

The Effects Server is a device designed to bridge two physical networks and introduce effects onto specific links between addresses on those two networks. These effects perturb network traffic in a configurable manner, including packet loss and reordering of packets.

The Effects Sever is designed to enable testing of applications by the easy reproduction of issues found in real-world environments, which may be difficult to otherwise recreate under lab conditions. In particular, the effects provided aim to provide realistic simulations of typical lossy and unreliable network connections.

Contents

- Hardware Specifications..... 2
- Connectivity and Safety.....3
- Installation..... 5
 - Mounting procedure..... 5
 - Physical wiring..... 6
 - Topology..... 7
 - Maintenance network..... 8
- Configuring the server..... 9
 - Command Line Interface..... 9
 - Command overview..... 10
 - Configuring a link..... 11
 - Adding effects to a link..... 13
 - Providing Metrics..... 13
 - Relaying Logs..... 15
- Command Reference..... 16
 - Modes..... 16
 - Control keys..... 17
 - Command Filters..... 17
 - Command Reference..... 18
- Effects Directory..... 20
 - Packet Loss.....20
 - Corruption..... 20
 - Duplication..... 20
 - Delay..... 21
 - Reordering..... 21
 - Drop all traffic..... 21
- Glossary.....22



Hardware Specifications

Physical specifications:

Item	Description
Operating temperature	0-60°C
Input voltage	90-264V AC
Internal PSU	Switched mode, 12V at 1.5A
Power	18W
Enclosure	1U 19" square-hole four-post rack mount
RoHS compliant	Yes
FCC compliant	No
CEC compliant	Internal PSU only

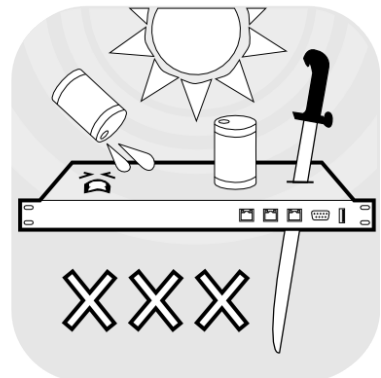
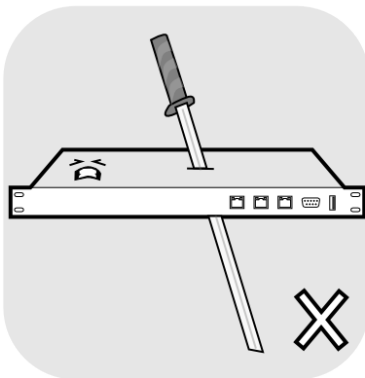
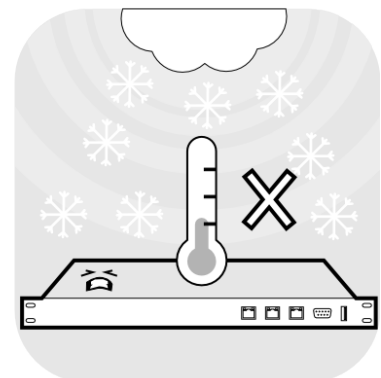
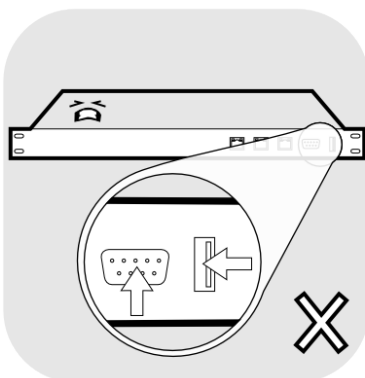
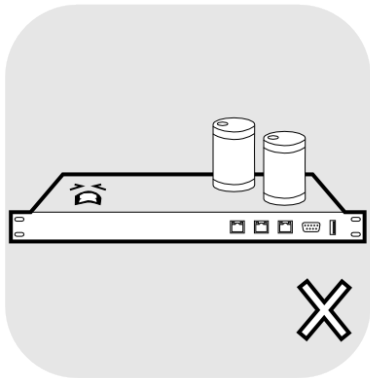
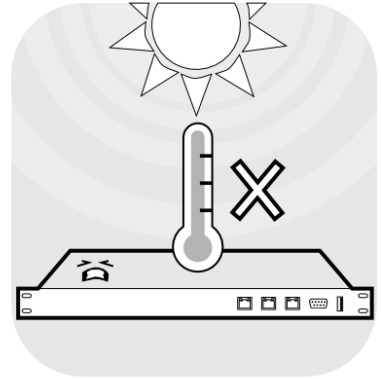
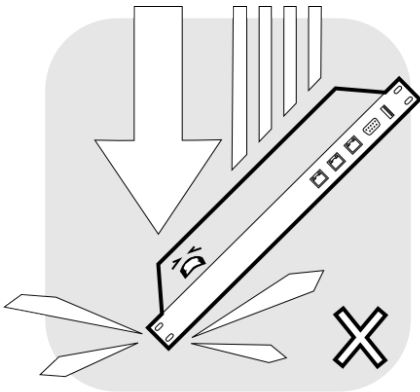
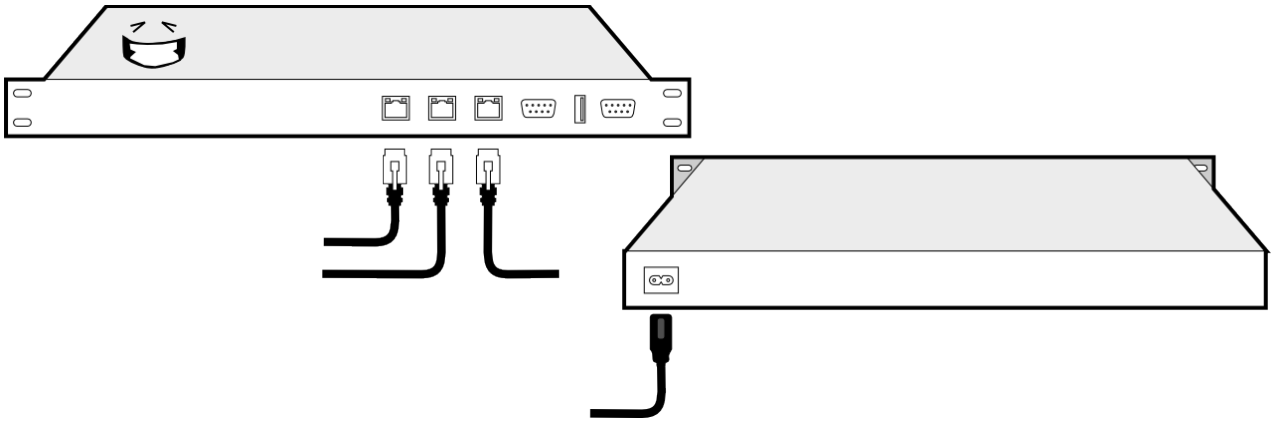
Electrical interfaces:

Item	Name	Interface	Purpose
2x RJ45	net0, net1	10/100 Ethernet	Networks to be bridged
1x RJ45	maint	10/100 Ethernet	Maintenance console
1x DB9	console	RS232	Reserved for future use as serial console
1x DB9	-	-	Unused
1x USB	-	-	Unused
1x Power	-	Non-polarised IEC320-C8 "shotgun" connector	Power

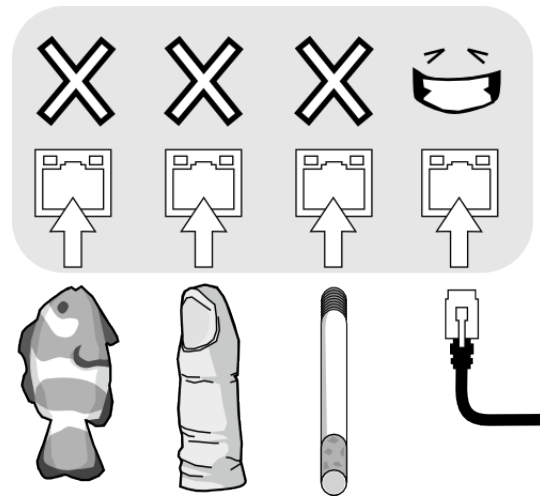
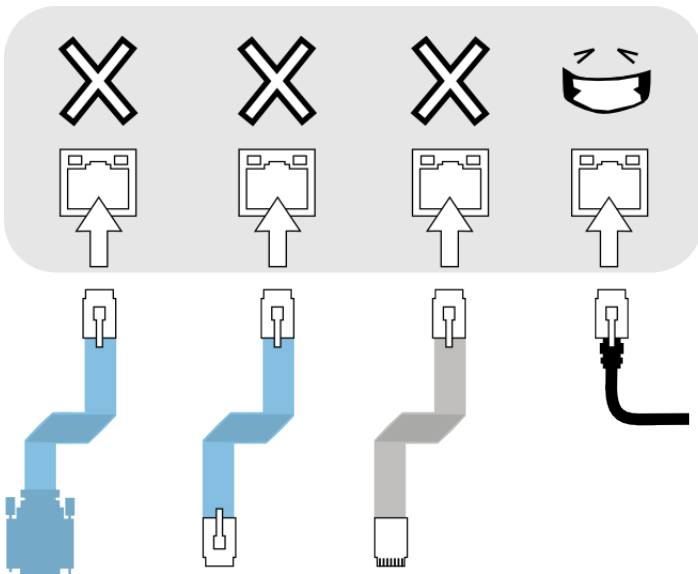
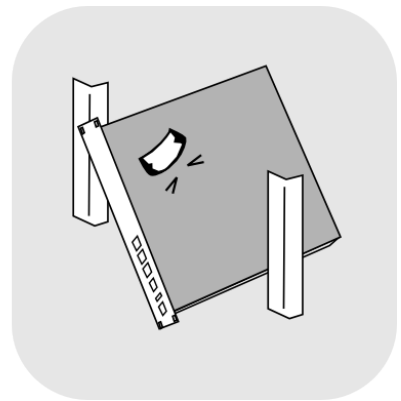
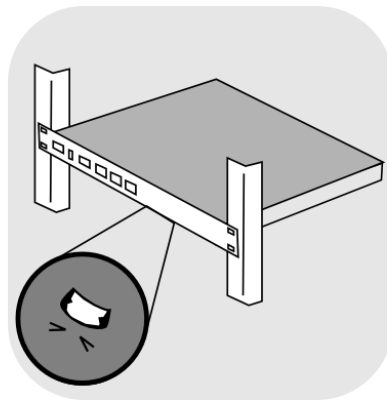
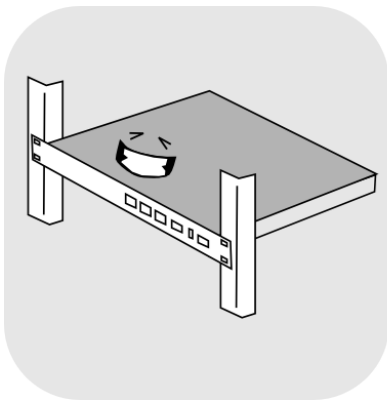
Protocols:

Protocol	Transport	Interface	Usage
Telnet	TCP/IPv4	maint	Command line interface
SNMP v1, v2c	UDP/IPv4	maint	Metrics
Syslog	UDP/IPv4	maint	Logging
IPv4	-	net0, net1	User traffic

Connectivity and Safety



Installation



Installation

The Effects Server is a half-depth 1U rack mounted machine.

The Effects Server is passively cooled. It is intended to be installed in a data centre or similar co-location environment with external cooling.

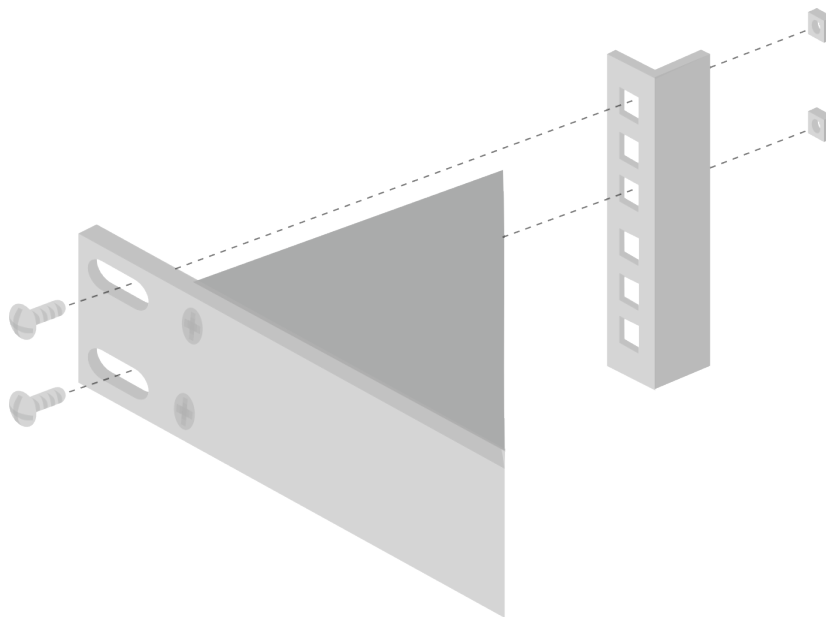


NOTE: It is possible, although not recommended, to mount another half-depth machine behind the Effects Server, in the same space, in racks which provide mounting from the rear.

The Effects Server has four mounting holes and should be bolted to a standard four-post rack using all four mounting holes. Extra care should be taken to ensure The Effects Server is mounted in the correct orientation as mounting the Effects Server upside down or at an angle may cause inverted packets and bad numbers.

See the section titled “mounting diagram” for the mounting diagram.

Mounting diagram



Mounting procedure

Installation requires two people.

1. Position the Effects Server within the rack, aligning the mounting holes to the frame as illustrated, such that the Effects Server occupies the intended rack unit of space. Do not mount the Effects Server at a non-integer multiple of rack units.
2. Place square-hole nuts behind the frame, and screw in the bolts. Do not over-tighten.
3. Connect the power.



NOTE: Bubblephone recommends the use of PAT tested power cables.

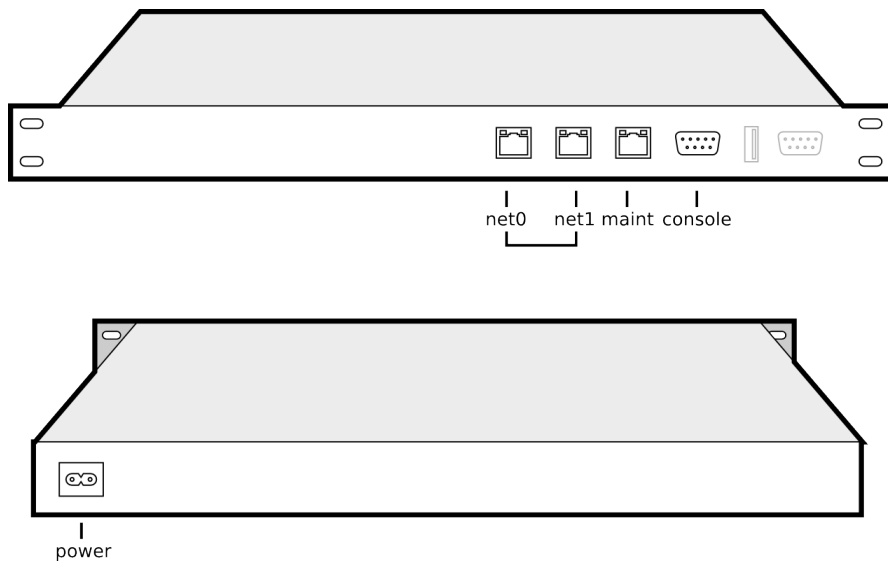
The Effects Server is now installed, and may be wired to networks and configured for use.

