Reliable Operation of Heterogeneous Wireless Networks with SON (Self-Organizing Networks) - SON coordination

Henning Sanneck, Szabolcs Nováczki

Nokia Siemens Networks
- Research

Tutorial DRCN 9th International Conference on Design of Reliable Communication Networks

Nokia Siemens Networks

© Nokia Siemens Networks 2012

SON operation: SON management + SON coordination

- **SON management:**
  - Enabling the control of the SON system by the human operator (governing the behaviour of the SON functions (target definition) based on business-level requirements) in a uniform way → „design-time“
  - Configuration (target setting, on/off, time scheduling, stop points / approval of actions), Monitoring (results) of the individual SON functions → „run-time“
- **SON coordination (a.k.a. „conflict management“, „semantic coordination“):**
  - Assuring the stable operation of the SON system as a whole by addressing the interactions between the use cases → „run-time“
  - Key characteristic: relates to more than one single function instance
  - SON coordination itself should be automated (e.g., by rule-based decision making), yet it may be run under close human supervision (manual approval of coordination actions) as well

→both SON management and SON coordination need a proper interface (GUI, tool chain) to the human operator

© Nokia Siemens Networks 2012

Why SON coordination is required

- Intra-area (optimization):
  - SON functions realize only parts of the overall optimization and are embedded into the live network
  - SON functions monitor specific KPIs and react autonomously to detected conditions by changing CM parameters
- Inter-area
  - healing – optimization: suspend optimization during recovery action execution
  - energy saving – healing: avoid false alarms
  - energy saving – optimization: adapt optimization execution

→ Avoid *run-time* concurrency issues (effect cancellations, race conditions, deadlocks) and oscillations („feature interaction“ in a distributed system)

→ Automatically suspend actions or trigger actions

SON coordination corresponds to human- and machine-level coordination activities in conventional management, e.g.,

- consistency checking at planning time for offline optimization
- suspending new CM plan download in case of a fault incident

© Nokia Siemens Networks 2012

Logical relationship of SON functions

Bell Labs Technical Journal 15(3), 5–18 (2010)**:**

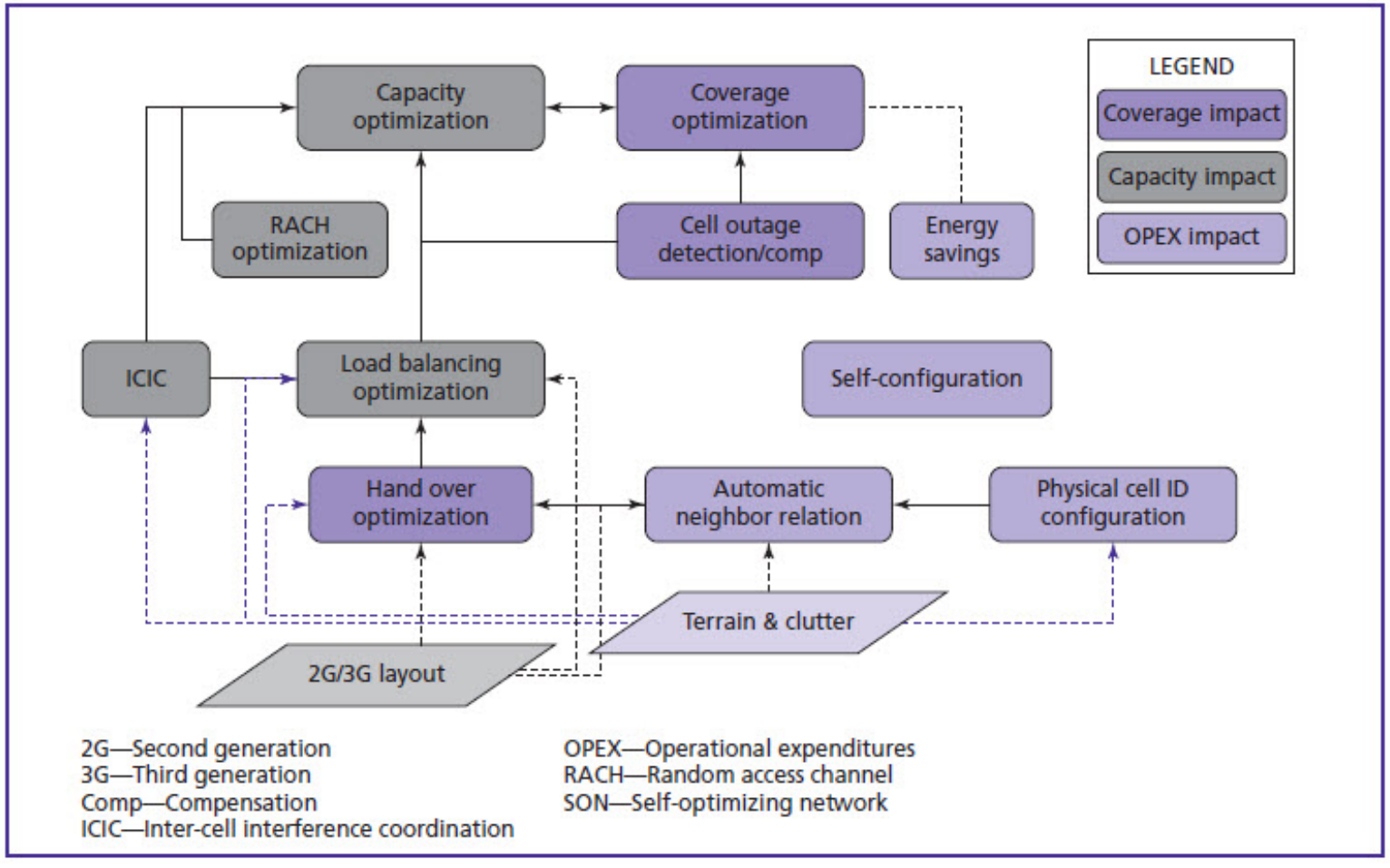
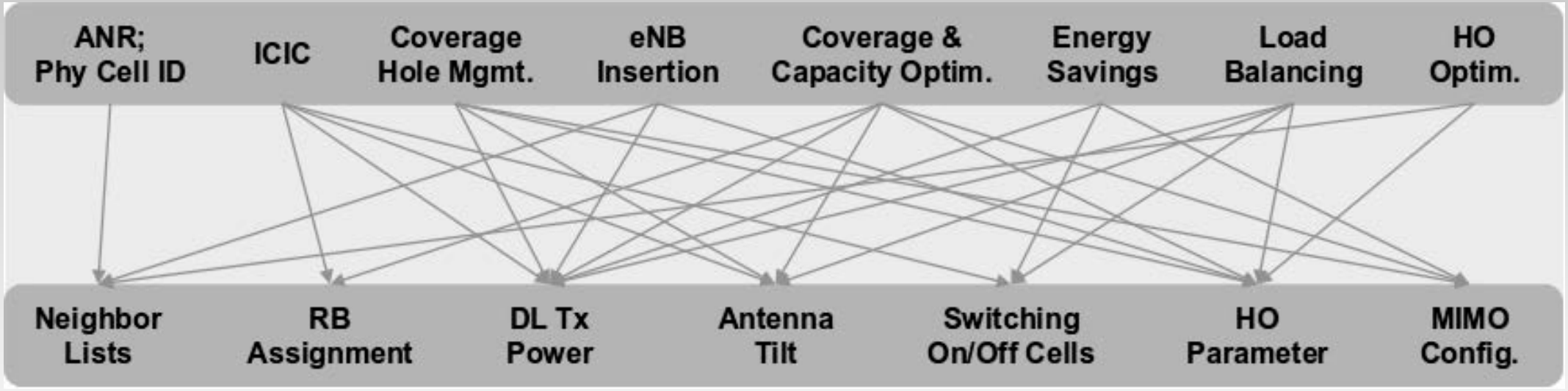


