

vnStatSVG

-- Lightweight web front-end for network traffic monitor: vnStat

[TinyLab.org](http://tinylab.org)

2013/8/4

<http://www.tinylab.org/project/vnstatsvg/> : Homepage

<http://gitorious.org/tinylab/vnstatsvg> : Git repo

<http://tinylab.org/vnstatsvg/> : Demo site

Table of Content

vnStatSVG: Monitor network traffic of Linux host, cluster and embedded device.....	3
1. Introduction	3
2. Take a look at a demo site	4
3. Getting start with the lightweight backend: vnStat	6
3.1 Install and use vnStat on Ubuntu	6
3.2 Compile and Install vnStat from source code	8
3.3 Cross compile it for ARM platform	8
4. Install vnStatSVG for different systems	9
4.1 Install vnStatSVG for Ubuntu	9
4.2 Install vnStatSVG for a Cluster	12
4.3 Install vnStatSVG for a Busybox based embedded system.....	13
5. Conclusion	14

vnStatSVG: Monitor network traffic of Linux host, cluster and embedded device

by falcon <wuzhangjin@gmail.com> of TinyLab.org

2013/08/03 02:33

1. Introduction

vnStatSVG is a lightweight AJAX based web front-end for network traffic monitoring.

To use it, its backend vnStat: <http://humdi.net/vnstat/> must be installed at first.

It only requires a CGI-supported http server but also generates a graphical report with SVG output, Compare to its counterparts, it has such features:

- ◆ Generates graphic output dynamically with SVG and AJAX
- ◆ Not only works with Apache but also works with Busybox httpd and even any other lightweight httpd servers
- ◆ Only need CGI support, No PHP and other modules needed
- ◆ Need very little bandwidth consumption for only transfer few data in XML format, the XSL, js and CSS files only need to be transferred once
- ◆ Supports to monitor multi-interfaces (eth0, eth1...) of a single host
- ◆ Supports to monitor multi-hosts (of a cluster) in one window
- ◆ Supports multi-protocols(http, ftp, file and even ssh) to transfer data between target hosts and the main web service host
- ◆ Support different web browsers, the latest chromium and firefox have been tested
- ◆ Support multi XML data dump methods, include a shell script version, a standalone C version and a 'plugin' to the official vnStat (only 1.6 currently)

In a word, vnStatSVG is friendly to generic Linux hosts, servers, embedded Linux systems and even Linux clusters.

This project is launched by Falcon in 2008 when I was a student and it has been hosted in <http://sourceforge.net/projects/vnstatsvg/>, the previous latest version is 1.0.7, after that, I moved to other topics and stopped its update.

In the past 2 weeks, I started to reactivate this project and continue its maintaining, now, It is hosted in gitorious.org: <http://gitorious.org/tinylab> for its friendly git access, see its repository address:

- ◆ Git homepage: <http://gitorious.org/tinylab/vnstatsvg>
- ◆ Git repository: [git://gitorious.org/tinylab/vnstatsvg.git](http://gitorious.org/tinylab/vnstatsvg.git)

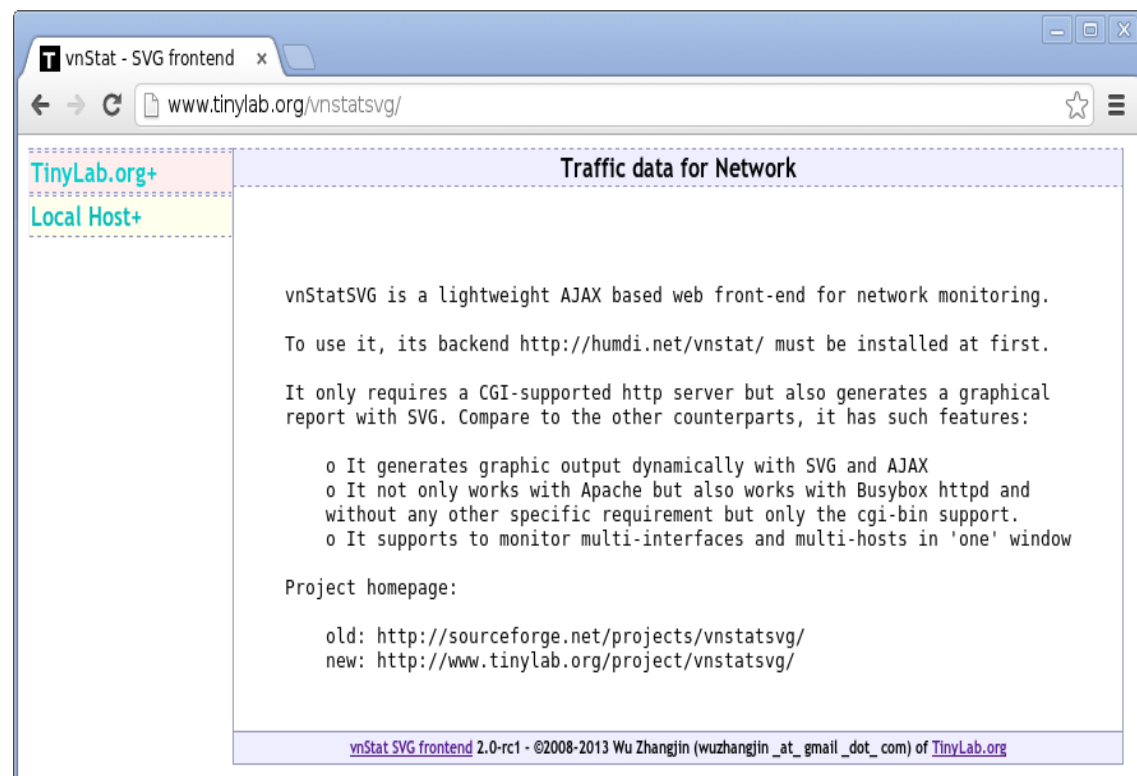
The 1st release candidate of version 2.0 is ready, it has fixed up some issues found in the early 1.0.7 and has added some new features. In the coming 2.0 release, no more new functions will be added, but new fixups will be added to make those new features become stable enough.

2. Take a look at a demo site

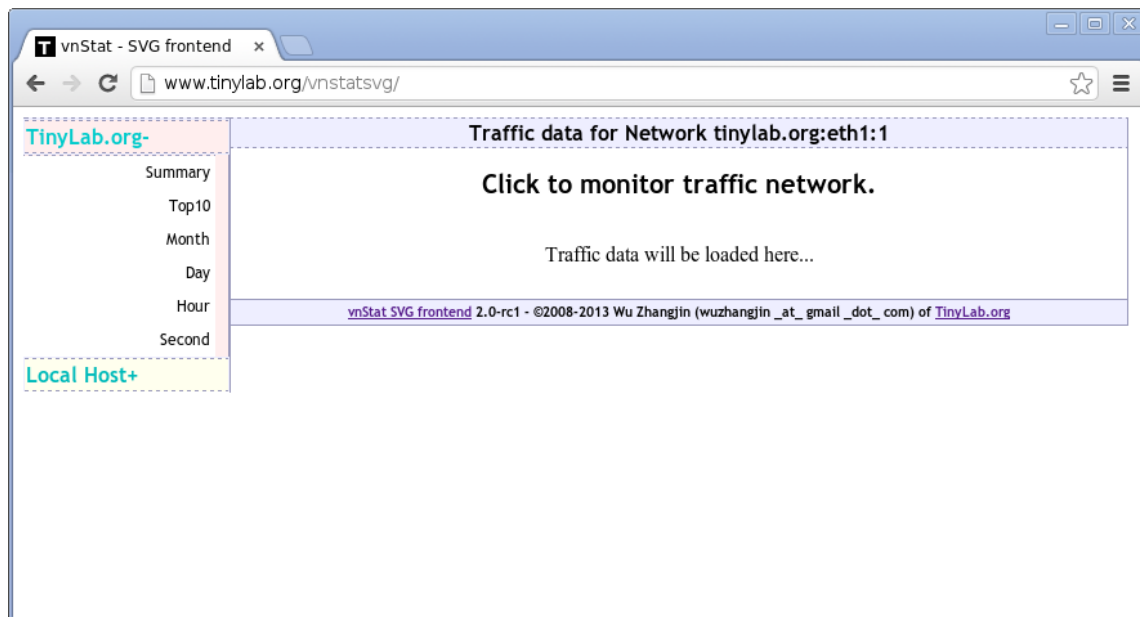
Before getting start to download, install and use vnStatSVG, Let's take a look at its demonstration site hosted on TinyLab.org:

<http://tinylab.org/vnstatsvg/>

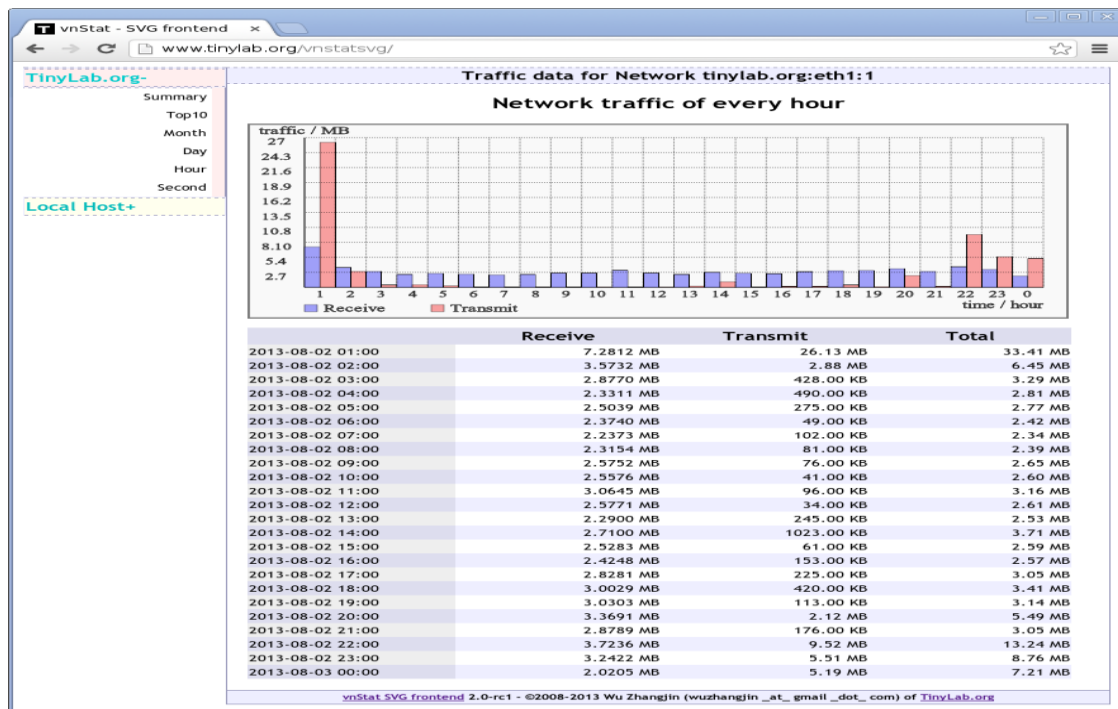
Open your chromium or firefox to access it, you should be able to get such a simple entry:



Take a look at the left sidebar, a list of hosts and the according network interfaces are present, hit any of them, you will get the hidden menu out:



Now, 6 different entries come out and the right part tells us the current monitored network host and interface, and allows us to monitor network traffic through click the entries, use 'Hour' data as an example:



The output include two parts, one is in SVG format, another is its original data listed in a table. The output style of the other entries are basically the same but with different data, the 'Second' entry is a little different, this data is not generated by the vnStat backend, but collected from the Linux network traffic statistic interface: **/proc/net/dev** directly every second, so, it is 'real time' data.

To monitor the other hosts and their interfaces, we can simply click them on the left sidebar and get the network traffic of 'Summary', 'top10 day', 'per day', 'per month', 'per hour' and 'per second'.

3. Getting start with the lightweight backend: vnStat

Take a look at its official introduction at first:

vnStat is a console-based network traffic monitor for Linux and BSD that keeps a log of network traffic for the selected interface(s). It uses the network interface statistics provided by the kernel as information source. This means that vnStat won't actually be sniffing any traffic and also ensures light use of system resources. However, in Linux at least a 2.2 series kernel is required.

In Linux, as we mentioned above, the network traffic statistic interface is **/proc/net/dev/**, its output looks below:

```
$ cat /proc/net/dev
Inter-|   Receive                               |   Transmit
face |bytes   packets errs drop fifo frame compressed multicast|bytes   packets errs
drop fifo colls carrier compressed
eth0: 180272431 154277 0 0 0 0 0 0 0 17004672 153875
0 0 0 0 0 0
lo: 1020338 8975 0 0 0 0 0 0 0 1020338 8975 0
0 0 0 0 0 0
```

As we can see, the **bytes** and **packets received** and **transmitted** of each network interfaces (eth0, lo...) have been listed.

But as we know, the files in **/proc** file system is in memory, which will be lost after system reboot, so, vnStat is written to save the data to a persistent file in a disk.

To save the data timely, vnStat should be run periodically, for example, get the data in every 5 seconds and provide data as fresh as possible, but to save cpu and power cost, the period should be not too small, 5 seconds should be enough for daily statistic.

3.1 Install and use vnStat on Ubuntu

Now, Let's install vnStat in Ubuntu at first:

```
$ sudo apt-get install vnstat
```

Ubuntu will install a daemon to execute vnStat periodically, no need to setup a cron service or write your own daemon again:

```
$ ps -ef | grep vnstat
root      1371      1  0 Aug02 ?          00:00:00 /usr/sbin/vnstatd -d
```

Now, let's take a look at how to use it, firstly, overview its options:

```
$ vnstat --help
vnStat 1.11 by Teemu Toivola
-q, --query          query database
-h, --hours          show hours
-d, --days          show days
-m, --months         show months
-w, --weeks          show weeks
```

```

-t, --top10          show top10
-s, --short          use short output
-u, --update         update database
-i, --iface          select interface (default: eth0)
-?, --help           short help
-v, --version        show version
-tr, --traffic       calculate traffic
-ru, --rateunit      swap configured rate unit
-l, --live           show transfer rate in real time

```

See also '`--longhelp`' for complete options list and '`man vnstat`'.

And show its daily traffic with the **-t** option:

```
$ vnstat -t
eth0 / top 10
```

#	day	rx	tx	total	avg. rate
1	07/27/13	1.09 GiB	1.28 GiB	2.38 GiB	230.62 kbit/s
2	07/07/13	1.00 GiB	179.97 MiB	1.17 GiB	113.84 kbit/s
3	06/29/13	636.57 MiB	246.67 MiB	883.24 MiB	83.74 kbit/s
4	07/16/13	792.17 MiB	35.14 MiB	827.31 MiB	78.44 kbit/s
5	07/21/13	679.28 MiB	97.92 MiB	777.21 MiB	73.69 kbit/s
6	07/20/13	421.73 MiB	87.03 MiB	508.76 MiB	48.24 kbit/s
7	07/06/13	466.12 MiB	27.79 MiB	493.91 MiB	46.83 kbit/s
8	07/24/13	394.07 MiB	57.00 MiB	451.07 MiB	42.77 kbit/s
9	07/30/13	386.93 MiB	38.82 MiB	425.74 MiB	40.37 kbit/s
10	06/30/13	374.99 MiB	28.08 MiB	403.07 MiB	38.22 kbit/s

Without the **-i** option, the default interface is '**eth0**'. To update the data and store it to the database immediately, use the **-u** option:

```
$ vnstat -u -i eth0
```

To learn more options, please use **--longhelp** option or access its homepage: <http://humdi.net/vnstat/>

The default database used by vnStat is under **/var/lib/vnstat/**, the database are named with the interface name. the database format is specifically designed by the author of vnStat, it can be parsed by vnStat itself, so, no extra database tool required. The default configure file is installed to **/etc/vnstat.conf**, you can configure the default interface and database storing directory there.

3.2 Compile and Install vnStat from source code

To use vnStat for a platform without prebuilt packages, especially for a new created embedded system, we may need to compile and even cross compile vnStat from source code and configure it ourselves, Herein will discuss it.

