



---

[News](#) | [Features](#) | [Download](#) | [GitHub](#) | [Documentation](#) | [Tutorial](#) | [FAQ](#) | [Contact](#)

Useful links:  
[Freecode](#)  
[IceWalkers](#)

[The Linux Foundation](#)  
[Linux Kernel Archives](#)  
[Linux Kernel Mailing List](#)  
[The Linux Documentation Project](#)



## *sar manual page*

### NAME

sar - Collect, report, or save system activity information.

### SYNOPSIS

```
sar [-A] [-B] [-b] [-C] [-d] [-F] [-H] [-h] [-p] [-q] [-R] [-r] [-S] [-t] [-u [ALL]] [-V] [-v] [-W] [-w] [-y] [-l {int [,...] | SUM | ALL | XALL}] [-P {cpu [,...] | ALL}] [-m {keyword [,...] | ALL}] [-n {keyword [,...] | ALL}] [-j {ID | LABEL | PATH | UUID | ...}] [-f [filename]] [-o [filename]] [-[0-9]+] [-i interval] [-s [hh:mm:ss]] [-e [hh:mm:ss]] [interval [count]]
```

### DESCRIPTION

The sar command writes to standard output the contents of selected cumulative activity counters in the operating system. The accounting system, based on the values in the count and interval parameters, writes information the specified number of times spaced at the specified intervals in seconds. If the interval parameter is set to zero, the sar command displays the average statistics for the time since the system was started. If the interval parameter is specified without the count parameter, then reports

are generated continuously. The collected data can also be saved in the file specified by the `-o filename` flag, in addition to being displayed onto the screen. If filename is omitted, sar uses the standard system activity daily data file, the `/var/log/sa/sadd` file, where the `dd` parameter indicates the current day. By default all the data available from the kernel are saved in the data file.

The sar command extracts and writes to standard output records previously saved in a file. This file can be either the one specified by the `-f` flag or, by default, the standard system activity daily data file. It is also possible to enter `-1`, `-2` etc. as an argument to sar to display data of that days ago. For example, `-1` will point at the standard system activity file of yesterday.

Without the `-P` flag, the sar command reports system-wide (global among all processors) statistics, which are calculated as averages for values expressed as percentages, and as sums otherwise. If the `-P` flag is given, the sar command reports activity which relates to the specified processor or processors. If `-P ALL` is given, the sar command reports statistics for each individual processor and global statistics among all processors.

You can select information about specific system activities using flags. Not specifying any flags selects only CPU activity. Specifying the `-A` flag selects all possible activities.

The default version of the sar command (CPU utilization report) might be one of the first facilities the user runs to begin system activity investigation, because it monitors major system resources. If CPU utilization is near 100 percent (user + nice + system), the workload sampled is CPU-bound.

If multiple samples and multiple reports are desired, it is convenient to specify an output file for the sar command. Run the sar command as a background process. The syntax for this is:

```
sar -o datafile interval count >/dev/null 2>&1 &
```

All data are captured in binary form and saved to a file (datafile). The data can then be selectively displayed with the sar command using the `-f` option. Set the interval and count parameters to select count records at interval second intervals. If the count parameter is not set, all the records saved in the file will be selected. Collection of data in this manner is useful to characterize system usage over a period of time and determine peak usage hours.

Note: The sar command only reports on local activities.

## OPTIONS

## -A

This is equivalent to specifying -bBdFHqrRSvwWy -I SUM -I XALL -m ALL -n ALL -u ALL -P ALL.

## -B

Report paging statistics. The following values are displayed:

### pgpgin/s

Total number of kilobytes the system paged in from disk per second.

### pgpgout/s

Total number of kilobytes the system paged out to disk per second.

### fault/s

Number of page faults (major + minor) made by the system per second. This is not a count of page faults that generate I/O, because some page faults can be resolved without I/O.

### majflt/s

Number of major faults the system has made per second, those which have required loading a memory page from disk.

### pgfree/s

Number of pages placed on the free list by the system per second.

### pgscank/s

Number of pages scanned by the kswapd daemon per second.

### pgscand/s

Number of pages scanned directly per second.

### pgsteal/s

Number of pages the system has reclaimed from cache (pagecache and swapcache) per second to satisfy its memory demands.

### %vmeff

Calculated as  $\text{pgsteal} / \text{pgscan}$ , this is a metric of the efficiency of page reclaim. If it is near 100% then almost every page coming off the tail of the inactive list is being reaped. If it gets

too low (e.g. less than 30%) then the virtual memory is having some difficulty. This field is displayed as zero if no pages have been scanned during the interval of time.

**-b**

Report I/O and transfer rate statistics. The following values are displayed:

**tps**

Total number of transfers per second that were issued to physical devices. A transfer is an I/O request to a physical device. Multiple logical requests can be combined into a single I/O request to the device. A transfer is of indeterminate size.

**rtps**

Total number of read requests per second issued to physical devices.

**wtps**

Total number of write requests per second issued to physical devices.

