Clase 08

Diseño y Programación Web

Materia:

Sistemas Operativos

Docente contenidista: CARLASSARA, Fabrizio

Revisión: Coordinación

Contenido

Clase 8

¡Te damos la bienvenida a la materia **Sistemas Operativos**!

En esta clase vamos a ver los siguientes temas:

En esta clase vamos a comenzar a empezar a introducirnos en cómo funcionan las redes. Entre otras cosas veremos:

- Componentes de una red.
- Modelo OSI y TCP/IP.
- IPv4 e IPv6.
- Máscaras de subred.
- Configuración de red en Linux.
- Cliente DNS.

Introducción a redes

Las redes de computadoras son sistemas que permiten la comunicación y el intercambio de información entre diferentes dispositivos informáticos, como computadoras, servidores, impresoras, dispositivos móviles, entre otros. Estas redes pueden ser tan simples como una conexión entre dos computadoras en una misma habitación o tan complejas como una red global que conecta millones de dispositivos en todo el mundo, como Internet.

Entre algunos de los elementos que componen una red podemos encontrar:

- **Dispositivos de red:** Los dispositivos de red son equipos electrónicos que se utilizan para conectar dispositivos entre sí en una red. Algunos ejemplos comunes incluyen:
    - **Computadoras:** Equipos que pueden enviar, recibir y procesar datos en la red.
    - **Router:** Dispositivo que interconecta redes y dirige el tráfico de datos entre ellas.
    - **Switch:** Dispositivo que conecta múltiples dispositivos en una red local y facilita la comunicación entre ellos.
    - **Punto de acceso inalámbrico (Access Point):** Dispositivo que permite a los dispositivos inalámbricos conectarse a una red cableada.
    - **Firewall:** Dispositivo que controla el tráfico de red basado en reglas de seguridad.
- **Medios de transmisión:** Son los medios físicos a través de los cuales se transmiten los datos en la red. Algunos ejemplos incluyen:
    - **Cable de cobre:** Utilizado en redes cableadas Ethernet.
    - **Fibra óptica:** Ofrece velocidades de transmisión más altas y mayor seguridad que el cable de cobre.
    - **Señales inalámbricas:** Utilizadas en redes Wi-Fi y redes celulares.
- **Protocolos de red:** Son conjuntos de reglas y convenciones que rigen la comunicación entre dispositivos en una red. Algunos protocolos comunes incluyen:
    - **TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet):** Protocolo

fundamental de Internet que facilita la comunicación entre dispositivos en redes IP.
    - **HTTP (Protocolo de Transferencia de Hipertexto):** Protocolo utilizado para transferir datos en la World Wide Web.
    - **SMTP (Protocolo Simple de Transferencia de Correo):** Protocolo utilizado para enviar correo electrónico.
    - **FTP (Protocolo de Transferencia de Archivos):** Protocolo utilizado para transferir archivos entre sistemas en una red.

- **Topología de red:** Es la disposición física o lógica de los dispositivos en una red. Algunas topologías comunes incluyen:
    - **Estrella:** Todos los dispositivos están conectados a un nodo central, como un switch.
    - **Bus:** Todos los dispositivos están conectados a un solo cable principal.
    - **Anillo:** Los dispositivos están conectados en un bucle cerrado, donde cada dispositivo está conectado al siguiente y al anterior.
    - **Malla:** Cada dispositivo está conectado a múltiples otros dispositivos, creando múltiples rutas para la comunicación.

