An Experimental System for SON Verification

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

***Abstract*—In a mobile Self-Organizing Network (SON) the usage of a SON coordinator is necessary to preclude the execution of conflicting SON function instances. Typically, such a coordinator is accountable for conflict prevention and resolution and does not take into consideration that the activity of function instances may induce an undesired network behavior. In this paper, we propose an experimental system for the verification of Configuration Management (CM) changes induced by the activity of SON function instances. The main part of this system is the SON verification function which is triggered when CM change requests get acknowledged by the SON coordinator. It analyses the resulting Performance Management (PM) data and in case it detects an undesired network behavior it sends a request to the coordinator to undo the responsible changes.**

I. Introduction

In a Self-Organizing Network (SON), several SON function instances may get active at the same time and try to perform conflicting Configuration Management (CM) changes. In such a case, a SON coordinator [1] is typically employed to reject the requests which would engage in conflicts and allow those which would guarantee a flawless network operation. Usually, the decisions of a coordinator are based on rules used to anticipate and avoid known conflicts between SON function instances. In literature, such an approach is typically referred to as *pre-action SON coordination*.

Unfortunately, approved CM changes may not necessarily lead to the performance targeted by the corresponding SON functions or by the operator himself. A SON coordinator does not consider the fact that the actions of a large number of deployed SON functions may cause an undesired network behavior [2]. Usually, a SON coordinator is designed for conflict prevention and resolution between SON function instances, e.g., taking care that two instances are not modifying the same CM parameter on the same cell. However, it does not observe whether such changes have had a negative impact on the network performance.

An anomaly detection and diagnosis mechanism [3] addresses such kind of problems. The purpose of such a mechanism is to detect an undesired network behavior, perform root cause analysis and provide a corrective action. In order to assess the impact of a set of configuration changes, an anomaly detector learns the faultless behavior of the network and later uses it as a basis of comparison to identify significant deviations from the usual behavior. To provide the corrective action, the diagnosis part typically learns the impact of faults on different performance indicators.

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

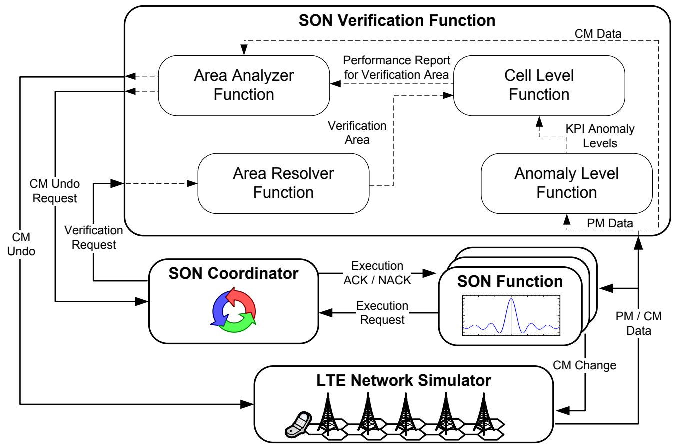

Figure 1. SON Simulation System (S3) Components

Motivated by the ideas of anomaly detection and diagnosis we have developed the SON verification function. The purpose of our function is to assess the impact of SON-induced CM changes and provide the corrective action in case they have caused an undesired or unusual network behavior. When action requests get acknowledged, the SON coordinator forwards the task of observing the performance impact of those changes to our verification function. The coordinator sends for this purpose a verification request message identifying the cells that have been reconfigured by a SON function instance. Moreover, the coordinator informs our function about the area influenced by that reconfiguration as well as the time the change has an effect on other running function instances. Based on this information, our function determines where to look for an anomaly in order to find the changes responsible for an undesired behavior to occur. The approach our function is following we also classify as *post-action verification*.

II. SON Simulation System

To evaluate the behavior of the our concept, we have developed the SON Simulation System (S3), as outlined in Figure 1. It consists of four components as described below.

1) LTE Network Simulator: We employ an Long Term Evolution (LTE) radio network simulator that allows us to reconfigure 12 Evolved NodeBs (eNBs) at run-time. The simulated network area equals to 50 km^2. The simulator periodically computes and exports Performance Management (PM) data for every cell in so-called simulation rounds, each corresponding to approximately 100 minutes in real time. At the beginning of a round, the simulator configures the network as defined by the CM parameter setup. During a round, 1500 uniformly distributed mobile users follow a random walk mobility model (speed 6 km/h) and use the network. The constant bit rate requirement of each user is set to 256 kpbs. Furthermore, the simulator makes use of the path loss radio propagation model. At the end of a round, the simulator exports PM data for each cell which is forwarded to the SON Function Engine (SFE) and our verification function as well.

