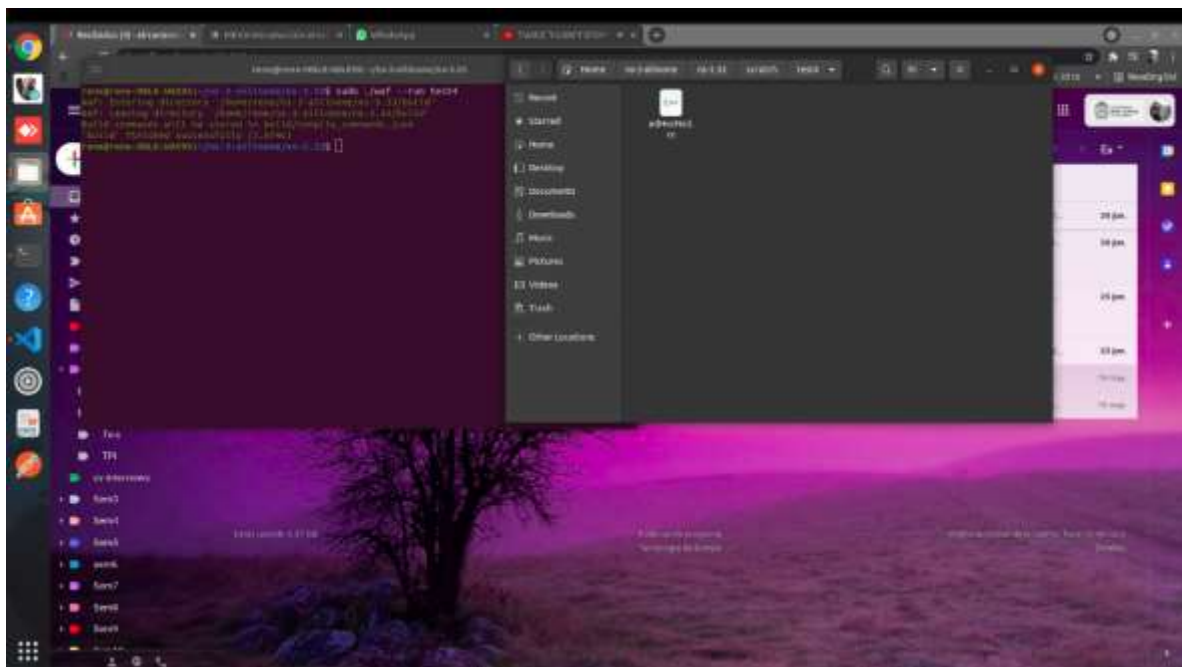


# Análisis y Resultados

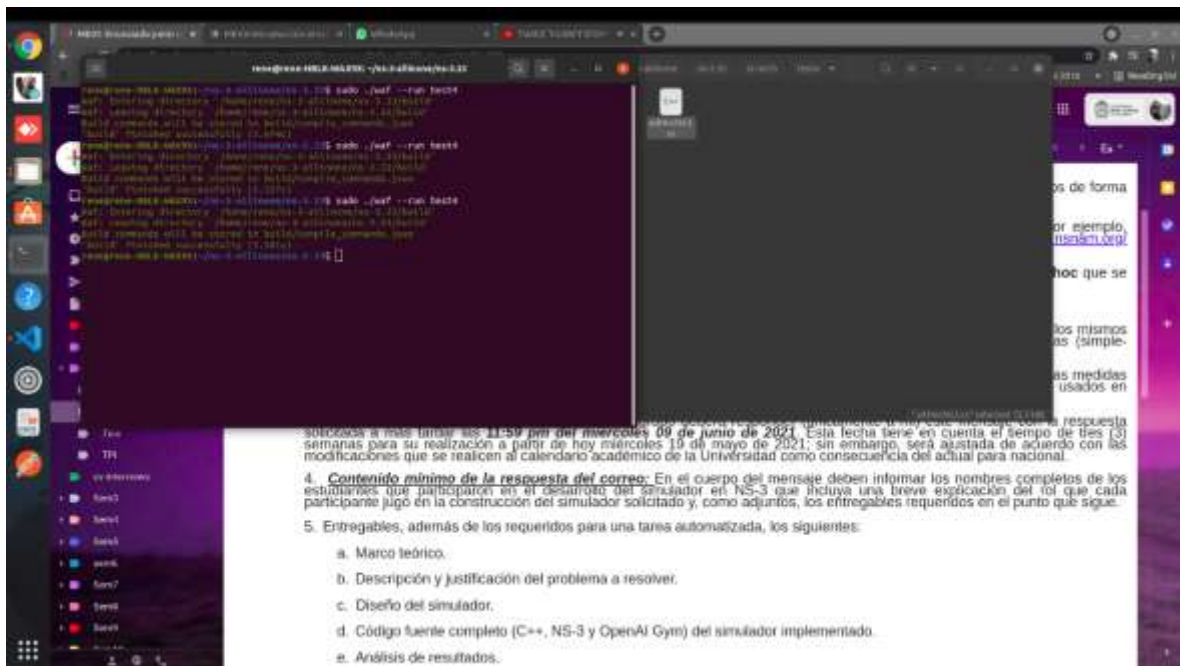
## Red Ad Hoc NS3 sin OpenAI Gym sin LoRaWan

Para los análisis y resultados se dispuso de 3 escenarios:

El primero es el código de NS3 escrito en C++ donde se crea la red ad hoc de NS3 sin el uso de OpenAI Gym, ni del modulo LoRaWan, esta red consta de 20 nodos con trafico de tipo tcp e internet, los tiempos de ejecución del archivo se evidencia a continuación en dos fotos la primera que nos muestra el tiempo usando YansWifiPhyHelper, para tener más conocimiento sobre los tipos de trafico usado se recomienda leer los comentarios en el fichero, durante el desarrollo del taller se encontró que los tiempos de esta red variaban si se anexaban más nodos, se determino que el tiempo de espera máximo para los nodos de prueba era de 15 segundos. La distribución de esta red se hizo de dos formas, la primera con el uso de grid(forma de rejilla) que ubicaba los nodos de la forma más cuadrada posible con la ayuda de MobilityHelper.



La segunda, donde ubicaba los nodos de forma lineal con valores entre 0 y 1000 unidades de distancia( ns3 simula que estas unidades de distancia se dan en metros, pero no es exacto).

