet addresses using IPv4. The IETF therefore designed a new version of the Internet Protocol; Internet Protocol version 6 (or IPv6). This was ratified as an Internet Standard in July 2017.

IPv6 uses a 128 bit address for each element in a hosts address. It also uses eight number groups (rather than 4) which are separated by a colon. Each number group has four hexadecimal digits.

The following illustrates what an IPv6 address looks like:

```
2001:0DB8:AC10:FE01:EF69:B5ED:DD57:2CLE
```

Uptake of the IPv6 protocol has been slower than was originally expected, this is in part because the IPv4 and IPv6 have not been designed to be interoperable but also because the utilisation of the IPv4 addresses has not been as fast as many originally feared (partly due to the use of private networks). However, over time this is likely to change as more organisations move over to using the IPv6.

38.8 Sockets and Web Services in Python

The next two chapters discuss how sockets and web services can be implemented in Python. The first chapter discusses both general sockets and HTTP server sockets. The second chapter looks at how the Flask library can be used to create web services that run over HTTP using TCP/IP sockets.

38.9 Online Resources

See the following online resources for information

- https://en.wikipedia.org/wiki/Network_socket Wikipedia page on Sockets.
- https://en.wikipedia.org/wiki/Web_service Wikipedia page on Web Services.
- https://codebeautify.org/website-to-ip-address Provides mappings from URLs to IP addresses.
- https://en.wikipedia.org/wiki/IPv4 Wikipedia page on IPv4.
- https://en.wikipedia.org/wiki/IPv6 Wikipedia page on IPv6.
- https://www.techopedia.com/definition/28503/dns-server For an introduction to DNS.