

“REDES DE INFRAESTRUCTURA EN NS-3”

Informe de laboratorio N°12
Laboratorio de Comunicaciones Inalámbricas

1st Melanny Dávila
Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

2nd Jonathan Álvarez
Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
jonathan.alvarez@epn.edu.ec

Abstract—En el presente informe se presentarán los resultados obtenidos durante la práctica de laboratorio, la misma que se desarrolló en NS-3 mediante el uso de VMware con dos diferentes protocolos: TCP y UDP.

Index Terms—NS-3, red de infraestructura, Linux, comandos, estructura, UDP, TCP.

I. INTRODUCCIÓN

NS-3 es un simulador de redes basado en eventos discretos. Se usa principalmente en ambientes educativos y de investigación. Permite simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes móviles ad-hoc. Implementa una amplia gama de protocolos tanto de redes cableadas como de redes inalámbricas.

NS-3 brinda modelos de cómo trabajan las redes en el envío de paquetes de datos. Algunas de las razones para usarlo incluyen realizar estudios que son más difíciles o imposibles de realizar con sistemas reales, estudiar el comportamiento del sistema en un entorno altamente controlado y reproducible, y aprender cómo funcionan las redes [1].

II. OBJETIVOS

- Conocer el simulador ns-3 y su aplicación en redes inalámbricas.
- Simular una red de infraestructura en ns-3.
- Entender la estructura de un programa en ns-3.

III. CUESTIONARIO

A. *Presente las capturas de los resultados obtenidos durante la práctica y comente cada una de las imágenes. Adicional genere un resumen donde explique los resultados de lo realizado en la práctica.*

Durante la sesión de laboratorio, se trabajó con el software NS-3 dentro del programa ap-example se modificó la dirección IP de red a la dirección: 192.169.60.0/24 tal y como se presenta en la figura 1.

```
168 Ipv4AddressHelper address;  
169 address.SetBase ("192.168.60.0", "255.255.255.0");  
170 Ipv4InterfaceContainer staNodeInterface;  
171 Ipv4InterfaceContainer apNodeInterface;
```

Fig. 1. Dirección IP de red modificada

Al guardar dichos cambios se debe ejecutar el programa desde el terminal de Linux mediante el uso del comando `/waf --run scratch/ap-example --vis` y de esta forma se desplegará una interfaz gráfica que permite analizar la información de los nodos y el enlace presente entre ellos.

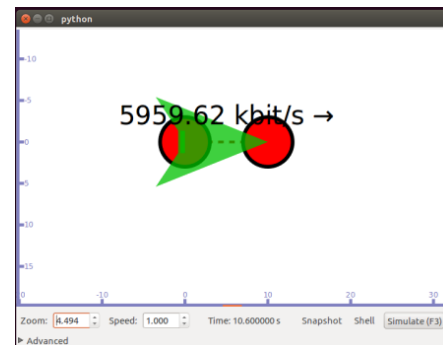


Fig. 2. Enlace establecido

- Características del enlace:

```
WiFi link between STA Node 0 and AP Node 1; distance=10.00 m.  
SSID: ns3-80211n  
BSSID: 00:00:00:00:00:02
```

Fig. 3. Propiedades del enlace

- Características del nodo 1:

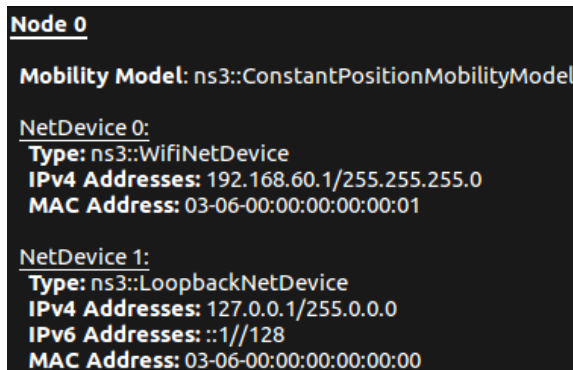


Fig. 4. Propiedades del nodo 1

- Características del nodo 2:

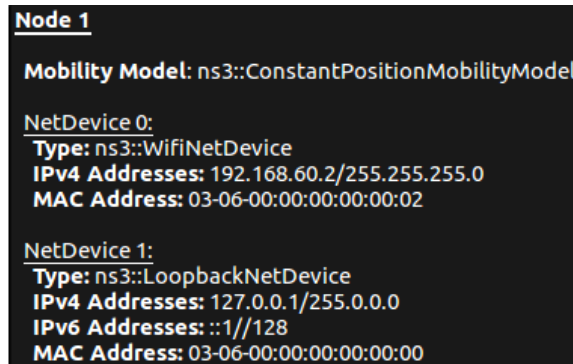


Fig. 5. Propiedades del nodo 2

Donde el throughput obtenido fue de 5.81381 [Mbps] para este caso, como evidencia se presenta la figura 6.

