Network Element Stability Aware Method for Verifying Configuration Changes in Mobile Communication Networks

Janne Ali-Tolppa, Tsvetko Tsvetkov

Network Element Stability Aware Method for Verifying Configuration Changes in Mobile Communication Networks

Janne Ali-Tolppa[1] and Tsvetko Tsvetkov[2]

[1] Nokia Bell Labs, Munich, Germany
janne.ali-tolppa@nokia.com
[2] Department of Computer Science, Technische Universität München
tsvetko.tsvetkov@in.tum.de

Abstract. Automatic Configuration Management (CM) parameter change assessment, the so-called Self-Organizing Network (SON) verification, is an important enabler for stable and high-quality modern SONs. However, it also presents a new set of challenges. While improving network stability and resolving unexpected conflicts caused by parallel configuration changes, the SON verification can also make the network optimization less dynamic. This flexibility can be desirable, especially when the network is in an unstable state. One such example is Network Element (NE) commissioning, after which it may be preferable for the SON functions to explore the configuration space more freely in order to find the optimal configuration. On the other hand, at some point in time, the NE has to converge to a stable configuration.
To address these challenges, we introduce in this paper the concept of Network Element Virtual Temperature (NEVT), which indicates the state of stability of a NE, and propose how it can be utilized to optimize the verification process. This approach is evaluated in a simulated environment and compared to other verification mechanisms. The results show that the proposed method allows the network to better react to changes without sacrificing on its stability.

1 Introduction

The Self-Organizing Network (SON) concept is a key enabler for managing the complex modern networks. It covers the tasks of self-configuration, self-optimization and self-healing [8]. A SON-enabled network is managed by a set of autonomous SON functions performing specific network management tasks. The functions are designed as control loops, which monitor Performance Management (PM) and Fault Management (FM) data, and based on their goals configure the Configuration Management (CM) parameters. For example, the Coverage and Capacity Optimization (CCO) function has been developed to optimize the coverage within a cell by changing the antenna tilt or the transmission power.

However, in a complex system the functions may have unexpected side-effects. Especially, when several SON function instances and human operators are operating independently in parallel. For this reason, policy-based pre-action coordination is applied to avoid known conflicts between SON function instances [3,12]. Such conflicts may include, for example, two SON function instances changing the same CM parameters (lost update problem) or one function influencing the input data of another (race condition), causing it to base its decisions on invalid data. However, pre-action coordination can only prevent potential conflicts that are known beforehand. As a consequence, the concept of SON verification has been developed [5, 7, 17]. It automatically verifies the impact of configuration changes and, in case a degradation is detected, returns the network to the last known stable configuration by performing a rollback of the changes.

The verification process introduces its own challenges too, however. While it makes the network more stable by quickly rectifying degradations, it can also make the system less dynamic by preventing optimization paths that may lead to a minor transient decrease in performance. A related concrete problem is the configuration of the verification observation window length. If it is too short, the verification mechanism may try to rollback short, transient performance degradations, and therefore generate false positive undo requests. On the other hand, if the observation window is configured long, it makes the verification cycle slower, increases the number of parallel and potentially conflicting verification operations and makes determining the CM change causing a degradation more difficult.

In this paper, we propose an approach that mitigates these issues. We introduce the concept of Network Element Virtual Temperature (NEVT), which indicates the state of stability of a Network Element (NE). The NEs with high NEVT are considered unstable and therefore, for example, more likely to accept optimization steps that lead to minor transient degradations. Similarly, as the NEs becomes more stable, i.e. there are no further changes in the NE as the time passes, NEVT "cools down" and the NE becomes less likely to accept major changes. Utilizing the NEVT, the verification process can give the SON functions the necessary freedom when needed, but ensure that the optimization will converge to a stable, well-performing configuration.

2 The Verification Process

The purpose of the SON verification is to automatically verify CM changes done in the network, either by SON functions or human operators, and in case a degradation is detected, return the network back to a stable configuration by performing a rollback, also known as an undo operation. The automatic verification process operates in three phases: (1) it divides the network into verification areas according to the CM changes, (2) observes each area during the so-called *observation window* and (3) in case a degradation is detected, marks the changes for an undo that are most likely responsible for it.

