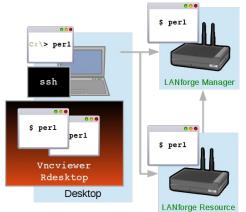


LANforge Scripting Cookbook

The LANforge Scripting Cookbook provides a series of detailed examples of how to craft testing scripts for unattended/automated operation. Each example intends to give the reader a runnable test script and a better understanding of how to use the LANforge Scripting.

Places to Run CLI Commands

You do not need to operate scripts directly from the LANforge server, and this allows you to code scripts in your preferred text editing environment. Likewise, you do not need to run a copy of LANforge Server on your desktop. Scripts will create a plain-text connection to the LANforger server you specify.



Windows Desktop

You can install a copy of the LANforge Server on your windows desktop (without a license) so that you have access to the Perl scripting libraries. Edit scripts and run them from your C:\Program Files\LANforge-Server\scripts directory.

Linux Desktop

You can copy the LANforge scripts folder directly from your LANforge server to your Documents directory with scp.

SSH or VNC connection to LANforge Server

Using vncviewer, rdesktop or ssh are all fine options to connect to the LANforge server to write and operate scripts. The LANforge server comes with a basic Linux desktop and you can use emacs, vim, pluma, or gedit text editors installed by default. When editing scripts on the LANforge server itself, be careful to back up your work before you upgrade LANforge. The LANforge install process will over-write scripts of the same name in the scripts directory.

Requirements for Scripts

Your desktop (or other computer) running CLI scripts needs to have a reliable (wired) connection to the management port of your LANforge server. If you are engaging in long running tests, you might consider running the scripts from the LANforge manager itself if your desktop machine needs to be powered off.

Script Libraries

CLI scripts are written using Perl. They require the libraries in /home/lanforge/scripts/LANforge Users may write scripts in other programming languages, such as python, but in that case, they will not be able to take direct avantage of the Perl scripts included in LANforge.

On Windows

LANforge is more fully featured on Linux, but basic support exists on Windows as well.

You can run CLI scripts from any Windows desktop as long as you have Perl installed. You can use ActiveState Perl or Perl from the Cygwin project. We also highly suggest installing PuTTY ssh client to access your LANforge server.

On Linux/OS X

Most Linux distributions come with and ssh client and Perl already installed.

LANforge Server Requirements

The following examples will create test scenarios that work on LANforge Linux systems running the LANforge software with the LANforge kernel and a sufficient license. If you are running LANforge server using another Linux kernel, you may not be able to operate some of the examples. (Features like Armageddon, operation of WiFi-AC radios, and WanLinks all require drivers included only in Candela provided kernels.)

Please contact us at support@candelatech.com if you have any questions

Before Starting LANforge-CLI Traffic Generation

Before attempting the examples below, ensure that you have successfully followed these software installation guides:

- LANforge-GUI Installation
- LANforge Server Installation

It is also recommended that you back up your current running LANforge Server database so that you may safely return to your current operating state. For instance:

su - root cd /home/lanforge tar -cvzf my_db_backup.tar.gz DB

LANforge-CLI Detailed Cookbook Examples

- 1. Perl: Perl CLI Scripts Introduction
- 2. Perl: Operating LANforge scripts from Windows
- 3. Perl: Monitor and Reset Ports with the portmod Script
- 4. Perl: Cross Connects and Endpoints Tutorial
- 5. Perl: Creating Connections with the FIREmod Script
- 6. Perl: Operating Endpoint Hunt Scripts with the CLI
- 7. Perl: Generating WiFi Traffic with the Associate Script
- 8. Perl: Changing Station WiFi SSID with the CLI API
- 9. CLI: Changing Station POST_IFUP Script with the CLI API
- 10. CLI: Scripting Attenuation with CSV data
- 11. Basics: LANforge Entity IDs
- 12. Basics: LANforge GUI Introduction
- 13. JSON: Querying the LANforge Client for JSON Data
- 14. CLI: Station CLI Operations
- 15. Perl: Generic Endpoint Scripts
- 16. Perl: Chamber View: Automated tests with script
- 17. Basics: LANforge Scripting Introduction
- 18. Python: Querying the LANforge JSON API using Python
- 19. Python: Managing WANlinks using JSON and Python
- 20. Perl: Control a chamber with the If_chamber.pl Script
- 21. Perl: Emulating Video Transmission with Layer 3 connections
- 22. Python: Create Test Scripts With the Realm Class
- 23. Python: Create Layer 4 Test Scripts With Python
- 24. Python: Create Generic Test Scripts With Python
- 25. Python: Create VAP Test Scripts With Python
- 26. Python: Load Scenarios And Control Test Groups With Python
- 27. Python: Record the results of a test as CSV from the REALM monitor script
- 28. Python: Record the results of a test as an Excel file from the REALM monitor script
- 29. Python: Define and Demonstrate Docstring Usage in Candelatech Python Scripts
- 30. Python: Scan for SSIDs, BSSIDs, and Signals of wireless APs
- 31. Python: Probe Ports for Information
- 32. Suite: Start Here: Initial Setup to Run Scripts Test Suite for AP Testing
- 33. Getting Started: A Simple Script: sta_connect2.py
- 34. Getting Started: Basic: Layer 3 Traffic Generation: test_13.py
- 35. Getting Started: Basic: Layer 4-7 HTTP Traffic Generation: test_14.py
- 36. Getting Started: Basic: Layer 4-7 FTP Traffic Generation: test_14.py
- 37. Getting Started: Configure and Run Dataplane Test
- 38. Multiplexed REST Access via Nginx Proxy

Introduction to CLI Scripts

Goal: You will be able to execute LANforge testing scripts from the command line from Windows, Linux, remote desktop or ssh connection.

Traffic emulation may be run unattended and automated using Perl scripts provided with the LANforge Server. These scripts can be run from within the LANforge server or outside the LANforge Server (on a Windows or other desktop). The output of the scripts should be redirected into a text file for you to process the results.



Where Do I Find Scripts?

On Windows

On most versions of windows, the LANforge Server installs scripts in

C:\Program Files (x86)\LANforge-Server\scripts

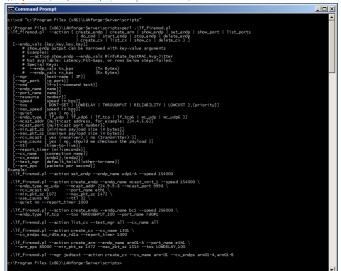
On Linux

In the home directory for user LANforge:

How to Run Scripts

Starting a script on Windows:

- 1. Make sure that perl is in your PATH. (See Inspecting DOS PATH)
 - 2. Open a CMD window (or a PowerShell window)
 - 3. and change directory to C:\Program Files (x86)\LANforge-Server\scripts
 - 4. Type perl .\script_name.pl ENTER to run the script.



Generally, the script will tell you that it needs more switches.

Finding Help

Most scripts have a -h or --help switch that explain what switches they expect.

Script Conventions

In general, scripts will expect you to tell them a few things, regardless of the script:

- The manager IP address to connect to:--mgr 127.0.0.1 or --manager 192.168.100.1. This
 often defaults to 127.0.0.1, but when connecting from outside the machine, please use the IP of
 the LANforge management port (often eth0).
- The manager port to connect to --port 4001 or --mgr_port 4001. This often defaults to 4001.
- Which resource to direct the command to. The manager is always resource 1. Resource 2
 would be your second LANforge server. -- resource 2. Some scripts use the older term card: -card
- If you need debugging output, turn off quiet mode: -q no or --quiet no, Some older scripts want you to turn quiet on explicitly: --quiet 1
- To capture the output, use the > operator to redirect the text output into a text file.
- Scripts are often executed from within a shell script (or batch file). Often the formatting of the
 commands includes '\' characters which indicate 'continue this command on the next line of
 input.' Here is an example of formatting a single script command on multipl lines:

\$./lf_portmod.pl \
manager 192.168.100.1 \
card 3 \
show port

 Comments begin with '#'. They are lines ignored by the shell, and they are also comments in perl.

Running on local LANforge manager

You can use ssh, VNC or Rdesktop to connect from your desktop to your LANforge manager server. (When using VNC, assume display :1). From there, in a terminal, you will execute your script from the /home/lanforge/scripts directory as shown in the example below:

- \$ cd /home/lanforge/scripts
- \$./lf_portmod.pl --help
 # help_appears
- #...help appears...
 \$./lf_portmod.pl --manager 192.168.100.1 --card 1 --show_port
- # ... displays port info

0	lanforge@jedtest:~/scripts	000
File Edit View	earch Terminal Help	
	-manager 127.0.0.1card 1quiet 1show_portport_name eth1 nds can be found in lf_portmod.txt	
>>RSLT: 0 Cmd: '	_show_port' '1' '1' 'eth1'	
Win32-Name: Wi Current: UP Supported: UP Advertis-UP Advertis-UP INS Servers: IPv6-6lobal: DE IPv6-6lobal: DE NAC: 00:98:0b12 But309:0b12 Bildge-Port-Cos Bildge-Port-Cos PAD: 78720 TRXDF07: 0 TK RXDF07: 0 TK	290:Dff;fc29:0f9/64 EFED 06:f9 DEV: ethl MTU: 1500 TX Queue Len: 1000 1907: e1000e Tx-Rate: 1000000kbps	

Running on local LANforge resource

If you connect to a LANforge resource and want to run a script, you must direct the script at the LANforge manager server and specify the resource you are interested in. For example, you might be on resource 2 (192.168.100.2) and desire to run tests on resource 3 (192.168.100.3):

<pre>\$ cd /home/lanforge/scripts</pre>			
<pre>\$./lf_portmod.plmanager</pre>	192.168.100.1	card 3	show_port
# displays port info			

Running on a Linux Desktop to a Remote LANforge

A more detailed set of steps follows. When running LANforge CLI scripts on a Linux desktop, you normally want to download and un-zip a copy of the LANforge-Server install file found on the Candela Technologies downloads page. Use a link similar to:

http://www.candelatech.com/private/downloads/r5.3.2/LANforgeServer-5.3.2_Linux-F21x64.tar.gz. For best results, use the scripts packaged with the version of LANforge to which your scripts will be connecting.

- 1. Open a terminal on your desktop, cd to your Downloads folder
- 2. use wget or curl to download the tar file:

wget "http://guest:guest@www.candelatech.com/private/dowmloads/r5.3.2/LANforgeServer-5.3.2_Linux-F21-x64.tar.gz"

3. Create a scripts directory in your Documents folder:

\$ cd ~/Documents/scripts

- 4. Expand the tar file in your Downloads directory:
- \$ tar xf LANforgeServer*tar.gz

5. Copy the scripts file into your Documents folder:

```
$ cp -r LANforgeServer-5.3.2/scripts/. ~/Documents/scripts/
```

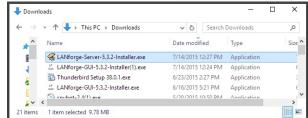
To use your scripts, in your terminal, change directories to ~/Documents/scripts and they will operate similar to the above examples.

\$ cd ~/Documents/scripts
\$./lf_portmod.pl --manager 192.168.100.1 --card 3 --show port

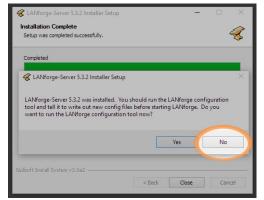
Running on a Windows Desktop to a Remote LANforge

The process for running CLI scripts on a Windows desktop is roughly similar, but involves running the Windows LANforge Server installer. This process does not require a Windows license as we will not be running the windows LANforge server. Perl is required to run Windows scripts. Start by installing that. You can use the perl that comes with the Cygwin project or if you just want perl, install the ActiveState ActivePerl package. ActivePerl should install update your environment **%PATH%** variable. If it does not immediately, you might need to log out and log back in.

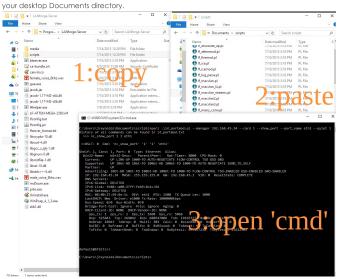
- Download the Windows version of the LANforge Server installer using your browser: http://www.candelatech.com/private/downloads/r5.3.2/LANforge-Server-5.3.2-Installer.exe. Use username guest, password guest.
- 2. In your Downloads folder, double click and install the LANforge-Server-Installer.exe.



Do not configure it, do not run LANforgeServer.



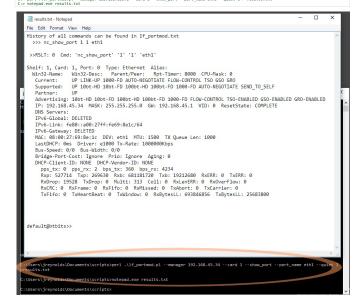
- You will not need to be running the LANforge GUI to do this install.
- 3. The installation scripts folder will be system-protected, so you want to copy the folder over to



4. Open a DOS terminal, either using Run→cmd Enter or Run→powershell Enter

- 5. Change to the new copy of the scripts directory, and then you can run scripts by giving them to perl:
 - C:> cd C:\Users\bob\Documents\scripts C:> perL.\\f_portmod.pl --manager 192.168.100.1 --card 3 --show_port --port_name ethl --quiet 1
- To capture output from a script, use the shell redirect operator. ≥. This example shows redirecting and browsing the results with Notepad:

C:> perl .\lf_portmod.pl --manager 192.168.100.1 --card 3 --show_port --port_name eth1 --quiet 1 > results.txt C:> notenad.exe results.txt



Operate LANforge Scripts from Windows.

Goal: Use an installation of LANforge and Perl on a Windows computer to operate tests and manage connections on remote LANforge computers.



1. Prepare your Windows computer:

2. Install LANforge Server

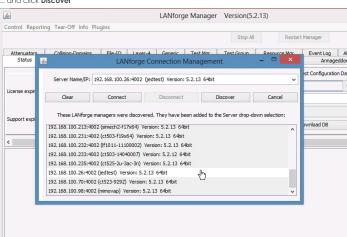
- A. You do not need to configure or start this server. Only the perl scripts directory of this installation will be used.
- B. You can download it from our current releases page. For more information see LANforge Server Installation

3. Install LANforge GUI

- A. You can download it from the same location.
- B. Make sure you can connect to your LANforge manager. In this example, the LANforge manager will be at 192.168.100.26
- C. Open the LANforge GUI

ntrol Reportin	ng Tear-Off Info Plu	igins										
						Stop All		Restart Mani	ager		Refresh	HEU
Attenuators Status	Collision-Domains		Laver-4 Ge			Test Group	Resource	X Ev	vent Log Armage	Alerts	Port Mgr W	Message anLinks
	Server Name/IP:							× st C	onfiguratio	n Database	2	
ense expir	our ren ridineja r									-	Load	
	Clear	Connect	Disc	connect	Die	scover	Cancel				Delete	
		Cir	k Discover to f	ind LANforce	Managers	45°				~	Save	
pport expl								pwnł	oad DB		Show Progr	285
		_	_			_						

D. ... and click Discover



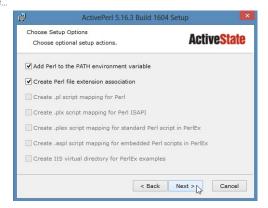
E. If you are able to connect you should be able to browse your ports and connections.

								Stop	A	Restart	Manager	Refi	resh	HELF
Stat	JS	1	Layer-3	L3 End	26	VoIP/	RTP	VoIP/RT	P Endps		Armageddon		WanLinks	
Attenua	tors	Col	ision-Domains	File-IO	Layer-4	Generic	Test Mgr	Test Group	Resource	e Mgr	Event Log Al	lerts Port N	lgr Mess	ages
			Disp: 192.168.10	0.135:0.0		Sniff Packets	d	ear Counters	Reset Po	rt D	slete			
			Rpt Timer: medi	um (8 s)	~	Apply		view Details	Create	M	dify Batch I	Modify		
						All Ether	net Interfaces (P	orts) for all Reso	urces.					
Port	Pha	Down	IP	SEC	Alias	Parent Dev	RX Bytes	RX Pkts	Pps RX	bps RX	TX Bytes	TX Pkts	Pps TX	1
.2.00			192.168.100.42	0	eth0	_	7.662.812.693	55,122,599	46	55.232	183.573.933	138.087.7	128	1
1.1.0		H H	192.168.100.26	0	eth0		184,280,308	143.090.9	166	1.473.093	10.246.150.576	53.885.054	133	
.2.17	in	in	10.26.1.19	0	sta9	wiphy0	76.280.536	57.360	0	0	986.461.208	650,662	0	
.2.16		H H	10.26.1.18	0	sta8	wiphy0	75,164,420	56.514	0	0	988,778,230	652.241	0	
.2.15			10.26.1.17	0	sta7	wiphy0	76,701,566	57,657	0	0	995,897,302	656,855	0	
.2.14			10.26.1.16	0	sta6	wiphy0	72,056,940	54,422	0	0	1,002,001,044	660,882	0	
.2.13			10.26.1.15	0	sta5	wiphy0	75,608,172	56,870	0	0	1,001,813,806	660,757	0	
.2.12			10.26.1.14	0	sta4	wiphy0	75,117,326	56,525	0	0	1,007,291,372	664,306	0	
.2.11			10.26.1.13	0	sta3	wiphy0	78,660,442	58,967	0	0	1,041,047,048	686,280	0	
1.2.10			10.26.1.12	0	sta2	wiphy0	75,568,352		0	0			0	
.2.09			10.26.1.11	0	sta 1	wiphy0	83,100,650	61,663	0	0	259,044,274,	168,875,6	0	
.2.02			10.26.1.10	0	sta0	wiphy0	35,447,362		0	0	191,966,135,	125,147,6	0	
							456,400,222,	301,529,1	N 0	0	735,641,436	514,331	0	
.2.01			10.26.0.3	0	eth1		458,047,909,	304,987,1	A: 1	1,985			1	
.1.1			10.26.0.2	0	eth1		1,260,067,886		1	1,997		304,948,3	1	
.2.27			10.26.0.19	0	eth1#9	eth1	980,969,979		0	156		58,097	0	
1.2.26			10.26.0.18	0	eth1#8	eth1	983,843,653		0	156		57,952	0	
.2.25			10.26.0.17	0	eth1#7	eth1	989,626,547	687,089	0	90		58,132	0	
1.2.24			10.26.0.16	0	eth1#6	eth1	996,399,789		0	90		57,987	0	
.2.23			10.26.0.15	0	eth1#5	eth1	996,768,295		0	155		58,078	0	
1.2.22			10.26.0.14	0	eth1#4	eth1	1,001,251,507		0	89		57,994	0	
1.2.21			10.26.0.13	0	eth1#3	eth1	1,031,768,323	714,516	0	89	83,405,440	58,106	0	

- For more information see LANforge GUI Installation
- 4. Install ActiveState Perl
 - A. Please download it from the ActiveState downloads page.
 - B. Begin and press next...

	ActiveP	erl 5.16.3 Build 1604 Setup	×
	ActiveState	Welcome to the ActivePerl 5.16.3 Build 1604 Setup Wizard	
	ActivePerl [®]	The Setup Wizard will allow you to change the way ActivePerl 5.16.3 Build 1504 features are related on your computer or even to remove ActivePerl 5.16.3 Build 1604 from your computer. Click Next to continue or Cancel to exit the Setup Wizard.	
	www.activestate.com		
		Next > Cancel	
Cpress next			
	户 ActiveP	erl 5.16.3 Build 1604 Setup	×
	Custom Setup Select the way you want fe	ActiveState	
	Click on the icons in the tree b	below to change the way features will be installed.	
	ActivePerl Perl PPM Document Examples	ActiveState ActivePerl is a quality-assured distribution of Perl and other value-additions such as PPM, Perl for ISAPI, PerlEx, and PerlScript.	
		This feature requires OKB on your hard drive. It has 4 of 4 subfeatures selected. The subfeatures require 80MB on your hard drive.	
	Location: C:\Perl\	Browse]
	Reset Disk Usage	< Back Next > Cancel]

D. ...press next...



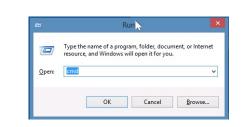
Eand then wait a f	ew minutes			
d₽	ActiveP	erl 5.16.3 Build 1604 Set	qu	×
	nstalling ActivePerl 5.16.3	Build 1604	Active Stat	e
	Please wait while the Se 1604. This may take se	tup Wizard installs ActivePe veral minutes.	rl 5.16.3 Build	
	Status:			
		< Back Ne	ext > 🔓 Cancel	
F. Now, you are done	, press Finish .			
B	ActiveP	erl 5.16.3 Build 1604 Set	цр	×
	Active <mark>State</mark>	Completing the ActivePe Setup Wizard	rl 5.16.3 Build 1604	
	Active Perl [®]	Using ActivePe Secure your critical syste production server licensin; Business or Enterprise Er today to learn about these ActivePerl binaries and su organizations of all sizes i Contact us today for more sales@activestate.com or 1.866 631 4631 (toll-free i 1.866 foil 4.861 (toll-free)	ms and get g with ActivePerl litions. Contact us quality-assured pport packages for that depend on Perl. information: phone	^
	www.activestate.com	Display the release no		
_		< Back	Cancel	

G. For advanced GUI scripting in Windows, you may also wish to view the Win32::GuiTest perl module page.

5. Using scripts from Windows

Α.

- A. The scripts installed on your Windows computer will communicate with the LANforge manager over the management port (TCP 4001).
- B. Open a cmd window. Click Start->Run, type cmd and press Enter





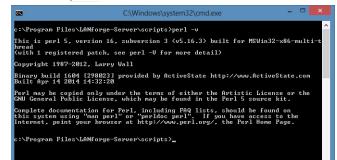
C. Change directory to C:\Program Files\LANforge-Server\scripts

C:\Windows\system32\cmd.exe	-	×	
Microsoft Windows [Version 6.2.9200] <c> 2012 Microsoft Corporation. All rights reserved.</c>			^
C:\Users\IEUser≻cd "c:\Program Files"			
c:\Program Files>cd LANforge-Server			1
c:\Program Files\LANforge-Server≻cd scripts			
c:\Program Files\LANforge-Server\scripts>			

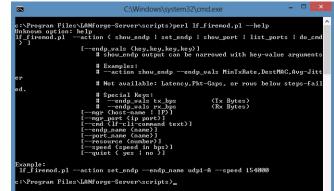
D. List the script files: dir *.pl

volume ser	ial Number is	D0DE-1E49	
Directory	of c:\Program	Files\LANforge-Server\scripts	
8/07/2014	04:32 PM	3,205 ftp-upload.pl 32,537 lf_associate_ap.pl 3,163 lf_attenmod.pl 7 370 lf firmerod pl	
18/07/2014	04:32 PM	32,537 lf_associate_ap.pl	
18/07/2014	04:32 PM	3,163 lf_attenmod.pl	
8/07/2014	04:32 PM	7,370 lf_firemod.pl	
18/07/2014	04:32 PM	10,162 lf_ice.pl	
8/07/2014	04:32 PM	4,322 lf_icemod.pl	
8/07/2014	04:32 PM	419 If_log_parse.pl	
18/07/2014	04:32 PM	7.370 H_3rendd.pl 10.162 H_1ce.pl 4.322 H_ice.pl 4.131 H_1cg_parte.pl 44.633 H_nacvlan2,pl 15.644 H_nacvlan2,pl 15.644 H_nacvlan3,pl 12.240 H_nacvlan3,l 19.122 H_nacvlan3,l 19.121 H_nacv	
18/07/2014	04:32 PM	15,644 lf_macvlan2.pl	
07/2014	04:32 PM	17,240 If_macvlan3.pl 19,122 lf_macvlan_14.pl	
0/07/2014	04:32 PM	17,419 lf_macvlan_streams.pl	
19/07/2014	04-32 PM	8557 lf moniton nl	
8/07/2014	04-32 PM	13,830 lf_many_conn.pl 8,557 lf_monitor.pl 18,031 lf_netoptics.pl	
18/07/2014	04:32 PM	32 819 lf nfe io nl	
8/07/2014	04:32 PM	11 610 lf nowtmod nl	
8/07/2014	04:32 PM	8,191 lf_port_walk.pl	
8/07/2014	04:32 PM	11.610 lf_portmod.pl 8.191 lf_port_walk.pl 6.751 lf_stress1.pl	
8/07/2014	04:32 PM	9,939 lf_stress3.pl	
8/07/2014	04:32 PM	9,939 If_stress3.pl 6,145 If_stress4.pl 24,688 If_verify.pl 22,242 If_voip.pl	
8/07/2014	04:32 PM	24,688 lf_verify.pl	
18/07/2014	04:32 PM	22,242 lf_voip.pl	
	24 File(s)	344,077 bytes	

E. Your installation of Perl should have put it into your path variable (%PATH%). Please verify that it did with this command: perl -v

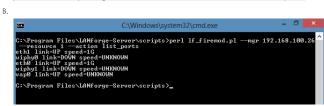


- F. If perl is not found [Command not found] then you might need to close your DOS window and open a new one, or your Windows computer might need a reboot for the PATH variable to take effect.
- G. Start the **lf_firemod.pl** script with the --help switch to see the options.

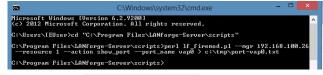


A. perl lf_firemod.pl --help

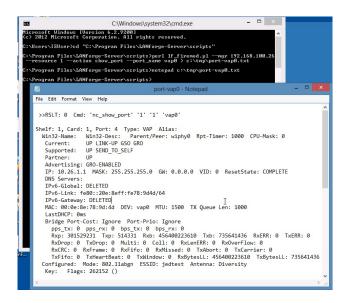
- H. Open a second cmd window so that you can see the help text in the first window. Change directory to C:\Program Files\LANforge-Server\Scripts
- I. Use this command to list ports available on 192.168.100.26:
 - A. perl lf_firemod.pl --mgr 192.168.100.26 --resource 1 --action list_ports



- J. Most command output shows considerably more text than the output of the previous command. You may want to pipe it to a file. In this example, the output is redirected to C:\tmp\port-vap0.txt and shown with Notepad.
 - A. Query the port stats using: perl lf_firemod.pl --mgr 192.168.100.26 --resource 1 --action show_port --port vap0 > c:\tmp\port-vap0.txt



B. Show the output with: notepad c:\tmp\port-vap0.txt



Inspecting Ports (Network Interfaces) using If_portmod

Goal: You will be able to report and reset ports on your LANforge server.

Port statistics can be programatically monitored using the script 1f_portmod.p1. This script can also reset ports, alter WiFi station settings, and pass arbitrary LANforge CLI commands directly to the LANforge manager.

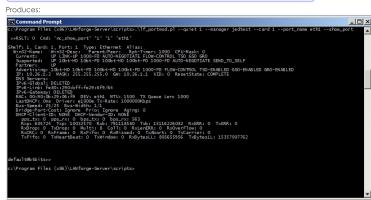


Ports of all kinds can be viewed with the **If_portmod.pl** perl script. You can also do some limited manipulation of ports as well.

Listing Ports

You can show statistic on a port with the --show_port argument: C:\> pert .\lf_portmod.pl --quiet 1 --manager jettest --card 1 --port_name eth1 --show.port

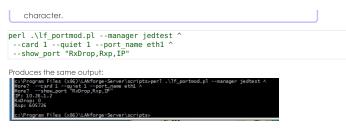
1 You can right-click to paste these commands into your DOS window



Listing Port Attributes

Individual port attributes can also be shown, which often makes automating reporting easier.

pert \\If_partmod.pt --manager jettest --card 1 --pait 1 --pait 1 --part name eth1 --show_part "Rubrap, Rep. IP" Produces: eth?pape, Rup_IP" eth?pape, Rup_IP" pt refore, Rup_IP" consider that is a lot of text to type. If we want, we can reformat that command. f Long DOS commands and be continued on the next line with the



Loading a test scenario

Saved test scenarios are often referred to as 'databases'

lf_portmod.pl --load day_238

This matches the same database name seen in the Status tab database dropdown.

ontrol Rep	orting Tear-Off	Info Plugins						
					Stop All	Restart Manage	r	Refresh HEL
Collision-Dor	nains File-IO	Layer-4 Gene	ric Test Mgr 1	est Group Res	ource Mgr	Event Log Alerts	Port Mgr M	essages
Status	Layer-3	L3 Endps	VoIP/RTP	VoIP/RTP E	Indps	Armageddon	WanLinks	Attenuators
	License Info		Current U	sers	1	Test Con	figuration Datab	ase
Licenses exp	oire in: 44 days.		Imin from:192.168.1 server from:127.0.0		List:	day_238	-	Load
					Name:	day_237		lete
					Load Beh	navior: day_236		ave
Support expi	res in: 44 days.					day_235		
		•	1	•	3	_day_234		Progress
						day_233		
				Virtual She	elf 1	day_232 day_231		
			Re	source 1	Resource 2	day 230		
						day 229		
						day 228		
						day 227		
			•• •			day 226		
						day_225		
					-	day_224		
						day_223		
						day_222		
			N	etsmith	Netsmith	day_221		
						day_220		
						day_219		
						day_219 day_218 day_217		_

Admin-down a port

lf_portmod.pl --manager 192.168.1.101 --card 1 --port_name eth2 --set_ifstate down

Resetting a Port

Resetting a port forces a port to unload and reload its configuration.

lf_portmod.pl --manager 192.168.1.101 --card 1 --port_name eth2 --cmd reset

Sending a specific CLI command to the LANforge manager:

It is possible to directly pass a command to the LANforge manager:

lf_portmod.pl --manager 192.168.1.101 --cli_cmd "scan 1 1 sta0"

Cross Connects and Endpoints Tutorial

Goal: Gain a better understanding on how you will use Cross connects, Connections and Endpoints to use the LANforge CLI scripts knowlegably.

Creating connections in the LANforgeGUI implies creating endpoints. These endpoint entities are created with predictable names and are usually created in pairs. Understanding these naming conventions and how they are created is fundamental to your proficiency with creating connections with LANforge CLI scripts.



Most examples in our cookbooks assume a dual-ended connection, also known as a cross-connect or abbreviated as CX.

Building Endpoints and Connections

Let's follow the creation of a Connection:

<u>\$</u>	tutorial-ex - C	reate	/Modify Cross Connect			
+ · All	Display	Syr	Batch-Create	Apply	ОК	Cancel
CX Name: CX Type:	Cross-Connect tutorial-cx LANforge / UDP			•		
	Endpoint A		Endpoint B			
Resource:	1 (jedtest)	•	1 (jedtest)	-		
Port:	4 (sta301)	-	1 (eth1)	-		
Min Tx Rate:	New Modem (56 Kbps)	-	New Modem (56 Kbps)	-		
Max Tx Rate:	Same	-	Same	-		
Min PDU Size:	AUTO	-	AUTO	-		
Max PDU Size:	Same	-	Same	-		
IP ToS:	Best Effort (0)	-	Best Effort (0)	-		
Pkts To Send:	Infinite	-	Infinite	-		

Using a terminal on the LANforge machine, we look at the /home/lanforge/DB/DFLT/endps.db file and inspect the commands issued that create that connection:

X lanforge∂jedtest:=/DB/DFLT	_ O X
lanforge@jedtest ~/DB/DFLT	
> grep 'tutorial[^]*' *db	
endp.db:add_endp tutorial-cx-A 1 1 sta301 lf_udp -1 N0 56000 0 N0 -1 0 INCREASING N0 32 0 0	
endp.db: set_endp_flag	and the second second
endp.db: set_endp_details tutorial_cx_f 0 0 4294967295 0 '00 90 0b 29 06 f9 ' 0 0 0 0 10000 0 NA NA NA	0.0.0.0 0
endp.db: set_endp_quiesce tutorial-cx-A 3	
endp.db: set_endp_addr tutorial-cx-A '00 0e 8e 24 1f 5b ' AUTO 0 0	
endp.db: set_endp_flag tutorial-cx-A ReplayLoop 0	
endp.db: set_endp_flag tutorial-cx-A EnableTcpNodelay 0	
endp.db: set_endp_flag tutorial-cx-A EnableRndSrcIP 0	
endp.db: set_endp_flag tutorial-cx-A EnableConcurrentSrcIP 0	
endp.db: set_endp_flag tutorial-cx-A EnableLinearSrcIP 0	
endp.db: set_endp_flag tutorial-cx-A EnableLinearSrcIPPort 0	
endp.db: set_endp_flag tutorial_cx_A QuiesceffterRange 0	
endp.db: set_endp_flag tutorial-cx_A QuiesceAfterDuration 0	
endp.db: set_endp_tos tutorial-cx-A DONT-SET 0	
endp.db:set_script tutorial-cx-A NA NA NA NONE 'NA' 0 0	
endp.db: set_endp_proxy tutorial-cx-A NO	
endp.db:rm_thresholds tutorial-cx-A all	
endp.db:set_endp_report_timer_tutorial-cx-A_5000	
endp.db: set_endp_flag tutorial-cx-A ClearPortOnStart 0	
endp.db:add_endp tutorial-cx-B 1 1 eth1 lf_udp -1 NO 56000 0 NO -1 0 INCREASING NO 32 0 0	
endp.db: set_endp_flag	
endp.db: set_endp_details tutorial-cx-B 0 0 4294967295 0 '00 0e 8e 24 1f 5b ' 0 0 0 0 10000 0 NA NA NA	0.0.0.0 0
endp.db: set_endp_quiesce tutorial-cx-B 3	
endp.db: set_endp_addr tutorial-cx-B '00 90 0b 29 06 f9 ' AUTO 0 0	
endp.db: set_endp_flag tutorial_cx_B ReplayLoop 0	
endp.db: set_endp_flag tutorial-cx-B EnableTcpNodelay 0	
endp.db: set_endp_flag tutorial-cx-B EnableRndSrcIP 0	
endp.db: set_endp_flag tutorial-cx-B EnableConcurrentSrcIP 0	
endp.db: set_endp_flag tutorial-cx-B EnableLinearSrcIP 0	
endp.db: set_endp_flag	
endp.db: set_endp_flag tutorial-cx_B QuiescenfiterDuration 0	
endp.db:set_endp_tos <mark>tutorial-cx-B</mark> DONT-SET 0 endp.db:set_script tutorial-cx-B NA NA NONE 'NA' 0 0	
endp.db: set_endp_proxy tutorial-cx-B NO endp.db:rm thresholds tutorial-cx-B all	
endp.db:rm_thresholds tutorial=cx=B all endp.db:set_endp_report_timer tutorial=cx=B 5000	
endp.db:set_endp_report_timer_tutorial_cx_B_5000 endp.db: set_endp_flag_tutorial_cx_B_ClearPortOnStart_0	
endp.db: set_endp_flag tutorial-cx-b tlearPortUnStart V tst_mgr.db:add_cx tutorial-cx default_tm tutorial-cx-A tutorial-cx-B	
tst_mgr.db:set_cx_report_timer_default_tm_tutorial-cx_H_tutorial-cx_b	
tst_mgr.ubtset_ex_report_timer default_tm tutorial_CX 5000 CX0niy	
lanforge@jedtest ~/DB/DFLT	
>	

That's a lot of commands. We will point out what is particularly necessary when using our Perl scripts.

Endpoints and Connections Naming Convention

The connection we created above is named **tutorial-cx**. Two endpoints also have names, tutorial-cx-A and tutorial-cx- ${\bf B}.$ The A-side of a connection is always managed. A B-side endpoint may be unmanaged. When you write CLI scripts that create connections, name your endpoints using a similar convention.

Endpoints are Created First

We can use the If_firemod.pl script to create endpoints and a cross connect in this order:

- \$./lf_firemod.pl --action create_endp --endp_name tutorial2-cx-A \ --speed 256000 --endp_type lf_tcp --port_name sta301

We can see the results of those script commands in our Layer-3 and L3 Endps tabs:

					S	top All	Re	start Mar	nager		Refres	h	HELP
Layer-4 Tes	t Mar T	est Group	R	esource I	lar Event	Log Alerts	Port	Mar	lessages				
Status	Layer			Endps		ageddon		WanLink			uators	Fil	e-10
Rpt Timer:	default	(5s)	▼ G	o Test M	tanager all	-		Select Al	I Star	t Stop	Quiesce	Cle	ar
View	0 - 200				▼ Go			Disp	lay Cr	eate <u>M</u>	dodify D	elete	
			-	Cro	oss Connects	for Selecte	d Test M	lanager-					
Name	Туре	State	P	kt Rx A	Pkt Rx B	Bps Rx	A	Bps	Rx B	Rx Drop	% A Rx Dro	р % В С	rop P
cx-sta300	LF/UDP	Run	33,	823,148	33,802,288	70	57,489		767,016	0.	066	0.113	22
tutorial-cx	LF/UDP			0	0		0		0		0	0	
tutorial2-cx	LF/TCP	Stopped		0	0		0		0		0	0	
4													1
				.10									
Logged in to:	192.168.1	00.26:40	02 a	s: Admin									
\$					L	B Endps						-	-0
						top All	Box	start Mar	ogor		Refrest		HELF
					3	top All	nes	start Mar	lager		Refrest	<u> </u>	HEL
in PDU Size AUT	ro	-	Go	Max PDI	J Size Same		- Go		Cht		0.1000		1
_									Start	Stop	Quiesce	Clear	
IN Tx Rate New	Modem (56 Ki	ops) 🔻	Go	MAX Tx I	Rate Same		▼ Go	_		1			
	400	-	Go					Display	Cr <u>e</u> ate	e <u>M</u> odify	Batch Mod	dify	Delet
iew 0 -													
iew 0 -						All Endpoints	s —						
ew 0 -											Rx Rate	Rx Rat	
ew 0 · Name	EI	D Run	Mng	Sc	ript	Tx Rate	Tx Rate (1 min)		Tx Rate	LL	IN Hace	11 1111	
Name	EI			Sc None	ript	Tx Rate 767,881	(1 min)	68,102		3,626	767,489		767,9
Name cx-sta300-A cx-sta300-B	1.1.2	.1 🖌	~	None None	ript		(1 min) 7		79		767,489	9	
Name cx-sta300-A cx-sta300-B cutorial-cx-A	1.1.2. 1.1.1. 1.1.4.	.1 🖌	× ×	None None None	ript	767,881 767,999 0	(1 min) 7	68,102 68,124 0	79	3,626 1,578 0	767,489 767,016	5	
Name cx-sta300-A cx-sta300-B utorial-cx-A utorial-cx-B	1.1.2 1.1.1 1.1.4 1.1.4	.1 🖌 .2 🖌 .3		None None None None	ript	767,881 767,999 0 0	(1 min) 7	68,102 68,124 0 0	79	3,626 1,578 0 0	767,489 767,016 (9 6 0	
Name cx-sta300-A cx-sta300-B utorial-cx-A utorial-cx-B utorial2-cx-A	1.1.2. 1.1.1. 1.1.4. 1.1.4. 1.1.4.	.1 🖌 .2 🖌 .3 🗌 .4 🛄		None None None None None	ript	767,881 767,999 0 0 0	(1 min) 7	68,102 68,124 0 0	79	3,626 1,578 0 0	767,489 767,016 (
	1.1.2. 1.1.1. 1.1.4. 1.1.4. 1.1.4. 1.1.4. 1.1.4.	.1 🖌 .2 🖌 .3 🗌 .4 🛄		None None None None	ript	767,881 767,999 0 0	(1 min) 7	68,102 68,124 0 0	79	3,626 1,578 0 0	767,489 767,016 (9 6 0	767,9 768,1

Licking on the Layer-3 connection automatically highlights the two endpoints.

Starting and Stopping: Connections have State

When a connection is first created, it is STOPPED. When you start it, it becomes RUNNING. When you set a connection to STOPPED, both endpoints immediately stop sending and recieving. That can have a consequence of leaving unacknowledged packets in flight. The safest way is to QUIECE the connection, which first stops the endpoints transmitting, waits a short time, and then stops the endpoints from recieving.

When there is just one Endpoint

Normally, if you see one endpoint, it should only be a multicast endpoint. A single endpoint can be seen in these situations:

- You have paused between creating the first and second endpoint for a connection. Continue working.
- Created by a script mistakenly, through a typo or other misconfiguration
- Left over from an interrupted script that deleted the cross-connect and one of two endpoints

A single endpoint is not an illegal entity, but lonely endpoints can add confusion. If you find endpoints that do not match any existing connections, we suggest deleting them.

A Cross-Connect can be one-sided, that is, have one unmanaged endpoint. The A side endpoint is a LANforge managed port transmitting to another device that's not a LANforge machine. Some connection types create this style of endpoint pairs, like **File-endpoints** and **Layer 4-7 connections**.

Multicast

Multicast endpoints are created differently both in the GUI and in the CLI scripting environment. This tutorial does not focus on multicast, but see the section Creating Endpoints section of Creating Connections with FIREMOd Script and the chapter on WiFi Multicast Download.

Creating Connections with the FIREmod Script

Goal: Create, destroy, start and stop connections and endpoints without needing to use the LANforge GUI.

Traffic emulation can be run unattended and using automated tools without use of the LANforgeGUI using Perl scripts provided with the LANforge Server. These scripts can be run from within the LANforge server or outside the LANforge Server (on a Windows deskhop). The output of the scripts needs to be redirected into a text file for you to process the results.



Script Capabilities

The 1f_firemod.p1 script has a lot of options because endpoints have a lot of features. Basic actions:

- Creating and Deleting Endpoints and Cross Connects: create_endp, delte_endp, create_cx,
 - delete_cx
- Modifying an Endpoints TX Speed: set_endp
- Listing and Monitoring Ports, Endpoints and Cross Connects: list_ports, show_endp, list_cx, show_cx
- Reporting on Ports, Endpoints and Cross Connects: show_port, show_endp, show_cx
- Controlling Traffic: do_cli, start_endp, stop_endp. To start bi-directional traffic, start both endpoints.
- Pass direct CLI commands: do_cmd. Use this to help configure aspects of your testing scenario that
 are options presented in this script. Like secondary IPs on a port.

Creating a basic cross connect requires two endpoints, and each endpoint requires a port (network interface). Script options often begin by stating the manager, resource and action:

C:\> perl .\lf_firemod.pl --mgr 192.168.100.1 --resource 2 --action create_endp ...more options...

Script Actions, arguments to --action

Creating Endoints: create_endp

We use these parameters when creating and endpoint:

--endp_name

name this endpoint

--port_name name of the port this endpoint uses

--speed

speed of the endpoint transmission in bps

--tos

type of service

--max_speed

Maximum port speed if different than minimum speed of port, in bps

--endp_type

Endpoint Types: tcp, udp, tcp6, udp6. To create multicast endpoint types, use mc_udp and mc_udp6.

--min_pkt_sz/--max_pkt_sz

Minimum and maximum packet sizes

--use_csums Enable checksums

--#1

\$./lf fire

packet Time To Live

--report_timer

the update interval for the endpoint

Example of creating a tcp connection endpoint with debugging:

Creating a multicast udp connection:

Create a connection with specific test-manager

Show Endpoint Stats: show_endp

By default, using the show_endp action shows all endpoints. It might be useful to place output like this right into a file or to immediate use grep to find the rows you want.

RSLT: 0 Cmd: 'nc_show	endp'							
FileEndp [e2#0-nfs-100] Shelf: 1, Card: 1 MinWriteRate: 15444 MinReadRate: 15444 NumFiles: 2 MinFil Directory: AUTO P Server-Mount: 10.43	Port: 10 Endpoin 000bps MaxWriteRa 000bps MaxReadRat LeSize: 26214400B refix: AUTO Volum	t: 1 Type: F1 te: 0bps MinF e: 1544000bps MaxFileSize: e:	LLE_NFS Pat Read/WriteSz QuiesceAf 26214400B	tern: INCREAS: : 40968 MaxRe terFiles: -1	ING	TRUNC)		
RptTimer: 1000ms F LastRpt: 0.000 secs				e Ahos				
RetryTimer: 1000ms								
Buffers Read:	Total: 0	Time: 0s	Cur: 0	0/s				
	Total: 0	Time: 0s		0/s				
	Total: 0	Time: 0s		0/s				
	Total: 0	Time: 0s	Cur: 0	0/s				
Buffers Written:	Total: 0	Time: 0s	Cur: 0	0/s				
Files Written:		Time: 0s		0/s				
Read CRC Failed:	Total: 0	Time: 0s	Cur: 0	0/s				
FileEndp [e2#0-nfs-101] Shelf: 1, Card: 1 MinWriteRate: 15440 MunReadRate: 15440 NumFiles: 2 MinFil Directory: AUTO P	Port: 12 Endpoin 000bps MaxWriteRa 000bps MaxReadRat LeSize: 26214400B	t: 2 Type: F1 te: 0bps MinF e: 1544000bps MaxFileSize:	LE_NFS Pat Read/WriteSz QuiesceAf	tern: INCREAS : 40968 MaxRe	ING	TRUNC)		
Server-Mount: 10.41				ions:				
RptTimer: 1000ms F								
LastRpt: 0.000 secs		eRate: Obps	RealReadRat	e: Obps				
RetryTimer: 1000ms								
Buffers Read:	Total: 0	Time: 0s	Cur: 0	0/s				
Bytes Read:	Total: 0	Time: 0s	Cur: 0	0/s				
	Total: 0	Time: 0s		0/s				
	Total: 0	Time: 0s		0/s				
Buffers Written:		Time: 0s Time: 0s		0/s				
Files Written: Read CRC Failed:		Time: 0s Time: 0s		0/s 0/s				
Read CRC Parted:	Totat: 0	1100: 05	cur: o	0/5				

You can redirect all output into a file:

\$./lf_firemod.pl --action show_endp --mgr cholla-f19 > /var/tmp/endp-stats.txt

It is possible to print out one-word attributes, such as MaxWriteRate tx_bps or rx_bps:

./lf_firemod.pl --mgr 127.0.0.1 --quiet 1 --action show_endp --endp_name cx_0-B --endp_vals tx_bps,rx_bps Rx Bytes: 99938104 Tx Bytes: 99993112

Configure Endpoint: set_endp

This is for changing the attributes of an endpoint, such as endpoint TX rate.

\$./lf_firemod.pl --mgr cholla-f19 --action set_endp --endp_name cx_0-A --speed 2000000

Show Port Stats: show_port

This is pretty useful for getting transmit rate on ports during a connection while not having to use the If_portmod script. If you do not specify --port_name, all ports will be listed.

\$./lf_firemod.plaction show_portmgr cholla-f19port_name eth2#0		
Shelf: 1, Card: 1, Port: 10 Type: MacVLAN Alias: Win32-Name: Win32-Desc: Parent/Peer: eth2 Rpt-Timer: 8000 CPU-Mask: 0 Current: UP LINK-UP TSO UFO GSO GRO PROBE_ERROR Supported: UP SEND_TO_SELF		
Partner: UP		
Advertising: 10bt-HD 10bt-FD 100bt-HD 100bt-FD 1000-FD TSO-ENABLED UFO-ENABLED G	0-ENABLED GRO	- ENABLED
IP: 10.41.0.10 MASK: 255.255.255.0 GW: 0.0.0.0 VID: 0 ResetState: COMPLETE		
DNS Servers:		
IPv6-Global: DELETED		
IPv6-Link: fe80::a00:27ff:fe09:183d/64		
IPv6-Gateway: DELETED		
MAC: 08:00:27:09:18:3d DEV: eth2#0 MTU: 1500 TX Queue Len: 0		
LastDHCP: Oms Driver: macvlan Tx-Rate: 1000000Kbps		
Bus-Speed: 0/0 Bus-Width: 0/0		
Bridge-Port-Cost: Ignore Prio: Ignore Aging: 0		
DHCP-Client-ID: NONE DHCP-Vendor-ID: NONE		
pps tx: 0 pps rx: 0 bps tx: 0 bps rx: 0		
Rxp: 5652 Txp: 21 Rxb: 1932984 Txb: 1826 RxERR: 0 TxERR: 0		
RxDrop: 0 TxDrop: 0 Multi: 5652 Coll: 0 RxLenERR: 0 RxOverFlow: 0		
RxCRC: 0 RxFrame: 0 RxFifo: 0 RxMissed: 0 TxAbort: 0 TxCarrier: 0		
TxFifo: 0 TxHeartBeat: 0 TxWindow: 0 RxBytesLL: 2068632 TxBytesLL: 2330		

List Ports, action: list_ports

This is the same as --show_port without the port_name option.

Direct LANforge Command: do_cmd

In case you wanted to pass a CLI command directly in. Below is an example of setting the TOS flag for an endpoint:

C:\> perl .\lf_firemod.pl	mgr 192.168.100.1action do_cmd \
cmd "set_endp_tos cx	_01-A LOWDELAY 10"

See the LANforge CLI User Guide for more info.

Remove endpoint: delete_endp

Remember to remove the cross connect before removing the endpoint.

\$./lf_firemod.pl --action delete_endp --mgr cholla-f19 --endp_name cx-0-A

Create Cross-connect: create_cx

First you want to create two endpoints. You will add those endpoints to your cross connect. This example below shows all three steps:

Below we see the endpoints created:

Control Reporting Tear-Off	Info Plugins			
	Stop All	Restart Manager	Refres	h HELP
Generic Test Mgr Test (Status Layer-3		ILINKS Alerts Port Mgr	Messages File-IO	Layer-4
	Go Max PDU Size Same	Go Sta		esce Clear
	Go MAX Tx Rate Same	Go Display Cr	eate Modify Bat	ch Modify De
0 - 400	GoAll E	indpoints-		
Name EID	Run Mng Script	Tx Rate (1 min)	Tx Rate LL	Rx Rate (1
cx_0-A 1.1.10.11	None None	0 0		0
cx_0-B 1.1.12.12	None None	0 0	0	0
•				Þ
Logged in to: cholla-f19:40	as: Admin			

and the CX details screen:

	Display	Sync	Batch-Create	App	oly OK	Cancel
CX Name: CX Type:	Cross-Connect cx 0 LANforge / TCP			•		
Resource: Port:	Endpoint A 1 (fedora19-64) 10 (eth2#0)	▼ 1 (f	lpoint B fedora19-64) (eth2#1)	•		
Min Tx Rate:	1000000 (1 Mbps)		0000 (1 Mbps)	• •		
Max Tx Rate: Min PDU Size:	1462 (1.42773 KB)	▼ 146	62 (1.42773 KB)	-		
Max PDU Size: IP ToS:	1462 (1.42773 KB) Best Effort (0)		62 (1.42773 KB) st Effort (0)	•		
Pkts To Send:	Infinite	▼ Inf	inite	-		

Show Cross Connects: list_cx

This shows the cross connects and their enpoints:

\$./	'lf_fir	emod.pl	actio	on list_c	<mgr< th=""><th>cholla-f19</th><th></th></mgr<>	cholla-f19	
CX cx	_0, en	dpoint	cx_0-A,	endpoint	cx_0-B		

Show Cross Connect Stats: show_cx

The output of this command begins with the basic stats for the CX and includes the statistics of each endpoint.

