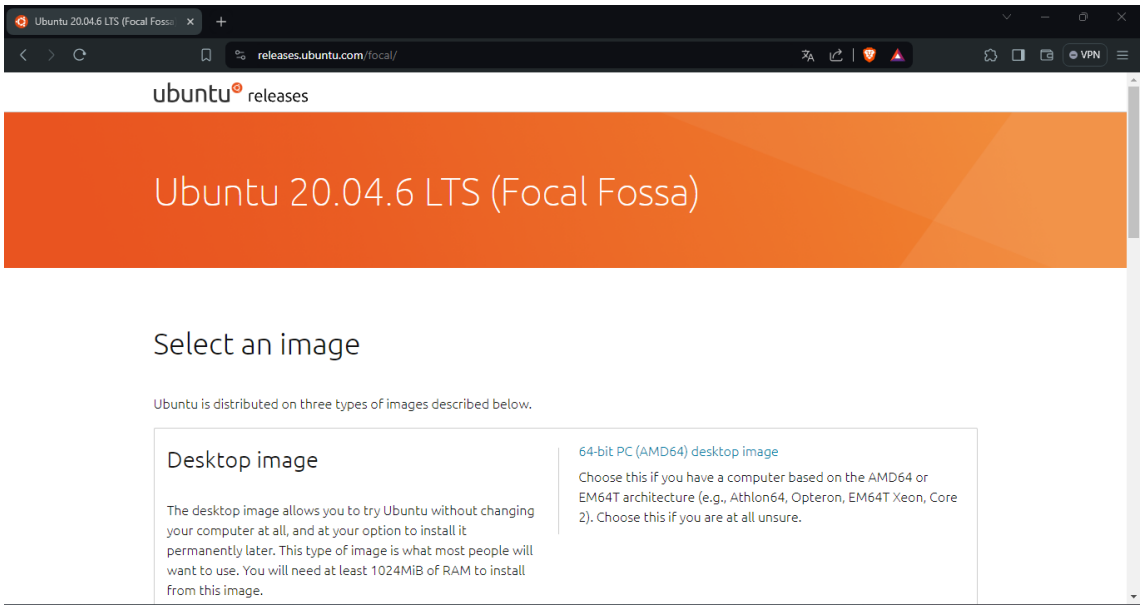
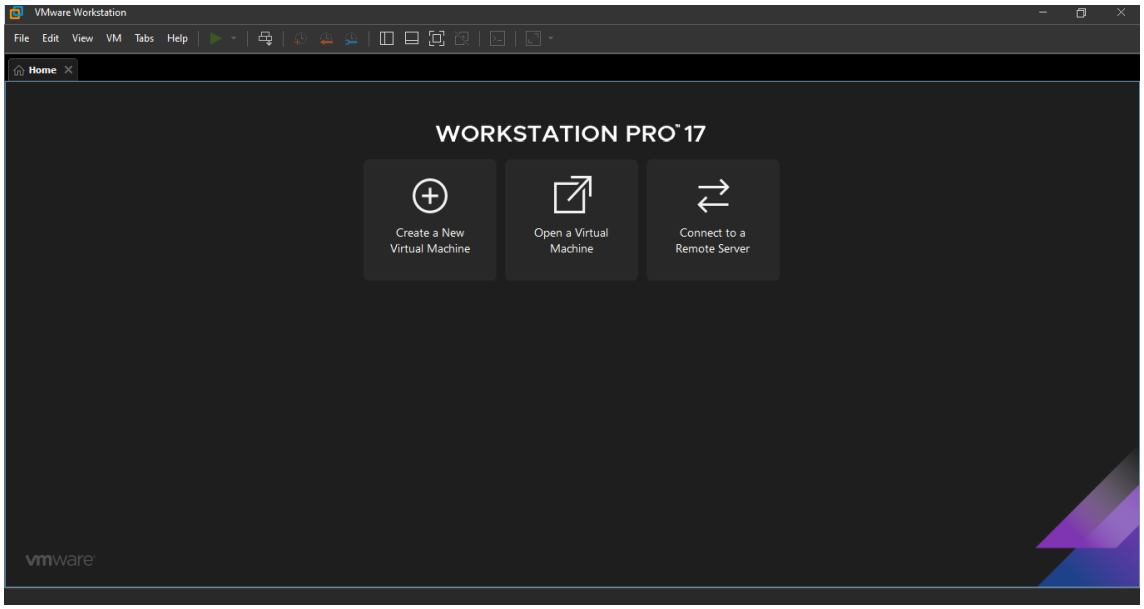
USER GUIDE

This process requires the use of Ubuntu 20.04.6 that works on a virtual machine (VMWARE), Ubuntu can be obtained from the following link of the official page (https://releases.ubuntu.com/focal/), as well as VMWARE can be obtained from its official page from the following link (https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html.html).

