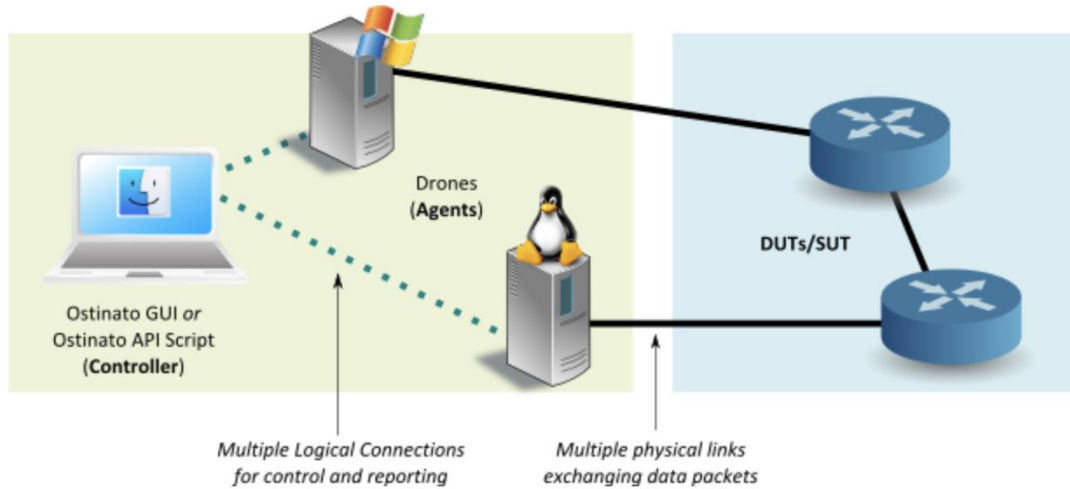


**GROUP-5**  
**Software Engineering Lab**  
**Assignment 2 Report**

**B05-511017020**  
**B05-510417090**  
**510517063**  
**510517010**

**Anjishnu Mukherjee**  
**Sunirban Sarkar**  
**Sankhasubhra Mandal**  
**Samrat Sarkar**

# Ostinato - Packet Generator



## What is Ostinato?

**Ostinato** is a popular network traffic and packet generator. This tool has a friendly GUI and boasts a Python API for automating network tests. It allows you to create your own traffic streams and offers surprising customization capabilities. The user interface is well designed, with many dynamic graphical representations.

Ostinato boasts features covering a wide scope of functionalities, which makes it a versatile tool. The features are intelligently designed and easy for the average user to get a handle on. One can not only create their own streams but can also configure them in an in-depth way, with consideration of stream rates, bursts, and number of packets. This is a tool suited to network load testing and functional testing. It affords the ability to visualize data on a "per stream" basis, giving an accurate look at packet loss measurements at a granular level.

Ostinato is also great for achieving real-time network measurement and monitoring. Its efficient and economical design enables the user to receive and transmit stats and rates at the level of the interface. There is also the fact that Ostinato is highly compatible and supportive. It covers most protocols, including VLAN, ARP, IPv4 and IPv6, IP in IP, TCP and UDP, ICMPv4 and ICMPv6, IGMP, MLD, IEEE 802.3 LLC, SNAP, and text-based protocols.

With Ostinato one can stack protocols in random order to test error cases and vary packet fields across packets at the runtime stage—for example, by changing the IP/MAC addresses. The tool features user-defined scripting, which can be utilized to substitute a protocol that hasn't been implemented. The user can also open, edit, replay, and save PCAP files. And despite there being only one controller, we can have multiple agents.

## What is it for?

Networking Protocol Development and Analysis

Functional Testing

Performance Testing

Security Testing

Penetration Testing

## Who is it for?

Programmers/Developers

QA Testers

Security Researchers

A brief idea about the target users of Ostinato.