ANFORGE_TCP CX: cx_0 : tx_endpoint: cx_0-A r:								
dpoint [cx 0-A] (NOT RU	WING, FIXED PLD	SIZE, RATE BURS	TY, IP PORT	AUTO)				
Shelf: 1, Card: 1 Pe								
MinTxRate: 1000000bp:					В			
DestMAC: 08:00:27:69 SrcMAC: 08:00:27:09	ia:3d Destipado	1r: 18.41.8.11	DestipPort:	0 Quiesce: 3				
Role: CONNECT RptTin	10:50 SICIP: 0	star free of the	Tel On Au	5: DUNI-SET PIIO	ity: 0			
Latency: 0 -:0:- 0	0000000000		8 1 (1)	y-310001. 0000				
Pkt-Gaps: 0 -:0:- 0								
Last-Rpt: 0.000 secs	ago RealTxRate:	Bbps RealRxRa	te: 0bps T	FL: 0 Conn-Timeo	t: 10000			
FileName: SendBadC								
RxDrop%-SEQ: 0.0000 PktsToSend: 0	HXDrops-CX: 0.6	sees conn-limer	: -1-885 C	onn-Pause: 0-005				
Multi-Conn: 0 Active-Co	onnections: A Fi	iles.Played: 8						
RunningInGroup: NONE			os-Failed: (9				
First-Rx: - Ins Mcast	t-Source: 0.0.0.0	9:0						
	Total: 0			0/s				
Rx Pkts (On Wire):	Total: 0	Time: 68s		0/s 0/s				
Rx Bytes: Rx Bytes (On Wire):	Total: 0	Time: 60s Time: 60s		8/s				
Rx 000 Pkts:	Total: 0	Time: 6Ac	Cur: 0	0/s				
RX Wrong Dev:	Total: 0	Time: 60s	Cur: 0	0/s				
RX CRC Failed:	Total: 0	Time: 68s	Cur: 0	0/s				
Rx Bytes (On Mire): Rx 000 Pkts: RX Wrong Dev: RX CRC Failed: RX Bit Errors: Rx Dropped Pkts:	Total: 0	Time: 3s Time: 3s	Cur: 0	0/s 8/s				
Cx Detected:	Otal: 0	11ne: 35	cur: e	8/s				
Rx Duplicate Pkts:		Time: 68s	Cur: A	A/s				
Tx Pkts:	Total: 0	Time: 68s	Cur: 0	0/s				
Tx Pkts (On Wire):		Time: 68s		0/s				
Tx Bytes:	Total: 0	Time: 68s		0/s 8/s				
Tx Bytes (On Wire) Tx Failed Pkts: Tx Failed Bytes:	Total: 0	Time: 3s Time: 60s	Cur: 0	8/s				
Tx Failed Bytes:	Total: 0	Time: 60s	Cur: 0	0/5				
			Cur: 0	0/s				
TCP Retransmits: Conn Timeouts:	Total: 0	Time: 3s	Cur: 0	0/s 8/s				
Conn Timeouts:	lotal: 0	11ne: 385	cur: e	8/s				
dpoint [cx 0-B] (NOT RU	WING, FIXED PLD	SIZE, RATE BURS	TY. IP PORT	AUTO)				
Shelf: 1, Card: 1 Pe	ort: 12 Endpoint	t: 12 Type: LANF	DRGE_TCP P	attern: INCREASIN				
Shelf: 1, Card: 1 Pr MinTxRate: 10000000pr	ort: 12 Endpoint MaxTxRate: 100	t: 12 Type: LANF 00000bps MinPkt	DRGE_TCP Pi Size: 1462B	MaxPktSize: 146	в			
Shelf: 1, Card: 1 Pr MinTxRate: 10000000pp DestMAC: 08:00:27:09	ort: 12 Endpoint MaxTxRate: 100 18:3d DestIpAdo	t: 12 Type: LANF 00000bps MinPkt ir: 10.41.0.10	DRGE_TCP P Size: 1462B DestIpPort:	HaxPktSize: 146 0 Quiesce: 3				
Shelf: 1, Card: 1 Pr MinTxRate: 10000000pp DestMAC: 08:00:27:09 SrcMAC: 08:00:27:69	ort: 12 Endpoint MaxTxRate: 100 18:3d DestIpAdo 1a:3d SrcIp: 0	t: 12 Type: LANF 300000bps MinPkt dr: 10.41.0.10 3.0.0.0 IpPort:	DRGE_TCP P Size: 1462B DestIpPort: 0-0 IpTO	Attern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio				
Shelf: 1, Card: 1 Pr MinTxRate: 10000000pp DestMAC: 08:00:27:09	ort: 12 Endpoint MaxTxRate: 100 18:3d DestIpAdo 1a:3d SrcIp: 0 er: 1000ms Runnd	t: 12 Type: LANF 30000bps MinPkt dr: 10.41.0.10 3.0.0.0 IpPort: LngFor: 0s Stop	DRGE_TCP P Size: 1462B DestIpPort: 0-0 IpTO In: 0s Avg	Attern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio				
Shelf: 1, Card: 1 P4 MinTxRate: 100000000 DestMAC: 08:00:27:09 SrcMAC: 08:00:27:09 Role: ACCEPT RptTim Latency: 0 -:0:- 0 Pkt-Gaps: 0 -:0:- 0	ort: 12 Endpoint MaxTxRate: 100 18:3d DestIpAdd 10:13d SrcIp: 0 ar: 1000ms Runni 0 0 0 0 0 0 0 [0 0 0 0 0 0 0	12 Type: LANF 30000bps MinPkt dr: 10.41.0.10 3.0.0.0 IpPort: ingFor: 0s Stop 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DRGE_TCP Pi Size: 14628 DestIpPort: 0-0 IpTO In: 0s Avg 0] (1) 0] (1)	HTTER: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio Jitter: 0ms	ity: 0			
Shelf: 1, Card: 1 P4 MinTXRate: 10000000p; DestMAC: 08:00:27:09 SrCMAC: 08:00:27:09 Role: ACCEPT RptTim Latency: 0 -:0:-0 Pkt-Gaps: 0 -:0:-0 Last-Rpt: 0.000 secs	ort: 12 Endpoint MaxTxRate: 100 18:3d DestIpAdo 10:3d SrcIp: 0 or: 1000ms Runni 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ago RealTxRate:	E: 12 Type: LANF 30000bps MinPkt dr: 10.41.0.10 3.0.0.0 IpPort: IngFor: 0s Stop 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DRGE_TCP Pi Size: 1462B DestIpPort: 0-0 IpTO In: 0s Avg 0] (1) 0] (1) te: 0bps T	Htern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo	ity: 0			
Shelf: 1, Card: 1 PV MinTXRate: 10000000p; DestMAC: 08:00:27:09 SrcMAC: 08:00:27:09 Role: ACCEPT RptTimu Latency: 0 -:0:-0 Pkt-Gaps: 0 -:0:-0 Last-Rpt: 0.000 secs FileName: SendBadC	Dart: 12 Endpoint MAXTXRate: 106 18:3d DestIpAdc 1a:3d SrcIp: 6 c: 1000ms Runni [0 0 0 0 [0 0 0 0 0 [0 0 0 0 0 [0 0 0 0 0 ago RealTXRate: c: RcvBuft: 0	E: 12 Type: LANF 100000ps MinPkt dr: 10.41.0.10 0.0.0.0 IpPort: IngFor: 0s Stop 0.0.0.0 0 0.0.0 0.0.0.0 0 0.0.0 0.0.0.0.	DRGE_TCP Pi Size: 1462B DestIpPort: 0-0 IpTO In: 0s Avg 0] (1) 0] (1) te: 0bps T D: 0 SND-MS	Htern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0	ity: 0			
Shelf: 1, Card: 1 P4 MinTXRate: 10000000p; DestMAC: 08:00:27:09 SrCMAC: 08:00:27:09 Role: ACCEPT RptTim Latency: 0 -:0:-0 Pkt-Gaps: 0 -:0:-0 Last-Rpt: 0.000 secs	Dart: 12 Endpoint MAXTXRate: 106 18:3d DestIpAdc 1a:3d SrcIp: 6 c: 1000ms Runni [0 0 0 0 [0 0 0 0 0 [0 0 0 0 0 [0 0 0 0 0 ago RealTXRate: c: RcvBuft: 0	E: 12 Type: LANF 100000ps MinPkt dr: 10.41.0.10 0.0.0.0 IpPort: IngFor: 0s Stop 0.0.0.0 0 0.0.0 0.0.0.0 0 0.0.0 0.0.0.0.	DRGE_TCP Pi Size: 1462B DestIpPort: 0-0 IpTO In: 0s Avg 0] (1) 0] (1) te: 0bps T D: 0 SND-MS	Htern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0	ity: 0			
Shelf: Î, Card: Î P MinTxRate: 10000000bp DestMAC: 08:00:27:00 Role: ACCEPT RptTim Latency: 0 -:0: 0 Pkt-Gaps: 0 -:0: 0 Last-Rpt: 0.000 secs FileName: SendBadC RxDroph-SEQ: 0.0000 PktsToSend: 0	ort: 12 Endpoint 5 MaxTxRate: 100 18:3d DestIPAdc 1a:3d SrCIp: 6 rr: 1000ms Runni [0 0 0 0 0 0 0 ago RealTxRate: rc: 0 RcvBuf: 0 RxDroph-CX: 0.6	E: 12 Type: LANF 300000ps MinPkt dr: 10.41.0.10 3.0.0.0 IpPort: ingFor: 0s Stop 0	DRGE_TCP Pi Size: 1462B DestIpPort: 0-0 IpTO In: 0s Avg 0] (1) 0] (1) te: 0bps T D: 0 SND-MS	Htern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0	ity: 0			
Shelf: 1, Card: 1 P(MinTKate: 100000600; DestMAC: 08:00:27:00 SrcMAC: 08:00:27:00 Role: ACCEPT RptTim Latency: 0 -:00: 0 Pht:Gaps: 0 -:00: 0 Last-Rpt: 0.000 secs FileName: SendBadC RxDroph.SEQ: 0.0000 PhtSToSend: 0 Multi-Conn: 0 Active-Ci RunningInGroup: NMU	rt: 12 Endpoint 5 MaxTxRate: 106 18:3d DestIpAdd 18:3d SrCIp: 6 (18:3d SrCIp: 6 (18:3d SrCIp: 6 (18:00 0 0 0 0 0 (18:00 0 0 0 0 0 0 (18:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (18:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (18:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (18:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (18:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (18:00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (18:00 0 0 0 0 (18:00 0 0 0 0 0 (18:00 0 0 0 0 (18:00 0 0 0 0 (18:00 0 0 0 (18:00 0 0 0 (18:00	12 Type: LANF 80000bps MinPkt fr: 10.41.0.10 8.0.0.0 IpPort: IngFo: 0s Stop 00 0	DRGE_TCP P: Size: 1462B DestIpPort: 0-0 IpTO! In: 0s Avg 0] (1) 0] (1) te: 0bps T D: 0 SND-MS: : -1-0ms Cu	httern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0 onn-Pause: 0-0ms	ity: 0			
Shelf: 1, Card: 1 P MinXRate: 10000000; DestMAC: 08:00:27:09 SrcMAC: 08:00:27:09 Role: ACCEPT RPTIM Latency: 0 -:0:-0 Pkt-Gaps: 0 -:0:-0 Last-Rpt: 0.000 sec: FileName: SendBadC Rubroph-SED: 0.0000 PktSToSend: 0 Multi-Conn: 0 Active-C RunningInGroup: NMME First-RwtIns Mcast	port: 12 Endpoint s MaXTxRate: 106 18:30 DestIpAdc tla:31 SrCIp: 6 pr: 1000ms Runni 0 0 0 ago RealTxRate: rc: 0 RxDrop%-CX: 0.6 manctions: 0 Fi Script-steps-cc 0.0.8.6	E: 12 Type: LANF 80000bps MinPkt 80000bps MinPkt IngFor: 0.41.0.10 8.0.0.0 IpPort: IngFor: 0.5 Stop 0.0.0.0.0.0.0 0.0.0.0.0.0.0 0.0.0.0.	DRGE_TCP P: Size: 1462B DestIpPort: 0-0 IpTO: In: 0s Avg 0 [(1) 0 [(1) te: 0bps T 0: 0 SND-MS: : -1-0ms Ci ps-Failed: (bitern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio -Jitter: 0ms FL: 0 Conn-Timeo 5: 0 pnn-Pause: 0-0ms 0	ity: 0			
Shelf: 1, Card: 1 P/ MinTKate: 100000600; ScrMAC: 08:00:27:00 ScrMAC: 08:00:27:00 Role: ACCEFT RptTim Latency: 0 -:0:-0 Pht:Gap: 0 -:0:-0 Pht:Gap: 0 -:0:-0 Last:Rpt: 0.000 sec: FileName: SendBadC, RxDroph.SEQ: 0.0000 Pht:SToSend: 0 Multi-Conn: 0 Active-Ct RunningInGroup: NMCB First:Rx: -Ins Kcast Rx Phts:	rt: 12 Endpoint : MaxTxRate: 106 :18:3d DestIpAdd :18:3d SrCIp: 6 :18:3d SrCIp: 6 :18:3d SrCIp: 6 :18:3d SrCIp: 6 :18:3d SrCIp :09:00 00 00 00 :19:00 00 00 00 :19:00 00 00 00 :19:00 00 00 :10:00 00	E: 12 Type: LANF 300000ps MinPkt fr: 10.41.0.10 3.0.0.0 IpPort: IngFor: 05 Stop 3.0.0.0 00 0 0.0.0.0.0 0.0.0.0.0 0.0.0.0.	DRGE_TCP P: Size: 14628 DestIpPort: 0-0 IpTO In: 0s Avg 0] (1) 0] (1) te: 0bps T D: 0 SND-MS : -1-0ms C ps-Failed: 0 Cur: 0	Bitern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DOMT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0 pnn-Pause: 0-0ms 0 0/s	ity: 0			
Shelf: 1, Card: 1 P MinXRate: 10000000; DestMAC: 08:00:27:09 SrcMAC: 08:00:27:09 Role: ACCEPT RPTIM Latency: 0 -:0:-0 Pkt-Gaps: 0 -:0:-0 Last-Rpt: 0.000 sec: FileName: SendBadC Rubroph-SED: 0.0000 PktSToSend: 0 Multi-Conn: 0 Active-C RunningInGroup: NMME First-RwtIns Mcast	<pre>rt: 12 Endpoint E MaxTX41: 106 :18:3d DestIpAdc 1a:3d SrCIp: 6 [0 0 0 0 0 0 0 0 ago RealTXRAte: rc: 0 RcVBuf: 0 RxDrop%-CX: 0.6 Dannections: 0 F; Script-steps.cc t-Source: 0.0.0 Total: 0</pre>	E: 12 Type: LAWE 30000bps MinPkt 30000bps MinPkt 3.0.0.0 IPPort: 0.0.0.0 IPPort: 0.0.0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DRGE_TCP Pi Size: 1462B DestIpPort: 0-0 IpTO: 1n: 0s Avg 0] (1) 0] (1) te: 0bps T 0: 0 SND-MS: : -1-0ms Ci ps-Failed: 1 Cur: 0 Cur: 0	bitern: INCREASIN MaxPktSize: 146 0 Quiesce: 3 5: DONT-SET Prio -Jitter: 0ms FL: 0 Conn-Timeo 5: 0 pnn-Pause: 0-0ms 0	ity: 0			
Shelf: 1, Cards 1 P WhitArkate 100000000 DestMAC: 08.00027:09 Role: ACCEPT RptTim Pht-Gasts 0.0002 Role: SondBack Rolroph-SEQ .0000 Nath-Rpt: 0.000 sec: Rolroph-SEQ .0000 PhtTSGand 0 Ruthicons: 0 Active.0 Multi-Cons: 0 Active.0 Rolroph-SEQ .0000 Firstback 0 Ruthicons: 0 Active.0 Rolroph-SEQ .0000 Firstback 0 Rolroph-SEQ .0000 Rolroph-SEQ .0000 Rolroph-Rolrop	rt: 12 Endpion: * MaxTMate: 100 18:3d DestIpAdd 18:3d DestIpAdd 8:00 - 00 0:00 - 00	1: 12 Type: LAW 30000bps: WinPkt 3: 10.41.0.10 3.0.0.0 JPPort: 5: 500 5: 0000 0 0 0 0 0 0 0 0 0 0 0 0 5: 0000 KalkAna 5: 0 CMM 1000 Conn.Timer 11cs.Played: 0 5: 0 Time: 605 Time: 605 Time: 605 Time: 605 Time: 605	DRGE_TCP P Size: 1462B DestIPPort: 0-0 IpTO: 1n: 0s Avg 0] (1) 0] (1) 10: 0 SNO-MS: : -1-0ms Ci Ds: 0 SNO-MS: : -1-0ms Ci ps-Failed: 1 Cur: 0 Cur: 0 Cur: 0 Cur: 0	itern: NNRRASH MuxPtSize: 146 0 Quiesce: 3 1: DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0 ponn-Pause: 0-0ms 0/s 0/s 0/s 0/s 0/s	ity: 0			
Shelf: 1, Card: 1 P WhinTaRate: 10000000p. DestMAC: 08:00:27:00 SrdMc: 08:00:27:00 Latency: 0 -10:-0 Pat-Cappe: 0 -10:-0 Pat-Cappe: 0 -10:-0 Pat-Cappe: 0 -0:-0 Rodropt:SGL: 0 -000 States - 0 Rodropt:SGL: 0 -000 Rodropt:SGL: 0 -0000 Rodropt:SGL: 0 -000 Rodropt:SGL: 0 -0000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -000000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -000000 Rodropt:SGL:	rt: 12 Endpoint = MaxtWate: 106 18:3d SrcIp: 6 r: 1090ms Runni (0 0 0 0 0 0 0 0 ago RealTwRate: 0 RedTops-CX: 0.6 Annections: 0 Fl Script-steps-cc -Source: 0.8.0 Total: 0 Total: 0 Total: 0 Total: 0	1: 12 Type: LAW 90000bcs MinPkt fr: 10.41.0.10 3.0.0.0 IpPort: IngFor: 85 Stop 5.000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DREE TCP P: Size: 1462B DestIPPort: 0-0 IpTO: 1n: 0s Aug 0] (1) 0] (1) 0] (1) 0] (1) 0 : 0 SND-MS: 1 : -1-0ms C ps-Failed: (Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0	itern: NNCREASH MaxPttDire: 146 0 Quidesce: 3 5: DONT-SET Prio -Jitter: 0ms FL: 0 Conn-Timeo 5: 0 0 0/s 0/s 0/s 0/s 0/s 0/s 0/s	ity: 0			
Shelf: 1, Card: 1 P WhinTaRate: 10000000p. DestMAC: 08:00:27:00 SrdMc: 08:00:27:00 Latency: 0 -10:-0 Pat-Cappe: 0 -10:-0 Pat-Cappe: 0 -10:-0 Pat-Cappe: 0 -0:-0 Rodropt:SGL: 0 -000 States - 0 Rodropt:SGL: 0 -000 Rodropt:SGL: 0 -0000 Rodropt:SGL: 0 -000 Rodropt:SGL: 0 -0000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -000000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -000000 Rodropt:SGL:	rt: 12 Endpoint = MaxtWate: 106 18:3d SrcIp: 6 r: 1090ms Runni (0 0 0 0 0 0 0 0 ago RealTwRate: 0 RedTops-CX: 0.6 Annections: 0 Fl Script-steps-cc -Source: 0.8.0 Total: 0 Total: 0 Total: 0 Total: 0	1: 12 Type: LAW 90000bcs MinPkt fr: 10.41.0.10 3.0.0.0 IpPort: IngFor: 85 Stop 5.000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DREE TCP P: Size: 1462B DestIPPort: 0-0 IpTO: 1n: 0s Aug 0] (1) 0] (1) 0] (1) 0] (1) 0 : 0 SND-MS: 1 : -1-0ms C ps-Failed: (Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0	itern: NNCREASH MaxPtSize: 146 0 Quiesce: 3 5 DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5:0 0nn-Pause: 0-0ms 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s	ity: 0			
Shelf: 1, Card: 1 P WhinTaRate: 10000000p. DestMAC: 08:00:27:00 SrdMc: 08:00:27:00 Latency: 0 -10:-0 Pat-Cappe: 0 -10:-0 Pat-Cappe: 0 -10:-0 Pat-Cappe: 0 -0:-0 Rodropt:SGL: 0 -000 States - 0 Rodropt:SGL: 0 -000 Rodropt:SGL: 0 -0000 Rodropt:SGL: 0 -000 Rodropt:SGL: 0 -0000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -000000 Rodropt:SGL: 0 -00000 Rodropt:SGL: 0 -000000 Rodropt:SGL:	rt: 12 Endpoint = MaxtWate: 106 18:3d SrcIp: 6 r: 1090ms Runni (0 0 0 0 0 0 0 0 ago RealTwRate: 0 RedTops-CX: 0.6 Annections: 0 Fl Script-steps-cc -Source: 0.8.0 Total: 0 Total: 0 Total: 0 Total: 0	1: 12 Type: LAW 90000bcs MinPkt fr: 10.41.0.10 3.0.0.0 IpPort: IngFor: 85 Stop 5.000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DREE TCP P: Size: 1462B DestIPPort: 0-0 IpTO: 1n: 0s Aug 0] (1) 0] (1) 0] (1) 0] (1) 0 : 0 SND-MS: 1 : -1-0ms C ps-Failed: (Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0	itern: NNCREASH MaxPttDire: 146 0 Quidesce: 3 5: DONT-SET Prio -Jitter: 0ms FL: 0 Conn-Timeo 5: 0 0 0/s 0/s 0/s 0/s 0/s 0/s 0/s	ity: 0			
Shelf: 1, Card: 1 P WhinFxBat: 0000000p. DestM4C: 08:00027:00 StrUMC: 08:00027:00 Latency: 0 -00:-0 Patt-Gape: 0 -00:-0 Patt-Gape: 0 -00:-0 Patt-Gape: 0 -00:-0 Radroph:SGL: 0 Radroph:SGL: 0 Radroph:SGL	rt: 12 Endpoint = MaxtWate: 106 18:3d SrcIp: 6 r: 1090ms Runni (0 0 0 0 0 0 0 0 ago RealTwRate: 0 RedTops-CX: 0.6 Annections: 0 Fl Script-steps-cc -Source: 0.8.0 Total: 0 Total: 0 Total: 0 Total: 0	1: 12 Type: LAW 90000bcs MinPkt fr: 10.41.0.10 3.0.0.0 IpPort: IngFor: 85 Stop 5.000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DREE TCP P: Size: 1462B DestIPPort: 0-0 IpTO: 1n: 0s Aug 0] (1) 0] (1) 0] (1) 0] (1) 0 : 0 SND-MS: 1 : -1-0ms C ps-Failed: (Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0 Cur: 0	itern: NNCR451M MaxPttic: 146 0 Quiesce: 3 5 DONT-SET Prio Jitter: 0ms FL: 0 Conn-Timeo 5: 0 0 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s	ity: 0			
Shelf: 1. Cards 1 T Winning to Boodbooks Winning the State Boodbooks Schukz Books Role: ACCEPT Rettime Latency: 0 - 10: 0 Book Role: ACCEPT Rettime Latency: 0 - 100 Book FileName: SendBack Books RunningIngto, 0 Book Running Charlong: NMME Firstik: Jime Kassi Running Charlong: MME Firstik: Jime Kassi Running Charlong: Running Charlong	rt: 12 Endpoint : MaxtRate: 106 18:3d Src1p: 6 r: 1000ms Runnal ago RealTMate: rc: Composition Radroph-CX: 0.6 nenections: 0 F1 Scr1pt-steps-cc -Source: 0.0.0.6 Total: 0 Total: 0 Total	E: 12 Type: LAW 900000ps HinPkt fr: 10.41.0.10 0.0.0 IpPort: 0.0.0 IpPort: 0.0.0 0000 0.00000 0.000000 0.000000 0.0000000 0.00000000	DRGE TCP P SIZE: 14628 DestIpPort: 0-0 IpTO: In: 6s Avg 0] (1) 0] (1) 10: 6 SNO.95 : -1-8ms Cr ps-Failed: (Cur: 0 Cur: 0	itern: NRCRASIN MaxPtSize: 146 0 Quiesce: 3 5 DONT-SET Prio .Jitter: 0ms 	ity: 0			
Shelf: 1. Cards 1 T Winning to Boodbooks Winning the State Boodbooks Schukz Books Role: ACCEPT Rettime Latency: 0 - 10: 0 Book Role: ACCEPT Rettime Latency: 0 - 100 Book FileName: SendBack Books RunningIngto, 0 Book Running Charlong: NMME Firstik: Jime Kassi Running Charlong: MME Firstik: Jime Kassi Running Charlong: Running Charlong	rt: 12 Endpoint : MaxtRate: 106 18:3d Src1p: 6 r: 1000ms Runnal ago RealTMate: rc: Composition Radroph-CX: 0.6 nenections: 0 F1 Scr1pt-steps-cc -Source: 0.0.0.6 Total: 0 Total: 0 Total	E: 12 Type: LAW 900000ps HinPkt fr: 10.41.0.10 0.0.0 IpPort: 0.0.0 IpPort: 0.0.0 0000 0.00000 0.000000 0.000000 0.0000000 0.00000000	DRGE TCP P SIZE: 14628 DestIpPort: 0-0 IpTO: In: 6s Avg 0] (1) 0] (1) 10: 6 SNO.95 : -1-8ms Cr ps-Failed: (Cur: 0 Cur: 0	itern: NRCRASIN MaxPktSine: 146 0 Quiesce: 3 5 DONT-SET Prio Jitter: 0ms TL: 0 Conn-Timeo 5:0 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/	ity: 0			
Sheif: 1, Cards 1 P HurthSate: 1000000pr. HurthSate: 1000000pr. Des:Mol: ACCPT PatTim Latency: 0 - 00-0 Pit-Gaps: 0 - 00-0 Hill: Gamma 1000000pr. Hill: Gamma 10000000000000000000000000000000000	rt: 12 Endpoint • MaxX#Att: 106 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 5 19:3d Src1p: 5 Src1pt: 5teps: c - Surcc: 0.0.0. Total: 0 Total: 0 Total	1: 12 Type: LAW 900000ps HinPkt fr: 10.41.0.10 9.0.0 IpPort: IngFor: 0k Stop 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DRGE TCP P SIZE: 14628 DestIpPort: 0-0 IpTO: In: 0s Avg 0] (1) 0 = (1) 10: 0 SNO.95; :-1-0ms Cr ps-Failed: I Cur: 0 Cur: 0	itern: NNCR451M MaxPtSize: 146 0 Quiesce: 3 5 DONT-SET Prio .Jitter: 0ms FL: 0 Conn-Timeo 5:0 0nn-Pause: 0-0ms 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/	ity: 0			
Sheif: 1, Cards 1 P HurthSate: 1000000pr. HurthSate: 1000000pr. Des:Mol: ACCPT PatTim Latency: 0 - 00-0 Pit-Gaps: 0 - 00-0 Hill: Gamma 1000000pr. Hill: Gamma 10000000000000000000000000000000000	rt: 12 Endpoint • MaxX#Att: 106 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 5 19:3d Src1p: 5 Src1pt: 5teps: c - Surcc: 0.0.0. Total: 0 Total: 0 Total	1: 12 Type: LAW 900000ps HinPkt fr: 10.41.0.10 9.0.0 IpPort: IngFor: 0k Stop 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DRGE TCP P SIZE: 14628 DestIpPort: 0-0 IpTO: In: 0s Avg 0] (1) 0 = (1) 10: 0 SNO.95; :-1-0ms Cr ps-Failed: I Cur: 0 Cur: 0	itern: NRCRASIN MaxPttinc: 146 0 Quiesce: 3 5 DONT-SET Prio .litter: 0ms FL: 0 Conn-Timeo 5: 0 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/	ity: 0			
Sheif: 1, Cards 1 P HurthSate: 1000000pr. HurthSate: 1000000pr. Des:Mol: ACCPT PatTim Latency: 0 - 00-0 Pit-Gaps: 0 - 00-0 Hill: Gamma 1000000pr. Hill: Gamma 10000000000000000000000000000000000	rt: 12 Endpoint • MaxX#Att: 106 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 6 18:3d Src1p: 5 19:3d Src1p: 5 Src1pt: 5teps: c - Surcc: 0.0.0. Total: 0 Total: 0 Total	1: 12 Type: LAW 900000ps HinPkt fr: 10.41.0.10 9.0.0 IpPort: IngFor: 0k Stop 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DRGE TCP P SIZE: 14628 DestIpPort: 0-0 IpTO: In: 0s Avg 0] (1) 0 = (1) 10: 0 SNO.95; :-1-0ms Cr ps-Failed: I Cur: 0 Cur: 0	itern: NNCR451M MaxPtSize: 146 0 Quiesce: 3 5 DONT-SET Prio .Jitter: 0ms FL: 0 Conn-Timeo 5:0 0nn-Pause: 0-0ms 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/	ity: 0			
Shaft T, Leris LT, Seris LT, Serie LT, Seri	<pre>rt: 12 Endpoint</pre>	1: 12 Type: LANF 2: 12 Type: LANF 2: 12 Jan 2: 10, 00 Type: 10, 00 T	DRGE_TCP P: DRGE_TCP P: E: DestIpPort: 0-0 IpTO N: 0s Avg 0] (1) 0] (1) 0] (1) 0] (1) 0] (1) 0] (1) 0] (1) 0] (1) 0] (1) 0] (1) 0]	itern: NNCRASIN MaxPktinc: 146 0 Quiesce: 3 i DONT-SET Prio Jitter: 0ms TL: 0 Conn-Timeo 5:0 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/	ity: 0			
Shaft T, Leris LT, Seris LT, Serie LT, Seri	<pre>rt: 12 Endpoint</pre>	<pre>E: 12 Type: LANF 00000000, Minite A. 0.0 0.0.0 Type: Tri 0.0.1 0.0.0 Type: Tri 0.0.0 0.0.0 Type: Tri 0.0.0 0.0 Type: Tri 0.0 Type: Tri 0.0.0 0.0 Type: Tri 0.0 Type: Tri 0.0</pre>	DRGE_TCP P SINC: 14628 DestIPPOrt: 0-0 [JTO: 0-0 [J	<pre>itten:: INCREASIN NEWRISIS:: UNCREASIN NEWRISIS:: USON: SET Proj. Different inter: Bes It: 0 Comm. Timese 0 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0</pre>	ity: 0			
Sheif: 1, Cards 1 P HurthSate: 1000000pr. HurthSate: 1000000pr. Des:Mol: ACCPT PatTim Latency: 0 - 00-0 Pit-Gaps: 0 - 00-0 Hill: Gamma 1000000pr. Hill: Gamma 10000000000000000000000000000000000	<pre>rt: 12 Endpoint</pre>	1: 12 Type: LANF 2: 12 Type: LANF 2: 12 Jan 2: 10, 00 Type: 10, 00 T	DRGE_TCP P DRGE_TCP P DRGE DestIPPOrt: 0-0 IpTO 0-0	itern: NRCRASIN MaxPitize: 146 0 Quiecce: 3 0 Jiter: 0ms)itter: 0ms 0 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s 0/s	ity: 0			

Remove Cross Connect: delete_cx

Remember to delete a cross connect before you delete its endpoints.

\$./lf_firemod.pl --action delete_cx --mgr cholla-f19 --cx_name cx_0

Controlling Traffic

You need to use do_cmd to control Unicast traffic.

By default, cross connects are created in the default_tm test manager. To control them, you want to specify default_tm in your set_cx_state CLI command.

/lf_firemod.pl --mgr 127.0.0.1 --quiet 0 --action do_cmd --cmd "set_cx_state default_tm cx_0 RUNNING"

The format of the command is specified in the CLI User Guide: set_cx_state. Possible CX states include:

- RUNNING
- SWITCH
- QUIECE
- STOPPED
- DELETED
- DILLILD

For Multicast traffic, use start_endp/stop_endp

\$./lf_firemod.pl --mgr cholla-f19 --action stop_endp --endp_name cx_0-A

Multicast Endpoints

There are different options for creating multicast endpoints.

<pre>\$./lf_firemod.plactio</pre>	n create_endpendp_name mcast_xmit_1 \
endp type mc udp	speed 154000 \
mcast addr 224.9.9.8	mcast port 9998 \
rcv mcast NO	port name eth1 \
min pkt sz 1072	max pkt sz 1472 \
use_csums NO	ttl 32

Add secondary IPs to a Port

This is not a default script option, so we use the **do_cmd** action:

C:\> perl .\lf firemod.plmgr	192.168.100.1
action do cmd "set sec ip 1	1 eth1 10.26.0.20-250/24"

See the LANforge CLI User Guide for more info.

Present Options

This is the output of lf_firemod.pl --help:

/lf_firemod.plaction { create_endp show_endp set_endp show_port list_ports do_cmd start_endp stop_endp delete_endp create cx list cx show cx delete cx }]	
[endp_vals {key,key,key,key}]	
# show_endp output can be narrowed with key-value arguments	
# Examples:	
#action show_endpendp_vals MinTxRate,DestMAC,Avg-Jitter	
# Not available: Latency,Pkt-Gaps, or rows below steps-failed.	
# Special Keys:	
#endp_vals tx_bps (Tx Bytes)	



Creating Endpoint Hunt Scripts with CLI API

Goal: Use the the CLI to operate the Endpoint Scripting features of the Layer-3 Endpoints you create.

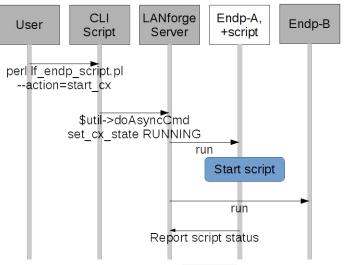
Layer-3 endpoints can manipulate their own transmission parameters using a variety of internal scripts, known as Endpoint Scripts. Using the **1f_endp_script.p1** CLI script, you can operate those internal endpoints behaviours.



This cookbook talks about Endpoint Scripts and CLI scripts at the same time. In this chapter, if the term **script** is used, assume **Endpoint Script**. Additionally, the terms operating and running can also be confusing. To keep the activities distinct, a LANforge user will **operate** a CLI script from a terminal. The LANforge server will **run** the Endpoint Script. A **CLI script** is a user-space perl script that issues CLI commands to a LANforge server. A **CLI command** is an instruction obeyed by the LANforge server.

The Forces at Play

There are a number of subsystems running while we operate an automated Endpoint Script, so let's review them:



- There will be Layer-3 connect constructed using 1f_firemod.p1. (Don't forget: create the endpoints before creating the cross connect.)
- A managed endpoint of that connection will be configured with an Endpoint Script.
- The attending engineer will operate a CLI script that changes state the Layer-3 connection to Running
- The Layer-3 connection starts both endpoints transmitting, one of them starts running it's Endpoint Script that sets it's transmit parameters.

Remember: Endpoint Scripts run inside the LANforge server process. CLI scripts run from the client side.

Let's Walk Thru Putting One Together

We recommend starting your first script off by the LANforge GUI to save an endpoint with an Endpoint Script. Next, inspect the LANforge database on the server for the script parameters. Take those parameters and adapt them to the operator's CLI script.

From the Layer-3 tab, open a connection tutorial-cx, and navigate to box 2. Click on the Script button.

<u>\$</u>									
+ · All					Display	Sync Batch-Create		Apply OK	Cance
CX Name: CX Type:	Cross-Connect tutorial-cx				Report Timer:	Cross-Connect default (5 s)			v
	Endpoint A		Endpoint B			Endpoint A	_	Endpoint B	T
Resource:	1 (jedtest)	-	1 (jedtest)	-	Pld Pattern Min IP Port:	AUTO	- -	AUTO	-
Port:	4 (sta301)	-	1 (eth1)	-	Max IP Port:	Same	-	Sane	
Min Tx Rate:	New Moden (56 Kbps)	-	New Modem (56 Kbps)	-	Min Duration:	Forever	-	Forever	-
Max Tx Rate:	Same	-	Same	-	Max Duration:	Same	-	Sane	-
Min PDU Size:	AUTO	_	AUTO	-	Min Reconn:	0 (0 ms)	-	0 (0 ns)	-
Max PDU Size:	Same	_	Sane	-	Max Reconn:	Same	-	Sane	-
IP ToS:	Best Effort (0)	-	Best Effort (0)	-	Multi-Conn:	Normal (0)	-	Normal (0)	-
Pkts To Send:	Infinite	-	Infinite	-		Script		Script	
						Thresholds		Thresholds	

2. Name your script

4		Add/Modify Sc	ript				
Endpoint Name:	tutorial-cx-A	Script Type:	NONE	-]		
Script Name:	bunny-script	Group Action:	All	-]		
Enable Script	Show Reports	Symmetric 🗌 Loop	Hide Iteratio	n Details 🔲	Hide Legend	🗌 Hide	CSV
Loop Count	Forever	Script Iterations	: NA		Estimated Du	uration:	NA
		Script Config	uration				1
	Show Previous Report	Svnc	Apply	ОК	Cancel		

3. Select your Script type, here we choose **ScriptHunt**

<u>\$</u>		Add/Modify Scrip	ot)×
Endpoint Name:	tutorial-cx-A	 Script Type: 	NONE	-			
Script Name:	bunny-script	Group Action:	NONE RFC-2544]		
🖌 Enable Script	Show Reports		ScriptHunt ScriptWL		Hide Legend	Hide	CSV
Loop Count	Forever	 Script Iterations: 	ScriptAtten		Estimated D	uration:	NA
		Script Configur	ation				
SI	how Previous Report	Sync	Apply	ОК	Cancel		

4. We immediately see the parameters for the script:

\$									
Endpoint Name: tutorial-cx-	A	Script Type:	[ScriptHunt	-				
Script Name: bunny-scri	pt	Group Action:	[All	-				
Enable Script Show	Reports 🔲 Sy	mmetric 🔲 Lo	ор	Hide Iteration I	Details	Hid	e Legend	🗌 Hide CS	V
Loop Count Forever	*	Script Iteration	1S: 1	180 (180)		Estima	ted Duratio	n: 18 m (1	8 m)
		Script C	onf	iguration					1
USE MSS 🗌 Show Du	ips 🔄 Show (nuation 🗹 Hide	Latency	Distrib	outions	Hide Hun	t St
Hide Constraints									
							(-)		
Run Duration:	5 s (5 s)	•	Pause Duration:		1 s	(1 s)		
Starting Rate:	New Modem (56	Kbps)	-	Max Iterations:		20			-
Max Drop Percent:	5% (5%)		-	Max-Tx-Underrun:		10% (10%)		-
Max Jitter:	high (100 ms)	-	Max RT Latency:		500ms	(500 ms)		-
Threshold:	3% (30,00	0)	-						
Payload Sizes	Α	Paylo	ad	Sizes B		-Atte	nuations (d	dB)	_
60 128	6	0 28				NONE		-	
256		56 12			100				- 1
1024	1	024			300				
1280		460			600				
1472		472 514			800 955				
1014		014							
					-	_		1	
Show Previ	ious Report	Sync		Apply	OK		Cancel		

5. Let's modify the parameters to match our CLI command example below:

\$		Add/Moo	lify Script	80
Endpoint Name:	tutorial-cx-A	Script Type:	ScriptHunt	
Script Name:	bunny-script		All	*
Enable Script	Show Reports 🔲 S	mmetric 🔲 Loop 🔲 Hi	de Iteration Details 🛛 🗌 H	iide Legend 🛛 Hide CSV
		Script Iterations:	540 (540)	Estimated Duration: 1.5 h (1.5 h
		Script	Configuration	
USE_MSS	Show Dups 🛛 🔲 Show	000 🕑 Show Attenuati	on 🛛 🗹 Hide Latency Dist	ributions 🔲 Hide Hunt Steps
🔲 Hide Constrain	ts			
Run Duration:	5 s	(5 s)	Pause Duration:	5s (5s)
Starting Rate:	Nev Mode	m (56 Kbps)	Max Iterations:	5
Max Drop Percent:	10% (10	%)	Max-Tx-Underrun:	10% (10%)
Max Jitter:	small	20 ms)	Max RT Latency:	high (100 ms)
Threshold:	3%	30,000)	-	
	load Sizes A		oad Sizes B	Attenuations (ddB)
60 128 256		1472		1.1.14 💌
512 1024 1280 1460 1472 1514				0,100,300,500,550,800,870,700,750,850,900,990
	Show Previous	Report Sync	Apply	0K Cancel
	Show Previous		Apply	OK Cancel db We will search for bunny

script and we'll inspect the resulting CLI command.

Eile	Edit	View	Search	Terminal	Help	
> C	d DB/D	FLT				^
> g endp 00,	rep bu .db:se 60,128	inny-so et_scr 8,256,		lb orial-cx-A 1,1280,146	bumy-script 37120 ScriptHunt '5000 5000 100000,20000,100000,20,56000,30000 0,1472,1514 1472 0,100,300,500,550,600,650,700,750,850,900,950 NONE' ALL 20	,1,1000
t_scr '500	ipt tut 0 5000	orial-cx	-A bunny-	script 37120	ScriptHunt 30000,1,100000, 60,128,256,512,1024,1280,1460,1472,1514 1472 0,100,300,500,550,600,650,700,750,850,9	00.050.00

7. Now we can craft this command into a CLI script. In a CMD window, we can write the formatted CLI script arguments:



In a Linux terminal, we can use double $\begin{tabular}{c} \begin{tabular}{c} \begin{tabular}{c} \end{tabular}$] quotes:

```
$ ./lf endp script.pl --mgr jedtest --resource 1 \
 --action set_script --script Hunt --script name bunny-script \
--action set_script --script --script name bunny-script \
--endp_name tutorial-CX-A -loops 1 --flags 37120 \
--private '5000 5000 100000,20000,100000,20,56000,30000,1,100000, 60,128,256,512,1024,1280,1460,1472,1514 1472 0,100,300,56
```

8. We can start the connection and the Endpoint Script will immediately begin running:

If endp script --mgr jedtest --resource 1 --action start cx --cx name tutorial-CX

9. If the number of loops is fixed, it will eventually quiesce and stop itself. If we need to stop it and let inflight packets come to rest, we can quiesce it:

lf_endp_script --mgr jedtest --resource 1 --action quiesce_cx --cx_name tutorial-CX

We could also use action **stop_cx** to immediately stop the connection.

10. If you have a LANforge GUI running, the Endpoint Script report will automatically display in a GUI window as soon as the connection starts. To display it to the terminal, you need to enable debug output:

lf_endp_script.pl --action show_report --endp_name tutorial-CX-A --quiet no

Or to save it to a text file:

lf_endp_script.pl --action show_report --endp_name tutorial-CX-A --quiet no > /home/lanforge/Documents/report.txt

11. To remove the script:

lf_endp_script.pl --action remove_script --endp_name tutorial-CX-A

At the CLI Command Level

Review of the set script CLI command

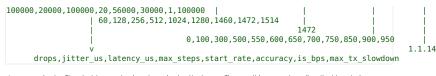
We have covered creating endpoints in earlier cookbooks. The perl script 1f_endp_script.p1 was created to modify endpoints and operate their Endpoint Scripts. That script is using the set_script CLI command (documented here). A call to it looks like:

set_script tutorial-cx-A bunny-script 37120 ScriptHunt '...' ALL 20

Endpoint Scripting Uses Large Parameters

That vague '...' section is the **private** parameter which is a parameter list each script type requires. The private parameter combines a series of constraints (sub-parameters). For the ScriptHunt, we might use:

run_duration	pause_duration	constraints	payload_sizes	_a payload_sizes_b	attenuations	attenuator
5000				1		
	5000	Ì	ĺ	ĺ	İ	Ì



Accuracy is also Threshold, max_tx_slowdown is also Underrun. The result is a very long line that has to be surrounded the the CLI level by one pair of single quotes: 000,1,100000, 60,128,256,512,1024,1280,1460,1472,1514 1472 0,100,300,500,550,600,650,700,750,850,900,950 NONE'

 $^{\pm}$ Write these parameters very carefully! Your first mistake is likely going to involve misplaced apostrophes.

Associating stations with the If_associate_ap_script.

Goal: Create, destroy, start and stop virtual stations without needing to use the LANforge GUI.

Automated wireless traffic is possible using the lf_associate_ap.pl script. This script can be run within the LANforge server or outside the LANforge Server (on a windows desktop). The output of the script should be redirected to a text file if you want to review the results. Use this file in conjunction with the <u>1f_firemod.p1</u> script to create traffic. Requires a LANforge CT520 (or better) system and an access point.



Script Capabilities

The lf_assocatiate_ap.pl script has many options, but here are the basic actions:

- Create stations and cross connects with them, running traffic for a specified amount of time (action: step1)
- Generate stress on the AP by repeatedly bringing up stations and taking them down (action: step2).

Before you begin

- 1. We assume you have a separate WiFi access point in routed mode. These examples can be used on a CT523 (or better) system with more than one radio if you want to practice the techniques. You would dedicate a radio to be a virtual AP (see cookbook).
- 2. For these examples, our AP will be open with no username or password, and the SSID will be jedtest
- 3. If you want to run scripts from your Windows desktop, you have ActivePerl installed.

Creating a virtual station with traffic

Using lf_associate_ap on Windows

1. In the LANforge GUI, we will inspect our wiphy0 radio.

					SI	top All	Restart	Manager		Refrest	HELI
File-IO	Layer	-4 G	Generic Test Mg	r Tes	t Group	Resource	Mgr Event L	og Alerts	Port Mg	Messa	ges
Status	Laye	r-3	L3 Endps VolP	RTP	VoIP/RTP	Endps	Armageddon	WanLinks	Attenuate	ors Coll	ision-Domain
Disp:	192.168.	100.17	78:0.0 SI	niff Pack	tets	Clea	ar Counters	Reset Por	t Dele	te	
Rpt Tin	ner: mea	iium	(8 s) 🔻	Apply		Vie	ew Details	Create	Mod	ify B	atch Modify
	_			-All Et	thernet Inte	erfaces (Pe	orts) for all Reso	urces.			
Port	Phan	Down	IP	SEC	Alias	Parent Dev	RX Bytes	RX Pkts	Pps RX	bps RX	TX Bytes
1.2.08			10.26.2.5	0	br2		0	0	0	0	1.00
1.1.0			192.168.100.26	0	eth0		34,000,280	43,610	20	109,478	46,409,34
1.2.00			192.168.100.42	0	eth0		42,404,600	44,639	6	6,116	32,655,59
1.1.1			10.26.1.2	0	eth1		17,427	160	0	0	95
1.2.01			10.26.1.1	0	eth1		17,427	160	0	0	1,49
1.2.09		~	0.0.0.0	0	vap0	wiphy0	0	0	0	0	
1.2.10		~	0.0.0.0	0	vap2	wiphy2	0	0	0	0	
1.1.2			0.0.0.0	0	wiphy0		0	0	0	0	
1.2.02			0.0.0.0	0	wiphy0		1,715	8	0	0	
1.1.3			0.0.0.0	0	wiphy1		0	0	0	0	
1.2.03			0.0.0.0	0	wiphy1		0	0	0	0	
1.1.4			0.0.0.0	0	wiphy2		0	0	0	0	
1.2.04			0.0.0.0	0	wiphy2		1,839	7	0	0	
1.1.5		2	0.0.0.0	0	wlan0	wiphy0	0	0	0	0	
1.2.05		2	0.0.0.0	0	wlan0	wiphy0	0	0	0	0	
1.1.6		2	0.0.0.0	0	wlan1	wiphy1	0	0	0	0	
1.2.06		~	0.0.0.0	0	wlan1	wiphy1	0	0	0	0	
1.1.7		~	0.0.0.0	0	wlan2	wiphy2	0	0	0	0	
1.2.07		V	0.0.0.0	0	wlan2	wiphy2	0	0	0	0	

Logged in to: 192.168.100.26:4002 as: Admin

And the radio should be set to channel -1 AUTO

			atus Informa	ation				
		LINK-DOW						
	Driver Info: 1	Port Type:	WIFI-Radio	Driver: ath9k	() Bus			
		Port	Configurable	es				
Enable			General In	terface Sett	ings			1
Set IF Down	Down	Aux-N	lgt					
Set MAC	DHCP-IPv6	DHCP	Release	DHCP Ven	tor ID:	None		-
Set TX Q Len	DHCP-IPv4	Seco	ndary-IPs	DHCP Clier	nt ID:	None		-
Set MTU	DNS Servers:	BLANK	-	Peer IP:		NA		
Set Offload	IP Address:	0.0.0.0		Global IPv6	:	AUTO		
Set PROMISC	IP Mask:	0.0.0.0		Link IPv6:		AUTO		
	Gateway IP:	0.0.0.0		IPv6 GW:		AUTO		
	Alias:			MTU:		1500		
	MAC Addr:	00:0e:8e:	4e:5a:56	TX Q Len		0		
	Rpt Timer:	medium	(8 s) 🔻	WiFi Bridg	e:	NONE		-
			WiF	i Settings				
	Max-VIFs:	2048 Ma	k-Stations: 20	048 Max-AP	s: 8 SL	pports: 802.1	1abgn	
	Country:	Unite	ed States (84	.0) 🔻				
	Channel/F	req: AUTO	(-1 Mhz)	-				
	Antenna:	All		-	Tx-Pow	ver: DEFAULT	г	
	RTS:	DEF	AULT		rag:	2346		
	Verbo	se Debug						
Print View Details	Logs	Pro	be Syr	IC	Арр	ју ОК		Canc
CMD window shorte		Pro emd 4	be Syr	10	App	ју ОК		Canc
CMD window shortc	cut: 🖽 R c	emd ୶						Canc
	cut: 🖽 R c	emd ୶						Canc
CMD window shortc	cut: R c	ram Fi	les\LANf					Canc
CMD window shortc	cut: R c at C: \Prog: ANforge-Serv	ram Fi	les\LANf					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/	cut: 🖽 R c e at C: \Prog: ANforge-Serv id.exe	ram Fi	les\LANf					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/J C(Windows\system32\em i\cd "Program Files (x85)\LWh?	cut: R c ANforge-Server Idverse-Server	erd e ram Fi. er\scri	les\LANf					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/J C(Windows\system32\em i\cd "Program Files (x85)\LWh?	cut: R c ANforge-Server Idverse-Server	erd e ram Fi. er\scri	les\LANf					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/ 3:G\Windows\system32\cm i\cd "Program Files (260)\	cut: R c cut: R cut: Cut:	er\scri	les\LAN1 pts					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/ C(Windows\system32\cm i\cd "Program Files (x66)\ i\Program Files (x66)\LWF6 ()\Program Files (x66)\LWF6 ()\Program Files (x66)\LWF6 ()\Program File (x68)\LWF6 ()\UNES \$\frac{1}{2}\$ \under \$\frac{1}{2}\$ C \UNES \$\frac{1}{2}\$ \under \$\under \$	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	emd er\scri er\scri	les\LAN1 pts					Cance
CMD window shortc LANforge Scripts are C:\Program Files\L/ C(Windows\system32\cm (\range Scripts) C(Windows\system32\cm (\range Scripts) C(S) (\range Scri	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	emd e ram Fi er\scri tipts ts>dir s-Server\s	les \LANf pts cripts					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/ C(Windows\system32\cm i\cd "Program Files (x66)\ i\Program Files (x66)\LWfo i\Program Files (x66)\LWfo i\Program Files (x66)\LWfo i\Program Files (x66)\LWfo i\Program Files (x66)\LWfo i\Program Files (x66)\LWfo i\Program File (x1) Madker is CCF Directory of ci\Program File (x1) Madker is CCF Directory of ci\Program File (x1) Madker is CCF	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	ram Fi er\scri -ipts s-Server\s s-Server\s sociate_ap sociate_ap	les \LANf pts cripts					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/ C(Windows\system32\cm (\range Scripts) C(Windows\system32\cm (\range Scripts) C(S) (\range Scri	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	ram Fi er\scri -ipts s-Server\s s-Server\s sociate_ap sociate_ap	les \LANf pts cripts					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/L C\Program Files (x65) \LNfo (Program Files (x65)\LNfo (Program Files (x65)\LNfo (Program Files (x65)\LNfo (S1) (Program Files (x65)\LNfo (S1) (S1) (S1) (S1) (S1) (S1) (S1) (S1)	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	emd e ram Fi er\scri tipts ts>dir s-Server\s	les \LANf pts cripts					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/ C(Windows\system32\em C:\Program Files (L65)\L/Mo C:\Program Files (Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	ram Fi ram Fi er\scri -ipts ts>dir s=Server\s product _p1 product _p1 produ	les\LANf pts cripts .pl					Canc
CMD window shortc LANforge Scripts are C:\Program Files\L/ C(Windows\system32\em C:\Program Files (L65)\L/LO C:\Program Files (L65)\L/LO Volume Script Number 100 CF Directory of c:\Program File C:\Stript Ostin Files (L65)\L/LO Volume Script Number 100 CF Directory of c:\Program File C:\Stript Ostin Files (L65)\L/LO C:\Stript Ostin C:\Stript Ostin Files C:\Stript Ostin Files (L65)\L/LO C:\Stript Os	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	rand ran Fi er\scri er\scri er\scri ersol persol parse,pl var,pl var,pl var,pl var,pl	les\LAN1 pts cripts .pl					Canc
CMD window shortc LANforge Scripts are C:\Program Files(LA C(Windows\system32\cm Locd "Program Files (x66) C:\Program Fil	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	rand ran Fi er\scri er\scri er\scri ersol persol parse,pl var,pl var,pl var,pl var,pl	les\LANf pts cripts .pl					Canc
CMD window shortc LANforge Scripts are C:\Program Files(LA C(Windows\system32\cm L\cd "Program Files (x66) LYPogram Files (x66) LWG "Program F	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	rand ran Fi er\scri er\scri er\scri ersol persol parse,pl var,pl var,pl var,pl var,pl	les\LANf pts cripts .pl					Canc
CMD window shortc LANforge Scripts are C:\Program Files(LA C(Windows\system32\cm L\cd "Program Files (x66) LYPogram Files (x66) LWG "Program F	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	rand ran Fi er\scri er\scri er\scri ersol persol parse,pl var,pl var,pl var,pl var,pl	les\LANf pts cripts .pl					Canc
CMD window shortc LANforge Scripts are C:\Program Files(LA C(Windows\system32\cm L\cd "Program Files (x66) LYPogram Files (x66) LWG "Program F	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	rand ran Fi er\scri er\scri er\scri ersol persol parse,pl var,pl var,pl var,pl var,pl	les\LANf pts cripts .pl					Cance
CMD window shortc LANforge Scripts are C:\Program Files(LA C(Windows\system32\cm L\cd "Program Files (x66) LYPogram Files (x66) LWG "Program F	Cut: R c ANforge-Server ANforge-Server dexe LANforge-Server csFro server>csfro c-Sfro csfro csf	rand ran Fi er\scri er\scri er\scri ersol persol parse,pl var,pl var,pl var,pl var,pl	les\LANf pts cripts .pl					Cance
CMD window shortc LANforge Scripts are C:\Program Files\L/L C:\Program Files(L/L) C:\Program Files(C66)\L/Mio (\Program Files(C66)\L/Mio	Cut: Image: Control of the second seco	ripts ri	les\LANf pts cripts .pl					Cance
CMD window shortc LANforge Scripts are C:\Program Files(L/ C:\Program Files(Cas) C:\Program Files(Cas) C:\Pro	Cut: ⊞ R c and C: \Prog: ANforge-Server decxe LaNforge-Server decxe LaNforge-Server * 6:00 \LaNforge * 6:00 \La	ipts cond of the second condition of the sec	les\LANf pts cripts .pl				<u>]</u>	Cance

3. per1 .\lf_associate_ap.pl --help Will show you the script options.



4. We can create a virtual station with this command:

step1 ta sta100	radio wi	phy0 ions 1	ssid je duratio	dtest ^ n 20 ^	
ommands ar	nd be continue	d on the	next line wit	h the 🔺	
stapirad ta staloorum p DHCC ta staloo 	io wiphyOssii duri tream ethisecuri Appoints O not present, not dstalOO.Old si eated 1 stations 0.1/1 sent to assoc 00 - ethi, done, test: cx-100dor -bps/Tx-B Total: 24998120 Total: 0 Total: 24998120	d jedtest ^ ation 20 ^ ity wpa2p2 t found, dor tations remov ciate ne. Time: 60s Time: 60s Time: 60s	ssphrase jedtes æ, ed Cur: 25000203 Cur: 0	416670/s 0/s 0/s	gr jedtast A
Tx-B ep-8100 Rx Obps / OB Obps / OB Obps / OB Obps / OB Bytes: Bytes: Bytes: Bytes: Bytes:	Total: 0 Total: 0 Total: 0 Total: 0 Total: 0	Time: 60s Time: 60s	Cur: 0 Cur: 0	0/s 0/s 0/s 0/s	-
	step1 ta stal00 p DHCP ommands ar step1	<pre>step1radio wi ta stal00num_stat p DHCPupstream ommands and be confinue stal ommands and be confinue stal p DHCPupstream ommands and be confinue stal p DHCP</pre>	<pre>step1radio wiphy0 ta sta100num_stations 1 p DHCPupstream eth1 ommands and be continued on the stp1radio wiphy0sid jadest A ta bit 00run_stations 1duration 20 commands and be continued on the stp1radio wiphy0sid jadest A ta bit 00run_stations 1duration 20 commands and be continued on the stp1radio wiphy0sid jadest A ta bit 00run_stations 1duration 20 commands and be continued on the stp1radio wiphy0sid jadest A ta bit 00run_stations 1duration 20 commands and be continued on the stp1radio wiphy0sid jadest A ta bit 00run_stations 1duration 20 commands and be continued on the stp1radio wiphy0sid jadest A ta bit 00run_stations 1 commands tatl contodoration 20 conto</pre>	step1 radio wiphy0 ssid jet ta sta100 num_stations1 duration p DHCP upstream eth1security v commands and be continued on the next line wit sta100 main stations1 commands and be continued on the next line wit station sid jetsect commands and be continued on the next line wit station sid jetsect commands sid jetsect sid jetsect comps/fesect	ta isloùruns_stations 1durstion 20 A portor purptreas ettosecurity Mo2passphrase jedtesti security Mo2passphrase jedtesti

5. We can see the port appear in the LANforge GUI:

							S	top All	Restar	t Manager		Refresh	HE
File-IO	Layer	-4 G	eneric Test Mg	Tes	st Group	Resourc	e Mgr Event l	og Alerts	Port Mgr	Messag	jes		
Status	- Li	ayer-3	L3 Endps	VolP	RTP	VolP/RTF	PEndps /	rmageddon	WanL	inks	Attenuators	Collision	n-Domains
	D	isp: 19	2.168.100.178:0.0		Sniff Pa	ackets	Clear	Counters	Reset Po	rt De	lete		
	R	pt Time	r medium (8 s)	-	Ap	ply	View	Details	Create	M	odify <u>B</u> ate	h Modify	
					AI	Ethernet	Interfaces (Port	s) for all Res	ources				
Port	Phan	Down	IP	SEC	Alias	Parent Dev	RX Bytes	RX Pitts	Pps RX	bps RX	TX Bytes	TX Pkts	Pps TX
2.08			10.26.2.5	0	br2		77,016,941	76,095	0	0	28,813,070	54,885	
1.00			192.168.100.26	0	eth0		187,505,665	330,818	52	244,940	286,076,926	323,699	
2.00			192.168.100.42	0	eth0		52,196,073	136,576	16	15,256	167,444,877	150,309	
.1.01			10.26.1.2	0	eth1		78,427,227	76,470	0	0	29,022,954	54,825	
.2.01			10.26.1.1	0	eth1		29,078,047	55,296	0	0	78,376,966	76,019	
1.08			10.26.2.40	0	sta1	wiphy0	17,367	123	0	0	36,081	182	
.1.09			10.26.2.43	0	sta100	wiphy0	818,403	12,386	0	0	27,041,709	23,753	
2.09			0.0.0.0	0	vap0	wiphy0	78,104,068	76,196	0	1	29,949,231	55,081	
2.10		×	0.0.0.0	0	vap2	wiphy2	0	0	0	0	792	8	
1.02			0.0.0.0	0	wiphy0		65,258,035	108,441	5	7,354	81,042,623	76,824	
2.02			0.0.0.0	0	wiphy0		135,873,550	242,060	18	36,031	31,115,199	56,252	
.1.03			0.0.0.0	0	wiphy1		0	0	0	0	0	0	
2.03			0.0.0.0	0	wiphy1		0	0	0	0	0	0	
.1.04			0.0.0.0	0	wiphy2		0	0	0	0	0	0	
2.04			0.0.0.0	0	wiphy2		12,328,608	50,175	0	0	21,576	116	
.1.05			0.0.0.0	0	wlan0	wiphy0	0	0	0	0	0	0	
.2.05			0.0.0.0	0	wlan0	wiphy0	0	0	0	0	0	0	
.1.06			0.0.0.0	0	wlan1	wiphy1	0	0	0	0	0	0	
.2.06			0.0.0.0	0	wlan1	wiphy1	0	0	0	0	0	0	
.1.07			0.0.0.0	0	wlan2	wiphy2	0	0	0	0	0	0	
.2.07		~	0.0.0.0	0	wlan2	wiphy2	0	0	0	0	0	0	

Logged in to: 192.168.100.26:4002 as: Admin

and we can inspect it.

			atus Inforn			
	Current:	LINK-UF	GRO Aut	orized		
	Driver Int	fo: Port Typ	pe: WIFI-ST	A Parent: v	viphy0	
		Port	Configurat	les		
tandard Configuration	Advanced Conf	iguration	Misc Con	figuration	Custo	m WiFi
Enable			General	interface Se	ttings	
Set IF Down	Down	Aux-M	lat			
Set MAC	_	DHCP	-	DHCP Ve	a de a IDa	None
Set TX Q Len	DHCP-IPv6	_		_		
Set MTU	DHCP-IPv4	Secon	idary-IPs	DHCP Cli	ent ID:	None
Set Offload	DNS Servers:	BLANK		Peer IP:		NA
Set PROMISC	IP Address:	0.0.0.0		Global IP	v6:	AUTO
Services —	IP Mask:	255.255.0	.0	Link IPv6	i:	AUTO
HTTP	Gateway IP:	0.0.0.0		IPv6 GW:		AUTO
FTP	Alias:			MTU:		1500
RADIUS	MAC Addr:	00:83:0c:5	57:37:d5	TX Q Len		1000
Low Level	Rpt Timer:	medium	(8 3)	WiFi Brid	ge:	NONE
PROMISC -			W	iFi Settings		
TSO Enabled		itest		AP:	AUT	~
UFO Enabled						
GSO Enabled	Key/Phrase: jed					.11abgn
LRO Enabled	Freq/Channel: 24			Rat		Default
GRO Enabled	WPA WP	A2 OS		Disable	e HT40	Disable SGI

Using lf_associate_ap on Linux

1. Double click on your PUTTY icon and open a connection to your LANforge machine.



- 2. The lf_associate_ap.pl script is in the scripts sub directory.

lanforge@jedtest ~ > cd scripts/				
<pre>lanforge@jedtest ~/scrip > ls lf_associate_ap.pl lf_associate_ap.pl</pre>				
	resource 1mg radio wiphy0 num_stations upstream eth1	ssid jedtest \ 1duration 20 \ security wpa2pa	ssphräse jedtestl <mark>e</mark>	
	,			
ⁱ Long shell comr character.	,		e next line with the 🛛	

4. We will see similar output:

lanforge@jedtes	st w/scripts						
		e 1mgr localhost					
		wiphy0ssid je					
		tations 1duratio					
		eam eth1securit					
	coss-connects, and e		у жраграз	sphrase Jeduesci			
	033-connects, and (00 - ep-B100) doi						
	sta100 /9 do						
		ne. ved sta100, Old s					
	stations to be removed attactions: stalloo (tations remo	vea			
		1/1 seen to asso					
Walting for sta	tions to associate.	1/1 seen to asso	clate				
Creating connec	tions: cx-100 (st	100 - eth1), done.					
		i test: cx-100do	ne.				
started upload							
	ps/Tx-B ep-B100 1	Rx-hps/Tx-B					
	10Mbps / 0B						
	10Mbps / 0B						
	10Mbps / 0B						
Obps / 24MB	10Mbps / 0B						
ep-A100:		Total: 25190840	Time: 60s	Cur: 25244695	420744/3		
		Total: 0			0/s		
ep-B100:	Tx Bytes:	Total: 0	Time: 60s	Cur: 0	0/3		
	Rx Bytes:	Total: 25190840			420737/8		
Adjusting tx ra	te for download	cx-100done					
Started downloa							
ep-A100 Rx-b	ops/Tx-B ep-B100 D	Rx-bps/Tx-B					
10Mbps / 0B	Obps / 6MB						
10Mbps / 0B	Obps / 12MB						
10Mbps / 0B	Obps / 18MB						
10Mbps / 0B	0bps / 24MB						
ep-A100:	Tx Bytes:	Total: 0	Time: 60s				
	Rx Bytes:	Total: 25182080					
ep-B100:	Tx Bytes:	Total: 25182080	Time: 60s	Cur: 25359597	422659/s		
	Rx Bytes:	Total: 0	Time: 60s		0/3		
lanforge@jedtes	st ~/scripts						
>							
		=0.00 - 0// =	8.0 Cm	- A (170		2 - 21	

More Traffic Examples

1. Creating Multiple stations that transm

./lf_associate_ap.pl -	-resource 1mgr	localhost \	
action step1	radio wiphy0	ssid jedtest \	
first sta sta100	num stations 10	duration 20 \	
first ip DHCP	upstream eth1	security wpa2passp	hrase iedtest1

2. Creating TCP/IP bursty traffic from 30Mbps to 450 Mbps

	action first_s first_i cxtype bps-max	p DHCP tcp	bps-	L	pstre	eam e					sphras	se jeo	ltes
	ups-illax	45010	μs										
lanforge@jedt	est=/sonpts												<u>=101></u>
arted downlow	00/Te-5 co-3100 Is		1 Re-bos/Is			ep-3112 3a-	tee/Tx-5 ep-51	02 Re-bos/In-B	ep-A103 Re-bri		ep-A104 Ra-boa/		Re-bea/Te
<pre>type/Tx-5 </pre>		Bits Rx-bps/Tx-B	ep-1106 3		ep-B106 Rx-bp		107 Rx-bps/Tx-8	ep-3107 Ra-kpe		ep-Bitte Rorbpi			ep-8109 8
	/ 385 39207	392ps / 08 0bp 15 0bps / 385	39207	/ 15 (bog /	385								1 296:04
	/ 425 32207	592pa / 08 0bp 105 0bpa / 495	39207										1 396:04
Alizpa / CA CB Cberr	Obps / 1268 / 695 355pe	592pe / 00 00p	9200	SHEPs / CO / E5 : Eben /	6002 / 5063 5M5		CB CEps / 1198						1 296:04
SN2pa / 00	0bps / 1663	Shipe / CB Cbp. CB Cbps / MS	/ 743 1	Gizpa / CG	(éps / 1296)								1 2042-p.8
	Tx Sytes: Dy Notes:	Total: 0 Total: 14454000	Time: 61s	/ 18 teps / Cars 0 Cars 14476923	0/#								
	Tx Bytes: Ex Bytes:	Total: 14454000		CEE: 14476196									
	TK Bytest		Time: 61s										
		Total: 6053240 Total: 6053240	Time: 60s	Czz: 6854839									
			Time: 60s Time: 60s										
	Ex Sytes: Tx Sytes:		Time: 60s										
	Ex Sytes: Tx Sytes:		Time: 60s Time: 60s										
	Dx Dytes: Tx Dytes:	Total: 13540040 Total: 12540040	Time: 00s	Car: 12629540 Car: 12620400									
	Dx Dytes: Tx Dytes:		Time: 60s										
-8104:	Rx Sytes: Tx Sytes:	Total: 9768560	Time: 60s	Car: 9020252 Car: 9012675	163670/s 163561/s								
			Time: 60s										
	Tx Bytes: Ex Dytes:		Time: 61s Time: 63s										
	Tx Dytes: Ex Dytes:		Time: 60s										
	Tx Bytes: Ex Bytes:	Total: 0 Total: 7920500	Time: 60s Time: 60s	Car: 1940001	0/# 132345/#								
	Tx Dytes: Dy Dytes:		Time: 61s Time: 60s	Cast: 1935517	192259/a								
		Total: 0 Total: 0 Total: 9206760	Timo: 60a		0/a 153632/a								
	Tx Dytes: Tx Dytes:	Total: 9206760	Time: 61s	Car: 9211980									
	Ex Dytes: Tx Dytes:		Time: 60s Time: 60s										
	Ex Dytes: Tx Dytes:		Time: 61s										
	Tx Dytes: Tx Dytes:		Time: 60s Time: 60s										
	To Dytes: To Dytes:	Total: 10647760 Total: 10647760	Time: 039	Car: 10721760 Car: 10714567									
	Dr. Dytest		Time: 00s										

3. Capturing that report with redirection

./lf_associate_ap.plresource 1mgr localhost \ action step1radio wiphy0ssid jedtest \ first_sta sta100num_stations 10duration 120 \ first_ip DHCPupstream eth1security wpa2passp cxtype tcpbps-min 30Mpbsbps-max 450Mbps &> report.txt	hrase jedtestl∖
i Both DOS and Linux command output can be saved to a file with the so operator.]
L Both DOS and Linux files can be viewed with the more command.	Ì

4. Creating steady UDP traffic to at 450Mbps

```
$ ./lf_associate_ap.pl --resource 1 --mgr localhost \
    --action step1 --radio wiphy0 --ssid jedtest \
    --first_sta sta100 --num_stations 10 --duration 120 \
    --first_ip DHCP --upstream eth1 --security wpa2 --passphrase jedtest1 \
    --cxtype udp --bps-min 450Mpbs \
    --bps-max SAME &> report.txt
$ more report.txt
```

5. Associating to an open AP

./lf_associate_ap.plresource 1	mgr localhost \
action step1radio wi	.phy0ssid jedtest \
first_sta sta100num_stat	ions 10duration 120 \
first ip DHCPupstream	eth1security open

6. Connecting a station at 802.11/abg speeds

./lf_associate_ap.pl	resource 1	mgr localhost \	
action step1	radio wiph	y0ssid jedtest	Λ
first_sta sta100	num_statio	ns 10duration 120	λ

--wifi_mode abg
7. Initializing your test secenario by pre-loading a database. The database is the same name as the

--upstream eth1 --security open $\$

	e Manager Ienorting		n(5.3.3) Info Plugir	28									
and the	tebound.	Tear-On	Into Findu	10		Sto	p All	Restart M	lanager			Refresh	HEL
ile-IO	Layer-4	Generic	Test Mgr	Test Group	Resource Mgr	Event Log	Alerts	Port Mgr	Messages	s			
Status	Layer	-3 1	L3 Endps	VoIP/RTP	VoIP/RTP Endps	Am	nageddon	WanLink	ks At	tenuators	5	Collision-D	omains
	Licen	se Info			Current Users	1		т	est Configu	ration Da	tabas	e	
					192.168.100.178		List:	day_2	36		-	Load	
icenses	expire in:	46 days.		Admin from:19				day_2					_
				gnuserver from	n:127.0.0.1		Name:	day_2				lete	
							Load Behav					ave	
upport e	xpires in:	46 days.						day_2	33				_
				•	11			day_2				Progr	ress
								day_2					
					Vi	tual Shelf	1	day_2					
					Resource 1	R	esource 2	day_2					
						_		day_2					
								day_2					
						1.0		day_2					
								day 2					
						1.00		day_2					
								day_2					
						1.00		day_2	21				
								day_2	20				
					Netsmith		Netsmith	day_2					
								day_2					
								day_2					
								day_2					
								day_2	15			•	
ogged in	to: 192.1	58.100.26	:4002 as: Ad	Imin							_		

--action step1 --radio wiphy0 --ssid jedtest \ --first_sta sta100 --num_stations 10 --duration 120 \ --first_ip DHCP --upstream eth1 --security open \ --db_preload day_236

8. Saving your test state after completing a traffic run

--first_ip DHCP

./lf_associate_ap.pl	resource 1mgr localhost \
action step1	radio wiphy0ssid jedtest \
first sta stal0	0num stations 10duration 120 \
first ip DHCP	upstream eth1security open \
	236db save station results

 Cleaning out your scenario settings after completing a traffic run. We can do this by loading the EMPTY database with the db_postload switch.

LANforge Manager Version(5.3.3) Control <u>Reporting Tear-Off</u> Info <u>P</u> lugi	ns					
		Stop All	Restart Manag	ler	Refresh	HELP
File-IO Layer-4 Generic Test Mg Status Layer-3 L3 Endps	Test Group Resource Mgr Even VolP/RTP VolP/RTP Endps	Log Alerts Armageddon	Port Mgr Mes WanLinks	sages Attenuators	Collision-D	omains
License Info	Virtual SI Resource 1	List: Name: Load Behr eff 1 Resource 2	day_236 day_211 day_210 (day_209 day_208 day_207 day_206 day_205	en-mvn h ty	Load	ress

Logged in to: 192.168.100.26:4002 as: Admin

/lf_associate_ap.pl	resource 1mgr localhost \
action step1	radio wiphy0ssid jedtest \
first_sta sta100	num_stations 10duration 120 \
first_ip DHCP	upstream eth1security open \
db_preload day_236	db_save station_resultsdb_postload EMPTY

Using If_associate_ap to stress test an AP

We can have a series of stations associate and unassociate over and over. This can be quite a bit of exercise for an AP. Below is a command that tests five clients connecting.

./lf_associate_ap.plmgr jedtest	action step2 \
ssid jedtestfirst_sta sta100	first_ip DHCP \
num_stations 10security wpa2	passphrase jedtest1

This will create set of ten stations bring them up and then take them down.

	jreynolds@atlas:-/btbits/x64_btbits/tools - Terminal	
deleting port stal00 deleting port stal01 deleting port stal02 deleting port stal03 deleting port stal04 deleting port stal06 deleting port stal06 deleting port stal06 deleting port stal08	step2esid jedtestfirst_sta stal00first_ip DHCPnum_stat	ions 10security wpa2passphrase jedtest
old stations should be gone now Created 10 stations, now polling for associati 10 stations associated, 10 stations with IPs	on	
Association took about 1 seconds	5 stal08 stal01 stal02 stal03 stal09 stal04 stal06 stal07 are admin	n down, done.
jreynolds@atlas ~/btbits/x64_btbits/tools	I	

Script Options

These might have been update since publication, please check --help output for your version of the script.

```
./lf_associate_ap.pl [--mgr {host-name | IP}]
    [--mgr_port {ip port}] # use if on non
    [--resource {resource}] # use if multipute
                                                                                                      # use if on non-default management port
# use if multiple lanforge systems; defaults to 1
                   [--quiet { yes | no }]
                                                                                                       # debug output; -q
                  ## AP selection
[--radio {name}]
                                                                                                       # e.g. wiphy2
                 [--said {name}]  # e.g. wipny2
[--ssid {ssid}]  # e.g. jedtest
[--security {open|wep|wpa|wpa2}] # station authentication type
[--passphrase {...}]  # implies wpa2 if --security not set
[--wifi_mode {a|abg|abgn|abgnAC|an|anAC|b|bg|bgn|g}]
                  ## station configuration
[--num stations {10}]
                  [--first_ip {DHCP |ip address]]
[--netmask {255.255.0.0}]
                  ##
                                              connection configuration
                  [--cxtype {tcp/tcp6/udp/udp6}] # use a tcp/udp connection, default tcp
[--upstream {name|eth1}]
                            # could be AP or could be port on LANforge
                 // perterms and perterms are perterms are perfected a
                  [--traffic type {separate|concurrent}]
                            # for step1: separate does download then upload
                            # concurrent does upload and download at same time
                  [--db preload {scenario name}]
                            # load this database before creating stations
                           # option intended as a cleanup step
                  [--db save {name}]
                            # save the state of this test scenario after running the
                           # connections, before --db_postload
                  [--db_postload {scenario name}]
                            # load this database after running connections,
                            # option intended as a cleanup step
```

Changing Station WiFi SSID with the CLI API

Goal: Programmatically change a stations SSID

Programatically creating LANforge virtual stations requires using the **add_sta** command. If you already have a station and need to change the SSID, you still use the **add_sta** command.



The general sequence of commands is:

1. if port is up, set port down with:

cur flags=0x1 interest flags=0x800002

- 2. issue add_port with changed SSID
- 3. issue set_port to bring it up with:

cur flags=0x0 interest flags=0x800002

We can create a station using this script command:

We format the parameters:

return fmt_cmd("add_sta", 1, \$resource, \$sta_wiphy, \$sta_name, "\$flags", "\$ssid", "NA", "\$key", \$ap, \$cfg_file, \$mac, \$mode, \$rate, \$amsdu, \$ampdu_factor, \$ampdu_density, \$sta_br_id, "\$flags_mask");

Changing Station POST_IFUP field with the CLI API

Goal: Programmatically change a station's POST_IFUP field.

Creating a series of scripts using the If_associate_ap.pl script is not adequate for negatiating a captive-portal environment, that script does not set the POST_IFUP parameter for the station. However, stations can be modified to gain that field.



Creating a station that negotiates a Captive Portal environment requires the POST_IFUP field to name a script. (Usually a portal-bot.pl script.) We can assign that port parameter with the **set_wlfi_extra2** command. At the time of this writing, there are no perl scripts using this CLI command, but I will show an example here:

1.	xtra2, # resource number
	# port name
	# flush-to-kernel
	# ignore probe
NA,	# ignore auth
NA,	# ignore assoc
NA,	# ignore reassoc
NA,	# corrupt gtk rekey mic
NA,	# radius ip
NA,	# radius port
NA,	# freq 24
NA,	# freq_5
# post	ifup script
	al-bot.plbot bp.pmuser "username"pass "secret"start url "http:
NA	# OCSD
	# 0(5)

The above command would never actually be formatted in in the way it appears above. It would all appear on one line without commentas.

In a perl script, the command could be formatted like:

my \$cmd = fmt cmd("	set wifi extra2", 1,
"sta100",	# port name
Θ,	# flush-to-kernel
"NA",	# ignore probe
"NA",	# ignore auth
"NA",	# ignore assoc
"NA",	<pre># ignore_reassoc</pre>



Important Notes

- the LANforge server treats single-quotes (apostrophes, ¹) as command delimiters. Use only doublequotes (ⁿ) to quote the arguments to the script.
- 2. Do not use newlines (\fbox{n} or carriage-returns (\fbox{n}). That will truncate the command and LANforge will process it immedately.
- 3. These parameters will be provided by the server:
- --mgt The management FIFO
- --ip4 The IP address of the port
- --ip6 The IPv6 address of the port
- --dnsv A comma-separated list of DNS addresses
- -- logout Signals logout

<u>Generating a series of attenuations using</u> <u>data in a CSV file.</u>

Goal: Using the attenuator_series.pl script and a specially formatted csv file, you will be able to re-play an arbitrary series of attenuations.