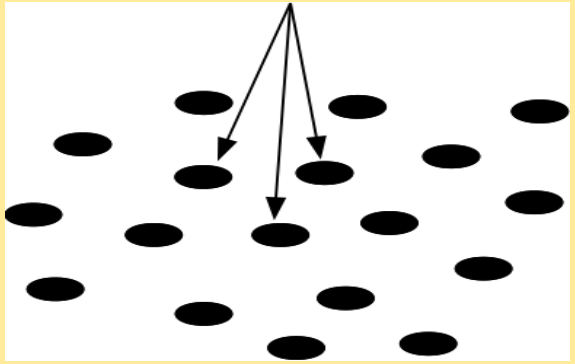
Figure 1.
Various SON use cases shown with their dominant optimization purpose (coverage, capacity, or OPEX reduction).

Parameter relationships of SON functions

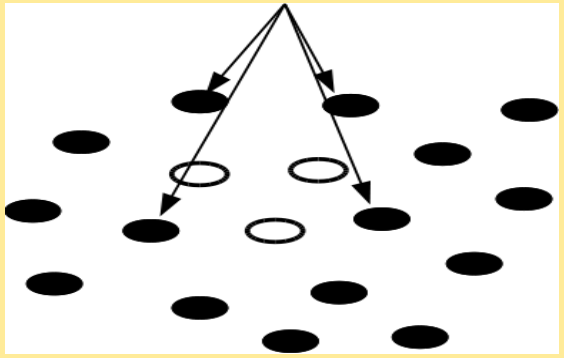


© Nokia Siemens Networks 2012

Impact-area



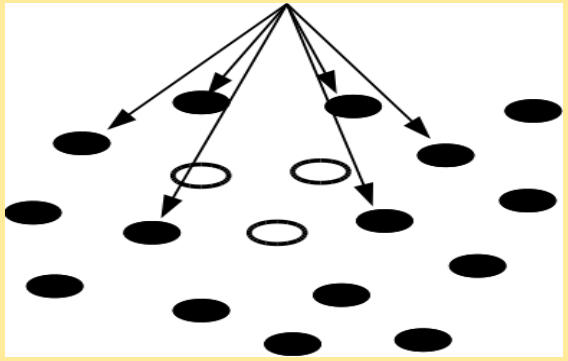
- **Function area:**
  - **Single SON function with three target cells**



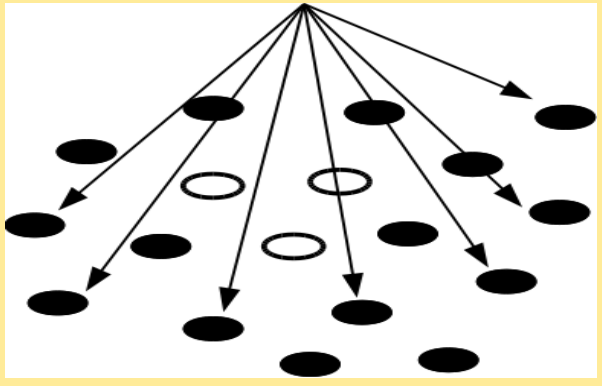
- **Input area**
  - **Input is provided from four additional cells**
  - **Function-area is indicated by non-filled circles**

© Nokia Siemens Networks 2012

Impact-area



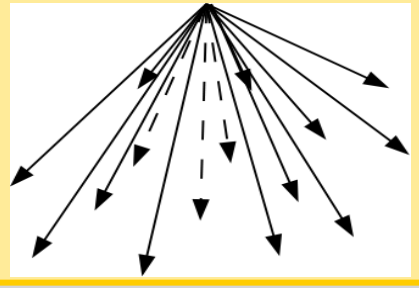
- **Effect area:**
  - The SON function has an effect on four of the surounding cells



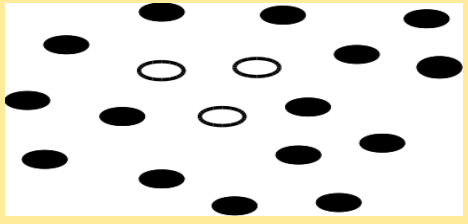
- **Safety margin:**
  - The Safety-margin for the SON function extends to seven neighboring cells

© Nokia Siemens Networks 2012

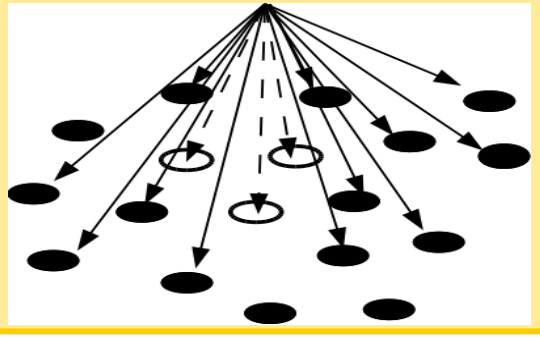
Impact-area: Run-time Mapping



- **Pre-defined abstract Impact area:**
  - Function-area indicated by dashed arrows
  - Remaining arrows indicate Input area, Effect area, Safety margin



- **Identification of SON functions Function area:**
  - Target cells in actual deployment are identified
  - Information is provided by SON function



- **Abstract Impact area is mapped to deployment**
  - Cells that form the Impact-area can easily be derived and indentified

© Nokia Siemens Networks 2012

Impact time

- Impact time defines the time interval during which an executed SON function instance has to be considered for coordination of other SON function instances
  ➔ Visibility of a particular SON function instance for the SON coordinator
  - Executed function impacts input data of another function
  - Operational guidelines require specific behavior ➔ To be enforced by coordination
- Impact time defined for each SON function individually
- SON coordinator applies SON function specific Impact time for each of the function instances separately
- Impact time is tightly coupled to the different conflict types
- Impact time is the time an executed SON function has potentially impact on other functions (not its impact on the network)

© Nokia Siemens Networks 2012

Impact time

- Potential conflicts depend on the respective timing, e.g., a „slow“ CCO has impact on a „fast“ MRO with lower probability
- Yet SON is not only about reducing OPEX as it is today, but doing network operations
  - Faster (more frequent executions) as today
  - More adaptive (variable user demand over the course of a day, cf. LiquidNet)
- For centralized SON there is usually a lower bound on data availability (granularity period: 15mins – 1h) which translates to a lower bound on the execution interval
  - → A number of SON functions will execute at this time interval
  - → Bursts of change requests appear within a short time period

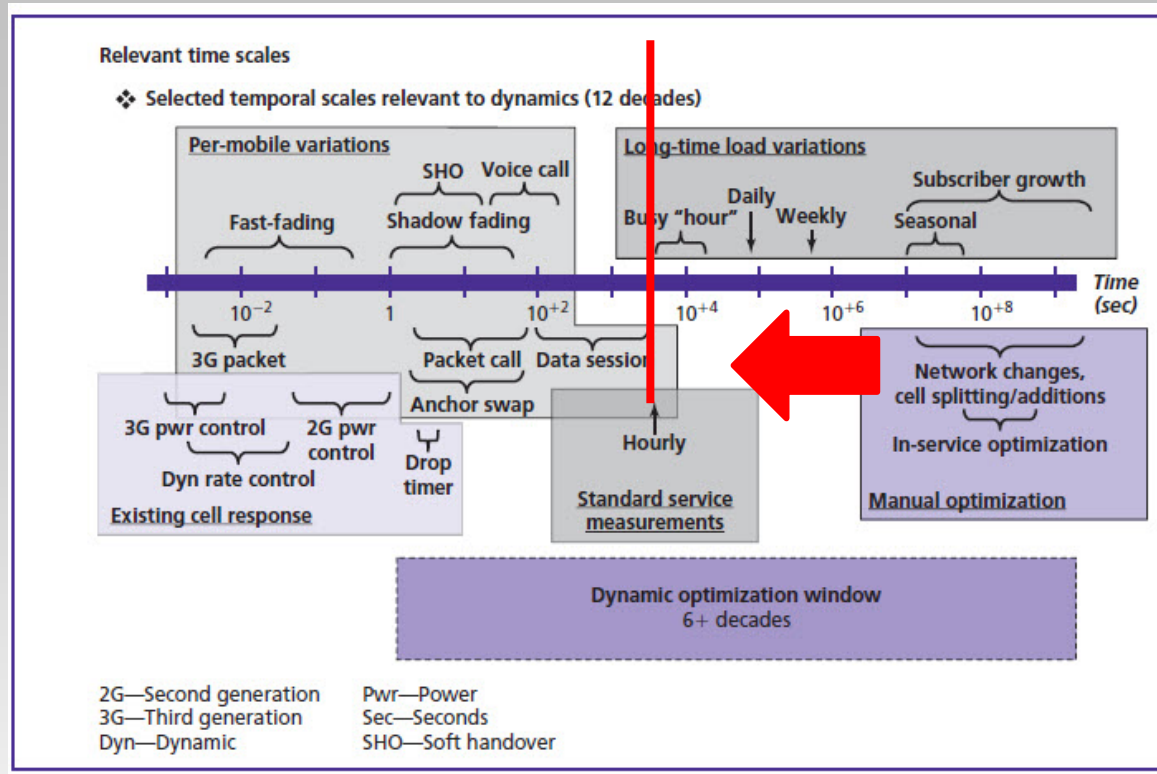
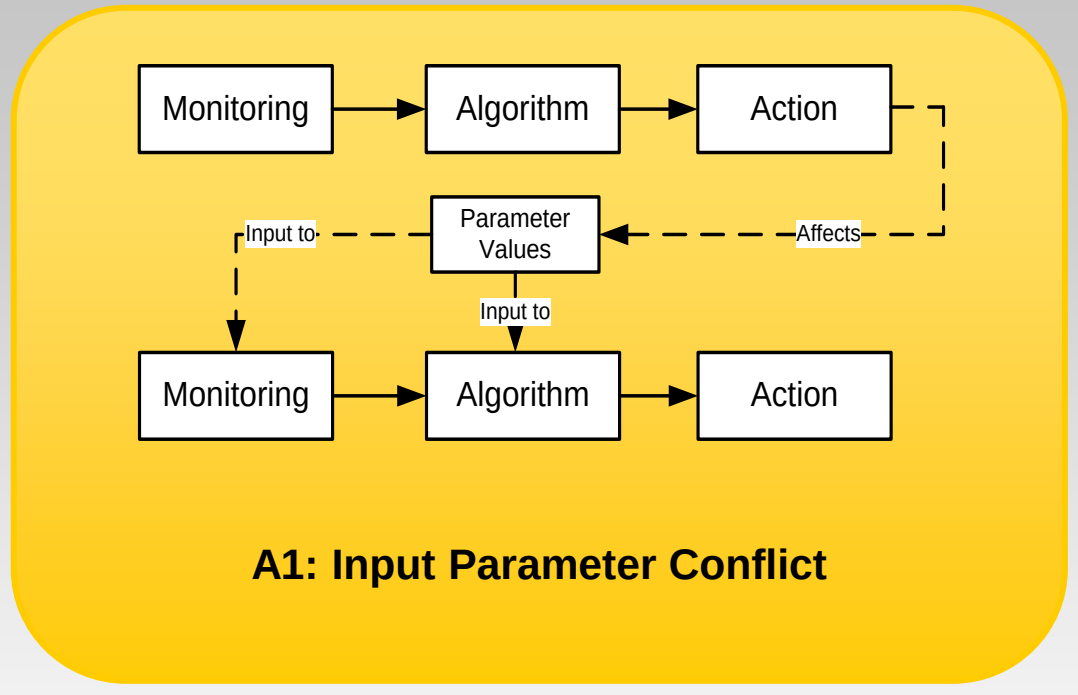


Figure 2.
A Bell Laboratories slide from 2003 showing relevant time scales for dynamic optimization.

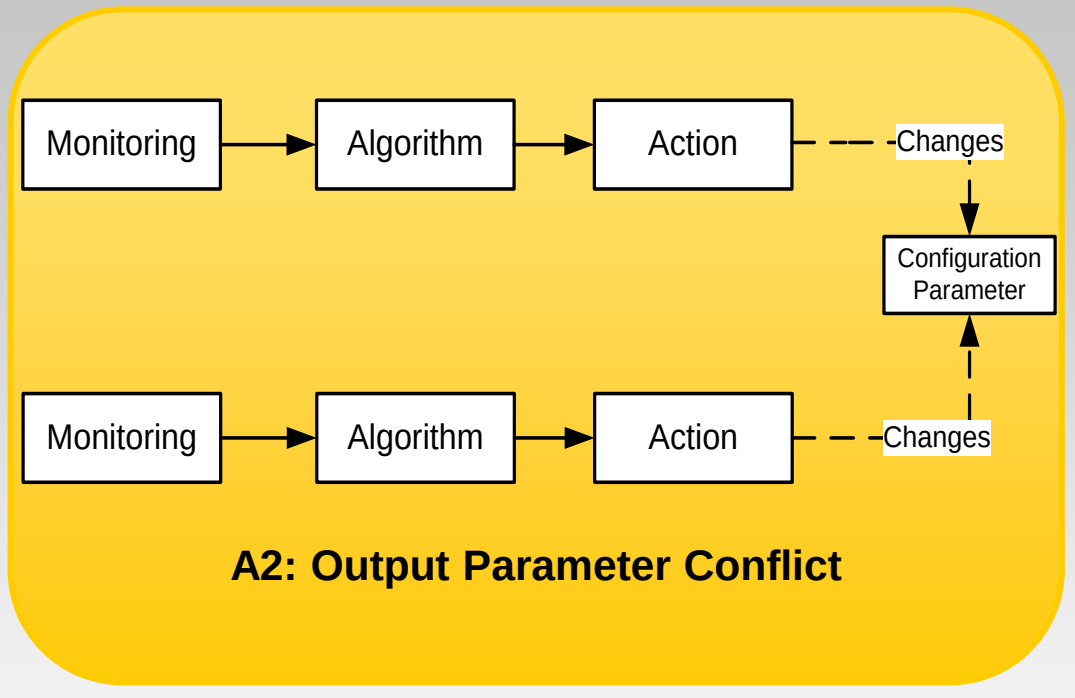
10 Bell Labs Technical Journal DOI: 10.1002/bltj

© Nokia Siemens Networks 2012

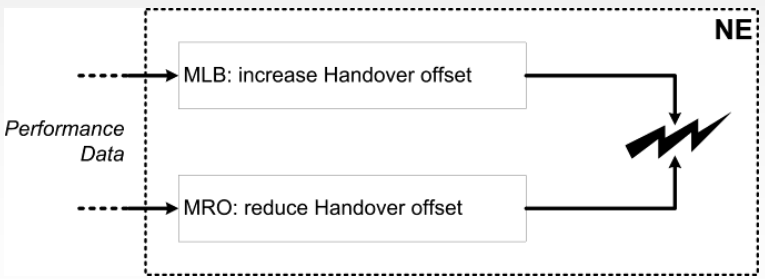
Conflict types



A1: Input Parameter Conflict



A2: Output Parameter Conflict

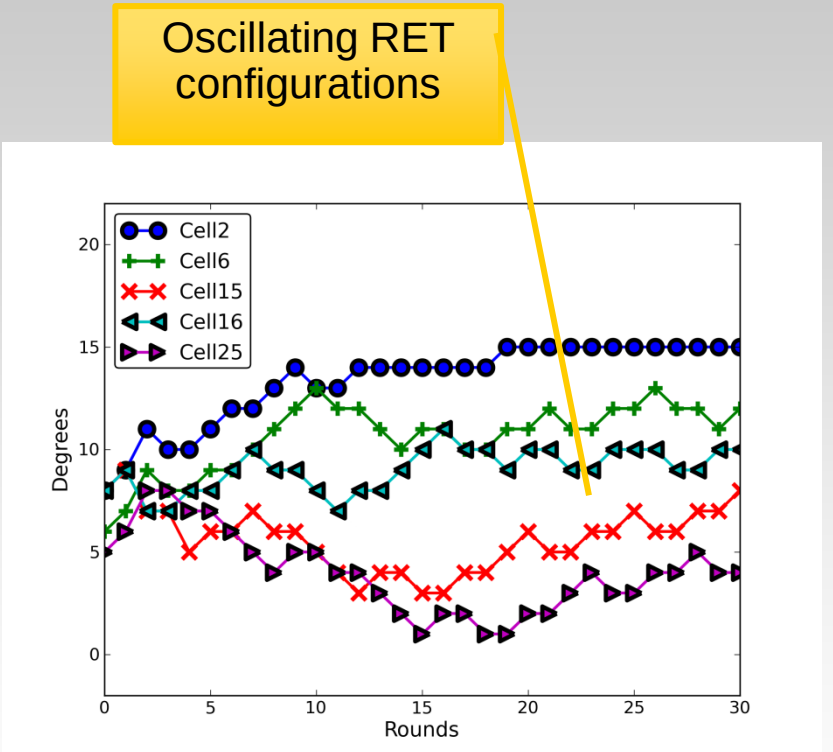


© Nokia Siemens Networks 2012

CCO / CCO (A2: output parameter conflict) (intra-area: self-opt)