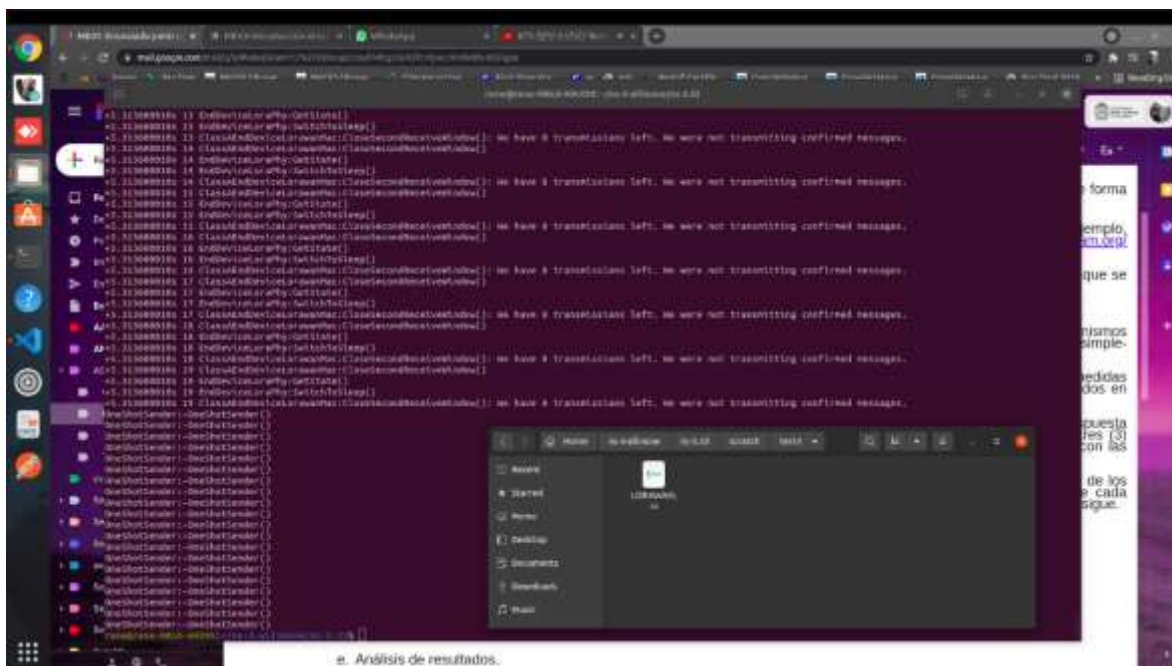


## Red Ad Hoc NS3 sin OpenAI Gym con LoRaWan

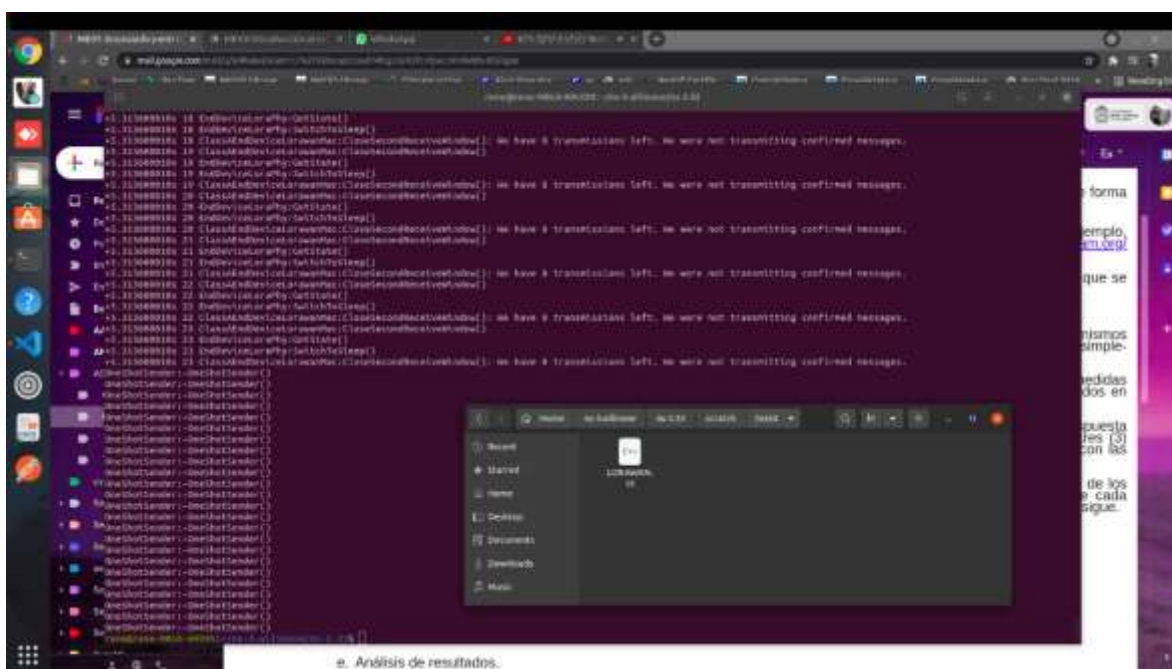
Para la implementación de la red ad hoc con LoRaWan se usó el código de complete-network-example y simple-network-example, en el cual se crea un canal con un delay y con una pérdida de 3,85 una referencia de 7,4, también se definió un tiempo de 3 segundos para el envío de paquetes, para este ejemplo no se asignaron direcciones ip de tipo v4 como en el ejercicio anterior, si no , que se usó un generador de direcciones de LoRaWan.

