

Tarea N°1: Mininet

Esta tarea está orientada al uso de la plataforma de emulación de redes conocida como Mininet. El objetivo consiste en introducir al alumno en los comandos más básicos de dicha plataforma, así como sus posibles aplicaciones prácticas. La tarea también incluye la simulación de un caso real de tele-operación a larga distancia.

SND

Las SND (Software Defined Networks) son un paradigma de redes de computadores que introducen el uso de recursos programables, de forma de reducir la dependencia del hardware. Para esta tarea, usted hará uso de la plataforma de virtualización de redes conocida como Mininet (mininet.org), la cual le permitirá implementar redes virtuales de forma realista a través de líneas de comandos o scripts.

Instalación

Cualquier problema con la instalación de Mininet, puede referenciarse en mininet.org/download/ en la cual ésta guía está basada.

1.- Instalar VirtualBox www.virtualbox.org/wiki/Downloads

*Usar versión 4.3 o anterior. Versiones más nuevas presentan errores con Mininet.

*Si lo desea puede usar alguna otra plataforma de virtualización.

2.- Descargar la imagen de Mininet desde [/github.com/mininet/mininet/wiki/Mininet-VM-Images](https://github.com/mininet/mininet/wiki/Mininet-VM-Images)

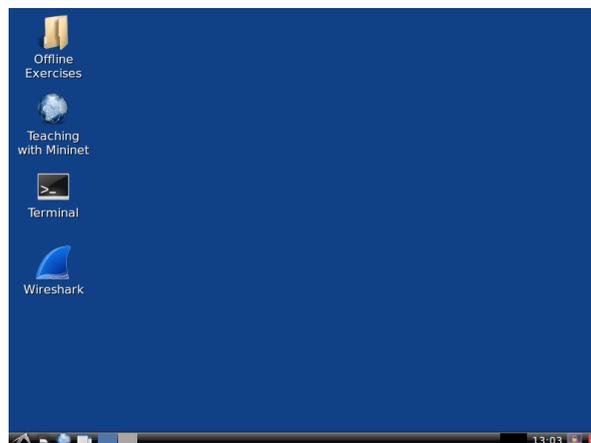
3.- Doble click en archivo .ovf, esto importará y creará la imagen con las configuraciones necesarias (asignará RAM, configuraciones por defecto, etc.).

4.- Correr Mininet desde la VM. Se le solicitarán los siguientes *user* y *pass*:

***User: mininet**

***Pass: mininet**

Luego de ingresar el *user* y la *pass* debería ver una pantalla como la siguiente:



Verifique a través del web browser de Mininet que tiene conexión hacia Internet. Esto será necesario para algunos pasos de la Tarea. En caso contrario deberá revisar la configuración de red de su VM.

Comandos básicos en Mininet

La siguiente guía incluye lo básico que necesitará para la Tarea. Para más información puede seguir la *walkthrough* de la página mininet.org/walkthrough/. Notará que la configuración del teclado en la máquina virtual no es la correcta. Para solucionar este problema, consulte el anexo al final del documento.

1.- Línea de comandos:

Note que para la guía se usarán la siguiente sintaxis:

\$: Precede comandos de Linux (como cd, ls, etc)

mininet> : Precede a comandos de Mininet

Para comenzar, abra la terminal y escriba lo siguiente:

```
$ sudo mn -h
```

Esto entregará una lista con los comandos más típicos de Mininet. En general, todos los comandos de Mininet desde la terminal de Linux se ejecutan con el comando *mn*. Para diseñar una red básica, puede simplemente introducir el comando sin ningún parámetro. Esto creará una red simple con un switch *s1* y dos hosts *h1* y *h2*. Puede crear otras topologías usando `--topo=TOPOLOGIA`. Algunas topologías admitidas son

```
--topo simple,<n>  
--topo linear,<x>,<y>  
--topo tree,<d>,<f>
```

Donde en este caso, la topología *simple* consiste en un switch y *n* hosts conectados a dicho switch, *linear* crea *x* switches con *y* hosts conectados a cada switch. En este caso, todos los switches están conectados entre sí. Por último, *tree* crea una topología de árbol, con una profundidad *d* y fanout de *f*.

Puede revisar los demás comandos viendo la ayuda (`mn -h`). Por ahora nos centraremos en los comandos mininet, para lo cual se usará una red simple con 1 switch y 2 hosts. Usando `sudo mn` se crea dicha red por defecto.

