

The `netstat` command

Hans Kruse and Carl Bruggeman

Jan 7, 2007

The `netstat` command

`netstat` (which is available on Unix as well as Windows) serves two primary purposes:

1. the `netstat` or `netstat -n` commands display the currently active IP connections.
2. the `netstat -r` or `netstat -rn` commands display the routing table.

What does the `netstat` output mean?

Below is a sample of what the `netstat -n` command might produce; the command generated almost 100 lines of output; only a few of them are shown below:

```
Active Internet connections
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp4   0      0 132.235.232.204.51197  132.235.8.44.993      ESTABLISHED
tcp4   0      0 132.235.232.204.51161  17.250.248.152.993   ESTABLISHED
...
tcp4   51     0 132.235.232.204.58977  130.239.18.159.21    CLOSE_WAIT
tcp4   0      0 127.0.0.1.1033        127.0.0.1.1015       ESTABLISHED
tcp4   0      0 127.0.0.1.1015        127.0.0.1.1033       ESTABLISHED
...
udp4   0      0 *.*                    *.*                    *
udp4   0      0 132.235.232.204.138   *.*                    *
udp4   0      0 132.235.232.204.137   *.*                    *
udp4   0      0 *.138                  *.*                    *
...
icm6   0      0 *.*                    *.*                    *
Active LOCAL (UNIX) domain sockets
Address Type  Recv-Q Send-Q Inode   Conn   Refs  Nextref Addr
399ecc0 stream  0      0      0       0      0     0      0
399eee0 stream  0      0      0       0      0     0      0
```

Proto tells you which type of transport protocol created the connection. For the most part, `tcp4` (TCP over IPv4) and `tcp6` (TCP over IPv6) are of interest to us. `netstat` also shows UDP and ICMP activity, but since these protocols don't use connections, information is limited.

Recv/Send-Q shows the amount of data waiting in the receive or send queues.

Local Address looks a bit odd; the first 4 numbers are the IP address, the fifth number is the TCP or UDP "port", which identifies the program that is using the connection. If you use `netstat` instead of `netstat -n`, `netstat` will attempt to display names for addresses and port numbers.

On most systems the Local Address is either the IP address of the network interface or the 127.0.0.1 local loop-back address. On machines with more than one interface the entry will identify the local address used in the connection.

Foreign Address has the same format as the Local Address but shows the remote side of the connection.

(state) has meaning only for TCP. An active TCP connection will be in the ESTABLISHED state, everything else shows a connection that is still being set up, or in the process of being disconnected. If you use `netstat -a` or `netstat -na` you will also see connections in the LISTEN state, these are server programs on your local machine that will accept incoming connection requests.

The list of IP-based connections may be followed by a listing of Unix sockets (not on Windows). These represent local connections between programs and are not usually of interest in a networking context. Lets now look at the `netstat -rn` output:

Routing tables

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	132.235.233.254	UGSc	51	3053	en0	
127.0.0.1	127.0.0.1	UH	11	649071	lo0	
132.235.232/23	link#4	UCS	18	0	en0	
132.235.232.138	0:13:21:61:b2:27	UHLW	0	508	en0	956
132.235.232.147	0:11:43:1a:dd:1	UHLW	0	0	en0	744
...						

Internet6:

Destination	Gateway	Flags	Netif	Expire
::1	link#1	UHL	lo0	
fe80::%lo0/64	fe80::1%lo0	Uc	lo0	
fe80::1%lo0	link#1	UHL	lo0	
fe80::%en0/64	link#4	UC	en0	
fe80::20d:93ff:feaf:b5ec%en0	0:d:93:af:b5:ec	UHL	lo0	
ff01::/32	::1	U	lo0	
ff02::/32	::1	UC	lo0	
ff02::/32	link#4	UC	en0	

Note that there are two blocks of information. For now we are only interested in the section labeled **Internet**. The IPv6 part (labeled **Internet6**) is something we come back to later in the quarter.

Destination is either the IP address of a single host, or a combination of a network address and a network mask. In our example the mask is shown in the “slash” convention, other `netstat` version may use the dotted decimal format. The entry 132.235.232/23 is the same as `address 132.235.232.0 netmask 255.255.254.0`. The special entry `default` stands for “everything else” and might also be shown as 0.0.0.0.