**bread/s**

Total amount of data read from the devices in blocks per second. Blocks are equivalent to sectors and therefore have a size of 512 bytes.

**bwrtn/s**

Total amount of data written to devices in blocks per second.

**-C**

When reading data from a file, tell sar to display comments that have been inserted by sadc.

**-d**

Report activity for each block device. When data are displayed, the device specification *dev m-n* is generally used (DEV column). *m* is the major number of the device and *n* its minor number. Device names may also be pretty-printed if option **-p** is used or persistent device names can be printed if option **-j** is used (see below). Note that disk activity depends on sadc options **"-S DISK"** and **"-S XDISK"** to be collected. The following values are displayed:

**tps**

Indicate the number of transfers per second that were issued to the device. Multiple logical requests can be combined into a single I/O request to the device. A transfer is of indeterminate size.

`rd_sec/s`

Number of sectors read from the device. The size of a sector is 512 bytes.

`wr_sec/s`

Number of sectors written to the device. The size of a sector is 512 bytes.

`avgrq-sz`

The average size (in sectors) of the requests that were issued to the device.

`avgqu-sz`

The average queue length of the requests that were issued to the device.

`await`

The average time (in milliseconds) for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.

`svctm`

The average service time (in milliseconds) for I/O requests that were issued to the device. Warning! Do not trust this field any more. This field will be removed in a future sysstat version.

`%util`

Percentage of CPU time during which I/O requests were issued to the device (bandwidth utilization for the device). Device saturation occurs when this value is close to 100%.

`-e [ hh:mm:ss ]`

Set the ending time of the report. The default ending time is 18:00:00. Hours must be given in 24-hour format. This option can be used when data are read from or written to a file (options `-f` or `-o` ).

`-f [ filename ]`

Extract records from filename (created by the `-o filename` flag).

The default value of the filename parameter is the current daily data file, the `/var/log/sa/sadd` file. The `-f` option is exclusive of the `-o` option.

## **-F**

Display statistics for currently mounted filesystems. Pseudo-filesystems are ignored. At the end of the report, `sar` will display a summary of all those filesystems. Note that filesystems statistics depend on `sadc` option `"-S XDISK"` to be collected. The following values are displayed:

### **MBsfree**

Total amount a free space in megabytes (including space available only to privileged user).

### **MBfsused**

Total amount of space used in megabytes.

### **%fsused**

Percentage of filesystem space used, as seen by a privileged user.

### **%ufsused**

Percentage of filesystem space used, as seen by an unprivileged user.

### **lfree**

Total number of free file nodes in filesystem.

### **lused**

Total number of file nodes used in filesystem.

### **%lused**

Percentage of file nodes used in filesystem.

## **-H**

Report hugepages utilization statistics. The following values are displayed:

### **kbhugfree**

Amount of hugepages memory in kilobytes that is not yet allocated.

### **kbhugused**

Amount of hugepages memory in kilobytes that has been allocated.

`%hugused`

Percentage of total hugepages memory that has been allocated.

`-h`

Display a short help message then exit.

`-I { int [,...] | SUM | ALL | XALL }`

Report statistics for a given interrupt. `int` is the interrupt number. Specifying multiple `-I int` parameters on the command line will look at multiple independent interrupts. The **SUM** keyword indicates that the total number of interrupts received per second is to be displayed. The **ALL** keyword indicates that statistics from the first 16 interrupts are to be reported, whereas the **XALL** keyword indicates that statistics from all interrupts, including potential APIC interrupt sources, are to be reported. Note that interrupt statistics depend on `sadc` option `"-S INT"` to be collected.

`-i interval`

Select data records at seconds as close as possible to the number specified by the interval parameter.

`-j { ID | LABEL | PATH | UUID | ... }`

Display persistent device names. Use this option in conjunction with option `-d`. Options **ID**, **LABEL**, etc. specify the type of the persistent name. These options are not limited, only prerequisite is that directory with required persistent names is present in `/dev/disk`. If persistent name is not found for the device, the device name is pretty-printed (see option `-p` below).

`-m { keyword [,...] | ALL }`

Report power management statistics. Note that these statistics depend on `sadc` option `"-S POWER"` to be collected.

Possible keywords are **CPU**, **FAN**, **FREQ**, **IN**, **TEMP** and **USB**.

With the **CPU** keyword, statistics about CPU are reported. The following value is displayed:

`MHz`

Instantaneous CPU clock frequency in MHz.

With the **FAN** keyword, statistics about fans speed are reported. The following values are displayed:

rpm

Fan speed expressed in revolutions per minute.

drpm

This field is calculated as the difference between current fan speed (rpm) and its low limit (fan\_min).

DEVICE

Sensor device name.

With the **FREQ** keyword, statistics about CPU clock frequency are reported. The following value is displayed:

wghMHz

Weighted average CPU clock frequency in MHz. Note that the cpufreq-stats driver must be compiled in the kernel for this option to work.

With the **IN** keyword, statistics about voltage inputs are reported. The following values are displayed:

inV

Voltage input expressed in Volts.

