IXNETWORK-NGPF QUICK REFERENCE GUIDE



TABLE OF CONTENTS

1.	Overv	ew	2
2.	Config	ure BGP through GUI	2
	2.1.	Add Chassis and Lock Ports	2
	2.2.	Add Topology	4
	2.3.	Emulate a Protocol	5
	2.4.	Device Group Multiplier	6
	2.5.	Edit Protocol Grid	.6
	2.6.	Configure BGP	7
	2.7.	Add Network Group	7
	2.8.	Start Protocols	.8
	2.9.	Configure Traffic	8
	2.10.	Add Endpoints to Traffic	9
	2.11.	Edit Packet and Setup Flow Groups	10
	2.12.	Setup Frame Size and Rate	11
	2.13.	Setup Flow Tracking and Protocol Behavior	. 12
	2.14.	Validate Traffic	13
	2.15.	Apply Traffic, Start Traffic, and Statistics View	.14
3.	Config	ure BGP through Automation (HLPyAPI)	.15
	3.1.	Initialize Environment	15
	3.2.	Add Chassis and Lock Ports	15
	3.3.	Create Topology and DeviceGroup	16
	3.4.	Create Ethernet Stack	17
	3.5.	Create Ipv4 Stack	18
	3.6.	Create BGP	19
	3.7.	Create Network Group	20
	3.8.	Start Protocols	.22
	3.9.	Enable Filter and Apply Changes on the Fly	.22
	3.10.	Retrieve Learned Info	23
	3.11.	Configure Traffic	23
	3.12.	Start Traffic and Get Statistics	24
4.	Other	Utilities	25
	4.1.	IxNetwork API Documentation Browser	25
	4.2.	Script Gen	26
	4.3.	F1 Option	27
5.	To Kno	ow More On NGPF	28
6.	Suppo	ort	28

1. Overview:

- NextGen Protocol Framework (NGPF) is Ixia's new protocol framework.
- > Upgraded from the classic protocol framework.
- > Built to meet and stay ahead of customer requirements in flexibility and scalability.
- > Designed to provide consistent and visual workflow across all protocols.
- > Designed to more closely simulate dynamic customer environments.
- Industry leading access, routing and SDN protocol coverage.
- Realistic subscriber emulation of mixed single and dual-stack subscribers.
- > Flexibility of scaling the number of emulated devices by using the multiplier feature.
- Granular session control by using configuration grids.
- System level statistics dashboards with on-demand drill-downs.
- Comparable feature set with IxN2X.

2. Configure BGP through GUI:

This section will walk through a scenario which configures BGP emulation manually to get the user introduced with most of the basic features of NGPF.

2.1 Add Chassis and Lock Ports:

The Port Selection window allows you to manage the ports



Fig 2.1 Selecting Ports



Select chassis by entering chassis IP or select chassis from the list of recently used chassis and click Connect all checked to add them to the configuration.

Nw Po	t Selection			
Chassi	s 🖶 Add Chassis 🔛 🔛 🖃 More 🗸	All ports 💌	Ports in configuration	🕂 Add Offline Ports
	Click Add Chassis		State	Name
	Recently Used Chassis			
G	<u>192.168.52.128</u>	>		
	Connect all checked	Nw Add Chassis		×
		Chassis Name or IP 10.13	4.162.46	
		For multipl	le chassis use comma	Enter Chassis IP
		192.168.52.128		
		3	Click OK	
		OK	Cancel Help	

Fig 2.1.1 Selects chassis by entering chassis IP

🛛 🗋 🗁 🔚 • 🏟 • 🛪 • 🖕 4	2 + 💿 💭 + 🔲 🛅 🤹 International Internati	
File Home Automation	Results / Reports Views	
Protocols All QuickTest	Imposer Script Imposer Script Appure Add Add Add Resource Ports* Protocols* Traffic * QuickTests * Manager* Options Statistics*	
Run	w Port Selection	- 🗆 X =
0 Overview	Chassis 🐥 Add Chassis 🎇 😒 🐌 🖃 More 🗸 🛛 All ports 🔹	All ports 👻
📽 Genario		
🕶 💮 Ports	Chassis/CardyPort Type Owner State Walle Chassis/CardyPort Value Chassis/CardyPort	Resource Group
🔓 Chassis	Port 01 2 PORT kita Vintual Load Module 2 Port 01 kita Vintual Port 2	
Protocols	Port 02 Ixia Virtual Port Add ports	CT.
≫ Traffic	Click Add Ports	
Impairments	1 Select Ports Assign to	
🕂 QuickTests	remaining	-
🖗 Captures	JPM Assign to	
	selected	
	selected	
		1t
	3	Click OK
		\sim
		}
	ок	Cancel Help

