

Cuaderno de Ingeniería



Marcos Folguera Rivera

Curso 2023/2024

Índice de contenidos

Índice de Figuras	
CP01: Parámetros de red y pruebas de conectividad con la herramienta ping.	Pag 3
1. Diagrama de topología (Figura)	
2. Situación inicial	
2.1 Desarrollo de la tarea	
3. Desarrollo	
4. Conclusiones	
CP02: Rastreo de rutas de Internet con la herramienta tracert (traceroute).....	Pag 6
1. Diagrama de topología (Figura)	
2. Situación inicial	
2.1 Desarrollo de la tarea	
3. Desarrollo	
4. Conclusiones	
CP03: Capturar paquetes con Wireshark y aplicar filtros por tipo de protocolo.....	Pag 9
1. Diagrama de topología (Figura)	
2. Situación inicial	
2.1 Desarrollo de la tarea	
3. Desarrollo	
4. Conclusiones	
CP04: Describir con Wireshark una resolución DNS a través de la captura de los datagramas UDP correspondientes.	Pag 11
1. Diagrama de topología (Figura)	
2. Situación inicial	
2.1 Desarrollo de la tarea	
3. Desarrollo	
4. Conclusiones	
CP05: Describir con Wireshark los mensajes de solicitud y respuesta para una comunicación HTTP	Pag 14
1. Diagrama de topología (Figura)	
2. Situación inicial	
2.1 Desarrollo de la tarea	
3. Desarrollo	
4. Conclusiones	
CP06: Seguimiento de las conexiones TCP en un PC mediante la herramienta netstat.	Pag 18
1. Diagrama de topología (Figura)	
2. Situación inicial	
2.1 Desarrollo de la tarea	
3. Desarrollo	
4. Conclusiones	

Cuaderno de Ingeniería. Curso 2023/2024

Marcos Folguera Rivera

CP07: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 25

1. Diagrama de topología (Figura)
2. Situación inicial
- 2.1 Desarrollo de la tarea
3. Desarrollo
4. Conclusiones

-----2º Entrega-----

CP08: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 31

1. Diagrama de topología (Figura)
2. Situación inicial
- 2.1 Desarrollo de la tarea
3. Desarrollo
4. Conclusiones

CP09: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 36

1. Diagrama de topología (Figura)
2. Situación inicial
- 2.1 Desarrollo de la tarea
3. Desarrollo
4. Conclusiones

CP010: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 39

1. Diagrama de topología (Figura)
2. Situación inicial
- 2.1 Desarrollo de la tarea
3. Desarrollo
4. Conclusiones

CP011: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 42

1. Diagrama de topología (Figura)
2. Situación inicial
- 2.1 Desarrollo de la tarea
3. Desarrollo
4. Conclusiones

CP012: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 46

1. Diagrama de topología (Figura)
2. Situación inicial
- 2.1 Desarrollo de la tarea
3. Desarrollo
4. Conclusiones

CP013: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 49

1. Diagrama de topología (Figura)

Cuaderno de Ingeniería. Curso 2023/2024

Marcos Folguera Rivera

- 2. Situación inicial
- 2.1 Desarrollo de la tarea
- 3. Desarrollo
- 4. Conclusiones

CP014: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.....Pag 52

- 1. Diagrama de topología (Figura)
- 2. Situación inicial
- 2.1 Desarrollo de la tarea
- 3. Desarrollo
- 4. Conclusiones

Índice de figuras

Figura 1.....

Figura 2.....

Figura 3.....

Figura 4.....

Figura 5.....

Figura 6.....

Figura 7.....

Figura 8.....

Figura 9.....

Figura 10.....

Figura 11.....

Figura 12.....

Figura 13.....

Figura 14.....

Figura 15.....

Figura 16.....

Figura 17.....

Figura 18.....

Figura 19.....

Figura 20.....

-----2º Entrega-----

Figura 21.....

Figura 22.....

Figura 23.....

Figura 24.....

Figura 25.....

Figura 26.....

Figura 27.....

Figura 28.....

Figura 29.....

Figura 30.....

Figura 31.....

Figura 32.....

Figura 33.....

Figura 34.....

Figura 35.....

Figura 36.....

Figura 37.....

Figura 38.....

Figura 39.....

Figura 40.....

Figura 41.....

Figura 42.....

Figura 43.....

Figura 44.....

Cuaderno de Ingeniería. Curso 2023/2024

Marcos Folguera Rivera

CP01: Parámetros de red y pruebas de conectividad con la herramienta ping.

1. Diagrama de topología (Figura 1)

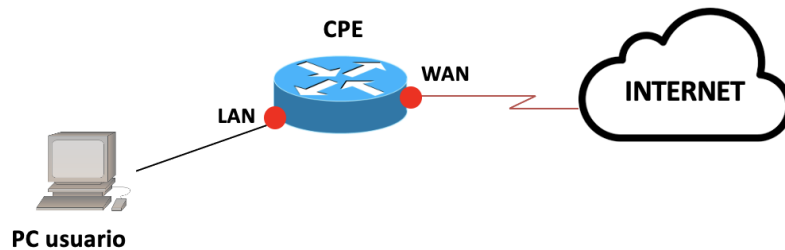


Figura 1. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida.

2.1 Desarrollo de la tarea

En esta tarea buscaremos probar la conectividad de red mediante la herramienta ping. Esta herramienta se utiliza para probar la conectividad de red TCP/IP, al igual que mide también el tiempo de ida y vuelta para los mensajes que se envían desde el host de origen al destino., para poder utilizarla se requiere obviamente un PC conectado al internet en el que se comprobará los parámetros de red y se podrá realizar las distintas pruebas de conectividad.

3. Desarrollo

3.1. Mostrar y describir la información de red de una NIC con el comando **ipconfig** y otras posibilidades de uso

Comando :**ipconfig**

Dirección IPv4:192.168.0.18

Máscara de subred: 255.255.255.0

Puerta de enlace predeterminada:192.168.0.1

Servidor DNS 1:158.49.17.21

Probar la conectividad de red mediante el comando ping

La herramienta que se utiliza para probar la conectividad se llama ping, esta herramienta envía paquetes de información al host remoto, luego el PC local observa si se recibe una respuesta por cada paquete y cuánto tardan en atravesar la red en caso positivo en ms. Lo primero que se debe realizar con ping es utilizarlo para comprobar que nos funciona correctamente la conexión hasta la puerta de enlace (figura 1), para hacer esto debemos entrar en la consola de comandos y escribir ping puerta_enlace, siendo puerta_enlace la dirección IP de nuestra puerta de enlace predeterminada

3.2. Realizar e interpretar una prueba de conectividad con el comando **ping** a la puerta de enlace predeterminada

Gracias al ejercicio anterior conocemos la dirección de nuestra puerta de enlace y procedemos a realizar el comando ping

```
C:\Users\Pc>ping 192.168.0.1

Haciendo ping a 192.168.0.1 con 32 bytes de datos:
Respuesta desde 192.168.0.1: bytes=32 tiempo=4ms TTL=64
Respuesta desde 192.168.0.1: bytes=32 tiempo=4ms TTL=64
Respuesta desde 192.168.0.1: bytes=32 tiempo=4ms TTL=64
Respuesta desde 192.168.0.1: bytes=32 tiempo=6ms TTL=64

Estadísticas de ping para 192.168.0.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 4ms, Máximo = 6ms, Media = 4ms
```

1 Figura .comando tracert

3.3. Realizar e interpretar la una prueba de conectividad con el comando **ping** al host remoto gaia.cs.umass.edu

Realizamos el comando ping a la dirección del servidor gaia

```
C:\Users\Pc>ping gaia.cs.umass.edu

Haciendo ping a gaia.cs.umass.edu [128.119.245.12] con 32 bytes de datos:
Respuesta desde 128.119.245.12: bytes=32 tiempo=116ms TTL=43
Respuesta desde 128.119.245.12: bytes=32 tiempo=118ms TTL=43
Respuesta desde 128.119.245.12: bytes=32 tiempo=119ms TTL=43
Respuesta desde 128.119.245.12: bytes=32 tiempo=128ms TTL=43

Estadísticas de ping para 128.119.245.12:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 116ms, Máximo = 128ms, Media = 120ms
```

2 Figura .comando ping

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctica con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	-
Capa de Transporte	-
Capa de Red	ICMP, IPv4
Capa de Enlace de Datos	Wi-Fi (IEEE802.11)
Capa Física	

Y las conclusiones para este contenido son:

- La puerta de enlace predeterminada es el primer router de acceso para conectarse a la red, es necesario para conectarse servidores desde fuera de la red local.
- La herramienta ping que sirve para ver los parámetros de conexión entre dos dispositivos de red (host, router, servidores) es extremadamente útil para comprobar problemas de conectividad (problemas de wifi haciendo ping a la puerta de enlace) y latencia (como ancho de banda insuficiente, distancia y número de nodos, entre otros).
- Los servidores DHCP sirven para asignar direcciones IP, configuración de puerta de enlace predeterminada, asignar direcciones de servidores DNS (servidores de dominio), reservar direcciones IP, entre otros protocolos para los hosts clientes cuando se conectan a una red.

CP02: Rastreo de rutas de Internet con la herramienta tracert (traceroute).

1. Diagrama de topología (Figura 2)

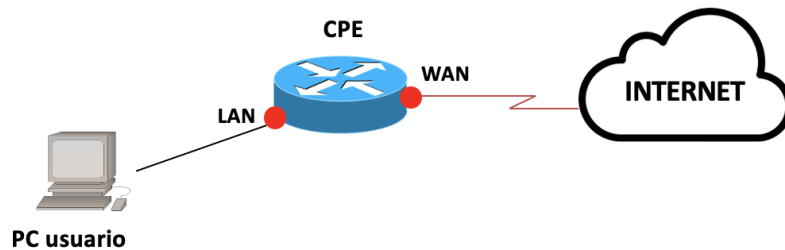


Figura 2. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida.

2.1 Desarrollo de la tarea

En esta tarea rastreamos una ruta a un servidor remoto utilizando la herramienta traceroute, esta herramienta permite a un usuario determinar la trayectoria o ruta seguida por los paquetes, así como el retardo (latencia) sufrido a través de la misma, además de ser utilizado para resolver problemas en redes IP, ya que permite al usuario ver la ruta tomada por los paquetes de datos que puede aclarar porque ocurren ciertos problemas.

Las rutas rastreadas pueden atravesar muchos “saltos”(routers) de distintos proveedores de servicios de Internet (ISP), según el tamaño del ISP y la ubicación de los hosts de origen y destino. Cada “salto” representa un router o enrutador.

3. Desarrollo

3.1. Mostrar la ruta seguida para llegar al host remoto *gaia.cs.umass.edu* con el comando **tracert**

Para mostrar la ruta deberemos utilizar el comando `tracert` y la dirección del servidor a acceder sobre una ventana del terminal como se muestra en la figura

```
C:\Users\Pc>tracert gaia.cs.umass.edu

Traza a la dirección gaia.cs.umass.edu [128.119.245.12]
sobre un máximo de 30 saltos:

  1      5 ms      4 ms      3 ms  www.adsl.vf [192.168.0.1]
  2      *          *          *      Tiempo de espera agotado para esta solicitud.
  3      *          *          *      Tiempo de espera agotado para esta solicitud.
  4     18 ms     17 ms     17 ms  ae7-100-xcr1.mat.cw.net [195.10.44.1]
  5     18 ms     18 ms     21 ms  ae2-xcr1.max.cw.net [195.2.30.89]
  6    2609 ms     17 ms     17 ms  ae1.cr0-mad5.ip4.gtt.net [154.14.148.5]
  7     113 ms    128 ms    113 ms  et-0-0-1.cr1-bos1.ip4.gtt.net [89.149.136.78]
  8     119 ms    117 ms    116 ms  ip4.gtt.net [65.175.24.206]
  9     115 ms    116 ms    115 ms  69.16.1.0
 10    234 ms    115 ms    116 ms  core2-rt-et-8-3-0.gw.umass.edu [192.80.83.113]
 11    117 ms    116 ms    121 ms  n1-rt-1-1-et-10-0-0.gw.umass.edu [128.119.0.120]
 12    115 ms    117 ms    118 ms  128.119.7.74
 13    122 ms    116 ms    117 ms  128.119.7.66
 14    118 ms    115 ms    122 ms  core1-rt-et-7-2-1.gw.umass.edu [128.119.0.217]
 15    118 ms    119 ms    117 ms  n5-rt-1-1-xe-2-1-0.gw.umass.edu [128.119.3.33]
 16    117 ms    118 ms    117 ms  cics-rt-xe-0-0-0.gw.umass.edu [128.119.3.32]
 17    120 ms    119 ms    118 ms  nscs1bbs1.cs.umass.edu [128.119.240.253]
 18    117 ms    116 ms    118 ms  gaia.cs.umass.edu [128.119.245.12]

Traza completa.
```

3 Figura .comando tracert

3.2. Interpretar (de forma general) el tipo de información que puede observarse por el comando anterior

Hay un total de 18 saltos en este ejemplo y podemos ver que en varios de estos se necesita más tiempo o incluso es demasiado largo y lo descarta. Hay que tener en cuenta que si realizáramos otro `tracert` a `cisco`, es probable que los nodos por los que pasásemos fueran distintos debido a la naturaleza de las redes interconectadas que componen Internet, pues están basadas en conexiones múltiples o múltiples caminos (topología física en malla o malla parcial) y la posibilidad de seleccionar distintas rutas por las cuales enviar los paquetes (redes de conmutación de paquetes). En este caso se dice que el camino de ida y vuelta de los paquetes es asimétrico (ASYM).

