

Instituto Tecnológico de Cancún.
Proyecto: Sistema de comunicación.

Kanxoc Ek Felix Gerardo.

Octubre 2020.

Ismael Jiménez Sánchez.

Fundamentos de telecomunicaciones.

Índice.

Contenido

Antecedentes	4
Vagrant	4
BOXES	4
VirtualBox	4
Máquinas virtuales	4
GNS3	5
puTTY	5
Python.	5
Wireshark.	5
Reporte de conclusiones.....	6
Fases	6
Conclusión del proyecto.....	9

Introducción.

En esta práctica se lleva a cabo la ejecución de un sistema de comunicación por medio de máquinas virtuales, usando distintos programas para poder llevar a cabo la realización del mismo. Así como la aplicación y entendimiento de los conceptos de la asignatura de Fundamentos de Telecomunicaciones; algunos de los programas a usar son Python, vagrant, virtualbox, entre otros. Todo con el fin de poder ver y entender la comunicación entre dos máquinas virtuales previamente desplegadas.

Antecedentes

Vagrant.

Vagrant es una herramienta que nos permite crear escenarios virtuales de una forma muy simple y replicable. Es un proyecto de código abierto escrito en Ruby y multiplataforma que nació en 2010 de la mano de Mitchell Hashimoto.

Vagrant es una herramienta muy útil e interesante, evitaremos tener cientos de bases de datos en nuestro ordenador, podremos tener aislado el entorno, y sobretodo evitaremos el "Pues en mi máquina si funciona".

BOXES.

Vagrant genera máquinas virtuales a través de imágenes ya creadas previamente por la comunidad. A estas máquinas se las denominan boxes.

Gracias a esto, podremos descargarnos entornos ya configurados para que el desarrollador solo se preocupe de empezar a programar y no de configurar el escenario, ahorrandonos así una gran cantidad de tiempo.

VirtualBox.

Es un software de virtualización para arquitecturas x86, creado originalmente por la empresa alemana innotek GmbH. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización.

Es un poderoso software de virtualización tanto para la empresa, como para el uso doméstico. Además, se caracteriza por ser la única solución profesional que está libremente disponible como software de código abierto bajo los términos de la Licencia Pública General de GNU.

Máquinas virtuales.

Una máquina virtual de sistema es aquella que emula a un ordenador completo.

GNS3.

es un simulador gráfico de red que permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellos. Con GNS3 los usuarios tendrán la posibilidad de poder escoger cada uno de los elementos que llegarán a formar parte de una red informática.

puTTY.

Es un emulador gratuito de terminal que soporta SSH y muchos otros protocolos. La mayoría de usuarios, especialmente los que trabajan sobre sistemas operativos Windows, lo encuentran muy útil a la hora de conectar a un servidor Unix o Linux a través de SSH

Python.

Python nació a principios de los 90 y, en un inicio, fue desarrollado como un hobby por Guido Van Rossum, un ingeniero holandés que trabajaba en ese momento en el CWI de Ámsterdam.

Es un lenguaje de programación versátil multiplataforma y multiparadigma que se destaca por su código legible y limpio. Python es ideal para trabajar con grandes volúmenes de datos ya que, al ser multiplataforma, favorece su extracción y procesamiento, por eso lo eligen las empresas de Big Data.