Fig 2.1.2 Port selection



2.2 Add Topology:

An IxNetwork instance supports one Scenario, which can contain multiple topologies. Each Topology is a collection of one or more test ports. Each port in a Topology is bound to a physical or virtual port and individual ports can be added or removed.



Nw Protocol Wizard			- 🗆 X
Ixia Ports	Ixia Ports		<u>IxNetwork</u>
	Chassis 🔮 Add Chassis 📡	All ports 💌	Ports in New Topology
	Chassis/Card/Port Type ✓ ④ 192.168.52.128 IxOS 8.30.1350.13 ▶ ■ Card 01 2 PORT Ixia Virtua	Owner EA, Protocol8.30 Load Module	Status Name Chassis/Card/Port
	Page Carol UI 2 POKT IXia Virtua	Append ports Assign to selected	2 Click Append ports
	Ports in Configuration + Add Offline Port	Remove selected	
	Status Name Chassis// 2 Ethernet - VM - 001 192.168.52 1 Ethernet - VM - 002 192.108.52	2128:01:02 .128:01:02 Select Port	
Z			<i>b</i> ₃
$\underline{\times}$		\langle	Click Next : 3
			Prev Next Cancel

Fig 2.2 Topology with selected ports



2.3 Emulate a Protocol:

- The Protocols page in the Protocol Wizard allows you to select the protocols in the Topology. The Protocols page lists the available Classic and NextGen protocols under respective tabs. Click NextGen, select required protocol for the test.
- > Presents all supported protocols in Next Generation Protocol Framework in a single window.

w Protocol Wizard							- ō X
kia Ports Initial Ports Protocols	Protocols Classic All the protocols you know with familiar features	n scaling, more control					<u>lxNetwork</u>
	Quick Select Protocols Protocol Wizards	Filter: 1	Click NextGen				
		Ethernet GREoIPv4	☐ IPv4 ☐ GREoIPv6	IPv4 Loopback	☐ IPv6	IPv6 Loopback	
		Access ACCP DHCPv6 Relay Agent L2TP Network Server Data Center Ethernet	DHCPv4 Client Lightweight DHCPv6 Relay Agent PPPoX Client	DHCPv4 Relay Agent DHCPv6 Server PPPoX Server	DHCPv4 Server	DHCPv6 Client L2TP Access Concentrator	
		Fabric-Path BGP EVPN VXLAN VPWS MPLS	TRILL	UXLAN	Geneve	BGP EVPN VXLAN	
0		Basic LDP RSVP-TE	☐ mLDP ☐ RSVP-TE (P2MP)	Basic LDPv6	Targeted LDP	Targeted LDPv6	
2		LDP PW/VPLS BGP AD for LDP VPLS Multicast	LDP Other PWs BGP EVPN	LDP VPLS using BGP AD BGP PBB-EVPN	BGP VRF/6VPE	BGP VPLS L2Site	
Û		IGMP Host	iPTVv4 PIMv4	GMP Querier	Click Lequired Protocol	IPTVv6	
Ζ		BFDv4	☐ BFDv6 ☐ OSPFv3	K BGP 2	BGP+	<u> </u> ISIS-1.3	
<u> </u>		PCEv4 OVSDB Server	PCCv4 Custom Protocol	OpenFlow Controller IPv6 SR Ext	OpenFlow Switch	OVSDB Cont Click Finish	3
						Prev	Finish Cancel

Fig 2.3 Selected BGP protocol

2.4 Device Group Multiplier:

- A Device Group has similar Devices per test port. A Device can be a router, host, switch, and so on. It can run multiple protocols and protocol stacks.
- ➤ A Device Group count is the number of instances in the group. A configuration can be scaled by modeling a group of *n* Devices per test port by changing the multiplier.