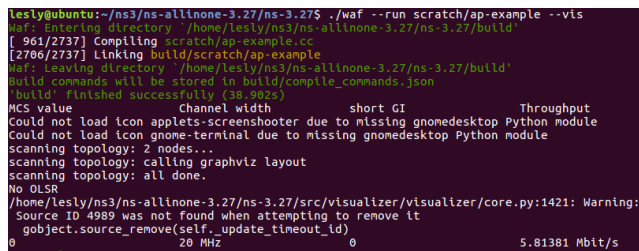


Fig. 6. Throughput obtenido

Dentro de la carpeta “pcapTraces” se encuentran los paquetes que fueron transmitidos en el enlace, Posterior a eso, se analizaron las trazas generadas mediante Wireshark.

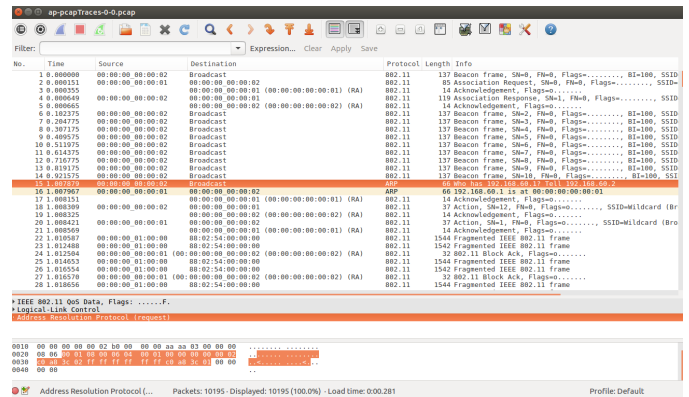


Fig. 7. Trazas UDP

Bajo el mismo escenario se trabajo con el protocolo TCP para lo cual se cambió el valor de la propiedad udp a “falso”, eso se presenta en la figura 8.

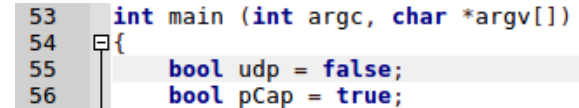


Fig. 8. Cambio de protocolo UDP a TCP

Permitiendo obtener como resultado un throughput de 5.10275 [Mbps].

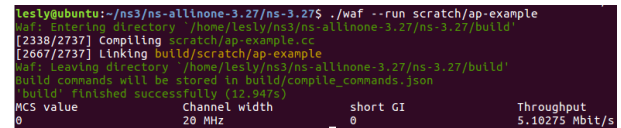


Fig. 9. Throughput obtenido

En este caso se puede observar diferencia en el valor resultante de throughput debido a que el protocolo UDP que no tiene ningún mecanismo de control de flujo, el sistema de ventanas hará que los datos se envíen con facilidad sin importar un desbordamiento. Por el contrario, TCP admite funciones de control de flujo y ventanas.

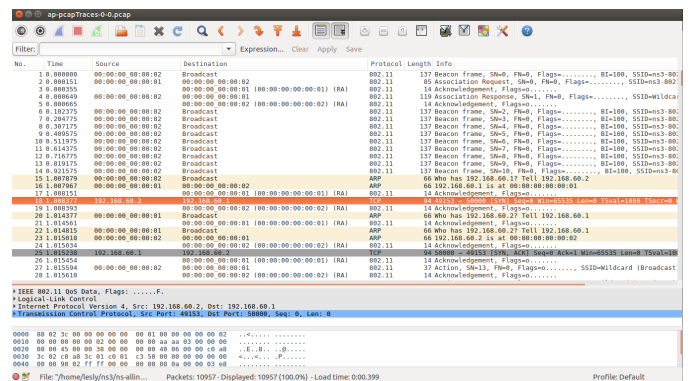


Fig. 10. Trazas TCP

Finalmente, se trabajaron tres casos en los cuales se tiene diferente distancia a la cual el enlace trabajará.

