Towards a Formal Model for the Network Alarm Correlation Problem

Jacques-H. Bellec, M-Tahar Kechadi
School of Computer Science & Informatics,
University College Dublin, Belfield, Dublin 4, Ireland.

Abstract

In telecommunication networks, alarms are usually useful for identifying faults, and therefore solving them. However, for large systems the number of alarms produced is so large that the current management systems are overloaded. One way of overcoming this problem is to filter and reduce the number of alarms before the faults can be located. In this paper, we describe a new approach for fault recognition and classification in large telecommunication networks. We introduce a new model and present another way of understanding the alarm correlation problem.

1 Introduction

Telecommunication networks are growing in size and complexity, and therefore their management is becoming more complicated. Each network element can produce a large amount of alarms when a fault is detected. The telecommunication network management system is in charge of the recording of the alarms generated by the nodes in the network and presents them to the operator. However, due to the high volume and the fragmented nature of the information, it is impossible to quickly solve the faults. Moreover, somes changes in the network such as new equipments, updated software, and network load, mean that the alarms can be very different in nature [1]. More precisely, when a fault occurs, devices or components can send messages to describe the problem that has been detected, but they only have a local view of the error. Due to the complex nature of these networks, a single fault may produce a cascade of alarms from the affected network elements. In addition, a fault can trigger other faults, for instance in the case of overloading. Even though failures in large communication networks are unavoidable, quick detection, identification of causes, and resolution of failures can make systems more robust, more reliable, and ultimately increase the level of confidence in the services that they provide [2].

Alarm correlation is a key functionality of a network management system that is used to determine the faults' origin, and to filter out redundant and spurious events. The alarm correlation systems generally combine causal and temporal correlation models with the network topology. The power and robustness of the models used and the algorithms developed vary from system to system. However, due to the absence of any simple, uniform, and precise presentation of the alarm-correlation problem, it is very difficult to compare their relative power, or even to analyze them for their properties. In general, data mining techniques are adapted towards the analysis of collections of data, as they can find the redundant data sequences. Generally, to analyse such a sequence, the most frequent episodes of data must be found. Unfortunately the domain of telecommunication networks has a particular behaviour compared to other data sets [3, 4], so most of the data mining techniques can not directly be applied to the alarm correlation problem

In this paper we focus on the Behavioural Proximity model (BP). Its main objective is to reduce considerably the number of alarms by clustering them according to their behaviour, to form events. Then these events are correlated to form clusters via the Event Duration Matching (EDM) algorithm. As a result, only importantl seeds of global events are presented to the network operator, helping him in identifying and solving the faults in the network. Our model incorporates a fuzzy core, which provides fuzzy results, namely results with a degree of trust. Due to a limited space, we will not give details of the different algorithms but only focus on the theorical model.

The paper is organised as follows: in the next section, we describe the work which has already been done in this domain. Then in section 3, we describe our model by defining each notion introduced in our approach. Finally, we conclude in section 4.

2 Background

In the past, network fault management were performed by human experts. The size and complexity of today's networks, however, have made the levels of human intervention required to perform this function prohibitively high. Currently, many systems employ event correlation engines to address this issue [5, 6]. The problem of an automatic identification of correlated events has been tackled from various perspectives. Model traversal approaches aim to represent the interrelations between the components of the network, [7] or the causal relations between the possible events in the network [8], or a combination of the two [9]. Rule-based [10] and code-based [11] systems also model the relations between the events in the system, specifying the correlations according to a rule-set or codebook. Other techniques, such as neural networks [12, 13] or decision trees, have also been applied to this task. These approaches vary in the level of expert knowledge required to train the system. Neural networks, for example, can require no expert input whereas model-based techniques may be fully reliant on the insights of human experts. The domain of sequential data mining addresses the specific problem of identifying relationships or correlations between events in a raw dataset, which is inherently sequential in nature, such as fault data consisting of a series of time stamped events. Mining sequential patterns can be viewed as a subset of the problem of mining associations between dataset elements in general, constrained by the temporal aspects of the data. But to deal with this, the temporal aspect is not the only factor that we have to consider. In fact, the particular nature of telecommunication networks gives some strong relationships between alarms behaviour that we cannot find in other kind of data sets.

In the field of telecommunication networks, related work done in [2] used association rules and frequent episodes to discover alarm patterns, which were subsequently used in the development of alarm correlation systems. However, the methods used in this research do neither capture the notion of alarm similarity, nor the incertitude related to the network. Futhermore, association rules and frequent episodes have the drawback of generating many uninteresting and redundant alarm patterns [14]. In the next sections, we will present our formalism which overcomes these drawbacks.