## Architecture & Deployment options

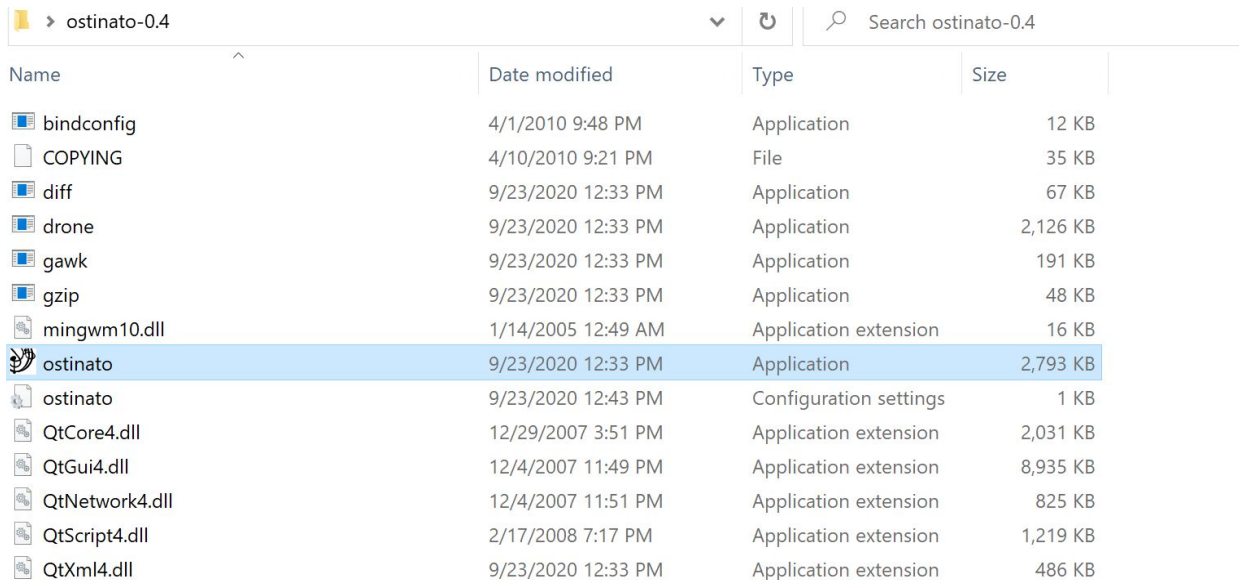
The image on the previous image gives an introduction to a very common setup for Ostinato-Drone across multiple Operating Systems. Basically Ostinato has a controller-agent architecture. Ostinato is the controller and Drone is the agent. The controller is available as a GUI application or via a python-ostinato API. And the binary for Drone is available separately. The agent does all the heavy lifting of traffic generation and capture. The controller instructs the agent and fetches reporting data like statistics etc. from the agent. Since the agent does the packet generation, hence the DUT (Device Under Test) is connected to the agent and not to the controller.



This image gives a pictorial representation of all architecture schemas for Ostinato.

## Installing ostinato

A zip file containing the executable binary along with all the dependencies can be downloaded directly from the [website](#). Unzipping the file, we can directly double-click the binary in the folder to run it.



ostinato-0.4				
Name	Date modified	Type	Size	
bindconfig	4/1/2010 9:48 PM	Application	12 KB	
COPYING	4/10/2010 9:21 PM	File	35 KB	
diff	9/23/2020 12:33 PM	Application	67 KB	
drone	9/23/2020 12:33 PM	Application	2,126 KB	
gawk	9/23/2020 12:33 PM	Application	191 KB	
gzip	9/23/2020 12:33 PM	Application	48 KB	
mingwm10.dll	1/14/2005 12:49 AM	Application extension	16 KB	
ostinato	9/23/2020 12:33 PM	Application	2,793 KB	
ostinato	9/23/2020 12:43 PM	Configuration settings	1 KB	
QtCore4.dll	12/29/2007 3:51 PM	Application extension	2,031 KB	
QtGui4.dll	12/4/2007 11:49 PM	Application extension	8,935 KB	
QtNetwork4.dll	12/4/2007 11:51 PM	Application extension	825 KB	
QtScript4.dll	2/17/2008 7:17 PM	Application extension	1,219 KB	
QtXml4.dll	9/23/2020 12:33 PM	Application extension	486 KB	

For Ubuntu, we can directly use the command `sudo apt install ostinato` from the terminal to install the last available free version of the software.

For MacOS, we can download the application from the website and then install it after providing necessary security permissions for accessing network interfaces.

## Common Issue after Installation - Running ostinato the first time

The displayed port group might not have any interfaces, indicated by a 0 beside the port number. Usually this is because drone is not running as expected. Basically, you need to ensure that drone is running with required privileges

## LINUX

- Easy: If the terminal supports sudo use `sudo ostinato`, otherwise become root by doing `su` and then run ostinato. Doing this provides full root privileges to both ostinato and drone which is unnecessary and insecure.