Ahora en la terminal de Mininet, puede probar los siguientes comandos:

```
mininet> help  
mininet> nodes  
mininet> net  
mininet> dump
```

Estos comandos muestran la ayuda, además de la información de los nodos, la red. Además, puede usar el comando *ifconfig* para obtener información de la interfaz de red de un host, de la misma forma que un equipo real:

```
mininet> h1 ifconfig -a
```

Note que la sintaxis pide especificar el nodo al comienzo del comando. Entre otros comandos de interés se encuentran *ps*, *xterm* y *ping*. Puede revisar la documentación de Mininet para probar más ejemplos.

2.- Wireshark en Mininet

El hecho de que los host creados sean emulados implica que los paquetes que son enviados y recibidos son reales. Esto permite, por ejemplo, la capacidad de visualizar dichos paquetes usando Wireshark. Usando la misma red creada, pruebe a ver el nombre de la interfaz del switch *s1*. Luego, abra Wireshark y comience monitorear el tráfico de alguna de las interfaces asociadas a los hosts (*s1-eth*) y ejecute el siguiente comando:

```
mininet> h1 ping -c 4 h2
```

El comando enviará cuatro pings desde el host *h1* al host *h2*. Podrá visualizar en Wireshark todo el tráfico asociado a los ping (requerimientos ARP y paquetes de request y reply) como si se tratara de una red real.

3.- API de Python

Para diseñar topologías más complejas y personalizadas, es recomendable usar la API de Python. Un ejemplo de esto se puede ver en la carpeta `/mininet/custom/topo-2sw-2host.py` el cual muestra cómo definir una topología de 2 switches y 2 hosts.

Mininet define cada topología como una **clase** que se puede instanciar como objeto usando Python. Cada clase crea sus nodos y enlaces usando **métodos**, para los cuales se pueden definir distintos parámetros. Los métodos más importantes para la creación de topología son **addHost**, **addLink**, y **addSwitch**.

`addHost(<nombre>)`: Crea un host con un nombre determinado por <nombre>

`addSwitch(<nombre>)`: Crea un switch con un nombre determinado por <nombre>

Nota: `addHost` y `addSwitch` no aceptan nombres que no sean de la forma *h1,h2,h3...* o *s1,s2,s3...* etc.

`addLink(<n1>, <n2>, bw=<BW>, delay=<D>,loss=<L>)`: Crea un enlace entre los nodos *n1* y *n2*, con los parámetros opcionales *BW*(Ancho de banda), *delay*, y porcentaje de pérdidas *L*. También admite otros parámetros como *jitter*, *latencia*, etc.

```
self.addLink( host, switch, bw=10, delay='5ms',loss=1)
```

Existe varias formas de implementar una topología a partir de un script. Una de las formas más simples se ejemplifica en el archivo adjunto `test_net.py`. Para la actividad se recomienda usar este archivo como base, ya que basta con modificar los métodos que crean los nodos y enlaces según se desea. Luego, para correr la red en mininet basta con ejecutar el script.

También puede intentar implementarlo usando el siguiente ejemplo de mininet como base:

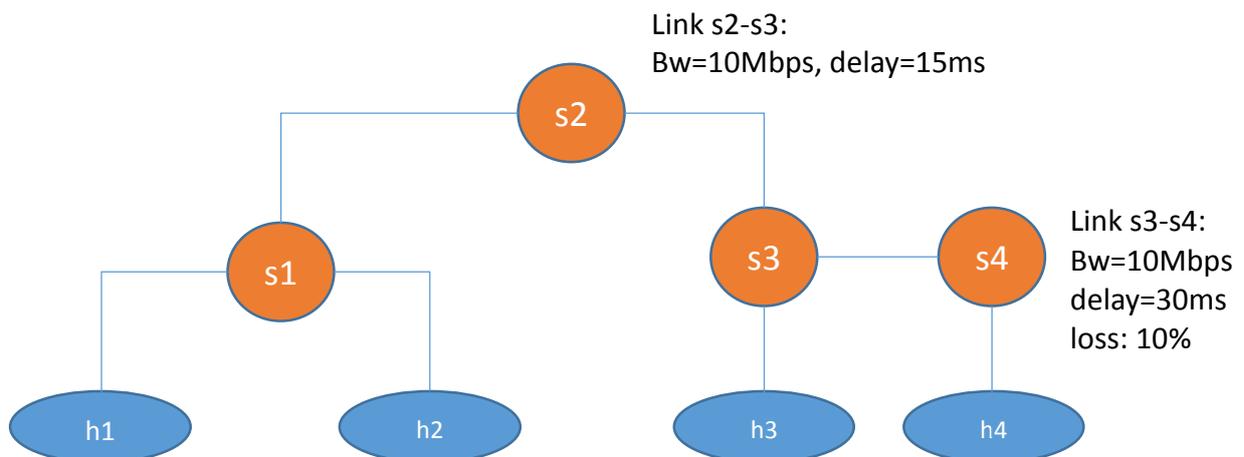
```
$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo
```

Modificar este ejemplo requiere de un conocimiento más detallado de Python, pero puede usarlo si lo desea.

