Self-Organization in Mobile Communication Systems

www.gs-mobicom.de

Andreas Mitschele-Thiel
Goals of the Course

• Understand self-organization principles
• Understand different methods of self-organization
• Know which self-organization method suits which problem
• Apply methods of self-organization to problems in communications
Course Organization

• Organization: Mohamed Abd Rabou Kalil
• Type of studies: lectures and discussions
• 10-12 lectures, Th 5 p.m.
• 2 credits/LP
• Exam: oral at the end of the semester

• Participants
  – Graduate School Mobicom (mandatory)
  – M.Sc. II, M.Sc. IN (elective)

• Prerequisites:
  – knowledge of communication systems and networks
  – interest in research issues
Content of Introduction

• Organization of Course

• Why is it important?
  => Motivation for Self-organization

• What does it mean?
  => Definition of Self-organization

• How does it work in practice?
  => Examples for Self-organization in Communication Systems
### Content of Course

#### Methods for self-organization
- Swarm intelligence and Ant algorithms
- Reinforcement learning
- Distributed Artificial Intelligence
- Game theory
- Neural Networks
- Data mining
- Evolutionary algorithms

#### Applications
- MANETs/VANETs
- Sensor networks
- Distributed information managements and P2P systems
- SO service/function placement
- Self-organized Networks (SON)
- Cognitive Radio and SO
- Bio-inspired adaptive autonomy

#### Methodology
- Systems Engineering concepts
- Analog vs. digital systems
Motivation for Self-Organization

- Problem of today's networks
  - Heterogeneity
  - Dynamics
  - Scalability

- Method: Self-organization
  - Fast, autonomous reaction to problems
  - Automation of control
  - Distributed control

- Some application scenarios:
  - Severe network impact due to disasters
  - Energy savings
  - Privately operated femto cells
Our Application Interest in Self-Organization

- **R** Reconfigurable Radio Interfaces
- **S** Self-organized Service Recovery
- **I** Decentralized Information Management
- **T** Cognitive Management of Transport Resources
Research Topics – Secure Network Operation

How can access to adhoc-deployed communication infrastructure be restricted?

- Innovative authentication techniques based on localization?
- How to ensure secure localization?
- How to configure/manage access rights, cryptographic keys etc.?
Research Topics – Integration of UAVs

Self-organized integration of mobile platforms (land-based and airborne)
Self-organized selection of MIMO clusters in distributed MIMO systems employing Ant or Swarm algorithms
Research Topics – Cognitive Radio Systems

Secondary use of spectrum not used by primary user
- large portion of frequency spectrum is unused most of the time
- distributed sensing of the medium to check for re-appearence of primary user
- distributed coordination of access to medium
- learn from past usage
Economic Drivers for Self-Organization

Decouple network cost from traffic volume!

Traffic volume

Network cost (existing technologies)

Revenue
Profit
Network cost (LTE)

Time

Voice dominated
Data dominated

Revenue
Profit
Network cost (LTE)
Self-Organization and Autonomous Systems

What’s the Name for this New Control Systems Science?

- Chaos Control?
- Complexity Control?
- Autonomous I/O Systems?
- Peer-to-Peer Control Systems?

Self-Organization
From Centralization to Self-Organization

- Monolithic / centralized systems
  Monolithic: Systems consisting of a single computer, its peripherals, and perhaps some remote terminals
  Centralized: systems with a well-defined centralized control process

- Distributed systems
  A collection of independent subsystems that appears to the application as a single coherent system

- Self-organizing autonomous systems
  Autonomously acting individual systems performing local programs and acting on local data but participating on a global task, i.e. showing an emergent behavior
Yates et al. (1987)

“Technological systems become organized by commands from outside, as when human intentions lead to the building of structures or machines. But many natural systems become structured by their own internal processes: these are the self-organizing systems, and the emergence of order within them is a complex phenomenon that intrigues scientists from all disciplines.”

Camazine et al. (2003)

“Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of a system. Moreover, the rules specifying interactions among the systems’ components are executed using only local information, without reference to the global pattern.”
Self-Organization – Example from Chemistry

Pattern formation in the Belousov-Zhabotinski reaction

Photography by Juraj Lipscher
Examples of Self-Organized Systems:

- **biology**: Ants building an ant colony, swarms of birds flying in some joint and well coordinated way
- **economy**: Capitalistic economy with minimal rules and lots of individuals with ideas and goals representing local algorithms
- **human society**: Formation of countries, any kind of organizations, communities, associations, worker unions, societies, clubs, etc.

But: From time to time there may be a breakdown, crash or collapse of the system, e.g.

- **human societies**: War, civil war
- **economy**: Crash of financial markets
- **nature**: Natural catastrophes
- **nuclear physics**: Chain reaction resulting from exceeding critical mass
- **mechanics**: Oscillating bridges
- **chemistry**: Different aggregate states of material (water: liquid, solid (ice), steam)
Self-Organizing Systems

Local system control

Local interactions (environment, neighborhood)

Simple local behavior

C

Self-organization in Communication Systems
A. Mitschele-Thiel
Ilmenau University of Technology
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| No central control        | No global control system  
                               No global information  
                               Subsystems perform completely autonomous |
| Emerging structures       | Global behavior or functioning of the system  
                               emerges in form of observable pattern or structures |
| Resulting complexity      | Even if the individual subsystems can be simple as well as their basic rules, the resulting overall system becomes complex and often unpredictable |
| High scalability          | No performance degradation if more subsystems are added to the system  
                               System performs as requested regardless of the number of subsystems |
Self-Organization – System Properties

• Absence of external control
• Adaptation to changing conditions
• Global order and local interactions
• Complexity
• Control hierarchies
• Dynamic operation
• Fluctuations and instability
• Dissipation (waste of resources)
• Multiple equilibria and local optima
• Redundancy
• Self-maintenance

Systems lacking self-organization
- Instructions from a supervisory leader
- Directives such as blueprints or recipes
- Pre-existing patterns (templates)
Definition **Self-Organization** (Camazine rephrased)

Self-organization is a process in which structure and functionality (pattern) at the global level of a system emerge solely from numerous interactions among the lower-level components of a system without any external or centralized control. The system's components interact in a local context either by means of direct communication or environmental observations without reference to the global pattern.

Definition **Emergence**

Emergent behavior of a system is provided by the apparently meaningful collaboration of components (individuals) in order to show capabilities of the overall system (far) beyond the capabilities of the single components.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-configuration</td>
<td>Methods for (re-)generating adequate configurations depending on the current situation in terms of environmental circumstances</td>
</tr>
<tr>
<td>Self-management</td>
<td>Capability to maintain systems and devices depending on the current system parameters</td>
</tr>
<tr>
<td>Adaptability</td>
<td>Ability of the system's components to adapt to changing environmental conditions</td>
</tr>
<tr>
<td>Self-diagnosis</td>
<td>Mechanisms to perform system autonomous checks and to compare the results with reference values</td>
</tr>
<tr>
<td>Self-protection</td>
<td>Capability to protect the system and its components against unwanted or even aggressive environmental influences</td>
</tr>
<tr>
<td>Self-healing</td>
<td>Methods for changing configurations and operational parameters of the overall system to compensate failures</td>
</tr>
<tr>
<td>Self-repair</td>
<td>Similar to self-healing but focusing on actual repair mechanisms for failing system parts</td>
</tr>
<tr>
<td>Self-optimization</td>
<td>Ability of the system to optimize the local operation parameters according to global objectives</td>
</tr>
</tbody>
</table>
Characteristics of Self-Organizing Systems

Self-organizing systems are dynamic and exhibit emergent properties.

Since these system-level properties arise unexpectedly from nonlinear interactions among a system’s components, the term emergent property may suggest to some a mysterious property that materializes magically.

Example: growth rate of a population

\[ x_{n+1} = r x_n (1 - x_n) \]

- \( 0 < r < 1 \): extinction
- \( 1 < r < 3 \): constant size after several generations
- \( 3 < r < 3.4 \): oscillating between two values
- \( 3.4 < r < 3.57 \): oscillating between four values
- \( r > 3.57 \): “deterministic” chaos
Consequences of Emergent Properties

A small change in a system parameter can result in a large change in the overall behavior of the system

- Adaptability
- Flexibility

Role of environmental factors

- Specify some of the initial conditions
- Positive feedback results in great sensitivity to these conditions

Self-organization and the evolution of patterns and structure

- Intuitively: generation of adaptive structures and patterns by tuning system parameters in self-organized systems rather than by developing new mechanisms for each new structure
- However: the concept of self-organization alerts us to the possibility that strikingly different patterns result from the same mechanisms operating in a different parameter range

Simple rules, complex patterns – the solution to a paradox?
Basis Methods in Self-Organizing Systems

