Introduction to the environment of Imunes

One of the objectives of Imunes-Sorbonne is to be used in TMEs. The purpose of TMEs is to understand how a network works. To do this, we will have to modify the network parameters of the machines. These modifications are not authorized on University machines since they require super-user (or root) access rights. To circumvent this limitation, we are going to use a network emulator which makes it possible to create and test in a realistic way several systems and equipment connected in networks.

Getting started with the network emulator

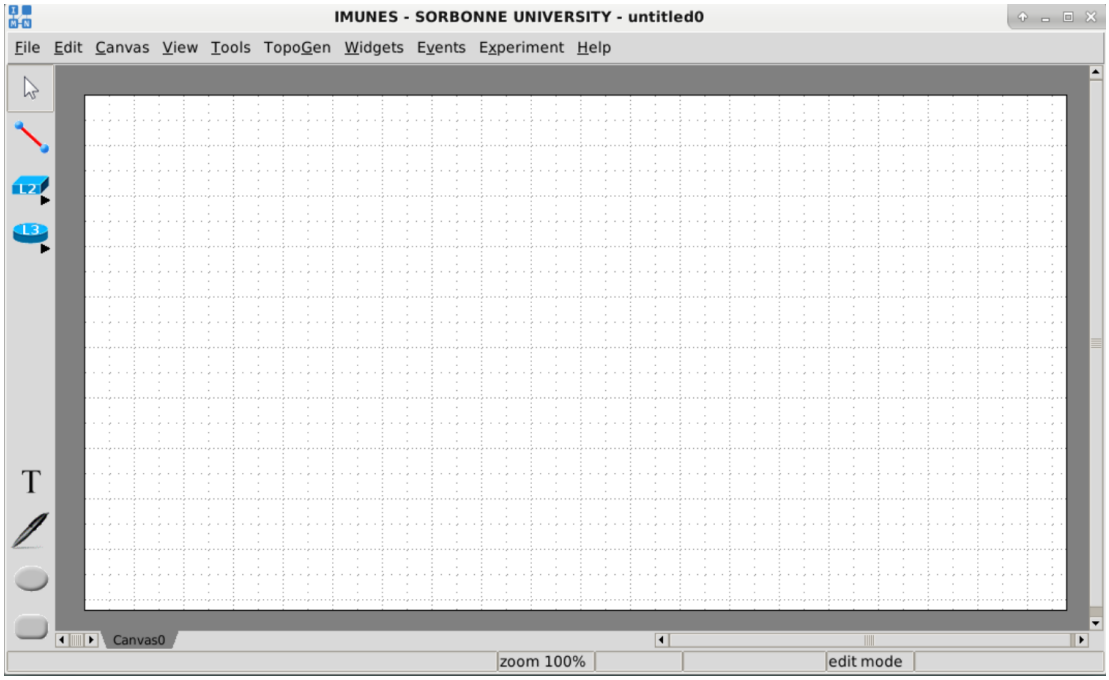
Most TMEs aim to study and illustrate network concepts, which then require a network 😊. We will work with network emulation software which allows from its graphic interface to create several network elements (router, switch, host) and connect them with wired links. When you run the emulation, the hosts will be accessible normally through a real classic terminal with a Linux shell. The routers (Linux) are configured via an interface similar to that of the OS of CISCO routers.

The switches are based on Open VSwitch which is a complete open source software switch. It is designed to allow massive automation of the network via several commands making it programmable, while supporting recent standard management interfaces and protocols such as 802.1q or NetFlow. Additionally, it is designed to support distribution across multiple physical servers similar to VMWare's vNetwork distributed vSwitch or Cisco's Nexus 1000V. It is now increasingly used in cloud environments to flexibly connect physical and virtual machines.

LAUNCH AND GUI

Launch the network emulator by clicking on the 'Network emulator' (IMN) icon on the panel at the bottom of your virtual machine's desktop.

The IMUNES GUI:



The white space in the middle is called a canvas. This is where you create the network topology using the toolbox. You can increase its size from the Canvas menu.

TOOLBOXHere is the description of some tools that allow, among other things, to create the nodes of a network:

 - allows you to select and move the elements.

 - creates the links between the nodes. Click on the first node, hold down the click, then point to the second node and release the click.

 - create a switch (bridge)

 - creates a router that primarily performs routing functions using Docker technology and Quagga routing software.

 computer- create a mini Linux PC based on Docker technology. In it, it is not possible to launch applications with a graphical interface (X Window).

