nProbe: an Open Source NetFlow Probe for Gigabit Networks

Luca Deri
NETikos S.p.A.
Via Matteucci 34/b
56124 Pisa, Italy
deri@ntop.org, http://www.ntop.org/

Abstract
Cisco NetFlow is an industry standard protocol suitable for monitoring network traffic. Although most of high-end network routers support NetFlow, very often flows are computed only on a small portion of the overall traffic due to performance limitation of NetFlow probe implementations.

This paper covers the design and implementation of an open source software NetFlow probe designed for handling Gigabit traffic. As nProbe uses little CPU and memory, it has been successfully used to monitor high-speed networks at full wire speed without packet sampling in scenarios where commercial NetFlow probes could not be used due to their limitations. Finally, the paper shows how nProbe has been successfully integrated into an embedded computer named nBox.

Keywords
Cisco NetFlow, Passive Network Monitoring, Packet Capture, Open Source Software.

1. NetFlow Traffic Monitoring: State of the Art

Cisco NetFlow© is a widespread standard for network traffic accounting. Leading network manufacturers such as Cisco, Juniper Network and Extreme Networks provide NetFlow agents as part of their operating systems. Unfortunately most of these implementations are not able to handle more than 5-10'000 packet/sec[1] unless some specialized and costly network boards such as Cisco MSFC (Multilayer Switching Feature Card) are used.

In the software world, there are several NetFlow collectors available [1]. They range from simple "capture and store the flow into the database" collectors written in Perl, to complex applications such as cFlowd and FlowScan. As probes are usually embedded in hardware, beside some rare exceptions [2] most the NetFlow software available has been designed for the collector side. The main consequence is that NetFlow is not very spread because:

- Only high-end routers support NetFlow or support it but are not really adequate for enabling the probe at reasonable speed.
- NetFlow-aware routers are relatively expensive and require some expertise for their setup.
- Sometimes NetFlow probes need to be installed on a computer (e.g. on the firewall) as network administrators have no access to the router that is often provided by an ISP.
- Enabling NetFlow on a router often slows down router's performance on high traffic networks.
- Most NetFlow probe implementations perform poorly hence packet sampling techniques need to be used.

On the collector side the situation is better as there is a plethora of applications available. Unfortunately most of the collectors have been designed for network experts so that very few network administrators really use

[1] High-end routers manufactured by Juniper Networks are able to handle million of packets/sec although can handle up to 7'000 packets/sec for the purpose of NetFlow traffic measurement.

NetFlow. The consequence of all this, is that most of the people still monitor their networks using SNMP MIB-II interface counters [3] and MRTG [4].

The author decided to fill this gap by implementing a software NetFlow probe named nProbe able to overcome the limitation of commercial router-based probes. In addition he extended ntop[5], a traffic monitoring application previously developed by the author, adding NetFlow support in order to provide a complete open source solution, both probe and collector, to traffic measurement using NetFlow. The following chapters describe the design and the implementation of nProbe and show how nProbe plus ntop can be effectively used to monitor networks.

2. nProbe Architecture

nProbe, acronym for NetFlow probe, is an open source NetFlow v5 probe. The application captures packets flowing on a Ethernet segment, computes NetFlow flows, and export them to the specified collectors. Users can fully control flows parameters (e.g. flow expire time) as well as flow collectors. Exported flows can be collected using commercial applications such as Cisco NetFlow Collector [6], or analyzed using open source tools such as ntop and flow-tools [7].

nProbe's main features include:

- Support for NetFlow v5.
- Ability to keep up with Gbit speeds on Ethernet networks handling thousand of packets per second without packet sampling on commodity hardware.
- Support for major OS including Unix, Windows and MacOS X.
- Resource (both CPU and memory) savvy, efficient, designed for environments with limited resources.
- Source code available under the GNU GPL license.

nProbe has been designed to be small, efficient and easy to embed in hardware. For this reason the application design is very simple. Packets are captured from the network using libpcap [8], a portable packet capture library, decoded and stored on a hash. Each hash bucket contains information about a flow (e.g. total number of packets, flow duration) in order to maintain flow state. Two threads are concurrently accessing the hash:

- The first thread captures network packets and updates hash buckets.
- The second thread periodically (e.g. every minute) walks the table looking for expired flows.

Whenever an expired flow is detected, the corresponding hash bucket is freed and the flow is emitted. In order to avoid emitting packets containing few flows, the user can specify the minimum number of flows per packet so that the probe does not sends the packet until the limit is reached.

During the application design, the author has carefully studied the hash performance. As buckets are frequently added and removed from the hash, a simple hash would not be enough because in this case it would be necessary to periodically rehash the entries causing packet loss as during this time the hash would be locked. For this reason the following design solution has been adopted:

1. The hash size is specified at startup in order to allocate all the memory at once. This is necessary as:
    - A dynamic hash (e.g. the one used by ntop) could prevent nProbe from running on systems (i.e. a hash extension could fail) with limited resources such as embedded systems. With this approach, if the probe can be started as there is enough memory, it cannot happen that the probe needs to quit at runtime due to lack of memory.
    - During hash resize the probe could experience packet loss.

2. Frequent calls to memory management functions (e.g. malloc, free) degrade the overall probe performance.
3. Hash indexing is more efficient with fixed-size hashes.

As every NetFlow probe, nProbe supports various aggregation facilities including port, address, AS (Autonomous System) source/peer, TOS (Type of Service), protocol. As the probe does not run on a router, the author had to provide AS information using an alternative way. At startup, the probe reads a file containing AS information that will then be used to fill flow information. In order to produce this file automatically, the probe comes with an utility that extracts the BGP table from Juniper Routers (similar tools exist on the Internet for Cisco routers), so that human intervention is not needed.

