📁 LPIC-2 / 🌐 Examen 205 - Configuración de Red - Ejercicios

Nota: Estos ejercicios implican usar herramientas de diagnóstico de red y, opcionalmente, configurar interfaces avanzadas. Las configuraciones avanzadas (bonding, bridging, VLANs) pueden romper la conectividad si no se hacen correctamente. ***Realiza las configuraciones avanzadas SIEMPRE en un entorno de prueba (VM) con interfaces virtuales adicionales dedicadas y un plan de recuperación.*** *Necesitarás privilegios de superusuario (`sudo`) para muchas herramientas y configuraciones.*

Ejercicio 5.2.1: Viendo el Estado de Interfaces y ARP Cache

- **Objetivo:** Usar comandos de línea de comandos para verificar el estado del enlace y la caché ARP.
- **Requisitos:** Acceso a la línea de comandos.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Verifica el estado del enlace (UP/DOWN) de tus interfaces:** Ejecuta `ip link show`. Busca la línea de estado después del nombre de la interfaz (ej: `<BROADCAST,MULTICAST,UP,LOWER_UP>`). `UP,LOWER_UP` indica que la interfaz está activa y el cable (o enlace virtual) está conectado.
    3. **Ver la caché ARP (moderno):** Ejecuta `ip neigh show`. Muestra las direcciones IP y MAC de los hosts que tu sistema ha aprendido en la red local, y su estado (`REACHABLE`, `STALE`, `FAILED`).
    4. **Ver la caché ARP (antiguo):** Ejecuta `arp -a`. Similar a `ip neigh show`.

Ejercicio 5.2.2: Capturando Tráfico de Red con `tcpdump`

- **Objetivo:** Usar `tcpdump` para observar paquetes de red.
- **Requisitos:** Acceso a la línea de comandos. El paquete `tcpdump` instalado (`sudo apt install tcpdump` o `sudo dnf install tcpdump`). Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Identifica tu interfaz de red principal:** Ejecuta `ip addr show`. Anota el nombre (ej: `enp3s0`).
    3. **Captura tráfico ICMP (ping) en la interfaz (en otra terminal):** En una *segunda* terminal, haz ping a un host (ej: `ping google.com`).
    4. **Captura tráfico en la interfaz principal (en la primera terminal):** Ejecuta `sudo tcpdump -i <tu_interfaz> -nnvvS`. Esto capturará paquetes, mostrará IPs y puertos numéricos, y será detallado. Mientras se ejecuta, haz ping desde la segunda terminal. Deberías ver los paquetes ICMP (echo request y echo reply) listados en la salida de `tcpdump`.

5. **Filtra tráfico por protocolo (ej: solo SSH):** Ejecuta `sudo tcpdump -i <tu_interfaz> -nnvvS tcp port 22`. Si accedes por SSH desde otra máquina, verás el tráfico.
6. **Filtra tráfico por host:** Ejecuta `sudo tcpdump -i <tu_interfaz> -nnvvS host <ip_de_un_host>`.
7. **Detén `tcpdump`:** Presiona `Ctrl+C`.
8. **(Concepto):** Para análisis más profundos, puedes usar `sudo tcpdump -i <interfaz> -w captura.pcap` para guardar el tráfico en un archivo y luego abrirlo con Wireshark en tu estación de trabajo.

Ejercicio 5.2.3: (Conceptual) Configurando Bonding (Active-Backup)

- **Objetivo:** Entender los pasos para configurar una agregación de enlaces básica.
- *Requisitos:* Privilegios de superusuario (`sudo`). **VM de prueba con múltiples interfaces de red virtuales (ej: `eth0`, `eth1` o `enpXsY`, `enpZsY`) que no sean esenciales para la conectividad de la VM.**
- **Desarrollo Paso a Paso (Basado en un método, elige el de tu VM):**
  - **Método Tradicional Debian (`/etc/network/interfaces`):**
    1. Edita `/etc/network/interfaces` (`sudo vi /etc/network/interfaces`).
    2. Asegúrate de que las interfaces miembro (`eth0`, `eth1`) estén configuradas como `manual`.

