System and Network Engineering

MSc System and Network Engineering
Research Project 2

Towards predicting network device failures: *An analysis of time-series and syslog data*

Chris Kuipers
Chris.Kuipers@os3.nl

Henk van Doorn
Henk.vandoorn@os3.nl

August 15, 2018

Supervisors:
P. Boers
M. Kaat

Assessor:
Prof. Dr. C.T.A.M. de Laat

Abstract

Predicting failures in networking has been a subject of interest for over 30 years. Time-series and syslog data are commonly used to monitor the state of IP networks. In this work we set out to determine to what extent network device metrics and syslog data can be used to predict network failures. We do this by analyzing a number of known outages in the production network of SURFnet, a large Dutch NREN. Through our experiments, we proved a relation between the dataset containing the syslog device messages and the dataset containing the time-series network metrics. By combining network device metrics with syslog data, our analysis shows that both datasets can complement each other. Yet, we also show disparities between the two. We give potential ways to handle these disparities. This work contributes to the understanding of the relationship between data sources commonly found in network environments.

Contents

1 Introduction

Predicting network errors and failures has been a subject of interest for over 30 years[1]. In most cases, networks are monitored with tools that visualize network performance and show metrics representing the current state of the network. Two common sources of these data are syslog messages[2] and Simple Network Management Protocol (SNMP)[3] metrics. This way of monitoring can be classified as *reactive*, meaning that network operators react on events once they have happened.

With the storage capacity and processing power increasing over time, we are now at a point where storage capacity is abundant and processing of large datasets is possible. This allows manual analysis of errors and events both real-time and back in time. The next step would be to identify anomalous behavior based on the data at hand and predict anomalies based on patterns in these data. Therefore, transitioning to a more proactive monitoring strategy.

SURFnet has been monitoring their network in a variety of ways since 1993 including syslog[2] and metric data. From 2013 and onwards, SURFnet has been storing this data in searchable database clusters. The network metric dataset ranges from device temperatures to link utilization. These data have proved to be invaluable to identify and characterize network failures in the past.

As SURFnet is storing a significant amount of network device data[4], the question arose whether these data could be used to predict failure. The data we are presented with are unstructured time-series data, stored in different data-stores and are currently used for monitoring, debugging and root cause analysis purposes. In addition to these datasets, we were given access to SURFnet's ticketing system.

To set a scope for our work, we analyzed the last five high impact outages on the SURFnet network, covering the period between 2015 and the first half of 2018. These outages, five in total, resulted in multiple customers losing their connectivity. Three of the outages were line-card related issues. During interviews with the SURFnet network operators, we obtained another set of distinct cases where one or more customers lost resiliency or connectivity for a maximum period of 15 minutes at a time.

After getting access to the ticketing data, we followed the methodology of Turner et al., using a combination of keyword searches and manual evaluation of the tickets[5]. The subset of tickets contained (1) a handful of outages attributed to failing power feeds, (2) a significant number of tickets related to issues with the fiber-optic cables (e.g. fiber-cuts and maintenance) and (3) seven cases attributed to spontaneous line-card reboots. These seven cases are in addition to the three high impact outages caused by malfunctioning line-cards we found earlier.

Based on this, we have chosen to focus on outages related to failing network devices. This decision was twofolded: (1) SURFnet has to rely on a third party that manages the physical fibers and often maintenance on the fiber is announced in advance. (2) The datasets describe the previous state of the network.

As a step toward mitigating these events in the future, this work sets out to determine to what extent it is possible to predict intermittent hardware failures based on network event data from multiple data sources collected by SURFnet. We will describe the current possibilities, limitations and usability of the datasets.

1.1 Research Question

We pose the following research question:

To what extent is it possible to predict intermittent network device failures based on network metrics from multiple data sources?

To answer this question, we pose the following sub-questions:

- What metrics are relevant to identify a failure?
- What pattern can be identified between the intermittent failures?
- Is it possible to prove correlation between multiple data sources?

2 Background

SURFnet, the Dutch National Research and Education Network (NREN) is a non-profit organization offering internet services to academic and educational institutes. Their network consists of 2 core routers and 400 switches[4]. Out-of-scope are the 800 other devices within the network.

The overview we are presenting here is concise and describes only the basic components our research is based on. The actual infrastructure and monitoring system of SURFnet is more complex and extensive than we have described here.

2.1 Devices

2.1.1 Core routers

Currently, SURFnet has two Juniper T4000 series routers which are the core routers within their network. They are located on two separate geographic locations for redundancy purposes. Their names are *jnr01.asd001a.surf.net* and *jnr01.asd002a.surf.net* respectively. We will be refering to them as:

- *Router-A*
- *Router-B*

These core routers play a pivotal role within the SURFnet network and are interconnected via two fiber-optic cables.

2.1.2 Ciena 5410 switches

Core components within the SURFnet network are the Ciena 5410 model switches. At this moment, SURFnet operates ~40 of these devices. The switches are equipped with either 1 or 10 gigabit fiberoptic interfaces which connect to customers. According to the network operators these switches can be polled with SNMP with a maximum frequency of five minutes. Polling with a higher frequency would cause the switch to drop the current SNMP polling process resulting in data gaps. The SNMP polling method will be further described in Section 2.2.1.

2.2 Data Sources

2.2.1 Network Device Metrics

Device and link-state information are stored as counter values. The device metrics are polled once every five minutes using SNMP[3], resulting in around 200.000 data points for every round of polling[4] These values are stored in InfluxDB[6], a time-series database. Grafana, a time-series visualization tool, retrieves the metrics from InfluxDB and plots them over time. Using these graphs, network operators can monitor the network. The architecture to gather the device metrics is outlined in Figure 1.

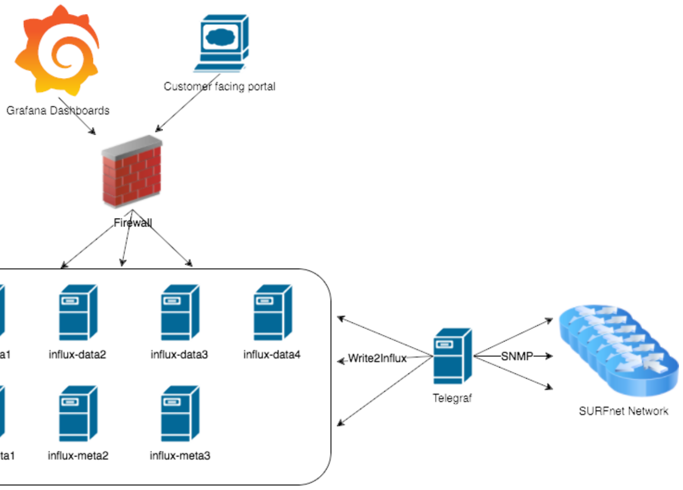

Figure 1: SNMP polling of network devices within the SURFnet architecture[4]

The SNMP polling as depicted in Figure 1 is implemented as an SNMP walk that retrieves a subtree of management values using SNMP GETNEXT requests[7]. This results in all available SNMP management information base **(mib)** values of a device. Telegraf timestamps all received metrics after which they will be stored in an InfluxDB cluster.

2.2.2 Syslog

When a device in the network generates a log message, the message is forwarded to Splunk[8] in the syslog[2] format, a standardized message format for system logs. Splunk is used as a central logging location, which offers a range of tools for querying, analyzing and visualizing the collected messages. Currently, the verbosity of the syslog messages is set to *informational.* The possible syslog logging is set by a numerical number which corresponds to a severity level which are depicted in Table 1.

level	**Severity**
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

Table 1: Syslog severity levels according to RFC 5424 - The Syslog Protocol[2]

Setting the syslog logging level to six (*informational*) results in all message starting from zero (*emergency*) up to and including level six (*informational*). Therefore, this setting does not generate debug-level messages which could contain in-depth technical information. The debugging-level is used in specific cases, for instance, to solve issues.

2.2.3 Ticketing system

As a third source, we used the information inside SURFnet's ticketing system. This system, called Jira[9], contains information about issues and incidents regarding the network. Here network operators log tickets and keep track of actions during and after incidents and problems.

Jira offers an overview of network related issues and measures taken to mitigate or solve these issues. The tickets are semi-structured and are used as a cross reference with the other two mentioned data sources. Listing 1 shows a typical ticket in Jira.

```
  Subject:  Spontaneous reboot LM5 Gv008a\_5410\_01
  Status:   Resolved
  Type:    Incident
  Origin:   Monitoring
  Impact:   P3
Description:
|5/1|CR link to Gv001a\_5410\_01 port|Up|0h34m57s|10G/FD|
|5/2|CR link to Gv002a_3930_01 port|Up|0h34m57s|10G/FD|
|5/3|CR link to Gv001a_5142_03 port|Up|0h34m54s|10G/FD|
|5/4|AVAILABLE|Down|0h0m0s|N/A |
```

Listing 1: Example of a spontaneous reboot trouble ticket (translated from Dutch)(Ticketing System)

3 Related Work

In 2010, Turner et al. created a timeline of network failures within the CENIC network [5]. Their work strongly relates to our work. CENIC, the Corporation for Education Network Initiatives in California and SURFnet both aim for similar goals; Offering internet services to educational institutes and research facilities[10, 11]. Although both organizations appear to be comparable, their networks differ. Turner et al. analyzed and classified network failures using router configuration snapshots, syslog data and administrator email logs. They encountered challenges of combining multiple data sources[5]:

- Inconsistencies in the data sources;
- Syslog data are ambiguous and prone to have omissions;
- Laborious to extract details from unstructured data.