 MyPC- creates a Linux PC by reusing the complete system of the machine. It is possible to launch any application.

 - emulates a physical interface (RJ45). This allows an emulator node to be connected to a real Ethernet interface.

 - creates a self-configured Wi-Fi hotspot. Should be placed in the canvas before Wi-Fi stations.

 - creates a self-configured Wi-Fi station associated with the previous access point.

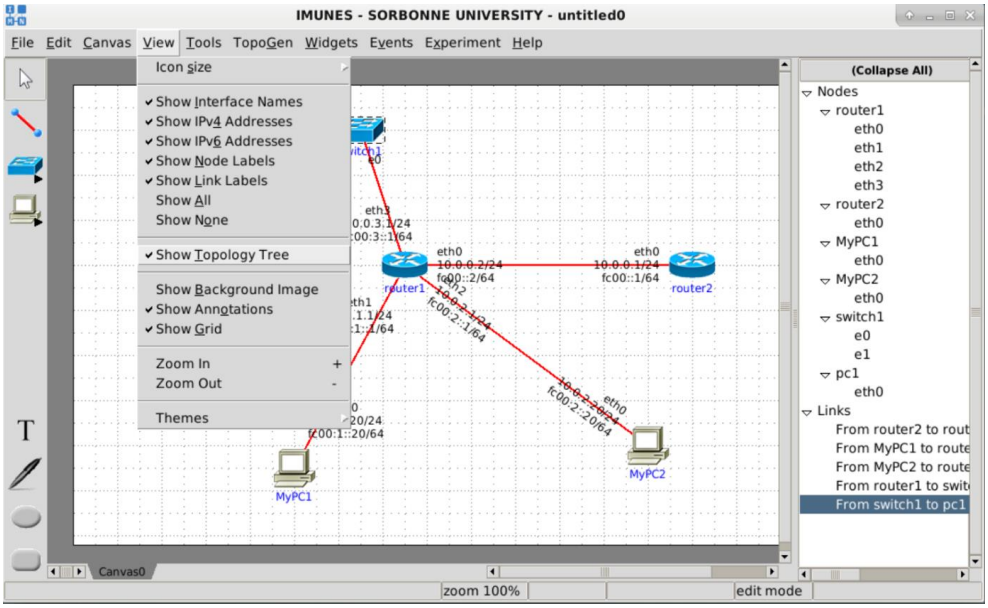
Other nodes are also defined. In addition, drawing tools allow you to add text, draw an oval, a rectangle or a free shape on the canvas.

CREATING AND CONFIGURING NODES/LINKS

To create an object on the canvas, simply choose the type of object in the toolbar and click on a place on the canvas where you want to place it.

To configure a link you can right click on the object and click on the "Configure" option. You can also double-click the link to bring up the configuration window.

It is also possible to display the complete list of objects by going through**`View->Show Topology Tree`**.



This will display details about all objects created by IMUNES and allow quick access to them.

START AN EXPERIMENT

Until you run an experiment, your network is just an image. In order to activate it and access the systems of the different nodes created, choose from the menu:

`Experiment->Execute`

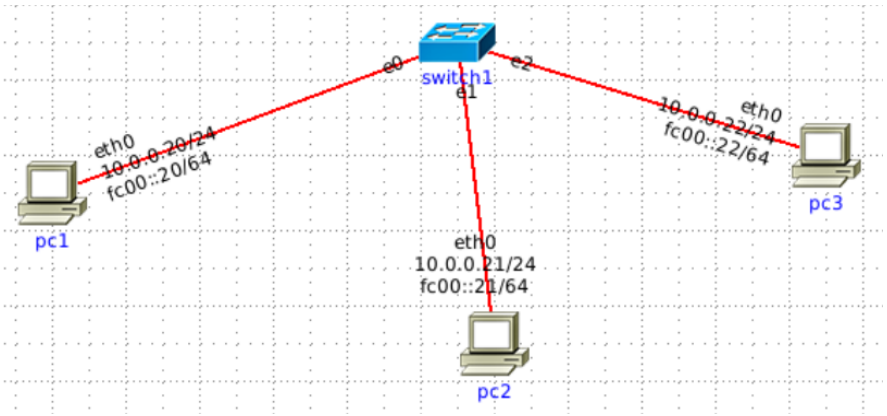
You can see all the actions performed by IMUNES in the status bar at the bottom of the GUI as soon as you launch the experiment.

