Boston University
Electrical & Computer Engineering

EC464 Capstone Senior Design Project

Second Prototype Testing Plan

CyberTap

by

Team 2
CyberTap

Team Members:
Felipe Dale Figeman fdale@bu.edu
Alex Fatyga afatyga@bu.edu
Evan Lang evanlang@bu.edu
Noah Malhi malhin@bu.edu
Justin Morgan justinfm@bu.edu

Required Materials:

Hardware:

- 1 Ethernet capable device
- TP-LINK TL-SG105E 5-Port Gigabit Easy Smart Network Switch
- Desktop PC
- PYNQ-Z1 Board
- 3 Ethernet Cables

Software:

- Xilinx C/C++ SDK 2019.1
- Jupyter Notebook
- Node.js Web Client
    - main.html
        - Front end that receives data from server.js and puts it into the table
    - server.js
        - Back end that reads the csv file, sends it to front end
- Packets.csv - as packets are parsed, they are written into this csv file which is read in server.js
- Python
    - sniffEx.py - sniffs network activity, parses it and writes it to a file
    - Main.py - runs all python files
    - Scapy terminal - used to create and send packets of various protocol types
- Bash script main.sh - runs the web client in the background and then runs python sniffing and parsing

Setup:

The test focuses on our packet sniffing, software parsing, modbus packet creation, and web application. After setting up the FPGA and running the bash script, the user will go to 192.168.137.99:8080/main.html and can view the parsed information of any incoming or sniffed packets. The user can refresh and see intercepted packets and their related information. Two

computers will be connected to the same switch as the FPGA and given static IP's 192.168.137.XXX/24. One computer will be used to ping the other using the Linux ping command. The ICMP packets from the first computer that sends to the second will be sniffed by the FPGA, parsed, and displayed on the web app. This will be done 3 times with 3 different ping intervals (.1s, .01s, and .001s) with 1000 pings being sent per run. When a ping run finishes, the packet sniffer will be immediately turned off to prevent unrelated packets from being included in the test. There are 3 metrics that need to be met in order for the test to be considered successful. First, the packet loss on the .1s interval ping test must be less than 1%, the packet loss on the .01s interval test must be less than 10%, and the packet loss on the .001s interval test must be less than 20%. Second, the web app must properly display the list of sniffed packets. Third, the information (source, destination, time sent, etc.) included in each packet sniffed must be accurate. After this test, a second test will be performed. The second test will involve the creation of packets of the following protocol variations (IP, TCP, UDP, Modbus/TCP) using the Scapy console on a sender computer. Through the console, these packets will then be sent to the receiver computer within the IP range described above, and the web app will be used to confirm that the newly created and sent packets are being picked up by the PYNQ and sent to the web app. In order to ensure that the packets picked up by the sniffer match those sent out via the Scapy console, the fields for sender IP, destination IP, and protocol type will be matched against the sender's configurations used to create and send said packets.

Additional note: It is very likely the web app will display more packets than were sent on the .1s ping test. That is because the test will run for significantly longer than the other 2, allowing packets unrelated to the test to also pass through the switch will the board is sniffing. If the number sniffed by the packet sniffer is greater than 1000 during this test, that will also be considered a success.

Pre-Testing Setup Procedure:

FPGA Side:

1. Connect the FPGA to a computer through microUSB

2. Turn on and boot the FPGA
3. Connect to the FPGA on the computer using a serial connection with a baud rate of 115200
4. Connect the FPGA to the switch on port 4
5. Open Jupyter Notebook on the desktop computer

Network Device Side:

1. Configure the network switch to mirror ingress packets from ports 1 and 2 to port 4.
2. On two devices, configure the ethernet adapter to use static IPV4 addresses on the same range as the FPGA
3. Connect the devices to the network switch on ports 1 and 2
4. Make sure the FPGA setup is complete

Modbus Packet Test Side:

1. Open the scapy console on the sender computer's terminal via the command 'sudo scapy'
2. Create packets of the following types: IP() (Raw IP packets), IP()/TCP(), IP()/UDP(), IP()/TCP()/Modbus()
3. Set the destination attributes of the IP packet configurations to the receiver computer's IP address
4. Confirm that the receiver computer is ready to receive said packets

