12. Ejercicio para entregar en grupo y defender en coloquio.

Resolver el direccionamiento IPv4 con el bloque IP asignado. Considerar los enlaces punto a punto salvo la red de n9, n10, n11, n13 y n14. Para la red de n9 considerar 40 hosts; para la red de n11 y n13, 328 hosts y para la red de n14, 500 hosts.

**Direccionamiento IP Asignado**

* **IPv4**: 140.222.16.0/21
* **IPv6**: 2001:fef8:df80::/48

Se utilizarán enlaces punto a punto en la mayoría de las redes, exceptuando la red compuesta por los nodos n9, n10, n11, n13 y n14.

1. Configurar la red detrás de n5 (n5,n13,n11) y la Red C con el bloque IPv6 asignado.

**Subredes IPv4 para Redes Específicas**

**Red de n14**

* **Requerimiento de hosts**: 500 hosts (9 bits para hosts, máscara /23).
* **Asignación**: 140.222.16.0/23
* **Direcciones IP**:
* Nodo n8: 140.222.16.1/23
* Nodo n14: 140.222.16.2/23

**Red de n11 y n13**

* **Requerimiento de hosts**: 328 hosts (9 bits para hosts, máscara /23).
* **Asignación**: 140.222.18.0/23
* **Direcciones IP**:
* Nodo n5: 140.222.18.1/23
* Nodo n11: 140.222.18.2/23
* Nodo n13: 140.222.18.3/23

**Red de n9**

* **Requerimiento de hosts**: 40 hosts (6 bits para hosts, máscara /26).
* **Asignación**: 140.222.20.0/26
* **Direcciones IP**:
* Nodo n8: 140.222.20.1/26
* Nodo n9: 140.222.20.2/26

**Redes Punto a Punto (Enlaces)**

Cada enlace requiere 2 bits para hosts, utilizando una máscara /30.

**Red n1 - n2**:

* Subred: 140.222.20.64/30
* IPs: n1: 140.222.20.65/30, n2: 140.222.20.66/30

**Red n2 - n3**:

* Subred: 140.222.20.68/30
* IPs: n2: 140.222.20.69/30, n3: 140.222.20.70/30

**Red n1 - n3**:

* Subred: 140.222.20.72/30
* IPs: n1: 140.222.20.73/30, n3: 140.222.20.74/30

**Red n3 - n5**:

* Subred: 140.222.20.76/30
* IPs: n3: 140.222.20.77/30, n5: 140.222.20.78/30

**Red n5 - n6**:

* Subred: 140.222.20.80/30
* IPs: n5: 140.222.20.81/30, n6: 140.222.20.82/30

**Red n3 - n6**:

* Subred: 140.222.20.84/30
* IPs: n3: 140.222.20.85/30, n6: 140.222.20.86/30

**Red n3 - n15**:

* Subred: 140.222.20.88/30
* IPs: n3: 140.222.20.89/30, n15: 140.222.20.90/30

**Red n4 - n15**:

* Subred: 140.222.20.92/30
* IPs: n4: 140.222.20.93/30, n15: 140.222.20.94/30

**Red n2 - n4**:

* Subred: 140.222.20.96/30
* IPs: n2: 140.222.20.97/30, n4: 140.222.20.98/30

**Red n2 - n7**:

* Subred: 140.222.20.100/30
* IPs: n2: 140.222.20.101/30, n7: 140.222.20.102/30

**Red n7 - n8**:

* Subred: 140.222.20.104/30
* IPs: n7: 140.222.20.105/30, n8: 140.222.20.106/30

**Red n1 - n10**:

* Subred: 192.168.1.0/24 (RFC-1918)
* IPs: n1: 192.168.1.1/24, n10: 192.168.1.2/24

**Subredes IPv6 para Redes Detrás de n5 y Red C**

a) la red detrás de n5 (n5, n13, n11)  y  la Red C con el bloque IPv6

**Red n5, n11 y n13**

* **Asignación IPv6**: 2001:fef8:df80:0::/64
* **Direcciones IP**:
* Nodo n5: 2001:fef8:df80:0::1/64
* Nodo n11: 2001:fef8:df80:0::2/64
* Nodo n13: 2001:fef8:df80:0::3/64

**Red C**

Esta red abarca los siguientes enlaces:

**Red n2 - n7**:

* Subred IPv6: 2001:fef8:df80:1::/64
* IPs: n2: 2001:fef8:df80:1::1/64, n7: 2001:fef8:df80:1::2/64

**Red n7 - n8**:

* Subred IPv6: 2001:fef8:df80:2::/64
* IPs: n7: 2001:fef8:df80:2::1/64, n8: 2001:fef8:df80:2::2/64

