Open Network Administrator (ONA) – A Web-Based Network Management Tool

Bruce Campbell and Robyn Landers – University of Waterloo

ABSTRACT

This paper presents *Open Network Administrator* (ONA), a web-based network management tool. Network administrators interact with ONA over the web, and ONA interacts with routers, switches and access points, using telnet and/or SNMP.

ONA provides a common web interface to a number of different vendors' network products, and provides granular access control, permitting network administrators across multiple departments to manage a network in a consistent manner. In addition to typical day-to-day operations like changing switch port speed, duplex, VLAN, etc., ONA performs a number of other maintenance operations, such as automatically backing up switch configurations, maintaining traffic graphs, and sending device reboot/up/down alerts via e-mail.

ONA leverages some existing components, such as RRDtool [1] and CVSweb [2]. A summary of existing tools, both commercial and open source, is provided, with comparisons made to ONA.

At the University of Waterloo, where there is both a central IT group and many sizeable decentralized autonomous IT groups, ONA rapidly gained widespread enthusiastic acceptance.

Introduction

The *managed* network switch offers significant advantages, including the ability to control individual port settings, speed/duplex, VLAN, etc. But there is a caveat: managed switches need to be managed. For a small network, the web-based management tool included with most switches is adequate. But as the network expands, as IT staffing grows, and as the number of features packed into edge devices increases, a need to treat the network edge as a *system* develops. Managing each switch independently becomes unwieldy.

The movement of network features from the core to the edge has not been matched by a corresponding increase in availability of edge device management tools. The traditional approach to edge device management is to try to keep the edge simple, and do everything smart in the core. This leaves powerful features of modern switches unleveraged, and ultimately leads to scalability issues.

Abrahamson, et al. [15] point out the importance of integrating a definitive source of edge device configuration data into a software tool that is easy to use and of significant benefit to its users. Many network vendors do offer a comprehensive network management tool which can do just about everything imaginable . . . with that vendor's latest and greatest network equipment. The prospect of upgrading an enterprise's entire network infrastructure, and then being tied to a single vendor, often makes this option unrealistic.

ONA can be dropped into an organization's existing heterogeneous network, with existing equipment, and existing network management structure (or lack thereof), and provide an immediate benefit.

Motivation

In 2000, the networking for the Davis Centre building at the University of Waterloo was upgraded from thinnet ethernet to switched ethernet. The networking in this building was managed by the IT group for the Faculty of Mathematics.

Extreme Networks hardware was selected, and it was initially managed through the Extreme Networks management tool, EPICenter [3].

The building also hosted researchers and administrative staff from the Faculty of Engineering, which had its own IT staff. In order to streamline the day-to-day network management activities (i.e., moves, adds, and changes), a number of IT staff from both faculties involved were given administrative access to EPICenter. Procedures were developed for documentation and change notification.

EPICenter did not support the notion of shared management well, and logging was minimal. There was no way to restrict the types of changes each staff member could make, and tracing back changes to a specific person was difficult. The learning curve for EPICenter was fairly steep, and the user interface made it easy to make errors. This created a need to give EPICenter administrative access to as few IT staff as possible.

Conversely, with a major network upgrade in progress, the number of moves, adds, and changes was significant. This created a need to give EPICenter administrative access to as many IT staff as possible.

We needed a tool that permitted IT staff to make the changes they needed to make, and prevented them from making changes they shouldn't make.

Requirements

The minimal requirements were for a secure, easily usable and accessible tool to permit network support staff to change speed, duplex, VLAN, etc., settings on network switch ports for which they have been authorized, and to log all such changes.

The requirements for being secure and easily accessible made web-based operation the preferred choice. Although it is possible to install custom software on the desktops and laptops of all network support staff, or to use X11 or RDP to connect to a server-based GUI application, these are considered significantly less practical than web-based operation.

Prior to investigating a switch port management tool, we had already developed systems to perform a number of other network related tasks (e.g., maintaining traffic graphs with MRTG [4], sending device down/reboot alerts with Nagios [5]), as well as some custom scripts for saving switch configurations to a TFTP server. Each of these systems had its own list of switch names and SNMP community strings. This was not ideal, and led to the requirement that the new tool also perform these tasks, such that our existing systems could be phased out.

Finally, it was recognized that networking technology and the way we use and manage it is constantly evolving. This led to a requirement that the new tool be extensible to meet currently unforeseen needs. Therefore, while not strictly a requirement, an open source solution would be viewed as preferable to a commercial solution.

Alternatives

As of the day of writing, there is no web-based, open source, network management tool to be found. Most of the network management tools available are not in fact device management tools. The areas of DNS management, network monitoring, and traffic graphing appear to be well covered. They break out like this:

- LANdb, a network documentation tool, appears feature rich according to its web site, and even promises device management in a future release. However, development appears to have been stalled since 2000.
- GxSNMP is a device management tool, albeit not web-based. Again, development appears to have been stalled since 2001.
- Kiwi CatTools is a popular commercial Windows-based tool for managing, monitoring and reporting on network devices and their configuration. A limited version can be downloaded for free from their website, with more powerful versions aggressively priced.
- HP OpenView is perhaps the most widely known and powerful enterprise network management tool. Pricing is targetted towards the enterprise.
- Splat is a network device management tool with capabilities generally similar to ONA. The most obvious difference is that Splat is command-line oriented while ONA is web-based. ONA satisifies most of the requirements listed by the authors of Splat [15]. Splat uses RANCID [14] for secure communication with devices; ONA uses authenticated SNMP/telnet. Splat relies on an authoritative inventory database to obtain MAC addresses and hostnames. This is not practical for ONA because ONA is used by many departments in several faculties at UW, each of which might have its own inventory database design, or none at all. ONA implements features such as multi-level access control, automated switch recovery, and initial switch configuration, which were future goals for Splat.

Design

ONA is a web-based application, written in PHP [17]. It interacts with network switches, routers, and access points, using SNMP and/or telnet scripting. Device configuration and data are stored in a MySQL [18] database.

The database consists of approximately 40 tables that store definitions (administrators and devices), group memberships, state information (ARP and MAC tables), and logs. The **devices** table contains the device host names, device types (switch, router, wireless access point, etc.), primary group membership,

NAME	PURPOSE	INTERFACE	AVAILABILITY
cacti [6]	traffic graphing	web	open source
IP Manager [7]	DNS manager	web	open source
Kiwi CatTools [8]	switch manager	Windows GUI	commercial
LANdb [9]	network documentation	web	open source
WhatsUp gold [10]	network monitoring, mapping	web, GUI	commercial
HP OpenView [11]	network manager	web	commercial
GxSNMP [12]	network manager	Linux GUI	open source
OpenNMS [13]	network monitoring	web	open source
RANCID [14]	switch configuration	script	open source
splat [15]	switch configuration	script	open source
SNMPc [16]	network monitoring	web	commercial

Table 1: Existing management tools.

and SNMP community strings. The **administrators** table contains the userids, primary group membership, and preferences for each administrator. The sections on *Granular Access Control*, *Change Logging*, and *MAC/IP Querying, Logging, Searching* explain the group membership tables, state information tables, and logs, respectively.

ONA uses Apache [19] .htaccess for user authentication, and thus any underlying authentication mechanism (e.g., RADIUS [20], LDAP [21]) can be used. ONA receives the successfully authenticated userid from Apache, and does not interact directly with authentication servers itself.

While VLAN and port settings from all devices are stored in the ONA MySQL database, the configurations on the switches themselves are still considered authoritative. That is, if a switch configuration is changed manually, ONA will not overwrite that change, but rather will update its own database to reflect the change, either within 24 hours, or immediately if an ONA administrator chooses that option.