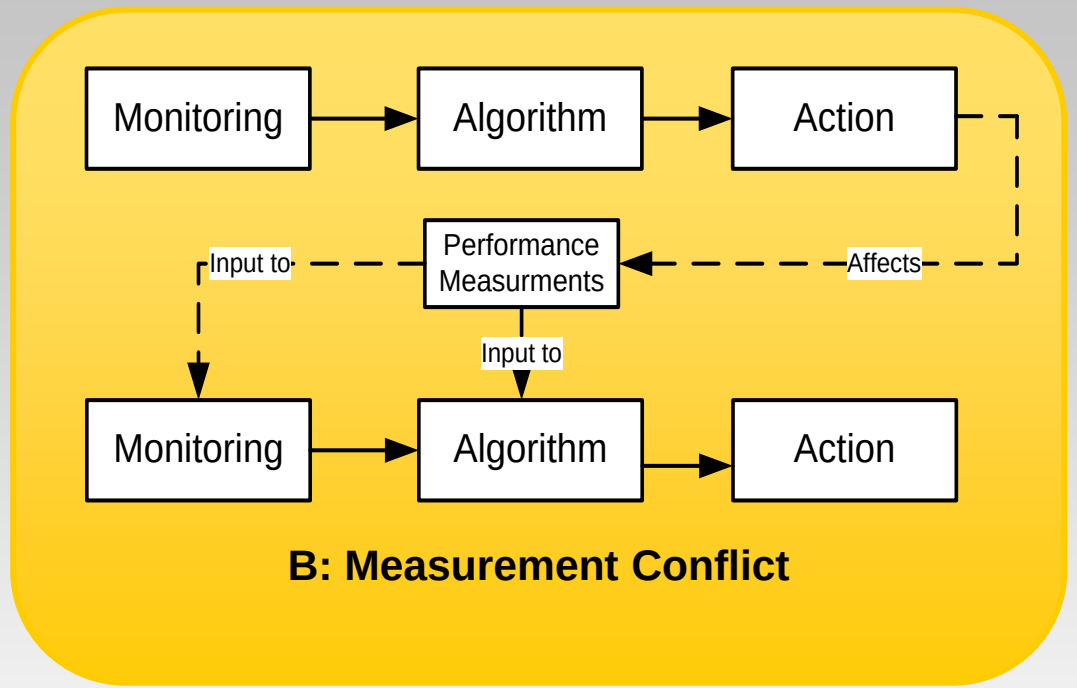
- Subsequent instances of CCO functions operate on identical target set
- Configuration is close to optimal
  - Performed changes to not provide a major benefit
  - SON function instances still try to optimize configuration of individual cells
    ➔ Oscillating reconfigurations (up – down – up – down)

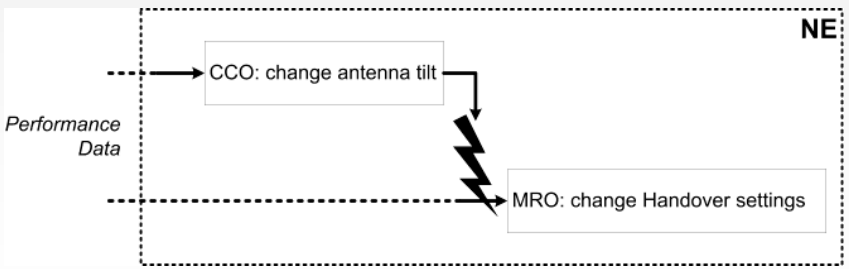
- Oscillation prevention:
  - Adaptation of SON functions
  - Coordination analyses requested changes and rejects them if oscillation is detected