Gateway identifies where packets to the given destination should be sent. This can be the IP address of a router, or the designation of a network interface (the naming will differ from system to system). If a network interface is named (or the address shown is the address of the local host itself) the routing entry is for addresses on the local network. If the gateway entry contains a list of hexadecimal numbers, it represents the hardware address for a host. Not all systems will show such entries.

Flags define a number of details about the routing entry. Check the man page if you are curious.

Netif is the designation (usually the same as the one used in `ifconfig`) of the network interface that will be used by this routing entry.

Expire shows how much longer a dynamic entry is valid. We will see examples of this in later labs.

While the `netstat` command itself is pretty similar on all the systems, the labeling and formatting of the output will be different. You should, however, be able to easily identify all the components we discussed above.

The netstat man page from Mac OS X

NETSTAT(1) BSD General Commands Manual NETSTAT(1)

NAME

netstat -- show network status

SYNOPSIS

```
netstat [-AaLlnW] [-f address_family | -p protocol] [-M core] [-N system]
netstat [-gilns] [-f address_family] [-M core] [-N system]
netstat -i | -I interface [-w wait] [-abdgt] [-M core] [-N system]
netstat -s [-s] [-f address_family | -p protocol] [-M core] [-N system]
netstat -i | -I interface -s [-f address_family | -p protocol] [-M core]
[-N system]
netstat -m [-M core] [-N system]
netstat -r [-Aaln] [-f address_family] [-M core] [-N system]
netstat -rs [-s] [-M core] [-N system]
```

DESCRIPTION

The netstat command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The first form of the command displays a list of active sockets for each protocol. The second form presents the contents of one of the other network data structures according to the option selected. Using the third form, with a wait interval specified, netstat will continuously display the information regarding packet traffic on the configured network interfaces. The fourth form displays statistics for the specified protocol or address family. The fifth form displays per-interface statistics for the specified protocol or address family. The sixth form displays mbuf(9) statistics. The seventh form displays routing table for the specified address family. The eighth form displays routing statistics.

The options have the following meaning:

-A With the default display, show the address of any protocol control blocks associated with sockets; used for debugging.

-a With the default display, show the state of all sockets; normally sockets used by server processes are not shown. With the routing table display (option -r, as described below), show protocol-cloned routes (routes generated by a RTF_PRCLONING parent route); normally these routes are not shown.

-b With the interface display (option -i, as described below), show the number of bytes in and out.

-d With either interface display (option -i or an interval, as described below), show the number of dropped packets.

-f address_family

Limit statistics or address control block reports to those of the specified address family. The following address families are recognized: inet, for AF_INET, inet6, for AF_INET6 and unix, for AF_UNIX.

-g Show information related to multicast (group address) routing. By default, show the IP Multicast virtual-interface and routing tables. If the **-s** option is also present, show multicast routing statistics.

-I interface
Show information about the specified interface; used with a wait interval as described below. If the **-s** option is present, show per-interface protocol statistics on the interface for the specified address_family or protocol, or for all protocol families.

-i Show the state of interfaces which have been auto-configured (interfaces statically configured into a system, but not located at boot time are not shown). If the **-a** options is also present, multicast addresses currently in use are shown for each Ethernet interface and for each IP interface address. Multicast addresses are shown on separate lines following the interface address with which they are associated. If the **-s** option is present, show per-interface statistics on all interfaces for the specified address_family or protocol, or for all protocol families.

-L Show the size of the various listen queues. The first count shows the number of unaccepted connections. The second count shows the amount of unaccepted incomplete connections. The third count is the maximum number of queued connections.

-l Print full IPv6 address.

-M Extract values associated with the name list from the specified core instead of the default /dev/kmem.

-m Show statistics recorded by the memory management routines (the network manages a private pool of memory buffers).

-N Extract the name list from the specified system instead of the default /kernel.

-n Show network addresses as numbers (normally netstat interprets addresses and attempts to display them symbolically). This option may be used with any of the display formats.

-p protocol
Show statistics about protocol, which is either a well-known name for a protocol or an alias for it. Some protocol names and aliases are listed in the file /etc/protocols. The special protocol name 'bdg' is used to show bridging statistics. A null response typi-

cally means that there are no interesting numbers to report. The program will complain if protocol is unknown or if there is no statistics routine for it.

-r Show the routing tables. Use with **-a** to show protocol-cloned routes. When **-s** is also present, show routing statistics instead. When **-l** is also present, netstat assumes more columns are there and the maximum transmission unit ('**mtu**') are also displayed.

-s Show per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.

-W In certain displays, avoid truncating addresses even if this causes some fields to overflow.

