

## fping Roland Schemers & David Papp

¡De antemano no confundir **fping** con [ping](#)! Por eso le dedicamos una entrada aparte de ping, como dice el refrán "juntos pero no revueltos", veamos qué es **fping**, su [historia y su uso](#)  
¡Aprendamos!

### ¿Qué es fping?

Le dedicamos su entrada totalmente aparte porque no todas las distribuciones GNU/Linux lo traen por defecto sino que tenemos que instalarlo desde los repositorios oficiales (de cada distro). El programa se caracteriza por trabajar de manera masiva conectandose a varios dispositivos por medio del protocolo [ICPM](#) enviando paquete de datos para hacer un diagnóstico y determinar a ciencia cierta si tenemos conexión con un anfitrión u objetivo (servidor, enrutador, tableta, teléfono móvil celular que use Android, etcétera). Este método nos ahorra tiempo en conexión y además ofrece resultados personalizados que podemos pasar luego a software de monitorización de servidores .

### Historia de fping.

En 1992 Roland Schemers, de la Universidad de Stanford (California, Estados Unidos) convirtiéndose *de facto* en una importante herramienta de diagnóstico de redes y mantuvo el programa sin llegar a liberar la versión 2.x.

En 2001 RL "Bob" Morgan y David Papp se dedicaron a desarrollar aún más destacando como hitos importantes la compilación en Mac OS (Benjamin Gardiner, julio 2001), soporte al protocolo IPv6 (Jeroen Massar, enero 2002) y su propio sitio web comercial -el software libre no quiere decir necesariamente gratis- [www.fping.com](http://www.fping.com) (Thomas Dzubin, septiembre de 2002, hoy fuera de línea y el dominio se encuentra en venta).

Para el año 2007 Thomas Dzubin decide mudar el código fuente y sitio web a [sourceforge.net](https://sourceforge.net) donde aún reposa una copia del programa y de donde tomamos estos [datos históricos](#).

Lamentablemente hasta 2011 mantuvieron el programa y aunque parecía destinado a quedarse estancado vino la mano invisible del software libre (de nuevo) a su rescate: [David Schweikert](#) intentó contactar al autor y no obtuvo respuesta alguna y decidió retomar el proyecto aparentemente abandonado. Para darle un renacimiento decidió comenzar a publicar con la nueva numeración 3.X y para aclimatarla a los nuevos tiempos le colocó un sitio en [GitHub](#) donde invita a

bifurcar el código. Para liberar la versión 3.0 el señor David Schweikert invirtió [438 sesiones de codificación](#). Nos enteramos también [por medio de este artículo](#) sobre "*listas grises de correo electrónico*" que David Schweikert también ha implementado en **lenguaje Perl** [una utilidad a tal efecto](#), así que como veís el sr. David es muy prolijo *¡gracias, mil gracias por "mover" al mundo de la programación!*

Entre los hitos de la versión 4.X destaca: un solo binario para protocolo IPv4 e IPv6, [nombres de opciones largas](#) para todos los parámetros, descarte de respuestas si exceden el parámetro de tiempo especificado (algo así como decimos por acá: "tarde piastes, pajarito"). El resumen completo lo podrán leer acá en [este enlace web](#). También podemos participar en el [grupo de correo electrónico alojado por cortesía de Google](#).

De acuerdo a la licencia que rige **siempre debemos dar el debido crédito a todos y cada uno de los autores y colaboradores del proyecto.**

## [Autores y colaboradores](#) de fping.

- Roland J. Schemers III, Universidad de Stanford, concepto y versiones 1.x.
- RL "Bob" Morgan, Universidad de Stanford, versiones 2.x.
- David Papp, versiones 2.3x y menores a 3.0.
- David Schweikert, versions 3.X y superiores.