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctico con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	-
Capa de Transporte	-
Capa de Red	ICMP, IPv4
Capa de Enlace de Datos	Wi-Fi (IEEE802.11)
Capa Física	

Y las conclusiones para este contenido son:

1. Si realizamos traceroute a un sitio varias veces, es probable que el camino sea distinto debido a que internet está basado en conexiones múltiples
2. El primer salto siempre será el de la puerta de enlace predeterminada.
3. Aunque es posible que se utilice una ISP (Las ISP actúan como intermediarios entre los usuarios y el resto de Internet) para todos los saltos realizados, es mucho más probable que se utilicen varias, estas se pueden identificar utilizando Whois

CP03: Capturar paquetes con Wireshark y aplicar filtros por tipo de protocolo.

1. Diagrama de topología (Figura 3)

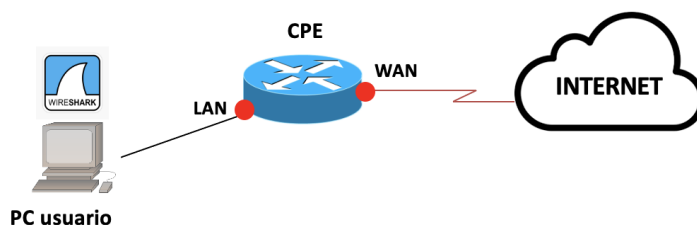


Figura 3. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida y con la herramienta software Wireshark instalada.

2.1 Desarrollo de la tarea

En esta tarea se capturar tramas ICMP con wireshark , mas adelante se explicara el contenido a realizar

3. Desarrollo

3.1. Describir cómo realizar una captura de paquetes con Wireshark para el comando ping gaia.cs.umass.edu

Primero deberemos abrir tanto el cmd como la aplicación wireshark.

Dentro de wireshark hacer doble click en el adaptador wifi y luego darle a start para empezar a capturar datos.

También se le pueden aplicar filtros , en este caso aplicaremos el siguiente filtro: ip.addr == 128.119.245.12 (Que es la ip de gaia.cs.umass.edu)

Ahora en la cmd haremos el comando ping gaia.cs.umass.edu

Captura de wireshark con filtro: ip.addr == 128.119.245.12

No.	Time	Source	Destination	Protocol	Length	Info
4594	337.795717	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=262/1537, ttl=128 (reply in 4595)
4595	337.911928	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=262/1537, ttl=43 (request in 4594)
4597	338.811959	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=263/1793, ttl=128 (reply in 4598)
4598	338.927618	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=263/1793, ttl=43 (request in 4597)
4606	339.824429	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=264/2049, ttl=128 (reply in 4607)
4607	339.939868	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=264/2049, ttl=43 (request in 4606)
4611	340.843251	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=265/2305, ttl=128 (reply in 4612)
4612	340.960758	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=265/2305, ttl=43 (request in 4611)

4 Figura .Captura wireshark ping a gaia

3.2. Describir cómo filtrar por el protocolo ICMP para analizar los paquetes Echo ping request y Echo ping reply

En este caso modificaremos el filtro de wireshark poniendo solamente ICMP , Cabe destacar que si quisiéramos ver los paquetes ICMP de gaia.cs.umass.edu habría que poner que coincidirían con la figura anterior.

Captura de wireshark con filtro ICMP

No.	Time	Source	Destination	Protocol	Length	Info
4594	337.795717	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=262/1537, ttl=128 (reply in 4595)
4595	337.911928	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=262/1537, ttl=43 (request in 4594)
4597	338.811959	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=263/1793, ttl=128 (reply in 4598)
4598	338.927618	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=263/1793, ttl=43 (request in 4597)
4606	339.824429	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=264/2049, ttl=128 (reply in 4607)
4607	339.939868	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=264/2049, ttl=43 (request in 4606)
4611	340.843251	192.168.0.18	128.119.245.12	ICMP	74	Echo (ping) request id=0x0001, seq=265/2305, ttl=128 (reply in 4612)
4612	340.960758	128.119.245.12	192.168.0.18	ICMP	74	Echo (ping) reply id=0x0001, seq=265/2305, ttl=43 (request in 4611)
6253	588.278528	192.168.0.18	8.247.223.126	ICMP	106	Echo (ping) request id=0x0001, seq=266/2561, ttl=3 (no response found!)
6337	592.219010	192.168.0.18	8.247.223.126	ICMP	106	Echo (ping) request id=0x0001, seq=267/2817, ttl=3 (no response found!)
6345	596.218889	192.168.0.18	8.247.223.126	ICMP	106	Echo (ping) request id=0x0001, seq=268/3073, ttl=3 (no response found!)
6368	600.218673	192.168.0.18	8.247.223.126	ICMP	106	Echo (ping) request id=0x0001, seq=269/3329, ttl=4 (no response found!)
6369	600.236452	4.68.74.133	192.168.0.18	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
6370	600.236957	192.168.0.18	8.247.223.126	ICMP	106	Echo (ping) request id=0x0001, seq=270/3585, ttl=5 (reply in 6371)
6371	600.254271	8.247.223.126	192.168.0.18	ICMP	106	Echo (ping) reply id=0x0001, seq=270/3585, ttl=60 (request in 6370)

5 Figura .Captura wireshark filtro ICMP

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctico con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	-
Capa de Transporte	-
Capa de Red	ICMP, IPv4
Capa de Enlace de Datos	Ethernet II*
Capa Física	

*Por defecto, si usas una NIC inalámbrica tipo Wi-Fi, Wireshark representa la trama Wi-Fi (protocolo IEEE802.11) como una trama del protocolo Ethernet II para normalizar su representación.

Y las conclusiones para este contenido son:

1. Wireshark permite capturar tramas y paquetes ICMP si se realiza un ping (conexión)
2. Wireshark Es capaz de decodificar y analizar el contenido de datos de protocolo PDU y ICMP
3. Gracias a la dirección MAC podemos observar el destino y origen inmediatos.

CP04: Describir con Wireshark una resolución DNS a través de la captura de los datagramas UDP correspondientes.

1. Diagrama de topología (Figura 4)

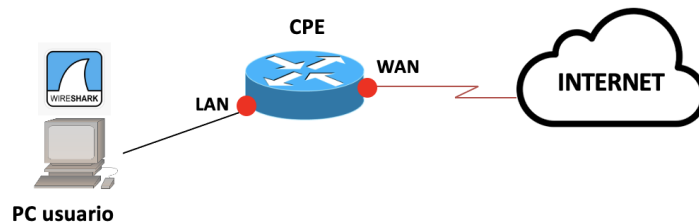


Figura 4. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida y con la herramienta software Wireshark instalada.

2.1 Desarrollo de la tarea

Vamos a utilizar wireshark para capturar paquetes de una resolución DNS entre cliente y servidor.

Antes debemos habilitar en nuestra configuración de dispositivo las propiedades avanzadas Configuración->Red e internet->Wifi->propiedades->y marcar las opciones de dirección IP automática y DNS automática

3. Desarrollo

3.1. Mostrar cómo liberar la caché DNS.

Usando el comando `ipconfig/flushdns`

3.2. Capturar y guardar con Wireshark la realización de una resolución DNS con el comando **nslookup** para el nombre FQDN `gaia.cs.umass.edu`

Primero dentro de wireshark y wifi pondremos el siguiente filtro:
`ip.addr==direcciónIPdelPC&&dns.qry.name==gaia.cs.umass.edu`
`ip.addr== 192.168.0.18&&dns.qry.name==gaia.cs.umass.edu`

Después abriremos una ventana en la cmd y podremos `nslookup gaia.cs.umass.edu`

No.	Time	Source	Destination	Protocol	Length	Info
26	13.581619	192.168.0.18	192.168.0.1	DNS	77	Standard query 0x0002 A gaia.cs.umass.edu
28	13.600677	192.168.0.1	192.168.0.18	DNS	93	Standard query response 0x0002 A gaia.cs.umass.edu A 128.119.245.12
29	13.612324	192.168.0.18	192.168.0.1	DNS	77	Standard query 0x0003 AAAA gaia.cs.umass.edu
30	13.637514	192.168.0.1	192.168.0.18	DNS	130	Standard query response 0x0003 AAAA gaia.cs.umass.edu SOA unix1.cs.umass.edu

6 Figura . Captura de wireshark para resolución DNS de gaia

Podemos observar que la primera trama será la consulta DNS y que la IP destino no coincide con la del Gateway predeterminado pero sí la dirección MAC destino
Y en la trama 2 dentro del campo Domain Name System->Answers podremos encontrar la respuesta con la dirección IP como muestra la siguiente figura

Domain Name System (response)
Transaction ID: 0x0002
Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
Queries
Answers
gaia.cs.umass.edu: type A, class IN, addr 128.119.245.12
Name: gaia.cs.umass.edu
Type: A (Host Address) (1)
Class: IN (0x0001)
Time to live: 15156 (4 hours, 12 minutes, 36 seconds)
Data length: 4
Address: 128.119.245.12

7 Figura . Captura de parámetros DNS wireshark

Para acceder a la información del puerto destino nos dirigimos a la 2ª trama con la respuesta y en UDP lo encontraremos como se muestra en la figura

User Datagram Protocol, Src Port: 53, Dst Port: 54979
Source Port: 53

8 Figura . Captura de parámetros UDP wireshark

3.3. Describir cómo filtrar por el nombre FQDN y la dirección lógica del PC en un filtro combinado para analizar los paquetes de la consulta DNS capturados

Aplicar un filtro por el protocolo DNS:dns

Filtrar por el nombre FQDN en la consulta DNS: dns.qry.name == gaia.cs.umass.edu

Filtrar por la dirección IP lógica (source IP) desde donde se hizo la consulta:
ip.src == 192.168.0.18

Combinar los filtros con un AND (&&):

`ip.addr== 192.168.0.18&&dns.qry.name==gaia.cs.umass.edu`

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctico con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	
Capa de Transporte	
Capa de Red	
Capa de Enlace de Datos	Ethernet II*
Capa Física	

*Por defecto, si usas una NIC inalámbrica tipo Wi-Fi, Wireshark representa la trama Wi-Fi (protocolo IEEE802.11) como una trama del protocolo Ethernet II para normalizar su representación.

Y las conclusiones para este contenido son:

1. Para poder ver las consultas DNS se necesita vaciar el historial DNS con `ipconfig/flushdns`
2. Fases para una resolución DNS:
 - El cliente DNS (como `nslookup`) hace una consulta recursiva al servidor DNS configurado preguntando por el FQDN.
 - El servidor DNS comienza la resolución solicitando la IP del dominio "edu" al servidor raíz.
 - El servidor raíz responde indicando qué servidor autoritativo gestiona "edu" (por ejemplo `ns.edu.gov`).
 - El servidor DNS ahora pregunta a `ns.edu.gov` por el dominio "umass.edu".
 - `ns.edu.gov` responde indicando qué servidor es autoritativo para "umass.edu" (por ejemplo `ns1.umass.edu`).
 - Entonces el servidor DNS le pregunta a `ns1.umass.edu` por el registro "gaia.cs.umass.edu".
 - Finalmente, `ns1.umass.edu` responde con la IP asociada a `gaia.cs.umass.edu`.
 - El servidor DNS devuelve esta IP al cliente `nslookup` como respuesta a la consulta inicial.
3. El puerto de destino no será siempre el mismo y dependerá de cuales se encuentren disponibles

CP05: Describir con Wireshark los mensajes de solicitud y respuesta para una comunicación HTTP.

1. Diagrama de topología (Figura 5)

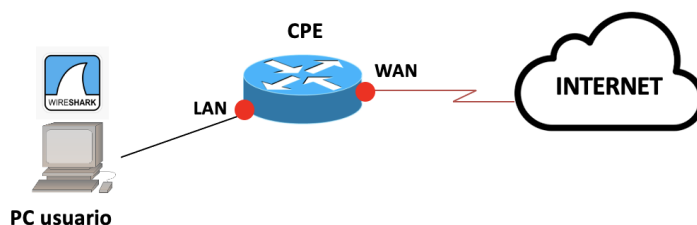


Figura 5. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida y con la herramienta software Wireshark instalada.