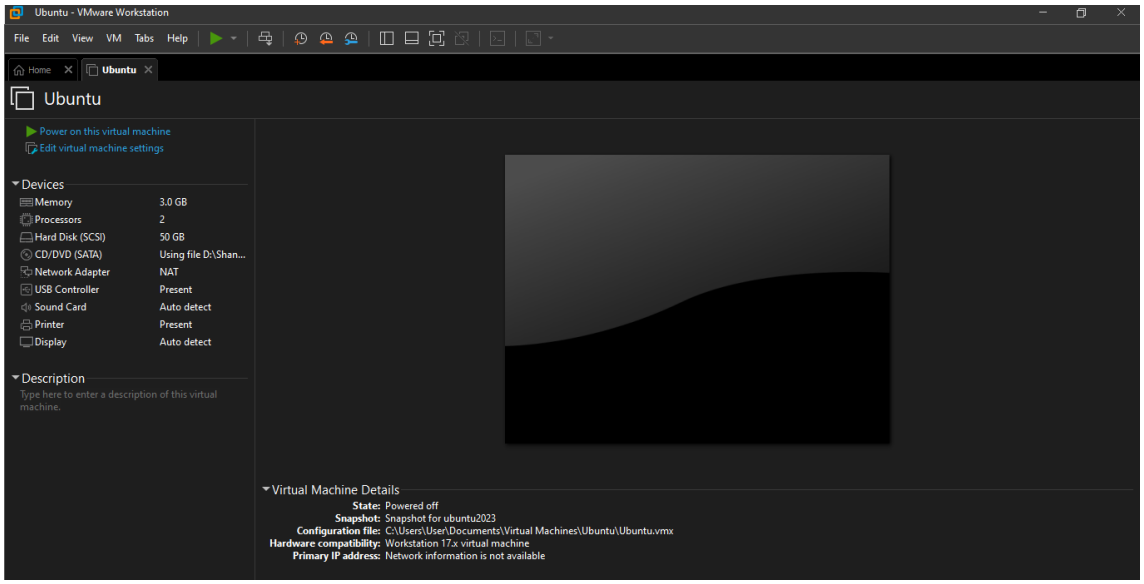


Ubuntu .iso file.



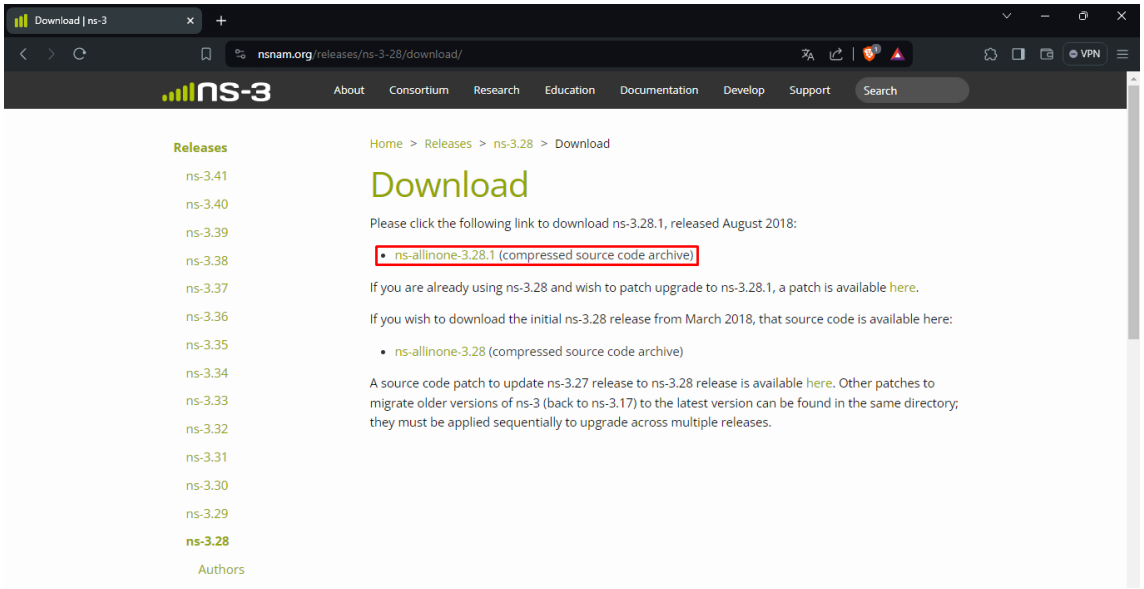
VMWARE workspace view.

