Impact of Rare Alarms on Event Correlation

Anne Bouillard
ENS / INRIA
45 rue d'Ulm
75230 Paris cedex 05, France
anne.bouillard@ens.fr

Aurore Junier
INRIA / IRISA
Campus de Beaulieu
35000 Rennes, France
aurore.junier@inria.fr

Benoit Ronot
Alcatel-Lucent Bell Labs
Villarceaux,
91620 Nozay, France
benoit.ronot@alcatel-lucent.com

***Abstract*—Nowadays, telecommunication systems are growing more and more complex, generating a large amount of alarms that cannot be effectively managed by human operators. The problem is to detect significant combinations of alarms describing an issue in real-time. In this article, we present a powerful heuristic algorithm that constructs dependency graphs of alarm patterns. More precisely, it highlights patterns extracted from an alarm flow obtained from a learning process with a small footprint on network management system performance. This algorithm helps to detect issues in real-time by effectively delivering concise alarm patterns. Furthermore, it allows the proactive analysis of the functioning of a network by computing the general trends of this network. We evaluate our algorithm on an optical network alarm data set of an existing operator. We find similar results as the expert analysis performed for this operator by Alcatel-Lucent Customer Services.**

I. Introduction

Network management, especially for issue resolution, has in the last decade become a very complex task. To face the increasing network complexity, more and more alarm types have been defined, and most of them no longer refer to any critical problem (see [5] for example). Networks can easily produce billions of alarms a day that can no longer be effectively managed manually. Furthermore, this huge quantity of information is highly redundant: a single fault can have several symptoms, can be propagated, or can persist, which generates numerous alarms.

The principle of event correlation is to group alarms referring to the same problem (to reduce redundancies) and to highlight those referring to probable faults. Many studies in recent years respond to these challenges, based on event correlation, or on alarm pattern retrieval algorithms, both of which are limited (see Section II). Therefore, automatic techniques that process the large quantity of alarms are now compulsory to build and operate reliable networks.

In this paper, our aim is to develop an efficient method that highlights alarms corresponding to root causes of problems and that can be applied to any kind of network without making any assumption on its characteristics. Our research is based on the assumption that the alarms that occur most frequently are those referring to general information about the network. As a consequence, we focus our study on the observation of non-frequent alarms.

The paper is organized as follows. In Section II, we discuss related works. Section III introduces the method that defines patterns of non-frequent alarms. Based on these, dependency graphs are constructed to identify alarms that might lead to severe faulty behavior (Section IV). Finally, we show the evaluation and the results of the proposed method in Section V.

II. State of the art

Here, we present examples of various approaches. A more detailed state of the art is presented in [2].

Method based on graph dependency. In [6], Katzela and Schwartz present a pioneer event correlation heuristic to find the root cause of anomalies. They first construct a statistical dependency graph of the network elements (nodes, links) and build, for each alarm appearing in the system, the set of objects that might have sent it. Finally, using this graph, they compare a set of localization algorithms.

Methods based on specific architecture. Many articles design a correlation engine directly in the network management system and improve some aspects to reduce the raw flow of alarms by only sending the most pertinent information. For example, in [7] Ross and White define an alarm architecture using the principle of Model-Based Reasoning and a rule-based approach. Their Alarm Correlation Engine of Northern Telecom has been implemented in Smalltalk and contains rule writers to maintain the knowledge of the network (*i.e.* the set of rules that defines a specific problem). However rule-based approaches are limited in their scalability.

Method based on linear algebra. In [3], an alarm is defined as redundant to another if it occurs close in time to the other for most of its occurrences (represented by a Gaussian curve centered on the actual occurrence time). A linear computation is performed to detect redundancies. However, this approach does not take into account alarms occurring between the alarms designated as redundant, which provides additional important information.

Method based on pattern definition. Another way to correlate alarms is to create patterns of frequent alarms by performing data mining on the flow of alarms (see Dousson and Duong, [4]). However, this iterative method is at least $O(i^5)$ in the number of iterations, which remains computationally too complex to be implemented in practice.

Method based on probabilistic finite state machine. In [8], Rouvellou and Hart correlate alarms using probabilistic finite state machines (PFSM) describing faults. A PFSM represents the succession of alarms that imply a particular

ISBN 978-3-901882-53-1, 9th CNSM and Workshops ©2013 IFIP

problem. Unfortunately, the set of possible faults (which can be very large) must be defined in advance.

Method based on correlation coefficient: Yang, in [10], presents a technique where the alarms are represented by Gaussian functions and statistics are used to correlate them. Then, the links between alarms are represented on a colored map. Unfortunately, this method is off-line which does not meet our dynamic requirements.

