The sys admin's daily grind: Pktstat

Traffic Spotting

When it comes to daily tasks such as monitoring network traffic, administrators should choose a tool that is sufficiently functional and not too complicated. *By Charly Kühnast*

Today, I'm talking about a task that isn't exactly a big thrill for most administrators: providing human-readable statistics for traffic on a network interface. For this task, I recently discovered pktstat [1] in the course of searching for a compromise between the monosyllabic IPTraf and the verbose Wireshark. Pktstat is included by most distributions, and the source code is available online. To see the current connections on an interface, you simply type:

```
pktstat -i eth0
```

In a view that is remotely reminiscent of `top`, pktstat shows you the network activity sorted by class (ICMP, TCP, UDP, and so on). If name resolution takes too long for your liking, you can simply disable it by setting the `-n` parameter. In the case of protocols such as HTTP, FTP, and X11, pktstat outputs more information about the data transferred, such as the path and the request method for HTTP (i.e., `GET` or `POST`).

Figure 1 shows the download status for the ISO image of the future Ubuntu LTS version 12.04. You might notice that pktstat doesn't show the full names of the source and target machines – only the bit up to the first dot – to ensure readability. If you really want the whole name, you need to enable the `-F` parameter in pktstat.

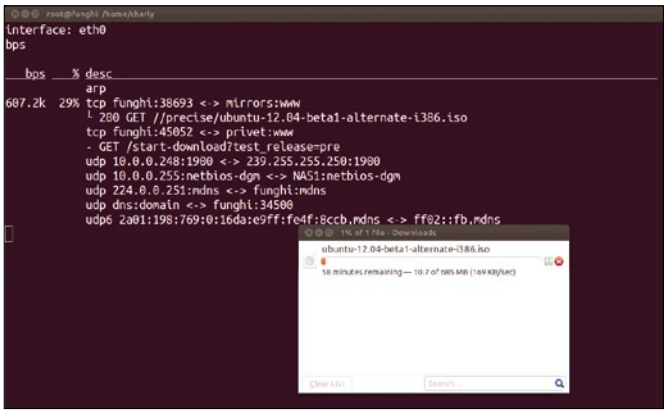

Figure 1: pktstat was bound to notice me downloading a whole ISO image. The source and target host names are deliberately curtailed.

You tend to lose visibility when things start to liven up on a network interface. To keep pace, you can resort to two tweaks. For one thing, pktstat deletes from its overview after 10 seconds those connections for which no data has been transferred. You can reduce this value to one second using the `-k` (keeptime) parameter.

Additionally, pktstat updates its overview every five seconds. Specifying `-w 1` speeds it up and refreshes the view every second. The `-w` parameter can be used in another way: pktstat offers a single-shot mode, which you enable like this:

```
pktstat -i eth0 -1 -w 10
```

The `-1` parameter initiates single-shot mode. Pktstat will now run without screen output for the number of seconds specified in `-w 10`. It then quits and leaves you a tidy overview of the connections it identified as its legacy.

Re-sorting

The tool offers some other parameters for influencing the output; the one I use most frequently is `-l` (last seen). This tells pktstat to sort the overview to show me the connections that were last active. The longer a communication is idle, the farther down the list it slides. The `-t` parameter (top mode) will push data streams that shovel the largest volume of data through the interface to the top of the list. Most command-line parameters also work interactively at pktstat run time; you can press the *L* key to enable last-seen mode in this way.

After working with pktstat for a while, I think you will agree with me that it provides administrators an uncomplicated approach to discovering the traffic situation on their networks.

For the classic question – Which process is currently grabbing all of the available bandwidth? – well, if you want to do some detective work, you still need Wireshark. ■■■

INFO

[1] Pktstat: *http://www.adaptive-enterprises.com.au/~d/software/pktstat/*

AUTHOR

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.