Composition of verification areas A verification area consists of the reconfigured cell, also called the *target cell*, and a set of cells impacted by the CM change(s), also referred to as the *target extension set* [16]. The extension set is often based on the neighbor relations of the target cell. Furthermore, if we have SON function activities in the network, we may define the verification area as the combined impact areas [2] of the SON functions that are verified in a given verification operation. We may also enlarge a verification area based on its location [6], e.g., more cells are added if they are part of known trouble spots or dense traffic.

Observation window After the verification area has been determined, the verification process monitors the performance of the area after the CM changes have been applied. It does this by monitoring the set of Key Performance Indicators (KPIs) selected to be used in verification. The assessment of the KPIs is usually done by profiling [9] the network behavior, which requires analyzing the network performance indicators and specifying how the network should typically behave and how it has behaved before and after the changes.

Generating CM undo requests At the end of the observation window the verification process calculates a *verification score* for the verified changes. This score can be compared to an *acceptance threshold*. If the score is higher than the threshold, the changes are accepted, otherwise the process attempts to return the degraded verification areas to a known previous stable configuration by performing an automatic CM undo [17]. It determines, which CM changes have an overlapping impact areas with the verification area, and can thus have contributed to the degradation, and creates an undo request for them. Several methods have been developed to optimize the selection of undo areas and requests [15]

3 Challenges

Many SON functions require not only one step, but several, during which they observe whether they have moved closer towards achieving their goal. In the process, the functions may induce a transient performance decrease in the network. For instance, the CCO function monitors the impact of its last deployed antenna tilt or transmission power changes, and adjusts them if required [8]. The same applies also to the combined, and often unpredicted, effect of several independent SON function instances running in parallel. They might introduce a transient degradation, to which the functions react and adapt to a more optimal configuration.

This poses a problem to the SON verification. Namely, how long should the verification observation window be? If the observation window is made too short, there is a risk that the true impact of the CM changes is not yet visible in the network. Changes are rolled back, although they are just transient and the SON

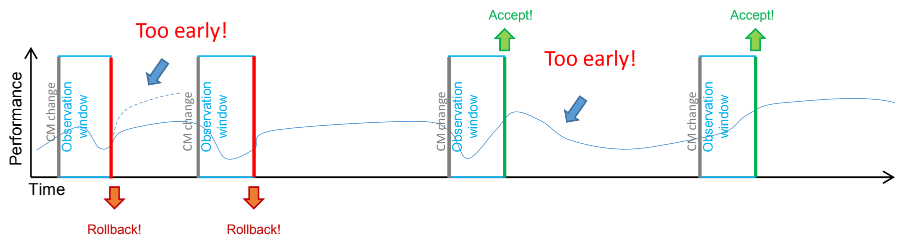


Fig. 1: The problem of choosing the correct observation window length

functions would adapt to the situation in the upcoming rounds. In the same manner, the opposite may happen and changes are accepted before the negative impacts show in the monitored KPIs. Figure 1 depicts this problem.

On the other hand, a long observation window is not desirable, because it means a longer verification cycle, which can block other SON functions from executing. Also, the longer observation window, the more difficult it becomes to find out which changes have actually caused a degradation, especially if there are several parallel changes made by more than one SON function instances. Yet one more problem with a long observation window is that as it makes the verification process longer, it increases the number of parallel verification operations, which can lead to conflicts between them [15]. This can be problematic, especially in more complex networks.

The CM scoring method proposed in [10] answers these questions partly. It introduces a mechanism that assesses the changes made by SON functions over a certain period of time and categorizes their behavior by assigning them to a certain zone. For example, changes leading to a drop in performance are assigned to a so-called red zone. The difficulty, however, is choosing the correct length of the required time period. Once again we would like to keep the observation window as short as possible, but if it is too short, the scoring mechanism does not work anymore.