Actividad a realizar

Las siguientes actividades deberán ser realizadas usando Mininet. Su tarea deberá incluir capturas de los pasos realizados, mostrando claramente la captura de paquetes en Wireshark además de una explicación de lo que se muestra. Si algún paso lo requiere, deberá incluir su código en python.

1.- Implemente en Mininet la siguiente red:



Puede usar línea de comandos de Mininet o la API de Python. Muestre las IP de cada host usando `ifconfig`. Haga 10 pings desde `h1` hasta `h2` usando la terminal `xterm`. Muestre los paquetes en Wireshark (incluido el requerimiento ARP). Luego, repita el procedimiento entre `h1` y `h3`. Compruebe que efectivamente se producen los retardos asociados a estos dos enlaces. Ejecute una cantidad suficientemente alta de pings entre `h3` y `h4` hasta lograr un error del 5% (95% de intervalo de confianza) que permita comprobar el porcentaje de pérdida de dicho enlace.

Referencia sobre intervalo de confianza:

<http://profesores.elo.utfsm.cl/~agv/miscellaneous/confidenceInterval/IntervaloConfianzaPromedio.html>

2.- Monte un servidor http en el host *h1* . Luego, haga un requerimiento GET al servidor desde el host *h2*. Para ello, use los siguientes comandos.

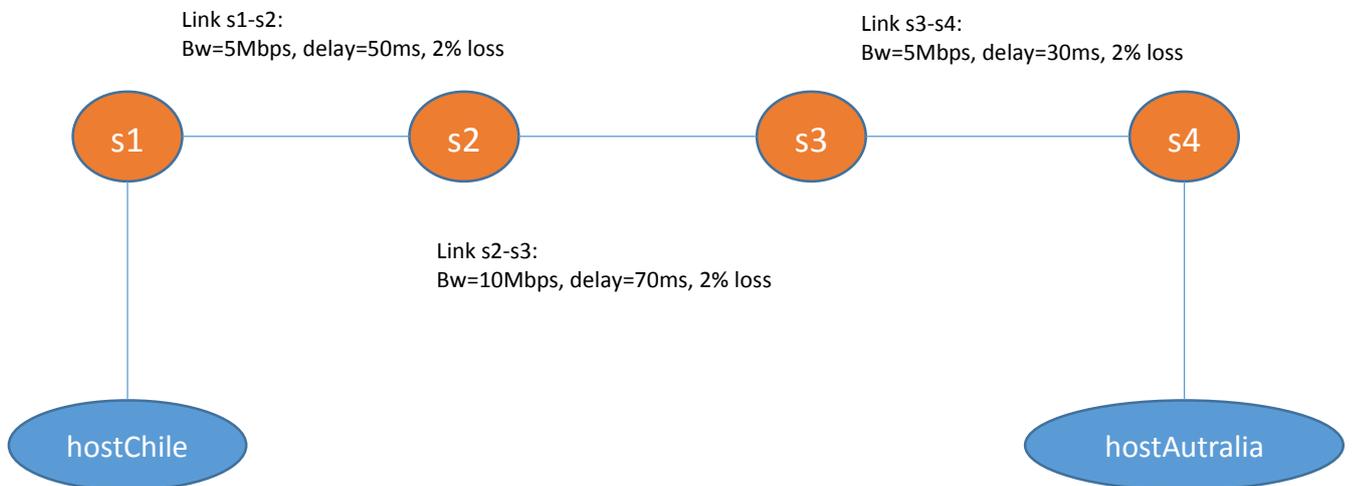
```
mininet> h1 python -m SimpleHTTPServer 80 &  
mininet> h2 wget -O - h1
```

Visualice los paquetes enviados en Wireshark, mostrando claramente el requerimiento GET. Compruebe que las IP corresponden a las de cada host.

3.- Ingrese a la carpeta `/mininet/examples/` y ejecute el script `nat.py`. Este archivo crea una red con topología *tree* con conexión a internet usando el protocolo NAT.

