Exercises: Networking

Part A

Write Python programs – one using socket, one using socket server, one using Twisted, and one using Asyncio, that act as network servers of time information. A TCP client should, on connecting to a server, receive the time in ISO 8601 format.

Part B

Multicast is a special form of UDP where a datagram is sent to all nodes, not just one. Each node can choose to listen (or not) for multicast datagrams sent to a given group. Multicast addresses are 224.0.0.0 and upwards, however ISPs and firewalls often block addresses below 239.0.0.0 to avoid multicast packets appearing on the Internet. Addresses 239.0.0.0 and above are specified to be local and so are not allowed to escape from the local network. Such packets are thus not blocked by firewalls.

Write Python programs to listen for and send multicast packets on group 239.3.2.1:58000. You will almost certainly want to do this with Asyncio or Twisted first, doing it directly with sockets gets a bit messy (especially compared to doing it with Asyncio or Twisted).

NB Ensure the time-to-live of the datagram is 1, we do not want to allow any possibility of the datagrams escaping out into the wild!

Part C

Write a Python program that is an HTTP server listening on port 8001 that returns as a top-level heading 'Hello World.'

Part D

Write a Python program that is a 'packet sniffer' that displays packet information using a graphical display.

NB PyQt, PySide, PyGObject, PyGTK and wxPython, as well as Twisted Internet use event-driven approaches and provide their own event loops. You will need to research which event loop to use for this application and how to integrate GUI and networking.