

**NAME**

dhcpemu – DHCP Single Client Packet Emulator

**SYNOPSIS**

```

dhcpemu [-w]
  [ --bcastbit ]
  [ --bpop=number ]
  [ --cookie=cookie-type ]
  [ --debug=debug-level ]
  [ --dstport=portnum ]
  [ --echoyi ]
  [ --file=filename ]
  [ --junk=junksize ]
  [ --hlen=number ]
  [ --htype=number ]
  [ --if=interface-name ]
  [ --persist ]
  [ --promiscuous ]
  [ --quiet ]
  [ --server=server-ip-address ]
  [ --size=size ]
  [ --srcip=source-ip-address ]
  [ --srcport=portnum ]
  [ --timeout=timeout-secs ]
  [ --use-cache ]
  [symbol=value] [symbol=value]...

```

**DESCRIPTION**

**dhcpemu** is a DHCP/BOOTP emulator, or more precisely a DHCP/BOOTP packet generator. It is important to note that with the options available it is quite possible to create an illegal packet; indeed this is one of the primary functions of **dhcpemu**: to test the behavior of servers when confronted with packets that do not conform to the standards.

The timeline for **dhcpemu** is as follows. A packet is constructed, is sent through the interface specified, and a reply awaited. The emulator will exit after the first reply is received or after a length of time specified by *timeout*. Any configuration received will be saved for later re-use ( **-w** option) in a file named *interface.dhc* in the current working directory. Depending on the parameters specified, and/or the DHCP server configuration, no reply may in fact be forthcoming. If no timeout is specified the emulator may be killed with any suitable asynchronous signal. The SIGINT signal (usually generated from the keyboard with CONTROL-C ) is available if **dhcpemu** is running in the foreground.

The format of the options (*symbol=value*) is identical to that in **dhcpcap** except in the one respect that they are separated one from another by whitespace instead of colons. The options are first scanned by the shell, so judicious use of quotes may be needed to protect special characters. There are two options with special formats whose semantics dictate that they are never encountered in **dhcpcap**, but which are valid for the emulator:

Option#55: request list **rv**

A list of options that the emulator wishes the server to return. Its format is **rv=opt,opt...** where *opt* is either an option number, or the symbolic code for that option (given in **dhcpcap** tags).

Option#53: message type **mt**

The type of DHCP/BOOTP packet that will be sent. It has the format **mt=keyword** where *keyword* is one of the following: **discover**, **request**, **release**, **decline**, **inform**, **leasequery**, **offer**, **ack**, **nak**, **known**, **unknown**. If no **mt** option is given, the packet will not have any message type, and will appear to be BOOTP not DHCP.

The packet sent is constructed as follows: a buffer of *size* octets (default 548) is zeroed out. Each DHCP option specified on the command line (*symbol=value*) is then added to the buffer. Options that correspond to fixed fields are placed at the appropriate offset: if any such are repeated, only the value in the last of them will be represented in the result. On the contrary, non fixed field options are placed in the variable part of the packet in the order in which they are entered on the command line, until the command line is consumed, or space in the buffer is exhausted. If the same option occurs occur more than once, it will be represented multiple times in the result.

After these placements are made, space allowing, *junksize* (default zero) octets of random data are appended.

Finally any fixed fields which have not been assigned values on the command line are set according to the following table:

Field Name	Offset	Set By	Default Value
<i>op</i>	0	<b>--bpop=number</b>	bootrequest (=1)
<i>htype</i>	1	<b>--htype=type</b>	ethernet (=1)
<i>hlen</i>	2	<b>--hlen=number</b>	6
<i>hops</i>	3		0
<i>xid</i>	4	<b>xi=number</b>	random number
<i>secs</i>	8	<b>sc=number</b>	0
<i>flags</i>	10	<b>-b</b> <b>--bcastbit</b>	0
<i>ciaddr</i>	12	<b>ci=ip-address</b>	0.0.0.0
<i>yiaddr</i>	16	<b>yi=ip-address</b>	0.0.0.0
<i>siaddr</i>	20	<b>si=ip-address</b>	0.0.0.0
<i>giaddr</i>	24	<b>gi=ip-address</b>	0.0.0.0
<i>chaddr</i>	28	<b>hw=hardware-address</b>	Hardware address of interface through which packet is sent
<i>sname</i>	44		all zeros
<i>file</i>	108	<b>-f string</b> <b>--file=string</b>	all zeros
<i>cookie</i>	236	<b>-m cookie-type</b> <b>--cookie=cookie-type</b>	0x63825363

