Use Cases, Requirements and Assessment Criteria for Future Self-organising Radio Access Networks

Kathleen Spaey
IBBT / University of Antwerp

December 10, 2008
IWSOS 2008 – Vienna, Austria

Joint work with Mehdi Amirijoo (Ericsson), Remco Litjens (TNO ICT), Martin Döttling (NSN), Thomas Jansen (TU Braunschweig), Neil Scully (Vodafone) and Ulrich Türke (Atesio), in the context of the FP7 SOCRATES project.
Outline

- Introduction: self-organising radio access networks
- Use cases
- Requirements
- Assessment criteria and benchmarking approach
- Conclusions and future work
Future radio access networks will exhibit a significant degree of self-organisation
- Self-optimisation, self-configuration, self-healing
- Broad attention: 3GPP, NGMN, FP7, …

Main drivers
- Effectuate substantial OPEX reductions
  - Improve effectiveness of human interventions
- Enhance network performance and service quality
  - Better adaptation to changing network characteristics and failures
Self-organisation
- Self-optimisation
  - Measurements, autonomous parameter adjustment, ...
  - Continuous loop
- Self-configuration
  - Incidental, intentional events
  - E.g., ‘plug-and-play’ installation of new base stations
- Self-healing
  - Incidental, non-intentional events

Focus on 3GPP LTE (E-UTRAN)
Self-organising radio access networks

- **Objectives**
  - Development of methods and algorithms for self-organisation
  - Quantitative assessment
    - Simulation of scenarios

- **Use cases**
  - Mechanisms for which self-organisation is anticipated to be effective and feasible

- **Requirements**
  - On the performance and functioning of the SON algorithm
  - To enable the development of solutions

- **Assessment criteria**
  - Evaluation of the SON algorithms
    - ‘Which algorithm is best’ if more than one algorithm for a use case
  - Assessment of the gains that can be achieved using SON
    - By comparison with manual network operation
  - Metrics and assessment approaches
Use cases

- **Self-optimisation**
  - Radio network optimisation
    - Interference coordination
    - Self-optimisation of physical channels
    - RACH optimisation
    - Self-optimisation of home eNodeB
  - GoS / QoS related parameter optimisation
    - Admission control parameter optimisation
    - Congestion control parameter optimisation
    - Packet scheduling parameter optimisation
    - Link level retransmission scheme optimisation
    - Coverage hole detection
  - Handover related optimisation
    - Handover parameter optimisation
    - Load balancing
    - Neighbour cell list

- **Others**
  - Reduction of energy consumption
  - Tracking areas
  - TDD UL/DL switching point
  - Management of relays and repeaters
  - Spectrum sharing
  - MIMO

- **Self-configuration**
  - Intelligently selecting site locations
  - Automatic generation of default parameters for NE insertion
  - Network authentication
  - Hardware / capacity extensions

- **Self-healing**
  - Cell outage prediction
  - Cell outage detection
  - Cell outage compensation
Use cases: packet scheduling parameter optimisation

- Self-optimisation use case
- Packet scheduling
  - Key radio resource management mechanism in LTE
  - Coordinates access to shared channels in time / frequency domain
  - Traffic types with distinct QoS requirements (delay, throughput, …)
  - A typical scheduler integrates proportional fairness and deadline-based principles
    - Various tuneable parameters → candidate parameters for self-optimisation
      - Capacity sharing between services
      - Degree of proportional fairness
      - …
- Self-optimisation based on
  - Observed performance and efficiency issues
  - Observed ‘environmental’ changes
    - Traffic characteristics, traffic mix, spatial distribution
    - Propagation conditions
    - User mobility
On the performance and functioning of the SON algorithm
To enable the development of solutions
Technical and business requirements
Technical requirements

- **Performance and complexity**
  - Trade-off, e.g., limitations on measurement related signalling overhead

- **Stability**
  - Important, as the algorithms should run without manual intervention
  - Iterations of the algorithm should converge
  - Only significant changes should trigger the recalculation of parameters

- **Robustness**
  - Algorithms should be able to deal with unexpected events
    - Missing, wrong or corrupted input (measurements)

- **Timing**
  - How often should an algorithm run, how fast should an algorithm react
  - Use case dependent, time scales range from ms to hours or days

- **Interaction**
  - Alignment with other algorithms in own cell and surrounding cells is required, particularly relating to common parameters

- **Architecture and scalability**
  - Centralised or decentralised algorithms ↔ requirements on architecture and interfaces
  - Use case dependent

- **Required inputs**
  - Counters, measurements, etc.
  - Use case dependent
Business requirements

- Cost efficiency requirements
  - OPEX / CAPEX reductions

- LTE deployment
  - Speed up roll-out of LTE networks → easily solving problems to ensure network quality
  - Simplify processes → solve real problems occurring in real networks
  - Do not introduce other manual efforts → no extra effort required to set-up / configure SON functionality
  - Easy deployment of new services → meet QoS requirements of new services
  - End user benefits → user should experience high GoS and QoS
  - Deployment trends → Impact of network sharing should be considered
Assessment criteria