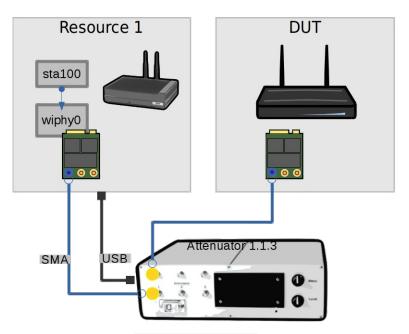
Playing back a series of WIFI attenuation levels using the attenuator_series.pl and a CSV file of attenuations make it possible to emulate the motion of a station (or stations) moving among a series APs. Or it could emulate interference in a crowd of moving people. Requires a LANforge CT703 (or better) and a LANforge CT520 (or better) system, and an access point.



Testing 1x1 with one attenuator

Our LANforge manager (resource 1.1) has an attenuator serial number 3 (resource 1.1.3) connected to the Device Under Test. The attenuator will be 1.1.1.3. There will be station sta100 on LANforge resource 1 and AP vap0 on LANforge resource 2. Cables connect the radios to the the attenuator. The radios are configured in 1x1 mode. The corresponding channel on the attenuator is 1.1.3.0

[See LANforge Entity IDs for more on numbering.]

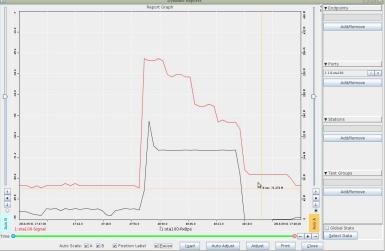


Let's script it with a simple data file: /home/lanforge/atten_test1.csv

channels,1.1.3.0
delay,5000
attenuate,250
attenuate, 320
attenuate,450
attenuate, 520
attenuate,820

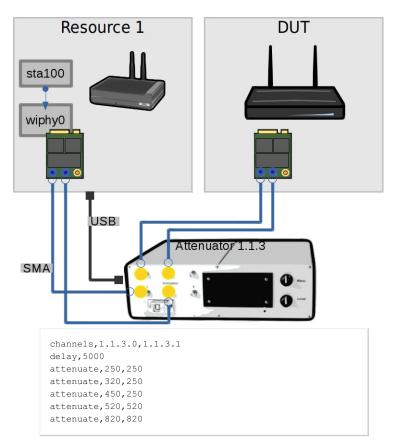
We run the script in our terminal:





Testing 2x2 with One Attenuator

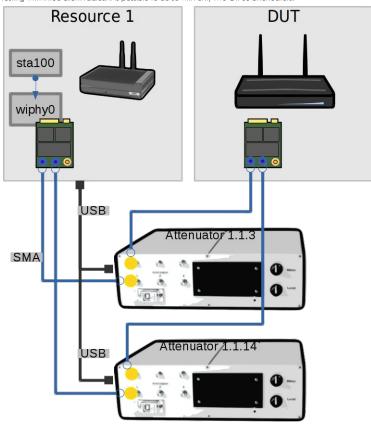
Next we cable up the second channel (1.1.3.1). We can update the csv test file, by adding a new column for the channel.



We can run the same command and watch the dynamic reports window to see a similar graph.

A 2x2 Example with Two Attenuators

The first radio on each LANforge is connected in 2x2 mode to both attenuators. This example is drawn to illustrate how you design the connection of your channels independently of their radios. Obviously, you don't need two attenuators for this scenario. However, if you had a CT523 with three radios and want to perform 2x2 testing with three client radios, it is possible to do so with only two CT703 attenuators.



We change the data file to specify the first channel on attenuator 14 (1.1.14.0): /home/lanforge/atten_test3.csv

channels,1.1.3.0,1.1.14.0 delay,5000 attenuate,250,250 attenuate, 320, 320 attenuate, 450, 450 attenuate, 520, 520 attenuate, 820, 820

We can run the script once in our terminal:

```
$ cd /home/lanforge/scripts
$ ./attenuate_series.pl -f ../atten_test3.csv
```

Watching the port signal in the dynamic display we will see a rise and dip at 10 second intervals.

Connecting up Multiple Radios

There is no different in attenuator control whether you have one radio in 3x3 or three radios in 1x1 to control. If you are testing multiple radios, you will be monitoring their **RX Signal** in the dynamic report.

File Format

Editing the test data file with a basic spreadsheet program than can save to CSV format is possible. You will want to save with comma format, without double-quoting the cells. These directives are converted to lower-case, so you can type them in UPPER-CASE or Mixed-Case if necessary.

The format of the CSV file allows you to specify many options that might also be specified on the command line.

Directives

comments

Rows that begin with a comment sign (#, ;, !) will be entirely ignored. Cells in column B or beyond will be ignored

channels

Each cell following this directive specifies an attenuator channel to control.

sleep, nap

The following cell specifies a one-time wait time in milliseconds

delay, naptime

The following cell specifies a standard wait time in milliseconds between each attenuate command

attenuate, _, ,

The following cells specify an attenuation value for channels specified by the last channels command.

minimum, min

Sets the minimum attenuation permitted. Values below this will be set to the minimum directed.

maximum, max

Sets the maximum attenuation permitted. Values above this will be set to the maximum directed.

Attenuation Values

- Inherently Positive Values, like 200 are absolute attenuation values, in deci-decibels. 200 means 20.0dB. The smallest unit of resolution is 0.5dB, so all your values will end in zero or five. E.G. (0, 5, 105, 200, 955). Values range between zero and 955.
- Explicitly Positive Values, that begin with @+, ++, + are increments with respect to the last value set on the channel.

attenuate,250 attenuate,@+50

Results in the channel at 30.0dB. Spreadsheets often omit signed values when saving, so **Q**+ will force a text type cell.

• Explicitly Negative Values, that begin with @-, --, - are decrements with respect to the last value set on the channel.

attenuate,300 attenuate,@-50

Results in the channel at 25.0dB. Spreadsheets often omit signed values when saving, so @- will force a text type cell.

• Basic Cell Math can be performed, but only against absolute cell values.

attenuate, 500, 400		
attenuate,=B1+50,=C1-50,	#	results in 550, 350
attenuate, =B2+5, =C2-5,	#	fails: B2 and C2 were formulas.

This feature is unlikely to be as useful as it sounds, because pasting a column of forumae will be pretty useless, since a spreadsheet processes them recursively. Also, most spreadsheets saved to CSV typically don't save formulae by default, you probably will get the computed values in your CSV file.

 Shortcuts include, NA, and ,, You can skip a computation on a cell by leaving a blank cell, underscore, or 'NA'. Careful: the value +0 will likely be truncated to 0, and set the channel to 0.0dB attenuation.

Script Options

The attenuate_series.pl script uses these arguments. They support long and short argument switch names:

-m mgr	LANforge manager host, like localhost or 192.168.101.1
-f file	CSV file with attenuation data
- d	

delay	Override of DELAY variable, milliseconds between applying rows
-l loop	Repeat indefinitely
-c channel	Override of channels variable, eg: 1.2.3.1,2.3.4.3
-i min minimum	Set minimum attenuation value (not lower than zero)
-x max maximum	Set maximum attenuation value (not higher than 955)
-n dry dryrun dry_run	Do not apply attenuation, just parse file, ignore nap times

Example CSV File

This CSV shows a working example that gives warnings.

- # example csv
- 2. channels, 1.1.14.0, 1.1.14.1, 1.1.14.2, 1.1.3.0, 1.1.3.1, 1.1.3.2,
- DELAY, 2000, , , , , ,
- 4. ATTENUATE, 950, 850, 750, 950, 850, 750,
- 5. attenuate,940,-10,-10,-10,-10,-10,
- 6. attenuate, 930, -10, NA, -10, -10,,
- attenuate, =B4-10, =C4+10, NA, -15, -15, ,
- 8. attenuate, -15, _, -15, , NA, -15,
- 9. sleep,1000,,,,,
- 10. attenuate, 110, 115, 215, 315, 415, 515,
- 11. _,=B10-20,=C10+20,=D10+20,=E10+20,=F10+20,=G10-20,
- 12. _,@+10,@+10,@-10,10,10,10,
- 13. # eof

Attenuators Tab

Here's the Attenuators tab used for the examples:

Sontrol R	eporting Te		forge Manage Plugins	r Version(5	5.3.5)		
	eporting Te	-	p All	Restart Mar	nager	Re	fresh HELP
Test Mgr Status	Test Grou Layer-3	p Resource L3 Endps	e Mgr Even Armageddo				ons Messages File-IO Layer-4
		Sta		Mo <u>d</u> if	y Delete	•	
Name	State	Script	Module 1	Module 2	Module 3	Module 4	Temperat
1.1.14	Idle	None	1.0	93.0	74.0	HOUGHE 4	87
1.1.3	Idle	None	1.0	74.0	93.0		85
					\$		

Opening and Saving CSV

Here are options used for the open dialog in LibreOffice Calc:

	Text Im	port - [seri	es.csv]		×
Import					
Character set:	Unicode (UTF-8)		~	N	
Language:	Default - English (USA)		~	\$	
From row:	1 +				
Separator Optic	ons				
 Fixed width 	1	Set	eparated by		
🗌 Tab	🗹 Comma 🗌 Ser	nicolon	Space	Othe	er
🗌 Merge de	limiters			Text delir	niter: " 🗸
Other Options					
Quoted fiel	d as text		etect specia	l numbers	
Fields Column type:	~				
Standard		Standard	Standard	Standard	Standard 🛆
2 channels	the attenuator loop	1.1.14.1	1.1.14.2	1.1.14.3	1.1.3.1
3 delay 4 950		1000 850	750	950	850
5 sleep		1000		000	
6 7 -10		-10	-10	-10	-10
°10		-10	NA	-10	-10
Help				ОК	Cancel

Here are the options used for the save dialog in LibreOffice Calc:

hannels	1.1.14	.1 1	1.1.14.2	1.1.14.3	1.1.3.1	1.1.3.2	1.1.	3.3			
lelay	1	000									
attenuate	1	950	850	750	950	850	(750			
sleep	1	000									
attenuate		940	-10	-10	-10	-10		-10		+	
attenuate		930	-10	NA	-10	-10	(
attenuate		920	-10	NA	-10	-10					
attenuate		-10	-10	-10	-10	-10		-10	0		Export Text File
attenuate	++10		10	10	10	10		10	Field Options		
attenuate		10	10	NA	10	10	NA		options		
attenuate			10	NA	10	10	NA		Character set		Unicode (UTF-8)
10		10	10	10	10	10	-				
[#] <u>99</u>		-							Field delimite	r:	,
									Text delimiter		
		_									
		-							Save cell	co	ontent as shown
								_	Save cell	for	omulas instead of calculated values
		-							Quote all	tex	ext cells
											ext cens
								_	Fixed colu	umi	nn width
											Or Count
									Help		OK Cancel

LANforge Entity IDs

Goal: Gain a better understanding of LANforge Entity IDs (EIDs)

Every port, radio, virtual port, endpoint and connection in LANforge has an ID known as an EID. These are an internal notation that expresses the hierarchy of the physical and virtual objects managed by LANforge realm.



Ports, Endpoints and Connections are Entities

Entity IDs (EIDs) are a dotted-decimal phrase. It expresses the Shelf, Resource Number, Port or Connection number, and if it is an endpoint, it gains a fourth decimal. An example:

1.2.8.4	: EID	
1	: shelf	
2	: resource	
8	: port	
	: endpoint	

ontrol	Repor	ting 1	ear-Off Info Plu	igins									
							Stop All		Restart Mar	nager	R	efresh	HELP
File-IO	Layer	r-4 1	est Mgr Test 0	Group	Resour	ce Mgr	Event Log Aler	ts Port I	Mgr Mess	ages			
Sta	atus		Layer-3		L3 Endp	s	Armageo	ldon	W	anLinks		Attenuators	i i
	Disp	: 192	.168.100.51:0.0		Sniff Pac	kets	Clear Cour	nters	Reset Port	Dele	ete		
	Rpt	Timer:	medium (8 s)		Apply	1	<u>V</u> iew Det	ails	Cr <u>e</u> ate	Mod	lify <u>B</u> atch	Modify	
					All E	thernet I	nterfaces (Ports) f	for all Reso	urces. ——				
Port	Pha	Down	IP	SEC	Alias	Parent Dev	RX Bytes	RX Pkts	Pps RX	bps RX	TX Bytes	TX Pkts	Pps
1.1.00			192.168.100.26	0	eth0		8,331,917,784	14,496	23	04 020	12.657.629	13.882	
1.1.00			10.26.1.2	0	eth1		60.084.186.586				59,768,550		
1.1.02			10.26.2.48	0	sta300	wiphy2	59,569,320,101				60.801.796		
1.1.03			0.0.0.0	0	wiphy2	mpriyz	72,934,033,707				61,946,091		
L.1.04			10.26.2.43	0	sta301	wiphv2	974,452			0			
1.1.05			0.0.0.0	0	sta302	wiphy2			0 0				
1.1.13			0.0.0.0	0	vphv0					0		0	
1.1.14			0.0.0.0	0	wiphy0					0	0	0	
1.1.15			0.0.0.0	0	wiphy1			0	0 0	0	0	0	
.1.16			0.0.0.0	0	wlan0	wiphy0	0		0 0	0	0	0	
1.1.17			0.0.0.0	0	wlan1	wiphy1	0		0 0	0	0	0	
.1.18			0.0.0.0	0	wlan2	wiphy2	0		0 0	0	0	0	
.1.19			0.0.0.0	0	hwsim0		0		0 0	0	0		
.2.0			192.168.100.42	0	eth0		1,651,134,987	5,419,636	8	6,678	6,846,252,	5,580,224	
.2.1			10.26.1.1	0	eth1		59,777,383,398	39,439,	. 65	792.097	60,075,036	39,998,	
1.2.2			0.0.0.0	0	wiphy0		0		0 0				
1.2.3			0.0.0.0	0	wiphy1		0			0			
1.2.4			0.0.0.0		wiphy2		70,987,511,003	63,077	. 105	903,372	61,170,347	39,756,	
2.5			0.0.0.0	0	wlan0	wiphy0	0		0 0		0		
2.6			0.0.0.0	0	wlan1	wiphy1	0						
2.7			0.0.0.0	0	wlan2	wiphy2	0		0 0				
L.2.8			10.26.2.1	0	vap0	wiphy2	59.916,602,410	40,014	. 65	788,665	60,400,686	39,417,	
4													

For now, assume the shelf number will always be 1. The Resource number will refer to the LANforge machine ID as reported on the Status tab. The Port ID is only unique within a LANforge machine. The Port ID also refers to hardware in a machine: radios get a third decimal. The fourth decimal refers to either endpoints or connections.

Only Some LANforge Entities Generate Connection Data

While some items with port numbers, notably radios and ports, do not generate traffic. Endpoints generate traffic, and typically endpoints are transmitting to an opposite endpoint. The exception to this are multicast endpoints.

EIDs Express Hierarchy

From the dotted-decimal perspective:

- Physical or virtual ports reside below a resource, except:
- ...for VLANs: A virtual port does not reside below it's physical port
- ...for bridge ports:: A port of a bridge has to exist before the bridge is created
- An endpoint resides below a physical or virtual port.

The formatting of the decimals might or might not be zero-padded. The picture below should convey how a connection (Layer 3) relates to two endpoints, and two ports:

4													
<u>4</u>					Lann		inger version(e						
Control Reportin	g Tear-Off	Info	Plug	lins				_					
							Stop All		Restart Man	ager		Refresh	HE
File-IO Layer-4			est Gr		Resource I	Mgr E	vent Log Alerts		rt Mgr 🛛 Messa				
Status	Lay	er-3	Ľ	l	L3 Endps		Armagedd	on	Wa	anLinks		Attenuat	tors
Rpt Tir	mer: defau	lt (5	s)	▼ Go	Test Mai	nager a	II 👻	S	Select All Sta	rt Stop	Quies	sce Clear	1
View	0 - 20	0			-	Go			Display C	reate M	odify	Delete	
							cts for Selected T	est Ma	anager				
Name	Type S	tate	EIC		En	dpoints ((A c) P)						
Name	Type 5	tate	LIL		LIN	uponits (A O DI						
cx-sta300 L	F/UDP Run		1.2	cx-st	a300-A <=	=> cx-sta	300-B						
	F/UDP Stop		1.3		rial-cx-A <=								
	F/TCP Stop		2.4	tuto	rial2-cx-A <	<=> tuto	rial2-cx-B						
		-											
		_											
Logged in to: 19	2.168.100.	26:40	02 as	: Admi	n								
5							L3 Endps		_				
							Stop All		Restart Man	ager		Refresh	HE
Min PDU Siz				Go	Max PDU	Sizo Sana	-	Go					
14111100 54				00	Munitoo	SILC Suit		100	Start	Stop	Quiesce	e Clear	
and the second second		-	-		a management								
MIN Tx Rate	New Modern (56 Kbps		Go	MAX Tx Ra	ate Seme		Go					
MIN Tx Rate View	0 - 400	56 Kbps	_	Go Go	MAX Tx Ra	ate Some	-	Go	Display Creat	te <u>M</u> odify	Batch I	Modify De	lete
		56 Kbps	_		MAX Tx Ra	ate Same		Go	Display Creat	te <u>M</u> odify	Batch	Modify De	lete
		56 Kbps	_		MAX Tx Ra	ate Same	-All Endpoints -	Go	Display Creat	te <u>M</u> odify	Batch	Modify De	lete
View	0 - 400		•	Go			—All Endpoints—				<u>B</u> atch I	Modify De	lete
			_	Go	MAX Tx Ra	A/B			Display Creat		Batch I	Modify De	lete
View Name	0 - 400 EID	Run	Mng	Go		A/B	— All Endpoints — Source Add	Ir	Destination	n Addr	<u>B</u> atch I	Modify De	lete
View Name cx-sta300-A	0 - 400 EID	Run	Mng	Go		A/B	— All Endpoints — Source Add 10.26.2.48 33009	Ir	Destination	n Addr	Batch	Modify	lete
View Name cx-sta300-A cx-sta300-B	0 - 400 EID 1.1.2.1 1.1.1.2	Run	Mng	Go		A/B A	— All Endpoints — Source Add 10.26.2.48 33009 10.26.1.2 33010	Ir	Destination	n Addr	Batch	Modify De	lete
View Name cx-sta300-A cx-sta300-B tutorial-cx-A	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3	Run		Go S Jone Jone Jone		A/B A B A	All Endpoints	Ir	Destination 10.26.1.2 330 10.26.2.48 330 10.26.1.2 0	n Addr	Batch	Modify De	lete
View Name cx-sta300-A cx-sta300-B tutorial-cx-A tutorial-cx-B	EID 1.1.2.1 1.1.4.3 1.1.1.4	Run		Go S Jone Jone Jone Jone		A/B A B A	All Endpoints	Ir	Destination 10.26.1.2 330 10.26.2.48 330 10.26.1.2 0 10.26.2.43 0	n Addr	Batch	Modify De	lete
View Name cx-sta300-A cx-sta300-B tutorial-cx-A tutorial-cx-B tutorial2-cx-A	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5	Run		Go Go S Jone Jone Jone Jone		A/B A B A B A	All Endpoints	Ir	Destination 10.26.1.2 330 10.26.2.48 330 10.26.1.2 0 10.26.2.43 0 10.26.1.2 0	n Addr	<u>B</u> atch I	Modify De	lete
View Name cx-sta300-A cx-sta300-B tutorial-cx-A tutorial-cx-B tutorial2-cx-A	EID 1.1.2.1 1.1.4.3 1.1.1.4	Run		Go S Jone Jone Jone Jone		A/B A B A B	All Endpoints	Ir	Destination 10.26.1.2 330 10.26.2.48 330 10.26.1.2 0 10.26.2.43 0	n Addr	Batch	Modify De	lete
View Name cx-sta300-A cx-sta300-B cutorial-cx-A cutorial-cx-B cutorial2-cx-A	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5	Run		Go Go S Jone Jone Jone Jone		A/B A B A B A	All Endpoints	Ir	Destination 10.26.1.2 330 10.26.2.48 330 10.26.1.2 0 10.26.2.43 0 10.26.1.2 0	n Addr	Batch	Modify De	lete
View Name cx-sta300-A cx-sta300-B utorial-cx-A utorial2-cx-B utorial2-cx-B	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5	Run		Go Go S Jone Jone Jone Jone		A/B B A B B B B B	All Endpoints- Source Add 10.26.2.48 33000 10.26.1.2 33010 10.26.1.2 0 10.26.1.2 0 10.26.1.2 0	Ir	Destination 10.26.1.2 330 10.26.2.48 330 10.26.1.2 0 10.26.2.43 0 10.26.1.2 0	n Addr	Batch I	Modify De	
View Name cx-sta300-A cx-sta300-B utorial-cx-A utorial2-cx-B utorial2-cx-B	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5	Run		Go Go S Jone Jone Jone Jone		A/B B A B B B B B	All Endpoints	Ir	Destination 10.26.1.2 330 10.26.2.48 33 10.26.2.43 0 10.26.2.43 0 10.26.2.43 0	n Addr 10 009	Batch		
View Name Ex-sta300-A tx-sta300-B tutorial-cx-A tutorial-cx-A tutorial2-cx-B	EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5 1.1.1.6	Run		Go S Jone Jone Jone Jone Jone	cript	A/B A B B B B B B I J	All Endpoints		Destinatioi 10.26.1.2 330; 10.26.2.48 33; 10.26.2.45 0 10.26.2.45 0 10.26.2.45 0 10.26.2.43 0 Restart Mana	n Addr 10 009	Batch I	Modify De Refresh	_ □
View Name cx-sta300-A cx-sta300-8 tutorial-cx-A tutorial2-cx-8 tutorial2-cx-8	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5	Run		Go S Jone Jone Jone Jone Jone		A/B A B B B B B B I J	All Endpoints		Destination 10.26.1.2 330 10.26.2.48 33 10.26.2.43 0 10.26.2.43 0 10.26.2.43 0	n Addr 10 009	Batch		_ □
View Name CX-5ta300-A CX-5ta300-B Uttorial2-CX-B Uttorial2-CX-B Disp:	EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.1.4 1.1.4.5 1.1.1.6	Run		Go S None Jone Jone Jone Jone Jone	cript hiff Packets Apply	A/B A B A B B A B B A S	- All Endpoints Source Add 10.26.2.48 33005 10.26.1.2 33010 10.26.1.2 3010 10.26.1.2 0 Port Mgr Stop All Clear Counte View Detail		Destinatio 10.26.1.2 330 10.26.248 33 10.26.1.2 0 10.26.243 0 10.26.2.43 0 10.26.2.43 0 Restart Mana Reset Port Create	n Addr 10 009			_ □
View Name CX-5ta300-A CX-5ta300-B Uttorial2-CX-B Uttorial2-CX-B Disp:	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.1.4 1.1.1.4 1.1.1.4 1.1.1.4 1.1.1.6	Run		Go S None Jone Jone Jone Jone Jone	cript hiff Packets Apply	A/B A B A B B A B B A S	All Endpoints		Destinatio 10.26.1.2 330 10.26.248 33 10.26.1.2 0 10.26.243 0 10.26.2.43 0 10.26.2.43 0 Restart Mana Reset Port Create	n Addr 10 009 ger Delete		Refresh	_ □
View Name CX-5ta300-A CX-5ta300-B Lttorial2-CX-B Lttorial2-CX-B Disp:	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.4.3 1.1.4.5 1.1.4.5 1.1.1.6	Run		Go S None Jone Jone Jone Jone Jone	cript hiff Packets Apply	A/B A B A B B A B B A S	- All Endpoints Source Add 10.26.2.48 33005 10.26.1.2 33010 10.26.1.2 3010 10.26.1.2 0 Port Mgr Stop All Clear Counte View Detail		Destinatio 10.26.1.2 330 10.26.248 33 10.26.1.2 0 10.26.243 0 10.26.2.43 0 10.26.2.43 0 Restart Mana Reset Port Create	n Addr 10 009 ger Delete		Refresh	_ □
View Name ex-sta300-A cv-sta300-B tutorial-ex-A tutorial2-ex-B Disp: Rpt Tin Port Pha Dc	0 - 400	Run		S S S S S S S S S S S S S S S S S S S	niff Packets Apply All Ether Parent Dev	A/B A B B B B B B C C C C C C C C C C C C	- All Endpoints Source Adc 10.26.1.2 33010 10.26.1.2 33010 10.26.2.43 0 10.26.1.2 0 10.26.1.2 0 Port Mgr Stop All Clear Counte Mew Detail rfaces (Ports) for Mask	Ir ers (Destination 10.26.1.2 330 10.26.2.48 331 10.26.2.45 0 10.26.2.45 0 10.26.2.45 0 Restart Mana Reset Port Create Sources	n Addr 10 009 ger Delete		Refresh	_ □
View Name CX-Sta300-A CX-Sta300-B Utorial-CX-A Utorial2-CX-B Disp: Rpt Tin Port Pha DC .1.00	0 - 400	Run	Mng V N N V N N V N N N N N N N N N N N N N	S S S S S S S S S S S S S S S S S S S	aiff Packets Apply All Ether Parent Dev	A/B A B A B B B B B C C C C C C C C C C C	- All Endpoints Source Add 10.26.2.48 33005 10.26.1.2 33010 10.26.1.2 3010 10.26.1.2 0 10.26.1.2 0 10.26.1.2 0 Port Mgr Clear Counte View Detail fraces (Ports) for Mask 255.255.255.0	Ir o s all Res 00:90	Destination 10.26.1.2 330. 10.26.2.4 23. 10.26.1.2 0 10.26.2.4 30 10.26.1.2 0 10.26.2.4 30 10.26.2.4 30 10.26.2.4 30 Restart Mana Reset Port Crgate sources. MAC	n Addr 10 009 ger Delete		Refresh	_ □
View Name cxsts300-A cxsts300-8 tutorial-cxA tutorial2-cx-B Disp: Rpt Tin Port Pha Dc 1.00	EID 1.1.2.1 1.1.1.2 1.1.1.2 1.1.1.4 1.1.4.3 1.1.1.4 1.1.4.5 1.1.1.6 192.168.10 mer: medium wwn 192.168.10 10.26.1	Run //	Mng V N V N V N V N V N V N V N V N	Go S S S S S S S S S S S S S S S S S S S	cript hiff Packets Apply —All Ether Dev e	A/B A 2 B A br>B A B A B B A B B A B B A B B A B B A B B A B	- All Endpoints Source Add 0.26.1.2 93010 10.26.1.2 93010 10.26.1.2 0 10.26.1.2 0 10.26.1.2 0 Port Mgr Stop All Clear Counte View Detail rfaces (Ports) for Mask 255.255.255.0	Ir 	Destination 10.26.2.40 331 10.26.2.40 331 10.26.2.40 331 10.26.2.43 0 10.26.2.43 0 Restart Mana Reset Port Crgate Sources. MAC :0b.29.06:f8 0b.29.06:f8	n Addr 10 009 ger Delete		Refresh	_ □
View Name CX-5ta300-A CX-5ta300-B Uttorial-CX-A Uttorial-CX-B Disp:	0 - 400 EID 1.1.2.1 1.1.4.3 1.1.4.4 1.1.4.5 1.1.1.6 192.168.10 mer: medium 192.168 10.26.1 10.26.2	Run //	Mng V N V N V N V N V N V N V N V N	S S S S S S S S S S S S S S S S S S S	niff Packets Apply —All Ether Parent e wiphy2 s	A/B A B B B B B S Device th0 th1 th300	- All Endpoints Source Add 10.26.2.48 33005 10.26.1.2 33010 10.26.1.2 3010 10.26.1.2 0 10.26.1.2 0 Port Mgr Stop All Clear Counte give Detail races (Ports) for Mask 255.255.255.0 255.255.255.0	Ir irs [] all Res 00:90 00:90 00:90	Destination 10.261.2.330 10.262.48.331 10.262.430 10.262.430 10.262.430 10.262.430 10.262.430 10.262.430 Restart Mana Reset Port Crgate sources. MAC cb.230.06:f9 .86:61.8555	n Addr 10 009 ger Delete		Refresh	_ □
View Name cxsta300.4 cxsta300.4 ctutorial-cx4 utorial-cx4 utorial-cx4 utorial2-cx4 Disp: Port Port Pha Dc 1.00	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.4.3 1.1.4.3 1.1.4.3 1.1.4.3 1.1.4.3 1.1.4.3 1.1.4.5 1.1.1.6 192.168.10 ner: medium 192.168.10 10.26.1 10.26.2 0.0.0.0	Run	Mng V N V N V N V N V N V N V N V N	Go S S S S S S S S S S S S S S S S S S S	cript aiff Packets Apply All Ether Dev e wiphy2 s wiphy2 s	A/B A B B B B B B B C A B B C A B B C C C C	- All Endpoints Source Add 0.26.1.2 30010 10.26.1.2 30010 10.26.1.2 0 10.26.1.2 0 10.26.1.2 0 10.26.2.43 0 10.26.1.2 0 Port Mgr Stop All Clear Counte View Detail faces (Ports) for Mask 255.255.255.0 255.255.255.0 0.0.0.0	ir ir s 00:90 00:0e 00:0e	Destination 10.26.2.40 331 10.26.2.40 331 10.26.2.40 331 10.26.2.43 0 10.26.2.43 0 10.26.2.43 0 Restart Mana Reset Port Crgate Sources. MAC :0b.29.06:f8 :0b.	n Addr 10 009 ger Delete		Refresh	
View Name CX-5ta300-A CX-5ta300-B Utorial-CX-A Utorial-CX-A Utorial2-CX-B Disp: Rpt Tin Port Pha DC .1.00 .1.01 .02 .03 .04 .04 .04 .04 .04 .04 .04 .04 .04 .04	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.4 1.1.4.5 1.1.1.6 1.1.4.5 1.1.1.6 1.1.4.5 1.1.1.6 1.1.1.6 1.1.2.1 1.1.4.5 1.1.1.6 1.1.2.1 1.1.4.5 1.1.2.1 1.1.4.5 1.1.4.5 1.1.2.1 1.1.4.5 1.1.5.5 1.0.5.5.5 1.0.5.5 1.0.5.5 1.0.5.5.5 1.0.5.5.5 1.0.5.5.5 1.0.	Run	Mng V N V N V N V N V N V N V N V N	Goo S S Jone Jone Jone Jone Jone Jone Jone Jone	aiff Packets Apply —All Ether Parent Dev e wiphy2 s wiphy2 s wiphy2 s	A/B A B B A B B A B B A B B C C C C C C C	- All Endpoints Source Add 10.26.2.48 33005 10.26.1.2 33010 10.26.1.2 3010 10.26.1.2 0 10.26.1.2 0 Port Mgr Stop All Clear Counte give Detail Taces (Ports) for Mask 255.255.255.0 0.0.0.0 255.255.255.0	Ir ir s all Res 00:90 00:90 00:0e 00:0e	Destination 10.261.2320 10.262.46331 10.262.46331 10.262.430 10.262.430 10.262.430 10.262.430 10.262.430 10.262.430 Restart Mana Reset Port Create sources. MAC cb:22.06:f8 cb:22.06:f9 ce:21:f55 ce:27:55 ce:2	n Addr 10 009 ger Delete		Refresh	lete HEL
View Name Cx-sta300-A Cx-sta300-B Utorial-Cx-A Utorial-Cx-A Utorial2-Cx-B Disp:	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.1.2 1.1.4.3 1.1.1.4 1.1.1.4 1.1.1.5 1.1.1.6 192.168.10 ner: medium 192.168.10 10.26.2 0.0.0.0 10.0 10.26.2 2 0.0.0.0	Run	Mng V N V N V N V N V N V N V N V N	Goo S S Jone Jone Jone Jone Jone Jone Jone Jone	cript hiff Packets Apply Apl Ether Parent Dev e wiphy2 s wiphy2 s	A/B A B B B B B B B B B B B B B B B B B	- All Endpoints Source Add 0.26.1.2 3000 0.26.1.2 3000 0.26.1.2 3000 0.26.2.40 3000 0.26.2.40 0 0.26.2.40 0 0.25.25.25.25.0 0.0.0.0 0 255.255.255.25.0 0.0.0.0 0	Ir 5 00:90 00:90 00:00 00:00 00:00 00:00	Destination 10.26.2.29 10.26.2.40 33 10.26.2.40 33 10.26.2.40 12 10.26.2.43 0 10.26.2.43 0 Restart Mana Reset Port Crgate Sources. MAC Nob.29.06:f8 Nob.29.06:f8 Nob.29.06:f8 Nob.29.06:f8 Sol.29.06	n Addr 10 009 ger Delete		Refresh	
View Name CX-5ta300-A CX-5ta300-A Utorial2-CX-A Utorial2-CX-A Utorial2-CX-B Disp:	0 - 400 EID 1.1.2.1 1.1.1.2 1.1.4.3 1.1.4 1.1.4.5 1.1.1.6 1.1.4.5 1.1.1.6 1.1.4.5 1.1.1.6 1.1.1.6 1.1.2.1 1.1.4.5 1.1.1.6 1.1.2.1 1.1.4.5 1.1.2.1 1.1.4.5 1.1.4.5 1.1.2.1 1.1.4.5 1.1.5.5 1.0.5.5.5 1.0.5.5 1.0.5.5 1.0.5.5.5 1.0.5.5.5 1.0.5.5.5 1.0.	Run	Mng	Goo S S Jone Jone Jone Jone Jone Jone Jone Jone	cript aiff Packett Apply All Ether Parent Dev e wiphy2 s wiphy2 s wiphy2 s	A/B A B B A B B A B B A B B C C C C C C C	- All Endpoints Source Add 10.26.2.48 33005 10.26.1.2 33010 10.26.1.2 3010 10.26.1.2 0 10.26.1.2 0 Port Mgr Stop All Clear Counte give Detail Taces (Ports) for Mask 255.255.255.0 0.0.0.0 255.255.255.0	Ir ir s all Res 00:90 00 00:90 00:90 00:90 00:90 00:90 00:90 00:90	Destination 10.261.2320 10.262.46331 10.262.46331 10.262.430 10.262.430 10.262.430 10.262.430 10.262.430 10.262.430 Restart Mana Reset Port Create sources. MAC cb:22.06:f8 cb:22.06:f9 ce:21:f55 ce:27:55 ce:2	n Addr 10 009 ger Delete		Refresh	

The exception is connections. Connections are numbered outside of this hierarchy.

Do I use EIDs in Scripts?

Usually not, for these reasons:

- EIDs are generated at LANforge manager start time, and might depend on the detection order of ports when the PCI bus on the host is enumerated at boot time.
- 2. New EIDs can be created by appending one database to another on non-conflicting devices
- New devices can be hot-added to a LANforge resource, like a programmable attenuator or a USB-Ethernet adapter, generating new Port IDs.

In scripts, it is legal to reference port numbers, but not advised to store them between sessions. If you reference an EID, it should be from within your present LANforge session. If your resources tend to disappear off the network and return (you had a machine reboot) those EIDs are not guaranteed to return.

¹ For ports, only the first two decimals (shelf and resource id) are actually stable across machine reboots.

If you look into the saved scenarios (in /home/lanforge/DB/DFLT) you will notice that ports, endpoints, and connections are refered to by name. Event though in the CLI Users's guide, where it states port number, use names in your scripts:



First-time User Introduction to LANforge: Scripting and GUI

Goal: This outline is a rough and generic overview of our GUI. This outline, that references other Candela Technologies documentation on our website, briefly covers basic GUI tasks and traffic generation that may be shown to a new customer whom has never used the GUI before, without overloading them with great detail.

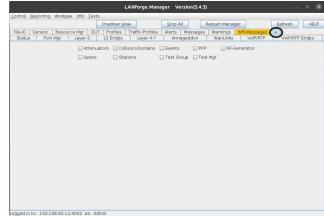


1. Table of Contents

- A. Basic GUI port manager layout and introduction
 - A. Editing the GUI tabs and Port Manager to display relevant information
 - B. Changing Columns in the Port Manager
- B. LANforge GUI Tab Introduction
 - A. Status
 - B. Port Mgr
 - C. Layer-3, Layer-3 Endps
 - D. Layer 4-7
 - E. Resource Mgr
 - F. Messages, Warnings, Wifi Messages
 - G. Using Netsmith
- C. Station Creation
 - A. Searching for Active SSIDs & Connecting to a Particular SSID
- D. MAC-VLAN Creation
- E. Bridge Creation
- F. Virtual AP (VAP) Creation
- G. Monitor Creation
- H. Layer-3 Cross-Connection
- I. Layer-4 Cross Connection
- J. Introduction to Chamber View & Running Scripts in Chamber View

2. Basic GUI Port Manager layout and introduction:

- After connecting the GUI, the interface will automatically open to the Status page. There are 28 tabs/pages that the GUI has, not including the Netsmith View and the Chamber View.
- A. Editing the GUI tabs and Port Manager to display relevant information
 - Upon opening the GUI, several default GUI tabs open as well. Depending on what upcoming WiFi testing must occur, more (or less) GUI tabs may need to be open than the ones defaulted.
 - When running python scripts aimed to automate the GUI, the tabs that the actions in the script are occuring in must be displayed in the GUI (unless the user is running the GUI in headless mode).
 - A. To **display** tabs that are hidden:
 - Click on the + tab under Refresh in the top right hand corner. Then, select which tabs to add to the GUI display.



- B. To hide tabs that are currently displayed:
 - Right-click the mouse on any tab that is aimed to remove and click Hide. This is
 - will remove the tab from the GUI interface currently and will be placed under the + category.

B. Customization of Column Display in the Port Manager

A. In the second tab, Port Manager, comes downloaded with all the tab columns selected to be displayed (73 columns). To change which columns are selected and displayed, Rightclick the mouse in any column space and select Add/Remove Table Columns. From that point, select the necessary columns wished to be displayed in the Port Manager. Nforce Mana

00/543

pro nadas Vinte		Table Columns	Y martine Y martine	Profiles	Alerts Laver		Varnings W Armaged	Ifi Message	+	WanLinks	VolP/	IRT.
4Way Time (us)		Nias	ANOP Time (us)	Dawn		lear Counters	Reset Po			101001065	4007	
Away Time (US)	Beacon	bps RX	bps RX LL									
					I	Disp[ay	Create	Moj	(fy	Batch Modify		
bps TX	bps TX LL	🔲 Bytes RX LL	Bytes TX LL	Interface	s (Port) for all Resourc	es					-
Channel	Collisions	Connections	Crypt		p	IP	Mode	Signa		MAC		
CX Ago	CX Time (us)	Device	DHCP (ms)	· · · ·	2	IP	MOOS	signa		MAL		
Down I	Gateway IP	IP IP	IPv6 Address	Not-Ass			AUTO 20			Se: 55 bd: f0		
Pv6 Gateway	KewPhrase	Login-Fail	Login-OK	Not-Ass			AUTO 20			:De: 5a: 6e:33		
Logout-Fail	Logout-OK	MAC	Mask	Not-Ass Not-Ass			AUTO 20 AUTO 20			8e:32:12:cf		
Misc	Mode	MTU	No CX (us)	4 Not-Ass			AUTO 20			Be 32:e6:cf		
				A Not-Ass			AUTO 20			:8e:32:e7:cf		
Noise	Parent Dev	Phantom Ø	Port Port	4 Not-Ass			AUTO 20			Se:32 de cf		
Port Type	Pps RK	Pps TX	QLEN .	1 Not-Ass			AUTO 20 AUTO 20			70.be.9e.0a		
Reset	Retry Failed	RK Bytes	RX CRC	11101-9455		192.168.92.12		U GOIN		7a:05:69:10		
RX Drop	RX Errors	RK Fifo	RX Frame			0.0.0.0				78:85:69:19		
RX Length	RX Miss	RX Over	RX Pkts			0.0.0.0	802.11ab			8e:32:12:cf		
BX-Rate	SEC .	Isignal	SSID			0.0.0.0	802.11ab			De: Sa: 6e: 33 70: be: Se: 0a		
G Status	Time-Stamp	TX Abort	TX Bytes			0.0.0.0	802.11ab			Se: 55 bd : f0		
		TX Fifo	TX HB	4		0.0.0.0	AUTO 20			Be:4c:e0:cf		
TX Crr	TX Errors			4		0.0.0.0	AUTO 20 AUTO 20			8e:3c:2e:cf 8e:2a:15:cf		
TX Pkts	TX Wind	TX-Failed %	TX-Rate	4		0.0.0.0	AUTO 20			Be: 53:46:cf		
WFI Retries				1								
	Select Al											
	Select Al	Apply	Gancel									

- B. After selecting the columns that wish to be displayed, Right-click the mouse again in the body/rows of the Port Manager and select Save-table Layout. This will make sure the changes don't revert the next time the GUI is opened and closed.
- C. After resizing, one can also Right-click the mouse in the body/rows of the Port Manager and select Auto-size, to auto-size the columns to make sure that all the words under each column are in vision at first glance.
- D. Tip: hot-keys are enabled throughout the entirety of the GUI. In some places in the GUI, there are lines underneath some letters in buttons. To use the keyboard shortcut for that button, press Alt + that letter underlined in the word to press the button. This also works for drop-down menus when the shortcut is enabled via an underlined letter in a word. Note: MAC users need to use key combo ctl+alt + letter to do shortcuts. Circled below are some examples of hotkeys enabled.

-				-	-	LANfor	ge Manager 👋	/ersion(5.4.3)				
ontrol	Be	201	rting Windge	vs info ;	Tests							
-	_			-	/	1	Chamber Mew		itop All	Restart	Manager	Refresh
VolP/R1			os File-IO		V	DUT Profiles Traffic-Pr	-					
				Generic	Resource Mgr					lessages		~
St	atus		P	ort Mgr	Layer	3 L3 Endps	Layer	4.7	Armageddor	-	WanLinks	VolP/RTP
			Di	sp: 10.25	3.1.30:1	Sniff Packets	Down 1 0	lear Counters	Reset Port	Dele		
						0						
			R¢	x Timer: a	redium (8 s)	 Apply 	VRF I	Display	Create	Mod	Batch Modify	
						All Ethernet	interfaces (Ports) for all Resource		_		
Port Ø		I	Parent Dev	Channel	Alias	SSID	AP	P	Mode	Signal	MAC	
.1.06		1	wiphy3		wlan3		Not-Associated	0.0.0.0	AUTO 20		00:0e:8e:56:bd:10	
.1.07					wian1		Not-Associated		AUTO 20		00:0e:8e:5a:6e:33	
1.1.08					wlan0		Not-Associated		AUTO 20		00:0e:8e:32:12:cf	
1.1.09					wlan2		Not-Associated		AUTO 20		00:19:70:be:Se:0a	
1.1.10					sta0000	jedway-wpa2-x2048-4			AUTO 20		00:0e:8e:32:e5:cf	
1.1.11					sta0001	jedway-wpa2+2048-4-4			AUTO 20		00:0e:8e:32:e7:cf	
.1.12					sta0002	jedway-wpa2-x2048-4-4			AUTO 20		00:0e:8e:32:de:cf	
.1.16					sta2000	jedway-wpa2+2048-4-1			AUTO 20		00:19:70:be:9e:0a	
.1.18		1	wiphy2		sta2002	jedway-wpa2-x2048-4-1	Not-Associated		AUTO 20		00:19:70:be:d0:0a	
.1.00					eth0			192.168.92.12				
.1.01					eth1			0.0.0.0			0c c4 7a 86 69 19 00 0c 8e 32 12 cf	
					wiphy0				802.11abgn		00 De Se 52 12 CT 00 De Se 5a 6e 33	
.1.03					wiphy1 wiphy2			0.0.0.0	802.11abgn 802.11an-AC		00:0e:8e:5a:6e:33 00:19:70:be:5e:0a	
.1.04					wiphy2 wiphy3			0.0.0.0	802.11ah.AL 802.11abgn		00:19:70:0e:5e:08 00:0e:8e:56:bd:f0	
.1.13			wiphy0		stal 234	ledway-wpa2-x2048-4		0.0.0.0	AUTO 20		00:0e:8e:4c:e0:cf	
.1.13					stal 234 stal 235	jedway-wpa2-k2048-4 jedway-wpa2-k2048-4		0.0.0.0	AUTO 20		00:0e:8e:3c:2e:cf	
1.1.15					stal 235	jedway-wpa2-x2048-4 jedway-wpa2-x2048-4		0.0.0.0	AUTO 20		00:0e:8e:2a:15:cf	
.1.17					stal 237	iedway-wpa2-x2048-4		0.0.0.0	AUTO 20		00:0e:8e:53:46:cf	
			mprijo	1.57	20022.37	Jeanaj-npaz-vz040-4		0.0.0.0	MOTO 20	o dum		

3. LANforge GUI Tab Introduction

A. Status tab:

Please read the see also below (LANforge Manager) to read about the LANforge Status tab. This is where information about the server is typically stored, configurations of the GUI are able to be saved, and where the Netsmith is.

Control Reporting Windows Info Test		orge Mana	iger Vei	sion(5.4.	.3)				-	0 😣	
Control Reporting windows into Test	Chamber View Stop			II Restart		Manager		Re	Refresh		
File-IO Generic Resource Mgr DU Status Port Mgr Layer-3		fic-Profiles Layer 4-7	Alerts	Message geddon		nings 🔁	Wifi-Messages VoIP/RTP	+	VoIP/RTP I	Endps	
License Info Licenses expire in: 27 days. Support expires in: 27 days. Status Vie <u>w</u> : Ports by Resource	Current Users * Admin from 10.2551.30 Admin from 127.00.1 gruserver from 127.00.1			Configuration: Download DB Name:		DFLT	Test Configura	-	Lo Del Sa	ete	
			er/Resource I								
Logged in to: 192.168.92.12:4002 as: A	dmin										

For more information see Step 2: LANforge Manager

B. Port Mgr tab:

The Port Mgr tab is where all the ports and representations of the radios, wifi objects, and ethernet connections are located. The Port Mgr (or Port Manager) includes the location/appearance of all further MAC-VLANs, 802.1Q-VLANs, Redirects, Bridges, Bonds, GRE Tunnels, WiFi Stations, WiFi VAPs, WiFi Monitors and WiFi Virtual Radios. Please read more about the Port Mgr tab next to see-also below

			-		Chon	nber <u>V</u> iew	Stop All	Bosto	t Manager	Refre	ab [HELE
							_		-		sn	HEU
File-IO		ene		rce Mgr		ofiles Traffic-Profiles			rnings Wifi-Me			
Status			Port Mgr	Layer-3	3 L3 E	ndps Layer 4-7	Armage	eddon [Wa	anLinks Vo	IP/RTP Vol	P/RTP En	idps
Disp	: 1	0.2	53.1.30:1		Sniff Pac	kets 🖌 Down	1 Clear	Counters	Reset Port	Delete		
Rot	Tim	en	nediun (8	s) 🔻	Apply	VBE	I D	isplay	Create	Modify Bat	ch Modif	
			neoran (o		14493	- All Ethernet Interface			orgate	200	en no an	,
						All Ethernet Interfact	es (Ports) for	all Resources		1		
Port	Ø	Ţ	Parent Dev	Channel	Alias	SSID		AP	IP	Mode	Signal	
1.1.06		-	wiphy3	1	wlan3			Not-Associated	0.0.0.0	AUTO 20	0 dBm	00.04
1.1.07				157	wlan1			Not-Associated		AUTO 20	0 dBm	
1.1.08				157	wlan0			Not-Associated		AUTO 20		00:0
1.1.09				0	wlan2			Not-Associated		AUTO 20		00:1
1.1.10				157	sta0000	iedway-wpa2-x2adas				AUTO 20		00:06
1.1.11				157	sta0001			Not-Associated		AUTO 20	0 dBm	00:0
1.1.12				157	sta0002			Not-Associated		AUTO 20	0 dBm	00:0
1.1.16				0	sta2000			Not-Associated		AUTO 20	0 dBm	00:19
1.1.18				0	sta2002			Not-Associated		AUTO 20	0 dBm	00:19
1.1.00					eth0				192.168.92.12			00:04
1.1.01					eth1				0.0.0.0			00:04
1.1.02				0	wiphv0				0.0.0.0	802.11abon		00:06
1.1.03				0	wiphy1				0.0.0.0	802.11abgn		00:06
1.1.04				161	wiphy2				0.0.0.0	802.11an-AC		00:19
				0	wiphy3				0.0.0.0	802.11abgn		00:00
1.1.04			wiphy0	157	stal234	jedway-w	pa2-x2048-4		0.0.0.0	AUTO 20	0 dBm	00:06
1.1.05			wiphy0	157	stal235	jedway-w	pa2-x2048-4		0.0.0.0	AUTO 20	0 dBm	00:00
1.1.05			wiphy0	157	sta1236	jedway-w	pa2-x2048-4		0.0.0.0	AUTO 20	0 dBm	00:00
				157	stal237	iedway-w	pa2-x2048-4		0.0.0.0	AUTO 20	0 dBm	00:04

For more information see Ports (Interfaces)

C. Layer-3 tab, L3 Endps tab:

The Layer-3 tab are where Layer-3 WiFIRE traffic connections are made, started, stopped, modified, and displayed. Each cross-connects have 2 endpoints each. These endpoints and the traffic/data associated with them are found and elaborated under the L3 Endps tab in the GUI. Please visit the introduction to Layer-3 Cross-Connects, linked below, for a general overview.

ontrol Repo	orting Tea	ar-Off Info	Plugins								
						Stop Al	I R	estart Manager	-	Refresh	HE
Attenuators	File-IO	Layer-4	Test Mgr	Test Group	Resource Mgr E	vent Log Alerts	Port Mgr	Messages			
	Status			Layer-3		L3 End	dps	· .	W	anLinks	
	Rpt Time	n fast	(1 s) 🔻	Go Test Man	ager all	- Select	All Sta	rt Stop	Quiesce	Clear	
	View	0 - 200			Go			Cr <u>e</u> ate <u>M</u> o	dify Del	ete	
					Cross Connects for	Selected Test Mar	nager				
Name	Туре	State	$Pkt\;Rx\:A\toB$	Pkt Rx A ← B	Rate $A \rightarrow B$	Rate A ← B	Rx Drop % A	Rx Drop % B D	rop Pkts A Dr	op Pkts B	Avg RTT
cdx-1	LF/UDP	Run	17,294	17,549	9,998,239	9,997,437	0	0	0	0	1
cdx-10	LF/UDP		17,377	17,716	9,997,632	9,996,340	0	0	0	0	0
cdx-2	LF/UDP		17,548		9,997,351	9,996,964		0	0	0	0
cdx-3	LF/UDP		17,633		9,997,891	9,996,964		0	0	0	0
cdx-4	LF/UDP		17,633		9,997,891	9,996,964		0	0	0	1
cdx-5	LF/UDP		17,718		9,997,947	9,992,326		0	0	0	1
cdx-6	LF/UDP		17,718		9,997,947	9,997.018			0	0	1
cdx-7	LF/UDP		17,718		9,997,947	9,997,018		0	0	0	1
cdx-8	LF/UDP		17,718		9,997,947	9,997.516			0	0	1
	LEUDD	0	17.710	17.044	0.007.047	0.007.010		-	0	0	- D
<											

For more information see Layer-3 Cross-Connects (FIRE)

D. Layer 4-7 tab:

The 'Layer 4-7' tab is currently where Layer-4 HTTP, HTTPS, FTP, FTPS, TFTP, SCP and SFTP endpoints are made. These are stateful protocols that will communicate properly with third-party servers. FTP, FTPS, TFTP, SCP and SFTP can upload and download, and the other protocols are only for downloading. The Layer 4-7 tab is used to manage Layer 4-7 endpoints.

						Stop Al	R	estart Manage	r	Refres	h HE
Layer-4	Generic	Test Mgr	Resource	Mgr Seria	d Spans PPF	P-Links Ev	ent Log Al	erts Port M	gr Messag	es	
Status	Layer-3	L3 En	dps Vo	IP/RTP	VoIP/RTP End	lps Arn	nageddon	WanLinks	Collision	n-Domains	File-I
View	0 - 200			▼ Go		Di	splay Ci	eate Mo	dify Bat	ch Modify	Delete
blama		Time	Cloture		er-4 Endpoints				Tu Pata(1)	Ru Data	Di Data/1
Name	EID	Type	Status	Total-URLs	ur-4 Endpoints	s for Selected Bytes-RD	Bytes-WR		Tx Rate(1)	Rx Rate	Rx Rate(1
o-lb-1	EID 1.1.18 1.1.0.62	Type L4/Gen L4/Gen	Status Stopped Run		URLs/s	Bytes-RD	Bytes-WR	Tx Rate	Tx Rate(1) 0	Rx Rate 0 50.211	
p-lb-1 pogle-0	1.1.18	L4/Gen	Stopped		URLs/s 0	Bytes-RD 0	Bytes-WR 0	Tx Rate	Tx Rate(1) 0 0	0	50,33
p-lb-1 oogle-0 oogle-0	1.1.18 1.1.0.62 1.1.0.63	L4/Gen L4/Gen	Stopped Run		URLs/s 0 0 1 0.143	Bytes-RD 0 307,084	Bytes-WR 0	Tx Rate 0 0	Tx Rate(1) 0 0 0	0 50,211	50,33 54,28
o-lb-1 oogle-0 oogle-0 oogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64 1.1.47.65	L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Run Uninitializ	Total-URLs 7 8 7	URLs/s 0.143 0.163 0.142 0.142	Bytes-RD 0 307,084 333,384 322,814 0	Bytes-WR 0 0 0 0 0	Tx Rate 0 0 0 0 0	0 0 0 0 0 0 0	0 50,211 54,177	50,33 54,28
p-lb-1 bogle-0 bogle-0 bogle-0 bogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64 1.1.47.65 1.1.48.66	L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Uninitializ Uninitializ	Total-URLs	URLs/s 0 0 0.143 0.163 0.163 0.142 0 0 0 0	Bytes-RD 0 307,084 333,384 322,814 0 0	Bytes-WR 0 0 0 0 0 0	Tx Rate 0 0 0 0 0 0	0 0 0 0 0	0 50,211 54,177 52,466 0 0	50,33 54,28
p-lb-1 oogle-0 oogle-0 oogle-0 oogle-0 oogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64 1.1.47.65	L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Run Uninitializ	Total-URLs 7 8 7	URLS/S 0 0 0.143 0.163 0.142 0 0 0 0 0 0	Bytes-RD 0 307,084 333,384 322,814 0	Bytes-WR 0 0 0 0 0	Tx Rate 0 0 0 0 0	0 0 0 0 0 0 0	0 50,211 54,177 52,466 0	Rx Rate(1 50,33 54,28 52,47

For more information see Layer 4-7

E. Resource Mgr tab:

The Resource Mgr tab displays information on all Resources discovered by the LANforge server and provides the ability to perform system functions on selected machines (one or more). The definition of a resource is a LANforge machine that belongs to a numbered realm. The realm 255 is always a stand-alone realm while the realm resource 1 is the manager. The Resource Mgr tab displays LANforge servers in the same realm. LANforge systems have to be manually numbered, two LANforge systems with the same resource ID will confuse the manager resource. Please visit the link below for more information on the Resource Mgr

						Stop Al	R	estart Manage	r	Refres	h HI
Layer-4	Generic	Test Mgr	Resource	Mgr Seria	Spans PPF	-Links Ev	ent Log 🛛 Al	erts Port M	gr Messag	jes	
Status	Layer-3	L3 En	dps Vo	IP/RTP	VoIP/RTP End	ps Arn	ageddon	WanLinks	Collisio	n-Domains	File-
Rpt Timer View	fast 0 - 200	(1 s)	Go Tes	t Manager al	I 🔽	Di	Select All	eate Mo		Quiesce tch Modify	Clear Delet
					r-4 Endpoints						
Name	EID	Type	Status	Laye Total-URLs	URLs/s	s for Selected Bytes-RD	Test Manage Bytes-WR		Tx Rate(1)	Rx Rate	Rx Rate(1
tp-lb-1	1.1.18	L4/Gen	Stopped		URLs/s 0	Bytes-RD 0	Bytes-WR 0	Tx Rate 0	Tx Rate(1) 0	0	
tp-lb-1 loogle-0	1.1.18	L4/Gen L4/Gen	Stopped Run		URLS/S 0 0.143	Bytes-RD 0 307,084	Bytes-WR 0		Tx Rate(1) 0	0	50,33
tp-lb-1 loogle-0 loogle-0	1.1.18 1.1.0.62 1.1.0.63	L4/Gen L4/Gen L4/Gen	Stopped Run Run		URLs/s 0 0.143 0.163	Bytes-RD 0 307,084 333,384	Bytes-WR 0 0 0	Tx Rate 0 0	Tx Rate(1) 0 0 0	0 50,211 54,177	50,33 54,28
tp-lb-1 loogle-0 loogle-0 loogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64	L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Run	Total-URLs 0 7 8 7	URLs/s 0 0.143 0.163 0.142	Bytes-RD 0 307,084	Bytes-WR 0 0 0 0	Tx Rate 0 0 0 0	Tx Rate(1) 0 0 0	0 50,211 54,177 52,466	50,33 54,28 52,47
tp-lb-1 loogle-0 loogle-0 loogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64 1.1.47.65	L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Run Uninitializ	Total-URLs 0 7 8 7 0	URLs/s 0 0.143 0.163 0.142 0	Bytes-RD 0 307,084 333,384 322,814 0	Bytes-WR 0 0 0 0 0	Tx Rate 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 50,211 54,177 52,466 0	50,33 54,28 52,47
tp-lb-1 poogle-0 poogle-0 poogle-0 poogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64 1.1.47.65 1.1.48.66	L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Uninitializ Uninitializ	Total-URLs 0 7 8 7 0 0 0	URLs/s 0 0.143 0.163 0.142 0 0 0	Bytes-RD 0 307,084 333,384 322,814 0 0	Bytes-WR 0 0 0 0 0 0	Tx Rate 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 50,211 54,177 52,466 0 0	Rx Rate(1) 50,33 54,28 52,47
tp-lb-1 poogle-0 poogle-0 poogle-0 poogle-0 poogle-0	1.1.18 1.1.0.62 1.1.0.63 1.1.0.64 1.1.47.65	L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen L4/Gen	Stopped Run Run Run Uninitializ	Total-URLs 0 7 8 7 0	URLs/S 0 0.143 0.163 0.142 0 0 0 0	Bytes-RD 0 307,084 333,384 322,814 0	Bytes-WR 0 0 0 0 0	Tx Rate 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 50,211 54,177 52,466 0	50,33 54,28 52,47

For more information see Resources (Data Generator Machines)

F. Messages, Warnings, Wifi-Messages Mgr tab:

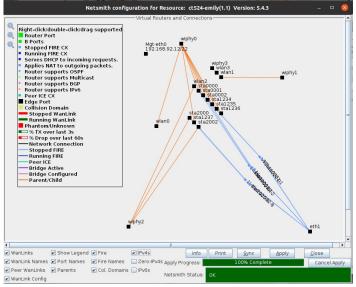
The Messages, Warnings and Wifi-Messages tab are all tabs that should be open at all times. All these tabs contain important information about the LANforge GUI Interface. The Messages tab displays detailed CLI command feedback from the LANforge Server. When scripting, command failures can be shown here. If <u>any</u> one of these 3 tabs are highlighted/have a yellow background in the tab bar, there is a new update in that yellowed tab.For information on any other tabs, besides the ones mentioned above, please visit the link below LANforge-GUI User Guide: Tab Display Preferences for further tab descriptions.