Topology 1	Click here	New Topology
BGP Peer 1 IPv4 1 Ethernet 1	Device Group Multiplier Multiplier 10 Per Port 10 Total 10 OK Cancel	Edit Multiplier

Fig 2.4 Emulate number of devices by using device group multiplier

2.5 Edit Protocol Grid:

The protocol stack displayed in the Scenario view is interactive. Click on a particular protocol stack, edit the values according to the requirement.

Topology 1	1x Image: Second seco
Details for IPv4 1	
Protocol Settings > < Pv4 Global Settings > < IPv4 Grouping Device Group Topology Device # Status Session Int Pv4 1: 1 ports Device Group 1 Topology 1 1 Not Started C Ethernet - VM - 001 Device Group 1 Topology 1 1 # 1 Not Started All IPv4 Select Views	fo Address Pro address address address address 000.10.2 2 3 20000000000000000000000000000000

Fig 2.5 Configuring interface sections



2.6 Configure BGP:

- Configure Interface IP Address to 20.20.20.2 and Gateway Address to 20.20.20.1 in device group 1 IP stack by using method 2.5. Configure Interface IP Address to 20.20.20.1 and Gateway Address to 20.20.20.2 in device group 2 IP stack using by method 2.5
- Similarly configure Local IP to 20.20.20.2 and DUT IP to 20.20.20.1 in BGP Stack in device group 1 by clicking the BGP stack. Configure Local IP to 20.20.20.1 and DUT IP to 20.20.20.2 in BGP Stack in device group 2 by clicking the BGP stack.

2.7 Add Network Group:

- A Network Group represents a set of L3 networks (sub-netted or switched) with optional information explaining the reachability to each of these networks.
- All Devices connected to a Network Group must belong to one of the networks modeled by that Network Group.



Fig 2.7 Route Profile addition by using network groups



2.8 Start Protocols:

> Click **Start All** to start all the protocols configured in the test.

Nw 🗋 🥭	🔚 - 🗳	• 🏹 • 🎥 🖉	• 🥘 👷 •		\$ 3} -		Scen	ario Too	ls	Ix	Network (TCL
File	Home	Automation	Results /	Reports	Views	Net	twork Gr	oup	Layout		
Protocols S	cenario Item •	tions Apply Changes	New Topology	New	Protocol Network Group	p (Protocol Options	Delete	Paste	/ ime	Grid Operation
Start A				Dune				Edit	•		
Stop A		Click	Start All					То	pology 1		
R Scen	nario -						etwork G	iroup 1 ess Pool	1x		Device Gro 1 device
Porte	s Chassis					BG	P 		<u></u>		BGP Peer 1 IPv4 1
- 🔂 Prote	ocols amework				 				L		Ethernet 1
0.	Protocol Int	Dot Dot	aile for Notu	work Gro	up 1						

Fig 2.8 Brings up all protocol stacks

2.9 Configure Traffic:

The Advanced Traffic Wizard helps to integrate the options for traffic configuration in the control plane and data plane of IxNetwork, thereby facilitating quick setup of large scale testing.



Fig 2.9 Configures L2-3 traffic

2.10 Add Endpoints to Traffic:

- The Endpoints dialog is the first dialog in a series that form the Advanced Traffic Wizard. To access the Endpoints dialog, click the Endpoints tab in the left pane of Advanced Wizard window.
- > The Endpoints dialog opens to display the options for selecting the traffic endpoints.



Fig 2.10 Configures source and destinations endpoints set

2.11 Edit Packet and Setup Flow Groups:

*Editing the packet and setting up flow groups is optional.



Fig 2.11 Customizing the packet and creating flow groups

Prev

Next

Einish

<u>C</u>ancel

Help

2.12 Setup Frame Size and Rate:

*Setting up the frame Size and Line rate is optional.





Fig 2.12 Setting up the frame size and line rate of the traffic

2.13 Setup Flow Tracking and Protocol Behavior:

*Setting up the flow tracking and Protocol Behavior is optional.



