

nProbe: an Open Source NetFlow Probe for Gigabit Networks

*Adapted by nPulse Networks LLC from an original paper by Luca Deri.
<http://www.npulsenetworks.com>*

Abstract

Cisco NetFlow is an industry standard protocol suitable for monitoring network traffic. Although most high-end network routers support NetFlow, very often flows are computed only on a small portion of the overall traffic due to performance limitations in the 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 can be successfully used to monitor high-speed networks at full wire speed without packet sampling in scenarios where commercial NetFlow probes could not deliver the required performance.

NetFlow Traffic Monitoring: State of the Art

Cisco NetFlow[®] is a widespread standard for network traffic accounting. Leading network manufacturers such as Cisco, Juniper Networks and Extreme Networks provide NetFlow agents as part of their operating systems. Unfortunately most of these implementations are not able to handle more than 5,000 to 10,000 packet/sec¹ unless additional specialized and costly network boards such as Cisco MSFC (Multilayer Switching Feature Card) are used.

At the highest level, NetFlow defines “probes”, which gather data from points within the network, and “collectors”, which aggregate the data, and may also analyze it and present reports.

In the software world, there are several NetFlow collectors available^A. 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, most NetFlow software available has been designed for the collector side. The main consequence is that NetFlow is not in very widespread use because:

- Only high-end routers support NetFlow, and only a limited part of the router’s valuable processing power can be assigned to non-routing tasks such as gathering packet statistics.
- NetFlow-aware routers are relatively expensive and require some expertise for their setup.
- 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.
- 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.

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 NetFlow. The consequence of all this, is that most of the people still monitor their networks using SNMP MIB-II interface counters^B and MRTG^C.

¹ Some high-end routers are able to handle millions of packets/sec but usually can only process a few thousand packets/sec for the purpose of NetFlow traffic measurement.

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

nProbe Architecture

nProbe is an open source NetFlow probe. The application captures packets flowing on an Ethernet segment, computes NetFlow flows, and exports them to the specified collector(s). Users can fully control flow parameters (e.g. flow expire time) as well as the flow collectors. Exported flows can be collected using commercial applications such as Cisco NetFlow Collector^E, or analyzed using open source tools such as ntop and flow-tools^F.

nProbe's main features include:

- Support for NetFlow v5, v9 extensions and flow templates
- Support of IPFIX (draft 3) over SCTP/TCP/UDP
- Ability to keep up with Gbit speeds on Ethernet networks, handling >1.3 million packets per second without sampling on commodity hardware
- Support for major operating systems 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^G, a portable packet capture library, and are 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 reported. In order to avoid transmitting to the collector packets containing few flows, the user can specify the minimum number of flows per packet. The probe will not send the packet until that limit is reached.

During the application design, the author carefully studied the hash performance. As buckets are frequently added and removed from the hash, a simple hash would not be adequate. This is because it would be necessary to periodically rehash the entries, resulting in the loss of packets arriving during the rehash period. For this reason the following design principles have been adopted:

- The hash size is specified at startup to allocate all the memory at once, because;
 - 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 there is sufficient memory to start the probe, then it is guaranteed that the probe will not quit during runtime due to lack of memory.
 - during hash resizing, the probe could experience packet loss.

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

Like all NetFlow probes, nProbe supports various aggregation facilities including port, address, AS (Autonomous System) source/peer, TOS (Type of Service) and protocol. As the probe does not run on a router, the author had to provide an alternative method for gathering AS information. 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 a utility that extracts the BGP table from network routers, so that human intervention is not needed.

The nProbe architecture enabled the probe to handle several hundred thousand packets per second, even on a very modest server platform and with standard I/O drivers. Tests² were 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 a driver from Intel's website plus a traffic simulator able to fill a Gbps line, with the following results:

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

These tests were performed using nProbe 1.x. A newer release, nProbe 2.x, is now available and able to deliver a significantly better performance due to a new hash management. The results demonstrate that nProbe on a Gigabit network handles at least an order of magnitude more packets more than the number 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 can be readily implemented on commodity hardware.

Validation

nProbe has been deployed for lengthy periods 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 Gbps 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 only limited per-flow traffic statistics. For this reason the author decided to use ntop as pure collector, and to let nProbe gather the traffic and export flow data to ntop. With this two level architecture, ntop has been able to scale at Gbit speeds while mostly³ 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 Gbps Internet link.

² The tests were performed by Hauman Technologies Inc.

³ NetFlow flows do not contain the payload information that is needed by ntop for detecting some protocols.

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 heavily-loaded Gbps networks using a high-end PC running 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.

About the Author

Luca Deri <deri@ntop.org> is currently a lecturer in the Computer Science Department at the University of Pisa. 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. Luca's home page is <http://luca.ntop.org/>.

About nPulse Networks LLC

nPulse is a global leader in the hardware-acceleration of open-source-based solutions for network monitoring, network security, traffic analysis and data management. With its customers and partners, nPulse works every day at the leading edge of network monitoring and security technology, providing deep insight into "network vital signs". nPulse Networks is headquartered in Reston, Virginia, and also maintains development centers in Charlottesville, VA and in Pisa, Italy. For more information, visit www.npulsenetworks.com.

References

- ^A SWITCH - The Swiss Education and Research Network, *Flow Measurement Tools*, <http://www.switch.ch/tftant/floma/software.html>, 2003.
- ^B K. McCloghrie, M.T. Rose, *Management Information Base for Network management of TCP/IP-based Internets: MIB-II*, RFC 1213, March 1991.
- ^C T. Oetiker, *Multi Router Traffic Grapher (MRTG)*, <http://www.mrtg.org/>.
- ^D L. Deri, R. Carbone, and S. Suin, *Monitoring Networks Using Ntop*, Proc. of IM 2001, Seattle, May 2001.
- ^E Cisco Inc., *FlowCollector/DataAnalyzer*, <http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/nfc/index.htm>.
- ^F *Flow-tools*, <http://www.splintered.net/sw/flow-tools/>, 2003.
- ^G Lawrence Berkeley National Labs, *libpcap*, Network Research Group, <http://www.tcpdump.org/>.