Save Clear Text Save Save Save Save Save Save Save Save	HE File-I
Chamber Jew Stop All Restart Manager Befresh Generic Resource Mgr DUT Profiles Traffic-Profiles Alerts Messages Warnings + WithRessages Status Port Mgr Layer-3 L3 Endps Layer 4-7 Ammageddon WanLiks VoiPiktP Endps F Victores to be LMMorge GU Manager That and Status VoiPiktP Endps F Save Clear Text The box 1513302 PF12020 WitConts in GU Manager That and Status F Save Clear Text Two Doct 3153302 PF12020 WitConts in Gu Manager That and Status F Save Clear Text Two Doct 315302 PF12020 WitConts in GLOBANTON Yvorimit SLI Compile on Web Doct 11011 FP11200 Save Clear Text Variants SLI Compile on Web Doct 11011 FP11200 Save Status Save Statu	
Generic Resource Mgr DUT Profiles Traffic-Profiles Alerts Messages Warnings + WithMessages Status Port Mgr Layer-3 L3 Endps Layer 4-7 Armageddon Wanniks VoiPATP Vo	
Status Port Mgr Layer-3 L3 Endps Layer 4-7 Armageddon WanLinks VolP/RTP VolP/RTP Endps F Stave Clear Tost <	File-I
Save Clear Text Vicenes to the Vicence	File-I
Save Clear Text Vicenes the full of Difference Links Vicenes to Links vicent Save Clear Text Vicenes the full candin inter allocates and information from the server. The Direct 31 5332 CPT 3333 Vicenes to Links vicent Vicenes the full candin inter allocates and the Vicenes to Links vicent Vicenes to Standard Vicenes Vicenes to Vicenes Vicenes to Vicenes Vi	
Valicense te ht Udlinge (01 Anapper) The Dec 23 51 326 20 27 3333 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for more information. The Dec 23 51 336 20 27 3030 VB/Cone te Udlinge, Enter help' for More te Udlinge, Enter help' for Help'	
The page will contain mixelineases information from the server. The Dec 29 15320 PT 020 PW income Net Hearts help for more information. The Dec 29 15320 PT 020 PW income Net Hearts help for more information. The Dec 29 15320 PT 020 PW income Net PW information PW information. The Dec 29 15320 PT 020 PW income Net PW information PW information. The Dec 29 15320 PT 020 PW income Net PW information PW information. The Dec 29 15320 PT 020 PW income Net PW information PW information. The Dec 29 15320 PT 020 PW income Net PW information PW	_
The Dec 20 13/30/1971200 witchme to LANGrgs. Enter help for more information. The Dec 20 13/30/2971200 The Dec 20 13/30/	
The Dec 20 153/02 PT 7200 The Dec 20 153/02 PT 7200 Version 5.4 Compiled on Wd Oc 2110118 PT 7200 Mersion 5.4 Compiled on Wd Oc 211018 PT 7200 Mersion 5.4 Compiled on Wd Oc 200 Mersion 5.4 Compiled on Wd Oc 200 Mersion 5.4 Compiled on Wd Oc 200 Mersion 5.4 Compiled on Wd Oc 200 Mersio	
Tao Dec 29 15 13 20 PT 2020 *********************************	
Arranse S.A.S. Complete an: Wed Dec 2 11:01:16 PET 2023 Windows S.A.S. Complete an: Wed Dec 2 11:01:16 PET 2023 Windows Linux 64bit Exercise Stahewise 2 Action 1 Perform 2006 A Complete State 54.6 Wanubias 6500 MJ.m. 16000 Wind-64m 1600 Windows 6400 Fine-CC 6400 Wanubias 6500 MJ.m. 16000 Wind-64m 1600 Windows 6400 Fine-CC 6400 Wanubias 6500 MJ.m. 16000 Wind-64m 1600 Secondary/Fir 64000 802110-CAses 6400 Fin33-0 Regel Andread State 54.0 Regel Andre	
By greater Do machine: Lance VEE 44.41.0.71-109/CE24.06.44.64.44.05 Licenses: Shows J. Cardel: 10 Ports: 10000 Active Ports: 66000 Files/CE 66000 Windows Licenses: Shows J. Cardel: 10 Ports: 10000 Active Ports: 66000 Files/CE 66000 BOILIAC Actives: 46400 File391.0 BOILIAC Actives: 64400 File391.0 BOILIAC ACTIVES: 84400 File391.0 BOILIAC A	
Mindform: Luna: 484t Luna: Lishuka: Cardi: 10 Ports: 10000 Active Parts: 68000 FVA-CC 68000 Vanulak: 68000 Min: 68000 Mindform: 6800 Windform: 68000 Ming 68 Mindform: 6800 Wanulak: 6800 Mindform: 6800 Mindform: 6800 Active All States 6600 Mindform: 6800 Wanulak: 6800 Mindform: 6800 Mindform: 6800 Active All States 6800 Windform: Landson form: 7800 Mindform: 7800 Active All States 6800 Windform: Landson form: 7800 Mindform: 7800 Active All States 6800 Windform: Landson form: 7800 Mindform: 7800 Active All States 6800 Windform: Landson form: 7800 Mindform: 7800 Active All States 7800	
An	
Vauluks Good W.J.m. GOOD W.J.G.M.GOOD W.J.G.M.GOOD W.J.S.M. GOO W.J.S.M. GOO W.J.S.G. GOO W.J.G. GO W.J.G.	
Vauluks Good W.J.m. GOOD W.J.G.M.GOOD W.J.G.M.GOOD W.J.S.M. GOO W.J.S.M. GOO W.J.S.G. GOO W.J.G. GO W.J.G.	
WarPeths: 460 / Amageden. 650 / V00F. 640 Secondary/#: £4000 Stallic Added: Secondary/#: £4000 Withm 0: V00F.0 Withm 0: V00F.0 Withm 0: V00F.0 December 2015 2000 Bio 2015 2010 December 2015 2010 2010 Bio 2015 2010 2010 Bio 2015 2010 2010 Bio 2015 2010 2010 2010 Bio 2015 2010 2010 2010 2010 Bio 2015 2010 2010 2010 2010 2010 2010 Bio 2015 2010 2010 2010 2010 2010 2010 2010	
AFGEN 0 AFGEN 12 Micron V Andone Windowski Status (Status V 1000 - 0 Wicron 0 Andone W Acidem C W Acide 0 W 1000 - 0 Micron 0 Andone Status (Status V 1000 - 0 Micron 0 Andone Status V 1000 - 0 Micron 0 Andone Status (Status V 1000 - 0 Micron 0 Andone Status V 1000 - 0 Micron 0 Andone Sta	
LANGForge Learness angles in 12:2 days. LANGForge Learness angles in 10:2 for FireCo D angles days. W.C.m. 01:4, dams 01:4, dams 01:4, dams 01:4, do 01:4, do 01:4, dams 01:4,	
Current Suize Prizz 1 (Inc. 5 Frie CAS & Annageddec 5 UND F 8 (Warth 9 Victories Victories Victories Anna (Lo 8 Victories 0 LANGrage Support and Safebare Victories Anna (Lo 8 Victories 0 LANGrage Support and Safebare Victories Anna (Lo 8 Victories 0 Rec 26 S35030 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3530 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3530 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3530 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3530 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3530 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3530 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 S3500 F71200 - SASEI 23 Can (Mana, Clanter) The Dec 35 Can (Mana, Clanter) The Dec 35 Can (Mana, Clanter) The Dec 35 Can (Man	
W.L.m. 0 WL.dm. V W.OBH 0 W.U.S.m. V W.L.G. 0 W.U.G.0 G Wardhard 0 2012.1.4.6.4.0.0 Wardhard 0 2012.1.4.6.4.0.0 Wardhard 0 2012.1.4.1.4.0.0.0 Wardhard 0 2012.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0	
Vandenko 2021/26. Audios: 0 Ministringe Support and Galance Upperdes expire in: 22 days. Two Dec 25 5530.0 PT 2020. Two Dec 25 5530.0 PT 2020. Solit: 252 Cond. Takow, Cleaner' Two Dec 25 5530.0 PT 2020. Solit: 253 Cond. Takow, Cleaner' Two Dec 25 5530.0 PT 2020. Solit: 254 Cond. Takow, Cleaner' Dec 25 5530.0 PT 2020. Solit: 254 Cond. Takow, Cleaner'	
The Dec 25 15:30 CF 97 2025 The Dec 25 15:30 CF 97 2025 SHLT 122 Cond 'thou, clients' The Dec 25 15:30 CF 97 2025 SHLT 122 Cond 'thou The Dec 25 15:30 CF 97 2025 SHLT 122 Cond 'thou The Dec 25 15:30 CF 7020 SHLT 120 Cond 'thou The Dec 25 15:30 CF 7020 SHLT 120 Cond 'thou SHLT 120 CF 97 2020 SHLT 120 SHLT 120 SHLT 120 CF 120 SHLT 120 S	
Tue Dec 29 15:3102 PF1 2020 >> >>85:1: 22: Cmd: Whong Clients' Tue Dec 29 15:3102 PF1 2020 >>85:1: 25:C md: Whong Tue Dec 29 15:3102 PF1 2020 >>85:1: 35:40 Cmd: 'hong dis' Tue Dec 29 15:3102 PF1 2020 >>85:1: 35:40 Cmd: 'hong dis'	
Tue Dec 29 15:33:02 PST 2020: >>RSUT: 192 Cmd: \nho" Tue Dec 29 15:33:02 PST 2020: >>RSUT: 542 Cmd: \nhow_dbs' Tue Dec 29 15:33:02 PST 2020: >>RSUT: 0 Cmd: Tienen 'NA' Yes'	
Tue Dec 29 15:33:02 PST 2020: >>RSLT: 3549 Cmd: 'show_dbs' Tue Dec 29 15:33:02 PST 2020: >>RSLT: 0 Cmd: 'license' NA' Ves'	
Tue Dec 29 15:33:02 PST 2020: >>RSLT: 0 Cmd: 'license' NA' Ves'	
Tue Dec 29 15:33:02 PST 2020: >>RSLT: 0 Cmd: 'show w' 'ALL' 'ALL'	
Tue Dec 29 15:33:02 PST 2020: >>RSLT: 0 Cmd: 'show_vrcx' 'ALL' 'ALL'	
Tue Dec 29 15:33:02 PST 2020: >> RSLT: 252 Cmd: 'show_clients' Tue Dec 29 16:33:02 PST 2020: >> RSLT: 192 Cmd: 'show_clients'	
Tue Dec 29 15:33:02 PST 2020: >> RSLT: 192 Cmd: Who' Tue Dec 29 15:33:02 PST 2020: >> RSLT: 0 Cmd: Who alerts'	
The De c 2 is 33:02 PT 2020. New Setting: ISBNF: 1	
Tue Dec 29 15:33:02 PST 2020: Push_Endp_Rpts flag set to: 0	
Tue Dec 29 15:33:02 PST 2020: Push_All_Rpts flag set to: 0	
Tue Dec 29 15:37:57 PST 2020: Adding new endpoint.	
Tue Dec 29 15:37:57 P97 2020: Setting Rate: VTsta0000-0-A min: 256000 max: 0 Tue Dec 29 15:37:57 P97 2020: Setting Payload Size: Shefi I Card: 1 Port: 65535 Endpoint: 142 Type: LANFORGE UDP min: -1 max: 0 Checksum: NA	
The Dec 25 15:37:57 PT 2020. Setting Payload Size; Shen: 1 Lard: 1 Port: 65555 Endpoint: 142 Type: LowPorde_ODF min: 1 max: 0 Checksum; NH The Dec 25 15:37:57 PT 2020. Endpoint values have been set.	
Tue Dec 29 15:37:57 PST 2020: Adding new endpoint.	
T	
Input:	
oaged in to: localhost:4002 as: Admin	

For more information see Tab Display Preferences

G. Using Netsmith tab:

In the LANforge GUI, on the Status page there is a small button named Netsmith. It is a tool used to help visualize the relationships of ports and cross connects defined in the resource you are viewing. There is a separate Netsmith view for each LANforge resource in your realm. There are several ways to edit the GUI objects in Netsmith, display the different up-to-date connections in the GUI, and what is shown in Netsmith. Please visit the link below to understand how to use Netsmith in greater detail.



For more information see Netsmith: Virtual Network Configurator

4. Station Creation:

Please visit Step 1 of the following cookbook below to learn how to Create a Station in the LANforge-GUI. Please visit the link at the bottom of this section on how to script a station in the GUI.

A. Searching for Active SSIDS & Connecting to a Particular SSID:

Often times, there may be an active network around, but the LANforge GUI does not have the network registered as "able to be connected to". To allow this network to be recognized, one must **scan** in the GUI object's settings to make sure that the object sees this network.

See below for an example:

I. Double-click or select Modify on a station in the Port Mgr to pop up Configure Settings window.

	LANforg	e Manager	Ve	rsion(5.4.3	+0	li-sock			•	0
ontrol <u>B</u> eporting Wind <u>o</u> ws info <u>T</u> e	sts									
	Chamber View	1	5	top All	Re	estart Manager		Refres	n	HELF
File-IO Generic Test Group Res Status Port Mor Laver-3	ource Mgr VAP	Stations DU Laver 4-7		Profiles Ale Armageddon	rts	Warnings + WanLinks	Wifi-Messa Attenuator		-Gener	retor
	Sniff Packets	Down	1	Clear Coun	ters	Reset Port	Delete			
Rpt Timer: medium (8 s) 💌	Apply	VRF	I	Display		Create	Modify	Bato	h Modif	ly

II. However, the GUI isn't registering that as a proper network because selecting Display Scan, Scan, and Sync in the Configure Settings shows no networks are found and discovered in the GUI.



III. Type in the desired SSID, Key/Phrase, and select the appropriate Security to be used (WPA/WPA2/WPA3... etc) located within the WiFi Settings panel (shown below). Select Apply. Apply will trigger the LANforge GUI to start searching for currentlyactive SSIDs.

		WN LINK-DOWN GRO NON rt Type: WIFI-STA Parent:			
		Port Configurab	les		
andard Configurat	ion Advanced	Configuration Mise	Configuration	Corruptions	Custom V
Enable		General Ir	terface Settings		
Set MAC	Down	Aux-Mgt	DHCP Hostname	: None	-
Set TX Q Len	DHCP-IPv6	DHCP Release	DHCP Vendor ID:	None	-
Set Offload	DHCP-IPv4	Secondary-IPs	DHCP Client ID:	None	-
Set PROMISC	DNS Servers:	BLANK	Peer IP:	NA	
	IP Address:	0.0.0.0	Global IPv6:	AUTO	
Services	IP Mask:	0.0.0.0	Link IPv6:	AUTO	
НТТР	Gateway IP:	0.0.0.0	IPv6 GW:	AUTO	
FTP	Alias:		MTU:	1500	
DNS	MAC Addr:	00:0e:8e:4e:3b:47	TX Q Len	1000	
RADIUS	Rpt Timer:	medium (8 s) 👻	WiFi Bridge:	NONE	-
IPsec-Upstream	IPSec GW:	0.0.0.0	IPSec Password:		
	IPSec Local ID.:		IPSec Remote ID		
Low Level		Wie	i Settings		
PROMISC	SSID: je	dway-wpa2-x2048-5-		AULT	
TSO Enabled		lway-wpa2-x2048-5-1			
UFO Enabled	Freq/Channel: 5			efault	
GS0 Enabled			The con	erauce	
LRO Enabled		2 WPA3 OSE	-		
GRO Enabled	UISADIE HT4	0 🔲 Enable VHT160	Disable SGI		

IV. Then, select Display Scan in the bottom bar, as highlighted in the picture above. Something similar to the Window in the picture below will pop up. Then click on Scan (circled below) and Sync. Now, the most recent active networks should be scanned and displayed in a similar window to below by the GUI. The example below indicates that the radio (wiphy0) has now found current, active networks. Also, the far right corner of the table displays the age of the networks, so if the Age is too old after the recent scanning, it might be time to restart the network or pick a new network.

Note: If there are no scan results, the radio is probably set to a specific channel. The radio channel configuration may need to be changed or the object must be created on a different radio.

	sta0	(ct524-genia)		r Version(5.4. Settings	2)		×	C	$\circ \circ \circ$
0			wij	phy0 Scan Resu	lts			-	
SSID	Channel	Info	Auth	BSS	Signal	Frequency	Beacon	Age	
CBCI-318F-2.4	1	3x3 MM0	WPA2	74:85:2a:4f:0c:38	-78.0	2412	100	46.15 \$	
CBCI-CE22-2.4	6	3x3 MM0	WPA WPA2	20:25:64:61:af:70	-75.0	2437	100	5.32 s	
CBCI-CE22-5	161-	3x3 MIM0	WPA WPA2	20:25:64:61:10:e0	-77.0	5805	100	2.79 s	
OSEN-AP	157+	3x3 MCS 0-9 AC	OSEN	04:10:21:df:42:df	-84.0	5785	240	43.27 s	
Silv-Wifi	11	2x2 MM0	WPA WPA2	24: f5:a2:6a:72:47	-83.0	2462	100	13.19 s	
Silv-Wifi-quest	11	2x2 MM0	Open	24: f5:a2:6a:72:49	-78.0	2462	100	5.09 s	
Success11	11	3x3 MM0	WPA WPA2	c0:7c:d1:0c:bd:f8	-79.0	2462	100	26.95 s	
XEINITY	161-	3x3 MM0	WPA2	20:25:64:61:10:e3	-78.0	5805	100	2.81 s	
[BLANK]	1	3x3 MM0	WPA2	f2:9f:c2:71:fa:8a	-55.0	2412	100	5.74 s	
[BLANK]	60	4x4 MCS 0-9 AC	WPA2	f2:9f:c2:69:55:12	-62.0	5300	100	4.49 s	
[BLANK]	6	4x4 MCS 0-9 AC	WPA2	f0:9f:c2:69:55:13	-58.0	2437	100	13.48 s	
bridged-APft	36+	3x3 MCS 0-9 AC	WPA2	04:10:21:2d:29:44	-61.0	5180	240	5.05 s	
dlink-dir878-perf	9	4x4 MCS 0-9 AC	Open	74:da:da:6d:73:78	-71.0	2452	100	5.22 s	
dlink-dir878-perf-5g	48	802.11a	WPA WPA2	74:da:da:6d:73:79	-81.0	5240	100	4.86 s	
jedway-abcd	36+	3x3 MM0	WPA2	00:0e:8e:e7:7b:07	-71.0	5180	240	5.05 s	
jedway-open-1	1+	3x3 MM0	Open	00:0e:8e:78:e1:76	-71.0	2412	240	5.59 s	
jedway-open-149	149+	3x3 MM0	Open	00:0e:8e:de:d4:e6	-75.0	5745	240	3.01 s	
jedway-r8000-11	11	3x3 MM0	WPA2	10:0c:6b:63:1c:9e	-41.0	2462	200	5.11 s	
jedway-r8000-153	153-	3x3 MCS 0-9 AC	WPA2	10:0c:6b:63:1c:9f	-46.0	5765	200	2.95 s	
jedway-r8000-36	36+	3x3 MCS 0-9 AC	Open	10:0c:6b:63:1c:9d	-60.0	5180	200	5.05 s	
jedway-wep-48	48	802.11a	WEP	04:f0:21:4d:9a:21	-48.0	5240	240	4.86 s	
jedway-wpa-1	1+	3x3 MM0	WPA	00:0e:8e:b9:b9:76	-71.0	2412	240	5.67 s	
jedway-wpa2-x128	44+	3x3 MCS 0-9 AC	WPA2	04:10:21:3d:66:41	-55.0	5220	240	4.93 s	
jedway-wpa2-x204	1	3x3 MM0	WPA2	00:0e:8e:05:70:76	-72.0	2412	240	5.75 s	
jedway-wpa2-x204	149	3x3 MM0	WPA2	00:0e:8e:ff:86:e6	-74.0	5745	240	3.01 s	
jedway-wpa2-x204	161-	3x3 MM0	WPA2	00:0e:8e:b3:68:e7	-75.0	5805	240	2.81 s	
jedway-wpa2-x204	157	3x3 MM0	WPA2	00:0e:8e:7b:df:9b	-73.0	5785	240	2.88 s	
jedway-wpa2-x204	36+	3x3 MM0	WPA2	00:0e:8e:e4:fa:07	-71.0	5180	240	5.05 s	
jedway-wpa2-x64-3-1	48	3x3 MCS 0-9 AC	WPA2	04:10:21:d3:8d:21	-47.0	5240	240	4.86 s	
jedway-wpa3-1	1+		WPA2 WPA3	00:0e:8e:22:e8:76	-71.0	2412	240	5.75 s	
jedway-wpa3-44	44+	3x3 MCS 0-9 AC	WPA2 WPA3	04:f0:21:0a:8b:41	-54.0	5220	240	4.93 s	
muffin	48-	3x3 MCS 0-9 AC	Open	04:10:21:18:92:21	-47.0	5240	240	4.86 s	
ubnt-hd-ent	60	Ax4 MCS 0-9 AC	WPA2	f0:9f:c2:69:55:12	62.0	5300	100	54.5	
				Paus	ie I	Scan	1 (3	ync)	Close
	_				-				
						-			
		-							

V. . Now, close the two windows opened previously by selecting Close. Go back to the Port Mgr tab and the desired object to be connected should be connected to that SSID. In Wifi-Messages, there should have also been a message saying that sta0 and wiphy0 are scanning for network SSIDs. This is another indication of the LANforge scanning software retrieving local SSIDs. LANforge now concludes that it can connect to the SSID by acquiring an AP and IP in the Port Mgr (see circled below).

	sta0	(ct524-genia)	Configure	Settings			(X)		
			wi	phy0 Scan Resu	Its			-	
SSID	Channel	Info	Auth	BSS	Signal	Frequency	Beacon	Age	
CBCI-318F-2.4	1	3x3 MM0	WPA2	74:85:2a:4f:0c:38	-78.0		100		
CBCI-CE22-2.4	6	3x3 MM0	WPA WPA2	20:25:64:61:af:70	-75.0		100	5.32 s	
C8CI-CE22-5	161-	3x3 MM0	WPA WPA2	20:25:64:61:10:e0	-77.0		100	2.79 s	
OSEN-AP	157+	3x3 MCS 0-9 AC	OSEN	04:10:21:df:42:df	-84.0		240	43.27 s	
Silv-Wifi	11	2x2 MM0		24:15:a2:6a:72:47	-83.0		100	13.19 s	
Silv-Wifi-guest	11	2x2 MM0	Open	24:15:a2:6a:72:49	-78.0		100	5.09 s	
Success11	11	3x3 MM0		c0:7c:d1:0c:bd:f8	-79.0		100	26.95 s	
XEINITY	161-	3x3 MM0	WPA2	20:25:64:61:10:e3	-78.0		100	2.81 s	
(BLANK)	1	3x3 MM0	WPA2	f2:9f:c2:71:fa:8a	-55.0	2412	100	5.74 s	
[BLANK]	60	4x4 MCS 0-9 AC	WPA2	f2:9f:c2:69:55:12	-62.0	5300	100	4.49 s	
[BLANK]	6	4x4 MCS 0-9 AC	WPA2	f0:9f:c2:69:55:13	-58.0	2437	100	13.48 s	
bridged-APft	36+	3x3 MCS 0-9 AC	WPA2	04:10:21:2d:29:44	-61.0	5180	240	5.05 s	
dlink-dir878-perf	9	4x4 MCS 0-9 AC	Open	74:da:da:6d:73:78	-71.0	2452	100	5.22 s	
dlink-dir878-perf-5g	48	802.11a	WPA WPA2	74:da:da:6d:73:79	-81.0	5240	100	4.86 s	
jedway-abcd	36+	3x3 MM0	WPA2	00:0e:8e:e7:7b:07	-71.0		240	5.05 s	
jedway-open-1	1+	3x3 MM0	Open	00:0e:8e:78:e1:76	-71.0		240	5.59 s	
jedway-open-149	149+	3x3 MM0	Open	00:0e:8e:de:d4:e6	-75.0		240	3.01 s	
jedway-r8000-11	11	3x3 MM0	WPA2	10:0c:6b:63:1c:9e	-41.0		200	5.11 s	
jedway-r8000-153	153-	3x3 MCS 0-9 AC	WPA2	10:0c:6b:63:1c:9f	-46.0		200	2.95 s	
jedway-r8000-36	36+	3x3 MCS 0-9 AC	Open	10:0c:6b:63:1c:9d	-60.0		200	5.05 s	
jedway-wep-48	48	802.11a	WEP	04:f0:21:4d:9a:21	-48.0		240	4.86 s	
jedway-wpa-1	1+	3x3 MM0	WPA	00:0e:8e:b9:b9:76	-71.0		240	5.67 s	
jedway-wpa2-x128	44+	3x3 MCS 0-9 AC	WPA2	04:10:21:3d:66:41	-55.0		240	4.93 s	
jedway-wpa2-x204	1	3x3 MM0	WPA2	00:0e:8e:06:70:76	-72.0		240	5.75 s	
jedway-wpa2-x204	149	3x3 MM0	WPA2	00:0e:8e:ff:86:e6	-74.0		240	3.01 s	
jedway-wpa2-x204	161-	3x3 MM0	WPA2	00:0e:8e:b3:68:e7	-75.0		240	2.81 s	
jedway-wpa2-x204	157	3x3 MM0	WPA2	00:0e:8e:7b:df:9b	-73.0		240	2.88 s	
jedway-wpa2-x204	36+	3x3 MM0	WPA2	00:0e:8e:e4:fa:07	-71.0		240	5.05 s	
jedway-wpa2-x64-3-1	48	3x3 MCS 0-9 AC	WPA2	04:10:21:d3:8d:21	-47.0		240	4.86 s	
jedway-wpa3-1	1+		WPA2 WPA3	00:0e:8e:22:e8:76	-71.0		240	5.75 s	
jedway-wpa3-44	44+	3x3 MCS 0-9 AC		04:f0:21:0a:8b:41	-54.0		240	4.93 s	
muffin	48-	3x3 MCS 0-9 AC	Open	04:10:21:18:92:21	-47.0		240	4.86 s	
ubnt-hd-ent	60	4x4 MCS 0-9 AC	WPA2	f0:9f:c2:69:55:12	62.0		100	54.5	
				Pau Pau	10	Scan) (2	mc)	Close
					-				

For more information see Station Creation : Step 1

Creating a MAC-VLAN on the LANforge-GUI is done in the Port Mgr.

Please visit Step 3 of the following cookbook on how to create a MAC-VLAN from the GUI. The following link will inform how to program the GUI to create a MAC-VLAN

For more information see Creating a MAC-VLAN in the GUI(Step 3)

For more information see Scripting a MAC-VLAN in the GUI

6. Bridge Creation:

Creating a Bridge on the LANforge-GUI is done in the Port Mgr.

Please visit Step 2 of the following cookbook on how to create a Bridge in Netsmith.

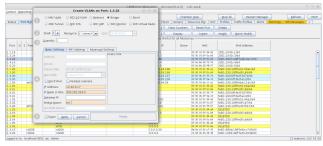
For more information see Creating a Bridge in Netsmith (Step 2)

A. Create a bridge in Port Mgr:

A. Click on the Port Mgr tab and Create in the top right corner.

ontrol	Be	por	ting Wind <u>o</u> w	ins into	Tests								
					Chambe	er <u>V</u> lew	S	top All	Resta	irt Manag	er	Refresh	HE
VolP/R	TP F	ndr	s File-IO	Generic	Resource	Mar DUT Profiles	77	Traffic-Profiles	Alerts	Warnin	nas + Wifi-Me	essages	
Stat			Port Mgr		Layer-3		Lay			geddon	WanLinks		P/RTP
Dis	p:]	92	168.92.198:	1	Sniff Packet	ts 🕑 Down	1	Clear Counter	s	Reset Po	ort Delete		
Rot	Tin	en	medium (8	s) 🔻	Apply	VRF	I	Display	1	Create	Modify	Batch Mo	dify
						All Ethernet Interfaces			_\		1.0000	Tarrenter	
						All currenter internaces	111	orts) for all Nesol	urces.	-			
Port	ø	I	Parent Dev	Channel	Alias	SSID		AP		Mode	IP	Noise	
1.1.00					eth0		-		-		192.168.92.198		00:3
1.1.01					ethl						0.0.0.0		00:3
1.1.02					eth2						10.0.0.103		00:3
1.1.03					eth3						0.0.0.0		00:3
1.1.04					eth4						0.0.0.0		0013
1.1.05					eth5						0.0.0.0		00:3
1.1.06					wiphy0)2.11ab			04:1
1.1.07					wiphy1					02.11an			04:1
1.1.08				0	wlan0			Not-Associated			0.0.0.0	-1 dBm	04:1
1.1.09			wiphy1	0	wlan1			Not-Associated	AL		0.0.0.0	-1 dBm	04:1
1.1.12					br1000						10.40.0.1		00:3
1.1.15				48	sta0000			04:F0:21:F8:92:21			10.40.0.12	-102 dBm	04:1
1.1.16				0	sta0001	jedway-wpa2-J					0.0.0.0	-1 dBm	04:1
1.1.17				0	vap201	jedway-roam	-36		AL		0.0.0.0	-1 dBm	04:1
1.1.18			eth0		eth0#0						192.168.94.56		00:3
1.3.00					eth0						192.168.93.166		00:3
1.3.01					eth1				_		0.0.0.0		00:3
1.3.02					eth2				_		10.40.0.6	-	00:3
1.3.03					eth3 eth4				_		10.40.0.5	-	00:3
1.3.04					eth5						0.0.0.0		00:3
1.3.05				48	wiphy0				0/	12.11ab			00:3
1.3.00				48	wiphy0 wiphy1)2.11ab)2.11ab			04:1
4	-	-		-4-4	wipiny 1		_		80	22.118D	0.0.0.0	_	04.1

B. After a new window pops up, Select Bridge in Step 1 of the new window. In Step 2, select the Shelf and Resource the bridge should use (from the drop down menus in each slot). Step 3, select the Quantity of the bridges to be created. In Step 4, under the Basic Settings tab, check the box if the bridge should be enabling DHCP-IPv4. If DHCP-IPv4 isn't enabled, give the bridge an IP Address and IP Mask. Lastly, give the bridge a name, Click Apply and Cancel. The bridge is now in the Port Mgr.



B. Adding a port to an existing bridge in Port Mgr:

	porting Win	dows Info	Tests									
			Chan	nber ⊻iew		Stop	All	Re	start Manag	er	Be	fresh Hi
RF-Generat	or File-IO	Generic	Test Grou	Resour	ce Mar VA	P Statio	ns DUT	Pro	files Alert	Warnin	as +	
Status	Port M		Layer-3	L3 End		aver 4-			geddon	WanLi		Attenuators
Disp: 1	92.168.92.1	-	Sniff Pac			0	lear Coun		Reset Po	ut De	lete	
	_						arear courr	icer 5				
Rpt Time	er: medium	(8 s) 🔻	Apply		VRF	I	Display		Create	M	dify	Batch Modify
	-			-All Ethen	net Interface	s (Port	s) for all Re	esourc	es.			
Port	Phantom Ø	Down I	IP	Alias	Parent Dev	AF	Cha	annel	Mode	SSID	MAC	
.1.0			192.168	eth0							0c:c4:7a:	
.1.1			10.40.11	eth1							0c:c4:7a:	
.1.10				wiphy3			0		802.11an		00:19:70:	
.1.2			10.40.9.1		wiphy0	00:0E:8	E: 157		802.11an	jedway-w		
				wiphy0			0		802.11ab			
1.8			0.0.0.0	wiphy2			0		802.11ab		04:f0:21:	
1.1.3 1.1.4 1.1.6 1.1.8			10.40.0.17 0.0.0.0				0 0 0		802.11ab 802.11ab 802.11ab		00:0e:8e: a6:50:b2: 00:0e:8e: 04:f0:21:	

A. To add a port, double click on the bridge you created or click once on the bridge in Port Mgr and select Modify. A window Configure Settings should pop up. At the bottom of the window, there is a small section that allows addition of ports.

		br17 (c	t524-genia) Con	figure Settings		\odot	
			Port Status Inform LINK-DOWN PROBE-ERROI Port Type: Bridge Drive	R TSO GSO GRO			
Enable Set MAC Set MC Set MU Set MU Set Bridge Info Set Bridge Info Set Dridge Char HTTP PTP PTP PTP PTP PTP PTP PT	Down DHCP-IPVg DHCP-IPVg DHCP-IPVg DHSsevers: IP Address: IP Mask: Gateway IP: Aljus: MAC Addh Rpt Timer. IPSec Own Bridd Configured Pc	General In Aux-Mgt P DHCP Release Secondary-IPs BLANK 10.40.0.17 255.255.254.0 0.0.00 00:00:00:00:00:00 aedium (8 s) 00:00 ge information	Port Configural DHCP Hostname: DHCP Vendor ID: DHCP Client ID: DHCP Client ID: DHCP Client ID: IDHCP Client ID: IDHCP Client ID: IDHCP Client ID: IDHCP Client ID: IVS Quen WFI Bridge: IPSec Remote ID: Bern	None None None NA Autro Autro Loso NONE V	Spanning-Tree Aging Time: Bridge Priority: Max Age: Hello Time: Forwarding Delay:	300 32768 20 2 2 15	
	Print	Display Pr	obe Sync	Apply <u>O</u> K	Cancel		

B. In the text box under the Add Ports button (circled below), type in the port name (ex: vap123, eth1, sta000) intended to be added to the bridge. In this example, eth1 to be added to br17.

		br17 (c	t524-genia) Con	figure Settings			• • •
		Current: Driver Info:	Port Status Inform LINK-UP PROBE-ERROR TS Port Type: Bridge Drive	SO GSO GRO			
			Port Configural	hles			
Enable		General I	nterface Settings	5100			
Set MAC	Down	Aux-Mat	DHCP Hostname:	None	Spanning-Tree		_
Set TX Q Len	DHCP-IPv6	DHCP Release	DHCP Vendor ID:		Aging Time:	300	-
Set MTU			-		Bridge Priority:	32768	•
Set Offload Set Bridge Info	DHCP-IPv4	Secondary-IPs	DHCP Client ID:	None	Max Age:	20	-
Set Bridge Info	DNS Servers:	BLANK	Peer IP:	NA	Hello Time:	2	-
	IP Address:	10.40.0.17	Global IPv6:	AUTO	Forwarding Delay:	15	-
	IP Mask:	255.255.254.0	Link IPv6:	AUTO			
	Gateway IP:	0.0.0.0	IPv6 GW:	AUTO			
	Alias:		MTU:	1500			
	MAC Addr:	de:0d:83:75:d0:3c	TX Q Len	1000			
	Rpt Timer:	medium (8 s) 🔻	WiFi Bridge:	NONE			
Services	IPSec GW:		IPSec Password:				
HTTP	IPSec Local ID.		IPSec Remote ID.:				
FTP		dge Information	Rem	ove Ports			
DNS RADIUS	Configured P	orts Current Port					
IPSEC-Client			Add	Ports			
IPsec-Upstream			eth 1				
	Print	Display Pr	obe <u>Sync</u>	Apply QK	Cancel		
	Eont	Disbiak El	ope <u>S</u> hic	Abbia DK	Saucei		

C. Select Add Ports (circled). This button will now categorize eth1 as a Configured Port. Then, select Apply and Sync to now see eth1 also be listed under Current Ports. Lastly, click OK to close the window. If the port inputted into the text box does not move to the Current Ports category after selecting Sync, this may mean that the port is already in a configuration that prevents it from being in a bridge (i.e. it may already be in a bridge... etc). To learn how to script a bridge in the GUI, please visit the link below.

For more information see Scripting the GUI to create a Bridge (Step 6)

7. Virtual Creation (VAP):

Please visit **Step 1** of the following cookbook to learn how to create a Virtual AP in the GUI. For more information see Scripting the GUI to create a VAP (Step 7)

8. Monitor Creation:

Please visit **Step 1** of the following cookbook to learn how to create a Monitor in the GUI. For more information see Scripting the GUI to create a Monitor

9. Layer 3 Creation:

Layer-3 Cross-Connects represent a stream of data flowing through the system under test. A Cross-Connect (CX) is composed of two Endpoints, each of which is associated with a particular Port (physical or virtual interface). The Layer-3 tab displays connections 0-200 by default.

Separated below are important sections to getting to know the Layer 3 tab:

For more information see How to Create and Modify Cross-Connects & Cross-Connect Information

For more information see Interpreting the Layer-3 Endps tab: Layer-3 Cross Connect Endpoints & Batch-Creating Cross-Connects

For more information see Scripting a Layer-3 Cross Connect (Step 8)

10. Layer 4-7 Traffic Generation:

The Layer 4-7 traffic is supposed to emulate curl commands. Endpoints can be created with the following protocols: HTTP, HTTPS, FTP, STP, SCP and SFTP. These are stateful protocols that will communicate properly with third-party servers. FTP, FTPS, TFTP, SCP and SFTP can upload and download, and the other protocols are only for downloading. The Layer 4-7 tab is used to manage Layer 4-7 endpoints.

Separated below are important sections to getting to know the Layer 4-7 tab:

For more information see Creating and Modifying Layer 4-7 Endpoints, L4 Endpoint Information, Batch-Create Layer 4-7 Endpoints

For more information see Layer 4-7 Endpoint Display

For more information see Setting up a Simple HTTP Get/Download in the GUI

For more information see Scripting the GUI to create Layer 4-7 traffic (Step 9)

Querying the LANforge GUI for JSON Data

Goal: The LANforge GUI now has an embedded webserver and a headless mode of operation. Read on for how to configure and query the client for JSON formatted data.

Updated 2019-11-21:New features in 5.4.1.

 Some of the CLI API parameter names have changed. Notably: nc_show_ports flags changed to probe_flags. Be aware that older scripts might break on upgrade.

Updated 2018-07-24:New features in 5.3.8.

The LANforge GUI (as of release 5.3.6) can be configured to start an embedded web server that can provide data about ports and layer-3 connections. This service can be queried with with any browser or AJAX connection. We're going to increasingly refer to it as the LANforge **client**. This feature provides these benefits:

- More rapid polling: using CLI scripts to poll ports on the LANforge manager can add stress and contention to the LANforge manager; polling the GUI will not tax your test scenario.
- Expanded array of data: the views found in the GUI, like Port Mgr and Layer-3 tabs, contain synthesized data columns not available through the CLI scripting API. Most of these columns can be returned in JSON format.
- Reduced effort when integrating with third party test libraries: many other testing libraries expect JSON formatted input.
- Web socket delivery of event data allows real-time reporting of interface changes and station scan results. This is also a channel for querying additional diagnostic data.
- There is a /help web page that allows you to build POST commands.
- A headless -daemon mode that will run the client without any GUI windows. This requires much less memory and has been queried for weeks at a time without crashing or memory leaks.

Present and potential drawbacks of the JSON feature:

- Actively being developed: the JSON views/schema of the objects is at a demonstation state. URLs and JSON structures have changed in 5.3.8.
- Now no longer possible to create Groovy plugins to add JSON features if you
 want to use the headless mode. JSON Features are compiled into the
 LANforge GUI from Java sources.
- In 5.3.8 we have limited the view of ports, have added URLs to post direct CLI
 commands, and have applied HTML application/x-www-form-urlencoded
 form posting submissions in name/value pairs. There is no multipart/formdata JSON submission at this time.

Client Settings

The LANforge GUI is started using a script (lfclient.bash or lfclient.bat). From a terminal, we call that script with the -httpd switch. By default the GUI will listen on port 8080:

<pre>\$ cd /home/lanforge \$./lfclient.bash -httpd</pre>
You can specify the port to listen on:
<pre>\$./lfclient.bash -httpd 3210</pre>
You can run the client headless with the -daemon switch as well:
<pre>\$./lfclient.bash -httpd -daemon</pre>

There is a setting in the 5.3.8 Control \rightarrow Preferences menu for setting a minimized mode and the HTTP port number as well.



LANF	forge-GUI Preferences
General LANforge Manager Tabs	
Confirmations	
Confirm Exit	Confirm Netsmith Sync
Auto-Submit	Skip delete warning in Reporting Manager
Skip delete warning in Netsmith	
Display	
🗹 Anti-Alias lines in Netsmith	Invert RX-Signal X Axis
Show Commas	Enable system look and feel (requires restart)
Display Graph Duration: 2 Minutes (2 min)	▼ Dynamic Report Duration 30 Minutes (30 min) ▼
Set Extended Title	
Features	
🗹 Poll Mgr	
Enable HTTP Service	HTTP Port 8080
🔲 Run GUI Minimized	User ID
Enable hotkey for entity deletion (A/t-t)	
Tool Tips	
🖌 Show Tooltips	Duration(ms) 10000 V Delay(ms) 250 V
Reset	t Config OK

Making Queries

From the terminal we can query the port to find a basic message from the GUI:

\$ curl -sq http://localhost:8080/

This first page (\vec{J}) will give you a JSON list of the resource URLs available. Most URLs will provide JSON as their default content type. Notably, /help defaults to HTML.

← → ♂ û	① localhost:8080 □2050 ···· ▼ ☆ Q Search III ● ■ Ξ
JSON Raw Data Hea	ders
Save Copy	▼ Fliter JSON
handler:	"candela.lamforge.HttpStatus"
uri:	
text:	"These urls are presently available"
▼ mappings:	
	"(This page) Provides status and configuration."
<pre>/cli-form/:cnd:</pre>	"Post key-value command values to a specifi command, eg: -H 'Content-type: application/x-www-form-urlencoded' -X POSTdata 'shelf=1& resource=1&port=1' http://chollad:8089/cli-form/nc show port"
▼/cli:	"Post literal (LI commands, eg: -H 'Content-type: application/x-www-form-urlencoded' -X POSTdata-urlencode 'cnd=nc_show_port 1 1 1' http://chollad.8888/cli"
<pre>/cli-json/:cnd:</pre>	"Post json object command values, eg: -H 'Content-type: application/json' -H 'Accept: application/json' -X POST '{\'alias\":\"cx0001 \",\"state\":\"RUNNING\")'http://cholla4:8889/cli-json/set_cx_state"
/alerts/:	"List alerts that have been reported"
<pre>▼/resource/:shelf_id:</pre>	"Lists ports on resource 1.2, eg: /resource/1/2/, see /port/:shelf_id/:resource_id/list"
▼/misc/:	"(This page) Provides access to less formatted data and other hacks, see /misc/help."

By default, most URLs will treat a default Accept: */* header as text/html. Compare the two techniques below:

JSON Output

\$ curl -sqv -H 'Accept: application/json' http://localhost:8080/resource/1/1
{"handler":"candela.lanforge.HttpResource\$JsonResponse","uri":"resource","candela.lanforge.HttpResource\$JsonResponse","uri":"resource","candela.lanforge.HttpResource\$JsonResponse","uri":"resource","candela.lanforge.HttpResource\$JsonResponse","uri":"resource","candela.lanforge.HttpResource\$JsonResponse","uri":"resource","candela.lanforge.HttpResource\$JsonResponse","uri":"resource\$JsonResponse","resource\$JsonRespon

Clearly, the JSON output is difficult to read. We cover formatting output below.

HTML Output

Most of the queries to the client will return JSON by default. The notable exception is the /help URL. To get HTML output in the terminal, you have to specify Accept: text/html to curl:

<pre>\$ curl -sqv -H 'Accept: text/html' http://localhost:8080/port/1/1/1</pre>
html
<html></html>
<head><title>/port</title></head>
<body></body>
<thead>EIDAPActivityChannel>Devic</thead>
1.1.10.0>td>>td >td >td >td >td >td >td >td >td >
<hr/>
· · ·

Formatting Results

JSON formatted text is pretty difficult to read, there are a few different utilities that can help you look at it: jq, json_pp, json_reformat, tidy, xmllint, yajl and jsonlint.

Example of installing formatters

On Fedora, install:

\$ sudo dnf install -y jq perl-JSON-PP tidy libxml2 yajl

On Ubuntu, install:

\$ sudo apt install -y jq libjson-pp-perl perltidy xmllint libxml2-utils yajl-tools

Now we can perform a query:

```
$ curl -sq /port/1/1/1
{
    "candela.lanforge.HttpPort" : {
    "duration" : "1"
},
```



Notice that the URI object list paths with colon-tagged positions in them, e.g.: /cli-form/:cmd. These are interpreted as URL parameters and not query string parameters, they cannot be moved into the query string.

Making your shell friendly

To save you typing, you might want to add this function to your .bash_aliases file:

<pre>function Json() { curl -sqv -H 'Accept: application/json' "http://localhost:8080\${@}" \ json_reformat less</pre>	
}	

Then you can make your calls this way:

\$ Json /port/1/1/1

Browsing results in table format

We can view a URL in a browser as well:

/port	× +)	+				
← → ♂ ŵ	i localhost:	8080/port/1/1/1/		•• ♥ ☆ () III'	\ 🙂	🖬 »	=
EID AD Astistic	honnel	Device Dov	n IP	Parent Dev	Phantom	Port	SSID	1	
EID AP ACTIVITY C									
EID AP Activity C 1.1.1 0.0 </td <td></td> <td></td> <td>e 0.0.0.0</td> <td></td> <td></td> <td>1.1.01</td> <td>3310</td> <td></td> <td></td>			e 0.0.0.0			1.1.01	3310		

Viewing Alerts and Events

You can both view and stream event data. Querying events and alerts are both quite similar:

A busy LANforge system will generate hundreds of thousands of events. Only the last few thousand can be recalled.

You can inspect a singular event:



We can view /alerts similarly.

```
$ Json /alerts/92
{
    "handler" : "candela.lanforge.HttpEvents$FixedJsonResponder",
    "uri" : "alerts/:event_id",
    "alert" : {
        "name" : "wlan0",
        "time-stamp" : "2018-07-02 16:23:30.880",
        "entity id" : "NA",
        "id" : "92",
        "eid" : "1.1.5",
    }
```

Streaming Events

Continually polling the /events URL is not as effective as streaming a websocket providing the same data. We need a web socket client. Websockets are built into modern browsers and there are python and perlutilities for the job as well. An easy to use python client is wsdump.

Installing wsdump

There is a useful python utility called wsdump (or wsdump.py). Try to install the python-websocket package to get it. There are many similar matches, but there is not one dedicated package that provides it. On Fedora:

root@fedora\$ dnf whatprovides `which wsdump` root@fedora\$ **dnf install -y python3-websocket-client** root@ubuntu\$ 1s -1 /usr/bin/wsdump /usr/bin/wsdump _ /etc/alternatives/wsdump root@ubuntu\$ 1s -1 /etc/alternatives/wsdump /stc/alternatives/wsdump _ /usr/bin/python2-wsdump root@ubuntu\$ dpkg-query -5 /usr/bin/python2-wsdump python-websocket: /usr/bin/python2-wsdump root@ubuntu\$ sudo apt install python-websocket

You might need to install pip, and that might be in the python3-pip package. Then you can install via:

sudo	apt	install	python-pip # or sudo dnf install python-pi	р
sudo	pip	install	upgrade pip	
pip s	searc	ch websoo	ket	
sudo	pip	install	websocket-client	

Streaming Using wsdump

Ş

Here's an example of wsdump below. Don't forget you are now using h the ws:// schema and not the http:// schema!

\$ /usr/bin/wsdump ws://localhost:8081/

It might take a few second to start showing results if your system is not very active. You should be able to prompt output by executing this message in the **Messages** tab: **gossip hi ben!**

	y restarting th						
	ip hi ben!						
Input: gos	ap in bend						
							_
Logged in	to: idtest:	4002 as:	Admin				
fi-event	:"1.8:	sta8105	(phv #3	B): scan	finished	: 5180, \"\""}	
					ssip, 'hi		
					started"		
		sta/101	(phv #)	l): scan	started"		

Streaming Using javascript

You can also use a web page to follow events because websockets are built into modern browsers. This is a screenshot of the

WebSocket Test
URL (ws://) ws://jed-f23:8081/ Connect Disconnected
<pre>{*wifi-event*:*1.3: IFNAME=sta3120 <3>CTRL-EVENT-CONNECTED · Connection to 00:0e:8e:d5:fa:e6 completed [id=0 id_str=]*}</pre>
<pre>""""""""""""""""""""""""""""""""""""</pre>
<pre>(*flags:0, event_type:0, "event_id*:400895,"eid_type:12,"sheif*:1,"resource:13,"port*:33,"endp*:0,"extra*:0,"pri ata3120 is Link DOWN.*, "name:"sta3120","eid*:11,33","event_eid*:{"type*:11,"event_id*:400895,"index*: / flags:0," isidert:"tue)</pre>
{"deleted-alert":400895}
<pre>(*flags*0, *event_type*0, *event_id*400956, *eid_type*2, *ehelf*1, *resource*3, *port*33, *endp*0, *extra*:0, *pri ata3120 is link DOWL*, *name*:*ata3120*,*eid*:*13,33*, *event_eid*:(*type*:11, *event_id*:400856, *index*:- , flags*0, *is_alert*:true), *is_alert*:true)</pre>
<pre>(*flags:0,*event_type:0,*event_id*:1023429,*eid_type*:2,*shelf*:1,*resource*:3,*port*:33,*endp*:0,*extra*:0,*pr eta3120 is Link DOWN,*nnme*:*eta3120,*eid*:13,33*,*event_eid*:(*type*:11,*event_id*:1023429,*index*:- 1,*flags*0,*flag_ider*:false);*is_leter*:false)</pre>
{"wifi-event":=1.3: sta3120: del station 00:0e:8e:d5:fa:e6=}
<pre>{*wifi-event*:=1.3: sta3120 (phy #1): deauth 00:0e:8e:a1:7d:45 -> 00:0e:8e:d5:fa:e6 reason 3: Deauthenticated because sending station is leaving (or has left) the IBSS or ESS*}</pre>
<pre>[*flag#:0, event_type=17, event_tdf:102340. edd_type=12, "helf*1, tranource*13, "port*133, "endp*10, "extra*10, "pr. [phy #]: disconnected [local request] reaction: 3: Deathclicated becau*, "name*:#cal200, "edd:*11.333, "event_edd"("type*11], "event_id*1023430, "index*:- . !flag#10," im_alter*1falme, 'im_alter*Ifalme)</pre>
<pre>{*wifi-event*:*1.3: sta3120 (phy #1): disconnected (local request) reason: 3: Deauthenticated because sending station is leaving (or has left) the IBSS or ESS*}</pre>
<pre>(*flags:0,*event_type:50,*event_id:1023431,*eid_type:2,*shalf*:1,*seource*:3,*port:32,*endp*:0,*extra*:0,*p sta3119 r_changed from 0.0.0.0 to 10.4113.111*,*name*:sta3119*,*eid*:11.32*,*event_eid*: [*ype*11,*event_id*:1023431,*index::1,*flag=10.*ind=1ex*:flag=10.*is_index*:flag= </pre>
DISCONNECTED

<u>Data Views</u>

URLs

/shelf

The /shelf/1/ URL provides a list of resources in your realm:

Json /shelf/1	
"handler": "candela.lanforge.HttpResource\$JsonResponse",	
"uri": "shelf/:shelf id",	
"candela.lanforge.HttpResource": {	
"duration": "0"	
"resources": [
1	
"1.1": {	
" links": "/resource/1/1",	
"hostname": "idtest.candelatech.com"	
}	
},	
"1.2": {	
" links": "/resource/1/2",	
"hostname": "hedtest"	
1 L	
6	_



/shelf/1/

Resource 1
Resource 2
Resource 3
Resource 4
Resource 5
Resource 6
Resource 7
Resource 8

/resource

The /resource URL provides a digest of ports available at the requested resource.



- cli-port: 4003 • cpu: Intel(R) Core(TM) i7-3555LE CPU (2137Mhz)(x4)
- /port

The /port URL provides a digest of ports and their state. You can request multiple ports by ID on this resource by appending the port IDs with commas. You can list ports on a resource:



We can query multiple ports at a time by their number or their name by placing a comma between the specifiers. Additionally, we can query for just the fields we desire. All field names are lower-case: ? fields=tx+crr,rx+fifo.



/cx

The /cx URL allows us to query Layer-3 connection information.



And individual connections:

```
$ Json /cx/udp:r3r2:3000$ Json 'cx/udp:r3r2:3000'
```

```
"uri" : "cx/:cx_id",
"41.1" : {
    "drop pkts b" : 0,
    "type" : "LF/UDP",
    "rx drop % a" : 0,
    "pkt rx a" : 0,
    "avg rtt" : 0,
    "avg rtt" : 0,
    "rx drop % b" : 0,
    "name" : "udp:r3r2:3000",
    "endpoints (a -- b)" : "udp:r3r2:3000-A <=> udp:r3r2:3000-B",
    "drop pkts a" : 0,
    "entity id" : "NA",
```

Technically, colons in URLs need to be encoded as **%3A**, so the above URL should be /cx/udp%3Ar3r2%3A3000, but curl is pretty damed forgiving.

/endp

Endpoints may be listed and inspected:

Json /endp/
{
"uri" : "endp",
"handler" : "candela.lanforge.HttpEndp\$JsonResponse",
"candela.lanforge.HttpEndp" : {
"duration" : "4"
},
"endpoint" : [
{
"1.2.8.55.2" : {
"_links" : "/endp/55",
"entity id" : "NA",
"name" : "sta3000-ep-B"
}
},
Json /endp/sta3000-ep-B

	"duration" : "1"		
},			
"ur:	i" : "endp/:endp_id",		
"end	dpoint" : {		
	"rx rate 11" : 0,		
	"pdu/s tx" : 0,		
	"bursty" : false,		
	"rx rate" : 0,		
	"tx pkts 11" : 0,		
	"rx bytes" : 0,		
	"run" : false,		
	"top rtx" : 0,		

Creating Ports

It is possible to create ports and connections by using the CLI commands. Your LANforge test scenarios (located in the /home/lanforge/DB/ directory) contain all the CLI commands that create your ports and connections. You can submit those commands over HTTP in two ways:

• /cli-json/\$command An example of using the gossip command:

Then check your LANforge GUI messages.

• /cli-form/\$command An example of using the gossip command:

curl -X POST -d 'message=hello+world' http://localhost:8080/cli/gossip

Then check your LANforge GUI messages.

 /cli/: use this method to submit a raw URL-encoded command. This might be useful if you are copying commands directly out of a database:

curl -X POST -d 'cmd=gossip hello' http://localhost:8080/cli/

Except for /cli-json, these methods accept application/x-www-form-urlencoded content type submissions. This is default for the NanoHttp library and default for curl.

These CLI commands do not return data, only a result code. All data that the Perl scripts would collect from command line queries is sent directly to the GUI. Some CLI commands send data over the websocket, like the diag command.

Command help

Commands are often complex and include a number of bitwise flags to set the state and features of ports. There is presently no tag-substitution for port flags, but there is a help utility that can help you compute them. http://localhost:8080/help/

← → ♂ ŵ	(i) localhost 8080/help/ ··· ♥ ☆ Q Search >>	≡
add_arm_endp add_bgp_peer add_bond add_br add_cd add_cd_endp	CLI Reference for add_arm_endp CLI Reference for add_bgp_peer CLI Reference for add_bond CLI Reference for add_br CLI Reference for add_cd CLI Reference for add_cd_endp	10
add_cd_vr add_channel_group add_cx add_endp	CLI Reference for add cd_vr CLI Reference for add_channel_group CLI Reference for add_cx CLI Reference for add_endp	

Select a command to see the field helper screen:

http://localhost:8080/help/set_port

Type values into the field inputs and the CLI command will be refreshed:

↔ → ♂ ŵ	(i) localhost:8080/help/set_port	🔽 🏠 🔍 Search	<u></u> <u> </u>	١II/	1	C	≡
	Command Co	omposer [set	_port]				
This is the curl com	mand:						

\$ echo '' > /tmp/curl_data \$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' http://atlas:8080/cli-form/set_port

This is the CLI command:

Click the **Parse Command** button and the values in the command box will be displayed in the curl command and the field inputs. (Notice this form is doing a GET request.)

This is the curl command:

\$ echo 'shelf=1&resource=3&port=sta3000¤t_flags=2147483649&interest=16384&report_timer=3' > /tmp/curl_data \$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' http://atlas:8080/cli-form/set_port

This is the CLI command:

 $\left[1\ 3\ \mathrm{sta3000}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{2147483649}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{NA}\ \mathrm{16384}\ \mathrm{3}\ \mathrm{NA}\ \mathrm{NA}$

Parse Command

You may find a list of flag fields that are organized by field names. The text area below the selection list is the sum of the selected fields. Copy the flag values into the input field above to incorporate it into your command.

current.audonet/flow-control current.auto-negotiate flags_BTMASS_DISCONNECT flags_BTMASS_DISCONNECT flags_BTMASS_POWER_DOWN flags_BTMASS_POWER_DOWN flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_BTMASS flags_SUPMORTS_SUP	
interest.Alias interest.BRIDGE interest.BYPASS interest.CPU MASK	
interest.DHCP interest.DHCP-RLS	
interest.DHCPv6 interest.CHO.OFFLOAD interest.IFDOWN interest.IVTERNAL_USE interest.IVT6_ADDR5	

Creating a WiFi Station

Please refer to the scripts **1f_associate_ap.p1** and **1f_vue_mod.sh** for examples of how to produce lists of CLI commands involved in creating stations. Please refer to:

- 1. Learn CLI Commands used to operate WiFi stations
- 2. and Changing Station WiFi SSID with the CLI API

These will provide ways of collecting the CLI commands in log files for you to place into the command /help/ page.

• Use ssh to log into your LANforge manager. Use the 1f_vue_mod.sh script to create a station:

cd scripts
./lf vue mod.shmgr localhostresource 3create staname sta3101 \
radio wiphylssid idtest-1000-openpassphrase '[BLANK]' \
log cli /tmp/clilog.txt
cat /tmp/clilog.txt
set_wifi_radio 1 3 wiphyl NA -1 NA NA NA NA NA NA NA NA Oxl NA
add_sta 1 3 wiphyl sta3101 1024 idtest-1000-open NA [BLANK] AUTO NA 00:0e:8e:cl:df:45 8 NA NA
set_port 1 3 sta3101 0.0.0.0 255.255.0.0 0.0.0.0 NA 2147483648 00:0e:8e:cl:df:45 NA NA NA 840

1. Enter each command into the your browser toolbar by altering the command into a url:

http://localhost:8080/help/**set_wifi_radio**?cli=1 3 wiphy1 NA -1 NA NA NA NA NA NA

Produces:

\$ echo 'shelf=1&resource=3&radio=wiphy1&channel=-1&flags=0x1' > /tmp/curl_data \$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' \ http://localhost:8080/cli-form/set_wifi_radio http://localhost:8080/help/add_sta?cli=1 3 wiphy1 sta3101 1024 idtest-1000-open

Produces:

\$ echo 'shelf=1&resource=3&radio=wiphy1&sta_name=sta3101&flags=1024&ssid=idtest-\$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' \ http://localhost:8080/cli-form/add_sta

http://localhost:8080/help/**set_port**?cli=1 3 sta3101 0.0.0.0 255.255.0.0 0.0.0.0

Produces:

\$ echo 'shelf=1&resource=3&port=sta3101&ip_addr=0.0.0.&netmask=255.255.0.&gate
\$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' \
http://localhost:8080/cli-form/set_port

2. Verify with the LANforge GUI the changes you wish to make.

Creating Connections

Using the /cli-json/add_endp and /cli-json/add_cx URLs, it is possible to create Layer-3 connections. Create the Layer-3 endpoints first, of course.

Create L3 Endpoints

Construct your command using the /help/add_endp page. For an example, use these parameters:

alias	enter udp1000-A
shelf	1
resource	2
port	b2000
type	select type.lf_udp
min_rate	1000000 (1 Mbps)
max_rate	SAME
payload_pattern	select payload_pattern.increasing

Command Composer [add_endp]

This is the curl command:

\$ echo 'alias=udp1000-A&shelf=l&resource=2&port=b2000&type=lf_udp&min_rate=1000000&payload_pattern=increasing' > /tmp/curl_data \$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' http://atlas:8080/cli-form/add_endp This is the CLI command:

udp1000-A 1 2 b2000 lf_udp NA NA 1000000 NA NA NA NA increasing NA NA NA NA

Click Parse Command and copy the resulting curl command into a text editor:

Fi	le E	Edit	Search	Options		
	1# 2 e	1/bi cho	n/bash 'alias=	udp1000		
	67				6	

And for the **B** endpoint, choose a station:

alias	enter udp1000-B
shelf	1
resource	7
port	sta7000
type	select type.lf_udp
min_rate	54000 (54 Kbps)
max_rate	SAME
payload_pattern	select payload_pattern.increasing

6	_		LANforge CLI Help - Mozi	lla Firefox					E	
LANforge CLI Help	×	C Queryin	g the LANforge GUI for JSC	× +						
← → ♂ ŵ	(i) local	host:8080/1	90% 💟	्र Search		$\overline{\mathbf{A}}$	111	1 7 (1	C	≡
	Cor	mmar	nd Compo	oser [add	l_end	lp]				
This is the curl comma	nd:									
\$ echo 'alias=udp1000-B& \$ curl -sqv -H 'Accept:								> /tmp/c	url_dat	a
This is the CLI comma	nd:				1					
(F 4000 NA NA	NIK NIK desember NIK (

Click Parse Command and copy the resulting curl command into a text editor:



We'll save this file as a shell script: -/create-endp.sh We can then run it from our terminal like so: bash -x create-endp.sh

jreynolds@atlas:~/btbits/x64_btbits/html/i 1%	jreynolds@atlas:~ X	jreynolds@atlas:/mnt/d2/pub/system-images	2 1	reynolds@fs1:/mnt/d1/pub/sysimage	i jreynolds@atlas:~/btbits/x64_btbits/client
\$ cd			_		
irevnolds@atlas:~;#					
\$ bash ./create-endp.sh					
* Trying 192.168.100.51					
* TCP NODELAY set					
* Connected to atlas (192.168.100.51)	port 8080 (#0)				
> POST /cli-form/add_endp HTTP/1.1					
> Host: atlas:8080					
> User-Agent: cur1/7.55.1					
<pre>> Accept: application/json > Content-Length: 101</pre>					
> Content-Type: application/x-www-for	an uniterested				
- concentryper apprication what for	and reneoued				
* upload completely sent off: 101 out	of 101 bytes				
< HTTP/1.1 200 OK					
< Content-Type: application/json					
< Date: Wed, 1 Aug 2018 00:54:53 GMT					
< Connection: keep-alive					
< Content-Length: 147					

We should see the endpoints we've created in the LANforge GUI Endps tab:

					Stop A	JI Res	tart Manager	R	efresh H
Layer-4 Generic Test Mg Status Layer-3	r Test Gr L3 Endps		Re VolP/		t Log Alerts P Endps	Port Mgr V. WanLinks	AP Stations M Attenuators	lessages RF-Gener	ator File
Min PDU Size		▼ G	-	Aax PDU Size Same			Start <u>S</u> top	Quiesce Cle	ar
MIN Tx Rate New Modern	SG Kbpz)	▼ G	0 1	AX Tx Rate Same	•				
View 0 - 1000		▼ G	0			Display	Cr <u>e</u> ate Mo <u>d</u> ify	Batch Modify	Delete
					All Endpoints —				
	1			(an enapointo		1	1	
Name	EID	Run	Mng	Script	Tx Rate	Tx Rate (1 min)	Tx Rate (last)	Tx Rate LL	Rx Rate (
								-	0
dp2.b2000-08.sta8261-B	1.2.8.440		~	None	0	0	0	0	
	1.2.8.440			None None	6,082		0	0	62,573
dp2.b2000-08.sta8262-A			V			4,908			
dp2.b2000-08.sta8261-8 dp2.b2000-08.sta8262-A dp2.b2000-08.sta8262-8 dp1000-A	1.8.272		~	None	6,082	4,908	0		62,573
dp2.b2000-08.sta8262-A dp2.b2000-08.sta8262-B	1.8.272		V V V	None None	6,082 112,799 0 0	4,908 92,467 0	0	0	62,573 6,055 0
dp2.b2000-08.sta8262-A dp2.b2000-08.sta8262-B dp1000-A	1.8.272 1.2.8.446 1.2.8.596		× × ×	None None None	6,082 112,799 0	4,908 92,467 0	0 0 0		62,573 6,055 0
dp2.b2000-08.sta8262-A dp2.b2000-08.sta8262-B dp1000-A dp1000-B	1.8.272 1.2.8.446 1.2.8.596 1.7.10		× × × × ×	None None None None	6,082 112,799 0 0	4,908 92,467 0 56,155 55,425	0 0 0	0	62,573 6,055 0

Create L3 Connection

With the creation of two endpoints, we can proceed with creating a Layer 3 cross-connect. This is much simpler, it really only takes the names of the two endpoints we created above. We'll choose $default_tm$ for the test manager.

← → ⊂ ŵ	i localhost	90% 🗸 🟠	Q Search	<u></u> ⊻ ≫ ≡
Co	ommand C	omposer	[add_cx]	
This is the curl comm s echo '' > /tmp/curl d s curl -sqv -H 'Accept. This is the CLI comma	ata application/json' -X P	OST -d '@/tmp/curl_dat	a' http://atlas:8086	/cli-form/add_cx
Parse Command Fields for the command with 01 alias 02 iest_mgr 03 tr_endp 04 rr_endp	I update when you change th	iem:		
alias	udp10	00		
test_mgr	defaul	It_tm		
tx_endp	udp10	00-A		
rx_endp	udp10	00-В		

Click the **Parse Command** button and copy the resulting **cur1** command into your editor with the shell script. Run the script again. It doesn't hurt to re-create the endpoints.

				LANforge /	lanager Version	(5.3.8)		N			
ontrol <u>Reporting</u>	Iear-Off Ir	nfo <u>P</u> lugins						M.			
					St	op All	Restart	Manager		Refresh	HELP
					50	op An	restart	manager		Refreati	TILL
Laver-4 Generic	Test Mar	Test Grou	p Resource	Mar Eve	nt Log Ale	erts Port	Mar VAP	tations	lessages		
Status Laver		B Endps	VoIP/RTP		TP Endps	WanLi		ttenuators		enerator	File-IO
Status Layer	-5 _ 13	cirups	VOIP/INTE	Voir/is	in Enups	want	IND P	attenuators	1.1.0	enerator	Filefio
Det Tie	ner: fast	(1 s)	▼ Go Test I	danagar al	-	5	elect All	tart Stop	Quiesce	Clear	
npt III	ier: Tast	(15)	• GO Test I	Manager an	`		ilect Air		Quiesce	cicur	
View	0 - 500			▼ Go			Display	Create	Mo <u>d</u> ify E	Delete	
View	0 - 300	,		• 00			Dispidy	cieace		velere	
			Cr	oss Connec	ts for Select	ed Test Mar	nager				
		1					5	100 00 0000		1	
Name	Type	State	Pkt Rx A	Pkt Rx B	Bps Rx A	Bps Rx B	Rx Drop %	Rx Drop %	Drop Pkts A	Drop Pkts B	Avg RTT
							А	в			~
	LF/UDP	Run	2,250,983	211,913	64,536	6,075	42.966	0.357	1,695,749	759	1,01
dp2.b2000-08.s	LF/UDP	Stopped	0	0	0	0	0	0	0	0	
dp2.b2000-08.s	LF/UDP	Stopped	1,576,746	152,583	62,573	6,055	44.527	0.452	1,265,599	693	67
dp1000	LF/UDP	Stopped	0	0	0	0	0	0	0	0	
udp:r3r2:3000	LF/UDP	Stopped	68	74	50,623	55,090	8.108		6		1,74
udp:r3r2:3001	LF/UDP	Stopped	40	77	28,802	55,445	48.052				1,71
4 August 2002	I C/UDD	Stoppod	62	77	45 292	55.460	10107		1.4	0	216-

Toggling the Connection

Cross connects have three good state: STOPPED, RUNNING, and QUIESCE. The command to change them is set_cx_state. You will have no trouble creating the command:

test_mgr	default_tm
cx_name	udp1000
cx_state	RUNNING

Command Composer [set_cx_state]

This	is th	e cur	command:

\$ echo '' > /tmp/curl_data
\$ curl -sqv -H 'Accept: application/json' -X POST -d '@/tmp/curl_data' http://atlas:8080/cli-form/set_cx_state

default_tm udp1000 RUNNING	
Parse Command	
Fields for the command will update when you change them:	Flag Fields for command will be computed when you select them, but you might need to actually write modified values into some fields (when you see token values like [string] o [name]).
01: test_mgr default_tm	ox_state.DELETED ox_state.QUIESCE
02: cx_name udp1000	cx_state.STOPPED
03: cx_state RUNNING	cx_state.SWITCH

Click Parse Command and then you can paste the resulting command into your editor

Command Com	poser [set	t_cx_state
-------------	------------	------------

This is the curl command:

\$ echo 'test_mgr=default_tm&cx_name=udp1000&cx_state=RUNNING' > /tmp/curl_data
\$ curl -sqv _H 'Accept: application/json' -X POST -d '@/tmp/curl_data' http://atlas:8080/cli-form/set_cx_state

I his is	tne	CLI COI	nmand:	
default	+m	udp1888	DUNNTNG	

Parse Command	
Fields for the command will update when you change them:	Flag Fields for command will be computed when you select them, but you might need to actually write modified values into some fields (when you see token values like [string] or [name]).
01: test_mgr default_tm	cx_state.DELETED cx_state.QUIESCE
02: cx_name udp1000	cx_state.RUNNING cx_state.STOPPED cx_state.SWITCH
03: cx_state RUNNING	The biolowing numbers are only valid for signed 64 bit values, so not all flags can be calculated as positive unsigned integers. If you see a negative number, first check that the Java flag was not entered as an int.

Advanced Techniques

You can make JSON submissions and you can also submit Base64 encoded values in both form an and JSON submission URLs.

Submitting Base64

Field names that end in -64 are interpreted as base64 encoded values. From a linux terminal, you can convert text to base64 encoded value using the **base64** command:

\$	echo	"RUNNING"	base	64
J	LVOTk1	.0Rwo=		

Below is a CLI command example. You typically would not care to spend the effort doing this unless the data you need to express is difficult to URL encode.

\$ echo 'test_mgr-64=YW55Cg==&cx_name-64=dWRwMTAwMAo=&cx_state=64=UIVOTK1ORwo=' > /tmp/curl_data
\$ curl -A 'Accept: application/json' -X POST -d @/tmp/curl_data http://host/cli-form?set_cx_state

Submitting JSON

Instead of posting to /cli-form, you can post to /cli-json and your submission will be parsed as a json object. The parameter names stay the same. The base64 name extensions are also available! You **need** to specify that your **Content-type** in the POST is **application/json**.