- Better: Do a [one-time privilege assignment](#). After doing that you can run ostinato as a normal user.

## MAC

- Easy: The not-recommended way of doing this is to use the command `sudo /Applications/Ostinato/Ostinato.app/Contents/MacOS/Ostinato`. Doing so provides full root privileges to both ostinato and drone which is unnecessary and insecure.
- Better: Install [Wireshark](#). It is a great tool to have alongside Ostinato. During Wireshark installation, it creates a startup task that sets up appropriate ownership and permissions to the `/dev/bpf*` devices so that capture/transmit applications such as Wireshark/Ostinato can be invoked normally without sudo. In other words, if you can run Wireshark without sudo, you can run Ostinato without sudo.

## WINDOWS

- Ostinato uses [WinPcap](#) on Windows. Typically WinPcap is installed to run at startup and applications using WinPcap don't need anything special to be done. If however, this isn't the case for you, you can right click on the Ostinato application and select "Run as administrator".

## Preview of Features

Once we are done with the installation and configuring of ports, we are ready to use it as a standalone application for many different kinds of tasks, the most basic one being highly customizable packet transfer. Let's see an example, where we use ostinato to craft and send a stream of ICMP packets.

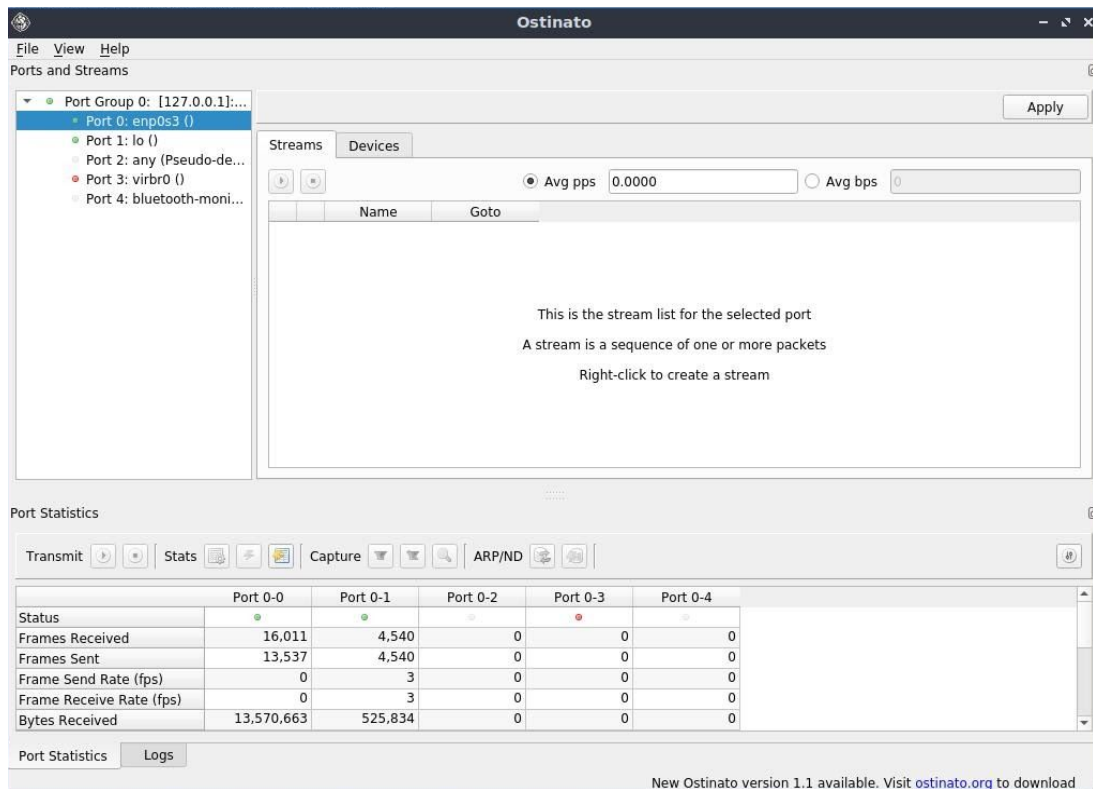
```

~$ ifconfig enp0s3
enp0s3: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::ff7:caa8:8381:4664 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:cc:b0:f4 txqueuelen 1000 (Ethernet)
RX packets 28425 bytes 25713844 (25.7 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 22478 bytes 3458583 (3.4 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

~$ arp -a
_gateway (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:db:42:67 [ether] on enp0s3

```

The screen below is the one we are greeted with when opening Ostinato. Do note that if there are no port groups or no available ports under a group, then you need to go back to the previous step and configure the application as above.



