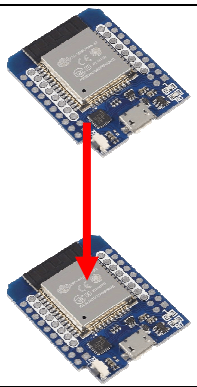
The Firewall and the ESP32 for IOT

Programming challenge 2.

Send UDP messages between a computer running Windows, macOS or Linux and an ESP32 IOT chip under different scenarios and combinations. For all cases, the sender/client sends a message to the receiver/server and the receiver replies with an acknowledgement message. The sender can be on a laptop or on the ESP32 chip. The receiver can also be on a laptop or on the ESP32 chip. Here are several different scenarios (ordered from easy to hard):

1. The sender and the receiver are both on the same network.

   One ESP32 is the sender and a second ESP32 is the receiver. Both chips are connected to the same WiFi network. One ESP32 sends a message to another ESP32, and the second ESP replies with an acknowledgement. (This is just testing out my sample ESP32 UDP sender and receiver programs.)

Outside CS Firewall	CS WiFi Firewall Router access point	Sender and receiver Inside CS Firewall
		
		Both connected to CS WiFi
	Outside IP 10.15.15.2 Inside IP 192.168.1.1	Sender IP 192.168.x.x Receiver IP 192.168.x.x
		Sender sends to 192.168.x.x

2. The sender is behind a firewall and the receiver is outside the firewall.

   The CS WiFi router is the firewall. The ESP32 sender is connected to CS WiFi so it is inside the firewall, and the ESP32 receiver is connected to Guest WiFi so it is outside the firewall. The sender inside the firewall will have a 192.168.x.x IP address, whereas the receiver outside the firewall will have a 10.15.x.x IP address. The firewall only blocks initial incoming traffic; it doesn't block outgoing traffic or incoming replies. So this scenario works without any changes to the firewall.

Receiver/server Outside CS Firewall	CS WiFi Firewall Router access point	Sender/client Inside CS Firewall
		
Connected to Guest WiFi		Connected to CS WiFi
Receiver IP 10.15.x.x	Outside IP 10.15.15.2 Inside IP 192.168.1.1	Sender IP 192.168.x.x
Receiver listening on port x		Sender sends to 10.15.x.x port x

3. The receiver is behind a firewall and the sender is outside the firewall.

The ESP32 receiver is connected to CS WiFi so it is inside the firewall, and the ESP32 sender is connected to Guest WiFi so it is outside the firewall. The receiver inside the firewall will have a 192.168.x.x IP address, whereas the sender outside the firewall will have a 10.15.x.x IP address. Since the firewall blocks initial incoming traffic, therefore, the sender outside will not be able to send messages to the receiver inside. This scenario will work only if you punch a hole through the firewall, i.e., configure the router to do port forwarding for whatever port you want to use.

The CS WiFi router has a fixed static IP address of **10.15.15.2**. The sender will send to this CS WiFi router IP address using a unique port number, say, 12345. In the router's setup page, you need to port forward all messages destined for port 12345 to the IP address of your receiver which has a 192.168.x.x value. You also specify the port number that the receiver is listening on, which can be different from 12345.

Sender/client Outside CS Firewall	CS WiFi Firewall Router access point	Receiver/server Inside CS Firewall
		
Connected to Guest WiFi		Connected to CS WiFi
Sender IP 10.15.x.x	Outside IP 10.15.15.2 Inside IP 192.168.1.1	Receiver IP 192.168.x.x
Sender cannot send to 192.168.x.x		Receiver listening on port x
Sender sends to 10.15.15.2	Port forward all messages destined for port 12345 to 192.168.x.x port x	

And here is the programming challenge question. How does someone using a web browser or an app running on a mobile phone control a WiFi smart light bulb inside the home? The smart light bulb inside the home is behind a firewall and the user's mobile phone is outside the firewall. This is like scenario #3 above, but you do not want a user to have to punch a hole through their firewall.

So the only solution is to use scenario #2 above. Remember that the firewall only blocks initial incoming traffic; it **doesn't**

block outgoing traffic or incoming replies.

Programming challenge

Follow scenario #2 to create a web server and a LED light client for turning on and off a LED light. The web server is outside the firewall and can be implemented either on a computer/laptop or on the ESP32 chip. The LED light client is inside the firewall and is implemented on the ESP32 chip.

Step 1 - The client initiates a connection with the server, so that the server will know how to reply to the client.

Step 2 – The mobile phone user will browse to a webpage on the web server. The webpage has buttons to turn on/off a light.

Step 3 - The web server will reply to the client with the on/off command. The light attached to the client will turn on/off accordingly.