Socket Timeouts in Python

Socket Timeouts

- It is often useful to initiate socket activity subject to a timeout.
- If no activity occurs within the timeout period, the Python script regains execution and continues on with other activities.
- An example of this is in Lab 3 where a client host does a service discovery scan to find what file sharing services are available on a particular UDP port.
  - The client sends IP broadcast packets and does a socket recv, waiting for responses for a set timeout interval. If no servers are available, the socket recv times-out and the client can regain control and print out a "No servers found." message.
  - Without socket timeouts, the recv would block indefinitely.

Socket Timeouts

- Socket timeouts are set by using

  ```
  my_socket.settimeout(5)
  ```

  which sets the timeout value to 5 seconds. Giving an argument of **None** disables timeouts.

- The timeout exception is then captured by a try/except block:

```
try:
    data = my_socket.recv(1024)
    (recv returned with data)
    ...
except socket.timeout:
    print("Socket recv timeout ...")
    ...
```

echo_server_timeout.py
echo_server_timeout_multiclient.py

service_announcement_discovery.py
file_download_protocol.py