\$ echo '{"test_mgr":"default_tm","cx_name":"udp1000","cx_state":"RUNNING"}' > /tmp/curl_data
\$ curl -sq -H 'Content-type: application-json' -H 'Accept: application/json' \
 -X POST -d@/tmp/curl_data http://localhost:8080/cli-json/set_cx_state

Handling Mismatched Column Errors

(This should be fixed as of 2018/08/14) When the LANforge cliet is in GUI mode, the **columns** of data that are returned match the GUI **table columns** displayed. You can use the Right-click \rightarrow Add/Remove Table Columns menu item to change this. We do not recommend doing this for querying JSON data though, because the table columns definitions will not match up to the data the webserver expects to return.

Generio Statu		t Mgr Layer		Group I L3 Endp	Resource Mgr s VolP/R		erts Port Mgr TP Endps	VAP St WanLink		essag tenu:	
Dis	p: 192	.168.1	00.51:0	0.0	Sniff Packets		1 Clear Cour	nters	Reset Po	rt	Delete
Rpt	Timer:	mediu	.m (8	s) 🔻	Apply	VRF	Iiew Det	ails	Cr <u>e</u> ate		Mo <u>d</u> ify
	198				All E	thernet Interface	es (Ports) for all I	Resource	s. —	1	
Port	Pha	Down	Parent Dev	Channel	Device	SSID	AP		IP	Ad	tivity
1.1.00					eth0			192.1	68.100.41	0	
1.1.01			· · · · · · · · · · · · · · · · · · ·		ethl			0.0.0	.0	0	
1.1.02		~	a	36	wiphy0			0.0.0	.0	0	
1.1.03		~		44	wiphy1			0.0.0		0	
1.1.04			2	48	wiphy2			0.0.0	.0	0	
1.1.05		~	wiphy0	36	wlan0		Not-Associated	0.0.0	.0	0	

"status" : "INTERNAL_ERROR"

The terminal you started the LANforge client on will also give a similar error:

1532480073953: names_to_col_ids size:71 java.lang.IllegalArgumentException: names_to_col_ids map is not going to work: 1532480073953: lfj_table columns:10

Reset the Table Layout

1. Right-clicking the Port Mgr and selecting Add/Remove Table Columns will allow you to change this.

lerts RTP End	Port Mg		AP St InLinl	ations <s< th=""><th></th><th>ssag tenu</th><th>es ators</th><th></th></s<>		ssag tenu	es ators	
î	Clear Co	unter	s	Res	et Por	t	De	le <u>t</u> e
ĩ	⊻iew D	etails		Cr	<u>e</u> ate		Мо	<u>d</u> ify
es (Po	rts) for al	II Reso	ource	es. —				1
	AP			IP		Ad	tivity	
			192.3	168.10	0.41	0		
_			0.0.0			0		
_			0.0.0 0.0.0			0		-
-			0.0.0			0		-
Not Not en a2 a2 a2 	Modify 5 Reset 5 Display Dyna Table R Coun <u>t</u> 5 <u>C</u> alcula View Lo Create	electe Select Select Select Select mic Re mic Re Select Select tions gs	ed ed ed (L eted eport	ogout))		D	
Not-	Addition						-	-
en Not-	Add/ <u>R</u> el				rt Col	umns	-	-
a2	<u>S</u> ave Ta							
а2	Reset T							
-	Auto-Re	esize (Colun	nns				

2. Clicking the Select All/None button and then Apply will get all the columns displayed, and returned in your queries.

<u></u>	Add or Remove	Table Columns	
🖌 Port	🖌 Phantom	Down	🖌 Parent Dev
🖌 Channel	🖌 Device	SSID	AP
🖬 IP	🖌 Activity	SEC SEC	🖌 Alias
🖌 RX Bytes	🗾 RX Pkts	🗹 Pps RX	🗾 bps RX
🗹 TX Bytes	🗾 TX Pkts	🗾 Pps TX	🖬 bps TX
🗹 Collisions	RX Errors	TX Errors	🗹 RX Drop
🗹 RX Length	RX Over	RX CRC	🖌 RX Frame
🗾 RX Fifo	🖌 RX Miss	🗹 TX Abort	🗹 TX Crr
🗾 TX Fifo	🖌 TX HB	🗾 TX Wind	🖬 bps TX LL
🗾 Bytes TX LL	🗾 bps RX LL	🗾 Bytes RX LL	🖌 Reset
🗹 TX-Rate	🖌 RX-Rate	🖌 Status	🗾 Signal
🗾 Noise	🖌 Connections	🗹 DHCP (ms)	🗹 CX Ago
🗹 No CX (us)	🗹 CX Time (us)	🗹 ANQP Time (us)	🖌 4Way Time (us
🗹 Crypt	🖌 Retry	Misc 🖌	🗾 Beacon
🖌 Key/Phrase	🖌 Login-OK	🖌 Login-Fail	🖌 Logout-OK
🖌 Logout-Fail	V QLEN	MTU	🖌 Mask
🖌 Gateway IP	MAC	🖌 IPv6 Address	🖌 IPv6 Gateway
	Select None	Apply	Cancel

3. Make sure to $\textbf{Right-Click} \rightarrow \textbf{Save Table Layout}$ so that your next session will show all the data.

						-		
s	Po		r VAP Stat		ssages			
6		Arma	geddon	WanLinks	A	ttenu	ato	
î	C	lear C	ounters	Reset Po	ort	Delete		
Ţ		⊻iew I	Details	Cr <u>e</u> ate		Mo <u>d</u>	dify	
5 (F	ort	s) for	all Resource	s. —				
Pa	arer	nt Dev	RX Bytes	RX Pkts	Pps	RX	b	
-			610.611	669.028	. 3	2.536	20,	
		Clear	Selected			8	16,	
		Modi	fy Selected			8		
			t Selected			4		
		Rese	t Selected (Logout)		6		
	phy		ay Selected	Logout)		0		
	ohy. ohy.		namic Repor			D 0		
WI	ony.	-				5		
MARIN	phy		Report			9	- ·	
	phy		t Selected			ō		
	phy.	-	ulations			ō		
-	-	View	Logs			0	<u> </u>	
		Crea	te Layer-3 C	X		6		
		Add/	Remove Tab	le Columns		0		
		_	Remove Tab		olumns	8		
			Table Layou		olamito	2		
	o buć		t Table Layo			4		
	phy.					0		
	phy		-Resize Colu	nns		0		
	, iy	~	6,581,71	5,595,679	9	89		

4. Restart the LANforge client

Learn CLI commands used to operate WiFi stations.

Goal: Compare and learn script and CLI commands used when creating and operating stations.



Examples of CLI commands

- 1. Creating Stations
- 2. Using Open Authentication
- 3. Using WPA2 Authentication
- 4. Static IP Addresses
- 5. Station DHCP IP Address
- 6. Creating a Station with a MAC Address Pattern
- 7. Admin Down
- 8. Admin Up
- 9. Deleting a Station
- 10. Creating Connections and Running Traffic
- 11. Starting and Stopping Traffic
- 12. Create a Layer 3 TCP Connection
- 13. Create a Layer 3 UDP Connection
- 14. Create a Layer 4-7 Web Connection

Setting for Examples

This was done in a two-machine LANforge cluster, the manager named jedtest and the second resource named kedtest. The CLI output of these CLI commands has been discarded as well as any show_port commands.

The **show_port** commands are useful for inspecting the results of previous commands. Often there is useful wait before issuing the **show_port** command to allow processing time on the manager. Please inspect the scripts in the **/home/lanforge/scripts** directory for how and when they tend to sleep.

These commands are also found in the /home/lanforge/DB/DFLT directory files. You cannot run those DB files directly, because they are executed in certain order. However, you can grep for connection- and station-names in those files to find results of GUI commands.



Using Open Authentication

This station is created with DHCP enabled. That is controlled via flags that are descibed in the add_sta command.

Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --create_sta --name sta100 --radio wiphy0 -security open --ssid jedtest

Perl script:

./lf_associate_ap.pl --mgr jedtest --resource 2 --quiet yes --action add --radio wiphy0 -security open --ssid jedtest --passphrase --first_sta sta100 --first_ip DHCP -num_stations 1

CLI command:

set_wifi_radio 1 2 wiphy0 NA -1 NA NA NA NA NA NA NA NA 0x1 NA add_sta 1 2 wiphy0 sta100 0 jedtest NA [BLANK] AUTO NA 00:0e:8e:8d:8d:e9 8 NA NA NA set_port 1 2 sta100 0.0.0.0 255.255.0.0 0.0.0.0 NA 2147483648 00:0e:8e:8d:8d:e9 NA

Using WPA2 Authentication

This station is created with DHCP enabled. That is controlled via flags that are described in the add_sta command.

Shell script:

\$./lf_vue_mod.sh --mgr jedtest --resource 2 --create_sta --name sta200 \
--radio wiphy1 --security wpa2 --ssid jedtest --passphrase jedtest1 \
--log_cli /tmp/clilog.txt

Perl script:

- ./lf_associate_ap.pl --mgr jedtest --resource 2 \
- --action add --radio wiphyl --security wpa2 --ssid jedtest \
 --passphrase jedtest1 --first_sta sta200 --first_ip DHCP --num_stations 1

CLI command:

```
set_wifi_radio 1 2 wiphy1 NA -1 NA NA NA NA NA NA NA NA 0x1 NA
add_sta 1 2 wiphy1 sta200 1024 jedtest NA jedtest1 AUTO NA 00:0e:8e:6f:01:62 8 NA N
set_port 1 2 sta200 0.0.0.0 255.255.0.0 0.0.0.0 NA 2147483648 00:0e:8e:6f:01:62 NA
```

Static IP Addresses

Here is an example of creating a virtual station with a static address: 10.26.2.14/255.255.254.0 Shell Script:

.....

Perl Script:

./lf_associate_ap.pl --mgr jedtest --resource 2 --action add --radio wiphy1 --first_sta sta203 --first_ip 10.26.2.4 --netmask 255.255.254.0 --ssid jedtest --security wpa2 -passphrase jedtest1 --num_stations 1 --wifi_mode abgnAC --log_cli /tmp/clilog.txt

CLI Command:

```
set_wifi_radio 1 2 wiphyl NA -1 NA NA NA NA NA NA NA NA 0x1 NA
show_port 1 2 wiphyl
add_sta 1 2 wiphyl
add_sta 1 2 wiphyl sta203 1024 jedtest NA jedtest1 AUTO NA 00:0e:8e:63:50:62 8 NA N
set_port 1 2 sta100 10.26.2.4 255.255.254.0 0.0.0.0 NA 0 00:0e:8e:63:50:62 NA NA NA
```

Station DHCP IP Address

For the station to gain a DHCP IP address, you have to admin-up the station.

Creating a Station with a MAC Address Pattern

The <code>lf_associate_ap</code> script contains logic that parses a MAC address pattern and produces new MAC addresses. This is not a feature of the LANforge Manager. Your CLI calls to the LANforge manager will not parse the mask.

The pattern nomenclature of the LANforge GUI can also be used when specifying a MAC address for stations:

xx keep parent radio octet

```
.
```

randomize this octet

00 - ff

assign this value to the octet

Shell script:

Perl script:

./lf_associate_ap.pl --mgr jedtest --resource 2 --action add --radio wiphy1 --first_sta sta205 --first_ip 10.26.2.4 --netmask 255.255.254.0 --ssid jedtest --security wpa2 -passphrase jedtest1 --num_stations 1 --mac-pattern '4e:xx:xx:*:01' --log_cli /tmp/clilog.txt

CLI command:

set_wifi_radio 1 2 wiphyl NA -1 NA NA NA NA NA NA NA NA 0x1 NA show_port 1 2 wiphyl add_sta 1 2 wiphyl sta205 1024 jedtest NA jedtest1 AUTO NA 4e:0e:8e:43:f1:01 8 NA N. set_port 1 2 sta205 10.26.2.4 255.255.254.0 0.0.0.0 NA 0 4e:0e:8e:43:f1:01 NA NA NA Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --down --name sta200 --log_cli /tmp/clilog.txt
--quiet 1

Perl script:

./lf_portmod.pl --manager jedtest --card 2 --port_name sta200 --set_ifstate down

CLI command:

set_port 1 2 sta200 NA NA NA NA 1 NA NA NA NA 8388610

Admin Up

Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --up --name sta200 --log_cli /tmp/clilog.txt -quiet 1

Perl script:

./lf_portmod.pl --manager jedtest --card 2 --port_name sta200 --set_ifstate up

CLI command:

set_port 1 2 sta200 NA NA NA NA 0 NA NA NA NA 8388610

Delete Station

Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --delete_sta --name sta200 --log_cli /tmp/clilog.txt --quiet 1

Perl script:

./lf_associate_ap.pl --mgr jedtest --resource 2 --action del --port_del sta200

```
CLI command:
```

rm vlan 1 2 sta100

Creating Connections and Running Traffic

LANforge can create Layer-3 and Layer 4-7 connections using the **1f_vue_mod.sh** script. When connections are created, they exist in a stopped state. Connections can then have their state changed to RUNNING to start traffic.

Starting and Stopping Traffic

Layer-3 and Layer 4-7 connections both subject to the states STOPPED, RUNNING, and QUIECSE.

./lf_vue_mod.sh --mgr jedtest --start_cx --name tcp200 --log_cli /tmp/clilog.txt --quiet 1

./lf_vue_mod.sh --mgr jedtest --stop_cx --name tcp200 --log_cli /tmp/clilog.txt --quiet 1

Perl script:

./lf_firemod.pl --mgr jedtest --action do_cmd --cmd "set_cx_state default_tm tcp200 RUNNING"

./lf_firemod.pl --mgr jedtest --action do_cmd --cmd "set_cx_state default_tm tcp200 STOPPED"

CLI commands:

<pre>set_cx_state defau</pre>	ult_tm tcp200 RUNNING	
<pre>set_cx_state defau</pre>	ult_tm tcp200 STOPPED	

Create a Layer 3 TCP Connection

Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --create_cx --name tcp200 --tcp --sta sta200 -port eth1 --bps 1000000 --log_cli /tmp/clilog.txt --quiet 1

Perl script:

```
./lf_firemod.pl --mgr jedtest --resource 2 --action create_endp --endp_name tcp200-.
./lf_firemod.pl --mgr jedtest --resource 2 --action create_endp --endp_name tcp200-.
./lf_firemod.pl --mgr jedtest --resource 2 --action create_cx --cx_name tcp200 --cx
```

CLI commands:

add_endp tcp200-A 1 2 sta200 lf_tcp -1 NA 1000000 1000000 NA -1 -1 increasing NO NA 0 0 set_endp_report_timer tcp200-A 5000 add_endp tcp200-B 1 2 ethl lf_tcp -1 NA 1000000 1000000 NA -1 -1 increasing NO NA 0 0 set_endp_report_timer tcp200-B 5000 add_cx tcp200 default_tm tcp200-A tcp200-B set_cx_report_timer default_tm tcp200 5000 NA

Create a Layer 3 UDP Connection

Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --create_cx --name upd200 --udp --sta sta200 -port eth1 --bps 2000000 --log_cli /tmp/clilog.txt --quiet 1

Perl script:

i) ci_iiii ciiodip c	-myi jeutest	resource z	log_cli	/tmp/clilog.txt	quiet Iac
./lf_firemod.pl	-mgr jedtest	resource 2	log_cli	/tmp/clilog.txt	quiet 1ac
./lf_firemod.pl	-mgr jedtest	resource 2	log_cli	/tmp/clilog.txt	quiet 1ac

CLI commands:

add_endp udp200-A 1 2 sta200 lf_udp -1 NA 2000000 2000000 NA -1 -1 increasing NO NA 0 0 set_endp_report_timer udp200-A 5000 add_endp_udp200-B 1 2 eth1 lf_udp -1 NA 20000000 2000000 NA -1 -1 increasing NO NA 0 0 set_endp_report_timer udp200-B 5000 add_cx udp200 default_tm udp200-A udp200-B set cx report timer default tm udp200 5000 NA

Create a Layer 4-7 Web Connection

Layer 4-7 connections are created with a one-sided technique, the curl command always operates on the **A-side** and the **B-side** is unmanaged. The endpoint and connection naming does not follow the Layer-3 convention.

Shell script:

./lf_vue_mod.sh --mgr jedtest --resource 2 --create_l4 --name yh200 --sta sta200 --url http://www.yahoo.com/ --utm 2400 --log_cli /tmp/clilog.txt --quiet 1

Perl script:

Commands are set using lf_firemod.pl --action do_cmd --cmd ...

CLI commands:

add_l4_endp yh200 1 2 sta200 l4_generic 0 10000 2400 'dl http://www.yahoo.com/ /dev	1
set_endp_tos yh200 DONT-SET 0	I
set_endp_flag yh200 L4Enable404 0	I
<pre>set_endp_report_timer yh200 5000</pre>	I
set_endp_flag yh200 ClearPortOnStart 0	I
set_endp_quiesce yh200 3	I
add_cx CX_yh200 default_tm yh200	2

Learn CLI commands used create Generic endpoints.

Goal: Compare and learn script and CLI commands used when creating and operating Generic endpoints.

Similar to the Layer3 perl script, **1f_firemod.p1**, the **1f_generic_ping.p1** script has been enhanced to use curl or other commands with parameter expansions. The **1f_curl.sh** script is a helper script that wraps curl commands and reports success or failure.

Introduced in LANforge 5.3.8.



Example Commands

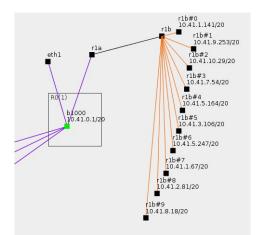
- 1. Creating ports and MAC VLANs
- 2. Creating ping endpoints
- 3. Creating curl endpoints
- 4. Using parameters for endpoint options
- 5. Using the 1f_cur1.shscript

Setting for Examples

Generic endpoints are effectively one-legged connections. You can ping an IP or use curl to download web content. Both of these types of connections can be pointed back at the LANforge itself. We can operate these connections from redirect interfaces. The same techniques can apply to WiFi stations, of course.

Create Redirects

- 1. Create a bridge br0 including eth1
- 2. In Netsmith, select br0 \rightarrow Modify and enable DHCP
- 3. Create a redirect r1a, r1b
- 4. Add r1a to br0
- 5. Create 10 MAC VLANs based off **r1b** with DHCP enabled



- After the MAC VLAN ports have addresses, you can verify that you can ping and download pages from LANforge webserver:
 - 1. In a terminal, begin by sourcing lanforge.profile:

/home/lanforge\$. lanforge.profile	
2. Ping the bridge from a MAC VLAN:	

/home/lanforge\$ ping -I 10.41.1.141 10.41.0.1

3. Grab the web page. (Apache is listening on all ports by default.)

/home/lanforge\$ curl -sq --interface 10.41.1.141 http://10.41.0.1/

Creating ping endpoints

In the /home/lanforge/scripts directory, the lf_generic_ping.pl script creates a wrapped ping command by default. There is another script, lfping, that reports ping results to LANforge. Here is an example of creating l ping endpoint:

Create a ping endpoint for every MAC VLAN parented by r1b:

\$./lf_generic_ping.pl --mgr localhost --resource 1 \
 --dest 10.41.0.1 --parent rlb

Create a ping endpoint for every virtual station parented by wiphy0:

Create a ping endpoint for every MAC VLAN beginning with r1b#1 (r1b#1, r1b#10):

Creating curl endpoints

To use other commands with the script, you can create a --and parameter. You can use curl directly if desired, but curl's output is not formatted well for LANforge to understand. By default, commands do not understand what port or IP they should be interacting as. We need to provide special parameters to help.

Parameter expansion

The 1f_generic_ping.p1 script will look for these tokens in the --cmd parameter:

- %i Expands to the IPv4 address of the port.
- %d Expands to the destination hostname or address
- %p Expands to the port name

The curl wrapper script

The scripts/lf_curl.sh script is a wrapper for curl that detects success or failure, and an operate the request in a loop. Expandable parameters are expanded by If_generic_ping.pl, not If_curl.sh.

You can use If_curl.sh from the command line to test it out:



Executes:

curl -sqLk --interface 10.0.0.1 -o /tmp/output_r1b#1 http://example.com/

So it is best used from If_generic_ping.pl to construct commands referencing this script:

/lf_generic_ping.plmgr localhostresource 1 \
name curl_exmatch 'r1b#'dest http://10.41.0.1/ \
cmd 'lf_curl.sh -o /tmp/curl_%p.out -i %i -d %d -p %p'

Uploading files with curl

It is possible to use the 1f_generic_ping.p1 script to create URL encoded form posts for uploading files.

1. If you don't have a file to upload, create one here:





2. Create a series of Generic endpoints:

```
cd /home/lanforge/scripts
./lf_generic_ping.pl --name up ∖
```

--match 'sta' \

In this test scenario, a script is used to bring up the WiFi Capacity test with a pre-configured configuration. The capacity test is then started and a report is generated. All of this is automated, and other

tests such as Dataplane are also supported. This feature requires

LANforge version 5.4.1 or higher.

- --dest http://192.168.48.1/ \
- --cmd 'curl -d @/var/www/html/data-2m.asc -o /tmp/curl_%p.out --interface %p --dns-ipv4-addr %i %d'

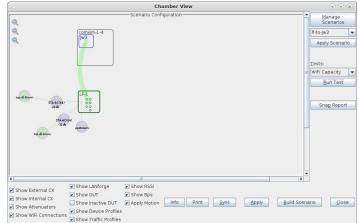
Automate WiFi Capacity and other GUI tests.

Goal: Use a command-line script to have the LANforge-GUI run the WiFi Capacity test and generate a pdf automatically.



1. Configure WiFi Capacity Test for automated run.

- A. For this to work, the LANforge GUI must be started with the -cli-socket 3990 argument. This causes it to open a socket to listen for text commands.
- B. Open Chamber View by clicking on the 'Chamber View' button in the LANforge-GUI. Create an appropriate scenario and DUT if you have not already done so. Other cookbook examples have more details of how to do this, please see those if you are unfamiliar with Chamber View.



C. Select WiFi Capacity test, and click **Run Test** to configure it as desired.

		WiF	i Capaci	y Test			\odot \odot
Settings Select Port	s Test Groups PC	U Mix Settings Advar	nced Setti	ngs Select Output	Notes		
	Station Incremen	t: 1,10,120	• [?]	Loop Iterations:	Single (1)	-	
	Duration:	20 sec (20 s)	•	🔲 Use Test Groups	Subset of Test G	iroup	
	Protocol:	UDP-IPv4	-	Layer 4-7 Endpoint:	NONE	-	
	Payload Size:	AUTO	-	MSS: AUTO			
	Total Download F	late:	-	1G (1 Gbps)		-	
	Total Upload Rat	e:	-	Zero (0 bps)		-	
	Percentage TCP I	Rate:		10% (10%)		-	
	Station-Down Qui	esce period:		0 (0 sec)		-	
		Save		DEFAULT			
		Load		DEFAULT		-	
		Delete		DEFAULT		-	
					Start	Pause	Cancel

D. Enter a name in the 'Save' field, click save, and make sure it shows up as a loadable configuration. In this case, we are saving the configuration as 'udp-dl-120'

		Wi	Fi Capaci	ty Test					 × ×
Settings	Select Ports Test Groups PE	U Mix Settings Adv	anced Setti	ings Select Output	Notes				
	Station Incremen	1,10,120	▼ [?]	Loop Iterations:	Single	(1)	-		
	Duration:	20 sec (20 s)	-	Use Test Groups	Subset o	of Test Gr	oup		
	Protocol:	UDP-IPv4	-	Layer 4-7 Endpoint:	NONE		-		
	Payload Size:	AUTO	-	MSS: AUTO					
	Total Download F	ate:	-	1G (1 Gbps)			-		
	Total Upload Rat	9:	-	Zero (0 bps)			-		
	Percentage TCP I	Rate:		10% (10%)			-		
	Station-Down Qui	esce period:		0 (0 sec)			-		
		Save		udp-dl-120					
		Load		DEFAULT			-		
		Delete		DEFAULT layer-4 testme					
				udp-dl-128 udp-dl-120			P	ause	Cancel

- 2. Use the If_gui_cmd.pl script to launch the WiFi Capacity Test.
 - A. Open an ssh session or terminal window and log into the LANforge system, or some other system with the LANforge scripts/ repository. On a LANforge system, this will usually be /home/lanforge/scripts In this case, the directory name is called If_scripts

	greearb@ben-t530:~/l	otbits/x64_btbits/server/lf_s	cripts	\odot \otimes \otimes
File Edit View Search Terminal	Help			
greearb@ben-t530 lf scripts]\$				
greearb@ben-t530 lf_scripts]\$				
reearb@ben-t530 lf scripts]\$				
greearb@ben-t530 lf scripts]\$				
		lf show events.pl	rand nmap.pl	
			README.md	
tenuator_series_example.csv				
lc autn.pl	lf icemod.pl	lf stress3.pl		
eate file assortment.bash			show-port-from-json.pl	
<pre>reate_file_assortment.bash reate-mounts.sh sorter.sh</pre>				
	lf 14 random speeds bash		strongswan-config	
cp-lease-list.pl		lf voip test.pl		
	lf_l4_reset.sh lf_log_parse.pl	lf vue mod.sh		
nctl lf0312.pl	lf loop traffic.sh	lf wifi dotlx.bash		
			timed ice pause.sh	
ur million.sh	lf macylan3.pl	lf wifi portal.bash		
			track call end.sh	
		lf zlt binary.pl		
		license.txt	wait on ports.pl	
		list phy sta.sh		
associate ap.pl				
		ocean-text.csv		
preearb@ben-t530 lf scripts]\$				

B. Run the If_gui_cli_cmd.pl script with appropriate arguments. Use --help for details. Once you run this, the WiFi Capacity test should be automatically opened and the test will be started. The script will end when the capacity test has completed. You may copy the results to some easily found location, such as a web server directory.

0	greearb@ben-t530:~/btbits/x64_btbits/server/lf_scripts		ð Ø)
File	Edit View Search Terminal Help			
Copy [gree [gree [gree [gree [gree [gree [gree [gree [gree [gree g udp	<pre>mding:cv pat 'mptest' Report Location:' tost results: par/hom/sread/thm/reports/wifi-capacity-2019-11-04-06-13-12 /tmp/ arabbent:30 [f scripti]s arabbent:530 [f scripti]s arabbent:540 [f scripti]s</pre>	te	confi	*
>>>Se OK	anding:cv load 'mytest' 'udp-dl-120'			
>>>Se OK	ending:cv click 'mytest' 'Auto Save Report'			
>>>Se OK	ending:cv click 'mytest' 'Start'			
>>>Se	indingicy get "mytest" (Report Location; multiple: mytest" (Report Location; multiple: get "mytest" (Report Location;			0 *

C. For details on what GUI-CLI commands are supported, please see the screen-shot below and look at the contents of the If_gui_cmd.pl script.

	Mate Terminal	\odot \land \times
File Edit	t View Search Terminal Help	
	@ben-dt4 ~]\$ telnet localhost 3990	-
Trying :		
	connect to address ::1: Connection refused	
	127.0.0.1 ed to localhost.	
	character is '^]'.	
	ed to LANforgeGUI version: 5.4.1	
lfaui# h		
Availabl	Le commands:	
	Gend LANforge CLI command. This can be useful for requesting the GU	I update
	ts data, for instance: cli show ports	
	ss: Flush spread-sheet info to the file system immediately.	
	eporting <start stop="" =""> <precision-in-seconds: 0="" best="" means=""> <"save-</precision-in-seconds:></start>	to-direct
ory"> ## For	automating chamber-view tests (plugins, etc)	
	et 'instance-name' 'object' 'value'	
	'eate 'instance-type' 'instance-name'	
	lick 'instance-name' 'object'	
cv lo	oad 'instance-name' 'config-name'	
	et 'instance-name' 'object'	
	pply scenario	
cv bu		
	elete 'instance-name' Give this help message.	
	Log out of this command session	
CALC.	Ebg but of this command session	
lfgui#		

Getting Started with Python scripting for LANforge

Goal: After reading this, a user will know what python modules which are available to use LANforge.

There are many python modules which control LANforge. These modules allow the user to automate many tasks. This cookbook will introduce the initial steps to start LANforge automation, the libraries that need to be imported to run Candela's python scripts, and how to create objects in python.

There are two options to run lanforge_scripts. If you are a programmer, you can clone the git repository locally by using git clone https://github.com/greearb/lanforge-scripts in your command line into the directory which the folder should be in.

1 Candela supports Fedora 27+ and Python 3.7+. Older python versions are not supported.

The advantage to cloning lanforge_scripts from git is that it is the latest code available, while the pip repository will lag slightly behind the git repo. The advantage to using pip is that there might be occasional errors in the git repository, which will be ironed out by the time we push it to the pip repository. The pip repository is always close to the master branch of the git repo, lagging no more than a couple weeks. Pick the one which best suits your needs.

¹ Please run Python as the **LANforge** user on your system.

If you run python as root, you might break your LANforge system. The LANforge Linux OS requires some certain python packages and versions to operate and configure networking correctly. Candelatech does not recommend updating python packages as root. Your package manager will automatically update these dependencies when you run updates as well, which can overwrite your changes.

The safe way to run python scripts is as a non-root user. If you have python dependencies installed locally by using **pip** --user, if a dependency breaks you can fix it without potentially harming files your operating system needs to operate. Broken python dependencies on your system Python can break your operating system. When your system updates, it might overwrite the changes you have made.

Users are responsible for making certain the version of Python running on their system is supported by the Python Foundation. An up to date list can be found at Python Foundation support. Candela Technologies does not support versions of Python which have been deprecated by the Python Foundation. Customers with a support contract can contact support@candelatech.com to get support upgrading their LANforge systems.

1. Importing libraries:

Please visit the readme.md page in lanforge-scripts/py-scripts to make sure all the libraries that are needed are imported.

2. Install necessary dependencies

A. In the root of lanforge_scripts, run pip3 install --user -r requirements.txt --upgrade

- 3. How to run a script from a git clone
 - A. We start in the /home/lanforge/lanforge-scripts/py-scripts directory on your lanforge system
 - B. On a Linux system, open your terminal and navigate to the py-scripts folder. You can run any python script by typing
 - python3 your_script_here.py --your_flag

into your command line. After the script, put any flags you need in order to run your test.

- 4. How to run a script from pypi
 - A. In your terminal, run python3
 - B. In your python environment, run import lanforge_scripts
 - c. You can use dir(lanforge_scripts) to see all of the different classes you can use in lanforge_scripts. This is useful if you want to make your own driver script which uses lanforge_scripts as a dependency.

5. Station Creation

	œ.													Mate Terminal 💿 🤅
					LAN	lforge Mana	ger Ver	sion(5.4.	3)					
Control	Bepi	rting Window	s info		iber View		Stop All		Restart Ma			lefresh	HELP	<pre>(base) [lanforge@lf0350-c0dc py-scripts]\$./create_station.pyradio wiphy6 sid lanforgepasswd passwordsecurity wpa2</pre>
VolP/RT	PEn	ps File-10	Resour							WitHessa				P Creating stations (base) [lanforge@lf0350-c0dc py-scripts]s
State		Port Mgr		Layer-3		Endps	Loyer 4-7		Armageddo		antinks	VolPJ	IRTP	
		2.168.1.4:1		Sniff Pac			1 Clea				elete			
Rpt	Time	edium (8	s) 💌	Appl		VRF ernet Interface		Dispjay		gate M	ogify	Batch Modi	fy	
Port	Ø	1 P	1	EC Alas	Dev	FX Bytes	FOC Pikts	Pps RX	bps for	TX Bytes	TX Pkts	Pps TX	bps T	an l
1.1.00		192.168.1.4		eth0 eth1		15,148,988		3	2,179				9,0	
1.1.02		0.0.0.0	0	eth2		0	0	6	0 0	0		0 0		
1.1.03		0.0.0.0	0	wiphy0 wiphy1		156.665.768	213,971	0	0 0	151.699.678		0 0		
1.1.05		0.0.0.0	8	wian0	wiphy0 wiphy1					0		0 0		
1.1.10		0.0.0.0	0	sta000	wiphy0	9				0		0 0		
1.1.11		0.0.0.0	0	staduo.	. wpnyd		0		3 0	U		0 0		
														,
4		-											•	
	n to:	ocahost 4002	as: Ad	min								4 stations (

- A. The create_station.py script creates wi-fi stations from your terminal.
- B. Type
 - ./create_station.py --radio wiphy0 --ssid lanforge --passwd password -- security wpa2

into your terminal. Please reemember to change the \protect{SID} , Passwd, and Security fields to match your network credentials.

- C. This will create 2 stations on your lanforge device off of your wiphy0 radio. You might need to wait 30 seconds for them to stop being phantom ports.
- D. You can specify the following flags as well:
 - A. num_stations Specify a different number of stations to create off of your antenna. The default is 2.

B. debug - call this flag if you want detailed diagnostic information

6. Find available networks

- A. You can find available WiFi networks on any Linux device by typing sudo iw dev wlan0 scan | grep SSID into your command line
- B. You can also find available WiFi networks by clicking on the station you created in the previous step,

7. Associate to a specific BSSID:

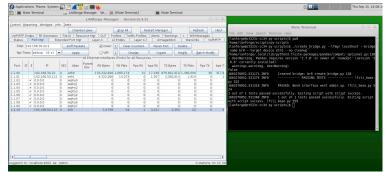
If you want to connect to a specific MAC Address for your router (which is called the BSSID) you can specify that on each station, you can do that in both the GUI and in the command line.

- A. Connect via Command line
 - A. Each of your scripts has an optional AP tag where you can define the MAC address of the router you want to connect to. To do so, simply append --ap to the end of the command line argument you are running followed by the router's MAC Address.

B. Connect via GUI

A. When you double click on a Station there is an AP field inside WiFi Settings. Type the MAC Address of your router in that box and everything will work assuming the MAC Address is correct.

8. Bridge Creation



- A. To create a bridge, you can use the create_bridge.py script in the py_scripts folder.
- B. Type

./create_bridge.py --lf_mgr localhost --bridge_name br0 --target_device eth1 --no_cleanup

Into your terminal, remembering to change the ssid, passwd, and security fields to match your network credentials.

- C. create_bridge requires the following arguments:
 - A. bridge_name Name of the bridge to create
 - B. target_device which device the bridge is going to connect

D. It is not valid to add stations to a bridge, they don't work like you would expect. Bridges can have: eth ports, redirects (rdd) ports, vaps, and qvlan ports. Ports in a bridge cannot have IP addresses.

9. VAP Creation

	and a												 Mate Terminal
				LANfor	ge Mar	ager Vers	ion(5.4.3)				• •	File Edit View Search Terminal Help
VoiP/RTI Statu	P Ende	Port Mgr	ource La	Chamber View Mgr DUT Profiles yer-3 L3 End	ps	Loyer 4-7	erts Warr	mageddon	wfi-Me	ssages WanUnks	Sefresh VoP/	HELP	(base) [Landrage[H389-d8b; p-script]s //create_station.pyradio wiph@- sit Landrage] passud passedscript ys //create_station.pyradio wiph@- laseD[Landrage[H389-d8b; p-script]s //create_bridge.pytanget_device w rescript].tandrage[H389-d8b; p-script]s //create_yap.yrradio wiph@sec
Disp	: 192.	168.1.4:1		Sniff Packets	Dow	n 1 Clear	Counters	Reset	Port	Delete			password
Ppt	Timer:	medium (8 s)	•	Apply	VRF		isp]ay	Crga	ste	Modify	Batch Modil	fy	Lanforge Creating VMPs
		1		Al Ethero	at Interfa	ces (Ports) for	all Resource	es.			1		Creating VAPs
Port	0 1	Р	SEC	Alias	Parent Dev	RX Bytes	RX Pkts	Pps RX	bps RX	TX Bytes	TX Pkts	Pps T	(base) [lanforga@if0350-c8dc py-scripts]\$ []
1.1.00		192.168.1.4	0	eth0		15,879,252			18,832				
1.1.01		192.168.42.7	8	eth1 eth2		149,792,808	99.018	0			0 148		
1.02		0.0.0.0	6	wiphy0		172,565,436				151.702.80			
1.1.04		0.0.0.0	0	wiphy1		0		0	0		0 0		
1.1.05	1	0.0.0.0	0	wten0	wphy0	0	0	0			0 0		
1.1.07		0.0.0.0	0	wian1 sta0000	wphyl	0	0	0			0 0		
1.1.10		0.0.0.0	6	sta0000	wphy0 wphy0		0	0			0 0		
1.1.12	- í	0.0.0.0	ő	br0	mpinjo			0					
1.1.14		0.0.0.0	0	1.1.vap0000	wiphy0	0	0	0	0	3,56	8 35		
1.1.16		0.0.0.0	0	1.1.vap0001	wphy0	0	0	0	0	17,51	2 70		
												•	
4													

- A. You can create a VAP from your terminal with the create_vap.py script.
- B. Type
 - ./create_vap.py --radio wiphy θ --security wpa2 --ssid lanforge --passwd password

into your terminal, remembering to change the ssid, passwd, and security fields to match your network credentials.

- C. create_vap supports the following flags:
 - A. num_vaps A user defined number of VAPs to create off of your antenna
 - B. upstream_port if your ethernet cable to your router is not eth1, define it using this flag. Eth1 is the default for this flag.
 - C. debug call this flag if you want detailed diagnostic information
- 10. MAC-VLAN Creation
 - A. In the py-scripts folder, there is a script named create_macvlan,py, which creates a mac-vlan based on the macvlan_parent , num_ports , first_macvlan_ip, netmask, and gateway input
- 11. Monitor Creation: Under Construction
- 12. Layer-3 Cross Connect
 - A. In your py-scripts, there is a create_I3 script which allows you to create stations in your terminal.
 - B. Type
 - ./create_l3.py --radio wiphy0 --num_stations 0

into your terminal, remembering to change the ssid, passwd, and security fields to match your network credentials.

- C. This will create a layer3 cross connect between each station you already built on your lanforge device and your wiphy0 radio. You might need to wait 30 seconds for them to stop being phantom ports. If you do not specify num_stations 0 it would have created two stations by default off your specified radio. You can change which port the cross connect will be connected to with the --upstream_port option.
- D. You can specify the following flags as well:
 - A. num_stations Specify a different number of stations to create off of your antenna.
 - B. upstream_port if your ethernet cable to your router is not eth1, define it using this flag. Eth1 is the default for this flag.
 - C. debug call this flag if you want to get error messages in case anything goes wrong.
- E. Your port manager will look similar to this

			stern 🔓	i 🖂 🖼 ڬ		2							🤹 Mon jan 11, 13
Mel Mel	te Terr	minal		2 V	Nicrge Mar	nager Versio	(0(5.4.3)	-		_	_		
e													 Mate Terminal
					forge Man	nager Vers	sion(5.4.3	0					File Edit View Search Terminal Help
orel Br	eportin	ng Winday	otri za										<pre>(base) [lanforge[lf0350-c8dc py-scripts]s ./create_station.pyradio wiphy0 - sig lanforgepassed passendsecurity epaz</pre>
				Chamber \Sew		Stop All		estart Mani			stresh	HELP	Creating stations
				rce Mgr DUT Prot				nings `+					(base) [lasforge[Lf0350-c8dc py-scripts]\$./create_bridge.pytarget_device w r0redip wight]
Status		Port Mgr			indps	Layer 4-7		mageddon		WanLinks	VOIP/F	RTP	(base) [Lanforge][10359-c8dc py-scripts]8 ./create_vap.pyradio wiphy8sec ity wpa2ssid Lanforgepassed passeord
		68.1.41		Snfl Packets	E Down		r Counters	Reset		Delete			password
Rpt Tir	men a	edium (8	5) 🔻	Δρφλγ	VRF		Display	Creat	ste	Hogity	Batch Modify	NY .	lanforge Creating VWp
	1.1			All Ethe	met interfa	aces (Ports) for	r all Resource	ces. —				_	Creating VMPs
ort o	οI	P		SEC Alas	Parent	RX Puters	RX Picts	Pps RK	bps RX	TX Puters	TX Pids	PosT	(base) [lanforge[tf0350-c8dc py-scripts]5 ./create_l3.pyradio wiphy0num_ ations 0
					Dev								Creating stations
00 01		192.168.1.				15.954.312 149.792.808			5 25,387				(base) [lanforge](f0350-c0dc py-scripts)5 [
12	1	0.0.0.0	4	eth2		0	0 0	i i	0 0	0 0	0		
03 D4		0.0.0.0				173,602,978				9 153,702,858			
D4 D6		0.0.0.0	- 6		wiphyo								
07	1	0.0.0.0	- 6		wiphy2								
10	1	0.0.0.0	6	sta0000	wiphy0	0			0 0	0 0	x 0		
	1	0.0.0.0		ste0001	wiphy0	0							
12		0.0.0.0			wiphyQ								
16		0.0.0.0	- 6		wiphyo								
ed in te	n p: loca	alhost 4002	2 31: 6	dmin							4 stations: D	×	

F. Your Layer 3 connections will look similar to this

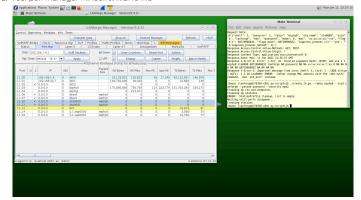
Latings Parager Mainting Parager </th <th>-</th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th>Mate Terminal</th>	-										Mate Terminal
Owner type Degree Appendix Degree Appendix <	0			LANforge	Manager V	Persion(5.4.3)				\odot \otimes \otimes	File Edit View Search Terminal Help
	AGP/ITTP End Status Rpt View Name Tsta0000-0	ps Pile-ID Port Mgr Timer: fast v 0-500 Type 9 UAUDP Step	Char tensource Mgr C Layer-3 (L s) • Go ste Pkt Rx A	rber tiew IUT Profiles I 3 Endps Test Manager a Cross Conr Pit Px 8 0	Traffic Profiles Layer 4 d • Nects for Selecto Rps Rx A	All Rost Alerts Warmin 1-7 Arms Select All Disp ed Test Manager- Aps Rx 8	rgs + Wil ageddon Start + Jay Crgate RicDrop % -	WarLin Stop - Qui Modify Rc Drop % 6	esce Delete	Drop Picts	Theory Clarkrepe (1993) de lay systematic J. creates statistics or - rates scape to later the statistical statistics of the statistical statistics of the statistics of the mean statistics of the statistics of t

13. Layer-4 Cross Connect

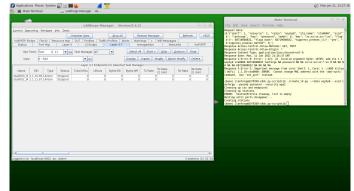
A. In your py-scripts, there is a create_14 script which allows you to create stations in your terminal.
 B. Type

./create_l4.py --radio wiphy0 --ssid lanforge --passwd password --security wpa2

- into your terminal
- C. This will automatically create 2 stations on your Lanforge device off of your wiphy0 radio and also create a cross connect from each station to your eht1 port. You can change which port the cross connect will be connected to with the --upstream_port option.
- D. You can specify the following flags as well:
 - A. num_stations Specify a different number of stations to create off of your antenna.
 - B. upstream_port if your Ethernet cable to your router is not eth1, define it using this flag. Eth1 is the default for this flag.
 - C. debug call this flag if you want detailed diagnostic information
- E. Your port manager will look similar to this



F. Your Layer 4 connections will look similar to this



- G. Your Netsmith display will look similar to this. The connections on this picture have been oriented for legibility.
- 14. Monitor and record an IPV4 variable time test. The purpose of this test is to detect whether your router is able to keep a steady signal when being barraged by multiple users.
 - A. In the first part of this tutorial, you are going to connect various numbers of stations, record them for 1 and 10 minutes, and then save as a CSV, excel, or pickle format. A station represents a device which is connected to a network, LANforge creates representations of stations which create real traffic on your network and then records statistics on that traffic. This module determines whether your device is able to sustain a heavy load of traffic or an user defined period of time. By recording the traffic it is then possible to go back and detect where any problems occurred which allows a network manager to fix problems which could be facing your network.
 - B. Navigate to the py-scripts folder and type the following command into your command line ./test_ipv4_variable_time.py --radio wiphy0 --security wpa2 --ssid lanforge -password password --output format csv

Replace the security, ssid, and password variables with the settings for the network you are testing. This will create 2 wiphy stations by default, connect them to the network you are testing, and report the results to a CSV file. You will find a file with a timestamp within the last 5 minutes in the report_data folder in your home directory. If you are running this script from another machine using the --mgr function, you will need to define the report_file variable.

- C. test_ipv4_variable_time accepts the following flags:
 - A. output_format The format you want to save your results to
 - B. col_names Which columns should be saved in the output file
 - C. test_duration how long you want the test to last.
 - D. report_file where you want the results to be stored
- D. Your port manager will look similar to this

Applications Place Mate Terminal		Jue Manager	. 🖻 [report-data]		
LANforge Man		sio • • •	1	■ Take Screensho e Terminal	۲
File-ID Resource Nor DUT Profiles Traffic- Status Port Nor Layer-3 L3 Endps Lay	top Al Destart Manager Profiles Alerts Warnings + yer 4-7 Armageddon Warta	Befresh HELP Wit-Messages ks VolPRTP Endps Overst Over	File Edit View Searc (base) (lanforge@lf0350-c8dc py-script security wpa2ssid lanforgepass Creating stations	h Terminal Help s)s ./test_ip/4_variable_time.pyradi word passwordoutput_format_csv	io wiph
Rpt Timer: #edium (8 i Apply Al Ethernet in	Down I Clear Counters Wer I Display Interfaces (Ports) for al Resources las Parent RX Bytes	Reset Port Delete Organe Mogify Batch M Rx Hits Pps Rx bps Rx Tx	Waiting for ips, timeout: 60 Starting CXs		
1.1.00 192168.1.4 0 eth0 1.1.01 192168.0.7 0 eth1 1.1.02 0 0.0.0 0 eth2 1.1.03 0.0.0.0 0 weby0 1.1.04 0.0.0.0 weby0 1.1.04 0.0.0.0 weby0	13,650,235 16,012,952	175,895 22 13,263 128.1 10,649 0 0			
11.00 392168.1.4 0 eth 11.01 392168.4.7 0 eth 11.02 # 0.0.8 0 wfrgd 11.03 # 0.0.4 0 wfrgd 11.04 # 0.0.4 0 wfrgd 11.05 # 0.0.4 # wfrgd wfrgd 11.05 # 0.0.2 # wfrgd wfrgd 11.10 # 0.0.2 # wfrgd wfrgd 11.121 172.16.42.52 # wfrgd wfrgd	284,745,280 1.579,923 wphy0 2.090 wphy0 0 wphy1 0	6.931.005 0 0 : 8 0 1.119 0 0 0			
1.1.11 172.16.42.52 D stw0001	wiphy0 2,256	10 0 1.228			
			Þ		
		A stations 21 21 00			
Your Layer 3 connec	tions will look				
Applications Place Mate Terminal			<u>₹</u> □ [report-data]	I Fri Jan 8, 12: ■ Take Screenshop	
Control Reporting Windows and Tests	LANfo	orge Manage	r Version(5.4.3)		
Status Port Hgr Layer-3 L3 Endps Layer 6	4-7 Armageddon Wani,inks Apt Timer: fast (1 s	volPJRTP VolPJRTP Endps FileH0	Chamber year 20 Resource Mgr DUT Profiles Traffic Profiles Ale Select All Start + 20p - Quesce	p Al Restart Manager Befree erts Warnings + Wit-Messages Clear	**
Name Type State Pit Rr A Pi	View 0 - 500	Cross Connects for S	Display, Crgate Modify Dele elected Test Manager xop Mits A Drop Hits B Avg RTT Timer ED		
Name Type State Pit Rx A Pi VTsta0100-0 (F/UDP Run 0 VTsta0201-1 (F/UDP Run 0	81 Pix 8 8ps Pix A 8 3,834 0 3,931 0	ps Rx B Rx Drop % A Rx Drop % B D 255,985 0 0 255,957 0 0		Endpoints (A => B) tao1001-0-4 -=> VTsta0100-0-B tao1001-1-4 -=> VTsta01001-1-B	
			4		
ogged in to: localhost 4002 as: Admin		ailar to this		4 gtz	ations: 2
(our Layer 3 endpoir		III ڬ 📰		۹۳۷ 40 Fri Jan 8, 12:	
Your Layer 3 endpoir Applications Place	es System 📄		anager Versi 🛚 [re	≪ Fri Jan 8, 12:	
Your Layer 3 endpoir Applications Place Mate Terminal corrol Beoring Windows into Seas	Es System LANfa	LANforge Manage	Anager Versi © [re r Version(5.4.3)	<pre></pre>	:08 ©
four Layer 3 endpoir Applications Place Mate Terminal Control Besofing Windows (No Sets	es System 📄 LANfo	Meine Manage	anager Versi • [re r Version(5.4.3)	Fri Jan 8, 12: eport-data]	:08 ©
Applications Place Applications Place and Applications Place and Applications Place and Applications Place and Applications a	LANKC	LANforge Ma Lanforge Ma Drge Manage Marsey Manage Marsey Manage Marsey	Anager Versi © [re version(5.4.3) Saure Mp Orf Perks Tationals Sa Saure Mp Orf Perks Tationals Saure Saur	40 Fri Jan 8, 12: eport-data]	:08 ©
(our Layer 3 endpoir Applications Place Mate Terminal Control Basena Weber 76 (José Seen Perky: Layer 3 (1366)	Es System		Anager Versi © [re r Version(5.4.3) Basers by Odf Preting Trail.Porting for the Section	40 Fri Jan 8, 12: 2port-data] pd. fauta Menage pd. fauta Menage	:08 。
Cour Layer 3 endpoin	Ess System Image: Control of the system LANFc Lanfc 47 Amageddan Warchoka 49 Poll Star Market 140 Star Te Ratio 15 Te Ratio Te Ratio 2555.00 256.00 256.00		Anager Versi © [re r Version(5.4.3)	40 Fri Jan 8, 12: eport-data] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• • • • •
Your Layer 3 endpoin Applications Place Mate Terminal Correl deserting windows phil laws Seen 1 Per High Support Laws News 20 mar/wg Support	Ess System Image: Control of the system LANFc Lanfc 47 Amageddan Warchoka 49 Poll Star Market 140 Star Te Ratio 15 Te Ratio Te Ratio 2555.00 256.00 256.00		Anager Versi © [re r Version(5.4.3)	40 Fri Jan 8, 12: eport-data] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• • • • •
Cour Layer 3 endpoin	Ess System Image: Control of the system LANFc Lanfc 47 Amageddan Warchoka 49 Poll Star Market 140 Star Te Ratio 15 Te Ratio Te Ratio 2555.00 256.00 256.00		Anager Versi © [re r Version(5.4.3)	40 Fri Jan 8, 12: eport-data] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• • • • •
Your Layer 3 endpoin Applications Place Mate Terminal Correl deserting windows phil laws Seen 1 Per High Support Laws News 20 mar/wg Support	Ess System Image: Control of the system LANFc Lanfc 47 Amageddan Warchoka 49 Poll Star Market 140 Star Te Ratio 15 Te Ratio Te Ratio 2555.00 256.00 256.00		Version(5.4.3)	40 Fri Jan 8, 12: eport-data] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• • • • •
Your Layer 3 endpoin Applications Place Mate Terminal Correl deserting windows phil laws Seen 1 Per High Support Laws News 20 mar/wg Support	Ess System Image: Control of the system LANFc Lanfc 47 Amageddan Warchoka 49 Poll Star Market 140 Star Te Ratio 15 Te Ratio Te Ratio 2555.00 256.00 256.00		Version(5.4.3)	40 Fri Jan 8, 12: eport-data] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	• • • • •
Your Layer 3 endpoint Applications Place Mate Terminal	Es System		version (5.4.3)	** Fri Jan 8, 12: copt.data] **	:08
Cour Layer 3 enclosin Applications Place Mate Terminal Court Breach where yet just Mate Terminal Court Breach where yet just Mare to provide yet just	es System LANfe Character the role of the role o	LANFORGE MA ALANFORGE MA ALA	Version(5.4.3)	** Fri Jan 8, 12: sport-data] *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** **** ****	208
Cour Layer 3 endpoint Applications Place Mate Terminal Mate Ter	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	** Fri Jan 8, 12: copt.data] **	:08
Cour Layer 3 endpoint Applications Place Mate Terminal Mate Ter	es System LANFe LANFe Low the second secon	AANforge Manage Aanage	version(5.4.3)	** Fri Jan 8, 12: cport-data] ************************************	ations : 2
Cour Layer 3 enclosin Applications Place Mate Terminal	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	** Fri Jan 8, 12: cport-data] ************************************	ations : 2
Applications Place Applications Place Applications Place Terminal Applications Place Appl	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	40 Fri Jan 8, 12: eport-data] edit manager	:08
Applications Place Applications	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	40 Fri Jan 8, 12: eport-data] edit manager	:08
Your Layer 3 endpoint Applications Place Mate Terminal The Mate Terminal The Mate Terminal The Mate Terminal The Mate Terminal Compared to Mater State Applications Place Applications Place Mate Terminal Mate Terminal	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	** Fri Jan 8, 12: cport-data] ************************************	:08
Applications Place Applications Place Mate Terminal Mate Terminal Mate Terminal Mate Terminal Mate Terminal Mate Terminal Mate Terminal Mate Terminal Mate Terminal Applications Place Mate Terminal	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	40 Fri Jan 8, 12: cort-data] 200	• ***
Applications Place Applications Place Mate Terminal Mate Terminal	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	version(5.4.3)	4) Fri Jan 8, 12: cport-data] add add add bit bit <td>2008: () () () () () () () () () ()</td>	2008: () () () () () () () () () ()
Applications Place Applications Place Mate Terminal Office Destrict, when yo loss Term Terminal Constructions Place Term Terminal Constructions Place Constructions Place Applications Place Mate Terminal Applications Place Mate Terminal	es System LANFe LANFe Low the second secon	LANFORGE MA CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD Calls Intel CANFORD CALLS CANFORD CALLS CA	b sn creating traffic. If we eports	4) Fri Jan 8, 12: sport-data] 4) Interference 1) Interference </td <td>2008: () () () () () () () () () ()</td>	2008: () () () () () () () () () ()
Applications Place Applic	es System LANFe LANFe Low the second secon	LANFORGE MANAGE ACTION AND A CONSTRUCTION OF A	version(5.4.3)	40 Fri Jan 8, 12: eport-data] edit interget internet	:08

Querying the LANforge JSON API using Python

Goal: Use Python scripts to query the LANforge Client JSON API. (See Querying the LANforge GUI for JSON Data) The provided Python scripts allow you the same API scope as the Perl scripts.

LANforge now provides Python scripts that query the REST API that the LANforge Client now exposes by default. This chapter steps through using each of the scripts. At the end we show an example of how to write a Python script. Scripts encourage Python 3. Requires LANforge 5.4.1 or later.

Client Settings

On your LANforge Server, the scripts directory is located at /home/lanforge/scripts. Under that directory, the py-json directory contains Python scripts.

Script	Purpose
initpy	Defines the py-json module
show_ports.py	Simplest example of querying all ports. This is typical of querying any of the APIs you browse from / or /help
create_sta.py	Script creates a virtual station
LANforge/initpy	Defines the py-json/LANforge module
LANforge/LFRequest.py	Use the LFRequest module to help make GET and POST requests to the LANforge Client
LANforge/LFUtils.py	Use the LFUtils module to help process JSON results

Getting Started

First, start the LANforge Client (LANforge GUI) and connect it to your LANforge Server. If you want to start the client in headless mode, open a terminal, and from the LANforgeGU_5.4.1 directory, start the script with the daemon argument:

\$./lfclient -daemon -s localhost

Querying Ports

Running the script

In the /home/lanforge/scripts/py-json directory:

<pre>\$ python3 ./show_ports.py</pre>	
{ '1.1.eth0': { '_links': '/port/1/1/0',	
'alias': 'eth0',	
'phantom': False,	
'port': '1.1.00'}}	
{ '1.1.ethl': { '_links': '/port/1/1/1',	
'alias': 'ethl',	
'phantom': False,	
'port': '1.1.01'}}	
{ '1.1.sta00500': { '_links': '/port/1/1/9',	
'alias': 'sta00500',	
'phantom': False,	
'port': '1.1.09'}}	
{ '1.1.sta00501': { '_links': '/port/1/1/10',	
'alias': 'sta00501',	
'phantom': False,	
'port': '1.1.10'}}	
{ '1.1.sta00502': { ' links': '/port/1/1/11',	
'alias': 'sta00502',	
'phantom': False,	
'port': '1.1.11'}}	

Looking inside the script

This script is a way of pretty-printing the results of GET http://localhost:8080/port/1/1/list

```
lf_r = LFRequest.LFRequest("http://localhost:8080/port/1/1/list")
json_response = lf_r.getAsJson()
j_printer = pprint.PrettyPrinter(indent=2)
for record in json_response['interfaces']:
        j_printer.pprint(record)
```

Other variations of this you can try are: /port/list This is an abbreviation /port/1/2/list If you have a second LANforge resource /port/1/2/eth0 Shaw a specific port

Example of Creating a Station

lanforge@ct524-debbie py-json]\$ python3 ./create_sta.py
<pre>xample 1: will create stations sta0200,sta0201,sta0202</pre>
x 1: Checking for station : http://localhost:8080/port/1/1/sta0200
x 1: Next we create stations
x 1: Next we create station sta0200
1: station up sta0200
Example 2: using port list to find stations
x 2: checking for station : sta0220
x 2: create station sta0220
1x 2: set port sta0220
xample 3: bring ports up and down
x 3: setting ports up
x 3: setting ports down
ports are down
Example 4: Modify stations to mode /a
sing add_sta to set sta0200 mode
xample 5: change station encryption from wpa2 to wpa3
sing add_sta to set sta0200 wpa3
xample 7: alter TX power on wiphy0

Looking inside the script

Create a station

Flags are a decimal equivalent of a hexadecimal bitfield you can submit as either 0x(hex) or (dec) a helper page is available at http://localhost:8080/help/add_sta

You can watch console output of the LANforge GUI client when you get errors to this command, and you can also watch the websocket output for a response to this command at ws://localhost:8081. Use \$ wsdump ws://localhost:8081/ to follow those messages.

Modes are listed at http://localhost/LANforgeDocs-5.4.1/lfcli_ug.html or at https://www.candelatech.com/lfcli_ug.html

The MAC address field is a pattern for creation: entirely random mac addresses do not take advantage of address mask matchin in Ath10k hardware, so we developed this pattern to randomize a section of octets:

ХХ

keep parent

randomize

chars [0-9a-f]

use this digit

If you get errors like "X is invalid hex character", this indicates a previous m_vlan call has not removed your station yet: you cannot rewrite mac addresses with this call, just create **new** stations.

The **staNewDownStaRequest()** creates a station in the Admin-Down state. This is a good way to efficiently create batches of stations because it defers all the PHY layer activity which takes significant time when you do it in a loop.

lf_r.addPostData(LFUtils.staNewDownStaRequest(sta_name, resource_id=resource_id, radio=radio, ssid=ssid, passphrase=passphrase) lf_r.formPost() sleep(0.05)

Sleeping for 50ms is not sufficient to interact with the station, but is a functional minimum to allow the LANforge to start processing the command; this is a good value to use in a loop that creates stations. Follow with:

LFUtils.waitUntilPortsAppear(resource_id, desired_stations)

Set station up

The LANforge API separates STA creation and Ethernet port settings. We need to revisit the stations we create and amend flags to add things like DHCP or ip+gateway, admin-{up,down} for sta_name in desired_stations:

```
lf_r = LFRequest.LFRequest(base_url+"/cli-json/set_port")
data = LFUtils.portDhcpUpRequest(resource_id, sta_name)
lf_r.addPostData(data)
lf_r.jsonPost()
sleep(0.05)
```

Set station down

Change station mode

There is not a set_sta command. Many LANforge CLI commands do a default modify if the entity already exists. This is how we can modify attributes of existing stations. For the mode values, see http://www.candelatech.com/lfcli_ug.php#add_sta

Change station protocol

Flags for add_sta and set_port are actually 64-bit values. When the values in the command below are read by the /he1p/add_sta page, Javascript cannot deal with integers greater than 32-bits long.

```
lf_r = LFRequest.LFRequest(base_url+"/cli-json/add_sta")
lf_r.addPostData({
    "shelf":1,
    "resource": resource_id,
    "radio": radio,
    "sta_name": sta_name,
    "mode": 0, # mode AUTO
```



Change radio power on radio wiphy0

Virtual stations do not have individual tx power states. You can set the radio transmit power. See http://www.candelatech.com/lfcli_ug.php#set_wifi_radio. The txpower is set through iwconfig, so see man 8 iwconfig. Power is in dBm, auto or off.

Not all flags in a JSON request are actually LANforge CLI parameters. The suppress_preexec_method parameter is a meta-flag tells the LANforge client to not check that the port exists before issuing the command. You would use this to expedite a script, because a check-port command is synchronous, not intended to be used in a loop.

lf r = LFRequest.LFRequest(base url+"/cli-json/set wifi radio") lf_r.addPostData({
 "shelf":1,
 "resource":resource_id,
 "radio":radio, "mode":NA, "txpower": "auto", "suppress_preexec_method": "true" }) lf_r.jsonPost()

Seeing Errors

Monitoring for Connection Errors

Use the wsdump utility on the LANforge to see the output of system errors and WiFi Events:

\$ wsdump ws://localhost:8081/

The output will be mostly similar to what you see in the WiFi-Messages tab in the GUI:

< ("wifi-event":"1.1: IFNAME=sta0200 <3>CTRL-EVENT-SCAN-STARTED","timestamp":"2019-11-21T16:00:50.0952")
< ("wifi-event":1.1: IFNAME=sta0200 <3>CTRL-EVENT-NETWORK-NOT-FOUND","timestamp":"2019-11-21T16:00:50.0952")
< ("wifi-event":1.1: sta0200 (pby #1): scan finiahed: 5745, \"\"," "timestamp":"2019-11-21T16:00:50.0952")
< ("wifi-event":1.1: sta0200 (pby #1): scan finiahed: 5745, \"\"," "timestamp":"2019-11-21T16:00:50.0952")
< ("wifi-event":1.1: IFNAME=sta0220 <3>CTRL-EVENT-NETWORK-NOT-FOUND","timestamp":"2019-11-21T16:00:50.0952")
< ("wifi-event":1:1: IFNAME=sta0220 <3>CTRL-EVENT-NETWORK-NOT-FOUND","timestamp":"2019-11-21T16:00:50.0952")
< ("wifi-event":1:1: IFNAME=sta0220 <3>CTRL-EVENT-NETWORK-NOT-FOUND","timestamp":"2019-11-21T16:00:50.0952")

The message CTRL-EVENT-NETWORK-NOT-FOUND indicates that the SSID we are attempting to connect to is unavailable.

Interpreting Python HTTP Error Output

It won't be uncommon to find errors similar to this:

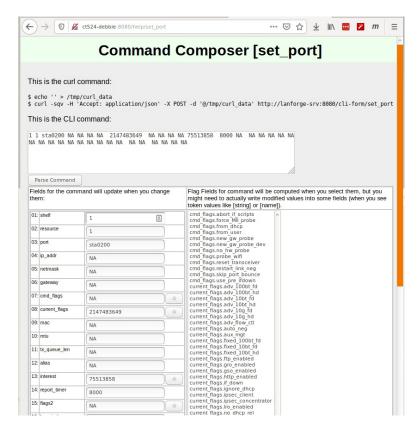
Url: http://localhost:8080/cli-form/set_port
Error: <class 'urllib.error.httperror'=""></class>
Request URL:
'http://localhost:8080/cli-form/set_port'
Request Content-type:
'application/x-www-form-urlencoded'
Request Accept:
'application/json'
Request Data:
(b'shelf=1&resource=1&port=sta0200¤t_flags=2147483649&interest=75513858&r'
b'eport_timer=8000')

The HTTPError exception is just some kind of 500 error and is often **timing** related. Perl scripts are subject to similar timing issues. When LANforge is busy creating and destroying stations, it is modifying the network stack during each modification...and this takes time.

You can decode this set_port request data by pasting the individual values into the /help/set_port page provided by your LANforge client: http://localhost:8080/help/set_port.

- shelf 1
- resource 1
- port sta0200
- current_flags 2147483649
- interest 75513858
- report timer 8000

For numerical flag fields, you can use the G button to try and decode the values of the flags.



Using the Scripts on Your Laptop

You can copy the **py-json** directory to your laptop or workstation. You may also use **git** and clone the LANforge-scripts repository: https://github.com/greearb/lanforge-scripts/.

