

Introduction

The User Datagram Protocol (UDP) is one of the core members of the Internet Protocol Suite, the set of network protocols used for the Internet. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without requiring prior communications to set up special transmission channels or data paths. UDP is sometimes called the Universal Datagram Protocol.

How to send UDP broadcast

It's possible to use UDP broadcast with PyS60 (e.g. with ad hoc Wi-Fi networks on the N95). However, the operation is slightly different from that of the standard Python libraries. There are two primary issues:

1) You do not need to (and indeed cannot) call `socket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)` to enable broadcast transmit. Also, the address "<broadcast>" will not be recognized. Use an explicit `255.255.255.255` instead.

Example:

```
#broadcasts "hello" to all machines on the local subnet, on port 8100

outsock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
msg = "hello"
port = 8100
outsock.sendto(msg, ("255.255.255.255", 8100))
```

2) You must bind to the address `0.0.0.0` to receive UDP broadcast packets. This will bind to all available interfaces. This corresponds to calling bind with an empty string in the standard Python socket implementation. Do not bind to the external IP address of the machine!

This example reads a packet from a UDP connection, and will respond to broadcast packets:

```
# receives a single packet, including broadcast packets

self.insock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
insock.bind(("", 8100))
(msg, addr) = recvfrom(65536)
```

As a further quick note, it seems that in the pys60 socket implementation, you can use nonblocking sockets by calling `socket.setblocking(False)`. However, this only works for `recv()` and not for `recvfrom()`.

If you need nonblocking behaviour with `recvfrom()`, use `socket._recv_will_return_data_immediately()` (which returns True if packets waiting) to test if there is a packet waiting, then call `recvfrom()` as required.

Reference link

- [User Datagram Protocol](#)