       ```
       auto eth0
       iface eth0 inet manual
       auto eth1
       iface eth1 inet manual
       ```

    3. Añade la definición del bond:

       ```
       auto bond0
       iface bond0 inet static
           address 192.168.1.200
           netmask 255.255.255.0
           gateway 192.168.1.1
           bond-mode active-backup
           bond-slaves eth0 eth1
       ```

    4. Guarda, y reinicia la red (o el sistema). `sudo systemctl restart networking.service` (si aplica) o `sudo ifdown eth0 eth1 bond0 && sudo ifup eth0 eth1 bond0`.
    5. Verifica el estado: `ip addr show bond0`, `cat /proc/net/bonding/bond0`.
  - **Método Tradicional Red Hat (`/etc/sysconfig/network-scripts/`):**
    1. Crea `ifcfg-bond0` (`sudo vi /etc/sysconfig/network-scripts/ifcfg-bond0`):

```
DEVICE=bond0
TYPE=Bond
BOOTPROTO=static
IPADDR=192.168.1.200
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
ONBOOT=yes
BONDING_OPTS="mode=1 primary=eth0" # mode 1 es active-backup,
primary opcional
```

2. Modifica `ifcfg-eth0` (`sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0`):

```
DEVICE=eth0
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

3. Modifica `ifcfg-eth1` (igual que eth0, cambiando `DEVICE`):

```
DEVICE=eth1
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

4. Guarda, y reinicia la red (o el sistema). `sudo systemctl restart network.service` (si aplica) o `sudo ifdown eth0 eth1 bond0 && sudo ifup eth0 eth1 bond0`.
5. Verifica el estado: `ip addr show bond0`, `cat /proc/net/bonding/bond0`.

- **Métodos NetworkManager/Netplan/systemd-networkd:** Implican usar `nmcli`, `netplan apply`, o editar archivos de configuración específicos del servicio. Los pasos son más variados pero conceptualmente similares (crear el bond lógico, asignar interfaces físicas como miembros/esclavos).

- **Verificación de Failover (Conceptual):** Una vez configurado, puedes intentar deshabilitar una interfaz física miembro (ej: `sudo ip link set eth0 down`) y verificar que la conectividad persiste y que `/proc/net/bonding/bond0` muestra la otra interfaz como activa.

Ejercicio 5.2.4: (Conceptual) Configurando un Puente de Red (Bridging)

- **Objetivo:** Entender los pasos para configurar un puente de red simple (ej: para virtualización básica).
- *Requisitos:* Privilegios de superusuario (`sudo`). **VM de prueba con múltiples interfaces virtuales dedicadas.**
- **Desarrollo Paso a Paso (Basado en un método, elige el de tu VM):**

- **Método Tradicional Debian (`/etc/network/interfaces`):**
  1. Edita `/etc/network/interfaces` (`sudo vi /etc/network/interfaces`).
  2. Configura la interfaz física que será parte del puente como manual:

     ```
     auto eth0
     iface eth0 inet manual
     ```

  3. Define la interfaz de puente y asígnale las interfaces físicas:

     ```
     auto br0
     iface br0 inet static # o dhcp
         address 192.168.1.254
         network 192.168.1.0
         netmask 255.255.255.0
         broadcast 192.168.1.255
         gateway 192.168.1.1
         bridge_ports eth0
     ```

  4. Guarda, y reinicia la red (o el sistema). `sudo systemctl restart networking.service` o `sudo ifdown eth0 br0 && sudo ifup eth0 br0`.
  5. Verifica: `ip addr show br0`, `ip link show eth0`, `brctl show` (si `bridge-utils` está instalado).
- **Método Tradicional Red Hat (`/etc/sysconfig/network-scripts/`):**
  1. Crea `ifcfg-br0` (`sudo vi /etc/sysconfig/network-scripts/ifcfg-br0`):