%in

Relative input value. A value of 100% means that voltage input has reached its high limit (in\_max) whereas a value of 0% means that it has reached its low limit (in\_min).

DEVICE

Sensor device name.

With the **TEMP** keyword, statistics about devices temperature are reported. The following values are displayed:

degC

Device temperature expressed in degrees Celsius.



`%temp`

Relative device temperature. A value of 100% means that temperature has reached its high limit (`temp_max`).

`DEVICE`

Sensor device name.

With the **USB** keyword, the `sar` command takes a snapshot of all the USB devices currently plugged into the system. At the end of the report, `sar` will display a summary of all those USB devices. The following values are displayed:

`BUS`

Root hub number of the USB device.

`idvendor`

Vendor ID number (assigned by USB organization).

`idprod`

Product ID number (assigned by Manufacturer).

`maxpower`

Maximum power consumption of the device (expressed in mA).

`manufact`

Manufacturer name.

`product`

Product name.

`-n { keyword [...] | ALL }`

Report network statistics.

Possible keywords are **DEV**, **EDEV**, **NFS**, **NFSD**, **SOCK**, **IP**, **EIP**, **ICMP**, **EICMP**, **TCP**, **ETCP**, **UDP**, **SOCK6**, **IP6**, **EIP6**, **ICMP6** and **UDP6**.

With the **DEV** keyword, statistics from the network devices are reported. The following values are displayed:

`IFACE`

Name of the network interface for which statistics are reported.

rxpck/s

Total number of packets received per second.

txpck/s

Total number of packets transmitted per second.

rxkB/s

Total number of kilobytes received per second.

txkB/s

Total number of kilobytes transmitted per second.

rxcmp/s

Number of compressed packets received per second (for cslip etc.).

txcmp/s

Number of compressed packets transmitted per second.

rxmcast/s

Number of multicast packets received per second.

%ifutil

Utilization percentage of the network interface. For half-duplex interfaces, utilization is calculated using the sum of rxkB/s and txkB/s as a percentage of the interface speed. For full-duplex, this is the greater of rxkB/S or txkB/s.

With the **EDEV** keyword, statistics on failures (errors) from the network devices are reported. The following values are displayed:

IFACE

Name of the network interface for which statistics are reported.

rxerr/s

Total number of bad packets received per second.

txerr/s

Total number of errors that happened per second while transmitting packets.

coll/s

Number of collisions that happened per second while transmitting packets.

rxdrop/s

Number of received packets dropped per second because of a lack of space in linux buffers.

txdrop/s

Number of transmitted packets dropped per second because of a lack of space in linux buffers.

txcarr/s

Number of carrier-errors that happened per second while transmitting packets.

rxfram/s

Number of frame alignment errors that happened per second on received packets.

rxfifo/s

Number of FIFO overrun errors that happened per second on received packets.

txfifo/s

Number of FIFO overrun errors that happened per second on transmitted packets.

With the **NFS** keyword, statistics about NFS client activity are reported. The following values are displayed:

call/s

Number of RPC requests made per second.

retrans/s

Number of RPC requests per second, those which needed to be retransmitted (for example because of a server timeout).

read/s

Number of 'read' RPC calls made per second.

write/s

Number of 'write' RPC calls made per second.

`access/s`

Number of 'access' RPC calls made per second.

`getatt/s`

Number of 'getattr' RPC calls made per second.

With the **NFSD** keyword, statistics about NFS server activity are reported. The following values are displayed:

`scall/s`

Number of RPC requests received per second.

`badcall/s`

Number of bad RPC requests received per second, those whose processing generated an error.

`packet/s`

Number of network packets received per second.

`udp/s`

Number of UDP packets received per second.

`tcp/s`

Number of TCP packets received per second.

`hit/s`

Number of reply cache hits per second.

`miss/s`

Number of reply cache misses per second.

`sread/s`

Number of 'read' RPC calls received per second.

`swrite/s`

Number of 'write' RPC calls received per second.

`saccess/s`

Number of 'access' RPC calls received per second.

## sgetatt/s

Number of 'getattr' RPC calls received per second.

With the **SOCK** keyword, statistics on sockets in use are reported (IPv4). The following values are displayed:

## totsck

Total number of sockets used by the system.

## tcpsck

Number of TCP sockets currently in use.

## udpsck

Number of UDP sockets currently in use.

## rawsck

Number of RAW sockets currently in use.

## ip-frag

Number of IP fragments currently in queue.

## tcp-tw

Number of TCP sockets in TIME\_WAIT state.

With the **IP** keyword, statistics about IPv4 network traffic are reported. Note that IPv4 statistics depend on sadc option "-S SNMP" to be collected. The following values are displayed (formal SNMP names between square brackets):

## irec/s

The total number of input datagrams received from interfaces per second, including those received in error [ipInReceives].

## fwddgm/s

The number of input datagrams per second, for which this entity was not their final IP destination, as a result of which an attempt was made to find a route to forward them to that final destination [ipForwDatagrams].

## idel/s

The total number of input datagrams successfully delivered per second to IP user-protocols (including ICMP) [ipInDelivers].