## Instalando fping.

Instalar **fping** se puede hacer de varias maneras:

### Con apt-get:

La manera más fácil es invocando **apt-get** (nuestro instalador favorito para GNU/Debian y sucedáneos) desde nuestros repositorios configurados, al día de hoy obtuvimos la versión 3.13:

```
apt-get install fping
```

## Descargando y compilando desde su página web [fping.org](http://fping.org)

Para descargar los binarios (y compilarlos) de la versión 4.0 podemos usar el siguiente código:

```
wget http://fping.org/dist/fping-4.0.tar.gz tar --extract -f fping-4.0.t
```

```
ar.gz cd fping-4.0/ ./configure make sudo make install
```

Nosotros hicimos la prueba pero cometimos el error de compilarlo como administrador y nuestro ejecutable debe ser lanzado con el comando **sudo**, mirad:

## Descargando desde GitHub.

Con Git simplemente debemos clonar desde el enlace:

```
git clone https://github.com/schweikert/fping.git
```

y repetimos los pasos que anteriormente vimos, ***pero con la ventaja de que si realizamos alguna mejora en el código fuente, y por medio del control de versiones, podremos proponerle a los encargados de fping.org el incorporarlo a la próxima versión de fping*** (atención: serán exigentes en el código que subamos, podrían negarse a agregarlo o incluso aceptarlo y mejorarlo así que salimos todos ganando).

## ¿Cómo funciona fping?

Como dijimos **fping** hace conexiones masivas pero detallaremos ahora cómo lo hace: para cada dispositivo u *objetivo* abre un *socket* por el cual envía una petición **ping** y lo deja abierto esperando la respuesta para de inmediato abrir otro *socket* y pasar al próximo objetivo. Si un objetivo no responde se vuelve a reintentar (más adelante podremos configurar el número de intentos) y si responde pues cierra el *socket* y hace debida cuenta en las estadísticas (también podremos configurar el informe: ver solamente los que responden o solo los que no responden o ambos, incluso ver los que no se puede resolver su dirección IP por dominio we, más adelante veremos ejemplos de ello).

## Opciones de fping.

Utilizar **fping** es más sencillo que **ping** ya que tiene valores predeterminados para rápidamente conocer el estado de cualquier dispositivo (que llamaremos *objetivo*), ejemplo:

```
jimmy@KEVIN:~$ fping
```

```
cantv.net cantv.net is alive jimmy@KEVIN:~$ fping
cantv.com.ve google.com google.com is alive cantv.com.ve is unreachabl
e
```

Sin embargo, como a nosotros "nos gusta complicarnos la vida" pues acá explicamos cada uno de los parámetros (que es el nombre correcto) para hacer el llamado a **fping**, veamos en detalle.

### **parámetro "-a":**

Muestra los objetivos que tienen conexión, recordemos que la novedad en **fping** es analizar varios objetivos al mismo tiempo, ejemplo:

```
jimmy@KEVIN:~$ fping
ubuntu.com debian.org cantv.com.ve ubuntu.com is alive debian.org is a
live cantv.com.ve is unreachable jimmy@KEVIN:~$ fping -a
ubuntu.com debian.org cantv.com.ve ubuntu.com debian.org
```

### **parámetro "-v":**

Muestra la versión que tenemos instalada:

```
jimmy@KEVIN:~$ fping
-v fping: Version 3.13 fping: comments to david@schweikert.ch
```

### **parámetro "-A":**

Al probar conexión con un nombre de dominio, esta opción muestra el o los objetivo(s) por dirección IP:

```
jimmy@KEVIN:~$ fping
-A ubuntu.com debian.org cantv.com.ve 91.189.94.40 is alive 149.20.4.1
5 is alive 161.196.27.230 is unreachable
```

### **parámetro "-r n":**

Número de reintentos a cada objetivo, por defecto es tres.

### parámetro "-b n":

Tamaño de paquete para enviar, si especificamos cero envía mínimo 28 bytes, al especificar uno envía 29 y así sucesivamente hasta un máximo de 4096 bytes (algunos dispositivos *pueden* negarse a responder si el valor es alto).

### parámetro "-c n":

Número de paquetes a enviar a cada objetivo, por defecto 1 (**ping** tiene una opción igual).