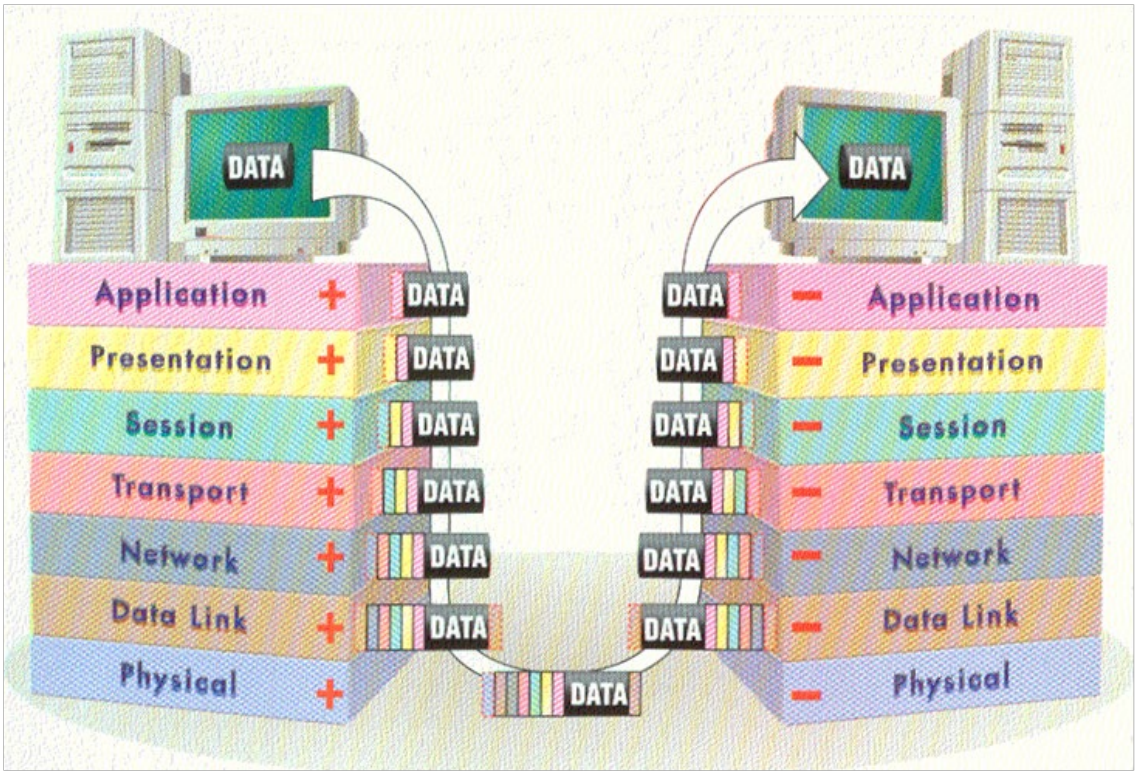
- **Modelos de red:** Son modelos teóricos que describen cómo las redes deberían funcionar y cómo los datos deberían fluir a través de ellas. Algunos modelos comunes incluyen:
    - **Modelo OSI (Open Systems Interconnection):** Modelo de referencia que divide las funciones de comunicación en siete capas, desde la física hasta la de aplicación.
    - **Modelo TCP/IP:** Modelo más utilizado en la práctica, que describe las funciones de comunicación en cuatro capas: red, transporte, Internet y aplicación.

Modelo OSI

El modelo OSI (Open Systems Interconnection) es un marco conceptual que define y estandariza las funciones de comunicación de un sistema de red en capas. Fue desarrollado por la Organización Internacional de Normalización (ISO) en la década de 1980 para

facilitar la interoperabilidad entre diferentes sistemas de red. El modelo OSI se compone de siete capas, cada una con funciones específicas que contribuyen a la comunicación entre dispositivos en una red.

Datos enviados desde un dispositivo a otro viajan por cada una de estas capas, donde cada una le agrega información adicional al paquete para que los dispositivos de la red sepan a donde enviarlo. Una vez en el receptor, este paquete de datos pasa por todas las capas en sentido contrario para que pueda desarmarse hasta obtener la información original.



Este modelo está compuesto por las siguientes capas:

1. **Capa física (Physical Layer):**
    a. **Función:** Esta capa se encarga de transmitir bits sin procesar a través de un medio de transmisión físico, como cables de cobre, fibra óptica o señales inalámbricas.
    b. **Responsabilidades:** Define las características eléctricas, mecánicas y funcionales de los dispositivos de red. Establece la conexión física entre los dispositivos y gestiona la transmisión de datos binarios a través del medio físico.
2. **Capa de enlace de datos (Data Link Layer):**

a. **Función:** Proporciona una comunicación fiable y sin errores entre nodos adyacentes a través de un medio de transmisión físico.
b. **Responsabilidades:** Dividida en dos subcapas:
    i. **Control de acceso al medio (MAC - Media Access Control):** Gestiona el acceso al medio compartido y controla la transmisión de datos.
    ii. **Control de errores (LLC - Logical Link Control):** Proporciona detección y corrección de errores, control de flujo y control de enlace lógico.

3. **Capa de red (Network Layer):**

    a. **Función:** Se encarga del enrutamiento de datos a través de una red interconectada de dispositivos.
    b. **Responsabilidades:** Determina la mejor ruta para la transmisión de datos desde el origen hasta el destino, controla el flujo de datos y realiza funciones de direccionamiento y enrutamiento.

4. **Capa de transporte (Transport Layer):**

    a. **Función:** Proporciona comunicación de extremo a extremo entre aplicaciones en dispositivos finales.
    b. **Responsabilidades:** Divide los datos en segmentos más pequeños para su transmisión, garantiza la entrega fiable de datos, controla el flujo de datos y realiza la segmentación y reensamblaje de datos.

5. **Capa de sesión (Session Layer):**

    a. **Función:** Establece, administra y finaliza las sesiones de comunicación entre dispositivos.
    b. **Responsabilidades:** Coordina la comunicación entre aplicaciones, maneja el inicio y cierre de sesiones y controla la sincronización y el diálogo entre los dispositivos.

6. **Capa de presentación (Presentation Layer):**

    a. **Función:** Se encarga de la representación y conversión de datos entre formatos de aplicación y formatos de red.
    b. **Responsabilidades:** Realiza la traducción, compresión y cifrado de datos, garantizando la compatibilidad entre diferentes sistemas y aplicaciones.