2.1 Desarrollo de la tarea

Vamos a capturar paquetes con datos HTTP entre un cliente y un servidor web.

Para ello necesitaremos la dirección IP del servidor y el programa wireshark, para obtener la dirección ip del servidor usaremos: `nslookup gaia.cs.umass.edu` en este caso 128.119.245.12.

Ahora abrimos el navegador y accedemos al sitio web: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

3. Desarrollo

3.1. Capturar y guardar una sesión HTTP para la URI <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

Ahora abriremos wireshark y accedemos a la red wifi y aplicamos el siguiente filtro

`http&&ip.addr==128.119.245.12`

y nos dará como resultado el mostrado en la figura de abajo.

La imagen muestra la interfaz de Wireshark con el filtro de captura `http&&ip.addr==128.119.245.12` aplicado. Se muestran cuatro paquetes de HTTP:

No.	Time	Source	Destination	Protocol	Length	Info
75381	3046.924823	192.168.0.18	128.119.245.12	HTTP	535	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
75405	3047.043120	128.119.245.12	192.168.0.18	HTTP	559	HTTP/1.1 200 OK (text/html)
75407	3047.514252	192.168.0.18	128.119.245.12	HTTP	505	GET /favicon.ico HTTP/1.1
75408	3047.632098	128.119.245.12	192.168.0.18	HTTP	538	HTTP/1.1 404 Not Found (text/html)

9 Figura . Captura wireshark filtro HTTP y server IP

A continuación dentro de wireshark haremos click en el cuadrado rojo para dejar de capturar y en archivo guardaremos la trama , como muestro en la siguiente figura



10 Figura . Captura de wireshark guardado archivo

3.2. Describir cómo filtrar por el protocolo HTTP y la dirección lógica del servidor Web en un filtro combinado para analizar los paquetes de la sesión HTTP capturados

Aplicar un filtro por el protocolo HTTP:**http**

Filtrar por la ip del servidor al que accederemos: **ip.addr==128.119.245.12**

Combinar los filtros con un AND (&&):

http&&ip.addr==128.119.245.12

3.3. Describir información relevante de los encabezados de la solicitud (incluir la línea de solicitud) y respuesta (incluir línea de estado y tamaño del archivo) HTTP del apartado 3.1

Primera trama (GET solicitud):

Es una solicitud GET al recurso /wireshark-labs/HTTP-wireshark-file3.html

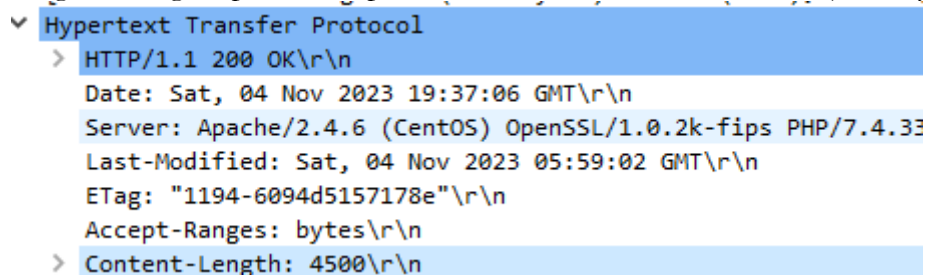
Sería un pedido al servidor por ese archivo HTML

Segunda trama (200 OK respuesta):

Es una respuesta HTTP con código 200 OK, indicando que el recurso fue encontrado exitosamente

El Content-Type es text/html, por lo que devuelve un documento HTML

En la siguiente figura podemos apreciar el tamaño del archivo HTML(4500 bytes)

A screenshot of a Wireshark packet capture. The packet list on the left shows a selected packet of type 'Hypertext Transfer Protocol'. The packet details pane on the right shows the following fields: 'HTTP/1.1 200 OK\r\n', 'Date: Sat, 04 Nov 2023 19:37:06 GMT\r\n', 'Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33', 'Last-Modified: Sat, 04 Nov 2023 05:59:02 GMT\r\n', 'ETag: "1194-6094d5157178e"\r\n', 'Accept-Ranges: bytes\r\n', and 'Content-Length: 4500\r\n'. The 'Content-Length' field is highlighted in blue.

```
▼ Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Sat, 04 Nov 2023 19:37:06 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33
    Last-Modified: Sat, 04 Nov 2023 05:59:02 GMT\r\n
    ETag: "1194-6094d5157178e"\r\n
    Accept-Ranges: bytes\r\n
    > Content-Length: 4500\r\n
```

11 Figura . Captura parámetro HTTP wireshark

Tercera trama (GET solicitud):

Nueva solicitud GET, en este caso para el archivo favicon.ico

Está pidiendo el ícono del sitio web

Cuarta trama (404 Not Found respuesta):

Respuesta 404 indicando que el recurso favicon.ico no fue encontrado

Nuevamente el Content-Type es text/html devolviendo el documento

El cliente recibe el recurso HTML en la segunda trama 200 OK, las tramas 3 y se dan porque el navegador web del cliente está tratando de descargar el ícono favicon.ico del sitio web, para mostrar el ícono junto a la URL.

Como ese ícono no existe en el servidor, este devuelve un error 404. Pero el navegador intenta descargarlo de todas formas cada vez que accede a un sitio.

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctico con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	
Capa de Transporte	
Capa de Red	
Capa de Enlace de Datos	Ethernet II*
Capa Física	

*Por defecto, si usas una NIC inalámbrica tipo Wi-Fi, Wireshark representa la trama Wi-Fi (protocolo IEEE802.11) como una trama del protocolo Ethernet II para normalizar su representación.

Y las conclusiones para este contenido son:

1. Se puede extraer información útil de los encabezados HTTP, como el Content-Type y Content-Length para conocer el contenido devuelto.
2. El código 200 OK indica que la solicitud del cliente fue exitosa y el recurso solicitado fue encontrado y devuelto por el servidor.
3. El código 404 Not Found representa que el recurso pedido no existe o no está disponible, lo cual se observa en la fallida solicitud al favicon.ico.

CP06: Seguimiento de las conexiones TCP en un PC mediante la herramienta netstat.

1. Diagrama de topología (Figura 6)

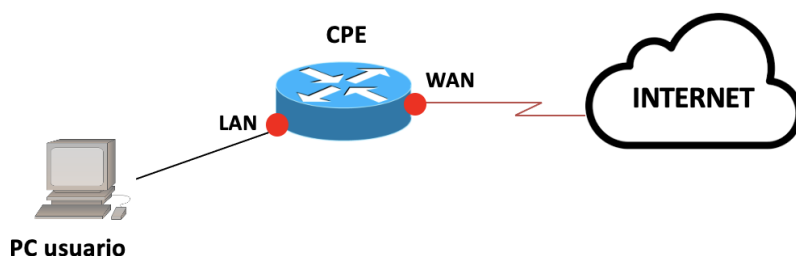


Figura 6. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida y con la herramienta software Wireshark instalada.

2.1 Desarrollo de la tarea

En esta tarea se explicarán las diversas funcionalidades del comando netstat así como para que sirven y como mostrar el contenido que queremos a través de las distintas opciones que ofrece este comando

3. Desarrollo

3.1. Completar la siguiente tabla con una breve descripción de la utilidad de los siguientes parámetros del comando netstat

Parámetros	Breve descripción
-a	Muestra todas las conexiones y puertos abiertos en escucha.
-n	Muestra las direcciones y puertos numéricos, no intenta resolver nombres.
-f	Muestra nombres FQDN completos
-o	Muestra el PID (process ID) de cada conexión.
-b	Muestra el nombre del programa que cuenta con la conexión abierta.
-q	Muestra todas las conexiones, puertos de escucha y puertos TCP enlazados que no sean para la escucha
-e / -s	Muestra estadísticas
-p <protocolo>	Filtra las conexiones por un protocolo específico como TCP o UDP.
<intervalo de tiempo>	Actualiza la salida cada x segundos para ver

	conexiones activas en tiempo real.
--	------------------------------------

Ejemplo de uso

`netstat -b -n -o -p TCP 1` (se pueden agrupar así `netstat -bnop TCP 1`)

Muestra las conexiones TCP (-p TCP)

Incluye el PID y nombre de programa (-o)

Muestra direcciones IP numéricas en lugar de nombres (-n)

Muestra el nombre de programa en cada conexión (-b)

Se actualiza cada segundo (1)

Para saber las opciones para el comando usar `netstat -h`

3.2. Describir cómo filtrar la salida del comando netstat con el comando findstr

El comando `findstr` en Windows nos permite filtrar la salida del comando `netstat` para mostrar solo líneas que coincidan con una cadena de búsqueda específica.

Ejemplos

Para filtrar la salida de `netstat` y ver solo conexiones TCP->`netstat | findstr /i TCP`

(se usa el parámetro `-i` para mostrar buscar tanto con mayúsculas como con minúsculas)

Véase en la siguiente figura

```
C:\Users\Pc>netstat | findstr /i TCP
TCP    192.168.0.18:61426    20.54.37.64:https    ESTABLISHED
TCP    192.168.0.18:61463    52.111.231.2:https    ESTABLISHED
```

12 Figura . Captura de cmd netstat TCP

3.3. Describir para qué sirve el uso del comando tasklist en combinación con netstat

El comando `tasklist` en combinación con `netstat` sirve para asociar las conexiones de red activas con los procesos o aplicaciones que las han originado.

Se puede utilizar para obtener el PID (ID de proceso) de una aplicación con `tasklist`, luego filtrar `netstat` para solo ver conexiones de ese PID. Esto muestra las conexiones de un programa específico.

Ejemplo.

Identificamos el nombre de la aplicación, por ejemplo el navegador Chrome

`tasklist | findstr chrome.exe`

Comando en cmd

```
C:\Users\Pc>tasklist | findstr chrome.exe
chrome.exe             12800 Console             8      200.800 KB
chrome.exe             15432 Console             8        9.380 KB
chrome.exe             1180  Console             8      118.128 KB
```

13 Figura . Captura de cmd tasklist navegador

4. Conclusiones

Y las conclusiones para este contenido son:

1. Netstat es una herramienta muy útil para obtener información en SO Windows permite identificar puertos abiertos, programas asociados, protocolos
2. Es importante ejecutar netstat con permisos elevados y verificar compatibilidad entre versiones de Windows para garantizar que las opciones y filtros funcionen correctamente.
3. Uso de los siguientes comandos en SO windows
netstat - Comando para ver estadísticas y conexiones de red en Windows.
findstr - Comando para buscar/filtrar cadenas en la salida de otros comandos.
tasklist - Muestra los procesos o aplicaciones en ejecución en el sistema.

CP07: Analizar con Wireshark los segmentos utilizados durante la negociación de una conexión TCP.

1. Diagrama de topología (Figura 7)

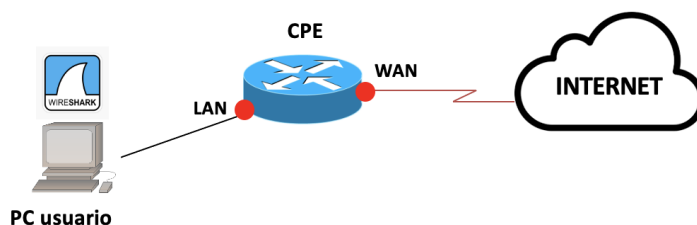


Figura 7. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida y con la herramienta software Wireshark instalada.

2.1 Desarrollo de la tarea

En esta tarea veremos y analizaremos las distintas fases y las distintas variables y opciones de una negociación TCP entre un cliente y un servidor, en este caso el cliente le pide al servidor un archivo HTML a descargar.

