On Design Principles for Self-Organizing Network Functions

Zwi Altman°, Mehdi Amirijoo*, Fredrik Gunnarsson*, Hendrik Hoffman[Δ], István Z. Kovács[+], Daniela Laselva[+], Bart Sas[X], Kathleen Spaey[X], Abdoulaye Tall°, Hans van den Berg[Φ], Kristina Zetterberg*

*Ericsson Research Linköping, Sweden

[X] iMinds, Antwerp, Belgium

°Orange Labs Issy-les-Moulineaux, France

[+]Nokia Aalborg, Denmark

[Φ]TNO, Delft, Netherlands

[Δ]TU Braunschweig Braunschweig, Germany

IWSON @ ISWCS, Barcelona, August 2014

Means to Manage Potential SON Conflicts

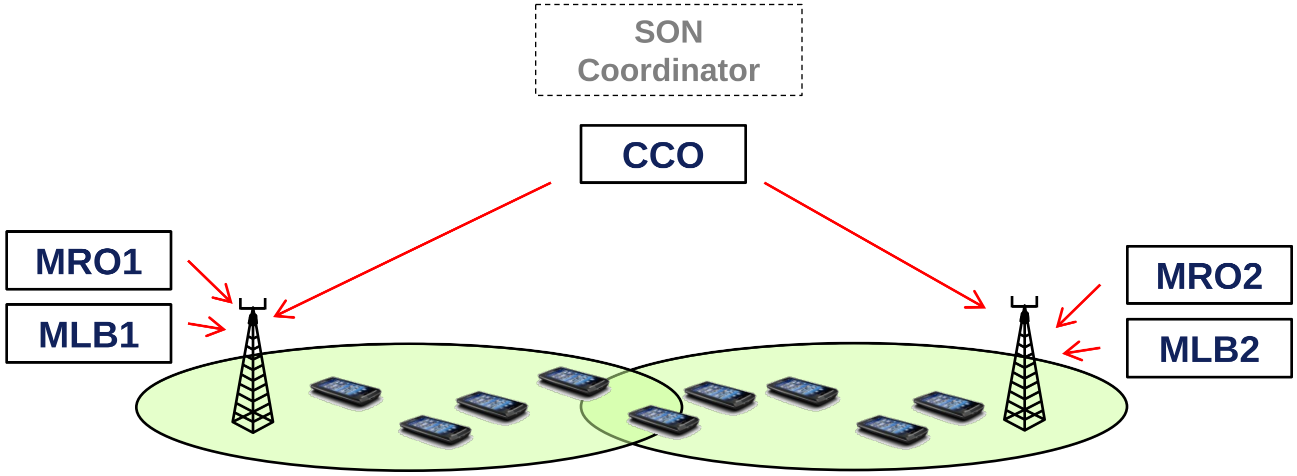
When introducing automation features such as **SON**, there can be concerns about the **overall behavior** such as **stability** of the system.

Previous discussions have proposed **SON coordination** mechanisms and **stabilizing feedback**.

This leads to two relevant questions:

- How shall a SON function be designed to be **prepared to be coordinated**?
- How can a SON function be designed to **reduce or even avoid coordination needs**?

- **Two cells**, each with an instance of decentralized Mobility Robustness Optimization (**MRO**) and Mobility Load Balancing (**MLB**), adjusting **handover triggering points**
- A **centralized CCO** algorithm, adjusting **antennas** and **transmission power**



- SON functions should be able to deal with the possible actions that are taken by the SON coordinator
  - Enabling/Disabling/Suspending SON Functions
  - Stopping/Suspending/Modifying SON Actions
  - Modifying Configuration Parameters

Design Principle 1.1: Updating Internal State

While they are disabled by the SON coordinator SON functions should keep updating their internal state based on changes in the network such that once they are enabled again they are able to swiftly operate optimally again.

Example:
The SON coordinator may disable MRO and MLB while CCO is updating parameters in the cell

MRO and MLB needs to monitor KPIs while being disabled.

Stopping/Suspending/Modifying SON Actions

Design Principle 1.2: Monitoring Output

SON functions should monitor the values of the controlled network parameters and take the current values into account when evaluating the system performance and taking decisions.

Example:

The SON coordinator may change the proposed parameters by MRO or MLB and instead use what CCO proposed.

MRO and MLB needs to monitor which parameters that actually are used.

Design Principle 1.3: Adaptability of Goals

SON functions should be designed such that they are able to deal with configuration parameter changes and can quickly operate optimally again after a configuration parameter change.

Example:

The SON coordinator or CCO may alter the target requirement of MRO or MLB, and they both need to adapt quickly.

If the handover trigger point change considerably, then MRO could change internal parameters significantly to quickly adapt.

- Aims at designing the SON function such that undesired interactions between SON functions of different types are avoided to as large extent as possible already in the design phase.

A Simple Linear Model

- Assume that the entire system of MRO and MLB in cells 1 and 2, and centralized CCO can be linearized, with parameters

  $x = [x_{MRO1}, x_{MRO2}, x_{MLB1}, x_{MLB1}, x_{CCO}]^T$

- $\dot{\boldsymbol{x}} = A\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{y}^t - \boldsymbol{y})$

- $$A = \begin{bmatrix} A_{MRO_1MRO_1} & A_{MRO_1MRO_2} & A_{MRO_1MLB_1} & A_{MRO_1MLB_2} & A_{MRO_1CCO} \\ A_{MRO_2MRO_1} & A_{MRO_2MRO_2} & A_{MRO_2MLB_1} & A_{MRO_2MLB_2} & A_{MRO_2CCO} \\ A_{MLB_1MRO_1} & A_{MLB_1MRO_2} & A_{MLB_1MLB_1} & A_{MLB_1MLB_2} & A_{MLB_1CCO} \\ A_{MLB_2MRO_1} & A_{MLB_2MRO_2} & A_{MLB_2MLB_1} & A_{MLB_2MLB_2} & A_{MLB_2CCO} \\ A_{CCOMRO_1} & A_{CCOMRO_2} & A_{CCOMLB_1} & A_{CCOMLB_2} & A_{CCOCCO} \end{bmatrix}$$

- Assume five measurements, $y = [y_1, y_2, y_3, y_4, y_5]^T$

- $$B = \begin{bmatrix} B_{MRO_1M_1} & B_{MRO_1M_2} & B_{MRO_1M_3} & B_{MRO_1M_4} & B_{MRO_1M_5} \\ B_{MRO_2M_1} & B_{MRO_2M_2} & B_{MRO_2M_3} & B_{MRO_2M_4} & B_{MRO_2M_5} \\ B_{MLB_1M_1} & B_{MLB_1M_2} & B_{MLB_1M_3} & B_{MLB_1M_4} & B_{MLB_1M_5} \\ B_{MLB_2M_1} & B_{MLB_2M_2} & B_{MLB_2M_3} & B_{MLB_2M_4} & B_{MLB_2M_5} \\ B_{CCOM_1} & B_{CCOM_2} & B_{CCOM_3} & B_{CCOM_4} & B_{CCOM_5} \end{bmatrix}$$

Design Principle 2.1: Separation of Control

Each SON function i *controls only its associated parameter x_i, and any SON function of a different type cannot control that parameter x_i.*

Example:

MRO and MLB both control the HO triggering point. Separation of concern can be implemented by separating the mobiles into two groups:

1. Mobiles subject to regular mobility
2. Mobiles that will be enforced to change cell due to load balancing reasons

Mobiles in each of the groups are configured with separate parameters and are handled by the two different SON functions.

Design Principle 2.2: Separation of Concern
Each SON function i *uses only its specific measurement y_i, and any SON function of a different type cannot use that measurement y_i. Furthermore, the measurements are made independent of each other.*

Example:
MRO and MLB both uses HO performance measurements such as handover failure and radio link failure. triggering point. Separation of concern can be implemented by separating statistics per groups:

1. The statistics S_1 gathered from mobiles in group 1, subject to regular mobility
2. The statistics S_2 gathered from mobiles in group 2, subject to load balancing

MRO will tune its parameter based on statistics in S_1, while MLB will tune its parameter based on statistics in S_2.

Separation of Time Scales

- Not always possible to avoid SON functions tuning the same parameter and/or using correlated measurements.
- The SON functions can still be separated if operating on very different time scales, e.g. a slow centralized SON function and a much faster decentralized SON function.

Design Principle 2.3: Separation of Time Scales

Two SON functions operating at different time scales are separated by time if the actions by the faster function can be assumed to have converged well within the update period of the slower one, and that measurements considered by the slower are averaged enough so that there is negligible correlation between the faster variations of the measurements.

Example:

Both MLB are CCO adjusts cell borders and uses cell load, but MLB is both operating faster than CCO, and they are separated in time.

A Simple Linear Model, Revisited

- If the separation principles are adopted, the majority of the entries in the A and B matrices are zero:

- $$A = \begin{bmatrix} A_{MRO_1 MRO_1} & A_{MRO_1 MRO_2} & A_{MRO_1 MLB_1} & A_{MRO_1 MLB_2} & A_{MRO_1 CCO} \\ A_{MRO_2 MRO_1} & A_{MRO_2 MRO_2} & A_{MRO_2 MLB_1} & A_{MRO_2 MLB_2} & A_{MRO_2 CCO} \\ A_{MLB_1 MRO_1} & A_{MLB_1 MRO_2} & A_{MLB_1 MLB_1} & A_{MLB_1 MLB_2} & A_{MLB_1 CCO} \\ A_{MLB_2 MRO_1} & A_{MLB_2 MRO_2} & A_{MLB_2 MLB_1} & A_{MLB_2 MLB_2} & A_{MLB_2 CCO} \\ A_{CCO MRO_1} & A_{CCO MRO_2} & A_{CCO MLB_1} & A_{CCO MLB_2} & A_{CCO CCO} \end{bmatrix}$$

- $$B = \begin{bmatrix} B_{MRO_1 M_1} & B_{MRO_1 M_2} & B_{MRO_1 M_3} & B_{MRO_1 M_4} & B_{MRO_1 M_5} \\ B_{MRO_2 M_1} & B_{MRO_2 M_2} & B_{MRO_2 M_3} & B_{MRO_2 M_4} & B_{MRO_2 M_5} \\ B_{MLB_1 M_1} & B_{MLB_1 M_2} & B_{MLB_1 M_3} & B_{MLB_1 M_4} & B_{MLB_1 M_5} \\ B_{MLB_2 M_1} & B_{MLB_2 M_2} & B_{MLB_2 M_3} & B_{MLB_2 M_4} & B_{MLB_2 M_5} \\ B_{CCO M_1} & B_{CCO M_2} & B_{CCO M_3} & B_{CCO M_4} & B_{CCO M_5} \end{bmatrix}$$

- Hence, inter-SON interactions are avoided.

Application of the Design Principles in SEMAFOUR

The six design principles have been adopted for the SEMAFOUR SON functions

- All Radio Access SON functions meet the design principles 1.x and are prepared for coordination actions
- The Radio Access SON functions are independent by adopting design principles 2.x, and there are no inter-SON function conflicts expected.

So, What Would Groucho Have Said?

Those are my principles. If you don't like them, I have others.

The challenge of potential SON conflicts should be addressed already in the design phase

- SON functions should be prepared for coordination actions
- SON functions can be designed for conflict-free inter-SON function operation.

The presented six design principles ensures this if met.