**Red n8 - n9**:

* Subred IPv6: 2001:fef8:df80:3::/64
* IPs: n8: 2001:fef8:df80:3::1/64, n9: 2001:fef8:df80:3::2/64

**Red n8 - n14**:

* Subred IPv6: 2001:fef8:df80:4::/64
* IPs: n8: 2001:fef8:df80:4::1/64, n14: 2001:fef8:df80:4::2/64

1. Resolver con ruteo estático la topología.

Se comienza definiendo la dirección de la red con mayor numero de hosts

1. Alternativo: Asignar a n10 una IP según RFC-1918 y configurar NAT en n1 para que pueda alcanzar al resto de los equipos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Realizar test con ping (ICMP) y traceroute para probar que funciona la topología.
2. Capturar puntualmente el tráfico de n13 hacia n7 y analizar: ARP e ICMP.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Realizar un traceroute entre los mismos equipos, capturar los mensajes.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Alternativo: Modificar los MTU para ver la fragmentación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Probar conectividad en las redes con IPv6 (por separado), capturar tráfico y analizar ICMPv6.

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

22. Ejercicios UDP a entregar. Correr sobre la topología de la figura 6:

a) Levantar un servicio UDP con la herramienta nc (netcat) en el host n9 y enviar información desde n13 usando el mismo comando. Ver el estado de los sockets en ambos extremos. ¿Qué significa el estado ESTABLISHED en UDP?

Interfaz de usuario gráfica

Descripción generada automáticamente

b) Levantar en el super daemon inetd o similar con el servicio UDP echo y probar con un cliente programado en el lenguaje de su elección mediante la API socket contra este servicio. Inspeccionar el estado. Ver de generar datos hacia el “servidor” desde más de un Nodo.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

c) Enviar información desde n13 a n9 a un port UDP donde no existe un proceso esperando por recibir datos. ¿Cómo notifica el stack TCP/IP de este hecho? Investigue la herramienta traceroute que ports utiliza y cómo usa estos mensajes (Ver ejercicio de IP con ruteo estático).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

d ) Para las pruebas anteriores capturar tráfico y ver el formato de los datagramas UDP y como se encapsulan en IP.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

23. Ejercicios TCP a entregar. Correr sobre la topología de la figura 6 (continuación de lo que venían realizando en la topología en la práctica anterior):

a) Programar en el lenguaje de su elección mediante la API socket un servidor TCP que escuche conexiones en el puerto 9000 y que los datos que reciba los descarte. Correr en el nodo n9 y enviar datos desde otro nodo usando la herramienta nc (netcat) o telnet.

Interfaz de usuario gráfica

Descripción generada automáticamente

b) En el nodo n9 levantar con el super daemon inetd algunos servicios extras, como tcp echo, discard y otros. Chequear los servicios TCP y UDP activos.

Interfaz de usuario gráfica

Descripción generada automáticamente

c) Desde el nodo n13 realizar una conexión TCP y enviar datos mediante un programa cliente de su elección al servicio discard, y al echo. Capturar el tráfico con la herramienta tcpdump o wireshark y analizar la cantidad de segmentos, los flags utilizados y las opciones extras que llevan los encabezados tcp.

Una captura de pantalla de una computadora

Descripción generada automáticamente

d) Sin cerrar las conexiones chequear los servicios activos y ver los Estados.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

e ) Generar nuevas conexiones hacia el nodo n9 e inspeccionar los estados. Por ejemplo realizar varias conexiones simultáneas al servicio tcp echo desde el mismo origen y desde otros nodos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

f) Intentar generar conexiones a un puerto donde no existe un proceso esperando por recibir datos. ¿Cómo notifica TCP de este hecho (ver flags)?

Una captura de pantalla de una computadora

Descripción generada automáticamente

g ) Cerrar las conexiones y ver el estado de los servicios en ambos lados. ¿En qué estado queda el que hace el cierre activo?

Interfaz de usuario gráfica

Descripción generada automáticamente

h) Observando la captura indicar la cantidad de segmentos y los flags utilizados. ¿Con cuántos segmentos se cerró la conexión? ¿Existen otras variantes de cierre?

i) Hacer un diagrama de los segmentos intercambiados con los números de secuencia absolutos para una de las sesiones TCP (Se puede usar la herramienta wireshark u otra).

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

j ) Alternativo: Realizar una conexión mediante nc indicando un puerto específico para el cliente. Luego cerrar la conexión desde el cliente e intentar abrirla nuevamente. ¿En qué estado está el socket? Investigar valor del 2MSL en la plataforma sobre la cual está haciendo los tests.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente