

Instituto Tecnológico de Cancún

Fundamentos de Telecomunicaciones

Proyecto Sistema de Comunicación

Prof. Ismael Jiménez Sánchez

Alumno(a). Laury del Rosario Mex Martin

Ciclo 2020-B

Fecha de Entrega: 2 de Noviembre de 2020

FASE 1: Instalar 2 centos8 en VirtualBox usando Vagrant.

```
Windows PowerShell
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
PS C:\Users\Laury\tele> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'centos/7' could not be found. Attempting to find it...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Loading metadata for box 'centos/7'
default: URL: https://vagrantcloud.com/centos/7
==> default: Adding box 'centos/7' (v2004.01) for provider: virt
default: Downloading: https://vagrantcloud.com/centos/boxes/
Download redirected to host: cloud.centos.org
default:
default: Calculating and comparing box checksum...
==> default: Successfully added box 'centos/7' (v2004.01) for 'v
==> default: Importing base box 'centos/7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'centos/7' version '2004.01' is up
==> default: Setting the name of the VM: tele_default_1601938801
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration
default: Adapter 1: nat
==> default: Forwarding ports...
default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few mi
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
default:
default: Vagrant insecure key detected. Vagrant will automatically replace
default: this with a newly generated keypair for better security.
default:
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
default: No guest additions were detected on the base box for this VM! Guest
default: additions are required for forwarded ports, shared folders, host only
default: networking, and more. If SSH fails on this machine, please install
default: the guest additions and repack the box to continue.
default:
default: This is not an error message; everything may continue to work properly,
default: in which case you may ignore this message.
==> default: Rsyncing folder: /cygdrive/c/Users/Laury/tele/ => /vagrant
PS C:\Users\Laury\tele> vagrant ssh
[vagrant@localhost ~]$ exit
logout
Connection to 127.0.0.1 closed.
PS C:\Users\Laury\tele>
```

Oracle VM VirtualBox Administrador

Archivo Ayuda

Herramientas

Nueva Configuración Descartar Iniciar

FreeBSD Apagada

WifiSlax Apagada

tele_default_1601938801... Corriendo

General

Nombre: WifiSlax

Sistema operativo: Oracle (64-bit)

Sistema

Memoria base: 2227 MB

Orden de arranque: Óptica, Disco duro

VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB

Controlador gráfico: VMSVGA

Servidor de escritorio remoto: Inhabilitado

Grabación: Inhabilitado

Almacenamiento

Controlador: IDE

Previsualización

WifiSlax

Windows PowerShell

PS C:\Users\Laury\tele> vagrant halt
==> default: Attempting graceful shutdown of VM...
PS C:\Users\Laury\tele> cd..
PS C:\Users\Laury> mkdir tele2

Directorio: C:\Users\Laury

Mode LastWriteTime Length Name

d----- 05/10/2020 06:09 p. m. tele2

PS C:\Users\Laury> cd tele2\
PS C:\Users\Laury\tele2> vagrant init centos/7
A 'Vagrantfile' has been placed in this directory. You are now ready to 'vagrant up' your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on 'vagrantup.com' for more information on using Vagrant.
PS C:\Users\Laury\tele2> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'centos/7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'centos/7' version '2004.01' is up to date...
==> default: Setting the name of the VM: tele2_default_1601939481855_20415
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
==> default: Forwarding ports...
default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
default:
default: Vagrant insecure key detected. Vagrant will automatically replace this with a newly generated keypair for better security.
default:
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
default: No guest additions were detected on the base box for this VM! Guest additions are required for forwarded ports, shared folders, host only networking, and more. If SSH fails on this machine, please install the guest additions and repack the box to continue.
default:
default: This is not an error message; everything may continue to work properly, in which case you may ignore this message.
==> default: Rsyncing folder: /cygdrive/c/Users/Laury/tele2/ => /vagrant

Oracle VM VirtualBox Administrador

Archivo Máquina Ayuda

Herramientas

FreeBSD

Apagada

WifiSlax

Apagada

tele_default_1601...

Apagada

tele2_default_160193948...

Corriendo

Nueva Configuración Descartar Iniciar

General

Nombre: tele_default_1601938801201_8993
Sistema operativo: Red Hat (64-bit)

Sistema

Memoria base: 512 MB
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB
Controlador gráfico: VBoxVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

Almacenamiento

Controlador: IDE

Previsualización

tele_default_1601938801201_8993

FASE 2: Conectar en GNS3, las dos VMs de CentOS con un switch ethernet.