After virtualizing Ubuntu on VMWARE, version 3.28.1 of the NS-3 software is required, which is available on the official nsnam.org website, at the following link (https://www.nsnam.org/releases/ns-3-28/download/).



View of virtual machine creation.

The initial NS-3 documentation can be found at the following page (https://www.nsnam.org/docs/release/3.28/tutorial/singlehtml/index.html), which covers aspects such as: supported platforms, prerequisites, ways to obtain NS-3, ways to build NS-3 and implementation.



.tar.bz2 file of Network Simulator 3.

Download and installation of the P1906 module

From the following link (https://github.com/ArmasSantiago?tab=repositories), there are 3 repositories that determine the 3 individually configured scenarios. Each repository consists of the README file which briefly shows how to run the specific scenario, the p1906.zip file contains the 9 classes and 21 files configured for the simulation in NS-3 and also provides documents representing the best practices of the standard.

Once the files have been downloaded from GitHub, where you can obtain the scenario-main.zip file for each of the scenarios. We proceed to open a terminal, and we usually go to the Downloads folder and then unzip the file scenario1-main.zip and the file p1906.zip.

```
santiago@unach:~$ cd Downloads/
santiago@unach:~/Downloads$ unzip escenario1-main.zip
Archive:  escenario1-main.zip
e70a5dbe9098bdd6ef0778abc107fbc7bd83c7e4
   creating: escenario1-main/
  inflating: escenario1-main/README
  inflating: escenario1-main/ieee-1906.1-recommended-practice-for-nanoscale-and-molecular-communicat.pdf
  inflating: escenario1-main/p1906.1.1-2020.pdf
  inflating: escenario1-main/p1906.zip
santiago@unach:~/Downloads$ cd escenario1-main/
santiago@unach:~/Downloads/escenario1-main$ unzip p1906.zip
Archive:  p1906.zip
   creating: p1906/
   creating: p1906/examples/
  inflating: p1906/examples/_RUN_EM_CHANNEL_CAPACITY_.sh
  inflating: p1906/examples/_RUN_MOL_Ancho_Pulso_Minimo_.sh
  inflating: p1906/examples/_RUN_MOL_Capacidad_Canal_.sh
  inflating: p1906/examples/_RUN_MOL_CHANNEL_CAPACITY_.sh
  inflating: p1906/examples/_RUN_MOL_Pulso_Retardo_.sh
  inflating: p1906/examples/em-example.cc
```

View of unzipped files.

Enter the /scenario1-main folder to copy the /p1906 folder to the /src folder of NS-3 using the following command:

```
santiago@unach:~/Downloads$ cp -r p1906/ ~/Documents/proyinvestigacion/ns-allinone-3.28.1/
ns-3.28.1/src/
```

Go to the folder containing the software, specifically /ns-3.28.1 in order to configure the simulator using the following command:

```
santiago@unach:~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1$ /usr/bin/python2
 ./waf configure
```

Finally, to build the project, without errors and ready to be included in the NS-3 modules, the following command is applied:

```
santiago@unach:~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1$ /usr/bin/python2
 ./waf
```

The way to check if this module is included and ready to be used is through the integration itself given in the console response as follows:

```
Modules built:
antenna                  aodv                     applications
bridge                   buildings                config-store
core                     csma                     csma-layout
dsdv                     dsr                      energy
fd-net-device            flow-monitor             internet
internet-apps            lr-wpan                  lte
mesh                     mobility                 mpi
netanim (no Python)      network                  nix-vector-routing
olsr                     p1906                    point-to-point
point-to-point-layout    propagation              sixlowpan
spectrum                 stats                    tap-bridge
test (no Python)         topology-read            traffic-control
uan                      virtual-net-device       wave
wifi                     wimax

Modules not built (see ns-3 tutorial for explanation):
brite                    click                    openflow
visualizer
```

Construction of p1906 module in NS-3 software.