4 Concept

To address the problems stated in Section 3, we propose a method to incorporate NE stability awareness into SON functions, especially in the SON verification. We introduce the concept of Network Element Virtual Temperature (NEVT), which indicates the state of stability of a NE. In the following, we are going to explain the concept in detail and describe how the NEVT can be used to improve SON verification.

4.1 Network Element Virtual Temperature

The Network Element Virtual Temperature (NEVT), similar to the temperature concept in simulated annealing [14], shall be understood as an indicator of the

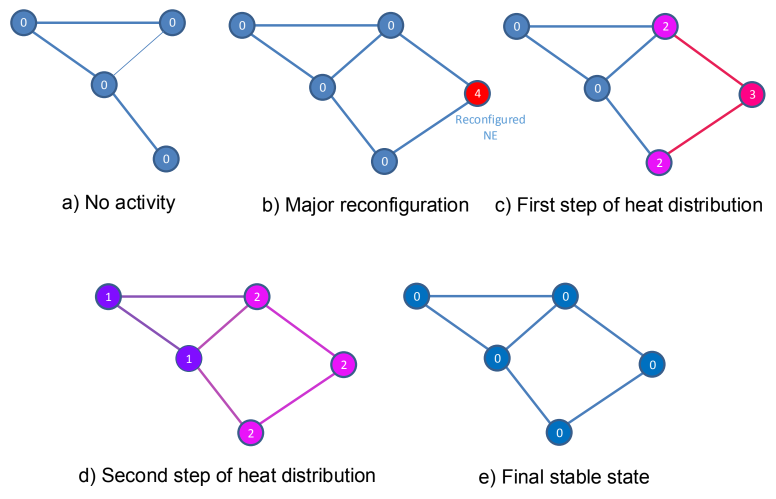


Fig. 2: Example of distributing NEVT ϵ [0; 4] in the network

stability of a NE, in particular of a cell. A NE becomes "hotter" after operations such as commissioning, SON function activity, or manual reconfiguration. Another example is when NE software is upgraded, which inserts new heat into the NE to indicate the reduced stability of its state. The resulting NEVT value depends on the event that has occurred in the network, i.e., a certain type of changes results in a much higher increase of the temperature than others. However, if there are no changes in the NE, it will cool down over time according to its *cooling factor* to indicate increased stability. A less stable NE with a high NEVT value is more likely to accept intermediate optimization steps that lead to somewhat reduced performance, in order to reach improved performance in the future. In contrast, a NE with a low NEVT value should retain its stability and is less likely to accept transient degraded optimization steps.

A change in a cell may not only affect the cell itself, but can also have an impact on its neighboring cells. For example, if a CCO function tilts an antenna up, this may not only influence the cell, where the tilt change was made, but also the neighboring cells may start experiencing higher interference. Therefore, the NEVT is transfered to the neighboring cells, analogous to heat conduction in physics. The amount of heat conducted to the neighboring cells depends on the selected *conduction factor*.

These three concepts, i.e., heat introduction, cooling factor and conduction factor, define the temperature of each NE over time. In Figure 2 we have outlined an example, where a new NE is introduced into an existing network resulting in introduction of a new, "hot" NE, heat conduction and subsequent cooling down.

4.2 Application of NEVT in SON verification

The SON verification function can use the NE stability indication, the NEVT, in several different ways:

- Choose to completely omit hot verification areas
- Select the observation window length based on the NEVT
- Accept changes causing minor degradations if the NE is hot

Firstly, if the verification function is heavily loaded, i.e., there are several, perhaps overlapping verification operations running, it may opt to completely omit hot verification areas to reduce the number of potential verification conflicts. Since the NEs in the verification area are unstable, also the verification decision would be unreliable and it might be better to let the SON functionality to continue optimizing the area.

Secondly, SON verification may adapt the observation window length according to the NEVT. For more unstable NEs, a longer observation window is used, in order to let them better stabilize after each change before making a verification decision. Likewise, for more stable NEs, a shorter window can be used.