SistemaDeComunicacion - GNS3

File Edit View Control Node Annotate Tools Help

```
graph TD; cliente --- Hub1; Hub1 --- server;
```

Topology Summary

| Node | Console |
|---------|-----------------------|
| cliente | telnet localhost:5000 |
| Hub1 | none |
| server | telnet localhost:5001 |

Servers Summary

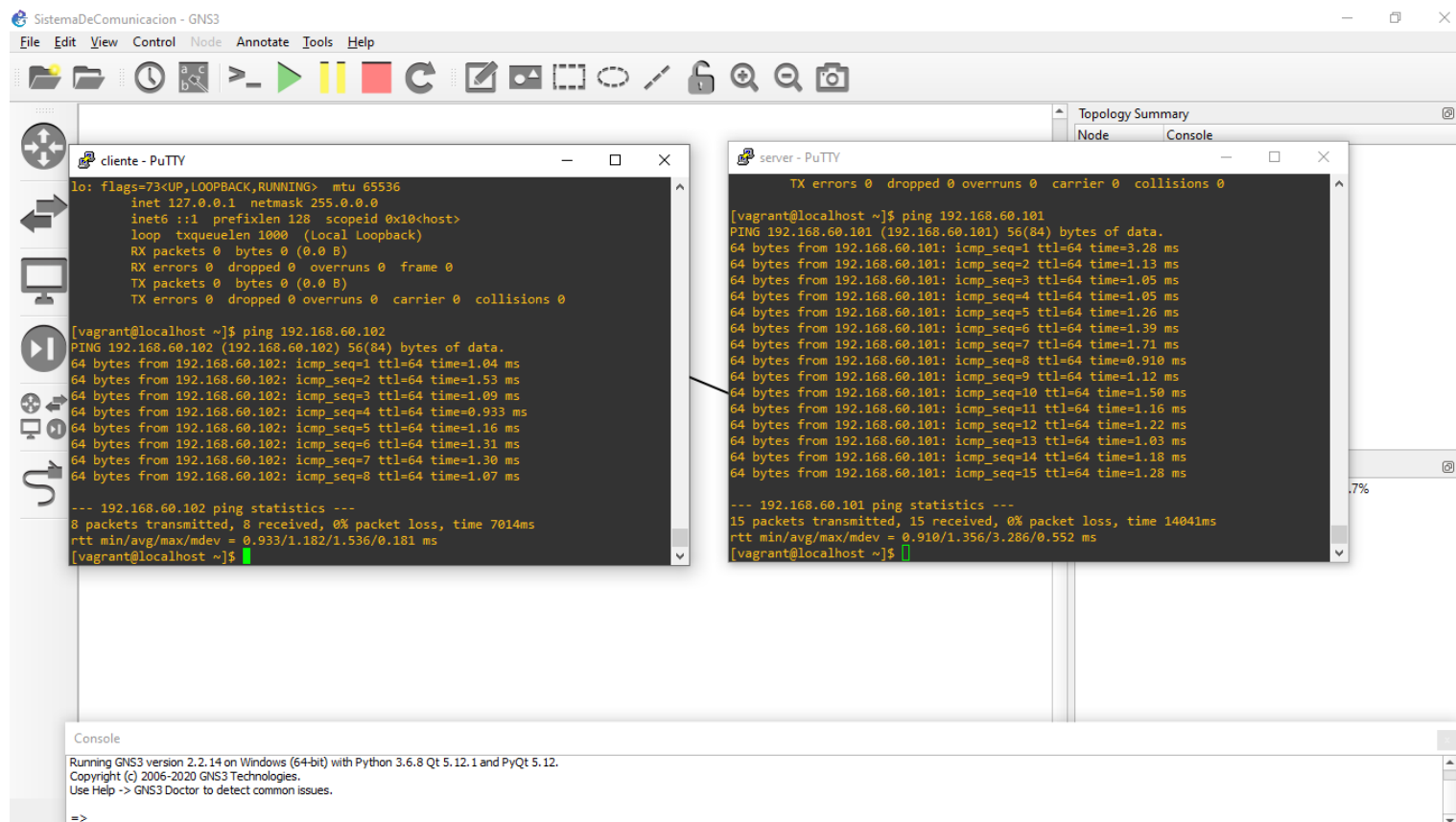
- DESKTOP-ORI6NAA CPU 23.5%, RAM 50.6%
- GNS3 VM is starting (GNS3 VM)

Console

Running GNS3 version 2.2.14 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.