- Positive and negative feedback
- Interactions among individuals and with the environment
- Probabilistic techniques
Positive and Negative Feedback

Simple feedback

Feedback

Input → System state → Output

Measurement

Amplification problems

Snowballing effect

Implosion effect
Interactions among Individuals and with the Environment

- Direct communication among neighboring systems
- Indirect communication via the environment (stigmergy)
- Interaction with (stimulation by) the environment

Stigmergy is a mechanism of spontaneous, indirect coordination between agents or actions, where the trace left in the environment by an action stimulates the performance of a subsequent action, by the same or a different agent. Stigmergy is a form of self-organization. It produces complex, apparently intelligent structures, without need for any planning, control, or even communication between the agents. As such it supports efficient collaboration between extremely simple agents, who lack any memory, intelligence or even awareness of each other.
Probabilistic Techniques

- Examples: stochastic processes, random walk
- Objectives: leaving local optima, stabilization

Simulation results

- Initialization
- Random walk
- Bumped into a wood chip
- Carrying a wood chip
- Drop chip

Pick-up chip

No

Simulation results
Design Paradigms for Self-Organizing Systems

Paradigm #1:
Design local behavior rules that achieve global properties

Paradigm #2:
Do not aim for perfect coordination: exploit implicit coordination

Paradigm #3:
Minimize long-lived state information

Paradigm #4:
Design protocols that adapt to changes
**Design Paradigms for Self-Organizing Systems**

**Paradigm #1: Design local behavior rules that achieve global properties**

To design a network function that establishes a global property, this paradigm is to distribute the responsibility among the individual entities: no single entity is "in charge" of the overall organization, but each contributes to a collective behavior. Following this paradigm, localized behavior rules must be devised, if applied in all entities, automatically lead to the desired global property (or at least approximate it). "localized" mean that entities have only a local view of the network and interact with their neighbors and changes in the network have initially only local consequences.

**Paradigm #2: Do not aim for perfect coordination: exploit implicit coordination**

Implicit coordination means that coordination information is not communicated explicitly by signaling messages, but is inferred from the local environment. A node observes other nodes in its neighborhood; based on these observations, it draws conclusions about the status of the network and reacts accordingly.

**Paradigm #3: Minimize long-lived state information**

To achieve a higher level of self-organization, the amount of long-lived state information should be minimized. In general, the more localized the interactions are and the less coordination is needed, the less state information must be maintained.
Paradigm #4: Design protocols that adapt to changes

The need for the capability of nodes to react to changes in the network and its environment typically arises from changed resource constraints, changed user requirements, node mobility, or node failures. Since there are no centralized entities that could notify the nodes about changes, each node has to continuously monitor its local environment and react in an appropriate manner. Three levels of adaptation can be distinguished.

- **Level 1**: A protocol is designed so that it can **cope with changes**, such as failure and mobility.
- **Level 2**: A protocol is designed to **adapt its own parameters** (e.g., value of timers, cluster size) as a reaction to changes in order to optimize system performance.
- **Level 3**: A protocol is designed so that it **realizes if the changes are so severe that the currently employed mechanism is no longer suitable**. To detect such situations the environment is observed, and significant changes in major parameters trigger a fallback or alternative solution.

Typically, these levels of adaptation are combined using control loops.
Design Paradigms for Self-Organizing Systems

Required functionality – system behavior (objectives)

Paradigm #1
- Local properties (divide and conquer)
- Local behavior rules

Paradigm #2
- Tolerable conflicts and inconsistencies (conflict detection and resolution)
- Implicit coordination

Paradigm #3
- Synchronized state (discovery mechanisms)

Paradigm #4
- Definition of severe changes and reactions (monitoring and control)
- Adaptive algorithms

Resulting protocol (behavior rules, messages, state, and control)
Limitations of Self-Organization

- Controllability
  - Predictability vs. scalability

- Cross-mechanism interference
  - Composition of multiple self-organizing mechanisms can lead to unforeseen effects

- Software development
  - New software engineering approaches are needed

- System test
  - Incorporation of the unpredictable environment
Self-Organized vs. Centralized Control

Self-organizing networks

- Local state
- Neighbor information
- Probabilistic methods

Networking functions for global connectivity and efficient resource usage

Centralized networks

- Global state (globally optimized system behavior)

Implicit coordination

Explicit coordination
Epidemic Data Replication

- Epidemic propagation of updates for data maintained redundantly on multiple sites
  - each incoming update is propagated to \( k \) randomly chosen neighbors with probability \( p \)
- Updates pass through the system like infectious diseases
  - All nodes are informed
  - No error handling (link failure etc.) needed!
Gossip-based Aggregation

- Compute a (global) aggregate value from distributed data
  - load balancing in networks
  - monitoring in industrial control applications
- Each node
  - maintains a local approximate value of the aggregate
  - communicates periodically with randomly chosen neighbors and updates the approximate value
- Only local pairwise interaction!

