# Report 1: Data Description and Analysis

Andres Chaves (706801)
achaves@student.unimelb.edu.au
*Melbourne School of Information*
*The University of Melbourne*

March 16, 2015

### Abstract

The purpose of this document is to describe and analyse the available data from which the experiments will be executed.

Throughout this document we will understand what is the data, how much data is available and we will see some examples of relations between the data.

# Introduction

The purpose of my Research Project is to explore how Machine Learning technologies can be used for aiding the Network Management process, specifically the Fault Management process.

One of the key Information Technology components in Fault Management is the Fetwork Management System (FMS). A FMS is a system which receives all the alarms from the different Network Elements (NEs), performs several data transformation, and at the end stores an enriched alarm and allows their visualisation:

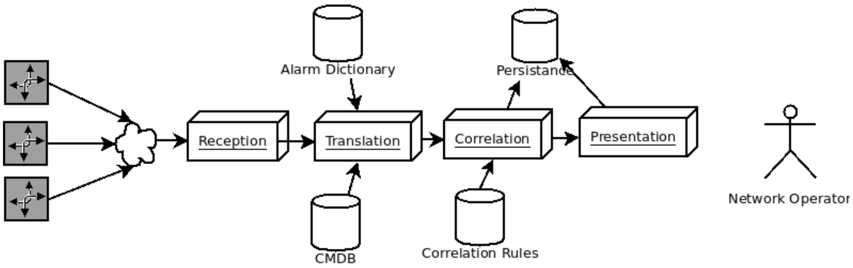


Figure 1: *A schematic of a generic Fault Management System*

The processing stages might include the following:

- Translation: Consist on receive the alarm in a given standard, de-codify it, translate it and enrich it into useful data. This is a required step.
- Correlation: Consist on establish rules for alarm suppression and correlation to present only key information to the network operator.

In order to conduct the experiment and analysis the candidate data will be the one stored by the NMS and for the experiments I will use the data of a specific NMS built in one of the companies I formerly worked on.

# Chapter 1

# Simple Network Management Protocol SNMP

- Protocol RFC, year - MIB, Trap, information, bindings, oid

# Chapter 2

# Fault Management System

Based on the overall components of a generic FMS in Figure 1, in the former company I worked for, we built a specific FMS using several Open Source software.

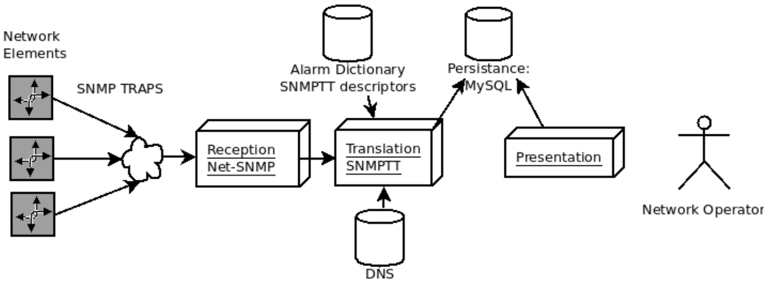


Figure 2.1: *Our built Fault Network Management System*

The FMS receives alarms using the SNMP protocol[1]. The Open Source components used are described as follows:

- Net-SNMP: Net-SNMP is an open source Linux and Unix package that implements the SNMP protocol. The role of this component will be the reception of alarms by using the snmptrapd daemon[2].
- SNMPTT: SNMPTT or SNMP-Trap Translator is an open source component that takes the SNMP trap (alarm) received by Net-SNMP and by using an alarm dictionary (to interpret the alarm name and its attributes) and a DNS (to gather the NE name) translates it into a more useful event. SNMPTT can also enrich the alarm from a database by executing a shell script.[3]
- MYSQL: MySQL is an open source relational database. For this project, it will be used as storage of the Alarms and Configuration Management Database[4].