### parámetro "-C n":

Igual que -c, el resultado del reporte en formato detallado. Veamos un ejemplo que ilustra las opciones "-r" "-b" y "-C" debidamente coloreada por nosotros para resaltar las diferencias y puntos clave:

```
jimmy@KEVIN:~$ fping
-r 7 -b 2 -C 3 python.org fping.org  fping.org  : [0], 30 bytes, 273 ms
(273 avg, 0% loss)  fping.org  : [1], 30 bytes, 277 ms (275 avg, 0% loss)
  python.org : [1], 30 bytes, 330 ms (330 avg, 50% loss)  fping.org  : [2
], 30 bytes, 272 ms (274 avg, 0% loss)  python.org : [2], 30 bytes, 325 m
s (327 avg, 33% loss)    python.org : - 330.20 325.48  fping.org  : 273.5
6 277.55 272.27  jimmy@KEVIN:~$ fping -A
-r 7 -b 2 -C 3 python.org fping.org  104.27.189.208 : [0], 30 bytes, 77.
4 ms (77.4 avg, 0% loss)  23.253.135.79  : [0], 30 bytes, 186 ms (186 av
g, 0% loss)  104.27.189.208 : [1], 30 bytes, 69.3 ms (73.4 avg, 0% loss)
  23.253.135.79  : [1], 30 bytes, 154 ms (170 avg, 0% loss)  104.27.189.2
08 : [2], 30 bytes, 69.7 ms (72.1 avg, 0% loss)  23.253.135.79  : [2], 30
bytes, 158 ms (166 avg, 0% loss)    23.253.135.79  : 186.10 154.74 158.
07 104.27.189.208 : 77.44 69.36 69.79
```

### parámetro "-B f":

Antes de rendirse, **fping** reenvía el paquete pero entre cada reintento espera una vez y media del tiempo de espera del intento anterior. Si queremos modificar este multiplicador debemos usar esta opción *y recordar que el separador decimal es el punto*.

### parámetro "-e":

Muestra el tiempo transcurrido sobre los paquetes de vuelta:

```
jimmy@KEVIN:~$ fping
ubuntu.org  ubuntu.org is alive  jimmy@KEVIN:~$ fping
-e ubuntu.org  ubuntu.org
is alive (179 ms)  jimmy@KEVIN:~$ fping
-c 3-e ubuntu.org  ubuntu.org : [0], 84 bytes, 234 ms (234 avg, 0% loss)
ubuntu.org : [1], 84 bytes, 262 ms (248 avg, 0% loss)  ubuntu.org : [2]
, 84 bytes, 265 ms (254 avg, 0% loss)  ubuntu.org : xmt/rcv/%loss = 3/3
/0%, min/avg/max = 234/254/265
```

## parámetro "-f archivo":

Esta es una de las fortalezas de **fping**: la capacidad que tiene de leer un archivo de texto los objetivos que queremos probar por conexión. Digamos, por ejercicio de imaginación, que tenemos varios dispositivos que reportan su dirección IP a una base de datos que tenemos en nuestra página web. *Saber la dirección IP de un objetivo es una cosa muy distinta a saber si dicho dispositivo está en línea.* Como lo más importante es tener la dirección IP -que ya tenemos grabada en nuestra base de datos- podemos emplear **wget** para obtenerlas y así guardarlas en un archivo de texto. **Dicho archivo de texto lo podremos poner a disposición de fping para pruebe todos y cada uno de ellos en un guion que escribamos y se ejecute cada cierto tiempo.**