FASE 3: Usar los scripts de Python para conectar las dos VMs usando sockets.

Antes realizar la conexión de los scripts comprobamos que haya comunicación entre las dos máquinas con un ping



The screenshot shows the GNS3 interface with two VMs connected. The 'cliente' VM (left) has an IP of 192.168.60.102 and is running a terminal window showing a successful ping to 192.168.60.101. The 'server' VM (right) has an IP of 192.168.60.101 and is also running a terminal window showing a successful ping to 192.168.60.102. The GNS3 console at the bottom indicates the version is 2.2.14 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.

```
cliente - PuTTY
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[vagrant@localhost ~]$ ping 192.168.60.102
PING 192.168.60.102 (192.168.60.102) 56(84) bytes of data.
64 bytes from 192.168.60.102: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from 192.168.60.102: icmp_seq=2 ttl=64 time=1.53 ms
64 bytes from 192.168.60.102: icmp_seq=3 ttl=64 time=1.09 ms
64 bytes from 192.168.60.102: icmp_seq=4 ttl=64 time=0.933 ms
64 bytes from 192.168.60.102: icmp_seq=5 ttl=64 time=1.16 ms
64 bytes from 192.168.60.102: icmp_seq=6 ttl=64 time=1.31 ms
64 bytes from 192.168.60.102: icmp_seq=7 ttl=64 time=1.30 ms
64 bytes from 192.168.60.102: icmp_seq=8 ttl=64 time=1.07 ms

--- 192.168.60.102 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7014ms
rtt min/avg/max/mdev = 0.933/1.182/1.536/0.181 ms
[vagrant@localhost ~]$
```

```
server - PuTTY
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[vagrant@localhost ~]$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=64 time=3.28 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=64 time=1.13 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=64 time=1.05 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from 192.168.60.101: icmp_seq=5 ttl=64 time=1.26 ms
64 bytes from 192.168.60.101: icmp_seq=6 ttl=64 time=1.39 ms
64 bytes from 192.168.60.101: icmp_seq=7 ttl=64 time=1.71 ms
64 bytes from 192.168.60.101: icmp_seq=8 ttl=64 time=0.910 ms
64 bytes from 192.168.60.101: icmp_seq=9 ttl=64 time=1.12 ms
64 bytes from 192.168.60.101: icmp_seq=10 ttl=64 time=1.50 ms
64 bytes from 192.168.60.101: icmp_seq=11 ttl=64 time=1.16 ms
64 bytes from 192.168.60.101: icmp_seq=12 ttl=64 time=1.22 ms
64 bytes from 192.168.60.101: icmp_seq=13 ttl=64 time=1.03 ms
64 bytes from 192.168.60.101: icmp_seq=14 ttl=64 time=1.18 ms
64 bytes from 192.168.60.101: icmp_seq=15 ttl=64 time=1.28 ms

--- 192.168.60.101 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14041ms
rtt min/avg/max/mdev = 0.910/1.356/3.286/0.552 ms
[vagrant@localhost ~]$
```

Console

Running GNS3 version 2.2.14 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

Script en el servidor y verificamos que el puerto se haya levantando con el comando netstat.

The screenshot displays a GNS3 network simulation environment. The main workspace shows a topology with three nodes: 'cliente' (a laptop icon), 'Hub1' (a switch icon), and 'server' (a laptop icon). They are connected in a star topology, with 'cliente' and 'server' both connected to 'Hub1'. The interface includes a top menu bar (File, Edit, View, Control, Node, Annotate, Tools, Help) and a toolbar with various icons for network management. On the left, there is a vertical toolbar with icons for adding nodes, connecting them, and other network functions. At the bottom, a console window shows the GNS3 version and copyright information.

Overlaid on the right side of the GNS3 window is a PuTTY terminal window titled 'server - PuTTY'. It displays a Python script for a TCP server and its execution output:

```
# print out what the client sends
request = client_socket.recv(1024)

print "[*] Received: %s" % request

# send back a packet
client_socket.send("ACK!")

client_socket.close()

while True:

    client,addr = server.accept()

    print "[*] Accepted connection from: %s:%d" % (addr[0],addr[1])

    # spin up our client thread to handle incoming data
    client_handler = threading.Thread(target=handle_client,args=(client,))
    client_handler.start()

"tcpserver.py" 37L, 753C written
[vagrant@localhost ~]$ python2 tcpserver.py
[*] Listening on 0.0.0.0:9999
```

