A Configuration Management Assessment Method for SON Verification

Tsvetko Tsvetkov[*], Szabolcs Nováczki[†], Henning Sanneck[‡], and Georg Carle[*]
[*]Department of Computer Science, Technische Universität München
Email: {tsvetko.tsvetkov, carle}@in.tum.de
[†]Nokia, Budapest, Hungary
[‡]Nokia, Munich, Germany
Email: {szabolcs.novaczki, henning.sanneck}@nsn.com

***Abstract*—Over the last years the complexity of mobile communication networks has significantly increased. Therefore, Self-Organizing Network (SON) features have been introduced to automate the process of fault-remedying, configuring Network Elements (NEs), and optimizing their operation. Such features are typically implemented by SON functions which actively perform changes to Configuration Management (CM) parameters in order to achieve certain objectives. However, having such features also requires a mechanism that allows us to verify their actions. In case certain NEs experience an undesired behavior, we have to be able to determine whether it is caused by CM changes and, if so, identify the responsible ones for that to happen. In this paper we propose a novel CM change performance assessment method with dynamic scope management for automated CM undo decision making. Our method includes two key properties: the ability to determine the minimal set of cells that are possibly influenced by a certain CM change, and generate a recommendation to accept or reject it based on the observed network performance. Results from our real data evaluation show our method is able to detect anomalous network behavior and identify the responsible changes.**

I. Introduction

Today, mobile operators need to find an efficient way of managing the increasing complexity of their communication networks. The rapid adaptation of mobile services by users significantly increases the generated data volume, the amount of signaling in the network, and number of generated control events. Hence, Self-Organizing Network (SON) features have been specified and developed to deal with the complex nature of mobile networks like Long Term Evolution (LTE) and LTE-Advanced. SON features usually target the optimal operation of the network, supervise the configuration and auto-connectivity of newly deployed Network Elements (NEs), and are responsible for fault detection and resolution [1].

A SON-enabled network is typically managed by a set of autonomous functions performing specific Network Management (NM) tasks, as shown in Figure 1. These SON functions are often designed as control loops which monitor Performance Management (PM) and Fault Management (FM) data, and based on their goals perform changes of Configuration Management (CM) parameters. Usually, a SON function's goal is given by the operator through a function's configuration [2]. For instance, the goal of the Mobility Load Balancing (MLB) function is to move traffic from high loaded cells to neighbors as far as coverage and interference allows.

978-1-4799-5863-4/14/$31.00 © 2014 IEEE

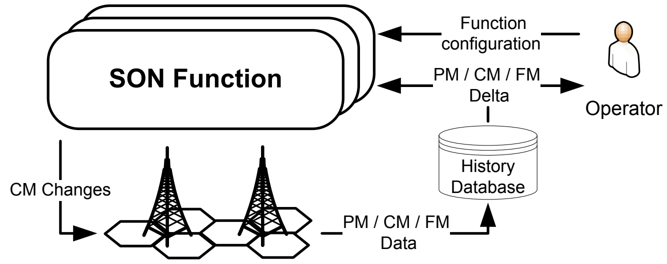

Figure 1. Overview of a SON

However, the increasing reliance on SON features to perform the correct optimization tasks introduces a new set of challenges. In a SON, the impact of each function's action on the environment depends upon the actions of other functions as well. For instance, if the Coverage and Capacity Optimization (CCO) function modifies the antenna tilt, the cell border changes physically which means that the received signal quality changes as well. Obviously, this affects the Handover (HO) performance of neighboring cells which is monitored by a function like Mobility Robustness Optimization (MRO). Therefore, an inappropriate change of the physical cell borders induced by CCO may negatively impact the HO performance and the upcoming decisions taken by MRO as well.

The concept of pre-action SON coordination [3], [4] goes into the direction of solving this problem by defining rules used to anticipate and avoid known conflicts between SON function instances. In addition, the idea of verifying a CM change triggered by a SON function instance and *rolling it back* based on certain rules has been introduced in [5]. Such rules, however, are defined with regards to SON coordination, i.e., only the priorities of the SON function instances are taken into consideration. As a result, we will rollback an accepted function instance's CM change only if another higher prioritized and conflicting one makes a request within the same area and time.

