# Monitoring Networks Using Ntop

*L. Deri, R. Carbone*
*Netikos S.p.A.*
*Via Matteucci 34/b*
*56124 Pisa*
*Italy*
*{deri, rocco}@ntop.org*

*S. Suin*
*Centro Serra, University of Pisa*
*Lungarno Pacinotti 43*
*56100 Pisa*
*Italy*
*stefano@ntop.org*

**Abstract**

Today's networks present several management challenges due to the variety of network types and the integration of different network media. Network administrators need automated tools to support human effort, gathering information about the status and behaviour of network elements. Network monitoring is probably the most fundamental aspect of automated network management.

The goal of this paper is to describe the design and implementation of ntop a simple, open-source, portable traffic measurement and monitoring tool, which supports various management activities, including network optimisation and planning, and detection of network security violations. In addition, it describes some scenarios where ntop can be effectively used for identifying common network problems as well as detecting intruders and security violations.

**Keywords**

Network monitoring, optimization, security and planning.

## 1. Background and Motivation

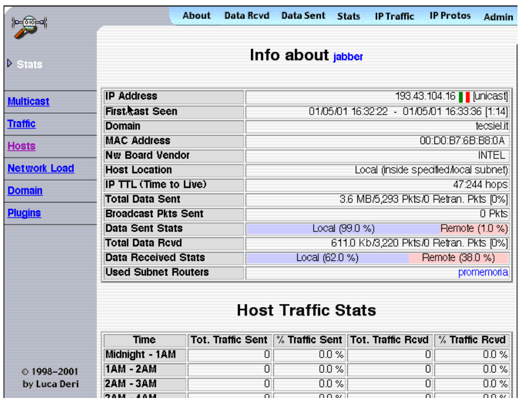
As an increasingly number of organisations rely on networks for communications and electronic commerce, existing LANs have often been quickly updated, extended, and interconnected to the Internet. In this dynamic scenario the task of network management is becoming an increasingly complex task, requiring automated tools to support human effort. According to [12], network monitoring is the most fundamental aspect of automated network management. Companies and organisations willing to deploy a commercial network monitoring system must face with a high monetary investment and complex deployment that often includes purchase of additional hardware and human training.

Some people, mostly from Universities and small ISPs (Internet Service Provider), who could not easily afford those commercial monitoring systems crafted their tools for basic network monitoring in order to have an idea of the type of traffic flowing across their network. When in 1997 the authors had to face with the task of monitoring the backbone of the University of Pisa, they decided first to see whether existing tools freely available on the Internet were good enough for monitoring such a complex and heterogeneous network. Traditional Unix tools for testing basic connectivity problems as well as network sniffers were considered not sufficient. These tools are very powerful for tracking specific network problems but they need off-line tools for better analysing and correlating captured data as well as identifying security violations.

After having done a survey of available tools, the authors decided to write a new application able to both monitor campus traffic and report information about the overall network security. This decision was motivated by the fact that none of the above tools offered all the features he needed while an integration of some of them would have been too challenging considering that most of the tools were not designed to be plugged together. This is why ntop had birth. Similar to the Unix top tool that reports processes CPU usage, the authors needed a simple tool able to report the network top users (hence the term ntop) for quickly identifying those hosts that were currently using most of the available network resources. Ntop was designed to give network administrators quick access to network monitoring, with a simple to use integrated web interface and minimal requirements. The idea was to make it available to network administrators with minimal (installation and learning) effort and cost, as opposed to expensive and complex (yet sophisticated and flexible) commercial management platforms. Nevertheless ntop is not targeted to replace commercial platforms but rather to fill a empty niche in the network monitoring arena: a lightweight, web-based, cross-platform traffic monitoring tool able to provide network administrators the information they need for keeping their network healthy and make informed decisions about network planning, accounting, and provisioning. This is one of the reasons why many ISPs deployed ntop since early releases. The other and most important reason is that ntop is open source software (OSS) [10]. The authors decided to base ntop on open source because they believed that this was probably the only way he could afford to write an application that:

- Does exactly what network administrators need: this is due to the fact that many requirements come directly from ntop users.
- It is robust and truly cross-platform as it has been tested by many people on different OSs, network topology, traffic conditions and media type.