After successful construction of the module, the mol-example.cc program is copied and placed in the /examples folder of the module and directed to the /scratch folder of the simulator.

```
santiago@unach:~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1/src/p1906$ cp exa
mples/mol-example.cc ~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1/scratch/
```

The following command executes the program with its default parameters.

```
santiago@unach:~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1$ /usr/bin/python2
 ./waf --run scratch/mol-example
```

Simulator execution

The result of the execution of the simulator of Scenario 1 is shown by console. The creation of the components that integrate the configured modules, the creation of objects and the call to methods in NS-3 are represented by sections. During the execution of the simulator the lines show, for example; creation of the object named 'P1906Medium' with a memory address '0x55a3db4433f0' and an associated text message. These objects call the methods associated with the built-in components to perform the program functions.

```
P1906Medium:P1906Medium(0x55bfe664b3f0, "===Sección-Creación componente Medio/Medium")
P1906Motion:P1906Motion(0x55bfe664b100)
P1906MOLMotion:P1906MOLMotion(0x55bfe664b100, "===Sección-Creación del componente Movimiento/Motion===")
P1906MOLMotion:SetDiffusionCoefficient(0x55bfe664b100, 1e-12)
P1906Medium:SetP1906Motion(0x55bfe664b3f0)
P Help etDevice:P1906NetDevice(0x55bfe666eda0)
P1906CommunicationInterface:P1906CommunicationInterface(0x55bfe664adf0)
P1906TransmitterCommunicationInterface:P1906TransmitterCommunicationInterface(0x55bfe664ae80)
P1906TransmitterCommunicationInterface:SetP1906NetDevice(0x55bfe664ae80)
P1906ReceiverCommunicationInterface:P1906ReceiverCommunicationInterface(0x55bfe664b240)
P1906ReceiverCommunicationInterface:SetP1906NetDevice(0x55bfe664b240)
P1906TransmitterCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664ae80)
P1906ReceiverCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664b240)
P1906MOLCommunicationInterface:P1906MOLCommunicationInterface(0x55bfe664adf0, "===Sección-Creación de Dispositivo de Red===")
P1906CommunicationInterface:SetP1906NetDevice(0x55bfe664adf0)
P1906CommunicationInterface:SetP1906Medium(0x55bfe664adf0)
P1906TransmitterCommunicationInterface:SetP1906Medium(0x55bfe664ae80)
P1906ReceiverCommunicationInterface:SetP1906Medium(0x55bfe664b240)
P1906TransmitterCommunicationInterface:P1906TransmitterCommunicationInterface(0x55bfe664dd30)
P1906TransmitterCommunicationInterface:SetP1906NetDevice(0x55bfe664dd30)
P1906MOLTransmitterCommunicationInterface:P1906MOLTransmitterCommunicationInterface(0x55bfe664dd30)
P1906ReceiverCommunicationInterface:P1906ReceiverCommunicationInterface(0x55bfe664ddf0)
P1906ReceiverCommunicationInterface:SetP1906NetDevice(0x55bfe664ddf0)
P1906MOLReceiverCommunicationInterface:P1906MOLReceiverCommunicationInterface(0x55bfe664ddf0)
P1906CommunicationInterface:SetP1906TransmitterCommunicationInterface(0x55bfe664adf0)
P1906TransmitterCommunicationInterface:~P1906TransmitterCommunicationInterface(0x55bfe664ae80)
P1906TransmitterCommunicationInterface:SetP1906NetDevice(0x55bfe664ae80)
P1906CommunicationInterface:SetP1906ReceiverCommunicationInterface(0x55bfe664adf0)
P1906ReceiverCommunicationInterface:~P1906ReceiverCommunicationInterface(0x55bfe664b240)
```