-w wait
Show network interface statistics at intervals of wait seconds.

OUTPUT

The default display, for active sockets, shows the local and remote addresses, send and receive queue sizes (in bytes), protocol, and the internal state of the protocol. Address formats are of the form '**host.port**' or '**network.port**' if a socket's address specifies a network but no specific host address. If known, the host and network addresses are displayed symbolically according to the databases **/etc/hosts** and **/etc/networks**, respectively. If a symbolic name for an address is unknown, or if the **-n** option is specified, the address is printed numerically, according to the address family. For more information regarding the Internet '**dot format**', refer to **inet(3)**. Unspecified, or '**wildcard**', addresses and ports appear as '*****'.

Internet domain socket states:

CLOSED: The socket is not in use.

LISTEN: The socket is listening for incoming connections. Unconnected listening sockets like these are only displayed when using the **-a** option.

SYN_SENT: The socket is actively trying to establish a connection to a remote peer.

SYN_RCVD: The socket has passively received a connection request from a remote peer.

ESTABLISHED: The socket has an established connection between a local application and a remote peer.

CLOSE_WAIT: The socket connection has been closed by the remote peer, and the system is waiting for the local application to close its half of the connection.

LAST_ACK: The socket connection has been closed by the remote peer, the local application has closed its half of the connection, and the system is waiting for the remote peer to acknowledge the close.

FIN_WAIT_1: The socket connection has been closed by the local application, the remote peer has not yet acknowledged the close, and the system is waiting for it to close its half of the connection.

FIN_WAIT_2: The socket connection has been closed by the local application, the remote peer has acknowledged the close, and the system is waiting for it to close its half of the connection.

CLOSING: The socket connection has been closed by the local application and the remote peer simultaneously, and the remote peer has not yet acknowledged the close attempt of the local application.

TIME_WAIT: The socket connection has been closed by the local application, the remote peer has closed its half of the connection, and the system is waiting to be sure that the remote peer received the last acknowledgement.

The interface display provides a table of cumulative statistics regarding packets transferred, errors, and collisions. The network addresses of the interface and the maximum transmission unit ('mtu') are also displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows a collection of information about the route stored as binary choices. The individual flags are discussed in more detail in the route(8) and route(4) manual pages. The mapping between letters and flags is:

1	RTF_PROTO1	Protocol specific routing flag #1
2	RTF_PROTO2	Protocol specific routing flag #2
3	RTF_PROTO3	Protocol specific routing flag #3
B	RTF_BLACKHOLE	Just discard packets (during updates)
b	RTF_BROADCAST	The route represents a broadcast address
C	RTF_CLONING	Generate new routes on use
c	RTF_PRCLONING	Protocol-specified generate new routes on use
D	RTF_DYNAMIC	Created dynamically (by redirect)
G	RTF_GATEWAY	Destination requires forwarding by intermediary
H	RTF_HOST	Host entry (net otherwise)
L	RTF_LLINFO	Valid protocol to link address translation
M	RTF_MODIFIED	Modified dynamically (by redirect)
R	RTF_REJECT	Host or net unreachable
S	RTF_STATIC	Manually added
U	RTF_UP	Route usable
W	RTF_WASCLONED	Route was generated as a result of cloning
X	RTF_XRESOLVE	External daemon translates proto to link address

Direct routes are created for each interface attached to the local host; the gateway field for such entries shows the address of the outgoing interface. The refcnt field gives the current number of active uses of the route. Connection oriented protocols normally hold on to a single route for the duration of a connection while connectionless protocols obtain a route while sending to the same destination. The use field provides a count of the number of packets sent using that route. The interface entry indicates the network interface utilized for the route.

When netstat is invoked with the -w option and a wait interval argument, it displays a running count of statistics related to network interfaces.

An obsolete version of this option used a numeric parameter with no option, and is currently supported for backward compatibility. By default, this display summarizes information for all interfaces. Information for a specific interface may be displayed with the -I option.

SEE ALSO

fstat(1), nfsstat(1), ps(1), sockstat(1), inet(4), unix(4), hosts(5), networks(5), protocols(5), services(5), iostat(8), trpt(8), vmstat(8)

HISTORY

The netstat command appeared in 4.2BSD.

IPv6 support was added by WIDE/KAME project.

FILES

/kernel default kernel namelist
/dev/kmem default memory file

BUGS

The notion of errors is ill-defined.

Darwin June 15, 2001 Darwin