In fact, although the authors designed the initial ntop architecture and coded most of it, many requirements come from ntop users as well as bug fixes and design changes. The current version of ntop, available on both Unix and Win32, features both a pure web-based application (ntop itself) and a command line application called intop (interactive ntop) initially conceived as a text-mode version of ntop targeted to those network administrators who prefer the shell to a web browser. The current intop has been mostly rewritten and turned into a sort of network shell that features a true shell and enhanced versions of several commands common network commands.



Info about jabber

| | |
|---|---|
| IP Address | 193.43.104.16 [unicast] |
| First/Last Seen | 01/05/01 16:32:22 - 01/05/01 16:33:36 [1:14] |
| Domain | tecsiel.it |
| MAC Address | 00:D0:B7:6B:B8:0A |
| Nw Board Vendor | INTEL |
| Host Location | Local (inside specified/local subnet) |
| IP TTL (Time to Live) | 47:244 hops |
| Total Data Sent | 3.6 MB/5,293 Pkts/0 Retran. Pkts [0%] |
| Broadcast Pkts Sent | 0 Pkts |
| Data Sent Stats | Local (99.0 %) Remote (1.0 %) |
| Total Data Rcvd | 611.0 Kb/3,220 Pkts/0 Retran. Pkts [0%] |
| Data Received Stats | Local (62.0 %) Remote (38.0 %) |
| Used Subnet Routers | promemoria |

Host Traffic Stats

| Time | Tot. Traffic Sent | % Traffic Sent | Tot. Traffic Rcvd | % Traffic Rcvd |
|---|---|---|---|---|
| Midnight - 1AM | 0 | 0.0 % | 0 | 0.0 % |
| 1AM - 2AM | 0 | 0.0 % | 0 | 0.0 % |
| 2AM - 3AM | 0 | 0.0 % | 0 | 0.0 % |

Figure 1: ntop Web Interface

Ntop currently focuses on the following functional areas:

- Traffic measurement.
- Traffic characterisation and monitoring.
- Network optimisation and planning.
- Detection of network security violations.

The following sections are structured as follows: section 2. covers the ntop architecture, section 3. presents in detail the previous functional areas showing what kind of information ntop can report, section 4 discusses some performance issues, and finally section 5 describes some open issues and future work.

## 2. Ntop Architecture Design

The ntop architecture design has been strongly influenced by the lessons taught by the Webbin [3] architecture, a web-based CMIP/SNMP management system [2]. Webbin pioneered the integration of the web with the world of network management and demonstrated that runtime application extensibility is mandatory in the management world, as new requirements have to constantly be accommodated due to rapid evolution of network technology. For this reason the ntop architecture preserved many design goals of Webbin such as:

- Portability across Unix and non-Unix (e.g. Win32) platforms.
- Simple, scalable, and efficient application kernel with low resource (both memory and CPU) usage.
- Minimal requirements (bare operating system) but capable of exploiting platform facilities, if present (e.g. kernel threads).
- Runtime application extensibility by using dynamically loadable software

components.

adding new ones:

- Ability to present data in a secure way both in a character-based terminal and in a browser (either web or WAP).
- The network analysis output should be rich in content and easy to understand.
- Adoption of the open source development paradigm.
- Use of open/industry standards (e.g. SNMP) for simplifying the integration with existing enterprise management systems.

