

Network Monitoring on the Console

BY BENEDICT REUSCHLING

This column covers ports and packages for FreeBSD that are useful in some way, peculiar, or otherwise good to know about. Ports extend the base OS functionality and make sure you get something done or, simply, put a smile on your face. Come along for the ride, maybe you'll find something new.

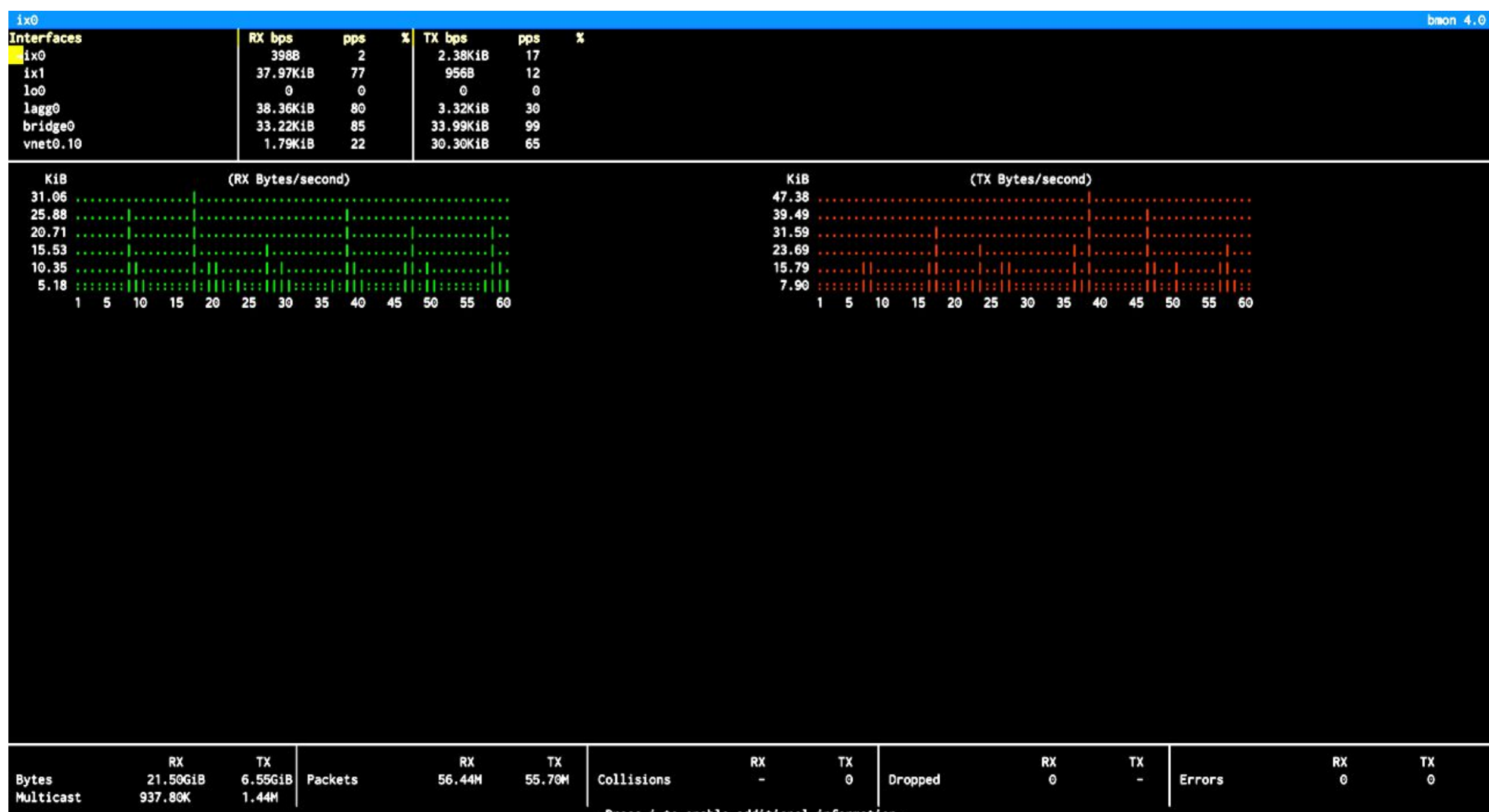
I've been doing a lot of network cabling on our big data cluster. Checking links and making sure that RX and TX lights are blinking are just a couple aspects of it. Once a link has been set up and documented, cables neatly tucked in the rack, IP addresses assigned, etc., it is time to see if the box allows the packets to flow in the right direction. There are a surprising number of tools available to help you with this. I usually breathe a sigh of relief that basic network diagnostic tools are still part of FreeBSD's base system. After all, how are people supposed to diagnose network problems if they have to install a package for that first—and from the network that is not working? Hey, who let the chickens loose in the server room? Look at all the eggs!