Thirdly, the verification may accept changes leading to minor degradations, in case the verification area has a high NEVT value. The verification process accepts all changes that have a verification score higher than the set acceptance threshold, but it may in addition accept changes with a score below the threshold with a certain probability depending on the verification score and the NEVT. This allows SON functions to explore the configuration space more freely, without SON verification interrupting the process with a CM undo, and thus "escaping" potential local optima. On the other hand, the likelihood of SON verification accepting such intermediate changes should become lower, as the verification area cools down. Analogous to simulated annealing, this ensures that the NE configuration will stabilize and converge to at least a local optimum solution. Contrary to offline optimization solutions based on simulated annealing [4], however, SON verification can always accept only minor degradations, since it operates in a live network, where sufficient Quality of Service (QoS) must always be ensured. In this paper we focus on this third alternative.

4.3 Components

The architecture consists of two components: (1) a NEVT aggregator and (2) a NEVT parser, as shown in Figure 3.

The NEVT aggregator consists of three sub-components. The first one is an event monitor that is responsible for monitoring the events occurring on the NE. Typical events are the change of CM parameters, software upgrades, etc. The second component, the NEVT calculator, computes the temperature. The third component is the NEVT distributor. It is the connection point, used to exchange NEVT values between the NEVT aggregators of neighboring NEs. Its task is to generate (send) and receive NEVT forward messages.

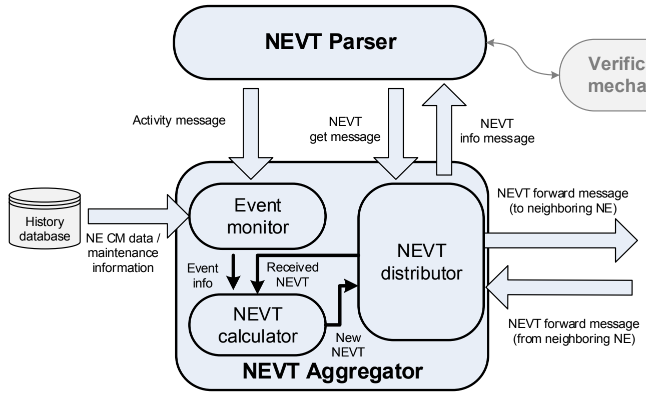


Fig. 3: Input, output, and components of the concept

Upon receiving a NEVT forward message, a NEVT distributor delegates the received information to the NEVT calculator. It uses the information to update the temperature of the NE.

The second main component, the NEVT parser, is the connection point to the verification mechanism [16] or any other network anomaly analyzer. First of all, it allows the verification mechanism to read the current NEVT. It can be also provide aggregated NEVT values for the current verification area.

Secondly, it gives a verification mechanism the ability to inform the NEVT aggregator about its planned undo activities. For example, if it requires a certain period of time to resolve verification collisions, an aggregator may artificially keep the temperature at a high level, indicating that the NEs have not stabilized.

4.4 Message flow

An example of the message flow in a heterogeneous mobile network is given in Figure 4.

1. A SON function triggers major CM changes within the fifth generation of mobile communications (5G) network.
2. The responsible NEVT aggregator computes the temperature.
3. The same NEVT aggregator distributes the computed values to the neighboring NEs, including those being part of the fourth generation (4G) network, by generating NEVT forward messages. The corresponding NEVT aggregators receive the message and adapt the temperature respectively. The receiving NEVT aggregators dynamically configure the temperature, based on the temperature of the sending entity, as well as the conductivity factor.
4. When the verification mechanism is triggered, it requests the temperature of the verification area from the NEVT parser.
5. The parser collects the NEVTs of the NEs by issuing a NEVT get message. The corresponding aggregators reply with an NEVT info message.