An alternative design which would make the ONA database authoritative was also considered. This could provide some advantages, such as the ability to make global configuration changes and have them automatically pushed out, or to replace a failed switch and have its configuration automatically updated. While the technical advantages were tempting, it would increase the risk that a bug in ONA could push a damaged configuration out to a large number of devices. Our experience with a commercial wireless access point manager has demonstrated this risk. We manually upgraded the firmware on a number of access points to fix a bug that was affecting our wireless deployment. The management software did not support the newer firmware, and rather than reverting to a "do not understand – do not do anything" state, it pushed an unusable configuration to the upgraded APs, putting them offline. This type of experience has the potential to chill enthusiasm, and was considered too risky for an environment that depends on voluntary adoption of solutions.

ONA is not a network documentation tool in itself. In addition to the standard port description field, ONA supports a comment field, which is stored in the ONA database. This field can be used to hold a jack number, hostname or other information as desired.

A word about terminology. An ONA *administrator* is someone who uses ONA to do network administration. An ONA *manager* is someone who has authority to manage or configure ONA itself (including for example granting permissions to administrators).

Main Features

Multi Vendor Support

ONA currently supports Extreme, HP Procurve, Nortel Baystack, Cisco 2900xl/3500xl and Cisco 2950/3550 switches, and Avaya AP-3 access points. Adding support for a new device involves writing PHP code to perform a number of primitive operations on the device, like fetching the port speeds and duplexes, setting tagged VLANs, etc. For example, the functions for managing the Nortel Baystack switch are as shown in Display 1.

The bulk of the work involved in adding support for additional devices is investigation of the SNMP involved. Beyond basic functions like querying the port statistics and interface descriptions, different vendors' SNMP implementations have little in common.

Each vendor typically relies on its own private MIBs, even for generic settings such as port speed and duplex. This is unfortunate as it reduces the opportunity to leverage existing code to support multiple vendors' equipment.

To give an idea of the effort required to add support for new devices, a student investigated the SNMP involved for the HP Procurve line over a period of two weeks, and then one of the authors developed the code, from scratch to debugged and in production, in about five days.

```
function set_nortel_port_tagged_vlans_via_snmp( $d, $portname, \
          $olduntaggedvlan, $untaggedvlan, $oldtaggedvlans, $taggedvlans )
function set_nortel_port_untagged_vlan_via_snmp( $d, $portname, \
          $oldvlan, $vlan, $istrunk )
function adjust_nortel_vlan_members( $d, $vlan, $remove_this_port, \
          $add_this_port )
function set_nortel_port_trunkmode_via_snmp( $d, $portname, \
          $trunkmode, $olduntaggedvlan, $untaggedvlan, $oldtaggedvlans, \
          $taggedvlans )
function get_nortel_vlan_configuration_via_snmp( $d, $signature )
function get_nortel_port_speeds_and_duplexes_via_snmp( $d, $signature )
function set_nortel_port_speed_duplex_via_snmp( $d, $portname, \
          $speed, $duplex )
function get_nortel_model_and_version_via_snmp( &$d )
function nortel_telnet_login( $d, $contin )
function nortel_telnet_logout()
function create_nortel_vlan_if_needed( $d, $vlan )
```

Display 1: Functions to manage the Nortel Baystack switch.

Traffic Graphs with RRDtool

RRDtool is integrated with ONA, and ONA automatically maintains RRD databases for all switch ports, via a cron job that runs every five minutes. Traffic graphs (or *traphs* as we like to call them) can be displayed in several formats. A summary traph page for each switch shows a 24-hour graph for each port. For each port, a number of traphs are available, ranging from a period of six hours to one year. A real time graph tool can show traffic on a specific port, updated every ten seconds. The real time traphs also show packet counts and error counts. An example summary traph page is shown in Figure 1.

Granular Access Control

Permissions are managed by means of relationships between administrators and devices. Administrators and devices are listed in tables that associate them with various groups. The groups are typically organizational units such as departments, faculties, IT support groups, and so on.

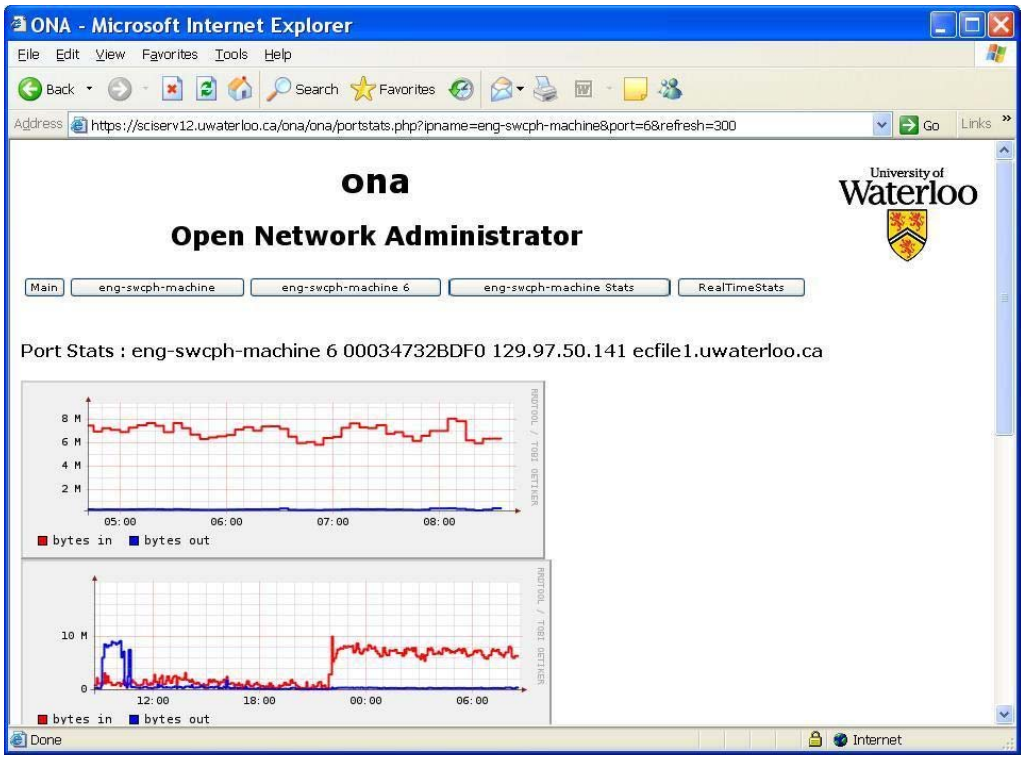
Devices and administrators each have a primary group to which they belong. The **devices** and **administrators** tables establish these primary group memberships. Additional group memberships are assigned through supplementary tables. Regular expressions may be used in these tables to specify, for example, ranges of ports, or devices whose names start with a common prefix, and so on. Each device (switch, router, or access point), switch port, VLAN tag, and administrator must be a member of at least one group.

Permission for an administrator to take a certain action on a certain device is determined by analyzing the group membership relationships. For example, an administrator may change the speed or duplex on a switch port if they have a group in common with the port or switch. They may change an untagged VLAN on a port if they have a group in common with the VLAN to which the port is being changed. If the port is a trunk port, then they also must have a group in common with all tagged and untagged VLANs on the port.

Every button in ONA and every settable parameter may be individually disabled or enabled on a per administrator basis. As an example of the level of granularity that this access control mechanism provides, an administrator could be given permission to change speed and duplex only, on one port only, and not have access to any of ONA's other features.

The following example illustrates how ONA uses groups and regular expression pattern matching.

- Two departments, "Engineering" and "Science," each have 10 switches.
- A core device "core-sw1" is managed by a department called "Central Services Department" (CSD)
- The Engineering switches are named eng-sw1 through eng-sw10
- The Science switches are named sci-sw1 through sci-sw10
- vlans 10 and 11 carry Engineering networks
- vlans 20 and 21 carry Science networks



Port Stats : eng-swcph-machine 6 00034732BDF0 129.97.50.141 ecfile1.uwaterloo.ca

Figure 1: Summary traffic graph page.

- The userid "engadmin," who works in Engineering, has full access on all Engineering switches
- The userid "sciadmin," who works in Science, has full access on all Science switches
- A user "manager," who works in CSD, has full access on all switches
- A user "operator," who works in CSD, has permission to disable/enable client ports on all switches

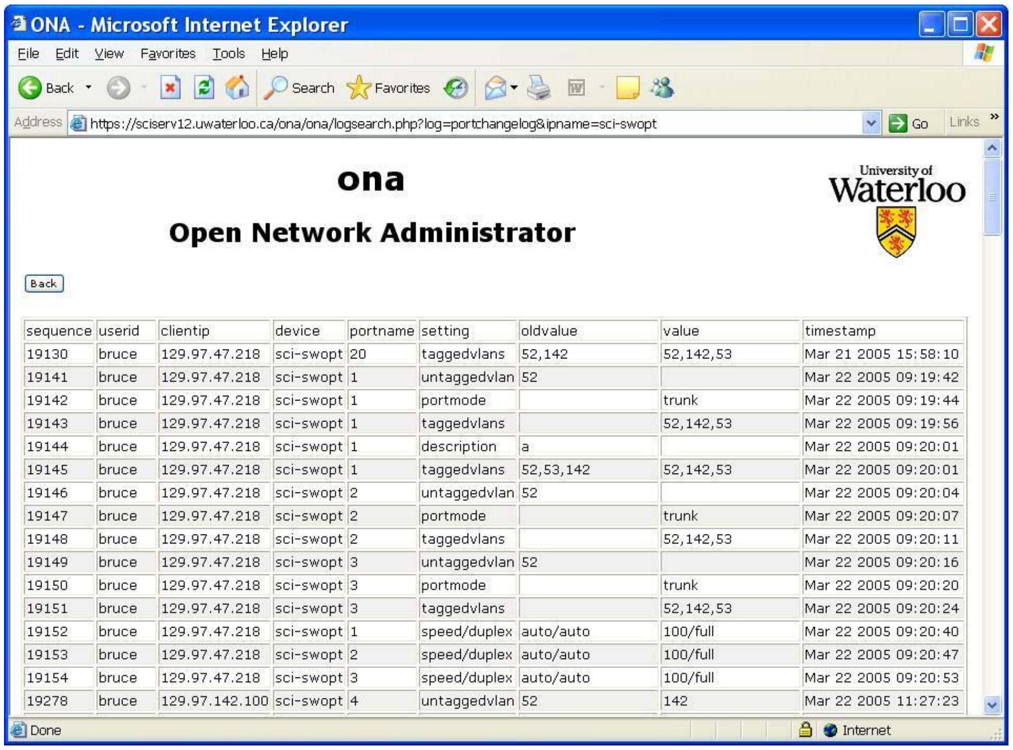
Implement the above using the managerial interface, as follows:

- Under "Devices," add eng-sw1 through eng-sw10, setting the groupid to "Engineering"
- Under "Devices," add sci-sw1 through sci-sw10, setting the groupid to "Science"
- Under "Devices," add core-sw1, setting the groupid to "CSD"
- Under "Administrators," add "engadmin," setting the groupid to "Engineering"
- Under "Administrators," add "sciadmin," setting the groupid to "Science"
- Under "Administrators," add "manager," setting the groupid to "CSD"
- Under "Administrators," add "operator," setting the groupid to "CSD," setting denytrunkchanges to "1," and setting allowededits to "portstate"
- Under "Device Group Memberships," add a groupid called "all," setting the ipname to "." (the regular expression consisting of a single period matches everything)
- Under "Vlan Group Memberships," add a groupid called "all," setting the vlan to "." (the regular expression consisting of a single period matches everything)
- Under "Administrator Group Memberships," add two entries, one for each of "manager" and "operator," setting the groupid to "all"
- Under "Vlan Group Memberships," add a groupid called "Engineering," setting the vlan to ˆ(10|11)$ (that is a regular expression which matches 10 or 11)
- Under "Vlan Group Memberships," add a groupid called "Science," setting the vlan to ˆ(20|21)$ (that is a regular expression which matches 20 or 21)

The above can be taken further. For example, engadmin and sciadmin could be given control over their respective ports on core-sw1, through use of "Port Group Memberships." Or, if Engineering and Science wanted to share switches in a given area, there are several ways to use group memberships to permit such sharing, without granting full access to non-shared switches.

Change Logging

For each change, the date, time, userid, IP address of the client, the parameter being changed, its previous value, and its new value, are all logged. For each device, a "Changes" button lists all changes in chronological order (See Figure 2). On the port edit screen, the changes for that particular port are displayed at the bottom of the screen. (All history is kept


ONA - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://sciserv12.uwaterloo.ca/ona/ona/logsearch.php?log=portchangelog&ipname=sci-swopt
Go Links
ona
Open Network Administrator
University of Waterloo
Back
Done
Internet

sequence	userid	clientip	device	portname	setting	oldvalue	value	timestamp
19130	bruce	129.97.47.218	sci-swopt	20	taggedvlans	52,142	52,142,53	Mar 21 2005 15:58:10
19141	bruce	129.97.47.218	sci-swopt	1	untaggedvlan	52		Mar 22 2005 09:19:42
19142	bruce	129.97.47.218	sci-swopt	1	portmode		trunk	Mar 22 2005 09:19:44
19143	bruce	129.97.47.218	sci-swopt	1	taggedvlans		52,142,53	Mar 22 2005 09:19:56
19144	bruce	129.97.47.218	sci-swopt	1	description	a		Mar 22 2005 09:20:01
19145	bruce	129.97.47.218	sci-swopt	1	taggedvlans	52,53,142	52,142,53	Mar 22 2005 09:20:01
19146	bruce	129.97.47.218	sci-swopt	2	untaggedvlan	52		Mar 22 2005 09:20:04
19147	bruce	129.97.47.218	sci-swopt	2	portmode		trunk	Mar 22 2005 09:20:07
19148	bruce	129.97.47.218	sci-swopt	2	taggedvlans		52,142,53	Mar 22 2005 09:20:11
19149	bruce	129.97.47.218	sci-swopt	3	untaggedvlan	52		Mar 22 2005 09:20:16
19150	bruce	129.97.47.218	sci-swopt	3	portmode		trunk	Mar 22 2005 09:20:20
19151	bruce	129.97.47.218	sci-swopt	3	taggedvlans		52,142,53	Mar 22 2005 09:20:24
19152	bruce	129.97.47.218	sci-swopt	1	speed/duplex	auto/auto	100/full	Mar 22 2005 09:20:40
19153	bruce	129.97.47.218	sci-swopt	2	speed/duplex	auto/auto	100/full	Mar 22 2005 09:20:47
19154	bruce	129.97.47.218	sci-swopt	3	speed/duplex	auto/auto	100/full	Mar 22 2005 09:20:53
19278	bruce	129.97.142.100	sci-swopt	4	untaggedvlan	52	142	Mar 22 2005 11:27:23

Figure 2: History of device changes.

indefinitely. The need has not yet arisen to truncate history or roll older portions behind a ''more'' button.)

Managerial changes (e.g., adding a new switch, changing an SNMP community string), and batch telnet commands, are also logged, but these logs are not available for display through ONA at this time.

All ONA logs are stored in SQL tables. The structures of the adminlog, portchangelog and telnetlog are shown in Tables 2, 3, and 4.

These features make use of the above tables:

- The switch display screen shows MACs and IPs on each port. For trunk ports, the number of MACs seen in the last seven days, and a link to the complete list are shown (see Figure 3).
- The MAC/IP search tool shows all ports the MAC has been seen on (including trunks), with a quick link to the port if the MAC was found uniquely on that port. Also shown is the MAC, ARP, and port history. (e.g., if a network card was changed in a computer, ONA will show when that happened, both in terms of the change to the ARP table, and the MAC address present on the port); see Figures 4 and 5.

MAC/IP Querying, Logging, Searching

A cron job queries the MAC tables (and ARP tables for routers) regularly throughout the day. The data are stored in several tables, as shown in Table 5.

Switch Configurations and CVS Repository

A nightly cron job saves the switch configurations to a TFTP server, and can optionally push the configurations to additional TFTP/FTP servers. The configurations can also be placed in a tar ball, for automated pulling to network operations staffs' laptop computers. (Instructions for doing this with Cygwin [22] are included in the documentation for ONA.)

The pushing of switch configurations to alternate servers can be done based on the primary groups of the devices. For example, the configurations for switches in group ''Engineering'' could be pushed to a specific server belonging to Engineering.

Plain text device configurations are also saved to two different CVS repositories. One contains the raw configurations, and is not made available through ONA or CVSweb. The other contains the configurations minus sensitive information, such as community strings. The latter is available through ONA, with a link to CVSweb.

Managerial Interface

ONA managers (administrators with the ''systemadmin'' setting) can add/change/delete administrators and devices, and can adjust group memberships, etc., through the managerial interface, as shown in Figures 6 and 7.

Up/Down/Reboot Alerts

ONA logs and e-mails alerts when devices reboot, become inaccessible, or reappear. Administrators can disable receiving these e-mail alerts in the ONA preferences window. The set of devices about which a user receives notification is controlled by regular expression manipulation of the groups to which the devices belong.

FIELD	DESCRIPTION
sequence	sequence number
userid	userid of person who made the change
clientip	IP address of client used by userid
info	raw SQL (minus sensitive information) showing what was changed in the ONA admin tables
timestamp	date/time of change

Table 2: Fields of adminlog.

FIELD	DESCRIPTION
sequence	sequence number
userid	userid of person who made the change
clientip	IP address of client used by userid
device	name of device changed
portname	name of port changed
setting	settings, i.e., speed, vlan, etc.
oldvalue	old value
value	new value
timestamp	date/time of change

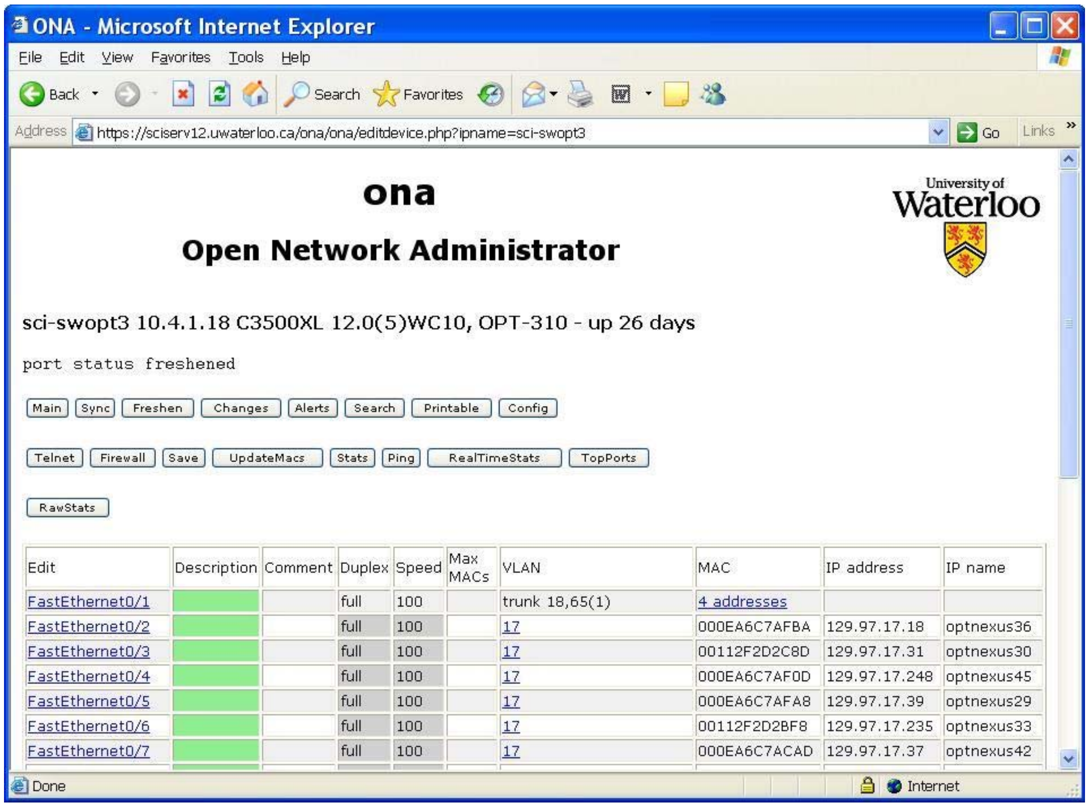
Table 3: Fields of portchangelog.

FIELD	DESCRIPTION
sequence	sequence number
userid	userid of person who made the change
clientip	IP address of client used by userid
device	name of device changed
announce	Boolean indicating whether this change should be announced in daily e-mail port (checkbox on telnet interface)
command	the telnet command that was run
timestamp	date/time of change

Table 4: Fields of telnetlog.

TABLE	DESCRIPTION
arp	IP to MAC
rarp	MAC to IP
arplog	changes in the ARP table
rawmac	MAC tables, with no analysis, i.e., raw data
mac	MAC tables, with results from trunk ports removed
maclog	changes in the MAC table

Table 5: Table names.

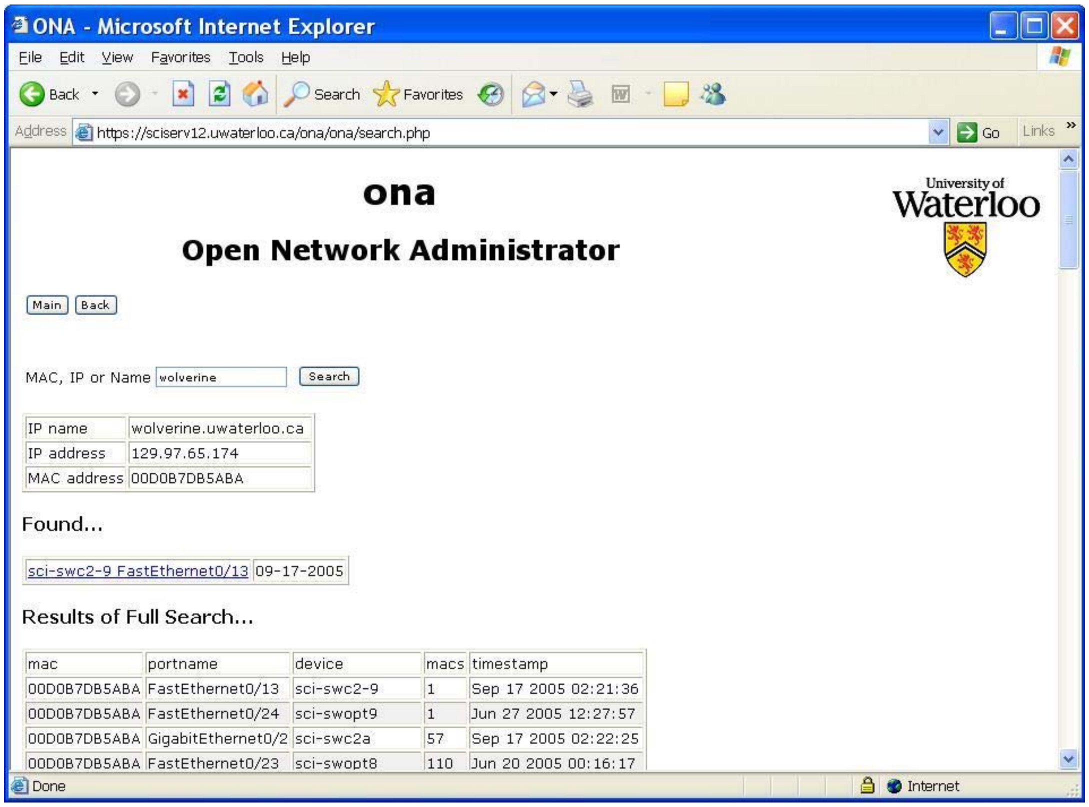

ONA - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://sciserv12.uwaterloo.ca/ona/ona/editdevice.php?ipname=sci-swopt3
Go Links
ona
Open Network Administrator
University of Waterloo
Main Sync Freshen Changes Alerts Search Printable Config
Telnet Firewall Save UpdateMacs Stats Ping RealTimeStats TopPorts
RawStats
Done Internet

sci-swopt3 10.4.1.18 C3500XL 12.0(5)WC10, OPT-310 - up 26 days

```
port status freshened
```

Edit	Description	Comment	Duplex	Speed	Max MACs	VLAN	MAC	IP address	IP name
FastEthernet0/1	🟩		full	100		trunk 18,65(1)	4 addresses		
FastEthernet0/2	🟩		full	100		17	000EA6C7AFBA	129.97.17.18	optnexus36
FastEthernet0/3	🟩		full	100		17	00112F2D2C8D	129.97.17.31	optnexus30
FastEthernet0/4	🟩		full	100		17	000EA6C7AF0D	129.97.17.248	optnexus45
FastEthernet0/5	🟩		full	100		17	000EA6C7AFA8	129.97.17.39	optnexus29
FastEthernet0/6	🟩		full	100		17	00112F2D2BF8	129.97.17.235	optnexus33
FastEthernet0/7	🟩		full	100		17	000EA6C7ACAD	129.97.17.37	optnexus42

Figure 3: Switch page.


ONA - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://sciserv12.uwaterloo.ca/ona/ona/search.php
Go Links
ona
Open Network Administrator
University of Waterloo
Main Back
Search
Done Internet

MAC, IP or Name wolverine

IP name	wolverine.uwaterloo.ca
IP address	129.97.65.174
MAC address	00D0B7DB5ABA

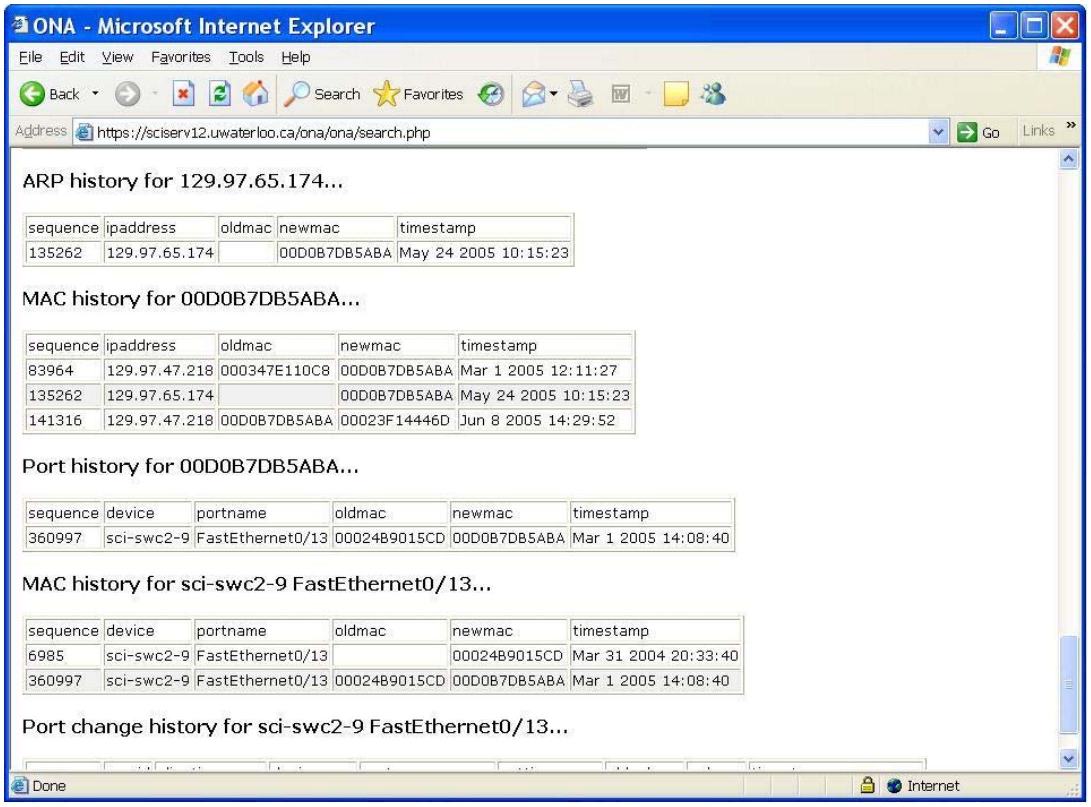
Found...

sci-swc2-9 FastEthernet0/13	09-17-2005

Results of Full Search...

mac	portname	device	macs	timestamp
00D0B7DB5ABA	FastEthernet0/13	sci-swc2-9	1	Sep 17 2005 02:21:36
00D0B7DB5ABA	FastEthernet0/24	sci-swopt9	1	Jun 27 2005 12:27:57
00D0B7DB5ABA	GigabitEthernet0/2	sci-swc2a	57	Sep 17 2005 02:22:25
00D0B7DB5ABA	FastEthernet0/23	sci-swopt8	110	Jun 20 2005 00:16:17

Figure 4: Search results.


ONA - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://sciserv12.uwaterloo.ca/ona/ona/search.php
Go Links
Done
Internet

ARP history for 129.97.65.174...

sequence	ipaddress	oldmac	newmac	timestamp
135262	129.97.65.174		00D0B7DB5ABA	May 24 2005 10:15:23

MAC history for 00D0B7DB5ABA...

sequence	ipaddress	oldmac	newmac	timestamp
83964	129.97.47.218	000347E110C8	00D0B7DB5ABA	Mar 1 2005 12:11:27
135262	129.97.65.174		00D0B7DB5ABA	May 24 2005 10:15:23
141316	129.97.47.218	00D0B7DB5ABA	00023F14446D	Jun 8 2005 14:29:52

Port history for 00D0B7DB5ABA...

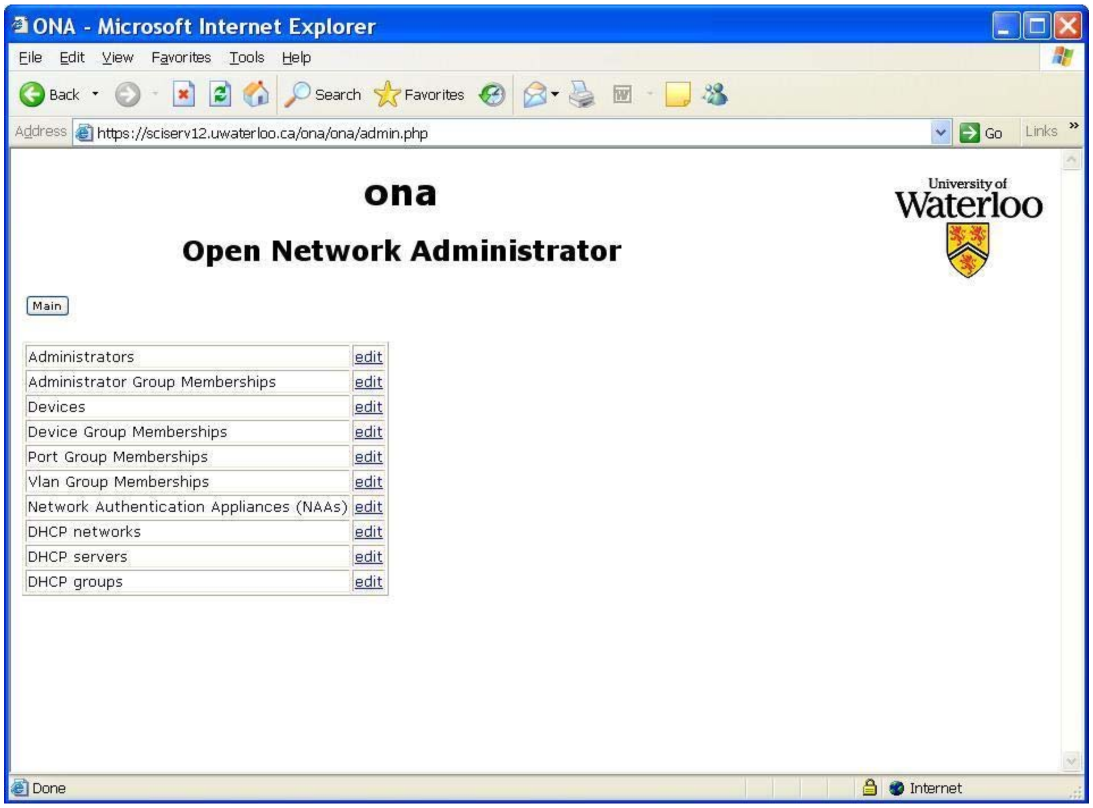
sequence	device	portname	oldmac	newmac	timestamp
360997	sci-swc2-9	FastEthernet0/13	00024B9015CD	00D0B7DB5ABA	Mar 1 2005 14:08:40

MAC history for sci-swc2-9 FastEthernet0/13...

sequence	device	portname	oldmac	newmac	timestamp
6985	sci-swc2-9	FastEthernet0/13		00024B9015CD	Mar 31 2004 20:33:40
360997	sci-swc2-9	FastEthernet0/13	00024B9015CD	00D0B7DB5ABA	Mar 1 2005 14:08:40

Port change history for sci-swc2-9 FastEthernet0/13...

Figure 5: History component of search results.


ONA - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address https://sciserv12.uwaterloo.ca/ona/ona/admin.php
Go Links
University of Waterloo
Main
Done
Internet

ona

Open Network Administrator

Administrators	edit
Administrator Group Memberships	edit
Devices	edit
Device Group Memberships	edit
Port Group Memberships	edit
Vlan Group Memberships	edit
Network Authentication Appliances (NAAs)	edit
DHCP networks	edit
DHCP servers	edit
DHCP groups	edit

Figure 6: Managerial interface – main screen.

Daily E-mail Report

Each night, ONA mails a summary of changes, alerts, and configuration differences from the previous day (minus sensitive information), to the administrators. Appendix 1 shows an example e-mail report.

Batch Telnet Interface

When specifically authorized in the **authgroups** table, administrators can perform batch telnet commands on a device or group of devices, for devices which support a command line interface. ONA is programmed to understand the telnet interfaces on the devices that it supports, so that it can supply the userid and password, transparently, and wait for command line prompts before sending the next in a series of telnet commands.

All commands run through the ONA batch telnet interface are logged. The administrator can choose whether to have telnet commands announced in the daily e-mail report, by use of a checkbox labelled Announce.

Commands run on a switch through telnet have the potential to make ONA's switch configuration database out of date. For example, if an administrator were to run commands via telnet to change the untagged VLAN on a port, ONA would continue to show the old VLAN for 24 hours, based on the values stored in its internal database. For this reason, the ONA telnet interface includes a radio button that can force ONA to synchronize its database immediately (Sync Now), or upon the next access to the switch through ONA's web interface (Sync Later).

Commands entered through the batch telnet interface bypass ONA's granular access control, so this interface is disabled by default, and is made available only to administrators who already know the actual switch passwords. An example of the use of this interface would be upgrading the firmware on a large number of devices, followed by rebooting the devices. The following commands were used to upgrade the firmware on approximately 100 Avaya access points:

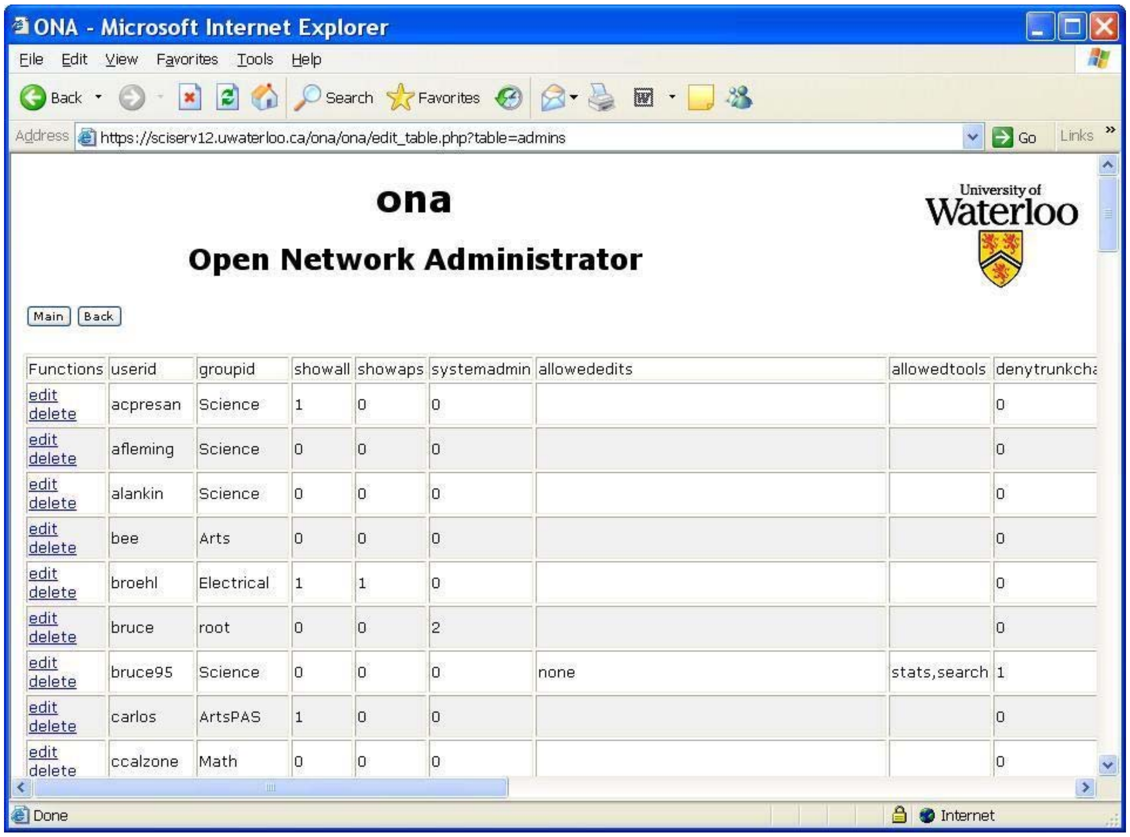
```
download 129.97.50.121 \
             tmpdata/AVAP3.bin img
reboot 2
```

After entering the above commands into the command window, and selecting the 100 access points via checkboxes, ONA ran the command on all 100 access points with no further input required.

Configuration Translator

ONA can generate a suggested configuration fragment for a device using the text configuration commands of a different vendor. A manager may use this to relatively easily replace a switch with one from a different vendor.

The configuration fragment generated by ONA consists only of the VLAN database, and speed, duplex, description, and VLAN(s) of all ports.



https://sciserv12.uwaterloo.ca/ona/ona/edit_table.php?table=admins

ona

Open Network Administrator

Functions	userid	groupid	showall	showaps	systemadmin	allowededits	allowedtools	denytrunkcha
edit delete	acpresan	Science	1	0	0			0
edit delete	afleming	Science	0	0	0			0
edit delete	alankin	Science	0	0	0			0
edit delete	bee	Arts	0	0	0			0
edit delete	broehl	Electrical	1	1	0			0
edit delete	bruce	root	0	0	2			0
edit delete	bruce95	Science	0	0	0	none	stats,search	1
edit delete	carlos	ArtsPAS	1	0	0			0
edit delete	ccalzone	Math	0	0	0			0

Figure 7: Managerial interface – editing administrators.

Display 2 shows sample configuration fragments for port 1, which is set 100/full, on VLAN 49.

The ONA Configuration Translator does not actually translate a given switch configuration text file from one vendor format to another. Rather, it generates the configuration fragment based on the VLAN and port information stored in the ONA database. For this reason, adding support for a new vendor involves writing fairly simple code to generate the text commands specific to that vendor, without knowledge of any other vendors' configuration file formats.

Security

Access to the ONA application is controlled through Apache .htaccess authentication. A configuration file contains the MySQL userid and password so that the ONA application may connect to the database.

The database contains the userids, passwords, and SNMP community strings for all devices. These are stored encrypted in the database, with a key contained in the configuration file.

ONA should be hosted on a server that does not permit general user login, or host web pages for general users, to help protect against unauthorized access to the configuration file and the database.

Access controls to limit administrative access should be applied on switches and access points, independently of whether one uses ONA. If a switch is stolen, the password or SNMP communities may be compromised. If all switches use the same passwords and communities, access controls restricting access to specific authorized hosts or subnets can reduce the impact.

Having a single network management tool handle a number of tasks means that userids, passwords and communities need be stored in only one location. This reduces the administrative burden of entering this information into several tools. By reducing this administrative burden, ONA makes it more likely that weak or widely known community strings will actually be changed, thus eliminating the common *I was meaning to get to that* explanation after a security compromise.

ONA's access controls limit what functions administrators can perform. Even in the case where administrators are given permission to do more than they actually need to do, ONA's logs indicate who did what and when.

ONA is also a handy tool when dealing with security incidents. When a computer is suspected of having a virus or is otherwise behaving badly on the network, ONA's "search" facility lets an administrator quickly search for the machine's connection point by either hostname, IP address, or MAC address. ONA shows the switch port on which the machine is connected, and one click takes the administrator to that port where it may be quickly disabled. A dialog box is available for the administrator to enter an explanatory message which is then e-mailed to the machine's user/owner/administrator as obtained by ONA from DNS.

Use of ONA At the University of Waterloo

Network management at UW is decentralized. The central IT group, Information Systems and Technology (IST), has responsibility for the external campus connection, internal campus backbone networking infrastructure out to points of presence in the faculties, and for all of student residence networking. The faculties may have responsibility for their own network infrastructure between their IST-provided point of presence and the end-user. Faculties such as Engineering and Mathematics have fairly large network infrastructures of their own; other faculties may have less or none (in which case IST provides it).

Before the development of ONA, network administrators used the vendor CLI tools for switch management. This meant learning the details of the tools for Cisco, Bay Networks (Nortel) and Extreme switches among others. Bay offered an ASCII GUI. Extreme offered both CLI and the EPICenter tool.

- Cisco:

```
interface FastEthernet0/1
speed 100
duplex full
switchport mode access
switchport access vlan 49
exit
```

- HP Procurve:

```
interface 1
   speed-duplex 100-full
exit
vlan 49
   name "cstc1net"
   untagged 1
exit
```

- Extreme:

```
configure port 1 auto off speed 100 duplex full
create vlan "cstc1net"
configure vlan "cstc1net" tag 49
configure vlan "cstc1net" add port 1 untagged
```

- Nortel Baystack:

```
interface FastEthernet1
speed 100
duplex full
exit
vlan ports 1 filter-tagged-frame enable
vlan ports 1 filter-untagged-frame disable
vlan ports 1 tagging disable
vlan create 49 name cstc1net type port
vlan members add 49 1
vlan ports 1 pvid 49
```

Display 2: Sample configuration fragments.

EPICenter was used for initial high level design but was abandoned due to its complexity and other shortcomings described earlier. Permission to modify switch settings was restricted to staff members who were directly responsible for network infrastructure.

ONA was well-received by senior network administrators. It offered a single easy interface to all the supported switches covering most of the device functionality. Significantly, it freed up senior network staff from the high volume of simple port configuration changes by enabling other technical staff to perform those simple changes themselves. Vendor CLI tools are still used in some circumstances when certain functionality is either not available via SNMP/telnet or not yet implemented in ONA.

ONA was developed in one department of the Engineering faculty, and was adopted voluntarily by most other Engineering departments, the Mathematics faculty, and others until at present four of the six faculties use it. In a decentralized environment where there are often tendencies towards preserving local control and resisting external influence, this widespread enthusiastic adoption of ONA is quite remarkable. The central IST department is now adopting ONA and plans are underway to bring all remaining campus network switches into ONA by the end of the year.

From the period May 1, 2004 to May 1, 2005, approximately 10,000 network port changes were made through ONA. ONA is the primary mechanism for switch port management, to the extent that the built-in web-based management tool that comes with the switches has been disabled, and is no longer depended on.

Even most veteran network operations staff, who are adept at the command line interfaces for multiple models of switches, choose to use ONA over telnetting into the switch directly.

Maintenance and Support for ONA

ONA was developed by one person, without the expectation in advance that it would be offered as an open-source tool to the community. It is implemented in roughly thirty PHP source files [23] and admittedly needs some polish to make it more readily amenable to both collaboration by multiple maintainers, and to outside use. However, it has been successfully installed by someone else without a great deal of difficulty.

ONA depends on having Apache web server, MySQL database, ModPHP (with SNMP, MySQL and FTP support enabled), CVSweb, and RRDTool installed. Then to bootstrap ONA, currently one must manually create the first manager and the default set of tables in SQL. (These bootstrapping steps are documented in detail [24].)

ONA is actively supported and developed by Bruce Campbell with contributions from Dawn Keenan of the Information Systems and Technology group at UW. With more recent perspective arising from ONA's enthusiastic acceptance at UW, and with the realization that there do not appear to be similar tools evolving elsewhere, there has emerged a small group of people spanning several UW IT groups who are interested in becoming active maintainers of ONA. Also, ONA is recognized at the administrative level of the University's IT organizations as an important tool worthy of a commitment of resources. We expect that these factors will lead to ensuring that ONA is maintainable and well-supported. Constructive feedback from readers of this paper would also help us move towards making ONA a well-packaged open-source tool.

System Requirements

ONA is written in PHP and has been tested on both FreeBSD and Linux. It should work on any UNIX platform.

To give some idea of the CPU etc. requirements to run an instance of ONA, some details of UW's production ONA installation are provided.

The system which runs http://ona.uwaterloo.ca/ is as follows:

- FreeBSD 4.11
- dual 2.80 GHz Xeon
- 512 MB memory
- 3ware SATA RAID

Devices:

- 277 switches (17288 ports)
- 299 wireless access points
- 6 routers

The cron jobs that query the port traffic statistics, and query the MAC address tables of all devices, place a moderate load on the server, with the load average typically sitting around 0.50 .

The cron job that updates the RRD databases takes about seven minutes to query and update the port statistics for the approximately 17,000 ports.

The MySQL database is approximately 400 MB, the bulk of which is the rawmac table.

Conclusions

ONA seems to fill an important niche that is currently comparatively vacant, and seems to fill it well. It is a secure, easy-to-use, open source, web-based network switch management tool.

Its ready acceptance by both traditional command-line-oriented "power user" network administrators and other IT support staff, in a strongly decentralized setting, suggests that its advantages are very appealing. Its numerous features and the simplicity of its user interface lead staff to prefer it over traditional methods.

ONA is designed to be friendly to changes made to switches by means other than ONA. This benign coexistence permits deployment in a staged manner, in which network operations staff can continue to manage switches directly as desired, and use ONA when they decide to. The deployment of ONA at UW did not require an overhaul of network management practices. ONA doesn't do everything – some operations, such as creating quality of service profiles, access control lists, etc., must still be done directly. But since ONA is comfortable with device changes made unbeknownst to it, ONA itself does not become an obstacle to using device features which it does not yet support.

ONA reduces the burden of network management by performing a variety of maintenance operations and handling the vast majority of typical switch port configuration changes.

It is surprising that there seem to be no other open source web-based edge device management tools available. We hope others will examine and use ONA, give us feedback as we move towards making it a well-formed open source tool, or perhaps be inspired to develop and contribute their own.

Availability

ONA is available for download at http://www.freebsd.uwaterloo.ca/twiki/bin/view/Freebsd/OnaInstallation [23]. Documentation is available at http://www.freebsd.uwaterloo.ca/twiki/bin/view/Freebsd/OpenNetworkAdministrator [24].

Author Information

Bruce Campbell is the Manager of the Science Computing department at the University of Waterloo. He has been at UW since 1984, initially as a systems administrator in the Engineering faculty. Bruce (the dirt-biker) is the developer of ONA, and can be reached at bruce@scimail.uwaterloo.ca or +1 519-888-4567 x6991 .

Robyn Landers has been a UNIX systems administrator in the Math Faculty Computing Facility at the University of Waterloo since 1989. Robyn (the sport-biker) can be reached at rblanders@math.uwaterloo.ca or +1 519-888-4567 x2030 .

References

[1] Oetiker, Tobi, *RRDtool*, Swiss Federal Institute of Technology, http://people.ee.ethz.ch/˜oetiker/webtools/rrdtool/ .

[2] Fenner, B., H. Zeller, A. Musha, V. Skytta, *CVSweb*, http://www.freebsd.org/projects/cvsweb.html .

[3] *EPICenter*, http://www.extremenetworks.com/libraries/prodpdfs/products/epicenter.asp .

[4] Oetiker, Tobi, *MRTG*, Swiss Federal Institute of Technology, http://people.ee.ethz.ch/˜oetiker/webtools/mrtg/ .

[5] Galstad, Ethan, *Nagios*, http://www.nagios.org/ .

[6] Berry, Ian, *cacti*, http://raxnet.net/ .

[7] Sorrell, Chadwick, *IP Manager*, http://unix.freshmeat.net/projects/ipman/ .

[8] *Kiwi CatTools*, http://www.kiwisyslog.com/cattools2.htm .

[9] Madden, John, *LANdb*, http://landb.sourceforge.net/about.shtml .

[10] *WhatsUp Gold*, http://www.ipswitch.com/Products/WhatsUp/index.asp .

[11] *HP OpenView*, http://www.openview.hp.com/ .

[12] Estrella, G., J. Friedrich, L. Liimatainen, G. McLean, J. Schulien, C. Small, *GxSNMP*, http://www.gxsnmp.org/ .

[13] *OpenNMS*, http://wiki.opennms.org/ .

[14] Kilmer, H., J. Heasley, A. Partan, P. Whiting, A. Schutz, *RANCID*, http://www.shrubbery.net/rancid/ .

[15] Abrahamson, C., D. Parter, M. Blodgett, A. Kunen, N. Mueller, "Splat: A Network Switch/Port Configuration Management Tool," *Seventeenth Large Installation Systems Administration Conference (LISA '03)*, p. 247, San Diego, CA, USENIX, http://www.usenix.org/publications/library/proceedings/lisa03/tech/full_papers/abrahamson/abrahamson_html/index.html, October 26-21, 2003.

[16] *SNMPc*, http://castlerock.com/ .

[17] *PHP*, http://www.php.net/ .

[18] *MySQL*, http://www.mysql.com/ .

[19] *Apache*, http://www.apache.org/ .

[20] *RADIUS – Remote Authentication Dial In User Service*, http://www.ietf.org/rfc/rfc2865.txt .

[21] *LDAP – Lightweight Directory Access Protocol*, http://www.ietf.org/rfc/rfc2251.txt .

[22] *Cygwin*, http://www.cygwin.com/ .

[23] *ONA download*, http://www.freebsd.uwaterloo.ca/twiki/bin/view/Freebsd/OnaInstallation/ .

[24] *ONA documentation*, http://www.freebsd.uwaterloo.ca/twiki/bin/view/Freebsd/OpenNetworkAdministrator/ .

Appendix 1: Example E-Mail Report

```
Subject: ona - 1 admin changes, 2 reboots, 4 port changes, 10 config lines

1 admin changes
2 switches have rebooted
4 port changes
2 switches 10 config lines changed

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1 admin changes...

#707  Jul 11 2005 09:52:21 vic      UPDATE devices
              SET ‘ipname‘=aco-swhh11 WHERE ‘ipname‘ LIKE aco-swhh1

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

2 switches have rebooted...

aco-swhh11                       last reboot was at Mon Jul 11 6:56:37
dccore-exsw05                    last reboot was at Mon Jul 11 10:23:27

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

4 port changes...

 seq       date/time        userid     switch     port    setting        value
______ ____________________ ______ _____________ _____ ____________ ______________
#26378 Jul 11 2005 09:58:14 bee      aco-swpas6 Fe0/7 speed/duplex       100/full
#26453 Jul 11 2005 14:13:20 peregi dccore-exsw20 5:4    description     5:4 ecenet
#26454 Jul 11 2005 14:13:20 peregi dccore-exsw20 5:4        comment DC-3579 A34 29
#26455 Jul 11 2005 14:13:21 peregi dccore-exsw20 5:4   untaggedvlan             90

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

2 switches 10 config lines changed...

#  aco-swpas6 changes   +3  -1
  !
  ! Last configuration change at 18:56:25 EST Sat Jun 18 2005
- ! NVRAM config last updated at 15:52:54 EST Thu Jul 7 2005
+ ! NVRAM config last updated at 08:58:11 EST Mon Jul 11 2005
  !
  version 12.0
  no service pad

   switchport access vlan 64
  !
  interface FastEthernet0/7
+  duplex full
+  speed 100
   switchport access vlan 64
  !
  interface FastEthernet0/8
==========================================================

#
#  dccore-exsw20 changes   +3  -3
#

- # Alpine3808 Configuration generated Fri Jul 8 22:43:08 2005
+ # Alpine3808 Configuration generated Mon Jul 11 22:46:58 2005

- configure vlan "VLSINet" add port 5:4 untagged

+ configure vlan "ECENet" add port 5:4 untagged

- configure port 5:4 display-string "5:4 vlsinet"
+ configure port 5:4 display-string "5:4 ecenet"
```