3 The Behavioural Proximity Model

This section presents the formal model of our Alarm correlation technique. Fault localization is a process of isolating faults responsible for the observable malfunctioning of the system. This can be done by trying to find some correlation between alarms. Given a set of alarms (dataset), the problem is to present to the operator only a small number of alarms that are highly considered to be the cause of root faults. Before presenting the technique in section 4, we need to define some key notions of the model.

- **Definition 1.** A fault is a disorder occurring in the hardware or software of the network [15]. Faults happen within the network components, and alarms are external manifestations of faults. Faults can be classified into four categories : hardware, software, telecommunication, and environment. We call F_S the total set of faults which can occur in a network, and F_i a particular fault defined as $F_i =< S_w, S_p, S_s > .i \in [1, n]$, where S_w is the set of alarms gathered in an event warning, before the fault appears, S_p is the set of alarms which can be viewed as the primary symptoms, and S_s the secondary symptoms. Off course, theses sets can be empty or composed by just one alarm, but they are finite and disjoint:

  $$S_p \cap S_w = \emptyset, S_p \cap S_s = \emptyset, S_w \cap S_s = \emptyset. \qquad (1)$$

  The faults behaviour can be caracterized by three states according to the faults duration time, as permanent, intermittent or transient. Permanent faults exist until some repair actions are taken. Intermittent faults occur on a discontinuous or periodic basis, causing degradation of service for short periods of time. Transient faults can cause a temporary and minor degradation of a service. They can be easily identified and repaired.

- **Definition 2.** An Error is the result of a fault. It can be identified by the difference between an observed, or measured value and a theoretically correct value or condition . Many errors may be generated by just a single fault. Some errors may result in a deviation of a delivered service from the specified service or component that is visible to the outside world. The term failure is used to denote this type of visible error. Some errors are not visible externally. Thus, errors may propagate within the network causing failures in other hardware or software components. In order to correct an error, its corresponding fault has to be resolved; therefore, errors are typically not handled directly. We can try to approximate an error by identifying the event, namely the bunch of alarms which have been sent to notify the presence of the error. To resume, we can write that the Faults F_i can lead to some errors Err_j visible under the form of alarm events e_k. $F_i \Rightarrow \sum_{j=1}^{n} Err_j \Rightarrow \sum_{k=1}^{n} e_k$.

- **Definition 3.** A Symptom is an external manifestation of failures. Symptoms are observed as alarms notifications of a potential failure. These notifications may originate from management systems which monitor the network status or from probes widespread around the network. Some faults may be directly observable, i.e., they are problems and symptoms at the same time. However, many types of faults are unobservable due to (1) their intrinsically unobservable nature, (2) local corrective mechanisms built into the management system that destroy evidence of fault occurrence, or (3) the lack of management functionality necessary to provide indications of fault existence. Examples of intrinsically unobservable faults include livelocks and deadlocks. Some faults may be partially observable the management system provides indications of fault occurrence, but the indications are not sufficient to precisely locate the fault.

  In a communications network, a single fault may cause a number of alarms to be delivered to the network management center. Multiple alarms may be a result of (1) fault re-occurrence, (2) multiple invocations of a service provided by a faulty component, (3) generation of multiple alarms by a device for a single fault, (4) detection and notification about the same network fault by many objects (hardware or software network components) simultaneously, and (5) error propagation to other network objects causing them to fail and generate additional alarms.

- **Definition 4.** An alarm consists of a notification of the occurrence of a specific event, which may or not represent an error. An alarm report is a kind of event report used in the transportation of alarm information. Alarms defined by vendors and generated by network equipment are messages observable by network operators, giving information about particular behaviours of the system. There may be many alarms generated for a single event. All the alarms are logged into a centralized management system, in text format files. According to the information architecture in telecommunication networks, an alarm can be thought of as an object. The attributes of the alarm try to describe the event that triggered it. This is a possible set of alarm attributes:

  – Event timestamp: gives the time the alarm was issued

  – Logged time: gives the time the alarm was recorded

  – Perceived severity: gives a state ranging from critical to indeterminate

  – Alarm ID: identifies the alarm by a unique serial number

  – Alarm Key: it is the key composed by all alarm attributes but the ones related to the time