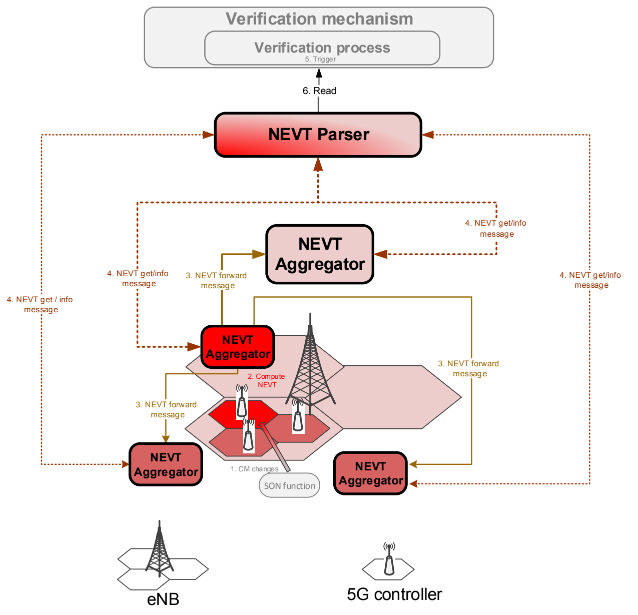


Fig. 4: Message flow

6. Based on the gathered information the verification mechanism is able to assess the change and decide whether or not it can be accepted.

4.5 Design of NEVT Aggregator

Let us define NEVT as $\chi \in [0, 100]$ and that the NEVT of a cell c at Granularity Period (GP) t is given by the function $\tau_t(c)$. An operation performed on a cell c, for example a major re-configuration, can increase the χ_c to indicate a reduced state of stability, i.e., $\tau_{t+1}(c) = \tau_t(c) + \delta$, where δ is the added heat. Note that δ must be scaled so that $\tau_{t+1}(c) \in [0, 100]$.

NEVT is also *conducted* to the neighboring cells. To simplify the model and to reduce the need for signaling, the NEVT conduction is done only when NEVT is added in the system. Let us denote the set of all cells as $\varSigma$. When δ of NEVT is added to a cell $c \in \varSigma$, the NEVT conducted to each neighbor of c is given by

$$\tau_{t+1}(c_n) = \tau_t(c_n) + K(\tau_t(c) + \delta - \tau_t(c_n)), \forall c_n \in N \tag{1}$$

, where $N \subset \varSigma$ is the set of cells neighboring c and $K \in [0, 1]$ is the NEVT conductivity factor.

Additionally, on each GP each cell cools down according to the *cooling factor*. That is, for each cell $c \in \varSigma$, $\tau_{t+1}(c) = F\tau_t(c)$, where $F \in [0, 1]$ is the cooling factor.

4.6 Design of NEVT Parser

For implementing the improved verification mechanism, we need the NEVT Parser to calculate the NEVT of a verification area. For this purpose, let us denote the set of all cells as Σ and the set of all verification areas as V. Furthermore, let us define an extraction function f_e that returns the cells of a verification area, i.e., $f_e\colon V \to \mathcal{P}(\Sigma) \setminus \{\varnothing\}$. Now the aggregated NEVT of a verification area $v \in V$ at a time t can be defined as

$$T(v) = \frac{1}{|f_e(v)|} \sum_{i=1}^{|f_e(v)|} \tau_t(c)_i \tag{2}$$

, where $\tau_t(c)$ is the NEVT of a cell $c \in f_e(v)$ as given by the NEVT Aggregator.

4.7 Enhanced verification process

In order to compute an anomaly score for verification, we define for each cell an anomaly vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ that is an element of $\mathbb{R}^n$. Each element $a_k \in \mathbb{R}, k \in [1, n]$ represents the deviation of a KPI from the expected value, also known as the *KPI anomaly level*. The value of n equals $|K|$, where K is the set of KPIs that are selected to determine the cell performance.