The main difficulty to tackle the correlation problem resides in the ability to deal with a huge quantity of data to detect dangerous network behaviors. Here we introduce a generic method that efficiently creates several dependency graphs of alarms. It is based on modifications of the sequence of alarms, which enables a fast study of the event correlation problem.

III. Construction of relevant patterns

We use the notations from the language theory. Consider A is a non-empty finite set. Then A^* (resp. A^+) denotes the set of finite sequences (resp. non-empty finite sequences) with elements in A. For $f \in A^*$, $|f|$ stands for the length of f and for $a \in A$, $|f|_a$ is the number of occurrences of a in f. The symbol '$\cdot$' represents the concatenation (if $f_1 = a_1 \cdots a_i$ and $f_2 = b_1 \cdots b_j$, then $f_1 \cdot f_2 = a_1 \cdots a_i b_1 \cdots b_j$). Also, $\mathcal{P}(A)$ stands for the set of subsets of A. The *support* of $f \in A^*$ is $\overline{f} = \{a \in \mathcal{A} \mid |f|_a \geq 1\} \in \mathcal{P}(A)$. With this notations and $A = \{a, b, c, d\}$, $f = abcbca \in A^*$ and $f \in \mathcal{A}^+$, $|f| = 6$, $|f|_a = 2$ and $\overline{f} = \{a, b, c\}$.

We now focus of a flow of alarms. Let $\mathcal{A}$ be the set of alarms names and f be the sequence of those alarms names in a log file. We assume that some alarms appear very frequently, while some others only appear a few times, which is what we observe in the log files studied. Our goal is to search for a group of non-frequent alarms. To highlight them we perform several transformations on the flow: a) Identification of the most frequent alarms; b) Construction of set patterns; c) Reducing the set patterns in size.

a) Identification of the most frequent alarms: This is simply done by counting the number of occurrences of each alarm $a \in \mathcal{A}$ that appears in f, $|f|_a$, and using a fixed threshold α. Then, the set of frequent alarms is defined by

$$M = \{a \in \mathcal{A} \mid |f|_a / |f| \geq \alpha\}.$$

b) Construction of set patterns: In the sequence of alarms, some alarms are sent several times. Therefore, if an alarm a arrives before an alarm b, this does not always mean that a is the cause of b or that a has been generated before b. Moreover, we may often find patterns $ababa$... in the log files. Furthermore, before being monitored, the alarms are temporarily buffered. Consequently, the exact order between the alarms has a weaker impact than for a real-time monitoring, which is rare and expensive in infrastructure costs. As a consequence, for alarms that are consecutive or almost consecutive, we may not have to keep them ordered and short patterns of f can be considered as sets of alarms instead of sequences.

Let $R = \mathcal{A} \setminus M$ be the set of alarms that are not the most frequent. Then, there is a unique decomposition of f as $f = m_0 \cdot r_1 \cdot m_1 \cdot r_2 \cdots r_\ell \cdot m_\ell$, where $m_0, m_\ell \in M^*$ $m_1, \ldots, m_{\ell-1} \in M^+$, and $r_1, \ldots, r_\ell \in R^+$. The sequence of alarms we focus on is the concatenation of non-frequent set patterns:

$$sp(f) = \overline{r_1} \cdot \overline{r_2} \cdots \overline{r_\ell} \in \mathcal{P}(R)^*.$$

Example 1: Due to lack of space we only give an example to illustrate the construction, from which an interpretation cannot be made. A more complete example is described in [2]. Take $\mathcal{A} = \{a, b, c, d, e\}$ and $f_{ex} = aabeaabacbabaccaabadbaceacaaceabacbabaacaaddcaeeaa$. We have $|f_{ex}| = 50$, and $|f_{ex}|_a = 24$. Then, with $\alpha = 0.4$, we find $M = \{a\}$ and $R = \{b, c, d, e\}$. Then, the sequence of non-frequent set patterns can be computed: $sp(f_{ex}) = \{b, e\}\{b\}\{b, c\}\{b\}\{c\}\{b\}\{b, d\}\{c, e\}\{c\}$ $\{c, e\}\{b\}$ $\{b, c\}\{b\}\{c\}\{d, c\}\{e\}$. We get $|sp(f_{ex})| = 17$ but the rarest alarm d appears in only 2 set patterns. A much more schematic view of the rare alarms is needed.

c) Reducing the set patterns in size: The third step consists of reducing the length of $|sp(f)|$. This can be done by setting transformation rules. For $u, v \in \mathcal{P}(R)$,

$$(R_1)\ uvu \to u \cup v \qquad (R_2)\ uv \to u \text{ if } v \subseteq u.$$

In other words, if $sp(f)$ can be written as $z_1 \cdot uvu \cdot z_2$, then it can be transformed into $z_1 \cdot (u \cup v) \cdot z_2$ using rule (R_1). If $v \subseteq u$ and $sp(f) = z_1 \cdot uv \cdot z_2$, then it can transformed into $z_1 \cdot u \cdot z_2$ using rule (R_2).

We recursively apply these rules until no further rule can be applied. Note that the choice of the order to apply the rule may slightly affect the result. We arbitrary chose repeatedly to apply first rule (R_1) from left to right and then (R_2) until no rule can be used. We denote by $rsp(f)$ the *reduced set pattern* obtained after applying rules (R_1) and (R_2).

Example 2: After the application of those rules to $sp(f_{ex})$ of Example 1, $rsp(f_{ex}) = \{b, e\}\{b, c\}\{b, d\}\{c, e\}$ $\{b, c\}\{c, d\}\{e\}$: its length is reduced from 17 to 7. The reduction is not drastic in this example because the number of alarms is too small. The reduction of the length will be greater for the sequences considered in Section V.

IV. Relevant patterns detection

In this section, we use the heuristic of the previous section in order to find patterns leading to rare alarms. Rare alarms only appear a few times in the log files: this corresponds to alarms that lead to faulty behavior of the system and are often repaired very soon after the occurrence of the alarm. Identifying patterns leading to that fault may then allow a proactive management by detecting the root cause of a problem before it implies a failure.

In order to handle this, we construct the *dependency graph* $\mathcal{G} = (V, E, w)$ of the set patterns found with our heuristic: $\mathcal{G}$ is a weighted directed graph with set of vertices V, set of edges E and weight function $w : E \to \mathbb{N}$, where

- $V = \overline{rsp(f)}$, the set patterns that appear in $rsp(f)$;
- $E = \{(u, v) \mid \exists z_1, z_2$ such that $sp(f) = z_1 \cdot uv \cdot z_2$;

- $w(u,v) = |\{(z_1, z_2) \mid sp(f) = z_1 \cdot uv \cdot z_2\}|$, the number of occurrences of uv in $rsp(f)$.

Example 3: Figure 1 represents the graph corresponding to the log file of Example 1. The weights of the edges are all equal to 1, so they are not represented. We observe that alarm d always appears with c or b, and is preceded by $\{b, c\}$. We may deduce that there is a correlation between those alarms.

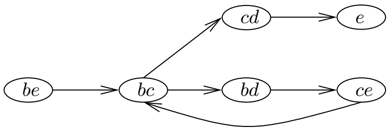

Fig. 1. Dependency graph of f_{ex}, the log file from Example 1.

A difficulty in analyzing the flow of alarms lies in the fact that the quantity of alarms to manage is huge and many pathologies can occur together. Consequently, we divide the dependency graph in sub-graphs, each focusing on the study of a small set of rare alarms. Observing these graphs gives a refined analysis of the rare alarms that provides hypotheses about the root cause that generated the alarms.

V. Experiments and results

We evaluate our method using the real network issues of an operator, a customer of Alcatel-Lucent Customer Services. We focus our analysis on a network element, NE, of the Optical Synchronous digital hierarchy (SDH), from a retrieved log file of 36K alarms over one year. Let us note that its full optical layer is composed of more than 62K elements that are able to generate alarms. Currently, each element analyzes the large amount of information received by a traditional management method, which is a difficult task.

We focus on a frequent issue: the laser failure of an optical SDH network element. Table I lists the alarms found in the log file studied. Several alarms are symptomatic of such an issue: LOS, AIS, RUP, and RUM. In particular, occurrences of alarm LOS indicate that the whole signal is unusable: it is replaced by an AIS consisting in continuously sending binary 1s. This produces occurrences of alarm AIS in every device downstream the fault (see [9]).

To guide the expert for analyzing the network, we first use the algorithm proposed in [1] to analyze the global flow of alarms. This algorithm allows us to detect on-line the *strong deviations* of the behavior of the flow. Here, the behavior means the rate at which the alarms arrive. This arrival rate is *sandwiched* between two affine curves with the same slope, which are automatically updated to fit the variations of the flow rate.

The flow f we consider is composed of the alarms and their arrival time. We discard the names of the alarms to perform the algorithm of [1]. Figure 2 represents the variation of alarms arrival rates on the global flow f computed with this algorithm. One can observe that rates computed are low and last a relatively long time up to $1.0\,10^7$ s. Afterwards, the range of rates progressively becomes wider with amplified variations. After $1.6\,10^7$ s, the fluctuations reach their maximum and remain so until the end of the observation. From this curve, one can assume that a major problem occurs after $1.0\,10^7$ s.