Physical wiring

Using standard Ethernet cables the RJ45 ports marked **net0** and **net1** should be connected to the two physical networks between which the effects server will bridge. Do not connect **net0** and **net1** to the same switch segment, or the Effects Server may explode.



NOTE: Bubblephone recommends the use of cat5e Ethernet cables. Ethernet cables are not provided with the Effects Server.



The **power** input must be provided with 90-264V AC. A power connector suitable for use with UK power outlets is provided with the Effects Server.

For configuration the **maint** RJ45 port should be used.

For general use the **console** DB9 port should not be used.

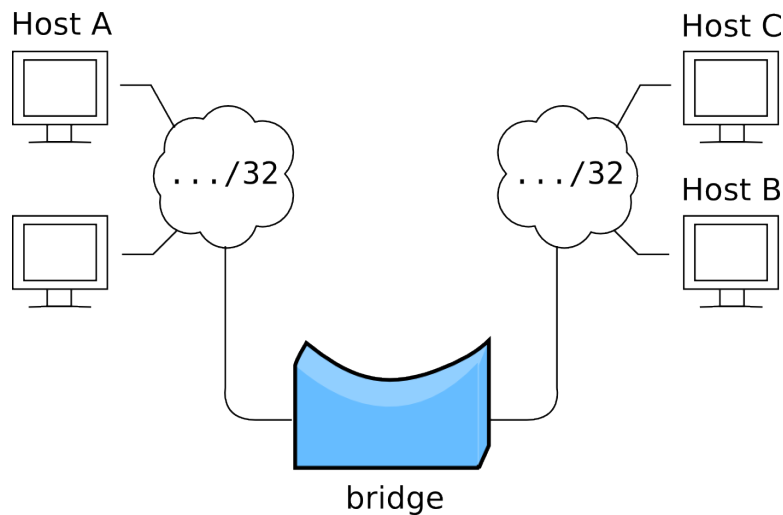
Any other available ports on the device should never be used.

All RJ45 ports are for Ethernet use, and must not be used with RJ45 serial cables.

Topology

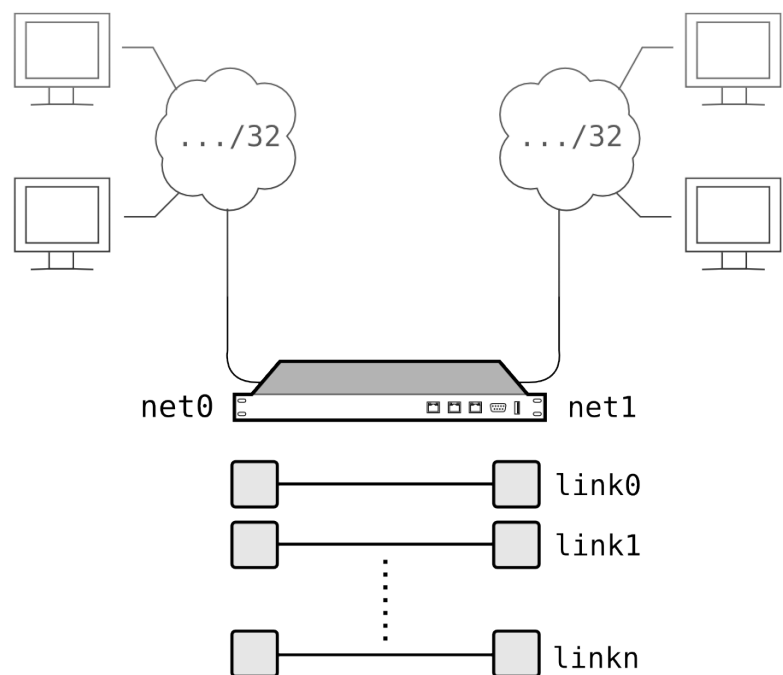
The Effects Server is designed to be a drop-in replacement for a bridge between two physical network segments which have no other connectivity via switching or routing.

The following diagram shows a network before the Effects Server has replaced a bridge. Host A is only able to communicate with host B via the bridge.



The above example illustrates two physically isolated networks, each with several hosts. The bridge maps traffic from hosts on one network to their counterparts. Note that this is not necessarily a one-to-one mapping (that is, host A need not communicate solely with host B).

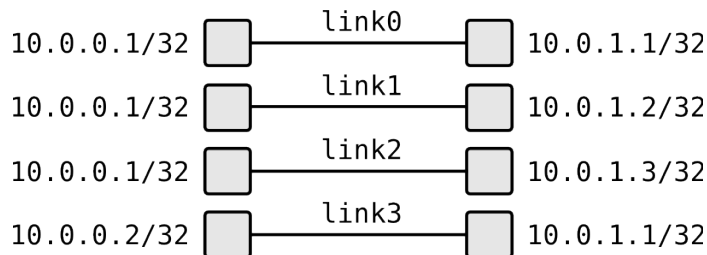
The Effects Server conceptually replaces this bridge. The two networks are connected via the **net0** and **net1** interfaces:



Bridging is provided by a set of user-defined *links*. A link is a mapping between a host on one network and its corresponding host on another. This is conceptually similar to a *route* in a router, except links operate at layer 2 in the OSI network layer model.