© Nokia Siemens Networks 2012

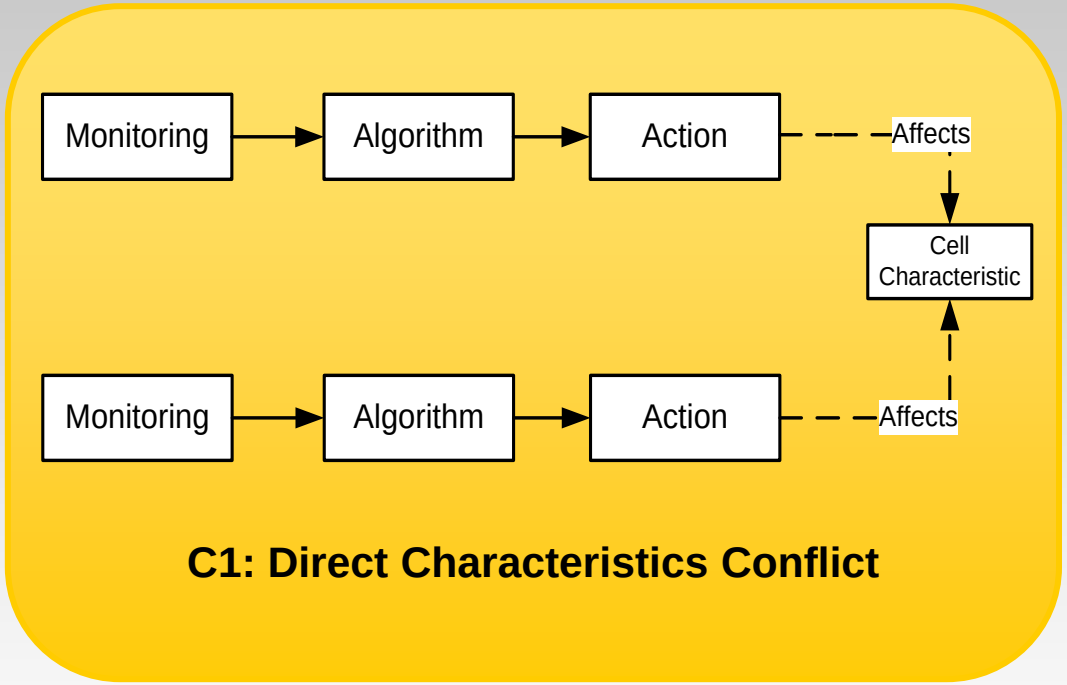
Conflict types



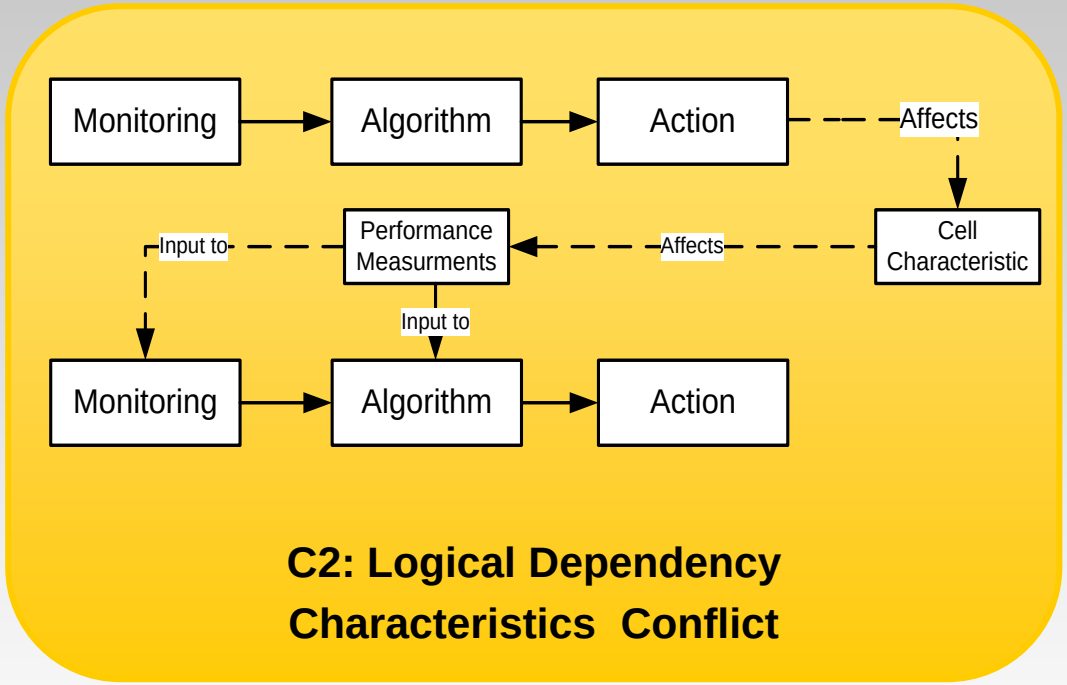


© Nokia Siemens Networks 2012

Conflict types



C1: Direct Characteristics Conflict



C2: Logical Dependency Characteristics Conflict

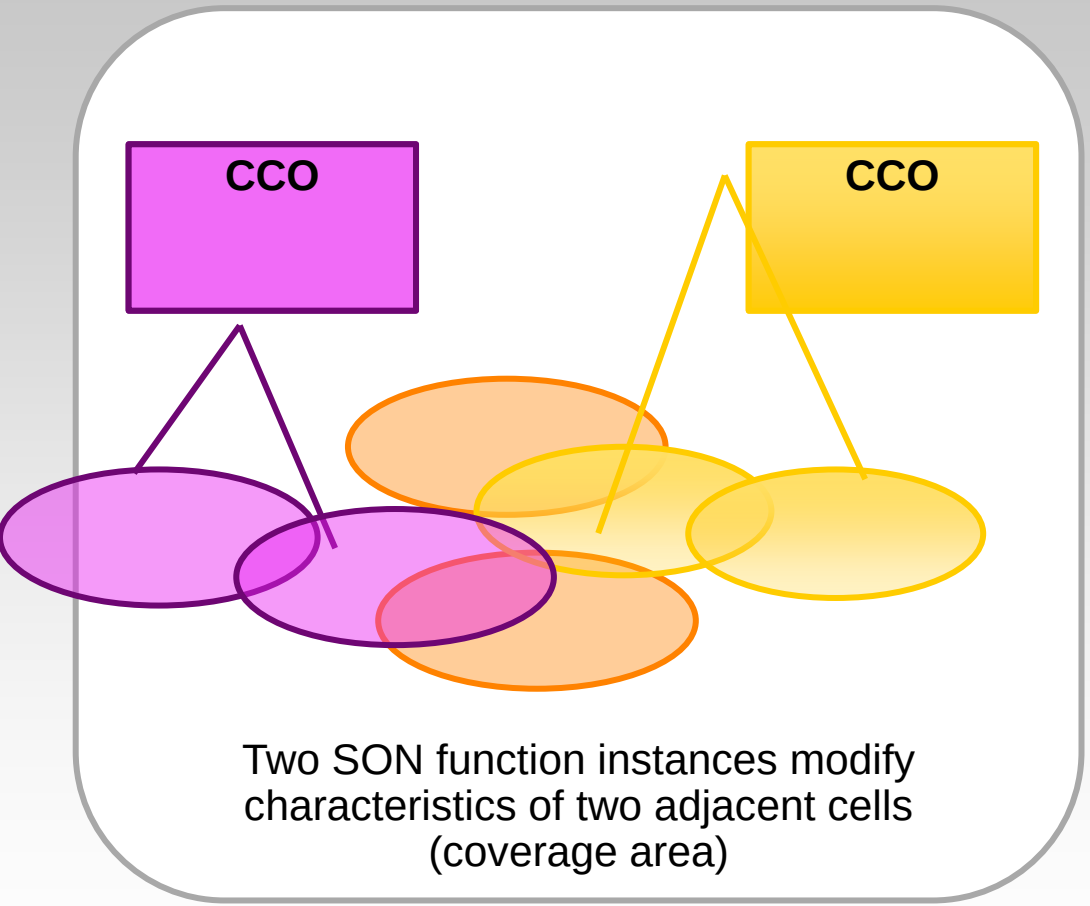
© Nokia Siemens Networks 2012

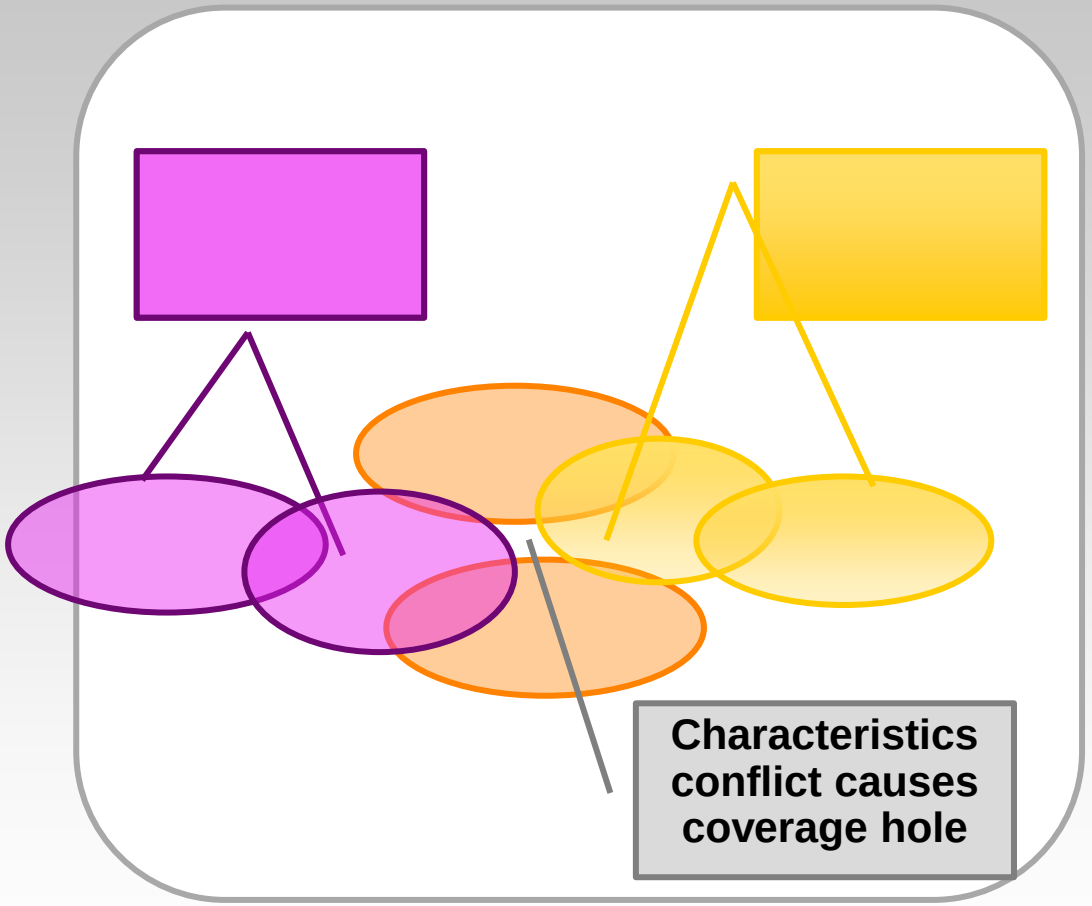
Conflict types: legacy concurrency control (OS, DBMS) viewpoint

- Input-related („race condition“): **A1, B, C2**
  - http://en.wikipedia.org/wiki/Race_condition#Example
    - both function instances („threads“) read system state (here: KPIs, CM data) corresponding to a certain time t. Instance 1 changes that state without Instance 2 being aware of it → potentially wrong / bad result
    - Solution: mutual exclusion (mutex)
- Output-related („lost update problem“): **A2, C1**
  - http://en.wikipedia.org/wiki/Concurrency_control#Why_is_concurrency_control_needed.3F