1 If you make a script on a Windows laptop and copy it back to your LANforge, please run **dos2unix** on the script to change the line-ending characters: **\$**

dos2unix myscript.py

Python Module Methods

LFRequest.py

Create a new LFRequest object to help create a request:

lf_r = LFRequest.LFRequest(base_url+"/port/1/1/wiphy0") wiphy0_json = lf_r.getAsJson()

Your REST requests are discussed in the Querying LANforge GUI for JSON Data chapter.

formPost(show_error=true)

This method formats post data as application/x-www-form-urlencoded data. There should be no significant differience between this and the jsonPost() method.

jsonPost(show_error=true)

This method formats post data as application/json data. There should be no significant difference between this and the formPost() method.

get(show_error=true)

Use this method to do a GET request with 'Accept: application/json' headers. You get unformatted results.

getAsJson()

Formats the results of get() into Objects using json.loads()

LFUtils.py

def staNewDownStaRequest(sta_name, resource_id=1, radio="wiphy0", flags=ADD_STA_FLAGS_DOWN_WPA2, ssid="", passphrase="", debug_on=False):

For use with add_sta. If you don't want to generate mac addresses via patterns (xx:xx:xx:81:*) you can generate octets using random_hex.pop(0)[2:] and gen_mac(parent_radio_mac, octet) See http://localhost:8080/help/add_sta

def portSetDhcpDownRequest(resource_id, port_name, debug_on=False):

Sets port admin down. See http://localhost:8080/help/set_port

def portDhcpUpRequest(resource_id, port_name, debug_on=False):

Sets port up and to use DHCP. See http://localhost:8080/help/set_port

def portUpRequest(resource_id, port_name, debug_on=False): Sets port up. See http://localhost:8080/help/set_port

def portDownRequest(resource_id, port_name, debug_on=False):

Sets port down. Does not change the use_dhcp flag See http://localhost:8080/help/set_port

def generateMac(parent_mac, random_octet):

Helps generate a random mac address.

def portNameSeries(prefix="sta", start_id=0, end_id=1, padding_number=10000):

This produces a named series similar to "sta000, sta001, sta002...sta0(end_id)" The padding_number is

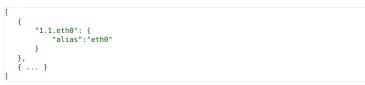
added to the start and end numbers and the resulting sum has the first digit trimmed, so $f(0, 1, 10000) \Rightarrow \{0000, 0001\}$

def generateRandomHex():

Use in conjuction with generateMac()

def portAliasesInList(json_list):

Return reverse map of aliases to port records. Normally, you expect nested records, which is an artifact of some ORM that other customers expect:



Naturally, this is more difficult to digest. This method returns a more intuitive structure:



def findPortEids(resource_id=1, port_names=(), base_url="http://localhost:8080"):
 returns PortEID objects matching requested port_names. Use after set_port

def waitUntilPortsAdminDown(resource_id=1, port_list=()):

Sleep and query until all ports report admin down. Use after set_port

- def waitUnfilPortsAdminUp(resource_id=1, port_list=()):
 Sleep and query unfil all ports report admin up. Use after set_port
- def waitUntilPortsDisappear(resource_id=1, port_list=()):
 - Sleep and query until requested ports have entirely gone away. Use after rm_vlan

def waitUntilPortsAppear(resource_id=1, port_list=()):

Sleep and query until requested ports have appeared. Use after add_sta

Managing WAN Links Using LANforge JSON API

Goal: Create and modify WAN Links Using LANforge JSON API. This cookbook provides examples in Python. (See Querying the LANforge GUI for JSON Data) The provided Python scripts allow you the same API scope as the Perl scripts.

This chapter steps through using Python scripts to create and manage WAN Links on a LANforge. Scripts require Python 3. Requires LANforge 5.4.1 or later. Examples require CT910, CT521a, or better.

Creating a WAN Link

We will start by creating a WAN Link between ports **eth2** and **eth5** on our ct522. Two Ethernet ports will be involved in this example.

The create_wanlink.py Script

This script is located in /home/lanforge/scripts/py-json or on the lanforge-scripts github page. You can copy this script to a new name and edit it to fit your environment. Remember, these JSON scripts will be querying a LANforge Client (GUI or headless). The URL you will see being queried is going to be http://localhost:8080/ for these exaples, assuming you are running the LANforge client on the same machine you are running your script.

This script performs the basic tasks you might use to manage WANlink connections:

- Listing existing WANlinks
- Removing a WANlink if it exists.
- Creating WANlink endpoints. You want to create the endpoints before creating the connection.
- Joining WANlink endpoints into a WANlink connection (CX)
- Starting the WANlink
- Modifying the WANlink. You can set endpoint tx rates and lossiness parameters while the endpoints are running.
- Stopping the WANlink
- Listing WANlink endpoint stats

Script Sections Explained

1. Listing Wanlinks

Notice that when we get a listing response, we are looking for items in the response that are dictionarys with a <u>links</u> key/value pair. There are other key/values used for diagnostics, such as uri, handler, warnings and errors.

base_url = "http://localhost:8080"

```
json_post = ""
json_response = ""
num_wanlinks = -1
# see if there are old wanlinks to remove
lf_r = LFRequest.LFRequest(base_url+"/wl/list")
try:
    json_response = lf_r.getAsJson()
# For debugging, this makes json response more legible:
    LFUtils.debug_printer.pprint(json_response)
    for key,value in json_response.items():
        if (isinstance(value, dict) and "_links" in value):
            num_wanlinks = 1
except urllib.error.HTTPError as error:
    num_wanlinks = 0;
```

JSON output

If there are no wanlinks, you will only see a warnings block telling you there are connections found that don't apply as WANlinks:

```
{
    "warnings" : [
    "HttpWl::selectColumnsFromRow: eid not in table: Cross-Connect cx_id: 17, type: 1, idx: -1",
    "HttpWl::myEvaluateGet: EidCx type 1 (LANforge / UDP) unavailable in WL table: 17.1",
    "HttpWl::selectColumnsFromRow: eid not in table: Cross-Connect cx_id: 18, type: 1, idx: -1",
    "HttpWl::myEvaluateGet: EidCx type 1 (LANforge / UDP) unavailable in WL table: 18.1"
    "handler" : "candela.lanforge.GenericJsonResponder",
    "uri" : "wl/:wl_id"
}
```

2. Removing a WANlink

If we found WANlinks, we can remove them by posting the data to the corresponding CLI command URIs: /cli-json/rm_cx and /cli-json/rm_endp.

```
Remember the naming convention: Layer-3 and WANlink endpoints end
with A and B. A WANlink named westin500 has endpoints named
westin500-A and westin500-B.
```

```
if (num_wanlinks > 0):
    lf_r = LFRequest.LFRequest(base_url+"/cli-json/rm_cx")
    lf_r.addPostData({
        'test_mgr': 'all', # could be 'default-tm', too
        'cx_name': 'wl_egl'
    })
    lf_r.jsonPost()
    sleep(0.05)
```

The parameters for each command can be found via the help page:

http://localhost:8080/help/rm_cx. Notice that slight pause between commands: 50ms is a good idea between deletion commands.



3. Creating WANLink Endpoints

Create the two endpoints first. Each side of a WANlink has its own transmission rate, buffer size and corruption parameters. Each WANlink requires an ethernet port. Side A will be 128,000bps with 75ms latency:

<pre>lf_r = LFRequest.LFRequest(base_url+"/cli-json/add_wl_endp")</pre>
lf_r.addPostData({
'alias': 'wl eg1-A',
'shelf': 1,
'resource': '1',
'port': 'eth3',
'latency': '75',
'max rate': '128000',
'description': 'cookbook-example'
})
lf r.jsonPost()
sleep(0.05)

Side B will be 256,000bps with 95ms latency:

<pre>lf_r = LFRequest.LFRequest(base_url+"/cli-json/add_wl_endp")</pre>
lf_r.addPostData({
'alias': 'wl_eg1-B',
'shelf': 1,
'resource': '1',
'port': 'eth5',
'latency': '95',
'max_rate': '256000',
'description': 'cookbook-example'
})
lf_r.jsonPost()
sleep(0.05)

4. Create the WANlink

Creating the WANlink is simple, we will add it to the default test manager default-tm:

ι	f_r = LFRequest.LFRequest(base_url+"/cli-json/add_cx")
l	f_r.addPostData({
	'alias': 'wl_eg1',

```
'test_mgr': 'default_tm',
    'tx_endp': 'wl_eg1-A',
    'rx_endp': 'wl_eg1-B',
})
[f_r.jsonPost()
sleep(0.05)
```

5. Start the WANLink

The LANforge server is very asynchronous. Before immediately changing the state on a connection or endpoint, test to see that it exists.

Polling for the WANlink

Note how we can request fields by name, in this case name, state, and _links.

```
try:
    json_response = lf_r.getAsJson()
    if (json_response is None):
        continue
```

If there is no response, or we get a 400 error, the wanlink has probably not finished creating. Our response will be **None**. In the reponse below, we're testing for dict entries that have the key **_____inks** in them. If the name value matches, our WANlink has been created:

```
for key,value in json_response.items():
    if (isinstance(value, dict)):
        if ("_links" in value):
            if (value["name"] == "wl_eg1"):
                seen = 1
```

It might be helpful to use these else clauses when getting started:

#else:
<pre># print(" name was not wl_eg1")</pre>
#else:
<pre># print("value lacks _links")</pre>
#else:
<pre># print("value not a dict")</pre>
<pre>except urllib.error.HTTPError as error: print("Error code "+error.code) continue</pre>

Change the WANLink State

Starting and stopping connections is done by changing the state:

```
lf_r = LFRequest.LFRequest(base_url+"/cli-json/set_cx_state")
lf_r.addPostData{
    'test_mgr': 'all',
    'cx_name': 'wl_eg1',
    'cx_state': 'RUNNING'
})
lf_r.jsonPost()
```

Polling the WANlink State

The connection might take a second to start. You can poll it similar to to how we polled it above:

6. Modifying the WANlink

The frequency fields below are in occurrance per million. Speeds are set in bits per second (bps). Latencies are in milliseconds.

7. Stopping the WANlink

Choose your Stop State

Stopping a WANlink is again, changing its state to either STOPPED or QUIESCE. The QUIESCE state stops transmission on both endpoints but does not close the connection so that in-flight packets can arrive. Choose QUIESCE if you want to make your accounting of packets the most accurate.

```
lf_r = LFRequest.LFRequest(base_url+"/cli-json/set_cx_state")
lf_r.addPostData{
    'test_mgr': 'all',
    'cx_name': 'wl_egl',
    'cx_state': 'STOPPED'
})
Lf_r.jsonPost()
```

Poll Untill Stopped

There might be a milliseconds to seconds of delay depending on how your connection is stopped. You might have to wait for slightly longer than QUIESCE-TIME before the connections are closed when using a QUIESCE stop. Polling the state of the connection is relatively simple:

```
running = 1
while (running > 0):
    sleep(1)
    lf_r = LFRequest.LFRequest(base_url+"/wl/wl_eg1?fields=name,state,_links")
    # LFUtils.debug_printer.pprint(json_response)
```

You might want to watch that debug output at first.

JSON Output

8. Listing WANlink Endpoint Stats

Each of the endpoints will show the amount of packets transmitted:

```
lf_r = LFRequest.LFRequest(base_url+"/wl_ep/wl_eg1-A")
json_response = lf_r.getAsJson()
LFUtils.debug_printer.pprint(json_response)
```

JSON Output

The key/value pairs are grouped for this example, but attribute order is not normally ordered.

{	
"endpoint" : {	
"name" : "wl eg1-A",	
"buffer" : 19936, # bytes	
"corrupt 1" : 0, # these corruptions are per-wanpath	
"corrupt 2" : θ, "corrupt 3" : θ,	
"corrupt 3" : 0,	
"corrupt 4" : 0,	
"corrupt 5" : θ, "corrupt 6" : θ,	
"corrupt 6" : 0,	
"delay" : 30000,	
"dropped" : 0,	
"dropfreq %" : 0.20000002980232,	
"dup pkts" : 0,	
"dupfreq %" : 0.132499992847443,	
"eid" : "1.1.3.82",	
"elapsed" : 7,	
"extrabuf" : 17408,	
"failed-late" : 0,	
"jitfreq %" : 2.51250004768372,	
"maxjitter" : θ, "maxlate" : 606	
maxtate . 000,	
"max rate" : 265333,	
"ooo pkts" : θ, "adisc" · "ETEO"	
da13c . 1110,	
"reordfrq %" : 0.319999992847443,	
"run" : false,	
"rx bytes" : 0,	
"rx pkts" : 0,	
"script" : "None", # applicable if 2544 script has be applied	
"serdelay" : 45648.296875,	
"tx bytes" : 0,	
"tx drop %" : 0,	
"tx pkts" : 0,	
"tx rate" : 0,	
"tx-failed" : 0, # below is a to-string debug value	
"wps" : "TblJButton, eid: Shelf: 1 Resource: 1 Port: 3 Endpoint: 82 Ty	oo, Wool ink"
	Je: WanLink ,
}, "uri" : "wl ep/:wl ep id",	
"candela.lanforge.HttpWlEndp" : {	
"duration" : "0"	
},	
"handler" : "candela.lanforge.HttpWlEndp\$JsonResponse"	
}	

<u>Control a Chamber with the</u> <u>lf_chamber.pl Script</u>

Goal: Monitor and manipulate a CT480a chamber

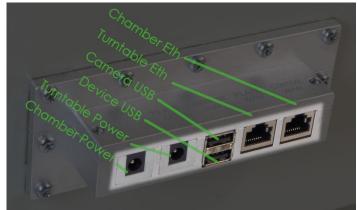
The CT840a chamber with rotating platform contains an embedded controller that is operated by the modbus protocol. For manual control of the turntable, the LANforge GUI may be used. For automation, you will want to use the lf_chamber.pl script.The lf_chamber.pl script allows you to monitor the door sensor, table angle, light state and fan state. It also provides control over lights, fans, and table position. This script requires a LANforge server version 5.4.1 or higher to communicate with the chamber. A CT521a or a virtual machine instance is adequate for the task. These instructions apply to the CT840a chamber.

1.



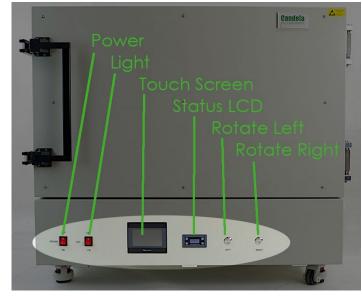
Configuring the CT840a

- A. The CT840a requires a network connection. Plug an Ethernet cable into the **Control LAN** port at the bottom rear of the chamber.
 - Depending on the date manufacture, these ports might be labeled differently.



- A. The chamber controller and lights are powered via AC cords at the outside bottom rear of the CT840a chamber.
- B. The Chamber Power or DC port is for 12v or other power required by devices inside the chamber. This runs below the turntable.
- C. The **Turnable Power** or Plate DC port is for a 12v or other power required by the DUT on the turntable. This is run up to the top of the turntable.
- D. Accessories or DUTs can be cabled to the **Device USB** port, or USB port.
- E. The USB camera has a dedicated USB port, **Camera USB** or Camera USB port.
- F. Below the turntable is an Ethernet jack for the DUT to use. That comes out at the Turnable Eth or Plate LAN port. It should be run up to the top of the turntable with the 12v power cord.
- G. The chamber controller is accessed on the network via the **Chamber Eth** or Control LAN port.

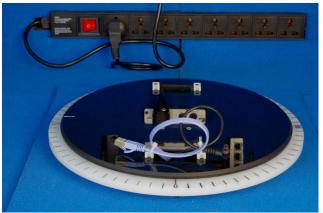
B. Use the front touch screen to set an IP address



- C. Make sure you can ping the chamber from your laptop and/or the machine running LANforge server. The LANforge server will communicate over the network to the Chamber Eth port.
- D. The rear ports are all accessory ports for the chamber.



A. 120v AC cord for internal power strip.



- I. This power plug provides power to the chamber modbus controller and the chamber lights.
- II. The turn table power cord plugs into the power strip.
- B. Pass-through DC barrel connectors. Use these for 12v (or other) power needed by devices in the chamber.
- C. SMA connectors. Seal these with terminators when not in use.
- D. Ethernet ports
- E. USB 3-A and USB C port
- F. Type F Coax port. Seal this with terminators when not in use.
- G. Fiber-optic pass-through. Seal this with screw-caps when not in use.
- H. HDMI ports

Configuring the Chamber in LANforge

A. In the Chamber View window, right-click on the main window and select New Chamber

	Scenario Configuration
LF-1	New Chamber New DUT New Profile New Traffic Profile Play All Paths

B. You will see the Create/Modify Chamber window.

2.

				Creat	e/Modify Chamb	er					
Vame	<auto create="" name="" new=""></auto>		Width	150		Heighti	150				
Chamber Type	Unknown (0)		Isolation	80		Speed (rpm)					
Furntable Type	CT850A (0)	-	Turntable			Position (deg)		Tilt (deg)			
Managed By:	None	-				Urtual	🔲 Open				
DUT-1			DUT-2		-						
DUT-3			DUT-4		-						
AMorge-1	None		LANforge-2	None	-						
ANforge-3	None	-	LANforge-4	None	-						
nt CICA	Int CX B		Int Atten	Ext CX A		Ext CX B	Ext Atten	Atten Floor	Zero-Atten RSSI 2.40hz		Zero-Atten RSSI Sõhz
	•	-	-		-	-		Cable (100 dd8) 💌	None (0 ddB)	-	None (0 ddB)
	•		-		~			Cable (100 ddB) 💌	None (0 ddB)		None (0 ddB)
		-	-		-	-	-	Cable (100 dd8) 👻	None (0 ddB)	-	None (0 ddB)
			-		-	-		Cable (100 dd8) 👻	None (0 ddB)	-	None (0 ddB)
		-	-		-	-		Cable (100 ddB) 👻	None (0 ddB)	-	None (0 ddB)
		-	-		-	-		Cable (100 ddB) 👻	None (0 ddB)	1	None (0 ddB)
			-		-	-		Cable (100 ddB)	None (0 ddB)	1	None (0 ddB)
		-	-		-	-	-	Cable (100 ddB) -	None (0 ddB)	-	None (0 ddB)
			-		-	-		Cable (100 dd0) 💌	None (0 ddB)	T	None (0 ddB)
		-	-		-	-		Cable (100 ddB)	None (0 ddB)	-	None (0 ddB)
		-	-		-	-	-	Cable (100 dd8) 👻	None (0 ddB)	T	None (0 ddB)
			-		-	-		Cable (100 dd8)	None (0 ddB)	1	None (0 ddB)
		-	-		-	-		Cable (100 ddB) -	None (0 ddB)	-	None (0 ddB)
			-		-	-		Cable (100 dd8)	None (0 ddB)	Ţ	None (0 ddB)
			-		-	-	-	Cable (100 ddB)	None (0 ddB)	T	None (0 ddB)
			-		-	-		Cable (L00 ddB)	None (0.ddB)	t	None (0 ddB)

C. Select the chamber and turntable type:

Name:	Chamber-0	
Chamber Type	2D Large (3)	-
Turntable Type	CT840A (2)	-
Managed By:	<custom> CT850A (0)</custom>	
DUT-1	ComXim (1)	
DUT-3	CT840A (2)	

A. For Chamber Type, select 2D Large

B. For Turntable Type, select CT840A

C. For Turntable, put in the IP address of the chamber

0			Create/Modify Chamber					
Name:	Chamber-0		Width:	150	Height:	150		
Chamber Type	2D Large (3)	-	Isolation	80	Speed (rpm)	0.0		
Turntable Type	CT840A (2)	Ŧ	Turntable	192.168.100.10	Position (deg)	0.0		
Managed By:	l (vm-48e4)	-	Turntable Rpt:	Position: 0.0 Tilt: 0.0 RPM: 0.0	D			

D. Select your LANforge server resource that manages the turntable

Create/Modify Chamber					
Name:	Chamber-0	Width:		Height:	
Chamber Type	2D Large (3)	 Isolation 	n 80	Speed (rpm)	0.0
Turntable Type	CT840A (2)	Turntab	le 192.168.100.10	Position (deg)	0.0
Managed By:	1 (vm-48e4)	Turntab	le Rpt: Position: 0.0 Tilt: 0.0 RPM	M: 0.0	

E. Click **OK**

D. You will see a new chamber, **C0** in the Chamber View window.

	Chamber View
	Scenario Configuration
LF-1	(C0

E. In the Chamber View window, right-click on the chamber CO and select Modify

F. Use the Speed and Position fields to adjust the turntable.

Speed (rpm)	0.0	
Position (deg)	0.0	-

G. Click Apply to send the configuration.

3.

Scripting Chamber Operations

- A. Connect to your LANforge system and open a terminal. An ssh connection is adequate.
- B. Become root: \$ sudo -s

- C. The lf_chamber.pl script lives in /home/lanforge # cd /home/lanforge
- D. To use the script, you must setup your environment variables located in /home/lanforge/lanforge.profile:
 - # source ./lanforge.profile
 - # source ./lantorge.protile
- 4. Now you may operate the script. Just using the script provides you a summary of options:

```
# ./lf_chamber.pl
Usage:
./lf_chamber.pl --angle 45 --speed 3 --targ 192.168.100.122
./lf_chamber.pl --adjust 5 --targ 192.168.100.122
./lf_chamber.pl --fan 1 --targ 192.168.100.122
./lf_chamber.pl --lights 1 --targ 192.168.100.122
./lf_chamber.pl --status 1 [ --id foo --mgt_pipe /foo/bar ] --targ 192.168.100.122
```

5. The following examples are going to use the example IP 10.0.0.9 for the chamber location on the network.

6. Chamber Status

- A. Use the command ./lf_chamber.pl --targ 10.0.0.9 --status 1
- Current-Angle: 3598 Door-Open: 0 Table-Moving: 0 Lights: 0 Fan: 0 Jog-Speed: 3 Return-Speed: 3 Absolute-Speed: 3 Jog Angle: 449
- B. If you see a lot more output, debugging has been enabled. You will see the individual mbpoll commands:

```
Corrent-Angle: mbpoll -a 1 -r 4139 -t 4 10.0.0.9 -1

Door-Open: mbpoll -a 1 -r 2094 -t 1 10.0.0.9 -1

0 Table-Moving: mbpoll -a 1 -r 2094 -t 1 10.0.0.9 -1

0 Table-Moving: mbpoll -a 1 -r 1283 -t 1 10.0.0.9 -1

0 Fan: mbpoll -a 1 -r 1284 -t 1 10.0.0.9 -1

0 Fan: mbpoll -a 1 -r 2184 -t 1 10.0.0.9 -1

3 Absolute-Speed: mbpoll -a 1 -r 4589 -t 4 10.0.0.9 -1

3 Absolute-Speed: mbpoll -a 1 -r 4589 -t 4 10.0.0.9 -1

3 Jog Angle: mbpoll -a 1 -r 4513 -t 4 10.0.0.9 -1
```

C. If a script is presently moving the table, you will see an error similar to: Current-Angle: /home/lanforge/local/bin/mbpoll: Connection failed: Operation now in progress COMM-FAIL

D. If there is a LANforge service currently engaging the chamber, you might see this error because the lanforge service polls the chamber frequently. If you want to stop the LANforge server, use the command: sudo service lanforge stop

7. Controlling the Platform

- A. Use the command ./If_chamber.pl --targ 10.0.0.9 --angle 45 --speed 3 to rotate the platform 45 degrees from zero.
- B. You will see the Current-Angle and Jog Angle reported in tenths of degrees, so 450 is 45.0 degrees.# ./lf_chamber.pl --targ 10.0.0.9 --status 1

Current-Angle: 450 Door-Open: 0 Table-Moving: 0 Lights: 0 Fan: 0 Jog-Speed: 3 Return-Speed: 3 Absolute-Speed: 3 Jog Angle: 450

- C. Change argument --angle 45 to --adjust 5 to add five more degrees of rotation:
- D. # ./lf_chamber.pl --targ 10.0.0.9 --adjust 5
 - Adjust 5

E. # ./lf_chamber.pl --targ 10.0.0.9 --status 1 Current-Angle: 500 Door-Open: 0 Table-Moving: 0 Lights: 0 Fan: 0 Jog-Speed: 3 Return-Speed: 3 Absolute-Speed: 3 Jog Angle: 500

- F. To return the plaform to zero rotation, use --angle 0 argument.
- G. The --speed argument modifies the rate of rotation. Speed 1 is very slow; use speed 3 to save time. Speed 6 might be too fast and your DUT might shift unexpectedly.
- H. If you see output that says mbpoll, you may ignore those lines.

8. Controlling the Fans

A. Use the command ./lf_chamber.pl --targ 10.0.0.9 --fan 1 to turn fan on.

Toggle fan /home/lanforge/local/bin/mbpoll -a 1 -r 2074 -t 0 10.0.0.9 -1 0 > /dev/null /home/lanforge/local/bin/mbpoll -a 1 -r 2074 -t 0 10.0.0.9 -1 1 > /dev/null

./lf_chamber.pl --targ 10.0.0.9 --status 1

Current-Angle: 3598 Door-Open: 0 Table-Moving: 0 Lights: 0 Fan: 1 Jog-Speed: 3 Return-Speed: 3 Absolute-Speed: 3 Jog Angle: 449

- B. Change argument --fan 1 to --fan 0 to turn the fan off:
- C. # ./lf_chamber.pl --targ 10.0.0.9 --fan 0

Toggle fan /home/lanforge/local/bin/mbpoll -a 1 -r 2074 -t 0 10.0.0.9 -1 0 > /dev/null /home/lanforge/local/bin/mbpoll -a 1 -r 2074 -t 0 10.0.0.9 -1 1 > /dev/null

- D. # ./If_chamber.pl --targ 10.0.0.9 --status 1
- Current-Angle: 3598 Door-Open: 0 Table-Moving: 0 Lights: 0 Fan: 0 Jog-Speed: 3 Return-Speed: 3 Absolute-Speed: 3 Jog Angle: 449 E. If you see output that says mbpoll, you may ignore those lines.
- E. In you see corport man says import, you may ignore me.

9. Chamber Lights

- A. Use the command ./lf_chamber.pl --targ 10.0.0.9 --lights 1 to turn lights on.
- B. Use the command ./lf_chamber.pl --targ 10.0.0.9 --lights 0 to turn lights off.
- C. The chamber lights are useful when setting up equipment but also for viewing the equipment with the USB camera.

10. USB Camera

- A. The USB camera is directly controlled by the connected computer. In the video demonstrations the LANforge system has a USB A-to-USB A cable connected to the chamber to use the camera. The software to use the camera is installed on the LANforge system, it would be one of these: xawty, cheese or camorama. There is nothing special about the camera, any laptop should be able to use it. For more information see USB Cable Types
- B. If you want to browse the camera from any machine on the network, the simplest way to do that is to use vncviewer/rdesktop to browse the camera software running on the LANforge desktop.

- C. You might notice that the default frame rate of the camera takes a lot of the LANforge CPU time. It should be possible to use v412-ct1 to set the frame rate of the camera.
- D. Please note that recording movies using the camera can be done but they will be very large files. We recommend doing timed frame captures every few seconds to save space.
- E. See also: Video for Linux documentation.

Emulate video streaming traffic with the <u>13_video_em.pl Script</u>

Goal: Emulate video stream traffic patterns using Layer-3 connections.

Using the 13_video_em.p1 and the 13_vid_group.p1, we assemble two test groups of connections, a group of Generic connections, and a group of Layer3 connections, that emulate the bursty buffer filling pattern of traffic that video streaming tends to resemble. Requires LANforge 5.4.2.



Begin with stations

A. Using a CT523c, we can create 16 stations, and for this script setup you probably do not want to create more than that. These scripts poll LANforge every 200ms and that loads the server quickly. If you are using a CT521a or CT522b, then consider starting with five or six stations. There needs to be a continuously named series of stations.

2									≙_□
Control	Repor	ting Tear O	ff <u>I</u> nfo <u>I</u>	Plugins					
				Cham	ber View	Stop All F	Restart Manager	Befresh	HELP
Status	Port	Mgr VolP	RTP	oIP/RTP End	lps DUT Profiles A	lerts Messages	Warnings +		
Disp	: 192.	168.92.14:0.	0	Sniff Pac	kets 🗹 Down	t Clear Counters	Reset Port	Delete	
Desk 7	Times	medium (8	5) -	Apply		I Display	Create	Modify Batch	Modify
npr	rimer.	mentum (0	»/ •	Abbi				Mogily	Modily
					-All Ethernet Interfaces	(Ports) for all Resou	rces.		
	100								
Port	Down	Parent Dev	Channel	Alias	SSID	AP	IP	MAC	
.1.00				eth0			192,168,92,14	0c:c4:7a:e2:01:e6	
.1.01				eth1			10.40.0.74	0c:c4:7a:e2:01:e7	
.1.02	-	wiphy0	153	sta000	jedway-wpa2-x2048-5-1	00:0E:8E:78:DE:98	10.40.10.155	00:0e:8e:25:a1:47	
.1.04	-		153	sta001	jedway-wpa2-x2048-5-1		10.40.10.153	00:0e:8e:c5:54:47	
.1.05			153	sta002	jedway-wpa2-x2048-5-1		10.40.10.150	00:0e:8e:de:10:47	
.1.06	0		153	sta003	jedway-wpa2-x2048-5-1		10.40.10.151	00:0e:8e:79:22:47	
.1.07	-		153	sta004	jedway-wpa2-x2048-5-1		10.40.10.157	00:0e:8e:13:4d:e3	
.1.09	-		153	sta005	jedway-wpa2-x2048-5-1		10.40.10.152	00:0e:8e:ce:al:e3	
.1.10			153	sta006	jedway-wpa2-x2048-5-1		10.40.10.161	00:0e:8e:d6:e9:e3	
.1.11	8		153	sta007	jedway-wpa2-x2048-5-1		10.40.10.159	00:0e:8e:e2:b5:e3	
.1.12	-		153	sta008	jedway-wpa2-x2048-5-1		10.40.10.164	04:f0:21:94:eb:03	
.1.14	-		153	sta009	jedway-wpa2-x2048-5-1		10.40.10.162	04:f0:21:57:12:03	
.1.15			153	sta010	jedway-wpa2-x2048-5-1		10.40.10.160	04:f0:21:82:0c:03	
1.16			153	sta011	jedway-wpa2-x2048-5-1		10.40.10.163	04:f0:21:ee:8c:03	
.1.17			153	sta012	jedway-wpa2-x2048-5-1		10.40.10.158	00:19:70:02:14:2d	
.1.19			153	sta013	jedway-wpa2-x2048-5-1		10.40.10.154	00:19:70:34:8e:2d	
.1.20			153	sta014	jedway-wpa2-x2048-5-1		10.40.10.156	00:19:70:7f:e1:2d	
1.21			153	sta015	jedway-wpa2-x2048-5-1		10.40.10.165	00:19:70:d5:c8:2d	
.1.03			0	wiphy0	, , ,		0.0.0.0	00:0e:8e:4e:59:47	
.1.08			0	wiphy1			0.0.0.0	00:0e:8e:5a:70:e3	
1.13			0	wiphy2			0.0.0.0	04:f0:21:20:37:03	
1.18				wiphy3			0.0.0.0	00:19:70:be:62:2d	
.1.22	V		153	wlan0	iedway-wpa2-x2048-5-1	Not-Associated	0.0.0.0	5a:12:5e:ea:af:d7	-
.1.23			153	wlan1	,	Not-Associated	0.0.0.0	06:78:0b:9c:c9:79	
	-	Luin have				Alak Associated	0.0.0.0	C- J- 7- 50 0- 60	

For more information see Creating Stations

1.

2.

B. In this example we will use eth1 as our upstream port. We will be referring to that using the EID format: 1.1.2 For more information see LANforge Entity IDs

Create Connections Using 13_vid_group.pl

A. The script 13_vid_group.p1 has help examples. You can do four tasks with the script.

Terminal - la	nforge@ct524-genia:~/scripts	
jreynolds@cholla5:~/git/lanforge-scripts	× lanforge@ct524-genia:~/scripts	×
lanforget:524-genia scripti5 ./l3 vid group.pl asge:./l3 vid group.pl create a large group of -action -a (create destroy start stor -otfersize b(stes K M = size of emulated RX -clear group -2 # empty test group first -crox mare -c (connection prefix) -empty test (top)b(l1_top)f_up) -empty test (top)b(l1_top)f_up) -empty = t (top)b(l1_top)f_up) -empty = t (lanforge porver) # default local -empr -n (lanforge server) # default local -empty = t (station resource) -resurce - (station resource) = for station secure) # list of streams maintained in l -test_gr = test of streams = solution name[lis # default ls (crome) # dl concectio # default ls crome) # dl concectio # default ls crome) # dl concectio # default	p) k buffer, default 308 s thost ti # default 100ps ti # default y-sdr.1080p30 13.videe em.pl se placed in this group € deeric connections in 1.3.(cv.name)_tg ter port; mple	
<pre>xamples: porteam porteam porteam porteam to need to create 30 stations emulating 720p HDR 60fps trans .73 vid group.plaction createbuffer size 8M endp type udpfirst_sta sta0000num cx 30 resource 2speed 200Mstream yt-hdr-720p60 upstream 1.2.br0</pre>	smitted from resource 2: Mclear_groupcx_name yt1080p60.1 \ \	
start test group: ./l3_vid_group.pl -a start -g yt60fps		
stop test group: ./l3_vid_group.pl -a stop -g yt60fps		
<pre># add 30 more stations on resource 3 to group ./l3_vid_group.pl -a create -b 8M -c yt1080p60.3 -s 200M -e yt-hdr-720p60 -g yt60fps -u 1.2.br0</pre>	-t udp -i sta0100 -n 30 -r 3 \	
f destroy test group ./l3_vid_group.pl -a destroy -g_yt60fps lanforqe@ct524-genia_scripts]5]]		

- A. Create groups of video emulators
- B. Start groups
- C. Stop groups
- D. Destroy groups
- B. We will create a group of 16 connections on our stations. Use the command: ./l3_vid_group.pl --action create --endp_type tcp --first_sta sta000 --num cx 16 --test grp sixteen --upstream 1.1.eth1

num_cx iotest_grp sixt	eenupsti	eam I.I.ethi	
Ternii Ternii			≙_ 0 ×
jreynolds@cholla5:~/git/lanforge-scripts	×	lanforge@ct524-genia:~/scripts	×
[lanforget524-genia script5] //l3 yid group.plactia No cross connects found for test group _l3 sixteen. No cross connects found for test group sixteen. Creating test group [sixteen]Creating test group [_l3. Creating Generic connections for videe emulation	sixteen]adding L3		xteenupstream 1.1.ethi

C. You will see two test group created. The group named **_L3_sixteen** contains the Layer-3 tcp connections. The group named **sixteen** contains Generic connections that control the Layer-3 connections.

2				A_0;
ontrol <u>Reporting</u>	Tear Off Info Plu	gins		
		Chamber <u>V</u> iew	Stop All Restart Manager	<u>R</u> efresh HELP
Status Port Mgr	Layer-3 L3 En	dps Generic Test Grou	p Alerts Warnings +	
	Rpt Timer: fas		Select All Start + Stop - Quiesce Display Create Modify Deleter	Clear
		LAN	Iforge Test Groups	
Name Run	Script		Cross Connects	
L3 sixteen	None sixteen-00	00 sixteen-0001 sixteen-0	002 sixteen-0003 sixteen-0004 sixteen-0005	sixteen-0006
sixteen 🗌	None CX sixteen-	0000 CX sixteen-0001 CX si	xteen-0002 CX sixteen-0003 CX sixteen-0004 C	sixteen-0005 C

D. The Generic tab will have 16 connections.

2													≙ _	
Control Repor	ting Tear Off	Info Plu	gins											
			Char	mber ⊻iew		Stop /	AU .	Restar	t Manag	er	B	efresh	Н	ELF
							_							
Status Port	Mgr Layer-	3 L3 End	dps ((Seneric Tes	t Group	Alerts	Warning	IS +						
								Select	All 61	art + S	top - Cl	ear		
	Rpt Timer: fa	st (1 :	5)	Go Test M	anager a	all	-	Jelect	All St		cop - Ci	ear		
									Cr <u>e</u> ate	Mo <u>d</u> ify	Delete			
				Gaporic	Endpointe	for Selec	tod Tort	Managor						
Name	EID	Status	Rpt#	Last Results				PDU/s TX	Dy Plete	PDU/s RX	Dropped	hos TY	bps R	vI
ixteen-0000	1.1.2.161	Stopped					0	0	0	0	o	0 bps	0 bps	è
ixteen-0001	1.1.4.163	Stopped				0 B	0	0	0	0	0	0 bps	0 bps	4
ixteen-0002	1.1.5.165	Stopped				0 B	0	0		0	0	0 bps	0 bps	-
ixteen-0003	1.1.6.167	Stopped				0 B	0	0	0	0	0	0 bps	0 bps	-
ixteen-0004	1.1.7.169	Stopped				0 B	0	0		0	0	0 bps	0 bps	-
ixteen-0005	1.1.9.171		0		0 B	0 B	0	0	0	0	0	0 bps	0 bps	-
ixteen-0006	1.1.10.173	Stopped	0		0 B	0 B	0	0	0	0	0		0 bps	1
ixteen-0007	1.1.11.175	Stopped			0 B	0 B	0	0	0	0	0	0 bps	0 bps	1
ixteen-0008	1.1.12.177		0		0 B	0 B	0	0	0	0	0	0 bps	0 bps	1
ixteen-0009	1.1.14.179	Stopped	0			0 B	0	0	0	0	0	0 bps	0 bps	1
ixteen-0010	1.1.15.181	Stopped	0		0 B	0 B	0	0	0	0	0	0 bps	0 bps	-
sixteen-0011	1.1.16.183	Stopped	0		0 B	0 B	0	0	0	0	0	0 bps	0 bps	Т
sixteen-0012	1.1.17.185	Stopped	0		0 B	0 B	0	0	0	0	0	0 bps	0 bps	
4			11											•

- Logged in to: localhost:4402 as: Admin
- E. The Layer-3 tab will have 16 connections.

ontrol <u>R</u> epor	ting Te	ar <u>O</u> ff Info	LANforge Manager Version(Plugins		1-sock	_	_		≙ _ □
			Chamber ⊻iew St	op All	Restart M	anager		Refresh	HELF
Status Port	Mgr L	ayer-3 L	3 Endps Generic Test Group Alert	s Warnings	+				
Rpt Ti	mer: fa	st (1 s	Go Test Manager all	Se	lect All St	art + St	op - Quie	sce Clear	
View	0 -	500	▼ Go		Disp]ay	Cr <u>e</u> ate	Modify	Delete	
			Cross Connects for Sele	ected Test Ma	nager				
Name	Туре	State	Endpoints (A \leftrightarrow B)	Pkt Rx A	Pkt Rx B	Bps Rx A	Bps Rx B	Rx Drop % R A B	x Drop
sixteen-0000	LF/TCP	Stopped	sixteen-0000-A <=> sixteen-0000-B	0	0	0	0	0	
sixteen-0001	LF/TCP	Stopped	_sixteen-0001-A <=> _sixteen-0001-B	0	0	0	0	0	
sixteen-0002	LF/TCP	Stopped	_sixteen-0002-A <=> _sixteen-0002-B	0	0	0	0	0	
sixteen-0003	LF/TCP	Stopped	sixteen-0003-A <=> sixteen-0003-B	0	0	0	0	0	
sixteen-0004	LF/TCP	Stopped	sixteen-0004-A <=> sixteen-0004-B	0	0	0	0	0	
sixteen-0005	LF/TCP	Stopped	sixteen-0005-A <=> sixteen-0005-B	0	0	0	0	0	
sixteen-0006	LF/TCP	Stopped	sixteen-0006-A <=> sixteen-0006-B	0	0	0	0	0	
sixteen-0007	LF/TCP	Stopped	sixteen-0007-A <=> sixteen-0007-B	0	0	0	0	0	
sixteen-0008	LF/TCP	Stopped	sixteen-0008-A <=> sixteen-0008-B	0	0	0	0	0	
sixteen-0009	LF/TCP	Stopped	sixteen-0009-A <=> sixteen-0009-B	0	0	0	0	0	
					2	~	~		•

F. When we inspect one of the Generic connections, we can see the command it uses.

/home/lanforge/scripts/l3_video_em.pl --mgr localhost --mgr_port 4001 --cx_name _sixteen-0000 --max_tx 1000000000 --buf_size 3145728 --stream yt-sdr-360p30 --tx_style bufferfill --quiet yes

You can paste this command into a shell prompt on your LANforge and use it. We discuss the options in the following section.

			Creater	Westify Generic Ends		≙ -
Name:	sixteen-0000	Rpt Timer:	default (5 s)	Test Manager	default_tm	
Shelf:	1	Resource:	1 (ct524-genia)	▼ Port:	2 (sta000) 💌 Endp ID: 161	
Command	Builders	generic		-		
Command:	/home/lanforge/scripts/l3	video em.plmar l	ocalhostmor po	rt 4001cx name si	teen-0000max tx 1Gbuf size 3145728stream vt-s	dr-1.080p30qujet ve
Command:	[home/lanforge/scripts/l3]	_video_em.plmgr l	ocalhostmgr_po		teen-0000 - max, tx 1G -buf, size 3145728 -stream yt-4	dr-1080p30quiet ye
Command:	(home/lanforge/scripts/l3,	_video_em.plmgr li	localhostmgr_po	rt 4001cx_name_sli Command Output		dr-1080p30quiet ye
Command:	/home/lanforge/scripts/l3	_video_em.plmgr k	ocalhostmgr_po			dr-1080p30quiet ye
Command:	[home/lanforge/scripts/l3,	_video_em.plmgr l	localhostmgr_po			dr-1080p30quiet ye
Command:	[home/lanforge/scripts/l3	"video_em.plmgr li	ocalhostmgr_po			dr-1080p30quiet ye
Command:	(home/lanforge/scripts/13	video_em.plmgr li	ocalhostmgr_po			dr-1080p30 -quiet ye
Command:	(nome/lanforge/scripts/l3	_video_em.plmgr k	ocalhostm <u>gr_po</u>			dr-1080p30 -quiet ye
Command:	[home/lanforge/scripts/i3]	_v(deo_em.plmgr l	ocalhostmgr_po			dr-1080p30 -quiet ye
Command:	(home/lanforge/scripts/l3	_video_em.plmgr k	ocalhost -mgr_po			dr-1080p30 -quiet ye

G. You can highlight the command in the window and copy it with Ctrl-C

ommand: [home/lanforge)scripts/13_video_em.pl -mgr localhost -mgr_port 4001 -oc, name_sixteen-0000 -max_tx 1G -buf_size 3145728 -stream yt-sdr-1080p30 -quiet yes Command Output

H. You can paste the command into the shell with Ctrl-Shift-V

2			A _ D 3
jreynolds@cholla5:~/git/lanforge-scripts	×	lanforge@ct524-genia:~/scripts	×
lanforget534-genia scripts]s Lanforget534-genia scripts]s Lanforget534-genia scripts]s Lanforget534-genia scripts]s /home/lanforge/scripts/l3_video_em.	plmgr localhostmgr_port 4	′ 001cx_name_sixteen-0000max_tx 1Gbuf_size 3145728stre	ean yt-sdr-1000p30quiet yes []

- I. When we inspect the Layer-3 connection, we see these aspects:
 - It is a TCP connection. This is optional, you can create UDP connections; the Android YouTube app uses TLS over UDP (QUIC protocol) connections.
 - Both endpoints are set at 0 bps transmit. The Generic script will control the throttle on the B-side of the connection.
 - The PDU size is auto. This doesn't have much bearing on TCP, but might have bearing on UDP connections.
 - The Report Timer is set to three seconds. This value is too long to graph with much detail in the Dynamic Report, but you can shorten it to 500ms if you desire to see more resolution in the Dynamic Report graph. This Report Timer value directly impacts processor load, so use it judiciously.
 - Auto-Helper is a new feature intended to reduce CPU load, it has little impact at the moment.
 - Multi-con is not desired for this style of connection.

A 8		_			1223		-		-
CX Name: CX Type:	Cross-Connect sixteen-0000 LANforge / TCP				Report Timer:	Cross-Connect medium (3 s)			-
	LANIOI ge / ICP	-				Endpoint A (Client)		Endpoint B (Server)	
	Endpoint A (Client)		Endpoint B (Server)		Pld Pattern	increasing	-	increasing	-
Resource:	1 (ct524-genia)	-	1 (ct524-genia)	-	Min IP Port:	AUTO	-	AUTO	-
Port:	2 (sta000)	-	1 (eth1)	-	Max IP Port:	Same	-	Same	
Min Tx Rate:	Zero (0 bps)	-	Zero (0 bps)	-	Min Duration:	Forever	-	Forever	1.
Max Tx Rate:	Same	-	Same	-	Max Duration:	Same	-	Sane	
Min PDU Size:	AUTO	-	AUTO	-	Min Reconn:	0 (0 ms)	-	0 (0 ms)	
Max PDU Size:	AUTO	-	AUTO	-		Same	Ţ	Same	
IP ToS:	Best Effort (0)	-	Best Effort (0)	-	Max Reconn:				-
	Infinite	-	Infinite		Multi-Conn:	Normal (0)	•	Normal (O)	
Pkts To Send:	Intinite	-	Infinite	-		Auto-Helper		Auto-Helper	
						Script		Script	
						Thresholds		Thresholds	

Exploring 13 video em.pl

- A. The options for 13_video_em.p1 are available with --he1p. The most important options for tuning video streaming emulation are:
 - tx_style: bufferfill is default and models present video playback
 - max_tx: this is the starting TX rate. The I3_vid_group script defaults this to 1Gbps, which is
 unrealistic for common WiFi connections. The script will regularly poll the station side for a
 TX-RATE value of the station to determine a more realistic upper bound for maximum rate.
 The more stations that share the same channel, the less realistic this rate becomes. We
 want to know this to some degree so that we can determine a realistic pause between
 buffer fills at a given bit rate.
 - buf_size: observation of packet captures indicate that a video plugin on a browser buffers
 three to four megabytes of video. Between this number and our max_tx rate, we can
 calculate when to transmit to fill the video buffer before it empties.
 - stream_res: This is a list with broadly agreed upon estimates of video bitrates. When
 people mention frame-rate, that is just part of the bitrate calculation; audio quality, color
 depth, and resolution are all part of the bitrate value.

			≙_□
jreynolds@cholla5:~/git/lanforge-scripts	×	lanforge@ct524-genia:~/scripts	;
qcf 446:169, hd:2400-169, ccf 3000-169, y 144616; q, 11004-13, 4004:14, 14004:14, 144616; q, 11004-14, 4004:14, 14004:14, 144616; q, 11004-14, 1404; 1404, 1404	<pre>e emulate a video server the server, is to fill holf offer en off ill holf offer "72000", y-thdr 2200000, wide "72000", y-thdr 2200000, wide "72000", y-thdr 2200000, wide "72000", y-thdr 2200000, wide "72000", y-thdr 220000, wide and y-thdr 20000, /pre>	66.43. yt.dcf 220000 reg23000 yeg-150. yt.dcf Her/140600, opera13, yt.dcf 14000, 20013, yt.dcf yt.dcf 20000, 210015, yt.dcf 14000, 20013, yt.dcf 3.7200, rea00005, yt.Hcf 14000, 20013, yt.dcf 3.7200, rea00005, yt.Hcf 14000, 20013, yt.dcf 140000, scyp-16.5, rea0001600 40000, yt.gcf 2000port_amme ctl 4.0000, yt.dcf 4.0000, yt.dcf 4	hdr-1080p30, yt-sdr-48(30, 480p162, 108p4:3, y yt-sdr-360p30, yt-sdr-1

B. A table of **stream resolutions** are available when you use the --list option. By default, the 13_vid_group.pl script uses the yt-sdr-1080p30 stream size. That name can be decoded like so:

yt: YouTube (but any popular stream, really)

sdr: Standard Dynamic Range color, hdr: High Dynamic Range color

1080: frame height

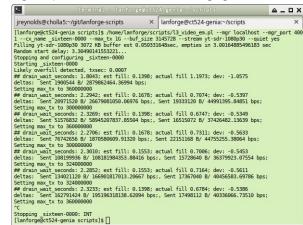
p: progressive, i: interlaced

30: 30 frames per second; smaller bitrates might be 29.9, 25, or 24

	Term		_			_		-/scripts	_	-	
-	ynolds@cholla5:						-	ct524-genia:~/scripts			>
lar	nforge@ct524-gen defined Video St	ia so	rip	ts]\$./l3_vi	.deo_em.p	ιι	ist			
rea	defined video St	reams									
	Stream		W	н		Audio+					
	108p4:3]		4 x		using		kbps				
	qcif-48k-4:3]		4 x 0 x		using		kbps kbps				
	sqvga-16:9] sqvga-4:3]		θx		using		kbps				
	qcif-48k-16:9]		2 x		using		kbps				
	qvga-16:9]		0 x		using	48	kbps				
	qvga-4:3]		0 x		using		kbps				
	144p16:9]		2 x 2 x		using		kbps kbps				
	<pre>qcif-96k-4:3] qcif-96k-16:9]</pre>		6 x		using using		kbps				
	216p4:3]		8 x		using		kbps				
	cif-300k-4:3]	28	8 x	216	using	300	kbps				
	cif]		2 x		using		kbps				
	216p16:9]		4 x		using		kbps				
	cif-300k-16:9] 240p4:3]		4 x 0 x		using		kbps kbps				
	cif-500k-4:3]		0 x		using		kbps				
	cif-500k-16:9]		4 x		using		kbps				
	360p4:3]	48	0 x		using	800	kbps				
	480i4:3]		0 x		using		kbps				
	480p4:3]		0 x		using		kbps				
	d1-800k-4:3] d1-800k-16:9]		0 x 2 x		using		kbps kbps				
	yt-sdr-360p30]		2 X 0 X		using	1128					
	d1-1200k-4:3]		e x		using	1200					
	480p16:9]	85	2 x		using	1200	kbps				
	d1-1200k-16:9]		2 x		using	1200					
	yt-sdr-360p60]		0 x		using	1628					
	720p] hd-1800k-16:9]		0 x 0 x		using	1800 1800					
	hd-2400k-16:9]		0 x		using	2336					
	yt-sdr-480p30]		2 x		using	2628					
	yt-sdr-480p60]	85	2 x		using	4128	kbps				
	yt-sdr-720p30]		0 x		using	5384					
	yt-hdr-720p30]		0 x		using	6884					
	yt-sdr-720p60] vt-sdr-1080p30]		0 x		using	7884 8384					
	yt-hdr-720p60]		0 x		using	9884					
,	yt-hdr-1080p30]				using	10384					
	yt-sdr-1080p60]				using	12384					
	yt-hdr-1080p60]				using	15384					
	yt-sdr-1440p30] yt-hdr-1440p30]				using	16512 20512					
	yt-sdr-1440p50]				using using	20512					
	yt-hdr-1440p60]				using	30512					
	yt-sdr-2160p30]				using	40512					
	yt-hdr-2160p30]				using	50512					
	yt-sdr-2160p60]				using	61512					
3	raw720p30]		0 x 0 x		using using	76012 221184					
	raw720p30] raw720p60]		0 x		using	442368					
	raw1080i]		0 x		using	1486512					
I	raw1080i30]		0 x		using	1488000					
	raw1080i60]			540	using	1488000					
	raw1080p]			1080 ts]\$[using	2976000	kbps				

C. Running the command

- D. When we run the 13_video_em.p1 command that we copied and pasted into our terminal above,
 - we'll see regular output ever several seconds. Lets discuss whats going on:



- A. Filling yt-sdr-1080p30 3072KB buffer: tells us our video bitrate, and our buffer size (3MB)
- B. est 0.0503 sec: estimate of how long at 1Gbps filling the buffer will take
- C. empties in 3.0016 sec: playback rate before buffer is fully played.
- D. Random start delay: 3.304sec...: the script is waiting this long before starting. This is so that we avoid a load spike, false detections of constant transmit, and more realistic transmit pattern.
- E. Likely overfill detected This warning appears when you transmit longer than your buffer fill takes. Estimates are inaccurate at the start.
- F. drain_wait_seconds is the computed time between stopping and restarting the next transmission. This is our empty time minus our transmit time.
- G. Actual fill: describes how long transmitting a full buffer took.
- H. dev: the difference between estimated fill time and actual fill time.
- Setting max_tx to 360000000: indicates we have detected our RX-Rate for our station was detected, and estimates could be better in range. Estimates are only for an isolated station.

Starting and Stopping Connections

- A. You can use the LANforge GUI Test Groups tab or the <u>13_vid_group.p1</u> script to start and stop connections.
- B. First, make sure you stations are associated. The scripts will not admin-up your stations.

C. Using the LANforge GUI

4.

5

- A. Highlight the Test Group holding the Generic connections. In our example that is the group named sixteen
- B. Press the Start button.
- C. The Generic scripts will control the Layer-3 scripts in the **L3_sixteen** group.
- D. Stopping the test group will also stop the Layer-3 connections.

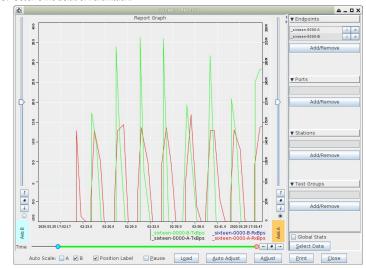
D. Using the 13_vid_group.pl script

- A. Starting the groups uses the --action start argument: ./l3_vid_group.pl --test_grp sixteen --action start
- B. Stopping the groups uses the --action stop argument: ./l3_vid_group.pl --test_grp sixteen --action stop

Observing Performance

Status Port Mgr Layer-3 L3 Endps Generic Test Group Alerts Warnings + Rpt Timer: fast (1 s) Go Test Manager all Select All Stat + Stop - Quiesce View 0.500 Go Display Crgate Modify Delet Cross Connects for Selected Test Manager Crgate Modify Delet Cross Connects for Selected Test Manager - Select All Stat + Stop - Quiesce Sixteen-0000 LF/TCP Stoped sixteen-0000-B 3.347 081,395,677 0 Sixteen-0001 LF/TCP Stoped sixteen-000-A <<>> sis State-Selected Pka 0 0 Sixteen-0001 LF/TCP Stoped sixteen-000-A <<>> sis Selected Minue 0 0 Sixteen-0001 LF/TCP Stoped sixteen-000-A <<>> sis Selected Minue 0 0 Sixteen-0001 LF/TCP Stoped sixteen-000-A <<<>> sis Class Selected Minue 0 0 Sixteen-0003 LF/TCP Stoped sixteen-000-A	Fy Delete Rx B Rx Drop % Rx Drop % 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
Rpt Timen: Tast (1 s) ▼ Go Test Manager all ▼ Select All Start + _ Stop Quiesce View 0 - 500 ▼ Go Display Crgate Modify Delet Cross Connects for Selected Test Manager Sixteen-0000. E/TCP Stopped sixteen-000.8 3.347 0.81, 395,677 0 Sixteen-0000.4 <=>> sixteen-000.8 Stat< Selected Monas 0 0 Sixteen-000.4 <=>> sixteen-000.8 Stat< Selected Monas 0 0 Sixteen-000.4 <=>> sixteen-000.4 <=>> sixteen-000.8 Stat 0 0 Sixteen-000.4 <=>> sixteen-000.4 <=>> sixteen-000.4 Selected Monas 0 0 Sixteen-000.4 <=>> sixteen-000.4 <=>> sixteen-000.4 Selected 0 0 Sixteen-000.5 LF/TCP Stopped sixteen-000.4 <=>> sixteen-000.4 Selected 0 0 Sixteen-000.5 LF/TCP Stopped sixteen-000.4 <=>> Sixteen-000.4 Selected 0 0 Sixteen-000.5 LF/TCP Stopped sixteen-000.4 <=>> Sixteen-000.4 Selected 0 0 <td <="" colspan="2" th=""><th>Fy Delete Rx B Rx Drop % Rx Drop % 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</th></td>	<th>Fy Delete Rx B Rx Drop % Rx Drop % 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</th>		Fy Delete Rx B Rx Drop % Rx Drop % 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
View 0 - 500 ✓ 60 Display Crgate Modify Delety Cross Connects for Selected Test Manager Name Type State Endpoints (A + 8) Pkt Rx A Pkt Rx B Bps Rx A Bps Rx B Rx Dr sixteen-0000 LF/TCP Stopped sixteen-0001.4 State Sixteen-0002.8 State 0.01, 395,677 0	Fy Delete Rx B Rx Drop % Rx Drop % 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
Cross Connects for Selected Test Manager Type State Endpoints (A ↔ B) Pkt Rx A Pkt Rx B Bps Rx A Bps Rx B Rx Dr Sixteen-0000 LF/TCP Stopped sixteen-0004.A <=> sixteen-0000-B 3.347 O(B1,395,677	Rx B Rx Drop % Rx Drop % B 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
Name Type State Endpoints (A ↔ B) Pkt Rx A Pkt Rx B Bps Rx A Bps Rx B Bps Rx B A Dx Dr sixteen-0000 LF/TCP Stopped sixteen-0004.4 <-> sixteen-0000.8 3.247 0.81,395,677 0	HX B A B 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
Name Type State Endpoints (A + B) PKt KA P	HX B A B 0 0.624 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
sixteen-0001 LF/TCP Stopped sixteen-0002.4 A >>> Start Selected Minus 0 0 sixteen-0003 LF/TCP Stopped sixteen-0003.4 Start Selected 0<	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
sixteen-0001 LFTCP Stopped steten-0001.4 A<-> s Start Selected Mm.s 0 0 sixteen-0003 LFTCP Stopped sixteen-0003.4 Mm.s 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		
sixteen-0002_LF/TCP Stopped sixteen-0002.A ≤ > s Stop Selected Minus 0 0 0 sixteen-0003_LF/TCP Stopped sixteen-0004.A < > s Stop Selected 0 <td>0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>	0 0 0 0 0 0 0 0 0 0 0 0 0 0		
bitcher-0003 [JTCP] Stopped steter-0004.x x steter 000 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0		
sinteen-0004 [F/TCP Stopped] sinteen-0005.4 ≤> s [Clear Selected] 0 0 sinteen-0005. [F/TCP Stopped] sinteen-0005.4 <> s Modify Selected × 0 0 sinteen-0005. [F/TCP Stopped] sinteen-0005.4 <> s Modify Selected × 0 0 sinteen-005.7 [F/TCP Stopped] sinteen-006.4 <> s Notify Selected × 0 0 sinteen-0005. [F/TCP Stopped] sinteen-0005.4 <> s s Table Report T 0 0 sinteen-001. [F/TCP Stopped] sinteen-001.6 < <-> s Count Selected E 0 0 sinteen-001.1 [F/TCP Stopped] sinteen-001.4 < <-> s Clucidations C 0 0 sinteen-001.1 [F/TCP Stopped] sinteen-001.4 < <-> s Clucidations 0 0			
subten-0005 [F/TCP] Stopped stoten-0005.4 * 0 Modify Selected * 0 0 subten-0006 F/TCP] Stopped stoten-0005.4 > 0	0 0 0 0 0 0		
Statemologic C/T/C Stopped Statemologic Figure 3 Constraint Report D O D Statemologic L/T/C Stopped Statemologic Statemologic Constraint Report T O D Statemologic Statemologic Statemologic Constraint Statemologic Statemologic <t< td=""><td>0 0 0</td></t<>	0 0 0		
sixteen-0007 LF/TCP Stopped sixteen-0008 LF/TCP Stopped sixteen-0018 Count Selected E 0<	0 0		
andren-0008 [JF/TCP Stopped] sintern-0008A <>> s Table Report T 0 0 0 sintern-0010 [JF/TCP Stopped] sintern-001A <>> s Count Selected E 0 0 0 sintern-0010 [JF/TCP Stopped] sintern-001A <>> s Calculations C 0 0 0 sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-0010 [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] Sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] Sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] Sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] Sintern-001A <<>> s Calculations C 0 0 Sintern-001A [JF/TCP Stopped] Sintern-001A [JF/TCP [JF/TCP Stopped] Sintern-001A [JF/TCP [JF/TCP [JF/TCP Stopped] Sintern-001A [JF/TCP [JF/TC			
sixteen-0009 [FT/CP] Stopped sixteen-0000+A sixteen-0010+A e 0 0 sixteen-0011 [FT/CP] Stopped sixteen-0011-A <	0 0		
sixteen-0010 LF/TCP Stopped sixteen-0010.4 <=> s Calculations c 0 0 sixteen-0011 LF/TCP Stopped sixteen-0011.4 <=> s Calculations c 0 0 sixteen-0012 LF/TCP Stopped sixteen-0012.4 <=> s Calculations 0 0 0	0 0		
sixteen-0011 LF/TCP Stopped sixteen-0011.A <=>_5 Calculations C O O sixteen-0012 LF/TCP Stopped sixteen-0012.A <=>_5 Add/Remove Table Columns O O O			
sixteen-0012 LF/TCP Stopped sixteen-0012-A <=>_s Add/Remove Table Columns 0 0			
	0 0		
sixteen-0015 LE/TCP Stopped sixteen-0015-A <-> s Save Table Layout			
Reset Table Layout			
Auto-Resize Columns			
Enforce Fairness			
Etitolice Pairtiess			

- A. You can observe the performance of the Layer-3 connections using the Dynamic Reports window.
- B. First, use the **Rpt Timer** combo box to apply a 500ms report timer to the connection. Press **Go** to apply
- C. Next, right click the connection and select Dynamic Report (or press D)
- D. Observe the bursts of transmission.



- A. Use the Adjust button to adjust your time window:
- B. Select 30 for max-time-ago
- C. Select **0** for min-time-ago
- D. Click Apply

1.

Create Python Scripts Utilizing the Realm Library

Goal: Create a python script to create stations and Layer-3 cross connects

Using the realm.py library we will write a script that will allow us to automate the creation of stations and Layer-3 cross connects. We will also be able to start and stop traffic over the cross connects using the script. We will be referencing the script, test_ipv4_variable_time.py, as an example throughout this cookbook. Requires LANforge 5.4.2. A. Setting up inheritance for our object

A. In order for our script to be platform independent we will need to import sys. Then use

if 'py-json' not in sys.path: sys.path.append(os.path.join(os.path.abspath('..'), 'py-json'))

- B. When creating our object we will need to import the LFCliBase module from the LANforge module using from LANforge.lfcli_base import LFCliBase
- C. After importing LFCliBase we can create our Class and inherit from LFCliBase

B. Setting up the main method

- A. The main method will typically follow a pattern:
 - I. First, the creation of a list of stations. This can be done in many ways. Example: station_list = LFUtils.port_name_series(prefix_="sta",
 - start_id_=0,
 end_id_=4, padding_number_=10000) II. Following the station list, we can initialize our object: ip_var_test = IPV4VariableTime(lfjson_host, lfjson_port, number_template="00", sta list=station list,
 - name_prefix="var_time", ssid="testNet", password="testPass", resource=1,

 - security="wpa2",
 test_duration="5m"
 - side a min rate=256
 - side_b_min_rate=256)
 - III. After our object has been initialized we can begin the testing process. The preferred order for running our tests is to:
 - i. Call cleanup() to prevent stations, cross-connects, and endpoints within our list from having creation issues if anything exists with the same name.
 - ii. Call the buildO method in our class to setup the basic versions of the stations, cross-connects, and endpoints.
 - iii. Call the start() method that will start the test itself, as well as any bring up any stations and start traffic on cross-connects that need it.
 - iv. Call the stop() method to stop the traffic and bring down any stations that are up.
 - v. Verify that the tests passed using our inherited passes() method.
 - vi. After verifying a pass we can then call our cleanup function again to clean up everything we worked with.
- C. Example Main Method

LAUI	npie Main Merriod
def	main(): lfjson host = "localhost"
	lfjson port = 8080
	station list = LFUtils.portNameSeries(prefix ="sta", start id =0, end id =4, padd
	<pre>ip_var_test = IPV4VariableTime(lfjson_host, lfjson_port, number_template="00", st. name_prefix="var_time",</pre>
	ssid="testNet",
	password="testPass",
	resource=1,
	security="wpa2", test duration="5m",
	side_a_min_rate=256, side_b_min_rate=256),
	ip var test.cleanup(station list)
	ip var test.build()
	if not ip_var_test.passes():
	<pre>print(ip_var_test.get_fail_message()) exit(1)</pre>
	ip var test.start(False, False)
	ip var test.stop()
	if not ip var test.passes():
	print(ip var test.get fail message())
	exit(1)
	time.sleep(30)
	ip var test.cleanup(station list)
	if ip var test.passes():
	if ip_val_cesc.passes().

print("Full test passed, all connections increased rx bytes")

Test Methods Available With Realm

A. Using lfcli_base._pass() and lfcli_base._fail()

- A. Since our class is inheriting Ifcli_base.py, we have access to methods that will help us keep track of passes and fails during our tests. We can access them using self._pass() or self._fail(). They will take two parameters, a string message and an optional boolean print_pass and print_fail for _pass() and _fail() respectively. If print_pass or print_fail are set to True, they will write the message to stdout whenever the functions are called
- B. Ifcli_base will add a "PASSED: message" or "FAILED: message" to a list when the tests pass or fail. This list can be accessed using the methods get_result_list()
 - get_failed_result_list() qet fail message() get_all_message()

B. Using Ifcli_base to check test success

A. passes() will return a boolean depending on whether or not there were any fails in the test. If it finds a fail message it will return False, if none are found it will return True. get_result_list() will return all logged pass/fail messages as a list. get_failed_result_list() will return a list of only fail messages.

get_fail_message() will return a list of string of fail messages separated by newlines get_message() will return a list of string of all messages separated by newlines

2.