Console view of the execution of the IEEE1906.1 standard using the mol-example.cc file, representing the creation of the Motion component and the first Network device.

```
P1906ReceiverCommunicationInterface:SetP1906NetDevice(0x55bfe664b240)
P1906TransmitterCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664dd30)
P1906ReceiverCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664ddf0)
P1906Specificity:P1906Specificity(0x55bfe6634420)
P1906MOLSpecificity:P1906MOLSpecificity(0x55bfe6634420, "===Componente de Especificidad/Specificity===")
P1906Field:P1906Field(0x55bfe65ec520)
P1906Field:P1906Field(0x55bfe65ec520)
P1906MOLField:P1906MOLField(0x55bfe65ec520)
P1906MOLField:P1906MOLField(0x55bfe65ec520, "===Sección-Creación Componente de Campo/Field===")
P1906Perturbation:P1906Perturbation(0x55bfe6643800)
P1906MOLPerturbation:P1906MOLPerturbation(0x55bfe6643800, "===Sección-Creación componente Perturbación/Perturbation===")
P1906MOLPerturbation:SetPulseInterval(0x55bfe6643800, +2000000000.0ns)
P1906MOLPerturbation:SetMolecules(0x55bfe6643800, 10000)
P1906MOLSpecificity:SetDiffusionCoefficient(0x55bfe6634420, 1e-12)
P1906NetDevice:P1906NetDevice(0x55bfe664da30)
P1906CommunicationInterface:P1906CommunicationInterface(0x55bfe664e140)
P1906TransmitterCommunicationInterface:P1906TransmitterCommunicationInterface(0x55bfe664e1b0)
P1906TransmitterCommunicationInterface:SetP1906NetDevice(0x55bfe664e1b0)
P1906ReceiverCommunicationInterface:P1906ReceiverCommunicationInterface(0x55bfe664e220)
P1906ReceiverCommunicationInterface:SetP1906NetDevice(0x55bfe664e220)
P1906TransmitterCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664e1b0)
P1906ReceiverCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664e220)
```

Console view of the IEEE1906.1 standard execution using the mol-example.cc file, representing the creation of the Specificity, Field and Perturbation component.

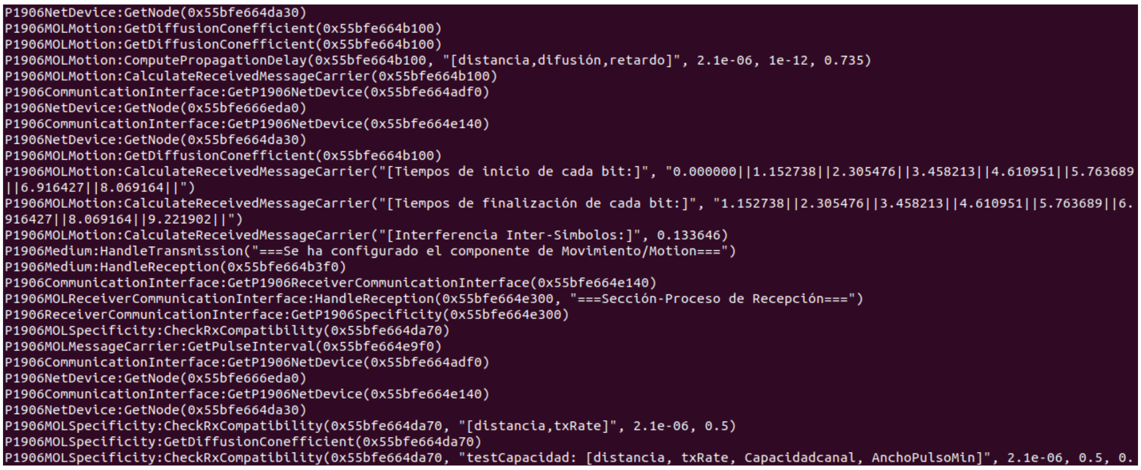
```
P1906MOLTransmitterCommunicationInterface:P1906MOLTransmitterCommunicationInterface(0x55bfe664e290)
P1906ReceiverCommunicationInterface:P1906ReceiverCommunicationInterface(0x55bfe664e300)
P1906ReceiverCommunicationInterface:SetP1906NetDevice(0x55bfe664e300)
P1906MOLReceiverCommunicationInterface:P1906MOLReceiverCommunicationInterface(0x55bfe664e300)
P1906CommunicationInterface:SetP1906TransmitterCommunicationInterface(0x55bfe664e140)
P1906TransmitterCommunicationInterface:~P1906TransmitterCommunicationInterface(0x55bfe664e1b0)
P1906TransmitterCommunicationInterface:SetP1906NetDevice(0x55bfe664e1b0)
P1906CommunicationInterface:SetP1906ReceiverCommunicationInterface(0x55bfe664e140)
P1906ReceiverCommunicationInterface:~P1906ReceiverCommunicationInterface(0x55bfe664e220)
P1906ReceiverCommunicationInterface:SetP1906NetDevice(0x55bfe664e220)
P1906TransmitterCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664e290)
P1906ReceiverCommunicationInterface:SetP1906CommunicationInterface(0x55bfe664e300)
P1906Specificity:P1906Specificity(0x55bfe664da70)
P1906MOLSpecificity:P1906MOLSpecificity(0x55bfe664da70, "===Componente de Especificidad/Specificity===")
P1906Field:P1906Field(0x55bfe65ea850)
P1906Field:P1906Field(0x55bfe65ea850)
P1906MOLField:P1906MOLField(0x55bfe65ea850)
P1906MOLField:P1906MOLField(0x55bfe65ea850, "===Sección-Creación Componente de Campo/Field===")
P1906Perturbation:P1906Perturbation(0x55bfe6695590)
P1906MOLPerturbation:P1906MOLPerturbation(0x55bfe6695590, "===Sección-Creación componente Perturbación/Perturbation===")
P1906MOLPerturbation:SetPulseInterval(0x55bfe6695590, +2000000000.0ns)
P1906MOLPerturbation:SetMolecules(0x55bfe6695590, 10000)
P1906MOLSpecificity:SetDiffusionCoefficient(0x55bfe664da70, 1e-12)
```