Para este ejemplo se usaron las misma distribuciones y tráficos del código anterior, se hicieron 2 pruebas una de 20 nodos y otra de 24.

Ejemplo de 20 nodos:



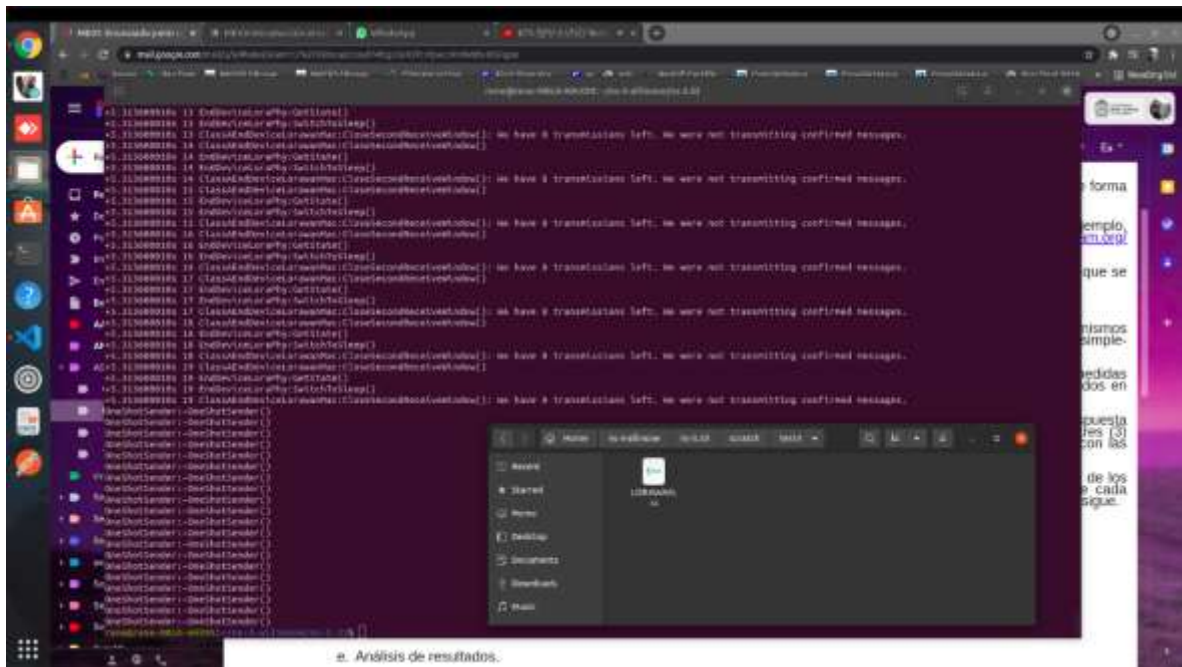
Ejemplo con 24 nodos:



## Red Ad Hoc NS3 con OpenAI Gym con LoRaWan

Para este escenario se tienen 2 archivos el primero es la parte de ns3 con LoRaWan donde se definen características y parámetros como en los ejemplos anteriores, hacemos énfasis en el numero de nodos, en los parámetros de envío de paquetes y particularmente en la distancia y los rewards, Lo que pudimos observar de la distancia y cómo se comporta esta con la cantidad de nodos es que entre más nodos o mas distancia se tiene a tener más perdida de paquetes.

El proceso con OpenAI Gym se descompone en 3 etapas, la primera de ellas es la identificación del objeto a estudiar (para este caso red ad hoc, se considera objeto dado que pueden ser redes eléctricas entre otras), la segunda es la parte de recompensas en esta se asigna el valor de 1 hasta que se llegue al número máximo de valor de recompensa del agente, este valor se define en la etapa 1, Y por ultimo tenemos la etapa de cierre donde si el agente ve que el valor de recompensa es el máximo da como terminado el proceso.