Building a Station

A. Build Method

- A. We will need to do a number of things to setup our build method.
 - I. To begin we will set the security type of our stations using station_profile.use_security()
 - II. We will then use station profile.set number template() to name our stations
 - III. After this we can set our command flags and parameters using self.station_profile.set_command_flag("add_sta","create_admin_down",1
 self.station_profile.set_command_param("set_port","report_timer",1500
 self.station_profile.set_command_flag("set_port","rpt_timer", 1)_____
 - IV. Once our parameters and flaas are set, we can pass a list of stations to station_profile.create() and cx_profile.create(). Our build function could look like this: for station in range(len(self.sta list)):

temp_sta_list.append(str(self.resource)+"."+self.sta_list[station])
self.station profile.create(resource=1, radio="wiphy0", sta names =self.sta list, debuq=Fals self.cs_profile.create(resource=, radio="wipny0", sta_names_=self.sta_list, debug=Fals self.cs_profile.create(endp_type="lf_udd", side_a=temp_sta_list, side_b="l.ethl", sleep_time self._pass("FASS: Station build finished")

1 The naming convention for the sides will look like foo-A for side a and foo-B for side b, foo will be set based on the names in the list of stations given.

B. StationProfile

A. The preferred method for creating a station_profile is to use the factory method new_station_profile() found in realm

- I. We will need to assign some variables for the creation of our stations before we can call create().
 - i. self.station_profile.use_security(security_type, ssid, passwd) is the preferred method to use when setting the security type, ssid, and password variables Example:

self.station_profile.use_security("wpa2", "testNet", "testPass")

ii. self.station_profile.number_template_ is the numerical prefix for stations. Using a number_template of "00" will have stations look like sta01, sta02...sta10 Example

self.station profile.number template ="00"

iii. self.station_profile.mode determines the wifi mode used by the stations. See here for available modes Example:

self.station_profile.mode=0

Cross Connects

- A. Starting and Stopping Traffic
 - A. In order for us to be able to start traffic, our stations will need to be admined up, associated, and with an IP. We can bring them up using station_profile.admin_up(). We can then use realm.wait_for_ip(resource, sta_list) to wait for our stations, as well as eth1, to get an IP address.
 - B. Once we are sure all of our stations have ip addresses, we can use cx_profile.start_cx() to start the traffic for our cross-connects. When we decide to stop the traffic we can just as easily use cx_profile.stop_cx() to stop traffic.

B. L3CXProfile

- A. self.local_realm.create_new_13_cx_profile() is the preferred method for creating a new Laver 3 CX Profile.
 - I. We will need to assign some variables for the creation of our stations before we can call create().
 - i. self.cx_profile.name_prefix will be used to specify the name prefix for the cx. Assigning self.cx_profile.name_prefix to "test_" would produce cross-connects named test_sta00 with the numbers being dependent on station_profile's number_template. Example:

self.cx_profile.name_prefix="test_"

ii. Set the _min_bps to the desired amount. _max_bps can be set but typically defaults to 0 which sets it to the same as the minimum bps. Example:

self.cx_profile.side_a_min_bps=56000 self.cx_profile.side_b_min_bps=56000

5

Using TTLS

A. TTLS setup requires a few pieces of information to work correctly. StationProfile has a set wifi extra() method for setting the relevant variables. See here for the available options B. We will need a key management type (key_mgmt), an EAP method (eap), an EAP identity string (identity), an EAP password string (passwd), an 802.11u realm (realm), an 802.11u domain (domain), and an 802.11u HESSID (hessid) Example: key_mgmt="WPA-EAP" eap="TTLS" identity="testuser" passwd="testpasswd" realm="localhost.localdomain" domain="localhost.localdomain" hessid="00:00:00:00:00:01" We can then use these variables to call the set_wifi_extra() method Example: station_profile.set_wifi_extra(key_mgmt, eap, identity, passwd, realm, domain, hessid]

Cleaning Up

A. Cleanup stations and cross connects

6.

7.

- A. We have two options for cleaning up everything once we finish:
 - The preferred method to cleanup is to use the individual cleanup methods found in StationProfile and L3CXProfile. These are station_profile.cleanup(resource, desired_station_list) and
 - cx_profile.cleanup(). These methods are preferred because they will only delete stations, cross-connects, and endpoints created during the test while leaving others untouched.This is useful if you are running other scripts in the background.
 - II. The other method for cleanup is to use Realm's remove_all_stations(), remove_all_endps(), and remove_all_cxs() methods. These will remove all stations, cxs, and endpoints that exist. These are good for doing a full cleanup, and it is recommended to use them in the order of cx, endpoint, station to prevent potential issues or missed deletions.

Debugging Stations

- A. Debug information for station creation can be output by setting <u>_debug_on=True</u> in StationProfile.create()
 - A. There are a few important debug outputs to pay attention to:
 - I. This is the debug output that appears when using the add_sta command. This is used frequently in StationProfile.create(). This debug output will allow you to troubleshoot any flags or other information that is being set when creating your stations. It will output the name at the top and the raw JSON data will follow.
 - 381 sta0000-('flag : 132096, 'flag mak': 68719608832, 'key': 'testPass', 'mode': 0,' mode': 0,' mode': 0,' radio': 'viphy0', 'resource': 1, 'shelf': 1, 'shelf': 1, 'shelf': 'testNet, 'sta_mame': 'sta0000')
 - The next bit of debugging output comes from using the set_port command. We are able to see all of the JSON data that is posted, and can use this to check our flags and other info.
 - ('current_flags': 2147483649, 'interset': 8437762, 'report': ista0000', 'resource': 1, 'heilf': 1) CLANforge.LFRequest.LFRequest object at 0x7f13dbc56850> -381
- B. There are a few steps we can take to make validating the information we get through debugging easier.
 - A. We can use the help page available on the address of the machine LANforge is running on. http://127.0.0.1/help/ will take us to a page containing all of the commands we can get help with.

et help with.	
LANforge CLI Help - Mozilla	Firefox
LANforge CLI Help 🛛 🗡	+
\leftrightarrow > C $\textcircled{0}$	① 127.0.0.1:8080/help/
add arm endp	CLI Reference for /help/add arm endp
add bgp peer	CLI Reference for /help/add bgp peer
add bond	CLI Reference for /help/add_bond
add br	CLI Reference for /help/add_br
add cd	CLI Reference for /help/add_cd
add_cd_endp	CLI Reference for /help/add_cd_endp
add_cd_vr	CLI Reference for /help/add_cd_vr
add_chamber	CLI Reference for /help/add_chamber
add_chamber_cx	CLI Reference for /help/add_chamber_cx
add chamber path	CLI Reference for /help/add_chamber_path
<u>add_channel_group</u>	<u>CLI Reference for /help/add_channel_group</u>
add_cx	CLI Reference for /help/add_cx
add_dut	CLI Reference for /help/add_dut
add_dut_notes	CLI Reference for /help/add_dut_notes
add_endp	CLI Reference for /help/add_endp
add_event	CLI Reference for /help/add_event

B. Using http://127.0.0.1/help/add_sta will bring us to a page specific to the add_sta

command.	

LANforge CLI Help -	Mozilla Firefox					000
	× +					
< → ୯ û	0 3 127.0.0	.1:8080/help/add_sta		⊠ ☆		• * =
	Co	mmand Com	poser [ad	d_sta]		
These are the curl o	commands:					
echo "" > /tmp/curl curl -sqv -H "Accep		n" -X POST -d '@/tmp/curl_da	ata' http://ctlt2-log	gan:8080/cli-fo	rm/add_sta	
This is the JSON ve	ersion:					
echo "}" > /tmp/jso curl -sqv -H "Accep		on" -H "Content-type: applica	ation/json" -X POST	-d '@/tmp/json	_data' http://cf	tlt2-logan:8080
This is the CLI com	mand:					
			li			
Parse Command						

C. Here we can enter all of the data we got from our debugging output into the correct areas.

Fiel	ds for the cor	nmand will update when you change t	hem:	Flag Fields for command will be computed when you select them, but you might need to a values into some fields (when you see token values like [string] or [name]).		
	resource	1		flags.802111_pmska_cache flags.80211u_addlional flags.80211u_auto flags.80211u_e911		
03:	radio	wiphy0		- flags.80211u_e911_unauth flags.80211u_enable		
04:	sta_name	sta0000		flags.80211u_gw flags.8021x_radius flags.create admin down		
36:	flags	132096		flags.custom_conf flags.disable_fast_reauth		
06:	ssid	testNet		flags.disable_gdaf flags.disable_ht80		
07:	nickname	NA		flags.disable_roam flags.disable_sgi - flags.hs20 enable		
	key	testPass		flags.ht160_enable flags.ht40_disable		
	ар	NA		flags.ibss_mode flags.lf sta migrate		
	wpa_clg_file	NA		flags.mesh_mode flags.no-supp-op-class-ie flags.soen enable		
	mac	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		flags.osen_enable flags.passive_scan flags.power save enable		
	mode	0	*	flags.scan_ssid		
	rate	NA	*	flags.use-wpa3 flags.verbose		
	max_amsdu	NA		flags.wds-mode flags.wep_enable		
	ampdu_factor	NA		flags.wpa2_enable flags.wpa_enable mode.802.11a		
	ampdu_density	NA		mode.AUTO mode.abg		
	sta_br_ip	NA		mode.abgn mode.abgnAC		
	flags_mask	68719608832		mode.abgnAX mode.an		
	ieee80211w	NA		mode.anAC mode.anAX mode.b		
	x_coord	NA		mode.bg mode.bgn		
	y_coord	NA		mode.bgnAC mode.bgnAX		
22:	z_coord	NA		mode.g rate./a/g		

D. Flag fields have a button next to them that will calculate and highlight relevant flags in the right hand column of the page. This can be useful for checking that the correct flags are being set.

ields for the co	mmand will update when you change them:	Flag Fields for command will be computed when you select them, but you might need values into some fields (when you see token values like [string] or [name]).
01: shelf	1	flags.80211r_pmska_cache flags.802111_additional
02: resource	1	flags.80211u_auto flags.80211u_e911
03: radio	wiphy0	[ags.80211u_e911_unauth [lags.80211u_enable [lags.80211u_ww
04: sta_name	sta0000	flags.8021n_radius flags.create admin down
05: flags	132096	flags custom conf
06: ssid	testNet	flags.disable_gdaf flags.disable_ht80
07: nickname	NA	flags.disable_roam flags.disable_sgi
08: key	testPass	flags.hs20_enable flags.ht160_enable flags.ht40_disable
09: ap	NA	flags.ifss_mode flags.if sta_migrate
10: wpa_cfg_file	NA	flags.mesh_mode flags.no-supp-op-class-ie
11: mac	XXXXXXXXXX	flags.osen_enable flags.passive_scan
12: mode	0	flags.power_save_enable flags.scan_ssid
13: rate	NA	flags.txo-enable flags.use-wpa3 flags.verbose
14: max_amsdu	NA	flags.webse flags.websenode flags.wep enable
15: ampdu_factor	NA	flags.wpa2_enable flags.wpa_enable
16: ampdu_density	NA	mode.802.11a mode.AUTO
17: sta_br_ip	NA	mode.abg mode.abgn mode.abgnAC
18: flags_mask	68719608832	mode.abgnAX mode.an
19: ieee80211w	NA	mode.anAC mode.anAX
20: x_coord	NA	mode.b mode.bg
21: y_coord	NA	mode.bgnAC
22: z_coord	NA	mode.bgnAX mode.g rate./a/q

E. After we have done this, we can click the **parse command** button towards the top of the data inputs. We can then enter this command into LANforge's messages tab in the input box.

Command Composer [add_sta]

These are the curl commands ente "Nalf-Stevarre-Dodlavskyhjälla vane-Nalf886/hgs-13388asid-tatterlägs-tattpakas-tattrakas-structurt" nääde-68/hgs_mak-6873988337 × /mg/crt_data cott syn = %cott syn = %cott syn syn = %cott syn =

cebs "[Cite[Fil/reserver11/redsr/wdpyr_Cta_samtriste0000",Flap:li2000,ridf="textNet","http:/flap.hst/iestriste0000;"/[apr.back":estriste00000;"/[apr.back":estriste00000;"/[apr.back":estriste00000;"/]] > /tsp/jos_data coll sep =#/cospir_plication/jue*=#_Costent-type:aplication/jue*==#.2007.ed="textNet".cospir_bub/clic_loger.bbl/cl This is the CLI command:

Parse Command

Create Python Scripts To Test Layer 4 <u>Traffic</u>

Goal: Create a script to test Layer 4 traffic using Realm

Using the realm.py library we will write a script that will allow us to automate the creation of stations and Layer 4 cross connects. We will also be able to start and stop traffic over the cross connects using the script. Station and Cross Connect creation is covered in the Realm Scripting Cookbook. Requires LANforge 5.4.2.

	Creating The Profile
A. We w objec	<pre>ill use the factory method self.local_realm.new_l4_cx_profile() to create our profile t.</pre>
B. After	we have done this we can set a few variables for our traffic:
	14_cx_profile.requests_per_ten will set our rate of requests per ten minutes. Setting requests_per_ten = 600 will set our URL request rate to 1 per second. There is no limit to what can be used as the rate but common rates are:
	■ 600:1/s
	■ 1200:2/s
	■ 1800 : 3/s
	2400 : 4/s
	14_cx_profile.url is the URL to be used in the requests. We will also need to specify the direction (al/ul) and a absolute path for the destination. See syntax here. Example:
	l4_cx_profile.url = "dl http://10.40.0.1 /dev/null"
	<pre>class IFV414(LFCliBase): definit(self, host, port, ssid, security, password, url, requests_pe target_requests_per_ten=600, number_template="00000", resource=1, num debug_on=False, exit_on_error=False, exit_on_fail=False): super()init(host, port, _debug=_debug_on, _halt_on_error=_exit_on_er self.host = host self.port = port self.sid = ssid self.sid = ssid self.security = security self.security = security self.security = security self.requests_per_ten = requests_per_ten self.requests_per_ten = requests_per_ten self.requests = number_template self.resource = resource self.num_tests = num_tests self.target_requests_per_ten = target_requests_per_ten self.local realm = realm.Realm(lfclient host=self.host, lfclient port=sel</pre>
	<pre>self.cx_profile = self.local_realm.new_l4_cx_profile() self.cx_profile.url = self.url self.cx_profile.requests per ten = self.requests per ten</pre>

2.

Starting Traffic

A. When running traffic, if you plan to measure the rate of requests, it is recommended to do so in 10 minute increments. An example of this can be seen here: test_ipv4_l4_urls_per_ten.py. To start the traffic we can use the 14_cx_profile.start_cx() method. To stop the traffic we can use the 14_cx_profile.start_cx() method.

<pre>def start(self, print_pass=False, print_fail=False): temp_stas = self.sta_list.copy() temp_stas.append("eth") cur_time = datetime.datetime.tow() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.l4_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.now() if self.l4_profile.check_errors(self.debug): if selfdprofile.check_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: test</pre>	def	build(self):					
<pre>print("Creating stations") self.station profile.create(resource=1, radio="wiphy0", sta_names_=self.sta_list temp_sta_list_station in range(len(self.sta_list)): temp_sta_list.append(str(self.resource) + "." + self.sta_list[station]) self.14_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, ss def start(self, print_pass=False, print_fail=False): temp_stas.append("eth1") cur_time = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time_slaep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfdprofile.check_errors(self.debug): if selff_check_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fai break interval_time = cur_time + datetime.timedelta(minutes=1) vark interval_time = cur_time + datetime.timedelta(minutes=1) vark interval_time = cur_time + datetime.timedelta(minutes=1) vark interval_time = cur_time + datetime.timedelta(minutes=1) interval_time = cur_time + datetime.timedelta(minutes=</pre>		# Build stations					
<pre>self.station_profile.create(resource=1, radio="wiphy0", sta_names_=self.sta_list temp_sta_list = [] for station in range(len(self.sta_list)): temp_sta_list.append(str(self.resource) + "." + self.sta_list[station]) self.14_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, si temp_stas = self.sta_list.copy() temp_stas.append("eth1") cur_time = datetime.datetime.timedelta(minutes=1) passes = 0 expected passes = 0 self.station_profile.admin_up(1) self.ld_profile.stat_cx() print("Starting test") for test in range(self.num_tests): expected passes += 1 while cur_time < datetime.now() if self.14_profile.check_errors(self.debug): if self.14_profile.check_errors(self.debug): if self.14_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		self.station profile.use security(self.security, self.ssid, self.password)					
<pre>temp_sta_list = [] for station in range(len(self.sta_list)): temp_sta_list.append(str(self.resource) + "." + self.sta_list[station]) self.14_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, st temp_stas = self.sta_list.copy() temp_stas.append("eth1") cur_time = datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.ld_profile.stat_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time_sleep(1) cur_time = datetime.now() if self.l4_profile.check_errors(self.debug): if self.l4_profile.check_errors(self.debug): if selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) rest = cur_time +</pre>		print ("Creating stations")					
<pre>temp_sta_list = [] for station in range(len(self.sta_list)): temp_sta_list.append(str(self.resource) + "." + self.sta_list[station]) self.14_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, st temp_stas = self.sta_list.copy() temp_stas.append("eth1") cur_time = datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.ld_profile.stat_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time_sleep(1) cur_time = datetime.now() if self.l4_profile.check_errors(self.debug): if self.l4_profile.check_errors(self.debug): if selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) rest = cur_time +</pre>		self.station profile.create(resource=1, radio="wiphy0", sta names =self.sta list					
<pre>for Tation in range(len(self.sta_list)): tem_sta_list.append(str(self.resource) + "." + self.sta_list[station]) self.14_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, ss def start(self, print_pass=False, print_fail=False): temp_stas.append("eth1") cur_time = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: self.fail("FAIL: Request rate did not exceed 90% target rate", prin break else: self.fail("FAIL: Errors found getting to %s " % self.url, print_fails break interval_time = cur_time + datetime.timedelta(minutes=1) } } </pre>							
<pre>temp_sta_list.append(str(self.resource) + "." + self.sta_list[station]) self.14_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, ss def start(self, print_pass=False, print_fail=False): temp_stas = self.sta_list.copy() temp_stas.append("ethl") cur_time = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.ld_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < datetime.now() if self.14_profile.check_errors(self.debug): if selfdeprofile.check_errors(self.debug): if selfdeprofile.check_errors(self.debug): if selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) for % % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) for % % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) for % % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) interval_time = cur_time + datetime.timedelta(minutes=1) </pre>							
<pre>def start(self, print_pass=False, print_fail=False): temp_stas = self.sta_list.copy() temp_stas.append("ethl") cur_time = datetime.datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.ld_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfdebck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) } }</pre>		<pre>temp_sta_list.append(str(self.resource) + "." + self.sta_list[station])</pre>					
<pre>temp_stas = self.sta_list.copy() temp_stas.append("ethl") cur_time = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.now() if self.14_profile.check_errors(self.debug): if self.14_profile.check_errors(self.debug): if self.14_profile.check_errors(self.debug): if selfdatetime.trate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		<pre>self.l4_profile.create(ports=temp_sta_list, sleep_time=.5, debug_=self.debug, su</pre>					
<pre>temp_stas = self.sta_list.copy() temp_stas.append("ethl") cur_time = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.now() if self.14_profile.check_errors(self.debug): if self.14_profile.check_errors(self.debug): if self.14_profile.check_errors(self.debug): if selfdatetime.trate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>							
<pre>temp_stas.append("ethl") cur_time = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.ld_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fails break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>	def	<pre>start(self, print pass=False, print fail=False):</pre>					
<pre>cur_ime = datetime.datetime.now() interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.14_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fails break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		temp stas = self.sta list.copy()					
<pre>interval_time = cur_time + datetime.timedelta(minutes=1) passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.now() if self.14_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		temp_stas.append("eth1")					
<pre>passes = 0 expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.ld_profile.start_cx(1) print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		<pre>cur_time = datetime.datetime.now()</pre>					
<pre>expected_passes = 0 self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.14_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.14_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa. break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		<pre>interval_time = cur_time + datetime.timedelta(minutes=1)</pre>					
<pre>self.station_profile.admin_up(1) self.local_realm.wait_for_ip(self.resource, temp_stas) self.l4_profile.statt_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		passes = 0					
<pre>self.local_realm.wait_for_ip(self.resource, temp_stas) self.ld_profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_pases += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.ld_profile.check_errors(self.debug): if selfcheck_request_rate(): pases += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		expected_passes = 0					
<pre>self.14 profile.start_cx() print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.14 profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Rerors found getting to %s " % self.url, print_fa. break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		self.station_profile.admin_up(1)					
<pre>print("Starting test") for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Rerors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		<pre>self.local_realm.wait_for_ip(self.resource, temp_stas)</pre>					
<pre>for test in range(self.num_tests): expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", print break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		<pre>self.l4_profile.start_cx()</pre>					
<pre>expected_passes += 1 while cur_time < interval_time: time.sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Rerors found getting to %s " % self.url, print_fa. break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		print("Starting test")					
<pre>while cur_time < interval_time: time sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		for test in range(self.num_tests):					
<pre>time.sleep(1) cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Rerors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>							
<pre>cur_time = datetime.datetime.now() if self.l4_profile.check_errors(self.debug): if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa. break interval_time = cur_time + datetime.timedelta(minutes=1) </pre>		while cur_time < interval_time:					
<pre></pre>		time.sleep(1)					
<pre>if selfcheck_request_rate(): passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa. break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		<pre>cur_time = datetime.datetime.now()</pre>					
<pre>passes += 1 else: selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>							
<pre>selffail("FAIL: Request rate did not exceed 90% target rate", prin break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>							
break else: selffail("FAIL: Errors found getting to %s " % self.url, print_fa break interval_time = cur_time + datetime.timedelta(minutes=1)		else:					
<pre>selffail("FAIL: Errors found getting to %s " % self.url, print_fa. break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		selffail("FAIL: Request rate did not exceed 90% target rate", prir break					
<pre>break interval_time = cur_time + datetime.timedelta(minutes=1)</pre>		else:					
		selffail("FAIL: Errors found getting to %s " % self.url, print_fai break					
if passes == expected passes:		interval_time = cur_time + datetime.timedelta(minutes=1)					

3.

1.

Examining The Results

- A. We can use http://localhost:8080/layer4/list to check our Layer 4 endpoints. Adding a ,?fields to the end of the URL will allow us to specify what we want to look at. We can separate fields by commas to show more than one at a time.
 - Example: http:/localhost:8080/layer4/list?fields=name,urls/s,total-urls
 - Using total-urls will show us the total requests made.
 - Using urls/s will show us the average URL rate per second.
 - Using rx rate and tx rate will show us the rates of received and transeferred traffic.

We can also use the url http://localhost:8080/layer4/all to see all of the available fields.

- B. When checking our results for Layer 4 tests we might want to check for common URL related errors:
 - acc. denied will show us the number of times we got an access denied error.
 - **bad-url** will show us the number of times a request was made with an invalid URL.
 - nf (4xx) will count the number of 400 errors recieved when making requests to our URL.

Create Python Scripts To Test Generic Traffic

Goal: Create a script to test Generic traffic using Realm

Using the realm.py library we will write a script that will allow us to automate the creation of stations and generic cross connects. We will also be able to start and stop traffic over the cross connects using the script. Station and Cross Connect creation is covered in the Realm Scripting Cookbook. Requires LANforge 5.4.2.

Creating The Profile

A. We will use the factory method self.local_realm.new_generic_cx_profile() to create our profile object.

B. After we have done this we can set a few variables for our traffic:

- A. gen_cx_profile.type will determine the type of command to execute.
- Example: self.cx_profile.type = "lfping"
- B. gen_cx_profile.dest is the destination IP address for the command. Example: self.cx_profile.dest = "127.0.0.1"
- C. gen_cx_profile.interval sets the interval at which the command is run in seconds. Example: self.cx_profile.interval = 1

```
D. Example Generic profile init:
```

```
self.host = host
self.port = port
self.ssid = ssid
self.radio = radio
self.upstream = upstream
self.sta_list = sta_list
self.security = security
self.password = password
                self.number_template = number_template
self.name_prefix = name_prefix
self.test_duration = test_duration
                self.local_realm = realm.Realm(lfclient_host=self.host, lfclient_port
self.cx_profile = self.local_realm.new_generic_cx_profile()
self.cx_profile.type = type
self.cx_profile.dest = dest
self.cx_profile.interval = interval
        # Station Profile init
```

```
2.
```

3.

Starting Traffic

- A. To start the traffic we can use the gen_cx_profile.start_cx() method. To stop the traffic we can use the gen_cx_profile.stop_cx() method.
- B. Example start and build method

Exar	nple start and build method:
def	build(self):
	self.station profile.use security(self.security, self.ssid, self.password)
	self.station profile.set number template(self.number template)
	print ("Creating stations")
	self.station profile.set command flag("add sta", "create admin down", 1)
	self.station profile.set command param("set port", "report timer", 1500)
	self.station profile.set command flag("set port", "rpt timer", 1)
	<pre>self.station profile.create(radio=self.radio, sta names =self.sta list, debug=sel;</pre>
	<pre>self.station_profile.create(profile.station profile.station names, sleep time=.5)</pre>
	<pre>self. pass("PASS: Station build finished")</pre>
C	seri. pass (TROS. Station build finished)
dof	<pre>start(self, print pass=False, print fail=False):</pre>
uer	
	<pre>self.station_profile.admin_up()</pre>
	<pre>temp_stas = self.sta_list.copy()</pre>
	temp_stas.append(self.upstream)
	if self.local_realm.wait_for_ip(temp_stas):
	<pre>selfpass("All stations got IPs", print_pass)</pre>
	else:
	selffail("Stations failed to get IPs", print_fail)
	exit(1)
	<pre>cur_time = datetime.datetime.now()</pre>
	passes = 0
	expected_passes = 0
	<pre>self.cx_profile.start_cx()</pre>
	time.sleep(15)
	end_time = self.local_realm.parse_time("30s") + cur_time
	print("Starting Test")
	while cur time < end time:
	cur time = datetime.datetime.now()
	gen results = self.json get("generic/list?fields=name,last+results", debug =s€
	if gen results['endpoints'] is not None:
	for name in gen results['endpoints']:
	for k, v in name.items():
	if v['name'] in self.cx profile.created endp and not v['name'].end
	expected passes += 1
	if v['last results'] != "" and "Unreachable" not in v['last re
	passes += 1
	else:
	<pre>selffail("%s Failed to ping %s " % (v['name'], self.cx_j</pre>
	break
	time.sleep(1)
	if passes == expected_passes:
	self. pass("PASS: All tests passed", print pass)

Examining The Results

For lfping we can use the last results of the endpoint to determine if the test was successful. An
example of this can be seen in our start method. The most common errors for lfping will either be
a blank last result or Destination Host Unreachable . Either of these results indicate a failed ping.
Successful pings will look like:
64 bytes from 10.40.0.1: icmp_seq=1 time=4.55 ms *** drop: 0 (0, 0.000) rx: 1 fail:
Results can also be seen in the generic tab in the LANforge Manager:
LANforge Manager Version(5.4.2)
Control Reporting Windows Info Tests
Chamber ⊻iew Stop All Restart Manager Refresh HELP
Generic Resource Mgr VAP Stations DUT Profiles Traffic-Profiles Alerts Messages Warnings Wifi-Messages + Status Port Mgr Layer-3 L3 Endps Layer 4-7 Armageddon WanLinks VolP/RTP VolP/RTP Endps File-10
Select All Start + Stop · Clear
Rpt Timer: fast (1 s) V Go Test Manager all V Create Modify Delete
Generic Endpoints for Selected Test Manager
Name EID Status Rpt# Last Results Tx Bytes Rx Bytes Tx Pkts PDU/s TX Rx sta0000_gen0 1.1.13.5 Stop 206 64 bytes from 10.40.0.1: icmp_seq=204 time=3.75 ms *** dr 0 B 0 B 0 0 0
sta0001_gen0 [1.1.14.7 Stop 206 64 bytes from 10.40.0.1: icmp_seq=204 time=1.61 ms *** dr 0 B 0 B 0 0 0
Logged in to: 192.168.92.12:4002 as: Admin
Double-clicking on an endpoint will allow you to see more specific results as well as the command
used by the endpoint. Using the sync button will allow you to see updated results.
Create/Modify Generic Endpoint 😑 🗊 🖉
Name: sta0000_gen0 Rpt Timer: default (5 s) 🗸 Test Manager: default_tm
Shelf: 1 Resource: 1 (ct524-emily) Port: 13 (sta0000) Endp ID:
Command Builders ping 👻
Size: Default 🔻 Interval: 1000000 (1 s) 💌 Count: Infinite 💌
How big should the ping packets be?
Target: 10.40.0.1
Payload:
Command: [lfping -i 1 -i sta0000 10.40.0.1
Command Output 64 bytes from 10.40.0.1: icmp_seq=105 time=2.36 ms = 40 (0,000 rr: 105 fail: 0 bytes: 12802
64 bytes from 10.40.0.1: icmp_seq=109 time=2.73 ms ^{xxxx} forp: 0 (0, 0.000) r.x: 109 fail: 0 bytes: 12096 64 bytes from 10.40.0.1: icmp_seq=109 time=2.73 ms ^{xxxx} forp: 0 (0, 0.000) r.x: 109 fail: 0 bytes: 12096
64 bytes from 10.40.0.1: icmp_seqw191 times.107 mm **** drop: 0 (0, 0.000) rx: 192 fail: 0 bytes: 12224 64 bytes from 10.40.0.1: icmp_seqw192 times.197 mm **** drop: 0 (0, 0.000) rx: 192 fail: 0 bytes: 12288
64 bytes from 10.40.0.1:icmp_seq=193 time=2.30 mm 5××× drop: 0 (0, 0.000) rr: 193 fail: 0 bytes: 1232 64 bytes from 10.40.0.1:icmp_seq=194 time=1.90 ms ××× drop: 0 (0, 0.000) rr: 194 fail: 0 bytes: 12416
64 bytes from 10.40.0.1: icmg.seq=195 time=2.11 ms ^{xxxx} drop: 0 (0, 0.000) rx: 195 fail: 0 bytes: 12420 64 bytes from 10.40.01: icmg.seq=106 time=2.20 ms ^{xxxx} drop: 0 (0, 0.000) rz: 195 fail: 0 bytes: 12544
64 bytes from 10.40.0.1: icnp_seq=137 time=1.69 ms **** drop: 0 (0, 0.000) rx: 197 fail: 0 bytes: 12606 64 bytes from 10.40.0.1: icnp seq=137 time=1.69 ms **** drop: 0 (0, 0.000) rx: 197 fail: 0 bytes: 12606
64 bytes from 10.40.0.1: icnp_seq=199 time=1.97 mm **** drop: 0 (0, 0.000) rx: 199 fail: 0 bytes: 12736 64 bytes from 10.40.0.1: icnp_seq=200 time=4.56 mm **** drop: 0 (0, 0.000) rx: 200 fail: 0 bytes: 12800
64 bytes from 10.40.0.1: icmp_seq=201 time>1.94 ms ^{seex} drop: 0 (0, 0.000) rx: 201 fail: 0 bytes: 12264 64 bytes from 10.40.0.1: icmp_seq=202 time>0.1 ms ^{seex} drop: 0 (0, 0.000) rx: 202 fail: 0 bytes: 12928
64 bytes from 10.40.0.1; icmp.seq=203 time=1.93 ms ^{xxxx} drop: 0 (0, 0.000) rx: 203 fail: 0 bytes: 12922 64 bytes from 10.40.0.1; icmp.seq=204 time=2.75 ms ^{xxxx} drop: 0 (0, 0.000) rx: 204 fail: 0 bytes: 13956
64 bytes from 10.40.0.1: icmp_seq=204 time=3.75 ms ××× drop: 0 (0, 0.000) rx: 204 fail: 0 bytes: 13056

Automate The Creation of VAPs With The Realm Python Library

Goal: Create a python script to create VAPs

Using the realm.py library we will write a script that will allow us to automate the creation of VAPs. Requires LANforge 5.4.2

VAPProfile	
A. The preferred method for creating a vap_profile is to use the factonew_vap_profile() found in realm	bry method
 We will need to set the name of our vap using vap_profile.va Example: 	ap_name
<pre>vap_profile.vap_name = "TestNet"</pre>	
II. vap_profile.use_security(security_type, ssid, pass preferred method to use when setting the security type, ssid, a variables. Available security types are wpa, wpa2, wpa3, wep Example: vap_profile.use_security(type="wpa2", ssid="testNo")	nd password , and open.
III. We can change the mode at any time before calling create(vap_profile.mode variable. Changing the mode will allow us 802.11 wireless standard the VAP uses. See here for available in Example: vap_profile.mode = 1	to specify the
IV. The channel to be used by the VAP can be set with the channel the create() method. Example:	nel parameter of

Bringing VAPs Up/Down

A. vap_profile.admin_up() and vap_profile.admin_down() can be used to bring the VAP up or down, as necessary.

	Using TTLS
A.	TTLS setup requires a few pieces of information to work correctly. VAPProfile has a <pre>set_wifi_extraO</pre> method for setting the relevant variables. See <pre>here</pre> for the available options
Β.	We will need a key management type (key_mgmt), an EAP method (eap), an EAP identity string (identity), an EAP password string (passwd), an 802.11u realm (realm), an 802.11u domain (domain), and an 802.11u HESSID (hessid)
	Example:
	key_mgmt="WPA-EAP" eap="TTLS" identity="testuser" passwd="testpasswd" realm="localhost.localdomain" domain="localhost.localdomain" hessid="00:00:00:00:00:01"
	We can then use these variables to call the set_wifi_extra() method
	Example:
	vap profile.set wifi extra(key mgmt, eap, identity, passwd, realm, domain, hessid

Cleaning Up

A. vap_profile.cleanup() can be used to remove any VAPs that were created by the profile

Load Scenarios And Control Test Groups With Python

Goal: Using a python script to load scenarios and start, stop, and quiesce test groups

This cookbook will demonstrate how we can use json to load DB scenarios and control test groups using python. We will be referencing the script scenario.py. Requires LANforge 5.4.2.

Running The Script

2.

4.

1.

A. Setting up

A. For this example we are using a database called fio_test_group. It has two stati	ons that
each have a read-only and write-only file-io endpoint attached to them.	

ANforge Manager	Version(5.4.3	3)							~ ^ >
ontrol Beporting	Windows Info								
Generic Test Gr		Chamber	View Profiles Tr	Stop All	Restar ts Warning	t Manager	lessages	<u>R</u> efresh	HELP
Status Port M		L3 Endps		Armageddon	WanLinks	VoIP/RTP	VoIP/RT	P Endps	File-IO
Disp: 10.253.1.4	6:0	Sniff Packe	ts 🖌	Down 1 Clear	Counters	Reset Port	Delete		
Rpt Timer: mediu	n (8s) 🔻	Apply		VRF I D	splay	Cr <u>e</u> ate	Mo <u>d</u> ify	Batch N	todify
			All Ethernet In	erfaces (Ports) for				-	
Ø I IP	Alias	Parent Dev	AP Channel	SSID	MAC				
192.168.92.	14 eth0				0c:c4:7a:e2:0				
0.0.0.0	eth1				0c:c4:7a:e2:0	01:e7			
10.40.2.47	sta0000 wiphy2	wiphy0	153	jedway-r8000-15	3 00:0e:8e:4e:7 04:10:21:20:3	7a:47 37:03			
10.40.9.142	sta0001 wiphy1	wiphy0	-1	jedway-r8000-15	3 00:0e:8e:4e:4				
0.0.0.0	wiphy0		0		00:0e:8e:4e:5	59:47			
0.0.0.0	wiphy3		0		00:19:70:be:6	52:2d			
gged in to: 192.1	58.92.14:4002	as: Admin						2 stations	:210I
ANforge Manager	Version(5.4.	3)							v . ^
ontrol <u>Beporting</u>	Windows Info	Tests		_	_				
		Chamber	2	<u>S</u> top All		t Manager		<u>R</u> efresh	HELF
Generic Test Gr Status Port N		Mgr DUT	Profiles Ti	affic-Profiles Ale	ts Warning WanLinks	s + Wifi-	lessages	R Endos	File-IO
Status Port N	lgr Layer-3	L3 Endps	Layer 4-7	Armageddon				P Endps	
Rpt Timer:	fast (1 s)	- Go Te	st Manager a	· •	Select All			sce Clea	
					Crgate	Mo <u>d</u> ify B	atch Modify	Delete	
			- Cross Conn	ects for Selected T	est Manager-				
Name	EID Typ	e Status	Read-Bps Rx-	Bps-20s Files-Read	Buf-RD Byte	es-RD Write-B	os Tx-Bps-20	s Files-WR	Buf-WR
o_sta0000_fio 1	.1.2.127 NFSv	4 Stopped	0		0 0	0		0 0	0
o_sta0001_fio 1 vo_sta0000_fio 1	.1.4.128 NFSv .1.2.125 NFSv	4 Stopped 4 Stopped	0			0		0 0	0
o_sta0001_fio 1	.1.4.126 NFSv	4 Stopped	0		0 0	0		0 0	0
her is test-c ANIONA Montage ontrol Reporting Generic Test Gr Status Port N Name Ru est-group-to est-group-to	r Version(G.4. Windows info oup Resource figr Layer-3 Rpt Timer: [None CX_ro_	Tests Chamber Mgr DUT L3 Endps fast (1 s Cross sta0000_fio C	Profiles Tr Layer 4-7	Armageddon Select All Display Nhforge Test Group	ts Warnings WanLinks Start + Sto r Crgate s r Test Group ime: test-gr	VoIP/RTP pp - Quiesce Mogify De	essages VolP/RTF ete Script :ts (CX)	efresh Endps Confi Free C X_wo_sta0C X_wo_sta0C	Xs 000 fio
ogged in to: 192.1 .ANforge Manage Control <u>B</u> eporting	r Version(5.4.)	3)							v . A
		Chambe		Stop All		t Manager		<u>R</u> efresh	HEL
Generic Test Gr Status Port N	lgr Layer-3	L3 Endps		Armageddon	WanLinks	s + Wifi-1 s VoIP/RTF op - Quiesc		P Endps	File-IO
	Rpt Timer:				y Cr <u>e</u> ate		elete		
test-group-ro	n Script None CX_ro_	sta0000 fio (Connects X_ro_sta0001_	tio	ify Test Group				× ^
test-group-wo	None CX_wo	_sta0000_fio	CX_wo_sta0001	_fio Test Group		group-wo Cross Conne	Script ects (CX)	🔲 Confi	
				Registe CX_wo_st: CX_wo_st:	ered CXs a0000_fio a0001_fio			Free CX_ro_sta0 CX_ro_sta0	0000_fio
		an Adapta							

2.

Script Examples

A. The Command and Available Options

A. The script is located in lanforge-scripts/py-scripts/. From that directory we can use ./scenario.py to run the script.

- B. The available options are:
 - 1. --load db_name This will load the database named db_name
 - --action (overwrite, append) | Optional argument to be used with --load, will specify an action to take when loading the database. The default action is to overwrite. The append option is more difficult to use and its use is discouraged. See here for more info.
 - 3. --clean_dut | Optional argument to be used with --load, will cleanup DUTs on load. See here for more info.
 - 4. --clean_chambers | Optional argument to be used with --load, will cleanup Chambers on load. See here for more info.
 - 5. --start group_name | This will start the cross-connects in the specified group
 - 6. --stop group_name | This will stop the cross-connects in the specified group
 - 7. --quiesce group_name | This will quiesce the cross-connects in the specified group

B. Examples of Running the Script

Loading fio_test_group with overwrite



Starting test-group-ro



Stopping test-group-ro



Quiescing test-group-wo



Record the results of a test as CSV from the REALM monitor script

Goal: Record the results of a LANforge test as a CSV file.

Some scripts in the LANforge library have a monitor function built in. We are going to be using the test_jpv4_variable_time script for this demonstration. This is useful for running a test and then analyzing the results afterwards.

^{1.} Start LANforge GUI. It is recommended to run this script on a fresh LANforge configuration with no stations loaded.

^{2.} Make sure you have lanforge-scripts on your device. If lanforge-scripts is already installed on your device, skip this step Navigate to py-scripts in the lanforge-scripts folder. If your LANforge device doesn't have this open source software yet you can clone them from Github To install lanforge-scripts paste git clone https://github.com/greearb/lanforge-scripts into your terminal.

Type the following command into your command line
 ./test_ipv4_variable_time.py --radio wiphy0 --security wpa2 --ssid lanforge --password
 password --output_format csv

Replace the security, ssid, and password variables with the settings for the network you are testing. This will create 2 wiphy stations by default, connect them to the network you are testing, and report the results to a CSV file.You can change the following fields in the Realm Monitor function:

^{4.} This creates a default file in your report-data folder under your home directory. The name will be in the format with today's timestamp and the name of the test you ran. It's a normal Excel file which you can use however you want..

- There are multiple commands you can use with this function, here is a list of the flag and what each of them mean:
 - A. report_file: Name the full path of the file you want to save results to. Default will save to your reportdata folder.
 - B. duration_sec: how long you want to run the test
 - C. output_format: The output format you want your file in. The following formats are supported:
 - A. xlsx DEFAULT
 - B. pickle
 - HINT: pickle is recommended if you are going to be manipulating data in python since it preserves formatting and can be quickly loaded into a Pandas DataFrame without any manipulation required
 - C. csv
 - D. json
 - E. pdf
 - WARNING: PDF is hard to export data from without an Adobe Acrobat license
 - F. png WARNING: png is going to export an image, do not use this if you are planning on manipulating your data because it does not preserve the numbers recorded
 - G. html
 - H. hdf
 - I. parquet
 - J. stata
 - D. ssid: REQUIRED Name of the network you are connecting to
 - E. password: REQUIRED Password to the network
 - F. radio: REQUIRED The radio which you are going to create stations from.
 - G. security: Match the security protocol of your router.
 - H. test_duration: Default is 60 seconds, write in a any number if you need. You can also use minutes or hours notation in this command, so for 42 minutes write 42m and for 8 hours write 8h.
 - upstream_port: Most users won't need to use this option, but it tells the program where to connect to the router
 - J. created_cx: List of the cross connects you are going to be analyzing. If you are starting with no stations created, you won't need to use this option.

Record the results of a test as an Excel file from the REALM monitor script

Goal: Record the results of a LANforge test as an Excel file.

Some scripts in the LANforge library have a monitor function built in. We are going to be using the test_jpv4_variable_time script for this demonstration. This is useful for running a test and then analyzing the results afterwards.

- 1. Start LANforge GUI. It is recommended to run this script on a fresh LANforge configuration with no stations loaded.
- Make sure you have lanforge-scripts on your device. If lanforge-scripts is already installed on your device, skip this step Navigate to py-scripts in the lanforge-scripts folder. If your LANforge device doesn't have this open source software yet you can clone them from Github To install lanforge-scripts paste git clone https://github.com/greearb/lanforge-scripts into your terminal.
- 3. Type the following command into your command line
- ./test_ipv4_variable_time.py --radio wiphy0 --security wpa2 --ssid lanforge --password
 password --output_format excel
- Replace the security, ssid, and password variables with the settings for the network you are testing. This will create 2 wiphy stations by default, connect them to the network you are testing, and report the results to an Excel file.
- 4. This creates a default file in your report-data folder under your home directory. The name will be in the format with today's timestamp and the name of the test you ran. It's a normal Excel file which you can use however you want..
- 5. There are multiple commands you can use with this function, here is a list of the flag and what each of them mean:
 - A. report_file: Name the full path of the file you want to save results to. Default will save to your reportdata folder.
 - B. duration_sec: how long you want to run the test

- C. output_format: The output format you want your file in. The following formats are supported:
 - A. xlsx DEFAULT
 - B. pickle
 - HINT: pickle is recommended if you are going to be manipulating data in python since it preserves formatting and can be quickly loaded into a Pandas DataFrame without any manipulation required
 - C. csv
 - D. json
 - E. pdf
 - WARNING: PDF is hard to export data from without an Adobe Acrobat license
 - F. png
 - WARNING: png is going to export an image, do not use this if you are planning on manipulating your data because it does not preserve the numbers recorded
 - G. html
 - H. hdf
 - I. parquet
 - J. stata
- D. ssid: REQUIRED Name of the network you are connecting to
- E. password: REQUIRED Password to the network
- F. radio: REQUIRED The radio which you are going to create stations from.
- G. security: Match the security protocol of your router.
- H. test_duration: Default is 60 seconds, write in a any number if you need. You can also use minutes or hours notation in this command, so for 42 minutes write 42m and for 8 hours write 8h.
- I. upstream_port: Most users won't need to use this option, but it tells the program where to connect to the router
- J. created_cx: List of the cross connects you are going to be analyzing. If you are starting with no stations created, you won't need to use this option.

Define and Demonstrate Docstring Usage in Candelatech Python Scripts

Goal: Use PEP 257 standards to properly document python scripts

This cookbook will demonstrate the proper method for documentation in Candelatech created test scripts using PEP 257 guidelines.

Any docstrings occurring after the attribute docstring will be referred to as "additional docstrings". Docstrings in Python are defined as a string literal that is the first statement in a module, function, class or method definition. Such string literals are referred to as "attribute docstrings" and will become the __doc__ attribute of the module, function, class, or method in which they are used.

PEP 257 establishes a standard for docstring usage. In order to keep consistency, triple double quotes should be used for **all** docstrings. Single-line docstrings should be contained entirely on one line. In the example given, a docstring for a function should briefly describe its purpose and specify the return type.

Example taken from PEP 257 page:

def function(a, b):
 """Do X and return a list."""

Multi-line docstrings should consist of a brief one line summary, followed by a blank line, and finally followed by a more elaborate description. The summary line may either be inline with the opening quotes or on the next line and the whole docstring should be on the same line of indentation as the opening and closing quotations. Closing quotes should exist on their own line, if part of a multi-line docstring, to prevent confusion.

```
Example taken from PEP 257 page:
```

```
def complex(real=0.0, imag=0.0):
    """Form a complex number.
    Keyword arguments:
    real -- the real part (default 0.0)
    imag -- the imaginary part (default 0.0)
    """
    if imag == 0.0 and real == 0.0:
        return complex_zero
    ...
```

Implementing Docstring Conventions With Candelatech Script Template

Candelatech scripts will follow PEP 257 specifications for module, function, class or method definitions. To keep things standardized the following example will cover the preferred format for module level docstrings.

#!/usr/bin/env python3
"""Module overview/one line description

More detailed summary of module, elaborate on when to use module/ test coverage of full script (Pass/ Fail conditions, columns/ information tested)

External scenario requirements:

Cookbook: http://www.candelatech.com/cookbook.php?vol=cli&book=

Copyright 2021 Candela Technologies Inc License: Free to distribute and modify. LANforge systems must be licensed.

Automated scanning of SSID, BSSID, and Signal of available wireless APs

Goal: Create a station and scan for SSID, BSSID, and Signal of available wireless APs

We will learn how to use a script to create a station and scan for available APs. We will then look at the /scanresults/ URI and the info we can get from a scan through JSON. Please refer to sta_scan_test.py as an example script.

	Using the Script
A. Com	nmand Line Options
	sta_name nameOfStation
	Specifies the name of the station to be created, if this option is used, the name will default to sta0000 .
	ssid nameOfNetwork
	Specifies the name of the network to connect to.
	This value must be used, however, the SSID does not have to exist and a fake name can be used.
	security {WEP, WPA, WPA2, WPA3, Open}
	Specifies the security type of the network to connect to. This value must be used, however, if a fake SSID is used the type should be open.
B. Runr	ning the script
A.	As an example, we can run the script using:
	./sta_scan_test.pysta_name sta0000ssid fake_ssidsecurity openradi
В.	This will produce output that looks like this:
	BSS Signal SSID

855	Signai	SSID
08:36:c9:e3:d4:da	-32.0	Logan-Test-Net
10:56:11:0c:04:02	-80.0	:)
22:56:11:0c:04:02	-79.0	xfinitywifi
32:56:11:0c:04:02	-80.0	NA

2.

The /scanresults/ URI

A. In order to view this page we will need to create a station and start a scan.

A. First we will create the station (Make sure to click on a radio in the Port Mgr tab first):

				Chambe	er View	Stop A	All R	estart Manager		Refresh HELP
1.1	0.0	RTP Endps File-I	0 0000		-	affic-Profil		Warnings +	Wifi-Messages	1
		tus Port N		Laver-3	L3 Endps	Layer		Armageddon	WanLinks	VolP/RTP
	รเล	itus Port N	ngr	Layer-3	L3 Endps	Layer	4-7	Armageddon	wantlinks	VOIP/RTP
D	sp	192.168.10.12:0	0	Sniff Pack	kets 🕑 Dov	vn 1	Clear Counter	s Reset Por	t Delete	
R	ot 1	Timer: nedium (8	3 s) 🔻	Apply		I	Display	Create	Modify	Batch Modify
_	_			_	-All Ethernet Interf	aces (Port	ts) for all Reso	urces.	roato a virtual in	terface of some type
ø	I	IP	Alias	Parent Dev	AP	Channel	SSID			are the hot-key acce
-		192.168.10.20	eth0					00:0d:b9:56:ad:e8	_	
		0.0.0.0	eth1					00:0d:b9:56:ad:e9		
		0.0.0.0	eth2					00:0d:b9:56:ad:ea		
		0.0.0.0	wiphy0			0		04:f0:21:94:d9:92		
		0.0.0.0	wiphyl			0		00:0e:8e:5c:63:82		
		0.0.0.0	wan1	wiphy1		1		00:0e:8e:5c:63:82		
		192.168.1.2	wlan0	wiphy0	08:36:C9:E3:D4:DA	44	Logan-Test	04:f0:21:94:d9:92		

1.

B. Next we will create the station, the default values can be used or a specific number for the station can be given:

Create	e VLANs on Port	: 1.1.04			~ ^
0	○ MAC-VLAN	○ 802. <u>1</u> Q-VLAN	O <u>R</u> edirect	O Bridge	Bond
U	GRE Tunnel	◉ <u>W</u> iFi STA	⊖ WiFi <u>V</u> AP	○ WiFi <u>M</u> onitor	🔾 WiFi Virtual Rad
0	Shelf: 1 🔻 F	Reso <u>u</u> rce: 1 (loga	n-lf2) 🔻	Port: 4 (wiphy0)	-
B	Quantity: 1				
	Basic Settings	WiFi Se <u>t</u> tings	Advanc <u>e</u> d S	Settings	
	VLAN ID:				
	STA ID:	0000			
	Parent MAC:	04:f0:21:94:d9:			
-	MAC Addr:	XXX:XXX:XXX:*:*:XXX		•	
4	DHCP-IPv4	Multiple Subn	ets		
	IP Address:				
	IP Mask or Bits:				
	<u>G</u> ateway IP:				
	#1 Redir <u>N</u> ame:				
	#2 Redir Name:				
6	Down Ap	y <u>C</u> ancel]	Ready	
0			changes, bu		e current window t

C. After creating the station, we will give the an SSID to connect to. (This doesn't have to be a real AP):

	Current: LI	INK-DOWN GRO NON	NE				
	Driver Info: Pr	Port Type: WIFI-STA	Parent:	wiphy0 wiphy	/0		
		Port Conf	igurab	oles			
Standard Configurat	ion Advanced	Configuration	Mis	c Configuration		Corruptions	Custom Wi
Enable		Ger	neral Ir	nterface Settin	igs		
Set MAC	Down	Aux-Mgt		DHCP Hostna	ame:	None	-
Set TX Q Len	DHCP-IPv6	DHCP Rele	ease	DHCP Vendor	r ID:	None	-
Set MID	DHCP-IPv4	Secondary		DHCP Client I		None	-
Set PROMISC	DNS Servers:	BLANK	-11-5	Peer IP:	ь.	NA	
	IP Address:	0.0.0.0		Global IPv6:		AUTO	
Services	IP Mask:	0.0.0.0		Link IPv6:		AUTO	
НТТР	Gateway IP:	0.0.0.0		IPv6 GW:		AUTO	
FTP	Alias:			MTU:		1500	
DNS	MAC Addr:	04:f0:21:31:5	a:92	TX Q Len		1000	
RADIUS	Rpt Timer:	nedium (8 s	s) 🗸	WiFi Bridge:		NONE	-
IPSEC-Client	IPSec GW:	0.0.0.0		IPSec Passwo	ord:	-	
	IPSec Local ID.:			IPSec Remote			
Low Level			WIF	Fi Settings			
PROMISC	SSID: fa	ake-ssid			DEFA	AULT	
TS0 Enabled	Key/Phrase:	k		(2) Mada	(000.0	11-k AV)	
UF0 Enabled	Freg/Channel: (identifier. NK] for empty S	SID	which mean	s use any a
GS0 Enabled		-			Contro.		a doo any 1
LRO Enabled		PA2 WPA3					
GRO Enabled	Disable <u>H</u> T4	40 🔲 Ena <u>b</u> le V	HT160) 📃 Disable SC	3I		

D. Clicking on Display Scan at the bottom of the station settings window will bring us to the Scan window:

		Port Status Inform	ation		
	Current: L	INK-DOWN GRO NONE			
	Driver Info: P	ort Type: WIFI-STA Paren	t: wiphy0 wiphy0		
		Port Configura	bles		
andard Configura	ition Advanced	Configuration Mis	sc Configuration	Corruptions Cus	stom V
Enable		General	Interface Settings		
Set MAC	Down	Aux-Mgt	DHCP Hostname	None	-
Set TX Q Len	DHCP-IPv6	DHCP Release	DHCP Vendor ID:	None	-
Set Offload	DHCP-IPv4	Secondary-IPs	DHCP Client ID:	None	-
Set PROMISC	DNS Servers:	BLANK	Peer IP:	NA	_
	IP Address:	0.0.0.0	Global IPv6:	AUTO	
Services	IP Mask:	0.0.0.0	Link IPv6:	AUTO	
HTTP	Gateway IP:	0.0.0.0	IPv6 GW:	AUTO	
FTP	Alias:		MTU:	1500	
DNS	MAC Addr:	04:f0:21:31:5a:92	TX Q Len	1000	
RADIUS	Rpt Timer:	nedium (8 s)	WiFi Bridge:	NONE	-
IPsec-Upstream	IPSec GW:	0.0.0.0	IPSec Password:		
-	IPSec Local ID.		IPSec Remote ID		
Low Level		W	Fi Settings		
PROMISC	SSID: fa	ake-ssid		AULT	_
TSO Enabled	Key/Phrase:			.11abgn-AX)	-
UFO Enabled	Freg/Channel:	0/0		afault	-
GSO Enabled		A2 WPA3 005			_
LRO Enabled		40 🔲 Enable VHT16			
GRO Enabled	Disable H14	to Enable vHITO	u 🔄 Disable sui		

E. Finally we'll be able to start the scan and see the results. Clicking on Scan and waiting a ds will show all of the APs available to the static

phy0 Scan Results								
SSID	Channel	Info	Auth	BSS	Signal			Age
	44+	3x3 MCS 0-9 AC	WPA2	10:56:11:0c:04:02	-79.0	5220	100	1.42 s
	1	3x3 MIM0	WPA2	10:56:11:0c:04:01	-74.0		100	12.60 s
RECT-85-HP ENV	4	1x1 MIM0	WPA2	f4:30:b9:05:0f:86	-75.0	2427	100	12.05 s
oDallasStars	11	3x3 MIM0	WPA2	f4:c1:14:69:66:8f	-82.0		100	10.03 s
ala madrid	11	0x0 MCS 0-0 AX	WPA2	ac:db:48:42:5f:45	-89.0	2462	100	10.00 s
andinh1987	11	3x3 MIM0	WPA2	84:00:2d:92:6a:f8	-62.0	2462	100	10.03 s
andinh1987	36+	3x3 MCS 0-9 AC	WPA2	84:00:2d:92:6b:00	-75.0	5180	100	9.76 s
gan-Test-Net	44+	3x3 MCS 0-9 AC	WPA2	08:36:c9:e3:d4:da	-32.0		200	1.39 s
achoWifi	11	2x2 MIM0	WPA WPA2	20:b0:01:5b:2f:ba	-80.0		100	10.01 s
almerston North	44+	4x4 MCS 0-9 AC	WPA2	bc:9b:68:de:a5:73	-81.0	5220	100	1.40 s
HAN	6	2x2 MIM0	WPA WPA2	5c:6a:80:1d:50:2b	-74.0	2437	100	11.31 s
cooterbooter	44+	3x3 MCS 0-9 AC	WPA WPA2	34:11:e4:e0:05:3b	-85.0	5220	100	1.38 s
cooterbooter	6	3x3 MIM0	WPA WPA2	34:1f:e4:e0:05:3a	-72.0	2437	100	11.31 s
ippery Weasel Tr	9	3x3 MIMO	WPA2	08:02:8e:da:23:aa	-84.0	2452	100	10.28 s
izardDen	8	3x3 MIM0	WPA2	94:a6:7e:14:8e:17	-11.0	2447	100	10.79 s
INITY	44+	3x3 MCS 0-9 AC	WPA2	42:56:11:0c:04:02	-79.0	5220	100	1.41 s
INITY	44+	4x4 MCS 0-9 AC	WPA2	bc:9b:68:de:a5:77	-81.0	5220	100	1.40 s
INITY	44+	3x3 MCS 0-9 AC	WPA2	66:1f:e4:e0:05:3b	-86.0	5220	100	1.41 s
INITY	36+	3x3 MCS 0-9 AC	WPA2	9a:00:2d:92:6b:00	-75.0	5180	100	9.78 s
BLANKI	44+	4x4 MCS 0-9 AC	WPA2	bc:9b:68:de:a5:76	-81.0	5220	100	1.37 s
BLANK]	44+	4x4 MCS 0-9 AC	WPA WPA2	bc:9b:68:de:a5:78	-80.0	5220	100	1.39 s
BLANK	44+	3x3 MCS 0-9 AC	WPA WPA2	32:56:11:0c:04:02	-74.0	5220	100	4.19 s
BLANKI	44+	3x3 MCS 0-9 AC	WPA WPA2	52:56:11:0c:04:02	-79.0	5220	100	1.38 s
BLANK	44+	3x3 MCS 0-9 AC	WPA WPA2	76:1f:e4:e0:05:3b	-86.0	5220	100	1.40 s
BLANKI	11	3x3 MIM0	WPA WPA2	92:00:2d:92:6a:f8	-62.0	2462	100	10.00 s
BLANKI	36+	3x3 MCS 0-9 AC	WPA WPA2	8a:00:2d:92:6b:00	-76.0	5180	100	9.74 s
initywifi	44+	3x3 MCS 0-9 AC	Open	22:56:11:0c:04:02	-78.0	5220	100	1.42 s
initywifi	44+	3x3 MCS 0-9 AC	Open	46:11:e4:e0:05:3b	-86.0	5220	100	1.42 s
initywifi	44+	4x4 MCS 0-9 AC	Open	bc:9b:68:de:a5:75	-80.0	5220	100	1.38 s
initywifi	36+	3x3 MCS 0-9 AC	Open	95:00:2d:92:6b:00	-75.0		100	9.78 s

3.

4.

JSON Response from /scanresults/

- A. Another way of viewing the same information is to use the /scanresults/ URI. This URL can be found at your LANforge ip using port 8080. Ex: 192.168.10.20:8080/scanresults. We will also need the shelf number, the resource number, and the station name. The final URL would look like this 192.168.10.20:8080/scanresults/1/1/sta0000
- B. The scan results can be viewed through JSON by using cURL on the same URL as before. The response will look like this:

{"handler":"candela.lanforge.HttpStationScan\$FixedJsonResponder","uri":" scanresults/:shelf_id/:resource_id/:port_id","candela.lanforge.HttpStationScan":
("duration":"1"),"scan-results".[{"1.1.4.08:36:c9:e3:d4:da":{"age":"2238","auth":"WPA. "beacon":"200","bss":"08:36:c9:e3:d4:da","channel":"44","entity id":"1.1.4", "frequency":"5220","info":"3x3 MCS 0-9 AC","signal":"-32.0","ssid":"Logan-Test-Net"}}

Accessing and Printing JSON Response with Python

- A. We will use sta_scan_test.py as an example for a start() method
 - A. First, we'll need to send a JSON post using realm. Use this cookbook as reference for getting started with realm. Our JSON will look something like this:
 - data = "shelf": 1, "resource": 1, "port": self.sta_list
 - B. We can then use json_post to send the request. We'll need to wait about 15 seconds to give the scan time to happen
 - self.json_post("/cli-json/scan_wifi", data)
 time.sleep(15)
 - C.
 - Next, we'll create a variable with the results from the scan using scan_results = self.json_get("scanresults/1/1/%s" % ','.join(self.sta_list))
 - D. Finally, we'll create a loop to iterate through the JSON response and print some nicely formatted output
 - print("{0:<23}".format("BSS"), "{0:<7}".format("Signal"), "{0:<5}".format("SS for result in scan_results['scan-results']:
 for name, info in result.items():
 print("%s\t%s\t%s" % (info['bss'], info['signal'], info['ssid']))

B. Final Results

C

- A. Our final function will look like this:
 - def start(self):
 self.station_profile.admin_up() print(self.sta list) print("Sleeping 15s while waiting for scan") data = {
 "shelf": 1,
 "resource": 1,
 "port": self.sta_list self.json_post("/cli-json/scan_wifi", data) time.sleep(15) scan_results = self.json_get("scanresults/1/1/%s" % ','.join(self.sta_lis print("{0:<23}".format("BSS"), "{0:<7}".format("Signal"), "{0:<5}".format for result in scan_results('scan-results']: for name, info_in result.items(): print("%s\t%s\t%s" % (info['bss'], info['signal'], info['ssid']))
- B. Our formatted output should look like this:

BSS	Signal	SSID
00:0e:8e:52:4e:82	-33.0	test-net
08:36:c9:e3:d4:db	-31.0	Logan-Test-Net
08:36:c9:e3:d4:dc	-27.0	Logan-Test-Net

Automated Probing of Ports for information

Goal: Probe a port for information on that port.

We will learn how to use a script to probe a port for more information.We will also look at the ouput from the GUI, JSON response, and the script itself. Use the port_probe.py script as a reference.

	Using the Script
A. Co	ommand Line Options
	Aport_eid <i>portEID</i>
	Specifies the eid of the port to be probed, if this option is used, the name will default to
	1.1.eth0.
B. Ru	nning the script
	A. As an example, we can run the script using:
	./sta_probe_test.pyport_eid 1.1.wlan1

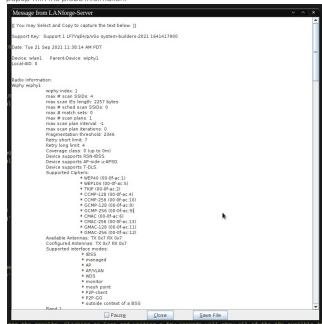
2.

Probe Results From the GUI

A. In order to view this page we will need to choose a port to use and start probing.

		Port Status Informa INK-DOWN GRO NONE ort Type: WIFI-STA Parent:		ohy1	1		
		Port Configurab					
Standard Configura	on Advanced Configuration Misc Configuration Corruptions Custom WiFi General Interface Settings						
Set MAC	Down	Aux-Mgt	DHCP Host	~	None	-	
Set MTU	DHCP-IPv6	DHCP Release	DHCP Vend	lor ID:	None	-	
Set Offload	DHCP-IPv4	Secondary-IPs	DHCP Clien	t ID:	None		
Set PROMISC	DNS Servers:	BLANK	Peer IP:		NA	NA	
	IP Address:	0.0.0.0	Global IPv6		AUTO		
Services	IP Mask: 0.0.0.0 Link IPv6: AUTO				AUTO		
HTTP	Gateway IP:	0.0.0.0	IPv6 GW:		AUTO		
FTP	Alias:		MTU:		1500		
DNS RADIUS	MAC Addr:	00:0e:8e:5c:63:82	TX Q Len		1000		
IPSEC-Client	Rpt Timer:	nedium (8 s) 🔻	WiFi Bridge		NONE	-	
IPsec-Upstream	IPSec GW:	0.0.0.0	IPSec Pass	word:			
	IPSec Local ID.		IPSec Remo	ote ID.:			
Low Level		WiF	i Settings				
PROMISC	SSID:		AP:	DEFA	NULT	_	
TS0 Enabled	Key/Phrase:		[?] Mode	(802.)	llabgn-AX)	-	
UFO Enabled	Freg/Channel:	2412/1	Rate:	OS Def	ault	-	
GS0 Enabled		A2 WPA3 OSEN	V WEP				
GRO Enabled		0 Enable VHT160		SGI			
- ono chabled			-				

B. Next we will click the probe button at the bottom of the window and another window will popup with the probe information:



This information is the formatted version of the probe. The other methods of accessing probe results will be unformatted JSON.

