Ostinato

Network Traffic Generator and Analyzer

I. Introduction

Ostinato is an open-source, cross-platform packet/traffic generator and analyzer written in QT. Ostinato using agent-controller architecture: ostinato is a GUI agent where we can do the configuration and controls, drone is the controller to realize the packets generation and capture, so we can use via friendly GUI or python API. It can send packets of several streams with different protocols at different rates. To deal with the combination of packages for various protocols, Ostinato has very flexible ways, also it can generate many non-standard packages, such as IP over IP, ARP over TCP, etc. It aims to be "Wireshark in Reverse" and thus become complementary to Wireshark. In our project, we can apply it to become our performance testing tool.

II. Installation

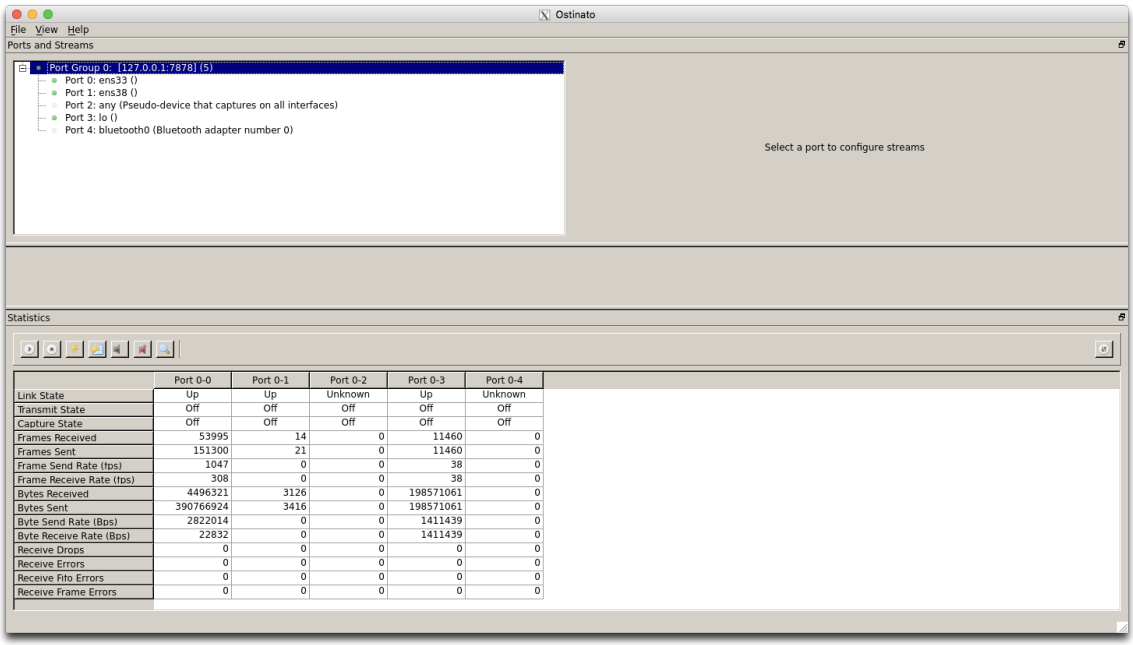
To compile and build the Ostinato as the generator for Linux system is quite easy, either we use the command line: `sudo apt-get install ostinato`, it will automatically build and form a new folder named ostinato in the current location, or we can build from the source code[1].
In my left VM I use the simple way, however when I tried to build from the source code, not only it requires multiple new compilers such as `protoc`, `qmake`, `qmake-qt4`, it still has problems when I tired `make` step. As a result, I install them both in a simple way.
As it introduced in the official document, this tool support the DPDK as an acceleration of Ostinato, but when I tried to compile this plugin it shows that the source is no longer available. I've conducted the developer through email for this issue.

To start this on my Ubuntu server, I need to enable the X11 forwarding from my device using command `ssh -Y username@my_vm_ip_address`.

[1] Build Instructions- Ostinato, https://devguide.ostinato.org/BuildingFromSource.html

III. Feature Layout & How To

Figure 1 shows the workspace after the startup, it is divided into three main sections: port list, data flow list, and statistics flow window.



	Port 0-0	Port 0-1	Port 0-2	Port 0-3	Port 0-4
Link State	Up	Up	Unknown	Up	Unknown
Transmit State	Off	Off	Off	Off	Off
Capture State	Off	Off	Off	Off	Off
Frames Received	53995	14	0	11460	0
Frames Sent	151300	21	0	11460	0
Frame Send Rate (fps)	1047	0	0	38	0
Frame Receive Rate (fps)	308	0	0	38	0
Bytes Received	4496321	3126	0	198571061	0
Bytes Sent	390766924	3416	0	198571061	0
Byte Send Rate (Bps)	2822014	0	0	1411439	0
Byte Receive Rate (Bps)	22832	0	0	1411439	0
Receive Drops	0	0	0	0	0
Receive Errors	0	0	0	0	0
Receive Fifo Errors	0	0	0	0	0
Receive Frame Errors	0	0	0	0	0

