

COMANDOS DIAGNÓSTICO DE REDES

PARTE 2

```
Simbolo del sistema
9  be2927.ccr21.elp01.atlas.cogentco.com [154.54.29.222]
10 be2930.ccr32.phx01.atlas.cogentco.com [154.54.42.77]
11 be2932.ccr42.lax01.atlas.cogentco.com [154.54.45.162]
12 be3360.ccr41.lax04.atlas.cogentco.com [154.54.25.150]
13 38.142.236.35
14 unassigned.cdn77.com [185.229.188.55]
15 unn-185-152-67-178.datapacket.com [185.152.67.178]

Procesamiento de estadísticas durante 375 segundos...
Origen hasta aquí Este Nodo/Vínculo
Salto RTT Perdido/Enviado = Pct Perdido/Enviado = Pct Dirección
0 [192.168.1.11]
1 32ms 0/ 100 = 0% 0/ 100 = 0% 192.168.1.1 [192.168.1.1]
2 56ms 0/ 100 = 0% 0/ 100 = 0% host-7.1.217.201.copaco.com.py [201.217.1.7]
3 50ms 0/ 100 = 0% 0/ 100 = 0% host-77.0.217.201.copaco.com.py [201.217.0.77]
4 49ms 0/ 100 = 0% 0/ 100 = 0% host-27.0.217.201.copaco.com.py [201.217.0.27]
5 202ms 1/ 100 = 1% 1/ 100 = 1% so-11-2-0.miami19.mia.seabone.net [89.221.41.34]
6 180ms 0/ 100 = 0% 0/ 100 = 0% be6762.ccr21.mia03.atlas.cogentco.com [154.54.9.17]
7 191ms 0/ 100 = 0% 0/ 100 = 0% be3400.ccr21.mia01.atlas.cogentco.com [154.54.47.17]
8 224ms 0/ 100 = 0% 0/ 100 = 0% be3569.ccr41.iah01.atlas.cogentco.com [154.54.82.241]
9 243ms 0/ 100 = 0% 0/ 100 = 0% be2927.ccr21.elp01.atlas.cogentco.com [154.54.29.222]
10 254ms 1/ 100 = 1% 1/ 100 = 1% be2930.ccr32.phx01.atlas.cogentco.com [154.54.42.77]
11 260ms 0/ 100 = 0% 0/ 100 = 0% be2932.ccr42.lax01.atlas.cogentco.com [154.54.45.162]
12 --- 100/ 100 =100% 100/ 100 =100% be3360.ccr41.lax04.atlas.cogentco.com [154.54.25.150]
13 --- 100/ 100 =100% 100/ 100 =100% 38.142.236.35
14 --- 100/ 100 =100% 100/ 100 =100% unassigned.cdn77.com [185.229.188.55]
15 282ms 0/ 100 = 0% 0/ 100 = 0% unn-185-152-67-178.datapacket.com [185.152.67.178]

Trazo completa.
```

ENSAYO N*5

NOMBRE DEL ESTUDIANTE: LLANO,ROSARIO

NOMBRE DEL PROFESOR: MARTÍNEZ,EUGENIO

MATERIA: INTRODUCCIÓN A REDES

CARRERA: TEC. EN DESARROLLO DE SOFTWARE

PATHPING

El comando pathping es una **herramienta de utilidad de línea de comandos de los sistemas operativos Windows**. Se suele utilizar para solucionar problemas de red, especialmente los relacionados con la latencia y el rendimiento de dicha red.

Considerada una alternativa a tracert y ping, combina lo mejor de ambas: al igual que la primera, pathping rastrea todos los saltos intermedios entre un origen y un destino. Y del mismo modo que ping, proporciona datos de latencia y pérdida de paquetes. Sin embargo, pathping mostrará resultados más detallados, ya que puede detectar qué routers tienen problemas y dónde están ubicados.

Así es como se ven los resultados de un comando pathping:

Cuando se introduce el comando, se listan todos los routers de origen y destino. Calcula las estadísticas para un tiempo determinado y recopila información sobre todos los routers. En el caso anterior, durante 200 segundos. A continuación, se muestran los datos.