Once created, this link can be subjected to effects. These effects then act on traffic transferred between the hosts. Multiple effects may be present on one link.

Hosts may be linked in a many-to-many model by providing multiple links. A host may be present on more than one link at a time. For example, in the IP scheme illustrated above, one host from **net0** (10.0.0.1) may be linked to multiple hosts on **net1**. Conversely, any one of those multiple hosts may be linked to a single host in **net0**.



The address spaces on each network need not be unique; the same IP may be given to different hosts on both **net0** and **net1**. These are illustrated using unique subnets for sake of clarity only. In this example 10.0.0/24 and 10.0.1/24 respectively.

Before any link is configured the Effects Server will drop all traffic between these hosts. Once a link is in place the Effects Server will bridge between these two hosts. After a link is in place effects may be applied to the link.

Each network interface on the Effects Server has a configurable MTU which must be set appropriately to suit the networks to which they are connected. See the **set mtu** command for details. The default values should suit typical networks.

Maintenance network

The **maint** interface automatically requests an IP address via DHCP. Should the Effects Server CLI be required to be accessible a suitable DHCP server must be provided on the network segment to which this interface is connected.

The maintenance interface is used to provide telnet access to the command line interface. It also serves to provide metrics by SNMP, and to optionally relay log messages to a syslog server. Once configured the maintenance port is not required for normal use but may be left connected for convenience, or if syslog or SNMP are to be used.

The maintenance interface may share the same network segment as either **net0** or **net1**, or may be kept isolated to its own VLAN.

Configuring the server

In order to use the effects server some configuration is required. This configuration needs only be performed once before use.

To configure the effects server a connection to the Effects Server Command Line Interface (EFCLI) is required. To connect to the EFCLI an Ethernet cable must be connected to the port marked **maint**. This cable must be connected to a network capable of providing an IP Address by DHCP to the interface.

Once an IP Address is known for this interface then a connection to the EFCLI can be made via telnet on the standard telnet port, port **23**.

```
% telnet 192.168.0.220
Bubblephone Effects server

ef0>
```

You are greeted with a welcoming banner, and have now connected to the EFCLI and can now examine the configuration at the prompt.



NOTE: The command prompt shows the current hostname, which may be altered for convenience by the **set hostname** command. Therefore, your prompt may differ from the example output.

To view the software version of the EFCLI use the **show version** command.

```
ef0> show version
efcli 1.0
Copyright 2011, Bubblephone Ltd.
```

Check the configuration of the device by using the **show interfaces** and **show links** commands before performing the basic setup to see if the default configuration meets your requirements.

Command Line Interface

The Effects Server provides a Command Line Interface (EFCLI), accessible via a telnet session through the **maint** network interface. This is the primary method for configuring the Effects Server.

The EFCLI allows examination of the current system setup and modification of parameters, such as creating and removing links and modifying the effects on those links.

Access to privileged commands is restricted by password authentication. Authentication is achieved by using the **enable** command and entering the correct password.



NOTE: Default passwords will be provided separately from this document.

Command overview

Commands within the EFCLI are hierarchically structured. Once connected to the EFCLI you will only have limited access to commands. To gain access to commands related to a particular feature (such as interfaces, links and effects) you must use the **configure** command to enter “configure” mode. While in configure mode you will have access to additional commands. For example, let us look at configuring an interface:

```
ef0> show interfaces
net0:
  mtu: 1500

net1:
  mtu: 1500
```

Commands may be abbreviated to their shortest unambiguous prefix. For example, instead of typing **show interfaces**, you could simply type **sh int**, which would do the same thing. For clarity, the full forms of commands are given in these examples.

To configure one of these interfaces we must first authenticate (the password is not shown).

```
ef0> enable
Password: *****
ef0#
```



NOTE: Passwords are sent over telnet in plain text and care should be taken over the choice of network used to configure the server.

Once enabled we are able to enter configure mode. Configuration data is entered from the terminal.

```
ef0# configure terminal

ef0(config)# help
  effect          Configure an effect
  interface       Configure an interface
  link            Configure a link

ef0(config)#
```

In configure mode we can see that from the **help** command that we have access to additional commands such as **effect**, **interface** and **link**. These commands enable us to modify effects, interfaces and links respectively.

Let us continue looking at the example of configuring an interface.

```
ef0(config)# interface net0

ef0(config-net0)# help
  add          Add an address
  delete      Remove an address
  set
```

We are now configuring the interface **net0**. Running the **help** command here shows that we have new commands available. These commands are specific to configuring an interface.

We can find out more about the **set** command by typing **set** and then typing a ? (question mark). You may use the **help** command or type a ? (question mark) at any point to get a list of valid commands for the current context.

```
ef0(config-net0)# set ?
  description  Assign a description
  mtu          Assign an MTU
```

To go back a level in the hierarchy you can use the **exit** command or press **^Z** (control-Z).

```
ef0(config-net0)# exit

ef0(config)# exit

ef0#
```

Configuring a link

In order for the Effects Server to let data flow over the bridge a link needs to be created between two endpoints. Without a link all packets are dropped.

