ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

TIK
Computer Engineering and
Networks Laboratory

Dynamic and Adaptive Allocation of Applications on MPSoC Platforms

Andreas Schranzhofer, Jian-Jia Chen, Luca Santinelli, Lothar Thiele

Introduction

- Multiprocessor System-on-Chips:
  - Performance, Cost, Flexibility
- Adaptivity
  - Adapt to altering environmental conditions
- Applications:
  - Multiple concurrent applications
  - Multimedia Devices (HDTV, SDR,..)
  - Avionics, automotive applications,….

Problem Statement

- Library of Processing Unit (PU) Types
- Multiple Applications
  - mutual exclusive execution modes
  - execution probabilities (varying)

UNCERTAINTY

- Architecture and Task Allocation
  - PU Types and instances
  - pinning of tasks is allowed
  (tasks cannot migrate)

DEGREE OF FREEDOM

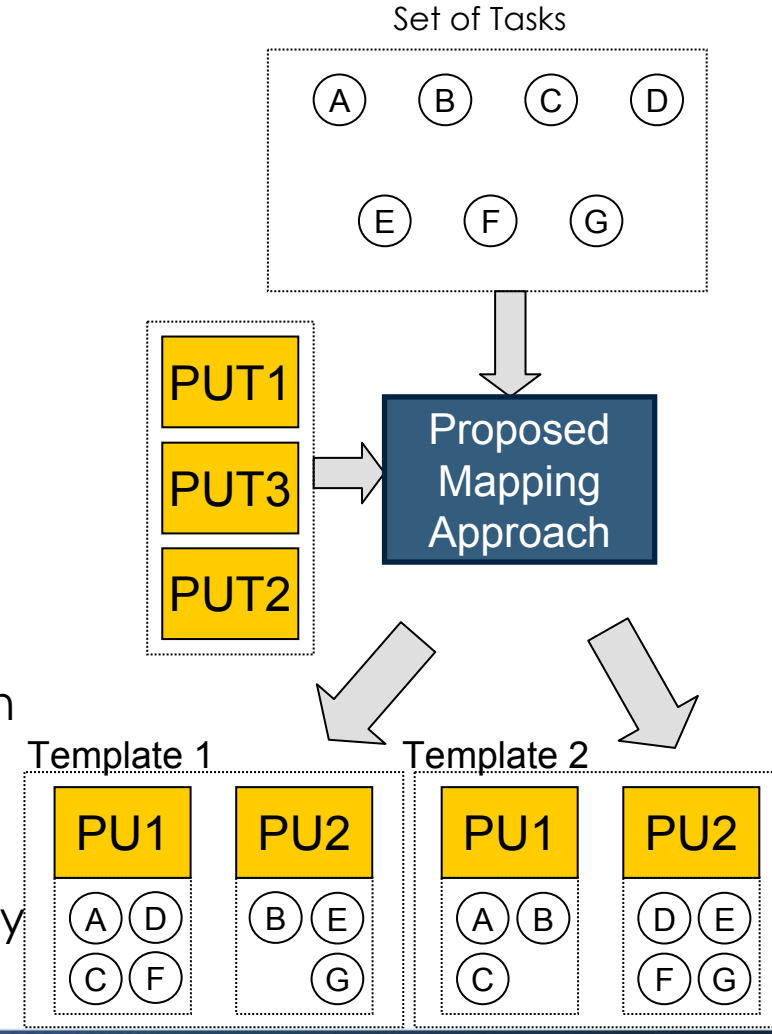
- Constraints
  - utilization bound
  - expected average power consumption

PREDICTABILITY

EFFICIENCY

- Dynamic mapping
  - Monitor execution pattern
  - Choose template mapping dynamically

ADAPTIVITY



Related Work

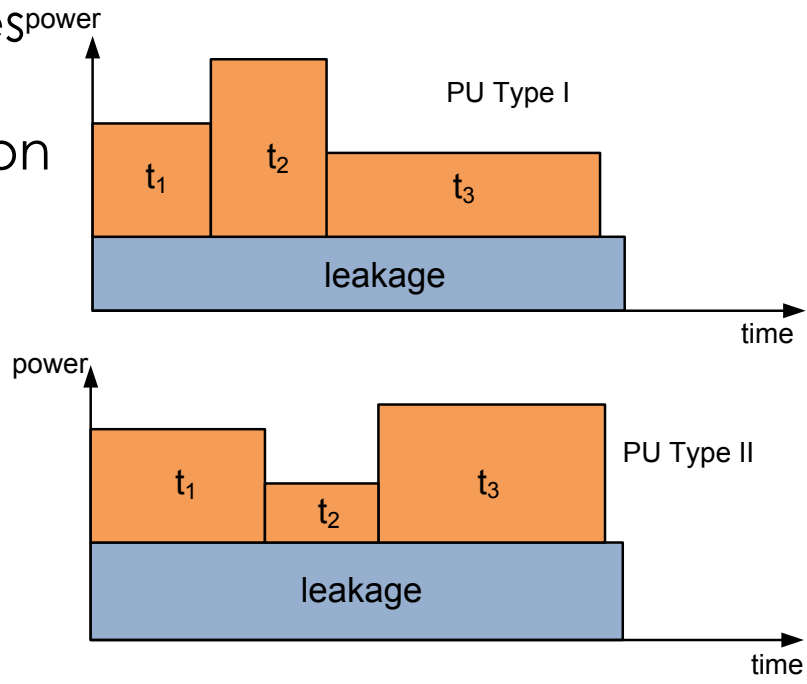
- [M. Schmitz et al. – TCAD 2005]
  - dynamic and static power consumption
  - execution probabilities for modes
  - genetic algorithm to compute mapping
- [Moreira et al. – SAC 2007]
  - online heuristic to assign tasks to processing element
  - no schedulability guarantee
  - evaluation using success rate
- [Benini et al. – ICLP 2008]
  - compute system configurations
  - transition between allocations at run-time
  - migration cost for task reallocation