- Caso 1: 10 [m]
 - Protocolo: TCP

```
lesly@ubuntu:~/ns3/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/ap-example
waf: Entering directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
[2344/2737] Compiling scratch/ap-example.cc
[2724/2737] Linking build/scratch/ap-example
waf: Leaving directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (153.146s)
MCS value      Channel width      short GI      Throughput
0              20 MHz              0             5.10275 Mbit/s
```

Fig. 11. Throughput para TCP a 10 [m]

- Protocolo: UDP

```
lesly@ubuntu:~/ns3/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/ap-example
waf: Entering directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
[2337/2737] Compiling scratch/ap-example.cc
[2720/2737] Linking build/scratch/ap-example
waf: Leaving directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.432s)
MCS value      Channel width      short GI      Throughput
0              20 MHz              0             5.81381 Mbit/s
```

Fig. 12. Throughput para UDP a 10 [m]

- Caso 2: 93 [m]
 - Protocolo: TCP

```
lesly@ubuntu:~/ns3/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/ap-example
waf: Entering directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
[2684/2737] Linking build/scratch/ap-example
waf: Leaving directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.202s)
MCS value      Channel width      short GI      Throughput
0              20 MHz              0             4.78651 Mbit/s
```

Fig. 13. Throughput para TCP a 93 [m]

- Protocolo: UDP

```
lesly@ubuntu:~/ns3/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/ap-example
waf: Entering directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
[ 961/2737] Compiling scratch/ap-example.cc
[2722/2737] Linking build/scratch/ap-example
waf: Leaving directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.022s)
MCS value      Channel width      short GI      Throughput
0              20 MHz              0             5.49047 Mbit/s
```

Fig. 14. Throughput para UDP a 93 [m]

- Caso 1: 103 [m]
 - Protocolo: TCP

```
lesly@ubuntu:~/ns3/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/ap-example
waf: Entering directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
[2338/2737] Compiling scratch/ap-example.cc
[2663/2737] Linking build/scratch/ap-example
waf: Leaving directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.584s)
MCS value      Channel width      short GI      Throughput
0              20 MHz              0             0.0011584 Mbit/s
```

Fig. 15. Throughput para TCP a 103 [m]

- Protocolo: UDP

```
lesly@ubuntu:~/ns3/ns-allinone-3.27/ns-3.27$ ./waf --run scratch/ap-example
waf: Entering directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
[2338/2737] Compiling scratch/ap-example.cc
[2669/2737] Linking build/scratch/ap-example
waf: Leaving directory '/home/lesly/ns3/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (6.584s)
MCS value      Channel width      short GI      Throughput
0              20 MHz              0             0.0011776 Mbit/s
```

Fig. 16. Throughput para UDP a 103 [m]

TABLA I
RESULTADOS OBTENIDOS

Distancia [m]	Throughput en base al protocolo [Mbps]	
	UDP	TCP
10	5.81381	5.10275
93	5.40047	4.78651
103	0.0011776	0.0011584

Los datos obtenidos en las figuras 11-16 se presentan en la siguiente tabla.

El rendimiento de TCP puede variar en el tiempo durante la transmisión de datos. Además, debido a que la longitud de su cabecera es de 20 bytes el throughput también disminuirá, mientras que la longitud de la cabecera de UDP es de 8 bytes. Asimismo, TCP es más lento que UDP debido a que no es un protocolo de mejor esfuerzo sino es un protocolo confiable que garantiza la transmisión de los datos.

B. Compare y analice, vía la visualización de los paquetes en Wireshark, las diferencias entre transmitir con UDP vs transmitir con TCP.

TCP es un protocolo orientado a la conexión, lo que significa que los dispositivos de comunicación deben establecer una conexión antes de transmitir datos y deben cerrar la conexión después de transmitir los datos. Es así como en las tramas capturadas se puede apreciar un handshake de tres vías previo a iniciar la conexión. Dado a que TCP proporciona mecanismos de comprobación de errores, permite implementar control de flujo y ACK; mediante su encabezado de longitud variable (20-60) bytes por lo que la tasa de de throughput disminuye.

UDP es el protocolo donde no hay “gastos” para abrir una conexión, mantener una conexión y terminar una conexión. La entrega de datos al destino no es garantizada, por lo que es eficiente para el tipo de transmisión de red de difusión y multidifusión debido a que solo tiene el mecanismo básico de verificación de errores mediante sumas de verificación debido a que la longitud de su cabecera es de 8 bytes.