Console view of the IEEE1906.1 standard execution using the mol-example.cc file, representing the creation of the second Network device, creation of the Specificity, Field and Perturbation component.

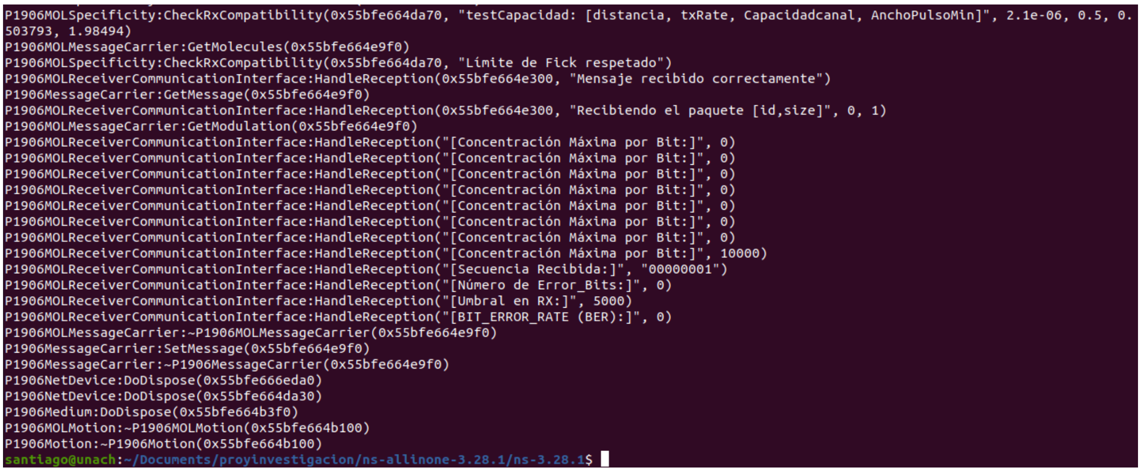
```
P1906TransmitterCommunicationInterface:SetP1906Medium(0x55bfe664dd30)
P1906ReceiverCommunicationInterface:SetP1906Medium(0x55bfe664ddf0)
P1906CommunicationInterface:GetP1906TransmitterCommunicationInterface(0x55bfe664adf0)
P1906TransmitterCommunicationInterface:SetP1906Perturbation(0x55bfe664dd30)
P1906CommunicationInterface:GetP1906TransmitterCommunicationInterface(0x55bfe664adf0)
P1906TransmitterCommunicationInterface:SetP1906Field(0x55bfe664dd30)
P1906CommunicationInterface:GetP1906ReceiverCommunicationInterface(0x55bfe664adf0)
P1906ReceiverCommunicationInterface:SetP1906Specificity(0x55bfe664ddf0)
P1906Specificity:SetP1906CommunicationInterface(0x55bfe6634420)
P1906Medium:AddP1906CommunicationInterface(0x55bfe664b3f0)
P1906NetDevice:SetNode(0x55bfe66b7680)
P1906NetDevice:SetNode(0x55bfe66b7680)
P1906NetDevice:SetIfIndex(0)
P1906NetDevice:SetReceiveCallback(0x7fffd37beff0)
P1906CommunicationInterface:SetP1906NetDevice(0x55bfe664e140)
P1906CommunicationInterface:SetP1906Medium(0x55bfe664e140)
P1906TransmitterCommunicationInterface:SetP1906Medium(0x55bfe664e290)
P1906ReceiverCommunicationInterface:SetP1906Medium(0x55bfe664e300)
P1906CommunicationInterface:GetP1906TransmitterCommunicationInterface(0x55bfe664e140)
P1906TransmitterCommunicationInterface:SetP1906Perturbation(0x55bfe664e290)
P1906CommunicationInterface:GetP1906TransmitterCommunicationInterface(0x55bfe664e140)
P1906TransmitterCommunicationInterface:SetP1906Field(0x55bfe664e290)
P1906CommunicationInterface:GetP1906ReceiverCommunicationInterface(0x55bfe664e140)
P1906ReceiverCommunicationInterface:SetP1906Specificity(0x55bfe664e300)
```