    - Both function instances write system state (here: CM data, change cell characteristic). Instance 2 overwrites the change of Instance 1 (without Instance 1 having taken effect)
    - Solution: locking
    - Special case: „oscillation“ (ping pong between change by Instance 1 and 2)
      - Solutions: choice of meaningful min interval between changes, specific oscillation detection
- Specifics in SON domain:
  - Problem: system state which needs to be protected
    - are both KPIs (*statistics on KPIs !*) and CM parameters (the correspondence to an operating system or a database would be only CM parameters)
    - is physically distributed (cells on different basestations, RNCs)
  - Solution:
    - Impact area and **time** definition related to both Instance 1 and 2
    - **„Virtual“ locking** at OSS (not NE) level: single mechanism for mutexes and locks

© Nokia Siemens Networks 2012

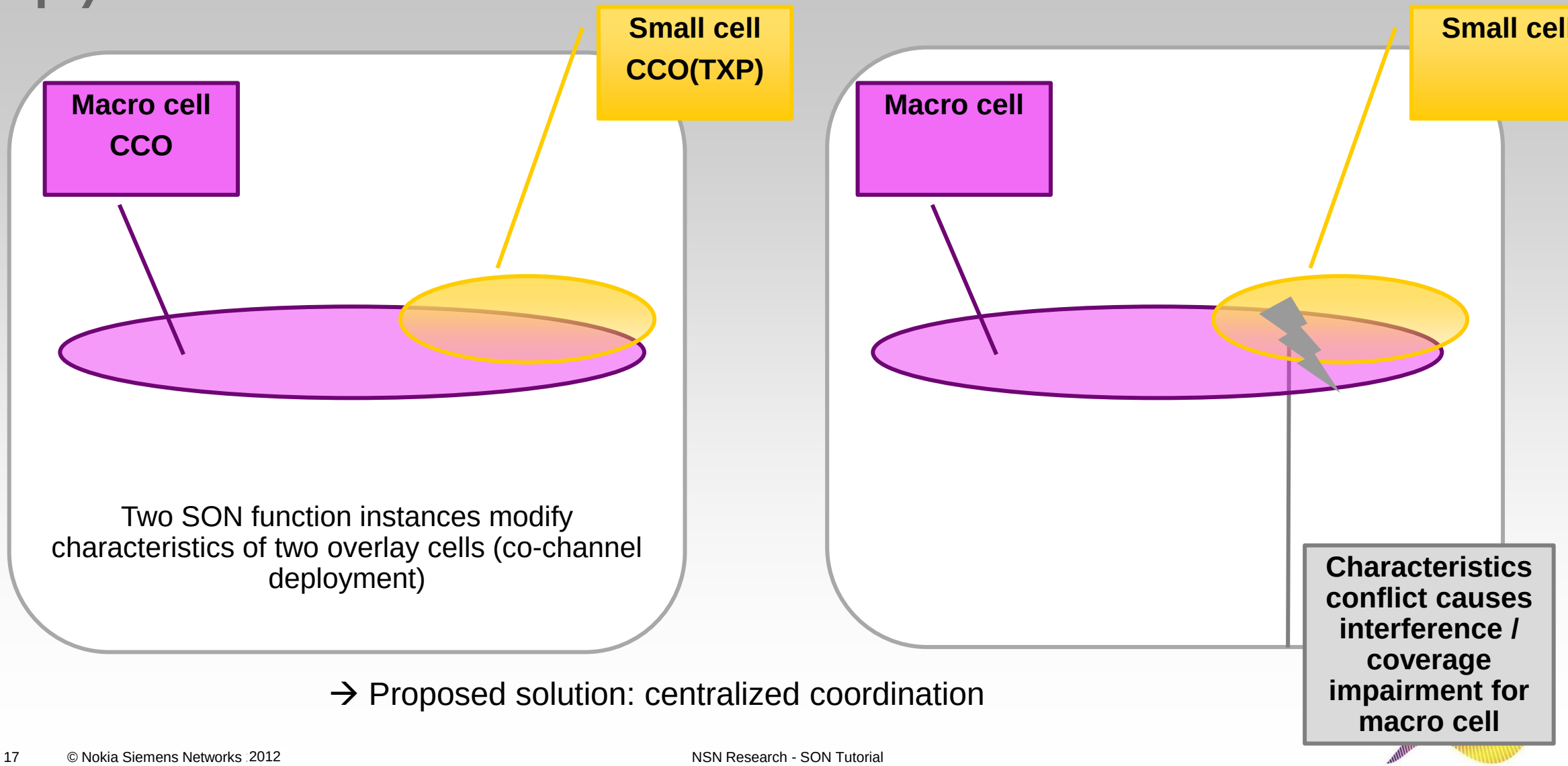
CCO / CCO (C2: logical characteristics conflict) (intra-area: self-opt)





© Nokia Siemens Networks 2012

CCO / CCO (C2: logical characteristics conflict) (intra-area: self-opt)

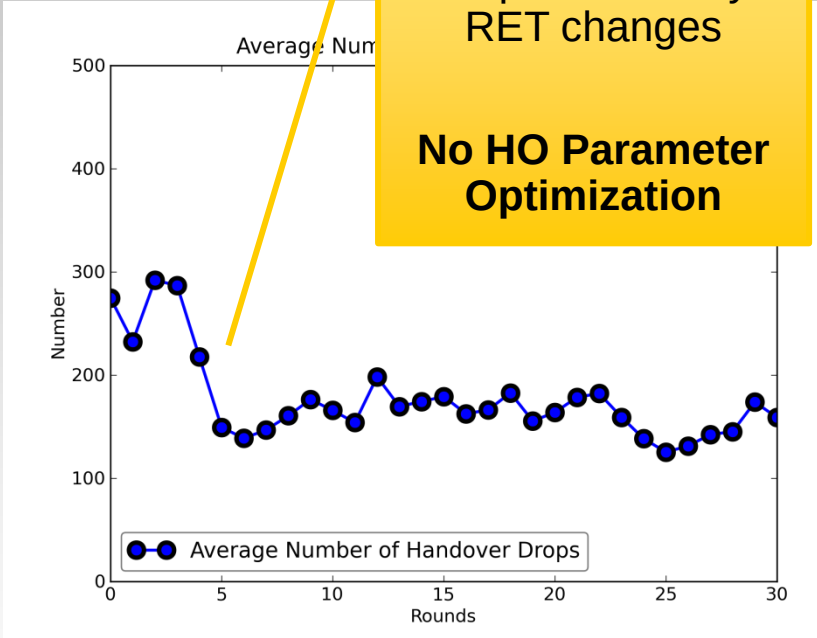


→ Proposed solution: centralized coordination

© Nokia Siemens Networks 2012

CCO / MRO (C2: logical characteristics conflict) (intra-area: self-opt)

- CCO(RET) instances change the cell boundaries
  ➔ especially overlap area
- Previously non-optimal handover parameter settings fit better to the new cell layout
  ➔ Improved handover performance without HO parameter optimization
- Optimizations performed by concurrently executed MRO instances could have negative effect



© Nokia Siemens Networks 2012

References

S. Hämäläinen, H. Sanneck, C. Sartori (eds.)
LTE Self-Organizing Networks (SON) - Network Management Automation for Operational Efficiency, Wiley, ISBN 978-1-119-97067-5, December 2011. Chapter 9

Baliosian, J., Sailhan, F., Devitt, A. and Bosneag, A.N. (2006) The omega architecture: towards adaptable, selfmanaged networks. Proceedings of the 1st Annual Workshop on Distributed Autonomous Network Management Systems, June, Dublin, Ireland.

Bandh, T., Romeikat, R., Sanneck, H. and Tang, H. (2011) Policy-based coordination and management of Self-Organising-Network (SON) Functions. IFIP/IEEE Symposium on Integrated Management, May, Dublin, Ireland.

Bandh, T., Sanneck, H. and Romeikat, R. (2011) An experimental system for SON function coordination. 1st IEEE International Workshop on Self-Organising Networks, May, Budapest, Hungary.

Band, T., Schmelz, C. (2012) Impact-time Concept for SON-Function Coordination. 2nd IEEE International Workshop on Self-Organising Networks, August, Paris.

Barth, U. and Kuehn, E. (2010) Self-Organisation in 4G mobile networks: motivation and vision. 7th International Symposium on Wireless Communication Systems, September, York, UK.

Calder, M., Kolberg, M., Magill, E.H. and Reiff-Marganiec, S. (2002) Feature interaction; a critical review and considered forecast. Computer Networks, 41, 115–141.

Cleaveland, R. and Smolka, S. (1996) Strategic directions in concurrency research. ACM Computing Surveys, 28(4), 607.

© Nokia Siemens Networks 2012

References

Doettling, M. and Viering, I. (2009) Challenges in mobile network operation: Towards self optimising networks. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, April.

Romeikat, R., Bauer, B., Sanneck, H. (2012) An Integrated SON Experimental System for Self-Optimization and SON Coordination, 2nd IEEE International Workshop on Self-Organizing Networks, August, Paris, France.

Romeikat, R., Sanneck, H., Bandh, T (2013) Efficient, Dynamic Coordination of Request Batches in C-SON Systems, 3rd IEEE International Workshop on Self-Organising Networks, June, Dresden, Germany.

Sanneck, H., Schmelz, C., Bandh, T. et al. (2010) Policy-driven workflows for mobile network management automation. 6th International Wireless Communications and Mobile Computing Conference, June, Caen, France.

Schmelz, C., Amirijoo, M., Eisenblaetter, A. et al. (2011) A coordination framework for self-organisation in LTE networks. IFIP/IEEE Symposium on Integrated Management, May, Dublin, Ireland.

SOCRATES (2008) Use Cases for Self-Organising Networks Deliverable D2.1, INFSO-ICT-216284 SOCRATES, March. Available at http://www.fp7-socrates.eu/, [accessed 30 August 2011].

SOCRATES (2011) Final Report on Self-Organisation and its Implications inWireless Access Networks. Deliverable D5.9, INFSO-ICT-216284 SOCRATES, January. Available at http://www.fp7-socrates.eu/, [accessed 30 August 2011].

Strassner, J., Agoulmine, N. and Lehtitet, E. (2006) FOCALE – A novel autonomic networking architecture. Latin American Autonomic Computing Symposium (LAACS), Campo Grande, MS, Brazil.