Introduction and Overview

- Nokia Research overview
- Networking moving to distributed control
  - System design moving from centralized to distributed
  - Reduce cost of operation
  - Emerging pervasive networks
- 4 design paradigms for self-organization
  - Based on analysis of existing protocols
- Two examples in multi-hop wireless networks
  - Distributed resource control in multi-hop networks
  - Gateway discovery and multi-hop handover
- New challenges for emerging ubiquitous networks
  - From distributed to self-organized system design

Definitions of Self-organization

- Merriam-Webster:
  1. organization of oneself or itself
  2. the act or process of forming or joining an organization (as a labor union).

- Wikipedia: Self-organization is a process in which the internal organization of a system, normally an open system, increases in complexity without being guided or managed by an outside source. Self-organizing systems typically (though not always) display emergent properties.

- Chemistry: The capability of a system to spontaneously generate a well-defined supramolecular entity by self-assembling from components in a given set of conditions.

- Economics: A market economy is sometimes said to be self-organizing.

- "We define a self-organizing system as one where a collection of units coordinate with each other to form a system that adapts to achieve a goal more efficiently."

Self-organization in Networking

- Distributed control, i.e., no central or external control
- Entity
- System
- Only interaction with nearby entities, i.e., behavior is based on local information

- Simple behavior rules
- Create system-wide properties
  - Order
  - Adaptability
  - Robustness
  - Fairness

- Need to understand typical „design patterns“ of self-organization
- Need to understand scope – which functions to be self-organized
Summary of Paradigms and Examples

**Paradigms**
1. Achieve global properties with local behavior
2. Do not aim for perfect coordination
   - Implicit coordination
3. Minimize long-lived state
4. Adaptation

**Examples**
- Addressing and Naming
  - IPv6, ad-hoc networks
- Resource and congestion control and access
  - TCP
  - Medium Access (MAC) protocols
  - Distributed resource control in multi-hop networks
- Resource discovery
  - Proactive and reactive discovery
  - Multi-hop handover
- Data collection in pervasive networks

---

**Paradigm #1: Achieve Global Properties with Local Behavior**

- Behavior rules based on local environment
  - Don’t need to know about global state
  - No need for central control
  - System-wide property from local behavior

**Examples**
- **TCP protocol** for Internet congestion control
  - TCP is fully distributed, very simple, flexible
    - End-points control resources in the network
    - TCP achieves „fair“ resource distribution
- **Addressing** in the Internet – IPv6
  - Stateless autoconfiguration
  - Assign a globally unique address
  - Hosts configure address themselves
    - Global Address = Network prefix + Interface ID

---

**Diagrams**

1. Router Solicitation and Router Advertisement with Network prefix
   - Network prefix
   - Interface ID
2. TCP connection 1
   - Connection 1
   - Bottleneck router
3. TCP connection 2
   - Connection 2
From Local Behavior to Self-organization

Some typical questions

• Local behavior and control are important – but does it work?

• When does local behavior lead to optimal solutions?
  • But optimal solutions are difficult in dynamic networks

• Local behavior rules may lead to inconsistencies
  • When is this preferable or acceptable?

Example for addressing

• Passive autoconfiguration for ad-hoc networks
  • Assign addresses locally based on local information
  • Maintain list of assigned addresses locally
    • Passively collect information about already assigned & used addresses
    • Detect address conflicts in a timely manner locally

Paradigm #2: Do not Aim for Perfect Coordination:
Implicit Coordination

„Talking is silver, silence is golden”

• Robustness often more important than perfect or optimal solution

• When can we admit inconsistencies?
  • Need to be detectable or contained

• What is implicit coordination?
  • E.g. Listening and observing, randomization
  • No explicit signaling, no dedicated, central controllers

Examples

• TCP way of implicit communication: packet loss
• Local address assignment: Usage of passive duplicate address detection
• WLAN MAC protocols with Carrier Sense Multiple Access
• Distributed resource control in multi-hop networks
  • Listening and overhearing to minimize control messages
WLAN MAC Protocol with Implicit Communication

