The "NetBoard": Network Monitoring Tools Integration for INFN Tier-1 Data Center

D De Girolamo¹, L dell’Agnello and S Zani

Istituto Nazionale di Fisica Nucleare, CNAF, Viale P. Berti Pichat 6/2 Bologna, IT

E-mail: donato.degirolamo@cnaf.infn.it

Abstract. The monitoring and alert system is fundamental for the management and the operation of the network in a large data center such as an LHC Tier-1. The network of the INFN Tier-1 at CNAF is a multi-vendor environment: for its management and monitoring several tools have been adopted and different sensors have been developed. In this paper, after an overview on the different aspects to be monitored and the tools used for them (i.e. MRTG, Nagios, Arpwatch, NetFlow, Syslog, etc), we will describe the "NetBoard", a monitoring toolkit developed at the INFN Tier-1. NetBoard, developed for a multi-vendor network, is able to install and auto-configure all tools needed for its monitoring, either via network devices discovery mechanism or via configuration file or via wizard. In this way, we are also able to activate different types of sensors and Nagios checks according to the equipment vendor specifications. Moreover, when a new device is connected in the LAN, NetBoard can detect where it is plugged. Finally the NetBoard web interface allows to have the overall status of the entire network "at a glance", both the local and the geographical (including the LHCOPN and the LHCONE) link utilization, health status of network devices (with active alerts) and flow analysis.

1. Introduction
The NetBoard is a network monitoring toolkit developed at INFN-CNAF. The basic idea is to install and integrate several monitoring tools in a single software system that works in a multi-vendor network environment. Moreover, in a complex data center, such as a LHC Tier-1, it is fundamental to have in a single “place” the overall status of the entire network "at a glance", either to minimize the reaction times of system engineers in case of errors, or to discover abnormal or unwanted behaviors. In particular, the NetBoard is able to install and auto-configure all tools needed for monitoring, and to build dynamically the entire network map too, both via network devices discovery mechanism and via configuration file. In this way, the NetBoard is able to activate different types of sensors and checks according to the equipment vendor specifications. Finally the NetBoard web interface integrates, in a single web page, the most important pieces of information coming from all network monitoring tools, giving a centralized and “effective” view of entire monitored IT infrastructure. Detailed status information of all devices is available as well.

¹ To whom any correspondence should be addressed.
2. Networking aspects to be monitored

Network monitoring means considering various aspects of the network: throughput, state of health of network devices, flow analysis and tools for day-by-day network management; for example we need a tool that shows at any time where every host in the network (or IP address) is plugged. At INFN TIER-1 the monitoring of the different aspects described above is committed to different independent systems, which characterize a single control environment: the “NetBoard”. Every used tool relies on open source software or on ad-hoc developed solution.

2.1. Throughput

The open source tool Multi Router Traffic Grapher (MRTG, [1]) takes care of collecting statistic on bandwidth utilization of all network devices. This system, based on Simple Network Management Protocol (SNMP, [2]), is able to read and to store network interfaces counters and to create graphs showing network trend on different time scale, from daily to yearly (Figure 1). Besides, thanks to MRTG saved counters, it has been possible to develop a kind of Loadmap (Figure 2), namely a graphics description of the uplinks load level of all network devices as well as the backbone, based on color code and progress bars. In this way, it’s easy to find out, for example, which switch is connected to the more bandwidth-consuming servers (and which they are) and to highlight bottlenecks. In particular, MRTG is very simple to use because it comes with a set of configuration tools which make configuration and setup very simple, so we don’t have to write manually any configuration file. The tools are cfgmaker and indexmaker. cfgmaker creates MRTG configuration files based on information pulled from a router or another SNMP manageable device. Instead indexmaker creates web pages containing the graphs of every monitored interface. Then the interfaces counters are read every 5 minutes via cron according to the information contained in the configuration file.

![Figure 1. MRTG Throughput Graph for a switch interface.](image)

![Figure 2. The Loadmap shows the uplink load of all network switches in a single view.](image)

2.2. Alert System

It can’t be possible to think about a Data Center without an alert system. System engineers have to know the “health status” of the network at every time and need to be immediately notified as soon as the trouble happens, in order to reduce downtimes. Nagios [3] has been integrated in the NetBoard as alert system, because of its stability, as well as for its features: monitoring of services and device
metrics; alert notification and escalation via e-mail and SMS; event handlers, for automatic problem solving, etc. In particular it is highly flexible. In fact it is a plugin-based monitoring system that allows users to easily develop their own checks or sensor to monitor any device, metric or service that is considered mission critical for the IT infrastructure. On the other side configuring Nagios can take quite a while, especially for first-time users. In fact it is necessary to create and edit properly several configuration files before monitoring anything, taking care in understanding how the various directives affect the monitoring itself. In particular the configuration files are: Main Configuration File that contains a number of directives that affect how the Nagios daemon operates; Resource Files that can be used to store user-defined macros. The main point of having resource files is to use them to store sensitive configuration information (like passwords), without making them available to the CGIs; Object Definition Files are used to define hosts, services, hostgroups, contacts, contactgroups, commands, etc. This is where you define all the things you want monitor and how you want to monitor them. CGI Configuration File contains a number of directives that affect the operation of the CGIs. It also contains a reference the main configuration file, so the CGIs know how you have configured Nagios and where your object definitions are stored (Figure 3). Therefore it is easy to understand that configuring Nagios by hand could be very difficult and tedious, and it is necessary to know the meaning of all Nagios configuration directives as well.

![Figure 3. Nagios Configuration Overview](image)

For this reason we thought to use a system that automatically creates all Nagios configuration files starting from a simple host list or from a “global” configuration file. We called that system the “Netboard”.

### 2.3. Arpwatch

Alongside Nagios, Arpwatch [5] has been integrated in NetBoard. It is used to keep track for [ethernet, ipaddress] pairings. Its textual database is always up-to-date because it uses pcap library to listen for ARP packets on virtual ethernet interfaces configured in trunk mode, in order to work with all VLANs defined in network. It logs and reports via e-mail (Figure 4) all new stations that “appear” in network, if a station change IP address or if there are two stations with the same IP address due to a misconfiguration (flip-flop). Furthermore, it is useful to discover where an IP address (or hostname) is plugged, just reading the L2 Forwarding Database of a switch or router via SNMP and parsing the result with arpwatch database. In order to work with all VLANs defined in LAN, it is necessary to have an host with a physical network interface where we create as many virtual network interface running in 802.1q trunk mode, as the number of VLAN we want to monitor (Figure 5). In this way we know at every time the ARP table of each VLAN without querying the router for that VLAN. Moreover, it should be noted that the switch port where we connect the arpwatch physical interface has to be a trunk port, otherwise we need n physical interfaces with arpwatch running, each of them on a different VLAN, but it could become very expensive.
3. The “NetBoard”

Since we have different monitoring systems dedicated to different aspects (MRTG, Nagios, etc.), when we have to add one or more new network device to the monitoring system, we have to edit several files per switch or router to configure various tools in order to monitor all aspects we need. By now we understood this action is very expensive in terms of time that a system engineer has to dedicate to it. On the other hand, when a user asks to a system engineer which is the status of a switch, he is forced to connect with its browser to more than one web server for viewing and correlate a lot of information from different monitoring systems. If we expand that question to the entire network we need a couple of hours or more to know the global status of our LAN. It is clear that in a complex data center network it is not feasible: system engineers have to know immediately the network status, at each instant. In this scenario the NetBoard, a collection of scripts and visualization tools written in Perl, tries to meet these needs and much more.

3.1. The NetBoard “global” configuration file

First of all NetBoard uses a single “global” configuration file that contains the list of network devices to monitor. In this way it is very simple and quick to add a switch or router to the monitoring system. In fact we don’t need to configure several monitoring tools, and edit by hand a lot of configuration files, but it is enough to add a line in the global configuration file and restart NetBoard. In fact a dedicated Perl script does the following operations:

1. parses the global configuration file,
2. learns the device type and vendor, with the sysDescr SNMP query,
3. enables specific Nagios plugins on vendor basis,
4. writes out all configuration files needed by Nagios and MRTG,
5. creates the web interface for every single device

Each line of the global configuration file representing a device contains the device name, IP address, uplink interfaces, bandwidth of uplinks, device to which it is connected. All these data could be discovered via another SNMP discovering tool (hostFinder) that queries the core routers and builds the map, from its own point of view (Cisco System [4]). But, since it could be “stressful” for the router CPU in a production network, we use the configuration file as described above (Figure 6).
3.2. The NetBoard Web Interface

The web interface is the other important component of NetBoard (Figure 7). The idea is to have a single web page that shows the overall status of the entire network “at a glance”, from all interesting point of view: throughput and health. It consists of a Flash application that takes in input an XML file containing objects describing the network and shows them in a web page. The XML file is created by NetBoard that collects information of all network devices from MRTG, Nagios and devices themselves in order to have a comprehensive overview of the LAN. In fact, as shown in the picture, the NetBoard web page has the following boxes:

1. Top Menu, that contains the links to detailed information of different monitoring aspects.
2. WAN Connection, with the graphs of geographical connections of the Data Center
3. Weather Map of core routers that shows the load of various backbone links based on color-code.
4. Top 10 Gigabit Interfaces that shows which are the top bandwidth-consuming 10G servers.
5. Top 20 Switches, that shows which are the top bandwidth-consuming switches.
6. Live Log, where are written the Nagios log or syslog in real time
7. Active Alert, that shows the “active” problems alerted by Nagios not yet solved.