First connect to the EFCLI using telnet and authenticate with the **enable** command. Then enter configure mode using **configure terminal**.



NOTE: Unique contractions of commands perform the same function as commands in full. For example **he** will show the same as **help**. In this case, we could use **con t** in place of **configure terminal**.

```
Bubblephone Effects Server
```

```
ef0> enable
password:
ef0# configure terminal

ef0(config)#
```

Before we can create a link we must add addresses for the endpoints to their respective interfaces. The interfaces provided are **net0** and **net1**. To add an address to an interface we must enter the configuration mode for that interface.

```
ef0(config)# interface net0  
ef0(config-net0)#
```

From here we can use the **add** command to add an address to the interface.

```
ef0(config-net0)# add 192.168.0.210  
ef0(config-net0)#
```

Once addresses have been added to both interfaces then we can create a link between them. Exit from configuration mode.

```
ef0(config-net0)# exit  
ef0(config)# exit  
ef0#
```

Once out of configuration mode use the **add** command to create a link. The **add** command takes five parameters; the link name (of the form link0, link1, link2 etc), two addresses and their respective interfaces.

```
ef0# add link0 net0 192.168.0.210 net1 192.168.0.220  
  
ef0# show links  
link0  
  endpoint: net0    192.168.0.210  
  endpoint: net1    192.168.0.220  
  link effects:  
  
ef0#
```

Here we have created a link between our endpoints and have used the **show** command to list the details of the link we have created. You will notice that there are no effects currently active on this link.

Now we have created a link, bridging will occur between these two endpoints.



NOTE: Once a link has been created it will persist even across system reloads until it is removed with the **delete** command.

Adding effects to a link

With a link created it is now possible to add any of the effects to that link. With no effects present, the link will carry traffic through without modification.

To see a list of the available effects with short descriptions you can use the **show effects** command.

```
ef0> show effects
tne6 -- Drop all traffic (after a certain time)
tne5 -- Drop all traffic
tne4 -- Packet reordering
tne3 -- Packet corruption
tne2 -- Packet duplication
tne1 -- Packet loss
tne0 -- Packet delay
```

For further information on effects please see the Effects Directory later in this document.

To add an effect to a link you will need to enter configuration mode for that link. Use the **add** command to add effects to that link. More than one effect may be added to a link.

```
ef0# configure terminal
ef0(config)# link link0
ef0(config-link0)# add tne0
ef0(config-link0)#
```

The effect **tne0** is now active on the link **link0**.

Use the **delete** command to remove an effect in a similar manner.

Providing Metrics

The Effects Server provides metrics via the Simple Network Management Protocol (SNMP). SNMP is configurable using the command line interface.

SNMP authentication is performed by providing a “community string”. This string can be set or unset using the Effects Server command line interface when in enable mode.

The SNMP community string is set with **set community** command. For example, to set the community string to **hello**:

```
ef0# set community hello
```

SNMP clients may now connect to the Effects Server on its management interface.

To disable the SNMP server, the community string is unset with **unset community** command:

```
ef0# unset community
```

The Effects Sever supports SNMP protocols 1 and 2c, and may be used with any SNMP client which supports those versions. The Effects Server provides the following MIBs:

```
IF-MIB::interfaces
SNMPv2-MIB::system
DISMAN-EVENT-MIB::sysUpTimeInstance
```

The Effects Server does not provide SNMP traps.

For example, gathering metrics using the **snmpbulkwalk** command provided by the freely-available Net-SNMP project:

```
% snmpbulkwalk -c hello -v2c 172.16.254.2
SNMPv2-MIB::sysDescr.0 = STRING: "Bubblephone Effects Server"
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (50808) 0:08:28.08
SNMPv2-MIB::sysName.0 = STRING: ef0
SNMPv2-MIB::sysServices.0 = INTEGER: 15
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (13) 0:00:00.13
SNMPv2-MIB::sysORID.1 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
...
IF-MIB::ifNumber.0 = INTEGER: 2
IF-MIB::ifDescr.3 = STRING: net1
IF-MIB::ifDescr.4 = STRING: net0
...
IF-MIB::ifInOctets.3 = Counter32: 4182155
IF-MIB::ifInOctets.4 = Counter32: 3926475
IF-MIB::ifOutOctets.3 = Counter32: 4255446
IF-MIB::ifOutOctets.4 = Counter32: 4536951
```

Here we can see the interface **net1** has transferred approximately 4MB in both directions. The presence of effects on links is evident, as the corresponding interface **net0** has transferred a different amount of traffic. Uninteresting information has been elided for brevity.

Rather than run a tool by hand, it is more typical to use a common automated system which gathers data periodically, and updates a "round robin" database. This can be used to provide convenient graphs showing activity over a period of time. There are many open-source tools available which help provide gathering and visualisation of data from SNMP; MTRG and RRDtool are popular choices.

Relaying Logs

The Effects Server logs both locally (visible by **show log**) and optionally relayed to an external machine via syslog. Syslog is configured using the command line interface in enable mode.

By default syslog relaying is disabled. To enable relaying, the address of the syslog server must be specified with the **set syslog** command.