3. Desarrollo

3.1. Recuperar la captura de la sesión HTTP para la URI <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

Abrimos la captura anteriormente realizada y nos aseguramos que tenga puesto viene el filtro:

`http&&ip.addr==128.119.245.12`

Siendo 128.119.245.12 la dirección ip de gaia

Tal y como se muestra en la siguiente figura

No.	Time	Source	Destination	Protocol	Length	Info
346	19.239723	192.168.0.18	128.119.245.12	HTTP	622	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
356	19.358991	128.119.245.12	192.168.0.18	HTTP	559	HTTP/1.1 200 OK (text/html)
359	19.827586	192.168.0.18	128.119.245.12	HTTP	505	GET /favicon.ico HTTP/1.1
379	22.777822	128.119.245.12	192.168.0.18	HTTP	538	HTTP/1.1 404 Not Found (text/html)

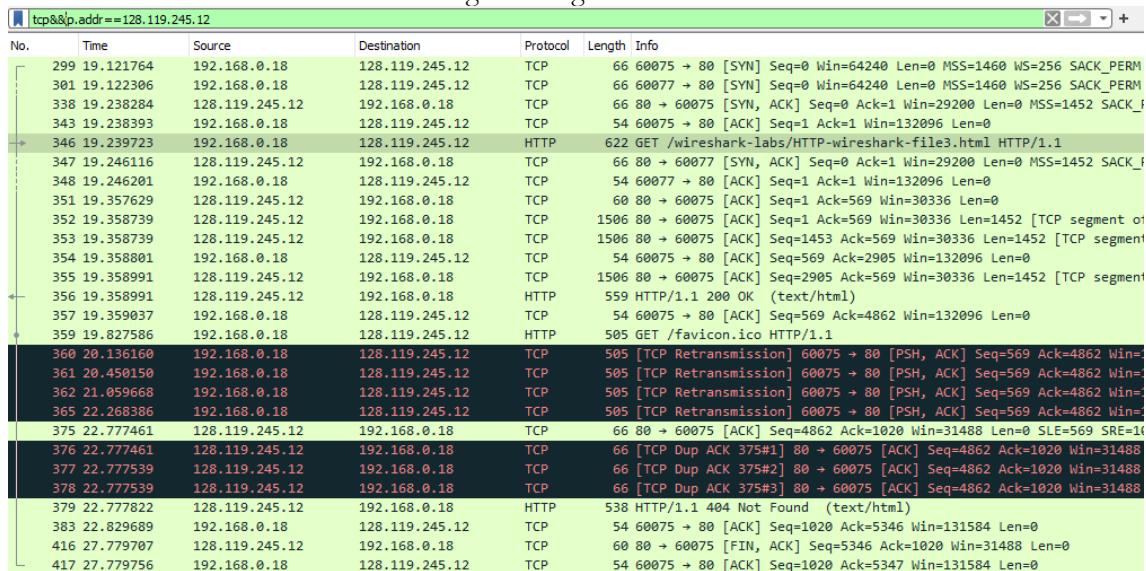
14 Figura . Captura wireshark filtro HTTP y server IP

3.2. Describir cómo filtrar por el protocolo TCP y la dirección lógica del servidor Web en un filtro combinado para analizar los segmentos de la sesión HTTP

Para mostrar las tramas del protocolo TCP debemos cambiar el filtro y poner el siguiente:

`tcp&&ip.addr==128.119.245.12`

Mostrando un resultado como el de la siguiente figura



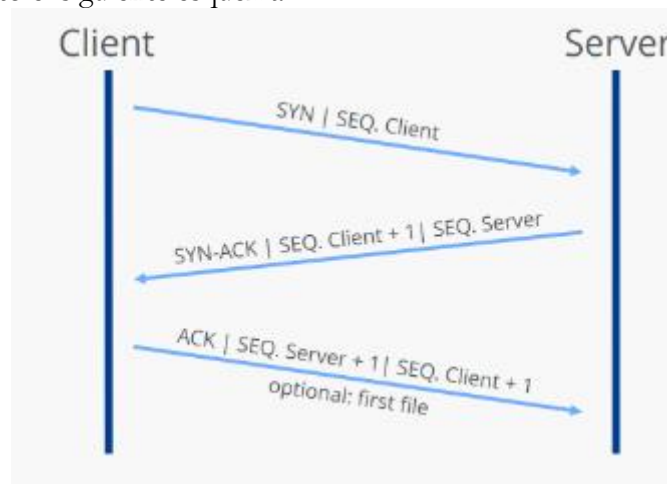
No.	Time	Source	Destination	Protocol	Length	Info
299	19.121764	192.168.0.18	128.119.245.12	TCP	66	60075 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
301	19.122306	192.168.0.18	128.119.245.12	TCP	66	60077 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
338	19.238284	128.119.245.12	192.168.0.18	TCP	66	80 → 60075 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK_f
343	19.238393	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
346	19.239723	192.168.0.18	128.119.245.12	HTTP	622	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
347	19.246116	128.119.245.12	192.168.0.18	TCP	66	80 → 60077 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK_f
348	19.246201	192.168.0.18	128.119.245.12	TCP	54	60077 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
351	19.357629	128.119.245.12	192.168.0.18	TCP	60	80 → 60075 [ACK] Seq=1 Ack=569 Win=30336 Len=0
352	19.358739	128.119.245.12	192.168.0.18	TCP	1506	80 → 60075 [ACK] Seq=1 Ack=569 Win=30336 Len=1452 [TCP segment or
353	19.358739	128.119.245.12	192.168.0.18	TCP	1506	80 → 60075 [ACK] Seq=1453 Ack=569 Win=30336 Len=1452 [TCP segment
354	19.358801	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=569 Ack=2905 Win=132096 Len=0
355	19.358991	128.119.245.12	192.168.0.18	TCP	1506	80 → 60075 [ACK] Seq=2905 Ack=569 Win=30336 Len=1452 [TCP segment
356	19.358991	128.119.245.12	192.168.0.18	HTTP	559	HTTP/1.1 200 OK (text/html)
357	19.359037	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=569 Ack=4862 Win=132096 Len=0
359	19.827586	192.168.0.18	128.119.245.12	HTTP	505	GET /favicon.ico HTTP/1.1
360	20.136160	192.168.0.18	128.119.245.12	TCP	505	[TCP Retransmission] 60075 → 80 [PSH, ACK] Seq=569 Ack=4862 Win=
361	20.450150	192.168.0.18	128.119.245.12	TCP	505	[TCP Retransmission] 60075 → 80 [PSH, ACK] Seq=569 Ack=4862 Win=
362	21.059668	192.168.0.18	128.119.245.12	TCP	505	[TCP Retransmission] 60075 → 80 [PSH, ACK] Seq=569 Ack=4862 Win=
365	22.268386	192.168.0.18	128.119.245.12	TCP	505	[TCP Retransmission] 60075 → 80 [PSH, ACK] Seq=569 Ack=4862 Win=
375	22.777461	128.119.245.12	192.168.0.18	TCP	66	80 → 60075 [ACK] Seq=4862 Ack=1020 Win=31488 Len=0 SLE=569 SRE=1
376	22.777461	128.119.245.12	192.168.0.18	TCP	66	[TCP Dup ACK 375#1] 80 → 60075 [ACK] Seq=4862 Ack=1020 Win=31488
377	22.777539	128.119.245.12	192.168.0.18	TCP	66	[TCP Dup ACK 375#2] 80 → 60075 [ACK] Seq=4862 Ack=1020 Win=31488
378	22.777539	128.119.245.12	192.168.0.18	TCP	66	[TCP Dup ACK 375#3] 80 → 60075 [ACK] Seq=4862 Ack=1020 Win=31488
379	22.777822	128.119.245.12	192.168.0.18	HTTP	538	HTTP/1.1 404 Not Found (text/html)
383	22.829689	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=1020 Ack=5346 Win=131584 Len=0
416	27.779707	128.119.245.12	192.168.0.18	TCP	60	80 → 60075 [FIN, ACK] Seq=5346 Ack=1020 Win=31488 Len=0
417	27.779756	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=1020 Ack=5347 Win=131584 Len=0

15 Figura . Captura wireshark filtro TCP y server IP

Podemos observar que la conexión TCP se inicia con una petición HTTP Get de la trama nº 346 Es entonces cuando empieza la negociación que se explicara en la siguiente tarea

3.3. Describir información relevante de los encabezados de los tres segmentos de la negociación TCP en tres fases y cómo reconocer qué corresponden con la sesión HTTP.

Antes de iniciar esta parte es necesario realizar Edit > Preferences y dentro de Protocols > TCP > Relative sequence numbers para obtener los números de secuencia relativos
Teniendo en mente el siguiente esquema



16 Figura . Captura esquema negociación TCP

Una vez identificado el inicio de la negociación podemos identificar la primera fase por el número de secuencia que suele empezar en 0 y el siguiente sería 1 y por el resto de flags que están a 0 excluyendo el SYN, y nótese también en la direcciones ip origen y destino

El cliente envía un segmento TCP con el flag SYN activado y un número de secuencia inicial aleatorio. Esto indica que desea abrir una conexión.

Figura de la fase 1

299	19.121764	192.168.0.18	128.119.245.12	TCP	66 60075 → 80 [SYN] Seq=0
301	19.122306	192.168.0.18	128.119.245.12	TCP	66 60077 → 80 [SYN] Seq=0
338	19.238284	128.119.245.12	192.168.0.18	TCP	66 80 → 60075 [SYN, ACK] Seq=0 Ack=1
343	19.238393	192.168.0.18	128.119.245.12	TCP	54 60075 → 80 [ACK] Seq=1
346	19.239723	192.168.0.18	128.119.245.12	HTTP	622 GET /wireshark-labs/HT
347	19.246116	128.119.245.12	192.168.0.18	TCP	66 80 → 60077 [SYN, ACK] Seq=0 Ack=1
348	19.246201	192.168.0.18	128.119.245.12	TCP	54 60077 → 80 [ACK] Seq=1
351	19.357629	128.119.245.12	192.168.0.18	TCP	60 80 → 60075 [ACK] Seq=1
352	19.358739	128.119.245.12	192.168.0.18	TCP	1506 80 → 60075 [ACK] Seq=1
353	19.358739	128.119.245.12	192.168.0.18	TCP	1506 80 → 60075 [ACK] Seq=1
354	19.358801	192.168.0.18	128.119.245.12	TCP	54 60075 → 80 [ACK] Seq=5
355	19.358991	128.119.245.12	192.168.0.18	TCP	1506 80 → 60075 [ACK] Seq=2
356	19.358991	128.119.245.12	192.168.0.18	HTTP	559 HTTP/1.1 200 OK (text/html)

Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 1842797174
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1000 = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
000. = Reserved: Not set
...0 = Accurate ECN: Not set
.... 0... = Congestion Window Reduced: Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...0 = Acknowledgment: Not set
....0... = Push: Not set
....0.. = Reset: Not set
....1. = Syn: Set

17 Figura . Captura wireshark negociación TCP fase 1

El servidor responde con un segmento que tiene los flags SYN y ACK activados. El flag SYN confirma que acepta abrir la conexión y el flag ACK está activado para confirmar el numero de secuencia del cliente.

338	19.238284	128.119.245.12	192.168.0.18	TCP	66 80 → 60075 [SYN, ACK] Seq=0 Ack=1
343	19.238393	192.168.0.18	128.119.245.12	TCP	54 60075 → 80 [ACK] Seq=1 Ack=1 Win=1
346	19.239723	192.168.0.18	128.119.245.12	HTTP	622 GET /wireshark-labs/HTTP-wireshark
347	19.246116	128.119.245.12	192.168.0.18	TCP	66 80 → 60077 [SYN, ACK] Seq=0 Ack=1
348	19.246201	192.168.0.18	128.119.245.12	TCP	54 60077 → 80 [ACK] Seq=1 Ack=1 Win=1
351	19.357629	128.119.245.12	192.168.0.18	TCP	60 80 → 60075 [ACK] Seq=1 Ack=569 Win=1
352	19.358739	128.119.245.12	192.168.0.18	TCP	1506 80 → 60075 [ACK] Seq=1 Ack=569 Win=1
353	19.358739	128.119.245.12	192.168.0.18	TCP	1506 80 → 60075 [ACK] Seq=1453 Ack=569
354	19.358801	192.168.0.18	128.119.245.12	TCP	54 60075 → 80 [ACK] Seq=569 Ack=2905
355	19.358991	128.119.245.12	192.168.0.18	TCP	1506 80 → 60075 [ACK] Seq=2905 Ack=569
356	19.358991	128.119.245.12	192.168.0.18	HTTP	559 HTTP/1.1 200 OK (text/html)

Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 564197105
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 1842797175
1000 = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
000. = Reserved: Not set
...0 = Accurate ECN: Not set
.... 0... = Congestion Window Reduced: Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...1 = Acknowledgment: Set
....0... = Push: Not set
....0.. = Reset: Not set
....1. = Syn: Set

18 Figura . Captura wireshark negociación TCP fase 2

El cliente confirma la conexión respondiendo con un segmento ACK, acusando recibo del SYN+ACK del servidor.