7. **Capa de aplicación (Application Layer):**

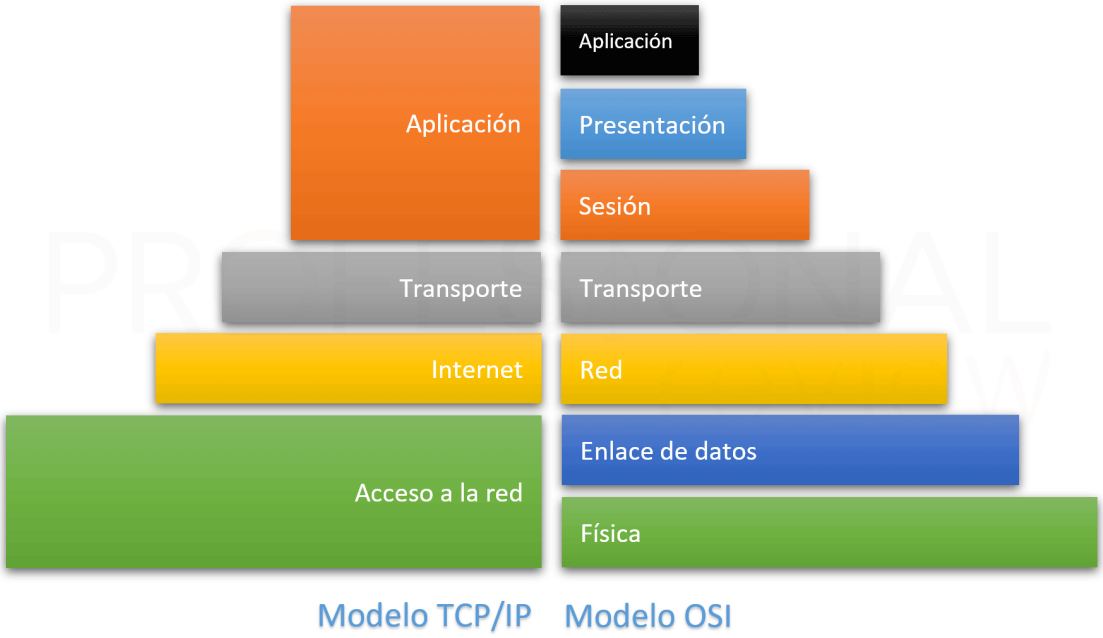
a. **Función:** Proporciona servicios de red directamente a las aplicaciones del usuario final.
b. **Responsabilidades:** Ofrece interfaces para acceder a servicios de red, como correo electrónico, transferencia de archivos, acceso web, entre otros.

Modelo TCP/IP

El modelo TCP/IP (Transmission Control Protocol/Internet Protocol) es un conjunto de protocolos de red que proporciona la base para la comunicación en Internet y en muchas otras redes. A diferencia del modelo OSI, el modelo TCP/IP no tiene una estructura de capas tan claramente definida, pero puede conceptualizarse en cuatro capas principales:

1. **Capa de Acceso a la Red (Network Access Layer):**
    a. Esta capa es equivalente a las capas físicas y de enlace de datos del modelo OSI.
    b. Se encarga de la transmisión de datos entre dispositivos en la misma red física.
    c. Incluye protocolos como Ethernet, Wi-Fi, PPP (Point-to-Point Protocol), y otros que se ocupan de la transmisión física y el acceso al medio.
2. **Capa de Internet (Internet Layer):**
    a. Esta capa es similar a la capa de red del modelo OSI.
    b. Su función principal es el enrutamiento de datos a través de redes interconectadas.
    c. El protocolo principal en esta capa es el Protocolo de Internet (IP), que se encarga de direccionar los paquetes de datos y determinar la mejor ruta para su entrega.
3. **Capa de Transporte (Transport Layer):**
    a. Esta capa es análoga a la capa de transporte del modelo OSI.
    b. Se encarga de la entrega de datos de extremo a extremo y de garantizar la fiabilidad y el control de flujo.
    c. Los protocolos principales en esta capa son el Protocolo de Control de Transmisión (TCP) y el Protocolo de Datagrama de Usuario (UDP).
4. **Capa de Aplicación (Application Layer):**

a. Esta capa es similar a la capa de aplicación del modelo OSI.
b. Proporciona servicios de red a las aplicaciones y usuarios finales.
c. Incluye una amplia gama de protocolos para servicios específicos, como HTTP para la World Wide Web, FTP para la transferencia de archivos, SMTP para el correo electrónico, y muchos otros.



Internet Protocol (IP)

Antes de seguir, tenemos que detenernos un momento para definir el Internet Protocol (IP). Este es un protocolo de comunicación de la capa de red del modelo TCP/IP que se utiliza para enrutar los datos a través de redes de computadoras. Proporciona la funcionalidad básica de direccionamiento y enrutamiento de paquetes de datos en Internet y en otras redes basadas en TCP/IP.