When the node creation is complete, you can now interact with the devices involved in your topology. Double-click on a device and a terminal will appear. If it is a PC this window will be a bash shell. If it's a router, then a vtysh configuration interface will appear. The vtysh shell of the routers, which is emulated by the Quagga software, is close to that of Cisco IOS. The commands you use will therefore be similar to those you would type if you are using a Cisco router.

You can use the**widgets**to quickly display certain PC or router information. For example, you can display the routing table of a node as soon as you point it on the canvas and without clicking by activating the option**Routing table**in the menu**widgets**.

A Concrete Step-by-Step Example: Switch + 3 Hosts

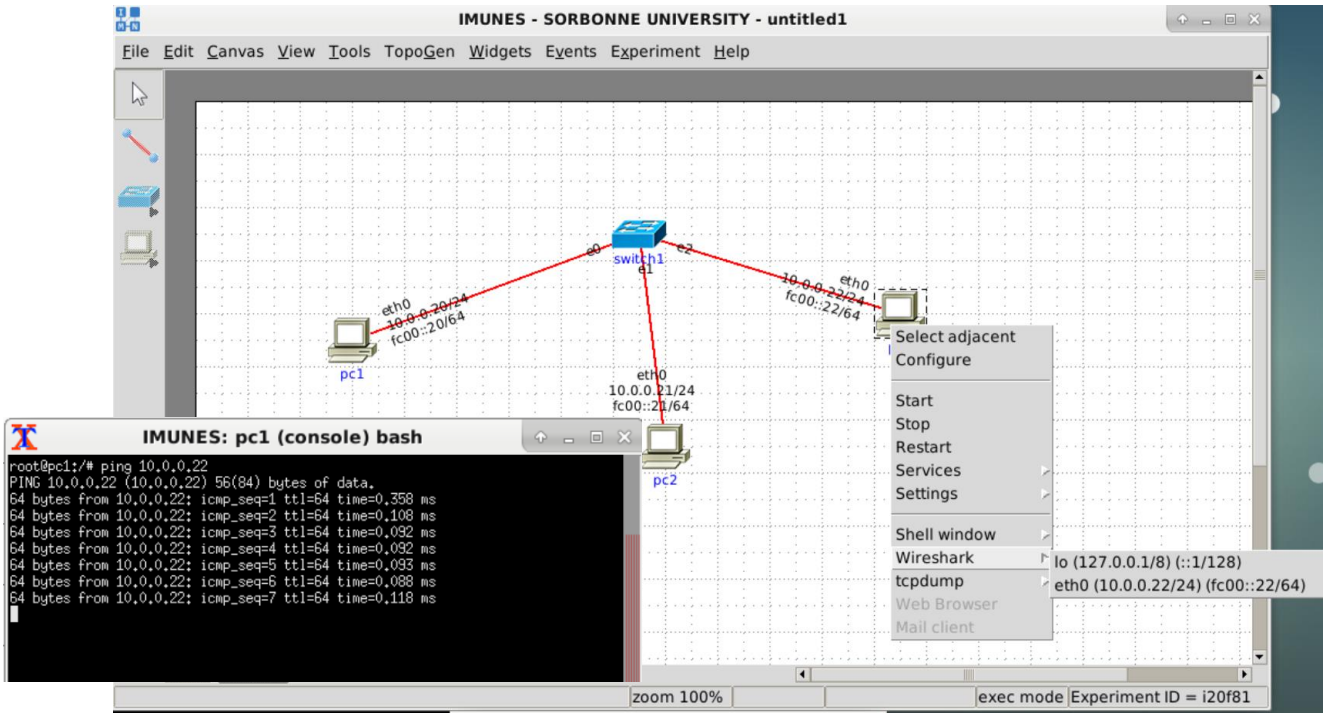
Create the following scenario using the list of toolbox button-icons on the left side of the GUI. Click on the 'Lan Switch' icon then on the canvas (the work surface) to add a Switch, on the 'PC' icon to add the hosts and use the 'Create Link' icon to connect the machines to the Switch .



Launch network emulation by choosing 'Experiment Execute' from the main menu. This operation may take several seconds.

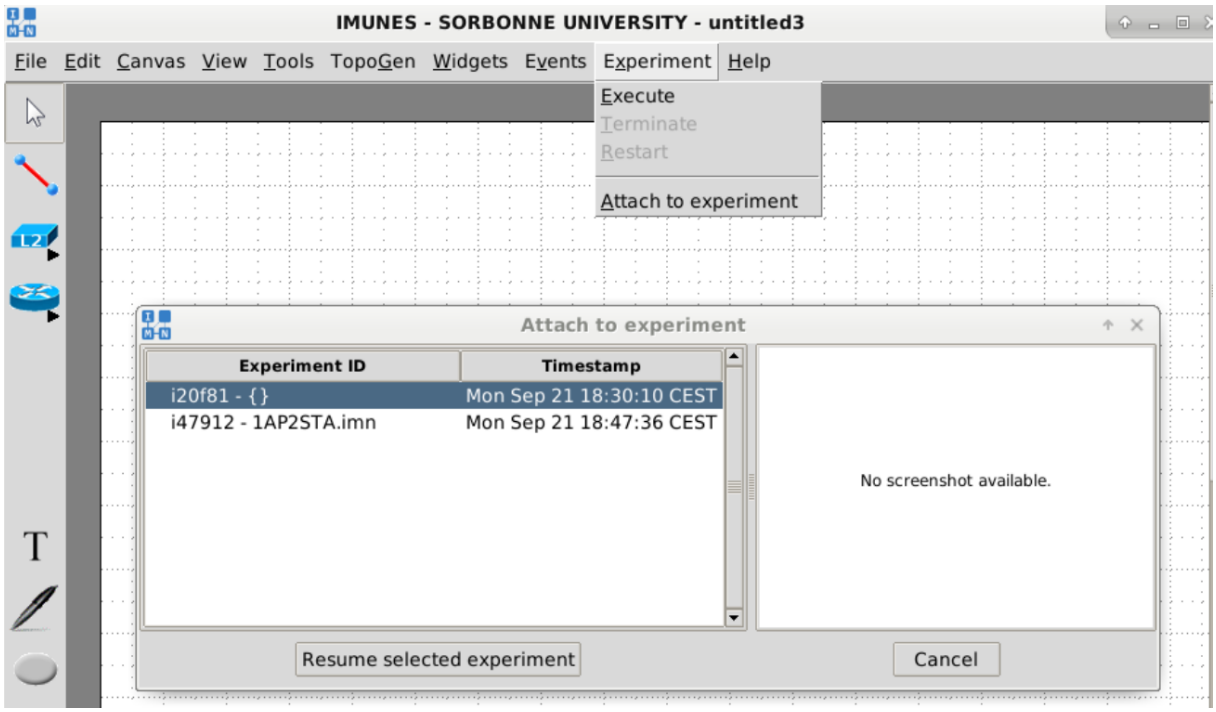
Then, inside the work surface, double click with the left mouse button on the pc1 machine icon in order to obtain a classic terminal giving you access to the pc1 machine. Alternatively, you can press and hold the right mouse button until a menu appears. Choose 'Bash' to get the terminal.

`ping the IP address of machine pc3 from machine pc1, then, without stopping the ping, run a wireshark on pc3's eth0 interface from the menu that appears by clicking the right mouse button. wireshark displays the data exchanged between pc1 and pc3 and which is generated by the ping command. We will study wireshark software later, and of course IP addressing.`



Before ending the experiment, you can save your configurations in a file named for example testsw (automatic .imn extension) with 'Save' from the 'File' menu or by simultaneously pressing Ctrl s. Do not hesitate to use it frequently in order to avoid losing your work in the event of a crash. 'Save As' or Ctrl e, saves an existing file under another name and location. Backup saves IP addresses, VLANs and routing tables in the .imn file, while instantly updating the graphical interface. We will see later how to save the configurations of CISCO routers.

If you quit the IMUNES emulator by pressing Quit or if you close the IMUNES window. The experiment is not over and it is still ongoing. To review the network topology again and access its nodes again, launch IMUNES and choose Experiment->Attach to experiment. When the current experiments window appears, select the experiment you want to resume and press Resume selected experiment. (See figure below)



Test by creating several experiments in parallel.

To end the experiment, choose Experiment□Terminate. This operation may take several tens of seconds.

Finally, you test the drawing/text tools:

Draw on the previous topology of the emulator with the 'Add a FreeForm' tool a line of green color and width 4 connecting pc1 and pc3. With the 'Add a Textbox' tool, insert the text "Data" above the green line.