If you already can see port groups, then select a port and move on to configure the packet streams. The next few images are sequential and show all the steps of this configuration process in order.

Ostinato

File View Help

Ports and Streams

Port Group 0: [127.0.0.1]:...

- Port 0: enp0s3 ()
- Port 1: lo ()
- Port 2: any (Pseudo-de...
- Port 3: virbr0 ()
- Port 4: bluetooth-moni...

Streams Devices

Avg pps 0.0000 Avg bps 0

Name Goto

This is the stream list for the selected port  
A stream is a sequence of one or more packets  
Right-click to create a stream

Port Statistics

Transmit Stats Capture ARP/ND

	Port 0-0	Port 0-1	Port 0-2	Port 0-3	Port 0-4
Status					
Frames Received	16,011	4,540	0	0	0
Frames Sent	13,537	4,540	0	0	0
Frame Send Rate (fps)	0	3	0	0	0
Frame Receive Rate (fps)	0	3	0	0	0
Bytes Received	13,570,663	525,834	0	0	0

Port Statistics Logs

New Ostinato version 1.1 available. Visit [ostinato.org](http://ostinato.org) to download

Ostinato

File View Help

Ports and Streams

Port Group 0: [127.0.0.1]:...

- Port 0: enp0s3 ()
- Port 1: lo ()
- Port 2: any (Pseudo-de...
- Port 3: virbr0 ()
- Port 4: bluetooth-moni...

Add Stream

Protocol Selection Protocol Data Variable Fields Stream Control Packet View

Basics

Name trial

Enabled

Frame Length (including FCS)

