MANET and Sensor Network Security
CSD - Computer Security Resource Center

mLAB User Manual

Written by

Aimilios Apostolopoulos and Manos Antonakakis

{aimilios,manos}@nist.gov

http://csrc.nist.gov/manet/index.html

National Institute of Standards and Technology

mLAB

MLAB 1
Chapter 1: INTRODUCTION 3
- 1.1. What is mLab? 3
- 1.2. Overview of mLab 3
- 1.3. Supported platforms 5
- 1.4. Photographs of mLab 6
- 1.5. mLab modules 8
- 1.6. GNU license 8
- 1.7. What mLab is NOT 8

Chapter 2: INSTALLATION 10
Chapter 3: mLab 11
- 3.1. Network Configuration 11
- 3.2. Create Scenario 13
  - 3.2.1. Create New Scenarios 13
  - 3.2.2. Preview – Start Scenario 15
- 3.3. Malicious actions 16
- 3.4. Create Simple Network traffic 17
- 3.5. Remote execution command and quick network testing 18

Chapter 4: mLab side tools 19
- 4.1. Network Information 19
- 4.2. mSniffing – Network Sniffing Tool 20

Chapter 1: INTRODUCTION

1.1. What is mLab?

So, you just designed your new mobile ad hoc network (MANET) algorithm/protocol/application/whatever and successfully evaluated its efficiency with extensive simulations. What is the next step? A field test of a prototype will show whether simulations were on the right track or not, but that's a big leap to take; going from the simulator directly to the real thing.

That is why there are testbeds to bridge that gap. Testbeds are emulators, typically a great number of devices with wireless capabilities deployed in a large space and waiting for you to program them, experiment with them and test your work in almost realistic conditions. However, there are few who can afford to have such a facility in their lab. Usually, it's possible to rent this equipment, run your experiments remotely and, then, receive the results and analyze them. The problem is that your work will be conducted under serious time constrains, especially if you are on a tight budget, not to mention that a lengthy learning curve could consume a significant portion of that budget.

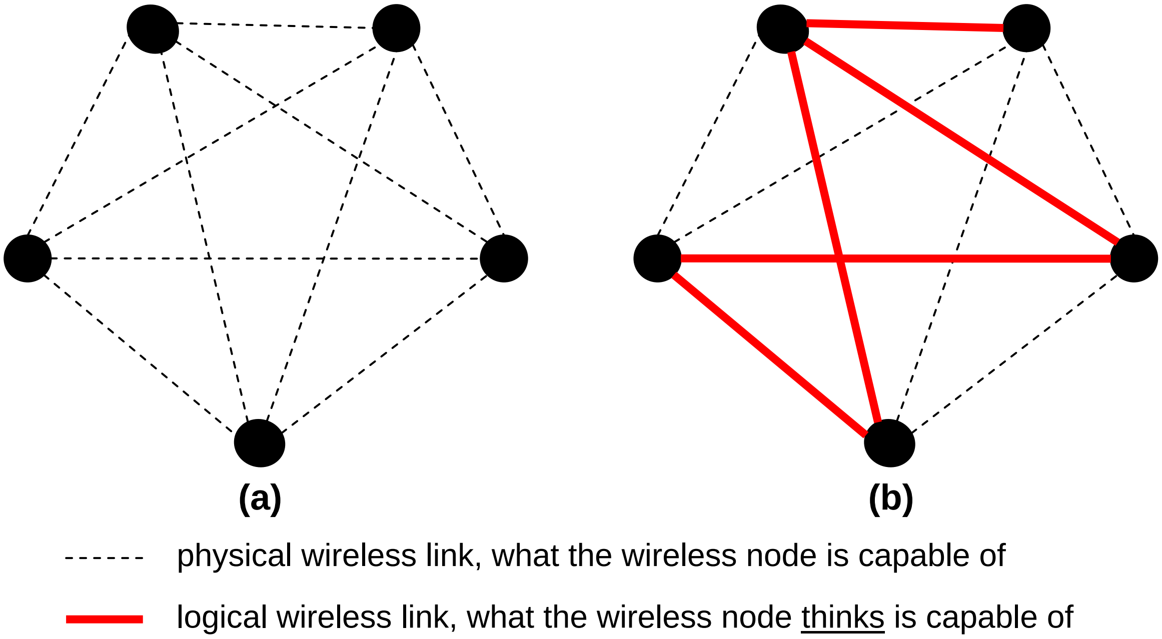
That's where mLab comes in. mLab is also a MANET emulator, a testbed, but the difference with other testbeds is that it is intended to operate in your lab. In brief, mLab's software allows users to automatically generate arbitrary logical network topologies, in order to perform real-time performance measurements of routing protocols or network applications. And on top of that, it's open-source!

1.2. Overview of mLab

The general idea of mLab is having a number of MANET nodes physically close to each other (inside your lab), but force them to "think" that they can only communicate with a selected few of them. That way, we can emulate a logical topology, on which we can run e.g. a routing protocol to analyze its behavior.

So, what does mLab need in order to work? First of all, there is the hardware. mLab is an emulator, not a simulator, **so you have to have the necessary hardware**

equipment [For client: any Linux system with 1 wireless interface, 1 wired interface, for server: Linux x86 (Rec: Debian stable) with 1 wireless interface, 1 wired interface]. That means that each node should be a device that has a wireless (802.11a/b) interface, so that it can communicate with other ad hoc nodes and run MANET protocols. In addition, the device should also have a wired interface (Ethernet or USB), which is used for administrative purposes. In other words, mLab uses the wired interface to transfer files needed for its operation to and from the node, and manipulate its networking elements in such a way that will create the logical topology we want (kind of like blinders are used with horses to force them to see only part of the bigger picture ☺). That leaves the wireless interface free of any interference and, most importantly, emulates an actual MANET, which was the whole point all along.


(a)

(b)

physical wireless link, what the wireless node is capable of
logical wireless link, what the wireless node thinks is capable of

Figure 1-1: (a) physical links, (b) logical links between MANET nodes

You will need a PC to act as a server, which will run the main components of mLab and from which you will control all your nodes. All nodes must have an Ethernet (or USB connection using USB networking) connection with the server, probably through a hub.

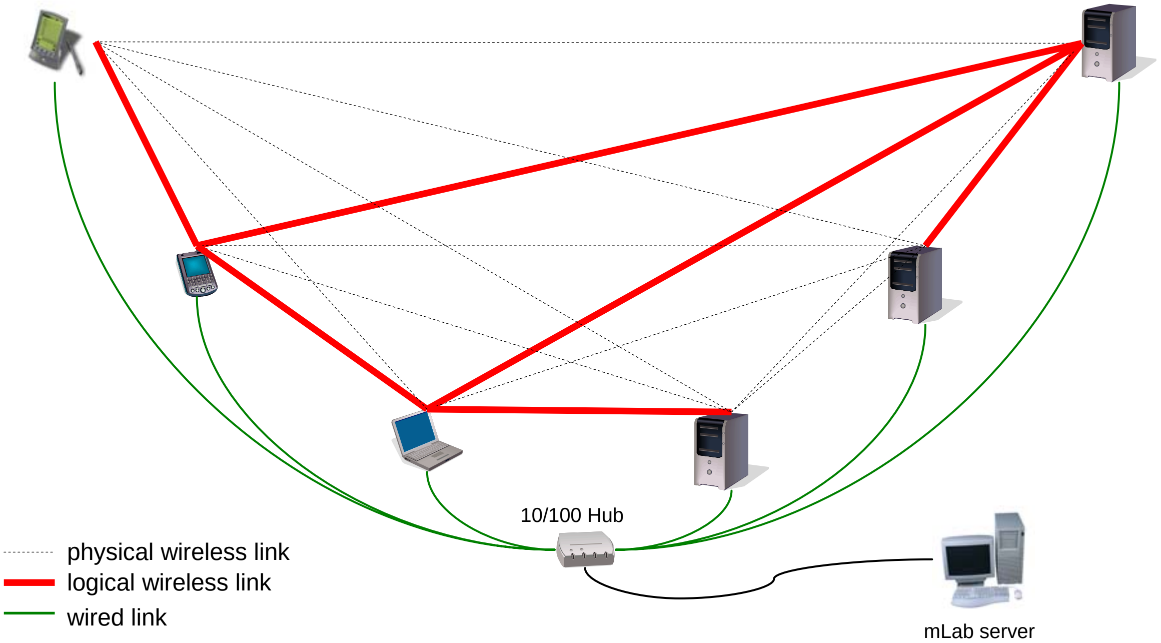


Figure 1-2: Logical diagram of mLab. We use the wired links (green lines) to manipulate the physical wireless links (dotted lines) and create logical wireless links (red lines)