- **Evaluation** of the SON algorithms
  - ‘Which algorithm is best’ if more than one algorithm for a use case

- **Assessment of the gains** that can be achieved using SON
  - By comparison with manual network operation

- **Metrics and benchmarking approaches**
Metrics

- Performance (GoS / QoS)
- Coverage
- Capacity
- OPEX
  - Often quoted as an important SON gain → important to be able to quantify the impact
  - Method which considers difference between OPEX without/with SON
    - Simplification of the reality, purpose of the model is to enable assessment
- CAPEX
  - SON effect on CAPEX will be a combination of reduced number of sites and increased equipment cost per site
### Metrics: OPEX

- **OPEX without SON** is determined by summing together all components that contribute to OPEX
  - Determine the cost of an individual task
    - Task is defined as optimising or adjusting a parameter or parameter set
    - Effort per task (days) = A + B + C
      - A = gathering input info (e.g., planning tool, performance counters, drive tests)
      - B = determine new settings (e.g., manual, computer assisted by planning tool / simulator)
      - C = apply new settings (e.g., automatic processes, site visits, etc.)
    - Cost per task (Euro) = effort per task (days) x cost per day (Euro)
  - Determine OPEX per task, per network, per year
    - OPEX per task / year = (cost per task) x (# changes per network) x (# changes per year)
  - Determine total OPEX per year
    - OPEX / year = SUM<sub>all tasks</sub> (OPEX per task / year)

- **OPEX with SON** is determined using the same method as for without SON, but by assessing the impact on the various components
  - In some cases OPEX may be reduced to zero, but definitely not always

- **Difference** is then assessed
Reduced number of sites
  – Specify a scenario
    • System parameters
    • Traffic / mobility, including traffic load per km²
    • QoS / GoS requirements
  – For both the case with and without SON
    • Determine maximum cell size, such that QoS / GoS requirements are still met
    • CAPEX is determined based on the number of sites needed, including costs for backhaul and core network elements

Increased equipment cost per site
  – Computational complexity
  – Network bandwidth requirements
    • Extra backhaul capacity for SON
  – Additional site equipment
    • E.g., electrical antenna tilt

The main challenge will be the exact quantification of increased equipment costs
Benchmark approach: comparing SON algorithms

- Compare different SON algorithms developed for a given use case
  - Specify a scenario
  - Consider different self-optimisation algorithms
    - SO_A, SO_B, SO_C, SO_D
  - Evaluate all metrics for each algorithm
  - Determine overall ranking
    - Combining different metrics using a utility function
    - Single target metric, with constraints on the other metrics
Benchmarking approach: gains from SON

- Compare SON algorithm with manual network optimisation
  - Specify a scenario
  - Consider different SO and MO (manual optimisation) approaches

self-optimisation

manual optimisation (benchmark)

assess e.g. OPEX/CAPEX gains of SO\textsubscript{A} with regard to benchmark MO\textsubscript{D}

OPTIM. EFFORT

CAPEX

OPEX gain

CAPEX gain

SO\textsubscript{A}

MO\textsubscript{D}
### Benchmarking approach: gains from SON

#### CAPEX GAINS

<table>
<thead>
<tr>
<th></th>
<th>SO_A</th>
<th>SO_B</th>
<th>SO_C</th>
<th>SO_D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO_A</td>
<td>0</td>
<td>-</td>
<td>--</td>
<td>---</td>
</tr>
<tr>
<td>MO_B</td>
<td>+</td>
<td>0</td>
<td>-</td>
<td>--</td>
</tr>
<tr>
<td>MO_C</td>
<td>++</td>
<td>+</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>MO_D</td>
<td>+++</td>
<td>++</td>
<td>+</td>
<td>0</td>
</tr>
</tbody>
</table>

#### OPEX GAINS

<table>
<thead>
<tr>
<th></th>
<th>SO_A</th>
<th>SO_B</th>
<th>SO_C</th>
<th>SO_D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO_A</td>
<td></td>
<td></td>
<td></td>
<td>+++</td>
</tr>
<tr>
<td>MO_B</td>
<td></td>
<td></td>
<td></td>
<td>+++</td>
</tr>
<tr>
<td>MO_C</td>
<td></td>
<td></td>
<td></td>
<td>++</td>
</tr>
<tr>
<td>MO_D</td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

- **Gains from SON will depend on current operator approach**
  - Operator invests a lot in manual optimisation ➔ gain is mainly in OPEX reduction
  - Operator invests very little in manual optimisation ➔ gain is mainly in network quality improvement
Conclusions and future work

- Identification of **use cases, requirements and assessment criteria** for future **self-organising radio access networks**

- Basis for a **framework** for the development of SON methods and algorithms
  - **Relation and dependencies** between different SON components

- **SON algorithms** for the identified use cases will be developed
  - Taking into account the identified requirements
  - Evaluated using the proposed assessment criteria

www.fp7-socrates.eu