Nótese en la figura como es solo el flag ACK el que se encuentra activo

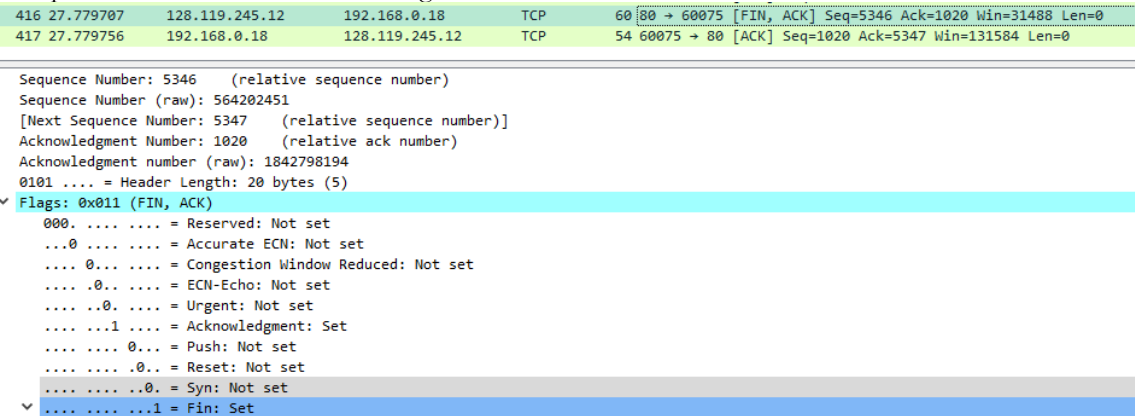
343	19.238393	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=1 Ack=1
346	19.239723	192.168.0.18	128.119.245.12	HTTP	622	GET /wireshark-labs/HTTP-wire
347	19.246116	128.119.245.12	192.168.0.18	TCP	66	80 → 60077 [SYN, ACK] Seq=0 A
348	19.246201	192.168.0.18	128.119.245.12	TCP	54	60077 → 80 [ACK] Seq=1 Ack=1
351	19.357629	128.119.245.12	192.168.0.18	TCP	60	80 → 60075 [ACK] Seq=1 Ack=56
352	19.358739	128.119.245.12	192.168.0.18	TCP	1506	80 → 60075 [ACK] Seq=1 Ack=56
353	19.358739	128.119.245.12	192.168.0.18	TCP	1506	80 → 60075 [ACK] Seq=1453 Acl
354	19.358801	192.168.0.18	128.119.245.12	TCP	54	60075 → 80 [ACK] Seq=569 Ack:
355	19.358991	128.119.245.12	192.168.0.18	TCP	1506	80 → 60075 [ACK] Seq=2905 Acl
356	19.358991	128.119.245.12	192.168.0.18	HTTP	559	HTTP/1.1 200 OK (text/html)

```

Sequence Number: 1      (relative sequence number)
Sequence Number (raw): 1842797175
[Next Sequence Number: 1      (relative sequence number)]
Acknowledgment Number: 1      (relative ack number)
Acknowledgment number (raw): 564197106
0101 .... = Header Length: 20 bytes (5)
Flags: 0x010 (ACK)
 000. .... = Reserved: Not set
...0 .... = Accurate ECN: Not set
.... 0... = Congestion Window Reduced: Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...1 = Acknowledgment: Set
.... .... 0... = Push: Not set
.... ..... 0.. = Reset: Not set
.... ..... .0. = Syn: Not set
.... ..... ...0 = Fin: Not set
    
```

19 Figura . Captura wireshark negociación TCP fase 3

Cabe destacar el ultimo segmento de la negociación TCP en el cual el servidor indica que ya se ha compartido los datos activando el flag FIN



20 Figura . Captura wireshark negociación TCP fase final

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctico con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	
Capa de Transporte	
Capa de Red	
Capa de Enlace de Datos	Ethernet II*
Capa Física	

*Por defecto, si usas una NIC inalámbrica tipo Wi-Fi, Wireshark representa la trama Wi-Fi (protocolo IEEE802.11) como una trama del protocolo Ethernet II para normalizar su representación.

Y las conclusiones para este contenido son:

- 1. Por lo general el numero de secuencia inicial se suele iniciar a 0
- 2. La primera trama del procedimiento TCP no sirve como acuse de recibo(ACK)
- 3. Si falla alguna fase, la conexión no se establece y se descarta después de algunos reintentos dependiendo del SO

CP08: Analizar con Wireshark los segmentos TCP de datos (PSH) correspondientes a una solicitud/respuesta HTTP.

1. Diagrama de topología (Figura 8)

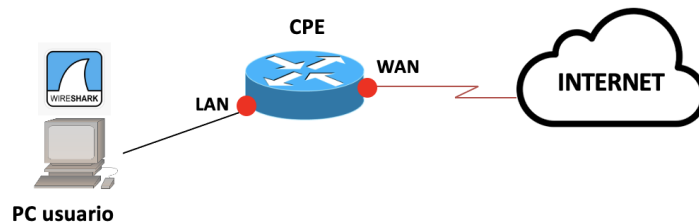


Figura 8. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC conectado a una red Wi-Fi distinta a la del Centro Universitario de Mérida y con la herramienta software Wireshark instalada.

2.1 Desarrollo de la tarea

En esta tarea se comprobará los parámetros wireshark cuando se captura la navegación a un sitio web haciendo incapie en TCP y HTTP en las peticiones GET y la Respuesta 200 ok para ver el msg dividido en segmentos ya que no cabe por la MSS.

3. Desarrollo

3.1. Recuperar la captura de la sesión HTTP para la URI <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

Primero accederemos al la url: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html> a través de nuestro navegador web

Después obtendremos la dirección ip del servidor con (Figura 21)

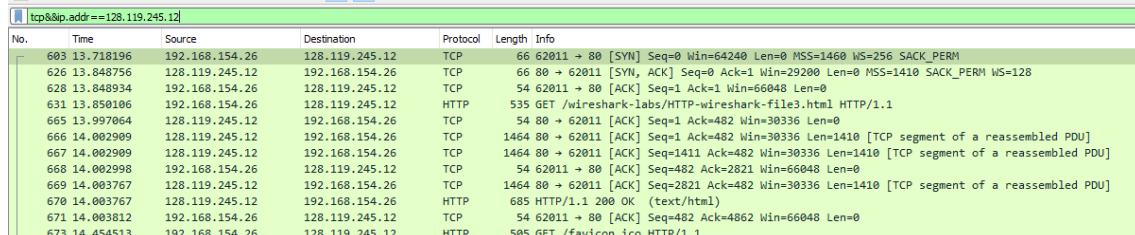
```
C:\Users\Pc>nslookup gaia.cs.umass.edu
Servidor: UnKnown
Address: 192.168.154.11

Respuesta no autoritativa:
Nombre: gaia.cs.umass.edu
Address: 128.119.245.12
```

Figura 21 .Comando nslookup

3.2. Describir cómo filtrar por el protocolo TCP y la dirección lógica del servidor Web en un filtro combinado para analizar los segmentos de la sesión HTTP

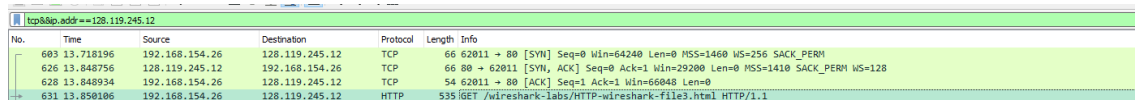
Y aplicaremos el filtro `tcp&&ip.addr==128.119.245.12` en wireshark (Figura22)



No.	Time	Source	Destination	Protocol	Length	Info
603	13.718196	192.168.154.26	128.119.245.12	TCP	66	62011 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
626	13.848756	128.119.245.12	192.168.154.26	TCP	66	80 → 62011 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1410 SACK_PERM WS=128
628	13.848934	192.168.154.26	128.119.245.12	TCP	54	62011 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
631	13.850106	192.168.154.26	128.119.245.12	HTTP	535	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
665	13.997064	128.119.245.12	192.168.154.26	TCP	54	80 → 62011 [ACK] Seq=1 Ack=482 Win=30336 Len=0
666	14.002909	128.119.245.12	192.168.154.26	TCP	1464	80 → 62011 [ACK] Seq=1 Ack=482 Win=30336 Len=1410 [TCP segment of a reassembled PDU]
667	14.002909	128.119.245.12	192.168.154.26	TCP	1464	80 → 62011 [ACK] Seq=1411 Ack=482 Win=30336 Len=1410 [TCP segment of a reassembled PDU]
668	14.002998	192.168.154.26	128.119.245.12	TCP	54	62011 → 80 [ACK] Seq=482 Ack=2821 Win=66048 Len=0
669	14.003767	128.119.245.12	192.168.154.26	TCP	1464	80 → 62011 [ACK] Seq=2821 Ack=482 Win=30336 Len=1410 [TCP segment of a reassembled PDU]
670	14.003767	128.119.245.12	192.168.154.26	HTTP	685	HTTP/1.1 200 OK (text/html)
671	14.003812	192.168.154.26	128.119.245.12	TCP	54	62011 → 80 [ACK] Seq=482 Ack=4862 Win=66048 Len=0
673	14.454513	192.168.154.26	128.119.245.12	HTTP	505	GET /favicon.ico HTTP/1.1

Figura 22 .Filtro tcp y http.

3.3. Describir información relevante de los encabezados del segmento que contiene la solicitud HTTP (Figura 23)



No.	Time	Source	Destination	Protocol	Length	Info
603	13.718196	192.168.154.26	128.119.245.12	TCP	66	62011 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
626	13.848756	128.119.245.12	192.168.154.26	TCP	66	80 → 62011 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1410 SACK_PERM WS=128
628	13.848934	192.168.154.26	128.119.245.12	TCP	54	62011 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
631	13.850106	192.168.154.26	128.119.245.12	HTTP	535	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1

Figura 23 .Petición Get completa.

Encabezado TCP(Figura 24):

Puertos de origen y destino: Identifican las aplicaciones de origen y destino.

Números de secuencia y acknowledgment: Indican la posición del paquete en la secuencia de datos y se utilizan para el control de flujo.

Longitud de la cabecera TCP: Indica la longitud de la cabecera TCP en palabras de 32 bits.

Banderas TCP: Pueden incluir SYN, ACK, PSH, FIN.

Solicitud HTTP

Método HTTP: GET.

Versión HTTP: HTTP/1.0, HTTP/1.1, etc.

Encabezados HTTP.

Cuerpo de la solicitud.

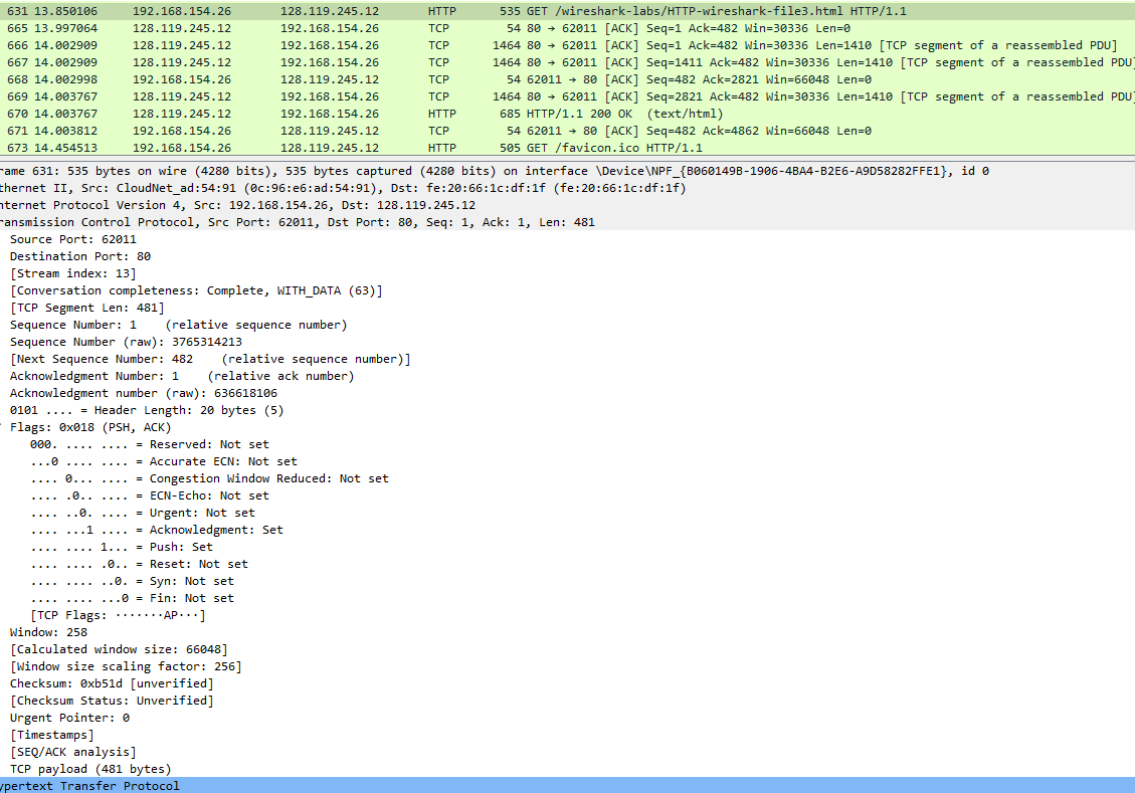


Figura 24 .Petición Get completa TCP.

El TCP Payload es de 481 bytes y podemos observar que se necesitó un total de 4 segmentos TCP para enviar la respuesta HTTP completa.

3.4. Describir información relevante de los encabezados del segmento que contiene la respuesta HTTP(Figura 25). Incluir cómo reconocer si fue necesario más de un segmento para enviar todo el mensaje de capa de aplicación.

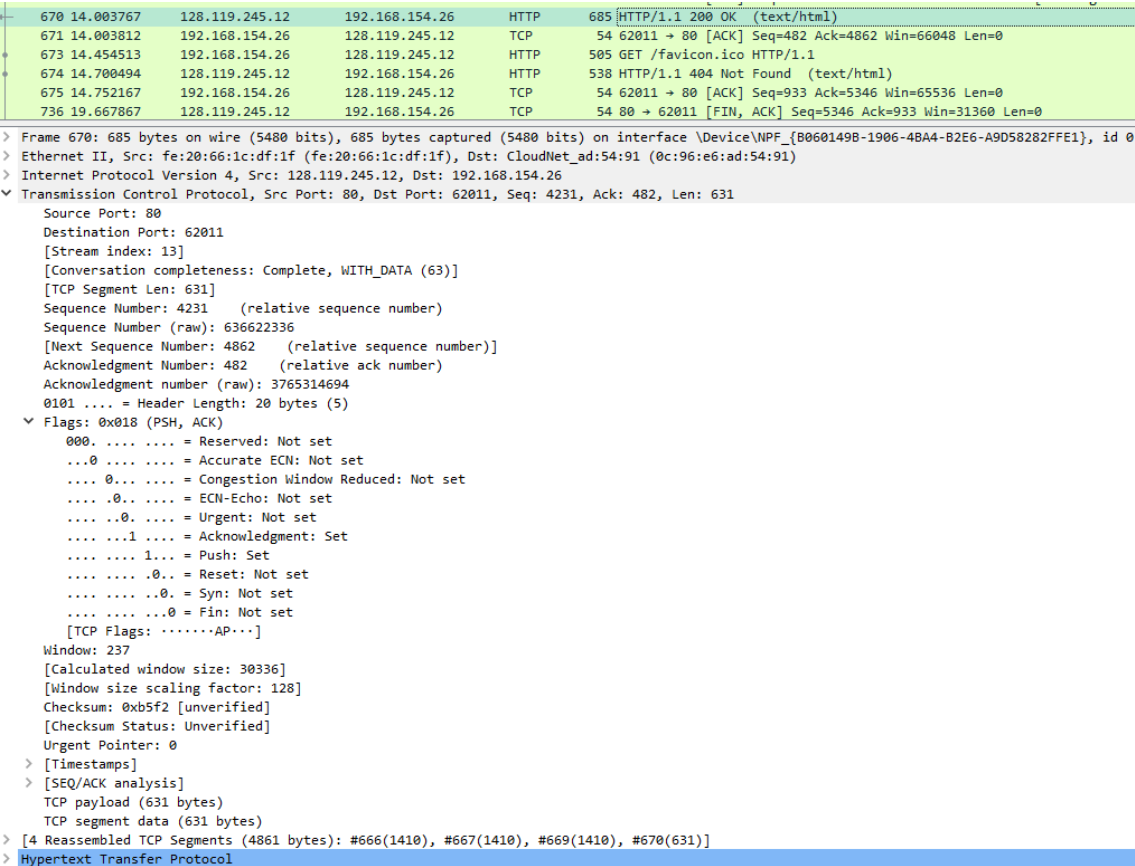


Figura 25 .Respuesta HTTP.

En esta trama podemos observar que solo tiene el indicador ACK, su número de secuencia es 4231 y el número de acuse de recibo es 482, este número debe coincidir con el número de bytes recibidos anteriormente + 1 (por lo que es correcto)

4. Conclusiones

Relaciona los protocolos que intervienen en desarrollo de este contenido práctico con las capas del modelo híbrido de Internet:

Modelo híbrido de Internet	Protocolos usados
Capa de Aplicación	
Capa de Transporte	
Capa de Red	
Capa de Enlace de Datos	
Capa Física	Ethernet II*

*Por defecto, si usas una NIC inalámbrica tipo Wi-Fi, Wireshark representa la trama Wi-Fi (protocolo IEEE802.11) como una trama del protocolo Ethernet II para normalizar su representación.

Cuaderno de Ingeniería. Curso 2023/2024

Marcos Folguera Rivera

Y las conclusiones para este contenido son:

1. El acuse de recibo es igual a los bytes enviados más uno.
2. PSH indica que los datos deben transmitirse a la capa de aplicación.
3. Si se envía una trama mayor que el mss establecido la información se divide en segmentos.

CP09: Usar la operación AND bit-a-bit para calcular la dirección de red/subred de una dirección IPv4 dada

1. Situación inicial

Este contenido no necesita usar ningún comando o herramienta software específica.

2. Desarrollo

2.1. Desarrollo de la tarea

En esta tarea vamos a calcular la dirección de red y de subred utilizando la operación bit and bit.

2.1. Explica la utilización de la operación binaria AND a partir de algún ejemplo(Figura26).

Descripción	Decimal	Binario
Dirección IP	192.168.68. 210	192.168.68. 11010010
Máscara de subred	255.255.255. 128	255.255.255. 01111111

Figura 26 .Tabla Direcciones IP Y SM.

Primero nos fijamos en la Mascara de subred y todos los octetos que no estén a 255 son los que tendremos que pasar a binario de la dirección IP y de la MAC ya que el resto al hacer la operación AND quedaran igual ya que $1 \text{ and } n = n$ siendo n el numero de la IP.

Ahora se hace la Operación and para obtener la dirección de subred->192.168.68.**82**

11010010

01111111

01010010 ->82 192.168.68.**82**

Clase C

Bits Subred=7

Subredes= $2^7 = 126$

Bits porción host=1

host por red/subred= $2^1 = 2$

Otros parámetros también útiles (Figura27):

-Dirección subred formato punteado=(en binario) IPv4 AND Mascara Subred

-Numero de bits de Msubred(contar el numero de 1 de la mascara sub red)=n

-Para calcular las subredes= $2^{(\text{numero de 1 en en la parte de subred final})}$

-Numero de bits de porción de host = numero de 0 en en la parte de subred final

-Cuantos host por subred(direcciones IP usables para dispositivos)= $2^{(\text{numero de 0 en en la parte de subred final})} - 2$.

Direccionamiento IPv4 legado (Classful IP addressing)

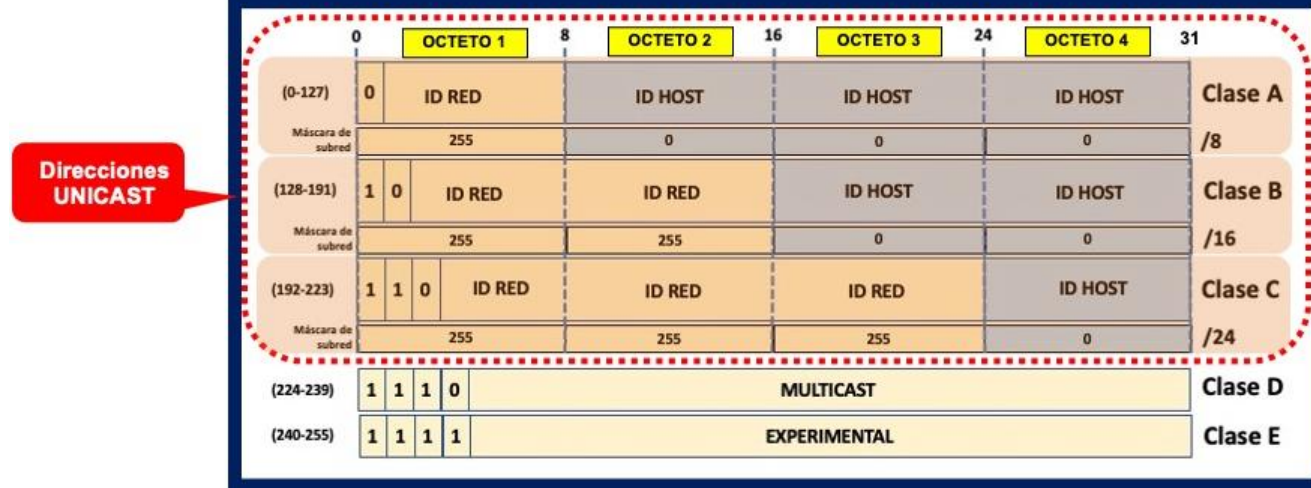


Figura 27 .Esquema Direccionamiento IP

2.2. Aplica la utilización de la operación binaria AND para determinar si dos hosts pertenecen a la misma subred según sus direcciones IP

IPA=192.168.1.18 IPB=192.168.1.33 Mascara Subred=255.255.255.240

IPA:11000000 10101000 00000001 00010010
 MR:11111111 11111111 11111111 11110000
 SUB1: 11000000 10101000 00000001 00010000

IPB:11000000 10101000 00000001 00100001
 MR:11111111 11111111 11111111 11110000
 SUB2:11000000 10101000 00000001 00100000

Ya que las subredes son distintas los 2 host no podrían comunicarse sin un router porque no se encuentran en la misma subred.

3. Conclusiones

Y las conclusiones para este contenido son:

1. -Si nos fijamos en la Mascara de subred y todos los octetos que no estén a 255 son los que tendremos que pasar a binario de la dirección IP y de la MAC ya que el

Cuaderno de Ingeniería. Curso 2023/2024

Marcos Folguera Rivera

resto al hacer la operación AND quedaran igual ya que $1 \text{ and } n = n$ siendo n el numero de la IP.

2. -Cuantos host por subred(direcciones IP usables para dispositivos) $= 2^{\text{(numero de 0 en en la parte de subred final)} - 2}$.
3. -Para calcular las subredes $= 2^{\text{(numero de 1 en en la parte de subred final)}}$.

CP10: Conocer la técnica de división en subredes de VLSM y la asignación estática de direcciones IPv4

1. Diagrama de topología (Figura 9)



Figura 9. Diagrama de topología

2. Situación inicial

Este contenido no necesita usar ningún comando o herramienta software específica.

2.1. Desarrollo de la tarea

En esta actividad vamos a completar la tabla de subredes de dos subredes con distinto número de host usando VLSM para así optimizar el direccionamiento.

3. Desarrollo

3.1. Describir la división en subredes IPv4 utilizando el proceso de máscara de longitud variable (VLSM).

Pasos para división en subredes VLSM

Subred A 120 host

Subred B 60 host

1) Ordenar de subredes de mayor a menor por número de host para Clase C 192.168.0.0/24

-Subred A 120 host $2^n - 2 = 120 \rightarrow n = 7$ si son 7 resto $8 - 7 = m$ bits de subred

 Son para múltiplos de 8

 Si fueran 10 serían $16 - 10 = 6 = m$

 Si fueran 20 $24 - 20 = 4 = m$

 Dirección de subred 192.168.0.00000000 bits de host 192.168.0.0

 Máscara Subred 255.255.255.10000000 m bits de subred 255.255.255.128

 Dir broadcast 192.168.0.01111111 \rightarrow 192.168.0.127 (se ponen los bits de host=1 en la DirSubred)

 Rango desde 192.168.0.1 (la siguiente a la DirSubred) hasta 192.168.0.126 (la anterior a la Dir Broadcast)

-Subred B 60 host $2^n - 2 = n \rightarrow n = 6$ si son 6 resto $8 - 6 = m = 2$ bits de subred

 Dirección de subred Se pone a partir de la anterior broadcast por lo que si fue 192.168.0.01111111 la siguiente será 192.168.0.10000000 \rightarrow 192.168.0.128

 Máscara Subred 255.255.255.11000000 m=2 bits de subred 255.255.255.192

 Dir broadcast 192.168.0.10111111 \rightarrow 192.168.0.191 (se ponen los bits de host=1 en la DirSubred)

Rango desde **192.168.0.129** (la siguiente a la DirSubred) hasta **192.168.0.190** (la anterior a la Dir Broadcast).

Nombre Red	Dirección de red/subred	Máscara de subred	Rango asignable	Dirección de broadcast
A	192.168.0.0	255.255.255.128	192.168.0.1-192.168.0.126	192.168.0.127
B	192.168.0.128	255.255.255.192	192.168.0.129-192.168.0.190	192.168.0.191

3.2. Describir la planificación de la asignación de direcciones IPv4 a los dispositivos finales e intermedios de una red

- Los hosts **PC1** y **PC2** usarán para la NIC la **primera dirección IP del rango asignable** en la subred correspondiente.
- El **Router1** utilizará **para cada una de sus interfaces** conectadas la **última dirección IP del rango asignable** en la subred correspondiente.

Para cada interface se le aplica la Mascara de Subred de la subred (igual que los PC).

Dispositivo	ID Interfaz	Dirección IP	Máscara de subred	Puerta de enlace <i>predeterminada</i>
Router1	Gi0/0	192.168.0.126	255.255.255.128	No aplicable
	Gi0/1	192.168.0.190	255.255.255.192	No aplicable

Cuaderno de Ingeniería. Curso 2023/2024

Marcos Folguera Rivera

PC1	NIC	192.168.0.1	255.255.255.128	192.168.0.126
PC2	NIC	192.168.0.129	255.255.255.192	192.168.0.190

4. Conclusiones

Y las conclusiones para este contenido son:

1. La dirección de broadcast se obtiene a partir de la de subred.
2. La dirección de Gateway por defecto de un ordenador coincide con la interfaz del router al que se conectan.
3. Hay que ordenar de subredes de mayor a menor por número de host.