The nProbe architecture enabled the probe to handle several hundred thousand packets. The tests[2] have been performed using a Dual AMD Athlon MP 1600 CPU, 1 GB RAM, Intel Pro 1000 GE NIC, running Debian GNU/Linux 3.0, Kernel 2.4.18 self compiled with driver from Intel website and a traffic simulator able to fill a Gbit line with the following results:

Packet Size	Network Load	nProbe Performance
64 bytes packets	142 Mbit	277'340 packet/sec
64-1500 bytes (random)	953.6 Mbit	152'430 packet /sec

The tests have been performed using nProbe 1.x, whereas at the time of the writing nProbe 2.x is available and able to deliver a significantly better performance than version 1, due to a new hash management. The results demonstrated that nProbe on a Gigabit network that is at least an order of magnitude more than the number of packets handled by many NetFlow probes embedded on commercial routers (e.g. Juniper M5 series or Extreme Networks Alpine).

Furthermore, due to its minimal resource requirements, nProbe has been successfully ported on a tiny, matchbox-size, embedded PC named nBox.

Fig. 1: nBox

The nBox is based on an embedded PC produced by Cyclades [9] based on Hard Hat Linux and powered by a dual PowerPC MPC855T. The box bas been selected as it has a small form factor with no moving parts, low power consumption (it can be powered with the Ethernet cable with no need of an external power supply), and based on open source software for easy customization with no need to pay any royalty. nBox has been designed to be a cheap NetFlow probe suitable for monitoring low speed networks (e.g. leased lines or 10 Mbit Ethernet) without the need to purchased expensive NetFlow-aware network appliances. Based on an embedded Linux operating system, it is a low-cost hardware NetFlow probe, easy to configure thanks to the embedded web HTTPS server that allows users to configure both the box and nProbe. nBox emits network flows in NetFlow v5 format using a customized version of nProbe. The peculiarity of nBox is that it has no moving parts (low noise and minimized hardware failure risk) and that its enclosure can fit on the palm of a hand. Thanks to its tiny size and low price, users can place a box per network trunk and use ntop as central flow collector, or use the box as a portable probe and display traffic statistics on a display that can be connected to the serial port.

[2] The tests have been performed by Hauman echnologies Inc.

3. Validation

nProbe and nBox have been deployed since several months at the University of Pisa for monitoring the campus backbone using ntop as collector. The probe, installed on a Linux PC close to the border gateway, continuously collects Internet traffic flowing on the Gbit backbone. Tests have demonstrated that nProbe is much more efficient than ntop due to the limited number of operations that it has to perform. This is because ntop provides several per-packet/host traffic statistics whereas nProbe computes limited per-flow traffic statistics that require a portion of the ntop time. For this reason the author decided to use ntop as pure traffic collector and let nProbe collect the traffic and send ntop the flows. With this two level architecture, ntop has been able to scale at Gbit speeds while mostly[3] providing the same level of accuracy provided by the original ntop.

During the test period, several network problems including attacks, viruses, and misconfiguration have been detected. Currently the probe is successfully analyzing the traffic flowing across an experimental 2.5 Gbit Internet link.

4. Final Remarks

This paper has described the design and architecture of nProbe. Lab tests and real network traffic have proved that the probe is suitable for monitoring Gbit networks using a high-end PC based on Linux. Although ntop cannot capture traffic at high speed when used as collector for nProbe flows, ntop can successfully scale at Gbit speeds while providing almost the same level of traffic analysis accuracy. Therefore the combination of nProbe and ntop allow high speed networks to be successfully monitored at low cost using commodity hardware.

5. Availability

Both ntop and nProbe are available for Unix, MacOS X and Windows under the GNU GPL licence at http://www.ntop.org/.

6. Vitae

Luca Deri <deri@ntop.org> is currently sharing his time between NETikos S.p.A. and the University of Pisa where he has been appointed as lecturer at the CS department. He received his Ph.D. in Computer Science with a thesis on Software Components from the University of Berne in 1997. He previously worked as research scientist at the IBM Zurich Research Laboratory and as research fellow at the University College of London. His professional interests include network management and monitoring, software components and object-oriented technology. His home page is http://luca.ntop.org/.

7. References

[1] SWITCH - The Swiss Education and Research Network, *Flow Measurement Tools*, http://www.switch.ch/tf-tant/floma/software.html, 2003.
[2] S. Astashonok, *fprobe: a NetFlow Probe*, http://fprobe.sourceforge.net/.
[3] K. McCloghrie, M.T. Rose, *Management Information Base for Network management of TCP/IP-based Internets: MIB-II*, RFC 1213, March 1991.
[4] T. Oetiker, *Multi Router Traffic Grapher (MRTG*), http://www.mrtg.org/.
[5] L. Deri, R. Carbone, and S. Suin, *Monitoring Networks Using Ntop*, Proc. of IM 2001, Seattle, May 2001.
[6] Cisco Inc., *FlowCollector/DataAnalyzer*, http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/nfc/index.htm.
[7] *Flow-tools*, http://www.splintered.net/sw/flow-tools/, 2003.
[8] Lawrence Berkeley National Labs, *libpcap*, Network Research Group, http://www.tcpdump.org/.
[9] Cyclades Inc., *TS100 Secure Server*, http://www.cyclades.com/, 2002.

[3] NetFlow flows do not contain any payload information that is used by ntop for detecting some protocols.