To calculate the anomaly vector, the expected behavior is learned in a training phase, during which we collect samples $p_1 \ldots p_t$ for each given KPI, where t marks a training period. Then, we measure the current value of the KPI, denoted as p_{t+1}. The collected data, i.e., $p_1 \ldots p_t, p_{t+1}$, is normalized by computing the *z-score* of each data point. The KPI anomaly level is the z-score of p_{t+1}. The z-score itself represents the distance between the given point and the sample mean in units of the standard deviation. It is negative, when the raw score is below the mean, and positive when above. In addition, the anomaly level is normalized so that a positive level value means better than average performance and negative level worse than average. For example, for a success KPI, such as Handover Success Rate (HOSR), we can use the z-score as such. However, for a failure KPI, such as Radio Link Failure (RLF), we use the negative z-score.

The verification is done for a verification area consisting of one or more cells. In order to calculate the verification score for the area, we collect the anomaly vectors of each cell in the verification area into a verification area anomaly matrix $\mathbf{A}$, where each column represents a cell and each row a KPI. Next, we aggregate all elements a_{jk} of the matrix $\mathbf{A}$ into one single z-score that represents the overall verification area performance. The aggregation function $\varphi(\mathbf{A})$ calculates it by taking the arithmetic average of all a_{jk}, as defined in Equation 3.

$$\varphi(\mathbf{A}) = \frac{1}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} a_{jk} \tag{3}$$

, where m is the number of cells in the verification area.

Now, following the proposal in [13] for simulated annealing, we can define an *acceptance function* $\psi(v, \mathbf{A}) \in [0, 1]$ that defines the probability of the verification process accepting a change, which is resulting in an anomaly matrix $\mathbf{A}$ in verification area $v \in V$, as

$$\psi(v, \mathbf{A}) = \begin{cases} 1, & \text{if } \varphi(\mathbf{A}) \geq \epsilon \\ 0, & \text{if } \varphi(\mathbf{A}) < \epsilon \textit{ and } T(v) = 0 \\ e^{\dfrac{-c(\epsilon - \varphi(\mathbf{A}))}{T(v)}}, & \text{otherwise} \end{cases} \quad (4)$$

, where ϵ is the acceptance threshold and c is a scaling factor.

For example, by setting $\epsilon = 0$, all changes having a positive verification score will always be accepted and changes with negative score will have a probability of acceptance depending on the verification score and the NEVT of the verification area.

5 Evaluation

In this section we present the results of our experimental case study with the presented NE stability aware verification method. We also give an overview of the used simulation system.

5.1 Environment

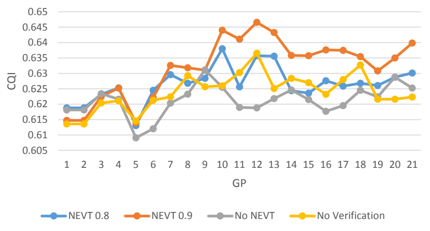
To evaluate the behavior of SON coordination and verification concepts, we have developed the SON Simulation System (S3), as presented in [15].

The simulator models a Long Term Evolution (LTE) macro cell network based on simulated mobile users and radio propagation models [11]. The simulated network consists of 32 cells spread over an area of 50 km^2. Furthermore, 1500 uniformly distributed mobile users follow a random walk mobility model and use the network. The PM data calculated based on the radio simulation is collected in rounds corresponding to approximately 100 minutes in real time, i.e. GPs, and forwarded to the SON functions.

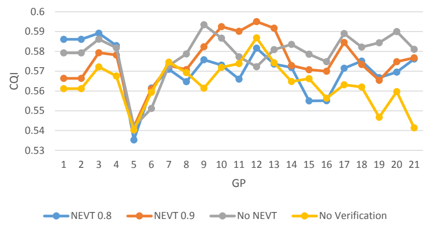
For all simulation test runs, we employ two optimization functions: Mobility Robustness Optimization (MRO) and CCO, and the verification function. The CCO function adapts the antenna tilt and transmission power and the MRO changes the Cell Individual Offset (CIO) parameters. An instance of each of the two functions is running for each cell in the network.

5.2 Setup and scenario

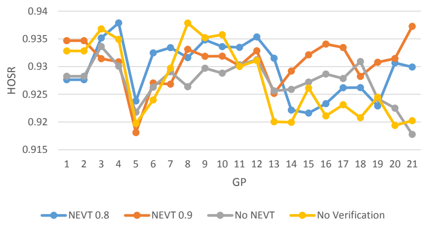
Each simulation round consists of 21 GPs. In the tested scenario, a disturbance is introduced after the 4th GP by turning one of the cells off. CCO and MRO adapt to this change and the verification function keeps monitoring their changes.



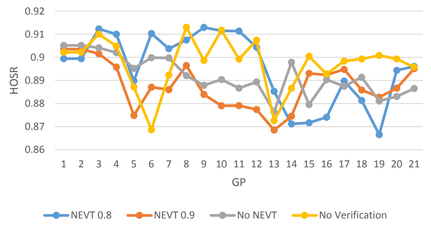
(a) CQI of the verification area



(b) CQI of the worst performing cell



(c) HOSR of the verification area



(d) HOSR of the worst performing cell

Fig. 5: Comparison of different verification strategies

During the 13th GP the cell is turned on again. When the cell is turned on and off, the NEVT calculator adds a fixed amount of 80 units of NEVT in the reconfigured cell.

The scenario is repeated by using four different configurations:

1. verification with NEVT, cooling factor 0.8
2. verification with NEVT, cooling factor 0.9
3. verification without NEVT, i.e., rejecting all changes with verification score below the acceptance threshold
4. no verification

Furthermore, NEVT conductivity factor is set to 0.8 and the verification acceptance function scaling factor to equal the maximum NEVT value of 100. Verification area is the target cell and its neighboring cells. A short observation window of only one GP was selected. Each scenario in each configuration is simulated 5 times and values for each GP averaged over the simulations. The followed KPIs are the Channel Quality Indicator (CQI) and the Handover Success Rate (HOSR), which are also used in the verification process. The CQI is computed as the harmonic mean over the channel efficiency [1].

5.3 Simulation Results

The simulation results show that the SON verification with the NEVT is able to adapt better to the disturbances while ensuring the stability of the system. Figure 5 shows the results. It depicts the CQI and HOSR values of the cell that was shutdown and its neighbors. Both the average value of the KPI of every cell in the whole verification area and the value of the worst performing cell in the area are shown. Note that all the values are averages of the five simulation rounds.

For the most important KPI, the CQI, we can see that using verification without NEVT slows down the reaction to the degradation introduced during GP 5 due to higher number of undo operations. Moreover, if we look at the CQI of the worst performing cell, we can see that without verification it ends up in a sub-optimal state.

If we look at the verification with NEVT, we can see that the SON functions are able to react to the disturbance fast and at the same time the degradation of the worst performing cell is avoided. With lower cooling factor of 0.9, in particular, the network stabilizes in clearly better average CQI values than without verification or with verification without NEVT. As for the HOSR we see less differences, but the NEVT-based verification mechanism achieves slightly better results.

6 Conclusion

The Self-Organizing Network (SON) verification is an important enabler of modern SONs. While it improves the stability of the network, verification can also make it less flexible, which can be undesirable in certain circumstances. For example, after commissioning a new Network Element (NE), it can be preferable to let the SON functions configure the Configuration Management (CM) parameters more freely to better adapt to the environment.

In this paper, we proposed the concept of Network Element Virtual Temperature (NEVT), which indicates the state of stability of a NE. Major reconfigurations of a NE increase its NEVT to indicate reduced stability. Over time, if there are no further reconfigurations, the NEVT decreases according to the *cooling factor* as the NE becomes more stable. NEVT is *conducted* to the neighboring NEs to indicate that they should also adapt to the changes. Utilizing the concept of NEVT, we introduced a NE stability aware verification process. In the process, all CM changes that exceed a set threshold for the verification score are accepted. However, also CM changes that do not exceed the threshold have a certain probability of being accepted, depending on the verification score and the NEVT of the NE.