  – Node ID: identifies the node in the subnetwork

  – Event Type: gives some indications of the nature of what happened

  – Probable Cause: gives some indication of why it happened

  – Specific Problem: clarifies what happened

  We call A the set of all possible alarms and a_k the k^{th} alarm in the dataset. Each alarm can be defined by a set of static parameters noted $\lambda(a_i)$ and by a set of parameters related to the time noted $\delta(a_i)$. In other words, $a_i = \lambda(a_i) + \delta(a_i)$. We define identical alarms if they have the same static content with different timestamps and logged times. Namely, a_i is identical to $a_{i'}$ only if $\lambda(a_i)$is equals to $\lambda(a_{i'})$. From a raw data set, we can gather the alarms with the same static attributes, and calculate the exact number of different kinds of alarms.

- **Definition 5.** We can caracterize the **behaviour of an alarm** a_i, firstly according to its redundancy, then to its periodic nature. If we can retreive one or multiple occurences of this particular alarm, then we can try to detect if there is a constant periode between them. And if it is the case, we can affirm that these set is composed by **Periodic alarms**. We call **Single alarms** the alarms that are unique, i.e we cannot find a similar alarm in the laspe time considered. We call **Twin alarms** the alarms that appear only twice. With only two alarms we cannot determine a periodic behaviour. When a set of alarms is not periodic, namely we cannot determine a specific period among them, we call them **Aperiodic alarms**. The choice between a periodic and aperiodic is not an easy one because of some delayed alarms, missing alarms and overlapped events, that can give a wrong standard deviation and so, give a wrong caracterization of the behaviour. We created an algorithm to answer to these needs, nammed ABR, which uses fuzzy logic to determine the nature of the behaviour of each family of alarm. Its main advantage is that it can pinpoint with accuraccy the behaviour via giving a degree of trust to different caracterizations. The fuzzy formalism of ABR is the following:

- Let A be a set of alarms noted as a_i. Thus, $A = \sum_{i=1}^{n} a_i$.
- Let e_i be a fuzzy set in A, characterized by a membership function $\psi e_k(a_i)$ which maps each point in A onto the real interval [0, 1].
- $e_k \in \emptyset \iff \forall a_i, \psi e_k(a_i) = 0.0$.
- $e_j = e_k \iff \forall a_i : \psi e_j(a_i) = \psi e_k(a_i)[or, \psi e_j = \psi e_k]$.
- $\psi e'_k = 1 - \psi e_k$.
- $e_j \subset e_k \iff \psi e_j \leq \psi e_k$.
- $e_l = e_j \cup e_k$, where: $\psi e_l(a_i) = MAX(\psi e_j(a_i), \psi e_k(a_i))$.
- $e_l = e_j \cap e_k$ where: $\psi e_l(a_i) = MIN(\psi e_j(a_i), \psi e_k(a_i))$.

- **Definition 6.** A event is a set of correlated alarms. Let E be the set of all possible sets of events and e_i an event of E. The event recognition is the first part of the recognition process in the BP model. The Alarm Behavioural Recognition (ABR) algorithm takes care of the event recognition and gives in output the sets representing all kind of different alarms we can find in the data.

The membership function for each event e_i is called $\psi e_i(a_k)$, $\forall a_k$ we have $\psi e_i(a_k) \in [0, 1]$. The first correlation rule is based on the static attributes of the alarms, and gives a crips set:

- If $\lambda(a_j) = \lambda(a_k)$, then $a_j \in e_i$ and $a_k \in e_i$, then $\psi e_i(a_k) = \psi e_i(a_j) = 1$.

We developped a new event correlation algorithm called EDM for Event Duration Matching. Its aims are twofold: it scores the events by importance and classifies them in three categories, and it proceeds to the correlation to give in output some clusters. Here is a possible set of event attributes:

- Event ID : it gives the unique ID of the event
- Start Time: gives the minimum apparition time of all embedded alarms in the event
- End Time: gives the maximum apparition time of all embedded alarms in the event
- Score: gives a score calculated according to the relevance of the attributes of the alarms
- Gravity : it is the average time of apparition of the alarms
- Event Key: the key composed by all alarm keys
- Code Type: identifies the nature of the event (Primary, secondary or tertiary)
- Nb Alarms : number of embedded alarms

The scoring function χ uses the static attributes defining the nature of the alarm representing the event. It is completly deterministe, but the classification which follows, uses fuzzy logic.