1.3. Supported platforms

mLab was developed and intended only for Linux machines. In other words, any device that can run Linux and can support wireless and wired interfaces can run mLab. Currently, the following platforms have been tested and are supported;

- ix86 (desktop and laptop PCs), running Debian Stable and kernel version 2.4.X (There are available already configured 2.4.27 kernels if anyone is interested)
- ARM (iPAQ, CerfCube, Zaurus), running Familiar OS and Kernel version 2.4.X (There are available already configured 2.4.19 kernels if anyone is interested)

There are pre-configured kernels for all of the above at http://csrc.nist.gov/manet/ that only need to be compiled for the intended platform. Check the installation instructions for more information.

In case you haven't figured it out already, since the ix86 architecture is supported and Linux can run even in 80386 machines (mLab requires, at least, Pentium II however), you can gather all those old PCs you intended to throw away, add a PCMCIA wireless card on each of them and set up a MANET testbed in your lab at a relatively very low cost!

1.4. Photographs of mLab

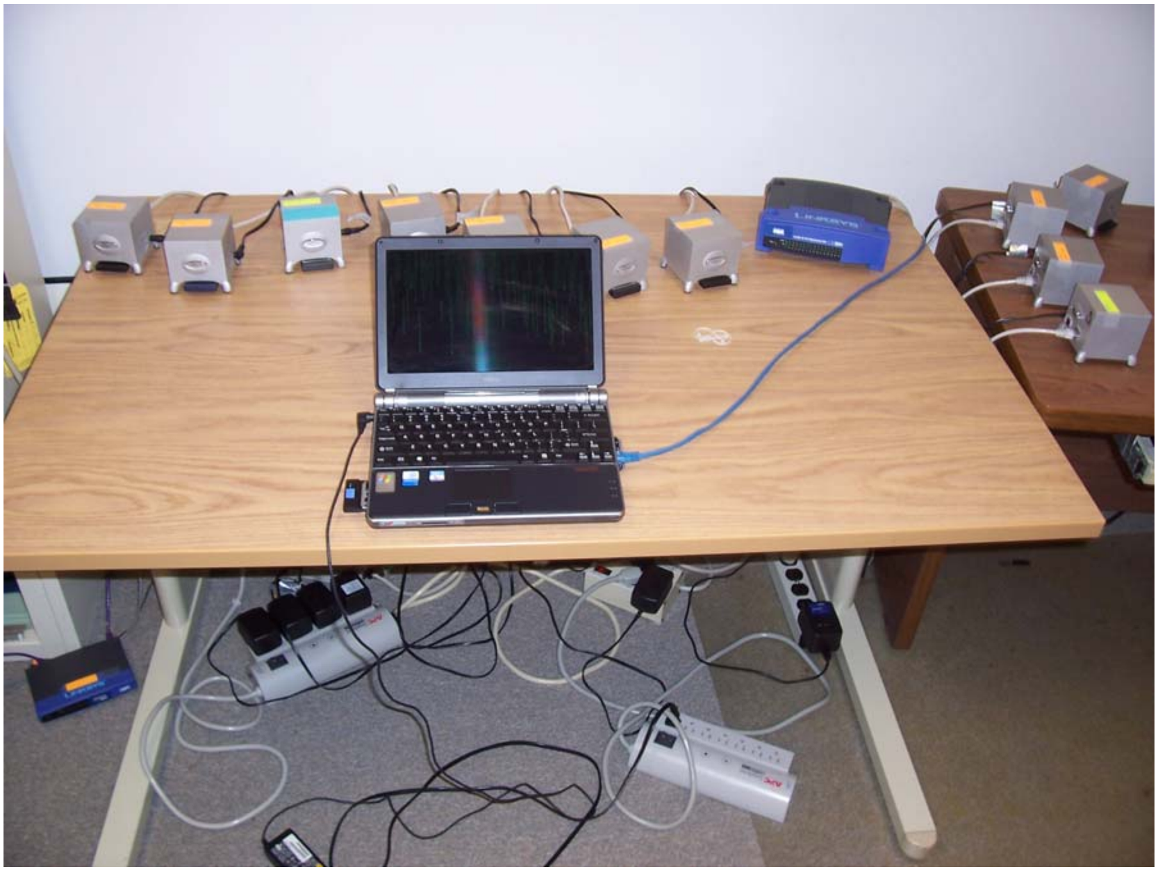


Figure 1-3: Hardware for a mLab. You can see 11 CurfCubes and 1 laptop that are all connected to the mLab Server via a 10/100 Hub

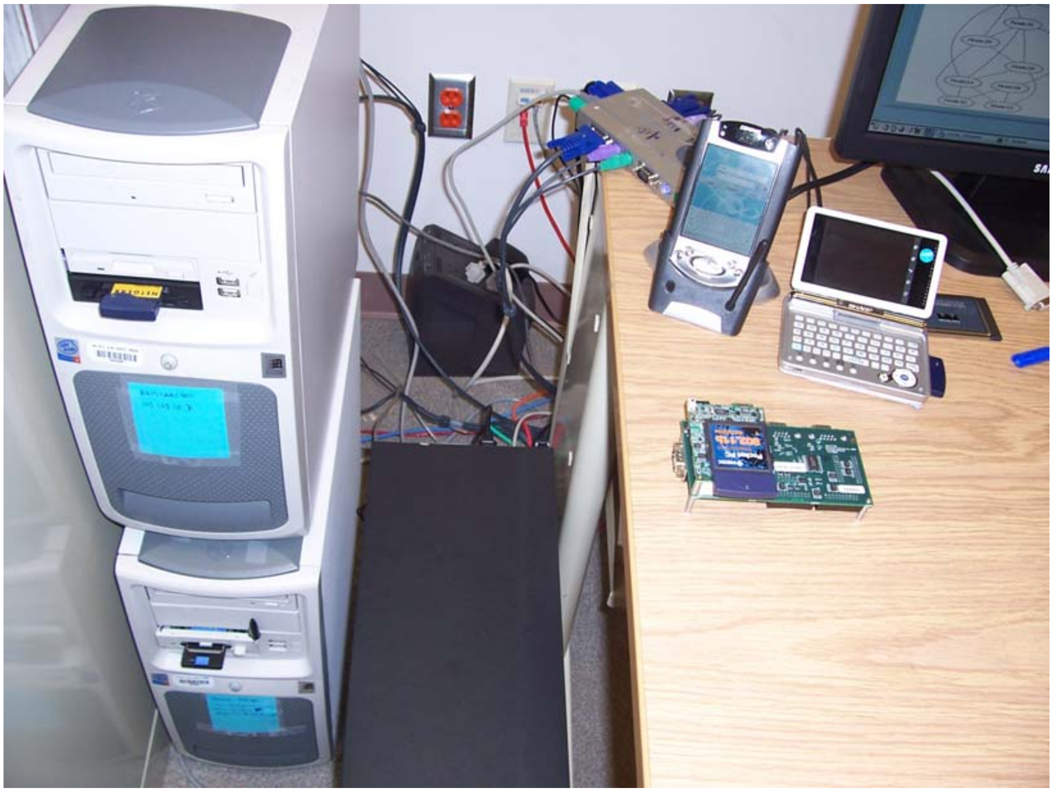


Figure 1-4: Another part of mLab. 2 PCs with PCMCIA wireless cards, a SONY ZAURUS, a COMPAQ iPAQ and a CurfBoard, all mLab nodes