At the bottom of the GNS3 window, a console window displays the following text:

```
Running GNS3 version 2.2.14 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>
```

```
[vagrant@localhost ~]$ [*] Listening on 0.0.0.0:9999
```

```
[vagrant@localhost ~]$ netstat -tapn | grep LISTEN
```

```
(Not all processes could be identified, non-owned process info  
will not be shown, you would have to be root to see it all.)
```

| | | | | | | |
|---------|---|---|--------------|-----------|--------|------|
| tcp | 0 | 0 | 0.0.0.0:9999 | 0.0.0.0:* | LISTEN | 985/ |
| python2 | | | | | | |
| tcp | 0 | 0 | 0.0.0.0:111 | 0.0.0.0:* | LISTEN | - |
| tcp | 0 | 0 | 0.0.0.0:22 | 0.0.0.0:* | LISTEN | - |
| tcp | 0 | 0 | 127.0.0.1:25 | 0.0.0.0:* | LISTEN | - |
| tcp6 | 0 | 0 | :::111 | :::* | LISTEN | - |
| tcp6 | 0 | 0 | :::22 | :::* | LISTEN | - |
| tcp6 | 0 | 0 | :::1:25 | :::* | LISTEN | - |

Script en el Cliente se ejecuta y vemos un mensaje de conexión

The screenshot displays the GNS3 interface with a topology consisting of a 'cliente' node (laptop icon) connected to a 'Hub1' node (switch icon). A terminal window titled 'cliente - PuTTY' is open, showing the execution of a Python script. The script attempts to connect to a server on localhost:5000, but it fails with a 'Connection reset by peer' error. The console at the bottom shows the GNS3 version and system information.

Topology Summary

| Node | Console |
|---------|-----------------------|
| cliente | telnet localhost:5000 |
| Hub1 | none |
| server | telnet localhost:5001 |

```
cliente - PuTTY
client = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
# connec the client
client.connect((target_host,target_port))
# send some data
client.send("GET / HTTP/1.1\r\nHost: google.com\r\n\r\n")
# receive some data
response = client.recv(4096)
print response

'''
'''
'''
'''
"tcpclient.py" 19L, 356C written
[vagrant@localhost ~]$ python2 tcpclient.py
Traceback (most recent call last):
  File "tcpclient.py", line 16, in <module>
    response = client.recv(4096)
socket.error: [Errno 104] Connection reset by peer
[vagrant@localhost ~]$
```

Console

Running GNS3 version 2.2.14 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

Revisamos el servidor y al momento de ejecutar el script del cliente nos aparece este mensaje de conexión (tuve un error de conversión del string a enviar).

The screenshot shows the GNS3 interface with two terminal windows. The 'cliente - PuTTY' window on the left shows a Python script that attempts to connect to a server. It prints the response and then shows a traceback error: 'TypeError: not all arguments converted during string formatting' at line 32 of 'tcpserver.py'. The 'server - PuTTY' window on the right shows the netstat output, which lists several listening ports. A green arrow points from the server window to the client window, indicating the connection attempt.

```
# create a socket object
client = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

# connec the client
client.connect((target_host,target_port))

# send some data
client.send("GET / HTTP/1.1\r\nHost: google.com\r\n\r\n")

# receive some data
response = client.recv(4096)

print response

~
~
~
"tcpclient.py" 19L, 356C written
[vagrant@localhost ~]$ python2 tcpclient.py
Traceback (most recent call last):
  File "tcpclient.py", line 16, in <module>
    response = client.recv(4096)
socket.error: [Errno 104] Connection reset by peer
[vagrant@localhost ~]$
```

```
[vagrant@localhost ~]$ [*] Listening on 0.0.0.0:9999

[vagrant@localhost ~]$ netstat -tapn | grep LISTEN
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 0.0.0.0:9999        0.0.0.0:*          LISTEN     985/
python2
tcp        0      0 0.0.0.0:111        0.0.0.0:*          LISTEN     -
tcp        0      0 0.0.0.0:22         0.0.0.0:*          LISTEN     -
tcp        0      0 127.0.0.1:25       0.0.0.0:*          LISTEN     -
tcp6       0      0 :::111             :::*               LISTEN     -
tcp6       0      0 :::22              :::*               LISTEN     -
tcp6       0      0 :::1:25            :::*               LISTEN     -

[vagrant@localhost ~]$ Traceback (most recent call last):
  File "tcpserver.py", line 32, in <module>
    print "[*] Accepted connection from: %s:%d" % (addr[0],addr[1])
TypeError: not all arguments converted during string formatting
netstat -tapn | grep LI
```

Console

Running GNS3 version 2.2.14 on Windows (64-bit) with Python 3.6.8 Qt 5.12.1 and PyQt 5.12.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=>

FASE 4 :Capturar el tráfico de la comunicación entre las dos VMs al momento de utilizar los scripts.