Fixed 64 Min 64 Max 1518

Simple

L1

- None
- Mac
- Other

L2

- None
- Ethernet II
- 802.3 Raw
- 802.3 LLC
- 802.3 LLC SNAP
- Other

L3

- None
- ARP
- IPv4
- IPv6
- IP 6over4
- IP 4over6
- IP 6over6
- Other

Payload

- None
- Pattern
- Hex Dump
- Other

VLAN

- Untagged
- Tagged
- Stacked

L4

- None
- ICMP
- IGMP
- MLD
- TCP
- UDP
- Other

Special

- None
- Signature

L5

- None
- Text
- Other

Trailer

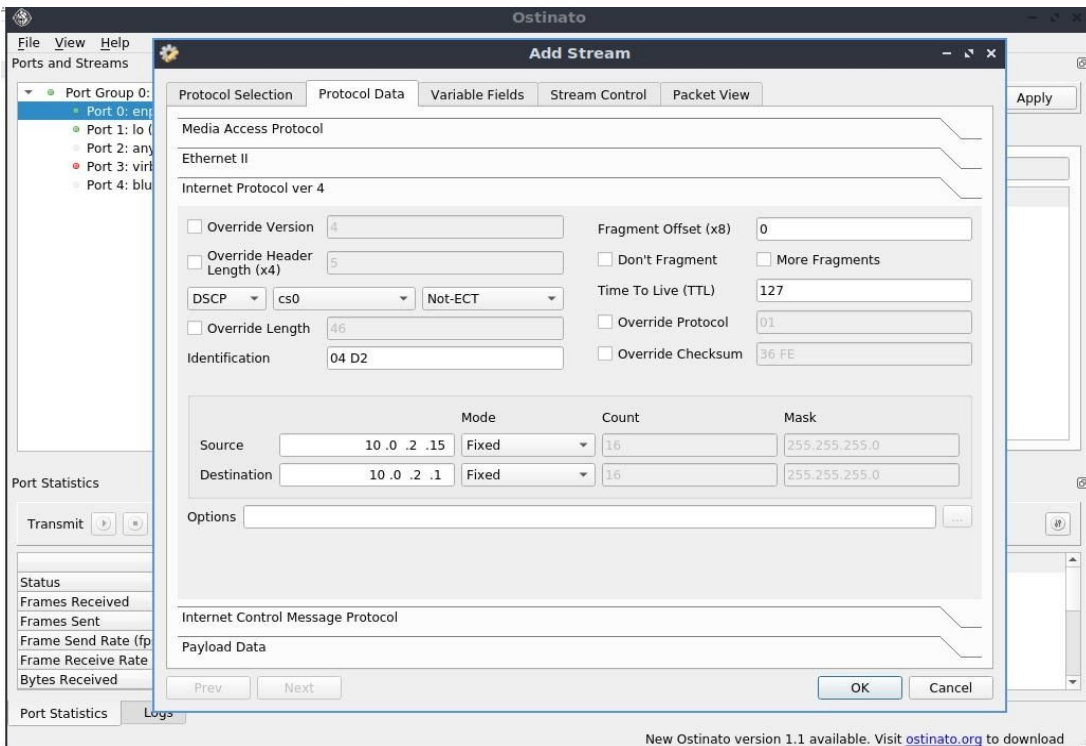
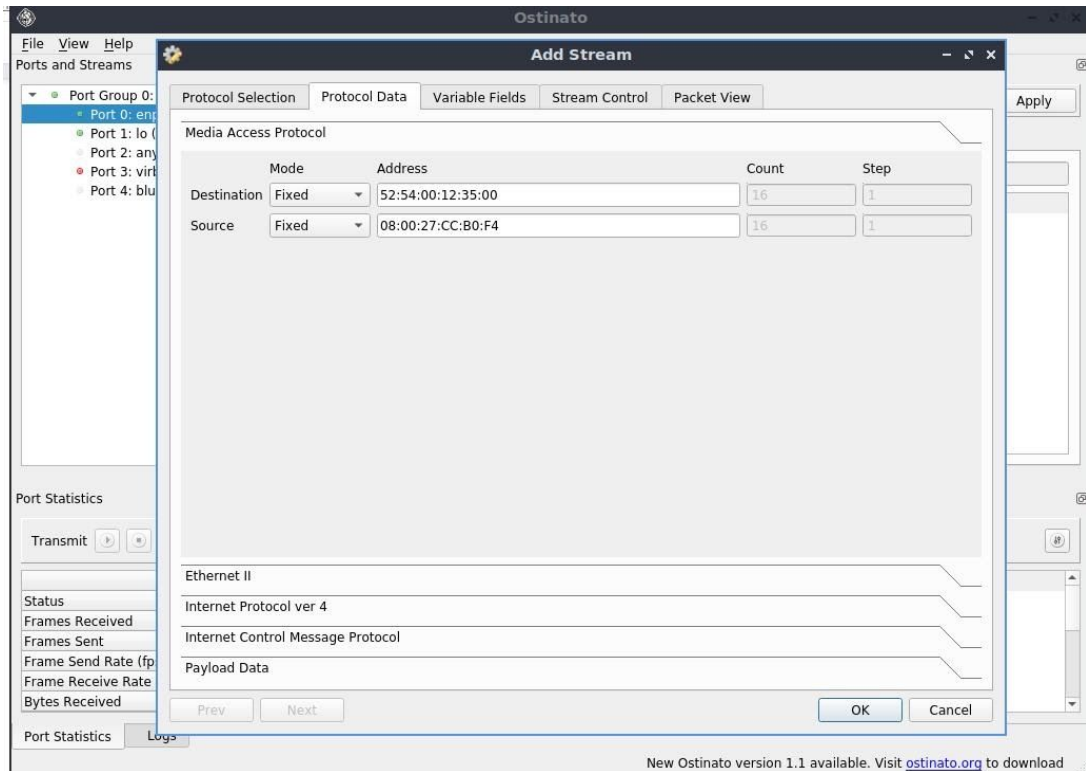
- None
- Other