The authors faces similar challenges and their work assisted us in interpreting our dataset. Their work assisted in the validation of events and analyzing the content of the datasets.

In their later work, Turner et al. used their method to analyze syslog messages[12]. They compared the network topology deduced from syslog messages to IS-IS update messages. Using the IS-IS update messages as *ground truth*, they found significant disparities between the IS-IS data and the changes computed based on the syslog information. The authors attribute these differences to the syslog data lacking precision. They conclude by stating that if one accounts for the challenges syslog data brings, syslog is still to be considered as a valuable source. This research contributed to our understanding of the limitations and possibilities of syslog.

Roughan et al. tried to detect IP forwarding anomalies using router SNMP data and BGP dynamic updates[13]. They highlight that SNMP data by itself lacks causality information and the quality of the data is uncertain. However, by combining these data with sources of comparable quality, the data still can leverage interesting viewing points as they give additional insights in anomalies. We observe similar findings in the work of other-sNadeem et al., Turner et al., Zhang et al., Turner et al., Markopoulou et al. This paper gave further insight in the behavior of SNMP when a failure occurs an a network device. They especially highlight the change of anomalies and false positives when using SNMP to measure performance.

4 Methodology

Following the methodology of Turner et al.[5], we identified and classified failure events based on the -presumed- cause of the failure. These data are retrieved both from the ticketing system by using keyword searches and manual interpretation of the data, and through interviews with the network operators. These are then used as process input, detailed below.

4.1 Process

The analysis is based on a iterative process of identification, analysis and validation. This process is depicted in Figure 2.

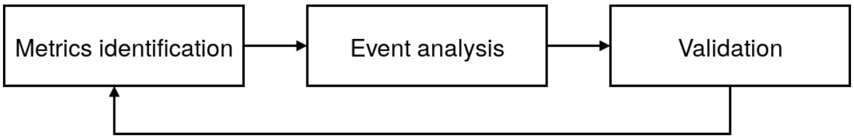

Figure 2: Event analysis process

4.1.1 Metrics identification

Using an initial opportunistic approach, we analyze on a per-event basis the device syslog messages and identify metrics that are related to the event. This is followed by an event analysis which was validated by interviewing the network operators. For example, we collected the metrics of all network interfaces of a line-card which has failed. The metrics and syslog messages that are analyzed on a per failure basis, are outlined in Section 4.3.

4.1.2 Event analysis

Using the collected data, we dove into the details of the failure event. With the use of Grafana, which in turn queries InfluxDB to visualize metrics, we analyzed the previously selected metrics. A timeframe was manually set for each event where we accounted for the impact and type of failure and compared it with a timeframe with known regular behavior. Therefore, our approach in setting a timeframe is dynamic and dependent on anomalies. Likewise, each syslog message corresponding to the aforementioned timeframe is analyzed.

Splunk was used to analyze the syslog messages both qualitatively and quantitatively. Using qualitative analysis, we tried to understand the cause and the meaning of the syslog messages, including the relationship between messages. By analyzing the syslog messages quantitatively, we were able to visualize the frequency of the messages over time. The technical documentation from the device manufacturers is used as input during the qualitative analysis of syslog messages.

4.1.3 Validation

We validated our results from the event analysis. Where possible, we tried to simulate the event by reproducing a similar setting using a test bed of the same model of network devices. Furthermore, root cause analysis documents have been used to verify our findings. At last, the findings have been validated by discussing them with the Team Expert Network-management (SURFnet TEN). They have in-depth knowledge about the SURFnet network and events which have happened.

4.2 Classification

Based on the information in SURFnet's ticketing system, we categorized the known past outages. The line-card related events are grouped according to the presumed cause of the outage as described in each trouble ticket. We classified the outages into two categories; spontaneous line-card reboots or permanent line-card failures.

During the course of this research we found frequent recurring drops in total switch throughput. This could be impacting connectivity and is included in the classification of events.

- Spontaneous line-card reboots;
- Permanent line-card failures;
- Unusual loss of throughput.

4.2.1 Spontaneous line-card reboots

Spontaneous line-card reboots are faults that happen intermittently and without a clear root cause. When the line-card fails and reboots, links connected to this line-card experience loss of connectivity for that period. After a short moment, connectivity is restored again. Listing 1 in Section 2.2.3 shows this ticket in the ticketing system.

In all seven of the reported events in the ticketing system, a Ciena 5410 switch was affected. According to the information available, no other switch models have had line-card reboots. As this messages is specific to this switch and is likely to impact connectivity, we will attempt to determine the cause, find omens and analyze if and how this failure can be predicted.

4.2.2 Permanent line-card failures

The permanent line-card malfunctions occurred on the core routers. These are faults where the line-cards drop network traffic and the connected links loose connectivity. This should not be an issue as a link-failure on the core routers should trigger a fail-over mechanism that reroutes the network traffic.

However, in this scenario traffic was being malformed and dropped without the fail-over mechanisms rerouting the traffic which is why this is a high impact event. The connectivity was only restored when the line-card was remotely shutdown or physically removed, triggering the fail-over mechanisms.

4.2.3 Unusual loss of throughput

Events related to unusual loss of throughput are not, directly or indirectly, mentioned in the Jira ticketing system. These type of events have been pinpointed by identifying and analyzing a hand-full of syslog events that were repeated many times. Looking at the network performance as depicted in Section 5.3 Figure 14, we can see a significant loss of throughput for a short period. These events differ from the previous as they do not indicate line-card issues but do effect connectivity.

4.3 Identification

4.3.1 Spontaneous line-card reboots

Using the Jira ticketing system and our classification model, we created a list with a number of these issues, containing the time and date of the event, device name and the impacted line-card. This resulted in a list of seven distinct reported cases in the past two years, as shown in Table 2.

Interestingly, in all of the reported cases the Ciena 5410 switches were affected. In a handful of other cases devices would suddenly reboot, however, further analysis showed that in these cases the power feed was the cause of the issue.

In two of the seven cases, the same line-card was affected only two weeks apart. In two other cases, the same chassis was affected twice in an eight month period, however, two individual line-cards rebooted. Spontaneous reboots trigger syslog messages as shown in Listing 2.

Date	Time	Hostname	Line-card
22-02-2018	09:59	Gv008a_5410_01	LM5
06-01-2018	13:55	Asd002A_5410_01	LM6
07-08-2017	19:07	Dt001B_5410_01	LM4
24-07-2017	01:58	Dt001B_5410_01	LM4
14-11-2016	11:19	Gn001A_5410_01	LM1
18-08-2016	21:14	Asd001A_5410_01	LM8
12-12-2015	13:37	Asd001A_5410_01	LM7

Table 2: Seven reported cases of spontaneous rebooting line-cards in Ciena 5410 switches (ticketing system)

```
Feb 22 09:59:39 active.5410-01.gv008a.dcn.surf.net [Local] 145.145.93.57
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 MODULE-6-THERMAL_STATE_CHANGE:
    ↪ module(1-A-LM5): :Thermal state changed from NORMAL to unknown
Feb 22 10:01:02 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 MODULE-6-OPER_STATE_UPDATE: module
    ↪ (1-A-LM5): :Module operational state changed to Init
```

Listing 2: Rebooting line-card in Ciena 5410 (syslog)

4.3.2 Permanent line-card failures

In the Jira tickets, network operators mentioned that syslog messages were generated prior to the failing of the line-cards which are listed in Listing 3.

```
 Sep 11 12:05:04 re1.JNR01.Asd001A.dcn.surf.net 1 2017-09-11T10:05:04.139
     ↪ Z re1-Router-A4000 - - - - fpc6 XMCHIP(1): FI: Packet CRC error -
     ↪ Stream 6, Count 1
```

Listing 3: Messages identifying line-card failure (syslog)

According to the vendor[18], this message indicates that a packet with an incorrect Cyclic Redundancy Check (CRC) value is to be sent through the switch fabric. Cyclic Redundancy Checks (CRCs) are used as an error-detecting code to detect accidental changes in transit.

The syslog *Packet CRC Error* messages and the *Interface Input Error* metrics stored in InfluxDB are correlated between the two devices. If *Router-A* generates *Packet CRC Error* syslog messages, it will result in *Router-B* receiving packets with an incorrect CRC. This in turn will increase the *Interface Input Error* counter. Figure 7 in Section 5.2 depicts the occurrence of these errors.

4.3.3 Unusual loss of throughput

During the course of this research we found significant and infrequent drops in the *total throughput* for multiple Ciena 5410 switches. The *total throughput* is a derived metric, consisting of the sum of the inbound traffic for all interfaces, plotted against the sum of the outbound traffic for all interfaces over time. This can either be plotted on a per line-card basis, or for the whole switch chassis. The time stamp of the anomalies in the metrics relate to the time stamp of the syslog message listed in Listing 4.

```
2018 May 24 09:50:33 active.5410-01.Asd001A.dcn.surf.net DATAPLANE-4-
    ↪ FLOOD_CONTAINMENT_THRESHOLD: chassis(1): :Flood Containment
    ↪ Threshold Event Container LIMIT_2 on l2-ucast EXCEEDED
```

Listing 4: Flood threshold exceeded messages (syslog)

For these type of events, we have used the *throughput per interface* metrics and syslog events.

