nBox, nProbe, n2disk
User’s Guide

Version 5.4

December 2010

© 2010-07

Table of Contents

Release History

Release 5.3 (December 2009)
Updated to nProbe 5.4.x (with FastBit)
Major Manual Rework

Release 5.2 (April 2009)
Updated nBox firmware
Added n2disk 1.0
Updated nProbe to 5.1x version.
Removed nDumper

Release 5.1 (November 2008)
Updated nBox firmware
Updated nProbe to 5.0.x version.
Added nDumper 0.6.x version.
Added n2n

Release 5.0 (February 2008)
Updated nBox firmware
Updated nProbe with latest features.
Updated ntop with latest 3.3.X version.

Release 4.0 (July 2007)
Updated nBox with latest 2.6 kernel series image
Updated nProbe with 4.9 version coverage.

Release 3.9 (April 2005)
Updated nBox section

Release 3.0.1 (February 2004)
Updated nBox section

Release 3.0 (January 2004)
Added nProbe 3.0 coverage

Release 2.2 (October 2003)
Added nBox coverage

Release 2.1 (June 2003)
Added nFlow support

Release 2.0.1 (February 2003)
Added the ability to save flows on disk (-P flag)

Release 2.0 (January 2003)

Added the ability to select multiple NetFlow collectors.

Added –p flag for ignoring TCP/UDP ports.

Added –e flag for slowing down flow export speed.

Added –u flag for identifying input NetFlow devices into emitted flows.

Added –z flag for preventing nProbe from emitting tiny flows.

Added –a flag for selecting the way flows are exported to several collectors (if defined).

Added the ability to control an LCD display where the probe can report traffic statistics.

Enhanced TCP flags support in exported flows.

Release 1.3 (July 2002)

First public release.

1. Introduction

Traffic measurements are necessary to operate all types of IP networks. Network admins need a detailed view of network traffic for several reasons and some of these could be security, accounting and management. The traffic compositions have to be analyzed accurately when estimating traffic metrics or when finding network problems. All measurements have to be made by inspecting all the packets flowing into the network trunk analyzed (such as router and/or switches). This analysis could be done on the fly or by logging all the packets and then post-processing them. But with the increasing network capacities and traffic volumes this kind of approach is not suitable for the most cases. Instead similar packets (packets with a set of common properties) can be grouped together composing what is called "flows". As an example, a flow can be composed of all flowing packets that share the same source and destination address so a flow can be derived using only some fields of a network packet. This way, similar types of traffic can be stored in a more compact format without losing the information we are interested in. This information can be aggregated in a flow datagram and exported to a collector able to report network metrics in a user-friendly format.

As soon as collected this information provides a detailed view of the network traffic.

Precise network metric measurements are a challenging task so a lot of work has been done in this filed. In commercial environments, NetFlow is probably the de-facto standard for network traffic accounting and billing. The NetFlow technology has originally been created by Cisco in 1996 and is now standardized as Internet Protocol Flow Information eXport (IPFIX – RFC 3917). NetFlow is based on the probe/collector paradigm. The probe, usually part of network appliance such as a router or a switch, is deployed on the measured network segment, it sends traffic information in NetFlow format towards a central collector.

nProbe is a software NetFlow v5/v9/IPFIX probe able to collect and aggregate network traffic, and export it using the standard Cisco NetFlow v5/v9/IPFIX format. It is available for most of the OSs on the market (Wndows, Solaris, Linux, MacOSX). When installed on a PC, nProbe turn it into a Network-aware monitoring appliance.

Many users, who used nProbe, realized that running a network probe on a PC is not always the best choice for several reasons:

PCs have moving parts that can break and make the probe unavailable.

PCs are large, need monitors and keyboards, whereas probes often need to be deployed on places where there is not much space available.

Administering PCs is not cheap and they require the purchase of an OS, its installation and maintenance.

In large networks divided in several trunks it is necessary to have several probes each analyzing a trunk. This requires that multiple PC running nProbe are deployed across the network.

The cost (for both hardware and maintenance) of a PC+nProbe is not neglectable in particular if several probes need to be deployed.

In many cases, no technician are available at the monitored site and something plug and play in needed.

To face these matters and to provide an All-in-One high-performance and reliable solution, nBox has been designed and developed.

nBox is based on Linux OS, and thanks to an optimized Linux kernel with the PF_RING module that significantly improves the packet capture process, nBox is able to monitor and analyze network trunks at full speed without the need for a hardware accelerator card. The nProbe application installed in the nBox server has been optimized and extended with respect to the version of the very popular open-source software. The new nProbe contains some features not included in the open version and the software has been carefully optimized to run on the nBox server.

If you are a user that does not want to bother with installing nProbe on a PC or you need to use a high performance and reliable network probe solution then you are probably an nBox user.

In some environments it would be nice to distribute light network probes on the network that send traffic information towards a central traffic analysis console such as ntop or any other NetFlow/IPFIX compliance collector. In order to satisfy the above requirements nProbe and ntop can be used together.

nBox includes both a NetFlow probe (nProbe) and a collector (ntop) for v5/v9/IPFIX NetFlow flows.

Based on your network speed and traffic volumes different nBox servers can be used. Please refer to Chapter 5 of this manual to have an idea of the different nBox configuration and for the typical usage scenario.

nBox can be effectively used:

To analyze NetFlow flows generated by your border gateway.

To replace the embedded, low-speed NetFlow probe available on your router/switch

As a NetFlow probe that sends flows towards one or more collectors either ntop or a commercial one (e.g. Cisco NetFlow Collector or HP-OV).

Both as a probe and collector at the same time. ntop can be used as collector and analyzer for nProbe-generated flows.

Finally it is worth to say that nBox is quite easy to administer using the very intuitive embedded web interface. nBox is easy to setup and it is immediately ready to use with little configuration effort. Throughout this document we are going to describe the main components of the nBox web interface.

This manual is divided in three main parts:

- The first one covers nProbe and shows how it can be configured and deployed on your network
- The second part covers the usage of nProbe with ntop flows collector
- The final part is dedicated to the nBox appliance.

2. nProbe

There are two main version of the nProbe probe:

The one that is public available and distributed in both source and binary format (see Appendix E for the license information).

The one (nProbe 5.1.3x Pro) distributed only in binary format with the nBox appliance.

The first version of nProbe is available for use with no further configuration. On the other hand the Pro version requires an nBox valid license in order to work (please refer to section 4.1.2 for more information about the license).

In the following sections we refer to the version included within nBox which have different features compared with the public version, both in term of performance and functionalities.

2.1 Main Features

Some of the nProbe features include:

- Limited memory footprint (regardless of the network size and speed) and CPU savvy.
- Designed for running on environments with limited resources (the nProbe binary < 100 Kb).
- Fully user configurable.
- Fully NetFlow v4/v5/v9 IPFIX compliant.
- High-performance probe: commercial probes included those embedded on routers and switches are often not able to keep up with high-speeds or, when able, their performance decreases dramatically handling small size packets.
- Ability to act as flow collector and proxy
- Ability to natively save flows into MySQL and SQLite, as well as text and binary.
- Native PF_RING support for high speed flow generation
- Support of acceleration cards such as Endace, Mutina Technology, and Tilera.
- Support of detect protocols via DPI (deep packet inspection) and report protocol name in flows for precise collector protocol accounting
- Ability to forge NetFlow interface Ids based on MAC/IP addresses.
- New nprobe architecture for better performance with respect to previous versions.
- Support of both flow and packet sampling

2.3 Usage

The nProbe probe has to be activated on a PC from where it is possible to capture all the traffic you are interested in. For this reason, in case of switched networks, it is necessary to either mirror traffic (VLAN or port mirror) or place the probe on a location (e.g. closed to the border gateway) where most of the traffic flows.

When activated, nProbe begins to collect traffic data and emit NetFlow in the format specified by the user (v5 if none) flows towards the specified collector. A set of packets with the same (src ip & port, dst ip & port, protocol #) is called flow (note that some protocols such as ICMP have no concept of ports). Every flow, even a very long-standing ISO CD image download, has a limited lifetime; this because the flow collector should periodically receive flow chunks to account traffic precisely.

In the following sections, we discuss all the nProbe 4.9.x Pro command line options and how to efficiently configure nProbe to run on your network.

2.3.1 nProbe Command Line Options

nProbe allows network administrators to precisely tune the flow generation policy. In particular, it is possible to specify a lot of command line options.

the available options and a detailed explanation of each option are listed below:

```
# nprobe -h

Welcome to nprobe v.6.0.0 ($Revision: 1713 $) for i386-apple-darwin10.4.1

Built on 09/30/10 08:36:02 PM
Copyright 2002-10 by Luca Deri <deri@ntop.org>

Usage:
nprobe -n <host:port|none> [-i <interface|dump file>] [-t <lifetime timeout>]
              [-d <idle timeout>] [-l <queue timeout>] [-s <scan cycle>] [-N]
              [-p <aggregation>] [-f <filter>] [-a] [-b <level>] [-G] [-O <# threads>]
              [-P <path>] [-F <dump timeout>] [-D <format>]
              [-u <in dev idx>] [-Q <out dev idx>]
              [-I <probe name>] [-v] [-w <hash size>] [-e <flow delay>] [-B <packet count>]
              [-z <min flow size>] [-M <max num flows>][-R <payload Len>]
              [-x <payload policy>] [-E <engine>] [-C <flow lock file>]
              [-m <min # flows>][-q <host:port>]
              [-S <sample rate>] [-A <AS list>] [-g <PID file>]
              [-T <flow template>] [-U <flow template id>]
              [-o <v9 templ. export policy>] [-L <local nets>] [-c] [-r]
              [-1 <MAC>@<ifIdx>][-3 <port>] [-4] [-5 <port>] [-6]
              [-9 <path>] [--black-list <networks>] [--pcap-file-list <filename>]
              [--fastbit <dir>] [--fastbit-rotation <mins>]
              [--fastbit-template <flow template>] [--fastbit-index <flow template>]
              [--fastbit-exec <cmd>]

[--collector|-n] <host:port|none>   | Address of the NetFlow collector(s).
```

```
                                    | Multiple collectors can be defined using
                                    | multiple -n flags. In this case flows
                                    | will be sent in round robin mode to
                                    | all defined collectors if the -a flag
                                    | is used. Note that you can specify
                                    | both IPv4 and IPv6 addresses.
                                    | If you specify none as value,
                                    | no flow will be export; in this case
                                    | the -P parameter is mandatory.
[--interface|-i] <iface|pcap>       | Interface name from which packets are
                                    | captured, or .pcap file (debug only)
[--lifetime-timeout|-t] <timeout>   | It specifies the maximum (seconds) flow
                                    | lifetime [default=120]
[--idle-timeout|-d] <timeout>       | It specifies the maximum (seconds) flow
                                    | idle lifetime [default=30]
[--queue-timeout|-l] <timeout>      | It specifies how long expired flows
                                    | (queued before delivery) are emitted
                                    | [default=30]
[--scan-cycle|-s <scan cycle>]      | It specifies how often (seconds) expired
                                    | flows are emitted [default=30].
                                    | If -P is used, the scan cycle will be
                                    | set to the value of the -F parameter
[--rebuild-hash|N]                  | Rebuild the hash at each scan. Useful for
                                    | producing flows that last as the scan
                                    | cycle as netflow collectors do. This
                                    | option is ignored when -P is not used.
[--aggregation|-p] <aggregation>    | It specifies the flow aggiregation level:
                                    | <VLAN Id>/<proto>/<IP>/<port>/<TOS>/<AS>
                                    | where each element can be set to 0=ignore
                                    | or 1=take care. Example '-p 1/0/1/1/1/1'
                                    | ignores the protocol, whereas
                                    | '-p 0/0/1/0/0/0' ignores everything
                                    | but the IP
[--bpf-filter|-f] <BPF filter>      | BPF filter for captured packets
                                    | [default=no filter]
[--all-collectors|-a]               | If several collectors are defined, this
                                    | option gives the ability to send all
                                    | collectors all the flows. If the flag is
                                    | omitted collectors are selected in
                                    | round robin.
[--verbose|-b] <level>              | Verbose output:
                                    | 0 - No verbose logging
                                    | 1 - Limited logging (traffic statistics)
                                    | 2 - Full verbose logging
[--daemon-mode|-G]                  | Start as daemon.
[--num-threads|-O] <# threads>      | Number of packet fetcher threads
                                    | [default=2]. Use 1 unless you know
                                    | what you're doing.
[--dump-path|-P] <path>             | Directory where dump files will
                                    | be stored.
[--dump-frequency|-F] <dump timeout>| Dump files dump frequencey (sec).
                                    | Default: 60
[--dump-format|-D] <format>         | <format>: flows are saved as:
                                    | b        : raw/uncompressed flows
                                    | t        : text flows
                                    | d        : SQLite
                                    | Example: -D b. Note: this flag has no
```

```
                                     | effect without -P.
[--in-iface-idx|-u] <in dev idx>     | Index of the input device used in the
                                     | emitted flows (incoming traffic). The default
                                     | value is 0. Use -1 as value to dynamically
                                     | set to the last two bytes of
                                     | the MAC address of the flow sender.
[--out-iface-idx|-Q] <out dev idx>   | Index of the output device used in the
                                     | emitted flows (outgoing traffic). The default
                                     | value is 0. Use -1 as value to dynamically
                                     | set to the last two bytes of
                                     | the MAC address of the flow receiver.
[--vlanid-as-iface-idx]              | Use vlanId (or 0 if the traffic isn't tagged)
                                     | as interface index. Note that this option
                                     | superseedes the --in/out-iface-idx options
[--nprobe-version|-v]                | Prints the program version.
[--flow-lock|-C] <flow lock>         | If the flow lock file is present no flows
                                     | are emitted. This facility is useful to
                                     | implement high availability by means of
                                     | a daemon that can create a lock file
                                     | when this instance is in standby.
[--help|-h]                          | Prints this help.
[--syslog|-I] <probe name>           | Log to syslog as <probe name>
                                     | [default=stdout]
[--hash-size|-w] <hash size>         | Flows hash size [default=32768]
[--flow-delay|-e] <flow delay>       | Delay (in ms) between two flow
                                     | exports [default=1]
[--count-delay|-B] <packet count>    | Send this many packets before
                                     | the -e delay [default=1]
[--min-flow-size|-z] <min flow size>| Minimum TCP flow size (in bytes).
                                     | If a TCP flow is shorter than the
                                     | specified size the flow is not
                                     | emitted [default=unlimited]
[--max-num-flows|-M] <max num flows>| Limit the number of active flows. This is
                                     | useful if you want to limit the memory
                                     | or CPU allocated to nProbe™ in case of non
                                     | well-behaved applications such as
                                     | worms or DoS. [default=4294967295]
[--payload-length|-R] <payload Len>  | Specify the max payload length
                                     | [default: 0 bytes]
[--payload-policy|-x] <policy>       | Specify the max payload export policy.
                                     | Format: TCP:UDP:ICMP:OTHER where all
                                     | parameters can se set to:
                                     | 0: no payload for the selected protocol
                                     | 1: payload for the selected protocol
                                     | 2: payload for TCP sessions with SYN flag
                                     | Example -x 2:0:0:0 [default=2:0:0:0]
[--netflow-engine|-E] <engine>       | Specify the engine type and id.
                                     | The format is engineType:engineId.
                                     | [default=0:146] where engineId is a
                                     | random number.
[--min-num-flows|-m] <min # flows>   | Minimum number of flows per packet
                                     | unless an expired flow is queued
                                     | for too long (see -l) [default=30
                                     | for v5, dynamic for v9]
[--sender-address|-q] <host:port>    | Specifies the address:port of the flow
                                     | sender. This optionis useful for hosts
                                     | with multiple interfaces or if flows
```

```
                                    | must be emitted from a static port
[--sample-rate|-S] <pkt rate>:<flow rate>
                                    | Packet capture sampling rate and flow
                                    | sampling rate. If <pkt rate> starts with '@'
                                    | it means that nprobe will report the specified
                                    | sampling rate but will not sample itself
                                    | as incoming packets are already sampled
                                    | on the specified capture device at the
                                    | specified rate. Default: 1:1 [no sampling]
[--as-list|-A] <AS list>            | GeoIP file containing the list of known ASs.
                                    | Example: GeoIPASNum.dat
[--city-list] <City list>           | GeoIP file containing the city/IP mapping.
                                    | Example: GeoLiteCity.dat
[--pid-file|-g] <PID file>          | Put the PID in the specified file
[--flow-templ|-T] <flow template>   | Specify the NFv9 template (see below).
[--flow-templ-id|-U] <templ. id>    | Specify the NFv9 template identifier
                                    | [default: 257]
[--flow-version|-V] <version>       | NetFlow Version: 5=v5, 9=v9, 10=IPFIX
[--flows-intra-templ|-o] <num>      | Specify how many flow pkts are exported
                                    | between template exports [default: 10]
[--local-networks|-L] <networks>    | Specify the local networks (see -c
                                    | and -r options)
[--local-hosts-only|-c]             | All the IPv4 hosts outside the local
                                    | network lists will be set to 0.0.0.0
                                    | (-L must be specified before -c).
                                    | This reduces the load on the probe
                                    | instead of discarding flows on the
                                    | collector side.
[--local-traffic-direction|-r]      | All the traffic going towards
                                    | the local networks (-L must also be
                                    | specified before -r) is assumed incoming
                                    | traffic all the rest is assumed outgoing
                                    | (see also -u and -Q).
[--src-mac-address|-1] <MAC>@<ifIdx>| Flow source MAC address (see below)
[--count|-2] <number>               | Capture a specified number of packets
                                    | and quit (debug only)
[--collector-port|-3] <port>        | NetFlow/sFlow collector port for incoming flows
[--tunnel|-5]                       | Compute flows on tunneled traffic rather than
                                    | on the external envelope
[--no-promisc|-6]                   | Capture packets in non-promiscuous mode
[--smart-udp-frags|-7]              | Ignore UDP fragmented packets with fragment offset
                                    | greater than zero, and compute the fragmented
                                    | packet length on the initial fragment header.
[--ipsec-auth-data-len|-8] <len>    | Length of the authentication data of IPSec
                                    | in tunnel mode. If not set, IPSec will not be decoded
[--dump-stats|-9] <path>            | Periodically dump traffic stats into the
                                    | specified file
--black-list <networks>             | All the IPv4 hosts inside the networks
                                    | black-list will be discarded.
                                    | This reduces the load on the probe
                                    | instead of discarding flows on the
                                    | collector side.
--pcap-file-list <filename>         | Specify a filename containing a list
                                    | of pcap files.
                                    | If you use this flag the -i option will be
                                    | ignored.
--csv-separator <separator>         | Specify the separator for text files (see -P)
```

```
                                    | Default is '|' (pipe)
--fastbit <dir>                     | Base directory where FastBit files will be created.
--fastbit-rotation <mins>           | Every <mins> minutes a new FastBit sub-directory is created
                                    | so that each directory contains at most <mins> minutes.
                                    | Default 5 min(s).
--fastbit-template <flow template>  | Fields that will be dumped on FastBit partition. Its syntax
                                    | is the same as the -T flag. If this flag is not specified,
                                    | all the specified flow elements (-T) will be dumped.
--fastbit-index <flow template>     | Index each directory containing FastBit files as soon as
                                    | the directory has been dumped. The flow template specifies
                                    | which columns will be indexed. Its syntax is the same as
                                    | the -T flag. This option requires that fbindex application
                                    | is installed or built. If this flag is not specified, all
                                    | columns will be indexed.
--fastbit-exec <cmd>                | Execute the specified command after a directory has been
                                    | dumped (and optionally indexed). The command must take an
                                    | argument that is the path to the directory just dumped.
--bi-directional                    | Force flows to be bi-directional. This option
                                    | is not supported by NetFlow V5 that by nature
                                    | supports only mono-directional flows
--account-l2                        | NetFlow accounts IP traffic only, not counting
                                    | L2 headers. Using this option the L2 headers
                                    | are also accounted
--dump-metadata <file>              | Dump flow metadata into the specified file
                                    | and quit
--event-log <file>                  | Dump relevant activities into the specified log file

Further plugin available command line options
---------------------------------------------------
30/Sep/2010 21:18:42 [plugin.c:145] Loading plugins [.so] from ./plugins
30/Sep/2010 21:18:42 [dbPlugin.c:72] Initializing DB plugin
[BGP Update Listener]
  --bgp-port <port>                                  | TCP port on which BGP updates will be sent

[MySQL DB]
  --mysql=<host>:<dbname>:<table_prefix>:<user>:<pw> | Enable MySQL database support configuration
  --mysql-skip-db-creation                           | Skip database schema creation

[DNS Protocol Dissector]
  --dns-dump-dir <dump dir>                          | Directory where DNS logs will be dumped

[HTTP Protocol Dissector]
  --http-dump-dir <dump dir>                         | Directory where HTTP logs will be dumped
  --http-exec-cmd <cmd>                              | Command executed whenever a directory has been
dumped
  --dont-hash-cookies                                | Dump cookie string instead of cookie hash
  --dont-nest-dump-dirs                              | Don't create subdirs on the dump directory
  --max-http-log-lines                               | Max number of lines per log file (default 10000)

[MySQL Plugin]
  --mysql-dump-dir <dump dir>                        | Directory where MySQL logs will be dumped
  --mysql-exec-cmd <cmd>                             | Command executed whenever a directory has been
dumped
  --max-mysql-log-lines                              | Max number of lines per log file (default 10000)
```

```
Note on interface indexes and (router) MAC addresses
---------------------------------------------------
When -u and -Q are specified, it is possible to also specify -1 (even multiple
times) for simulating a router running nProbe™. In this case nProbe™ works
as follows:

[Use Case] -u 1 -Q 2 -1 AA:BB:CC:DD:EE:FF@3 -1 11:22:33:44:55:66@4
           All the flows have direction 1->2 except those who are originated
           from MAC AA:BB:CC:DD:EE:FF that have 3 as source interface id
           and those who are originated from 11:22:33:44:55:66 that have
            4 as source interface (direction = flow interface index)

NetFlow v9/IPFIX format [-T]
----------------
The following options can be used to specify the format:

 ID   Flow Label                Description
------------------------------------------------
[  1] %IN_BYTES                   Incoming flow bytes (src->dst)
[  1] %SYSTEM_ID
[  2] %IN_PKTS                    Incoming flow packets (src->dst)
[  2] %INTERFACE_ID
[  3] %FLOWS                      Number of flows
[  3] %LINE_CARD
[  4] %PROTOCOL                   IP protocol byte
[164] %PROTOCOL_MAP               IP protocol name
[  4] %NETFLOW_CACHE
[  5] %SRC_TOS                    Type of service byte
[  5] %TEMPLATE_ID
[  6] %TCP_FLAGS                  Cumulative of all flow TCP flags
[  7] %L4_SRC_PORT                IPv4 source port
[167] %L4_SRC_PORT_MAP            IPv4 source port symbolic name
[  8] %IPV4_SRC_ADDR              IPv4 source address
[  9] %IPV4_SRC_MASK              IPv4 source subnet mask (/<bits>)
[ 10] %INPUT_SNMP                 Input interface SNMP idx
[ 11] %L4_DST_PORT                IPv4 destination port
[171] %L4_DST_PORT_MAP            IPv4 destination port symbolic name
[ 12] %IPV4_DST_ADDR              IPv4 destination address
[ 13] %IPV4_DST_MASK              IPv4 dest subnet mask (/<bits>)
[ 14] %OUTPUT_SNMP                Output interface SNMP idx
[ 15] %IPV4_NEXT_HOP              IPv4 next hop address
[ 16] %SRC_AS                     Source BGP AS
[ 17] %DST_AS                     Destination BGP AS
[ 21] %LAST_SWITCHED              SysUptime (msec) of the last flow pkt
[ 22] %FIRST_SWITCHED             SysUptime (msec) of the first flow pkt
[ 23] %OUT_BYTES                  Outgoing flow bytes (dst->src)
[ 24] %OUT_PKTS                   Outgoing flow packets (dst->src)
[ 27] %IPV6_SRC_ADDR              IPv6 source address
[ 28] %IPV6_DST_ADDR              IPv6 destination address
[ 29] %IPV6_SRC_MASK              IPv6 source mask
[ 30] %IPV6_DST_MASK              IPv6 destination mask
[ 32] %ICMP_TYPE                  ICMP Type * 256 + ICMP code
[ 34] %SAMPLING_INTERVAL          Sampling rate
[ 35] %SAMPLING_ALGORITHM         Sampling type (deterministic/random)
[ 36] %FLOW_ACTIVE_TIMEOUT        Activity timeout of flow cache entries
[ 37] %FLOW_INACTIVE_TIMEOUT      Inactivity timeout of flow cache entries
[ 38] %ENGINE_TYPE                Flow switching engine
[ 39] %ENGINE_ID                  Id of the flow switching engine
[ 40] %TOTAL_BYTES_EXP            Total bytes exported
[ 41] %TOTAL_PKTS_EXP             Total flow packets exported
[ 42] %TOTAL_FLOWS_EXP            Total number of exported flows
[ 56] %IN_SRC_MAC                 Source MAC Address
[ 57] %OUT_DST_MAC                Destination MAC Address
[ 58] %SRC_VLAN                   Source VLAN
[ 59] %DST_VLAN                   Destination VLAN
[ 60] %IP_PROTOCOL_VERSION        [4=IPv4][6=IPv6]
[ 61] %DIRECTION                  [0=ingress][1=egress] flow
[ 62] %IPV6_NEXT_HOP              IPv6 next hop address
[ 70] %MPLS_LABEL_1               MPLS label at position 1
[ 71] %MPLS_LABEL_2               MPLS label at position 2
[ 72] %MPLS_LABEL_3               MPLS label at position 3
[ 73] %MPLS_LABEL_4               MPLS label at position 4
[ 74] %MPLS_LABEL_5               MPLS label at position 5
[ 75] %MPLS_LABEL_6               MPLS label at position 6
[ 76] %MPLS_LABEL_7               MPLS label at position 7
[ 77] %MPLS_LABEL_8               MPLS label at position 8
[ 78] %MPLS_LABEL_9               MPLS label at position 9
[ 79] %MPLS_LABEL_10              MPLS label at position 10
[148] %FLOW_ID                    Serial Flow Identifier
[NFv9 57552][IPFIX 35632.80] %FRAGMENTS                     Number of fragmented flow packets
[NFv9 57554][IPFIX 35632.82] %CLIENT_NW_DELAY_SEC           Network latency client <-> nprobe (sec)
[NFv9 57555][IPFIX 35632.83] %CLIENT_NW_DELAY_USEC          Network latency client <-> nprobe (usec)
[NFv9 57556][IPFIX 35632.84] %SERVER_NW_DELAY_SEC           Network latency nprobe <-> server (sec)
[NFv9 57557][IPFIX 35632.85] %SERVER_NW_DELAY_USEC          Network latency nprobe <-> server (usec)
[NFv9 57558][IPFIX 35632.86] %APPL_LATENCY_SEC              Application latency (sec)
[NFv9 57559][IPFIX 35632.87] %APPL_LATENCY_USEC             Application latency (usec)
[NFv9 57570][IPFIX 35632.98] %ICMP_FLAGS                    Cumulative of all flow ICMP types
[NFv9 57573][IPFIX 35632.101] %SRC_IP_COUNTRY               Country where the src IP is located
[NFv9 57574][IPFIX 35632.102] %SRC_IP_CITY                  City where the src IP is located
[NFv9 57575][IPFIX 35632.103] %DST_IP_COUNTRY               Country where the dst IP is located
[NFv9 57576][IPFIX 35632.104] %DST_IP_CITY                  City where the dst IP is located
[NFv9 57577][IPFIX 35632.105] %FLOW_PROTO_PORT              L7 port that identifies the flow protocol or 0
if unknown
[NFv9 57578][IPFIX 35632.106] %TUNNEL_ID                    Tunnel identifier (e.g. GTP tunnel Id) or 0 if
unknown
[NFv9 57579][IPFIX 35632.107] %LONGEST_FLOW_PKT             Longest packet (bytes) of the flow
[NFv9 57580][IPFIX 35632.108] %SHORTEST_FLOW_PKT            Shortest packet (bytes) of the flow
[NFv9 57581][IPFIX 35632.109] %RETRANSMITTED_IN_PKTS        Number of retransmitted TCP flow packets (src-
>dst)
[NFv9 57582][IPFIX 35632.110] %RETRANSMITTED_OUT_PKTS       Number of retransmitted TCP flow packets (dst-
>src)
[NFv9 57583][IPFIX 35632.111] %OOORDER_IN_PKTS              Number of out of order TCP flow packets (dst-
>src)
[NFv9 57584][IPFIX 35632.112] %OOORDER_OUT_PKTS             Number of out of order TCP flow packets (dst-
>src)
[NFv9 57585][IPFIX 35632.113] %UNTUNNELED_PROTOCOL          Untunneled IP protocol byte
[NFv9 57586][IPFIX 35632.114] %UNTUNNELED_IPV4_SRC_ADDR     Untunneled IPv4 source address
[NFv9 57587][IPFIX 35632.115] %UNTUNNELED_L4_SRC_PORT       Untunneled IPv4 source port
[NFv9 57588][IPFIX 35632.116] %UNTUNNELED_IPV4_DST_ADDR     Untunneled IPv4 destination address
[NFv9 57589][IPFIX 35632.117] %UNTUNNELED_L4_DST_PORT       Untunneled IPv4 destination port

Plugin BGP Update Listener templates:
[NFv9 57762][IPFIX 35632.290] %SRC_AS_PATH_1                Src AS path position 1
[NFv9 57763][IPFIX 35632.291] %SRC_AS_PATH_2                Src AS path position 2
```

```
[NFv9 57764][IPFIX 35632.292] %SRC_AS_PATH_3                Src AS path position 3
[NFv9 57765][IPFIX 35632.293] %SRC_AS_PATH_4                Src AS path position 4
[NFv9 57766][IPFIX 35632.294] %SRC_AS_PATH_5                Src AS path position 5
[NFv9 57767][IPFIX 35632.295] %SRC_AS_PATH_6                Src AS path position 6
[NFv9 57768][IPFIX 35632.296] %SRC_AS_PATH_7                Src AS path position 7
[NFv9 57769][IPFIX 35632.297] %SRC_AS_PATH_8                Src AS path position 8
[NFv9 57770][IPFIX 35632.298] %SRC_AS_PATH_9                Src AS path position 9
[NFv9 57771][IPFIX 35632.299] %SRC_AS_PATH_10               Src AS path position 10
[NFv9 57772][IPFIX 35632.300] %DST_AS_PATH_1                Dest AS path position 1
[NFv9 57773][IPFIX 35632.301] %DST_AS_PATH_2                Dest AS path position 2
[NFv9 57774][IPFIX 35632.302] %DST_AS_PATH_3                Dest AS path position 3
[NFv9 57775][IPFIX 35632.303] %DST_AS_PATH_4                Dest AS path position 4
[NFv9 57776][IPFIX 35632.304] %DST_AS_PATH_5                Dest AS path position 5
[NFv9 57777][IPFIX 35632.305] %DST_AS_PATH_6                Dest AS path position 6
[NFv9 57778][IPFIX 35632.306] %DST_AS_PATH_7                Dest AS path position 7
[NFv9 57779][IPFIX 35632.307] %DST_AS_PATH_8                Dest AS path position 8
[NFv9 57780][IPFIX 35632.308] %DST_AS_PATH_9                Dest AS path position 9
[NFv9 57781][IPFIX 35632.309] %DST_AS_PATH_10               Dest AS path position 10

Plugin DNS Protocol Dissector templates:
[NFv9 57677][IPFIX 35632.205] %DNS_QUERY                    DNS QUERY
[NFv9 57678][IPFIX 35632.206] %DNS_QUERY_ID                 DNS query transaction Id
[NFv9 57679][IPFIX 35632.207] %DNS_QUERY_TYPE               DNS query type (e.g. 1=A, 2=NS..)
[NFv9 57680][IPFIX 35632.208] %DNS_RET_CODE                 DNS return code (e.g. 0=no error)
[NFv9 57681][IPFIX 35632.209] %DNS_NUM_ANSWER               DNS # of returned answers

Plugin dump templates:
[NFv9 57592][IPFIX 35632.120] %DUMP_PATH                    Path where dumps will be saved

Plugin HTTP Protocol Dissector templates:
[NFv9 57652][IPFIX 35632.180] %HTTP_URL                     HTTP URL
[NFv9 57653][IPFIX 35632.181] %HTTP_RET_CODE                HTTP return code (e.g. 200, 304...)
[NFv9 57654][IPFIX 35632.182] %HTTP_REFERER                 HTTP Referer
[NFv9 57655][IPFIX 35632.183] %HTTP_UA                      HTTP User Agent
[NFv9 57656][IPFIX 35632.184] %HTTP_MIME                    HTTP Mime Type

Plugin L7 Protocol Recognition templates:
[NFv9 57637][IPFIX 35632.165] %L7_PROTO                     Symbolic layer 7 protocol description

Plugin MySQL Plugin templates:
[NFv9 57667][IPFIX 35632.195] %MYSQL_SERVER_VERSION         MySQL server version
[NFv9 57668][IPFIX 35632.196] %MYSQL_USERNAME               MySQL username
[NFv9 57669][IPFIX 35632.197] %MYSQL_DB                     MySQL database in use
[NFv9 57670][IPFIX 35632.198] %MYSQL_QUERY                  MySQL Query
[NFv9 57671][IPFIX 35632.199] %MYSQL_RESPONSE               MySQL server response

Plugin RTP templates:
[NFv9 57622][IPFIX 35632.150] %RTP_FIRST_SSRC               First flow RTP Sync Source ID
[NFv9 57623][IPFIX 35632.151] %RTP_FIRST_TS                 First flow RTP timestamp
[NFv9 57624][IPFIX 35632.152] %RTP_LAST_SSRC                Last flow RTP Sync Source ID
[NFv9 57625][IPFIX 35632.153] %RTP_LAST_TS                  Last flow RTP timestamp
[NFv9 57626][IPFIX 35632.154] %RTP_IN_JITTER                RTP Jitter (ms * 1000)
[NFv9 57627][IPFIX 35632.155] %RTP_OUT_JITTER               RTP Jitter (ms * 1000)
[NFv9 57628][IPFIX 35632.156] %RTP_IN_PKT_LOST              Packet lost in stream
[NFv9 57629][IPFIX 35632.157] %RTP_OUT_PKT_LOST             Packet lost in stream
[NFv9 57630][IPFIX 35632.158] %RTP_OUT_PAYLOAD_TYPE         RTP payload type
[NFv9 57631][IPFIX 35632.159] %RTP_IN_MAX_DELTA             Max delta (ms*100) between consecutive pkts
```

```
[NFv9 57632][IPFIX 35632.160] %RTP_OUT_MAX_DELTA            Max delta (ms*100) between consecutive pkts

Plugin SIP templates:
[NFv9 57602][IPFIX 35632.130] %SIP_CALL_ID                  SIP call-id
[NFv9 57603][IPFIX 35632.131] %SIP_CALLING_PARTY            SIP Call initiator
[NFv9 57604][IPFIX 35632.132] %SIP_CALLED_PARTY             SIP Called party
[NFv9 57605][IPFIX 35632.133] %SIP_RTP_CODECS               SIP RTP codecs
[NFv9 57606][IPFIX 35632.134] %SIP_INVITE_TIME              SIP SysUptime (msec) of INVITE
[NFv9 57607][IPFIX 35632.135] %SIP_TRYING_TIME              SIP SysUptime (msec) of Trying
[NFv9 57608][IPFIX 35632.136] %SIP_RINGING_TIME             SIP SysUptime (msec) of RINGING
[NFv9 57609][IPFIX 35632.137] %SIP_OK_TIME                  SIP SysUptime (msec) of OK
[NFv9 57610][IPFIX 35632.138] %SIP_BYE_TIME                 SIP SysUptime (msec) of BYE
[NFv9 57611][IPFIX 35632.139] %SIP_RTP_SRC_IP               SIP RTP stream source IP
[NFv9 57612][IPFIX 35632.140] %SIP_RTP_SRC_PORT             SIP RTP stream source port
[NFv9 57613][IPFIX 35632.141] %SIP_RTP_DST_IP               SIP RTP stream dest IP
[NFv9 57614][IPFIX 35632.142] %SIP_RTP_DST_PORT             SIP RTP stream dest port

Plugin SMTP Protocol Dissector templates:
[NFv9 57657][IPFIX 35632.185] %SMTP_MAIL_FROM               Mail sender
[NFv9 57658][IPFIX 35632.186] %SMTP_RCPT_TO                 Mail recipient
```

