IN ICINGA AND CACTI

Presentation for the TF-NOC meeting in Cambridge 2014
Presented by Erik Ruiter
Presentation overview

- Background
- Ethernet OAM overview
- Icinga plugins
- Cacti templates
- Issues and experiences during testing / developing
About SURFsara

SARA was founded in 1971

In 2008 Vancis was created to handle market oriented activities

In 2013 SARA was merged with SURF and became SURFsara

Mission statement:
SURFsara supports research in the Netherlands by developing and offering advanced ICT infrastructure, services and expertise.
## Services

<table>
<thead>
<tr>
<th>National supercomputer Cartesius (capability computing)</th>
<th>National compute cluster Lisa (capacity computing)</th>
<th>Grid compute &amp; storage Gina (middleware services)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Cartesius Image" /></td>
<td><img src="image2" alt="Lisa Image" /></td>
<td><img src="image3" alt="Gina Image" /></td>
</tr>
<tr>
<td><strong>HPC Cloud IaaS (Do-it-yourself)</strong></td>
<td><strong>Hadoop – Data processing (map-reduce algorithm)</strong></td>
<td><strong>GPU cluster (Computing on a video card)</strong></td>
</tr>
<tr>
<td><img src="image4" alt="HPC Cloud" /></td>
<td><img src="image5" alt="Hadoop" /></td>
<td><img src="image6" alt="GPU Cluster" /></td>
</tr>
<tr>
<td><img src="image7" alt="Collaboratorium" /></td>
<td><img src="image8" alt="Render cluster" /></td>
<td><img src="image9" alt="Beehub / SURFDrive" /></td>
</tr>
<tr>
<td><strong>Collaboratorium Remote collaboration (video wall)</strong></td>
<td><strong>Render cluster (Data visualization)</strong></td>
<td><strong>Beehub / SURFDrive (Dropbox unlimited)</strong></td>
</tr>
<tr>
<td><img src="image10" alt="Collab Image" /></td>
<td><img src="image11" alt="Render Image" /></td>
<td><img src="image12" alt="Beehub Image" /></td>
</tr>
</tbody>
</table>

**BeeHub is...**

...a work in progress, but usable! This is "Release Candidate 1". It has all the bar...
SURFsara has developed:

- An Icinga / Nagios plugin for monitoring the CFM status of Ethernet OAM enabled devices in Icinga.

- A graphing template for Cacti to graph L2 Delay and jitter measurements.
Why?

- We wanted to demonstrate that it is possible to implement simple OAM monitoring without spending too much resources.

- We wanted to have this available in our existing OSS environment, so that we did not have to invest in additional software.

No additional software required for L2 monitoring, this saves time and resources when implementing Ethernet OAM.
What is Ethernet OAM

- A set of tools for Operations, Administration and Management (OAM) for Ethernet networks.
- Two standards available for Connectivity Fault management (CFM):

<table>
<thead>
<tr>
<th>IEEE 802.1ag</th>
<th>ITU-T Y.1731</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity Fault Management (CFM)</td>
<td>Operations, Administration, and Maintenance (OAM)</td>
</tr>
<tr>
<td>Connectivity messages</td>
<td>Connectivity messages + Performance messages</td>
</tr>
</tbody>
</table>

Ethernet OAM Messages

• Loopback (LB)
  • Layer 2 ping

• Linktrace (LT)
  • Layer 2 traceroute

• Continuity Check (CC)
  • one-way hello (comparable to BFD)

• Delay Measurement (DM)
  • one way delay, two way delay, jitter (Only Y.1731)

• Etc… (in Y.1731)
## Ethernet OAM Terminology

<table>
<thead>
<tr>
<th>MEP</th>
<th>Maintenance End Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIP</td>
<td>Maintenance Intermediate Point</td>
</tr>
<tr>
<td>MAID</td>
<td>Maintenance Association Identifier</td>
</tr>
<tr>
<td></td>
<td>&quot;Domain&quot; Identifier</td>
</tr>
<tr>
<td>MD Level</td>
<td>Level of a Maintenance Domain (0…7)</td>
</tr>
<tr>
<td></td>
<td>&quot;Scope&quot;: link segment or path monitoring</td>
</tr>
<tr>
<td>MEP ID</td>
<td>MEP Identifier (1…8191)</td>
</tr>
</tbody>
</table>
Ethernet OAM Layered approach

There are eight levels (0-7) which can be used to segment a OAM domain.