**OPTIONS**

Option	Synonym
<b>--bcastbit</b>	<b>-b</b>
<b>--cookie</b>	<b>-m</b>
<b>--debug</b>	<b>-d</b>
<b>--file</b>	<b>-f</b>
<b>--junk</b>	<b>-j</b>
<b>--persist</b>	<b>-g</b>
<b>--promiscuous</b>	<b>-p</b>
<b>--quiet</b>	<b>-q</b>
<b>--server</b>	<b>-a</b>
<b>--size</b>	<b>-s</b>
<b>--srcip</b>	<b>-S</b>
<b>--timeout</b>	<b>-t</b>
<b>--use-cache</b>	<b>-u</b>

- bcastbit** Set the broadcast bit.
- bpop=*number*** Set the value of the *bpop* field to *number*. This should be =1 for packets directed to servers from clients, and =2 for the converse.
- cookie=*cookie-type*** Sets the magic cookie in the outgoing packet according to *cookie-type* selected from this list: **rfc1048**, **cmu**, **none**, **junk**. **rfc1048** (the default) is required by DHCP. The values **none** and **junk** are self explanatory. **cmu** is an historical relic of some BOOTP implementations.
- debug=*number*** Set the level for debug output to *number*.
- dstport=*portnum*** The destination port number (default=67, the DHCP server port number).
- echoyi** Print to *stdout* IP address offered and the DHCP server's address on successful completion. This is intended to aid scripting.
- file=*filename*** Set the *file* field to the string *filename*
- hlen=*number*** Set the value of the *hlen* field in the DHCP packet.
- hlype=*number*** Set the value of the *hlype* field in the DHCP packet.
- if=*interface*** Send the packet through the selected network interface. If no interface is explicitly given, **dhcpemu** will select the first one allowing broadcast. If no hardware address is explicitly defined by the options, the *chaddr* field in the packet to be the hardware address of the interface.
- persist** Once a packet has been received, continue to listen for other responses (until *time-out* or killed by an asynchronous signal)
- promiscuous** Listen on all interfaces, not just the interface through which the outgoing packet was sent. Without this option broadcast replies from the server will not be received.
- quiet** Run quietly: display neither the contents of the incoming nor the outgoing packets.
- server=*server-ip-address***  
The packet is unicast to *server-address*. Without this option the packet will be broadcast.
- size=*size*** A packet of *size* octets is sent (by default 548 octets)
- srcip** Set the IP address in the IP header to be the address of the interface *interface* selected. Otherwise, this address is set to be 0.0.0.0 (from *init*, *init-reboot* and *selecting* states) or to the IP address which was previously configured (*renewing*

- and rebinding states).
- srcport=portnum** Set the source port number. The default is to allow the operating system to assign an unused port number.
  - timeout=timeout** Exit after *timeout* seconds if no responses are received. The default is to wait forever.
  - use-cache** Set the values of the *server address*, *requested address* and *ciaddr* to be those received in a previous DHCP OFFER. This allows several invocations of **dhcpemu** to be put into a script, and to conduct a complete DHCP transaction. This would not otherwise be possible, because the values of those fields would only be known at run-time.

## NOTES

In normal use, to verify that DHCP is working correctly, **dhcpemu** would like to receive a reply from the server even though it may be emulating a client with an arbitrary MAC address. The way to accomplish this is to set the *giaddr* in the packet to the IP address of the box on which the emulator is running (**gi=local-ip-address**) and to unicast the packet to the IP address of a known server (**--server=dhcp-server-ip-address**). Depending on how strict the server is, the source port number in the packet may also have to be set to 67 (**--srcport=67**).

## BUGS

Vendor options are not supported so a command line option like *vendor-option-symbol=value* will be ignored.

The **cmu** magic cookie implies a different encoding of the non-fixed-fields, but only **rfc1048** encoding is supported.

## SEE ALSO

**dhcptags(5)**, **RFC2031**, **RFC2032**.