CP11: Conectar los dispositivos de un diagrama de topología de red en Cisco Packet Tracer (PT)

1. Diagrama de topología (Figura 10)



Figura 10. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC con la herramienta de simulación de red Cisco Packet Tracer.

2.1. Desarrollo de la tarea

En esta actividad se conectarán los dispositivos que se encuentran en el diagrama de topología.

3. Desarrollo

3.1. Describir qué tipo de cableado se emplea para conectar un PC a un switch y cómo hacerlo en PT

Una vez realizada la tabla de direcciones y establecido las direcciones en cada uno de los PC se realizarán las conexiones.

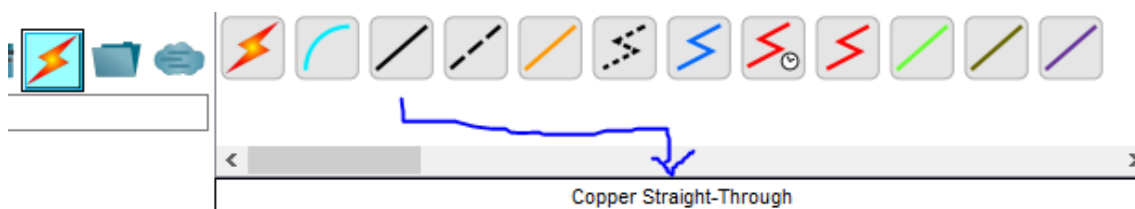


Figura 28 .Sección de cables PT

Entre un PC y Switch-> Usamos un Cable de cobre de conexión directa (Copper Straight-Through) (Figura 28) usando la interfaz de **switch** FastEthernet0/2 y la interfaz del **PC1** FastEthernet0 (Figura 29).

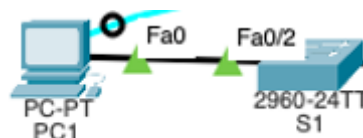


Figura 29 .Resultado en packet tracer conexión PC1-Switch1.

3.2. Describir qué tipo de cableado se emplea para conectar un switch a un router y cómo hacerlo en PT

Entre un Router y Switch-> Usamos en Cable de cobre de conexión directa (Copper Straight-Through) usando la interface de **switch** FastEthernet0/1 y la interface del **router1** FastEthernet0/0 (Figura 30).



Figura 30 . Resultado en packet tracer conexión R1-Switch 1.

3.3. Describir qué tipo de cableado se emplea para conectar dos routers entre ellos a través de una WAN tipo serial y cómo hacerlo en PT.

Entre dos router-> Usamos en Cable de cobre serie DCE (Serial DCE) (Figura 31). usando la interface de **R1** S0/0/0 y la interface del **R2** S0/0/0 (Figura 32).

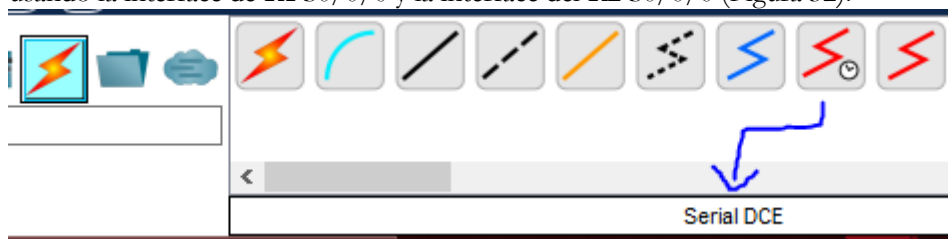


Figura 31 .Sección de cables PT.

nota:Se utiliza DCE para poder aplicarle un delay al router 1 con clock para simular mejor

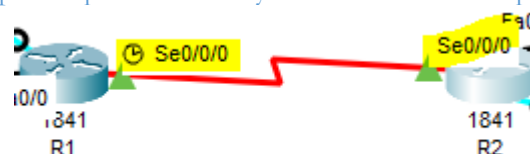


Figura 32 . Resultado en packet tracer conexión R1-R2.

3.4. Describir qué tipo de cableado se emplea para conectar un PC al puerto de consola de un router y cómo acceder al IOS para su administración en PT.

Entre dos router-> Usamos en consola(console)(Figura 33).
usando la interface de **R1** S0/0/0 y la interface del **R2** S0/0/0(Figura 34).

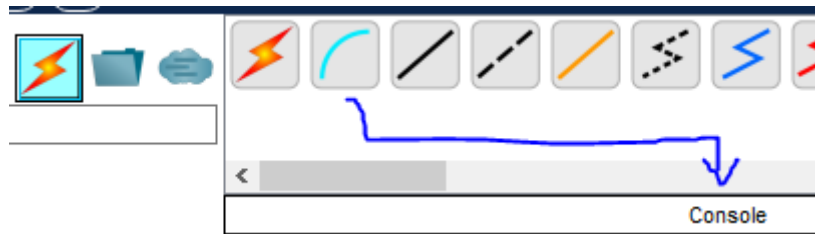


Figura 33 .Sección de cables PT

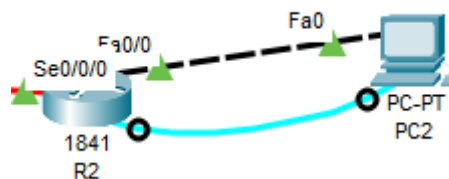


Figura 34 . Resultado en packet tracet conexión R1-R2.

Para poder configurar los parámetros de router es necesario conectarlo a un pc o laptop.
Una vez conectado con el cable de consola y asignado las direcciones IP MS y Gateway si hacemos click en el pc y nos dirigimos a la pestaña desktop .(Figura 35) aplicación terminal y pulsamos OK se nos abrirá una consola en la que si pulsamos enter tendremos acceso al router que conectamos anteriormente al pc por el cable de consola.

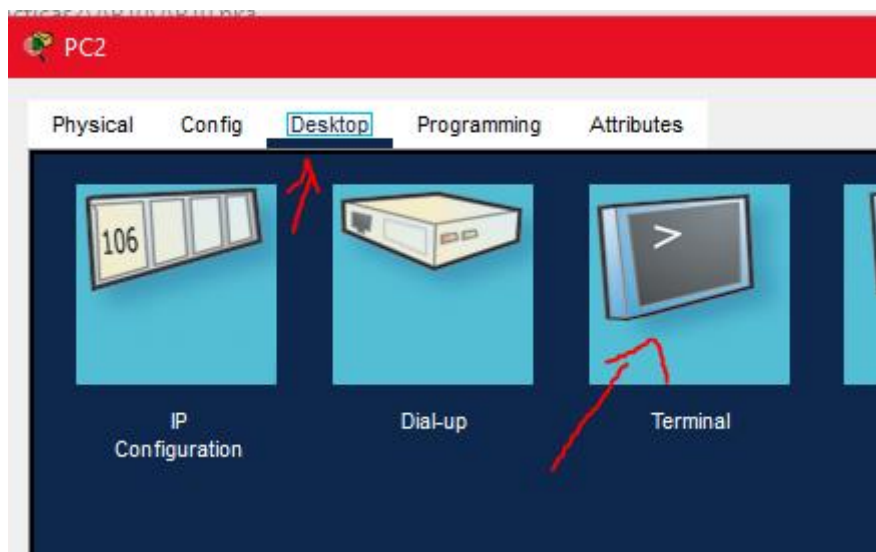


Figura 35 . Configurar router desde PC.

4. Conclusiones

Y las conclusiones para este contenido son:

1. Para que se puedan comunicar un PC con un Switch es necesario el cable (Copper Straight-Through) usando la interface de **switch** FastEthernet0/2 y la interface del **PC1** FastEthernet0.
2. Para que se puedan comunicar un Switch y un Router es necesario el cable (Copper Straight-Through) usando la interface de **switch** FastEthernet0/1 y la interface del **router1** FastEthernet0/0.
3. Para que se puedan comunicar dos Routers es necesario el cable (Serial DCE) usando la interface de **R1** S0/0/0 y la interface del **R2** S0/0/0.
4. Para poder configurar los parámetros de router es necesario conectarlo a un pc o laptop. Una vez conectado con el cable de consola y asignado las direcciones IP MS y Gateway si hacemos click en el pc y nos dirigimos a la pestaña desktop aplicación terminal y pulsamos OK se nos abrirá una consola en la que si pulsamos enter tendremos acceso al router que conectamos anteriormente al pc por el cable de consola.

CP12: Conocer la interfaz de comandos (CLI) de un router en Cisco Packet Tracer (PT)

1. Diagrama de topología (Figura 11)



Figura 11. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC con la herramienta de simulación de red Cisco Packet Tracer.

2.1 Desarrollo de la tarea

Utilizaremos el acceso a consola a través de un pc conectado a un router para así configurar el router sus interfaces y otros parámetros.

3. Desarrollo

3.1. Describir cómo cambiar entre los modos de la CLI de EXEC de usuario, EXEC privilegiado y configuración global de un router Cisco

Una vez en la consola (se accede como se explica en la CP12) le damos a enter hasta que aparezca **Router>**

1. Entra al modo de EXEC (ejecución) privilegiado:

Router>enable

Router# (representa que estamos en EXEC privilegiado)

2. Para volver a EXEC de usuario poner

Router#exit

3. Para acceder al modo de configuración global desde EXEC privilegiado

Router#configure terminal

Router(config)# (modo de configuración global)

3.2. Describir cómo lleva a cabo la configuración básica a través de la CLI de un router Cisco.

Seguiremos 5 pasos

- 1) Configura el **nombre del router** como **R1** desde el modo de configuración global:

```
Router(config)#hostname R1
```

- (2) Configura una **contraseña secreta del modo EXEC privilegiado** como **cisco**:

```
R1(config)#enable secret cisco
```

- (3) Desactiva la búsqueda de DNS:

```
R1(config)#no ip domain-lookup
```

- (4) Configura una **contraseña para el acceso por consola** como **cisco**:

```
R1(config)#line con 0
R1(config-line)#password cisco (modo de configuración de línea)
R1(config-line)#login
R1(config-line)#exit (volver a modo de configuración global)
R1(config)#
```

- (5) Configura una **contraseña para el acceso por terminal virtual (vty)** como **cisco**:

```
R1(config)#line vty 0 4
R1(config-line)#password cisco
R1(config-line)#login
R1(config-line)#exit (volver a modo de configuración global)
R1(config)#
```

3.3. Indicar qué tipo de información devuelve y para qué sirve el comando **show running-config** de la CLI un router Cisco.

El comando **show running-config** (Figura 36) sirve para revisar la configuración activa usada actualmente el router y poder así ver sus interfaces y que está cargada en la memoria RAM de este dispositivo, ejecuta el siguiente comando ejemplo:

```
R1# show running-config
<Resultado omitido>
Building configuration...
Current configuration : 1063 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname R1
enable secret 5 $1$i6w9$dvdpm6zV10E6tSyLdkR5/
no ip domain lookup
!
interface FastEthernet0/0
description LAN 192.168.1.0 default gateway
ip address 192.168.1.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
```

Figura 36 . Ejemplo (no significativo del ejercicio) del comando show running-config.

4. Conclusiones

Y las conclusiones para este contenido son:

1. Para poder comprobar las dirección ip y las interface configuradas, podemos usar show ip route o show ip interface.
2. Otra forma de comprobar las dirección ip y las interface configuradas es dejando el ratón encima del router en el diagrama de topología.
3. El # en Router# representa que estamos en EXEC privilegiado.
4. El (config)# en Router(config)# representa que estamos en modo de configuración global.

CP13: Configurar las interfaces de un router a través de la CLI en Cisco Packet Tracer (PT)

1. Diagrama de topología (Figura 12)



Figura 12. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC con la herramienta de simulación de red Cisco Packet Tracer.

2.1 Desarrollo de la tarea

Utilizaremos el acceso a consola a través de un pc conectado a un router para así configurar el router sus interfaces y otros parámetros..

3. Desarrollo

3.1. Describir cómo configurar una interfaz LAN tipo Ethernet de un router Cisco.

Para configurar las interface es necesario hacer todos pasos vistos en las dos actividades anteriores. Una vez realizado asegurarnos de que estamos en modo de configuracion global con (config)# y poner los siguientes comandos:

```
R1(config)#interface Fa0/0
R1(config-if)#description Para LAN1 (commando opcional)
R1(config-if)#ip address 192.168.1.62 255.255.255.224
R1(config-if)#no shutdown (commando para activar la interface)
R1(config-if)#exit
```

Dispositivo	ID Interfaz	Dirección IP	Máscara de
R1	Fa0/0	192.168.1.62	255.255.255.224
	S0/0/0 (DCE)	192.168.1.65	255.255.255.252

Figura 37 . Direcciones para poder configurar la interface de R1 desde el Pc.