C. Realice una nueva tabla comparativa para analizar las variaciones del throughput, esta vez mantenga constante la distancia (30 metros) y varíe el tiempo de simulación (mínimo 5 distintos valores de tiempo de simulación), realice este proceso para la transmisión con UDP y para TCP, así como para la banda de los 2.4 GHz y la banda de los 5 GHz. Comente los resultados obtenidos.

TABLA II
VARIACIONES DE THROUGHPUT PARA 2.4 GHz

Banda de 2.4 GHz		
Tiempo [s]	UDP[Mbps]	TCP [Mbps]
5	5.74433	5.02051
10	5.74669	5.01471
20	5.74904	5.01413
40	5.7511	5.02253
80	5.75155	5.0276

Como primer caso tomamos la banda de 2.4 GHz para tiempos que se duplican a partir de 5 segundos. Al observar la tendencia de UDP se puede apreciar que para un periodo de tiempo el crecimiento del throughput es constante aunque no es muy significativo, cuando se llegó al ultimo tiempo se obtuvo que la tendencia de crecimiento se redujo. Es importante considerar que para ambos casos pero especialmente para UDP a medida que aumentábamos el tiempo la duración de la simulación crecía de gran manera. En el caso de TCP podemos apreciar que en un inicio tuvimos una disminución del throughput de 5 a 10 segundos, a partir de los 20 segundos este valor comenzó a crecer pero sigue siendo menor a UDP.

TABLA III
VARIACIONES DE THROUGHPUT PARA 5 GHZ

Banda de 5 GHz		
Tiempo [s]	UDP [Mbps]	TCP[Mbps]
5	5.81028	5.10391
10	5.81263	5.10275
20	5.81499	5.09175
40	5.81676	5.10507
80	5.81734	5.10304

Para el caso de 5 GHz se aprecia una tendencia similar a la notada anteriormente. El throughput crece tanto para TCP y UDP al pasar de la banda de 2.4 GHz. UDP sigue presentando incrementos constantes hasta llegar al tiempo mayor donde en relación a los anteriores disminuye. La duración de la simulación para esta banda y con el protocolo UDP fue considerable. Para el protocolo TCP se presenta una reducción en el valor de Throughput para los primeros tres tiempos luego un crecimiento y finalmente una disminución.

```
int main (int argc, char *argv[])
{
    bool udp = true;
    bool pCap = true;

    double simulationTime = 5; //seconds
    double distance = 30.0; //meters
    double frequency = 2.4; //whether 2.4 or 5.0 GHz
    int mcs = 0; // modulation code scheme
    uint32_t channelWidth = 20; // channel width
    uint32_t sgi = 0; // channel width
}
```

Fig. 17. Parámetros configurados en el código para la simulación

D. Conclusiones:

- Wireshark es una herramienta poderosa cuando desea verificar lo que realmente está sucediendo en una red, debido a que proporciona las herramientas para profundizar en las trazas capturadas.
- TCP es un protocolo confiable ya que garantiza la entrega de datos al destino debido a todos los mecanismos utilizados por el protocolo. Mientras que la entrega de datos al destino no se puede garantizar en UDP debido a sus prestaciones.
- UDP obtiene mayor throughput que TCP al mantener constante valores como la distancia y tiempo. Esto podemos atribuir al tamaño de la cabecera en UDP siendo

menor que para TCP, también que el primero tiene como prioridad el envío de la información de manera mas rápida.

E. Recomendaciones:

- Para alcanzar mayores tasas de throughput, se debe trabajar en la banda de frecuencia de 5 [GHz] debido a que al realizar este cambio, la cantidad de datos que se puede transmitir es mayor.
- Cerrar todas las aplicaciones que puedan ocupar memoria para que las simulaciones puedan procesarse mas rápido

REFERENCES

- [1] "NS-3", ns-3. <https://www.nsnam.org/> (accedido ago. 28, 2021).
- [2] S. R. in Linux, O. Source, in N. on March 7, 2012, y 11:44 Pm Pst, "Two simple filters for wireshark to analyze TCP and UDP traffic", TechRepublic. <https://www.techrepublic.com/blog/linux-and-open-source/two-simple-filters-for-wireshark-to-analyze-tcp-and-udp-traffic/> (accedido ago. 30, 2021).
- [3] "Wireshark — Key Differences Between Tcpdump vs Wire-shark", EDUCBA, jul. 01, 2021. <https://www.educba.com/tcpdump-vs-wireshark/> (accedido ago. 30, 2021).