Our evaluation, based on radio environment and subscriber simulation, shows that the NEVT-based verification allows the network to react and adapt to major changes faster. At the same time it avoids the degradations that occurred, when no verification was used. Distribution of NEVT to neighboring NEs enables the whole subnetwork to react to changes in one NE and the "cooling down" of NEVT ensures that the network stabilizes to an optimal configuration.

Future work includes studying the application of NEVT to other Network Management (NM) use cases.

References

1. 3GPP: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures. Technical specification 36.213 v12.1.0, 3rd Generation Partnership Project (3GPP) (Mar 2014)
2. Bandh, T.: Coordination of autonomic function execution in Self-Organizing Networks. Phd thesis, Technische Universität München (Apr 2013)
3. Bandh, T., Romeikat, R., Sanneck, H.: Policy-based coordination and management of SON functions. In: 12th IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM 2011) Work. pp. 827–840. IEEE, Dublin, Ireland (May 2011)
4. Berger, S., Fehske, A., Zanier, P., Viering, I., Fettweis, G.: Comparing Online and Offline SON Solutions for Concurrent Capacity and Coverage Optimization. In: Proceedings of the Vehicular Technology Conference (VTC Fall). Vancouver, Canada (Sep 2014)
5. Ciocarlie, G., Connolly, C., Cheng, C.C., Lindqvist, U., Nováczki, S., et al.: Anomaly Detection and Diagnosis for Automatic Radio Network Verification. In: 6th International Conference on Mobile Networks and Management (MONAMI 2014). Würzburg, Germany (Sep 2014)
6. Ericsson: Transparent Network-Performance Verification For LTE Rollouts. White Paper, 284 23-3179 Uen (Sep 2012)
7. Gajic, B., Nováczki, S., Mwanje, S.: An Improved Anomaly Detection in Mobile Networks by Using Incremental Time-aware Clustering. In: IFIP/IEEE Workshop on Cognitive Network and Service Management (CogMan 2015). Ottawa, Canada (May 2015)
8. Hämäläinen, S., Sanneck, H., Sartori, C. (eds.): LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency. John Wiley & Sons, Chichester, UK (Dec 2011)
9. Nováczki, S.: An Improved Anomaly Detection and Diagnosis Framework for Mobile Network Operators. In: 9th International Conference on Design of Reliable Communication Networks (DRCN 2013) (Mar 2013)
10. Nováczki, S., Tsvetkov, T., Sanneck, H., Mwanje, S.: A Scoring Method for the Verification of Configuration Changes in Self-Organizing Networks. In: 7th International Conference on Mobile Networks and Management (MONAMI 2015). Santander, Spain (Sep 2015)
11. NSN: Self-Organizing Network (SON): Introducing the Nokia Siemens Networks SON Suite - an efficient, future-proof platform for SON. White Paper (Oct 2009)
12. R. Romeikat, H. Sanneck, T. Bandh: Efficient , Dynamic Coordination of Request Batches in C-SON Systems. In: In: IEEE Veh. Technol. Conf. (VTC Spring 2013), Dresden, Germany (Jun 2013)
13. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi: Optimization by Simulated Annealing. Science, New Series, Vol. 220 (May 1983)
14. Szu, H., Hartley, R.: Fast simulated annealing. In: Physics Letters A. vol. 122, pp. 157 – 162 (Jun 1987)
15. T. Tsvetkov, H. Sanneck and G. Carle: A Graph Coloring Approach for Scheduling Undo Actions in Self-Organizing Networks. In: IEEE IM, Ottawa, Canada (May 2015)
16. Tsvetkov, T., Frenzel, C., Sanneck, H., Carle, G.: A Constraint Optimization-Based Resolution of Verification Collisions in Self-Organizing Networks. In: IEEE Global Communications Conference (GlobeCom 2015). San Diego, CA, USA (Dec 2015)

17. Tsvetkov, T., Nováczki, S., Sanneck, H., Carle, G.: A Configuration Management Assessment Method for SON Verification. In: International Workshop on Self-Organizing Networks (IWSON 2014). Barcelona, Spain (Aug 2014)