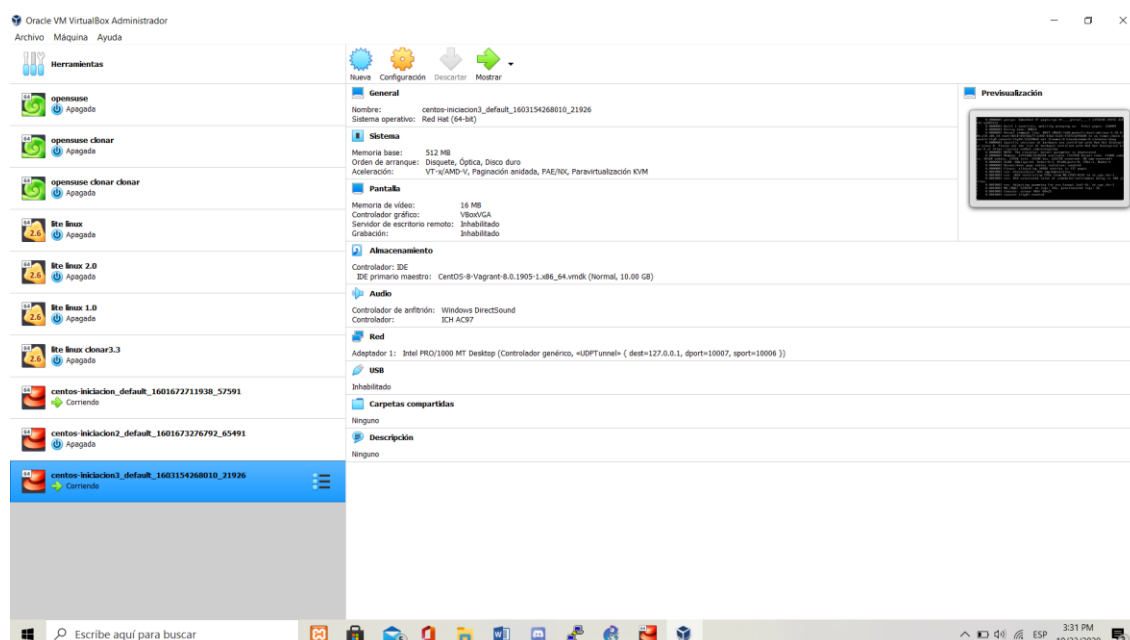
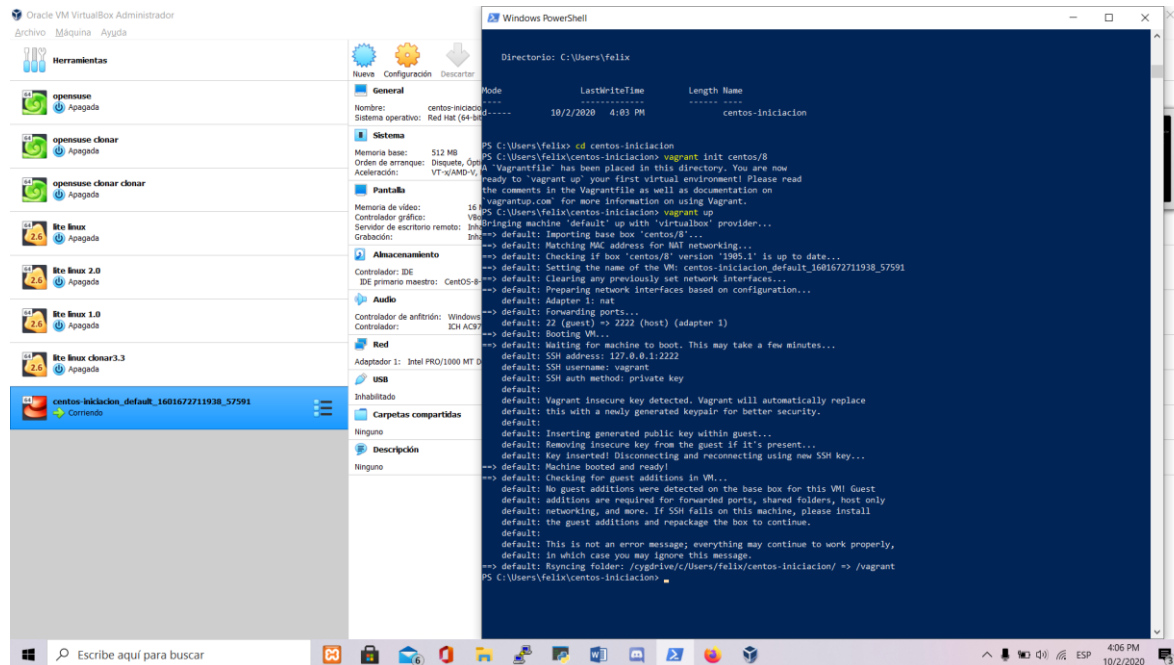
Wireshark.