JSON Response from /probe/

A. Another way of viewing the same information is to access the /probe/ page from LANforge. This can be done by going to the page at your LANforge ip using port 8080. Ex:

192.168.10.20:8080/probe. We will also need the shelf number, the resource number, and the port name.

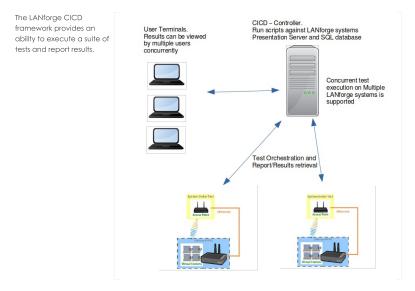
The final URL would look like this: 192.168.10.20:8080/probe/1/1/wlan1 and the page will look similar



3.

Basic CICD AP Testing with LANforge

Goal: Set up Basic CICD a LANforge system, Regression Automation and Reporting with data from previous runs.



- 1. The following steps are discussed
 - A. Set Up CICD Controller and Environment
 - B. Set Up The JSON Configuration Files
 - C. Test Execution
 - D. Test Results
- 2. Set Up CICD Controller and Environment
 - A. clone lanforge-scripts from https://github.com/greearb/lanforge-scripts
 - B. run /lanforge-scripts/update_dependencies.py to install python packages for generating output
 - C. Install web server:
 - The web server is to allow for viewing of results from User Terminals

The CICD - Controller is not dependent on a web server, results may be viewed locally on CICD - Controller

- A. LANforge LANforge installation using kinstall.pl installs a web server on LANforge LANforge installation installs an httpd server, LANforge may be used for storing and displaying results.
 - For the following example a separate LANforge system (Fedora) was used as the CICD Controller and httpd web server.
- Fedora install httpd and configure server
- C. Ubuntu install apache2 and configure server.
- D. Install mail service for email of links to results
 - For the example below Linux mailx program was used

Installation of mail services is dependent on the environment in which the CICD - Controller is installed.

- The CICD Controller is not dependent on email services
- E. Install database sqlite3

\$ sudo dnf install sqlite3 =
A. Fedora
\$ sudo apt-get update =
\$ sudo apt-get install sqlite3 =
B. Ubuniu

- F. Create a html-reports directory. On lanforge /home/lanforge/html-reports
- G. Determine sqlite3 database name and location, sqlite3 db will be created. ./tools/qa_sqlite3.db

3. Set Up The JSON Configuration Files

- A. There are three JSON configuration input files described below. For all the JSON configuration files the CAPITALIZED parameters allow for a value to be entered into one location and used in multiple areas of the CICD framework. For example in ssid_indx=1 the SSID_USED is set to asus11ax-5. For the test suite below the SSID_USED may be entered instead of asusu11ax-5, thus if the SSID changes, the SSID will need to be modified in ct_AX88U_dut, the ct_tests,ison will remain untouched. This reduces the need to modify the ct_test,ison for SSID changes that would affect multiple tests
 - A. --json_rig test_rig.json this JSON file describes LANforge test rig, Example ct_test_rig.json The test_rig.json describes the LANforge system and test parameters for the CICD -Controller

```
B. --json_dut ct_AX88U_dut.json this JSON file describes the AP, Example ct_AX88U_dut.json
                the ct_AX88U_dut.json describes the device under test parameters, DUT_SET_NAME:
                DUT_NAME ASUSSRT-AX88U for example is used by Chamberview Tests
        C. --json_test ct_tests.json this JSON file describes the tests, Example ct_tests.json
                The tests may use the CAPITALIZED variables or may be entered with the command line
                arguments as they would be entered on the command line.
                The tests are not limited to only python tests
B. test_rig.json
                  "test_rig":{
                              "Notes":[
                                       "This JSON file describes LANforge system and test run configuration"
              J
},
"test_rig_parameters":{
    "TEST_BED": "CT-TEST-001",
    "TEST_RIG": "CT-TEST-001",
    "DATABASE_SQLITE": "./tools/qa_sqlite3.db",
    "LF_MGR_ID": "192.168.100.116",
    "LF_MGR_PORT": "8080",
    "LF_MGR_PORT": "8080",
    "LF_MGR_PASS": "lanforge",
    "LF_MGR_PASS": "lanforge",
    "UPSTREAM_PORT": "1.1.eth2",
    "TEST_TIMEOUT": 600,
    "TEST_PEOULT": "Support@candelate
    "TEST_PEOULT".

                            1
                             UPSITEAM_POIT: 11.1eU12 ,
"TEST_TIMEOUT": 600,
"EMAIL_LIST_PRODUCTION": "support@candelatech.com",
"EMAIL_LIST_TEST": "support@candelatech.com",
"EMAIL_TITLE_TXT": "Lanforge QA Testing",
"EMAIL_TXT": "Lanforge QA Testing"
                  }
C.
        ct_AX88U_dut.json
                   "ct_AX88U_dut":{
                             "Notes":[
"The device undertest configuration is contained in this file"
                             1
                   },
                     .
test_dut":{
                             "DUT_SET_NAME": "DUT_NAME_ASUSRT-AX88U",
"USE_DUT_NAME": "ASUSRT-AX88U",
                              "wireless_network_dict":{
                                       "ssid_idx=0":{"ssid_idx":"0","SSID_USED":"asus11ax-2","SSID_PW_USED":"hello123","BSSID":"3c:7c:3f:55:4d:60","SECURITY
"ssid_idx=1":{"ssid_idx":"1","SSID_USED":"asus11ax-5","SSID_PW_USED":"hello123","BSSID":"3c:7c:3f:55:4d:64","SECURITY
                             }
                  }
D. ct_tests.json
                   "ct tests 001":{
                               'Notes":
                                        "This JSON file describes tests to be run by LANforge system"
                             1
                    "test_suites":{
"suite_wc":{
                                      "create_chamberview_dut_wc":{
"enabled":"TRUE",
"load_db":"skip",
"command":"create_chamberview_dut.py",
                                                   "args":""
                                                  "args_list":[
                                                           JS_LISU :[

"--lfmgr LF_MGR IP --port LF_MGR_PORT --dut_name DUT_NAME",

"--ssid 'ssid_idx=0 ssid=SSID_USED security=SECURITY_USED password=SSID_PW_USED bssid=BSSID'",

"--ssid 'ssid_idx=1 ssid=SSID_USED security=SECURITY_USED password=SSID_PW_USED bssid=BSSID'",

"--sw_version_DUT_SW --hw_version_DUT_HW --serial_num_DUT_SERIAL --model_num_DUT_NAME"
                                                 ]
                                       },
                                        "create_chamberview_wc":{
    "enabled":"TRUE",
    "load_db":"skip",
                                                  "command":"create_chamberview.py",
                                                   "args":""
                                                  "args_list":[
                                                                --lfmgr LF_MGR_IP --port LF_MGR_PORT --delete_scenario",
                                                           " --trag Lr --poit L _-non-twin __decet_________"
 --create _scenario _scenario_upa2_wc",
" --raw_line \"profile_link 1.1 STA-AC 19 'DUT: DUT_NAME Radio-1' NA wiphy7,AUTO -1 NA\" ",
" --raw_line \"profile_link 1.1 upstream-dhcp 1 NA NA UPSTREAM_PORT,AUTO -1 NA\""
                                                 1
                                       }.
                                       },
"wifi_capacity":{
    "enabled":"TRUE",
    "timeout":"600",
    "iterations":"1",
    "load_db":"skip",
    "command":"1f_wifi_capacity_test.py",
    "arror":""
                                                   "args":'
                                                   "args_list":[
                                                           " --mgr LF_MGR_IP --port LF_MGR_PORT --lf_user LF_MGR_USER --lf_password LF_MGR_PASS --instance_name scenario
" --upstream UPSTREAM_PORT --batch_size 1,10,19 --loop_iter 1 --protocol UDP-IPv4 --duration 6000",
" --pull_report --local_Lf_report_dir REPORT_PATH --test_tag 'wpa2_wc'",
" --test_rig TEST_RIG",
                                                             " --set DUT_SET_NAME'
                                                 ]
                                      },
"lf_qa":{
"crab
                                                 ua:[
"enabled":"TRUE",
"timeout":"600",
"load_db":"skip",
"command":"./tools/lf_qa.py",
                                                  "args":""
                                                  "args_list":[
                                                                 --path REPORT_PATH --store --png --database DATABASE_SQLITE"
                                                 1
                                   }
                           }
                 }
```



4. Set Up The JSON Configuration Files

- A. The If_check.py is run form the lanforge-scripts/py-scripts/tools directory
- B. If_check.py uses three JSON files as input:
 - For Example:
 - ct_test_rig.json describes the LANforge test rig configuration
 - ct_AX88U_dut.json describes the device under test
 - ct_tests.json -describe the tests to be run.
- 5. If_check.py execution, simple command example



6. Sample email sent on run

🖓 Get Messages 👻 🖋 Write 🔍 Chat 📱 Address Book 🛛 🗞 Tag 👻 🖓 Quick Filter		=
From LANforge <lanforge@ :candelatech.com=""> \$</lanforge@>	ঠ Reply 🗳 Reply All 🗸 → Forward 🖻 Archive 👌 Junk	🗊 Delete 🗌 More 👻
ubject Lanforge QA Testing [192.168.95.6] 2021-10-13 06:21:14.071302		7:21 AM
To support@candelatech.com 🛊		
anforge QA Testing lanforge target 192.168.100.116 esults from 192.168	1-10-13-06-18-12-lf_check.html	
A Report Dashboard: ttp://192.168. //html-reports/ct_results_directory/2021-10-13-06-18-12_lf_check/20	<u>121-10-13-06-21-11_lf_ga/2021-10-13-06-21-11-lf_ga.html</u>	

7. If_check.py: sample If_check.py Report

	6-18-12		TECHNOLOGIES	
bjective				
n QA Tests				

	Teraor	Termen	Thrandi	date		
ct523c-3b	7b s.15.0- rc5+	Version: 5.4.4 Complied on: Mon 11 Oct 2021 05:51:28 PM PDT	5.4.4	"Mon 11 Oct 2021 06:59:52 PM PDT"	167462c3107bd9b139b240edc0ab7437871bcb3191	e98881d23d0cce429de7b877b86661e408ec7360\n

LANForge Radios

Radio	WIFI-Radio Driver	Radio Capabilities	Pirmware Version	max_sta	max_vap	max_vits
1.1.wiphy0	ath10k(9984)	802.11bgn-AC	10.4b-ct-9984- xtH-13-774502ee5	128	24	64
1.1.wiphy1	ath10k(9984)	802.11an-AC	10.4b-ct-9984- xtH-13-774502ee5	128	24	64
1.1.wiphy2	ath9k()	802.11abgn	<ath9k firmware="" lack="" radios=""></ath9k>	2048	32	2048
1.1.wiphy3	ath10k(988x)	802.11abgn-AC	10.1-c1-8xx1H-022- bcdb24ff	127	24	64
1.1.wiphy4	Iwtwffl(AX200)	802.11abgn-AX	release/core62:5ecbd6da	1	1	1
1.1.wiphy5	hyterff(AX210)	802.11abgn-AX	release/core62:5ecbd6da	1	1	1
1.1.wiphy6	Iwtwitt(AX210)	802.11abgn-AX	release/core62:5ecbd6da	1	1	1
1.1.wiphy7	mt7915e()	802.11abgn-AX	<no data="" firmware=""></no>	19	16	19

LF Check Test Results

Test	Command	Duration	Start	End	Result	STDOUT	STDERR
create_chamberview_dut_wc	./create_chamberview_dut.py	0d 2s 825293 ms	2021-10-13-06-18-16	2021-10-13-06-18-19		STDOUT	
	102.160.000.116part 8000dat_mu d_idx+1 mid-amarilax-5 mecurity-spat um A20507-A0200						,NA
create_chamberview_wc	./create_chamberview.py	0cl 5s 987380 ms	2021-10-13-06-18-19	2021-10-13-06-18-25		STDOUT	
	.152.100.115port 9000delete_sco raw_line "profile_link 1.1 upstro			raw_line "profile_lin	R 1.1 STA-J	C 19 'DUT: 0	SUSET -
wft_capacity	./If_wifl_capacity_test.py	0d 165s 770807 ms	2021-10-13-06-18-25	2021-10-13-06-21-11	Success	STDOUT	
1,10,19 Loop_iter 1 protocol UDF	17.150.100.11dport 00001f_user 1 -DPv4duration 6000pull_report _tag 'spa2_sc'test_rig CT-TEST-000	-local_lf_report_dir //	nome/lanforge/html-re			1.1.eth2	alch_size
If_qq	././took/tt_qa.py	0d 2s 220803 ms	2021-10-13-06-21-11	2021-10-13-06-21-13		STDOUT	
/./tools/lf_qs.pypath /home/lan			If_checkstorep		/qa_sqlitel	l.db	

Generated by Candela technologies LANitarge network tealing tool	Can
www.condeldlech.com	TCHN
	TECH

dela

8. **If_qa.py**

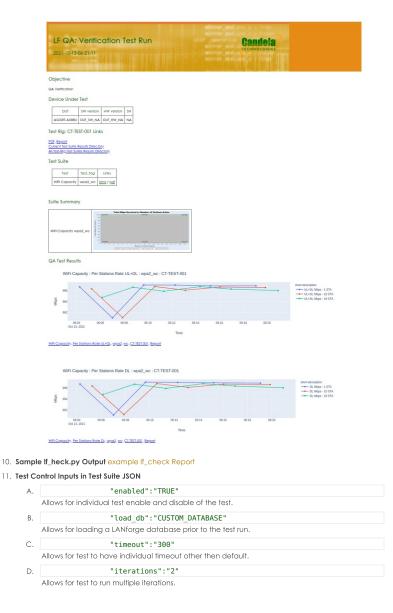
If_qa.py: process kpi.csv, produces html/pdf results, produces plotly png and interactive graphs from test run kpi

```
sample command:
```

./lf_qa.py --path /home/lanforge/html-reports/ct_results_directory/(results_dir of lf_check.py)\ --store \ --png \

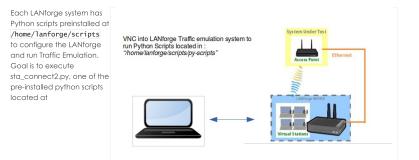
```
--database ./tools/qa_aqlite3.db
```

9. If_qa.py: sample If_qa.py Report



Start Here: Introduction to Executing Python Script on LANforge

Goal: Run First Python Script on LANforge



/home/lanforge/scripts/py-script/sta_connect2.py

The Script sta_connect2.py will create a station, create TCP and UDP traffic, run traffic for a short amount of time, and verify whether traffic was sent and received. It also verifies the station connected to the requested BSSID if bssid is specified as an argument. The script will clean up the station and connections at the end of the test. An html and pdf or the results will be generated and placed in /home/lanforge/html-reports directory The script will clean up the station.

 Start the LANforgeGUI if GUI not running: To start the LANforgeGUI navigate to : //home/lanforge/LANforgeGUI_5.4.5 Execute : ./lfclient.bash

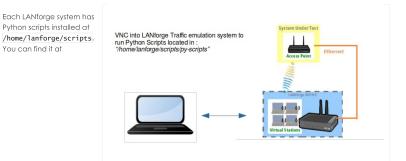
Select 'Connect' to connect to: localhost:4002 Local Machine Address

1	Server Name/IP:	localhost:4002 Local I	Andros	\$	
Candela	Clear		Disc <u>o</u> nnect	Discover	<u>C</u> ancel
TECHNOLOGIES	Discover Resources				
-		Click Discover to fir	id LANforge Mar	hagers	
/here Do I Find Scrip Preinstalled Python So xample script sta_co	cripts Location				
nitial Information to				:	
Note: An example of a The DUT informat		t is an Access Poi port generation.		ormation may	be optiona
		[] = = = [] = = = []			
 A. The LANforge manage B. The LANforge upstream 	-				
C. The LANforge radio:		pointerin point			
D. The Device Under Test		1			
E. The Device Under Test		-			
F. The Device Under Test					
			-1 - 11		
G. The Device Under Test			,		
H. The Device Under Test	r Haraware version:	aut nw version inv			
I. The Device Under Test		-dut_sw_version [sw v	version]		
I. The Device Under Test J. The Device Under Test		-dut_sw_version [sw v	version]		
J. The Device Under Test Example Command for	f Serial Number:du sta_connect2.py: • localhostups Ldut_ssid axe1 F-AXE11000dut_ M32A	-dut_sw_version [sw v ut_serial_num [serial r tream_port 1.1.eth 1000_5gdut_pass nw_version 1.0d	version] number] 2 wd 1f_axe11	on V3.0.0.4.	386.0000
J. The Device Under Test Example Command for s /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M	i Serial Number:du sta_connect2.py: • localhostups Ldut_ssid axe1 F-AXE11000dut_ 132A	-dut_sw_version [sw v ut_serial_num [serial r tream_port 1.1.eth L000_5gdut_pass	version] number] 2 wd 1f_axe11	on V3.0.0.4.	
J. The Device Under Test Example Command for s /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M File Edit View Search	Serial Number:du sta_connect2.py: - localhostups Ldut_ssid axe1 F-AXE11000dut_ 132A Terminal Help	-dut_sw_version [sw v ut_serial_num [serial r tream_port 1.1.eth 1000_5gdut_pass nw_version 1.0d	version] number] 2 wd 1f_axe11	on V3.0.0.4.	386.0000
J. The Device Under Test Example Command for s /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M	<pre>Serial Number:du sta_connect2.py: localhostups Ldut_ssid axe1 -AXE11000dut_ 432A Terminal Help py-scripts]\$ pwd /py-scripts]\$./st dut ssid axe11 </pre>	-dut_sw_version [sw v +_serial_num [serial r tream_port 1.1.eth 1000_5gdut_pass 100_version 1.0d Mate Terminal 1_connect2.pymgr 100_5gdut_passwd	version] number] 2 wd lf_axel1 ut_sw_versi localhost - lf axel1000	on V3.0.0.4. -upstream_po)_5gdut_se	386.0000
J. The Device Under Test Example Command for s /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M File Edit View Search [lanforge@ct523c-3ba3 /home/lanforge@ct523c-3ba3 th2radio 1.1.wiphy1 pa2dut_model_num GT	<pre>Serial Number:du sta_connect2.py: localhostups Ldut_ssid axe1 -AXE11000dut_ 432A Terminal Help py-scripts]\$ pwd /py-scripts]\$./st dut ssid axe11 </pre>	-dut_sw_version [sw v +_serial_num [serial r tream_port 1.1.eth 1000_5gdut_pass 100_version 1.0d Mate Terminal 1_connect2.pymgr 100_5gdut_passwd	version] number] 2 wd lf_axel1 ut_sw_versi localhost - lf axel1000	on V3.0.0.4. -upstream_po)_5gdut_se	386.0000
J. The Device Under Test Example Command for s /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M File Edit View Search [lanforge@ct523c-3ba3 /home/lanforge@ct523c-3ba3 th2radio 1.1.wiphy1 pa2dut_model_num GT dut_serial_num M32A	<pre>Serial Number:du sta_connect2.py: localhostups localhostups localhostups localhostdut_ localhostdut_ localhostdut_ localhost localhostdut_ localhost localhost</pre>	-dut_sw_version [sw v tream_port 1.1.eth 1000_5gdut_pass w_version 1.0d Mate Terminal Connect2.pymgr 100_5gdut_passwd version 1.0dut	version] number] 2 wd lf_axe11 ut_sw_versi lf_axe11006 _sw_version	on V3.0.0.4.	386.0000
J. The Device Under Test Example Command for s /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M File Edit View Search [lanforge@ct523c-3ba3 /home/lanforge@ct523c-3ba3 th2radio 1.1.wiphy1 pa2dut_model_num M ja2radio 1.1.wiphy1 pa2dut_model_num M dut_serial_num M32A Results for sta_conner dissultable.friedDevice -A 193600 bps %55D: uopts0000 -A 193600 bps	<pre>Serial Number:du sta_connect2.py: localhostups Ldut_ssid axel Terminal Help py-scripts]\$ pwd /py-scripts]\$./stdut 5sid axellAXE11000dut_h </pre>	-dut_sw_version [sw v tream_port 1.1.eth 1000_5gdut_pass w_version 1.0d Mate Terminal Connect2.pymgr 100_5gdut_passwd version 1.0dut	version] number] 2 wd lf_axel1 ut_sw_versi lf_axel1006 _sw_version	on V3.0.0.4. -upstream_po) 5gdut_se V3.0.0.4.386	386.0000
J. The Device Under Test Example Command for : /sta_connect2.pymgr radio 1.1.wiphy1 dut_model_num GT dut_serial_num M File Edit View Search (lanforge@ct523c-3ba3 /home/Lanforge%ct523c-3ba3 th2radio 1.1.wiphy1 pa2dut_model_num GT dut_serial_num M32A Results for sta_conne dut_serial_num M32A Results for sta_conne dut_serial_num M32A Results for sta_conne dut_serial_num M32A Stability duptideen -A 130500 bps ASSE: uptiteen -B 130500 bps ASSE: uptitee	<pre>Serial Number:du sta_connect2.py: localhostups localhostups localhostups localhostups localhostupt localhostupt localhostupt localhostupt localhost localhostupt localhost localhost</pre>	-dut_sw_version [sw v ream_port 1.1.eth 1000_5gdut_pass 100_5gdut_pass 100_5gdut_pass Mate Terminal 100_5gdut_pass 100_5gdut	version] number] 2 wd lf_axel1 ut_sw_version lf_axel1006 _sw_version e/html-repc P: 192.168.50.152	on V3.0.0.4.	336.0000 (→ (→ (→ (→ (→ (→ (→ (→ (→ (→ (→ (→ (→ (

- Script produces both html and pdf results Sample sta_connect2.py Script HTML Output: example of html output Sample sta_connect2.py Script pdf Output: example of pdf output Other script options may be shown by typing _./sta_connect2.py --help

Basic: Layer3 Traffic Generation: test_13.py

Goal: Use Python Script test_I3.py to Generate Layer3 Traffic



/home/lanforge/scripts/py-script/test_13.py

The script test_I3.py will:

- create stations (on multiple radios),
- create TCP and UDP cross connects
- run traffic at specified data rates for a specified time.

The traffic prioritization is configurable:

- BE Best Effort
- BK Background
- VI Video
- vo Video

The upload and download statistics are recorded at the end of each polling interval. The script will verify whether traffic is sent and received. The script cleans up the station and connections at the end of the test. An HTML and PDF report of the results will be generated and placed in the **/home/lanforge/html-reports** directory.

1. Start the LANforgeGUI if GUI not running:

To start the LANforgeGUI navigate to : /home/lanforge/LANforgeGUI_5.4.5 Execute : ./lfclient.bash Click the Connect button to connect to: localhost:4002 Local Machine Address

	LANforge	Connection Management	$\odot \odot \odot$
	Server Name/IP:	localhost:4002 Local Machine Address	
0 and ala	Clea <u>r</u>	Connect Disconnect Discover	<u>C</u> ancel
Candela	Discover Resources		
TECHNOLOGIES		Click Discover to find LANforge Managers	

2. Where Do I Find Scripts?

Preinstalled Python Scripts Location on LANforge: /home/lanforge/scripts/py-scripts Example script test_I3.py location: /home/lanforge/scripts/py-scripts/test_I3.py

 Initial Information to Gather as input to test_13.py script: Note: An example of a Device Under Test is an Access Point. The DUT information is used in report generation. The DUT information may be optional.

- A. The LANforge manager IP address: -- 1fmgr [localhost]
- B. The LANforge upstream port: --upstream_port [eth port]
- C. The LANforge end point type: --endp_type 'lf_udp,lf_tcp'
- D. The LANforge type of service: --tos 'BK,VI'
- E. The side 'a' tx bit rate (upload) --side_a_min_bps [bits per second]
- F. The side 'b' tx bit rate (download) --side_b_min_bps [bits per second]
- G. The LANforge radio information : --radio 'radio==[radio] stations==[number] ssid==[ssid] ssid_pw==[password] security==[security]'
- H. The Test Durations : --test_duration [value] (s seconds, m minutes, h hours)
- I. The Polling Interval : --polling_interval [value] (s seconds, m minutes, h hours)
- J. The Test Rig: --test_rig [test system id]
- K. The Test Tag: --test_tag [unique test id]
- L. The Device Under Test Model Number: --dut_model_num [model]

M. The Device Under Test Hardware Version: --dut_hw_version [hw version] N. The Device Under Test Software Version: --dut_sw_version [sw version] O. The Device Under Test Serial Number: --dut_serial_num [serial number] 4. Example Command for test_I3.py: --tos bk,ui (--side_min_bps 256000 \ --side_b_min_bps 102400000 \ --radio 'radio==wiphy1 stations==1 ssid==asus_5g ssid_pw==lf_asus_5g security==wpa2' \ --test_duration 30s \ --polling_interval 5s \ --polling_interval 5s \ --test_rig CT_LAB_104 \ --test_tag Layer_3_Example \ --dut_model_num RT-AX88U \ --dut_hw_version A1.1 --dut_sw_version 3.0.0.4.384 \ --dut_serial_num M1IAHP000003 Mate Terminal \bigtriangledown \land \times File Edit View Search Terminal Tabs Help X Mate Terminal Mate Terminal Mate Terminal X Mate Terminal
[lanforge@ct523c-3b8d py-scripts]\$ pwd
/home/Lanforge/scripts/py-scripts
[lanforge@ct523c-3b8d py-scripts]\$./test_l3.py \
 `#[test configuration]`\
 --Ungr 192.168.00.104 \
 --upstream_port 1.1.eth2 \
 --condp type 'lf udp.lf tcp' \
 --tos 'BK,VI' \
 --side a min_bps 256000 \
 --side b min_bps 102400000 \
 --radio 'radio==wiphyl stations==2 ssid==asus_5g ssid_pw==lf_asus_5g security==wpa2' \ --test duration 30s \ --polling_interval 5s \ '[#report_configuration]'\ --test_rig CT_LAB 104 \ --test_tag 'Layer_3 Example' \ --dut model_num RT-AX88U \ --dut_mversion Al.l \ --dut_sw_version 3.0.0.4.384 \ --dut_serial_num MIIAHP000003

5. Results for test_I3.py located in /home/lanforge/html-reports

	Mate Terminal 📀 📀 🤇
File Edit View Search Ter	minal Help
1659359110.236749 INF0	Stopping CXs l3_cxprofile.py 422
1659359110.654813 INFO	Cleaning up cxs and endpoints l3_cxprofile.py 433
1659359111.699352 INFO	Cleaning up stations station_profile.py 376
1659359111.845134 INFO	LFUtils: Waiting until 2 ports disappear LFUtils.py 572
1659359112.848988 INFO	LFUtils::wait_until_ports_disappear:: Request returned None
[http://192.168.0.104:80	080/port/1/1/sta0000,sta0001?fields=alias] LFUtils.py 610
1659359112.849085 INFO	Full test passed, all connections increased rx bytes test_13
.py 2183	
1659359112.855821 INFO	write_output_html: /home/lanforge/html-reports/2022-08-01-06
-03-52_test_l3/2022-08-01	l-06-03-52-test_l3.html lf_report.py 335
1659359112.855963 INFO	write_output_index_html: /home/lanforge/html-reports/2022-08
-01-06-03-52_test_l3/inde	
1659359113.516891 INFO	
1659359113.517003 INFO	PASSED: PASS: Stations & CX build finished: created/updated
2 stations and 8 connect	
1659359113.517036 INFO	PASSED: PASS: Stations & CX build finished: created/updated
2 stations and 8 connect	
1659359113.517074 INFO	
<-> -1 All tests passed	
	successfully. Exiting script with script success.
1659359113.517143 INFO	
ript success. lfcli_base.	
[lanforge@ct523c-3b8d py-	-scripts]\$

6. Results for test_I3.py is located at /home/lanforge/html-reports:

Script produces both HTML and PDF results:

• example of HTML output

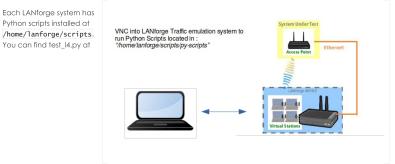
example of PDF output

• example of kpi.csv output

Other script options may be show by typing ./test_13.py --help

Basic: Layer4 HTTP Traffic Generation: test_14.py

Goal: Use Python Script test_I4.py to Generate Layer4 HTTP Traffic



/home/lanforge/scripts/py-script/test_l4.py

The script test_I4.py will:

- Create stations (on the specified radio).
- Create Layer 4-7 endpoints.
- Monitor the bytes-rd attribute of the created endpoints.

The test type attribute is configurable:

bytes-rd monitor the bytes read

urls monitor the url's per second

The monitored Layer 4-7 attribute statistics are recorded at the end of each polling interval. Test_I4.py will monitor the urls/s, bytes-rd, or bytes-wr attribute of the layer 4-7 endpoints. These attributes can be tested over FTP using a --ftp flag. If the monitored value does not continually increase, this test will not pass. The script cleans up the stations and connections at the end of the test. An HTML and PDF report of the results will be generated and placed in the //home/lanforge/html-reports directory.

 Start the LANforgeGUI if the GUI is not running: To start the LANforgeGUI navigate to : /home/lanforge/LANforgeGUI_5.4.5 Execute : ./lfclient.bash

Click the Connect button to connect to: localhost:4002 Local Machine Address

0	LANforge	Connection Management	$\odot \odot \otimes$
	Server Name/IP:	localhost:4002 Local Machine Address	-
	Clear	Connect Disconnect Discover	<u>C</u> ancel
Candela	Discover Resources		
TECHNOLOGIES		Click Discover to find LANforge Managers	

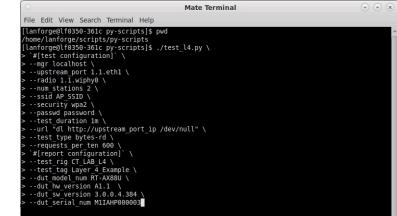
2. Where Do I Find Scripts?

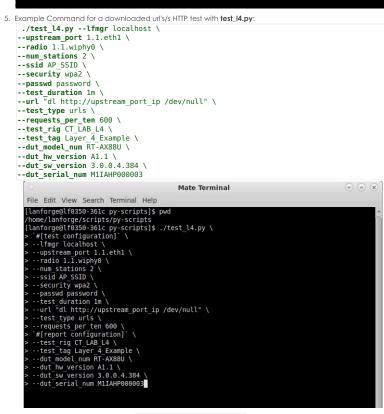
Preinstalled Python Scripts Location on LANforge: /home/lanforge/scripts/py-scripts Example script test_l4.py location: /home/lanforge/scripts/py-scripts/test_l4.py

- Initial Information to Gather as input for the test_14.py script: Note: An example of a Device Under Test is an Access Point. The DUT information is used in report generation. The DUT information may be optional.
 - A. The LANforge manager IP address: --mgr [localhost]
 - B. The LANforge upstream port: --upstream_port [eth port]
 - C. The LANforge radio information : --radio 'radio==[radio] stations==[number] ssid==[ssid] ssid_pw==[password] security==[security]'
 - D. The LANforge station creation amount: --num_stations [2]
 - E. The AP SSID name: --ssid [SSID]
 - F. The AP security type: --security [open, wpa, wpa2, wpa3]
 - G. The AP SSID password: --passwd [password]
 - H. The Test Duration : --test_duration [value] (s seconds, m minutes, h hours)
 - I. The Test URL: --url "dl http://upstream_port_ip /dev/null"
 - J. The Test Type: --test_type [bytes-rd, urls]
 - K. The Service Request Interval: --requests_per_ten [600]
 - L. The Test Rig: --test_rig [test system id]

- M. The Test Tag: --test_tag [unique test id]
- N. The Device Under Test Model Number: --dut_model_num [model]
- O. The Device Under Test Hardware Version: --dut_hw_version [hw version]
- P. The Device Under Test Software Version: --dut_sw_version [sw version]
- Q. The Device Under Test Serial Number: --dut_serial_num [serial number]
- Example Command for a downloaded bytes-rd HTTP test with test_I4.py:

<pre>./test_l4.pylfmgr localhost \</pre>
upstream_port 1.1.eth1 \
radio 1.1.wiphy0 \
num_stations 2 \
ssid AP_SSID \
security wpa2 \
passwd password \
test_duration 1m \
url "dl http://upstream_port_ip /dev/null" \
test_type bytes-rd \
requests_per_ten 600 \
test_rig CT_LAB_L4 \
test_tag Layer_4_Example \
dut_model_num RT-AX88U \
dut_hw_version A1.1 \
dut_sw_version 3.0.0.4.384 \
dut_serial_num M1IAHP000003





6. Results for test_I4.py are located in /home/lanforge/html-reports:

	Mate Terminal	\odot \otimes \otimes
File Edit View Search Term	inal Help	
1		
659551704.658922 INF0	item sta0000 l4 test l4.py 256	
659551704.659637 INFO	item sta0001 l4 test l4.py 256	
659551704.661618 INF0	self.csv results file -results.csv test 14.py 216	5
659551704.662043 INFO	csv results file: -results.csv test l4.py 810	
659551704.727016 INFO	write output html: /home/lanforge/html-reports/20	922-08-03-11-32-
	2-25-test_l4.html lf_report.py 335	
659551704.728430 INF0	write output index html: /home/lanforge/html-repo	orts/2022-08-03-
1-32-25_test_l4/index.htm	l lf_report.py 323	
L659551706.899609 INFO	Stopping CXs l4_cxprofile.py 71	
659551707.116932 INFO	Cleaning up cxs and endpoints 14 cxprofile py 13	1
659551707.358503 INF0	Cleaning up stations station_profile.py 376	
659551707.584649 INF0	LFUtils: Waiting until 2 ports disappear LFUti	
659551709.141561 INF0	LFUtils::wait until ports disappear:: Request ref	turned None: [n
659551709.142251 INFO	/1/sta0000,sta0001?fields=alias] LFUtils.py 610	1 570
659551709.156003 INFO	LFUtils: Waiting until 2 ports disappear LFUti LFUtils::wait until ports disappear:: Request ref	
	/1/sta0000,sta0001?fields=alias] LFUtils.py 610	turned None: [n
.659551709.157082 INFO	Full test passed test 14.py 879	
659551709.157956 INFO	PASSING TESTS lfcli k	bace ny 522
659551709.158563 INFO	PASSED: PASS: Station build finished lfcli base.	
659551709.158879 INFO	PASSED: All stations got IPs lfcli base.py 524	
	ccessfully. Exiting script with script success.	
659551709.159600 INFO	2 out of 2 tests passed successfully. Exiting sci	ript with script
success. lfcli base.py 55		
lanforge@lf0350-361c py-s		

7. Results for test_I4.py are located in /home/lanforge/html-reports:

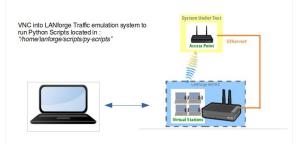
- The script produces both HTML and PDF results:
- example of HTML output
- example of PDF output
- example of **kpi.csv** output

Additional script options may be shown by typing ./test_14.py --help

Basic: Layer4 FTP Traffic Generation: test_14.py

Goal: Use Python Script test_14.py to Generate Layer4 FTP Traffic

Each LANforge system has Python scripts installed at /home/lanforge/scripts. You can find test_14.py at



/home/lanforge/scripts/py-script/test_l4.py

The script test_I4.py will:

- Create stations (on the specified radio).
- Create Layer 4-7 endpoints.
- Monitor the bytes-rd attribute of the created endpoints.

The test type attribute is configurable:

bytes-rd monitor the bytes read

bytes-wr monitor the bytes write

urls monitor the url's per second

The monitored Layer 4-7 attribute statistics are recorded at the end of each polling interval. Test_14.py will monitor the urfs/s, bytes-rd, or bytes-wr attribute of the layer 4-7 endpoints. These attributes can be tested over FTP using a --ftp flag. If the monitored value does not continually increase, this test will not pass. The script cleans up the stations and connections at the end of the test. An HTML and PDF report of the results will be generated and placed in the //home/lanforge/html-reports directory.

Start the LANforgeGUI if the GUI is not running:
 To start the LANforgeGUI navigate to : /home/lanforge/LANforgeGUI_5.4.5
 Execute : ./lfclient.bash
 Click the Connect button to connect to: localhost:4002 Local Machine Address

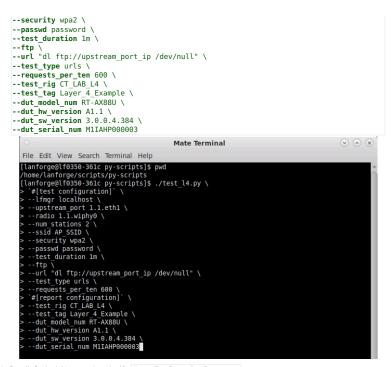
	LANforge	Connection Ma	nagement		$\odot \odot \otimes$
	Server Name/IP:	localhost:4002 Lo	ocal Machine Addres	s	•
Orndala	Clear	Conn <u>e</u> ct	Disc <u>o</u> nnect	Discover	Cancel
Candela	Discover Resources				
TECHNOLOGIES		Click Discover	to find LANforge Ma	nagers	

- 2. Where Do I Find Scripts?
 - Preinstalled Python Scripts Location on LANforge: /home/lanforge/scripts/py-scripts Example script test_14.py location: /home/lanforge/scripts/py-scripts/test_14.py
- Initial Information to Gather as input for the test_l4.py script: Note: An example of a Device Under Test is an Access Point. The DUT information is used in report generation. The DUT information may be optional.
 - A. The LANforge manager IP address: --mgr [localhost]
 - B. The LANforge upstream port: --upstream_port [eth port]
 - C. The LANforge radio information : --radio 'radio==[radio] stations==[number] ssid==[ssid] ssid_pw==[password] security== [security]'
 - D. The LANforge station creation amount: --num_stations [2]
 - E. The AP SSID name: --ssid [SSID]
 - F. The AP security type: --security [open, wpa, wpa2, wpa3]
 - G. The AP SSID password: --passwd [password]
 - H. The Test Duration : --test_duration [value] (s seconds, m minutes, h hours)
 - I. The FTP Test Switch: --ftp [enables FTP testing]
 - J. The Test URL: --url "dl ftp://upstream_port_ip /dev/null"
 - K. The Test Type: --test_type [bytes-rd]
 - L. The Service Request Interval: --requests_per_ten [600]
 - M. The Test Rig: --test_rig [test system id]
 - N. The Test Tag: --test_tag [unique test id]
 - O. The Device Under Test Model Number: --dut_model_num [model]
 - P. The Device Under Test Hardware Version: --dut_hw_version [hw version]
 - Q. The Device Under Test Software Version: --dut_sw_version [sw version]
 - R. The Device Under Test Serial Number: --dut_serial_num [serial number]

4. Example Command for a downloaded bytes-rd FTP test with test_14.py:

<pre>./test_l4.pylfmgr localhost \upstream_port 1.1.ethl \radio 1.1.wiphy0 \num_stations 2 \sscid AP_SSID \security wpa2 \passwd password \test_duration lm \test_duration</pre>
radio 1.1.wiphy0 \ num_stations 2 \ ssid AP_SSID \ security wpa2 \ password \ test_duration 1m \
num_stations 2 \ ssid AP_SID \ security wpa2 \ password \ test_duration 1m \
ssid AP_SSID \ security wpa2 \ passwd password \ test_duration lm \
security wpa2 \ passwd password \ test_duration 1m \
passwd password \ test_duration 1m \
test_duration 1m \
ftp \
url "dl ftp://upstream_port_ip /dev/null" \
test_type bytes-rd \
requests per ten 600 \
test_rig CT_LAB_L4 \
test_tag Layer 4 Example \
dut_model_num RT-AX88U \
dut_hw_version A1.1 \
dut_sw_version 3.0.0.4.384 \
dut_serial_num M1IAHP000003
○ Mate Terminal (♥ ♠ 🛠
File Edit View Search Terminal Help
<pre>/home/lanforge/scripts/py-scripts [lanforge@lf0303-s6lc py-scripts]\$./test_l4.py \ > `#[test configuration]` \ >Ifmgr localhost \ >ufmgr localhost \ >radio 1.1.wiphy0 \ >security wp22 \ >security wp22 \ >security wp24 \ >test_duration 1m \ >test_type bytes-rd \ >test_type bytes-rd \ >reducets per to 600 \ </pre>

5. Example Command for a downloaded url's/s FTP test with test_l4.py: ./test_l4.py --lfmgr localhost \ -upstream_port 1.1.ethl \ --radio 1.1.wiphy0 \ -num_stations 2 \ --ssid AP_SSID \



6. Results for test_14.py are located in /home/lanforge/html-reports:

	Mate Terminal	\odot
File Edit View Search Ter	minal Help	
7		
1659719453.671809 INFO	item sta0000_l4 test_l4.py 256	
1659719453.672634 INFO	item sta0001 l4 test l4.py 256	
1659719453.674645 INFO	self.csv_results_file -results.csv test_l4.py 216	
1659719453.675257 INFO	csv results file: -results.csv test l4.py 810	
L659719453.741720 INFO	write_output_html: /home/lanforge/html-reports/2022-08	-05-10-06-
l5_test_l4/2022-08-05-10	06-15-test_l4.html lf_report.py 335	
659719453.743767 INFO	write output index html: /home/lanforge/html-reports/20	922-08-05
0-06-15_test_l4/index.ht	ml lf_report.py 323	
659719455.664310 INF0	Stopping CXs l4_cxprofile.py 71	
659719455.958650 INFO	Cleaning up cxs and endpoints l4_cxprofile.py 131	
659719456.176748 INF0	Cleaning up stations station_profile.py 376	
659719456.398688 INFO	LFUtils: Waiting until 2 ports disappear LFUtils.py	
659719457.430698 INF0	LFUtils::wait_until_ports_disappear:: Request returned	None: [h
	1/1/sta0000,sta0001?fields=alias] LFUtils.py 610	
659719457.431388 INFO	LFUtils: Waiting until 2 ports disappear LFUtils.py	
659719457.443248 INF0	LFUtils::wait_until_ports_disappear:: Request returned	None: [h
	1/1/sta0000,sta0001?fields=alias] LFUtils.py 610	
659719457.444070 INFO	Full test passed test_l4.py 879	
659719457.444743 INFO	PASSING TESTS lfcli_base.py	
659719457.445157 INF0	PASSED: PASS: Station build finished lfcli_base.py 524	
659719457.445669 INFO	PASSED: All stations got IPs lfcli_base.py 524	
	uccessfully. Exiting script with script success.	
L659719457.446345 INFO	2 out of 2 tests passed successfully. Exiting script w	ith scrip
success. lfcli_base.py		
[lanforge@lf0350-361c py-	scripts]\$	

7. Results for test_I4.py are located in /home/lanforge/html-reports:

- The script produces both HTML and PDF results:
 - example of HTML output
 - example of PDF output
 - example of **kpi.csv** output

Additional script options may be shown by typing ./test_14.py --help

Customize and Run the Python Dataplane Script

Goal: Edit and run the bash script that runs both Python Dataplane and Create Chamberview scripts according to a customized LANforge Setup.

This cookbook describes how to edit a bash script, cv_dataplane_script.sh, that executes the 'Create Chamber DUT, 'Create Chamberview' and the 'Dataplane' test python scripts (create_chamberview_dut,py, create_chamberview,py, If_dataplane_test,py). These 3 python scripts are broken up into sections within this one bash script, that have their own arguments passed into each python script. The python scripts will run in consecutive order within the bash script and the LANforge GUI will reflect when each python script runs. Requires LANforge 5.4.2.

- Open the bash script and edit the variables at the top of the script. Also, understand a little bit about this bash script.
 - A. This bash script, 'cv_dataplane_script.sh, is comprised of 3 different python scripts, broken up into their own sections. This cookbook explains how to run these 3 different python scripts.

B. In this code, when executing a python script, the arguements are denoted by a '--' in front of them. This is then followed by the arguement name, a equals sign, and then the actual arguement (user input). The format: '--argument_name_1=[USER INPUT ARGUMENT 1] --argument_name_2=[USER INPUT ARGUEMENT 2]. Only 1 space must be between the end of an argument input and next arguement name. Never include a space between the equal sign and arguement/arguement name.



C. Variables are denoted at the top in the format 'VARIABLE_NAME=YOUR_VALUE' When the variables are used in the script, the format is '\${VARIABLE_NAME}'. More or less variables can be added and removed from the script if used in this format.

#!/bin/bash			
<pre># Define some common variables. MGR=192.168.102.211 PORT=8080 DUT_NAME="TEST_DUT"</pre>			

- D. Throughout the script, there are backslashes at the end of the line if the arguments are continuing over to the next line. The backslashes indicate the code to combine both lines together when running.
- E. Lastly, to see any further detail about scripts, run the script in the same format as the bash code: go to the py-scripts directory and run the --help argument. In the create_chamberview_dut python script this would look like: '/create_chamber_dut.py --help'. This --help argument gives more detail

-dut_name \${DU

about the script.	
[lanforge@ct523c-Scale-Mobsta ~]\$ cd /home/lanforge/lanforge-scripts/py-scripts [lanforge@ct523c-Scale-Mobsta py-scripts]\$./create_chamberview_dut.pyhelp	ļ

2. Create the DUT: edit the arguments to pass into create_chamberview_dut.py

--ssid "ssid_idx=1 ssid=eero-tesh-lanforge security=hPA2 password=lanforge bssid=64:97:14:64:09:07" A. First, the argument names --If_mgr, -o (port) and --dut_name are all passed in from the top (and they are all required). DUT Flags are flags used by the server, below is a screenshot of all the flags. To calculate the number for all the flags needed, t

dut flaas:

STA MODE	0x1	# (1) DUT acts as Station.
AP_MODE	0x2	# (2) DUT acts as AP.
INACTIVE	0x4	# (3) Ignore this in ChamberView, etc
WEP	0x8	# Use WEP encryption on all ssids, deprecated, see add_dut_ssid.
WPA	0x10	# Use WPA encryption on all ssids, deprecated, see add_dut_ssid.
WPA2	0x20	# Use WPA2 encryption on all ssids, deprecated, see add_dut_ssid.
DHCPD - LAN	0x40	# Provides DHCP server on LAN port
DHCPD-WAN	0x80	# Provides DHCP server on WAN port
WPA3	0×100	# Use WPA3 encryption on all ssids, deprecated, see add_dut_extras.
11r	0x200	# Use .11r connection logic on all ssids, deprecated, see add_dut_ssid.
EAP-TTLS	0x400	<pre># Use EAP-TTLS connection logic on all ssids, deprecated, see add_dut_ssid.</pre>
EAP - PEAP	0x800	# Use EAP-PEAP connection logic on all ssids, deprecated, see add_dut_ssid.
NOT - DHCPCD	0x1000	# Station/edge device that is NOT using DHCP.
		# Otherwise, automation logic assumes it is using dhcp client.

B. Next, add the sid lines. Each –ssid argument is followed by a string with several individual arguments. sid_idx is the number which ssid it is. This number is just a sequential number, the first one is 1, second one is 2, etc. 'ssid' is sid from the AP. This is the same for password, security, BSSID. Multiple securities example for a SSID is shown in the second –line for ssid_idx 2.

Nome Tot DuT Name Observation Observation					Create/Mod	ify DUT							 × ×
Start Munch Serial point Anti-restion Anti-restion NUN MN MN MN Start Munch MN MN MN MN Start Munch MN MN MN MN MN Start Munch Massord Munch Markenson MSDD Munch MN MN MN MA LAPER Start Munch Massord Munch Markenson MSDD Munch MASDD Munch MAN MAN MAN MAN MAN LAPER MAN	Name	TEST_DUT	Image file	NONE	Choose Image	×							
WM MA MA Statu Statu <td>SW Info</td> <td></td> <td>HW Info</td> <td></td> <td>Model Number</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	SW Info		HW Info		Model Number								
SDD-1 Resource1 Parsson/2 Revirage SDD-1 K4972464026 WP WP WPA WPA DD-1/r C.APFAL SDD-2 recome/hardinge Parsson/2 referinge BSDD-2 REF71640200 WP WP WPA	Serial Number		Serial port		API version	0							
SSD-2 Research Ladrug Passend 2 Refurge SSD-2 SSD-2<	WAN		LAN										
SSID-3 Personed 3 BSID-3 D0000000000 WPP WPA WPA 2 WPA 3 B02117 EAP-FLA SSID-4 Personed 4 BSID-4 D0000000000 WPP WPA 2 WPA 3 B02117 EAP-FLA SSID-4 Personed 4 BSID-4 D0000000000 WPP WPA 2 WPA 3 B02117 EAP-FLA SSID-4 Personed 4 BSID-7 D0000000000 WPP WPA 2 WPA 3 B02117 EAP-FLA SSID-4 Personed 7 BSID-7 D0000000000 WPP WPA 2 WPA 3 B02117 EAP-FLA SSID-4 Personed 7 BSID-7 D0000000000 WPP WPA 2 WPA 3 B02117 EAP-FLA SSID-4 Personed 7 BSID-7 D0000000000 WPP 3 WPA 2 WPA 3 B02117 EAP-FLA SSID-4 D0000000000 WPP 3 WPA 3 B02117 EAP-FLA BAP-FLA Mark Hado 10 Num Ane Hado 2 B0000 WPP 3 WPA 3 <	SSID-1	eero-mesh-lanforge	Password-1	lanforge	BSSID-1	64:97:14:64:d9:06	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
SSID-4 Password 4 SSID-4 Password 5 SSID-4 Password 5 SSID-5 Password 5 SSID-5 Password 6 SSID-6 Password 7 SSID-7 Password 7	SSID-2	eero-mesh-lanforge	Password-2	lanforge	BSSID-2	64:97:14:64:d9:07	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
SSID-5 Password 5 BSSID-5 D000000000000 WPP WPA WPA2 WPA3 B021tr EAP-FEA SSID-6 Password 6 BSSID-6 D00000000000 WPP WPA2 WPA3 B021tr EAP-FEA SSID-6 Password 6 BSSID-7 D0000000000 WPP WPA3 B021tr EAP-FEA SSID-7 Password 7 BSSID-7 D0000000000 WPP WPA3 B021tr EAP-FEA SSID-7 Password 7 BSSID-7 D000000000 WPP WPA3 B021tr EAP-FEA CAHO MgIP B00.0 D000000000 WPP WPA3 B021tr EAP-FEA Nam Act Bodo 10 Nam Act Bodo 20 Nam Act Bodo 10 D EAP-FEA EAP-FEA Chacker Phondes DHCP on LM DHCP/Gites Phondes DHCP on LM DHCP/Gites Phondes DHCP on LM EAP-FEA	SSID-3		Password-3		BSSID-3	00:00:00:00:00:00	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
SSD-4 Personold BSSD-4 BSSD-4 BOSD-4 BOSD-7 BOSD-	SSID-4		Password-4		BSSID-4	00:00:00:00:00:00	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
SSDP 1 Personed 7 SSDP 3 Personed 7 SSDP 4 Personed 7	SSID-5		Password-5		BSSID-5	00:00:00:00:00:00	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
SSID-4 Password 8 SSID-8 00000000000 WFP WFA WFA2 WFA3 B62.11r EAA-FLIX	SSID-6		Password-6		BSSID-6	00:00:00:00:00:00	WEP	WPA	WPA2	WPA3	802.11r	EAP-TTLS	EAP-PEA
CANDO Mg/LP 00.0 Num Are Bado 10 Num Are Bado 2 Num Are Bado 3 0 Schware EXtranse EXtranse EXtranse EXtranse EXtranse	SSID-7		Password-7		BSSID-7	00:00:00:00:00:00	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
Num Ark Readio 2 Num Ark Readio 2 Num Ark Readio 3 2 Provides DHCP on UNN III DHCP Client. III Provides DHCP on WINN IIII AP DUT	SSID-8		Password-8		BSSID-8	00:00:00:00:00:00	WEP	WPA	WPA2	WPA3	🔲 802.11r	EAP-TTLS	EAP-PEA
Active Provides DHCP on LAN DHCP Client Provides DHCP on WAN AP DUT	EAP-ID		Mgt IP	0.0.0.0									
	Num Ant Radio 1	0	Num Ant Radio 2	0	Num Ant Radio 3	0							
Notes	Active Active	Provides DHCP on LAN	DHCP Client	Provides DHCP on WAN	AP DUT								
	Notes												
					Apply QK	Cancel							

3. Create the Chamber View scenario: edit the flags to pass into create_chamberview.py

The image below highlights the section of the script to be edited.

Create/update chamber view scenario and apply and build it.
echo "Build Chamber View Scenario"
#change the lfmgr to your system, set the radio to a working radio on your LANforge system, same with the ethernet port.
./create_chamberview.pymgr \${MGR}mgr_port \${PORT}delete_scenariocreate_scenario "TEST_SCENARIO" \
line "Resource=1.1 Profile=STA-AX Amount=1 Uses-1=wiphy0 DUT="\${DUT_NAME}" DUT_Radio=Radio-1 Traffic=http Freq=-1" \
line "Resource=1.1 Profile=upstream-dhcp Amount=1 Uses-1=eth2 Traffic=NA Freq=-1 " \
line "Resource=1.1 Profile=uplink-nat Amount=1 Uses-1=eth3 Uses-2=eth2 Traffic=voip DUT=upstream DUT_Radio=LAN Freq=-1"

A. As shown in the example, the first line comprises of the following flags: --mgr (lanforge IP address), -mgr_port (the port through which this script will use), --delete_scenario, and --cs (name under which this new scenario will be saved under in the database). The mgr and mgr_port are passed in through the variables at the top of this bash script and the scenario name can be anything. If the scenario name passed in as --create_scenario is already in the GUI, using the '--delete_scenario' flag will override that already created scenario. ,/create_chamber/stevy --gr \$1000 --gr port \$(2001) --delete_scenario -rcreate_scenario *TEST_SCEURE(0* \) B. The next arguments, '--line' are the lines that show up in the GUI if the scenario is created manually. These lines will translate in the GUI after the command is executed. After the '--line' is the actual line, a string. The string has details of the objects of the dataplane test (such as amount, radio, etc.), some required, some not. In the example is used a 'station' line, 'upstream-dhcp' line, and 'uplinknat' line (as all these are objects in our dataplane test).1st, 'Resource' (required) is resource number the object is located on. The first number will most likely always be 1 (Shelf) and the second number is Resource (in this case, also 1). Notation for resource 4 would be '1.4'. Profile (required) is the name of the profile wanted for an object. Profiles can be created and found in the profiles tab in the GUI. The profile used in the example is 'STA-AX' (with the station profile type), for the station profile. 'Amount' (required) is the amount of objects to be created in chamber view. For stations, the amount can be multiple, for ethernet object creations it will most likely be 1.'Uses-1' (required) is the object this new created object will use or reside on. For an upstream object, eth1 through eth3 might be the 'Uses-1'. For station objects, this is the radio that the station will use. 'Uses-2' is optional and an alternative to 'Uses-1'. Traffic' (optional) is background traffic that object runs and can be either voip, http and others found in the 'Traffic' dropdown of the Scenario Creation GUI.

			rofile		Amount		Uses-1				Frequ	ency.			Maps To		Traffic-1		Traffic-2		Traffic-3		Traffic-6		Traffic-5		
È	×	•	STA: STA-AX	٠	1 (1)	-	wiphyd		ило		AUT0	(-1 Mhz)		NA.	DUT: TEST_DUT Radio-1	۳	http	٠	NA.	Ŧ	NA	٠	NA.	-	NA	-	
			Upstream: upstream-dhcp	٠	1 (1)	*	ett2	-	kuito	٣	auto.	(-1 Mha)		NA	DUT: TEST_DUT Radio-1	*	NA	٣	NA.	¥	NA	٠	NA.	-	NA		
8	×	Ŧ	Uplink uplink nat		1 (1)	-	e#3		192	-	AUTO.	(+1 Mmg)	-	NA.	DUT: upstream LAN	-	voip	¥	NA.	•	NA	¥	NA.	-	NA	-	

C. 'DUT' and 'DUT_Radio' are not optional. In the station line, '\$DUT' is taking in the DUT name variable passed in at the top, but it can be any DUT name that is already created in the GUI. 'DUT_Radio corresponds with the SSID number in the DUT object, 'Radio-1' is corresponding to 'SSID-1'. For the upstream object, the 'Radio' is the 'LAN' port on the DUT. Finally, 'Freq' is the frequency the object's radio should be on. This mainly is for stations and objects that use radios." BSSID-1 64:97:14:64:d9:06 WEP WPA WPA2 WPA3 802.11r EAP-TTLS EAP-PEAP SSID-1 Password-1

64:97:14:64:09:07 WEP WPA WPA2 WPA3 802.11r EAP-TTLS EAP-PEA

4. Edit the arguments to run the dataplane test.

85%

- A. Similar to other python scripts run in this bash script, '--mgr', '--o' (port) and '--dut' are passed in through the variables at the top. '--instance_name is the name of the new window that will hold the dataplane test.--upstream_port is more than likely an ethernet port, one used earier when creating the scenario. The example used below is eth3. This notation is the '[shelf].[resource].[name] notation. The shelf and the resource can be found in the 'Port Manager', under the 'Port' column. The shelf and the resource are the first 2 numbers separated by the first dot. This same notation is used for the '--station' flag too. The 'station' flag is the station used in the dataplane test.
- B. Upload and download speed are in percentages or Kbps/Mbps/Gbps. It is the requested connection traffic speed. If a percentage is entered, the rate will be calculated from the theoretical throughput. Opposite Rate: 10%
- C. The '--raw_line' flags are similar to those used earlier in other python scripts. They are permutations/combinations of the dataplane test, which will reflect in the window that pops up within the GUI. They all need to have the same format used as in the example (same spacing, units if need be).

-

	D	ataplane Test (dataplane-instance)		
Settings Advanced Configuration	Pass/Fail Report Configuration	on Report † 🗙			
Selected DUT:	eero-root	-	Duration:	15 sec (15 s)	
Downstream/WiFi Port:	1.01.wlan0	-	Upstream/Other Port:	1.01.eth2	
Rate:	85%	-	Opposite Rate:	OKbps	
Path Loss:	10		AP Tx Power:	0	
Thannels (STA DUT)	Mode		Packet Size	Custom Packet Sizes	
AUTO	Auto	•	Custom	88 1200	
No-Change	= 802.11a 802.11b		60 142		
	802.11g	1	256		
3	802.11abg		512		
4	802.11abgn	-	1024		
	802.11bgn		MTU 4000		
7	802.11bg 802.11abgn-AC				
Spatial Streams	Security (STA DUT)		Bandwidth (STA DUT)		
AUTO	AUTO		No-Change		
1	Open		AUTO		
2	WEP WPA		20		
4	WPA2		80		
	WPA3		160		
Traffic Type	Attenuator 1:		Attenuator 2:	Turntable	
UDP	NONE (0)	-	NONE (0)	 NONE (0) 	
TCP	¥1¥2¥	3 🖌 4		4 0.+45.359	
Arm-UDP			25 26 27 2		
Direction	0.+50.550		0.+50.950		
010000					
DUT Transmit					
DUT Receive					
Test running Loop: 1/1 II	terations per Loop: 1/4 Total: 1	/4	top Skip	Another Iteration	Cancel

5. Run the bash script!

[lanforge@ct523c-Scale-Mobsta py-scripts]\$./cv_dataplane_script.sh

Multiplexed REST Access via Nginx Proxy

Goal: Configure an NGINX proxy to allow REST traffic to a variety of isolated LANforge machines

It is possible to configure a Nginx proxy in a manner to allow remote REST clients access to multiple isolated LANforge systems. This leverages the proxy_pass feature in Nginx. There are multiple ways to configure proxy access.

For the example below, we will assume these values:

- public proxy hostname is bizproxy, 10.39.0.44
- bizproxy is running Nginx
- Isolated LAN with LF machines: 192.168.92.0/24
- Example LANforge machines:
 - 192.168.92.10 ct523-jedway1
 - 192.168.92.11 ct522-jedway3
- the LANforge machines need to have GUIs configured to start automatically

LANforge GUI HTTP Processing

The HTTP library that the LANforge GUI incorporates is very simple. It is not configured to parse Host: headers. There is no need to rewrite the Host header when proxying to port 8080.

Proxying to Apache on LANforge (mgt_ip, port 80) is different. If you want to proxy requests to a LF Apache instance on port 80, you should incorporate Host header rewriting. (No examples below, sorry.)

Proxy Request Rewriting

Three ways of making proxy requests include:

- Port Rewriting. Works best with our python libraries.
- Hostname Rewriting, more difficult, but still works with python libraries.
- URL (path-name) Rewriting: this does NOT work well with our python libraries.

Port Rewriting

This manner of proxying just translates different server listening ports to the target machines. It is another easy transformation, but it opens up quite a number of high-numbered ports on bizproxy. Nainx confia:

```
server {
   listen 1910;
   server name
   root /usr/share/nginx/html;
    location / {
       rewrite
                           /(.*) /$1 break;
      proxy pass
proxy redirect
                          http://192.168.92.10:8080;
                          off;
       proxy set header
                          Host $host;
       proxy set header
                          X-Real-Ip $remote addr;
       proxy set header
                          X-Forwarded-For $remote addr;
    }
server {
   listen 1911;
   server name
   root /usr/share/nginx/html;
    location / {
       rewrite
                           /(.*) /$1 break;
       proxy pass
                          http://192.168.92.11:8080;
       proxy redirect
                          off:
       proxv set header
                          Host $host:
       proxy set header
                          X-Real-Ip $remote addr;
       proxy_set_header
                          X-Forwarded-For $remote_addr;
    }
}
```

Use curl to test access:

curl -sqv -H 'Accept: application/html' http://bizproxy:1910/port/1/1/list

Example script usage:

./scenario.py --mgr bizproxy --mgr port 1910 \ --load BLANK --action overwrite

Hostname Rewriting

It is possible to rewrite hostnames and host headers to isolated LF systems. This is **complicated** rewrite because the DNS names need to be present at the developer's workstation. (It is unlikely that the the headers in the HTTP request can be manipulated to add the Host header.) Ideally, the non-isolated LAN DNS can be configured to return the return the IP of bizproxy.corp.me when hostnames like ct523-jedway1.bizproxy.corp.me are requested.

On the developer workstation, this is possible with extra effort on the user side by manipulating the /etc/hosts file on a workstation:

ct523-jedway1

ct523-jedway1.bizproxy.corp.me

10.39.0.44 Nginx config:

}

etc/hosts

```
server {
   listen 80;
    server name ct523-jedway1;
    root /usr/share/nginx/html;
   location / {
        rewrite
                           /(.*) /$1 break;
                           http://192.168.92.10:8080;
        proxy pass
       proxy redirect
                           off;
Host $host;
        proxy set header
        proxy set header
                           X-Real-Ip $remote addr;
                           X-Forwarded-For $remote_addr;
       proxy_set_header
   }
```

Check the URL access using curl:

```
# check by IP:
$ curl -sqv \
    -H 'Host: ct523-jedway1' \
    -H 'Accept: application/json' \
    http://10.39.0.44/port/1/1/list
# check by hostname
$ curl -sqv \
    -H 'Accept: application/json' \
    http://ct523-jedway1.bizproxy.corp.me/port/1/1/list
```

Example script usage:

```
./scenario.py --mgr ct523-jedway1 --mgr_port 80 ∖
--load BLANK --action overwrite
```

Logging HTTP Access

The bizproxy logs should be located in /var/log/nginx. In LF 5.4.6, the GUI can send messages to syslog. Messages from the GUI would look like:

1685573102952: ip[192.168.92.1] sess[] GET url[/port/1/1/list]

Appendix

URL Rewriting is mentioned here so the reader can understand what not to configure.

URL Rewriting

Below is an example permitting REST access to LF hosts by way of a URL prefix. For example, the URL http://bizproxy/92.11/port/1/1/list becomes the URL http://192.168.92.11:8080/port/1/1/list . This is not the best kind of proxy rewriting, but it is the easiest. Using a URL prefix is less ideal because it inherently conflicts with the LANforge python libraries provided. Nginx config:

```
server {
    listen
                  80;
    server name
                 /usr/share/nginx/html;
    root
   # Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
    location /92.10 {
       rewrite
                            /92.10/(.*) /$1 break;
                            http://192.168.92.10:8080;
       proxy pass
       proxy redirect
                            off;
       proxy set header
                           Host biz lflab5 9210;
       proxy set header
                           X-Real-Ip $remote addr;
                           X-Forwarded-For $remote_addr;
       proxy_set_header
    location /92.11 {
                            /92.11/(.*) /$1 break;
       rewrite
                            http://192.168.92.11:8080;
       proxv pass
       proxy redirect
                            off;
       proxy set header
                           Host $host;
                           X-Real-Ip $remote addr;
       proxv set header
       proxy_set_header
                           X-Forwarded-For $remote addr;
    }
}
```

Use curl to query the REST endpoint:

\$ curl -sqv -H 'Accept: application/json' http://bizproxy/92.10/port/1/1/list

This is not compatible with the py-scripts library.

www.candelatech.com | sales@candelatech.com | +1.360.380.1618