Figure 1-5: … Wireless networking …

1.5. mLab modules

mLab consists of these modules;

- Network Configuration module, with which the user can assign Wired IP, Wireless IP and Wireless MAC to every node.
- mNet (Create Scenario) module, which is responsible for generating random logical topologies based on user-defined rules.
- Malicious actions module, where the user can select and optimize from a number of attacks against the network.
- Network traffic generator (Create Simple Network Traffic), where the user can create simple traffic between nodes using a number of different protocols (UDP-TCP [sockets with any possible port available] and ICMP).
- Network Information tool, where the user can view and set several options of the wireless network, such as signal strength, wireless network administrate fixtures, wireless adapter hardware and software info.
- Sniffing (Network Sniffing Tool), which is a sniffing tool and allows the user to observe all overheard traffic and apply filters on it.

The next chapters describe, in more detail, how these features work.

1.6. GNU license

mLab is under the GNU license, which means that it's open-source. Visit http://www.gnu.org/ for further details. So, you are free, welcome and expected to download mLab, use it, alter it, improve it, etc.

1.7. What mLab is NOT

A really important concept you must realize from the start is that mLab emulates a MANET at the network layer of OSI (layer 3) and above. That means that it has nothing to do with neither the physical nor the data link layer (layers 1 and 2). Don't forget that the nodes will, most probably, be close to each other in your lab, so every node can "hear" every packet transmitted by any node, but ignores most of them, according to what mLab dictates. So, in order to avoid any future misunderstandings, mLab does NOT emulate a MANET at the data link layer, it emulates a MANET at the network layer. If you want to use mLab to test your data link protocol, unfortunately, you should seek something else and you can stop reading this manual (who likes manuals, anyway? ☺).

Another thing, which you should have in mind, is that mLab was designed and developed having security-related issues in mind. As a result, some of its aspects focus on security, rather than e.g. network performance. However, it's an open-source tool, so you are free to alter it any way you see fit.

Also, since mLab is not a commercial release, you will find that it lacks extensive error control routines in its GUI (text-boxes, etc). You cannot damage any hardware, don't worry, and just be a little careful with your input and with buttons that say stuff like

“Send to node” (when you haven’t a node connected), if you want to avoid error tracking, which you probably do!!

Chapter 2: INSTALLATION

The installation of the mLAB is very simple. First of all you must be root. After that you extract the mLAB.tar.gz and you enter the mLAB folder. Then you should run the "./CHECK_ENV" script that will inform you whether or not you have the necessary applications in order to install the mLAB. The most important applications that you must have preinstall before you are ready to install mLAB are:

- Greaphviz [http://www.graphviz.org/]
- QT3 [http://www.trolltech.com/]
- Libpcap0.9.4 [http://www.tcpdump.org/]
- Expect [http://expect.nist.gov/]
- Sed ,fgrep,xargs,console (usually these are in the standard installation of any Linux system.)

When the "./CHECK_ENV" script finish without any errors then you are almost ready to run the "install.sh" which will actually compile the mLAB. First you must change the /root/.bashrc file and add these 3 lines:

- **export MLABROOT=/usr/local/mlab** or any other path of the directory that you wish the mLAB to install in.
- **export PATH=$PATH:$MLABROOT/scripts:$MLABROOT**
- **export MLABSERVER=192.168.10.1** The servers wired IP (not the WIRELESS).

After editing the .bashrc file and restarting your console, you are ready to install the mLAB. Just run the install.sh script and follow the instructions.

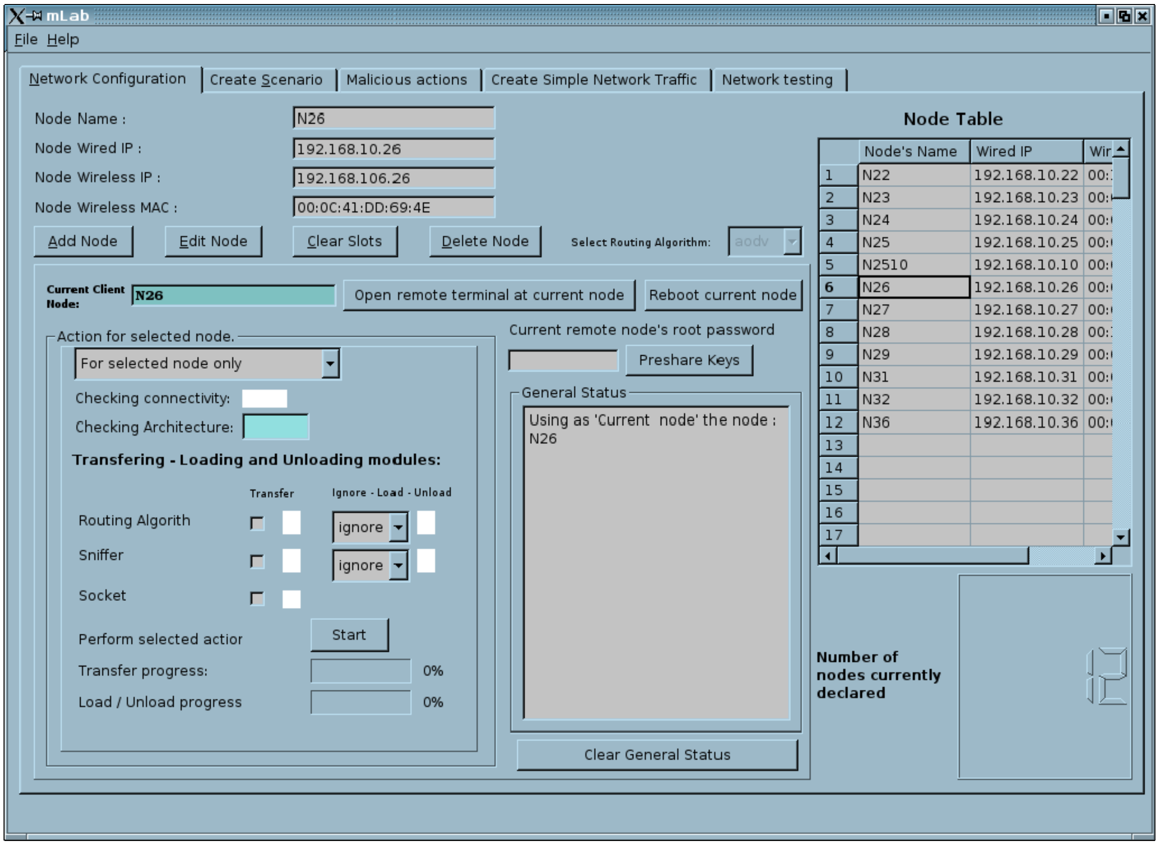
Chapter 3: mLab

This chapter describes, in detail, what each GUI element of each mLab module does.

