

ifstat Command

Purpose

ifstat - Displays rolling statistics on interface utilization.

Syntax

ifstat [options] [[ALL] | <if> ...] [<interval_time> [<iterations>]]

Parameters

Interface list

A list of interfaces to monitor. A list of valid interface names can be retrieved with the **-I** option. The pseudo interface **ALL** can be specified although it is implied if no interfaces are listed.

Interval time

The time between samplings (in seconds)

Iterations

How many iterations to run before stopping

Options**-a**

Auto-factor bytes to appropriate level. Byte values greater than 10000 Bytes will be auto factored to 10 KBytes, greater than 10000 K will be factored to 10 M, etc. The resolution is set at 10000 so that accuracy is not sacrificed for brevity. The **-a** switch is incompatible with **-f**, **-k**, **-m**, and **-g**.

The **-k**, **-m**, and **-g** options are more consistent than **-a** in that they always apply factoring rules while **-a** factors each value separately. Explicitly specifying a factoring value of **-k**, **-m**, or **-g** may mean that some non-zero values may factor to 0.

-e

Summarize data at end of run. When the user hits Ctrl-C or the iteration period ends **ifstat** will print out totals, averages, minimum and maximum values over the period that **ifstat** has been running.

-f

Suppress the factor character value (B, K, M, G) for Byte data. This **-t** and the **-H** options may be useful when capturing the output to a CSV file. Because the output from the **-a** option varies by each data point, this option is incompatible with **-a**.

-g

Factor all byte data to GBytes. The data will be followed by the letter G to denote that it is in GBytes unless the **-f** option is used.

-h

Show help (and exit). Any other switches will be ignored.

-H

The default is to show descriptive column headers in output. The **-H** option suppresses this behavior. This option is useful when capturing output to a CSV format.

-k

Factor all byte data to KBytes. The data will be followed by the letter K to denote that it is in KBytes unless the **-f** option is used.

-I

List interfaces known by the performance API. Additional **-I** switches can be stacked (**-II**) to provide additional detail on the interfaces such as the interface MTU and bitrate.

-m

Factor all byte data to MBytes. The data will be followed by the letter M to denote that it is in MBytes unless the **-f** option is used.

-p

By default headers will be displayed each iteration. This changes the display of headers to each

page of results rather than each iteration of results. This option is incompatible with the **-H** option.

- t** The default output will show a totals summary for all interfaces each iteration. This option will suppress that output.
- T** Show only interface totals and suppress individual interface output.
- v** Show Version information (and exit)
- x** Show extended stats as described in the **Analytics** section of this page. Multiple **-x** switches can be stacked (**-xxx**) for additional stats.

Analytics

Additional derived statistics can be displayed using (multiples of) the **-x** switch. These include percentage of bitrate and percentage of MTU.

- x**
 - %BRX - Percentage of Bitrate RX
 - %BTX - Percentage of Bitrate TX
- xx**
 - RXErr - RX Errors
 - TXErr - TX Errors
 - Coll - Collisions
- xxx**
 - ABPRX - Average Bytes Per Packet RX
 - %M_RX - ABPRX as % of MTU
 - ABPTX - Average Bytes Per Packet TX
 - %M_TX - ABPTX as % of MTU

Notes on sampling and averaging

The statistics gathered over an interval are averaged to a one second time period. The two results of this are a smoothing of the data when using larger intervals and data averaged to nonsensical values in low traffic times (such as X Bytes sent in 0 packets)

It is important to realize the effects of intervals and averaging and use them accordingly. Generally speaking, larger intervals tend to reduce the effects of traffic spikes and valleys. As a result, the max and min values in large intervals are not well representative of the actual maximum and minimum values.

A shorter interval will provide more accurate maximum and minimum values but will produce a greater amount of data to sift through.

One option is to capture data on smaller intervals, and display the averaged data on larger intervals. As of this writing **ifstat** does not provide this capability.

Examples

1. Display basic auto-factored statistics every two seconds with summary results on exit:

```
ifstat -ae 2
```

2. Display extended auto-factored statistics for only en0, every two seconds:

```
ifstat -xxxat en0 2
```

3. Create space-delimited output as input for a graphing or analytical application:

```
ifstat -Htkf en0 1 > ifstat_out.csv
```

Notes

Statistics for **ifstat** are provided by the **libperfstat** API.

Location

/usr/bin/ifstat

Related Information

The **netstat**, **topas**, and **entstat** commands and the **libperfstat** API.