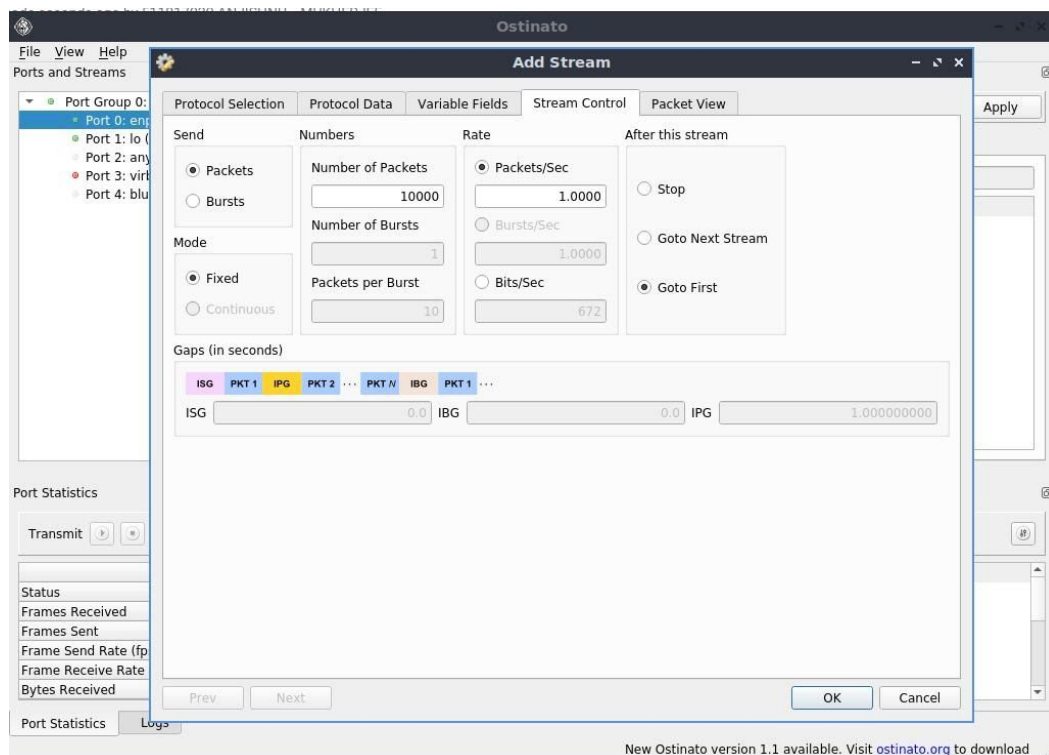
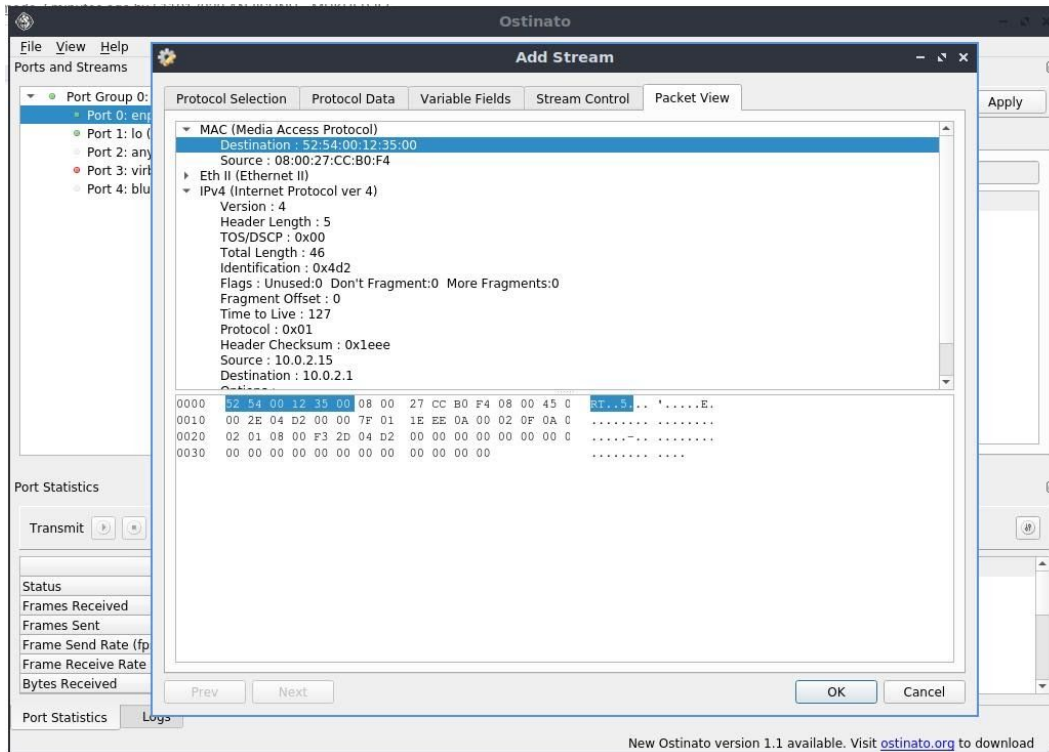
Advanced

Prev Next OK Cancel

New Ostinato version 1.1 available. Visit [ostinato.org](http://ostinato.org) to download







The image displays two screenshots of network tools. The top screenshot shows the Ostinato application window, which is used for configuring and transmitting network packets. The bottom screenshot shows the Wireshark application window, which is used for capturing and analyzing network traffic.