Pongamos por caso que queremos probar la conectividad de los [siete primeros sitios web](#) mundiales según alexa.com, grabamos dichos dominios en un archivo de texto y le aplicamos **fping**:

```
jimmy@KEVIN:~$ nano lista.txt  jimmy@KEVIN:~$ fping -f
lista.txt  wikipedia.org is alive  youtube.com is alive  facebook.com is
alive  google.com is alive  google.co.in is alive  yahoo.com is alive  b
aidu.com is alive  jimmy@KEVIN:~$ fping -A
-f lista.txt  216.58.219.238 is alive  208.80.154.224 is alive  157.240.
14.35 is alive  172.217.10.238 is alive  172.217.9.227 is alive  98.138.2
53.109 is alive  111.13.101.208
is alive  jimmy@KEVIN:~$ fping -A -r 7 -b 2 -C 3
-f lista.txt  172.217.12.206 : [0], 30 bytes, 217 ms (217 avg, 0% loss)
208.80.154.224 : [0], 30 bytes, 124 ms (124 avg, 0% loss)  172.217.10.78
: [0], 30 bytes, 200 ms (200 avg, 0% loss)  172.217.7.3 : [0], 30 by
tes, 123 ms (123 avg, 0% loss)  206.190.36.45 : [0], 30 bytes, 156 ms (1
56 avg, 0% loss)  157.240.14.35 : [0], 30 bytes, 306 ms (306 avg, 0% los
```

```
s) 111.13.101.208 : [0], 30 bytes, 409 ms (409 avg, 0% loss) 172.217.12
.206 : [1], 30 bytes, 181 ms (199 avg, 0% loss) 172.217.10.78 : [1], 30
bytes, 160 ms (180 avg, 0% loss) 208.80.154.224 : [1], 30 bytes, 90.5 m
s (107 avg, 0% loss) 172.217.7.3 : [1], 30 bytes, 106 ms (115 avg, 0%
loss) 206.190.36.45 : [1], 30 bytes, 147 ms (152 avg, 0% loss) 157.24
0.14.35 : [1], 30 bytes, 265 ms (286 avg, 0% loss) 111.13.101.208 : [1]
, 30 bytes, 366 ms (388 avg, 0% loss) 172.217.12.206 : [2], 30 bytes, 10
1 ms (166 avg, 0% loss) 172.217.10.78 : [2], 30 bytes, 97.5 ms (152 avg
, 0% loss) 208.80.154.224 : [2], 30 bytes, 89.8 ms (101 avg, 0% loss) 2
06.190.36.45 : [2], 30 bytes, 147 ms (150 avg, 0% loss) 172.217.7.3
: [2], 30 bytes, 125 ms (118 avg, 0% loss) 157.240.14.35 : [2], 30 byte
s, 229 ms (267 avg, 0% loss) 111.13.101.208 : [2], 30 bytes, 354 ms (376
avg, 0% loss) 172.217.12.206 : 217.02 181.78 101.01 172.217.10.78 :
200.48 160.14 97.59 157.240.14.35 : 306.86 265.73 229.37 111.13.101.2
08 : 409.75 366.93 354.05 208.80.154.224 : 124.15 90.50 89.83 206.190.3
6.45
: 156.
18 147.92 147.14 172.217.7.3 : 123.82 106.56 125.41 jimmy@KEVIN:~$
```

## parámetro "-g":

Si NO utilizamos un archivo para la lista de objetivos, podemos decirle a **fping** que la genere, por supuesto dándole un inicio y un final (*no mostramos la salida completa del comando*):

```
jimmy@KEVIN:~$ fping -A -g 192.168.1.1 192.168.1.128 192.168.1.1 is aliv
e 192.168.1.47 is alive 192.168.1.80 is alive ICMP Host Unreachable fr
om 192.168.1.47 for ICMP Echo sent to 192.168.1.2 ICMP Host Unreachabl
e from 192.168.1.47 for ICMP Echo sent to 192.168.1.3 ICMP Host Unreachabl
e from 192.168.1.47 for ICMP Echo sent to 192.168.1.4 ICMP Host Unreacha
ble from 192.168.1.47 for ICMP Echo sent to 192.168.1.5 ICMP Host Unreac
hable from 192.168.1.47 for ICMP Echo sent to 192.168.1.6 ICMP Host Unre
achable from 192.168.1.47 for ICMP Echo sent to 192.168.1.7 ICMP Host Un
reachable from 192.168.1.47 for ICMP Echo sent to 192.168.1.8 ICMP Host
Unreachable from 192.168.1.47 for ICMP Echo sent to 192.168.1.9 ICMP Hos
t Unreachable from 192.168.1.47 for ICMP Echo sent to 192.168.1.10 ICMP
Host Unreachable from 192.168.1.47 for ICMP Echo sent to 192.168.1.11 IC
MP Host Unreachable from 192.168.1.47 for ICMP Echo sent to 192.168.1.12
ICMP Host Unreachable from 192.168.1.47 for ICMP Echo sent to 192.168.1.
13 ICMP Host Unreachable from 192.168.1.47 for ICMP Echo sent to 192.168
.1.14 ICMP Host Unreachable from 192.168.1.47 for ICMP Echo sent to 192.
168.1.15 ICMP Host Unreachable from 192.168.1.47 for ICMP Echo sent to 1
```

92.168.1.16 ( ... )

También podremos suministrar una máscara de red tal como "**fping -g 192.168.1.0/24**".