The following figure depicts the ntop architecture:

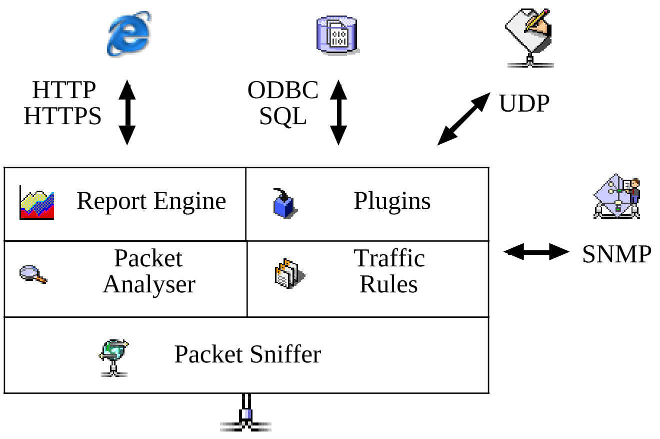


Figure 2: Ntop Architecture

The main architecture components are:

- The packet sniffer, capable of handling multiple network interface simultaneously, is based on the libpcap (http://www.tcpdump.org/) library that provides a portable and unified packet capture interface, whereas other operating systems provide proprietary capture facility. Thanks to a good design of libpcap, the authors decided to port libpcap to non-Unix platforms embedding native platform capture facilities (e.g. NDIS on Win32) into it. This allowed the ntop source code to be unique across various platforms.
- The packet analyser processes captured packets, correlates packets belonging to the same flow (e.g. TCP connection), learns about the network topology (e.g. maintains a list of the current network routers and DNS servers), maintains traffic statistics of known hosts. Host's information is stored in a dynamically growable hash table whose entries contain several counters that keep track of the data sent/ received by the host, sorted according to the supported network protocols. Since it is not possible to predict the number of different hosts whose packets will be

handled by the application, ntop allocates up to a specified amount of memory. Either periodically or if there are no entries left ntop purges the host table in order to avoid exhausting all the available memory and creating huge tables that are then slow to walk.

- The report engine contains an embedded web server that allows users to view traffic reports using a web browser without the need to install and configure an external web server. The server features access lists, TSL (Trusted Socket Layers) support via OpenSSL (http://www.openssl.org), TCP wrapper (ftp://ftp.porcupine.org/pub/security/) support and user authentication for restricting ntop access. Traffic reports make extensive use of dynamic charts that have been implemented using the GDchart (http://www.fred.net/brv/chart/) library. Although HTML is probably the best way to navigate through ntop reports, intop allows users to take advantage of ntop by means of a command line interface. In addition to HTML, ntop offers a WML [14] interface for accessing traffic information via a mobile GSM WAP phone [4].

If available, all the above components can run in separate threads. This guarantees not only a better performance with respect to a single threaded application, but it also significantly decreases the packet loss probability due to potentially time-consuming operations such as address resolution. Ntop makes extensive use of plugins, software components implemented as a shared library (DLL, Dynamic Loadable Library, in Windows terminology).The use of plugins allowed the authors to:

- Selectively extend ntop without having to modify the ntop core: currently ntop sports some plugins for in depth analysis of protocols such as ARP, ICMP, and NFS, or for accessing ntop via ODBC/SQL/SNMP.
- Give the user the chance to activate just the needed features, and not just all the available ones.
- Selectively activate/deactivate at runtime specific ntop features.
- Integrate external code without requiring to change/relink ntop.

As ntop makes use of GNU tools such as libtool, autoconf, and automake, it is possible to specify at compile time the list of features that ntop has to support and those that are supported by the underlying operating system. The ability to tailor ntop both at compile time and runtime according to user needs, allowed people to deploy ntop in very heterogeneous environments. In fact it is possible to build a minimal ntop that supports just the basic features with no plugins, as well a version that features multithreading, SQL database integration and dynamic graphics. Thanks to this ease of tailoring, we have been able to run ntop on very different hardware and operating system configuration ranging from small Cobalt (http://www.cobalt.com/) boxes to powerful multiprocessor Sun servers. The ability to run embed ntop in small boxes, has allowed us to build a small, cheap, ready to use network probe that needs just to be plugged in the network with no other configuration needed beside host IP address configuration.