Figure 1: The Main Workspace interface

There is a port group entry for "127.0.0.1:7878" in the port list with green(connect) status, and under the group we can see all the ports on my local system. To create and send the packets, I selected the port(ens38) and create a new stream. The creation steps I followed the user guide from [2] They are:

- First I create a new stream and do the editing,
- On the "Protocol Selection" tab, select the protocols Mac - Ethernet II - IPv4 - UDP
- On the "Protocol Data" tab - Go to the Internet Protocol ver 4 (IPv4) section and enter source IP (self IP, which is 192.168.104.129 in my case) and destination IP (the sink virtual machine ip: 172.16.216.130)
- Go to the Media Access Control (Mac) section and set destination address to the mac address of the destination (according to the instruction, since these 2 devices are not under the same subnets, I need to clearify it by using the

[2]Quickstart Tutorial, https://userguide.ostinato.org/Quickstart.html

mac address of the default gateway mac address, in that case I use the mac address of port ens33) and source address to self mac(shown in the Figure 4

- On the "Stream Control" tab, configure number of packets as 100
- On the "Packet View" tab, you can preview how your packets will look(see Figure 6)

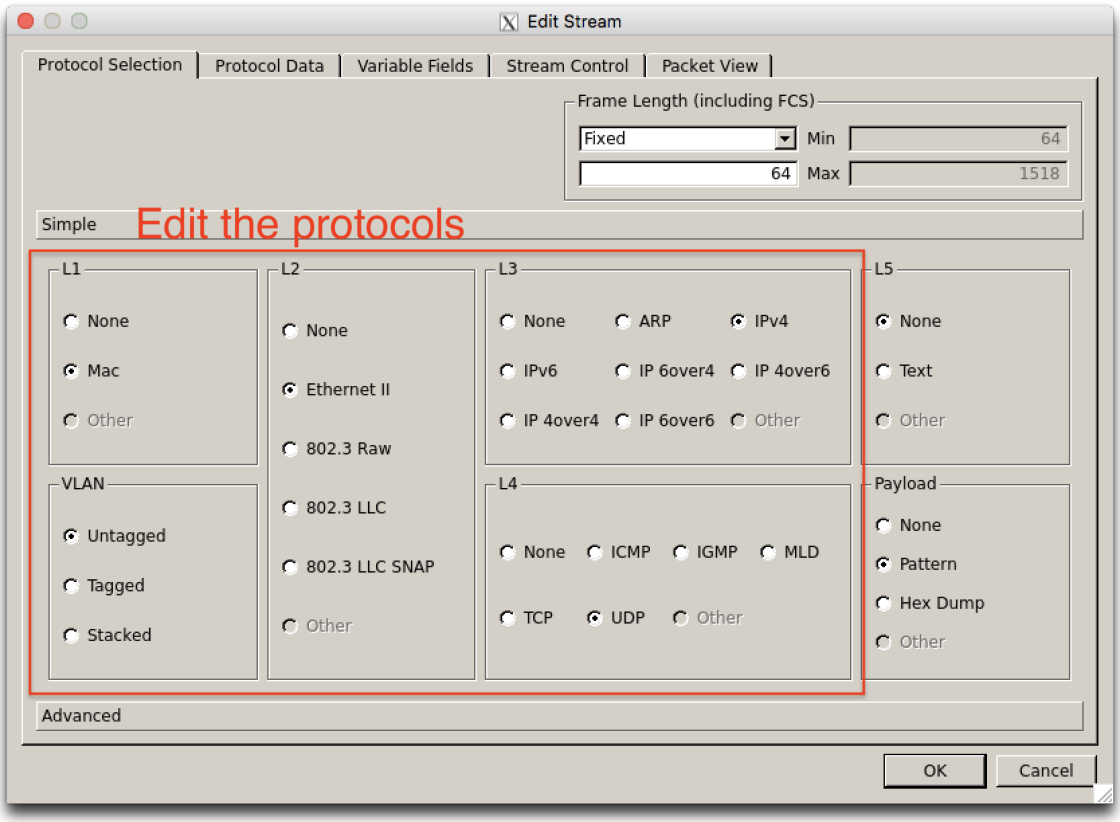


Figure 2: Protocol selection

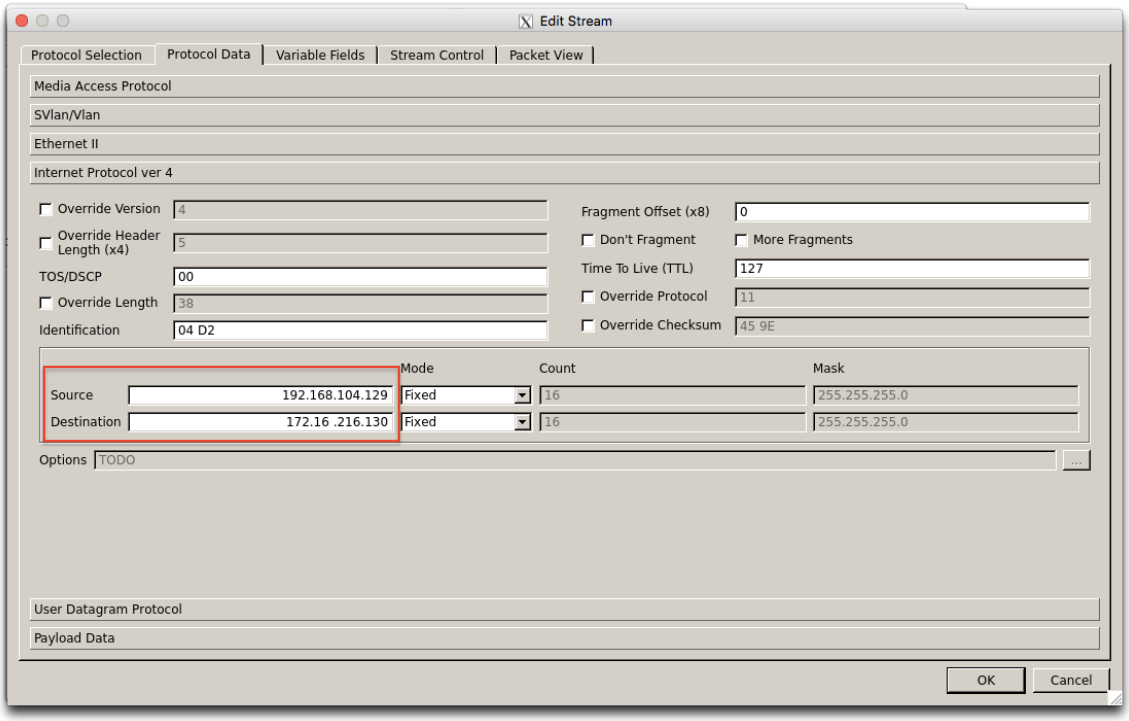


Figure 3: IPv4 setup

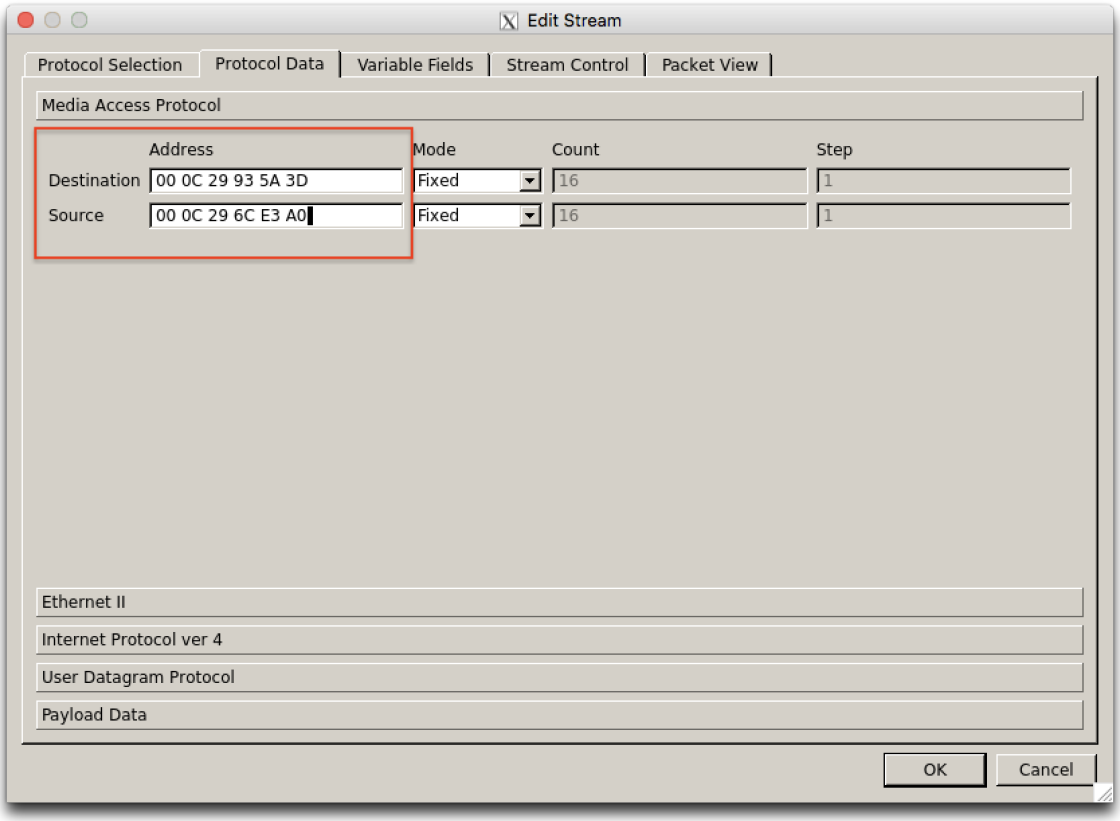


Figure 4: Mac address setup

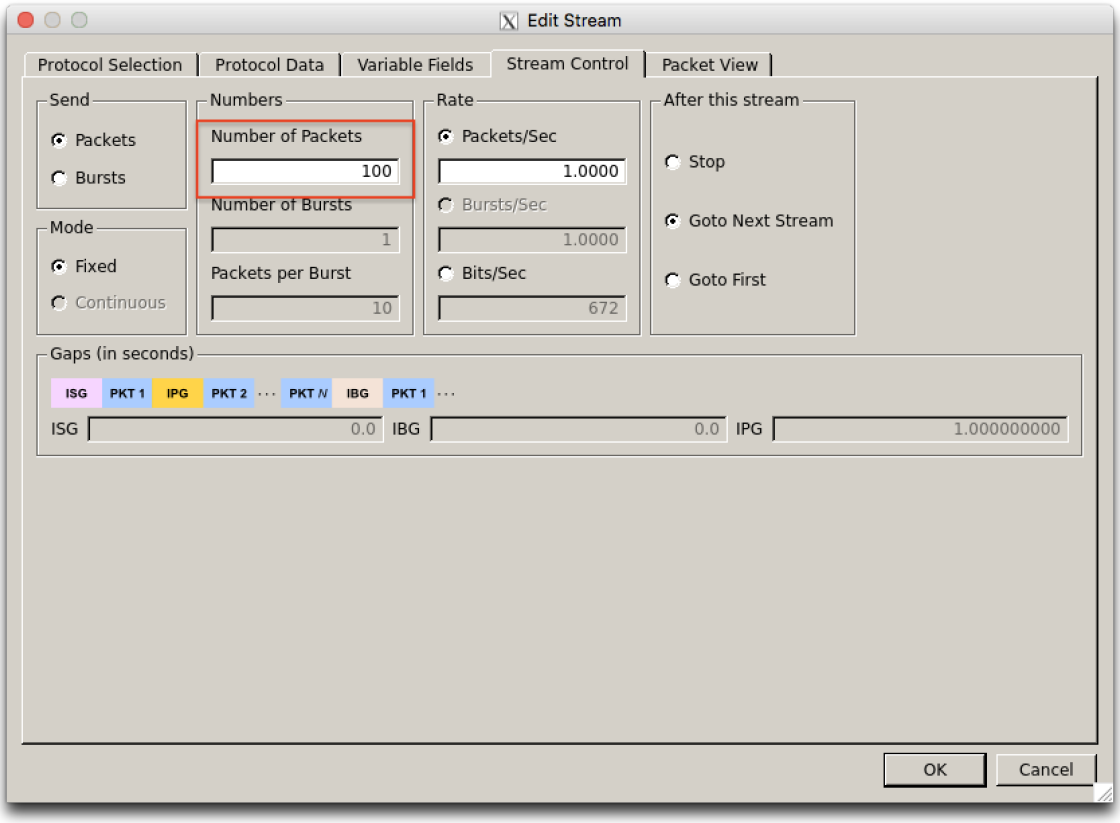


Figure 5: Packets number

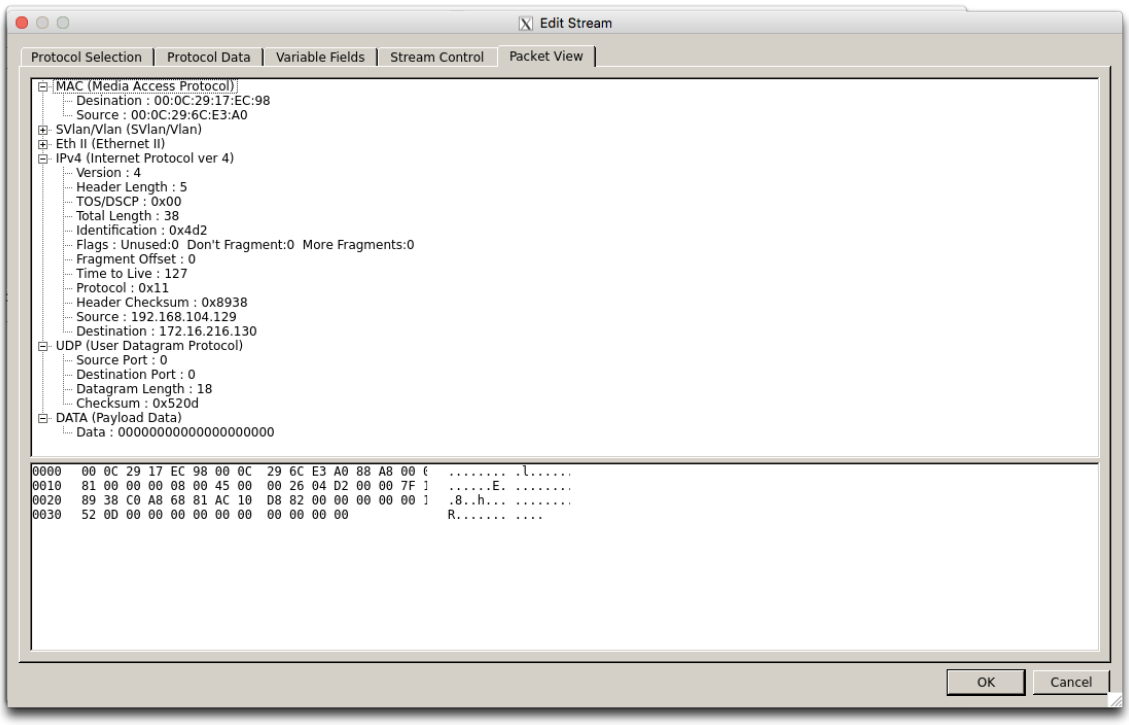


Figure 6: Packets information

After I click on "OK" and "Apply" button, the configuration with packets is done. Then In the statistics window, I could choose the port I want to send with and click on "Start Transmit", the packets will be sent as the *Frames Sent* row starts to increase. And all the information of this packets forwarding can be seen under the port column, see in Figure 7.