*Actualizado el domingo 24 de marzo de 2019*

Con el comando anterior obtendremos 256 líneas y es un tanto engorroso de leer, por lo que podremos filtrar con el comando **grep** las líneas que contengan la palabra clave **alive**. Esto es excelente porque en realidad solo nos interesa saber los que están conectados, en este ejemplo en particular nuestra red privada de área local. El otro detalle está en que también debemos filtrar los mensajes de error pero para ello no es necesario una palabra clave *sino filtrar los resultados que son emitidos por **STDERR*** :

```
jimmy@KEVIN:~$ sudo fping -g 192.168.1.0/24 2>/dev/null | grep alive 192
.168.1.1 is alive 192.168.1.47 is alive 192.168.1.174 is alive jimmy@K
EVIN:~$
```

## parámetro "-H n":

Sirve para especificar el tiempo de vida de los saltos, útil para largas distancias. Por lo general se considera que 200 milisegundos es un tiempo bastante prudencial como para decir que un objetivo está a tiempo, sin retardo apreciable. Aunque sabemos que las comunicaciones prácticamente van a la velocidad de la luz, muchas veces los objetivos que queremos llegar están en redes a los cuales hay que hacerles un largo rodeo por varias redes puente, de ahí la utilidad de establecer un tiempo de vida de los paquetes. Este resultado ilustra la diferencia entre hacer ping a un servidor que está lejos contra nuestro propio enrutador inalámbrico, para el primero al colocarle 3 milisegundos expira la consulta, para el segundo 10 milisegundos y nos sobra tiempo:

```
jimmy@KEVIN:~$ fping
-H 3 -c 3-e ubuntu.org ICMP Time Exceeded from 172.26.22.97 for ICMP Ec
ho sent to ubuntu.org (172.26.22.97) ICMP Time Exceeded from 172.26.22.9
7 for ICMP Echo sent to ubuntu.org (172.26.22.97) ICMP Time Exceeded fro
m 172.26.22.97 for ICMP Echo sent to ubuntu.org (172.26.22.97) ubuntu.
org : xmt/rcv/%loss = 3/0/100% jimmy@KEVIN:~$ fping
-H 200 -c 3-e ubuntu.org ubuntu.org : [0], 84 bytes, 224 ms (224 avg, 0
% loss) ubuntu.org : [1], 84 bytes, 180 ms (202 avg, 0% loss) ubuntu.or
g : [2], 84 bytes, 192 ms (199 avg, 0% loss) ubuntu.org : xmt/rcv/%los
s = 3/3/0%, min/avg/max = 180/199/
224 jimmy@KEVIN:~$ fping
```



```
-H 10 -c 5 -e 192.168.1.1 192.168.1.1 : [0], 84 bytes, 0.38 ms (0.38 avg, 0% loss) 192.168.1.1 : [1], 84 bytes, 0.29 ms (0.33 avg, 0% loss) 192.168.1.1 : [2], 84 bytes, 0.36 ms (0.34 avg, 0% loss) 192.168.1.1 : [3], 84 bytes, 0.28 ms (0.32 avg, 0% loss) 192.168.1.1 : [4], 84 bytes, 0.27 ms (0.31 avg, 0% loss) 192.168.1.1 : xmt/rcv/%loss = 5/5/0%, min/avg/max = 0.27/0.31/0.38
```

### parámetro "-i n":

Especifica la cantidad de milisegundos entre un paquete y otro. Dado el caso que no tenga éxito **fping** reintentará 1,5 veces el tiempo por defecto o el que especifiquemos (ver parámetro "-B").

### parámetro "-l":

Lo que **ping** tiene por defecto, **fping** hay que especificárselo: hacer conexión constante con los objetivos (algunos dispositivos pudieran apagar el servicio ping ante tal insistencia).

### parámetro "-p n":

Si utilizamos la opción anterior "-l" (o también con las opciones "-c" y "-C" ) con esto podemos establecer un valor personalizado entre ronda y ronda. Por defecto está fijado en 1000 milisegundos.

### parámetro "-n" y "-d":

Muestra los objetivos por nombre, esto es útil si nuestro dominio es compartido pues nos da una información adicional:

```
jimmy@KEVIN:~$ fping
-n ks7000.net.ve
host.caracashosting20.com is alive
jimmy@KEVIN:~$ fping
-n dronena.com
dronena.com: Name or service not known
jimmy@KEVIN:~$ fping -n android.com lga34s18-in-f4.1e100.net is alive
```

### parámetro "-q" y "-Q n":

Simplemente silencio, si lo usamos sin ningún otro parámetro pues simplemente no obtenemos nada... y así pues no tiene chiste. La documentación insiste en que muestra un resumen pero no experimentamos eso, la otra opción **"-Q"** debería mostrar dicho(s) resumen(es) cada *n* segundos especificados pero tampoco lo pudimos comprobar. Es entonces que para que sea útil podemos combinarlo al menos con la opción **"-c"** (que es para la cual está diseñada *exprofeso*):

```
jimmy@KEVIN:~$ fping -q 192.168.1.1  jimmy@KEVIN:~$ fping  
-q -c 5 192.168.1.1 192.168.1.1 : xmt/rcv/%loss = 5/5/0%, min/avg/max =  
0.25/0.27/0.28  jimmy@KEVIN:~$
```