### orq/s

The total number of IP datagrams which local IP user-protocols (including ICMP) supplied per second to IP in requests for transmission [ipOutRequests]. Note that this counter does not include any datagrams counted in fwddgm/s.

### asmrq/s

The number of IP fragments received per second which needed to be reassembled at this entity [ipReasmReqds].

### asmok/s

The number of IP datagrams successfully re-assembled per second [ipReasmOKs].

### fragok/s

The number of IP datagrams that have been successfully fragmented at this entity per second [ipFragOKs].

### fragcrt/s

The number of IP datagram fragments that have been generated per second as a result of fragmentation at this entity [ipFragCreates].

With the **EIP** keyword, statistics about IPv4 network errors are reported. Note that IPv4 statistics depend on sadc option "-S SNMP" to be collected. The following values are displayed (formal SNMP names between square brackets):

### ihdrerr/s

The number of input datagrams discarded per second due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc. [ipInHdrErrors]

### iadrerr/s

The number of input datagrams discarded per second because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., 0.0.0.0) and addresses of unsupported Classes (e.g., Class E). For entities which are not IP routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address [ipInAddrErrors].

## iukwnpr/s

The number of locally-addressed datagrams received successfully but discarded per second because of an unknown or unsupported protocol [ipInUnknownProtos].

## idisc/s

The number of input IP datagrams per second for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space) [ipInDiscards]. Note that this counter does not include any datagrams discarded while awaiting re-assembly.

## odisc/s

The number of output IP datagrams per second for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer space) [ipOutDiscards]. Note that this counter would include datagrams counted in fwddgm/s if any such packets met this (discretionary) discard criterion.

## onort/s

The number of IP datagrams discarded per second because no route could be found to transmit them to their destination [ipOutNoRoutes]. Note that this counter includes any packets counted in fwddgm/s which meet this 'no-route' criterion. Note that this includes any datagrams which a host cannot route because all of its default routers are down.

## asmf/s

The number of failures detected per second by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc) [ipReasmFails]. Note that this is not necessarily a count of discarded IP fragments since some algorithms can lose track of the number of fragments by combining them as they are received.

## frag/s

The number of IP datagrams that have been discarded per second because they needed to be fragmented at this entity but could not be, e.g., because their Don't Fragment flag was set [ipFragFails].

With the **ICMP** keyword, statistics about ICMPv4 network traffic are reported. Note that ICMPv4 statistics depend on sadc option "-S SNMP" to be collected. The following values are

displayed (formal SNMP names between square brackets):

[imsg/s](#)

The total number of ICMP messages which the entity received per second [icmpInMsgs]. Note that this counter includes all those counted by ierr/s.

[omsg/s](#)

The total number of ICMP messages which this entity attempted to send per second [icmpOutMsgs]. Note that this counter includes all those counted by oerr/s.

[iech/s](#)

The number of ICMP Echo (request) messages received per second [icmpInEchos].

[iechr/s](#)

The number of ICMP Echo Reply messages received per second [icmpInEchoReps].

[oech/s](#)

The number of ICMP Echo (request) messages sent per second [icmpOutEchos].

[oechr/s](#)

The number of ICMP Echo Reply messages sent per second [icmpOutEchoReps].

[itm/s](#)

The number of ICMP Timestamp (request) messages received per second [icmpInTimestamps].

[itmr/s](#)

The number of ICMP Timestamp Reply messages received per second [icmpInTimestampReps].

[otm/s](#)

The number of ICMP Timestamp (request) messages sent per second [icmpOutTimestamps].

[otmr/s](#)

The number of ICMP Timestamp Reply messages sent per second [icmpOutTimestampReps].



#### iadrmk/s

The number of ICMP Address Mask Request messages received per second [icmpInAddrMasks].

#### iadrmkr/s

The number of ICMP Address Mask Reply messages received per second [icmpInAddrMaskReps].

#### oadrmk/s

The number of ICMP Address Mask Request messages sent per second [icmpOutAddrMasks].

#### oadrmkr/s

The number of ICMP Address Mask Reply messages sent per second [icmpOutAddrMaskReps].

With the **EICMP** keyword, statistics about ICMPv4 error messages are reported. Note that ICMPv4 statistics depend on sadc option "-S SNMP" to be collected. The following values are displayed (formal SNMP names between square brackets):

#### ierr/s

The number of ICMP messages per second which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.) [icmpInErrors].

#### oerr/s

The number of ICMP messages per second which this entity did not send due to problems discovered within ICMP such as a lack of buffers [icmpOutErrors].

#### idstunr/s

The number of ICMP Destination Unreachable messages received per second [icmpInDestUnreachs].

#### odstunr/s

The number of ICMP Destination Unreachable messages sent per second [icmpOutDestUnreachs].

#### itmex/s

The number of ICMP Time Exceeded messages received per second [icmpInTimeExcds].

#### otmex/s

The number of ICMP Time Exceeded messages sent per second [icmpOutTimeExcds].