Nos dirigimos al WireShark para ver la captación del tráfico que se realizó al usar los scripts.

Captura del Servidor

The screenshot shows the Wireshark interface with a capture filter applied: "Apply a display filter ... <Ctrl-/>". The packet list shows 86 packets. The selected packet is Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface -, id 0. The packet details pane shows the following structure:

- Ethernet II, Src: RealtekU_4d:77:d3 (52:54:00:4d:77:d3), Dst: IPv6mcast_16 (33:33:00:00:00:16)
- Internet Protocol Version 6, Src: ::, Dst: ff02::16
- Internet Control Message Protocol v6

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 33 33 00 00 00 16 52 54 00 4d 77 d3 86 dd 60 00 33...RT·Mw...
0010 00 00 00 24 00 01 00 00 00 00 00 00 00 00 00 00 ...$.
0020 00 00 00 00 00 00 ff 02 00 00 00 00 00 00 00 00 .....
0030 00 00 00 00 00 16 3a 00 05 02 00 00 01 00 8f 00 .....:
0040 f7 69 00 00 00 01 04 00 00 00 ff 02 00 00 00 00 ..i.....
0050 00 00 00 00 00 01 ff 4d 77 d3 .....Mw..
```

The status bar at the bottom indicates: "Ready to load or capture", "Packets: 86 · Displayed: 86 (100.0%)", and "Profile: Default".

Captura del Cliente

Capturing from - [cliente Ethernet0 to Hub1 Ethernet0]

FileEditViewGoCaptureAnalyzeStatisticsTelephonyWirelessToolsHelp

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|-------------------|-------------------|----------|--------|---|
| 1 | 0.000000 | 192.168.60.101 | 192.168.60.102 | TCP | 74 | 51462 → 9999 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=5306691 TSecr=0 WS=64 |
| 2 | 0.000977 | 192.168.60.102 | 192.168.60.101 | TCP | 60 | 9999 → 51462 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 3 | 5.004807 | RealtekU_4d:77:d3 | RealtekU_4d:77:d3 | ARP | 60 | Who has 192.168.60.102? Tell 192.168.60.101 |
| 4 | 5.005444 | RealtekU_4d:77:d3 | RealtekU_4d:77:d3 | ARP | 60 | 192.168.60.102 is at 52:54:00:4d:77:d3 |

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface -, id 0

> Ethernet II, Src: RealtekU_4d:77:d3 (52:54:00:4d:77:d3), Dst: RealtekU_4d:77:d3 (52:54:00:4d:77:d3)

> Internet Protocol Version 4, Src: 192.168.60.101, Dst: 192.168.60.102

> Transmission Control Protocol, Src Port: 51462, Dst Port: 9999, Seq: 0, Len: 0

0000 52 54 00 4d 77 d3 52 54 00 4d 77 d3 08 00 45 00 RT·Mw·RT·Mw··E·

0010 00 3c fc 23 40 06 40 06 44 7c c0 a8 3c 65 c0 a8 ·<#00@·D|··<e·

0020 3c 66 c9 06 27 0f 01 2f 05 60 00 00 00 00 a0 02 <f····/·`·····

0030 72 10 eb 9b 00 00 02 04 05 b4 04 02 08 0a 00 50 r·····P

0040 f9 43 00 00 00 00 01 03 03 06 ·C·····

Ready to load or capture

Packets: 4 · Displayed: 4 (100.0%)

Profile: Default

Fase 5 Conclusiones.

Podemos ver en las capturas el proceso del protocolo de 3 vías (triple way handshake), donde primero el cliente envía una solicitud de conexión al servidor y envía un SYN (sincronizar número de secuencia) para comenzar una conmutación con el servidor, tanto en el servidor y en el cliente se refleja este es el primer paso para la comunicación, luego se ve el SYN, ACK que es donde el servidor responde la solicitud del cliente; este solo se refleja en el servidor ya que solo esta aceptando la solicitud. Y al final de la comunicación vemos un ACK de parte de ambas capturas del servidor y cliente que significa que el cliente ya reconoce la respuesta del servidor y ya se puede tener una conexión confiable entre estos.

Este sistema funciona con el flujo de de datos full-duplex.