Usually, the task of *discovering subnormal network behavior*, induced either by running SON function instances or by a change made by the operator, is targeted by anomaly detection techniques. Such methods very significantly in the underlying mathematical models, the assumptions about the data they are observing and scope of detection. For instance, in [6] performance indicator normalization is used to detect whether cells are showing an expected behavior or not. In [7], an ensemble method is suggested to compute Key Performance Indicator (KPI) degradation levels that indicate the severity of the experienced anomaly. However, these approaches do not provide an answer to two essential questions: ”why did it happen?” and ”how we can overcome the detected problem?”.

In this paper we try to give an answer to these two questions by introducing a combination of both ideas. We present a *CM assessment method* that evaluates CM changes from performance effects point of view. In case one or more CM changes occur in the network, our method is triggered to generate a scope that includes the set of entities (e.g., cells) that might be possibly influenced by those changes. The resulting scope is then observed whether the included entities are showing anomalous behavior, like a degradation in performance. Based on these observations our CM assessment method generates recommendations to either accept or undo the corresponding CM changes.

The rest of the paper is organized as follows. In Section II we give a general overview of verification in SON. In Section III we present the main building blocks of our CM assessment method. Section IV is dedicated to the evaluation of our approach on real network data. Finally, in Section V we conclude our paper with a summary.

II. Verification in SON

The idea of evaluating whether a certain system or service complies with its requirements and targets has been known for quite a long time. For example, in the field of software engineering verification is known as the process of determining whether the developed software can fulfill all expected requirements. Usually, this is a complex process that may include the usage of formal methods for proving or disproving the correctness of the implemented algorithms and may additionally require extensive tests that examine the behavior of the software at execution-time.

Within the SON area, the term verification, or at least the idea behind it, has been widely used. For instance, in self-configuration the purpose of site build verification is to detect problems that may occur during the installation or integration process of a NE [8]. Such a type of verification may also include shakedown tests that check the network’s accessibility or reliability. In the field of self-optimization, SON functions may track their own CM changes and trigger a CM undo in case they have moved further away from achieving their objective. Even in the area of SON coordination [3], the decision to acknowledge or reject a function’s request can be based on past experiences, i.e., a SON coordinator is not only designed for conflict prevention and resolution, but also verifies whether its decisions have a positive impact on the network performance as well. Such an approach has been introduced in [9], where reinforcement learning was applied in order to improve the coordination of two distributed SON functions, namely MRO and MLB.

In the area of self-healing, several anomaly detection and diagnosis approaches have been proposed. In [10], a framework has been developed to detect anomalous network behavior, perform root cause analysis and provide a corrective action. The presented framework requires each NE to monitor its own operation by measuring several types of performance indicators and uploading them to the Operations Support System (OSS) database. An anomaly detector determines whether the stored data is significantly deviating from the expectations. The corrective action is supplied by the diagnosis part. It learns the impact of different faults on the performance indicators. Furthermore, it employs a scoring system that rewards a given action in case it has had a positive impact on the network.

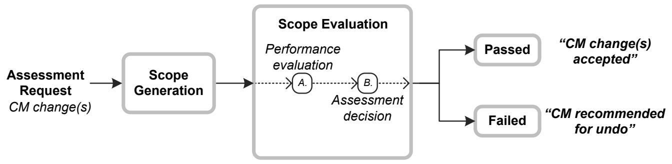

Figure 2. Overview of the CM assessment method

III. Concept Overview

In this section we describe our CM assessment method used to evaluate CM changes from performance effects point of view and place automated recommendations to either accept or undo the corresponding CM change. Note that here we refer to CM changes that may include any type of configuration change possible in a mobile network. For example, this could be a parameter modification of existing NEs, the introduction of new NEs or any other larger scale modification of multiple parameters. The high level overview of the proposed method is depicted in Figure 2.

The CM assessment method may be triggered on demand, for instance, when a SON coordinator accepts the request for a CM change of a SON function instance. Beyond that, it can be executed at regular time intervals. Such an interval may equal the PM granularity interval which can correspond to every hour in case of hourly PM data. The triggering of the assessment process occurs when an *assessment request* is received from the network. Such a message specifies the duration of the assessment process, also called the *assessment interval*. Moreover, it contains information about the CM changes that have been recently made and, therefore, need to be verified. The information about those changes serves as an input to the *scope generation step*.

The outcome of the scope generation is the smallest set of cells whose performance might be affected by the identified CM change(s). The computation can be either manually defined or be based on the impact area of the SON function instance that has made the change. As stated in [3], the impact area gives us information about which parts of the network (e.g., set of cells) are affected by the execution of a particular SON function instance. We see the cells that have been reconfigured by a SON function instance, i.e., the cells included in the function area, as most prone for experiencing anomalies. Furthermore, we consider the effect area and the safety margin as well. The main reason why we take the effect area into account is because it covers all cells that are supposed to experience side-effects after the execution of a function instance. For instance, the load of a cell may change if the transmission power of a neighboring cell has changed. However, the effect area can differ from its original definition. For example, due to an increased network density the effect

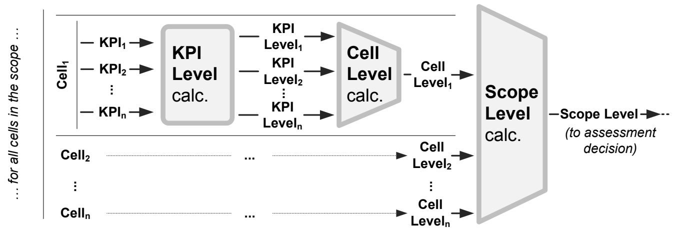

Figure 3. The performance evaluation phase

area can be much larger than assumed. This is why we take the safety margin as well. Note that the safety margin is supposed to extend the border of the impact area which should provide a higher degree of protection against undesired effects.

As soon as the scope generation process is completed, the *scope evaluation step* for the identified CM change(s) is triggered. It consists of two phases which we are going to describe in the following two subsections.

A. Performance Evaluation Phase

The performance evaluation phase of our assessment method is an aggregation scheme with the raw KPI values of individual cells at the input side and a scope-wide performance indicator, called the *scope level*, at the output side. The latter one is devoted to deliver high level performance information about the cells that are in the scope of the CM change being under assessment. In order to compute it, each of the raw KPI values of every individual cell in the scope is converted to a unified anomaly indicator, the *KPI level*. The KPI level represents the similarity of the actual KPI value to the expected (i.e., normal in case of healthy operation) domain. Furthermore, it is a normalized metric which takes values close to one if the actual values are close to the expectations and close to zero if there is a significant difference. The normal behavior is defined by *profiles*, either computed manually or learned from the data set.

Of course, there are several other ways of how we can calculate a KPI level. For instance, we may use a two-sample Kolmogorov-Smirnov test, to compare the distributions of two sets of KPI samples [11]. The first sample set can be collected during a training phase during which the network is showing an expected behavior, and the second set can be collected when the training phase is completed. The distance between the empirical distribution function of the KPI sample and the cumulative distribution function of the reference distribution can be regarded as the level of a KPI.

The KPI level values are aggregated for each individual cell to create a performance indicator, the *cell level*, that reflects the overall cell performance. The aggregation method used depends on the number of input KPIs. In case of a few input KPIs (typically less than 40) the recommended method is to use a weighted average of the KPI levels, where weights represent the importance of the KPIs in the aggregation. Otherwise, it is reasonable to identify, e.g., the 5th percentile of the cell level distribution. Finally, the cell levels are aggregated in the same manner, which provides the scope level used in the next phase: the assessment decision.

B. Assessment Decision Phase

The assessment decision phase is an anomaly detection and recommendation process which uses the scope level to detect performance degradation of the cells within the scope. There are two possible outcomes of the process:

- case ***passed***: it means that the assessment has passed and, therefore, the system recommends the acceptance of the corresponding CM change.
- case ***failed***: it means that the assessment has failed which leads our method to exit the assessment process with the concluding recommendation that the CM change should be undone.

There are several ways of how we can build a detector. In the simplest case the detector can be based on setting a threshold for the scope level and detect whether it has been exceeded. However, more sophisticated detectors might be required to increase the reliability and accuracy of the detection and the final recommendation.

IV. Evaluation

In this section we present an evaluation of the introduced CM assessment approach. At first, we give a description of the used data set. Then, we present the results from a selected showcase of our verification method detecting a temporal Physical Cell Identity (PCI) collision case.

A. Data Set Description

The data set we are using consists of PM and CM data dumps for a time period of approximately three weeks generated by a live LTE network. It includes the performance and configuration data of 1230 Evolved NodeBs (eNBs) for an LTE macro network. In addition, we know that two SON functions have been actively performing changes during this time period: the Automatic Neighbor Relation (ANR) and the PCI allocation function. Furthermore, we take the following eight KPIs with hourly granularity into consideration:

- `RRC_CONN_SETUP_SR` : success rate of the elementary procedure Radio Resource Control (RRC) connection setup.
- `EUTRAN_ERAB_SETUP_SR_nGBR`: success rate of the elementary procedure EUTRAN Radio Access Bearer (E-RAB) setup for non Guaranteed Bit Rate (GBR) services.
- `ERAB_DR`: E-RAB drop rate. The rate of abnormally dropped bearers.
- `INTER_eNB_HO_SR`: inter eNB HO success rate.
- `EUTRAN_RLC_PDU_RETR_R_DL`: retransmission rate for Radio Link Control (RLC) Protocol Data Units (PDUs) in downlink direction.
- `EUTRAN_RLC_PDU_RETR_R_UL`: retransmission rate for RLC PDUs in uplink direction.
- `EUTRAN_CELL_AVA_excBLU`: shows cell availability, excluding blocked by user state (BLU).

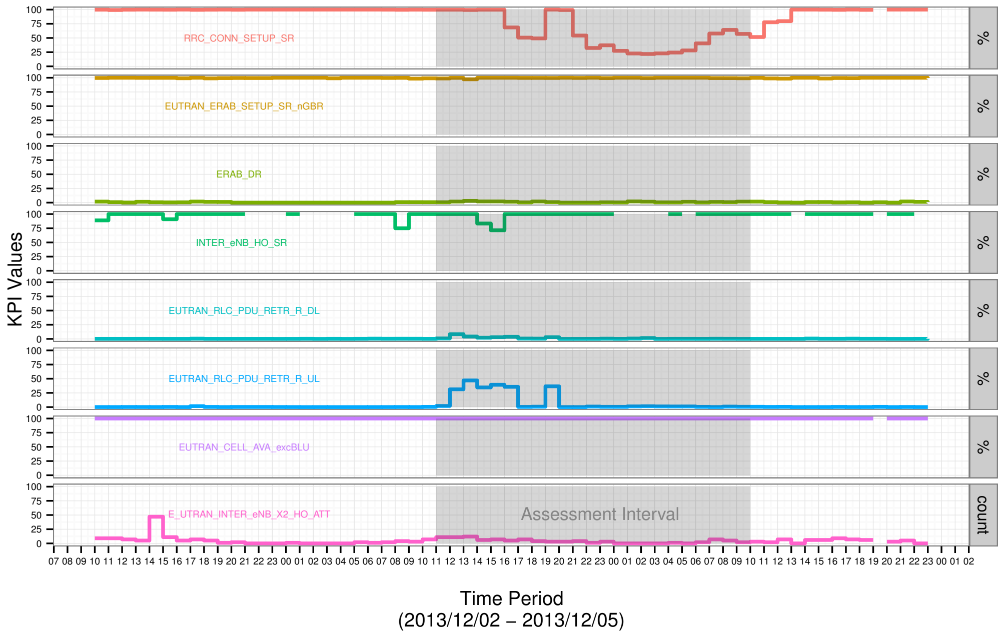

Figure 4. The KPI statistics of the degraded cell

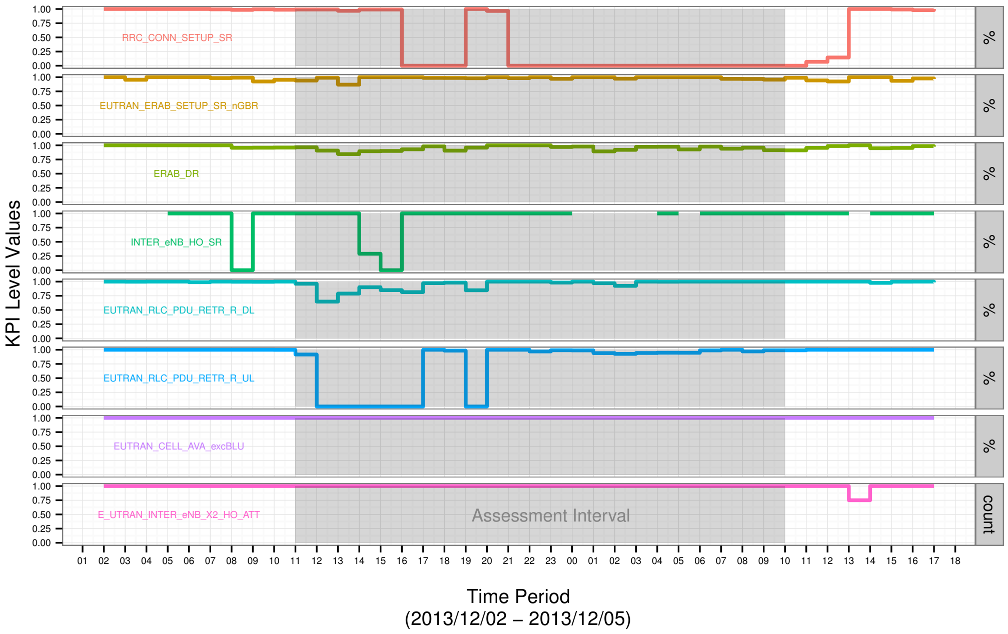

Figure 5. The KPI level of the degraded cell

- `EUTRAN_INTER_eNB_X2_HO_ATT`: number of inter eNB (X2-based) HO attempts.

As for the CM data, we observed the changes made by the PCI function instances that have been active during the given time period. More precisely, we evaluated the actual PCI changes of cells as well as the update of the PCI parameter of the corresponding adjacencies. Furthermore, we have grouped the CM changes hourly and per base station, i.e., the subject base station.

B. *Results*

The scope was selected by taking the cells of the target base station and all of the radio neighbors of these cells (based on the neighbor relation lists), which resulted in 107 cells being part of the scope. The profiles of the KPIs and the KPI levels were computed as described in [11]. The computation of the cell level is based on the average KPI level. The required scope level was computed as the 5th percentile of the distribution of the cell level values. In addition, we have set the detection threshold to 0.95, and declared an assessment failure in case

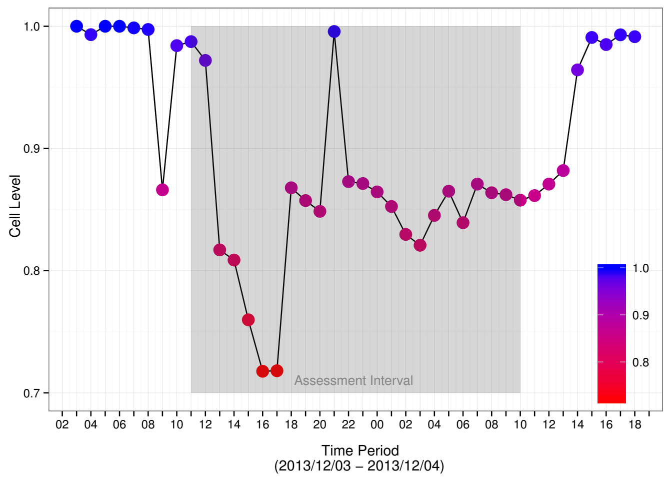

Figure 6. The cell level result

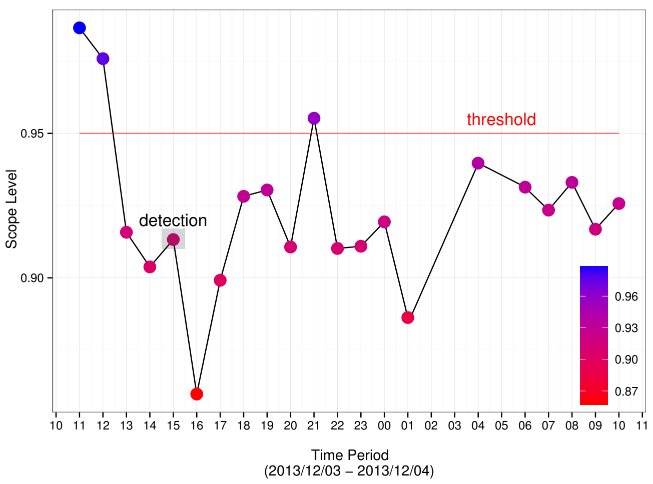

Figure 7. The scope level result

the scope level is below that threshold after three consecutive hours.