- Compruebe que los hosts efectivamente tienen conexión a internet haciendo un ping a alguna página conocida. Use el DNS (e.g. www.unapagina.cl) en vez de la IP. Muestre los paquetes correspondientes por Wireshark.
- Repita el punto 2 sobre alguno de los hosts de la red. Use el browser de Mininet para ingresar a la página web creada en el host. Use la IP de dicho host para ingresar.

4.- Implemente la siguiente red en un script en Python, que emulará un enlace entre Chile y Australia.



a.- Ejecute en mininet un iperf entre los hosts de Chile y Australia. Interprete la información entregada en relación a los parámetros del enlace.

b.- Abra Wireshark y he inicie una captura de paquetes en la interfaz de h1 y s1. Ejecute una transferencia de archivos usando FTP h1 y h2. Para ello siga las siguientes instrucciones:

1.- Cree un archivo de 10 Mb usando

```
fallocate -l 10MiB un-archivo.png
```

2.- Instale un servidor FTP en Mininet usando "sudo apt-get install ftpd".

3.- Inicie simultáneamente las terminales xterm para los host h1 y h2.

4.- Inicie un servidor FTP en el host h2 usando inetd.

5.- Inicie una captura tcpdump (a un archivo) para paquetes entre h1 y h2 en el puerto 21. e.g:

```
tcpdump -w ftp-transfer host 10.0.0.1 and host 10.0.0.2 and port 21
```

6.- En el host h1, cambie el directorio a /tmp, use "cd /tmp". Luego, inicie una sesión ftp usando el comando ftp 10.0.0.2 en una terminal. Ingrese usando Name: mininet y Password: mininet.

7.- Obtenga el archivo creado usando get (e.g. get un-archivo.png)

8.- Responda lo siguiente:

- ¿Qué problemas puede observar en la transmisión TCP de paquetes? ¿Por qué se producen? ¿Cómo resuelve TCP este problema?
- ¿Sería conveniente usar un enlace UDP para la transferencia de archivos en un enlace de este tipo? Justifique.

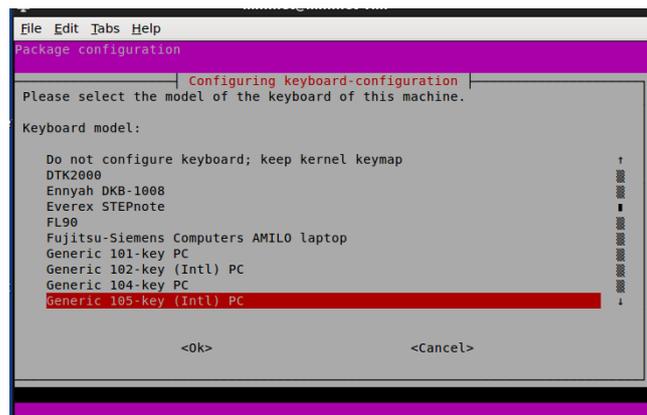
Adjunte para este desarrollo capturas con la red creada en Mininet (una vez ejecutado el script). También adjunte capturas de la transmisión ftp (proceso de transmisión y una vez finalizada la transferencia).

Anexo – Cambiar configuración de teclado en mininet

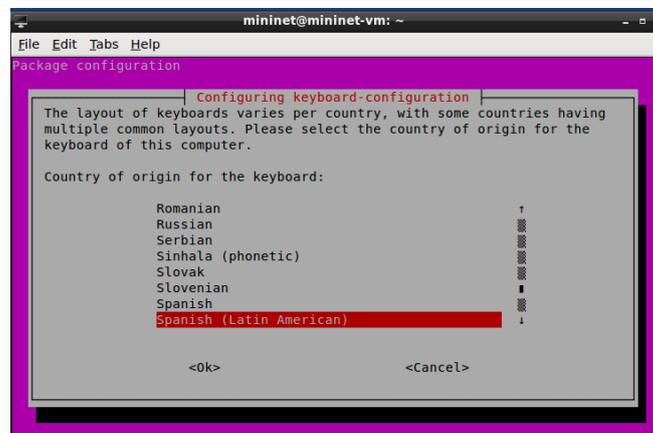
Cuando empiece a usar la terminal de comandos notará que la configuración de teclado no es la correcta. Para solucionar esto, ejecute el siguiente comando en la terminal:

```
$ sudo dpkg-reconfigure keyboard-configuration
```

Con esto debería aparecer la siguiente pantalla:



Seleccione la opción que sale por defecto. En la siguiente pantalla debería aparecer el siguiente menú:



Aquí, seleccione la opción *Spanish (Latin American)*. Luego, seleccione por defecto las demás opciones. Con esto, el teclado quedará configurado correctamente.