

NAME

bmon – Portable bandwidth monitor and rate estimator

SYNOPSIS

```
bmon [ -awShV ] [ -i <mod> ] [ -o <mod> ] [ -I <mod> ] [ -O <mod> ]
      [ -f <path> ] [ -p <policy> ] [ -r <float> ] [ -s <float> ]
```

DESCRIPTION

bmon is a portable bandwidth monitor with multiple input methods and output modes. A set of architecture specific input modules provide the core with the list of interfaces and their counters. The core stores this counters and provides rate estimation including a history over the last 60 seconds, minutes, hours and days to the output modules which output them according to the configuration.

The set of counters is dependant on the input module and may vary. Secondary input and output modules may be used to collect counter values from other nodes or to write HTML statistics. This input/output architecture minimizes the work needed to port it to other architectures or generate specific statistics.

OPTIONS

- i** Set primary *input module* and its configuration. The argument "list" will result in a list of available input modules. See INPUT MODULES for more details.
- o** Set primary *output module* and its configuration. The argument "list" will result in a list of available output modules. See OUTPUT MODULES for more details.
- I** Set secondary *input modules* and their configuration. The argument "list" will result in a list of available secondary input modules.
- O** Set secondary *output modules* and their configuration. The argument "list" will result in a list of available secondary output modules.
- f** Set alternative configuration *path*.
- p** Set interface acceptance policy. See INTERFACE SELECTION for more details.
- a** Include interfaces even if their status is down.
- r** Set reading *interval* in which the input module will be called. The default for this is one second. Should be less or equal than 1 or a factor of it. Values not equal to 1 will result in additional rate calculation with the read interval as its unit.
- s** Set sleeping *interval* between calls to output short interval callbacks for interactive output modules. Changing this can affect the variance of read intervals.
- w** Enable signal driven output intervals. The output module will only be invoked upon receiving of SIGUSR1. Use bmon -S - to send the signal to a running bmon instance in signal driven mode.
- S** Send SIGUSR1 to a running bmon instance. This argument takes either - which will result in invoking ps to find bmon instances or a *pid* directly.
- h** Prints a help text and exits.
- V** Prints the version and exits.

INPUT MODULES

Input modules provide the core with interface statistics. Two kinds of modules exist, primary and secondary input modules. Their main difference is usage, there may be only one primary module running at the same time while the number of secondary input modules is not limited.

Every input module has a description, help text and list of options available which can be seen by adding the option "help" to the module options:

bmon -i netlink:help

See MODULE CONFIGURATION for mode details.

PRIMARY INPUT MODULES

netlink (Linux)

Requires libnl and uses an rtnetlink to collect interface statistics. This input module also provides statistics about traffic control qdiscs and classes. It is the preferred input module on Linux.

kstat (SunOS)

Provides interface statistics on SunOS operating systems in form of 32bit and 64bit counters. It is the preferred input module on SunOS.

sysctl (BSD, Darwin)

Provides interface statistics on BSD and Darwin operating systems. Is is the preferred input module on any BSD alike system.

proc (Linux)

Provides interface statistics on Linux using the proc filesystem (/proc/net/dev). It is one of the fallback input modules on Linux and will work on nearly every Linux kernel version.

sysfs (Linux)

Provides interface statistics on Linux using the sys filesystem (/sys/class/net/). It may be used together with newer Linux kernel versions but has no real advantage over the netlink input module. It caches open file descriptors to speed it up and is used as fallback method.

netstat (POSIX)

Provides limited interface statistics on almost any POSIX operating system by invoking netstat -i -a. Only use this as last hope.

dummy (any)

The purpose of the dummy input module is for testing. It generates in either a static or randomized form.

nulll (any)

Does not provide any interface statistics and thus can be used to disable local interface collection.

SECONDARY INPUT MODULES

distribution

Collects interface statistics from other nodes. It is the counterpart of the secondary output module called distribution. Its purpose is to distribute statistics in real time with not too much bandwidth consumption itself. See DISTRIBUTION for more details.

OUTPUT MODULES

Output modules are feeded with rate estimations and graphs from the core and print them out to the configured output device. Two kinds of modules exist, primary and secondary output modules. Their main difference is usage, there may be only one primary module running at the same time while the number of secondary output modules is not limited.

Every output module has a description, help text and list of options available which can be seen by adding the option "help" to the module options:

```
bmon -o ascii:help
```

See MODULE CONFIGURATION for mode details.

PRIMARY OUTPUT MODULES

ascii The ascii output modules prints out the diagrams and lists to standard output. The output format is highly configurable and suits as vmstat alike tool for interface statistics.

curses Interactive curses user interface providing real time rate estimations and graphs. The default view is a list of all interfaces grouped per node. The graphical history diagram and a list of detailed counters may be enabled/disable during runtime. Press '?' while the UI is running to see the quick reference.