# Chapter 3

# Network Description

layers (AC, AG, CN), elements in a Node, Suppliers

### 3.0.1 Host Naming Convention

It is important to describe the host naming convention used for the network elements because it provides useful information. A host name is a 19 character string which follows the convention NEROLE_BRAND_MODEL_STATE_TOWN_C (i.e: AC_ER_SP10_ST_VCH_1):

- Network Element Role: Two letters that correspond to the type of node where the network element is located. The type of node can be AC (Access), AG (Agregator), CN (Concentrator), CR (CORE), IG (Internet Gateway) and RC (Rectifier).
- Brand: Two letters that correspond to the fabricator name of the equipment. Somes examples are: ER, AL, DT.
- Model: Four letters that codifies to the NE's model. Some examples are: SP10, SA30, DP29.
- State: Two letters that represents the state where the network element is physically located.
- Town: Three letters encoding the town where the network element is physically located.

# Chapter 4

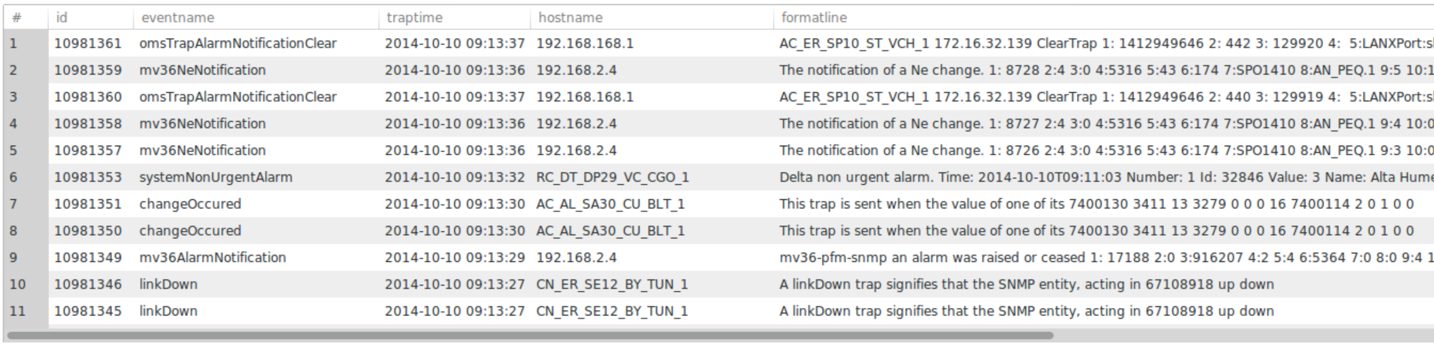
# Data Specification

The FMS specified in the last section stores the alarms in a table on a Relational Database System. Each month of alarms is stored in a separated table. For the purpose of this analysis we used the table for the month of October, 2014.

## 4.1 Data Attributes

The relevant attributes of the alarms stored are:

- Id: Unique numeric identifier of each alarm.
- Eventname: Correspond to the translated name of the alarm's OID as informed by the SNMPTT dictionary.
- Traptime: Timestamp of arrival of the trap.
- Hostname: Ip or host name of the entity that sends the alarm. A DNS is queried in order to translate the IP Address, if the translation is successful the name is stored otherwise the ip is inserted. Most of the host names follow a naming convention.
- Formatline: Semi-structured formatted line of the alarm. Normally it includes all the alarm's trap bindings.