5 Findings

5.1 Spontaneous reboots

The data we have identified in Section 4.3.1 are used to analyze the cause of the spontaneous reboots.

In each of the documented cases, we observed that just before the line-card is crashing, two Connectivity Fault Management (CFM) control messages are logged. Connectivity Fault Management is specified in the IEEE 802.1ag standard[19]. The protocol includes a Continuity Check Protocol that provides a means to detect connectivity failures in an Ethernet service[20].

Failing CFM Tunnel

The first control message is displayed in Listing 5 and is immediately followed by the second message, displayed in Listing 6. It indicates that a RDI (Remote Defect Indicator) has changed for a CFM.

```
Feb 22 09:59:20 active.5410-01.gv008a.dcn.surf.net [Local] 145.145.93.57
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 CFM-1-CFM_SERVICE_FAULT_SET:
    ↪ chassis(1): :cfm service fault state changed to RDI for service PBT
    ↪ -000002252-3474 index 4
```

Listing 5: Ciena 5410 RDI Fault state changed (syslog)

The message in Listing 5 indicates that one or more remote MEP (Maintenance association End Point) have detected service faults[21]. The normal process of CFM RDI messages is shown in Figure 3.

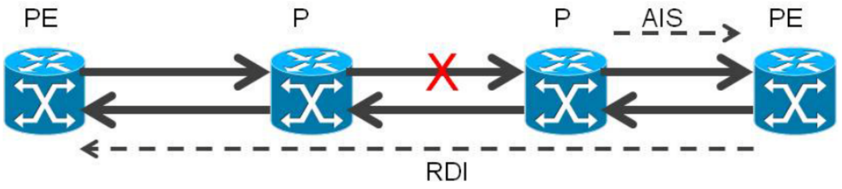

Figure 3: CFM process - detecting a remote defect (Ciena[21])

```
Feb 22 09:59:20 active.5410-01.gv008a.dcn.surf.net [Local] 145.145.93.57
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 CFM-3-CFM_R_MEP_FAULT_SET: chassis
    ↪ (1): :cfm fault status set to RDI for remote mep id 5, service PBT
    ↪ -000002252-3474
```

Listing 6: Ciena 5410 RDI Fault set (syslog)

The message in Listing 6 indicates that a Cross Connect Defect (CCM) heartbeat has not been received from a remote MEP for a certain period[21].

Line-card initialization

Shortly after these two messages, the line-card triggers a thermal warning as shown in Listing 2. This warning indicates that the thermal state of the line-card switched to *unknown*. A few seconds later, the switch logs that it tries to (re)initialize the card and the interfaces, as is shown in Listing 7. After this process completes, the line-card returns to its normal operational state.

```
Feb 22 10:01:02 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 MODULE-6-OPER_STATE_UPDATE: module
    ↪ (1-A-LM5): :Module operational state changed to Init
Feb 22 10:02:04 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 MODULE-6-OPER_STATE_UPDATE: module
    ↪ (1-A-LM5): :Module operational state changed to Enabled
Feb 22 10:02:06 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 XCVR-6-OPTIC_INSERTED: xcvr(port
    ↪ 5/1): :Pluggable transceiver inserted
Feb 22 10:02:06 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 PORT-4-STATE_CHANGE: port(port
    ↪ 5/1): :Port Link up
```

Listing 7: Initializing Line-card (syslog)

The thermal state warning is triggered because the switch is not able to probe the line-card for a thermal reading during the period between the line-card failure and the initialization process. Once the line-card becomes operational, it reports the correct thermal reading. Listing 8 displays these thermal messages.

```
Feb 22 10:01:03 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 MODULE-6-THERMAL_STATE_CHANGE:
    ↪ module(1-A-LM5): :Thermal state changed from unknown to COOL
Feb 22 10:02:26 remote.5410-01.Gv008A.dcn.surf.net [Local] 145.145.91.8
    ↪ 00:03:18:9a:1c:5f Gv008A_5410_01 MODULE-6-THERMAL_STATE_CHANGE:
    ↪ module(1-A-LM5): :Thermal state changed from COOL to NORMAL
```

Listing 8: Thermal states during the initalization of the Line-cards in the Ciena 5410 switches (syslog)

Given these insights and the order at which these messages arrive, it appears that a failure at a remote MEP would cause these CFM messages to be triggered.

Yet, the supposedly failing switch itself gives no indication that a failure has occurred. The interconnected switches do not provide any indication that a component or service has failed and no messages prior to the failure were generated.

Messages on connected switch

When a line-card on the Ciena 5410 crashes, the connected switches start logging that the interfaces and tunnels towards the affected Ciena 5410 switch are going down. These messages are displayed in Listing 9.

```
Feb 22 09:59:20 remote.5142-03.gv001a.dcn.surf.net [local] 145.145.91.112
    ↪  2c:39:c1:e8:70:20 Gv001A_5142_03 VCTUNNELING-1-TUNNEL_DOWN:
    ↪ chassis(1): :Tunnel 1B4-005UL0 down
Feb 22 09:59:20 remote.5142-03.gv001a.dcn.surf.net [local] 145.145.91.112
    ↪  2c:39:c1:e8:70:20 Gv001A_5142_03 VCTUNNELING-1-
    ↪ TUNNEL_ACTIVE_STATE_CHANGED: chassis(1): :Tunnel 1B4-005UL0 active
    ↪ state changed to inactive, b-vid 3472
Feb 22 09:59:20 remote.5142-03.gv001a.dcn.surf.net [local] 145.145.91.112
    ↪  2c:39:c1:e8:70:20 Gv001A_5142_03 PORT-4-STATE_CHANGE: port(port
    ↪ AGG1): :Port Link down
```

Listing 9: Messages on switch connected to a failing Ciena 5410 line-card (syslog)

After the line-card resumes normal operation, the connected switches log that the interfaces and tunnels are up again.

Network metrics

In all the cases we came across, we identified this exact same pattern between the CFM RDI error messages, thermal state warnings and (re)initialization of the card.

In contrast to this clear pattern of anomalies in the syslog messages, we were not able to find a similar pattern in the network metrics of the switch. We would have expected to identify a loss of overall throughput during the period that the line-card was rebooting. These expectations proved to be wrong. The throughput is plotted in Figure 4. These values are non-negative integer counter values and are plotted as non-negative derivatives.

We were unable to pinpoint why no traffic was dropped during the reboot as no other unusual syslog messages were generated and metric data showed usual behavior as can be seen in Figure 4.

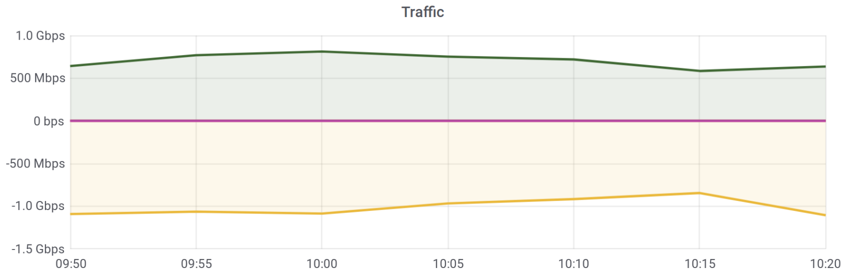

Figure 4: Switch Gv008a_5410_01 Line Module 5 - Total throughput (SNMP)

The total time between failure of the line-card and returning back to normal operation is between two and three minutes.

The class of line-card failures we have analyzed, only occurred on one specific model of switches; only the Ciena 5410 switches are affected. No other switches in the SURFnet network follow this specific set of events.

Using the pattern we have identified, we were able to find 73 cases in the last year. Figure 5 quantifies these events over time. Interestingly, most of these cases appear to go unnoticed, since the tickets in Jira only mention 7 cases. The metrics indicate that the throughput is unaffected during a reboot of the line-card, as depicted in Figure 4.

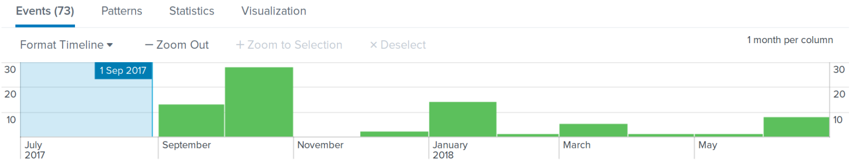

Figure 5: Number of cases of spontaneous line-card reboots plotted over time (syslog)

5.2 Line-card errors on core routers

Data overview

The *Packet CRC Error* messages are observed in the syslog dataset and form the startingpoint of the analysis. These messages only occur in combination with the failure of the line-card.

The *Interface Input Errors* mentioned in this chapter are device metrics stored in InfluxDB. We will show the relation between the messages generated by one core router to the metrics gathered from the other core router.

5.2.1 Failure at *Router-A* on 14th of September 2017

The loss of resilience was noticed on the 14th of September at 09:30 and identified as a faulty line-card at 10:00. The interfaces attached to the line-card did drop traffic but did not trigger failover features, interfaces on the line-card were manually shutdown triggering fail-over. Notable entries in the ticketing system are mentioned in Table 3.