$$\chi(e_i) = \chi(a_j) = \chi(\lambda(a_j)) \tag{2}$$

In order to identify the most relevent events, EDM classifies them in three categories, as *Primary, Secondary and Tertiary.* Primary events are the most important as they can be viewed as highly probable primary symptoms. Tertiary events are low interesting because they describe some telecommunication failures which are the most common effects in a telecommunication network. Finally, secondary events are somewhat interesting but not enough to be considered in the first place. As we said, it is not easy to find the boundaries between each class, a not much better score does not mean for sure that the event is in the upper class. To answer to this problem, we built in EDM with a fuzzy system, to get a result with a degree of trust and then give different results according to the network operator's point of view. The fuzzy formalism of EDM is the following:

- Let e_i be an event $\in E$, c_k a cluster of events $\in C$, $\Gamma(E)$ a subset of E containing primaries events and a minimal bound of degree of trust α . Thus, $E = \sum_{i=1}^{n} e_i$.
- If $\Psi(e_1, e_j) \leq \alpha$, $\forall j \in [2, n]$, and $e_1 \in Gamma(E) \Rightarrow e_1 \in c_k$.
- Let $\Delta(E)$ a subset of E containing Secondaries rated events, and $\Theta(E)$ a subset of E containing Tertiaries rated events. Thus, $\Gamma(E) \cup \Delta(E) \cup \Theta(E) = E$

- If $a_j \in e_i$ and $a_k \in e_{i+1} \Leftrightarrow \lambda(a_j) \neq \lambda(a_k)$
- If $a_j \in e_i$ and $a_k \in e_i \Leftrightarrow \lambda(a_j) = \lambda(a_k)$
- If $a_j \in e_i \Rightarrow \exists a_k \in e_i / \lambda(a_j) = \lambda(a_k)$
- Let the membership function of $\Gamma(E)$ be $\Phi_\Gamma(e_i)$ for an event $e_i \in E$, $e_i \in \Gamma(E) \Leftrightarrow \Phi_\Gamma((e_i) \geq High_Bound$
- Let the membership function of $\Delta(E)$ be $\Phi_\Delta(e_i)$ for an evente$_i \in E$, $e_i \in \Delta(E) \Leftrightarrow High_Bound \geq \Phi_\Delta(e_i) \geq Low_Bound$
- Let the membership function of $\Theta(E)$ be $\Phi_\Theta(e_i)$ for an event $e_i \in E$, $e_i \in \Theta(E) \Leftrightarrow \Phi_\Theta(e_i) \leq High_Bound$
- $|\Gamma(E)| \geq |C|$

- **Definition 7.** A cluster is a set of correlated events. In other words, it is what our technique produces and presents to the network operator. The number of clusters is not predefined, but must be significantly low compared to the number of raw input alarms. Basically, the number of clusters returned by the BP technique represents more or less the number of faults that should be solved by the operator. We call C the set of all possible cluster c_k, as $C = \sum_{k=1}^{n} c_k$. A cluster is a final result according to a degree of trust α preliminary etablished during the first phase of our technique by the network operator. The aggreated events which compose the clusters are said independant with a certain degree of trust α and with a specified scope.

  This is a possible set of cluster attributes:

  - Cluster ID : it gives the unique ID of the event
  - Start Time: gives the minimum apparition time of all embedded alarms in the event
  - End Time : gives the maximum apparition time of all embedded alarms in the even
  - Total Score: gives a total score calculated according to the sum of all events scores which composed the cluster.
  - Gravity : it is the average time of apparition of the alarms
  - Event Root: the root event
  - List of Events : the list of all events
  - Nb Alarms : number of embedded alarms

  The clusterisation rule is the following:
  Let e_i, e_j two events in$E$ θc_k the membership function of c_k.

- if $e_i \in Prim(E)$ or $e_j \in Prim(E)$, and $\theta(e_i, e_j) \geq \alpha$ then $e_i \cup e_j \in c_k$.

- **Definition 8.**

  We define Inter-correlated events by the fact: at least one of the correlated events belongs to the different class of network elements. By Inter-correlation constraint, we make focus on discovering multi-level correlation relations among the differents natures of the components.

- **Definition 9.**

  We define Intra-correlated events by the fact: all of the events in the correlation rules must belong to the same class of network element i.e. MS, BTS or BSC. By Intra-correlation constraint, we can discover the correlation relations in the same level of the telecommunication netwok.