#### iparmpb/s

The number of ICMP Parameter Problem messages received per second [icmpInParmProbs].

#### oparmpb/s

The number of ICMP Parameter Problem messages sent per second [icmpOutParmProbs].

#### isrcq/s

The number of ICMP Source Quench messages received per second [icmpInSrcQuenchs].

#### osrcq/s

The number of ICMP Source Quench messages sent per second [icmpOutSrcQuenchs].

#### iredir/s

The number of ICMP Redirect messages received per second [icmpInRedirects].

#### oredir/s

The number of ICMP Redirect messages sent per second [icmpOutRedirects].

With the **TCP** keyword, statistics about TCPv4 network traffic are reported. Note that TCPv4 statistics depend on sadc option "-S SNMP" to be collected. The following values are displayed (formal SNMP names between square brackets):

#### active/s

The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state per second [tcpActiveOpens].

#### passive/s

The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state per second [tcpPassiveOpens].

### iseg/s

The total number of segments received per second, including those received in error [tcpInSegs]. This count includes segments received on currently established connections.

### oseg/s

The total number of segments sent per second, including those on current connections but excluding those containing only retransmitted octets [tcpOutSegs].

With the **ETCP** keyword, statistics about TCPv4 network errors are reported. Note that TCPv4 statistics depend on sadc option "-S SNMP" to be collected. The following values are displayed (formal SNMP names between square brackets):

### atmptf/s

The number of times per second TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times per second TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state [tcpAttemptFails].

### estres/s

The number of times per second TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state [tcpEstabResets].

### retrans/s

The total number of segments retransmitted per second - that is, the number of TCP segments transmitted containing one or more previously transmitted octets [tcpRetransSegs].

### isegerr/s

The total number of segments received in error (e.g., bad TCP checksums) per second [tcpInErrs].

### orsts/s

The number of TCP segments sent per second containing the RST flag [tcpOutRsts].

With the **UDP** keyword, statistics about UDPv4 network traffic are reported. Note that UDPv4 statistics depend on sadc option

"-S SNMP" to be collected. The following values are displayed (formal SNMP names between square brackets):

idgm/s

The total number of UDP datagrams delivered per second to UDP users [udpInDatagrams].

odgm/s

The total number of UDP datagrams sent per second from this entity [udpOutDatagrams].

noport/s

The total number of received UDP datagrams per second for which there was no application at the destination port [udpNoPorts].

idgmerr/s

The number of received UDP datagrams per second that could not be delivered for reasons other than the lack of an application at the destination port [udpInErrors].

With the **SOCK6** keyword, statistics on sockets in use are reported (IPv6). Note that IPv6 statistics depend on sadc option "-S IPV6" to be collected. The following values are displayed:

tcp6sck

Number of TCPv6 sockets currently in use.

udp6sck

Number of UDPv6 sockets currently in use.

raw6sck

Number of RAWv6 sockets currently in use.

ip6-frag

Number of IPv6 fragments currently in use.

With the **IP6** keyword, statistics about IPv6 network traffic are reported. Note that IPv6 statistics depend on sadc option "-S IPV6" to be collected. The following values are displayed (formal SNMP names between square brackets):

irec6/s

The total number of input datagrams received from interfaces

per second, including those received in error [ipv6IfStatsInReceives].

#### fwddgm6/s

The number of output datagrams per second which this entity received and forwarded to their final destinations [ipv6IfStatsOutForwDatagrams].

#### idel6/s

The total number of datagrams successfully delivered per second to IPv6 user-protocols (including ICMP) [ipv6IfStatsInDelivers].

#### orq6/s

The total number of IPv6 datagrams which local IPv6 user-protocols (including ICMP) supplied per second to IPv6 in requests for transmission [ipv6IfStatsOutRequests]. Note that this counter does not include any datagrams counted in fwddgm6/s.

#### asmrq6/s

The number of IPv6 fragments received per second which needed to be reassembled at this interface [ipv6IfStatsReasmReqds].

#### asmok6/s

The number of IPv6 datagrams successfully reassembled per second [ipv6IfStatsReasmOKs].

#### imcpck6/s

The number of multicast packets received per second by the interface [ipv6IfStatsInMcastPkts].

#### omcpck6/s

The number of multicast packets transmitted per second by the interface [ipv6IfStatsOutMcastPkts].

#### fragok6/s

The number of IPv6 datagrams that have been successfully fragmented at this output interface per second [ipv6IfStatsOutFragOKs].

#### fragcr6/s

The number of output datagram fragments that have been

generated per second as a result of fragmentation at this output interface [ipv6IfStatsOutFragCreates].

With the **EIP6** keyword, statistics about IPv6 network errors are reported. Note that IPv6 statistics depend on sadc option "-S IPV6" to be collected. The following values are displayed (formal SNMP names between square brackets):

#### ihdrer6/s

The number of input datagrams discarded per second due to errors in their IPv6 headers, including version number mismatch, other format errors, hop count exceeded, errors discovered in processing their IPv6 options, etc. [ipv6IfStatsInHdrErrors]

#### iadrer6/s

The number of input datagrams discarded per second because the IPv6 address in their IPv6 header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (e.g., ::0) and unsupported addresses (e.g., addresses with unallocated prefixes). For entities which are not IPv6 routers and therefore do not forward datagrams, this counter includes datagrams discarded because the destination address was not a local address [ipv6IfStatsInAddrErrors].