Before you start using mLab, you must deploy your network and assign each node with a wired IP and a wireless IP, as indicated in the installation instructions.

Note, also, that you should be logged in as root in your PC that acts as a server, in order for mLab to work properly.

3.1. Network Configuration



Node Name : N26

Node Wired IP : 192.168.10.26

Node Wireless IP : 192.168.106.26

Node Wireless MAC : 00:0C:41:DD:69:4E

Select Routing Algorithm: aodv

Current Client Node: N26

For selected node only

Routing Algorith ☐ ignore

Sniffer ☐ ignore

Socket ☐

Transfer progress: 0%

Load / Unload progress 0%

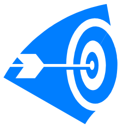
Using as 'Current node' the node : N26

Node Table

	Node's Name	Wired IP	Wir
1	N22	192.168.10.22	00:
2	N23	192.168.10.23	00:
3	N24	192.168.10.24	00:
4	N25	192.168.10.25	00:
5	N2510	192.168.10.10	00:
6	N26	192.168.10.26	00:
7	N27	192.168.10.27	00:
8	N28	192.168.10.28	00:
9	N29	192.168.10.29	00:
10	N31	192.168.10.31	00:
11	N32	192.168.10.32	00:
12	N36	192.168.10.36	00:
13			
14			
15			
16			
17			

Figure 3-1: Network Configuration module tab

In this tab you can declare your MANET nodes to mLab and transfer required files to the nodes. As you can see in figure 3-1, at the top left section, you can Add, Edit or Delete a node in mLab. Of course, you must know each node’s wired and wireless IP; otherwise, mLab won’t be able to find and communicate with the node. The right section of the tab is the list of all the nodes you have declared and their total number.



It is a good idea to use similar numbers for each node's wireless and wired interface, so that you won't get confused. For example, you can use 192.168.10.26 for the Ethernet IP of node 26 and 192.168.106.26 for its wireless IP.

This configuration of mLab can support up to 150 nodes. Theoretically, there is no limit to the number of nodes mLab can handle, and all you need to do to support more nodes is change the size of the node table in the source code and re-compile. Of course, then you will have other problems, such as stuffing a single room with 500 PCs, whose wireless cards must all be able to communicate with each other, wiring all machines with Ethernet and power cables (think of the number of hubs you need!), not to mention the ridiculously great number of packet collisions, that will, most likely, ruin every experiment! Just thinking ahead… 150 nodes should be more than enough.

At the left bottom section, you can configure what each node will be loaded with. You can select to transfer to the selected node or all nodes the required files for the:

- *Routing algorithm*. As the observant reader/user can realize, only the AODV routing algorithm is supported in this distribution. If you want to use other routing, feel free to add it ☺.
- Sniffer. This refers to mSniffing, the sniffing tool used by mLab to overhear all ongoing traffic. If you wish the clients to be able to run the sniffer module, then you must make sure that the client node has already installed the libpcap-0.7.2 or later module.
- Socket. Two nodes need the associated files in order to achieve peer-to-peer communication. That is needed when you will need to create network traffic yourself (with the Network Traffic Generator module).

For all of the above groups of files and mechanisms (except Socket) you can select load or unload them, which means you can transfer what is required to the node (load) or delete it from the node (unload). The option "ignore" does exactly what it implies; it does nothing with the files. If they are in the node, it ignores them flamboyantly and leaves them be, if they are not in the node, it leaves the node be. That's it!

Under normal conditions, you just have to select and load all options in all nodes and continue on with the rest of mLab. So, don't worry about it too much. When all your selections are set, just press the 'Start' button. The 'General Status' text-box in the middle-bottom part of the screen will be displaying messages as to how the transfer progresses. You should see the little boxes next to each option turn green if everything went as it should be, red if something went wrong and black if you had selected "ignore". These procedures will probably take a few seconds per node.

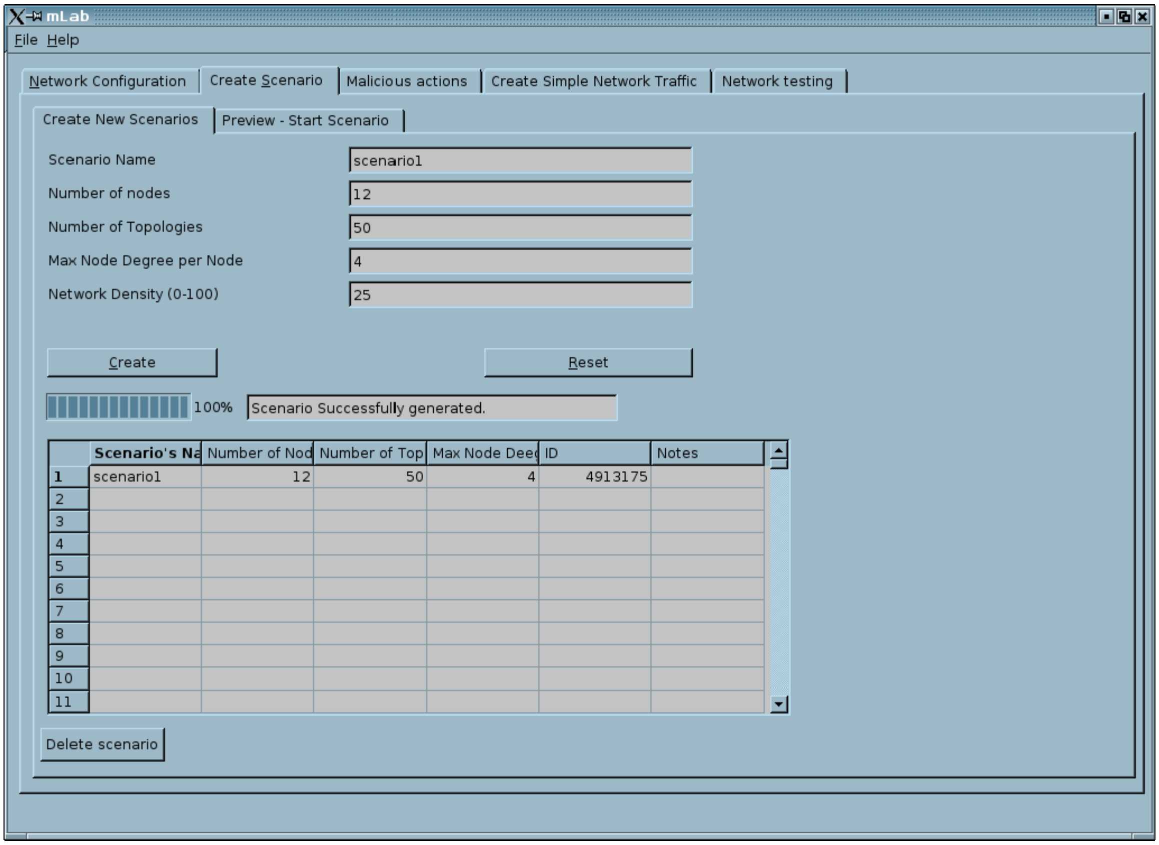
You only need to transfer something one time. After it is transferred it stays in the node, you don't have to do it again every time you change something in its configuration.

Sometimes, these files may be successfully transferred and loaded into the node, but the relevant box will turn red. That may happen because the box is anxious to get colored, so it checks the process list in the remote node, before they are fully loaded. Don't worry about it, just click the button again. If it doesn't work…again, you probably have a loose cable somewhere…

3.2. Create Scenario

In these tabs you can create, preview and start topology scenarios. A scenario is a series of random topologies. After setting various options, several random topologies are generated by the mNet module. You can then change between topologies, while routing algorithms/protocols/applications are running. That way, you can observe how they react to these topology changes.

