OPEN ACCESS
OPUS AUGSBURG

Universität Augsburg
Universitätsbibliothek

Detection and resolution of ineffective function behavior in Self-Organizing Networks

Christoph Frenzel, Tsvetko Tsvetkov, Henning Sanneck, Bernhard Bauer, Georg Carle

Angaben zur Veröffentlichung / Publication details:

Frenzel, Christoph, Tsvetko Tsvetkov, Henning Sanneck, Bernhard Bauer, and Georg Carle. 2014. "Detection and resolution of ineffective function behavior in Self-Organizing Networks." In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, 19 June 2014, Sydney, NSW, Australia*, 1–3. Los Alamitos, CA: IEEE. https://doi.org/10.1109/wowmom.2014.6919014.

Nutzungsbedingungen / Terms of use: **licgercopyright**

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

Detection and Resolution of Ineffective Function Behavior in Self-Organizing Networks

Christoph Frenzel*‡, Tsvetko Tsvetkov†, Henning Sanneck‡, Bernhard Bauer* and Georg Carle†
*Department of Computer Science, University of Augsburg, Augsburg, Germany
Email: {frenzel, bauer}@informatik.uni-augsburg.de
†Department of Computer Science, Technische Universität München, Munich, Germany
Email: {tsvetkov, carle}@net.in.tum.de
‡Nokia Solutions and Networks Research, Munich, Germany
Email: henning.sanneck@nsn.com

Abstract—**The Self-Organizing Network (SON) paradigm enables automated management of next generation mobile communication networks by introducing a set of autonomous functions that jointly achieve self-configuration, self-optimization, and self-healing. Each SON function is driven by a specific objective to optimize one or more Key Performance Indicators (KPIs). If a function encounters situations in which the given targets cannot be achieved, it can produce undesired behavior, resulting in an impaired SON. This paper presents the idea of SON Operational Troubleshooting function and outlines a concept for the automatic detection, analysis, and mitigation of operational problems.**

I. Introduction

The Self-Organizing Network (SON) approach is seen as a key enabler for automated network operations in next generation mobile communication networks by introducing self-configuration, self-optimization, and self-healing capabilities [1]. A SON-enabled network is managed by a set of autonomous functions performing specific network management tasks. These SON functions are designed as control loops which monitor Performance Management (PM) data, Fault Management (FM) data, and Configuration Management (CM) data, and adjust network configuration parameters. Thereby, each SON function tries to achieve an operator-given objective to optimize one or a set of Key Performance Indicators (KPIs). For instance, the Mobility Robustness Optimization (MRO) SON function aims to keep the rate of handover failures between two mobile network cells below a given threshold.

Since the objectives are interrelated, the SON functions can interfere with each other at run-time. This might be caused by different SON functions that monitor the same or a partially overlapping set of KPIs in the same or adjacent network areas. Likewise, equal network parameters may be adjusted by different SON functions. It is the task of SON coordination to prevent or resolve such conflicts, which might cause unwanted behavior and, hence, preclude flawless network operation [2].

At run-time, a SON function might encounter situations in which it cannot achieve its targets. However, the problems that cause the function failure can often be resolved by another SON function. For instance, an MRO SON function might not be able to reduce handover failures if there is a coverage hole, which can be handled by a Coverage and Capacity Optimization (CCO) SON function though. The operation of SON functions is usually not monitored which leaves such problems unnoticed and rendering the affected SON functions valueless. Even worse, due to coordination, such problems may also affect the operation of other SON functions, impairing the performance of a whole SON.

This paper presents the idea of SON Operational Troubleshooting (SONOT), i.e., the detection of situations in which a SON function cannot achieve its objective and the determination of countermeasures to overcome the problem, particularly preempting the execution of that SON function and taking the appropriate further steps beyond its scope. Furthermore, a concept for a SONOT function is outlined that realizes the idea, in the spirit of SON, as a SON function. As shown by the results of a preliminary evaluation, the concept can improve network performance in specific problem situations.

II. Operational Troubleshooting in SON

A SON function might encounter numerous problems at run-time that prevent it from achieving its objective. Typically, such problems are outside of the scope and control of the SON function itself. A simple case is that the network is not sufficiently dimensioned to cope with the demand or there is an equipment failure. For instance, if there is a sports game or a concert, the capacity of the network might just not be enough to cope with the traffic. There can also be more subtle problems, e.g., an MRO function continuously tries to optimize handover network configuration parameters, however, due to an undetected coverage hole, the handover KPIs do not improve.

The major problem of an ineffective SON function is that, besides being unable to achieve its own objective, it may also impede other SON functions from achieving theirs. This is due to the interactions between the functions and their coordination. A widespread scheme for SON coordination is run-time action coordination [2] which requires all SON functions to request for permission to change some network parameter at a SON coordination function. This function then determines conflicting requests, e.g., contrary changes of the same network configuration parameters, computes a set of non-conflicting SON functions that can be executed at the same time in the same area, and triggers their execution. Thereby, the conflict resolution is often based on operator priorities [1].

If a function with a high priority encounters a problem, it may block other functions from being executed. This monopolization of the network results in a SON that is trapped in a deadlock, although, the initial problem might be resolved by another SON function. For instance, consider that a high priority CCO(RET) SON function, i.e., a CCO function that optimizes the Remote Electrical Tilt (RET) in a cell [1], detects a coverage hole. If the Transmission Power (TXP) is not correctly set, the CCO(RET) function continuously adapts the tilt of a cell but never closes the coverage hole. Consider also a low priority CCO(TXP) SON function, i.e., a CCO function that solely controls the transmission power in a cell [1], which is able to solve the problem. Due to the high priority requests by the CCO(RET) function, the CCO(TXP) function is constantly blocked and cannot close the coverage hole.

Therefore, a SON function is required that is able to detect situations in which a SON function cannot achieve its objectives. This SONOT function can analyze the problem using a network-wide view and determine possible remedy actions, e.g., blocking functions that cannot achieve their objectives as well as triggering other functions.

There have been only a few research efforts which, at least partially, identified the problem of troubleshooting a SON. The authors in [3] present a coordination approach which prevents monopolization of the network by a SON function through alternately increasing and decreasing priorities. However, the root cause for monopolization has not been analyzed. The research projects SOCRATES [2] and UniverSelf [4] describe concepts to detect and delegate operational problems to higher-layer decision makers or a human operator. Unfortunately, the ideas are rather abstract without further details, e.g., an implementation or case study. The SEMAFOUR project aims at creating a sophisticated SON coordination function that is able to detect oscillations and frequent requests by functions due to a misconfiguration [5]. Upon such a problem, a management component adapts and improves the configuration of the SON functions. However, due to its recent start, there is currently not any further information about the concept.

III. Concept Outline

The concept for operational troubleshooting in a SON, depicted in Figure 1, is twofold. On the one hand, there is an *SON function monitoring* step that detects that a SON function ran into some problematic state which it cannot handle by itself and raises an alarm. On the other hand, this alarm is evaluated and a corrective action is taken in the *analysis and remedy* step. The concept is implemented in a new SON Operational Troubleshooting (SONOT) SON function which can perform complex computations separated from SON coordination.

A. SON Function Monitoring

The system has to detect the problem that hinders a SON function from fulfilling its objective. Based on the input data, there are two general approaches for this: *state-based* and *history-based*. The former solely considers the current network state. In other words, it takes data into account which is directly available at the moment of execution. The latter allows time series-based analyses of the system behavior, and the evaluation of the impact of changes of network parameters. Hence, it enables the analysis of performance trends and the prediction of future network states which allows reasoning

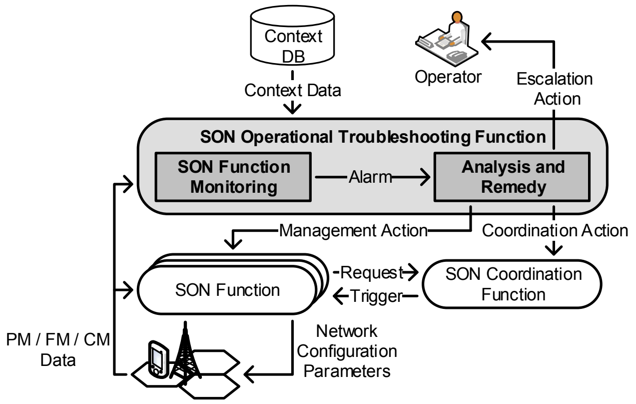

Figure 1. The SONOT function and its interaction with other SON functions

whether a SON function might be able to satisfy its objective. Moreover, it can also be used to track SON functions that are continuously modifying parameters that do not lead to significant improvement, i.e., configuration oscillations.

The SONOT function can continuously analyze the network through its SON function monitoring component. It can draw on a broad view on the network with respect to PM, FM, and CM data which is usually not accessible to regular SON functions. The SONOT function can use a statistical, time-series analysis to determine whether the execution of a SON function actually has a positive impact on network performance or whether a SON function produces oscillations. For instance, if a CCO function is continuously running, this might indicate a sever coverage problem caused, e.g., by an equipment failure. Upon detection of a problem, the SON function monitoring component generates an alarm indicating the detected problem and passes it to the analysis and remedy component for further handling.

B. Analysis and Remedy

The alarms trigger the second step of the troubleshooting process: the analysis and remedy of the problem. The analysis and remedy component analyzes the alarms, infers the root cause of an indicated problem, determines suitable countermeasures and, if required, blocks the execution of non-effective SON functions. The corrective actions can be manifold, e.g., the request for execution of another SON function at the SON coordination function, the adaptation of the configuration of a SON function, or the creation of a trouble ticket to inform the operator about the issue and request manual inspection. For example, if a CCO function raises an alarm that it found an unrecoverable coverage hole, the alarm resolver can infer that the problem might be caused by an equipment failure and, thus, blocks the execution of CCO and triggers a self-healing function like Cell Outage Compensation (COC). Comprehensive contextual information enables the system to establish a detailed picture of the current network status allowing well-informed decisions. It is possible to employ simple decision making approaches like a production rule systems or fuzzy logic system, or sophisticated approaches like influence diagrams or planners.

Requests for the execution of a SON function by the SONOT function need to be coordinated against regular SON function requests. This raises the question how the requests of the SONOT function are prioritized. Thereby, it is necessary that SON coordination considers the priority of the SON function which raised the alarm as well as the SON function that is requested by the SONOT function. Specifically, the SON coordination function needs to set the priority of the request by alarm-triggered functions such that their priority is the maximum of the priority of the alarming and the alarm-triggered function.

IV. Evaluation

In the following, the results of a preliminary evaluation are presented in order to show the advantages of the SONOT concept. Therefore, a scenario which requires the intervention of the SONOT function has been simulated with a SON simulation system similar to the one presented in [6]. Three SON functions are considered: an MRO function that minimizes handover problems, an CCO(RET) function that maximizes a cell's coverage and capacity through the antenna tilt, and a CCO(TXP) function that maximizes a cell's coverage and capacity through the transmission power. These functions are coordinated by a SON coordination function implementing an adaptation of the pre-action batch coordination concept with dynamic priorities [3]. For simplicity, the SON coordination function considers all requested actions concerning the same cell as conflicting.

The SON function monitoring component detects the monopolization of the network by a SON function through history-based analysis. Using exponential smoothing on the read KPI data, it can determine whether a SON function is able to improve a KPI value or whether the continuous adjustments of the network parameters are useless. In the latter case, an alarm is generated indicating the ineffective SON function. The analysis and remedy component uses a policy to map an alarm to coordination actions, specifically, the blockage of the useless SON function in the respective network area and the requesting of another SON function to resolve the problem.

In the scenario, a high priority MRO function cannot improve the handover performance within a cell due to a coverage problem. Figure 2 depicts the performance of the cell around the coverage hole for the case that a SONOT function is present and for the case without it. During the first three simulation rounds the Radio Link Failures (RLFs) and the throughput stay at almost the same level although the MRO function is actively changing the handover parameters. In round 4, however, the SON function monitoring component detects the monopolization of the network because KPIs do not improve and generates an alarm. The analysis and remedy component then blocks the MRO function for several rounds and triggers the CCO(TXP) function. This remedy action changes the transmission power leading to a significant decrease of the RLFs and an increase in the throughput. In contrast, the performance does not improve similarly if the SONOT is not present. In this case, each SON function tries to improve the performance but, due to the alternating execution induced by the SON coordination, the CCO(TXP) requires much more time for the optimization.

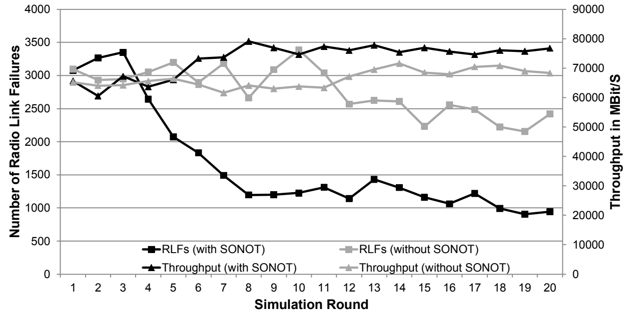

Figure 2. Overall performance with and without the SONOT function

V. Conclusion

This paper presented several issues that can arise if Self-Organizing Network (SON) functions encounter problems that hinder them to achieve their objectives. This leads to the idea of SON Operational Troubleshooting (SONOT), i.e., the detection of such situations and the reasoning for possible countermeasures to overcome the problem, particularly preempting the execution of that SON function and taking the appropriate further steps beyond its scope. A SONOT function is outlined that realizes the idea through a SON function monitoring component and an analysis and remedy component. In a simulated problem scenario, it has been shown that the SONOT function can improve network performance significantly.

In the future, it is planed to extend the problem analysis and evaluation in order to find further situations in which the SONOT function can provide advantages. Furthermore, new methods for the detection of problem situations are required to extend the applicability of the concept.

Acknowledgment

The authors would like to thank Péter Szilágyi (Nokia Solutions and Networks Research, Budapest, Hungary) for his contributions to the development of the initial concept.

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] L. C. Schmelz, M. Amirijoo, A. Eisenblaetter *et al.*, "A Coordination Framework for Self-Organisation in LTE Networks," in *Proc. 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, Dublin, Ireland, May 2011, pp. 193–200.

[3] R. Romeikat, H. Sanneck, and T. Bandh, "Efficient, Dynamic Coordination of Request Batches in C-SON Systems," in *Proc. IEEE 77th Vehicular Technology Conference (VTC Spring 2013)*, Dresden, Germany, Jun. 2013, pp. 1–6.

[4] K. Tsagkaris, A. Galani, N. Koutsouris *et al.*, "Unified Management Framework (UMF) Specifications Release 3," UniverSelf Project, Deliverable D2.4, Nov. 2013.

[5] S. Ben Jemaa, C. Frenzel, D. Götz *et al.*, "Integrated SON Management - Requirements and Basic Concepts," SEMAFOUR Project, Deliverable D5.1, Dec. 2013.

[6] T. Bandh and R. Romeikat, "An Integrated SON Experimental System for Self-Optimization and SON Coordination," in *Proc. 2nd International Workshop on Self-Organizing Networks (IWSON 2012)*, Paris, France, Aug. 2012.