## 3. Monitoring Networks Using Ntop

Ntop has been originally designed as a traffic monitoring application for tackling performance problems of the campus network backbone, able to give positive answers to frequent questions including:

- Why is the local network performance so poor?
- What host is using most of the available network bandwidth?
- What is the bandwidth percentage actually used by my computer?
- What are the contacted peers and the amount of network traffic produced by each of the processes running on my local computer?

Over the time, as users requested new features, the application evolved towards a more complex and comprehensive network analysis application. The current ntop version focuses on the following functional areas:

1. Traffic measurement.
2. Traffic monitoring.
3. Network optimisation and planning.
4. Network intrusion detection.

The following sections cover in detail each of the above areas.

### 3.1 Traffic Measurement

Ntop differs from many traffic monitoring tools because it transparently processes traffic data while capturing packets, and provides live traffic information in a human readable format. Captured packets are associated with source/destination hosts and stored in a dynamic hash table. Each new captured packet is used for both updating traffic statistics and learning about network topology. A hash slot is used to store information about one host. As the number of hosts can increase dramatically due to traffic condition, whenever ntop has to perform garbage collection, traffic information about purged hosts is stored in a local persistent cache implemented using GNU gdbm. In this way, when a purged host generates new traffic hence a hash slot needs to be allocated, ntop checks first if the cache contains information about the host. If the searched information is found, ntop restores it otherwise a brand new hash slot is created. The use of a local cache allows ntop to handle a large amount of hosts without losing traffic statistics when memory exhausts. For each host, the following information is recorded:

- The total traffic (volume and packets sent/received) generated/received by the host classified according to network protocol (IP, IPX, AppleTalk, etc.) and, when applicable, IP protocol (TCP, UDP, ICMP, FTP, HTTP, NFS, etc.).

- IP multicast traffic statistics.
- TCP session history: source/destination, duration, TCP sliding window size and TTL statistics, retransmitted data, fragmented packets percentage.
- Host used TCP/UDP services, operating system type (using the nmap tool, http://www.nmap.org), address tracking by means of DHCP monitoring.
- Traffic distribution (local vs. remote traffic), network usage (contacted peers, traffic generated by each running application), overall used bandwidth (actual, peak, and average), local subnet traffic matrix.
- Packets distribution: total number of packets sorted by packet size, unicast vs. multicast vs. broadcast, and IP vs. non-IP traffic. This information is very similar to the one reported by RMON probes.
- Protocol utilisation and distribution according to both protocol and source/destination (local vs. remote).
- Network Flows
  Traffic statistics for each user-defined flow. A network flow is a stream of packets that matches a user-specified rule. Rules are specified using BPF expressions at ntop start-up. Similar to NeTraMet [1] flows, ntop network flows can be used to specify traffic of particular interest. Ntop applies all the stored flow filters to each captured packet. When a packet matches a filter, the flow counters are updated. Please note that packet processing time increases with the number of defined flows and the complexity of the associated filters.

It is worth to note that the current ntop version comes with a couple of plugins that provide detailed statistics about NFS/ARP/ICMP protocol usage and display statistics about such protocols.

### 3.2 Traffic Monitoring

Traffic monitoring is the ability to identify those situations where network traffic does not comply with specified policies or when it exceeds some defined thresholds. In general, network administrators specify some policies to which all the hosts must obey, sometimes known as SLA (Service Level Agreement). Inside the traffic monitoring category fall all those situations caused by both misconfigured and faulty applications generating traffic that should not normally flow across healthy networks.

Ntop natively provides support for detecting some network configuration problems including:

- Duplicate IP Addresses Identification.
- Subnet Gateway Misconfiguration
  Ntop identifies the subnet routers by checking the association destination IP/MAC address in each captured packet (not just the ones directed to non-local IP

addresses). Subnet routers are identified by the destination MAC address whereas hosts with misconfigured netmasks are identified because they send a router those packets that are directed to hosts belonging to the local subnet.