-n: collector addresses

This specifies the NetFlow collectors addresses to which nProbe will send the flows. If more than one is specified, they need to be separated with a comma or the –n flag can be repeated several times (e.g. -n 172.22.3.4:33,172.22.3.4:34 and -n 172.22.3.4:33 –n 172.22.3.4:34 are equivalent). When multiple collectors are defined, you can control the way flows are exported using the –a option (see below); if on a collector address the destination port is omitted, flows are sent to 2055 port and whereas if all the option is not specified, by default, flows are sent to the loopback interface (127.0.0.1) on port 2055. If this parameter is used, nProbe exports flows towards collector running at 127.0.0.1:2055.

-i: interface name

It specifies the interface from which packets are captured. If -i is not used, nProbe will use the default interface (if any). In case a user needs to activate nProbe on two different interfaces, then we need to activate multiple nProbe instances one per interface. For debugging purposes it is possible to pass nProbe a .pcap file from which packets will be read.

-t: maximum flow lifetime

Regardless of the flow duration, a flow that has been active for more than the specified maximum lifetime will be considered expired and will be emitted. Further packets belonging to the same flow will be accounted on a new flow.

-d: maximum flow idle lifetime

A flow is over when the last packet received is older than the maximum flow idle lifetime. This means that whenever applicable, (e.g. SNMP walk) UDP flows will not be accounted on 1 packet/1 flow basis, but on one global flow that accounts all the traffic. This has a benefit on the total number of generated flows and on the overall collector performance.

-l: maximum queue timeout

It specifies the maximum amount of time that a flow can be queued waiting to be exported. Use this option in order to try to pack several flows into fewer packets, but at the same time have an upper bound timeout for queuing flows into the probe.

-s: flows scan cycle

This flag specifies how often expired flows are emitted towards the specified collector.

-N: rebuild the hash at each scan

In this case at each scan cycle (-s) the hash is rebuilt from scatch

-p: flow aggregation

Flows can be aggregated both at collector and probe side. However probe allocation is much more effective as it reduces significantly the number of emitted flows hence the work that the collector has to carry on. nProbe supports various aggregation levels that can be selected specifying with the –p flag. The aggregation format is <vlanid>/<proto>/<IP>/<port>/<TOS>/<AS> where each option can be set to 0 (ignore) or 1 (take care). Ignored fields are set to a null value. For instance the value 0/0/1/0/0/0 is useful for creating a map of who's talking to who (network conversation matrix).

-f: packet capture filter

This BPF filter (see the appendix for further information about BPF filters) allows nProbe to take into account only those packets that match the filter (if specified).

-a: select flow export policy

When multiple collectors are defined (see –n option), nProbe sends them flows in round robin. However it is possible to send the same flow to all collectors as a flow redirector does if the –a option is used.

-b: enable verbose logging

Using this flag, nProbe generates verbose output that can be used to tune its performance (see chapter 2.4). Zero is the lowest level (little information is printed), 1 displays traffic statistics, 2 is really verbose. Example of traffic statistics:

```
04/Jul/2007 18:16:00 [nprobe.c:1129] Average traffic: [1.7 pkt/sec][1 Kb/sec]

04/Jul/2007 18:16:00 [nprobe.c:1134] Current traffic: [1.9 pkt/sec][1 Kb/sec]

04/Jul/2007 18:16:00 [nprobe.c:1140] Current flow export rate: [0.9 flows/sec]

04/Jul/2007 18:16:00 [nprobe.c:1144] Buckets:

[active=13][allocated=21][free=8][toBeExported=0][frags=0]

04/Jul/2007 18:16:00 [nprobe.c:1149] Fragment queue: [len=0]

04/Jul/2007 18:16:00 [nprobe.c:1153] Num Packets: 111 (max bucket search: 0)

04/Jul/2007 18:16:00 [nprobe.c:1170] 115 pkts rcvd/0 pkts dropped
```

-G: start nprobe as a daemon.

Useful when starting nprobe as daemon.

-O: set the number of threads that fetch packets out of the network interface.

In general the more threads are available, the better is the performance. However it is not suggested to have too many threads as in some platforms this can slow down the probe. Start with 1 and increase it if necessary.

-P: dump flows

This path specifies the directory where flows will be dumped. The dump format is a 4 bytes number that specifies the flow length and then the raw flow. nProbe comes with a simple file for reading dumped flows. This option is useful to save flows on disks for further processing with tools such as flow-tools.

-F:

It specifies the frequency at which files are dumped on disk

-D: dump flows format

Flows stored on disks can be stored in two formats: text with user-specified format or SQLite format. Using flow SQLite format (-D d) can significantly reduce the size of stored files, although all the collectors might not support this format. Text flows (-D b) are the safest setting if you want to use a standard collector able to read flows dump on disk. In order to let the user specify the more appropriate format, nProbe allows users to define the flow format similar to what C programmers do. The available options are specified in Appendix B. Note that this flag has no effect unless –P is used.

-u: input device index

The NetFlow specification contains a numeric index in order to identify flows coming from different interfaces of the same probe. As multiple nProbe instances can be started on the same host but on different devices, the collector to divide flows according to interface number can use this flag. If –u is not used, then nprobe will use as interface index the last two bytes of the mac address of the flow sender.

-Q: output device index

Similar to –u but for the output interface.

-v: print version

This flag is used to print the nProbe version number and date.

-C: flow export lock

This is a simple way to implement high-availability. Start two probes capturing the same data. The master probe emits flows, the slave probe is started with –C <path>. As long as <path> exists, the slave works but no flow is emitted. If the <path> file is deleted (e.g. using an external program for controlling the master/slave such as heartbeat) the slave starts emitting flows. If the file is restored, the slave is silent again.

-h: print help

Prints the nProbe help.

-I: log to syslog

nProbe logs on stdout unless the –g flag (see above) is used. If the syslog needs to be used instead of a file, this flag instruments nProbe to log on it using the specified name (this is useful when multiple nProbe instances are active on the same host). Please note that –g is ignored if –I is used, and this option is not available on nProbe for Win32.

-w: size of the hash that stores the flows

The default size is 16384 and it should be enough for most of networks. In case flows are not emitted often and with strong traffic conditions it would be necessary to increase the hash. In general, it is a good practice to set hashes whose size is a multiple of the original size (i.e. do not extend the hash to 17000, but rather make it 32000 buckets large). To know more about nProbe tuning, see the following chapter.

-e: flow export delay

Some collectors cannot keep up with nProbe export speed. This flag allows flows to be slow down by adding a short delay (specified in ms) between two consecutive exports. The maximum allowed delay is 1000 ms.

-B: packet count delay

It specifies how many flow packets need to be sent before –e is applied,

-z: minimum TCP low size

Peer-to-peer applications, attacks or misconfigured applications often generate a lot of tiny TCP flows that can cause significant load on the collector side. As most collector setups often discard those flows, it is possible to instrument nProbe via the –z flag not to emit such flows. Note that the –z flag affects only the TCP protocol (i.e. UDP, ICMP and other protocols are not affected).

-M: maximum number of active flows

It is used to limit the maximum number of concurrent flows that the probe can sustain. This is useful to prevent the probe from creating as many flows as needed and hence to take over all the available resources.

-R/-x: used to specify how to handle packet payload.

This is useful for tuning those plugins (e.g. VoIP) that need payload inspection. The –R flag specifies the maximum length of the payload that can be stored into a flow. The –x option allows to tune the payload export . Its format is a:b:c:d where a is the export policy for TCP, b for UDP, c for ICMP and d for other protocols (e.g. IGMP). Each format value can be set to 0 (no payload export for the selected protocol) and 1 (export payload for the selected protocol). For TCP it is also possible to specify 2 to enable payload export for TCP sessions with SYN flag (e.g. export the initial payload of a TCP connection).

-E: netflow engine

Specify the netflow engineType:engineId into the generated flows.

-m: minimum number of flows per packet

In order to minimize the number of emitted packets containing flows, it is possible to specify the minimum number of flows that necessarily need to be contained in a packet. This means that the packet is not emitted until the specified number of flows is reached.

-q: flow sender address

This option is used to specify the address and port from which the packets containing flows are coming from. Usually the operating system prevents people to send packets from different addresses of those assigned to the network interfaces.

-S: sample rate

nProbe uses all the captured packets to calculat flows. In some situations (e.g. strong traffic conditions) it is necessary to reduce the number of packets that need to be handled by nProbe. This option specifies the sampling rate, i.e. the number of packets that are discarded between two packets used to produce flows.

-A: AS file

Network probes are usually installed on systems where the routing information is available (e.g. via BGP) in order to specify the AS (Autonomous System) id of the flow peer. As nProbe has no access to BGP information, users need to provide this information by means of a static file whose format is <AS>:<network>. AS information can be retrieved from routers using publicly available tools (e.g. Juniper routers allow BGP information to be retrieved using JunoScript) and saved on a file for nProbe access. The file can be stored in both plain text and gzip format.

-g:

It specifies the path where nProbe will save the process PID.

-T: flow template definition

Contrary to NetFlow v5 where the flow format is fixed, NetFlow V9 and IPFIX flows have a custom format that can be specified at runtime using this option as specified in appendix C.

-U: flow template id

NetFlow v9 and IPFIX flows format is specified in a template which definition is sent by nProbe before the start of sending flows. The flow format is defined by –T, where –U is used to set the template identifier. This option should not be used unless the default template value (257) needs to be changed.

-V: flow export version

It is used to specify the flow version for exported flows. Supported versions are 5 (v5), 9 (v9) and 10 (IPFIX).

-o: intra templates packet export.

It specifies the number of flow packets that are exported between two templates export.

-L: local networks

Use this flag to specify (format network/mask, e.g. 192.168.0.10/24) the list of networks that are considered local (see –c).

-c: track local hosts only

It allows nProbe to set to 0.0.0.0 all those hosts that are considered non-local (see –L). This is useful when it is necessary to restrict the traffic analysis only to local hosts.

-r: set traffic direction

When this option is used (-L must be specified before –r), all the traffic that goes towards the local networks is considered incoming, all the rest is outgoing. This has effect on the –u/-Q that are then forced with –r.

-1: specify a mapping between MAC address/Interface index

In mirrored environments, it is possible to simulat a switched environment by playing with MAC addresses. This option allows users to bind a MAC address to a specified interfaceId. For instance AA:BB:CC:DD:EE:FF@3 binds the MAC address AA:BB:CC:DD:EE:FF to port 3. On this way it is possible to simulate the fact that all the traffic (regardless of the IP address) that is routed by a router with MAC AA:BB:CC:DD:EE:FF is flowing through interfaceId 3. Note that -1 can be specified multiple times.

-2: debug only

Let the probe capture only up to the specified number of packets.

t-3 NetFlow collector port

It is now possible to use the nProbe as NetFlow proxy. With -3 we can see the incoming NetFlow port on which flows are received instead of sniffing packets. nProbe is able to convert flows from various versions. For instance "nprobe -3 2055 –i 192.168.0.1:2056 –V 10" converts each flow received on port 2055 to IPFIX and sends them to 192.168.0.1:2056.

-5:

Let the probe receive captured packets from a remote pcap client.

-6:

With this option nProbe does not use promiscuous mode to capture packets.

-9:dumps some flow statistics on file

Periodically dump NetFlow statistics on the specified file. The format of the file is:

<time when the flow was emitted,total packets,total bytes,total flows>

- -vlanid-as-iface-idx:

Using this option you tell nProbe to use the VLAN id as NetFlow interface index.

- -black-list:

With this option you can specify a list of networks or hosts from which all the incoming packets will be discarded by the probe. The accepted notation can be CIDR format or the classical network/netmask format.

- -mysql: enable MySQL database support configuration.

Using this option the emitted flows will be dumped on the specified MySQL database.

As some people prefer to have a configuration file containing the options that otherwise would be specified on the command line, it is also possible to start nProbe as follows:

```
nprobe <configuration file path>
```

where the configuration file contains the same options otherwise it is specified on the command line. The only difference between the command line and the configuration file is that different options need to be specified on different lines. For instance:

```
nprobe –n 127.0.0.1:2055 –i en0 –a -p
```

is the same as:

```
nprobe /etc/nprobe.conf
```

where /etc/nprobe.conf contains the following lines:

```
# cat /etc/nprobe.conf
-n=127.0.0.1:2055
-i=en0
-a=
-p=
```

Note that flags with no parameter associated (e.g. –a) also need to have '=' specified.

Any standard NetFlow collector (e.g. ntop) can be used to analyze the flows generated by nProbe. When used with ntop, the nProbe can act as a remote and light traffic collector and ntop as a central network monitoring console. See chapter 3 for further information about this topic.

2.3.2 nProbe on Windows

Some Windows versions NT/2K/XP/Vista support the concept of service. nProbe is activated as service or application depending on the Windows version. The nProbe installer registers the service and creates an entry on the Start menu. On Windows 95/98/ME, nProbe can be activated only on a console or on a batch file, whereas on NT/2K/XP/Vista can also be activated as a service (default).

```
E:\ntop\Source\nprobe\Debug>nprobe /h
Available options:
/i [nprobe options] - Install nprobe as service
/c [nprobe options] - Run nprobe on a console
/r- Deinstall the service
Example:
Install nprobe as a service: 'nprobe /i -i 0 -n 192.168.0.1:2055'
Remove the nprobe service: 'nprobe /r'
```

Notes:

Type 'nprobe /c -h' to see all options

In order to reinstall a service with new options it is necessary to first remove the service, then add it again with the new options.

Services are started/stopped using the Services control panel item.

If nProbe is started on the console, the /c flag needs to be used (e.g. nprobe /c –n 127.0.0.1:2055). If used as service, the command line options need to be specified at service registration and can be modified only removing and adding the service. The nProbe installer registers nProbe as a service with the default options. If you need to change the nProbe setup, you need to do the following:

```
nprobe /r      Remove the service
nprobe /i <put your options here>        Install the service with the specified options.
```

Services are started and stopped using the Services application part of the Windows administrative tools.

As network interfaces on Windows can have long names, a numeric index is associated to the interface in order to ease the nProbe configuration. The association interface name and index is shown typing the 'nprobe /c –h'

```
C:\ntop\nprobe\Debug>nprobe.exe/c -h
```

```
Running nProbe for Win32.
Welcome to nprobe v.5.1.3 for Win32
Built on 05/03/07 10:35:28
Copyright 2002-07 by Luca Deri <deri@ntop.org>
[…]
Available interfaces:
[index=0] 'Adapter for generic dialup and VPN capture'
[index=1] 'Realtek 8139-series PCI NIC'
 […]
```

For instance, in the above example the index 1 is associated to the interface Realtek 8139-series PCI NIC, hence in order to select this interface nprobe needs to be started with –i 1 option.

2.4 Tuning nProbe Performance

As nProbe can be deployed on very different environments, it is necessary to tune it according to the network where it is active. In order to achieve a good probe setup, it is necessary to understand how nProbe is working internally. Each captured packet is analyzed, associated to a flow, and stored onto a hash. Periodically, the hash is analyzed and expired flows are emitted[1]. The hash size is static (-w flag)[2] and_ this allows nProbe to:

Allocate all the needed memory at startup (this is compulsory on embedded systems where memory is limited and it is necessary to know at startup whether a certain application can operate with the available resources).

Avoid exhausting all the available memory in case of attacks that can produce several flows.

Selecting the hash size is a matter of trade-off between efficiency (an efficient hash is at least 1/3 empty) and memory usage. This statement does not mean that a huge hash is always the solution as the flow export process can be slower (and more CPU cycles are needed) as a large hash needs to be explored.

On the other hand, the hash size is just a part of the problem. In fact, the hash fill percentage can be also controlled by other factors such as:

[1] It is worth to remark that packets are captured while nProbe performs flow export (i.e. packet capture is not stopped during flow export).

[2] Note that as of nProbe 4.x the basic hash has a static size specified by –w that can grow as needed according to traffic conditions.

- Reducing the flow lifetime (-t)
- Reducing the maximum flow idle time (-d)
- Increasing how often the hash is walked searching expired flows (-s)

nProbe allows users to ease the tuning process by printing the status of internal hashes using the –b flag. Users who experience severe nProbe performance problems, packet loss or high CPU usage, should start nProbe with –b in order to find out whether their probe setup is optimal.

2.5 Frequently Asked Questions

Q: I'm sending 60 bytes ping packets using 'ping –s 60' but nProbe reports 92 bytes packets.

A: nProbe counts the packet size at IP level. An ICMP Echo Request packet with 60 bytes payload is 92 bytes long.

Q: I need to capture traffic from several interfaces but nProbe allows just one interface to be used. What can I do?

A: You can start several instances of nProbe, each on a different network interface.

Q: nProbe is exporting flows too fast and my collector cannot keep up with it. How can I slow down nProbe export rate?

A: nProbe has been for high-speed networks (1Gb and above) so its export rate can be high due to traffic conditions. There are several solutions available:

Specify a minimum intra-flow delay (-e flag)

Use several collectors and send them flows in round robin (-n flag) in order to balance load among the collectors.

3. Using nProbe with ntop

On the Internet there are several NetFlow collectors (see Reference paragraph) that can be used to handle flows generated by nProbe. Among them ntop is included. This section explains how to configure ntop to take advantage of nProbe.

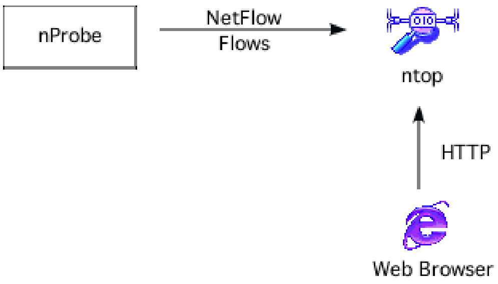


Fig. 1– Using ntop with nProbe

Suppose to run ntop on host which IP address is a.b.c.d and nProbe on host e.f.g.h (note: both ntop and nProbe can run on the same host).

ntop Configuration

From the ntop menu select the entry Plugins, than NetFlow and than the View/Configure link. Note that you have to enable the NetFlow plugin by clicking on the Activate link on the same menu. On the plugin you need to specify the collector port, i.e. the UDP port on which nProbe will send the flows. If you want that ntop displays NetFlow data which it receives from nProbe, you must specify the UDP port to listen to. The default port used for NetFlow is 2055, if you want you can choose this port or another port you like. It is important to explicit set the port and click on the Set Port button. It is done: ntop will now wait for flows sent by nProbe. Remember that ntop will collect flows on a virtual NetFlow interface and not on the default interface. If you want to select this interface you need to go into the Admin menu and select Switch Interface and click on the desired NetFlow interface.

nProbe Configuration

It is very simple to configure nProbe: nprobe –n a.b.c.d:2055 where a.b.c.d is the IP address where ntop is running. If needed, you can add further flags (e.g. –i to select a different interface from which flows are captured).

4. n2disk

N2disk is a network traffic disk recorder application. With n2disk you can capture full-sized network packets at gigabit rate from a live network interface, and write them into files without any packet loss. N2disk has been designed to write files into disks for very long periods, you have to specify a maximum number of distinct file that may be written during the execution, and if n2disk reaches the maximum number of files, it will start recycling the files from the oldest one. On this way you can have a complete view of the traffic for a fixed temporal window, knowing in advance the amount of disk space needed.

n2disk uses the industry standard PCAP file format to dump packets into files so the resulting output can be easily integrated with existing third party or even open/source analysis tools (like Wireshark or nTop).

n2disk has been designed and developed mainly because most network security systems rely on capturing all packets (both header as payload), since any packets may have been responsible for the attack or could contain the problems that we are trying to find. Netflow information is more manageable and requires less disk space to be stored, but in some cases, like deep-packet-inspection analysis or controlled traffic regeneration, it is not useful. When we need to collect the entire packet, because we need all the information, n2disk has to be used.

n2disk can be effectively used to perform numerous activities, among these:

- Off-line network packets analysis by feeding a specialized tools line Snort or nTop
- Reconstruct particular communication flows or network activities
- Reproduce the previous captured traffic to a different network interface

4.1 Main Features

Some of the n2disk features include:

Fully user configurable.

Use of the standard PCAP file format.

High-performance packet to disk recording.

Support for timed disk dumping based on network throughput condition (both Mbits and pps).

BPF filters supports (using the same format as in the popular tcpdump tool) to filter out the unwanted network packets from the recording process.

Multi-core support. n2disk has been designed with multicore architectures in mind. It uses 2 threads: one for the packet capture and one for the disk writing. The communication between the two threads has been carefully optimized.

PF_RING kernel module acceleration. n2disk exploit the packet capture acceleration offered by the PF_RING linux kernel module.

Direct-IO disk access. n2disk uses the Direct IO access to the disks in order to obtain maximum disk-write throughput.

4.2 Usage

In order to save all the traffic into disks, the n2disk application has to be activated on an interface from which it is possible to see/capture all the traffic you are interested in. Once activated, n2dsik will save the traffic data into the specified directory recycling the files already written, starting from the oldest one, this in case the maximum number of created files is reached.

In the following sections, we discuss all the n2disk 1.x command line options and how to efficiently configure n2disk to capture all the traffic flowing in your network.

4.2.1 n2disk Command Line Options

Below the available options and a detailed explanation of each option are listed:

n2disk v.1.0 ($Revision: 962 $)

```
[-v] [-V] [-h] [-i <device>]
[-s <snaplen>] [-f <filter>]
[-b <buffer len>] -o <dir>
-p <pcap file len> [-t <sec>] [-a]
[-m <max files>] [-n <max dirs>]
[-x <file prefix>] [-y <sample rate>] [-r]
Usage:
-v                     | Verbose
-V                     | Print application version
-h                     | Help
-i <device>            | Ingress packet device
-s <snaplen>           | Max packet capture length
-f <filter>            | BPF (tcpdump-like) ingress packet filter
-b <buffer len>        | Buffer length (MBytes)
-o <directory>         | Directory where dump files will be saved
-p <pcap file len>     | Max pcap file length (MBytes)
-t <seconds>           | Max pcap file duration (sec).
                       | Default is 0 that means no max duration
-a                     | Archive pcap file (rename to .old) instead of
          overwriting if already present on disk
 -m <max files>        | Max number of files before restarting file name
-n <max dirs>          | Max number of nested dump sub-directories
-x <file prefix>       | Dump file prefix
-y <sample rate>       | Packet sample rate (e.g. 100 means 1:100 sampling)
-r                     | Disable Direct I/O (experts only)
```

Example:

```
# ./n2disk -o /tmp/dumper -p 10
```

4.3 Tuning n2disk performance

In order to achieve a good n2disk setup able to obtain the maximum performance on a given nBox appliance, it is important to take into account the following aspects.

n2disk uses the PF_RING Linux kernel module accelerator to capture packets from a live network interface, so it is particularly important, especially in case of many small packets per second, to reserve enough ring buffer space inside the kernel. Furthermore, in order to reduce the number of clock-cycles needed to capture the packets and cross the network stack, it is possible to turn off the PF_RING Transparent Mode flag. Please refer to the General configuration section to see how to change these values.

Regarding the n2disk start-up parameters particularly important are the following options:

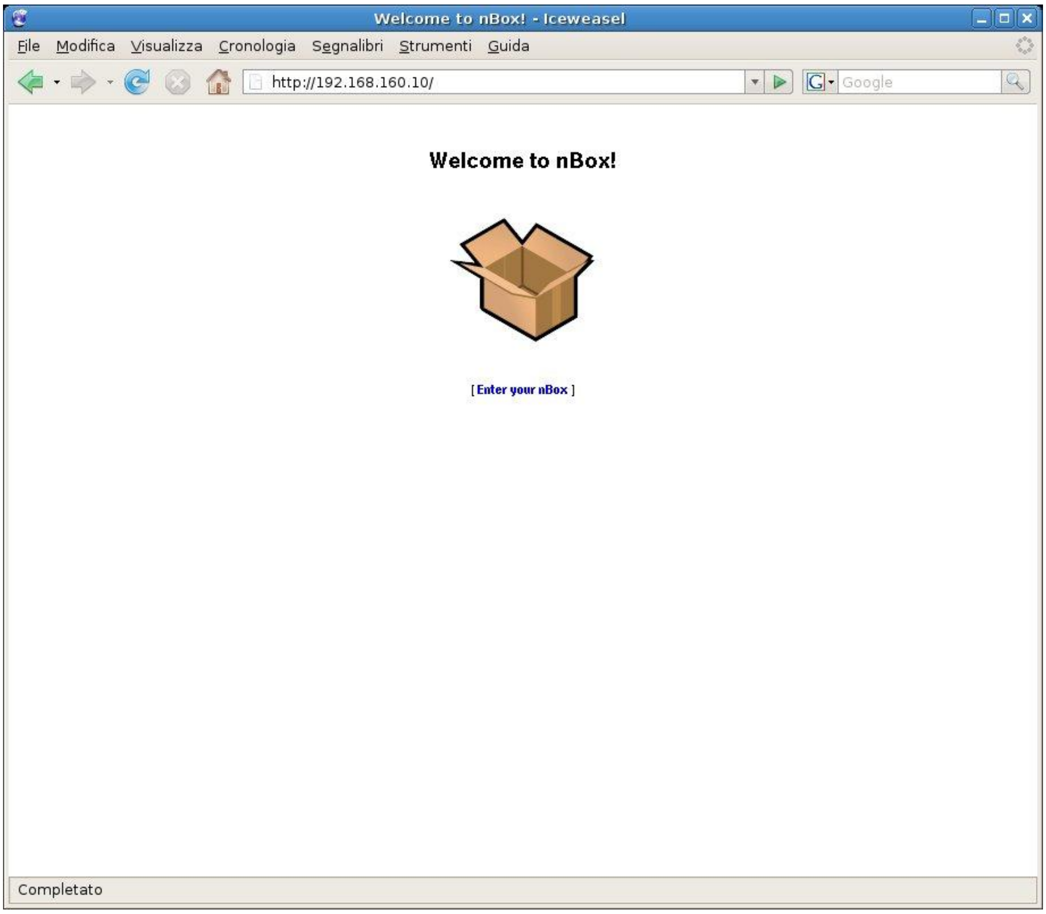
- -b The buffer length has to be big enough. 1 GB is sufficient in most cases.
- -c The write chunk size has to be greater than of equal to 64Kbytes.
- -p.The maximum file size should not be very small. A good value has to be more than 64Mbytes.
- -m The number of files per directory should not be very high.

5. Using the nBox

This section describes how to use the nBox web interface. nBox is a customized version of the Linux operating system, (based on GNU/Linux Debian) modified to use the PF_RING kernel module which acts as a packet capture accelerator. nBox hosts the ntop collector and the nProbe Pro network probe.

5.1 Usage Guidelines

Using the nBox is very simple. Startup the box and connect an Ethernet cable to it. From another PC open a web browser and go to http://192.168.160.10/ (this is the default IP address of your nBox), otherwise use SSH to connect to your box.


Welcome to nBox! - Iceweasel
File Modifica Visualizza Cronologia Segnalibri Strumenti Guida
http://192.168.160.10/
Google
Completato

Welcome to nBox!

[Enter your nBox]

Fig. 2– Initial nBox web page

The default nBox configuration is the following:

1. IP address 192.168.160.10
2. Default SSH user is root with password nbox
3. Default Web user is admin with password nbox

You can change the nBox configuration using the web interface, or if you are a power user using the command line interface.

When you enter the box main page you can find on the left side a list of links divided in three main sections: Configuration, Administration and Diagnostics.

The Configuration section includes the following links:

1. General: general nBox configuration (network, DNS, timezone,etc)
2. Users: add remote and manage web and console users
3. ntop: basic ntop configuration page
4. nProbe: allows to fully configure nProbe on the box
5. n2disk: allows to fully configure n2disk
6. n2n: Network-2-Network VPN configuration
7. SQL Database: configure local or remote SQL database on the box
8. IPMI: basic IPMI information (only if the IPMI card is installed)
9. High-Availability: allows to configure nBox in High-Availability service
10. Firewall: nBox firewall configuration
11. License: allows registering the license or finding out the nBox license code.

The Administration section includes the following links:

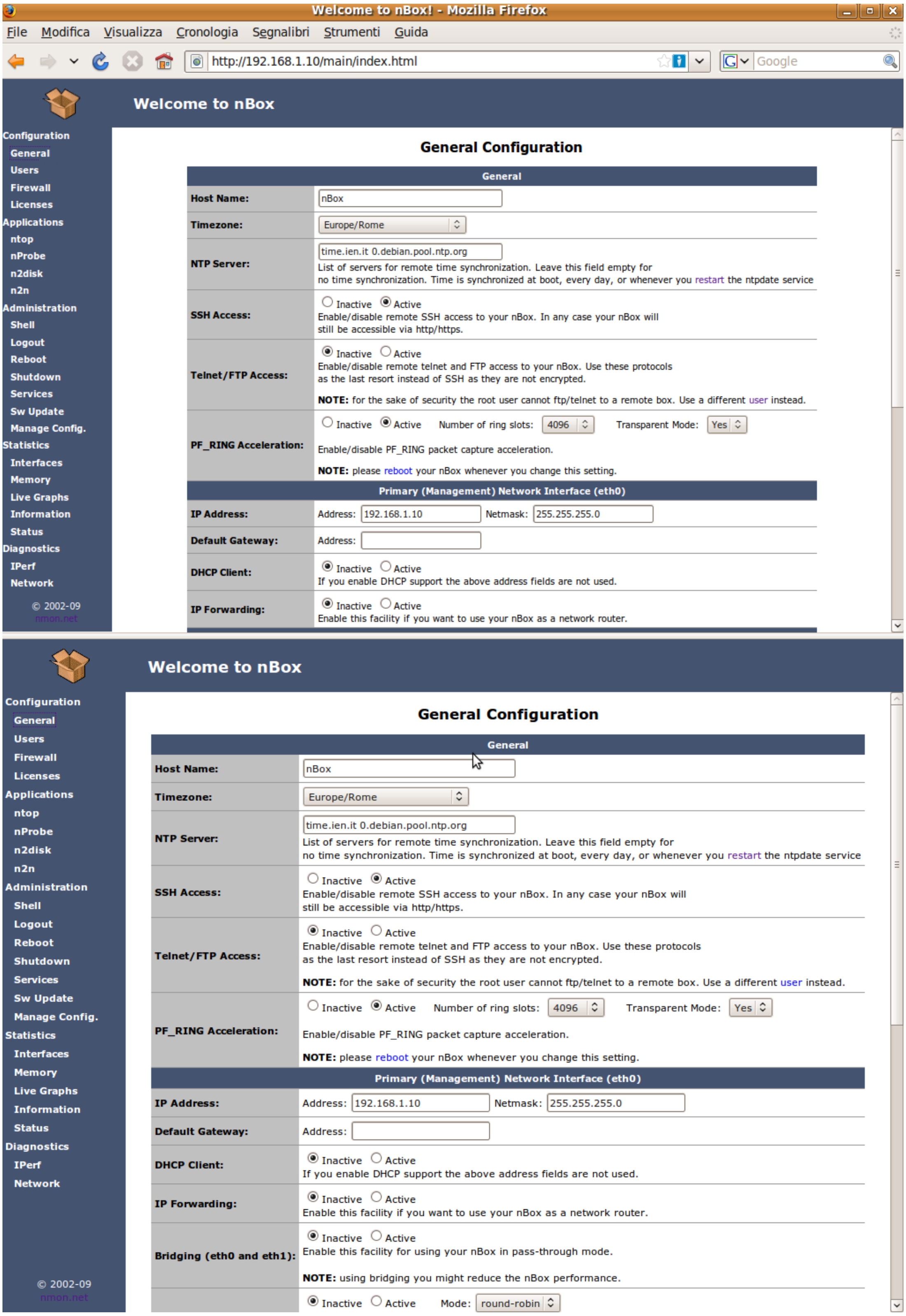
1. Shell: jave-based nBox shell
2. Logout: logout from the web interface
3. Reboot: allows to restart the nBox
4. Shutdown: allows to power-off the nBox
5. Services: allows to start/stop/restart the nBox services
6. Update: packages update
7. Configuration: backup/restore system configuration

The Diagnostics section includes the following links:

1. IPerf: network performance measure tool
2. Network: simple network diagnostic tools
3. Interfaces: network interface information (historical graphs)
4. Memory: shows memory usage (historical graph)
5. Live Graphs: shows cpu and network interfaces utilization
6. Information: report nBox system information
7. Status: allows to download the current status of the system configuration

In the following pages we describe the most important things related to each section.

5.1.1 Configuration: General



General Configuration

General	
Host Name:	nBox
Timezone:	Europe/Rome
NTP Server:	time.ien.it 0.debian.pool.ntp.org List of servers for remote time synchronization. Leave this field empty for no time synchronization. Time is synchronized at boot, every day, or whenever you restart the ntpdate service
SSH Access:	○ Inactive ● Active Enable/disable remote SSH access to your nBox. In any case your nBox will still be accessible via http/https.
Telnet/FTP Access:	● Inactive ○ Active Enable/disable remote telnet and FTP access to your nBox. Use these protocols as the last resort instead of SSH as they are not encrypted. **NOTE:** for the sake of security the root user cannot ftp/telnet to a remote box. Use a different user instead.
PF_RING Acceleration:	○ Inactive ● Active Number of ring slots: 4096 Transparent Mode: Yes Enable/disable PF_RING packet capture acceleration. **NOTE:** please reboot your nBox whenever you change this setting.
Primary (Management) Network Interface (eth0)	
IP Address:	Address: 192.168.1.10 Netmask: 255.255.255.0
Default Gateway:	Address:
DHCP Client:	● Inactive ○ Active If you enable DHCP support the above address fields are not used.
IP Forwarding:	● Inactive ○ Active Enable this facility if you want to use your nBox as a network router.

Welcome to nBox

General Configuration

General	
Host Name:	nBox
Timezone:	Europe/Rome
NTP Server:	time.ien.it 0.debian.pool.ntp.org List of servers for remote time synchronization. Leave this field empty for no time synchronization. Time is synchronized at boot, every day, or whenever you restart the ntpdate service
SSH Access:	○ Inactive ● Active Enable/disable remote SSH access to your nBox. In any case your nBox will still be accessible via http/https.
Telnet/FTP Access:	● Inactive ○ Active Enable/disable remote telnet and FTP access to your nBox. Use these protocols as the last resort instead of SSH as they are not encrypted. **NOTE:** for the sake of security the root user cannot ftp/telnet to a remote box. Use a different user instead.
PF_RING Acceleration:	○ Inactive ● Active Number of ring slots: 4096 Transparent Mode: Yes Enable/disable PF_RING packet capture acceleration. **NOTE:** please reboot your nBox whenever you change this setting.
Primary (Management) Network Interface (eth0)	
IP Address:	Address: 192.168.1.10 Netmask: 255.255.255.0
Default Gateway:	Address:
DHCP Client:	● Inactive ○ Active If you enable DHCP support the above address fields are not used.
IP Forwarding:	● Inactive ○ Active Enable this facility if you want to use your nBox as a network router.

Bridging (eth0 and eth1):	● Inactive ○ Active Enable this facility for using your nBox in pass-through mode. **NOTE:** using bridging you might reduce the nBox performance.
	● Inactive ○ Active Mode: round-robin

Fig. 3– nBox: General Configuration Page

The General configuration page allows you to setup the basic system configuration for your nBox. In particular you can specify and configure the following things:

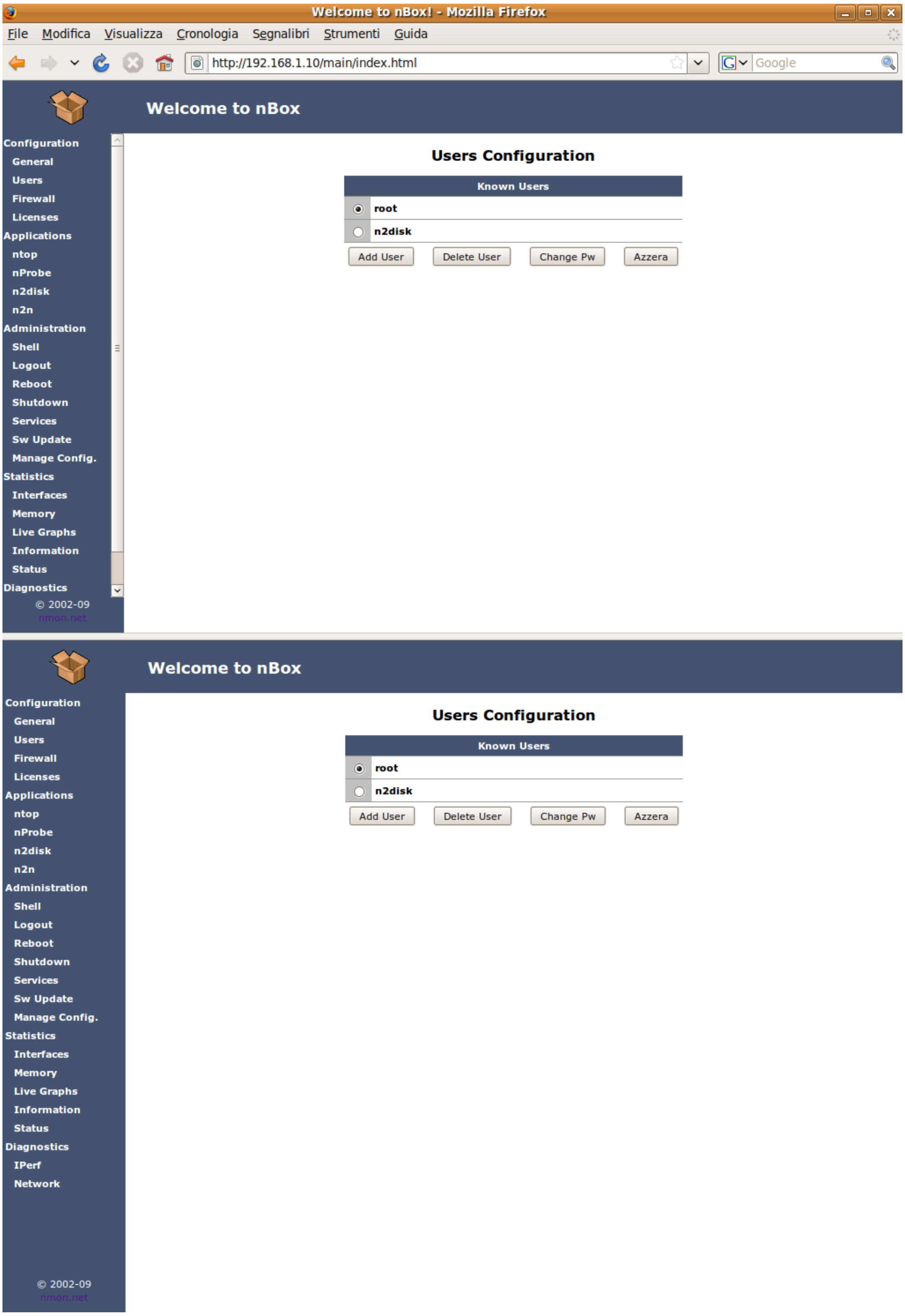
- The host name of the box.
- The time zone.
- A list of NTP servers to use and to keep the box synchronized.
- Enable or disable SSH access (enabled by default).
- Telent/FTP access (both services are disabled by default).
- PF_RING Acceleration (enabled by default).
- The IP address of the management interface (the address can be either a static address or a dynamic one by enabling the DHCP service).
- Enable or disable IP forwarding for the management interface (disabled by default).
- Enable or disable pass-through mode (disabled by default).
- Enable or disable network aggregation also known as Bonding (disabled by default).
- The address for all network interfaces installed on your box other then the management one.
- The primary and secondary DNS service and the nBox domain name
- The remote syslog server

Please refer to paragraph 4.2 to know the "pitfalls" related to the configuration of the nBox in pass-through mode.

In the section PF_RING Acceleration of the General configuration page you can change the number of ring slots in the PF_RING internal buffer and disable the Transparent Mode behaviour. Note that by disabling Transparent Mode, you might improve packet capture performance, but you can only use the interface, on which PF_RING has been enabled, for capturing packets and not for transmitting data.

5.1.2 Configuration: Users

Users (both web and shell) can be added or removed using the Users configuration menu.



Users Configuration

Known Users
● root
○ n2disk

Fig. 4 – nBox: Users Configuration Page

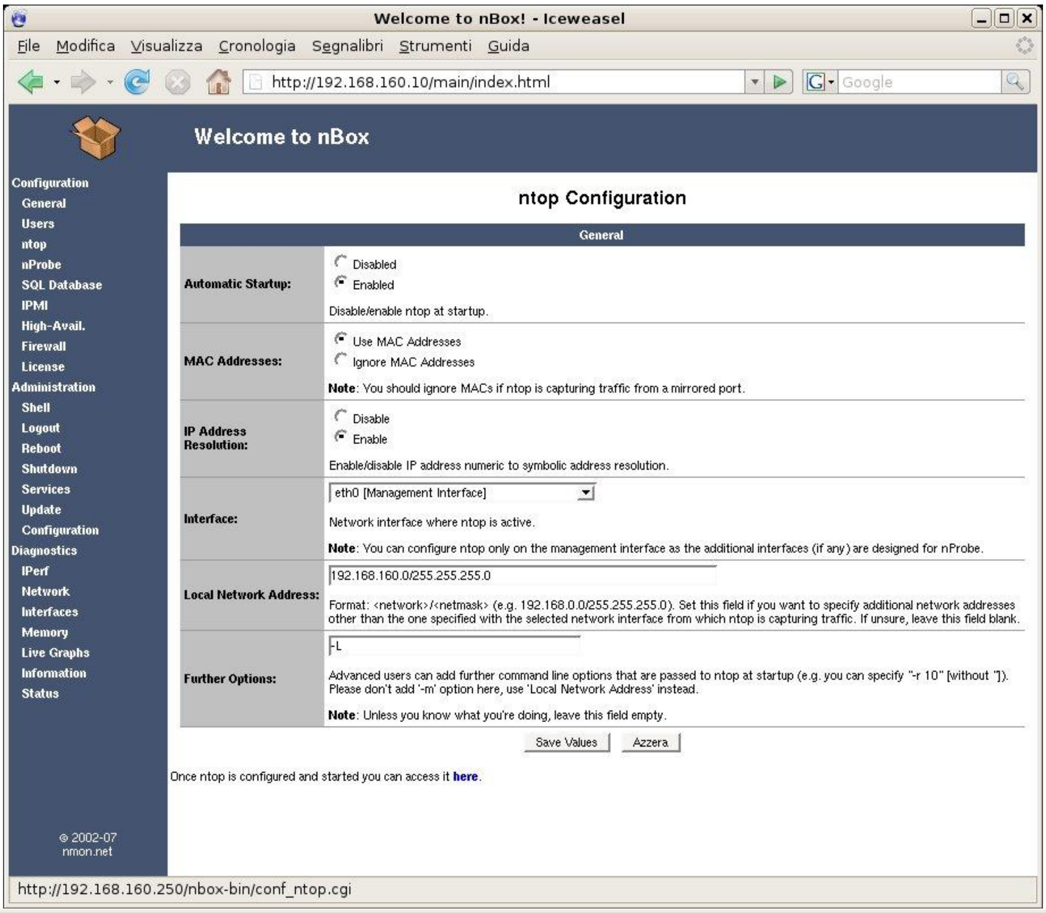
By default there is only the root user enabled in your Box for console or shell access and the admin user enabled for web access.

From the User Configuration Page you can:

- Add a new user
- Remove a user
- Change the password for a user.
- Reset the password for a user (next time the user will access the box he will be prompted to insert a new password).

5.1.3 Configuration: ntop

A simple configuration page for ntop allows you to configure the basic ntop start-up parameters and also access the ntop GUI, if it is enabled and running.



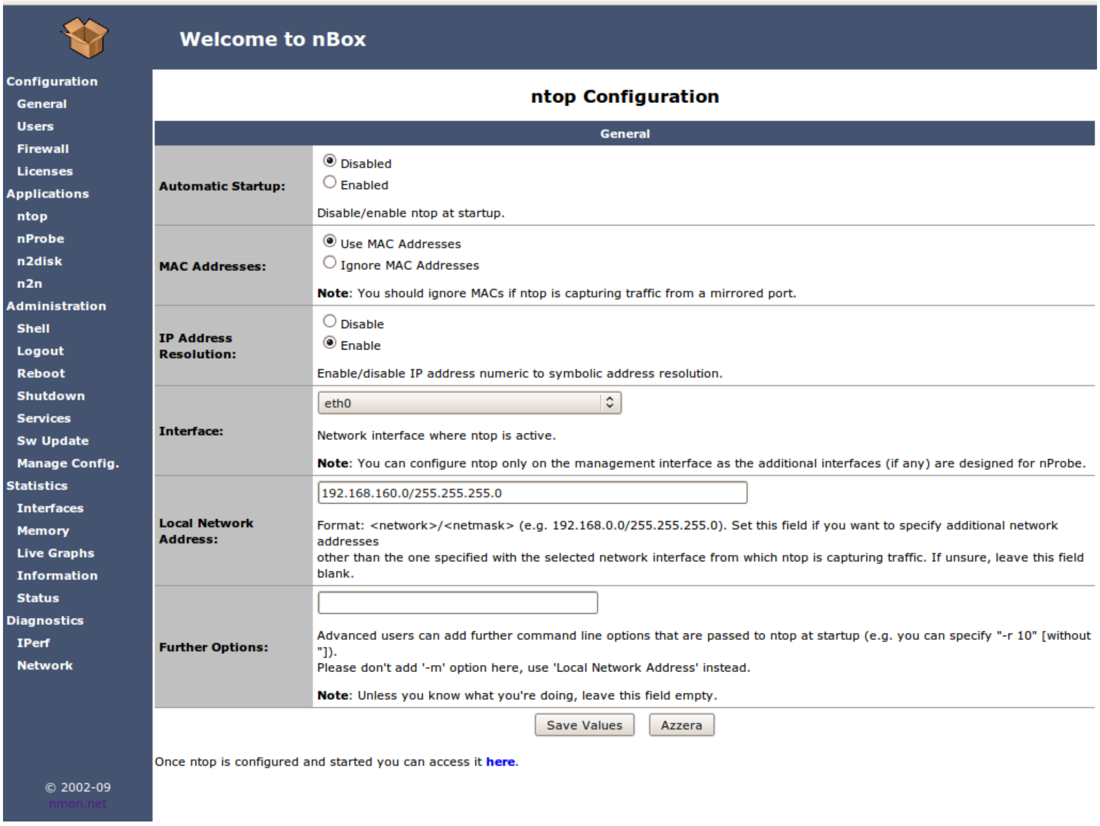
ntop Configuration

General

Automatic Startup:	○ Disabled ● Enabled Disable/enable ntop at startup.
MAC Addresses:	● Use MAC Addresses ○ Ignore MAC Addresses **Note**: You should ignore MACs if ntop is capturing traffic from a mirrored port.
IP Address Resolution:	○ Disable ● Enable Enable/disable IP address numeric to symbolic address resolution.
Interface:	eth0 [Management Interface] Network interface where ntop is active. **Note**: You can configure ntop only on the management interface as the additional interfaces (if any) are designed for nProbe.
Local Network Address:	192.168.160.0/255.255.255.0 Format: <network>/<netmask> (e.g. 192.168.0.0/255.255.255.0). Set this field if you want to specify additional network addresses other than the one specified with the selected network interface from which ntop is capturing traffic. If unsure, leave this field blank.
Further Options:	-L Advanced users can add further command line options that are passed to ntop at startup (e.g. you can specify "-r 10" [without "]). Please don't add '-m' option here, use 'Local Network Address' instead. **Note**: Unless you know what you're doing, leave this field empty.

Save Values | Azzera

Once ntop is configured and started you can access it **here**.



ntop Configuration

General

Automatic Startup:	● Disabled ○ Enabled Disable/enable ntop at startup.
MAC Addresses:	● Use MAC Addresses ○ Ignore MAC Addresses **Note**: You should ignore MACs if ntop is capturing traffic from a mirrored port.
IP Address Resolution:	○ Disable ● Enable Enable/disable IP address numeric to symbolic address resolution.
Interface:	eth0 Network interface where ntop is active. **Note**: You can configure ntop only on the management interface as the additional interfaces (if any) are designed for nProbe.
Local Network Address:	192.168.160.0/255.255.255.0 Format: <network>/<netmask> (e.g. 192.168.0.0/255.255.255.0). Set this field if you want to specify additional network addresses other than the one specified with the selected network interface from which ntop is capturing traffic. If unsure, leave this field blank.
Further Options:	Advanced users can add further command line options that are passed to ntop at startup (e.g. you can specify "-r 10" [without "]). Please don't add '-m' option here, use 'Local Network Address' instead. **Note**: Unless you know what you're doing, leave this field empty.

Save Values | Azzera

Once ntop is configured and started you can access it **here**.

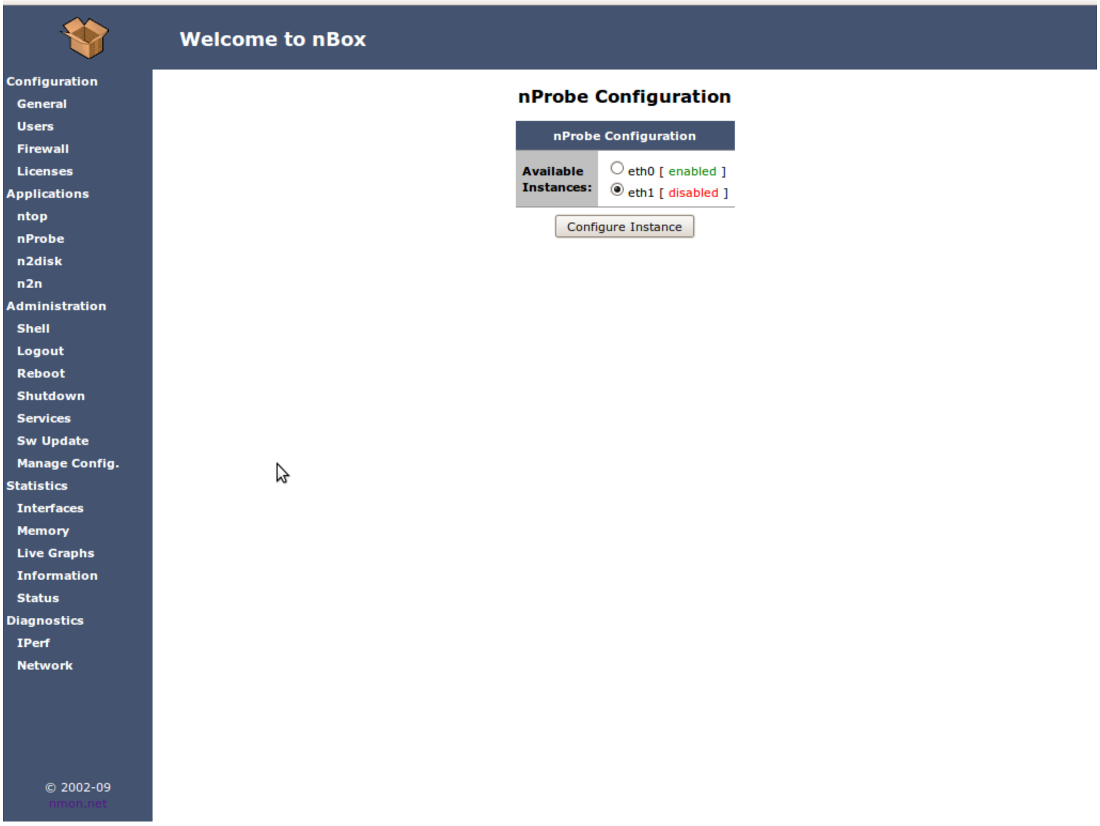
Fig. 5 – nBox: ntop Configuration Page

By default ntop runs at port 3000 of the nBox host.

Note that for fine tuning ntop configuration, you have to use the embedded ntop configuration page.

5.1.4 Configuration: nProbe

The nProbe configuration page allows you to enable and fully configure the probe on one or more network interface of your box.



Welcome to nBox

nProbe Configuration

nProbe Configuration	
Available Instances:	○ eth0 [enabled] ● eth1 [disabled]

Fig. 6 – nBox: nProbe Configuration Page, selection of the network interface.

Once you have selected the network interface where you want nProbe to listen to for incoming packets, you can set, using the web interface, all the nProbe start-up options described in Section 2.3.1. Note that nProbe can be configured on more then one network interfaces.

You can configure nProbe to act like a v5, v9 or IPFIX probe by selecting the radio button in the "NetFlow Version" web page section. By default nProbe is configured to manage v5 flows.

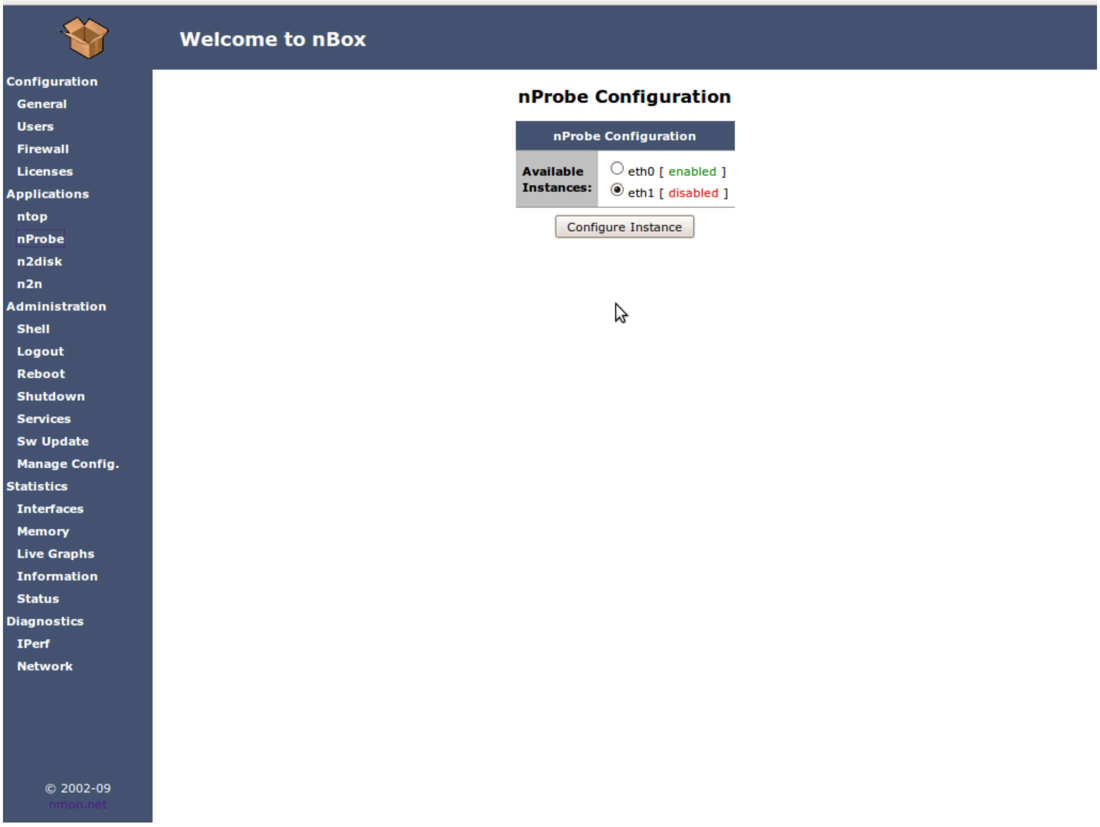
Important notes about ntop usage:

The ntop included into the nBox has been designed mostly as a network debugging tool as the box (usually) lacks of a hard disk hence ntop cannot save persistent data.

In case the box does not have an hard-disk installed, ntop saves temporary data in a ramdisk mounted on the /var/log/ntop directory. If this directory gets full ntop stops as it cannot operate. The default ramdisk size is around 64 Mbytes. If you want to increase its size edit the file /etc/fstab.

If you want that nProbe manages v9 or IPFIX flows you have to check the v9/IPFIX radio-button in the "NetFlow Version" section and than save the configuration. Next you have to return back to the configuration page of the interface that you are configuring and specify all the NetFlow parameters you want to be exported in the v9/IPFIX format.

It is worth to say that not all the NetFlow collectors can handle the v9/IPFIX flows protocol.



nProbe Configuration

nProbe Configuration

Available Instances: ○ eth0 [enabled] ● eth1 [disabled]

Configure Instance

Fig. 7 – nBox: nProbe Configuration Page for the selected network interface.

By default, in the v9 flows export format, nProbe is configured to export the following packet fields:

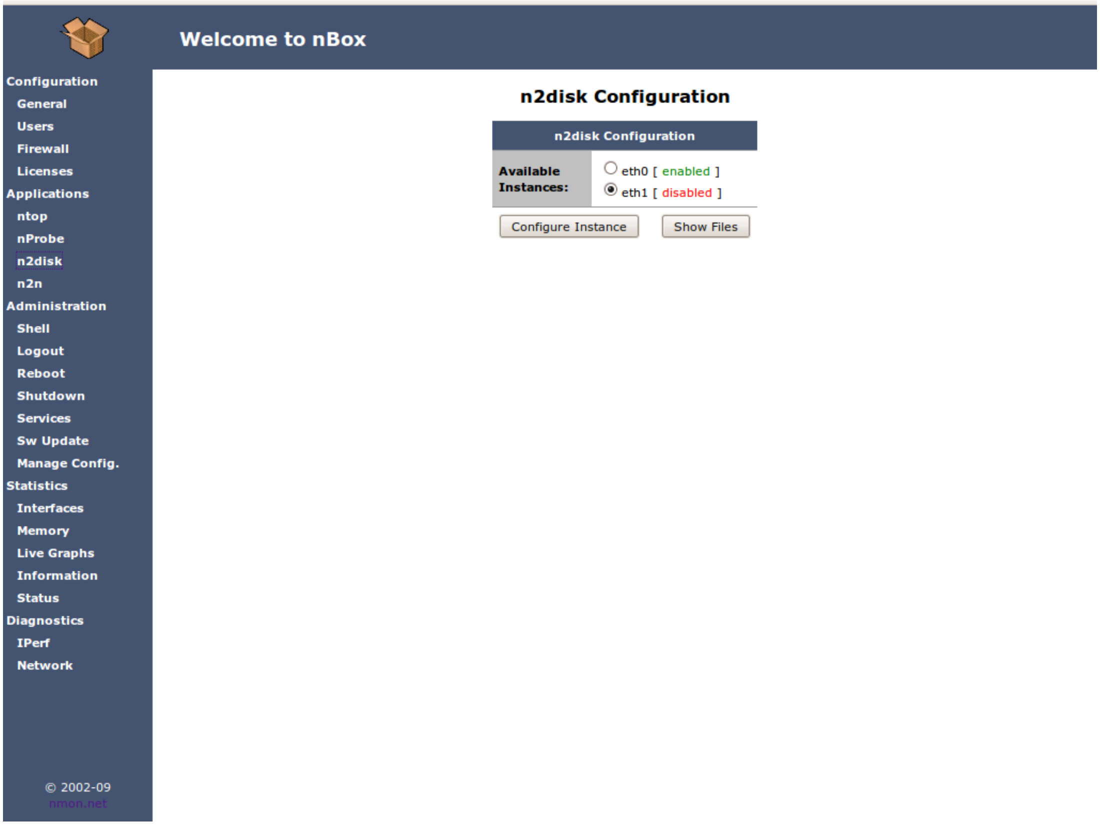
Incoming flow bytes, Incoming flow packets, IP protocol byte, Type of Service byte, Cumulative of all flow TCP flags, IPv4 source port, IPv4 source address, Input Interface SNMP idx, IPv4 destination port, IPv4 destination address, Output Interface SNMP idx, SysUptime (msec.) of the last flow pkt, SysUptime (msec.) of the first flow pkt.

If you want to specify NetFlow v9 flows in a format similar to v5 flows you have to select the following fields:

IPv4 source address, IPv4 destination address, IPv4 next hop address, Input Interface SNMP idx, Output Interface SNMP idx, Incoming flow packets, Incoming flow bytes, SysUptime (msec) of the last flow pkt, SysUptime (msec) of the first flow pkt, IPV4 source port, IPV4 destination port, Cumulative of all flow TCP flags, IP protocol byte, Type of Service byte, Source BGP AS, Dstination BGP AS, Source subnet mask, Dest subnet mask.

5.1.5 Configuration: n2disk

The n2disk configuration page allows you to enable and fully configure the n2disk application in order to capture packets and write them into disks, for one or more network interface of your box.

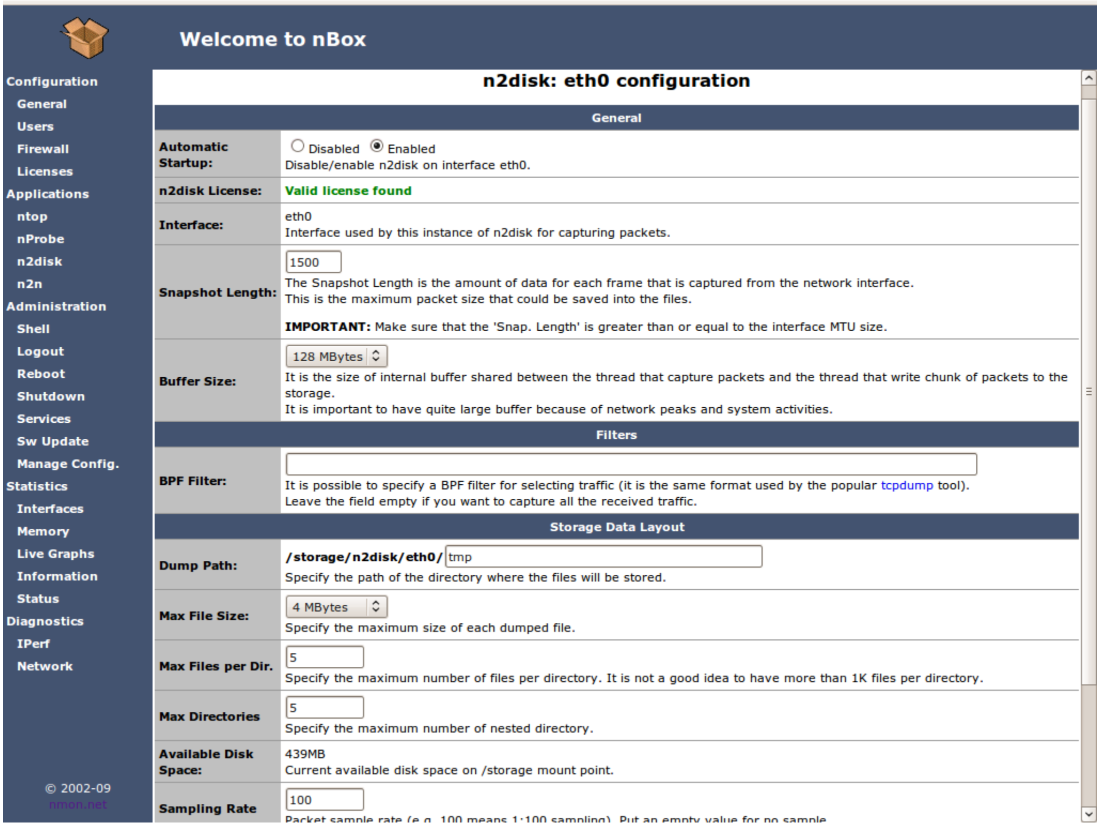


n2disk Configuration

n2disk Configuration	
Available Instances:	○ eth0 [enabled] ● eth1 [disabled]

Fig. 8 – nBox: main n2disk Configuration Page, selection of the network interface.

If you want to configure n2disk on a specific interface, you have to select the interface using the radio button and then click on the 'Configure Instance'. Once you have selected the network interface where you want n2disk to listen to for incoming packets, you can set, all the n2disk start-up options described in Section 4.2.1. Fig. 9 shows a screenshot of the first part of the n2disk configuration page for the interface eth0. Note that n2disk can be activated and configured on more then one network interfaces.



n2disk: eth0 configuration

General	
Automatic Startup:	○ Disabled ● Enabled Disable/enable n2disk on interface eth0.
n2disk License:	**Valid license found**
Interface:	eth0 Interface used by this instance of n2disk for capturing packets.
Snapshot Length:	1500 The Snapshot Length is the amount of data for each frame that is captured from the network interface. This is the maximum packet size that could be saved into the files. **IMPORTANT:** Make sure that the 'Snap. Length' is greater than or equal to the interface MTU size.
Buffer Size:	128 MBytes It is the size of internal buffer shared between the thread that capture packets and the thread that write chunk of packets to the storage. It is important to have quite large buffer because of network peaks and system activities.
Filters	
BPF Filter:	It is possible to specify a BPF filter for selecting traffic (it is the same format used by the popular tcpdump tool). Leave the field empty if you want to capture all the received traffic.
Storage Data Layout	
Dump Path:	**/storage/n2disk/eth0/** tmp Specify the path of the directory where the files will be stored.
Max File Size:	4 MBytes Specify the maximum size of each dumped file.
Max Files per Dir.	5 Specify the maximum number of files per directory. It is not a good idea to have more than 1K files per directory.
Max Directories	5 Specify the maximum number of nested directory.
Available Disk Space:	439MB Current available disk space on /storage mount point.
Sampling Rate	100 Packet sample rate (e.g. 100 means 1:100 sampling). Put an empty value for no sample

Fig. 9 – nBox: n2disk Configuration Page for the selected network interface (eth1).

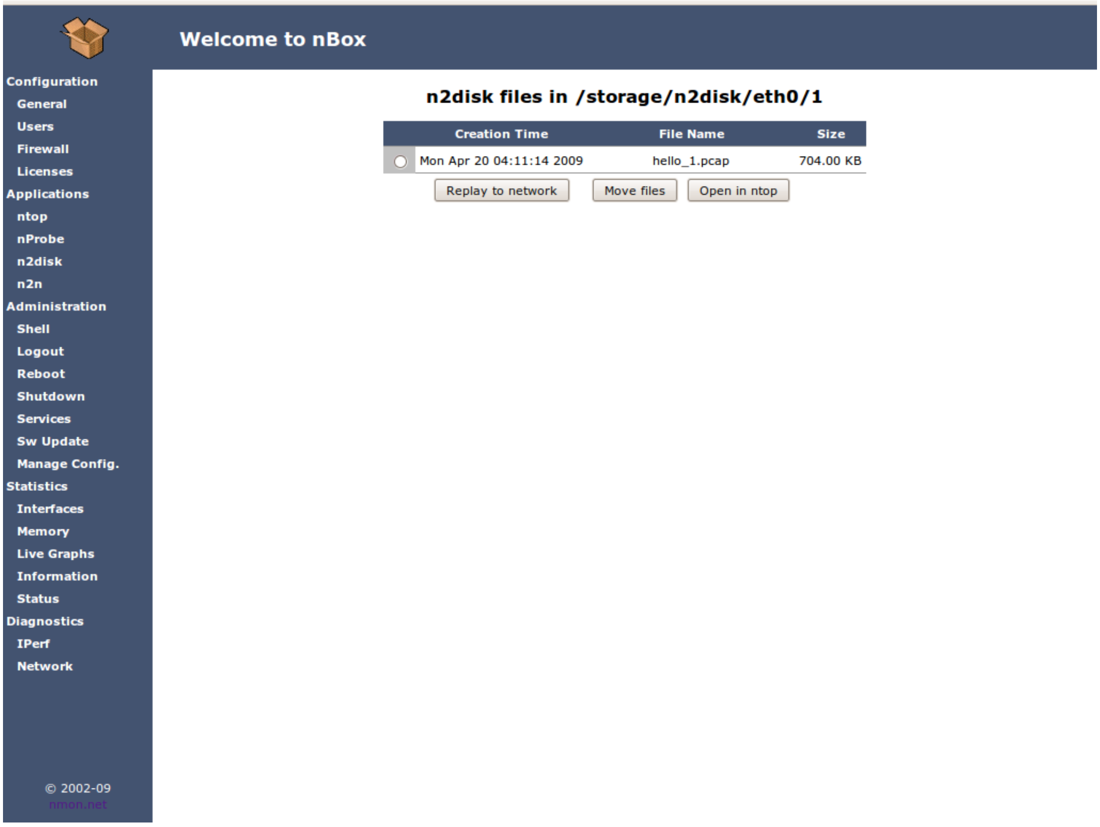
As already said, in the configuration page you can set and tune all the command line options available in the n2disk application. In particular you can enable the dumping only during certain time period and/or when the throughput reaches a specified value. It is worth saying, that if you enable timed dumping and/or dumping on throughput conditions, you can view the n2disk status from the Activity Report web page accessible from the 'Show Stats' button (see Fig. 9). Furthermore, in the configuration, you can specify a set of BPF filters in order to prevent the unwanted network packets to be dumped into files.

All the dumped files are stored in the /storage/n2disk/<network-interface> directory (in the following n2disk root directory) where <network-interface> is the name of the network interface where n2disk is listening to for incoming packets. Inside the n2disk root directory there are a set of subdirectories each of them saves the dumped files.

Particularly important is the snapshot length parameter (Snap. Length). In order to capture the entire packet you have to specify a snapshot length value big enough so that each flowing packets has a length less than, or equal to the snapshot length.To be sure to capture the whole packets you have to specify a length greater than or equal to the MTU size.

Note that using very large snapshot length values reduce the amount of buffering space available.

You can add to each dumped file a Tag string (by default no Tag is specified) just to distinguish some files from the others or because you want to re-start n2disk and do not want to overwrite any previously written files. A particular case is when you want to regenerate some files or you want to open them within nTop and you want to be sure that the files are not overwritten during that operation, so you can change or add a Tag string and restart (or start) n2disk.



n2disk files in /storage/n2disk/eth0/1

	Creation Time	File Name	Size
○	Mon Apr 20 04:11:14 2009	hello_1.pcap	704.00 KB

Fig. 10 – nBox: n2disk Show Files page.

Clicking on the 'Show Files' button on the n2disk page, you can access the Show Files page (Fig. 10) where you can browse the directories created in the n2disk root directory. Each directory may contain, at most, double the number of files specified in the configuration page (if archiving option is enabled). The number of subdirectories inside the root directory, is given by dividing the total number of files allowed by the maximum number of files per directory.

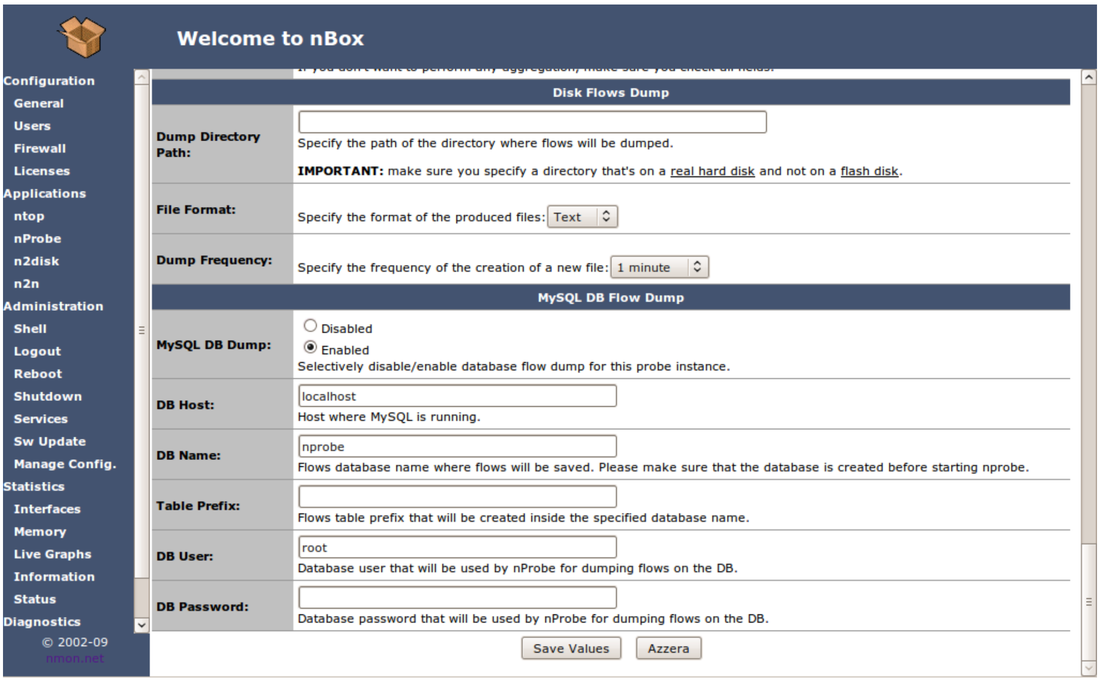
Once you have selected the files you are interested in, you can perform the following operations:

- Open the files within nTop to view packets statistics (Open in nTop)

- Regenerate the packets contained in the selected files (Replay files):into a different network interface using the tcpreplayapplication (Replay into network) generating the NetFlows flows using nProbe (Open with nProbe)
- Move the selected files to a different directory (Movefiles). You can download the files moved into a specified directory by accessing the nBox via FTP or SSH using the user 'n2disk'.

5.1.6 Configuration: SQL Database

It is possible to export nProbe flow in a MySQL database. If you want to do this nProbe stores the flow on a SQL database, but first you have to enable the database service on your nBox.



If you don't want to perform any aggregation, make sure you check all fields.

Disk Flows Dump	
Dump Directory Path:	Specify the path of the directory where flows will be dumped. **IMPORTANT:** make sure you specify a directory that's on a real hard disk and not on a flash disk.
File Format:	Specify the format of the produced files: Text
Dump Frequency:	Specify the frequency of the creation of a new file: 1 minute

MySQL DB Flow Dump	
MySQL DB Dump:	○ Disabled ● Enabled Selectively disable/enable database flow dump for this probe instance.
DB Host:	localhost Host where MySQL is running.
DB Name:	nprobe Flows database name where flows will be saved. Please make sure that the database is created before starting nprobe.
Table Prefix:	Flows table prefix that will be created inside the specified database name.
DB User:	root Database user that will be used by nProbe for dumping flows on the DB.
DB Password:	Database password that will be used by nProbe for dumping flows on the DB.

Fig. 12 – nBox: SQL Database Configuration Page, remote service

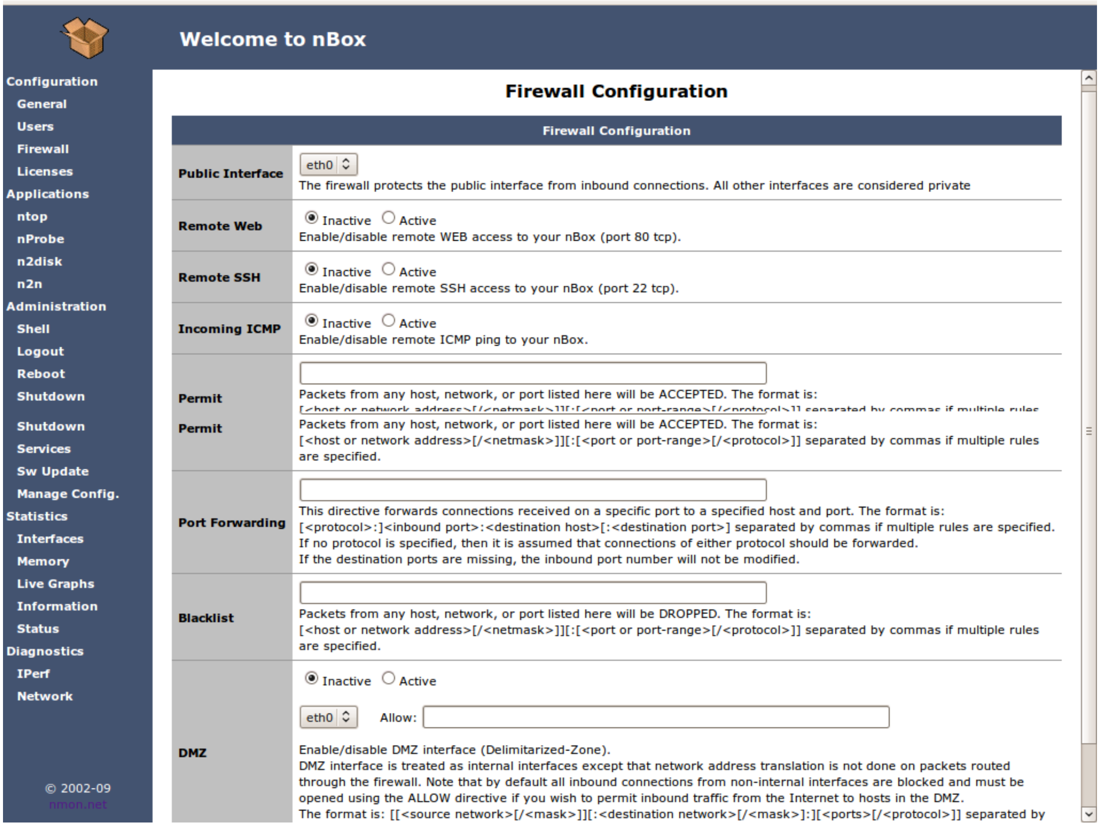
Entering the SQL Database configuration page, you have to choose between two options:

1. Activate MySql on the local box.
2. Configure the parameters to access a remote database service.

In the first case you have to define: database-name, the user and the password to access the database. In the second case, you have to specify also the hostname and the port where the DB service is waiting to receive connections. When you enable the MySQL support in the nProbe configuration page, the database connection settings you specify in the SQL Database page will be used automatically.

5.1.7 Configuration: Firewall

The Firewall link in the left frame allows you to enable or disable a simple firewall to protect access to your nBox.



Firewall Configuration

Firewall Configuration	
Public Interface	eth0 The firewall protects the public interface from inbound connections. All other interfaces are considered private
Remote Web	● Inactive ○ Active Enable/disable remote WEB access to your nBox (port 80 tcp).
Remote SSH	● Inactive ○ Active Enable/disable remote SSH access to your nBox (port 22 tcp).
Incoming ICMP	● Inactive ○ Active Enable/disable remote ICMP ping to your nBox.
Permit	Packets from any host, network, or port listed here will be ACCEPTED. The format is: [<host or network address>[/<netmask>]][:[<port or port-range>[/<protocol>]] separated by commas if multiple rules
Permit	Packets from any host, network, or port listed here will be ACCEPTED. The format is: [<host or network address>[/<netmask>]][:[<port or port-range>[/<protocol>]] separated by commas if multiple rules are specified.
Port Forwarding	This directive forwards connections received on a specific port to a specified host and port. The format is: [<protocol>:]<inbound port>:<destination host>[:<destination port>] separated by commas if multiple rules are specified. If no protocol is specified, then it is assumed that connections of either protocol should be forwarded. If the destination ports are missing, the inbound port number will not be modified.
Blacklist	Packets from any host, network, or port listed here will be DROPPED. The format is: [<host or network address>[/<netmask>]][:[<port or port-range>[/<protocol>]] separated by commas if multiple rules are specified.
DMZ	● Inactive ○ Active eth0 Allow: Enable/disable DMZ interface (Delimitarized-Zone). DMZ interface is treated as internal interfaces except that network address translation is not done on packets routed through the firewall. Note that by default all inbound connections from non-internal interfaces are blocked and must be opened using the ALLOW directive if you wish to permit inbound traffic from the Internet to hosts in the DMZ. The format is: [[<source network>[/<mask>]][:<destination network>[/<mask>]:][<ports>[/<protocol>]] separated by

Fig. 13 – nBox: Firewall Configuration Page

Once you have enabled the firewall you can access to the Firewall configuration page where you can specify the following settings:

- Public network interface. This is the interface which is protected by the firewall. You want to restrict access on that interface.
- Enable/Disable remote web access (port 80). If it is disabled you cannot manage to access to the nBox web GUI from the public network interface.
- Enable/Disable SSH access (port 22). If it is disabled you can not manage to access the nBox via any ssh clients from the public network interface.
- Enable/Disable incoming ICMP. The box doesn't reply to ICMP requests.
- Permit. Allows you to specify a list of networks or hosts from which allpackets will be ACCEPTED.
- Port Forwarding. Allows you to forward connections to a port to a specific host and port.

- Blacklist. Allows you to specify a list of networks or hosts from which all packets will be DROPPED.
- DMZ. Allows you to configure a Demilitarized Zone on a specific interface.

If you want to know the rules added by the firewall you can connect with a shell to the box as root user and type the command iptables-save. By redirecting the output to a file, you can modify the rules and than reload the whole rule by typing:cat filename | iptables-restore.

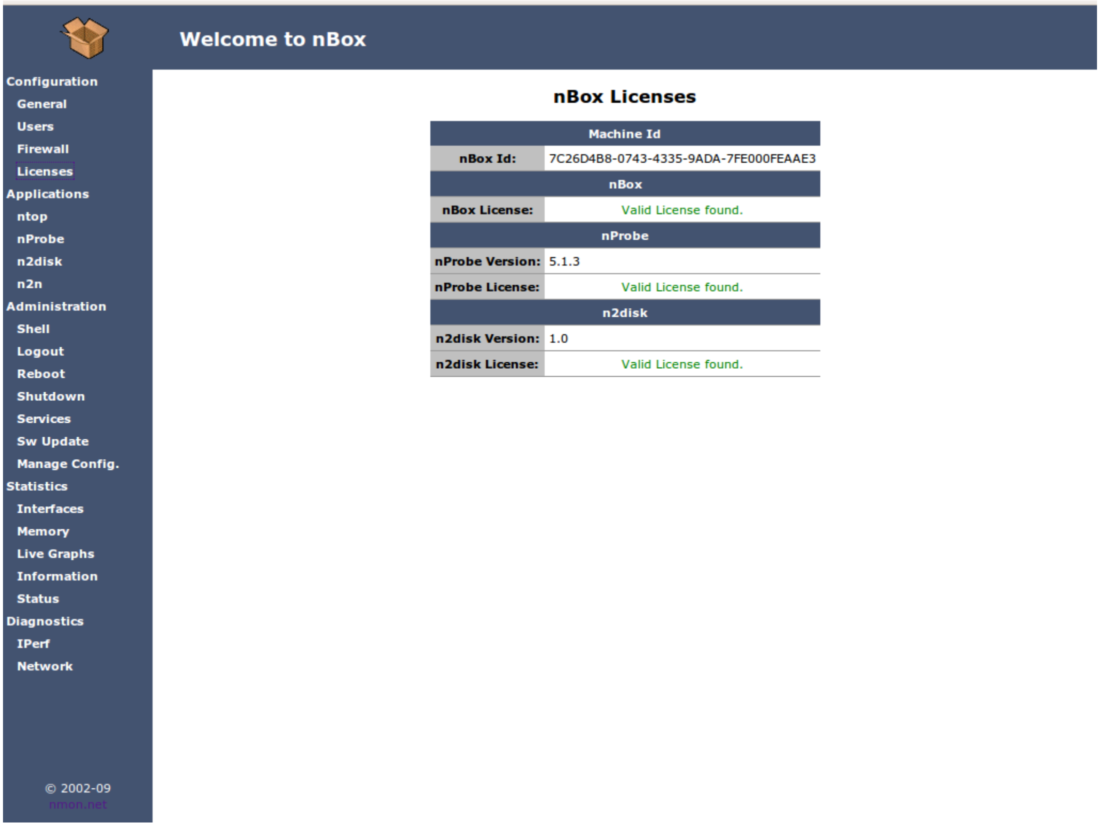
In he following we have reported the rules added by the firewall in case where only SSH and WEB access is allowed from the public network interface eth0.

```
nBox:~# iptables-save
*mangle
:PREROUTING ACCEPT [40:2740]
:INPUT ACCEPT [39:2688]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [21:2253]
:POSTROUTING ACCEPT [21:2253]
COMMIT
*nat
:PREROUTING ACCEPT [1:100]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o eth0 -j SNAT –to-source 192.168.160.10
COMMIT
*filter
:INPUT DROP [8:452]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [24:2585]
:TRUSTED - [0:0]
-A INPUT -m state –state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m state –state NEW -m tcp –dport 80 -j ACCEPT
-A INPUT -p tcp -m state –state NEW -m tcp –dport 22 -j ACCEPT
-A INPUT -i eth1 -m state –state NEW -j ACCEPT
-A INPUT -i eth1 -p udp -m state –state NEW -m udp –dport 67 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state –state NEW -j TRUSTED
-A FORWARD -m state –state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i eth1 -m state –state NEW -j ACCEPT
-A FORWARD -m state –state NEW -j TRUSTED
-A OUTPUT -o eth1 -p icmp -j ACCEPT
-A OUTPUT -p icmp -m state –state INVALID -j DROP
-A TRUSTED -j DROP
COMMIT
```

5.1.8 Configuration: License

In order to use all the nBox features, the nBox must have a valid license. A valid license is required:

- To the new nProbe 5.x Pro releases to run (either nProbe and n2disk).
- To have access to the nBox repository in order to get updates via Internet (see Configuration: Update section for details).



nBox Licenses

Machine Id	
nBox Id:	7C26D4B8-0743-4335-9ADA-7FE000FEAAE3
nBox	
nBox License:	Valid License found.
nProbe	
nProbe Version:	5.1.3
nProbe License:	Valid License found.
n2disk	
n2disk Version:	1.0
n2disk License:	Valid License found.

