An Optimization Approach for the Network Synthesis of Distributed Embedded Systems

Alex Malfatti, Davide Quaglia
Outline

• Context & background
• Current limitations
• Methodology
• Experimental results
• Conclusions & Future developments
Context & Background
Distributed Embedded Systems (DESs)

- Distributed applications of Networked Embedded Systems (NESs) interacting together
  - Example: Distributed control of building temperature

- Different types of channels and protocols

- Each NES acts as a node of the network

- New design goal
  - Good behavior of the global application
Network Synthesis

• Design process starting from a high-level specification of DES
• It finds the actual configuration in terms of
  – mapping of application tasks onto network nodes
  – their spatial displacement
  – the type of channels and protocols among them, and the network topology
CASSE (1)

- Communication Aware Specification and Synthesis Environment (CASSE), is an extended design flow, which addresses the network synthesis, in terms of nodes, tasks, data flows, abstract channels, zones and contiguities.
  - **Tasks**
    - A task represents a basic functionality of the whole application; it takes some data as input and provides some output.
    - *Relevant attributes*: computational requirements, mobility.
  - **Data flows**
    - A data flow (DF) represents communication between two tasks; output from the source task is delivered as input for the destination task.
    - *Relevant attributes*: communication requirements.
  - **Nodes**
    - A node can be seen as a container of tasks.
    - *Relevant attributes*: available computational resources, intrinsic power consumption, power consumption due to tasks, mobility, economic cost.
CASSE (2)

– Abstract Channels
  • An abstract channel (AC) interconnects two or more nodes.
  • Relevant attributes: available communication resources, economic cost, wireless/wired.

– Zones
  • A zone is a partition of the space which contains nodes; each zone is characterized by environmental attributes which are application-specific.

– Contiguities
  • Zones are related by the notion of contiguity defined as follows:
    – Two zones are contiguous if nodes belonging to them can communicate each other.
    – Contiguity represents not only the physical distance between zone, but it can be used also to model environmental obstacles like walls.
  • Relevant attributes: resistance.
UML Deployment Diagram
Purely analytical Optimization

• Process that explores the solutions using techniques unaware of the network context.
• Formulation of the network optimization problem in the form of MILP (Mixed Integer Linear Programming) problem, whose optimization techniques are well known in literature (e.g. Simplex method, Branch-and-bound method).
Current limitations

• The increase of the complexity of distributed applications
  – Computer-aided design for the communication infrastructure between nodes of a DES

• Low scalability of pure analytical approaches (e.g., MILP) to the size of a real application
  – Analytical modeling could be combined with **Network Manipulation** driven by an optimization process

• Gap between the ideal model of the network, and the real network
  – Mixed analytical and simulation-based methodology is needed
Methodology
Contributions

• NW-aware approach for the optimal Network Synthesis of DES.
• Methodology to manipulate the DES description from a high level specification to simulation.
• Definition of manipulation rules to alter the network setup according to given optimization goals.
• Use of the network simulation to validate the optimization results and explore other possible solutions.
Flow for optimal Network Synthesis

- Network scenario high-level description
- Optimization
  - Economic cost
  - Power consumption
  - Transmission delay
  - Error rate
- Optimization Strategies
  - Move task
  - Move data flow
  - Remove data flow
  - Add node
  - Remove node
  - Move channel
  - Add channel
  - Remove channel
- Manipulation

Simulation-based refinement
- Optimized Network scenario simulation model

Network scenario simulation model
Optimization problem (1)

- nodes N0, N1, N2, N3
- tasks T0, T1, T2, T3
- channels C0, C1
- data flows D0, D1
Optimization problem (2)

PROBLEMS:

• How to split this transformations into elementary steps?
• How to link elementary transformations to optimization goals?
NW-Aware Optimization

**Optimization objectives**
- Economic cost minimization
- Power consumption minimization
- Transmission delay minimization
- Error rate minimization

**Optimization strategies**
- Nodes removal
- Channels removal
- Tasks reallocation
- Data flows reallocation

**Manipulation rules**
- Move task
- Move data flow
- Remove data flow
- Add node
- Remove node
- Add channel
- Move channel
- Remove channel
Simulated Annealing

• Generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function.

• Aims to find a global optimum when many local optima are present.