3.2.1. Create New Scenarios



Scenario Name: scenario1
Number of nodes: 12
Number of Topologies: 50
Max Node Degree per Node: 4
Network Density (0-100): 25

100% Scenario Successfully generated.

	Scenario's Na	Number of Nod	Number of Top	Max Node Deeg	ID	Notes
1	scenario1	12	50	4	4913175	
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						

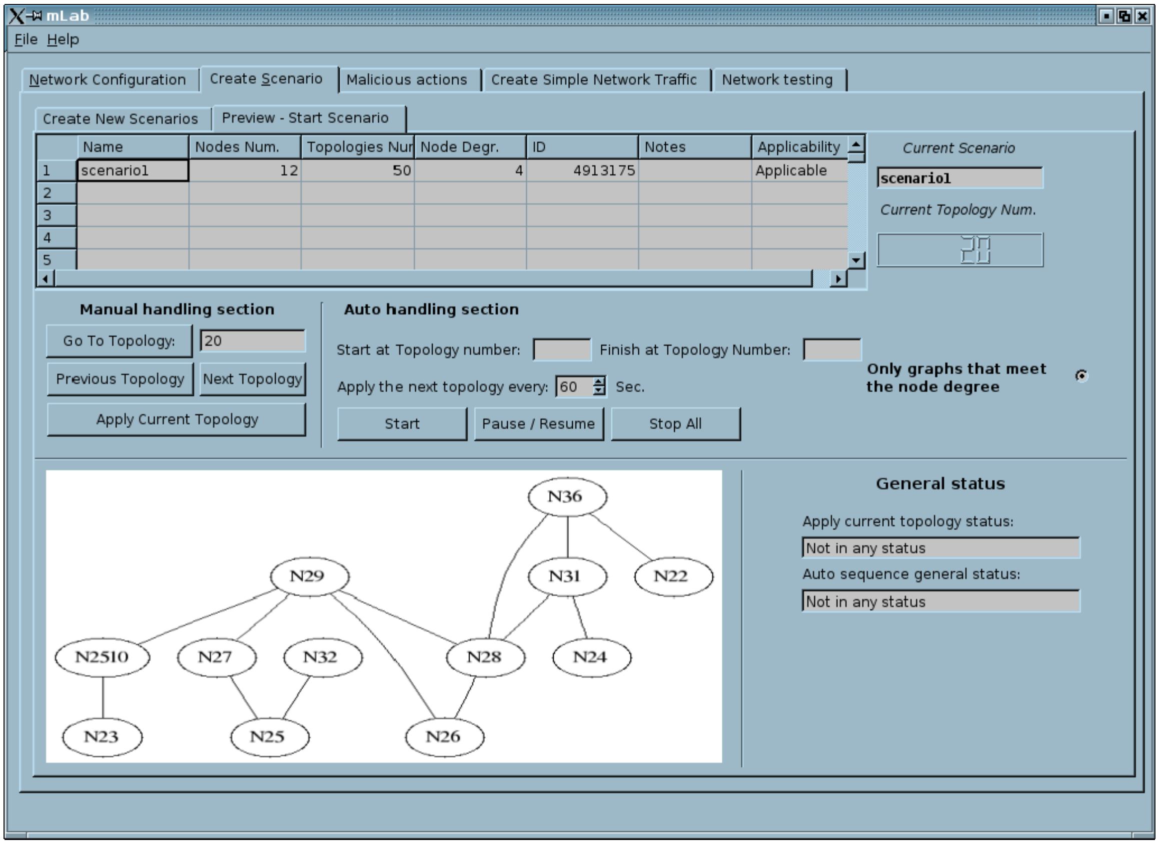
Figure 3-2: Create New Scenarios tab

Before creating a new scenario, you must set these options;

- **Scenario name.** The name of the scenario (big surprise!).
- **Number of nodes.** How many nodes the generated topologies will have. Of course, this should be less or equal to the number of nodes you have declared in the Network Configuration tab.
- **Number of Topologies.** How many topologies this scenario will have.
- **Max Node Degree per Node.** The degree of a node is how many connections it has with other nodes. When generating random topologies a node will have a random number of connections with other nodes from zero to the value you set here.
- **Network Density (0-100).** How dense the network will be. That is determined by an adjacency matrix, where element every element (x,y) determines if there is a connection between node x and node y. The number you set here is the possibility to have a ‘1’ in each position (x,y), therefore, have a connection. That’s all the math there is around here, don’t be too alarmed.

As you can see, we mLab assumes Boolean connectivity between nodes. This means that two nodes are either connected or they are not. No link quality issues here. Once again, mLab operates at the network layer, not the data link layer…

3.2.2. Preview – Start Scenario



	Name	Nodes Num.	Topologies Nur	Node Degr.	ID	Notes	Applicability
1	scenario1	12	50	4	4913175		Applicable
2							
3							
4							
5							

Current Scenario: scenario1

Current Topology Num.: 20

Go To Topology: 20

Start at Topology number: Finish at Topology Number:

Apply the next topology every: 60 Sec.

Only graphs that meet the node degree ●

Apply current topology status: Not in any status

Auto sequence general status: Not in any status

Figure 3-3: Preview - Start Scenario Tab

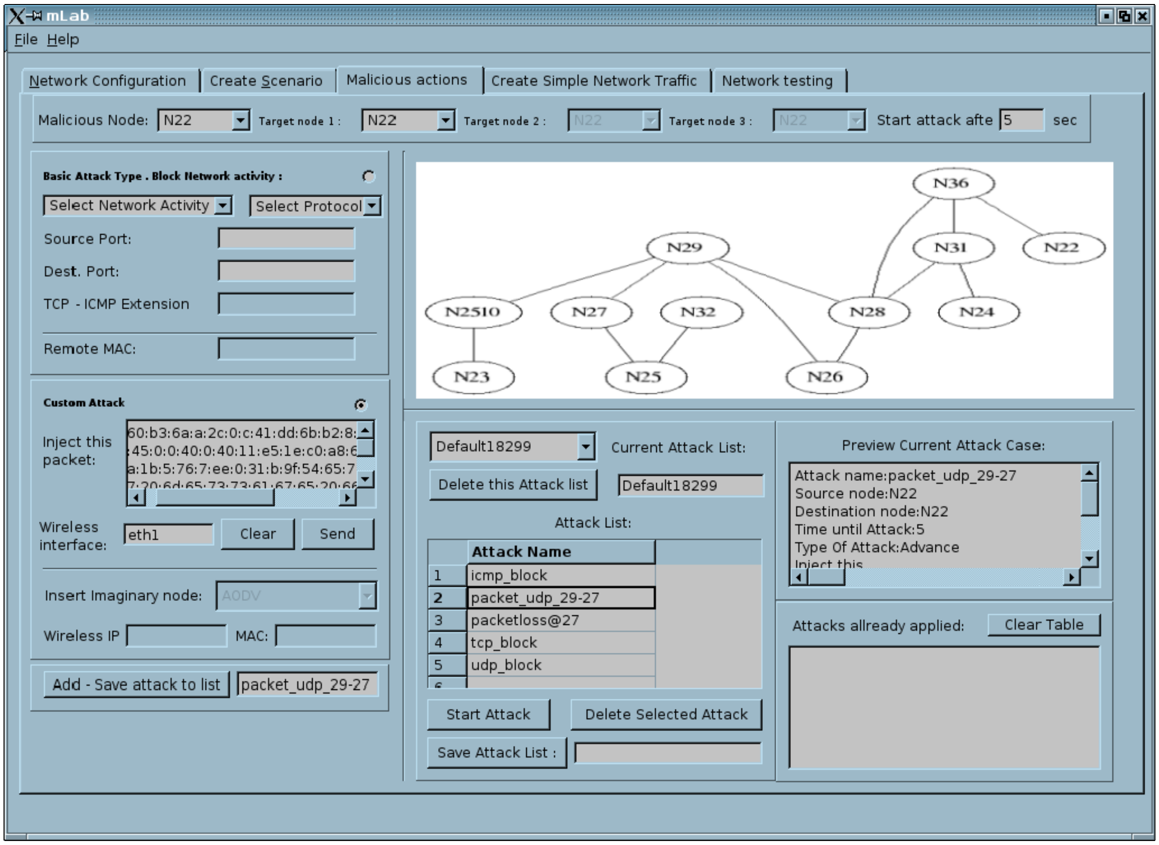
In this tab you can select one of the scenarios you have created and change between the generated topologies.