The **set syslog** command takes a single parameter, the IP address of the syslog server, with an optional port. If not specified, the port will default to port 514, the standard syslog port. Messages are relayed using the standard syslog protocol over UDP.

For example:

```
ef0# set syslog 172.16.254.1
ef0# set syslog 172.16.254.1:1234
```

The relay may be unset (disabling syslog relaying) by using the **unset syslog** command:

```
ef0# unset syslog
```

The syslog protocol provides several means of categorising messages, including by priority, facility and by tag. The messages logged by the Effects Server have the syslog tag **efcli**. This tag can be used to identify log messages produced by the effects server. Some syslog servers provide filtering and redirection of messages based on the message tag; this can be useful for automating notifications when important events occur.

The Effects Server logs messages from the command line interface and Effects Server internal systems. These typically include notifications of events, such as changing the password, or reloading the Effects Server.

Some example log messages, relayed to a UNIX syslog server:

```
# tail -3 /var/log/messages
Oct 26 09:59:53 172.16.254.2 ef0 efcli: enable password set
Oct 25 16:53:02 172.16.254.2 ef0 efcli: hostname set
Oct 25 15:21:17 172.16.254.2 ef0 efcli: reloading
```

Different syslog servers present messages in different ways.

Command Reference

This section provides a reference for features of the CLI, and an index of all commands. Note that not all commands are simultaneously accessible when using the CLI, depending on the current mode and authentication state.

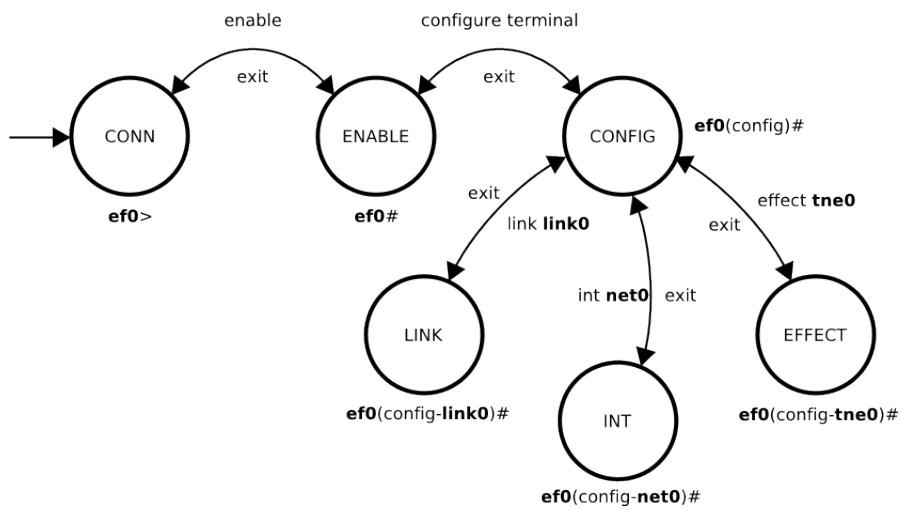
Modes

The effects server CLI operates in one of the following modes:

- CONN** Connected and unauthenticated
- ENABLE** Successfully authenticated, with access to modify the server
- CONFIG** Configuration
- LINK** Configuring a specific link
- INT** Configuring a specific interface
- EFFECT** Configuring a specific effect

The current mode is indicated by the prompt.

Navigation between these modes is illustrated below:



Control keys

Key	Ctrl	Command
↑ Up arrow	^P	History backwards
↓ Down arrow	^N	History forwards
← Left arrow	^B	Cursor left
→ Right arrow	^F	Cursor right
	^A	Cursor to SOL
⌫ Backspace	^H	Delete character
	^W	Delete word
	^U	Clear line
	^K	Clear to EOL
	^L	Redraw
	^D	EOT
	^Z	Exit to enable mode
→ Tab	^I	Tab completion
↵ Enter	^M	Execute command

Command Filters

The output from commands may be filtered through a pipeline. Multiple filters may be piped together, to filter the output from a previous filter. The available filters are:

Filter	Description
... begin pattern	Lines after the first matching line
... between pattern pattern	Lines between the first and last matching lines
... count	Count of lines
... exclude pattern	Exclude lines that match
... include pattern	Include lines that match
... grep { -iev } regex ... egrep { -iev } regex	Include lines that match POSIX BRE/ERE -i Invert case -e ERE dialect (same as egrep) -v Invert regex

Where **pattern** is a literal string, and **regex** is a regular expression.

Command Reference

Prompt	Command	Description
ef0>	help	Show available commands
ef0>	quit	Disconnect
ef0>	logout	Disconnect
ef0>	exit	Exit from current mode
ef0>	history	Show previously run commands
ef0>	enable	Turn on privileged commands
ef0>	show version	Show server version
ef0>	show syslog	Show syslog relay
ef0>	show effects	List available effects
ef0>	show interfaces	List available interfaces
ef0>	show links	List available links
ef0#	show log	List internal messages
ef0#	add link0 net0 1.2.3.4 net0 1.2.3.4	Create a link
ef0#	delete link link0	Remove a link
ef0#	reload	Warm reboot
ef0#	set hostname "ef0"	Assign a new hostname
ef0#	set password "password"	Assign a new password
ef0#	set syslog 1.2.3.4[:1234]	Assign a syslog relay
ef0#	set community "hello"	Assign a community string
ef0#	configure terminal	Enter configuration mode
ef0#	unset syslog	Disable syslog relay
ef0#	unset community	Disable SNMP server