SECONDARY OUTPUT MODULES

html Writes all interface statistics and diagrams as HTML files including a navigation menu for all nodes and their interfaces. The layout can be easily changes by altering the stylesheet which will not be overwritten.

distribution (any)

Distributes all statistics over the network using an UDP based statistic distribution protocol. The default configuration will use the multicast address all-nodes but it may also be configured so send to a unicast address.

MODULE CONFIGURATION

ARGUMENT ::= modulename:OPTS[,modulename:OPTS[,...]]

OPTS ::= OPTION[;OPTION[;...]]

OPTION ::= type[=value]

If you specify multiple primrary input or output modules the first reported to be working module will be used.

If you specify multiple secondary input or output modules all of them will get invoked.

DISTRIBUTION

Statistic distribution is a powerful method to spread the statistic all over the network and make the available on every machine. The advantage over web based statistic overviews and multi terminal remote shell based solutions is its nearly realtime accuracy while being lightweight and not polluting the network too much. The protocol is UDP based and thus not reliable and optimized on size.

See include/bmon/distribution.h for the protocol specification.

DIAGRAM TYPES

You will find the following diagram types being used by all output modules in some form:

list A list of interfaces including their byte and packets rate (bps/pps).

graphical history diagram

A graph showing the history of a counter over the last 60 (read interval/ seconds/minutes/hours/days). The outer left column is the most recent rate while the outer right column is the most outdated. The preferred diagram to impress co-workers.

detailed

Detailed counters such as error counters or other attributes assigned to this interface. The list of attributes may vary depending on the input module and architecture of the host OS.

INTERFACE SELECTION

SELECTION ::= NAME[,NAME[,...]]

NAME ::= [!]interface

The interface name may contain the character '*' which will act as a wildcard and represents any number of any character type, i.e. eth*, h*0, ...

Examples:

lo,eth0,eth1

eth*,!eth0

CONFIGURATION FILE

Bmon will try and read configuration data from the following files in the specified order: /etc/bmon.conf, \$HOME/.bmonrc.

None of the above files will be read if the path to the configuration file was specified using the -f option.

Configuration possibilities:

input <module configuration>

Specify primary input module (-i), see INPUT MODULES.

secondary_input <module configuration>

Specify secondary input modules (-I), see INPUT MODULES.

output <module configuration>

Specify primary output module (-o), see OUTPUT MODULES.

secondary_output <module configuration>

Specify secondary output modules (-O), see OUTPUT MODULES.

policy <policy>

Set interface acceptance policy (-p), see INTERFACE SELECTION.

read_interval <interval>

Set reading interval in which the input module will be called (-r).

sleep_time <interval>

Set sleeping interval between calls to output short interval callbacks for interactive output modules. (-s)

show_all

Include interface even if their status is down. (-a)

include <file>

Include *file* and read it as configuration file.

Color layouts

See COLOR LAYOUTS.

Bindings

See BIND INTERFACE.

COLOR LAYOUTS

The layout is used to specify the look'n'feel of the curses output module. The color "default" represents the terminal color which can be used to keep the background transparent for transparent terminals.

Colors: default, black, red, green, yellow, blue, magenta, cyan, white

Flags: reverse

Layouts: Default, Statusbar, Header, List, Selected,

Prototype:

Layout <name> <foreground> <background> <flags>

Example:

Layout Statusbar red black reverse

Feel free to submit patches extending the configurability using layouts.

BIND INTERFACE

The bind interface can be used to bind not yet assigned keys to shell scripts. It currently works in the curses output module but it might be ported to other output modules in the future. The interface name of the currently selected interface is provided to the script via the first argument.

Prototype:

Bind <key> <Executable>

Example:

bind D /sbin/intf_down.sh

EXAMPLES

To run bmon in curses mode monitoring the interfaces eth0 and eth1:

bmon -i eth0,eth1 -o curses

To run bmon in acii mode printing the detailed diagram with fixed y-axis unit:

bmon -o 'ascii:diagram=detailed;yunit=kb'

To run bmon in signal driven mode drawing the graphical diagram with customized drawing characters and fixed x and y axis:

bmon -s -o 'ascii:diagram=graph;fgchar=#;bgcar=_;xunit=min'

To run bmon with no primary output (daemon) but distribute the statistic over the network:

bmon -o null -O distribution

To run bmon collecting local and remote statistics and show it in curses mode:

bmon -I distribution:multicast -o curses

To build a relay and collect remote statistic and send them to a unicast address while ignoring errors because the destination is not available:

bmon -i null -I distribution:multicast -o null -O 'distribution:ip=10.0.0.1;errignore;forward'

To collect local statistics and those from the whole network and generate a HTML page out of the those statistics:

bmon -I distribution:multicast -o null -O html:path=/var/istats/

KNOWN ISSUES

The curses output modules doesn't work properly on NetBSD < 2.0 because getch() cannot be set to be non-blocking.

sysctl input segfaults on sparc64 OpenBSD.

FILES

/etc/bmon.conf
\$HOME/.bmonrc

SEE ALSO

ifconfig(8), kstat(1M), netlink(3)

AUTHOR

Thomas Graf <tgraf@suug.ch>