Console view of the IEEE1906.1 standard execution using the mol-example.cc file, representing the configuration and connection of components and interfaces.

Console view of the IEEE1906.1 standard execution using the mol-example.cc file, depicting packet transmission, transmission on the communication interface, message establishment on the carrier, generation and configuration of the modulation technique describing the program logic.



Console view of the IEEE1906.1 standard execution using the mol-example.cc file, representing the calculation of the pulse delay, calculations with the message received on the associated node, analysis of the calculation related to the inter-symbol interference.



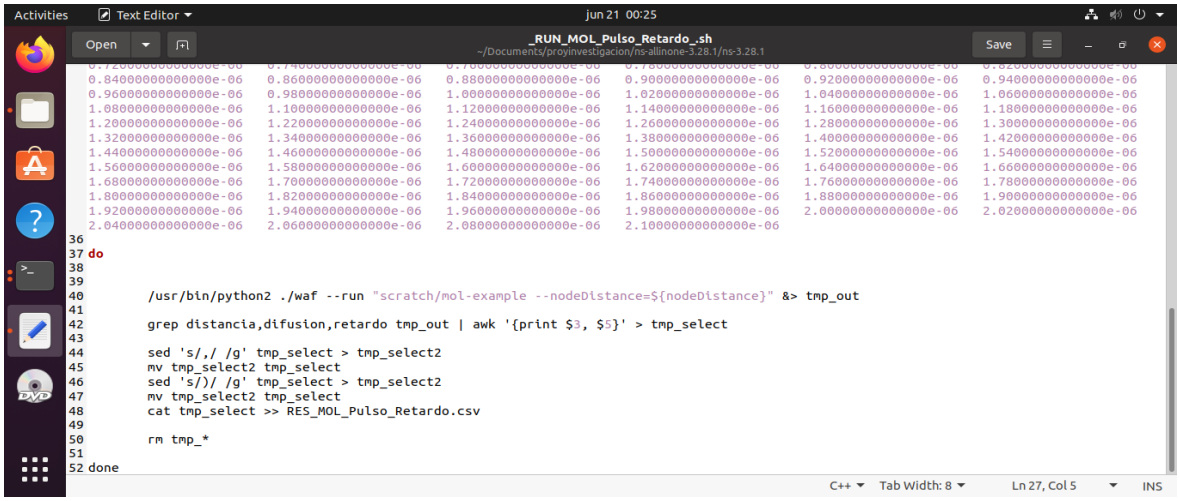
Console view of the IEEE1906.1 standard execution using the mol-example.cc file, representing the reception and processing of the compatibility in the receiver, obtains the modulation associated to the carrier and starts the demodulation process and the handling of the information.