---

Figure 6. NetBoard Global Configuration File

Figure 7. NetBoard Web Interface
In addition to the main web pages, all network devices have their own web page (Figure 8) that shows the details of the device itself. In this case the page shows not only the status of every single interface and the load in percentage, but using the tool hostFinder, it shows which host is plugged on each interface, too. The problem is that different switch brand has different SNMP query and response. For that reason the tool, first check the vendor, with sysDescr SNMP query, and then run the vendor specific “finder” script. Furthermore the finder is able to discover which are the uplink of the switch or using the following algorithm: if a FDB port ID belongs to all VLAN defined on the switch, it is a trunk port, otherwise, if a FDB port ID belongs to a subset of VLANs, or if a FDB port ID shows more than one MAC address, then on that interface there is either an hypervisor with some virtual machine or a switch dedicated to a specific VLAN.

Figure 8. NetBoard Switch Details

3.3. The hostFinder tool

Another useful tool of NetBoard is hostFinder (Figure 9), a Perl script and a web interface for host searching. The Perl script is able to find on which port of which switch is connected a MAC Address. But, as written before, correlating the L2 FDB of a switch or router and the Arpwatch database, it can be possible to say where an IP address or a hostname is plugged. The host/IP address search algorithm is the following:

1. Discover what is the router for that network with a traceroute
2. Parse the hostname/IP address in the Arpwatch database to find the MAC address
3. SNMP query the router for that MAC address and discover the FDB port ID
4. Discover the switch connected to that FDB Port ID via NetBoard global configuration file
5. SNMP query that switch to find the FDB Port ID which the MAC address is connected to

Figure 9. hostFinder
3.4. Netflow and sFlow

The network monitoring based on Netflow protocol allows to overcome some inherent limitations of traditional systems based on SNMP protocol (MRTG, Cacti, Cricket, etc…). These allow to gathering statistics only on the amount of traffic that crosses network devices. The systems that use the Netflow or sFlow protocol, instead, give full visibility of network utilization because allow to obtain detailed information on the nature of traffic in terms of host/network conversation: protocols and applications used and the source and destination of traffic flows also for high speed network. This type of information is available in most of network devices using sFlow (sFlow.org, [6]) and NetFlow (Cisco NetFlow, [7]) agents, which cache data related to TCP/UDP communications and periodically export them to a collector/analyzer. The analysis of data extracted from network flows is very important because helps network administrators to control the right use of network resources. Several advantages for network managers are, for example, the possibility to know which applications are more bandwidth consuming (GridFTP, P2P, Web, DNS, etc.) and which hosts are using them; thus it is possible to identify unauthorized network activities, to mark out source of possible network attacks and to define profile, priorities and traffic optimization and engineering. At INFN Tier-1 we just started to integrate in NetBoard the open source suite NfSen [8] and Nfdump [9] as a collector/analyzer software.

4. Conclusions

The use of a single system that covers all the aspects of network monitoring has proven to be cost-effective. By minimizing the reaction times of system engineers in case of troubleshooting, errors or abnormal and unwanted behaviors (e.g. bottlenecks), it drastically increases the availability and quality of service of the data center. Monitoring one or more new devices simply adding one or more lines to a global configuration file, regardless of model or manufacturer, is very useful in terms of time savings. Moreover, it is enough to have the NetBoard “global” configuration file, in order to have a fully functioning monitoring system replicated on as many machines as you want. Finally, other main advantages resulting from using integrated independent monitoring systems are flexibility and scalability; as network infrastructure becomes more complex, monitoring system grows with it, integrating new kind of checks, applications and devices by new manufacturers, without modifying the existing structure.

References

[1] MRTG: http://oss.oetiker.ch/mrtg
[2] SNMP: http://tools.ietf.org/html/rfc1157
[3] Nagios: http://www.nagios.org
[4] Using SNMP to Find a Port Number from a MAC Address on a Catalyst Switch, http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a00801c9199.shtml
[5] Arpwatch: http://ee.lbl.gov
[6] sFlow: http://www.sflow.org
[7] Cisco IOS NetFlow, http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html
[8] NfSen: http://nfsen.sourceforge.net
[9] Nfdump: http://nfdump.sourceforge.net