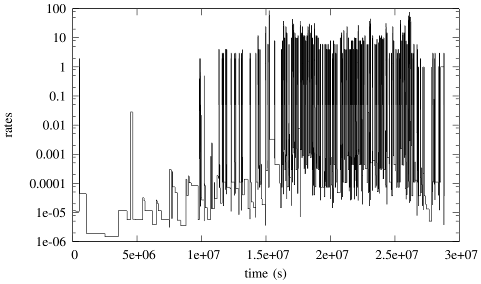

Fig. 2. Arrival rates of the global flow of alarms f.

Let us now focus on the behavior of some alarms: RUP (important for the use-case), but also EBER and FO. For the sake of simplicity, we denote f_a the sub-flow of f containing only occurrences of alarm a.

Figure 3 depicts computed rates for the sub-flows f_{RUP} and f_{EBER}. This graphic shows that RUP occurs in f only at the beginning of the flow, during the first $7.0\,10^6$ s. This ensures the presence of the use-case. Alarm EBER appears all along the observation. However, one can detect bursts of arrivals: before $1.0\,10^7$ s. and after $2.0\,10^7$ s. Processing the same way with alarm FO indicates three arrival time: a single burst at $5.0\,10^5$ s. and at $1.0\,10^7$ s and more occurrences appear after $2.2\,10^7$ s, as shown in Figure 4. RUP seems correlated with EBER and FO. One can observe that the instability on the global flow starts when RUP is not emitted any more. This might highlight that the use-case we focus on is not the major problem of this node.

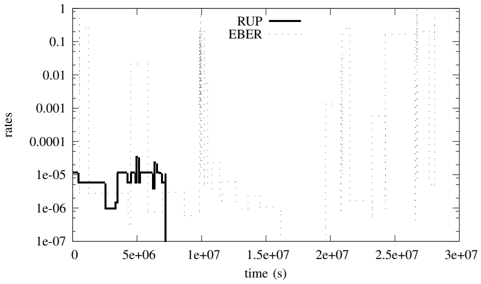

Fig. 3. Arrival rates of sub-flows f_{RUP} and f_{EBER}.

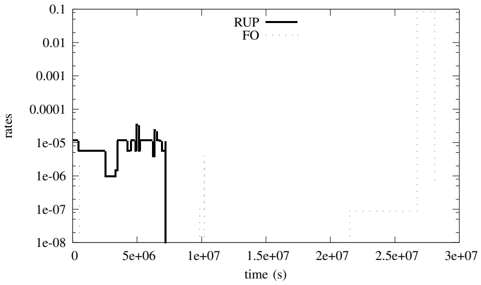

Fig. 4. Arrival rates of sub-flows f_{RUP} and f_{FO}.

Let us now focus on the approach described in Section III. The two columns NE_{α_1} and NE_{α_2} of Table I represent the partition of the alarms between M and R with parameters α_1 and α_2.

Acronym	full name	NE_{α_1}	NE_{α_2}
AIS	Ais	R	R
CP	Cabling problem	R	R
CSF	Communication subsystem failure	R	R
DS	Degraded signal	R	M
EBER	Excessive ber	R	R
FO	Frequency offset	R	R
HK	House keeping	R	R
LOS	Loss of signal	M	M
LOT	Loss of timing sources	R	R
NI	Node isolation	R	M
RDI	Remote defect indication	R	R
RI	Resource isolation	R	R
RUM	Replaceable unit missing	R	R
RUP	Replaceable unit problem	R	R
SSF	Server signal failure	R	R
U	Unequipped	M	M
UT	Unavailable time	R	R

TABLE I
LIST OF THE ACRONYMS OF THE ALARMS.

Our heuristic algorithm used with $\alpha_1 = 0.3$ identifies $M = \{$U, LOS$\}$, which represents 97% of the total occurrences of alarms. Figure 5 shows the full dependency graph created that represents the partial ordering of the 7 set patterns (of non-frequent alarms) detected in f. Note that here, the graph is so simple that it is acyclic. Group denotes the set of alarms $\{\mathrm{DS}, \mathrm{NI}, \mathrm{CSF}, \mathrm{EBER}, \mathrm{RDI}\}$ and the double slash bar indicates that the occurrences of the two set patterns are separated by a long time period. This means that between the occurrence of these two set patterns only alarms in M appear. Consequently, alarms relevant to a problem only occur at the beginning and the end of f.