New Advanced Traffic Wizard *		\times
Endpoints Protocol Behaviors	IxNetv	vork
Packet / QoS Packet / QoS Packet / QoS Provide a conception of the corresponding traffic packet fields on the fly with the information learned from protocols provide a conception of the corresponding traffic packet fields on the fly with the information learned from protocols provide a conception of the corresponding traffic packet fields on the fly with the information learned from protocols		
Image: Click to Update traffic on the fly with the information learned from Protocols 100 100 Group 200 DUT		
Preview Label Provider Preference	_	
Validate Transport LSP: RSVP, Basic LDP Inter AS/Region LSP: BGP (RFC 3107), Targeted LDP LISP RLOC Ordinal Value Preference		
Ordinal Value 0 For out of bound value last available RLOC will be used		
Open Flow Preference Enable Open Flow Nexthop selection for EVPN unicast traffic destined to Multi-homed CE Ordinal Value O In classic, first nexthop is used for out of bound ordinal. NGPF will give it multicast treatment		
Multicast Multicast forwarding mode:		
Prev Next Einish Cancel	H	ielp

Fig 2.13 Setting up the flow tracks and traffic update option

2.14 Validate Traffic:

Me Advanced Traffic Wizard *		– 🗆 X						
Endpoints	Preview	lxNetwork						
Packet / QoS	Flow Groups/Packets	Current Traffic Item Call Traffic Items View Flow Groups/Packets						
Flow Group Setup	Flow Group	Traffic Item						
Frame Setup		Groups/Packets						
Rate Setup	No Flo	w Groups to display.						
Flow Tracking	Please click 'View How Groups/P	ackets' button above to show the How Groups.						
Protocol Behaviors		The Preview dialog is used to preview the						
Preview	Flow Group packets	configured traffic items before validating						
Validate	N	the errors and exiting from the wizard.						
	43							
\geq	No.	asket to display						
	NO packet to display. Please click 'View Flow Groups/Packets' button above to show the packets.							
		2 Click Next						
	1 99 99 99 9							
		Prev Next Finish Cancel Help						



Fig 2.14 Viewing the flow groups and validating the traffic

2.15 Apply Traffic, Start Traffic, and Statistics View:

🔟 🗋 🗁 🔚 • 🎕 • 🕅	• 涛 🖉	• 🕘 💭 • 🛄 🚺	\$ } ₹	Traffic Tools	IxNe	twork (TCL:8009 SDN	9009 REST:11009 REST Session	n:1) [default_pramodm.7144	4.ixncfg]	
File Home A	utomation	Results / Reports	Views	Configuration						
L2-3 I4-7 Traffic Traffic Traffic	Add L2-3 Traffic	Click Apply L2-2	Traffic	Apply Changes	© ⊙ Options	Find Grid Operation	Grid/Column Profiles ▼ ∰ Group Rows By ▼	0% Ethernet	- VM - 001 1	00.0000%
Apply L2-3 Traffic		Click Start L2-3	Traffic				Grid	Rate Control (1 ou	t of 1 Flow Groups selecte	d for this port)
Start L2-3 Traffic	< \$	🔐 🕮 Traffic 🕨	C L2-3 Flow	Groups						
2 Sect 2.3 Tolk		Transmit State	Suspend	Tx Port		Rx Ports	Flow Group Name	Configured Frame Size	Applied Frame Size	Frame Rate
📲 Scenario) 🔨 🕘 🔪 🖩 🖬 Tr	affic Item N	lame: Traffic Item 1	TX Mode	: Interleaved, Src/Ds	t Mesh: OneToOne, Route Me	sh: OneToOne, Uni-directi	onal	
	1			Ethernet - VM - 001	Ethernet	- VM - 002;	Traffic Item 1-EndpointSet	Fixed: 64		10% Line Rate
▼ 🕘 Ports										

Fig 2.15 Applying and Starting traffic

X: Irathc X: I2-3 Traffic Items X: I2-3 Flow Groups Impairments QuickTests X: Captures	v Groups Check Traffic Statistics								
10. 200 - C	Summary Settings Tradking and Latency All 🐴)							
	Select Views. Port CPI Statistics 12-13 Test Summary Statistics Port Statistics Global Perform Statistics Traffic Item Statistics	• x (
	Triffic Base Tv Ermer Dr Ermer Dab and C Tv Ermer Dab Dv Ermer Dab Dv Ermer Dab (Bac) Dv 1 0 da (Bac) Dv Dab	Dy Date (Vher) Ty Date (Mher) Dy Date (Mher)							
	1 annu Letin 1 Arrianies (alies) names vena (vos %) 1 A name (alies) names (a rianie name (a rianie name (a rianie (alies) (a rianie) (a	76 744 272 76 744 772 76 744 76 77							
	1 I IIIIIII. Lalii 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	10,741.272 70.74 70.74							
	E Tella 1/1 total rows: 1) K K A Coss I troughout Latency 🥥								
1									