During the experiment, we verified 505 PCI changes which were applied during the observed time interval along with the corresponding parameter changes in the configuration database. Figure 4 to 7 depict the results that caught our attention. While several assessment intervals (duration of 24 hours) detected moderate degradations directly after PCI modifications, only during one of them we could detect an actual PCI collision case which led to a higher than usual performance degradation. Our observations have shown that there was a temporal PCI collision causing service degradation within one of the cells. In the first figure we see the KPIs of that particular cell for the time period during which these events took place. The second figure shows the resulting KPI levels. The last two figures depict the computed cell level and the scope level. In particular, the last figure outlines the assessment interval of interest and the exact moment where our CM assessment method has detected the degradation and, therefore, recommended the corresponding CM change for an undo. Note that the figures have some undefined data points due to missing data from the network.

V. Conclusion

In this paper we proposed a CM assessment method used for the verification of CM changes in a SON. Our approach consists of two major steps: the generation of the scope that includes the cells possibly impacted by the given changes, and the scope evaluation step during which we observe the selected area from a performance point of view. The goal of our method is to recommend the CM changes that have occurred during that time interval either for an accept or an undo.

The results from our experiment, which is based on CM and PM data from a LTE macro network, show that the CM assessment method is able to detect network anomalies caused by CM changes. In particular, we were able to detect PCI collisions, identify the degraded cell, and recommend the corresponding CM change for an undo.

The performance evaluation and assessment decision steps described in this paper represent one possible assessment strategy which we consider as an initial approach. Our future work will be devoted to further evaluation including more KPIs, either from real network data or from simulations. Furthermore, we also will target alternative anomaly detection and diagnosis techniques.

References

[1] S. Hämäläinen, H. Sanneck, and C. Sartori, Eds., *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Chichester, UK: John Wiley & Sons, Dec. 2011.

[2] 3GPP, "Telecommunication management; Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)," 3rd Generation Partnership Project (3GPP), Technical Specification 32.522 V11.7.0, Sep. 2013.

[3] T. Bandh, "Coordination of autonomic function execution in Self-Organizing Networks," PhD Thesis, Technische Universität München, Apr. 2013.

[4] T. Bandh, R. Romeikat, and H. Sanneck, "Policy-based coordination and management of SON functions," in *12th IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM 2011) Work.* Dublin, Ireland: IEEE, May 2011, pp. 827–840.

[5] R. Romeikat, H. Sanneck, and T. Bandh, "Efficient , Dynamic Coordination of Request Batches in C-SON Systems," in *IEEE Veh. Technol. Conf. (VTC Spring)*. Dresden, Germany: IEEE, Jun. 2013.

[6] P. Kumpulainen, M. Särkioja, M. Kylväjä, and K. Hätönen, "Finding 3G Mobile Network Cells with Similar Radio Interface Quality Problems," in *Engineering Applications of Neural Networks, L. Iliadis and C. Jayne, Eds. Springer Berlin Heidelberg*, 2011, pp. 392–401.

[7] G. Ciocarlie, U. Lindqvist, K. Nitz, S. Nováczki, and H. Sanneck, "On the Feasibility of Deploying Cell Anomaly Detection in Operational Cellular Networks," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, May 2014.

[8] Ericsson, "Transparent Network-Performance Verification For LTE Rollouts," White Paper, 284 23-3179 Uen, Sep. 2012.

[9] O. Iacoboaiea, B. Sayrac, S. B. Jemaa, and P. Bianchi, "SON Coordination for Parameter Conflict Resolution: A Reinforcement Learning Framework," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2014.

[10] P. Szilágyi and S. Nováczki, "An Automatic Detection and Diagnosis Framework for Mobile Communication Systems," *IEEE Trans. Netw. Serv. Manag.*, vol. 9, no. 2, pp. 184–197, Jun. 2012.

[11] S. Nováczki, "An Improved Anomaly Detection and Diagnosis Framework for Mobile Network Operators," in *9th International Conference on Design of Reliable Communication Networks (DRCN)*, Mar. 2013.