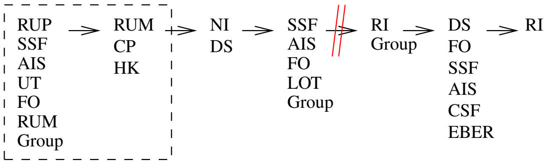

Fig. 5. Set pattern graph with α_1= 0.3.

The alarms leading to a repair can be identified as RUP and RUM. Indeed, those alarms appear less frequently than the others. From Figure 5, one can deduce the probable fault leading to RUM or RUP. It is also clear that HK and CP are related to RUM. As RUP comes in a very large set pattern (may not be very meaningful), we detail the dashed rectangle of Figure 5 on Figure 6 by setting $\alpha_2 = 0.003$. This gives $M = \{$U, LOS, DS, NI$\}$ (column NE_{α_2} in Table I). Now $|sp(f)| = 59$. The correlation between RUM, HK and CP is maintained. Also, RUP mainly appears after the occurrence of SSF and CSF.

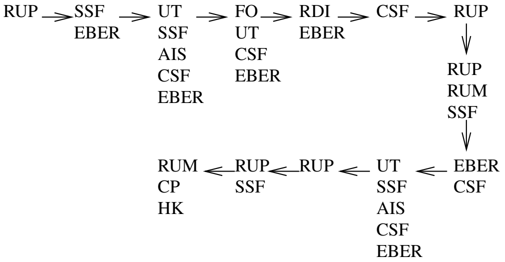

Fig. 6. Detail of the set pattern graph with $\alpha_2 = 0.003$ during the occurrences of RUP and RUM.

To conclude, the observation of EBER and UT in the dependency graph shows that the issue is affecting a demodulator of the node. This is confirmed by the presence of alarm FO in a set pattern with UT and EBER. At the end, RUM and CP show that the element has been changed, as they are directly followed by HK. Figure 5 indicates that no further RUP or RUM is emitted after a time, meaning that the problem has been fixed. Figure 2 shows that a later issue, impacting more NE than the former one, exists. Figure 5 highlights that the problem is linked to occurrences of alarms CSF and SSF (present in almost each set pattern). This indicates that a problem comes from the optical link of the network element or from its neighbor connected through this link.

Due to space restriction, a more complex network element is presented in [2].

VI. Conclusion

This article presents a method to correlate events in a network. Our idea is based on the principle that due to the huge alarm variety, which progressively increases, most alarms no longer refer to any critical problem. Consequently, we believe that a fault is highlighted by non-frequent alarms.

The method developed is realized in two steps. We first use an earlier work to provide an overview of network functioning. Then, the new algorithm is used to create a dependency graph of sequences of alarms (set patterns) from a studied flow. From this graph, rare alarms (probably referring to critical problems) are extracted. Finally, we focus on these alarms and small parts of the graph to express hypotheses about the network state.

Both algorithms used are very light in computational complexity and memory usage. Currently, this method studies one flow of alarm at a time. Our future work will enhance it by providing an automatic expertise combining the correlation results of all flows of alarms from every network element.

References

[1] Anne Bouillard, Aurore Junier, and Benoît Ronot. Hidden anomaly detection in telecommunication networks. In *Conference on Network and Service Management (CNSM'12)*, pages 82–90, 2012.

[2] Anne Bouillard, Aurore Junier, and Benoît Ronot. Alarms correlation in telecommunication networks. Research Report RR-8321, INRIA, 2013.

[3] ControlArtsInc, editor. *Alarm System Engineering*. e-book, 2010. http://www.controlartsinc.com/Support/Publications.html.

[4] Christophe Dousson and Thuang Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 620–626. Morgan Kaufmann Publishers, 1999.

[5] Bill Hollifield and Eddie Habibi. *The Alarm Management Handbook: Seven Effective Methods for Optimum Performance*. ISA, 2007.

[6] Irene Katzela and Mischa Schwartz. Schemes for fault identification in communication networks. *IEEE/ACM Transactions on Networking*, 3(6):753–764, 1995.

[7] Nial Ross and Tony White. An architecture for an alarm correlation engine. In *Object Technology*, 1997.

[8] Isabelle Rouvellou and George W. Hart. Automatic alarm correlation for fault identification. In *IEEE International Conference on Computer Communications (INFOCOM'95)*, pages 553–561, 1995.

[9] International Telecommunication Union. ITU-T recommendation G.774 SDH management information model for the network element view, 2001.

[10] Fan Yang. Improved correlation analysis and visualization of industrial alarm data. In *18th IFAC World Congress*, pages 12898–12903, 2011.