• How to split resources of a shared (wireless) medium?
• Carrier Sense Multiple Access / Collision Detection, (IEEE 802.11 CSMA / CD)
• “The CSMA/CD protocol functions somewhat like a dinner party in a dark room”
  • Charles Spurgeon
• E.g. A and C send data at the same time to B => Collision
  • A and C may not hear each other
• Collision events give information about status of the network
  • Capetanakis, 1979
• Random backoff after collision

---

DARE: Distributed Resource Control in Multi-hop Networks

**Goal:** Extend WLAN 802.11 MAC to make it suitable for
- Resource allocation for real-time applications
- Distributed control of multi-hop reservations

**Approach:** End-to-end reservation of resources (time slots)

**Protocol:** DARE
- Distributed allocation of time slots for real-time traffic end-to-end

---

E. Carlson, C. Prehofer, C. Bettstetter, H. Karl, A. Wolisz.
*A Distributed End-to-End Reservation Protocol for IEEE 802.11-based Wireless Mesh Networks.*
IEEE Journal on Selected Areas in Communications (JSAC), 2006.
DARE: Reservation Setup

- **Request-to-reserve**: Information about requested time slot [length and periodicity]
- **Clear-to-reserve**: Reserved resources on end-to-end path

Potential interferers are informed and do not transmit during time slots.

DARE: Real-time Data Transmission

- **Real-time data** [length and periodicity]
- **Implicit acknowledgement** by overhearing

No RTS/CTS needed (as in IEEE 802.11)
No explicit ACK needed (as in IEEE 802.11)
DARE: Protocol Design Summary

- Locally controlled end-to-end reservations
  - Nodes on reserved path maintain (soft) reservation state
  - Need to inform vicinity about reserved slots
  - Need to agree with other reservations in vicinity - first come first serve basis
- Continuously informing other nodes about reservations
  - “Biggy-packing” of information on other messages
  - How far to spread information about reserved slots – vs reuse of time slots
  - Especially difficult in case of mobility
- No central coordination or optimization of reservations and time slots

Performance Result: Dare vs EDCA (802.11 with Priorities)

- EDCA to give priority to real-time traffic
  - Retransmission timers, node queues
- DARE gives higher throughput and stable delays for higher load
  - DARE creates minimal signaling overhead

Evaluation Settings with NS2 Simulator

- 400 nodes in 2 x 2 km square, 230m range
- Random on-off of nodes
- IEEE 802.11 DCF and EDCA
- 10 Real-time flows with 40kb/s
- Background traffic with 20 or 50kb/s
- AODV routing protocol
**Paradigm #3: Minimise Long-lived State Information**

- Reduce state which is *synchronized* between two or more nodes

**Example**

- Discovery of gateways in multi-hop access network
  - Self-organizing ad-hoc network(s) connected to Internet via gateways (GW)
  - Nodes discovery optimal gateway and obtain global internet connectivity
- Comparison of proactive and reactive discovery
- Proactive discovery
  - Gateways send periodic announcement
  - Nodes select best gateway

![Diagram of Mobile Network with ad-hoc routing](image1)

**Reactive Discovery of Gateways**

- Reactive Discovery
  - Node sends announcement
  - Gateways reply
  - Nodes select best gateway
- Hybrid discovery option
  - Gateways in proactive mode
  - Nodes only proactive if gateway known

![Diagram of Ad Hoc Network](image2)
Gateway Selection & Handover with Mobile Nodes

- MRAN protocol for gateway discovery/selection
  - Autoconfiguration of global address
  - Packet forwarding to/from fixed network

- Gateway selection strategies and handover
  - Proactive and reactive gateway selection
  - Forced handovers to gateway if old gateway lost
  - Optimizing handovers to closer gateway