Results Viewing Device Side:

1. Configure a device to have a static IPV4 on the same range as the FPGA

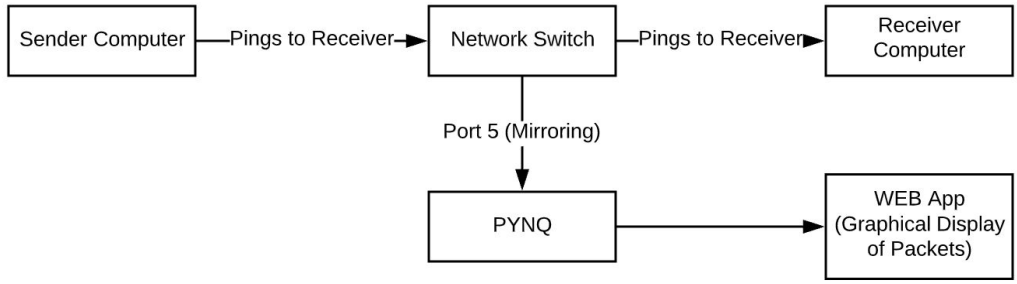
Testing Procedure:

1. Run ./main.sh in terminal
2. On a browser open <fpga_IP>:8080/main.html
3. Send pings from sender computer to receiver computer using command ping -i [Interval time in seconds] <fpga_ip_range>.XX/24
4. Once the sequence of test packets has been sent, stop packet sniffing with CTRL-Z
5. Refresh page on the viewing device
6. Compare number of packets sent vs. received and ensure IP addresses are correct

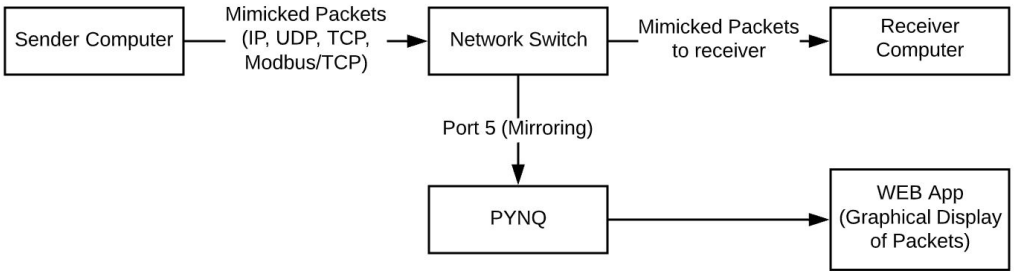
7. Run steps 1 through 5 three times, first with .1s interval, second with .01s interval, third with .001s interval
8. (Modbus Test) In Scapy console via a terminal window, perform the following steps after pre-testing setup procedure is complete
9. Create a while loop in the scapy console to send these packets every 100ms
10. Observe the sniffed packets and confirm that the sniffer is receiving said packets and displaying the correct protocols for each via the web app

Testing Diagrams:

Stress Testing via Pinging:



Modbus (and Other Protocols) Packet Testing:



Measurable Criteria:

The criteria for successful running and output is as follows:

1. The laptop sends the packet, as show in its terminal
2. The FPGA receives the packet, as shown in its terminal
3. The web application displays the received packet with its correct information, as shown in the web application's table

4. For the .1s interval test there is at most 1% packet loss
5. For the .01s interval test there is less than 10% packet loss
6. For the .001s interval test there is less than 20% packet loss

Score Sheet:

Test 1: 0.1s Interval Ping

Number of Packets Sent	Number of Packets Received
Packet Loss: _____%	

Test 2: 0.01s Interval Ping

Number of Packets Sent	Number of Packets Received
Packet Loss: _____%	

Test 3: 0.001s Interval Ping

Number of Packets Sent	Number of Packets Received
Packet Loss: _____%	

Web App Test:
Are the packets and there information correctly displayed? _____

Modbus Test:
Does the Web app correctly label the packets as TCP packets? _____