| # | id | eventname | traptime | hostname | formatline |
|---|---|---|---|---|---|
| 1 | 10981361 | omsTrapAlarmNotificationClear | 2014-10-10 09:13:37 | 192.168.168.1 | AC_ER_SP10_ST_VCH_1 172.16.32.139 ClearTrap 1: 1412949646 2: 442 3: 129920 4:  5:LANXPort:s |
| 2 | 10981359 | mv36NeNotification | 2014-10-10 09:13:36 | 192.168.2.4 | The notification of a Ne change. 1: 8728 2:4 3:0 4:5316 5:43 6:174 7:SPO1410 8:AN_PEQ.1 9:5 10:1 |
| 3 | 10981360 | omsTrapAlarmNotificationClear | 2014-10-10 09:13:37 | 192.168.168.1 | AC_ER_SP10_ST_VCH_1 172.16.32.139 ClearTrap 1: 1412949646 2: 440 3: 129919 4:  5:LANXPort:s |
| 4 | 10981358 | mv36NeNotification | 2014-10-10 09:13:36 | 192.168.2.4 | The notification of a Ne change. 1: 8727 2:4 3:0 4:5316 5:43 6:174 7:SPO1410 8:AN_PEQ.1 9:4 10:0 |
| 5 | 10981357 | mv36NeNotification | 2014-10-10 09:13:36 | 192.168.2.4 | The notification of a Ne change. 1: 8726 2:4 3:0 4:5316 5:43 6:174 7:SPO1410 8:AN_PEQ.1 9:3 10:0 |
| 6 | 10981353 | systemNonUrgentAlarm | 2014-10-10 09:13:32 | RC_DT_DP29_VC_CGO_1 | Delta non urgent alarm. Time: 2014-10-10T09:11:03 Number: 1 Id: 32846 Value: 3 Name: Alta Hume |
| 7 | 10981351 | changeOccured | 2014-10-10 09:13:30 | AC_AL_SA30_CU_BLT_1 | This trap is sent when the value of one of its 7400130 3411 13 3279 0 0 0 16 7400114 2 0 1 0 0 |
| 8 | 10981350 | changeOccured | 2014-10-10 09:13:30 | AC_AL_SA30_CU_BLT_1 | This trap is sent when the value of one of its 7400130 3411 13 3279 0 0 0 16 7400114 2 0 1 0 0 |
| 9 | 10981349 | mv36AlarmNotification | 2014-10-10 09:13:29 | 192.168.2.4 | mv36-pfm-snmp an alarm was raised or ceased 1: 17188 2:0 3:916207 4:2 5:4 6:5364 7:0 8:0 9:4 1 |
| 10 | 10981346 | linkDown | 2014-10-10 09:13:27 | CN_ER_SE12_BY_TUN_1 | A linkDown trap signifies that the SNMP entity, acting in 67108918 up down |
| 11 | 10981345 | linkDown | 2014-10-10 09:13:27 | CN_ER_SE12_BY_TUN_1 | A linkDown trap signifies that the SNMP entity, acting in 67108918 up down |

Figure 4.1: *An example of the stored data*

## 4.2 Data Overview

For this particular month of October 2014 we have the following overall information:

| | |
|---|---|
| Total Alarms | 2350714 |
| Different Alarm Types | 351 |
| Different Host Names | 3272 |
| Average of Alarms Received per Minute | 53 |
| Peak of Alarms Received in a Minute | 714 |

## 4.3 Data Analysis

In order to have a better insight of the data and to understand better what should be the work required by the learning algorithm, we described 5 different examples of correlation between two alarms.

For each correlation example we established the statistical frequency of, in a time window of 1 minute, what is the occurrence of alarm T1, the occurrence of the alarm T2, the occurrence of the alarm T1 given the occurrence of the alarm T1.

$\forall x \in X, \quad \exists y \leq \epsilon$ The following table summarises the frequencies:

| Scenario | P(at) | P(a2) | P(a2—at) |
|---|---|---|---|
| AC Outage + Link Down | 0.2214 | 0.3886 | 0.11 |

### 4.3.1 Scenario 1: AC Outage + Link Down, Fabricator 1

### 4.3.2 Scenario 2: AC Recovery + Link Up, Fabricator 1

### 4.3.3 Scenario 3: AC Outage + Link Down, Fabricator 2

### 4.3.4 Scenario 4: AC Recovery + Link Up, Fabricator 2

### 4.3.5 Scenario 5: Fuse Breaker Alarm + Link Down, Fabricator 1

# Bibliography

[1] IETF RFC 1157, http://www.ietf.org/rfc/rfc1157.txt?number=1157.

[2] Net-SNMP website, http://www.net-snmp.org/.

[3] SNMPTT website, http://snmptt.sourceforge.net/docs/snmptt.shtml.

[4] MySQL website, http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html.