#### iukwnp6/s

The number of locally-addressed datagrams received successfully but discarded per second because of an unknown or unsupported protocol [ipv6IfStatsInUnknownProtos].

#### i2big6/s

The number of input datagrams that could not be forwarded per second because their size exceeded the link MTU of outgoing interface [ipv6IfStatsInTooBigErrors].

#### idisc6/s

The number of input IPv6 datagrams per second for which no problems were encountered to prevent their continued processing, but which were discarded (e.g., for lack of buffer space) [ipv6IfStatsInDiscards]. Note that this counter does not include any datagrams discarded while awaiting re-assembly.

#### odisc6/s

The number of output IPv6 datagrams per second for which no problem was encountered to prevent their transmission to their destination, but which were discarded (e.g., for lack of buffer

space) [ipv6IfStatsOutDiscards]. Note that this counter would include datagrams counted in fwddgm6/s if any such packets met this (discretionary) discard criterion.

#### inort6/s

The number of input datagrams discarded per second because no route could be found to transmit them to their destination [ipv6IfStatsInNoRoutes].

#### onort6/s

The number of locally generated IP datagrams discarded per second because no route could be found to transmit them to their destination [unknown formal SNMP name].

#### asmf6/s

The number of failures detected per second by the IPv6 re-assembly algorithm (for whatever reason: timed out, errors, etc.) [ipv6IfStatsReasmFails]. Note that this is not necessarily a count of discarded IPv6 fragments since some algorithms can lose track of the number of fragments by combining them as they are received.

#### fragf6/s

The number of IPv6 datagrams that have been discarded per second because they needed to be fragmented at this output interface but could not be [ipv6IfStatsOutFragFails].

#### itrpck6/s

The number of input datagrams discarded per second because datagram frame didn't carry enough data [ipv6IfStatsInTruncatedPkts].

With the **ICMP6** keyword, statistics about ICMPv6 network traffic are reported. Note that ICMPv6 statistics depend on sadc option "-S IPV6" to be collected. The following values are displayed (formal SNMP names between square brackets):

#### imsg6/s

The total number of ICMP messages received by the interface per second which includes all those counted by ierr6/s [ipv6IfIcmpInMsgs].

#### omsg6/s

The total number of ICMP messages which this interface attempted to send per second [ipv6IfIcmpOutMsgs].

### iech6/s

The number of ICMP Echo (request) messages received by the interface per second [ipv6IflcmpInEchos].

### iechr6/s

The number of ICMP Echo Reply messages received by the interface per second [ipv6IflcmpInEchoReplies].

### oechr6/s

The number of ICMP Echo Reply messages sent by the interface per second [ipv6IflcmpOutEchoReplies].

### igmbq6/s

The number of ICMPv6 Group Membership Query messages received by the interface per second [ipv6IflcmpInGroupMembQueries].

### igmbr6/s

The number of ICMPv6 Group Membership Response messages received by the interface per second [ipv6IflcmpInGroupMembResponses].

### ogmbr6/s

The number of ICMPv6 Group Membership Response messages sent per second [ipv6IflcmpOutGroupMembResponses].

### igmbrd6/s

The number of ICMPv6 Group Membership Reduction messages received by the interface per second [ipv6IflcmpInGroupMembReductions].

### ogmbrd6/s

The number of ICMPv6 Group Membership Reduction messages sent per second [ipv6IflcmpOutGroupMembReductions].

### irtsol6/s

The number of ICMP Router Solicit messages received by the interface per second [ipv6IflcmpInRouterSolicits].

### ortsol6/s

The number of ICMP Router Solicitation messages sent by the



interface per second [ipv6IflcmpOutRouterSolicits].

irtad6/s

The number of ICMP Router Advertisement messages received by the interface per second [ipv6IflcmpInRouterAdvertisements].

inbsol6/s

The number of ICMP Neighbor Solicit messages received by the interface per second [ipv6IflcmpInNeighborSolicits].

onbsol6/s

The number of ICMP Neighbor Solicitation messages sent by the interface per second [ipv6IflcmpOutNeighborSolicits].

inbad6/s

The number of ICMP Neighbor Advertisement messages received by the interface per second [ipv6IflcmpInNeighborAdvertisements].

onbad6/s

The number of ICMP Neighbor Advertisement messages sent by the interface per second [ipv6IflcmpOutNeighborAdvertisements].