**Ostinato Screenshot:**

The Ostinato window has a menu bar (File, View, Help) and a toolbar. The main area is divided into two panes: "Ports and Streams" on the left and "Streams" on the right. The "Ports and Streams" pane shows a tree view of ports, including "Port 0: enp0s3 ()", "Port 1: lo ()", "Port 2: any (Pseudo-de...", "Port 3: virbr0 ()", and "Port 4: bluetooth-moni...". The "Streams" pane shows a table of streams with columns "Name" and "Goto". A stream named "trial" is listed with the "Goto" value "Goto first". Below the streams pane, there are input fields for "Avg pps" (1.0000) and "Avg bps" (672). The "Port Statistics" section at the bottom shows a table of statistics for various ports.

	Port 0-0	Port 0-1	Port 0-2	Port 0-3	Port 0-4
Status	●	●	●	●	●
Frames Received	71	105	0	0	0
Frames Sent	84	105	0	0	0
Frame Send Rate (fps)	5	3	0	0	0
Frame Receive Rate (fps)	6	3	0	0	0
Bytes Received	21,773	12,741	0	0	0

**Wireshark Screenshot:**

The Wireshark window shows a packet capture on the "enp0s3.GRTVTN" interface. The packet list pane shows a list of ICMP Echo (ping) requests and replies. The packet details pane shows the details of the selected packet (No. 1), including the Ethernet II header, Internet Protocol Version 4 header, and Internet Control Message Protocol header. The packet bytes pane shows the raw data of the packet.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.15	10.0.2.1	ICMP	60	Echo (ping) request
2	0.000348	10.0.2.1	10.0.2.15	ICMP	60	Echo (ping) reply
9	1.000021	10.0.2.15	10.0.2.1	ICMP	60	Echo (ping) request
10	1.000198	10.0.2.1	10.0.2.15	ICMP	60	Echo (ping) reply
15	2.000019	10.0.2.15	10.0.2.1	ICMP	60	Echo (ping) request
16	2.000314	10.0.2.1	10.0.2.15	ICMP	60	Echo (ping) reply
25	3.000019	10.0.2.15	10.0.2.1	ICMP	60	Echo (ping) request
26	3.000280	10.0.2.1	10.0.2.15	ICMP	60	Echo (ping) reply

Packet Details (No. 1):

- Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
- Ethernet II, Src: PcsCompu\_cc:b0:f4 (08:00:27:cc:b0:f4), Dst: RealtekU\_12:35:00 (52:54:00:12:35:00)
- Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.1
- Internet Control Message Protocol

Packet Bytes:

```

0000  52 54 00 12 35 00 08 00 27 cc b0 f4 08 00 45 00  RT...5...E...
0010  00 2e 04 d2 00 00 7f 01 1e ee 0a 00 02 0f 0a 00  ....
0020  02 01 08 00 f3 2d 04 d2 00 00 00 00 00 00 00 00  ....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....

```

As we can see, we can configure every aspect of the stream at first, and then view the packets once before we start the transmission. Then the statistics section of the tool shows us all the information needed about every frame. Alternatively, we can just open Wireshark, as above and observe the packets being sent.

## Advanced Use Case - Device Emulation

Ostinato version 0.8 onwards, users can emulate multiple devices/hosts with ARP/NDP and ping support. Streams can be configured to resolve their MAC addresses using ARP/NDP if corresponding device(s) are configured.

The screenshot shows the Ostinato configuration window with the 'Protocol Data' tab selected. The 'Media Access Protocol' section is active, displaying a table for configuring MAC addresses. The table has four columns: Address, Mode, Count, and Step. There are two rows for 'Destination' and 'Source' addresses, both set to '00 00 00 00 00 00', 'Resolve' mode, and a count of 16. Below the table, a note states: 'Please ensure that a corresponding device is configured on the port to enable source/destination mac address resolution. A corresponding device is one which has VLANs and source/gateway IP corresponding to this stream.' The bottom of the window shows a protocol stack: Ethernet II, Internet Protocol ver 4, User Datagram Protocol, and Payload Data.