3.2. Describir cómo configurar una interfaz WAN tipo Serial DCE de un router Cisco

Para configurar las interface es necesario hacer todos pasos vistos en las dos actividades anteriores. Una vez realizado asegurarnos de que estamos en modo de configuracion global con (config)# y poner los siguientes comandos:

Realizar la configuración de la interfaz WAN Serial 0/0/0 DCE
Se necesita un reloj para sincronizar la transmisión y recepción de datos entre los dispositivos conectados.

```
R1 (config) #interface S0/0/0
R1 (config-if) #description Para WAN
R1 (config-if) # ip address 192.168.1.65 255.255.255.252
R1 (config-if) #clock rate 2000000
R1 (config-if) #no shutdown
R1 (config-if) #end
R1#
```

Utilizamos las direcciones:

Dispositivo	ID Interfaz	Dirección IP	Máscara de	Puerta de enlace
R1	Fa0/0	192.168.1.62	255.255.255.224	No aplicable
	S0/0/0 (DCE)	192.168.1.65	255.255.255.252	No aplicable
R2	Fa0/0	192.168.1.30	255.255.255.224	No aplicable
	S0/0/0	192.168.1.66	255.255.255.252	No aplicable
PC1	NIC	192.168.1.33	255.255.255.224	192.168.1.62
PC2	NIC	192.168.1.1	255.255.255.224	192.168.1.30

Figura 38 . Direcciones para poder configurar la interface de R1 desde el Pc

Nota: el comando **clock rate** sirve para simular la capacidad en bits/s (velocidad) del enlace cuando no existe un ISP. Es OBLIGATORIO configurarlo en el extremo DCE del cable serial que conecta dos routers en una conexión WAN punto-a-punto.

3.3. Describir cómo configurar una interfaz WAN tipo Serial DTE de un router Cisco.

Para configurar las interface es necesario hacer todos pasos vistos en las dos actividades anteriores. Una vez realizado asegurarnos de que estamos en modo de configuracion global con (config)# y poner los siguientes comandos:

Realizar la configuración de la interfaz WAN Serial 0/0/0

```
R2 (config) #interface S0/0/0
R2 (config-if) #description Para WAN DTE
R2 (config-if) # ip address 192.168.1.66 255.255.255.252
R2 (config-if) #no shutdown
R2 (config-if) #end
R2#
```

Utilizamos las direcciones:

Dispositivo	ID Interfaz	Dirección IP	Máscara de	Puerta de enlace
R1	Fa0/0	192.168.1.62	255.255.255.224	No aplicable
	S0/0/0 (DCE)	192.168.1.65	255.255.255.252	No aplicable
R2	Fa0/0	192.168.1.30	255.255.255.224	No aplicable
	S0/0/0	192.168.1.66	255.255.255.252	No aplicable
PC1	NIC	192.168.1.33	255.255.255.224	192.168.1.62
PC2	NIC	192.168.1.1	255.255.255.224	192.168.1.30

Figura 39 . Direcciones para poder configurar la interface de R1 desde el Pc.

3.4. Indicar qué tipo de información devuelve y para qué sirve el comando **show ip interface brief** de la CLI un router Cisco.

Desde el modo EXEC privilegiado -> **show ip interface brief** (Figura 40) sirve para revisar la configuración de las interfaces.

R2#**show ip interface brief**

```
R2#show ip interface brief
Interface      IP-Address      OK? Method Status      Protocol
FastEthernet0/0 192.168.1.30    YES manual up          up
FastEthernet0/1 unassigned      YES unset  administratively down down
Serial0/0/0     192.168.1.66    YES manual up          up
Serial0/0/1     unassigned      YES unset  administratively down down
Vlan1           unassigned      YES unset  administratively down down
```

Figura 40 . Ejemplo comando R2#show ip interface brief.

4. Conclusiones

Y las conclusiones para este contenido son:

1. El comando `clock rate` sirve para simular la capacidad en bits/s (velocidad) del enlace cuando no existe un ISP. Es OBLIGATORIO configurarlo en el extremo DCE del cable serial que conecta dos routers en una conexión WAN punto-a-punto.
2. El comando `R2(config-if)#no shutdown` es necesario y sirve para activar una interface después de configurarla.
3. El comando `R2#show ip interface brief` sirve para revisar la configuración de las interfaces en este caso de R2.

CP14: Configurar una ruta estática por defecto y rutas estáticas de red específica en un router en Cisco Packet Tracer (PT)

1. Diagrama de topología (Figura 13)

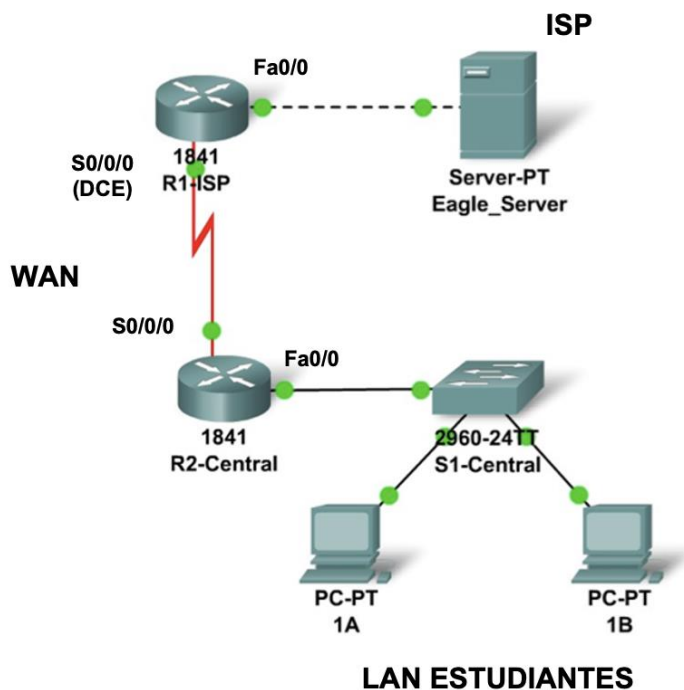


Figura 13. Diagrama de topología

2. Situación inicial

Este contenido debe desarrollarse desde un PC con la herramienta de simulación de red Cisco Packet Tracer.

2.1 Desarrollo de la tarea

Utilizaremos el acceso a consola a través de un pc conectado a un router para así configurar el router sus interfaces y otros parámetros.

3. Desarrollo

3.1. Describir cómo configurar una ruta estática por defecto en un router Cisco.

Existen dos opciones para configurar una ruta estática por defecto en IPv4(ip6 es distinto).

Opción 1 con dirección IP de siguiente salto.

R1(config)#ip route 0.0.0.0 0.0.0.0 **ip_siguiente_salto**

Opción 2 con identificador de la interfaz de salida.

R1(config)#ip route 0.0.0.0 0.0.0.0 **interface_salida** **ej:s0/0/0**

3.2. Describir cómo planificar y configurar una ruta estática de red específica en un router Cisco.

Configurar una ruta estática de red específica en IPv4 (ip6 es distinto).

Opción 1 con dirección IP de siguiente salto.

R1(config)#ip route **dir_red_destino** **mascara_subred** **ip_siguiente_salto**

Ej: R1(config)#ip route **192.168.111.0** **255.255.255.128** **192.168.111.137**

Opción 2 con identificador de la interfaz de salida.

R1(config)#ip route **dir_red_destino** **mascara_subred** **interface_salida**

Ej: R1(config)#ip route **192.168.111.0** **255.255.255.128** **s0/0/0**

Cabe destacar que según tengo entendido (no afirmo al 100%) el ultimo octeto de la **dir_red_destino** si es clase C dos últimos si es B y 3 ultimos si es A puede ser cualquier valor dentro del rango pero mejor poner 0.

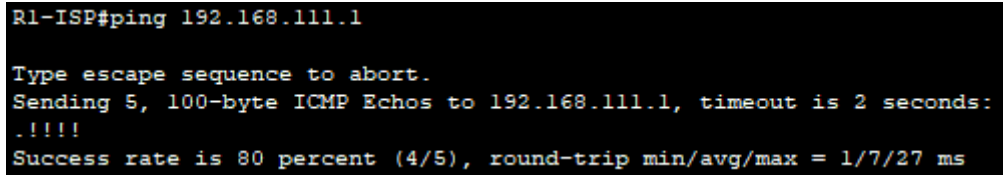
Dispositivo	ID Interfaz	Dirección IP	Máscara de subred	Puerta de enlace predeter.	Dirección servidor DNS
R1-ISP	Fa0/0	192.168.111.134	255.255.255.248	No aplicable	No aplicable
	S0/0/0 (DCE)	192.168.111.138	255.255.255.252	No aplicable	No aplicable
R2-Central	Fa0/0	192.168.111.126	255.255.255.128	No aplicable	No aplicable
	S0/0/0	192.168.111.137	255.255.255.252	No aplicable	No aplicable
PC 1A	NIC	192.168.111.1	255.255.255.128	192.168.111.126	192.168.111.133
PC 1B	NIC	192.168.111.2	255.255.255.128	192.168.111.126	192.168.111.133
Server-PT	NIC	192.168.111.133	255.255.255.248	192.168.111.134	No aplicable

Figura 41 . Direcciones para poder configurar una ruta estática de R1 a R2 .

3.3. Describir cómo realizar una prueba de conectividad para comprobar el funcionamiento de las rutas estáticas configuradas.

Desde R1 si hacemos un ping para acceder fuera a PC1 (Figura 42).

R1# ping 192.168.111.1



```
R1-ISP#ping 192.168.111.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.111.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/7/27 ms
```

Figura 42 . Comando ping desde R1 hacia PC1.

Veremos 2 opciones

-Éxito en las respuestas: Si recibes respuestas exitosas, significa que la ruta estática está funcionando correctamente y que el router puede alcanzar la red de destino a través de la ruta configurada.

-Falló la respuesta: Si no recibes respuestas o si obtienes mensajes de error, podría haber un problema en la configuración de la ruta estática, la configuración de las interfaces, la configuración de la máscara de subred, o puede que el dispositivo en la dirección de destino no esté respondiendo.

Para verificar que todo esta correcto puedes verificar la tabla de enrutamiento con el comando **show ip route** para asegurarte de que la ruta estática esté presente y activa.

3.4. Indicar qué tipo de información devuelve y para qué sirve el comando **show ip route** de la CLI un router Cisco.

Para verificar que todo esta correcto (conexiones, rutas estáticas o por defecto) puedes verificar la tabla de enrutamiento con el comando **show ip route** (Figura 43).

```
R1-ISP#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

192.168.111.0/24 is variably subnetted, 3 subnets, 3 masks
S    192.168.111.0/25 [1/0] via 192.168.111.137
C    192.168.111.128/29 is directly connected, FastEthernet0/0
C    192.168.111.136/30 is directly connected, Serial0/0/0
```

Figura 43 . Comando show ip route desde R1 .

Otra opción de cerciorarse de que las rutas y las interfaces estén activas y funcionales es poniendo el raton encima del router (en PT) tal y como se muestra en la siguiente imagen Figura 44.

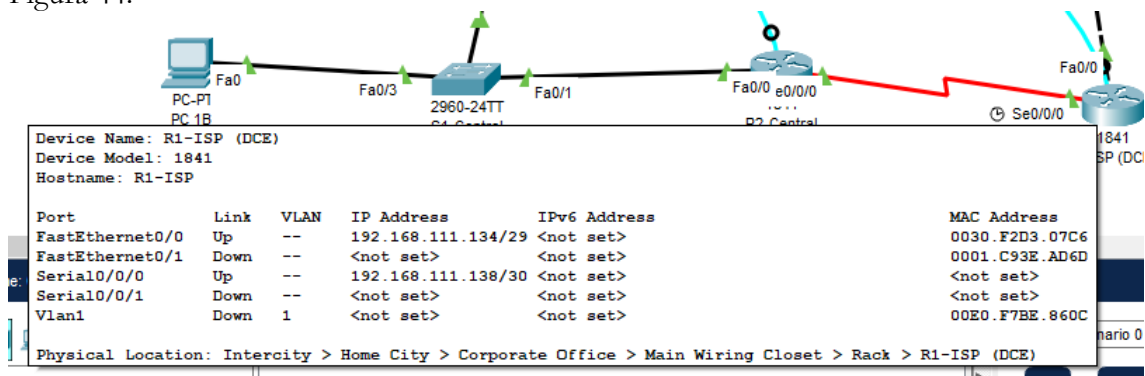


Figura 44 . Comando show ip route desde R1 .

4. Conclusiones

Y las conclusiones para este contenido son:

1. Cada router toma sus decisiones individualmente basándose en la información que posee en su propia tabla de enrutamiento.
2. El hecho de que un router posea determinada información en su tabla de enrutamiento no significa que otros routers posean la misma información.
3. La información de enrutamiento acerca de una ruta desde una red a otra no brinda información de enrutamiento acerca de la ruta inversa o de la ruta de regreso.