- **Definition 10.**

  Scope defines the area of alarms/events occuring in the network. The correlation process can be done for a particularly interesting scope for the network operator. It can be viewed as a constraint which limits the correlation process to a certain level of the network hierarchie. it can be particularly interesting for the network operator, to get only a close up to a certain scale of the network, and so have a more accurate view of what is going on there.

- **Problematique**

  We previously introduced some notions, it is now time to explain the problem we try to solve according to our model. As we said, the main goal is to present to the operator a small number of alarms, highly representative of the faults which appeared in the network. To do so, we have to find the correlations rules for the alarms. These rules are most of the time unknown to the operator, who identify the caracteristic sympthoms of the fault and then, the fault itself, according to his knowledge about the network, and the history of fault recovery. When a fault appears, the different symptoms which follow are a cascade of errors. Errors are not directly viewable, but if we retreive good correlation rules we would be able to identify the events which are the external magnifestation of the errors. And this, according to a degree of trust, whose limit is specified by the user. The first step consists into the identification of the errors which follow the fault appearance. As we have just a limited view of what is going on in the netwok, we can try to approximate the errors by identifiying the events of alarms. Then a correlation between these events must be done, to recognize dependant and independant events and then present to the operator some plausible sets of events which can represent the faults with a strong degree of trust.

4 Conclusion

The main contribution of this paper is the proposition of a new model for alarm correlation which satisfies the network opertor's needs. It provides the main roots of faults which appeared in the network in the form of clusters. This model has been implemented and is being evaluted with real data sets from a live 3GPP telecommunication network, and as far as we are, it shows some interesting results. For further improvement of our model, we are now integrating some training skills with the use of neural network reasoning.

References

[1] Himberg, J., Korpiaho, K., Mannila, H., Tikanmaki, J., Toivonen, H.: Time series segmentation for context recognition in mobile devices. In: Proc. of the IEEE International Conference on Data Mining, San Jose, California, USA (2001) 203–210

[2] Bouloutas, A., Galo, S., Finkel, A.: Alarm correlation and fault identification in communication networks. IEEE Trans. on Communications **4**(2/3/4) (1994) 523–533

[3] Bellec, J.H., Kechadi, M.T., Carthy, J.: Study of telecommunication system behavior based on network alarms. In: ECML/PKDD'05, Workshop on Data Mining for Business, Porto, Portugal (2005)

[4] Bellec, J.H., Kechadi, M.T., J.Carthy: A new efficient clustering algorithm for network alarm analysis. In: The 17th IASTED Int'l. Conference on Parallel and Distributed Computing and Systems, (PDCS'05), Phoenix, AZ, USA (2005)

[5] Yamanishi, K., Maruyama, Y.: Dynamic syslog mining for network failure monitoring. In: KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, New York, NY, USA, ACM Press (2005) 499–508

[6] Julisch, K.: Clustering intrusion detection alarms to support root cause analysis. ACM Trans. Inf. Syst. Secur. **6**(4) (2003) 443–471

[7] Meira, D., Nogueira, J.: Modelling a telecommunication network for fault management applications. In: Proc. of NOMS'98. (1998) 723–732

[8] Gopal, R.: Layered model for supporting fault isolation and recovery. In: IEEE/IFIP, Proc. of Network Operation and Management Symposium, Honolulu, Hawaii (2000)

[9] Steinder, M., Sethi, A.: Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system. In: Proc. of ICCCN'01, Arizona (2001) 374–379

[10] Liu, G., Mok, A., Yang, E.: Composite events for network event correlation. In: IM'99. (1999) 247–260

[11] Yemini, S., Kliger, S., Mozes, E., Yemini, Y., Ohsie, D.: High speed and robust event correlation. IEEE Communications Magazine **34**(5) (1996) 82–90

[12] Gardner, R., Harle, D.: Alarm correlation and network fault resolution using kohonen self-organising map. In: IEEE Global Telecom. Conf. Volume 3., New York, NY, USA (1997) 1398–1402

[13] Wietgrefe, H., Tuchs, K.D., Jobmann, K., Carls, G., Frohlich, P., Nejdl, W., Steinfeld, S.: Using neural networks for alarm correlation in cellular phone networks. Proc. of IWANNT (1997)

[14] Bellec, J.H., Kechadi, M.T., J.Carthy: Performance evaluation of two data mining techniques of network alarms analysis. In: The 2006 Int'l Conference On Data Mining, (DMIN'06), Las Vegas, NV, USA (2006)

[15] Gardner, R., Harle, D.: Methods and systems for alarm correlation. In: Proc. of Globecom'96, London, UK (1996) pp.136–140