Be that as it may, FreeBSD contains `ping(8)` (which does both IPv4 and v6 in one as of late [<https://reviews.freebsd.org/rS368045>]) for basic ICMP checking. If you do get a link and want to see some network activity, `systat(1)` has a handful of utilities available for TCP, UDP and interface statistics monitoring. Of course, you could get really fancy and let `www/grafana` paint the most beautiful graphs on your 4k display with a bit of effort. Nothing against it, just that a lot of us prefer something in between ping and all the bells and whistles of a fully-fledged browser application. Thanks to `ncurses` and friends, we don't have to give up "graphics" because we chose to stay in the terminal (a.k.a. that black and white text UI).

There are a surprising number of tools available to help you

Let's beef up our ping output with some bar graphs to check long term trends. From a C library called liboping (octo's ping) stems the noping utility (`net/liboping`). Once installed, you ping a target IP and you'll see your familiar packets and sequence numbers rolling up the screen. At the bottom though, there is a trend over time showing any loss in packets. Of course, when I tried to generate a screenshot for you, not a single packet would get lost. So, I refer you to [<https://noping.cc/>] for an example.

How about something for TCP tailored towards the humans among us? You'll certainly find `net/bmon` appealing. I couldn't help myself and included an image of `bmon`'s output from one of my (not so busy) boxes. As a bandwidth monitor and rate estimator—as the description tells us—it definitely produces a nice output for that one computer screen in your office that makes you look busy. I expect to see the ASCII art for the bar graphs in a future computer-focused blockbuster movie.



As an aside, piping your “`zfs list -o space`” output to `misc/nms` gives you a 1992 Sneakers movie feeling. It's exactly the right thing to do when see you a snooping colleague approaching to glance at your screen. Impress that person by unscrambling your screen output with a single press of a button. Repeat it for other commands with a lot of text output.

If you prefer a display like `top(1)`, take a look at `net-mgmt/tcpttrack`. It captures packets via `/dev/bpf` (yes, root-permissions only) on any kind of device posing as a NIC connected to your system. Once a connection has been established, source, destination, port, state information, bandwidth usage, and zodiac sign (OK, that last one was slightly exaggerated) of the packets are shown. If you like that, then you should allow `net-mgmt/iftop` to occupy some storage space in your system. The three-column output tells you exactly who your computer talks to all day long.

But who in the name of all the networking gods (the old ones and the new) is using up all that bandwidth? I have several suspects in the form of processes, so I let `net/netlogs` analyze

that for me. Instead of many nettop-like tools, it gives me bandwidth by process ID. I had always suspected that running services as root was a bad idea. To a jail(8) with you, bad process!

I sometimes look with envy at software that is not yet ported to FreeBSD. For example, speedometer [<https://excess.org/speedometer/>] would be nice to have. Maybe by the time this column appears, some busy ports committer or contributor will have already ported it. That would really make my day—do I hear a “challenge accepted” somewhere? After all, this column would probably not exist if there weren’t people out there working hard to make sure FreeBSD has a good, third-party software ecosystem. And I believe we take that for granted sometimes. So, to all the ports people, a big thank you for your tireless efforts in keeping things up(dated) and running!

The next time you install a tool you like, why not drop the maintainer a small thank you to brighten their day? You can find them on freshports.org (huge thanks to Dan Langille for that site!) or in the Makefile for the port itself. Unless it is ports@freebsd.org, then the port is up for you to give it some love. And when you do, you can teach others (including me) how to do that, because there can never be enough port maintainers. Check out the WantedPorts page on FreeBSD’s wiki [<https://wiki.freebsd.org/WantedPorts>] for more ports that could be included in the Ports Collection.

If you don’t mind more colors in your network output, then check out `sysutils/glances`. It’s as if `sysutils/htop` and `vmstat` had a love affair and `glances` is the result. There is even disk activity in there, in addition to the `top(1)`-like information in almost every corner of the screen. But of the many `top(1)` clones out there, only FreeBSD’s lists my ZFS ARC statistics at a glance, thanks to Allan Jude’s addition to it. I come back to it often, even with all the other applications the ports collection provides.

As a topic, `top(1)`-like ports could fill a column of its own. And it will, so stay tuned for more in a future column. If you know a great tool that should be included here, send me an email at bcr@freebsd.org.

To all the ports people,
a big thank you for your
tireless efforts in keeping
things up(dated) and
running!

BENEDICT REUSCHLING is a documentation committer in the FreeBSD project and member of the documentation engineering team. He serves on the board of directors of the FreeBSD Foundation as vice president. In the past, he served on the FreeBSD core team for two terms. He administers a big data cluster at the University of Applied Sciences, Darmstadt, Germany. He’s also teaching a course “Unix for Developers” for undergraduates. Together with Allan Jude, he is host of the weekly `bsdnow.tv` podcast.