Download the latest version from its homepage: <http://humdi.net/vnstat/> and decompress it:

```
$ wget -c http://humdi.net/vnstat/vnstat-1.11.tar.gz
$ tar xzf vnstat-1.11.tar.gz
```

And now, compile it and install it:

```
$ make ; make install
```

Then, let **cron** service update the database periodically with the default cron config

file: **examples/vnstat.cron**, copy it to **/etc/cron.d/** or use '**crontab -e**' command to edit it. Herein, will introduce a simple daemon written in shell script which works like cron but without its dependency, which allows us to update on the embedded system without the installation of a cron service, see below:

```
#!/bin/sh
# vnstat-update.sh -- a simple daemon to update database of vnstat in specified period
VNSTAT=/usr/bin/vnstat
IFACE=eth0
PERIOD=5

while ;;
do
    $VNSTAT -u -i $IFACE
    sleep $PERIOD
done
```

Save the above script to **/usr/bin** and append a simple line at the end of your system's **/etc/rc.local** (or similar script file will run during boot) may let it work.

```
/usr/bin/vnstat-update.sh &
```

3.3 Cross compile it for ARM platform

vnStat itself support cross compiling, but no direct method, herein, let's discuss how to compile it for ARM platform as an example.

Firstly, let's install a cross compiler, in Ubuntu 12.10 (or newer) host, we can simply install it:

```
$ sudo apt-get install gcc-arm-linux-gnueabi
```

On other Linux systems which don't provide a prebuilt cross compiler package, the Linaro one is recommended, see: <https://launchpad.net/linaro-toolchain-binaries>

Now, Let's cross compile vnStat, at first, should replace the **host gcc** with **cross gcc**:

```
$ sed -i -e 's/gcc/arm-linux-gnueabi-gcc/g' src/Makefile
$ make
```

If want a standalone binary, link it statically with '**-static**', Let's configure it for **CFLAGS**:

```
$ sed -i -e 's/\-02/\-static \-02/g' src/Makefile
```

Afterwards, we will get the ARM binary:

```
$ file src/vnstat
```


src/vnstat: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.31,
BuildID[sha1]=0xbb188e3e644cee77282f1d99652aae0285062607, not stripped

Then, install it to the target file system specified with **DESTDIR** environment variable, Let's assume the target embedded file system is **../target**. To install it, we should at first specify the right **cross strip** for the **install** command and then install to **../target**:

```
$ sed -i -e 's/install -s/install -s --strip-program=arm-linux-gnueabi-strip/g'  
Makefile  
$ mkdir ../target/  
$ DESTDIR=../target/ make install
```

At last, configure your cron service or start our **vnstat-update.sh** or use the just built **vnstatd** service. Please get more help from the **INSTALL** file in the root directory of vnStat's source code.

4. Install vnStatSVG for different systems

In theory, vnStatSVG is available for any Linux platforms which have installed a CGI supported web server, but we have only tested it on Ubuntu and a Busybox based embedded system, so, we will only discuss its installation on them.

4.1 Install vnStatSVG for Ubuntu

At first, Let's install the web server: apache2,

```
$ sudo apt-get install apache2
```

Then, Let's clone the latest version from its git repository:

```
$ git clone git://gitorious.org/tinylab/vnstatsvg.git  
$ cd vnstatsvg
```

And then, configure and install it:

```
$ ./configure  
$ make  
$ sudo -s  
$ make install
```

By default, files under **src/admin** will be installed to the web root: **/var/www**, and files under **src/cgi-bin** will be installed to the cgi-bin root **/usr/lib/cgi-bin**. The last section will show which files should be installed. To customize the installation targets, please get help below:

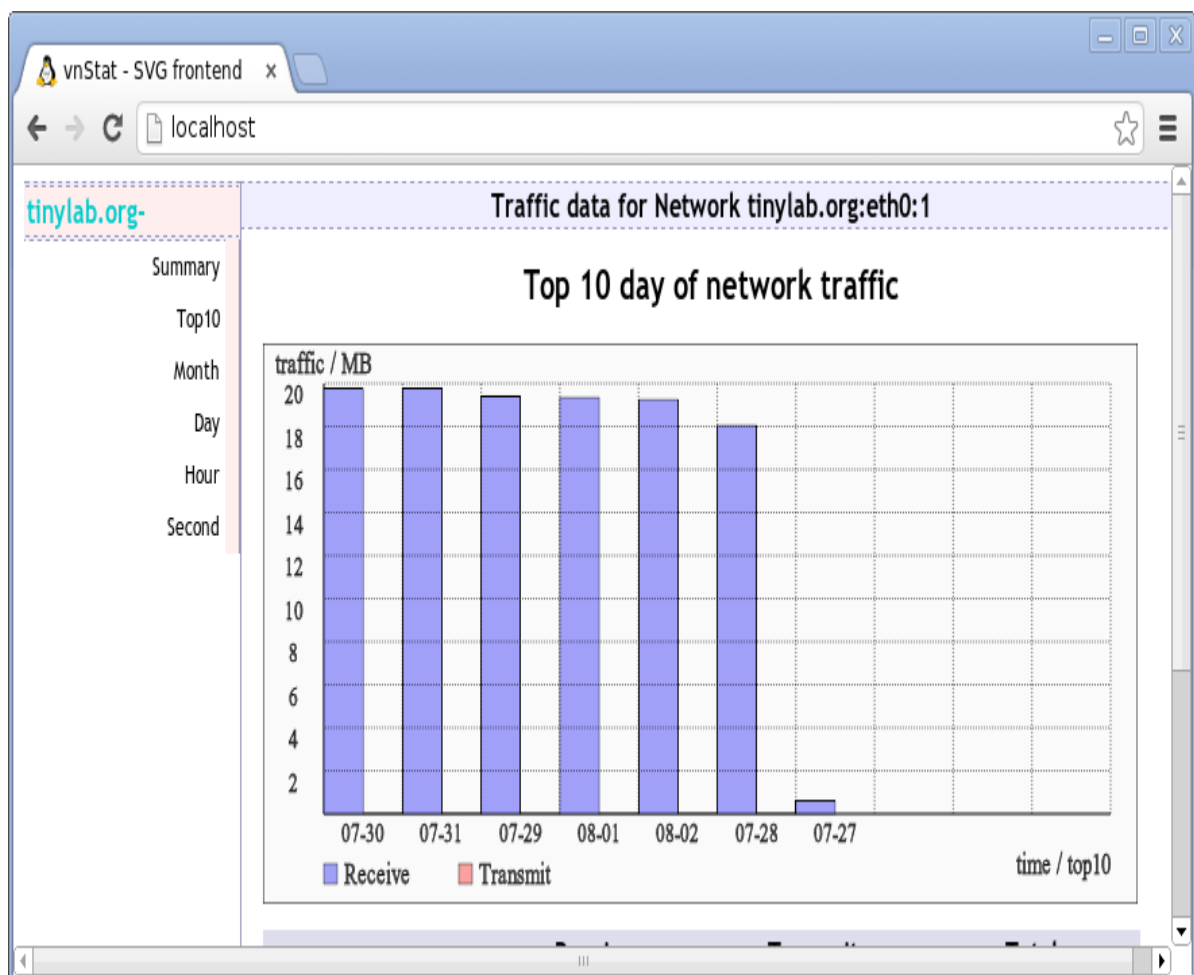
```
$ ./configure -h  
Usage:  
    $ ./configure -c /path/to/cgi-bin  
                  -w /path/to/web-root  
                  -d a directory in web-root  
                  -m [c|p|shell]  
                  -h get help
```

- c indicates the cgi-bin directory
- w indicates the web root directory
- d indicates a directory in web root to store the index page of vnstatsvg
- m indicates the method to dump XML data, there are three choices, one is using vnStatXML(c), another is using (shell) script with the --dumpdb option provided by vnStat. the third is using the --dumpxml option after patch vnStat with vnStatXML.
- h print this information

To cross compile it with static linking:

```
$ ./configure
$ make CFLAGS=' -static ' CROSS_COMPILE=arm-linux-gnueabi-
$ make install
```

Now, you should be able to access vnstatsvg web service, open <http://localhost> in your chromium or firefox browser. It will only list the default demo host and interface we have added:



To monitor your own host and interface, you can modify `/var/www/sidebar.xml`, for example, monitor **eth0** of **localhost**:

```
$ cat /var/www/sidebar.xml
<?xml version='1.0' encoding='UTF-8' standalone='no' ?>
<sidebar id="sidebar">
```

```

<iface>
    <name>eth0</name>
    <host>localhost</host>
</iface>
</sidebar>