Fig 2.15.1 Check for traffic statistics

3. Configure BGP through Automation (HLPyAPI):

IxNetwork provides a wide array of automation APIs. The REST API is the recommended method for creating new test scripts for IxNetwork. Please refer to the IxNetwork REST API quick reference guide for more details. The legacy API library continue to be supported for existing test scripts. This section will walk through to reproduce the same BGP emulation scenario through High Level Python API's to get the user introduced with most of the HLPyAPI's used in NGPF framework.

3.1 Initialize Environment:

Import the Required Packages and Check for the sanity of the System. *import sys, os import time, re sys.path.append('C:/Program Files (x86)/Ixia/hItapi/8.40.1123.18/TcIScripts/lib/hItapi/library/common/ixiangpf/python/') from ixiatcl import IxiaTcl from ixiahlt import IxiaHIt from ixiangpf import IxiaNgpf from ixiaerror import IxiaError ixiangpf = IxiaNgpf(ixiahlt)*

3.2 Add Chassis and Lock Ports:

ixiangpf.connect: Connects to the Ixia chassis with selected ports using the specified port handles

chassis_ip	= "192.168.52.198"					
tcl_server	= "10.38.66.146"					
ixnetwork_tcl_server	= "10.38.66.146:8009"					
port_list	= "1/11 1/12"					
connect_result = ixiangpf.connect (
ixnetwork_tcl_server	= ixnetwork_tcl_server,					
tcl_server	= tcl_server,					
device	= chassis_ip,					
port_list	= port_list,					
break_locks	= 1,					
reset	= 1					
)						

```
ports = connect_result['vport_list'].split()
```

*Note High Level API's are highlighted in Orange and all other handles are highlighted in Green.

Overview	<	>	٢	Ports	•
		S.		Name	Connection Status
📭 🗧 Scenario	1	0		1/1/11	
	2			1/1/12	10.39.64.132:01:12
👻 🥯 Ports					Dente
Chassis				(Locked

Fig 3.2: Chassis connected and selected ports locked



3.3 Create Topology and DeviceGroup:

Ixiangpf.topology_config: Adds Topology to the specified port handle and returns the topology handle and Device Group handle which can be used to configure Device Groups.

```
topologyConfig1 = ixiangpf.topology config (
                                          = """BGP 1 Topology""",
  topology name
  port_handle
                                          = ports[0],
)
topology 1 handle = topologyConfig1['topology handle']
deviceGroup1 = ixiangpf.topology_config (
  topology handle
                                          = topology 1 handle,
                                          = """BGP 1 Device Group""",
  device group name
  device group multiplier
                                          = "1",
                                          = "1".
  device group enabled
)
deviceGroup 1 handle = deviceGroup1['device group handle']
topologyConfig2 = ixiangpf.topology_config (
                                          = """BGP 2 Topology""",
  topology name
  port_handle
                                          = ports[1],
)
topology 2 handle = topologyConfig2['topology handle']
deviceGroup2 = ixiangpf.topology_config (
  topology handle
                                          = topology 2 handle,
                                          = """BGP 2 Device Group""",
  device group name
                                          = "1",
  device group multiplier
  device group enabled
                                          = "1",
)
```

```
deviceGroup_2_handle = deviceGroup2['device_group_handle']
```



Fig 3.3: Device groups added to respective topologies



3.4 Create Ethernet Stack:

Ixiangpf.interface_config: Configures the protocol stack with the Specified Options by using the Device Group Handle and returns the particular protocol stack handle

interfaceConfig1 = ixiangpf.interface_config (
 protocol_name = """Ethernet 1""",
 protocol_handle = deviceGroup_1_handle,
 mtu = "1500",
 src_mac_addr = "18.03.73.c7.6c.b1",
 src_mac_addr_step = "00.00.00.00.00",
)

ethernet_1_handle = interfaceConfig1['ethernet_handle']

interfaceConfig2 = ixiangpf.interface_config ((
protocol_name	= """Ethernet 2""",
protocol_handle	<pre>= deviceGroup_2_handle,</pre>
mtu	= "1500",
src_mac_addr	= "18.03.73.c7.6c.01",
<pre>src_mac_addr_step</pre>	= "00.00.00.00.00.00",
)	

```
ethernet_2_handle = interfaceConfig2['ethernet_handle']
```