At the top section you can see all scenarios with their options and, also, whether they are applicable in your network. For example, a scenario involving 50 nodes is not applicable if your network only haw 10. You can select a scenario, simply by clicking on it. You can see which scenario is selected at the top right section.

Right below that, there is the handling section of the scenario. You can choose to manually change between topologies or set a timer that will automatically proceed to the next topology.

After setting these options, your network will be changing between MANET topologies. You are ready now to generate traffic or create malicious activity on these topologies.

3.3. Malicious actions



Malicious Node: N22
Target node 1 : N22
Target node 2 : N22
Target node 3 : N22
Start attack afte 5 sec

Inject this packet:

```
60:b3:6a:a:2c:0:c:41:dd:6b:b2:8:
:45:0:0:40:0:40:11:e5:1e:c0:a8:6
a:1b:5:76:7:ee:0:31:b:9f:54:65:7
```

Wireless interface: eth1

Insert Imaginary node: AODV

Add - Save attack to list: packet_udp_29-27

Default18299 Current Attack List: Default18299

	Attack Name
1	icmp_block
2	packet_udp_29-27
3	packetloss@27
4	tcp_block
5	udp_block

Preview Current Attack Case:

```
Attack name:packet_udp_29-27
Source node:N22
Destination node:N22
Time until Attack:5
Type Of Attack:Advance
Inject this
```

Figure 3-4: Malicious actions tab

In this tab you can select from a range of the most common attacks against some of the most common protocols or just inject into the network any packet you want in hexadecimal form. Assuming that you have created a scenario and have it running as explained in the previous section, you should see a logical diagram of your wireless network.

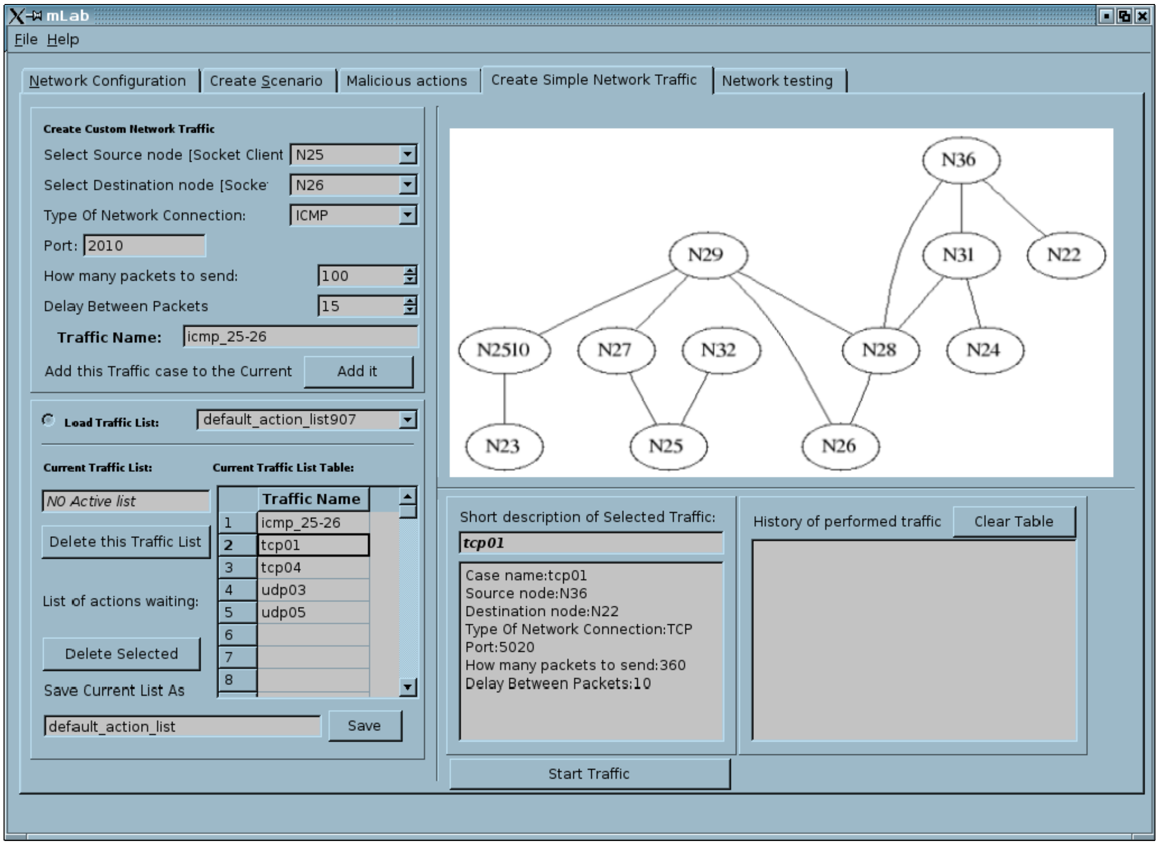
You can choose to attack against TCP, UDP or ICMP protocol. The types of attacks are:

- blocking all incoming traffic
- blocking all outgoing traffic
- blocking both incoming and outgoing traffic
- generating packet loss. More specifically, it will generate packet loss for 5 seconds, return to normal for the next 35 seconds and start over again for a total of 10 times.

Packet loss details are hard-coded. You can change these times or add your own version of the attack.

After, spitefully, planning the attack, give it an appropriately evil name at the bottom left section of the screen and add it to the attack list and save it, in order to repeat it anytime. You must only select your attack and press the ‘Start Attack’ button to unleash it! Then, you can use any network monitoring tool, like Ethereal, to find out what you have just done to the poor network…

3.4. Create Simple Network traffic



Select Source node [Socket Client: N25
Select Destination node [Socke: N26
Type Of Network Connection: ICMP
Port: 2010
How many packets to send: 100
Delay Between Packets: 15
Traffic Name: icmp_25-26
Add this Traffic case to the Current

○ **Load Traffic List:** default_action_list907

Current Traffic List: *NO Active list*

Current Traffic List Table:

	Traffic Name
1	icmp_25-26
2	tcp01
3	tcp04
4	udp03
5	udp05
6	
7	
8	

Save Current List As: default_action_list

Short description of Selected Traffic: ***tcp01***

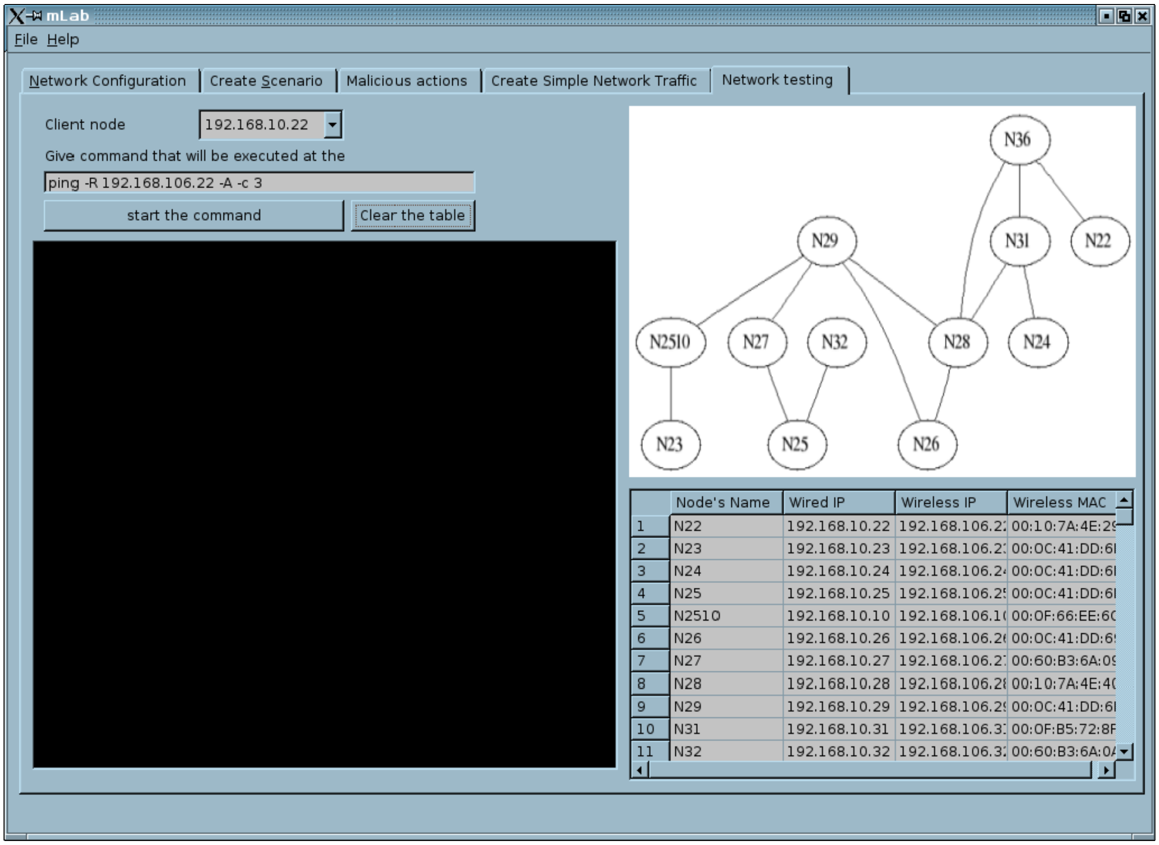
Case name:tcp01
Source node:N36
Destination node:N22
Type Of Network Connection:TCP
Port:5020
How many packets to send:360
Delay Between Packets:10

Figure 3-5: Create Simple Network Traffic tab

Here, you can create a simple, constant data flow between two nodes in the network (peer-to-peer connection). Again, you should see a logical diagram of the network at the upper right section of the screen.

You can select the source and destination nodes, the protocol (TCP, UDP, ICMP), the port you will use, how many packets you will send and how long the delay will be between two consecutive packets. Like in the ‘Malicious Actions’ tab, you can save your traffic cases in a list. In the bottom part of the screen you can see a short description of the selected traffic case.

3.5. Remote execution command and quick network testing



Client node 192.168.10.22

Give command that will be executed at the

```
ping -R 192.168.106.22 -A -c 3
```

	Node's Name	Wired IP	Wireless IP	Wireless MAC
1	N22	192.168.10.22	192.168.106.2	00:10:7A:4E:2
2	N23	192.168.10.23	192.168.106.2	00:0C:41:DD:6
3	N24	192.168.10.24	192.168.106.2	00:0C:41:DD:6
4	N25	192.168.10.25	192.168.106.2	00:0C:41:DD:6
5	N2510	192.168.10.10	192.168.106.1	00:0F:66:EE:6
6	N26	192.168.10.26	192.168.106.2	00:0C:41:DD:6
7	N27	192.168.10.27	192.168.106.2	00:60:B3:6A:0
8	N28	192.168.10.28	192.168.106.2	00:10:7A:4E:4
9	N29	192.168.10.29	192.168.106.2	00:0C:41:DD:6
10	N31	192.168.10.31	192.168.106.3	00:0F:B5:72:8
11	N32	192.168.10.32	192.168.106.3	00:60:B3:6A:0

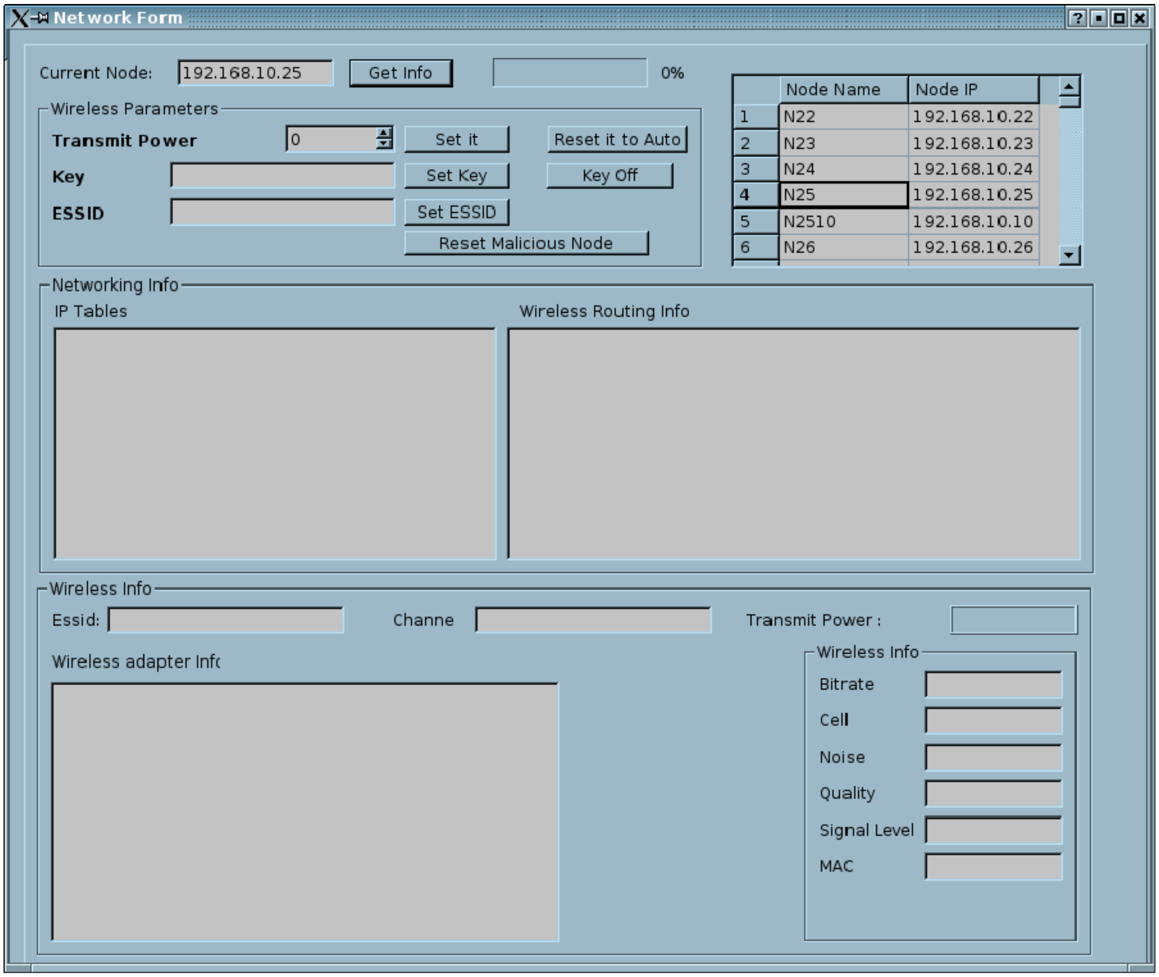
Figure 3-6: Create Network Traffic tab

This is a simple tool to check whether or not your new topology case has been successfully applied. Also, you can execute any command line command you wish on any remote node. Please note that while the remote command is being executed, the mLAB is being focused on it and no other mLAB fixture is available during the command execution.