With the **EICMP6** keyword, statistics about ICMPv6 error messages are reported. Note that ICMPv6 statistics depend on sadc option "-S IPV6" to be collected. The following values are displayed (formal SNMP names between square brackets):

ierr6/s

The number of ICMP messages per second which the interface received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.) [ipv6IflcmpInErrors]

idtunr6/s

The number of ICMP Destination Unreachable messages received by the interface per second [ipv6IflcmpInDestUnreachs].

odtunr6/s

The number of ICMP Destination Unreachable messages sent by the interface per second [ipv6IflcmpOutDestUnreachs].

itmex6/s

The number of ICMP Time Exceeded messages received by the interface per second [ipv6IflcmpInTimeExcds].

otmex6/s

The number of ICMP Time Exceeded messages sent by the interface per second [ipv6IflcmpOutTimeExcds].

iprmpb6/s

The number of ICMP Parameter Problem messages received by the interface per second [ipv6IflcmpInParmProblems].

oprmpb6/s

The number of ICMP Parameter Problem messages sent by the interface per second [ipv6IflcmpOutParmProblems].

iredir6/s

The number of Redirect messages received by the interface per second [ipv6IflcmpInRedirects].

oredir6/s

The number of Redirect messages sent by the interface by second [ipv6IflcmpOutRedirects].

ipck2b6/s

The number of ICMP Packet Too Big messages received by the interface per second [ipv6IflcmpInPktTooBigs].

opck2b6/s

The number of ICMP Packet Too Big messages sent by the interface per second [ipv6IflcmpOutPktTooBigs].

With the **UDP6** keyword, statistics about UDPv6 network traffic are reported. Note that UDPv6 statistics depend on sadc option "-S IPV6" to be collected. The following values are displayed (formal SNMP names between square brackets):

idgm6/s

The total number of UDP datagrams delivered per second to UDP users [udpInDatagrams].

odgm6/s

The total number of UDP datagrams sent per second from this entity [udpOutDatagrams].

## noport6/s

The total number of received UDP datagrams per second for which there was no application at the destination port [udpNoPorts].

## idgmer6/s

The number of received UDP datagrams per second that could not be delivered for reasons other than the lack of an application at the destination port [udpInErrors].

The **ALL** keyword is equivalent to specifying all the keywords above and therefore all the network activities are reported.

## -o [ filename ]

Save the readings in the file in binary form. Each reading is in a separate record. The default value of the filename parameter is the current daily data file, the /var/log/sa/sadd file. The -o option is exclusive of the -f option. All the data available from the kernel are saved in the file (in fact, sar calls its data collector sadc with its option "-S ALL". See sadc(8) manual page).

## -P { cpu [,...] | ALL }

Report per-processor statistics for the specified processor or processors. Specifying the **ALL** keyword reports statistics for each individual processor, and globally for all processors. Note that processor 0 is the first processor.

## -p

Pretty-print device names. Use this option in conjunction with option -d. By default names are printed as dev m-n where m and n are the major and minor numbers for the device. Use of this option displays the names of the devices as they (should) appear in /dev. Name mappings are controlled by /etc/sysconfig/sysstat.ioconf.

## -q

Report queue length and load averages. The following values are displayed:

## runq-sz

Run queue length (number of tasks waiting for run time).

## plist-sz

Number of tasks in the process list.

## ldavg-1

System load average for the last minute. The load average is calculated as the average number of runnable or running tasks (R state), and the number of tasks in uninterruptible sleep (D state) over the specified interval.

## ldavg-5

System load average for the past 5 minutes.

## ldavg-15

System load average for the past 15 minutes.

## blocked

Number of tasks currently blocked, waiting for I/O to complete.

## -R

Report memory statistics. The following values are displayed:

## frmpg/s

Number of memory pages freed by the system per second. A negative value represents a number of pages allocated by the system. Note that a page has a size of 4 kiB or 8 kiB according to the machine architecture.

## bufpg/s

Number of additional memory pages used as buffers by the system per second. A negative value means fewer pages used as buffers by the system.

## campg/s

Number of additional memory pages cached by the system per second. A negative value means fewer pages in the cache.

## -r

Report memory utilization statistics. The following values are displayed:

## kbmemfree

Amount of free memory available in kilobytes.

## kbmemused

Amount of used memory in kilobytes. This does not take into account memory used by the kernel itself.

`%memused`

Percentage of used memory.

`kbbuffers`

Amount of memory used as buffers by the kernel in kilobytes.

`kbcached`

Amount of memory used to cache data by the kernel in kilobytes.

`kbcommit`

Amount of memory in kilobytes needed for current workload. This is an estimate of how much RAM/swap is needed to guarantee that there never is out of memory.

`%commit`

Percentage of memory needed for current workload in relation to the total amount of memory (RAM+swap). This number may be greater than 100% because the kernel usually overcommits memory.

`kbactive`

Amount of active memory in kilobytes (memory that has been used more recently and usually not reclaimed unless absolutely necessary).

`kbinact`

Amount of inactive memory in kilobytes (memory which has been less recently used. It is more eligible to be reclaimed for other purposes).

`kbdirty`

Amount of memory in kilobytes waiting to get written back to the disk.

`-S`

Report swap space utilization statistics. The following values are displayed:

`kbswpfree`

Amount of free swap space in kilobytes.

`kbswpused`

Amount of used swap space in kilobytes.