Wireshark existe desde 1998. Fue ideado por Gerald Combs y, en un principio, llamado Ethereal. A lo largo de los años, ha recibido grandes cantidades de soporte y parches de la comunidad, y es ampliamente aceptado como el analizador de protocolo de red de facto disponible en la actualidad

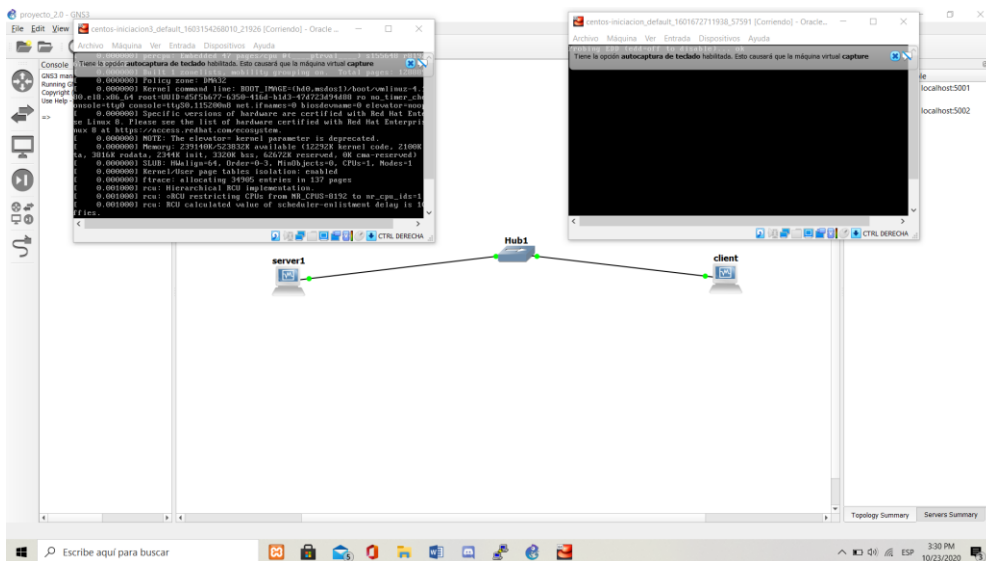
Reporte de conclusiones.

Fases

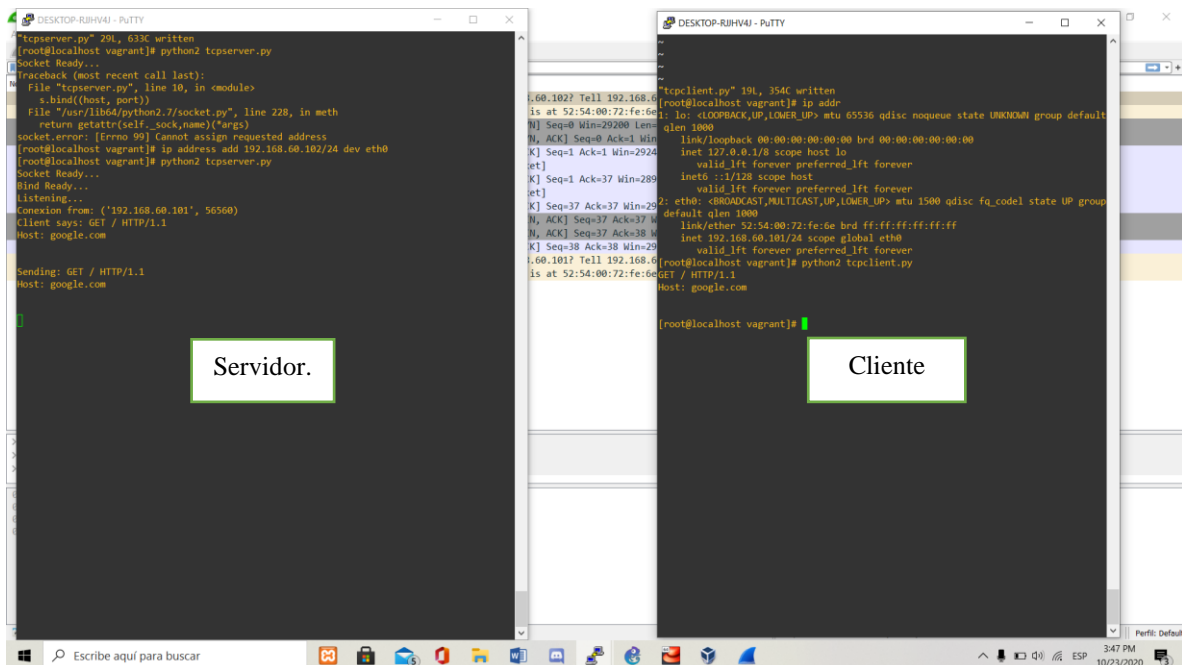
Fase1: utilizando vagrant se instalaron dos máquinas virtuales con CentOS 8, para esto se tuvo que utilizar la terminal powershell de Windows, utilizando comandos para poder instalarlos.