Fig. 14 – nBox: License Configuration Page

The license codes are written in the /etc/ directory and different text files are used for each service (nbox.license, nprobe.license, n2disk.license). When nProbe starts, it searches for its license file in the /etc directory. If nProbe does not find any valid license or the license code is invalid it won't start.

The license file (and by consequent the flash card) could not be moved from one nBox to another one because the license code is strictly tied to the underling machine.

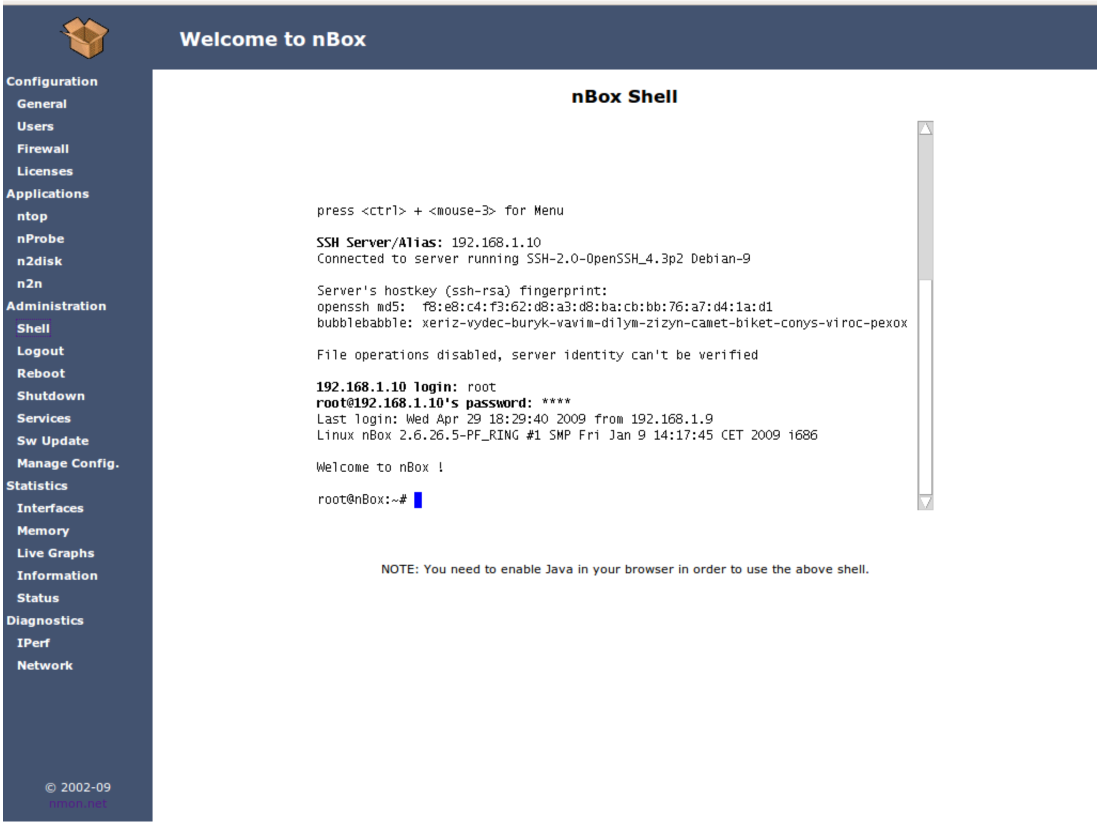
A license never expires so you can run nProbe 5.x Pro without any limitation. On the other hand, in order to get continuous package updates from the nBox Internet repository (www.nmon.net/update), you need to renew the license year-by-year.

If for some reason you loose the license file, you can send the nBox serial-id via e-mail to the mailbox nbox@nmon.net asking for the license code. The nBox serial-id can be retrieved using the Information page (nBox-id field) or by typing in the nBox console the following command as root user :license -s .

Everytime a new version of the nProbe package is released, each license code (nProbe and n2disk) has to be updated in order to keep the these features fully functional.

5.1.9 Administration: Shell

As not everybody has an SSH client installed, the web interface contains a Java applet that implements an SSH client that allows you to access the nBox via SSH without the need to use an SSH client.



nBox Shell

```
press <ctrl> + <mouse-3> for Menu

SSH Server/Alias: 192.168.1.10
Connected to server running SSH-2.0-OpenSSH_4.3p2 Debian-9

Server's hostkey (ssh-rsa) fingerprint:
openssh md5:  f8:e8:c4:f3:62:d8:a3:d8:ba:cb:bb:76:a7:d4:1a:d1
bubblebabble: xeriz-vydec-buryk-vavim-dilym-zizyn-camet-biket-conys-viroc-pexox

File operations disabled, server identity can't be verified

192.168.1.10 login: root
root@192.168.1.10's password: ****
Last login: Wed Apr 29 18:29:40 2009 from 192.168.1.9
Linux nBox 2.6.26.5-PF_RING #1 SMP Fri Jan 9 14:17:45 CET 2009 i686

Welcome to nBox !

root@nBox:~#
```

NOTE: You need to enable Java in your browser in order to use the above shell.

Fig. 15 – nBox: nBox Java-based SSH Shell

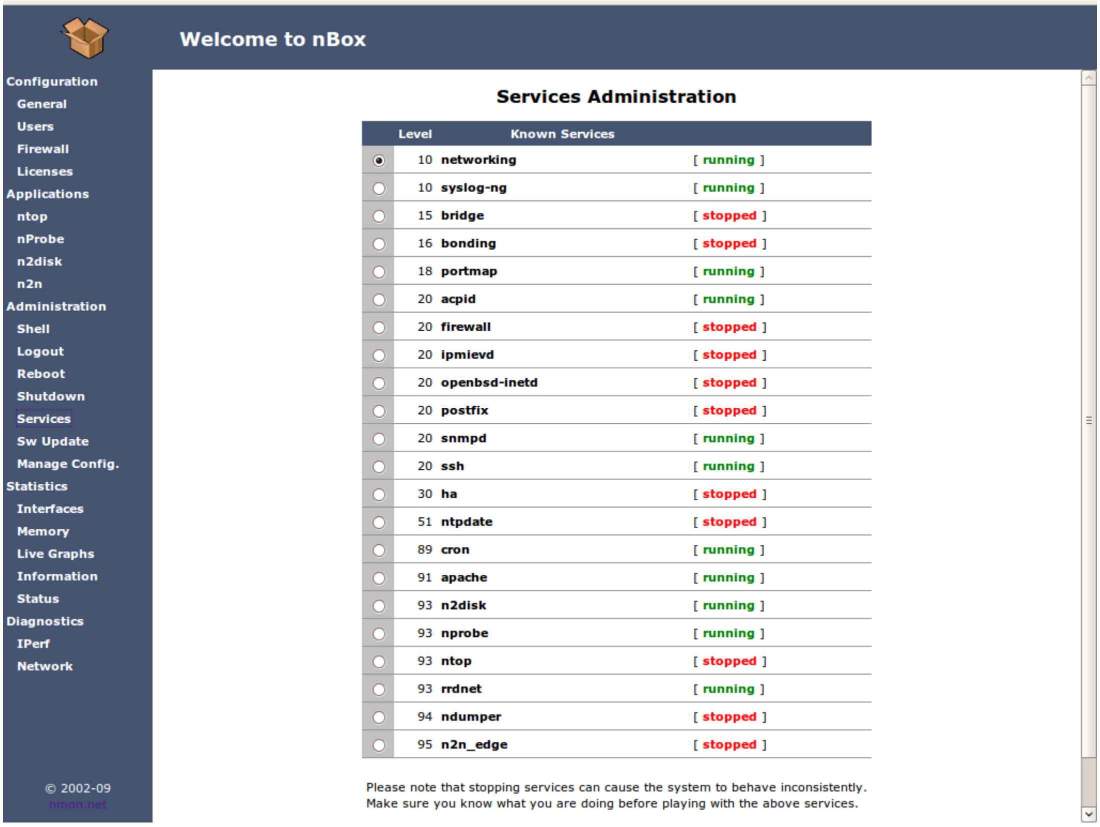
In order to use the java-based SSH client you have to enable Java plugins in your browser.

Important notes about DB usage:

It is safe to enable the DB services locally on the box only if you have an nBox with a hard-disk on-board because of the limited size of the flash-disk. The database root directory is /storage/mysql. If this directory gets full the system could hang.

5.1.10 Administration: Services

The available services can be managed from the web interface by selecting the radio-button corresponding to the service you want to manage and clicking on the buttons below (start, stop, restart).



Services Administration

	Level	Known Services	
●	10	networking	[running]
○	10	syslog-ng	[running]
○	15	bridge	[stopped]
○	16	bonding	[stopped]
○	18	portmap	[running]
○	20	acpid	[running]
○	20	firewall	[stopped]
○	20	ipmievd	[stopped]
○	20	openbsd-inetd	[stopped]
○	20	postfix	[stopped]
○	20	snmpd	[running]
○	20	ssh	[running]
○	30	ha	[stopped]
○	51	ntpdate	[stopped]
○	89	cron	[running]
○	91	apache	[running]
○	93	n2disk	[running]
○	93	nprobe	[running]
○	93	ntop	[stopped]
○	93	rrdnet	[running]
○	94	ndumper	[stopped]
○	95	n2n_edge	[stopped]

Please note that stopping services can cause the system to behave inconsistently.
Make sure you know what you are doing before playing with the above services.

Fig. 16 – nBox: Service Management Page

nProbe and ntop start with a watchdog program named runsv. If for some reason they crash, runsv restarts the service in a few seconds.

Note that the services are shown in running state only if the corresponding configuration has been enabled and if the service has been started. If you try to start or restart a disabled service it won't start.

rrdnet is a internal service that is enabled by default. The rrdnet service produces the data needed to display the network and memory usage graphs like those shown in the Memory page and in the Interfaces page respectively. If you disable the rrdnet service, the only effect is that the graphs aforementioned won't be displayed.

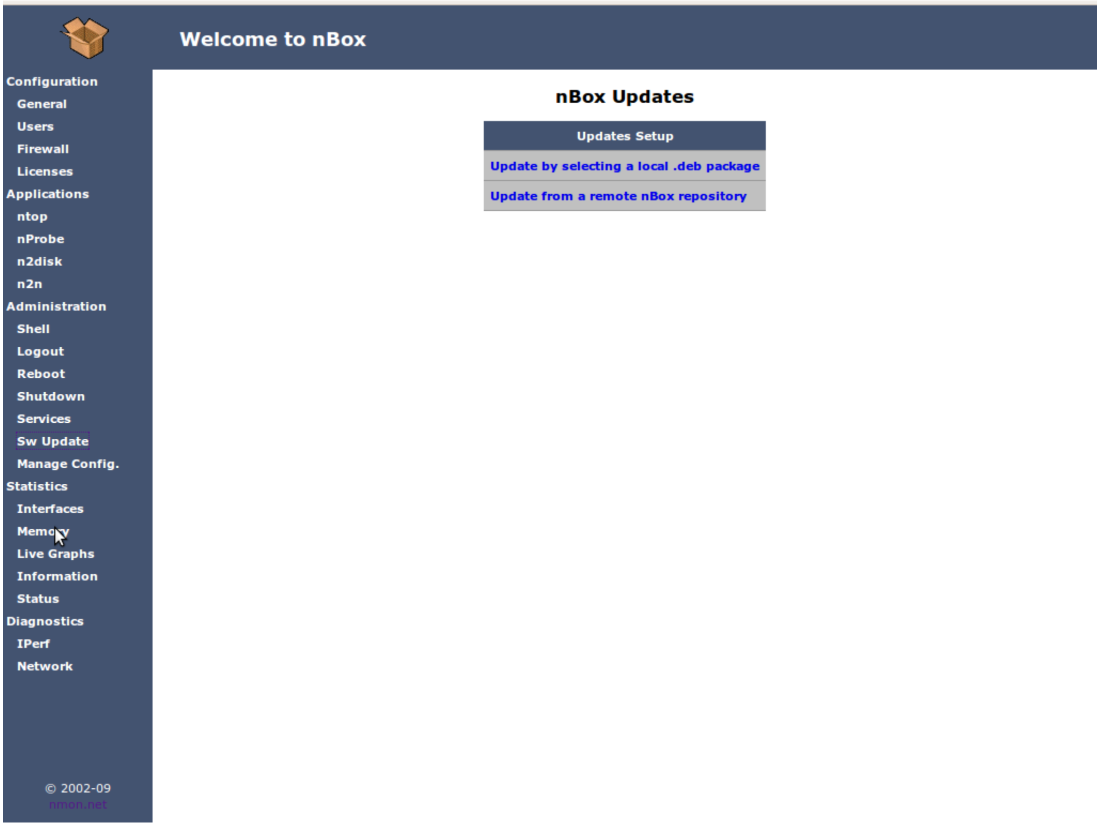
5.1.11. Administration: Update

The nBox firmware has been designed to be updated from remote without the need to override the current firmware image.

This facility allows the nBox administrator to remotely update the system without the need to be physically connected to the box.

There are two main possibilities to upgrade the nBox, both available from the Update page:

1. Upload and install on the box a single .deb package from a PC that can reach the nBox.
2. Uses the remote nBox repository to automatically download and install updates in your box.



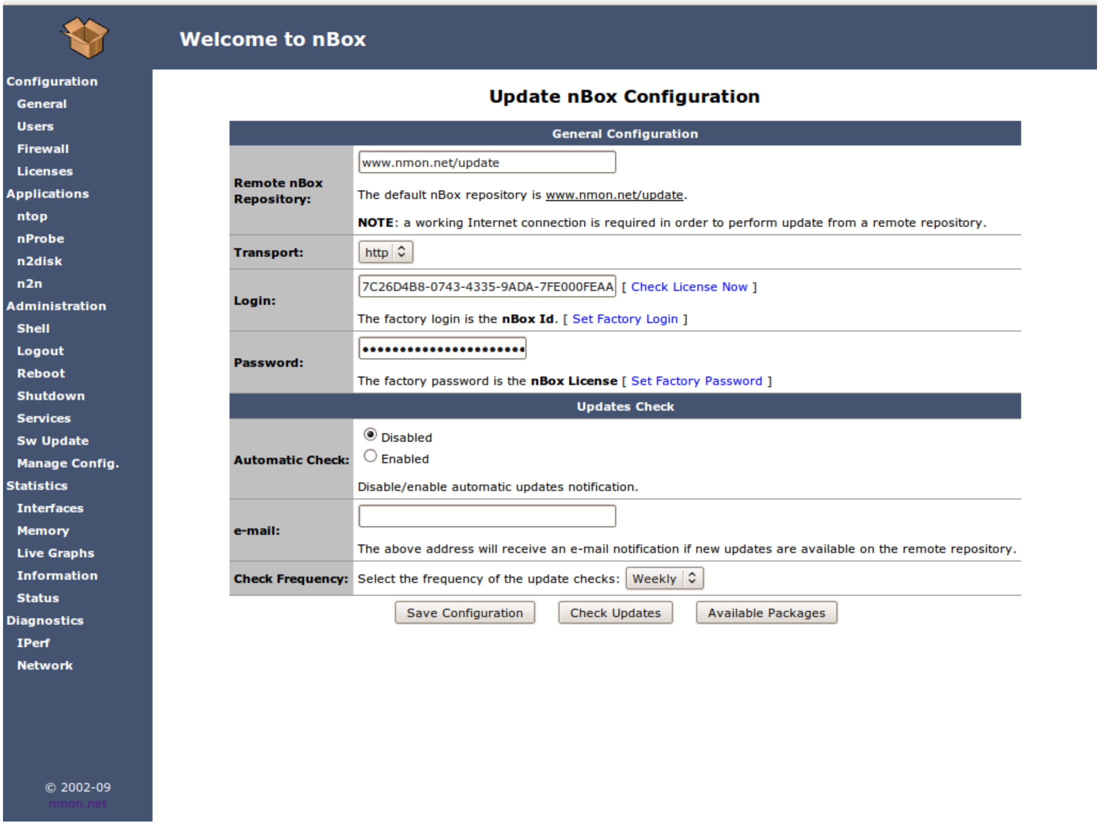
nBox Updates

Updates Setup
Update by selecting a local .deb package
Update from a remote nBox repository

Fig. 17 – nBox: Update Administration Page, local vs. remote update

In the first case you already have a Debian package (.deb) and you want to install or upgrade the package on your box, so you have to click on the link "Update by selecting a local .deb package" from the Update web page. The package will be uploaded on the box and the command dpkg -i <package.deb> will be executed.

In the second case, you have to select the "Update from a remote nBox repository" link. By default the main nBox repository address (www.nmon.net/update)has been already set, and the default login and password corresponding respectively with your nBox-id and nBox license code.



Update nBox Configuration

General Configuration

Remote nBox Repository: www.nmon.net/update

The default nBox repository is www.nmon.net/update.

NOTE: a working Internet connection is required in order to perform update from a remote repository.

Transport: http

Login: 7C26D4B8-0743-4335-9ADA-7FE000FEAA [Check License Now]

The factory login is the **nBox Id**. [Set Factory Login]

Password: ••••••••••••••••••••••

The factory password is the **nBox License** [Set Factory Password]

Updates Check

Automatic Check: ● Disabled ○ Enabled

Disable/enable automatic updates notification.

e-mail:

The above address will receive an e-mail notification if new updates are available on the remote repository.

Check Frequency: Select the frequency of the update checks: Weekly

Fig. 18 – nBox: Update Administration Page: remote update

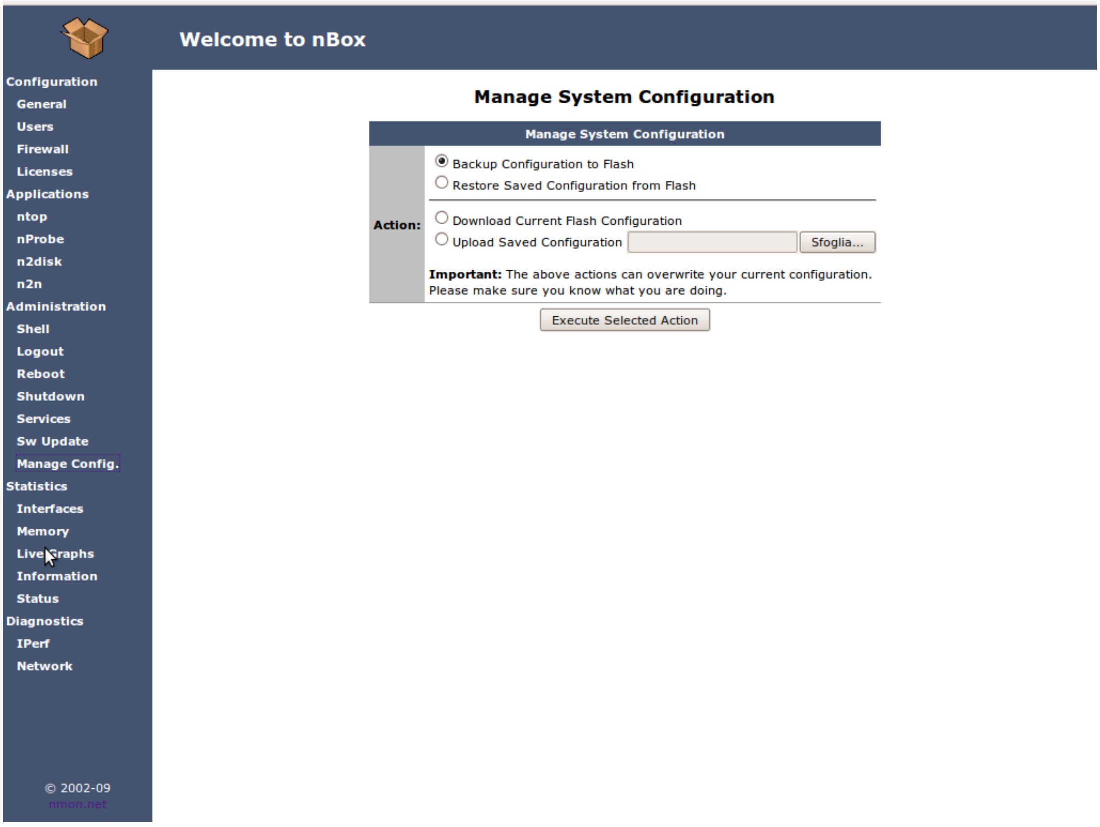
If for any reason you change or delete the login and password fields, you can restore both fields by clicking on the link "Set Factory Login" and "Set Factory Password".

The "Check Updates" button allows you to verify if there are updates available for your system. The nBox system will try to connect (using the http protocol) to the remote repository, with the login and password you provided, searching for updates. If there are new packages available, a list of the packages will be prompted and you can choose which packages to upgrade selectively.

It is also possible to be automatically notified when there will be updates available in the remote repository. You just have to enable the "Automatic Check" radio button and enter a valid e-mail address. When there will be updates available in the remote repository, you will receive an e-mail to the mailbox specified, with a list of available new packages. Note that in this case the system does not automatically install the new packages, so if you want to download and install the updates you have to use the "Check Updates" or "Available Packages" button.

5.1.12 Administration: Configuration

As all boxes, even the nBox can break, the web interface also allows people to backup the box configuration to a secondary flash partition or to the PC where the web browser is running.



Manage System Configuration

Manage System Configuration

Action:
● Backup Configuration to Flash
○ Restore Saved Configuration from Flash
○ Download Current Flash Configuration
○ Upload Saved Configuration [ ] Sfoglia...

Important: The above actions can overwrite your current configuration. Please make sure you know what you are doing.