- Misconfiguration of software applications
  The analysis of some protocol traffic data allows administrators to guess that there is something wrong on a certain host. For instance, the use of ntop has allowed us to detect the installation of an unauthorised caching DNS that was filling up its cache very frequently and a misconfigured NTP client that was asking the time of the day once every five seconds.
- Service misuse detection by identifying hosts/users that do not make use of the specified proxies or that route packets through suboptimal gateways.
- Protocol misuse by identifying those computers that run unnecessary protocols For instance, the Windows operating system installs by default protocols such as NetBEUI and IPX while most people need just TCP/IP.
- Excessive host bandwidth utilisation.

## 3.3 Network Optimisation and Planning

Whenever it is necessary to upgrade/update an existing network in order to provide an adequate speed, it is not often easy to know where are the real bottlenecks. Sometimes the way some hosts are configurated causes troubles to the whole networks, whereas in other situations network administrators might need assistance for knowing how the network is used in 24 hours in order to better plan extensions. For instance, if network performance is adequate during most of the day it might be cheaper to postpone some tasks in order to avoid traffic peaks other than buying new expensive network equipment. Therefore, in order to make decisions about network extensions, network administrator must be able to know:

- How the overall network traffic changes over the time in order to decide whether the network is slow most of the time or just in some specified time frames.
- The number and type of network assets and their categorisation according to collected usage.
- What hosts, if any, are misconfigured in order to see whether those few hosts have a great negative impact on the overall network performance.

In particular ntop allows administrators to:

- Discover and categorise assets according to collected usage
  ntop is able to report information about network assets including running operating system and list of known capabilities (e.g. DNS Server, gateway). In addition, ntop contains decoders for protocols such as IPX/Netbios/AppleTalk that provide much more information than IP regarding host information. For instance Appletalk NBP (Name Binding Protocol) that is similar to Internet DNS (Domain Name Server), contains information not limited to host name but also

informs about the running applications and the asset type.

- Identify unnecessary protocols
  Sometimes traffic is generated by hosts/routers that have not been configured properly and that attempt to communicate with peers using protocols that nobody else beside them is using. In addition, ntop can easily identify cases where in pure IP networks some misconfigured hosts use protocols such as IPX/Netbios/AppleTalk that generate some periodic broadcast traffic propagated to the whole subnet.
- Identify suboptimal routing
  The icmpWatch ntop plugin is responsible for handling ICMP packets. It is possible to identify machines that use a non-optimal routing just by keeping track of ICMP redirect messages or by periodically analysing the list of subnet routers.
- Traffic characterisation and distribution
  Ntop allows administrators to understand how traffic is distributed with respect to the protocol and origin (local vs. remote traffic). The study of traffic patterns helps administrators to understand how the network is used both locally and from remote and hence to improve, if possible, the global network topology and configuration.
- Reduction of the number of protocols used by replacing some old/legacy protocols with new ones (e.g. replace NetBIOS with NetBIOS-over-IP when possible) in order to reduce the number of protocols (hence to simplify network administration) without losing any existing functionality.
- Wiser bandwidth usage by studying how certain protocols are used, hence add applications such as proxies that allow traffic to be significantly reduced by caching information.

## 3.4 Network Intrusion Detection

Requirements for ntop security facilities have been drawn from our experience administering the campus backbone. In general we have noticed that most of the attacks are originated from inside the university, because of lack of security or because some students install software applications that make intruders life much simpler. Ntop offers simple facilities for recognising issues at TCP level (e.g. portscan, synflood, and land) and implements a lightweight NIDS (Network Intrusion Detection System) which allows users to specify traffic rules using a simple language. Basic security features include:

- Portscan detection
  The classic (send a packet to every port) and slow (a kind of portscan where port scan happens very slowly in order to make its detection more difficult) portscan (stealth scan) [6] can be easily detected.
- Spoofing detection identification by means of the arpWatch plugin that warns the user when two distinct IP addresses map to the same hardware address. Please

note that spoofing detection should be used properly on networks where proxy ARP routers are installed or whenever a host has enabled IP aliasing support.