Prompt	Command	Description
ef0 (config)#	effect tne0	Configure an effect
ef0 (config- tne0)#	set description "hello"	Assign a description
ef0 (config)#	interface net0	Configure an interface
ef0 (config- net0)#	add 1.2.3.4	Add an address
ef0 (config- net0)#	delete 1.2.3.4	Remove an address
ef0 (config- net0)#	set description "hello"	Assign a description
ef0 (config- net0)#	set mtu 1234	Assign an MTU
ef0 (config)#	link link0	Configure a link
ef0 (config- link0)#	add tne0	Add an effect
ef0 (config- link0)#	delete tne0	Remove an effect
ef0 (config- link0)#	set description "hello"	Assign a description

Effects Directory

This section provides a reference for all effects included with the Effects Server.

Packet Loss.....	20
Corruption.....	20
Duplication.....	20
Delay.....	21
Reordering.....	21
Drop all traffic.....	21

Packet Loss

The packet loss effect causes a percentage of the packets on a link to be dropped.

This effect will cause approximately 10% of packets to be lost in total. Each successive packet's probability depends by a 10% correlation on the previous one. This correlation makes the effect less random, and emulates burst packet loss.

$$r = [-1,1] \in \mathbb{R}$$
$$loss = 0.1$$
$$correlation = 0.1$$

$$P_n = \begin{cases} loss \cdot r & \text{if } n = 1 \\ correlation \cdot P_{n-1} + (1 - correlation) \cdot r & \text{if } n > 1 \end{cases}$$

This is intended for simulating generic packet loss as encountered on networks under typical operating conditions. This is not suitable for simulating networks saturated to their throughput capacity, as the loss model significantly differs.

Corruption

The corruption effect introduces a single bit error at a random offset in 15% of packets.

Corruption occurs at layer 2, so data link layer constructs such as the MAC address may be affected.

This is intended for emulating low quality connections such as a digital protocol over an analogue bearer. For example, IP over radio or modem.

Duplication

This effect duplicates 15% of packets.

Duplication occurs at layer 2.

This is intended for emulating the duplication found on typical routed networks.

Delay

The delay effect delays packets by a large value with some small variation. This effect needs to be active on a link for the reordering effect to be enabled on that link.

This effect will delay packets by 200ms with a random variance of 15ms and a correlation of 25% to emulate burst packet delay. Each successive packet's delay depends by a 25% correlation on the previous delay.

$$\begin{aligned}r &= [-1,1] \in \mathbb{R} \\ \text{delay} &= 200\text{ms} \\ \text{variance} &= 0.25 \\ \text{correlation} &= 0.25\end{aligned}$$

$$D_n = \begin{cases} \text{delay} - \text{variance} \cdot r & \text{if } n = 1 \\ \text{delay} - \text{correlation} \cdot D_{n-1} + (1 - \text{correlation}) \cdot r & \text{if } n > 1 \end{cases}$$

This is intended for simulating laggy connections with a reasonably consistent delay. For example, to simulate an SDSL connection operating under normal conditions.

Reordering

The reordering effect necessarily delays a percentage of packets to cause reordering. The delay effect must be active on a link for reordering to be enabled on the link. If the delay is not active an error message will be produced after an attempt to add reordering to a link.

Since delay is required for reordering, the reordering effect must be removed from a link before the delay can be removed.

This effect will cause approximately 75% of packets to be delayed in total, by 200ms each. Each successive packet's probability depends by a 10% correlation on the previous one.

$$\begin{aligned}r &= [-1,1] \in \mathbb{R} \\ \text{reorder} &= 0.25 \\ \text{correlation} &= 0.1\end{aligned}$$

$$P_n = \begin{cases} \text{reorder} \cdot r & \text{if } n = 1 \\ \text{correlation} \cdot P_{n-1} + (1 - \text{correlation}) \cdot r & \text{if } n > 1 \end{cases}$$

The overall effect of some packets being delayed is the reordering of these packets.

Reordering is intended for simulating packets taking different routes to their destination and arriving out of order.

Note that use of a protocol that provides ordering, such as TCP streams, will perceive this effect as a delay only, as the application layer is not responsible for reordering.

Drop all traffic

This effect will cause the link to drop all traffic.

This is intended for emulating a dead connection. For example, by unplugging a modem.

Glossary

BRE	Basic Regular Expression
CLI	Command Line Interface
CEC	Consumer Electronics Control
DHCP	Dynamic Host Configuration Protocol
Effect	A perversion of network traffic from its canonical form
EOL	End of Line
EOT	End of Transmission
ERE	Extended Regular Expression
FCC	Federal Communications Commission
IP	Internet Protocol
Link	A bridge between two subnets on different segments
MTU	Maximum Transmission Unit
OSI	Open Systems Interconnection
PEM	Packet Eating Monster
PSU	Power Supply Unit
RoHS	Restriction of Hazardous Substances
SDSL	Symmetric Digital Subscriber Line
SOL	Start of Line
TCP	Transmission Control Protocol
TNE	Test Network Effect
USB	Universal Serial Bus
VLAN	Virtual Local Area Network