Al igual que ping y tracert, pathping utiliza el **protocolo ICMP**.

La **sintaxis de pathping** es bastante sencilla: abre un símbolo del sistema y escribe pathping seguido de la URL o dirección IP.

La sintaxis básica para utilizar pathping es:

```
pathping -n [target]
```

Por ejemplo, este es el comando para realizar un pathping en facebook.com:

```
pathping www.facebook.com
```

Para **solucionar problemas de red**, puedes personalizar pathping con varios parámetros. Si bien el comando suele ser lento, es posible acelerar el proceso con estos parámetros.

Por ejemplo, cuando se utiliza '/n', pathping no resuelve las direcciones IP de los routers implicados. Una vez más, tomando el ejemplo de facebook.com, el comando tendrá el siguiente aspecto:

```
pathping /n www.facebook.com
```

También permite especificar el número máximo de saltos antes de detener el pathping usando /h y la cantidad de peticiones echo enviadas a cada router con /q. El comando se detendrá después de 30 saltos por defecto, pero si te enfrentas a un problema de red, esto podría significar que el paquete nunca llegó al destino. Asimismo, es posible acelerar el proceso reduciendo el número de peticiones echo utilizando el parámetro /q.

Para usar el **parámetro /h**, el comando se verá así:

```
pathping /h 4 www.facebook.com
```

En el ejemplo anterior, el número de saltos está limitado a 4.

Para utilizar el **parámetro /q**, la sintaxis será la siguiente:

```
pathping /q 20 www.facebook.com
```

Con pathping, también es posible especificar el tiempo entre pings consecutivos (/p), el de espera para una respuesta (/w), y muchos otros parámetros e incluso ordenarle que el comando utilice sólo IPv4 (/4) o IPv6 (/6).

En **Linux**, la sintaxis es similar, salvo que se usa mtr en lugar de pathping. Por ejemplo:

```
mtr www.facebook.com
```

O bien,

```
mtr 157.240.23.35
```

pathping se utiliza para medir la calidad de una conexión de red informática de host a host, enviando solicitudes del Protocolo de Mensajes de Control de Internet o Internet Control Message Protocol (ICMP) a cada destino apuntado. A continuación, muestra el tiempo de respuesta y la pérdida de paquetes de los devueltos desde cada destino. Si el extravío no es cero y el tiempo de ida y vuelta sigue siendo inferior a 1000 milisegundos, es probable que la conexión esté experimentando una latencia significativa

Pathping es una herramienta estándar utilizada para solucionar problemas de red. Generalmente, primero encuentra los routers entre el origen y el destino.

Luego, la latencia y la pérdida de paquetes entre el origen y todos los routers a lo largo del camino, mediante el envío de solicitudes echo.

La sintaxis del comando es bastante simple: pathping seguido de la dirección IP o la URL. Puedes personalizarlo con muchos parámetros. Aquí detallamos los más comunes:

- /q - para establecer el número de peticiones echo
- /h - para fijar el número máximo de saltos
- /w - para instaurar el tiempo de espera de una respuesta
- /n - para evitar que el comando resuelva la dirección IP de los routers

Netsh

El comando netsh es una herramienta esencial para administrar y solucionar problemas de red en entornos Windows. Ya sea que necesites configurar parámetros de red, administrar el firewall, gestionar conexiones inalámbricas o realizar diagnósticos de red, netsh te brinda la capacidad de realizar todas estas tareas de manera eficiente a través de su interfaz de línea de comandos.

CONFIGURACIÓN DE INTERFAZ DE RED: Una de las funciones más comunes de netsh es la configuración de parámetros de red. A través de este comando, podrás cambiar la dirección IP, la máscara de subred, la puerta de enlace y los servidores DNS de tu computadora. También puedes agregar y eliminar interfaces de red, habilitar o deshabilitar adaptadores, entre otras acciones. Para configurar la interfaz de red con netsh, simplemente debes especificar los valores que deseas modificar mediante los comandos correspondientes.