Fig 3.4: Ethernet stacks added to device groups

3.5 Create Ipv4 Stack:

ipv4config1 = ixiangpf.interface_config (

protocol_name	= """IPv4 1""",
protocol_handle	= ethernet_1_handle,
ipv4_resolve_gateway	= "1",
ipv4_manual_gateway_mac	= "00.00.00.00.00.01",
gateway	= "20.20.20.1",
gateway_step	= "0.0.0.0",
intf_ip_addr	= "20.20.20.2",
intf_ip_addr_step	= "0.0.0.0",
netmask	= "255.255.255.0" <i>,</i>

```
)
```

ipv4_1_handle = ipv4config1['ipv4_handle']

```
ipv4config2 = ixiangpf.interface_config (
```

protocol_name	= """IPv4 2""",
protocol_handle	= ethernet_2_handle,
ipv4_resolve_gateway	= "1",
ipv4_manual_gateway_mac	= "00.00.00.00.00.01",
gateway	= "20.20.20.2",
gateway_step	= "0.0.0.0",
intf_ip_addr	= "20.20.20.1",
intf_ip_addr_step	= "0.0.0.0",
netmask	= "255.255.255.0",

ipv4_2_handle = ipv4config2['ipv4_handle']



Fig 3.5: IPv4 stacks added to ethernet stacks

3.6 Create BGP:

Ixiangpf.emulation_bgp_config: Configures BGP stack with the specified options by using IPv4 handle and returns the BGP Stack handle

bgpIpv4Peer_1_handle = bgpConfig1['bgp_handle']

bgpConfig2 = ixiangpf.emulati	on_bgp_config (
mode	= "enable",
active	= "1",
handle	= ipv4_2_handle,
remote_ip_addr	= "20.20.20.2",
)	

bgpIpv4Peer_2_handle = bgpConfig2['bgp_handle']



Fig 3.6: BGP stacks added to ipv4 stacks



3.7 Create Network Group:

Ixiangpf.multivalue_config: Configures multivalue with specified options by using Device Group handle and topology handle and returns the multivalue handle

multiValueConfig1 = ixiangpf.multivalue_config (

pattern	= "counter",				
counter_start	= "200.1.0.0",				
counter_step	= "0.1.0.0",				
nest_step	= "0.0.0.1,0.1.0.0",				
nest_owner	= '%s,%s' % (deviceGroup_1_handle				
	topology_1_handle),				
nest_enabled	= "0,1",				

```
)
```

multivalue_4_handle = multiValueConfig1['multivalue_handle']

roup_config (
= deviceGroup_1_handle,
= "BGP_1_Network_Group1",
= "1",
= "1",
<pre>= ethernet_1_handle,</pre>
= "ipv4-prefix",
= multivalue_4_handle,
= "24",

)

networkGroup_1_handle = networkGroupConfig1['network_group_handle']



Fig 3.7 Adding BGP network group to device group 1

Ixiangpf.network_group_config: Configures Network Group with specified options by using Device Group handle, topology handle, ethernet handle and returns the Network Group handle

multiValueConfig2 = ixiangpf.multivalue_config (

)

pattern	= "counter",
counter_start	= "201.1.0.0",
counter_step	= "0.1.0.0",
counter_direction	= "increment",
nest_step	= "0.0.0.1,0.1.0.0",
nest_owner	= '%s,%s' % (deviceGroup_2_handle,
	topology_2_handle),
nest_enabled	= "0,1",

multivalue_10_handle = multiValueConfig2['multivalue_handle']

networkGroupConfig2 = ixiangpf.network_group_config (protocol handle = deviceGroup 2 handle, protocol name = "BGP 2 Network Group1", = "1", multiplier enable device = "1". connected to handle = ethernet 2 handle, = "ipv4-prefix", type = multivalue 10 handle, ipv4_prefix_network_address = "24", ipv4_prefix_length)

networkGroup_3_handle = networkGroupConfig2['network_group_handle']