	Address	Mode	Count	Step
Destination	00 00 00 00 00 00	Resolve	16	1
Source	00 00 00 00 00 00	Resolve	16	1

Please ensure that a corresponding device is configured on the port to enable source/destination mac address resolution. A corresponding device is one which has VLANs and source/gateway IP corresponding to this stream.

Ethernet II

Internet Protocol ver 4

User Datagram Protocol

Payload Data

Before triggering stream transmit, ARP/NDP resolution needs to be invoked using the Resolve Neighbors button in the Statistics Window. The agent will resolve all the Gateway IP addresses and all the destination IP addresses configured in each stream by sending a ARP/NDP request to the DUT. If ARP/NDP is not resolved (or failed), 00:00:00:00:00:00 will be used as the MAC address. You can check if all ARP/NDP are resolved in the Device Information pane.




In the reverse direction, if the DUT sends ARP/NDP request for any of the emulated device's IP address, the agent will reply back with a suitable ARP/NDP reply. Similarly, if the DUT sends a ping (IPv4 or IPv6) request, it would reply with a ping reply.

## Advantages







Ostinato is a feature-heavy tool. There are many different advantages of Ostinato, the most basic one being that it has a very simple GUI and setup process for the complicated task of network testing. Apart from that, the following are other noteworthy points.

- Scales well, from low-end to high-end use cases
- Viable alternative to commercial tools
- Cost savings
- Productivity improvement
- Facilitates new testing scenarios
- Frees up ports for where they are really needed

## Comparison with existing solutions

Low-End Freeware		Medium-End Open Source		High-End Commercial	
					
→ PC → Win Only → Simplistic → Crippled → Don't scale		→ Cross Platform → CLI (mostly) → Limited Protocols → Limited Controls → Narrow Focus → No Controlled Environment		→ Fully Featured → Dedicated H/W → Expensive => Insufficient → Overkill	

	Free Trial?	Top Features			Bottom Line
SolarWinds WAN Killer Network Traffic Generator 	14-Day	Define traffic for proactive testing	Flexible, straightforward tool	Integrates with other ETS tools	Part of Engineer's Toolset, WAN Killer allows for custom traffic generation and analysis.
Packet Sender 	Free, Open Source	Uses command line and GUI	Mobile versions	Test network APIs	Sends TCP, UDP, and SSL on your chosen ports for custom testing.
Nping 	Free, Open Source	Create custom packets	Troubleshoot firewalls/routing	Includes echo mode	Multifunctional, from ping to RAW packets, DoS, tracer, and ARP poisoning.
Ostinato 	Free, Open Source	Friendly GUI	User-defined scripting	Not for website traffic generation	An easy-to-use tool for load testing and functional testing.
NetScanTools Pro 	30-Day	Packet generating and crafting	Save and play back past files	Primarily UDP	Generator and Flooder tools offer simple and useful traffic generation.
TRex 	Low-Cost, Open Source	Stateless traffic streams	Various support types for streams	Fairly versatile	For experienced users, this tool offers a decent range of customization options.

## Drawbacks

Ostinato is a feature-heavy tool. There are, however, a few important functions it can't do. The following are some of the noteworthy mentions.

- It's not possible to use ostinato for generating fake traffic and sending it to websites.
- Ostinato is stateless, which means it doesn't support stateful connection-oriented TCP connections.
- Tracking stream level statistics is CPU intensive and may have an impact on max transmit rate.
- Tx Stream Statistics is counted only after transmit is finished. Fetching stats before transmit is finished will return 0 values.

## Read the Docs

Ostinato has a very good documentation website, both for [end users](#) and people who want to contribute to the development of the software. Since the software is available [open-source on GitHub](#), and can be built locally very easily using the [Qt application builder](#), developers can easily set up a dev environment to contribute features that they want to be included in the next iterations of the library. Apart from this, the primary [author](#) of the library leads efforts in spreading understanding of different common use cases via slide-decks, blog posts and youtube videos on different forums all over the internet. These are very helpful from an end user perspective, and will allow most users to get started easily with the common use cases.



Ostinato is a packet crafter, network traffic generator and analyzer with a friendly GUI and powerful Python API for network test automation.