- **Ejemplo:** netsh interface ipv4 set address
name="Ethernet" static 192.168.1.100
255.255.255.0 192.168.1.1

CONFIGURACIÓN DE FIREWALL: Con este comando, podrás crear reglas de entrada y salida, permitir o bloquear aplicaciones o puertos específicos, y establecer excepciones para el tráfico de red. Esto te brinda un mayor control sobre la seguridad de tu computadora y te permite protegerla de posibles amenazas externas.

- **Ejemplo:** netsh advfirewall firewall add rule
name="Allow Port 80" dir=in action=allow
protocol=TCP localport=80

ADMINISTRACIÓN DE REDES INALÁMBRICAS: Con este comando, podrás conectar tu equipo a una red inalámbrica, configurar parámetros de conexión, administrar perfiles de red y solucionar problemas de conectividad.

Ejemplo: `netsh wlan connect name="NombreDeRed"`

***DIAGNÓSTICO DE RED:** netsh también cuenta con herramientas de diagnóstico y monitoreo de red. Estas herramientas te permiten realizar pruebas de conectividad ICMP, verificar el estado de los puertos TCP/UDP, consultar información de enrutamiento y mucho más. Si te encuentras con problemas de conectividad o necesitas analizar el estado de tu red, utilizar netsh puede ayudarte a identificar y resolver problemas fácilmente.

- **Ejemplo:** `netsh interface tcp show global`

CURL

Curl (Client URL) es una herramienta de línea de comandos, que permite transferir datos hacia o desde un servidor sin interacción del usuario utilizando la biblioteca libcurl. curl también se puede utilizar para solucionar problemas de conexión.

El comando curl es compatible con la siguiente lista de protocolos:

- HTTP y HTTPS
- FTP y FTPS
- IMAP e IMAPS
- POP3 y POP3S
- SMB y SMBS
- SFTP
- SCP
- TELNET
- GOPHER
- LDAP y LDAPS
- SMTP y SMTPS

Comando curl: sintaxis y opciones

```
curl <options> <url>
```


| Opciones | Descripción |
|----------|---|
| -# | Muestra una barra de progreso al descargar un archivo usando curl |
| -o | Guarda el archivo descargado con un nombre diferente especificado en el sistema del cliente |
| -O | Cuando se usa, guardará el archivo descargado con el mismo nombre que la <url> |
| -T | Se utiliza para cargar un archivo a un servidor FTP. |
| -X | Accede a la <url> a través de un servidor proxy. |
| -w | Hacer que curl muestre información en stdout después de una transferencia completa |

● Guardar un archivo usando el comando curl

Cuando usas el comando curl sin ninguna <opción>, simplemente imprime el código fuente de la página web a la que estás enviando la solicitud. Para guardar este resultado en un archivo en su sistema de archivos Linux, use el comando curl con el indicador -o:

```
curl <options> -o <file_name_to_save> <url>
```

Por ejemplo:

```
curl -o index.html https://test.rebex.net/
```

● Probar si un servidor está disponible o no

Con el indicador -I, puede utilizar el comando curl en Linux para comprobar si un servidor está disponible o no.

La sintaxis es:

```
curl <options> -I <url>
```

• Acceder a las cookies con el comando curl

Cada vez que visita una URL, cierta información se almacena en su sistema, que luego se utiliza la próxima vez que visite la misma URL. Entonces, simplemente use la --cookie-jar bandera con el comando curl para acceder a las cookies almacenadas en su sistema de archivos Linux para la <url> dada:

```
curl <options> --cookie-jar <file_to_write_cookies> <url>
```

Por ejemplo, aquí está el comando para almacenar todas las cookies de <https://test.rebex.net/> en el archivo cookies.txt:

```
curl --cookie-jar cookies.txt https://test.rebex.net/
```

• Descargar archivos desde el servidor FTP

Normalmente, al acceder a un servidor FTP seguro, debe ingresar la contraseña cada vez que inicia sesión en el servidor. Pero con curl, puedes especificar directamente el nombre de usuario y la contraseña con la siguiente sintaxis:

```
curl -u <username>:<password> -O <url>
```

Por ejemplo, puede utilizar el siguiente comando para descargar el archivo readme.txt del servidor FTP test.rebex.net con «demo» como nombre de usuario y «contraseña» como contraseña:

```
curl -v -u demo:password -O ftp://test.rebex.net/readme.txt
```