- Spy detection
  A spy [8] is an host whose network card is set in promiscuous mode for capturing packets independently of whether they are directed to the host or not. Neped is a tool that sends to each host X of the local subnet an ARP request containing as target IP address the X IP address. Ntop periodically runs neped and warns users about hosts whose cards are set in promiscuous mode. Unfortunately, the algorithm just described does not work for all operating systems hence neped is used to report information about hosts that certainly have their NIC set in promiscuous mode whereas nothing can be said about the remaining hosts.
- Vulnerability Scanners Detection
  Attacks usually happen in two phases: in the first phase, the attacker learns about network topology and vulnerability, whereas in the second phase the real attack begins. In the first phase, quite often attackers use vulnerability detection tools (e.g. Saint and Nessus) in order to rapidly identify the easiest way to break the target network. Ntop can detect such scanners as it logs suspicious packets (e.g. SYN/FIN packets), identifies connections not properly terminated, and records ports usage making easy to identify suspicious situations (e.g. ports with incoming and no outgoing traffic) that need further investigation by network administrators.
- Trojan Horses detection by monitoring usage of ports that are used by well known trojan applications such as BO2K.
- Denial of service
  synflood [9], smurf [7] and network melt-down can be detected by analysing the traffic sent/received statistics and glancing through the throughput graphs.
- Network Discovery identification by monitoring ARP (local network discovery) and ICMP (local/remote network discovery) traffic. Peak of unanswered requests in a given amount of time are usually the proof of existence of such applications running on the network.
- Suspicious Packets
  Using libraries freely available on the Internet, hackers can easily forge packets for the purpose of exploiting security flaws of the TCP/IP protocol suite and weakness of some TCP/IP stack implementations. For this reason ntop is able to recognise peaks of packets having the RST (reset) flag set, overlapping offsets of fragmented packets and packets with SYN-ACK or SYN/FIN flag that do not belong to an established connection.

The ntop NIDS provides a more sophisticated way to approach intrusion detection. In fact, it allows users to specify simple IDS rules in a way similar to tools such as Snort [11]. Traffic rules are specified at ntop startup on the command line. Those rules are used for instructing both the correlation engine and the alarming system.

While capturing packets, ntop learns network topology and hosts relationships (i.e. routers, DNS, networks) and stores this information in a network knowledge database that is also used by the correlation engine for making decisions (e.g. a use can specify that ICMP redirect packets can be sent only by gateways). Whenever the current traffic matches the specified rules and an alert has to be emitted, the alarming system performs the action specified in the traffic rules and add an entry in the log. Logging is quite important as it allows to perform a post-attack anal-ysis, to detect attacks distributed over the time or outside of business hours.

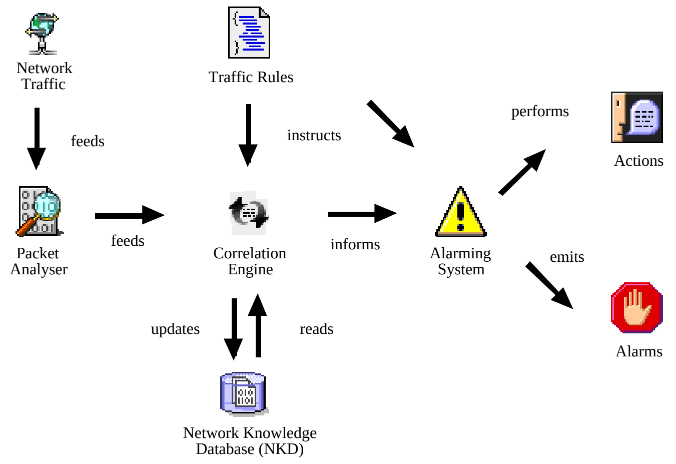


Figure 3: Ntop NIDS Architecture

Ntop's network knowledge makes it different from any other NIDS tool. This allows users to specify rules in a simple and compact way without having to know the network topology where those rules will be used as shown below:

- `icmp route-advertisement ICMP_REDIRECT !gateway/any action alarm`: emit an alarm when a host receives an ICMP redirect from another host that is not a gateway.
- `tcp root-ftp any/ftp any/any contains "230 User root logged in." action alarm`  
  Emit an alarm when somebody logged in as root on a FTP server.
- `tcp syn-without-ack any/any any/any flags S action mark expires 6`  
  `tcp ack revert any/any any/any flags A clears syn-without-ack all`: Emit an alarm for each TCP connection that has not completed the three-way handshake within 6 seconds.

## 4. Performance Issues

Users have tested ntop extensively on various network types running at different speeds. The current ntop version contains several optimisation that have allowed us to produce a mostly passive network application as users can decide not to run at all external tools that produce some network traffic such as nmap. In order to significantly reduce DNS queries, ntop processes DNS reply packets and caches mappings for future use. An important performance factor is the extensive use of several internal caches and hash tables for decreasing the packet processing time. This means that ntop can run in background on a 100 Mbit Ethernet with no noticeable performance degradation and very low (if any) packet loss that is strongly influenced by the percentage of CPU available to ntop.

Extensive cache usage has the drawback that memory usage is increased. In order to specify an upper bound to memory usage, ntop implements a secondary cache on disk. This means that when the primary cache (kept in memory) is full, ntop moves to disk information about hosts that have not transmitted/received since a long time. Instead, semi-persistent information such as IP address resolution (mapping numeric/symbolic IP address) and remote host operating system (computed using the neped) is always kept on disk in a persistent database. This solution guarantees that ntop will not use more that 16Mbytes of memory (this parameter is user configurable at startup) regardless of network traffic conditions.

## 5. Open Issues and Future Work

Ntop has been designed as a self-contained monitoring application targeted for small organisations that could not afford an enterprise management console. Over the time, many users equipped with a commercial console realised that their system lacked of some features present in ntop. Those users demanded a way to integrate ntop in their existing environment. We realised that the web interface is a good way for human operators to glance through ntop reports but it is just not adequate for a large enterprise. For this reason, we decided to expose via SNMP most of the information that ntop provides via HTTP. As ntop is in many ways similar to an RMON probe, the first MIB that ntop will implement is be RMON MIB. Currently ntop implements just the Ethernet stats group part of RMON 1. In the near future we plan to add new groups in order to implement most (if not all) of the RMON MIB.

Another limitation that has been reported by users is the very limited customisation of HTML reports. Some users demanded different reports, others wanted to add new ones but they simply could not do that because ntop is coded in C and they know only scripting languages such as Perl and PHP. In order to satisfy these requirements, we are currently implementing a plugin that embeds a Perl interpreter. This solution allows developers to extend ntop reports using Perl rather than C. The main advantages are that most of the people who play with the web are able to code in Perl, and also that reports can be changed on the fly without having to recompile ntop.

The current ntop version lacks of an automatic way to detect such misconfigurations. Namely the network administrator will detect problems by analysing specific traffic values according to his/her past experience. The authors are currently working at an automatic procedure for providing users hints about network problems without requiring them a strong networking experience. In fact, we are investigating whether it is possible to make ntop more intelligent as it is today. In fact, ntop features very powerful traffic analysis tools but it does not analyse automatically the traffic. For instance we would like ntop to produce one very simple page containing information such as “please check host X as it is currently sending too many ARP requests” and “host Y is misconfigured: its IP address is out of range”. We are currently studying whether neural networks or other techniques can be effectively used for giving a positive answer to this open issue.