• Often used when the search space is large and also discrete.
Ideal model vs. Actual behavior (1)

For each $i$-th data flow we define its throughput as:

$$Th(D_i)$$

Ideally, the used capacity of channel $C$ should be:

$$\sum_{i=1}^{n} Th(D_i)$$

NOT ALWAYS TRUE!  
e.g., overhead of the wireless protocols
Non-idealities

• In the Network-aware Optimization process we make some ideality assumptions which often don’t correspond to reality.
• For this reason we make a list of non-ideality factors that we are not able to take fully into account in the Network-aware optimization approach:
  – from the point of view of the Network
  – from the point of view of the Nodes
  – from the point of view of the Power consumption
  – from the point of view of the HW Architecture
Ideal model vs. Actual behavior (2)

- **Objective metric**
- **Design space**
- **Final actual max**
- **Initial ideal max**
- **Local search by manipulation**

- **Ideal model (analytical)**
- **Realistic model**
Simulation

• To take into account the non-ideality factors in the optimization process, we are going to use the simulation applied to:
  – Points close to the optimal obtained with the manipulations
    • Neighbors in the solution space
  – Points close to the optimal obtained with the manipulations
    • Neighbors in the ranking of the best solutions found

• Network Simulation lets us to refine the choice of candidate solutions for those objectives that operate on parameters verifiable through simulation (e.g. delay, error rate).
Simulation-based refinement

Considering neighbors of the optimum ideal max in

1. the space of solutions
2. the ranking (with respect to the optimization metric)
Toolchain

- Papyrus
  - UML Deployment diagram
  - uml2rad

- RadCASE
  - RadCASE design
  - rad2hif + univercm2hdl

- HIF
  - customize4scnsl
  - Network Manipulation
  - ns4opt

- SCNSL
  - Network Simulation
  - hif2sc

- XML
  - Manipulation rules to apply

- Scilab
  - Network Optimizer

- TECHNOLOGICAL LIBRARY
Toolchain

- Papyrus
  - UML Deployment diagram
    - uml2rad

- RadCASE
  - RadCASE design

- HIF
  - HIF
    - rad2hif + univercm2hdl
    - hif2sc

- SCNSL
  - Network Simulation

- Scilab

- XML
  - Manipulation rules to apply

- Network Optimizer

- TECHNOCLOGICAL LIBRARY
  - ns4opt
Experimental results
NW-Aware Optimization

**Optimization objectives**
- Economic cost minimization
- Power consumption minimization
- Transmission delay minimization
- Error rate minimization

**Optimization strategies**
- Nodes removal
- Channels removal
- Tasks reallocation
- Data flows reallocation

**Manipulation rules**
- Move task
- Move data flow
- Remove data flow
- Add node
- Remove node
- Add channel
- Move channel
- Remove channel

26
Transmission delay minimization

**Optimization objectives**
- Economic cost minimization
- Power consumption minimization
- Transmission delay minimization
- Error rate minimization

**Optimization strategies**
- Nodes removal
- Channels removal
- Tasks reallocation
- Data flows reallocation

**Manipulation rules**
- Move task
- Move data flow
- Remove data flow
- Add node
- Remove node
- Add channel
- Move channel
- Remove channel
Experimental results (before)

Delay = 4 ms

\[ d = 2 \]
Experimental results (after)

Delay = 2 ms
Validation through simulation

Network scenario high-level description

Optimization
- Transmission delay

Optimization Strategies
- Move task
- Move data flow
- Remove data flow
- Remove node
- Move channel
- Add channel

Manipulation

Optimized
Network scenario simulation model

Network scenario simulation model

800 us 10a data sent: A bytes: 1
800 us 10b data sent: B bytes: 1
2800 us T2aa data received: A bytes: 1
2800 us T2ab data received: B bytes: 1

800 us 10a data sent: A bytes: 1
800 us 10b data sent: B bytes: 1
1800 us T2aa data received: A bytes: 1
1800 us T2ab data received: B bytes: 1
Conclusions & Future developments
Conclusions

• New methodology for the optimal network synthesis of Distributed Embedded Systems.
• Validation of the optimization process results through network simulation.
• Set of tools in support of the optimization process from the UML description to the SCNSL simulation model.

Future developments

• Extension of the set of optimization objectives
  – Introduction of new optimization strategies
  – Introduction of new manipulation rules
• Further integration of the network simulation in the optimization process.
• Adaptation of the methodology for a Multi-objective optimization.