```

If have another host and another interface eth1, you can add it as below:

```

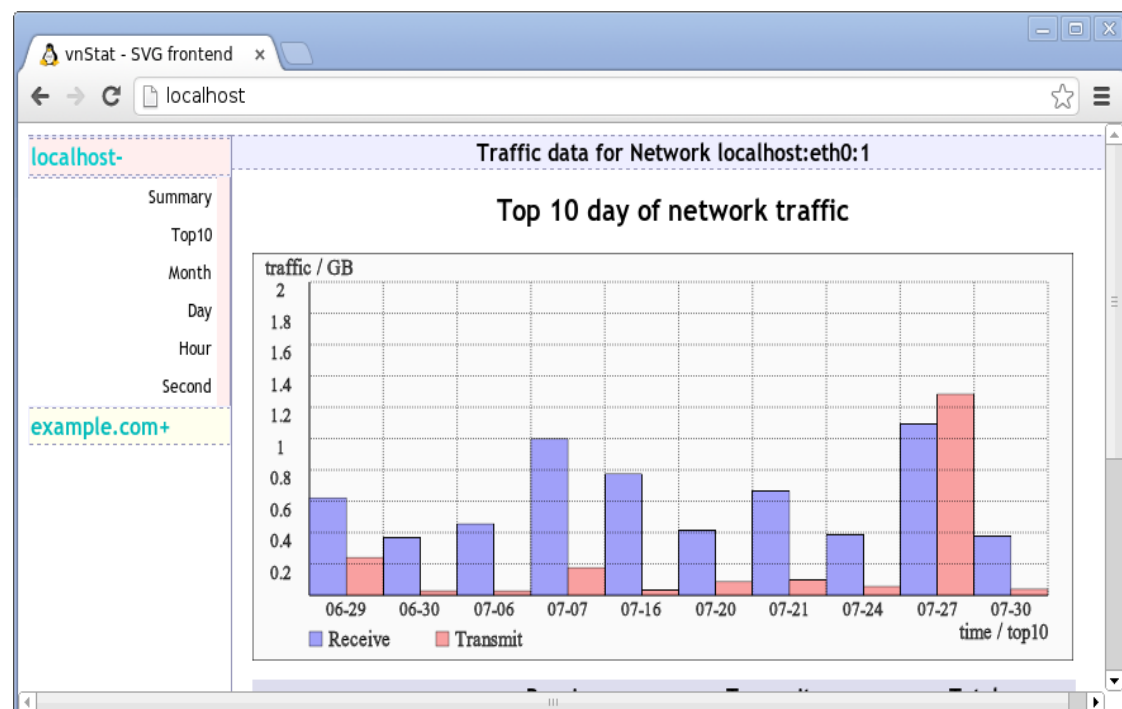
$ cat /var/www/sidebar.xml
<?xml version='1.0' encoding='UTF-8' standalone='no' ?>
<sidebar id="sidebar">
<iface>
    <name>eth0</name>
    <host>localhost</host>

</iface>
<iface>
    <name>eth1</name>
    <host>example.com</host>

</iface>
</sidebar>

```

To monitor the **example.com** host together in 'one' monitor window, you may need to install a web server there and install the **src/cgi-bin** files. No need to install **src/admin** files for vnStatSVG allows to share other files except the XML formatted network traffic data and vnStatSVG can grab the XML data from the other hosts smartly and transparently with a proxy mechanism. Now, we get such web output, just like the first demo picture we have seen above:



Currently, only 'eth0' and 'eth1' network interfaces are allowed to be monitored, to monitor other interfaces, please modify **/usr/lib/cgi-bin/vnstat.sh** yourselves. In the coming 2.0 release, I plan to provide a better solution to configure the network interfaces.

To display a detailed information of your host, use the **description** attribute the **iface** node in **/var/www/sidebar.xml**, see the examples in **src/admin/sidebar.xml**. Please get more information about installation from the **INSTALL** document.