## parámetro **"-l"**:

Cuando uno tiene un servidor "de verdad, verdad" con componentes redundantes tales como fuentes de poder, discos duros, **tarjetas de red**, etc pues entonces resulta útil esta opción. También podemos practicarla con nuestros ordenadores portátiles "modernos" (año 2009 para acá) que traen para conectar ethernet y wifi. la idea es mandar los ping de manera obligatoria por la tarjeta que nosotros ordenemos *y para nuestra sorpresa NO TIENE NADA QUE VER LA DIRECCIÓN IP QUE TENGA LA TARJETA EN CUESTIÓN, esto es debido a la forma y manera en que está programado GNU/Linux* (acá tienen [un pequeño pero denso artículo](#) al respecto, si queréis ahondar mucho en la materia).

## opción **"-t n"**:

Tiempo de espera inicial, por defecto 500 milisegundos.

## opción **"-u"**:

Mostrar los objetivos que son inalcanzables.

## opción **"-O n"**:

Establece el Tipo de Servicio ([TOS](#)) para los paquetes ICMP a fin de que los enrutadores le den un tratamiento específico. Básicamente es para indicar que el paquete es urgente, normal o de baja importancia (simplificado al máximo) *la cosa es que los dispositivos intermedios respeten dichas "sirenas" que ululan en el paquete.*

**-S addr** Dirección de origen del conjunto

## Respuestas para procesos por lotes **"batch"** o **"bash"**.

- Cero (0) si todos los objetivos tienen conexión.
- Uno (1) si uno o más objetivos fallan.
- Dos (2) si alguna de las direcciones IP no fueron encontradas.
- Tres (3) para línea de argumentos inválidos.
- Cuatro (4) para falla general del sistema.

## Ejemplos con fping:

*Las direcciones IP y nombres de dominio aquí utilizados son solamente para ejemplos didácticos, queda la responsabilidad de cada quien su utilización (o dicho de otra manera, cada quien es responsable por sus propios actos).*

En su forma más pura y simple:

```
fping 172.31.0.251
```

Una revisión completa a una red enviando un solo paquete a cada objetivo:

```
fping -a -r 0 -g 172.31.0.0/24
```

O cuatro paquetes a cada objetivo con la opción de reintentar:

```
fping -a -r 3 -g 172.31.0.0/24
```

Se puede hacer una exploración a una red de una manera mucho más rápida utilizando el parámetro "-i 1", esto indica que cada milisegundo envía un *ping* (necesita derechos de administrador):

```
sudo fping -a -i 1 -r 0 -g 172.31.0.0/24
```

La instrucción anterior pero leyendo de un archivo de texto con los nombres o direcciones de los objetivos (también podemos usar la opción "-f"):

```
sudo fping -a -i 1 -r 0
```

La opción "-s" muestra las estadísticas acumuladas hasta finalizar su ejecución:

```
fping -s -a -r 0 -g 172.31.0.0/24
```

## Otros usos para fping.

Hemos detectado que algunos otros proyectos han incrustado el código para una versión específica de **fping** adaptandolo para el lenguaje Python, [por ejemplo Riemman-fping](#).

## Fuentes consultadas.

### En idioma castellano.

- «[ping](#)» en Wikipedia.
- «[Detección de intrusos en la red con fping](#)» por Tannhausser.
- «[Realizar Barridos Ping utilizando fping](#)» por Alonso Eduardo Caballero Quezada.

### En idioma inglés.

- «fping.[org](#)»
- «fping(8) - [Linux man page](#)».
- «fping [0.0.1a2 for Python](#)».
- «Install and use Fping app on [RHEL/CentOS 6.x/7.x](#)» by Charls Vels.
- «How To Install Fping [on CentOS 7](#)»
- «fping: [A program to ping multiple hosts](#)».
- «Fping - [A Command-Line Tool to Ping Hosts In Parallel Under Ubuntu/Linux Mint](#)».
- «Code Snippet: [SO\\_BINDTODEVICE](#)».