Características del IP

Entre algunas de las características de este protocolo tenemos:

- **Direccionamiento IP:** El IP asigna direcciones únicas a cada dispositivo en una red para identificarlos y permitir que se comuniquen entre sí. Estas direcciones están formadas por una serie de números binarios, que se suelen representar en formato decimal separado por puntos (IPv4) o en formato hexadecimal (IPv6).
- **Enrutamiento de Paquetes:** El IP determina cómo se enrutan los paquetes de datos a través de una red, utilizando información de las direcciones de origen y destino de los paquetes.
- **Fragmentación y Reensamblaje:** IP permite la fragmentación de paquetes de datos más grandes en fragmentos más pequeños para su transmisión a través de redes con diferentes límites de tamaño de paquete. Además, IP también es responsable del reensamblaje de estos fragmentos en el destino final.
- **Servicios sin Conexión:** El IP es un protocolo sin conexión, lo que significa que no se establece una conexión explícita entre los dispositivos antes de enviar datos. Cada paquete se envía de forma independiente y puede seguir rutas diferentes a través de la red.

IPv4

IPv4, o Internet Protocol versión 4, es la cuarta revisión del protocolo de comunicación de la capa de red del conjunto de protocolos TCP/IP, que es la base de Internet y muchas redes locales. IPv4 es el protocolo de red más utilizado en la actualidad y proporciona las direcciones IP que identifican de manera única a cada dispositivo en una red IP.

Las direcciones IPv4 se componen de 32 bits, divididos en cuatro octetos de 8 bits cada uno. Cada octeto se representa en formato decimal y está separado por puntos. Por ejemplo, una dirección IPv4 típica tiene el formato XXX.XXX.XXX.XXX, donde cada XXX representa un número decimal de 0 a 255.

En este protocolo existen las máscaras de subred. Éstas se utilizan para dividir la dirección IPv4 en dos partes: la parte de red y la parte de host. Se compone de una serie de unos seguidos de una serie de ceros. Por ejemplo, una máscara de subred típica para una red de clase C es 255.255.255.0. Esto es el equivalente a escribir que usaremos los primeros tres octetos o los primeros 24 bits de la dirección como máscara de subred, por lo que es común ver que luego de la dirección de IP se escriba /24 para definir la máscara.

La parte de red de una dirección IPv4 es la porción de la dirección que identifica la red a la que pertenece el dispositivo. Está determinada por los bits de la dirección IPv4 que están cubiertos por unos en la máscara de subred. Todos los dispositivos en la misma red deben tener la misma parte de red en sus direcciones IPv4. El router utiliza la parte de red de la dirección IPv4 para determinar a qué red se debe enviar un paquete de datos.

La parte de host de una dirección IPv4 es la porción de la dirección que identifica de manera única a un dispositivo dentro de una red. Está determinada por los bits de la dirección IPv4 que están cubiertos por ceros en la máscara de subred. Cada dispositivo en una red debe tener una parte de host única en su dirección IPv4. La parte de host se utiliza para identificar un dispositivo específico dentro de una red y dirigir los paquetes de datos al dispositivo correcto.

Por ejemplo, consideren una dirección IPv4 de 192.168.1.100 con una máscara de subred de 255.255.255.0. Esta máscara de subred en binario se escribe como 11111111.11111111.11111111.00000000.

- Parte de Red: Los primeros tres octetos de la dirección IPv4 (192.168.1) están cubiertos por unos en la máscara de subred, por lo que forman la parte de red.
- Parte de Host: El último octeto de la dirección IPv4 (100) está cubierto por ceros en la máscara de subred, por lo que forma la parte de host.

En este ejemplo, 192.168.1 identifica la red a la que pertenece el dispositivo, mientras que 100 identifica de manera única al dispositivo dentro de esa red.

Clases de redes

En IPv4, las direcciones IP se clasifican tradicionalmente en tres clases principales: A, B y C. Cada clase de dirección IP tiene un rango específico de direcciones disponibles y asigna diferentes porciones de la dirección para identificar la red y el host. Sin embargo, con la introducción de las máscaras de subred, esta clasificación ya no es tan relevante como lo fue en el pasado.

- **Clase A:**
  - Rango de Direcciones: 0.0.0.0 a 127.255.255.255.
  - Prefijo de Máscara de Subred: /8.
  - Identificación de Red: Los primeros 8 bits (primer octeto) de la dirección IP identifican la red y los 24 bits restantes identifican el host.
  - Cantidad de Direcciones por Red: Cerca de 16 millones de direcciones por red.
  - Ejemplo: 10.0.0.0/8.
- **Clase B:**
  - Rango de Direcciones: 128.0.0.0 a 191.255.255.255.
  - Prefijo de Máscara de Subred: /16.
  - Identificación de Red: Los primeros 16 bits (primeros dos octetos) de la dirección IP identifican la red y los 16 bits restantes identifican el host.
  - Cantidad de Direcciones por Red: Cerca de 65,000 direcciones por red.
  - Ejemplo: 172.16.0.0/16.
- **Clase C:**
  - Rango de Direcciones: 192.0.0.0 a 223.255.255.255.
  - Prefijo de Máscara de Subred: /24.
  - Identificación de Red: Los primeros 24 bits (primeros tres octetos) de la dirección IP identifican la red, y los 8 bits restantes identifican el host.
  - Cantidad de Direcciones por Red: Cerca de 254 direcciones por red.
  - Ejemplo: 192.168.1.0/24.