4.2 Install vnStatSVG for a Cluster

To use vnStat for a Linux cluster and not install web server in all of the compute nodes in the cluster, you can write a daemon to sync XML data to local **/var/lib/vnstat** directory periodically and store the data with the name like **cluster_node_name-iface_name**, for example, **example.com-eth0**, then, you can configure the **sidebar.xml** as below:

```
$ cat /var/www/sidebar.xml
<?xml version='1.0' encoding='UTF-8' standalone='no' ?>
<sidebar id="sidebar">
  <iface>
    <name>eth0</name>
    <host>localhost</host>
  </iface>
  <iface>
    <name>example.com-eth0</name>
    <host>localhost</host>
    <description>example.com</description>
  </iface>
</sidebar>
```

To sync data from remote computer nodes to the web service node, we can use NFS service, SSH service, FTP service and the others, we will not demonstrate how for there are lots of tutorials in the internet. To sync data 'real time', the virtual **file** protocol should give a help. with this protocol, you can specify your own **dump_tool**, that means, you can write your own shell script to sync data and dump the XML data out with the help of our **/usr/lib/cgi-bin/vnstat.sh**. see a simple example:

```
$ cat /var/www/sidebar.xml
<?xml version='1.0' encoding='UTF-8' standalone='no' ?>
<sidebar id="sidebar">
  <iface>
    <name>eth0</name>
    <host>localhost</host>
  <protocol>file</protocol>
    <dump_tool>/usr/lib/cgi-bin/vnstat.sh</dump_tool>
    <description>Local Host</description>
  </iface>
</sidebar>
```

4.3 Install vnStatSVG for a Busybox based embedded system

The only difference here may be, we may need cross compiling here and need to do some manual configure for the lightweight web server of **Busybox: httpd**.

We have just discussed the cross compiling of vnStat and vnStatSVG above, will not rediscuss them. To continue the left parts, please install vnStat and the update script: **vnstat-update.sh** or the **vnstatd** daemon built with vnstat yourselves to get vnStat database.

To let httpd transfer the XML, XSL and XHTML files with right MIME types, the following lines must be added to the configure file of httpd, for example, **/etc/httpd.conf**:

```
.xml:text/xml
.xhtml:text/xml
.xsl:text/xml
```

Since embedded system may not use the standard **web root**, **web port** and **cgi-bin** directory, Let's assume the web root is **/data/www**, the default cgi-bin directory of httpd will be **/data/www/cgi-bin** and assume the port is **8080**. Now, Let's start the httpd service on Busybox based embedded system:

```
$ httpd -h /data/www/ -p 8080 -c /etc/httpd.conf
```

To let our vnStatSVG work, we must install the files under **src/admin** to **/data/www** and the files under **src/cgi-bin** to **/data/www/cgi-bin**. Let's refer to Makefile, the files of **ADMIN_FILES** under **src/admin** must be installed to **/data/www**, the files of **CGI_FILES** under **src/cgi-bin** must be installed to **/data/www/cgi-bin**.

```
$ grep -e '^ADMIN_FILES|^CGI_FILES' -ur Makefile
ADMIN_FILES=index.xhtml index.xsl sidebar.xml sidebar.xsl vnstat.xsl vnstat.js
vnstat.css menu.xml menu.xsl
CGI_FILES=httpclient proxy.sh vnstat.sh
```

Note, there is no real **vnstat.sh** under **src/cgi-bin**, but there are three similar files named with suffix, **-c**, **-p** and **-shell**, by default we use **-shell**. If not want to install vnStat separately, please use the **-p** method to compile vnStat and vnStatSVG together, Let's assume **../target** is our embedded system's root:

```
$ mkdir -p ../target/data/www/cgi-bin/; mkdir -p ../target/usr/bin
$ ./configure -c ../target/data/www/cgi-bin/ -w ../target/data/www/ -
b ../target/usr/bin -m p
$ make CFLAGS=' -static ' CROSS_COMPILE=arm-linux-gnueabi-
$ make install CROSS_COMPILE=arm-linux-gnueabi-
```

With the above steps, you should be able to configure **sidebar.xml** and after that, package your Busybox file system and start it on your target board. Then, after network configuration, vnStatSVG should work.

5. Conclusion

As we can see from the above exercises, with the backend vnStat, vnStatSVG is really powerful to monitor network traffic for different systems, from a single Linux host, to a Linux cluster and even for a simple Busybox based embedded system.

And it is very friendly to the resource limited embedded system which have no enough disk and memory to install apache2 with PHP modules and which have only a limited bandwidth of network connection.

Besides, vnStatSVG is configurable, scalable and easy to use.