Fig 3.7.1 Adding BGP network group to device group 2



3.8 Start Protocols:

ixiangpf.test_control: Start/Stop all the protocols configured in the test session

testControl = ixiangpf.test_control (action='start_all_protocols')
print("Waiting for 45 seconds for the protocols to converge")
timer = 30
time.sleep(timer)

-		Select Views		Protocols	Summary Port 0	CPU Statistics 🗡	Port
	Port	Sessions Up	See	sions Down	Sessions Not Started	Sessions Total	
Þ	1/1/11	. 2		0	0	2	
2	1/1/12	2		0	0	2	
				BGF	• Sessions Up		

Fig 3.8 Starting all protocol stacks to come up

3.9 Enable Filter and Apply Changes on the Fly:

bgp_1_status = ixiangpf.emulation_bgp_config (





Fig 3.9 Enabling the route filter and applying the changes on the fly

3.10 Retrieve Learned Info:

```
bgpLearnedInfo = ixiangpf.emulation_bgp_info (
    handle = bgpIpv4Peer_1_handle,
    mode = 'learned_info');
pprint(bgpLearnedInfo)
```

bgpLearnedInfo = ixiangpf.emulation_bgp_info (
handle	= bgpIpv4Peer_2_handle,						
mode	= 'learned_info');						
pprint(bgpLearnedInfo)							

Details for BG	P Peer 2														
Protocol Setting	rotocol Settings > < BGP Peer Global Settings > < BGP Peer Learned Information > < IPv4-Prefixes														
Grouping	Index#	Topology	Device Gro	ų	IPv4 Prefix	Prefix Length	Path ID	IPv4 Next Hop	IFV	6 Next Hop	MED	Local Preference	Origin	AS Path	Comm
Session 20.	1	BGP_2 Topolo	BGP_2 Devi	e											
· 1/1/12	# 1				200.1.0.0	24	NA	20.20.20.2		\sim	^	0	TOD		
												Learned IPv4 Ro	utes		

Fig 3.10 Showing details of BGP learned routes

3.11 Configure Traffic:

ixiangpf.traffic_config: Configures the traffic streams on the specified ports with specified options

```
trafficConfig = ixiangpf.traffic config (
  mode
                                          ='create',
  traffic generator
                                          ='ixnetwork 540',
  endpointset_count
                                          =1,
                                          =networkGroup 1 handle,
  emulation_src_handle
  emulation dst handle
                                          =networkGroup 3 handle,
  track_by
                                          ='sourceDestEndpointPairO trackingenabledO',
  rate_pps
                                          =1000,
 frame size
                                          =512,
)
```



Fig 3.11 L2-3 Traffic configured with the specified options



3.12 Start Traffic and Get Statistics:

ixiangpf.traffic_control: Start/stop traffic and allows to modify global traffic options





Fig 3.12 Running traffic

ixiangpf.traffic_stats: Collect Traffic statistics with the specified options

protostats = ixiangpf.traffic_stats (
 mode = 'all',
 traffic_generator = 'ixnetwork_540',
 measure_mode = 'mixed'
)



4. Other Utilities:

4.1 IxNetwork API Documentation Browser:

The main feature of this application is the ability to browse the API meta data in a hierarchical format. Access each level of the hierarchy with a view of siblings, attributes, execs, errors, and children by on clicking on BROWSE.



Fig 4.1 IxNetwork API documentation link



4.2 Script Gen:

- ScriptGen is a tool that may be used to generate a script that reflects the current configuration of IxNetwork.
- It is intended to be used after IxNetwork has been successfully configured. The generated scripts can be used to re-create a configuration as the basis for a new test.



Fig 4.2 IxNetwork ScriptGen link



4.3 F1 Option:

Move the mouse pointer over any field in the GUI, and then press F1 to get more information about the field.



Fig 4.3 IxNetwork F1 option usability



5. To Know More on NGPF:

https://www.youtube.com/watch?v=A0mbZuP94jo http://openixia.com/sampleScripts//lxNetwork/HighLevelApi/Ngpf/Python

6. Support:

For more information: <u>https://support.ixiacom.com/</u> For support assistance, contact : <u>support.ix@keysight.com</u>





https://github.com/openixia For queries : support.ix@keysight.com