- Testbed implementation
  - Protocol implementation in Linux
  - 13 nodes connected via WLAN
  - Mobility emulated (via packet filtering)
  - Different ad-hoc routing protocols (AODV, OLSR)

P. Hoffmann, C. Bettstetter, C. Prehofer, 
Performance Impact of Multihop Handovers in an IP-based 
Multihop Radio Access Network, ACM SIGMOBILE Mobile Computing 

Reactive vs Proactive Gateway Discovery

Evaluation Results

- Little impact on data delivery fraction
- Proactive discovery performs better for higher mobility

Parameters for performance evaluation
- Number of mobile nodes 10
- Number of gateways 3
- Constant bit rate of each MN 10 kbit/s
- Measurement period 30 min
- Number of experiments 10
Multi-hop Handover Optimization

- Optimizing handovers faster than forced ones
  - Optimizing ~ 5ms
  - Forced ~ 35 ms
- Optimizing handovers increase number of handovers
- Optimizing handovers improve system performance

![Graphs showing handover rates and delivery fractions with and without optimization]

Locality and State

- TCP congestion control
- Unsolicited service advertisements
- Address assignment with DHCP
- Time-division multiple access (TDMA)
- Clustering in ad hoc networks
- CSMA with collision avoidance (CSMA/CA)
- CSMA with collision detection (CSMA/CD)
- Reactive ad hoc routing
- Swarm forming
- Carrier sense multiple access (CSMA), ALOHA
- Passive duplicate address detection

Observation based, broadcast, broadcast, implicit, notification, single reply, 2 party coordination, n party coordination
Paradigm #4: Adaptation

„Adapt or perish, now as ever, is nature's inexorable imperative“ H. G. Wells

- Need to adapt when local/implicit mechanisms do not suffice
  - Monitor and adapt – often implicit communication

Levels of Adapation

- **Level 1**: A protocol is designed so that it adapts continuously (e.g. failure, mobility)
  - Adapt parameters continuously
  - E.g. TCP transmission window control adapts to round trip time (RTT)

- **Level 2**: A protocol is designed to adapt to specific events and conditions
  - Adapt parameters (e.g. value of timers, cluster size) in order to optimize system performance
  - E.g. TCP reduces sending rate upon congestion

- **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
  - Major reaction and change of behavior
  - E.g. TCP time out upon consecutive packet loss

Summary of Design Paradigms & Dependencies

What makes self-organization difficult in practice?

1. Achieve global properties with local behavior
   - E.g. resource control, addressing
   - Local control

2. Do not aim for perfect coordination
   - Permit local inconsistencies if detectable or tolerable
   - Implicit coordination (by detecting inconsistencies)

3. Minimize long-lived state
   - E.g. discovery instead of fixed bindings

4. Adaptation
   - Detect when employed mechanisms are at their limit
From Networking to Pervasive Computing & Services

How to design pervasive applications?
- Where to maintain state information?
- What level of control can we get?
- How to adapt to changing environment?
  - Robustness is key
- Hope to cope with heterogeneous environment?
  - Devices, protocols, radio
  - E.g. short range of RFID tags

Example: Opportunistic data collection
- Can we use the same principles for pervasive computing?

Local book finder example:
- All books in a building have RFID tags
- Phones in the building record books in vicinity
- Phones exchange data when people move
  - Optimized data storage and data exchange with network coding
  - Requests can be sent to local devices, e.g. “Find a Java book close by”
Summary and Outlook

Design principles of self-organization
- Identification of main paradigms of self-organized networks
- Self-organization leads to new research approaches and system design
- Towards “design patterns” and methodology for self-organized systems

Related trends & opportunities
- Context-aware pervasive systems
  - How to manage context information?
- Web services & service oriented architecture for pervasive systems
  - Stateless services and “document-oriented computing”
- Self-organized security mechanisms
  - Bio-inspired anti-body generation for mal-ware detection