     ```
     DEVICE=br0
     TYPE=Bridge
     BOOTPROTO=static # o dhcp
     IPADDR=192.168.1.254
     NETMASK=255.255.255.0
     GATEWAY=192.168.1.1
     ONBOOT=yes
     ```

  2. Modifica `ifcfg-eth0` (`sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0`):

     ```
     DEVICE=eth0
     TYPE=Ethernet
     BOOTPROTO=none
     ONBOOT=yes
     BRIDGE=br0
     ```

  3. Guarda, y reinicia la red (o el sistema). `sudo systemctl restart network.service` o `sudo ifdown eth0 br0 && sudo ifup eth0 br0`.
  4. Verifica: `ip addr show br0`, `ip link show eth0`, `brctl show` (si `bridge-utils` está instalado).

    - **Métodos NetworkManager/Netplan/systemd-networkd:** Implican usar `nmcli`, `netplan apply`, o editar archivos de configuración específicos. Conceptualmente similar (crear el bridge, añadir interfaces como puertos).
- **Verificación:** Después de configurar el bridge, la interfaz física (`eth0`) no debería tener dirección IP; la dirección IP se asigna al bridge (`br0`).

Ejercicio 5.2.5: Usando Herramientas de Troubleshooting en un Escenario Hipotético

- **Objetivo:** Aplicar varias herramientas para diagnosticar un problema.
- **Escenario Hipotético:** No puedes hacer SSH al servidor `server.example.com` (IP 192.168.1.10), pero sí puedes hacer ping a otros hosts en tu red local (ej: 192.168.1.1 - tu gateway).
- **Requisitos:** Acceso a la línea de comandos. Herramientas de troubleshooting instaladas (`ping`, `ip`, `tcpdump`, `nc`, `host`/`dig`).
- **Desarrollo Paso a Paso (Simulado):**
    1. Abre una terminal.
    2. **Paso 1: Verificar IP de destino (si usas hostname):** Ejecuta `host server.example.com`. Si no resuelve a 192.168.1.10, el problema es DNS (Ej. 109.4). Si resuelve, continúa.
    3. **Paso 2: Verificar conectividad IP básica:** Ejecuta `ping 192.168.1.10`. Si falla, el host está caído, no responde ICMP, o hay un problema en la ruta. Si `ping 192.168.1.1` funciona, tu red local y gateway están bien.
    4. **Paso 3: Verificar ARP (si el destino es local):** Si 192.168.1.10 está en tu red local (verifica tu máscara de subred con `ip addr show`), verifica la caché ARP: `ip neigh show 192.168.1.10`. Si no aparece o el estado es `FAILED`, tu máquina no puede comunicarse a nivel de Capa 2.
    5. **Paso 4: Verificar la ruta:** Ejecuta `ip route show`. Asegúrate de que hay una ruta para 192.168.1.10 (debería ser una ruta local si está en la misma subred, o la ruta por defecto vía tu gateway si está en otra red).
    6. **Paso 5: Verificar si el puerto SSH está abierto/escuchando en el destino:** Ejecuta `nc -zv 192.168.1.10 22`. Si falla ("Connection refused"), el servicio SSH no está corriendo, un firewall lo bloquea, o el puerto es incorrecto.
    7. **Paso 6: Analizar tráfico con `tcpdump`:** Ejecuta `sudo tcpdump -i <tu_interfaz> -nnvvS host 192.168.1.10 and port 22`. Intenta hacer SSH de nuevo desde otra terminal. ¿Ves paquetes TCP SYN saliendo? ¿Ves alguna respuesta SYN/ACK o RST del servidor? Esto te dirá si los paquetes llegan al servidor y cómo responde.
    8. **Paso 7: Verificar firewalls:** Revisa el firewall local en tu máquina y, si tienes acceso, en el servidor remoto.
    9. **Paso 8: Verificar el servicio en el servidor remoto (si tienes acceso):** Ejecuta `systemctl status ssh.service` (o `sshd.service`). Verifica los logs de SSH.