2) SON Function Engine: The SFE is a runtime environment for SON functions which handles their communication and configuration. Every time the LTE network simulator completes a round, the SFE triggers the monitoring phase of all SON functions. Should a CM change request be generated, it is immediately forwarded to the SON coordinator. Based on the coordinator's decision, the SFE deploys the requested CM parameter changes to the simulator. In total, the SFE includes three optimization functions: the Mobility Robustness Optimization (MRO), Remote Electrical Tilt (RET), and Transmission Power (TXP) function, as defined in [4]. Furthermore, an instance of all three functions is running for each cell in the network.

3) SON Coordinator: The SON coordinator performs pre-action coordination by employing the batch coordination concept with dynamic priorities, as defined in [1]. The concept is designed for batch processing of SON function requests. More precisely, every SON function instance has an assigned bucket and dynamic priority. The bucket initially contains a number of tokens which are reduced every time a request by the function instance is accepted and increased if a request is rejected. If the bucket gets empty, the priority is set to minimum. The priority can be increased again if a request is rejected. The coordinator collects all requests for a simulation round, determines the conflicts and sends an Acknowledgment (Ack) for the requests with the highest priority and a Non-Acknowledgment (Nack) for the others.

4) SON Verification Function: Our post-action verification approach involves a tight integration with SON coordination. The SON verification function we propose analyzes the network performance for acknowledged action request of SON function instances. In case the activity of a given instance causes an undesired network behavior, our function requests permission to undo the responsible CM changes from the SON coordinator for the affected area. To achieve its task, though, the SON verification function function makes use of four helper functions: (1) an anomaly level, (2) a cell level, (3) an area resolver, and (4) an area analyzer function. The anomaly level function allows us to differentiate between normal and subnormal cell Key Performance Indicator (KPI) values. The output is a *KPI anomaly level* which depicts the deviation of a KPI from its expectation. The cell level function creates an overall performance metric of individual cells. The output is the sum of the weighted anomaly levels which corresponds to the *cell level*. The area resolver function defines the spatial scope we are going to observe for anomalies. It consists of a set Σ that includes the cells that have been reconfigured by a SON function instance and a set of cells Ω that have been possibly influenced by that reconfiguration process. We call Σ the *CM change base* and Ω the *CM change extension area*. Altogether (i.e., $\Sigma \cup \Omega$) they compose the *verification area* V which is the spatial scope we observe for anomaly.

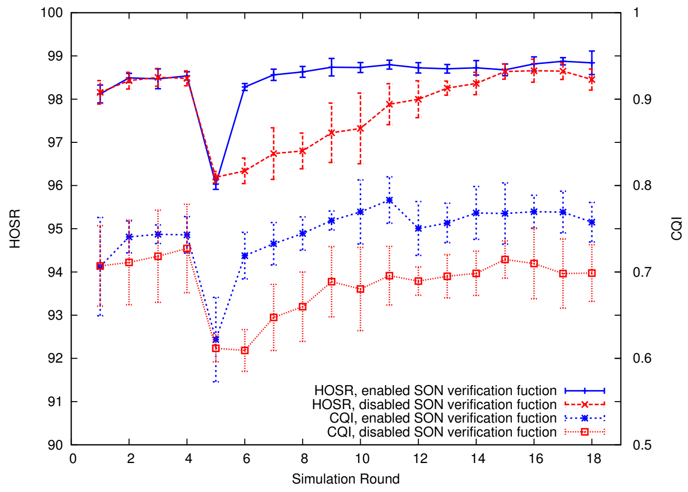

Figure 2. Impact on the HOSR and CQI

The area analyzer function determines whether the cells within that scope are showing subnormal behavior and triggers a CM undo for the CM change base if such a state can be confirmed.

III. Evaluation

Let us assume the following scenario: should TXP change the transmission power in such a way that it induces a negative impact on the KPIs monitored by the RET function, the latter one may try to provide a corrective action which may not necessarily lead to an improvement. Moreover, since we employ a pre-action coordination mechanism that dynamically adapts the priority of the running function instances, a high prioritized RET function may even prevent a low prioritized TXP from undoing its change. Figure 2 outlines the result of the Handover Success Rate (HOSR) and Channel Quality Indicator (CQI) captured by five cells in such a situation. Note that the latter one is computed as the weighted harmonic mean of the CQI channel efficiency. Up to round 5, the observed set of five cells are showing a normal and usual behavior. At the time where the TXP function makes a wrong decision by decreasing the transmission power of just two cells, all five cells begin to experience an anomalous behavior. At this point, the advantage of employing our function can be seen. Instead of letting the functions slowly provide a solution, i.e., let RET and TXP adapt the coverage and the let MRO adjust the Handover (HO) parameters, our function triggers a CM undo which returns the set of cells to the normal state experienced before.

References

[1] 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.

[2] T. Kürner, M. Amirijoo, I. Balan, H. van den Berg, A. Eisenblätter *et al.*, "Final Report on Self-Organisation and its Implications in Wireless Access Networks," Self-optimisation and self-configuration in wireless networks (SOCRATES), Deliverable D5.9, Jan. 2010.

[3] 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.

[4] 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.