IPv6

IPv6, o Internet Protocol versión 6, es la versión más reciente del Protocolo de Internet y está diseñada para abordar las limitaciones y

desafíos del protocolo IPv4. A diferencia de IPv4, que utiliza direcciones IP de 32 bits, IPv6 utiliza direcciones IP de 128 bits, lo que permite un espacio de direcciones mucho más grande y resuelve el problema de agotamiento de direcciones que IPv4 enfrenta.

IPv6, o Internet Protocol versión 6, representa una evolución significativa con respecto a su predecesor, IPv4. Mientras que IPv4 utiliza direcciones IP de 32 bits, IPv6 emplea direcciones de 128 bits, proporcionando así un espacio de direcciones prácticamente ilimitado. Esto resuelve uno de los problemas más críticos de IPv4: el agotamiento de direcciones.

Con la abundancia de direcciones que ofrece IPv6, cada dispositivo y servicio en Internet puede tener su propia dirección única, facilitando enormemente la expansión y la conectividad de dispositivos en red. Además, IPv6 presenta mejoras en la eficiencia de enrutamiento y fragmentación de paquetes, simplificando el manejo de datos en la red. También incorpora seguridad integrada mediante soporte nativo para IPsec, lo que fortalece la protección de las comunicaciones en línea.

Otra característica destacada de IPv6 es su capacidad para la autoconfiguración de direcciones, lo que simplifica enormemente la configuración de red para los dispositivos y servicios. Aunque IPv4 sigue siendo ampliamente utilizado, IPv6 está ganando terreno como la nueva norma para las comunicaciones en Internet, gracias a su mayor espacio de direcciones, mejoras en el enrutamiento y la seguridad, y su capacidad para adaptarse a las demandas del crecimiento continuo de Internet.

Configuración de red

Vamos ahora a ver cómo hacer para que nuestro sistema operativo pueda tener acceso a la red y, a través de ella, a internet. En primer lugar, tenemos que configurar el adaptador de red de nuestro software de virtualización. Una vez que eso esté en orden, vamos a poder configurar la red de nuestro sistema operativo.

Adaptador de red

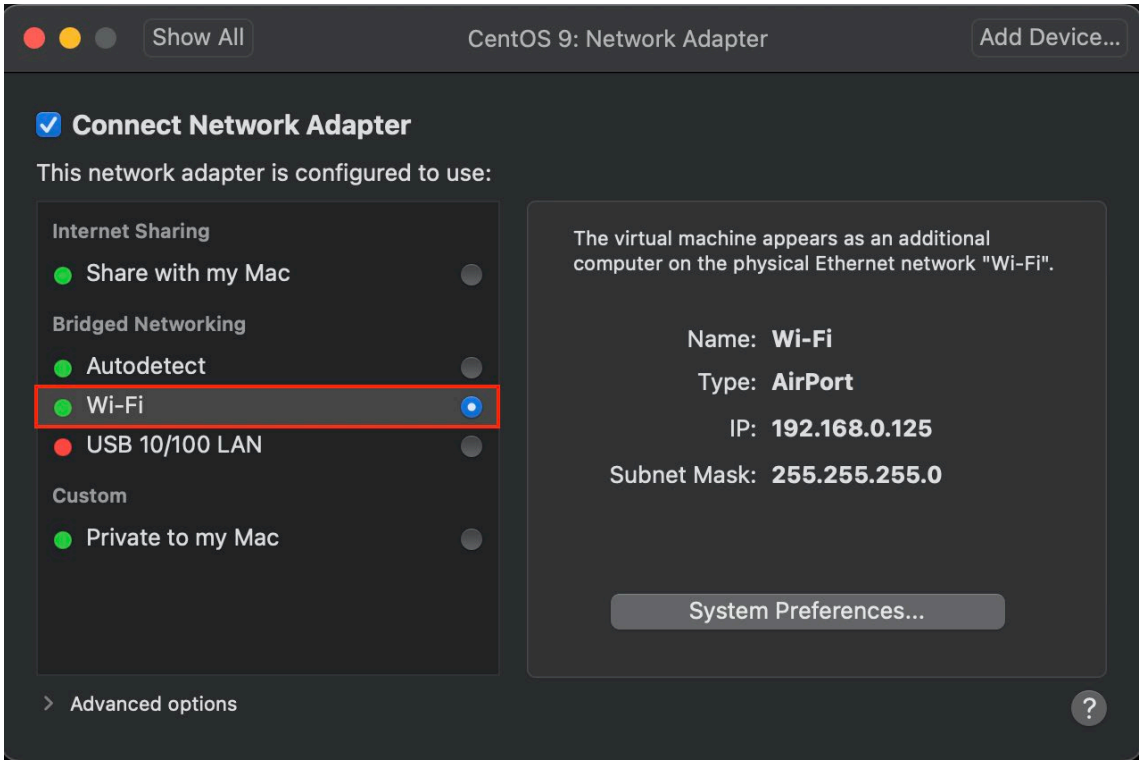
En los adaptadores de red de los software de virtualización, tendremos que elegir mayormente entre dos opciones: NAT y bridged. Algunas características de estas son:

- **NAT (Network Address Translation):**
    - NAT es una técnica que permite que múltiples dispositivos en una red privada compartan una única dirección IP pública para comunicarse con redes externas, como Internet.
    - En una configuración de NAT para una máquina virtual, el software de virtualización asigna una dirección IP privada a la máquina virtual y utiliza su propia dirección IP pública para todas las comunicaciones salientes.
    - El software de virtualización traduce automáticamente las direcciones y puertos de las comunicaciones salientes para que parezca que provienen de la dirección IP pública del host físico.
    - NAT proporciona una capa adicional de seguridad al ocultar las direcciones IP internas de la red virtual detrás de una dirección IP pública única.
- **Bridged (Puente):**
    - En una configuración de bridged para una máquina virtual, el adaptador de red virtual de la máquina virtual se conecta directamente a la red física del host.
    - Esto permite que la máquina virtual obtenga una dirección IP de la misma red que el host físico y que se comunique directamente con otros dispositivos en la red física, como si fuera una máquina física independiente.
    - El tráfico de red de la máquina virtual pasa a través del hardware de red del host físico, sin realizar traducción de direcciones IP ni otros tipos de manipulación.

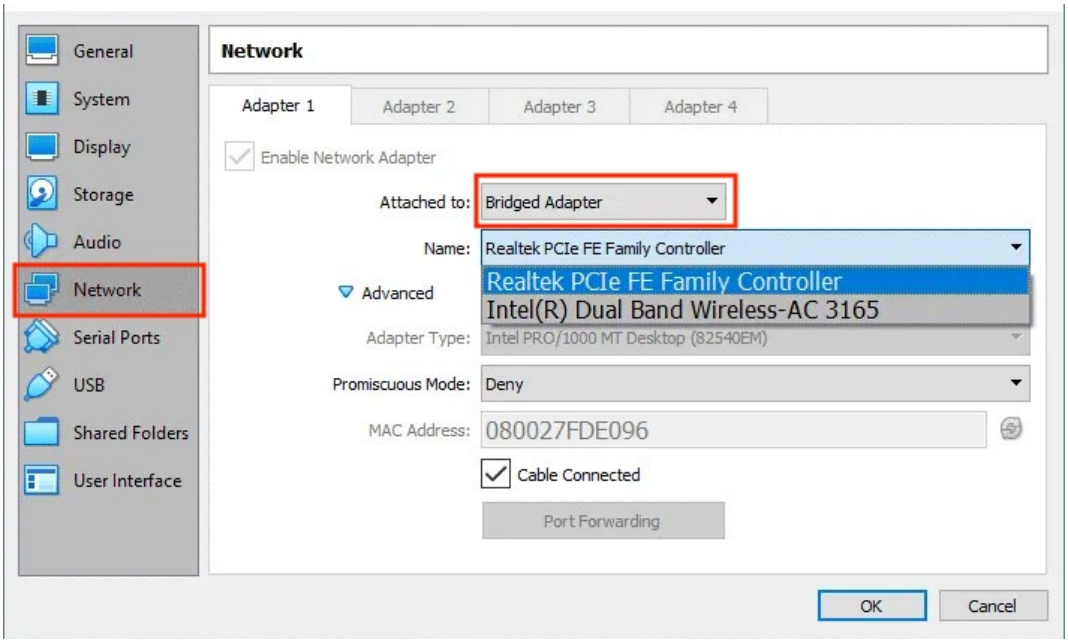
- Bridged proporciona una conectividad más transparente y directa a la red física, lo que puede ser útil en escenarios donde es necesario acceder a recursos de red externos o cuando se requiere una mayor flexibilidad en la configuración de la red de la máquina virtual.

Para nuestro caso, vamos a asegurarnos que nuestro adaptador de red funcione en modo bridged para exponer la máquina virtual a nuestra red y trabajarla como si fuera una computadora independiente.

En VMware vamos a buscar las opciones de configuración de nuestra máquina virtual y vamos al adaptador de red y buscamos la opción de Wi-Fi.



En VirtualBox vamos a buscar nuestra máquina virtual y buscar la configuración de red. Luego ahí, vamos a asegurarnos que esté en adaptador puente.



Configuración de red

Una vez que nuestro adaptador de red esté configurado, vamos a configurar la red para nuestro CentOS. En primer lugar, tenemos que averiguar cuál es la dirección de IP de nuestra red y el gateway o puerta de enlace. Podemos encontrar estos datos en las propiedades de Wi-Fi de nuestra computadora.