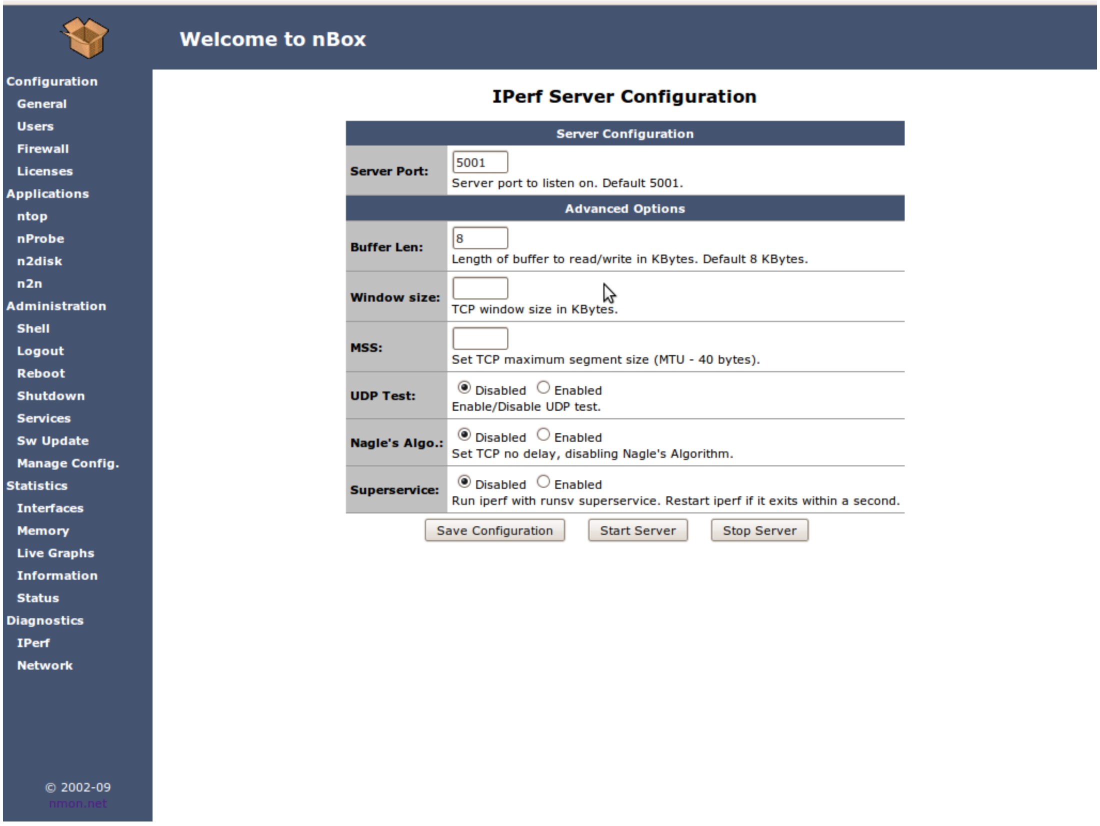
Execute Selected Action

Fig. 19 – nBox: Box Configuration Management Page

You can choose to backup and restore your box so that in case of hardware failure you can reinstall the box (after having replaced the broken hardware component) and restore the whole configuration with one click, simply uploading the configuration file you previously downloaded. This facility allows you to also rapidly clone boxes in case you need to build several boxes with the very same configuration. Note that in the latter case you have to ask the nBox team for a new nBox license id (see section 4.1.7 for more information about the nBox license).

5.1.13 Diagnostics: IPerf

IPerf is a tool to measure TCP and UDP network bandwidth performance. nBox has the IPerf software already installed and provides the user with a web page to configure the IPerf command line parameters and runs the program.



IPerf Server Configuration

Server Configuration	
Server Port:	5001 Server port to listen on. Default 5001.
Advanced Options	
Buffer Len:	8 Length of buffer to read/write in KBytes. Default 8 KBytes.
Window size:	 TCP window size in KBytes.
MSS:	 Set TCP maximum segment size (MTU - 40 bytes).
UDP Test:	● Disabled ○ Enabled Enable/Disable UDP test.
Nagle's Algo.:	● Disabled ○ Enabled Set TCP no delay, disabling Nagle's Algorithm.
Superservice:	● Disabled ○ Enabled Run iperf with runsv superservice. Restart iperf if it exits within a second.

Fig. 20 – nBox: IPerf Diagnostics Page

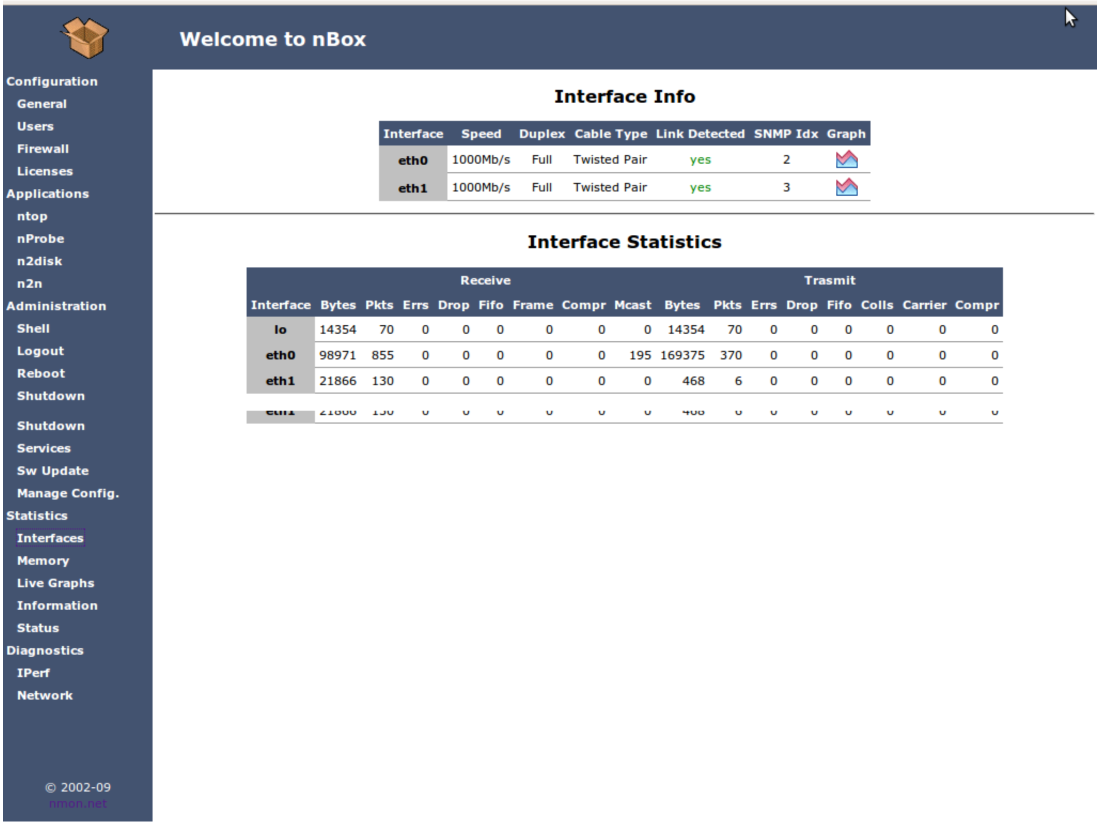
You can run IPerf both as client and server mode. Depending on the run mode, different options can be specified. In the server mode you can run IPerf as a continuous running service using the watchdog program runsv which is able to restart the IPerf process, within a few seconds, in case it crashes for any reasons.

Note that IPerf is not an nBox service, so if the system is powered off or rebooted, IPerf must be manually started.

Please refer back to the IPerf manual (see the Reference chapter) to know all the IPerf options details.

5.1.14 Diagnostics: Interfaces

If you want to take a look at the status of your network interfaces, you can use the Interfaces page.



Interface Info

Interface	Speed	Duplex	Cable Type	Link Detected	SNMP Idx	Graph
eth0	1000Mb/s	Full	Twisted Pair	yes	2	
eth1	1000Mb/s	Full	Twisted Pair	yes	3	

Interface Statistics

	Receive								Trasmit							
Interface	Bytes	Pkts	Errs	Drop	Fifo	Frame	Compr	Mcast	Bytes	Pkts	Errs	Drop	Fifo	Colls	Carrier	Compr
lo	14354	70	0	0	0	0	0	0	14354	70	0	0	0	0	0	0
eth0	98971	855	0	0	0	0	0	195	169375	370	0	0	0	0	0	0
eth1	21866	130	0	0	0	0	0	0	468	6	0	0	0	0	0	0

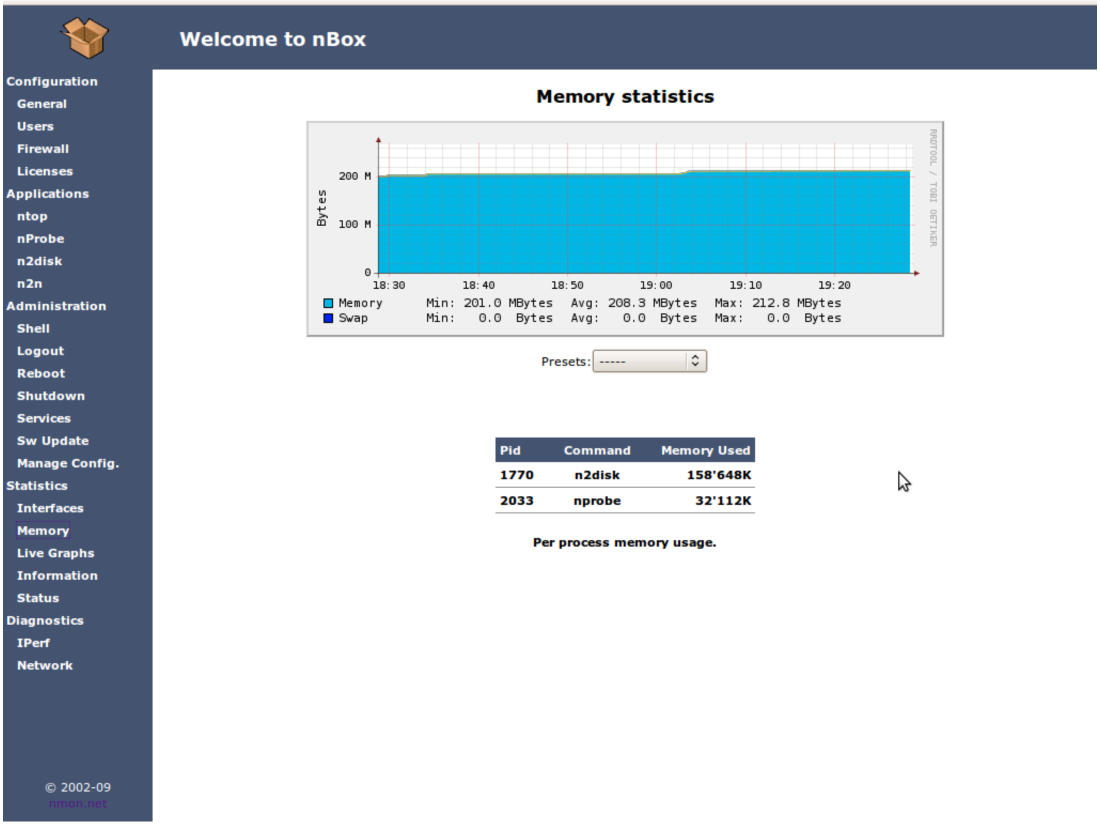
Fig. 21 – nBox: Interfaces Diagnostics Page

From the Interface page you can retrieve the SNMP idx parameters that you may need to use with nProbe (-u and -Q command line options).

If you want to know the throughput history of the traffic flowing through a specific network interface, you can just click on the graph associated with each network.

5.1.15 Diagnostics: Memory

To know the memory usage history of your nBox you can use the Memory page.



Welcome to nBox

Memory statistics

```
■ Memory    Min: 201.0 MBytes   Avg: 208.3 MBytes   Max: 212.8 MBytes
■ Swap      Min:   0.0  Bytes   Avg:   0.0  Bytes   Max:   0.0  Bytes
```

Presets: -----

Pid	Command	Memory Used
1770	n2disk	158'648K
2033	nprobe	32'112K

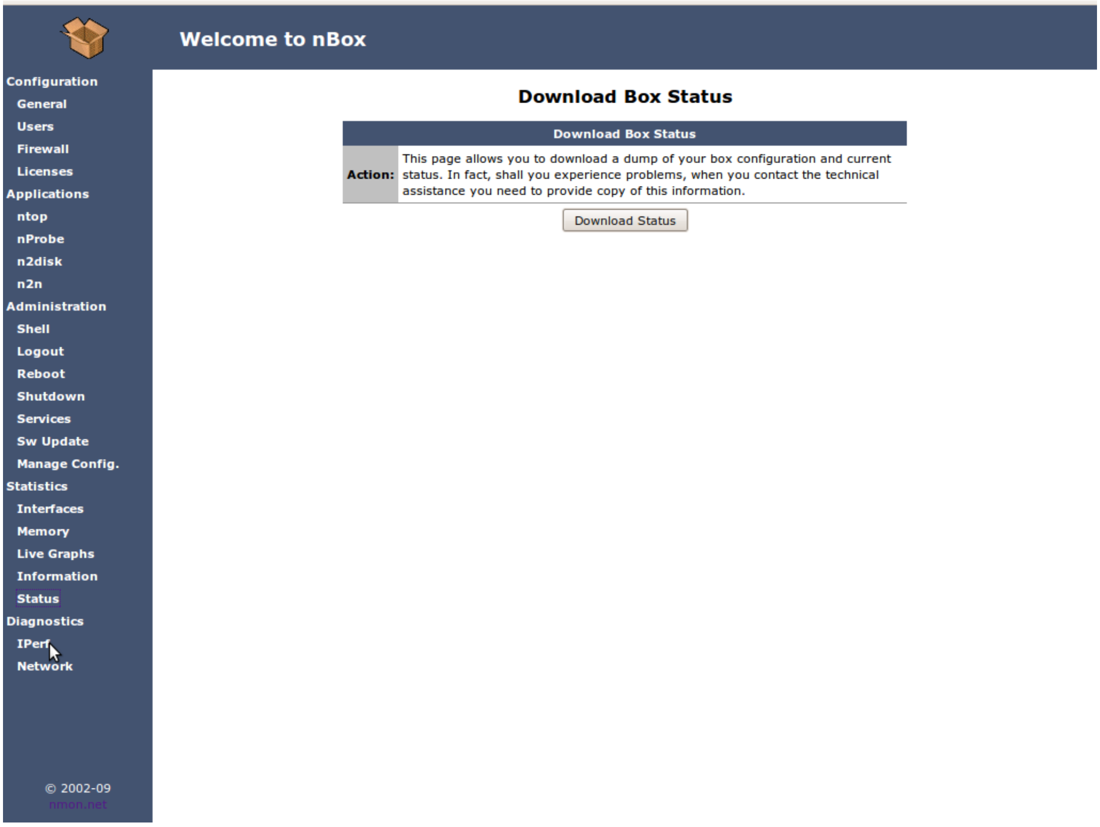
Per process memory usage.

Fig. 22 – nBox: Memory Diagnostics Page

In the Memory page you can find a graph showing the amount of memory used (main memory and also the swap space if you have a box with an hard-disk and swap area configured) within a time range. There is also a table containing the memory usage of some specific processes (if running) as ntop, nprobe and MySQL.

5.1.16 Diagnostics: Status

In case you need assistance with your nBox, you can download the whole box configuration to your PC and mail it to the nBox team.



Download Box Status

Download Box Status	
Action:	This page allows you to download a dump of your box configuration and current status. In fact, shall you experience problems, when you contact the technical assistance you need to provide copy of this information.

Fig. 23 – nBox: Status Diagnostics Page

Please include the file obtained from the Status page every time you need assistance, so that we can quickly provide you the information you need without the need to access your box for having a look to its configuration.

5.2 nBox in pass-through mode

In case you own a box with three network interfaces, you can use it in trunk-mode to analyze up to three different network trunks, or in pass-through mode for transparently analyzing your network traffic. In this configuration the interfaces 1 and 2 are bridged, whereas the interface 3 is used to access the box and emit NetFlow flows. In trunk-mode the interfaces 1 and 2 can be connected to different network trunks so that the box can simultaneously analyze two network trunks.

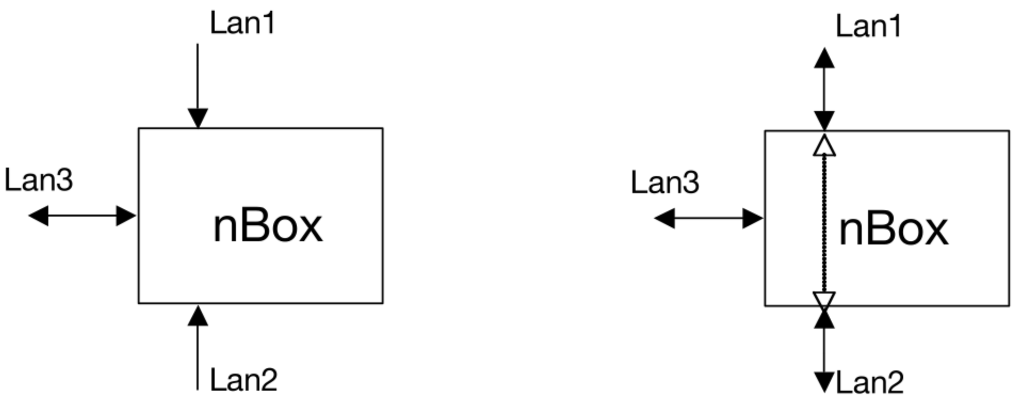


Fig. 24 - nBox: Trunk vs. Bridge mode

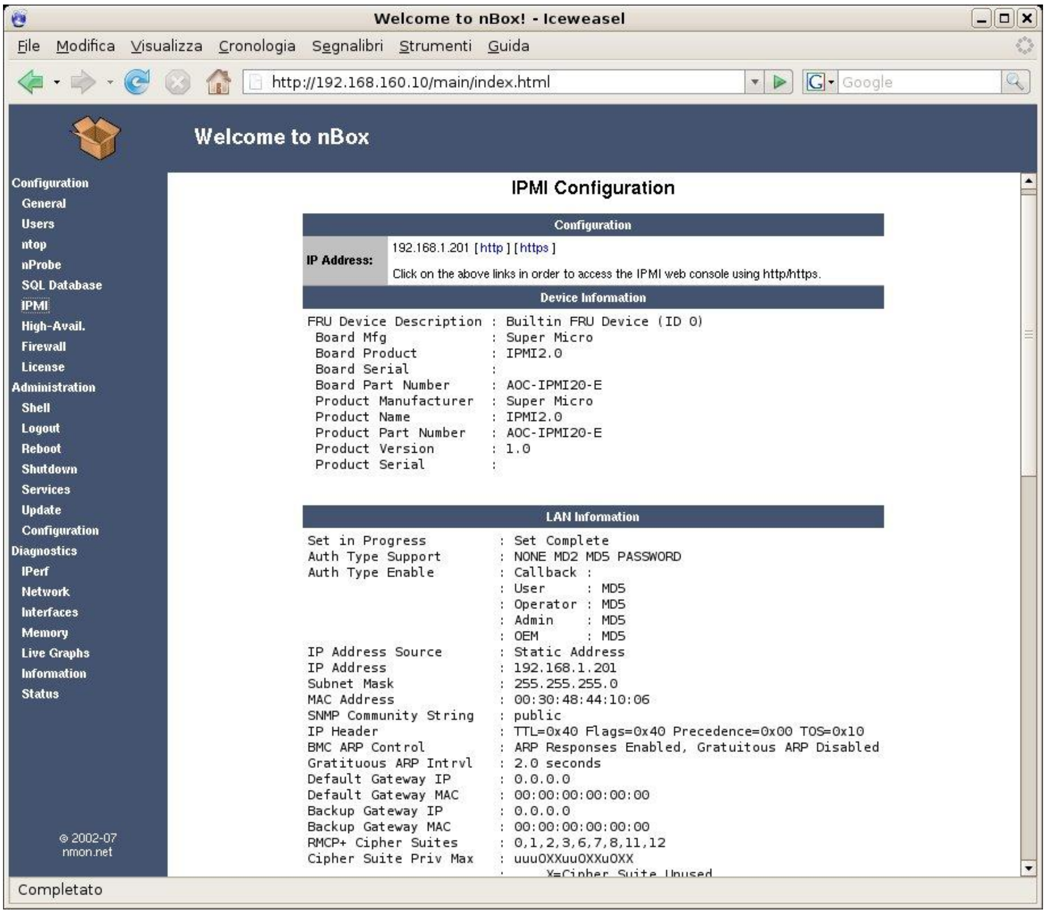
Note that in pass-through mode the network traffic flows are inside your box, so in case of hardware failure, shutdown or misconfiguration the network operations are interrupted as packets cannot be moved from interface 1 to 2 (and vice versa). Furthermore it is possible to configure nBox in pass-through mode also in case the box has only 2 network interfaces but you must be warned that in case of network failure you will not be able to reach the box.

5.3 Accessing nBox via IPMI

IPMI stands for Intelligent Platform Management Interface and is an open standard protocol for machine control and health monitoring. In many systems, the main processor must either constantly poll subsystems or deal directly with common subsystem faults and alerts. With IPMI, this processing burden shifts to an IPMI-compliant embedded service processor. The service processor (also known as BMC - Baseboard Management Controller) handles system event management, freeing the main processor for other tasks. Because the BMC is a separate service processor, the monitoring and control functions work regardless of CPU operation suppose the cases where the main processor is failing or system power-on status.

Here are some useful things that IPMI can do:

1. checks hardware health
2. console redirection over IP
3. remote access to the system console and the BIOS
4. remote management without the need for specific OS support.



IPMI Configuration

Configuration

IP Address: 192.168.1.201 [http] [https]

Click on the above links in order to access the IPMI web console using http/https.

Device Information

```
FRU Device Description : Builtin FRU Device (ID 0)
 Board Mfg             : Super Micro
 Board Product         : IPMI2.0
 Board Serial          :
 Board Part Number     : AOC-IPMI20-E
 Product Manufacturer  : Super Micro
 Product Name          : IPMI2.0
 Product Part Number   : AOC-IPMI20-E
 Product Version       : 1.0
 Product Serial        :
```

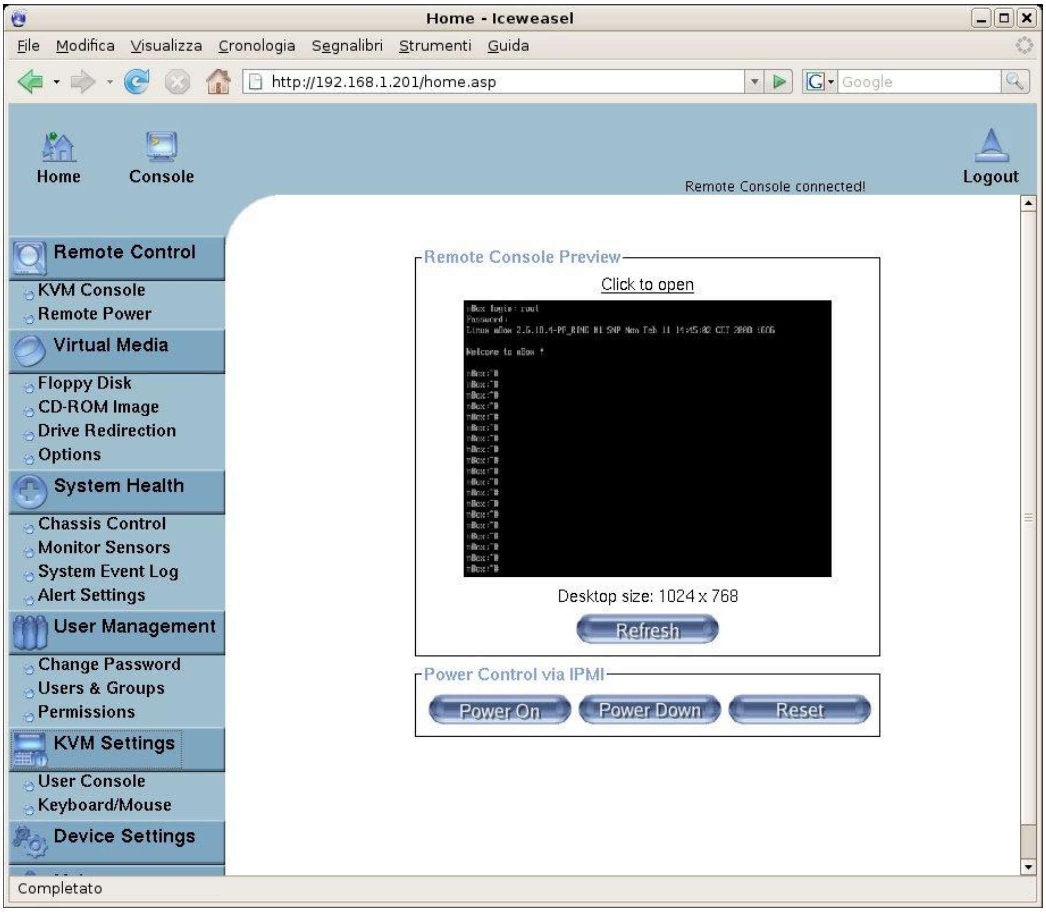
LAN Information

```
Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 PASSWORD
Auth Type Enable        : Callback :
                        : User     : MD5
                        : Operator : MD5
                        : Admin    : MD5
                        : OEM      : MD5
IP Address Source       : Static Address
IP Address              : 192.168.1.201
Subnet Mask             : 255.255.255.0
MAC Address             : 00:30:48:44:10:06
SNMP Community String   : public
IP Header               : TTL=0x40 Flags=0x40 Precedence=0x00 TOS=0x10
BMC ARP Control         : ARP Responses Enabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 2.0 seconds
Default Gateway IP      : 0.0.0.0
Default Gateway MAC     : 00:00:00:00:00:00
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
RMCP+ Cipher Suites     : 0,1,2,3,6,7,8,11,12
Cipher Suite Priv Max   : uuuOXXuuOXXuOXX
                        :     X=Cipher Suite Unused
```

Fig. 25 - nBox: IPMI Configuration Page

Console redirection is mainly useful during system restarts and when there are system failures. Administrators can then gain full remote access to text-based system information and control for BIOS, utilities, operating systems and applications. IPMI enables remote access to server's environmental status as well as power control like reset, reboot and shutdown. All of this functionality comes regardless of the state of the operating system. The IPMI protocol brings-up an out-of-band network, which provides a path to the system when regular in-band connectivity is lost or is unresponsive. An administrator can access the BMC by using an IPMI-compliant management application loaded on a PC or via a Web interface.