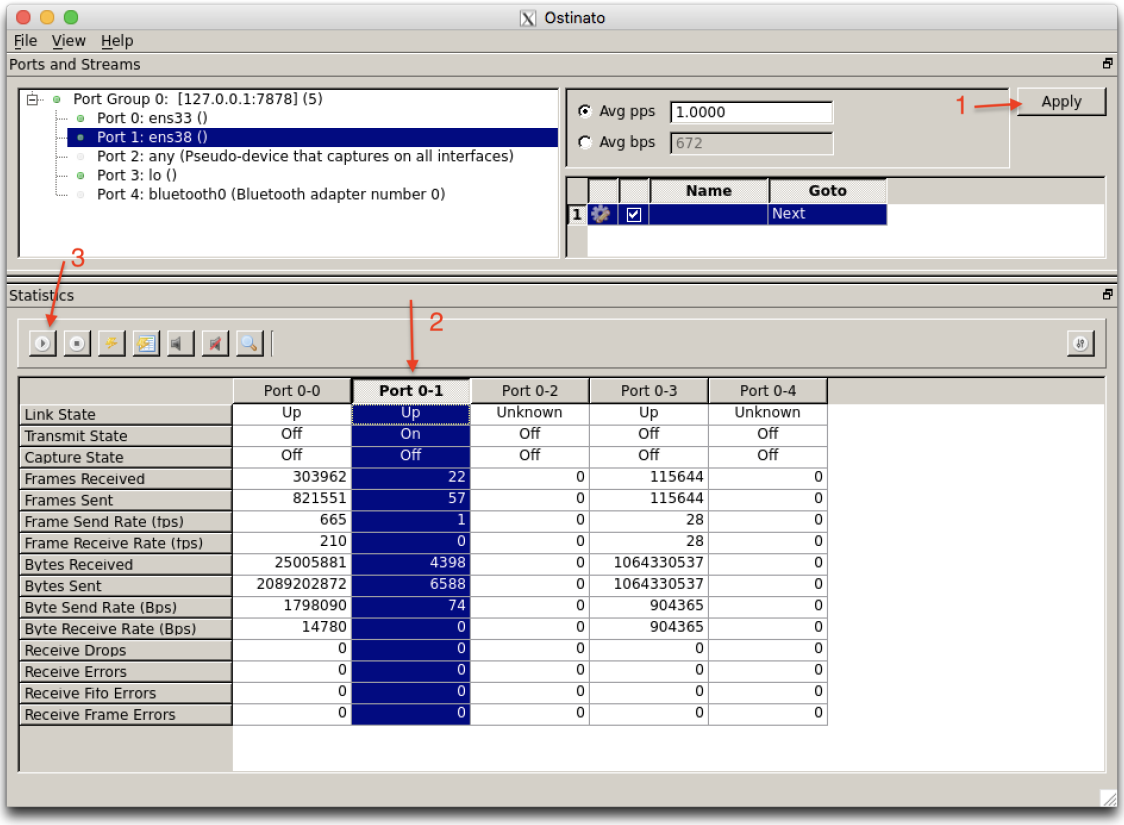


Figure 7: Send Packets

IV. Advanced

As I read the documents I found out that the stream editing can be various, there's more protocols in the "Advanced Mode" under "Protocol Selection" part, and to create multiple stream is as simple as to create the first one, we can change the protocols for each stream and other editing or simple right click on existing stream to duplicate a certain number of it. To test that, we need to click on "Clear Stats" to Clears statistics on selected port, then restart the transmit. The result Figure shown in **??**

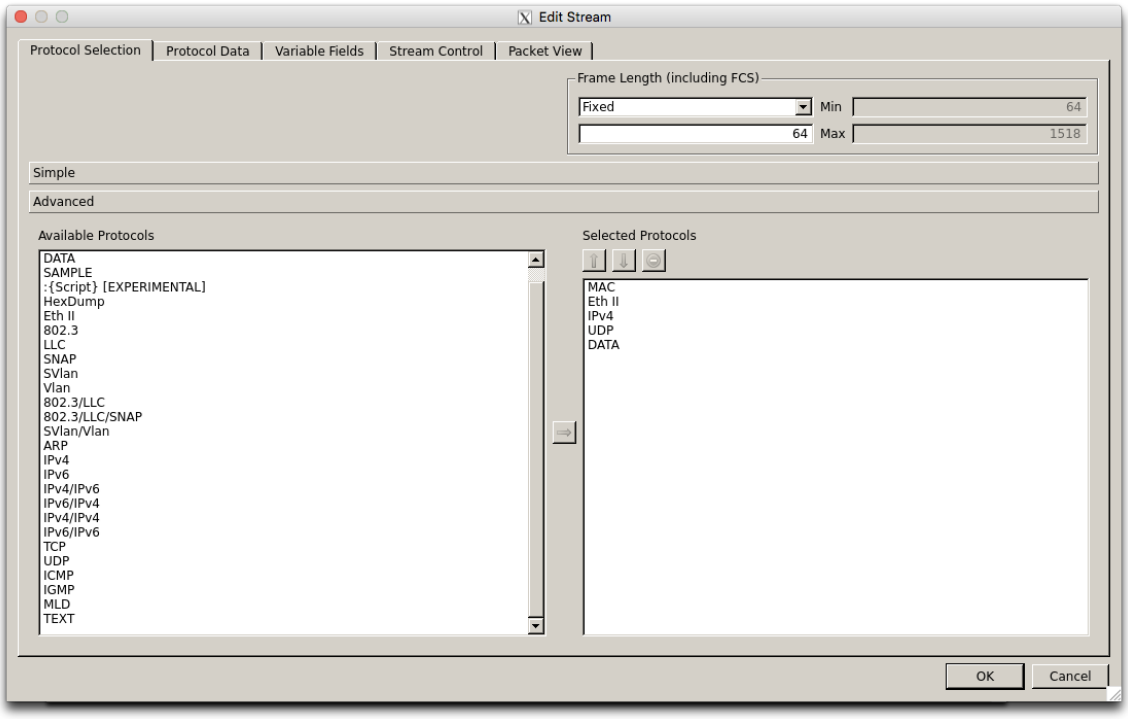


Figure 8: Advanced setup in protocol

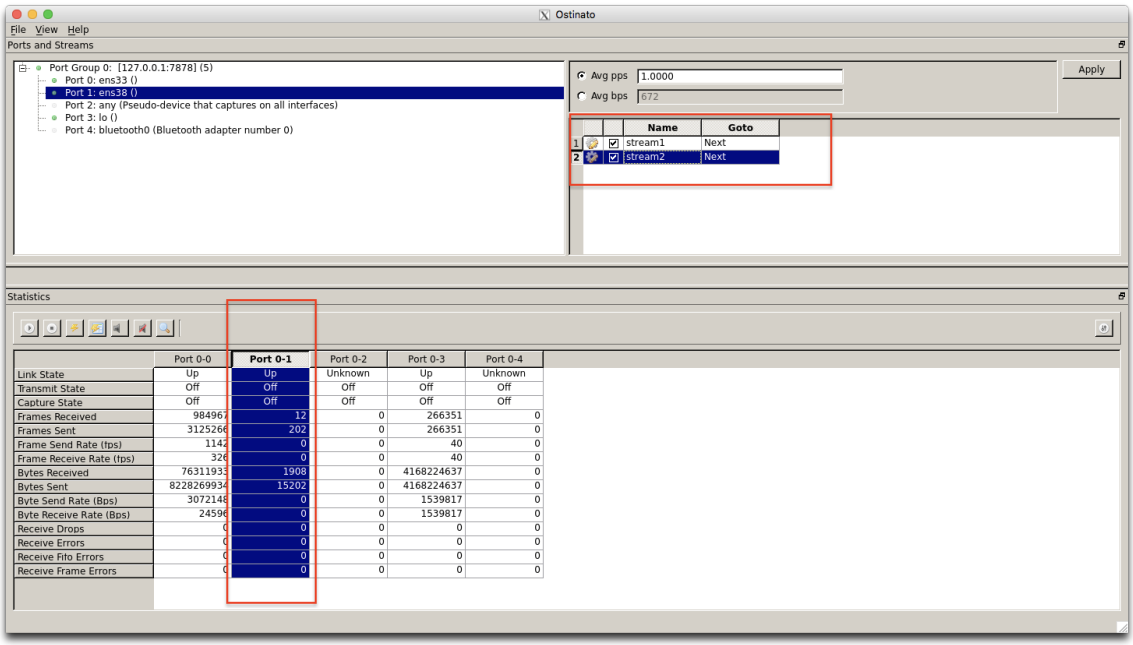


Figure 9: 2 Streams transmission and the results

V. Capture

In my sink device, I also start the Ostinato tried to capture the packets, this operation could be done by select the port and click on the "Capture" button, then we can wait for the packets this particular port received. Seen in Figure 10

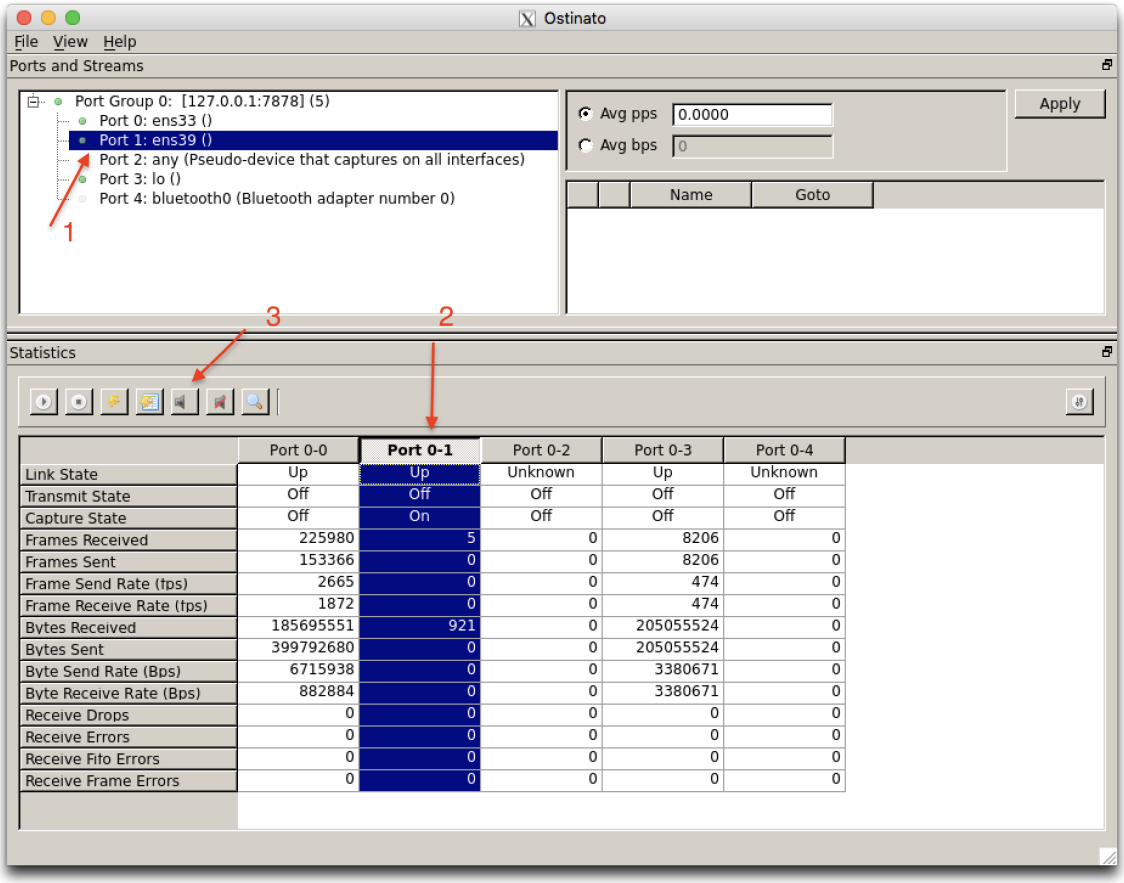
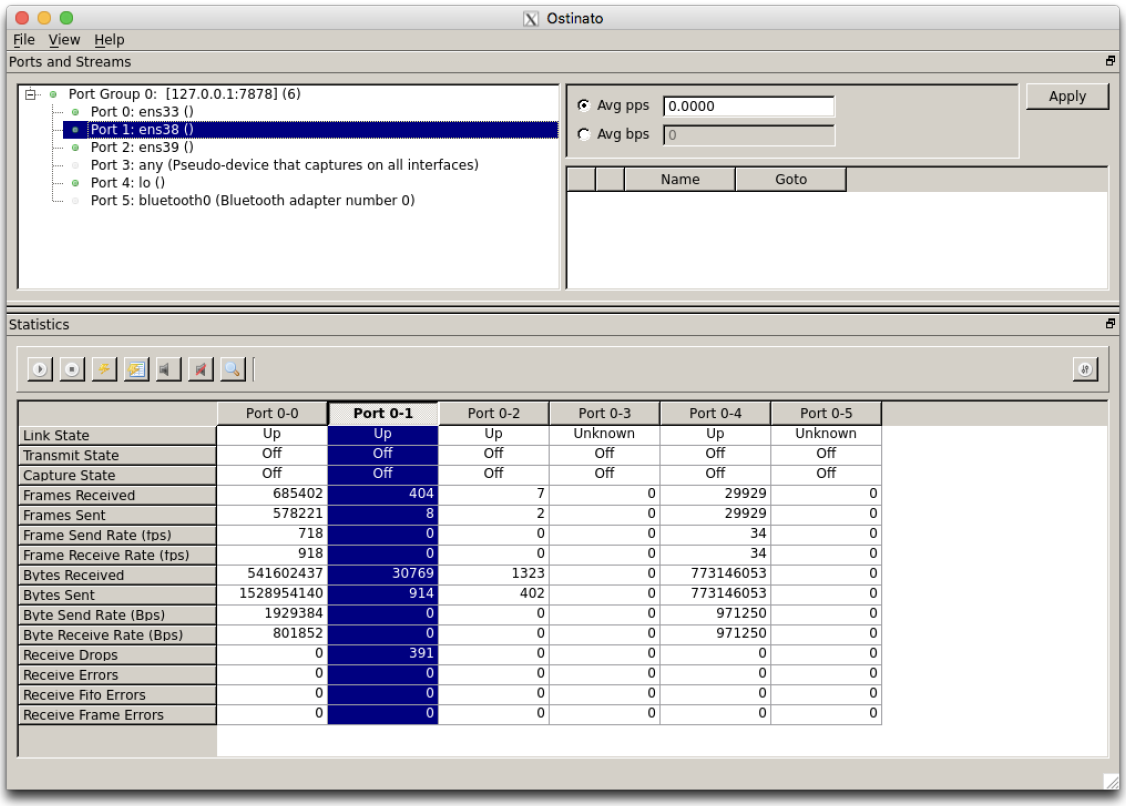