Habitualmente estas direcciones de IP suelen comenzar en *192.168.x.x* y el gateway suele ser *192.168.0.1*.

Una vez que tengamos este dato, vamos a hacer uso del Network Manager a través del comando **nmcli** para configurar la red.

En primer lugar, usando **nmcli connection show** podemos ver que interfaces de red tenemos disponibles para configurar. Podemos entender a una interfaz de red como un punto de conexión que permite que el sistema operativo se comunique con una red física o virtual. Cada interfaz de red está asociada con un dispositivo de red específico, como una placa de red Ethernet, una interfaz inalámbrica Wi-Fi, o una interfaz de red virtual.

Al correr el comando de arriba tendremos un resultado como este:

Vamos a ver por lo menos la interfaz *lo* que representa al *loopback*, esta es una interfaz de red especial que permite que el sistema se comunique consigo mismo.

Aparte de esta interfaz especial, encontraremos varias más de acuerdo con la cantidad de dispositivos de red que tengamos, los nombres variarán de acuerdo al tipo, pero en nuestro caso, encontraremos una con el nombre *ens160* que es la que representa una conexión de Ethernet de nuestra máquina virtual.

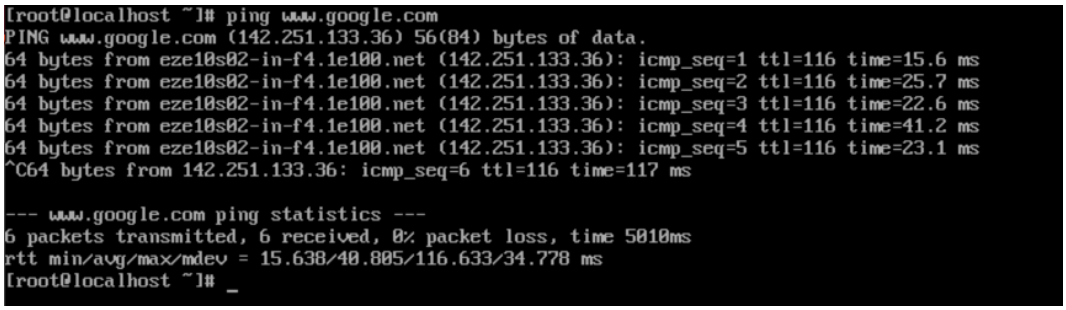
Una vez identificada nuestra interfaz de red, podemos configurar distintas propiedades de la interfaz de la siguiente forma:

- **nmcli connection modify [INTERFACE] ipv4.addresses [IPV4]/[MASK]** con este comando podemos elegir una dirección de IPv4 para nuestra interfaz, siempre y cuando esté disponible. Por ejemplo, *nmcli connection modify ens160 ipv4.addresses 192.168.0.127/24* asigna la dirección 192.168.0.127 con una máscara de 24 bits a la interfaz ens160.
- **nmcli connection modify [INTERFACE] ipv4.gateway [IPV4]** le asigna a la interfaz elegida la ruta hacia la puerta de enlace para poder dirigir el tráfico de red hacia otros dispositivos. Por ejemplo, *nmcli connection modify ens160 ipv5.gateway 192.168.0.1* asigna a la interfaz ens160 el gateway en 192.168.0.1. Este gateway tiene que estar en la misma red que la asignada con el comando anterior.
- **nmcli connection modify [INTERFACE] ipv4.dns [IPV4]** le asigna uno o varios servidores DNS a nuestro dispositivo para que podamos hacer uso de URLs en vez de direcciones de IP en casos conocidos. Por ejemplo, *nmcli connection modify ens160 ipv4.dns 8.8.8.8* asigna a la interfaz ens160 el servidor DNS en 8.8.8.8.

Algunos comandos útiles de nmcli también son:

- **nmcli connection down [INTERFACE]** da de baja una interfaz de red para que no pueda usarse.
- **nmcli connection up [INTERFACE]** da de alta una interfaz de red para que podamos empezar a usarla.
- **nmcli connection show [INTERFACE]** muestra una lista de datos detallada sobre la interfaz de red elegida.

Una vez que hayamos configurado nuestra red correctamente, podemos probar si tenemos acceso a internet mandando un ping a Google para ver si nos responde usando **ping www.google.com**. Deberíamos ver mensajes como estos:



Esto es un buen indicio, ya que los paquetes de prueba están llegando al servidor de Google y vuelven. Si vemos mensajes como "Red inaccesible" o "Nombre o servicio desconocido" es una indicación de que nuestra red está mal configurada.

Cliente DNS

El cliente DNS en Linux es una parte fundamental del sistema que se encarga de resolver los nombres de dominio a direcciones IP. Cuando un usuario intenta acceder a un sitio web o servicio utilizando su nombre de dominio (por ejemplo, "www.google.com"), el cliente DNS es responsable de traducir este nombre de dominio a una dirección IP que pueda ser entendida por la red y los servidores de destino.