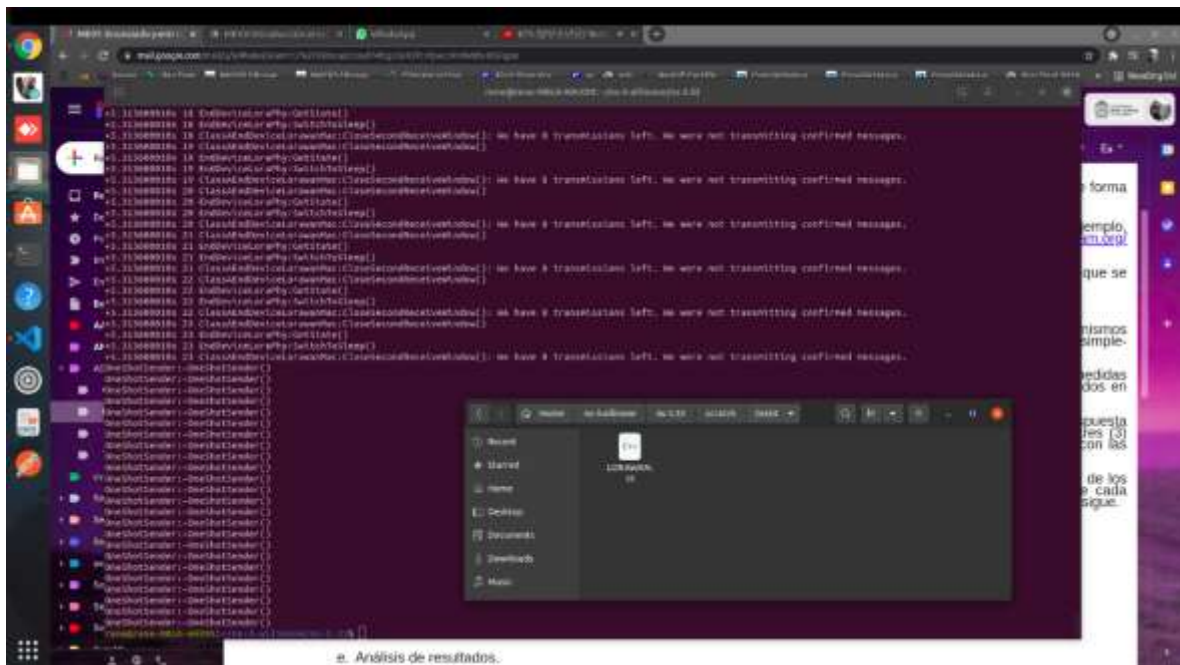
En la imagen anterior vemos que se tienen 20 nodos donde la distancia entre cada uno no es mayor a 20 metros(Unidades de ns3).

Para el segundo ejemplo de esta red se utilizaron 24 nodos, en las imágenes aparecen 19 y 23, esto se debe a que el nodo inicial toma el valor 0, desde la perspectiva de ingeniería seria normal ver que entre más distancia más porcentaje



de perdida habrá dado que por eventualidades en el ambiente las condiciones climáticas, entre otros factores son los generan los fallos.

Otro factor que pudimos ver es la cantidad de nodos, esto se debe a que una cantidad de nodos muy grande en un espacio muy pequeño o considerablemente reducido genera que la red, tenga más tráfico del que puede llegar a soportar, en tiempos de espera y envío, por lo que se considera que la topología de la red influye en los tiempos y posibles fallos o pérdidas de paquetes.



## Referencias

Las referencias mencionadas a continuación fueron parte de apoyo extra al marco lógico definido anteriormente, hacemos mención especial porque nos pareció interesante las diferentes formas de aplicación de ns3 y las librerías y módulos como OpenAI Gym y LoRaWan

1. <https://github.com/signetlabdei/lorawan>
2. <https://blog.uclm.es/felixvillanueva/>