`%swpused`

Percentage of used swap space.

`kbswpcad`

Amount of cached swap memory in kilobytes. This is memory that once was swapped out, is swapped back in but still also is in the swap area (if memory is needed it doesn't need to be swapped out again because it is already in the swap area. This saves I/O).

`%swpcad`

Percentage of cached swap memory in relation to the amount of used swap space.

`-s [ hh:mm:ss ]`

Set the starting time of the data, causing the sar command to extract records time-tagged at, or following, the time specified. The default starting time is 08:00:00. Hours must be given in 24-hour format. This option can be used only when data are read from a file (option -f).

`-t`

When reading data from a daily data file, indicate that sar should display the timestamps in the original local time of the data file creator. Without this option, the sar command displays the timestamps in the user's locale time.

`-u [ ALL ]`

Report CPU utilization. The **ALL** keyword indicates that all the CPU fields should be displayed. The report may show the following fields:

`%user`

Percentage of CPU utilization that occurred while executing at the user level (application). Note that this field includes time spent running virtual processors.

`%usr`

Percentage of CPU utilization that occurred while executing at the user level (application). Note that this field does NOT include time spent running virtual processors.

`%nice`

Percentage of CPU utilization that occurred while executing at the user level with nice priority.

#### %system

Percentage of CPU utilization that occurred while executing at the system level (kernel). Note that this field includes time spent servicing hardware and software interrupts.

#### %sys

Percentage of CPU utilization that occurred while executing at the system level (kernel). Note that this field does NOT include time spent servicing hardware and software interrupts.

#### %iowait

Percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.

#### %steal

Percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor.

#### %irq

Percentage of time spent by the CPU or CPUs to service hardware interrupts.

#### %soft

Percentage of time spent by the CPU or CPUs to service software interrupts.

#### %guest

Percentage of time spent by the CPU or CPUs to run a virtual processor.

#### %gnice

Percentage of time spent by the CPU or CPUs to run a niced guest.

#### %idle

Percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.

Note: On SMP machines a processor that does not have any activity at all (0.00 for every field) is a disabled (offline)

processor.

**-V**

Print version number then exit.

**-v**

Report status of inode, file and other kernel tables. The following values are displayed:

**dentunusd**

Number of unused cache entries in the directory cache.

**file-nr**

Number of file handles used by the system.

**inode-nr**

Number of inode handlers used by the system.

**pty-nr**

Number of pseudo-terminals used by the system.

**-W**

Report swapping statistics. The following values are displayed:

**pswpin/s**

Total number of swap pages the system brought in per second.

**pswpout/s**

Total number of swap pages the system brought out per second.

**-w**

Report task creation and system switching activity.

**proc/s**

Total number of tasks created per second.

**cswch/s**

Total number of context switches per second.

**-y**



Report TTY devices activity. The following values are displayed:

`rcvin/s`

Number of receive interrupts per second for current serial line. Serial line number is given in the TTY column.

`xmtin/s`

Number of transmit interrupts per second for current serial line.

`framerr/s`

Number of frame errors per second for current serial line.

`prtyerr/s`

Number of parity errors per second for current serial line.

`brk/s`

Number of breaks per second for current serial line.

`ovrun/s`

Number of overrun errors per second for current serial line.

## ENVIRONMENT

The `sar` command takes into account the following environment variables:

`S_TIME_FORMAT`

If this variable exists and its value is **ISO** then the current locale will be ignored when printing the date in the report header. The `sar` command will use the ISO 8601 format (YYYY-MM-DD) instead.

`S_TIME_DEF_TIME`

If this variable exists and its value is **UTC** then `sar` will save its data in UTC time (data will still be displayed in local time). `sar` will also use UTC time instead of local time to determine the current daily data file located in the `/var/log/sa` directory.

## EXAMPLES

```
sar -u 2 5
```

Report CPU utilization for each 2 seconds. 5 lines are displayed.

```
sar -I 14 -o int14.file 2 10
```

Report statistics on IRQ 14 for each 2 seconds. 10 lines are displayed. Data are stored in a file called int14.file.

```
sar -r -n DEV -f /var/log/sa/sa16
```

Display memory and network statistics saved in daily data file 'sa16'.

```
sar -A
```

Display all the statistics saved in current daily data file.

## BUGS

/proc filesystem must be mounted for the sar command to work. All the statistics are not necessarily available, depending on the kernel version used. **sar** assumes that you are using at least a 2.6 kernel.

## FILES

/var/log/sa/sadd

Indicate the daily data file, where the dd parameter is a number representing the day of the month.

/proc contains various files with system statistics.

## AUTHOR

Sebastien Godard (sysstat <at> orange.fr)

## SEE ALSO

[sadc\(8\)](#), [sa1\(8\)](#), [sa2\(8\)](#), [sadf\(1\)](#), [mpstat\(1\)](#), [iostat\(1\)](#), [pidstat\(1\)](#), [vmstat\(8\)](#)

<http://pagesperso-orange.fr/sebastien.godard/>