Fase 2: En esta fase, con ayuda de GNS3 se conectaron las dos máquinas virtuales de CentOS 8, con un hub ethernet para que estos puedan comunicarse.



Fase 3: En esta fase, con ayuda de los comandos necesarios modificamos la ip de las computadoras, posteriormente con uso de los scripts tcp client y tcp server, conectamos las dos computadoras virtuales, asignamos el puerto con el que vamos a comunicarnos, en el cual uno va a funcionar como el receptor (en este caso el server), y el otro va a funcionar como el emisor (en este caso el cliente), ya cuando estén conectados se va a mandar un mensaje al server y con esto termina el enlace y/o acaba la comunicación.



Fase 4: En esta ultima fase, se puede concluir que estamos utilizando los componentes de un sistema de comunicación.

- Emisor (En este caso el cliente).
- Receptor.(En este caso el server).
- Mensaje (datos enviados por el cliente-emisor).
- Codigo (En este caso los scripts de python).
- Protocolo (El protocolo viene siendo en este caso TCP).
- Medio de comunicación (La conexión de las computadoras con el hub).

Tambien podemos observar que el script hecho en python puede funcionar como comunicación full duplex, pero se tendria que configurar para que este pueda funcionar.

Tambien podemos notar que estamos utilizand el triple handshake, el cual funciona de la siguiente manera.

[SYN]: El cliente desea establecer una conexión en el servirdor, para esto manda un segmento con SYN, en la cual le informa al servidor que es probable que inicie la conversacion.

[SYN, ACK]: El servidor responde a la solicitud que hace el cliente con los segmentos SYN-ACK establecidos, en el cual [ACK] significa que recibio el segmento enviado, [SYN] significa el numero de secuencia con el cual se comunicara.

[SYN]: En esta parte el cliente reconoce la respuesta que hace el server y ambos empiezan a establecer la conexión con el cual iniciaran el traspaso y/o transferencia de los datos.

Capturing from - [client Ethernet0 to Hub1 Ethernet1]

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		RealtekU_72:fe:6e	Broadcast	ARP	60	Who has 192.168.60.102? Tell 192.168.60.101
2 0.000699		RealtekU_72:fe:6e	RealtekU_72:fe:6e	ARP	60	192.168.60.102 is at 52:54:00:72:fe:6e
3 0.000699		192.168.60.101	192.168.60.102	TCP	74	56560 → 500 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2981545625 TSecr=0 WS=64
4 0.001674		192.168.60.102	192.168.60.101	TCP	74	500 → 56560 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1086675032 TSecr=2981545625 WS=64
5 0.002681		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [ACK] Seq=1 Ack=37 Win=29248 Len=0 TSval=2981545639 TSecr=1086675032
6 0.002681		192.168.60.101	192.168.60.102	ISAKMP	102	[Malformed Packet]
7 0.003212		192.168.60.102	192.168.60.101	TCP	66	500 → 56560 [ACK] Seq=1 Ack=37 Win=28992 Len=0 TSval=1086675033 TSecr=2981545639
8 0.004168		192.168.60.102	192.168.60.101	ISAKMP	102	[Malformed Packet]
9 0.004168		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [ACK] Seq=37 Ack=37 Win=29248 Len=0 TSval=2981545642 TSecr=1086675034
10 0.007204		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [FIN, ACK] Seq=37 Ack=37 Win=29248 Len=0 TSval=2981545644 TSecr=1086675034
11 0.007204		192.168.60.102	192.168.60.101	TCP	66	500 → 56560 [FIN, ACK] Seq=37 Ack=38 Win=28992 Len=0 TSval=1086675037 TSecr=2981545644
12 0.008075		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [ACK] Seq=38 Ack=38 Win=29248 Len=0 TSval=2981545645 TSecr=1086675037
13 5.477343		RealtekU_72:fe:6e	RealtekU_72:fe:6e	ARP	60	Who has 192.168.60.101? Tell 192.168.60.102
14 5.477878		RealtekU_72:fe:6e	RealtekU_72:fe:6e	ARP	60	192.168.60.101 is at 52:54:00:72:fe:6e