Finally, the authors are trying to add a new analysis dimension to ntop. Currently traffic is analysed at packet and host level. This means that when a network problem happens, network administrators have to analyse the information reported by ntop with additional information (e.g. host X belongs to user Y) in order to associate a problem with a user. In past months, the authors tried to add a new dimension: the users dimension. Latest ntop snapshots, contain code that allow ntop to learn who is the user of a certain hosts and what are his/her customs. For instance it is possible to identify the name/email address of a host user sitting on a certain host, by analysing SMPT/IMAP/POP headers. User identity is very useful in intrusion detection because it not only allow hackers to be identified, but also because it allow to classify users. For instance if user X is a secretary and usually uses the network from 8AM to 5PM, if at some point in time ntop finds out that the same user is playing with napster at midnight, then it might be that someone else has stolen his identity. Beside technical problems, the main challenge for the authors is to find out how to perform this analysis by preserving the user’s privacy.

## 6. Final Remarks

This paper covered the design and implementation of ntop an open source application for traffic monitoring and intrusion detection. Ntop has been designed as a web-based application able to present network traffic and security information in a simple fashion, without the need to purchase expensive commercial tools. In addition ntop has been made available to network administrator with minimal (installing, learning) effort and cost, as opposed to expensive and complex (yet sophisticated and flexible) management platforms. Features such as embedded HTTP server for web and WAP access, support of various network media types, lightweight CPU utilisation, portability across various platforms, storage of traffic information into a SQL database, SNMP support, extensibility via software components and integration with many network tools, make ntop attractive for traffic analysis and network security.

## 7. Availability

Ntop for both Unix and Win32 is distributed under the GPL2 licence and can be downloaded free of charge from both the ntop home page (http://www.ntop.org/) and other mirrors on the Internet. Some Unix distributions including but not limited to FreeBSD and Linux, come with ntop preinstalled.

## 8. References

[1] N. Brownlee, “Network Traffic Meter and NeTraMet Manager/Collector”, Version 4.3, http://www.auckland.an.nz/net/Accounting/, October 1999.

[2] L. Deri, “Surfin' Network Management Applications Across the Web”, Proceedings of *2nd Int. IEEE Workshop on System and Network Management*, 1996.

[3] L. Deri, “A Component-based Architecture for Open, Independently Extensible Distributed Systems”, Ph.D. Thesis, University of Berne, Switzerland, 1997.

[4] L. Deri, “Beyond the Web: Mobile WAP-based Management”, *Journal of Network and Systems Management*, Special Issue on Web-Based Management, (in press) 2000.

[5] P. Ferguson, and D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing”, *RFC 2267*, January 1998.

[6] Fyodor, “The Art and Detection of Port Scanning”, *Sys Admin Magazine*, Nov. issue, 1998.

[7] C. Huegen, “The Latest in Denial of Service Attacks: Smurfing”, http://www.quadrunner.com/~chuegen/smurf.txt, December 1998.

[8] B. Mukherjee, and others, “Network intrusion detection”, *IEEE Network*, 8(3), 1994.

[9] “TCP/IP Security: TCP SYN Flooding”, *Phrack Magazine*, 7(48), 1996.

[10] E. Raymond, “The Cathedral & the Bazaar”, O’Reilly & Associates, ISBN 1-56592-724-9, 1999.

[11] M. Roesch, “Snort - Lightweight Intrusion Detection for Networks”, Proceedings of *LISA ‘99*, 1999.

[12] W. Stallings, “SNMP, SNMPv2, SNMPv3 and RMON 1 and 2”, Third Edition, Addison Wesley, September 1999.

[13] S. Waldbusser, “Remote Network Monitoring Management Information Base”, RFC 1757, 1995.

[14] WAP Forum, “Wireless Markup Language”, http://www.wapforum.org/, April 1998.