In case you purchased your nBox with IPMI card (please take a look at the table in Chapter 5), you have an extra Ethernet interface that is used for accessing the nBox's IPMI card. Accessing the IPMI services with nBox is very simple, just select IPMI in the Configuration menu on the left side, it will display the "IPMI Configuration" page with some basic Device and LAN information about the box.



Remote Console connected!

```
nBox login: root
Password:
Linux nBox 2.6.18.4-PF_RING #1 SMP Mon Feb 11 14:45:02 CET 2008 i686

Welcome to nBox !

nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
nBox:~#
```

Desktop size: 1024 x 768

Fig. 26 - IPMI Management Page

All the available IPMI services can be managed through the IPMI Web interface that can be accessed by clicking on the http or https page links or directly via a web browser pointing to the displayed IP address.

The default IPMI Web user is ADMIN and default password is ADMIN (both upper case).

You can change the default IPMI configuration settings using the IPMI Web interface browsing "Device Settings" -> "Network". For further information please refer back to the IPMI User's Guide that is provided with the companion CD part of your nBox.

6. nBox models summary

In the following table are briefly described the main features of the nBox models currently available and the typical monitoring scenario where each model could be used.

	nBox-EL	nBox-L	nBox-M	nBox-H	nBox-10G
Form Factor	Desktop	Rackmount (19'')			
Management Port	1				
Input Monitoring Ports (Min/Max)	1/2	1/5	1/9	1/5	1/1
IPMI Port	No	Optional		Yes	
HW Acceleration	None (Software Acceleration via PF_RING kernel module)				Tilera or Endace
Line to monitor	10/100 Mbit	10/100/1000 Mbit			10Gbit
Maximum Traffic Volume	Up to 100Mbit	Up to 200Mbit	Up to 500Mbit	1Gbit	10Gbit
Concurrent Flows/s	2000	Up to 5000	Over 5000	10000	Over 10000
Typical Monitoring Scenario	xDSL	100Mbit Network	Lightly Loaded Gbit Network	Heavy Loaded Gbit Network	Heavy Loaded 10Gbit Network

In the following table the nBox-Recorder models are summarized:

	nBox-R1	nBox-R3	nBox-R8
Form Factor	Rackmount (19'')		
Maximum Traffic Volume	Up to 250 Mbit	Up to 600 Mbit	1Gbit
Packets per seconds	Up to 150,000	Up to 250,000	Up to 400,000
Storage size	1 TB	3 TB	8 TB
RAID	None	Software RAID 0 Optional HW RAID 0,5	HW RAID 0,5, 10

For more information about the nBox models please check the nBox web site on: www.nmon.net/nBox.html and www.nmon.net/recorder.html

7. References

Introduction to Cisco NetFlow, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm

ntop, http://www.ntop.org/

nProbe, http://www.ntop.org/nprobe.html

nBox, http://www.nmon.net/nBox.html

Linux Debian, http://www.debian.org/

tcpdump, http://www.tcpdump.org/

Extreme Happy Netflow Tool, http://ehnt.sourceforge.net/

Libpcap, http://www.tcpdump.org/

Winpcap, http://winpcap.polito.it/

PC Engines, http://www.pcengines.ch/

SQLite, http://www.sqlite.org

IPerf, http://dast.nlanr.net/Projects/Iperf

tcpreplay, http://tcpreplay.synfin.net/

runsv, http://runsv.sourceforge.net

8. Credits

NetFlow is a trademark of Cisco Systems.

Windows is a trademark of Microsoft Corporation.

Appendix A

BPF Packet Filtering Expressions

This section has been extracted from the tcpdump man page and it describes the syntax of BPF filters you can specify using the –f flag.

The expression consists of one or more primitives. Primitives usually consist of an id (name or number) preceded by one or more qualifiers. There are three different kinds of qualifier:

type

qualifiers say what kind of thing the id name or number refers to. Possible types are host, net and port. E.g., `host foo', `net 128.3', `port 20'. If there is no type qualifier, host is assumed.

dir

qualifiers specify a particular transfer direction to and/or from id. Possible directions are src, dst, src or dst and src and dst. E.g., `src foo', `dst net 128.3', `src or dst port ftp-data'. If there is no dir qualifier, src or dst is assumed.

proto

qualifiers restrict the match to a particular protocol. Possible protos are: ether, fddi, ip, arp, rarp, decnet, lat, moprc, mopdl, tcp and udp. E.g., `ether src foo', `arp net 128.3', `tcp port 21'. If there is no proto qualifier, all protocols consistent with the type are assumed. E.g., `src foo' means `(ip or arp or rarp) src foo' (except the latter is not legal syntax), `net bar' means `(ip or arp or rarp) net bar' and `port 53' means `(tcp or udp) port 53'.

[`fddi' is actually an alias for `ether'; the parser treats them identically as meaning ``the data link level used on the specified network interface.'' FDDI headers contain Ethernet-like source and destination addresses, and often contain Ethernet-like packet types, so you can filter on these FDDI fields just as with the analogous Ethernet fields. FDDI headers also contain other fields, but you cannot name them explicitly in a filter expression.]

In addition to the above, there are some special `primitive' keywords that don't follow the pattern: gateway, broadcast, less, greater and arithmetic expressions. All of these are described below.

More complex filter expressions are built up by using the words and, or and not to combine primitives. E.g., `host foo and not port ftp and not port ftp-data'. To save typing, identical qualifier lists can be omitted. E.g., `tcp dst port ftp or ftp-data or domain' is exactly the same as `tcp dst port ftp or tcp dst port ftp-data or tcp dst port domain'.

Allowable primitives are:

dst host host

True if the IP destination field of the packet is host, which may be either an address or a name.

src host host

True if the IP source field of the packet is host.

host host

True if either the IP source or destination of the packet is host. Any of the above host expressions can be prepended with the keywords, ip, arp, or rarp as in: ip host host

which is equivalent to: ether proto \ip and host host

If host is a name with multiple IP addresses, each address will be checked for a match.

ether dst ehost

True if the ethernet destination address is ehost. Ehost may be either a name from /etc/ethers or a number.

ether src ehost

True if the ethernet source address is ehost.

ether host ehost

True if either the ethernet source or destination address is ehost.

gateway host

True if the packet used host as a gateway. I.e., the ethernet source or destination address was host but neither the IP source nor the IP destination was host. Host must be a name and must be found in both /etc/hosts and /etc/ethers. (An equivalent expression is ether host ehost and not host host

which can be used with either names or numbers for host / ehost.)

dst net net

True if the IP destination address of the packet has a network number of net, which may be either an address or a name.

src net net

True if the IP source address of the packet has a network number of net.

net net

True if either the IP source or destination address of the packet has a network number of net.

dst port port

True if the packet is ip/tcp or ip/udp and has a destination port value of port. The port can be a number or a name used in /etc/services. If a name is used, both the port number and protocol are checked. If a number or ambiguous name is used, only the port number is checked (e.g., dst port 513 will print both tcp/login traffic and udp/who traffic, and port domain will print both tcp/domain and udp/domain traffic).

src port port

True if the packet has a source port value of port.

port port

True if either the source or destination port of the packet is port. Any of the above port expressions can be prepended with the keywords, tcp or udp, as in: tcp src port port which matches only tcp packets.

less length

True if the packet has a length less than or equal to length. This is equivalent to: len <= length.

greater length

True if the packet has a length greater than or equal to length. This is equivalent to: len >= length.

ip proto protocol

True if the packet is an ip packet of protocol type protocol. Protocol can be a number or one of the names icmp, udp, nd, or tcp. Note that the identifiers tcp, udp, and icmp are also keywords and must be escaped via backslash (\), which is \\ in the C-shell.

ether broadcast

True if the packet is an ethernet broadcast packet. The ether keyword is optional.

ip broadcast

True if the packet is an IP broadcast packet. It checks for both the all-zeroes and all-ones broadcast conventions, and looks up the local subnet mask.

ether multicast

True if the packet is an ethernet multicast packet. The ether keyword is optional. This is shorthand for `ether[0] & 1 != 0'.

ip multicast

True if the packet is an IP multicast packet.

ether proto protocol

True if the packet is of ether type protocol. Protocol can be a number or a name like ip, arp, or rarp. Note these identifiers are also keywords and must be escaped via backslash (\). [In the case of FDDI (e.g., `fddi protocol arp'), the protocol identification comes from the 802.2 Logical Link Control (LLC) header, which is usually layered on top of the FDDI header. ntop assumes, when filtering on the protocol identifier, that all FDDI packets include an LLC header, and that the LLC header is in so-called SNAP format.]

decnet src host

True if the DECNET source address is host, which may be an address of the form ``10.123'', or a DECNET host name. [DECNET host name support is only available on Ultrix systems that are configured to run DECNET.]

decnet dst host

True if the DECNET destination address is host.

decnet host host

True if either the DECNET source or destination address is host.

ip, arp, rarp, decnet

Abbreviations for: ether proto p where p is one of the above protocols.

lat, moprc, mopdl

Abbreviations for: ether proto p where p is one of the above protocols. Note that ntop does not currently know how to parse these protocols.

tcp, udp, icmp

Abbreviations for: ip proto p where p is one of the above protocols.

expr relop expr

True if the relation holds, where relop is one of >, <, >=, <=, =, !=, and expr is an arithmetic expression composed of integer constants (expressed in standard C syntax), the normal binary operators [+, -, *, /, &, |], a length operator, and special packet data accessors. To access data inside the packet, use the following syntax: proto [expr : size] Proto is one of ether, fddi, ip, arp, rarp, tcp, udp, or icmp, and indicates the protocol layer for the index operation. The byte offset, relative to the indicated protocol layer, is given by expr. Size is optional and indicates the number of bytes in the field of interest; it can be either one, two, or four, and defaults to one. The length operator, indicated by the keyword len, gives the length of the packet.

For example, `ether[0] & 1 != 0' catches all multicast traffic. The expression `ip[0] & 0xf != 5' catches all IP packets with options. The expression `ip[6:2] & 0x1fff = 0' catches only unfragmented datagrams and frag zero of fragmented datagrams. This check is implicitly applied to the tcp and udp index operations. For instance, tcp[0] always means the first byte of the TCP header, and never means the first byte of an intervening fragment.

Primitives may be combined using:

A parenthesized group of primitives and operators

(parentheses are special to the Shell and must be escaped).

Negation (`!' or `not').

Concatenation (`&&' or `and').

Alternation (`||' or `or').

Negation has highest precedence. Alternation and concatenation have equal precedence and associate left to right. Note that explicit and tokens, not juxtaposition, are now required for concatenation. If an identifier is given without a keyword, the most recent keyword is assumed. For example, not host vs and ace is short for not host vs and host ace which should not be confused with not (host vs or ace). Expression arguments can be passed to nProbe as either a single argument or as multiple arguments, whichever is more convenient. Generally, if the expression contains Shell metacharacters, it is easier to pass it as a single, quoted argument. Multiple arguments are concatenated with spaces before being parsed.

Examples

1. To select all packets arriving at or departing from sundown:
2. nprobe -f “host sundown”
3. To select traffic between helios and either hot or ace:
4. nprobe -f “host helios and (hot or ace)”
5. To select all IP packets between ace and any host except helios:
6. nprobe -f “ip host ace and not helios”
7. To select all traffic between local hosts and hosts at Berkeley:
8. nprobe -f “net ucb-ether”

    a. To select all ftp traffic through internet gateway snup: (note that the expression is quoted to prevent the shell from (mis-)interpreting the parentheses):

9. nprobe -f “gateway snup and (port ftp or ftp-data)”

    a. To select traffic neither sourced from nor destined for local hosts (if you gateway to one other net, this stuff should never make it onto your local net).

10. nprobe -f “ ip and not net localnet”

    a. To select the start and end packets (the SYN and FIN packets) of each TCP conversation that involves a non-local host.

11. nprobe -f “tcp[13] & 3 != 0 and not src and dst net localnet”
12. To select IP packets longer than 576 bytes sent through gateway snup:
13. nprobe -f “gateway snup and ip[2:2] > 576”

    To select IP broadcast or multicast packets that were not sent via ethernet broadcast or multicast:

14. nprobe -f “ether[0] & 1 = 0 and ip[16] >= 224”
15. To select all ICMP packets that are not echo requests/replies (i.e., not ping packets):
16. nprobe -f “icmp[0] != 8 and icmp[0] != 0"

Appendix B

Text Flow Dump Format

The –D flag enabled users to specify the dump format: text file or SQLite file. The format options currently supported by nProbe are those specified with –T (see appendix C), so for instance if you want to dump the flows on a text file with the following packet fields:

- protocol
- source and destination address,
- source and destination port

you can use the following command:

```
nprobe –i eth0 –F 120 -P /tmp –D b -T "%PROTOCOL_MAP %IPV4_SRC_ADDR %L4_SRC_PORT_MAP %IPV4_DST_ADDR %L4_DST_PORT_MAP"
```

As result, the dump of the flows will be in the directory /tmp (-P option). It will be divided in a set of subdirectory (with the format <year>/<month>/<day>/<hour>/) where each leaf directory contains a list of files. Each file will have a name with the format <minute>.flows where minute is the time minute when the file is dumped.Note that in the example above the flows will be dumped every 2 minutes (-F option).

Appendix C

NetFlow v9/IPFIX Flow Format

The –T flag enables users to specify the format of NetFlow v9/IPFIX flows. The format options currently supported by nProbe are those specified in the NetFlow v9 RFC, namely (in square brackets it is specified the field Id as defined in the RFC):

[1] %IN_BYTES Incoming flow bytes

[2] %IN_PKTSIncoming flow packets

[3] %FLOWSNumber of flows

[4] %PROTOCOL IP protocol byte

[132] %PROTOCOL_MAP IP protocol name

[5] %SRC_TOSType of service byte

[6] %TCP_FLAGSCumulative of all flow TCP flags

[7] %L4_SRC_PORTIPv4 source port

```
[135] %L4_SRC_PORT_MAPIPv4 source port symbolic name
[8] %IPV4_SRC_ADDRIPv4 source address
[9] %SRC_MASK Source subnet mask (/<bits>)
[ 10] %INPUT_SNMP Input interface SNMP idx
[ 11] %L4_DST_PORTIPv4 destination port
[139] %L4_DST_PORT_MAPIPv4 destination port symbolic name
[ 12] %IPV4_DST_ADDRIPv4 destination address
[ 13] %DST_MASK Dest subnet mask (/<bits>)
[ 14] %OUTPUT_SNMPOutput interface SNMP idx
[ 15] %IPV4_NEXT_HOPIPv4 next hop address
[ 16] %SRC_AS Source BGP AS
[ 17] %DST_AS Destination BGP AS
[ 21] %LAST_SWITCHEDSysUptime (msec) of the last flow pkt
[ 22] %FIRST_SWITCHED SysUptime (msec) of the first flow pkt
[ 23] %OUT_BYTESOutgoing flow bytes
[ 24] %OUT_PKTS Outgoing flow packets
[ 27] %IPV6_SRC_ADDRIPv6 source address
[ 28] %IPV6_DST_ADDRIPv6 destination address
[ 29] %IPV6_SRC_MASKIPv4 source mask
[ 30] %IPV6_DST_MASKIPv4 destination mask
[ 32] %ICMP_TYPEICMP Type * 256 + ICMP code
[ 34] %SAMPLING_INTERVALSampling rate
[ 35] %SAMPLING_ALGORITHM Sampling type (deterministic/random)
[ 36] %FLOW_ACTIVE_TIMEOUTActivity timeout of flow cache entries
[ 37] %FLOW_INACTIVE_TIMEOUTInactivity timeout of flow cache entries
[ 38] %ENGINE_TYPEFlow switching engine
[ 39] %ENGINE_IDId of the flow switching engine
[ 40] %TOTAL_BYTES_EXPTotal bytes exported
[ 41] %TOTAL_PKTS_EXP Total flow packets exported
[ 42] %TOTAL_FLOWS_EXPTotal number of exported flows
[ 56] %IN_SRC_MAC Source MAC Address
[ 57] %OUT_DST_MACDestination MAC Address
[ 58] %SRC_VLAN Source VLAN
[ 59] %DST_VLAN Destination VLAN
[ 60] %IP_PROTOCOL_VERSION[4=IPv4][6=IPv6]
[ 61] %DIRECTION[0=ingress][1=egress] flow
[ 70] %MPLS_LABEL_1 MPLS label at position 1
[ 71] %MPLS_LABEL_2 MPLS label at position 2
```

[72] %MPLS_LABEL_3 MPLS label at position 3

[73] %MPLS_LABEL_4 MPLS label at position 4

[74] %MPLS_LABEL_5 MPLS label at position 5

[75] %MPLS_LABEL_6 MPLS label at position 6

[76] %MPLS_LABEL_7 MPLS label at position 7

[77] %MPLS_LABEL_8 MPLS label at position 8

[78] %MPLS_LABEL_9 MPLS label at position 9

[79] %MPLS_LABEL_10MPLS label at position 10

[80] %FRAGMENTED 1=some flow packets are fragmented

[81] %FINGERPRINTTCP fingerprint

[82] %CLIENT_NW_DELAY_SECNetwork latency client <-> nprobe (sec)

[83] %CLIENT_NW_DELAY_USEC Network latency client <-> nprobe (usec)

[84] %SERVER_NW_DELAY_SECNetwork latency nprobe <-> server (sec)

[85] %SERVER_NW_DELAY_USEC Network latency nprobe <-> server (usec)

[86] %APPL_LATENCY_SEC Application latency (sec)

[87] %APPL_LATENCY_USECApplication latency (sec)

[96] %IN_PAYLOAD Initial payload bytes

[97] %OUT_PAYLOADInitial payload bytes

[98] %ICMP_FLAGS Cumulative of all flow ICMP types

Plugin SIP templates:

[130] %SIP_CALL_IDSIP call-id

[131] %SIP_CALLING_PARTYSIP Call initiator

[132] %SIP_CALLED_PARTY SIP Called party

[133] %SIP_RTP_CODECS SIP RTP codecs

[134] %SIP_INVITE_TIMESIP SysUptime (msec) of INVITE

[135] %SIP_TRYING_TIMESIP SysUptime (msec) of Trying

[136] %SIP_RINGING_TIME SIP SysUptime (msec) of RINGING

[137] %SIP_OK_TIMESIP SysUptime (msec) of OK

[138] %SIP_ACK_TIME SIP SysUptime (msec) of ACK

[139] %SIP_RTP_SRC_PORT SIP RTP stream source port

[140] %SIP_RTP_DST_PORT SIP RTP stream dest port

Plugin Efficiency calculation templates:

[165] %EFFICIENCY_SENTAvg. transmission efficiency % (send)

[166] %EFFICIENCY_RCVDAvg. transmission efficiency % (rcvd)

Plugin Video protocol detection (skeleton plugin) templates:

[188] %VIDEO_PROTOSimple counter

Plugin SMTP Protocol Dissector templates:

[185] %SMTP_MAIL_FROM Mail sender

[186] %SMTP_RCPT_TO Mail recipient

Plugin Flow Serial Identifier templates:

[190] %FLOW_IDSerial Flow Identifier

Plugin HTTP Protocol Dissector templates:

[180] %HTTP_URL HTTP URL

[181] %HTTP_RET_CODEHTTP return code (e.g. 200, 304...)

Plugin dump templates:

[100] %DUMP_PATHPath where dumps will be saved

Plugin RTP templates:

[150] %RTP_FIRST_SSRC First flow RTP Sync Source ID

[151] %RTP_FIRST_TS First flow RTP timestamp

[152] %RTP_LAST_SSRCLast flow RTP Sync Source ID

[153] %RTP_LAST_TSLast flow RTP timestamp

[154] %RTP_IN_JITTERRTP Jitter (ms * 1000)

[155] %RTP_OUT_JITTER RTP Jitter (ms * 1000)

[156] %RTP_IN_PKT_LOSTPacket lost in stream

[157] %RTP_OUT_PKT_LOST Packet lost in stream

[158] %RTP_OUT_PAYLOAD_TYPE RTP payload type

[159] %RTP_IN_MAX_DELTA Max delta (ms*100) between consecutive pkts

[160] %RTP_OUT_MAX_DELTAMax delta (ms*100) between consecutive pkts

For instance if you want to specify NetFlow v9 flows in a format similar to v5 flows you can do as follows:

```
nprobe -T "%IPV4_SRC_ADDR %IPV4_DST_ADDR %IPV4_NEXT_HOP %INPUT_SNMP %OUTPUT_SNMP %IN_PKTS %IN_BYTES %FIRST_SWITCHED %LAST_SWITCHED %L4_SRC_PORT %L4_DST_PORT %TCP_FLAGS %PROTOCOL %SRC_TOS %SRC_AS %DST_AS %SRC_MASK %DST_MASK"
```

Note that the fields start with a % and are separated by a space. ??????????

Appendix D

nBox Firmware Upload into a CF

nBox can be installed simply uploading the firmware image you received to a compact flash (CF) or an ATA flash disk. It is also possible to use an (old) hard disk to store the firmware image that takes less than 512 MB. This is a simple operation that can be performed at no risk by end users. However, considered the price of a CF, we suggest you to use an additional compact flash for experiments so that at worst the original compact flash can still work. The CF can be seen as a normal hard disk with little differences in term of read/write speed and capacity.

The quickest way to do this is described below:

Get the firmware image from ntop.org and make sure it fits into your CF (you need a 512 MB compact flash or larger). Suppose that the image is named nBox.dd.

Plug your CF into a CF reader (there are many USB readers or CF to IDE adapters).

On Unix (Linux/BSD) do "dd if=nBox.dd of=/dev/XXX" where XXX is the name of the device where the CF is connected (note that XXX is the name of a device, e.g. hda, and not the name of a device partition such as hda1). On Windows you can use rawrite to perform the same operation.

If everything goes well you can reinstall the CF into your box and power it.

Some computers come with an internal compact flash socket, you might find convenient to get one of them. Instead if you plan to use the nBox on an existing PC you can either use a CF+ATA adapter or an ATA flash disk that's basically a solid-state hard disk. The boot loader expects the system to be mounted as secondary master disk. In case you used the disk as primary master you need to pass "root=/dev/hda1" to the boot loader. In case you want to make this change permanent you need to edit the boot loader configuration file (/etc/lilo.conf), replace hda with hdc, save the file, run lilo and then reboot.

The Linux kernel included with your nBox contains, compiled as modules, all the available Linux network card drivers. By default, in case of an unconfigured box, the system probes the available network cards and loads the modules for the cards you have. Module probing requires a few seconds, therefore in order to decrease the startup time, we suggest you to edit the file /etc/modules.conf and specify the correct modules for your network cards.

Appendix E

nProbe License

nProbe is open-source software (see http://www.opensource.org/) and is distributed under the GNU GPL2 license.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA, 02111-1307, USA.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License.The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.(Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope.The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program).

Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when running, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole.If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works.But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code.(This alternative is allowed only for non commercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it.For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License.Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it.However, nothing else grants you permission to modify or distribute the Program or its derivative works.These actions are prohibited by law if you do not accept this License.Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License.If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all.For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices.Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded.In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time.Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.If the Program specifies a version number of this License which applies to it and "any later version", you have the option to follow the terms and conditions either of that version or of any later version published by the Free Software Foundation.If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission.For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this.Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.