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
 > Ethernet II, Src: RealtekU_72:fe:6e (52:54:00:72:fe:6e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff 52 54 00 72 fe 6e 08 06 00 01  ....RT..n....
0010  08 00 06 04 00 01 52 54 00 72 fe 6e c0 a8 3c 65  ....RT..n...ce
0020  00 00 00 00 00 00 c0 a8 3c 66 00 00 00 00 00  ....<f:.....
0030  00 00 00 00 00 00 00 00 00 00 00 00  ....
  
```

Preparado para cargar o capturar Paquetes: 14 - Mostrado: 14 (100.0%) Perfil: Default

Escribe aquí para buscar

Aplique un filtro de visualización ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		RealtekU_72:fe:6e	Broadcast	ARP	60	Who has 192.168.60.102? Tell 192.168.60.101
2 0.000699		RealtekU_72:fe:6e	RealtekU_72:fe:6e	ARP	60	192.168.60.102 is at 52:54:00:72:fe:6e
3 0.000699		192.168.60.101	192.168.60.102	TCP	74	56560 → 500 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2981545625 TSecr=0 WS=64
4 0.001674		192.168.60.102	192.168.60.101	TCP	74	500 → 56560 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1086675032 TSecr=2981545625 WS=64
5 0.002681		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [ACK] Seq=1 Ack=37 Win=29248 Len=0 TSval=2981545639 TSecr=1086675032
6 0.002681		192.168.60.101	192.168.60.102	ISAKMP	102	[Malformed Packet]
7 0.003212		192.168.60.102	192.168.60.101	TCP	66	500 → 56560 [ACK] Seq=1 Ack=37 Win=28992 Len=0 TSval=1086675033 TSecr=2981545639
8 0.004168		192.168.60.102	192.168.60.101	ISAKMP	102	[Malformed Packet]
9 0.004168		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [ACK] Seq=37 Ack=37 Win=29248 Len=0 TSval=2981545642 TSecr=1086675034
10 0.007204		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [FIN, ACK] Seq=37 Ack=37 Win=29248 Len=0 TSval=2981545644 TSecr=1086675034
11 0.007204		192.168.60.102	192.168.60.101	TCP	66	500 → 56560 [FIN, ACK] Seq=37 Ack=38 Win=28992 Len=0 TSval=1086675037 TSecr=2981545644
12 0.008075		192.168.60.101	192.168.60.102	TCP	66	56560 → 500 [ACK] Seq=38 Ack=38 Win=29248 Len=0 TSval=2981545645 TSecr=1086675037
13 5.477343		RealtekU_72:fe:6e	RealtekU_72:fe:6e	ARP	60	Who has 192.168.60.101? Tell 192.168.60.102
14 5.477878		RealtekU_72:fe:6e	RealtekU_72:fe:6e	ARP	60	192.168.60.101 is at 52:54:00:72:fe:6e

Conclusión del proyecto.

Como último, la elaboración de este proyecto estuvo interesante ya que nos permite aterrizar los conceptos vistos a lo largo de la unidad impartida, con el cual nos sirvió bastante, entendiendo más como es que estos funcionan cuando se aplican a un caso implementado en la práctica, teniendo en cuenta que fue un proyecto sencillo pero muy importante ya que nos permite ver que el uso de las telecomunicaciones es muy importante hoy en día.