Customer OAM level 7

Provider OAM level 5

Operator OAM level 3

inter-node OAM level 0
Configuring a MEP and CCM session on a Juniper EX-4200

```bash
{master:0}[edit protocols oam ethernet connectivity-fault-management]
user@ex4200# show
maintenance-domain md7 {
    level 7;
    maintenance-association customer_L7{
        continuity-check {
            interval 100ms;
        }
        mep 700 {
            interface ge-0/0/0.0 vlan-id 1234;
            direction down;
            auto-discovery;
            remote-mep 701;
        }
    }
}
```

Customer OAM level 7

MA/MEG="customer_L7"
Ethernet OAM in production

SURFsara is currently working on having Ethernet OAM connectivity with adjacent LHCOPN nodes. This will allow better monitoring and troubleshooting in case of outages on remote links.
The following plugins are available for Icinga

- check_ethping
- check_ethtrace
- check_cfm_state

Usage and examples are explained on the following slides
Icinga plugins: check_ethping

= LoopBack Request (LBR) towards 0000.0000.0300

= LoopBack Message (LBM) towards Icinga Host

TF-NOC meeting Cambridge 2014 – Monitoring Ethernet OAM in Icinga and Cacti
Icinga plugins: check_ethping

$ ./check_ethping.py --help
Usage: check_ethping.py [options] destination_MAC

Options:
- h, --help show this help message and exit
- i INTERFACE, --interface=INTERFACE
  interface to use
- v VLAN, --vlan=VLAN vlan to query
- l MDLEVEL, --mdlevel=MDLEVEL
  OAM Maintentance Level
- c COUNT, --count=COUNT
  number of ethpings to send
- w WARN_ON_PACKETLOSS, --warn_on_packetloss=WARN_ON_PACKETLOSS
  Return warning on packetloss 1=yes 0=no

(default=1)

Notes:
The Icinga host needs to be inband, since it is participating in the OAM network using dot1ag-utils.
Icinga plugins: check_ethtrace

TF-NOC meeting Cambridge 2014 – Monitoring Ethernet OAM in Icinga and Cacti
$ ./check_ethtrace.py --help
Usage: check_ethtrace.py [options] destination_MAC

Options:
- h, --help                      show this help message and exit
- i INTERFACE, --interface=INTERFACE(interface to use)
- v VLAN, --vlan=VLAN vlan to query
- l MDLEVEL, --mdlevel=MDLEVEL   OAM Maintentance Level
--hops=HOPS                     Allowed number of hops (number or range eg. 2:3)
--mac_path=MACPATH              Specified trace path (use comma separated mac addresses)

Notes:
The Icinga host needs to be inband, since it is participating in the OAM network using dot1ag-utils.
Icinga plugins: check_ethtrace

= LoopBack Request (LBR) towards 0000.0000.03000

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngo-1.lighthouse.sara.nl</td>
<td>check_ethping</td>
<td>OK</td>
<td>PING 00:00:00:00:03:00 OK - Packet loss - 0%, RTA - 7.8806 ms</td>
</tr>
<tr>
<td></td>
<td>check_ethtrace</td>
<td>WARNING</td>
<td>ETH-TRACE b646.935:103 WARNING - hops = 2 - Wrong path detected (configured: 00:00:00:00:00:00:08:00,00:00:00:00:05:00:00:00:03:00 detected:00:00:00:00:00:00:00:03:00)</td>
</tr>
</tbody>
</table>
Icinga plugins: check_cfm_state

The check_cfm_state plugin monitors the CCM state of a Remote MEP and reports changes in the CCM status.
Icinga plugins: check_cfm_state

$ ./check_cfm_state_8021ag.py --help
Usage: check_cfm_state_8021ag.py [options] hostname

Options:
  -h, --help            show this help message and exit
  -v SNMP_VERSION, --version=SNMP_VERSION
                        Use specific SNMP version default = 1
  -p PORT, --port=PORT  SNMP port default = 161
  -c COMMUNITY, --community=COMMUNITY
                        SNMP community
  -m LIST, --mep=LIST   comma separated list to specify remote MEPs to
                        monitor, (all = all available MEPs)

Notes:
• No support for monitoring missed CCM messages between polls
• No filtering for MA’s or levels, all detected remote MEPs are reported
• There are three versions implemented: check_cfm_state_8021ag,
  check_cfm_state_ciena and check_cfm_state_juniper.
• Sometimes misleading alarms on monitored MEPs
Icinga plugins: check_cfm_state

This example shows how CCM sessions behave in a broadcast domain.

- MEP 800 has CCM sessions with all MEPs
- Other MEPs only have a CCM session with MEP 800

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch_mep800.surfsara.nl</td>
<td>check_cfm_state</td>
<td>OK</td>
<td>Remote MEP 800 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 802 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 803 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td>switch_mep801.surfsara.nl</td>
<td>check_cfm_state</td>
<td>OK</td>
<td>Remote MEP 800 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 802 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 803 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td>switch_mep802.surfsara.nl</td>
<td>check_cfm_state</td>
<td>OK</td>
<td>Remote MEP 800 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 802 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 803 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td>switch_mep803.surfsara.nl</td>
<td>check_cfm_state</td>
<td>OK</td>
<td>Remote MEP 800 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 802 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 803 OK - Level: 7 MAID: customer_L7</td>
</tr>
</tbody>
</table>
Icinga plugins: check_cfm_state

This example shows how CCM sessions behave in a broadcast domain.

- When 803 fails, 800 sets the RDI flag in its CCM frames.
- 801 and 802 receive these frames and set the RDI state for 800.
- When MEP 803 fails, all other MEPs think there is an issue with MEP 800.

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch_mep800.surfsara.nl</td>
<td>check_cfm_state</td>
<td>WARNING</td>
<td>Remote MEP 801 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 802 OK - Level: 7 MAID: customer_L7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote MEP 803 NO DATA</td>
</tr>
<tr>
<td>switch_mep801.surfsara.nl</td>
<td>check_cfm_state</td>
<td>WARNING</td>
<td>Remote MEP 800 WARNING - Level: 7 MAID: customer_L7 -- RDI Error Detected</td>
</tr>
<tr>
<td>switch_mep802.surfsara.nl</td>
<td>check_cfm_state</td>
<td>WARNING</td>
<td>Remote MEP 800 WARNING - Level: 7 MAID: customer_L7 -- RDI Error Detected</td>
</tr>
<tr>
<td>switch_mep803.surfsara.nl</td>
<td>check_cfm_state</td>
<td>WARNING</td>
<td>Remote MEP 800 NO DATA</td>
</tr>
<tr>
<td>Name</td>
<td>Functionality</td>
<td>Supported platform</td>
<td>Method</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------</td>
<td>-------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>check_cfm_state_8021ag.py</td>
<td>check CCM state</td>
<td>Any 8021.ag supported device</td>
<td>SNMP standardized MIB</td>
</tr>
<tr>
<td>check_cfm_state_ciena.py</td>
<td>check CCM state</td>
<td>Ciena</td>
<td>SNMP proprietary MIB</td>
</tr>
<tr>
<td>check_cfm_state_juniper.py</td>
<td>check CCM state</td>
<td>Juniper</td>
<td>Netconf</td>
</tr>
<tr>
<td>check_ethping</td>
<td>check LBR/LBM</td>
<td>Any 8021.ag supported device</td>
<td>dot1ag utils</td>
</tr>
<tr>
<td>check_ethtrace</td>
<td>check LTR/LTM</td>
<td>Any 8021.ag supported device</td>
<td>dot1ag utils</td>
</tr>
</tbody>
</table>
Ethernet OAM support in Cacti

Custom template for graphing DMM results: L2 delay and jitter.

- Shows 2-way Jitter and Delay for each individual DMM session detected on the device
- Is working for Ciena 3960 and Juniper EX.
- You can use the Cacti realtime plugin for realtime graph updates!
Building a custom Cacti template

Takes a lot of effort …

<table>
<thead>
<tr>
<th>Simple single value query: Eg. Load of CPU of control plan of switch</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data template</strong></td>
</tr>
<tr>
<td><strong>Graph template</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complex query, Eg throughput, errors and packet-loss of all interfaces on a switch</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data query</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Different methods for retrieving DMM data:

- Ciena DMM statistics -> Get SNMP data input method
- Juniper DMM statistics -> Get Script data input method

Juniper does not support SNMP for Ethernet OAM functions.

But… Cacti only Supports SNMP authentication 😞

Solution:

Juniper DMM poller script uses Netconf, and looks up username / password using a separate file with authentication information (netconf_auth):

127.0.0.1:user:password:22
192.168.1.1:user2:password2:22

The hostname / ip address is used as key
Used libraries / tools

Net-SNMP and Python Net-SNMP bindings:

• General SNMP toolset, with an extension to build python scripts.

Ncc-client:

• Is a Netconf client API in Python. Is required for querying Eth-OAM settings / results in Juniper Systems.
• There are a number of forks existing, from which the Juniper fork was used.
Dot1ag-utils:

- A software implementation of 802.1ag
- Allows you to enable ETH OAM on a Linux system, or an open Linux based switching platform (eg. Arista EOS).
- Provides L2 ping, L2 trace and CCM functionality
- Was developed inhouse at SARA by a former colleague. (Ronald van der Pol, now works at SURFnet)
The tools were developed and tested on a shared OAM testbed, together with TNO.

- Juniper Ex4200
- Ciena 3960
- Ciena 3960
- Cisco catalyst 2950
  - (for mgmt purposes)
- Overture ISG24

Uplink towards TNO using SURFnet infrastructure
Notes and Issues

Vendor support is growing, but not complete:
• Juniper provides dot1ag and Y.1731 support on MX an EX platform, but only on CLI and Netconf, no SNMP (Planned in Roadmap for 2014)
• Cisco has Ethernet Support on 3400 ME models, but not on mainstream C3750 or smaller datacenter switches.

Tested devices:
• Juniper MX 960
• Juniper EX4200
• Ciena 3960
• Overture 24
• MRV OptiSwitch 904
• Fujitsu FDX2400M

We sometimes encounter issues in the OAM implementation of devices.
• Bugs in device firmware.
• Errors in MIB files
Ethernet OAM Monitoring

Want to know more?

https://github.com/sara-nl/eth-oam

Thank you!
Erik.ruiter@surfsara.nl

www.surfsara.nl