Date	Time	Event
14-09-2017	09:30	Resilience loss noticed
14-09-2017	10:00	Problem identified
14-09-2017	10:30	Interfaces manually brought down
14-09-2017	17:30	Line-card reset

Table 3: Notable events mentioned in the ticketing system

Syslog messages

The entries in Table 3 are compared to the occurrences in syslog. We plotted the number of *Packet CRC error* syslog messages over a timeline to visualize the occurrences.

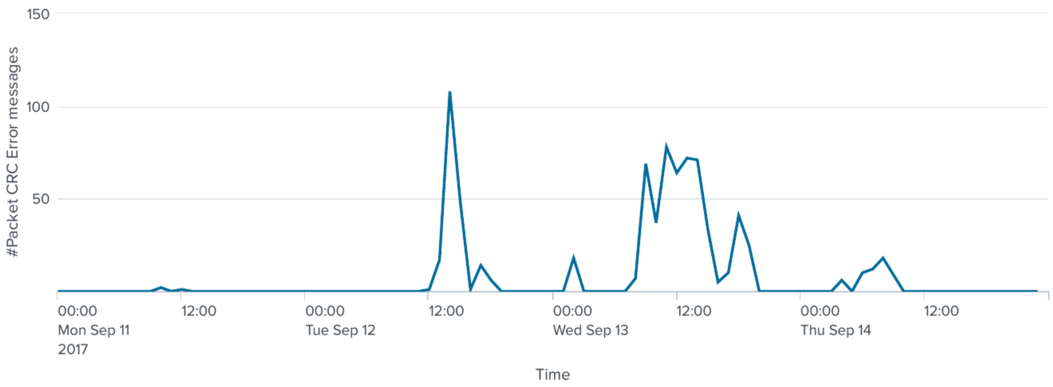

Figure 6: Number of Packet CRC Error messages (SNMP)

As can be seen in Figure 6, the first *Packet CRC Error*, which occurred on September 11th, went unnoticed. Surprisingly, most syslog messages occurred on 13th of September while connectivity remained unaffected according to the ticketing system. This can also be observed in the throughput metrics where no apparent drops can be seen. Next, the metrics stored in influxDB are analyzed and are depicted in Figure 7 and Figure 8.

Device metrics

Because packets are arriving at the switch fabric of *Router-A* with incorrect CRC, we expect to see input errors on *Router-B*, as this router is directly connected. The interface input errors are stored as a counter value in InfluxDB, the counter value represents the number of received packets on that interface with an incorrect CRC. We plotted this information using Grafana and is depicted in Figure 7.

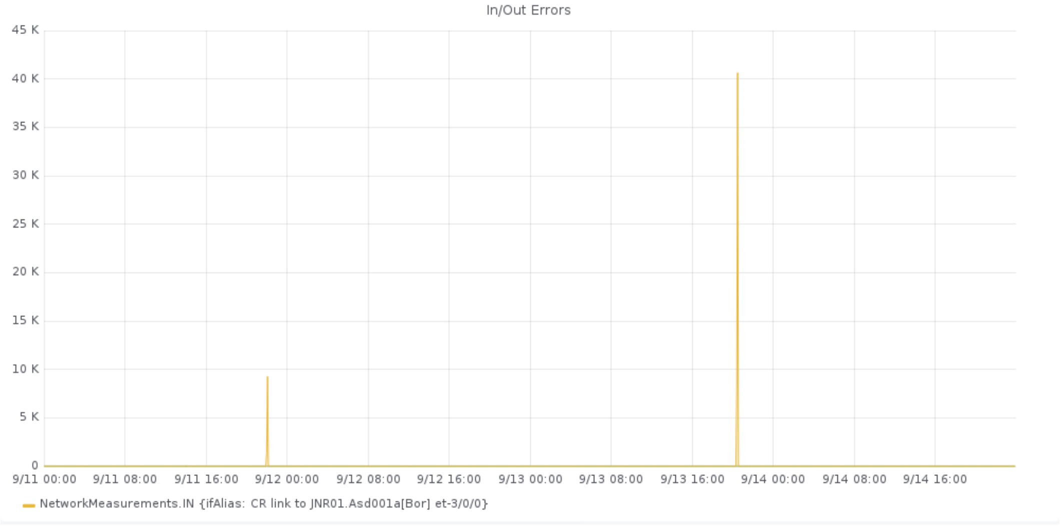

Figure 7: Interface input errors on *Router-B* (SNMP)

As can be seen in Figure 7, two major spikes of input interface errors occurred during this timeframe on *Router-B* interface which is connected to *Router-A*. No input errors occurred on all other active interfaces within this timeframe. This also shows that although *Packet CRC Error* messages were generated, they do not increase the *Interface Input Error* counter on *Router-A*. Neither the syslog message or the device metrics indicate why *Packet CRC Error* not necessarily result in *Interface Input* errors. It seems likely that the router drops the packet internally although we were unable to find syslog messages or device metrics which relates to this suggestion.

First, we will zoom into the first spike which is depicted in Figure 8.

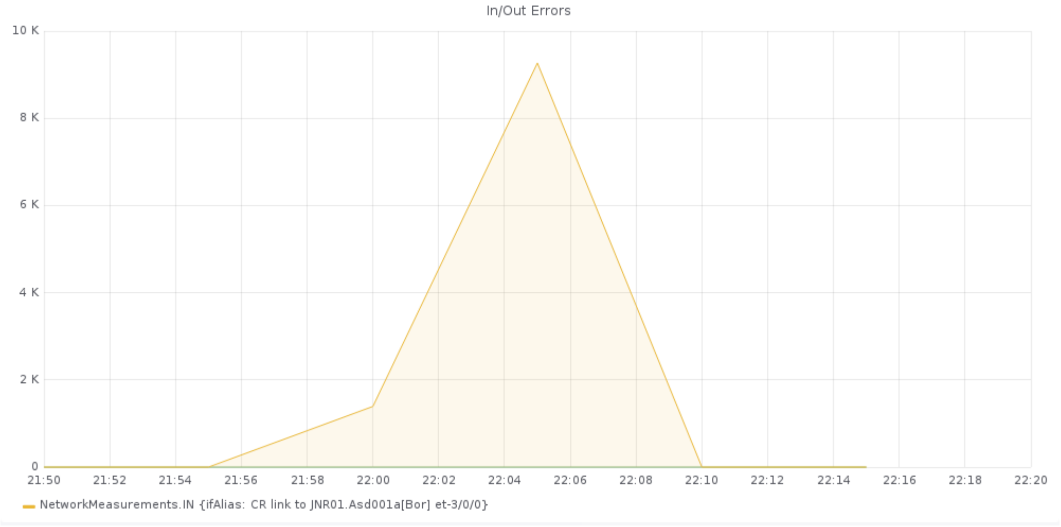

Figure 8: Graph of first spike in Interface input errors on *Router-B* on 12th of September (SNMP)

In Figure 8, each data point is clearly visible. Over a period of thirty minutes, there are 6 data points. This is due to the SNMP polling interval as described in the SURFnet monitoring architecture[4] and in Section 2.2.1.

Due to this polling resolution, we are unable to plot the metrics in a more granular manner.

Data comparison

As mentioned in Section 4.3.2 the syslog messages depicted in Figure 6 on September 11th 2017 warns that messages with an incorrect CRC will traverse the switch fabric. Therefore, it could indicate that connected network devices will receive packets with an incorrect CRC as this packet is likely to be transmitted through another port. The specific events are specified in Table 4.

Date	**Time**	**Event**
11-09-2017	10:13:25	Packet CRC Error
11-09-2017	10:13:27	Packet CRC Error
11-09-2017	12:05:00	Packet CRC Error

Table 4: Packet CRC messages with timestamp (ticketing system

The syslog and metrics show the events occurring within the same timeframe. The significant rise in *Interface Input Errors* at *Router-B* indicate that at *Router-A* a line-card is imminent to fail.

The second spike of interface input errors which occurred on September 13th is depicted in Figure 9.

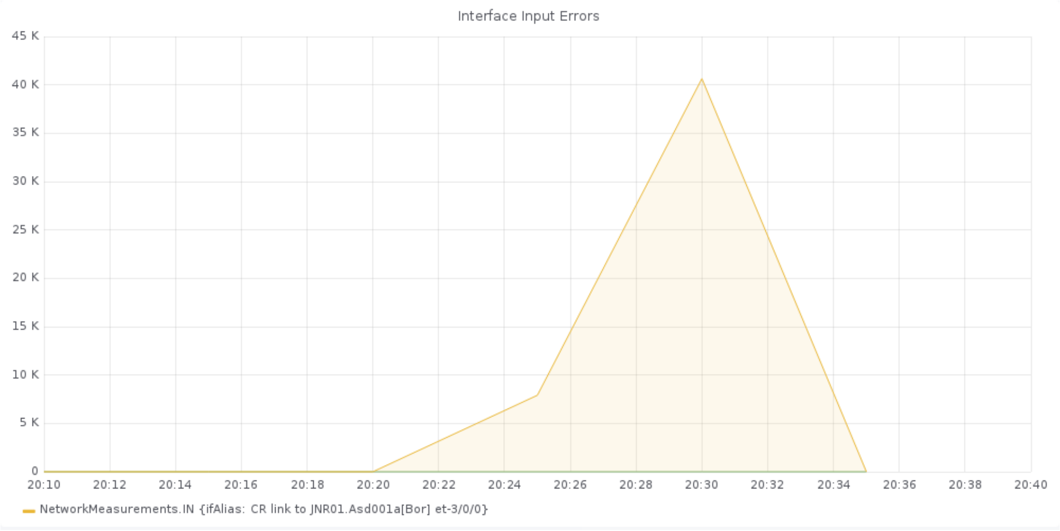

Figure 9: Graph of second spike of Interface Input errors on *Router-B* on 13th of September (SNMP)

Figure 9 again shows the limited data resolution of one data point per five minutes within a timeframe of thirty minutes similar to Figure 8. Grafana plots the values using just two non-zero data-points stored in InfluxDB.

The number of incorrect CRC messages over time, before the input interface errors start to happen, is depicted in Figure 10.

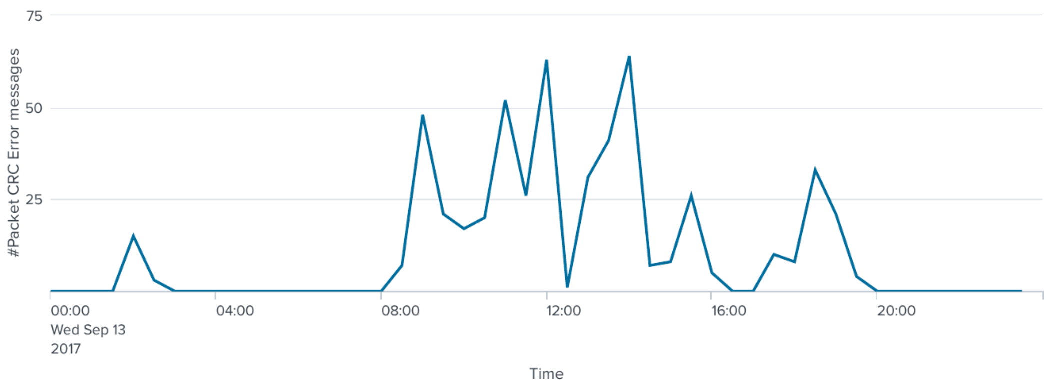

Figure 10: Syslog messages prior to interface input errors (Syslog)

As can be seen in Figure 10 *Router-A* generates syslog messages prior to the interface input errors seen in Figure 9. As with the first spike of interface input errors, the Syslog messages do not occur during the receival of the faulty packets. Based on this event it seems the *Packet CRC Error* generated by *Router-A* are an omen to line-card failure. An increase of *Packet CRC Error* messages seems to indicate that connected routers will receive packets with an incorrect CRC. Since another event of line-card failure occurred, we will compare this in Section 5.2.2.

5.2.2 Failure at *Router-B* on 6th of March 2018

As with the results described in Section 5.2.1 we will take the Syslog message as mentioned in Listing 3. We have taken the number of *Packet CRC Error* syslog messages occurrences and plotted those in Figure 11, all of those messages were generated by *Router-B*. First, we will analyze the number of generated syslog messages and secondly, the *Interface input errors* metrics will be analyzed and compared to the syslog messages.

Notable events in the ticket systems are mentioned in Table 5. It seems that the first errors had no impact on connectivity since no issues were reported by connected customers.

Date	**Time**	**Event**
06-03-2018	08:15	First messages in Syslog
06-03-2018	17:30	Starts dropping traffic
06-03-2018	17:48	Network issues reported by customers
06-03-2018	18:45	Issues diagnosed, failing line-card identified
06-03-2018	19:00	Failing line-card shutdown manually initiating failover restoring connectivity

Table 5: Notable events mentioned in the ticketing system

Syslog messages

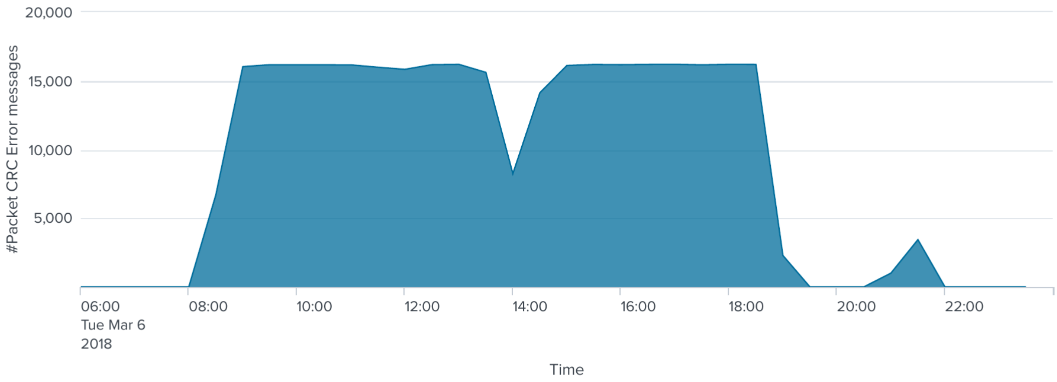

Figure 11: Number of Packet CRC Errors messages (Syslog)

Shown in Figure 11 are the number of *Packet CRC Error* syslog messages generated by *Router-B*. Connectivity loss was reported at 17:48. Therefore, the *Packet CRC Error* message could be seen as an omen to the failing of the line-card. According to the ticketing system the line-card was switched off at 19:00, which initiated the fail-over connection and restored connectivity. Shutting down the faulty line-card had an immediate impact on the number of *Packet CRC Error* syslog messages being generated which is depicted in Figure 12.

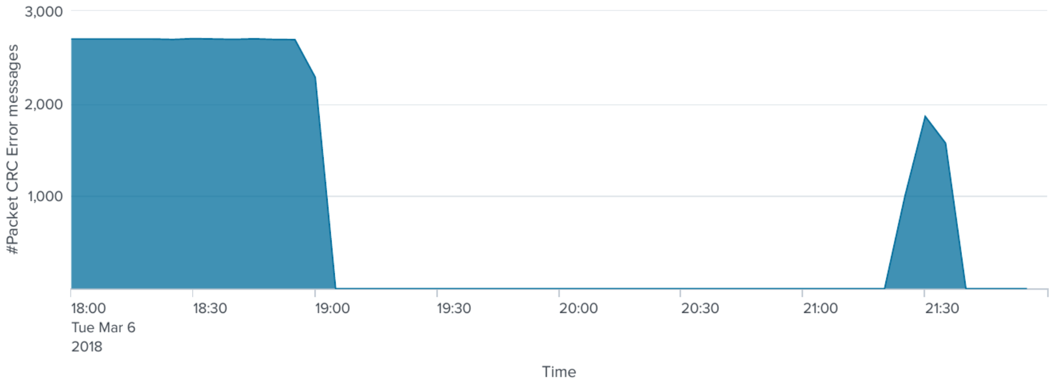

Figure 12: Impact on generated *Packet CRC Error* (Syslog)

Figure 12 shows the drop in generated messages after the faulty line-card was disabled. This corresponds with the events logged in the ticketing system listed in Table 5, a short recurrence can be seen at 21:30, we were unable to link those messages to an event given the available data because no mention was made in the ticketing system and no customers reported connectivity loss. Nonetheless, connectivity could have been impacted but not have been noticed.

Next the device metrics are plotted after which we attempt to relate those with syslog messages.

Device metrics

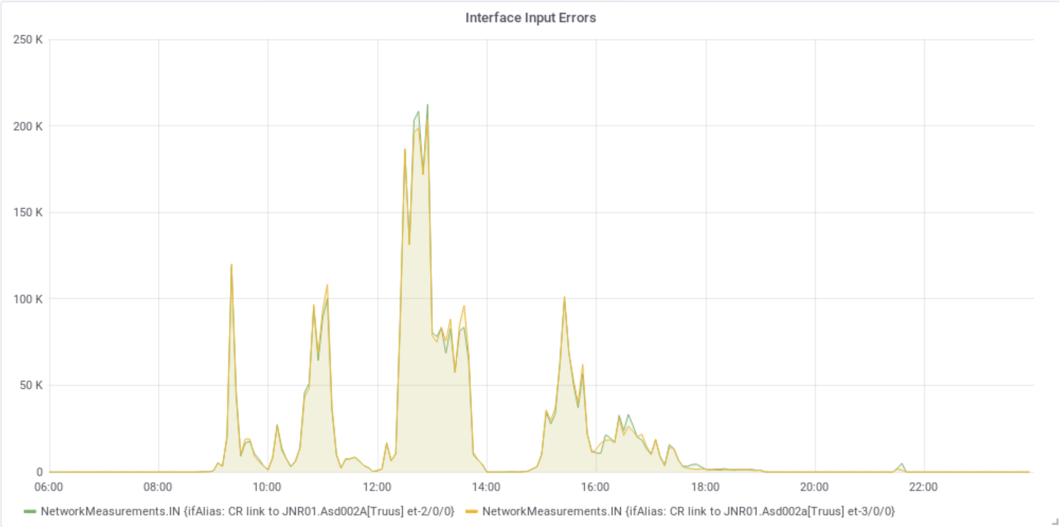

Figure 13: Interface errors occurring on Router B (SNMP)

As depicted in Figure 13 interface input errors on *Router-A* which is connected to *Router-B* reach up to 200.000 faulty packets received in five minutes. The number of errors is clearly abnormal since normally the number of errors remains well within double digits.

Data comparison

It is likely that there is a relation between the *interface input errors* and the *Packet CRC Error* syslog messages when a line-card is imminent to fail or has failed. The advent significant amount of syslog messages introduce an immense rise in *Interface Input Error* metrics which only occurred when a line-card failed or malfunctioned.

5.2.3 Analysis

Both line-card events show *Packet CRC Error* syslog messages before connectivity was impacted and notified. Furthermore, based on the two events we analyzed it seems likely that routers generating these syslog events will transmit packets with incorrect CRC as those packets travel over the switch fabric. The impact can be seen by analyzing the *Interface Input Error* metrics on other routers, linking the syslog messages and devices metrics. In the ticketing system, a timeline states when the line-cards were shutdown, reset or replaced. Upon these measures, the syslog events ceased and connectivity was restored.

5.3 Spontaneous throughput loss

During the research we found a peculiar event where all of a sudden the throughput of a switch significally drops, as depicted in Figure 14.

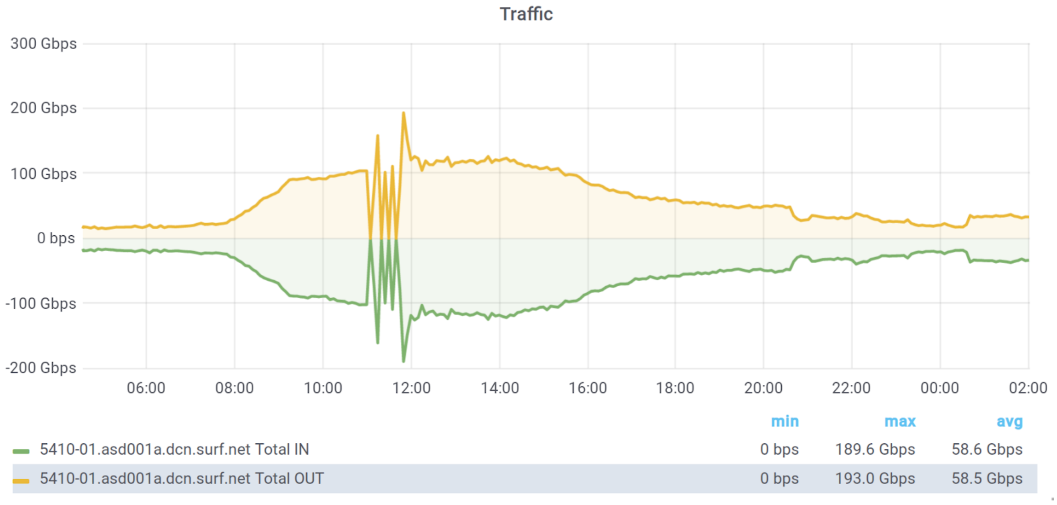

Figure 14: Throughput loss input and output (SNMP)

The metrics as shown in Figure 14, have been correlated in time to the syslog messages depicted in Listing 10. These syslog messages indicate that the switch exceeds a predefined flooding threshold. This warning is being triggered if the amount of frames being flooded exceeds a set threshold.

```
2018 May 24 09:50:33 active.5410-01.Asd001A.dcn.surf.net DATAPLANE-4-
    ↪ FLOOD_CONTAINMENT_THRESHOLD: chassis(1): Flood Containment
    ↪ Threshold Event Container LIMIT_2 on l2-ucast EXCEEDED
```

Listing 10: Flood threshold exceeded (syslog)

Next we plotted the occurrences over time starting from January 1st of 2018 up to and including the 30th of June 2018. Every bar in the graph represents one day, the total number of generated *Flood Containment* within this period is 82,394. The graph is depicted in Figure 15.

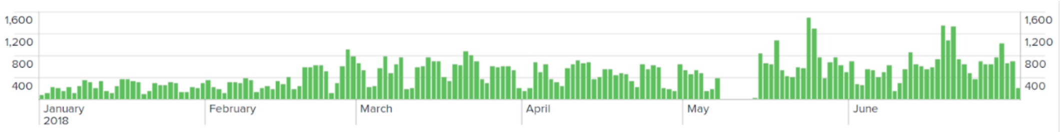

Figure 15: Occurrences of *Flood Containment Threshold* messages plotted over time (Syslog)

In this graph a trend can be seen as there are an increasing number of syslog messages indicating that the *Flood Containment Threshold* has been exceeded. Also, in the weekend the number of *Flood Containment* messages is consistently lower than during the working days. This trend is even more apparent during the May holidays. Thus, it seems likely that network activity is related to the number of *Flood Containment* events occurring.

5.3.1 Analysis

There are only a handful of scenarios where a switch will flood a frame to all interfaces. One case is if the switch receives unknown unicast frames. Another reason would be broadcast frames, but the metrics suggest that this was not the case, since there was no measured change in the broadcast traffic.

Looking back at the total throughput metrics in Figure 14, we see that the throughput is severely impacted.

After analyzing the documentation of the Ciena 5410 switches and discussing these findings with the SURFnet TEN team, we concluded that the internal flood channel of the switch, a dedicated backplane, is only limited to a maximum throughput of 500 Mbit/s. This flood channel is shared between the chassis and all line-cards.

An explanation could be that somehow all unicast packets are being flooded, and the internal flood channel is completely congested. For this to be true, the switch would need to flush its MAC-table (or parts of it).

After we identified numerous similar events, we noticed that a large portion of these events had one thing in common: just before the total throughput of the switch would decrease and the *flood contamination* messages would be logged into syslog, one or more interfaces on that switch would go down. Using the information about these events in the ticketing system, we found that in almost all cases that the interfaces went down, there was an explanation. Most of the times, the interface would go down because of work on the fiber-optic cable, or maintenance on the other side of the link.

Combining these findings, we hypothesize that the switch would loose its MAC-table each time when an interface would be shutdown. Using an experiment, we set out to test our hypothesis.

Therefore, our hypothesis for this experiment is:

A change in the state of any interface causes the switch to flush or drop its MAC-table, and therefore, results in the flood-channel to be congested, limiting the total throughput

5.3.2 Experiment

We suspected that changes to an interface influences the MAC-table. Therefore, we executed an experiment where we sent one million protocol data units (*PDU*) per second through a Ciena 5410 switch, while monitoring the number of received PDUs using a packet generator and a test bed provided by SURFnet. The size of each PDU is equal to 1,25 kilobytes, which simplifies throughput measuring. Therefore, one million packets per second is equal to ten gigabit per second traffic. The links between the switches and the hosts were all 10 Gbit/s Ethernet fiber-optics. We have measured the following:

1. Transmitted PDUs;
2. Received PDUs;
3. Return traffic.

The experimental setup is depicted in Figure 16. The transmitted PDUs (1) are the number of PDUs generated by the packet generator. The number of transmitted PDUs is measured at the egress port at Host A. Received PDUs (2) are the number of PDUs which have traversed through the switch and are received by Host B. Every 30 seconds, a PDU is returned to keep the switch's MAC-table populated or to relearn the MAC-address.

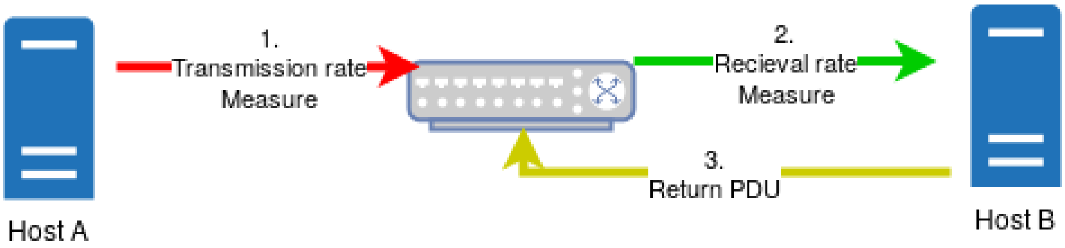

Figure 16: Depiction of experimental setup

First, we conducted a baseline-experiment, to verify our setup and compare our results to. This experiment is depicted in Figure 17. This baseline experiment shows that the number of received PDUs -red- is equal to the number transmitted.

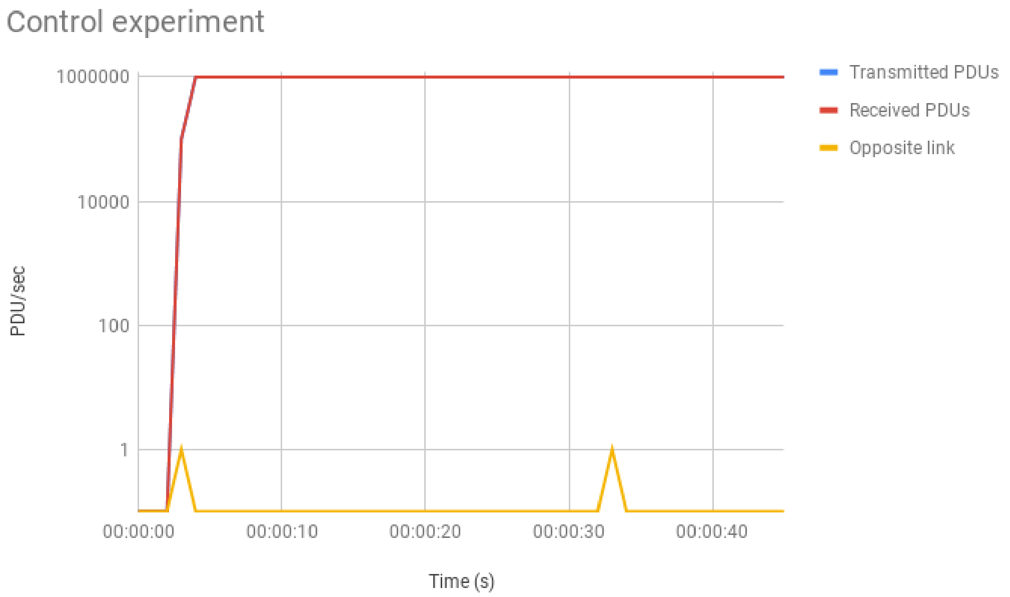

Figure 17: Control experiment

During the experiment we have triggered a couple of changes by shutting down an unrelated interface. For the duration of the experiment, the MAC-table and throughput was monitored. The experiment consisted of four stages.

1. Start transmission;
2. Introduce state change;
3. PDU in opposite direction of traffic;
4. Normal state restored.

The results of the experiment as described in Figure 18. The numbered arrows indicate the stages of the experiment.

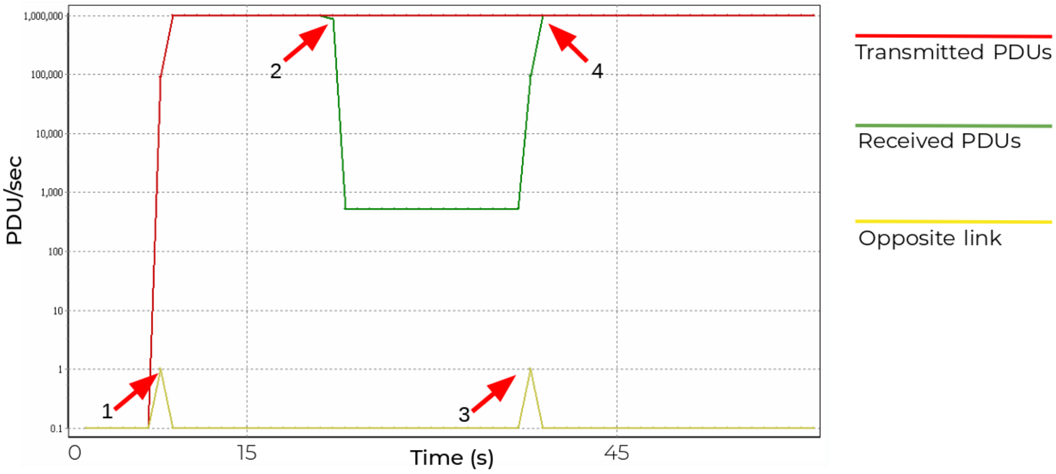

Figure 18: Throughput measurement of Protocol Data Units while shutting down a random interface

At the start of the experiment, indicated by arrow number 1, all PDUs transmitted successfully traversed the switch. When we introduce **any** change in the link-state of **any** interface, indicated by the arrow 2, the **entire** MAC-table was dropped. In this experiment, we shut the interface down on the remote side of the link. Listing 11 shows the MAC-table during the period between the points 2 and 3. Therefore, if a link state change was caused by a third party, the -global- MAC-address table would be dropped, impacting all traffic traversing the switch.

```
Asd001A_5410-01T*> mac-addr show vs pdb-os3
generating MAC table, please wait...
+----------------------------- MAC TABLE -----------------------------+
|          |         |D| Logical Interface | Oper |
| Virtual Switch |    MAC    |S| Type | Name     | LPort |
+-----------------+-----------------+-+--------+------------+-------+
|pdb-os3      |00-01-00-00-06-68|D|sub-port|pdb-os3_4/3 |4/3   |
|                                    |
| Entries: 1                              |
+---------------------------------------------------------------------+
```

Listing 11: State of the MAC-table during the period between points 2 and 3 of the experiment (Device CLI)

The induced state change caused the entire MAC-table to be dropped, thus, the flood-channel to be congested. Thereby, since this channel only has a maximal throughput of 500 Megabits per second, the total throughput is being limited.

During the period between points 2 and 3, the syslog displays the same messages as we have observed ourselves in Splunk. The messages are displayed in Listing 12.

```
June 15, 2018 12:03:20.090 [local] Sev:6 port(port 4/4): :Port Link down,
    ↪ operSpeed     , operDuplex     , operPause     , operAneg
June 15, 2018 12:03:21.950 [local] Sev:6 chassis(1): :Flood Containment
    ↪ Threshold Event Container LIMIT_2 on sub-port pdb-os3_4/3, l2-ucast
    ↪ EXCEEDED
June 15, 2018 12:03:37.925 [local] Sev:6 health(port 4/4): :Port Link:
    ↪ State has experienced a negative change in health Normal to Warning
June 15, 2018 12:03:45.046 [local] Sev:6 chassis(1): :Flood Containment
    ↪ Threshold Event Container LIMIT_2 on sub-port pdb-os3_4/3, l2-ucast
    ↪ normal
```

Listing 12: messages during the period between 2 and 3 of the experiment (Device CLI)

Upon relearning the MAC-address as indicate by point three, normal operation was restored as indicated by the fourth state as depicted in 18.

This experiment also shows that the switch does not give any indication that its entire MAC-table has been dropped. Even when accessing the console and setting the highest debugging level output, nothing indicates directly that the MAC-table has been dropped. Only the *flood containment threshold* hints at the fact that the MAC-table was dropped.

We repeated the experiment, where we removed a *service*, being a virtual switch within the switch. This virtual switch functions as an independent switch within the chassis. We wanted to validate if this also affects the switch in any way The results are shown in Figure 19.

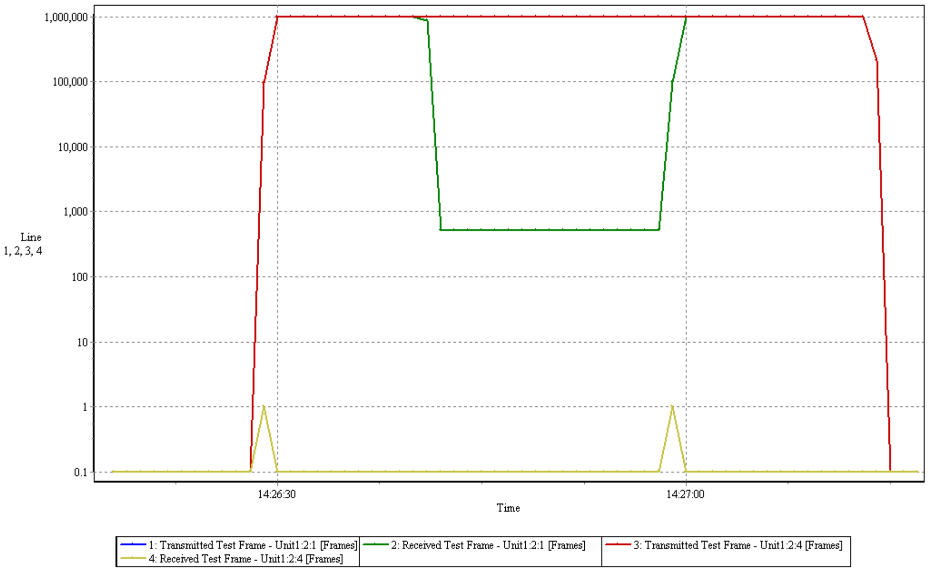

Figure 19: Throughput measurement of Protocol Data Units while removing a virtual switch

Immediately after removing a virtual switch the number of received PDUs dropped to exactly the bandwidth of the flood-channel. This is depicted by the green line in the chart. After (re)learning the MAC-address, the number of received PDUs is again equal to the number of transmitted PDUs. Thus, verifying that removing an virtual switches causes the switch to drop the -global- MAC-address table.

6 Discussion

Using the experiment as outlined in Section 5.3, we proved that a Ciena 5410 switch looses its global MAC-table each time any interface goes down. Until the switch relearns the destination MAC-address, the throughput is severely impacted.

The limitation of our experiment is that we have simulated an extreme case. Dropping the MAC-table will have little effect on interfaces with a continuous bidirectional data flow because the switch will populate the MAC-table near-instant. This results in limited or no dropped traffic. However, the results indicate that external factors, such as faulty customer equipment, fiber-cuts and flapping links can severely impact the throughput of the entire switch as any of those factors would cause the switch to drop its entire MAC-table.

In Section 5.1, we have analyzed syslog messages indicating that line-cards rebooted spontaneously. However, the network device metrics do not indicate that the throughput was impacted. This behavior is surprising as packets arriving at a rebooting line-card would likely to be dropped. The impact of a rebooting line-card is not reflected in the throughput metrics. The metrics would indicate that the reboots would go unnoticed, but customers did experience loss of connectivity. This disparity between the events observed by customers and the event as observed from the metrics is striking. We noticed that the resolution of the metrics data is lower than the time it takes to reboot a line-card. In other words, the line-card could reboot within the polling frequency of SNMP, in turn, masking the effect of the reboot. The work of others describe similar anomalies[5, 12, 14] and with SNMP in particular[13].

While our findings show that combining the syslog and network device metrics datasets do yield valuable insight in past failures, we were not able to identify metrics in the current datasets that could be used to predict spontaneous loss of throughput and spontaneous line-card reboots. In the case of permanent line-card failure, we were able to identify a likely indication of a impending line-card failure. However, given that the event only occurred two times, it is premature to conclude this definitively.

The current datasets are unsuitable as is for automated processing. The data format is different between each source, and the sources contain numerous inconsistencies and ambiguities. There were discrepancies between the time stamps in syslog messages. Nadeem et al. have also encountered these challenges, they worked around this issue by excluding these events [14]. This makes an automated effort in effect hard to do and manual intervention necessary. To certain extent, humans are capable of accounting for these errors more easily as they can interpret the data[1]. However, the sheer amount of data would be too laborious to be vetted for automated processing.

A large part of our analysis involved manual interpretation of the data. Partly, this is due to the fact that an in-depth understanding of the network is necessary. In addition to that, all datasets contained inconsistencies and ambiguities which we had to manually account for. The inconsistencies were related to incorrect time stamps, device names and counter-wrapping. The ambiguities were related to subtle differences in the meaning of syslog messages.

Our work contributes to the understanding of the relationship between data sources commonly found in network environments.

6.1 Future work

Now that we have identified the most challenging issues surrounding the datasets we have used, measures can be put in place to prevent or mitigate these challenges. In particular, inconsistencies related to improper time settings on the network devices could be fixed by using consistent time-zone settings and clock synchronization. Issues related to device names could be adjusted by adhering to one universal naming convention.

After implementing these propositions, we believe it would be an interesting field of study to explore the possibilities of automated ways to interpret the data and identify recurring patterns between outages.

6.1.1 Trend Analysis

During the analysis of the spontaneous throughput loss, we very briefly touched upon the subject of trend analysis. We consider this to be a stepping stone towards predicting failures. Network device failures could be preceded by anomalous trends in device metrics or syslog messages.

At last, we have tried to implement a trend analysis framework to identify anomalies in the network metrics data. Due to constraints both in the Grafana and InfluxDB toolset and in time, we were not able to successfully implement such a framework.

6.1.2 Supervised Machine Learning

Duenas et al. have trained a Random Forest model to predict future failure events based on the historical failures[22]. To train the model, they used a labelled dataset.

Future work would be to evaluate if their framework is applicable to syslog and network metrics data. In order to use supervised learning, a dataset needs to be labelled. This can either be done manually or through the use of Natural Language Processing (NLP) of ticketing data.

6.1.3 Unsupervised Machine Learning

Nadeem et al. show great potential in the use of Long-Short Term Memory Networks[14]. Although their work specializes in identifying network failures based solely on syslog, future work could be to use their framework to predict these failures beforehand. An interesting prospect is the potential ability to predict failures we have not encountered before.

7 Conclusion

We set out to determine to what extent it is possible to predict intermittent network device failures based on data from multiple sources. Given the current set of data and event descriptions, we were unable to create a reliable model to predict network device failures.

However, we identified metrics and syslog messages relevant to the permanent failure of line-cards on the core routers. We are capable of identifying three types of line-card failures using patterns between different data sources. We found 73 instances where the total throughput of a Ciena 5410 switch was severely impacted. We posed a hypothesis to validate our belief. Not only did we prove our hypothesis, we also proved the correlation between the device syslog messages and network metrics.

Furthermore, we also proved that some events will not generate any messages that could indicate anything happening, but we showed that these events can be retraced given the right device metrics. Also, the SNMP polling frequency of once every five minutes limits the data resolution. A more in-depth analyses would be possible given a more frequent polling frequency.

References

[1] Felix Salfner, Maren Lenk, and Miroslaw Malek. "A Survey of Online Failure Prediction Methods". In: *ACM Comput. Surv.* 42.3 (Mar. 2010), 10:1–10:42. ISSN: 0360-0300. DOI: 10.1145/1670679.1670680. URL: https://dl.acm.org/citation.cfm?doid=1670679.1670680.

[2] R. Gerhards. *The Syslog Protocol.* RFC 5424. http://www.rfc-editor.org/rfc/rfc5424.txt. RFC Editor, Mar. 2009. URL: http://www.rfc-editor.org/rfc/rfc5424.txt.

[3] R. Presuhn. *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP).* STD 62. http://www.rfc-editor.org/rfc/rfc3416.txt. RFC Editor, Dec. 2002. URL: http://www.rfc-editor.org/rfc/rfc3416.txt.

[4] *Next generation netwerkmonitoring.* https://blog.surf.nl/next-generation-netwerkmonitoring-waar-kiest-surfnet-voor/. Accessed: 2018-06-19.

[5] Daniel Turner et al. "California fault lines: understanding the causes and impact of network failures". en. In: *ACM SIGCOMM Computer Communication Review* 40.4 (Aug. 2010), p. 315. ISSN: 01464833. DOI: 10.1145/1851275.1851220. URL: http://dl.acm.org/citation.cfm?doid=1851275.1851220 (visited on 08/08/2018).

[6] InfluxData. *InfluxDB - Time Series Database Monitoring Analytics.* July 2018. URL: https://www.influxdata.com/.

[7] *SNMPWALK.* http://net-snmp.sourceforge.net/docs/man/snmpwalk.html. Accessed: 2018-07-12.

[8] splunk. *SIEM, AIOps, Application Management, Log Management, Machine Learning, and Compliance | Splunk.* July 2018. URL: https://splunk.com.

[9] Atlassian. *Jira - Ticketing and Issue Tracking Software.* July 2018. URL: https://www.atlassian.com/software/jira.

[10] Corporation for Education Network Initiatives in California. *About CENIC.* https://cenic.org/about/about-overview. 2018.

[11] SURFnet. *About SURFnet.* https://www.surf.nl/en/about-surf/subsidiaries/surfnet. 2018.

[12] Daniel Turner et al. “A comparison of syslog and IS-IS for network failure analysis”. en. In: ACM Press, 2013, pp. 433–440. ISBN: 978-1-4503-1953-9. DOI: 10.1145/2504730.2504766. URL: https://sci-hub.tw/http://dl.acm.org/citation.cfm?doid=2504730.2504766 (visited on 08/08/2018).

[13] Matthew Roughan et al. “Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies”. In: *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems.* SIGMETRICS ’04/Performance ’04. New York, NY, USA: ACM, 2004, pp. 416–417. ISBN: 1-58113-873-3. DOI: 10.1145/1005686.1005745. URL: http://doi.acm.org/10.1145/1005686.1005745.

[14] Moin Nadeem et al. “Automating Network Error Detection using Long-Short Term Memory Networks”. en. In: *arXiv:1806.02000 [cs]* (June 2018). arXiv: 1806.02000. URL: http://arxiv.org/abs/1806.02000 (visited on 08/09/2018).

[15] Shenglin Zhang et al. “PreFix: Switch Failure Prediction in Datacenter Networks”. In: *Proc. ACM Meas. Anal. Comput. Syst.* 2.1 (Apr. 2018), 2:1–2:29. ISSN: 2476-1249. DOI: 10.1145/3179405. URL: https://dl.acm.org/citation.cfm?doid=3203302.3179405.

[16] A. Markopoulou et al. “Characterization of Failures in an Operational IP Backbone Network”. In: *IEEE/ACM Transactions on Networking* 16.4 (Aug. 2008), pp. 749–762. ISSN: 1063-6692, 1558-2566. DOI: 10.1109/TNET.2007.902727. URL: https://sci-hub.tw/http://ieeexplore.ieee.org/document/4456903/ (visited on 08/08/2018).

[17] *SURFnet protected/redundant fiber paths.* https://www.surf.nl/diensten-en-producten/surflichtpaden/aansluitmogelijkheden-surflichtpaden/protected-redundante-lichtpaden/index.html. Accessed: 2018-07-14.

[18] *Syslog message: CRC link error detected for FPC.* PFE.* fabric plane.*.* https://kb.juniper.net/InfoCenter/index?page=content&id=KB19554&cat=AIS&actp=LIST. Accessed: 2018-06-21.

[19] N. Finn, D. Mohan, and A. Sajassi. *802.1ag - Connectivity Fault Management.* STD. http://www.ieee802.org/1/pages/802.1ag.html. IEEE, Dec. 2007. URL: http://www.ieee802.org/1/pages/802.1ag.html.

[20] Wikipedia. *IEEE 802.1ag — Wikipedia, The Free Encyclopedia.* http://en.wikipedia.org/w/index.php?title=IEEE%20802.1ag&oldid=672571099. [Online; accessed 15-July-2018]. 2018.

[21] Ciena. *5410 Service Aggregation Switch, Fault and Performance, SAOS 7.3.* Docs. 009-3219-009 Rev B. Dec. 2007.

[22] J. C. Duenas et al. “Applying Event Stream Processing to Network Online Failure Prediction”. In: *IEEE Communications Magazine* 56.1 (Jan. 2018), pp. 166–170. ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1601135.