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 resizes.

/proc/net/stat/rt_cache
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).