Average value?

3.3, 3.75
4.3, 5
Example: TCP Flow Control and RED

TCP flow control

Loss of a single packet (≥3 Dupacks)

Major problem (Timeout)

Random Early Detection (RED)

Always drop

Non-zero likelihood of drop

Never drop

Average Occupancy
Example: Media Access Control in 802.11

Distributed Coordination Function (DCF) with Exponential Backoff

- **DIFS**: Interframe space
- **bo**: Busy
- **bo_e**: Elapsed backoff time
- **bo_r**: Residual backoff time
- **busy**: Medium not idle (frame, ack etc.)
- **station 1**
- **station 2**
- **station 3**
- **station 4**
- **station 5**

Packet arrival at MAC
Example: RL-based Reservation and Routing

Reinforcement learning (RL) to learn from success of reservation requests

1. learn optimal routing strategies and
2. recover from link failures

- Local control actions (e.g. hop decisions)
- RL-Agent with adaptive policy
- Evaluation of routing quality (# of hops, QoS, …)
- Reinforcement

Example: Network Opt. based on Evolution Theory

Replacement and selection rely on some cost function defining the quality of each solution.

Crossover selection is typically random.

General parameters:
- size and meaning of population (each item represents part of the global solution)
- crossover operation and mutation probability
- local candidate selection strategy (mapping quality on probability)
- local replacement strategy (replace parents or neighbors, replace weakest)

Application-specific parameters:
- mapping of problem on appropriate coding
- handling of invalid solutions in codings
Example: Head Election based on Clustering

**k-means (not self-organized!)**

**Principles**
1. elect some given number of k cluster heads, one per cluster
2. each node associates itself with its nearest cluster head
3. for each cluster derived that way, select a new cluster head that is closest to the center (e.g., barycenter) of the cluster (hard to implement in distributed manner)
4. goto 2 until cluster-head decision is stable (or redo in dynamic systems)

**Application to select concentrator nodes (cluster heads) in tree networks**
Example: Head Election based on Clustering

LEACH: Low-Energy Adaptive Clustering Hierarchy

Application:
- ensure even distribution of power consumption of individual nodes in sensor networks

Assumptions
- each node can act as a regular node or as a cluster head
- regular nodes only communicate with their cluster head
- cluster heads supports data aggregation and data forwarding to neighboring cluster heads or base stations

LEACH represents a protocol supporting
- cluster head advertisements
- cluster set-up and
- steady-state communication
Example: Head Election based on Clustering

LEACH: Low-Energy Adaptive Clustering Hierarchy

Principles

- nodes elect themselves to become cluster heads at any given time with a certain probability (in HEED extension based on remaining energy)
- the cluster heads broadcast their status to the other nodes in the network
- each node determines to which cluster it wants to belong by choosing the closest cluster head (e.g. which requires the minimum communication energy)

![Diagram of LEACH clustering](image)
Example: Multi-Radio Load Balancing

Force-based selection of target cell
- Criteria/forces:
  - free capacity of target cell
  - supported QoS in target cell
  - migration penalty
  - handover cost

Problems: Predictability

Yield = density - loss

Fires don’t matter!

Density

Cold
Problems: Predictability

Everything burns!

Burned

Density
Problems: Predictability

Critical point
**Problems: Edge-of-Chaos**

Edge Of Chaos (EOC)
Self-Organized Criticality (SOC)

**Claims:**
- Interesting phenomena is at criticality (or near a bifurcation)
- Life, networks, the brain, the universe and everything are at “criticality” or the “edge of chaos.”
Lessons learned

Self-organization: what does it mean?
- distributed, autonomous control of systems
- simple local algorithms or behavior
- interaction with neighbors and/or environment only

Lots of applications of self-organization already
- self-organization keeps today’s Internet together
- development and tuning of self-organized algorithms is far from simple
- stability and predictability are important issues
Self-organization in general


Application of Self-organization to communication networks