

**NAME**

lnstat – unified linux network statistics

**SYNOPSIS**

**lnstat** [*options*]

**DESCRIPTION**

This manual page documents briefly the **lnstat** command.

**lnstat** is a generalized and more feature-complete replacement for the old **rtstat** program. It is commonly used to periodically print a selection of statistical values exported by the kernel. In addition to routing cache statistics, it supports any kind of statistics the linux kernel exports via a file in `/proc/net/stat/`.

Each file in `/proc/net/stat/` contains a header line listing the column names. These names are used by **lnstat** as keys for selecting which statistics to print. For every CPU present in the system, a line follows which lists the actual values for each column of the file. **lnstat** sums these values up (which in fact are counters) before printing them. After each interval, only the difference to the last value is printed.

Files and columns may be selected by using the **-f** and **-k** parameters. By default, all columns of all files are printed.

**OPTIONS**

lnstat supports the following options.

**-h, --help**

Show summary of options.

**-V, --version**

Show version of program.

**-c, --count <count>**

Print <count> number of intervals.

**-d, --dump**

Dump list of available files/keys.

**-f, --file <file>**

Statistics file to use, may be specified multiple times. By default all files in `/proc/net/stat` are scanned.

**-i, --interval <intv>**

Set interval to 'intv' seconds.

**-j, --json**

Display results in JSON format

**-k, --keys k,k,k,...**

Display only keys specified. Each key **k** is of the form **[file:]key**. If **<file>** is given, the search for the given key is limited to that file. Otherwise the first file containing the searched key is being used.

**-s, --subject [0-2]**

Specify display of subject/header. '0' means no header at all, '1' prints a header only at start of the program and '2' prints a header every 20 lines.

**-w, --width n,n,n,...**

Width for each field.

**USAGE EXAMPLES**

**# lnstat -d**

Get a list of supported statistics files.

**# lnstat -k arp\_cache:entries,rt\_cache:in\_hit,arp\_cache:destroys**

Select the specified files and keys.

**# lnstat -i 10**

Use an interval of 10 seconds.

**# lnstat -f ip\_conntrack**

Use only the specified file for statistics.

**# lnstat -s 0**

Do not print a header at all.

**# lnstat -s 20**

Print a header at start and every 20 lines.

**# lnstat -c -1 -i 1 -f rt\_cache -k entries,in\_hit,in\_slow\_tot**

Display statistics for keys entries, in\_hit and in\_slow\_tot of field rt\_cache every second.

**FILES****/proc/net/stat/arp\_cache, /proc/net/stat/ndisc\_cache**

Statistics around neighbor cache and ARP. **arp\_cache** is for IPv4, **ndisc\_cache** is the same for IPv6.

**entries** Number of entries in the neighbor table.

**allocs** How many neighbor entries have been allocated.

**destroys** How many neighbor entries have been removed.

**hash\_grows** How often the neighbor (hash) table was increased.

**lookups** How many lookups were performed.

**hits** How many **lookups** were successful.

**res\_failed** How many neighbor lookups failed.

**rcv\_probes\_mcast** How many multicast neighbor solicitations were received. (IPv6 only.)

**rcv\_probes\_ucast** How many unicast neighbor solicitations were received. (IPv6 only.)

**periodic\_gc\_runs** How many garbage collection runs were executed.

**forced\_gc\_runs** How many forced garbage collection runs were executed. Happens when adding an entry and the table is too full.

**unresolved\_discards** How many neighbor table entries were discarded due to lookup failure.

**table\_fulls** Number of table overflows. Happens if table is full and forced GC run (see **forced\_gc\_runs**) has failed.

**/proc/net/stat/ip\_conntrack, /proc/net/stat/nf\_conntrack**

Conntrack related counters. **ip\_conntrack** is for backwards compatibility with older userspace only and shows the same data as **nf\_conntrack**.

**entries** Number of entries in conntrack table.

**searched** Number of conntrack table lookups performed.

**found** Number of **searched** entries which were successful.

**new** Number of conntrack entries added which were not expected before.

**invalid** Number of packets seen which can not be tracked.

**ignore** Number of packets seen which are already connected to a conntrack entry.

**delete** Number of conntrack entries which were removed.

**delete\_list** Number of conntrack entries which were put to dying list.

**insert** Number of entries inserted into the list.

**insert\_failed** Number of entries for which list insertion was attempted but failed (happens if the same entry is already present).

**drop** Number of packets dropped due to conntrack failure. Either new conntrack entry allocation failed, or protocol helper dropped the packet.

**early\_drop** Number of dropped conntrack entries to make room for new ones, if maximum table size was reached.

**icmp\_error** Number of packets which could not be tracked due to error situation. This is a subset of **invalid**.

**expect\_new** Number of conntrack entries added after an expectation for them was already present.

**expect\_create** Number of expectations added.

**expect\_delete** Number of expectations deleted.

**search\_restart** Number of conntrack table lookups which had to be restarted due to hashtable re-sizes.

#### **/proc/net/stat/route**

Routing cache statistics.

**entries** Number of entries in routing cache.

**in\_hit** Number of route cache hits for incoming packets. Deprecated since IP route cache removal, therefore always zero.

**in\_slow\_tot** Number of routing cache entries added for input traffic.

**in\_slow\_mc** Number of multicast routing cache entries added for input traffic.

**in\_no\_route** Number of input packets for which no routing table entry was found.

**in\_brd** Number of matched input broadcast packets.

**in\_martian\_dst** Number of incoming martian destination packets.

**in\_martian\_src** Number of incoming martian source packets.

**out\_hit** Number of route cache hits for outgoing packets. Deprecated since IP route cache removal, therefore always zero.

**out\_slow\_tot** Number of routing cache entries added for output traffic.

**out\_slow\_mc** Number of multicast routing cache entries added for output traffic.

**gc\_total** Total number of garbage collection runs. Deprecated since IP route cache removal, therefore always zero.

**gc\_ignored** Number of ignored garbage collection runs due to minimum GC interval not reached and routing cache not full. Deprecated since IP route cache removal, therefore always zero.

**gc\_goal\_miss** Number of garbage collector goal misses. Deprecated since IP route cache removal, therefore always zero.

**gc\_dst\_overflow** Number of destination cache overflows. Deprecated since IP route cache removal, therefore always zero.

**in\_hlist\_search** Number of hash table list traversals for input traffic. Deprecated since IP route cache removal, therefore always zero.

**out\_hlist\_search** Number of hash table list traversals for output traffic. Deprecated since IP route cache removal, therefore always zero.

## SEE ALSO

**ip(8)**

## AUTHOR

lnstat was written by Harald Welte <laforge@gnumonks.org>.

This manual page was written by Michael Prokop <mika@grml.org> for the Debian project (but may be used by others).