Figure 10: Capture Packets

VI. Analysis

When the capture process has been on for a while, we can manually stop it and see the packets information using the "View Capture Buffer" button, it will automatically start the Wireshark on the selected port and you can see the packets send to the port with detailed information, since I haven't figured out how to equipped it with DPDK, so I tried send 400 packets in UDP protocol from the left VM(`192.168.104.129`) to NAT device(`172.16.216.128`). I analyzed it when the port received 404 packets and the results can be shown in Figure12



	Port 0-0	Port 0-1	Port 0-2	Port 0-3	Port 0-4	Port 0-5
Link State	Up	Up	Up	Unknown	Up	Unknown
Transmit State	Off	Off	Off	Off	Off	Off
Capture State	Off	Off	Off	Off	Off	Off
Frames Received	685402	404	7	0	29929	0
Frames Sent	578221	8	2	0	29929	0
Frame Send Rate (fps)	718	0	0	0	34	0
Frame Receive Rate (fps)	918	0	0	0	34	0
Bytes Received	541602437	30769	1323	0	773146053	0
Bytes Sent	1528954140	914	402	0	773146053	0
Byte Send Rate (Bps)	1929384	0	0	0	971250	0
Byte Receive Rate (Bps)	801852	0	0	0	971250	0
Receive Drops	0	391	0	0	0	0
Receive Errors	0	0	0	0	0	0
Receive Fifo Errors	0	0	0	0	0	0
Receive Frame Errors	0	0	0	0	0	0

Figure 11: Capture 404 packets on port ens38

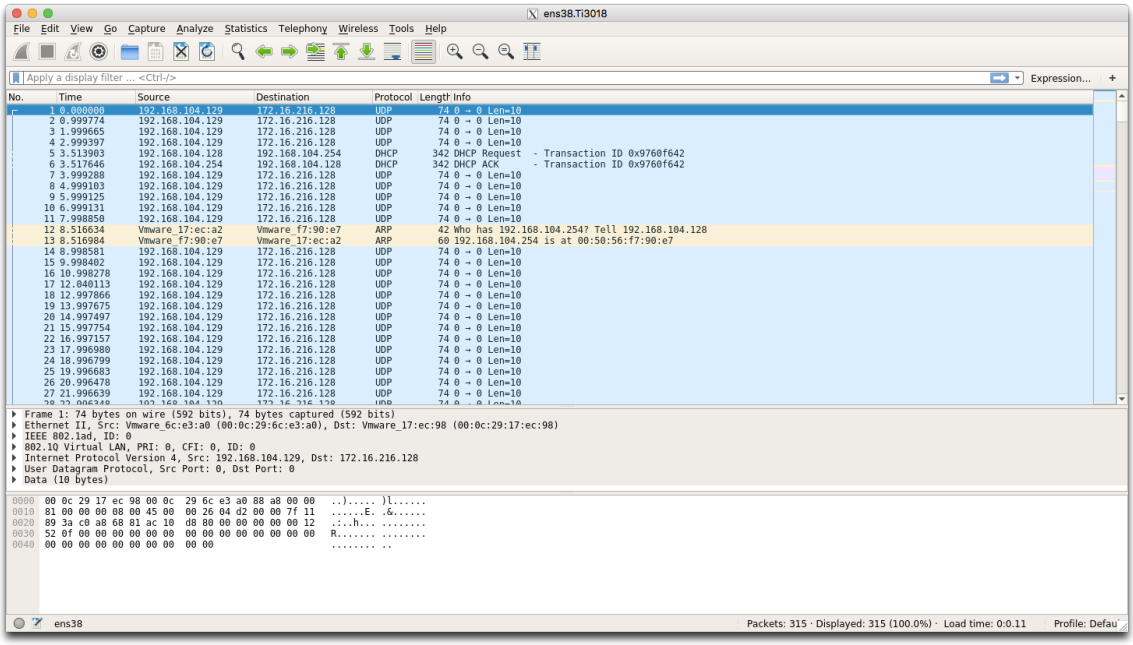


Figure 12: Analyze Packets