Chapter 4: mLab side tools

mLab comes with two auxiliary tools that give general information about the network.

4.1. Network Information



Current Node: 192.168.10.25

0%

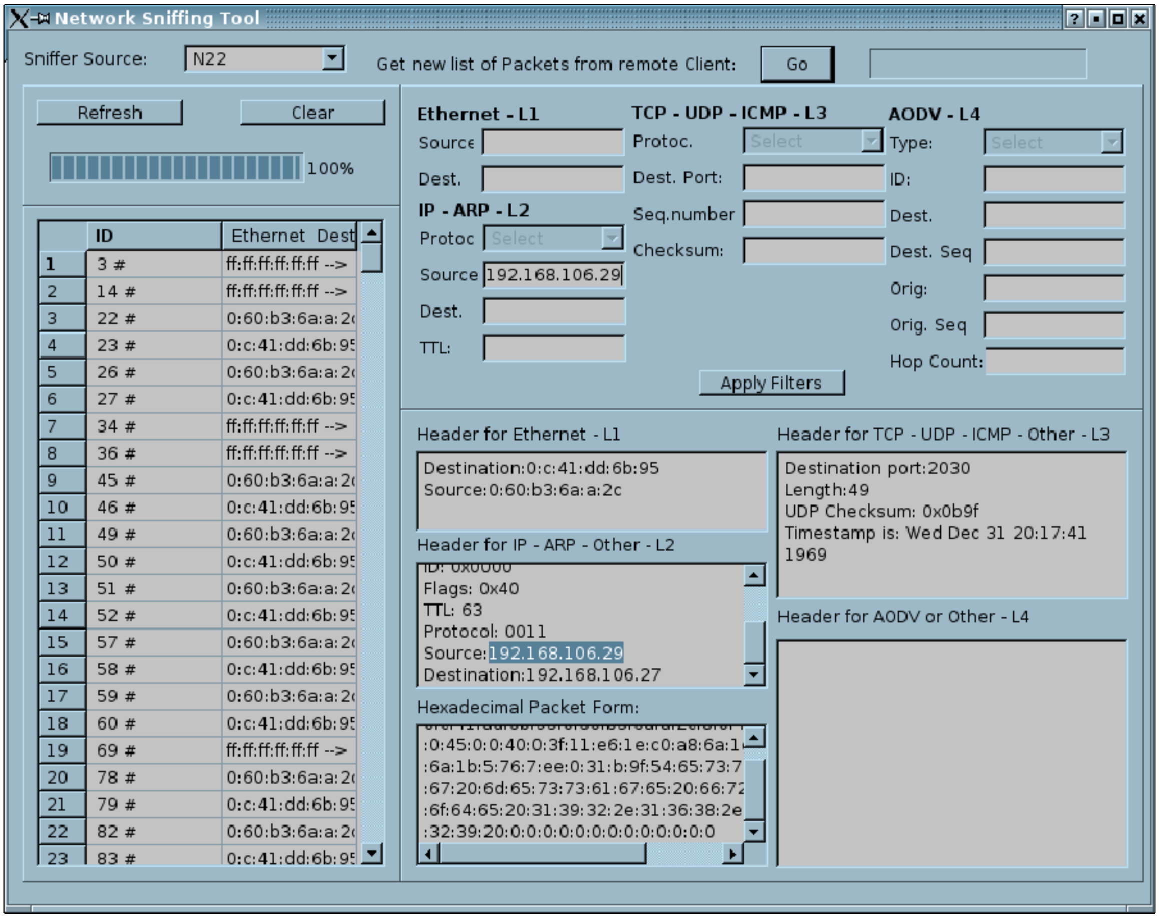
Transmit Power 0

	Node Name	Node IP
1	N22	192.168.10.22
2	N23	192.168.10.23
3	N24	192.168.10.24
4	N25	192.168.10.25
5	N2510	192.168.10.10
6	N26	192.168.10.26

Figure 4-1: Network Information form

Not much to say here; this tool gives you information about the wireless characteristics of your network. Just select the node you want from the list at the top left section and push the 'Get Info' button. You can see and alter the transmit power, the WAP key it will use and the ESSID. The 'Reset Malicious Node' button cancels any malicious activity of the node and resets him to normal routing (if only it was that simple in real communication…). You can also see IP tables information, wireless routing information, etc.

4.2. mSniffing – Network Sniffing Tool



Sniffer Source: N22

Source 192.168.106.29

	ID	Ethernet Dest
1	3 #	ff:ff:ff:ff:ff:ff -->
2	14 #	ff:ff:ff:ff:ff:ff -->
3	22 #	0:60:b3:6a:a:2
4	23 #	0:c:41:dd:6b:9
5	26 #	0:60:b3:6a:a:2
6	27 #	0:c:41:dd:6b:9
7	34 #	ff:ff:ff:ff:ff:ff -->
8	36 #	ff:ff:ff:ff:ff:ff -->
9	45 #	0:60:b3:6a:a:2
10	46 #	0:c:41:dd:6b:9
11	49 #	0:60:b3:6a:a:2
12	50 #	0:c:41:dd:6b:9
13	51 #	0:60:b3:6a:a:2
14	52 #	0:c:41:dd:6b:9
15	57 #	0:60:b3:6a:a:2
16	58 #	0:c:41:dd:6b:9
17	59 #	0:60:b3:6a:a:2
18	60 #	0:c:41:dd:6b:9
19	69 #	ff:ff:ff:ff:ff:ff -->
20	78 #	0:60:b3:6a:a:2
21	79 #	0:c:41:dd:6b:9
22	82 #	0:60:b3:6a:a:2
23	83 #	0:c:41:dd:6b:9

```
Destination:0:c:41:dd:6b:95
Source:0:60:b3:6a:a:2c
```

```
Flags: 0x40
TTL: 63
Protocol: 0011
Source:192.168.106.29
Destination:192.168.106.27
```

```
:0:45:0:0:40:0:3f:11:e6:1e:c0:a8:6a:1
:6a:1b:5:76:7:ee:0:31:b:9f:54:65:73:7
:67:20:6d:65:73:73:61:67:65:20:66:72
:6f:64:65:20:31:39:32:2e:31:36:38:2e
:32:39:20:0:0:0:0:0:0:0:0:0:0:0:0:0
```

```
Destination port:2030
Length:49
UDP Checksum: 0x0b9f
Timestamp is: Wed Dec 31 20:17:41
1969
```

Figure 4-2: mSniffing - Network Sniffing Tool

This tool is a Sniffer that operates in the nodes. First, you must select the node from which you will get the sniffing information and then push the ‘Go’ button to communicate with the node and receive this information. You will see a list of all the overheard traffic at the bottom left section of the screen. Unless you are permanently brain-damaged or your name is Neo, you might want to use the filtering options at the top section of the screen.

When you click on a packet, it is parsed and all information is presented in a more readable form, at the bottom right section of the screen.

There is also the hexadecimal form of the packet there, so you can copy and paste it in the ‘Malicious Actions’ tab to create a simple Replay attack!