Obtaining results

The method used to obtain the simulation results of scenario 1 is presented. .sh files are executed, which facilitate the execution of multiple simulations and the capture of the desired values, such as the delay pulse.

View of the code fragment of the _RUN_MOL_Pulso_Retardo.sh file for obtaining result.

The _RUN_MOL_Pulso_Retardo.sh file can be obtained in the examples folder of the p1906.zip file. The figure shows the first code fragment, where the program modifies the behavior of the terminal when executing the script with the help of the 'set-x' and 'set -e' commands that are helpful in tracking and understanding the flow of execution. The 'for' loop iterates through the list of values depending on the distance by assigning the current value of the variable 'nodeDistance' from the mol-example.cc file.



View of the code fragment of the _RUN_MOL_Pulso_Retardo.sh file for obtaining result.

The figure represents the continuation of the code, which executes the program and saves the output in a temporary file 'tmp_out'. Then, a series of manipulations are performed on the contents of the file to extract the specific information and save it to 'tmp_select'. Finally, the results are added to the file '_RUN_MOL_Pulso_Retardo.csv' and the temporary files are deleted. To run the program from the terminal in Ubuntu, execute permissions must be given using the following command:

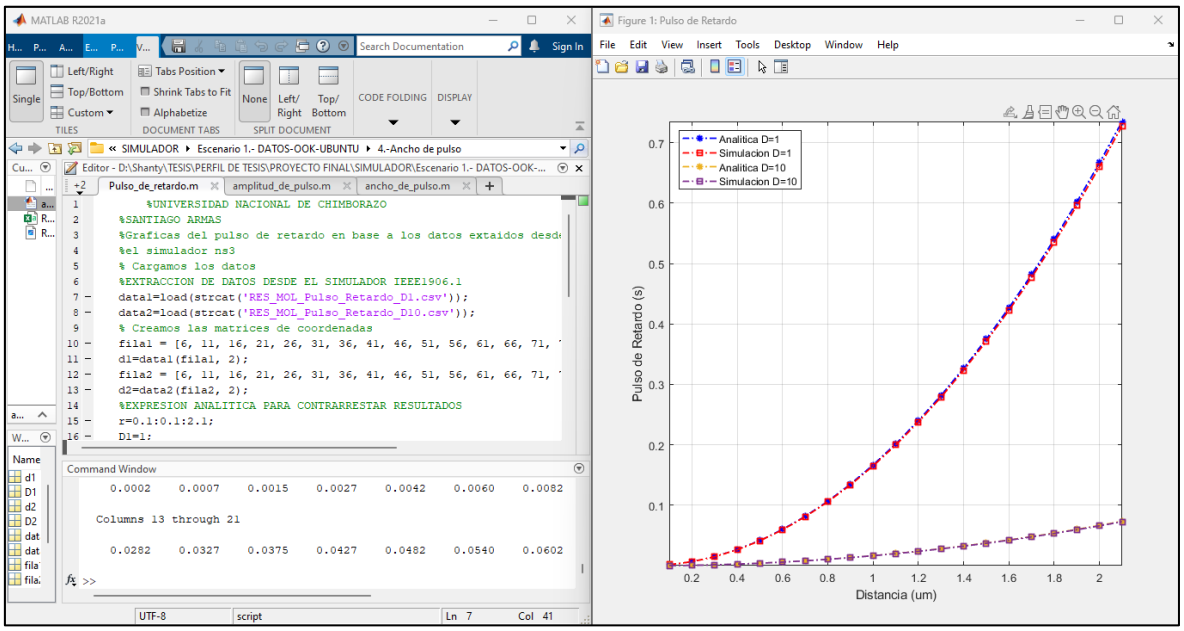
```
santiago@unach:~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1$ chmod +x _RUN_MOL_Pulso_Retardo_.sh
```

And start the execution in the NS-3 environment using the following command:

```
santiago@unach:~/Documents/proyinvestigacion/ns-allinone-3.28.1/ns-3.28.1$ ./_RUN_MOL_Pulso_Retardo_.sh
```

Data export to MatLab

To generate the graphs, data are loaded from .csv files in user-defined variables in the MatLab environment. Then, the analytical calculation is performed to represent the resulting curves and to be able to perform the comparison of analytical results with the simulation data.



View of MatLab code for obtaining delay pulse plots.