Para que esto funcione correctamente, tenemos que registrar un servidor DNS donde nuestro cliente va a solicitar la IP de algún dominio. Existen varios servidores muy populares, uno de ellos es el 8.8.8.8 que registramos en la sección anterior con el comando nmcli connection modify ens160 ipv4.dns.

El funcionamiento del cliente DNS en Linux implica varios pasos:

1. **Consulta a los Servidores DNS:** Cuando un programa o aplicación en Linux necesita resolver un nombre de dominio, envía una consulta al cliente DNS. Esta consulta puede ser realizada por el comando nslookup, dig, o por librerías de resolución de nombres de dominio como gethostbyname() o getaddrinfo().
2. **Búsqueda en la caché local:** El cliente DNS primero busca en su caché local para ver si ya ha resuelto previamente ese nombre de dominio. Si la respuesta está en la caché y aún es válida (no ha caducado), entonces se devuelve la dirección IP

almacenada en la caché sin necesidad de realizar una consulta externa.

3. **Consulta a los servidores DNS configurados:** Si el nombre de dominio no está en la caché local o si la entrada ha caducado, el cliente DNS envía consultas a los servidores DNS configurados en el archivo /etc/resolv.conf en busca de la dirección IP correspondiente al nombre de dominio.
4. **Recibir respuesta y almacenar en caché:** Una vez que el cliente DNS recibe una respuesta del servidor DNS, la dirección IP correspondiente al nombre de dominio se almacena en la caché local para futuras consultas. Esto ayuda a mejorar el rendimiento, ya que evita tener que realizar la misma consulta DNS repetidamente.
5. **Actualización y expiración de la caché:** Las entradas en la caché local del cliente DNS tienen un tiempo de vida (TTL) asociado, que determina cuánto tiempo se mantendrá en caché la respuesta. Después de que el TTL expire, la entrada se elimina de la caché y se requerirá una nueva consulta DNS para resolver el nombre de dominio nuevamente.

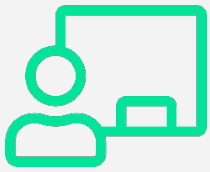
En resumen, el cliente DNS en Linux es responsable de traducir los nombres de dominio a direcciones IP mediante consultas a servidores DNS configurados. Almacenando las respuestas en una caché local, se mejora el rendimiento y se reduce la necesidad de realizar consultas DNS repetitivas para los mismos nombres de dominio.

Conclusiones

Hemos llegado al final de esta clase en donde hemos desarrollado temas como:

- Componentes de una red.
- Modelo OSI y TCP/IP.
- IPv4 e IPv6.
- Máscaras de subred.
- Configuración de red en Linux.
- Cliente DNS.

El entender cómo funcionan las redes va a ayudarnos a poder entender cómo solucionar posibles problemas de configuración que tengan que ver con el tráfico de datos entre distintos dispositivos.



Te recomendamos que puedas realizar el desafío semanal para que puedas verificar lo que aprendiste.

Estaremos en contacto ante cualquier consulta que tengas.

¡Hasta la próxima clase!

Hemos llegado así al final de esta clase en la que vimos:

- Componentes de una red.
- Modelo OSI y TCP/IP.
- IPv4 e IPv6.
- Máscaras de subred.
- Configuración de red en Linux.
- Cliente DNS.

El entender cómo funcionan las redes va a ayudarnos a poder entender cómo solucionar posibles problemas de configuración que tengan que ver con el tráfico de datos entre distintos dispositivos.

Te esperamos en la **clase en vivo** de esta semana.
No olvides realizar el **desafío semanal**.

¡Hasta la próxima clase!

Bibliografía

Eckert, J. W. (2020). Linux+ and LPIC-1: Guide to Linux Certification. Cengage.

Para ampliar la información

Cloudflare. (s. f.). ¿Qué es el modelo OSI? En Cloudflare. Recuperado de https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/

Fortinet. (s. f.). ¿Qué es un modelo TCP/IP? En Fortinet. Recuperado de https://www.fortinet.com/lat/resources/cyberglossary/tcp-ip#:~:text=El%20modelo%20TCP%2FIP%20define,los%20datos%20en%20las%20redes.

Red Hat. (s. f.). Comando nmcli. En Red Hat Enterprise Linux 8 System Design Guide. Recuperado de https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/system_design_guide/ref-frequent-nmcli-commands_getting-started-with-nmcli

International IT. (s. f.). Topología de red. En International IT. Recuperado de https://www.internationalit.com/post/topologia-de-red-conozca-los-principales-tipos?lang=es

freeCodeCamp. (s. f.). Hoja de trucos de subred - Mascara de subred /24 /30 /26 /27 /29 y otras referencias de red CIDR de dirección IP. En freeCodeCamp. Recuperado de https://www.freecodecamp.org/espanol/news/hoja-de-trucos-de-subred-mascara-de-subred-24-30-26-27-29-y-otras-referencias-de-red-cidr-de-direccion-ip/