Proposed Methodology - Overview

- Hardware/Application Model
  - dynamic / static power model
  - multiple concurrent multi-mode applications
- Application Model and Scenarios
- Task Allocation for Scenario Sequences
  - Mapping for each Scenario Sequence – *Templates*
- Online Mapping
- Experimental Results

Hardware Specification

- Heterogeneous MPSoC
  - Multiple Processing elements
  - Different types of Processing elements
- Processing Unit
  - Available computational resources **[cycles/time unit]**
  - Static/Leakage power consumption **[power unit]**
  - Dynamic power consumption **[power unit/resource demand]**
    - dependent on the tasks utilization
  - No dynamic power management

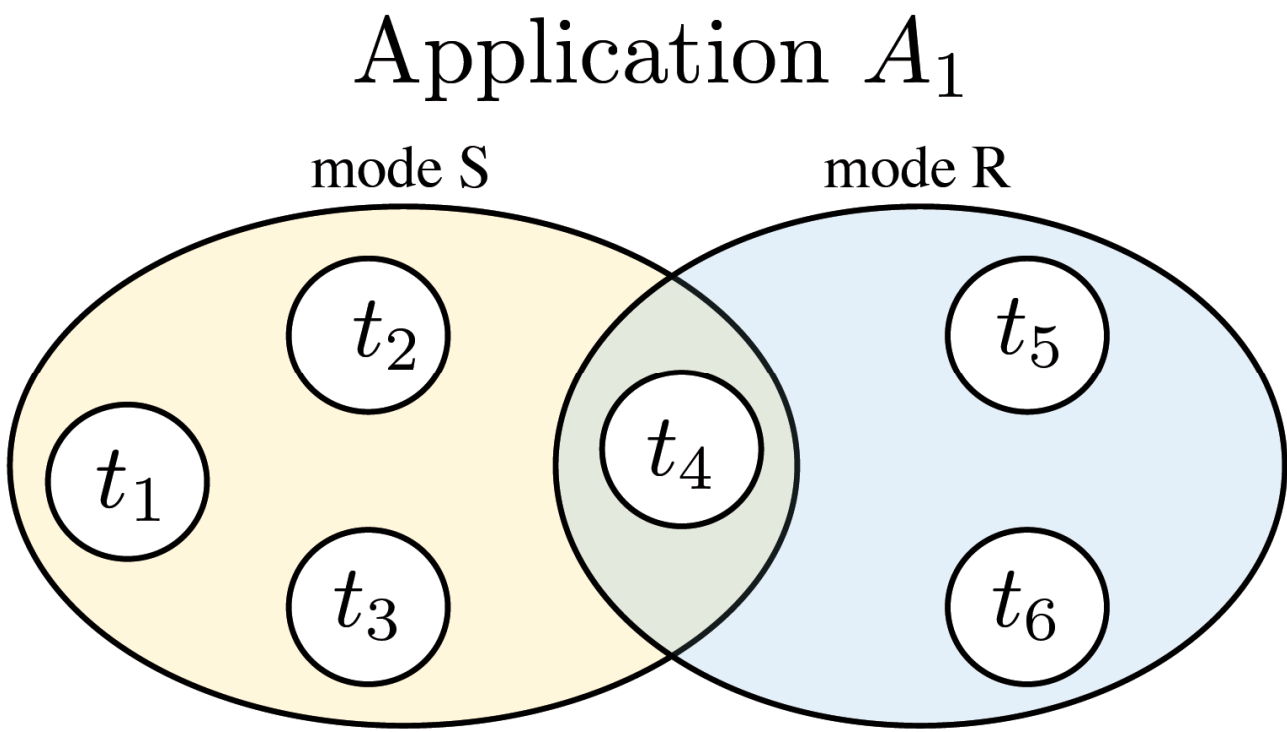


Application Model

Tasks as the basic computational unit

modes are annotated with probabilities

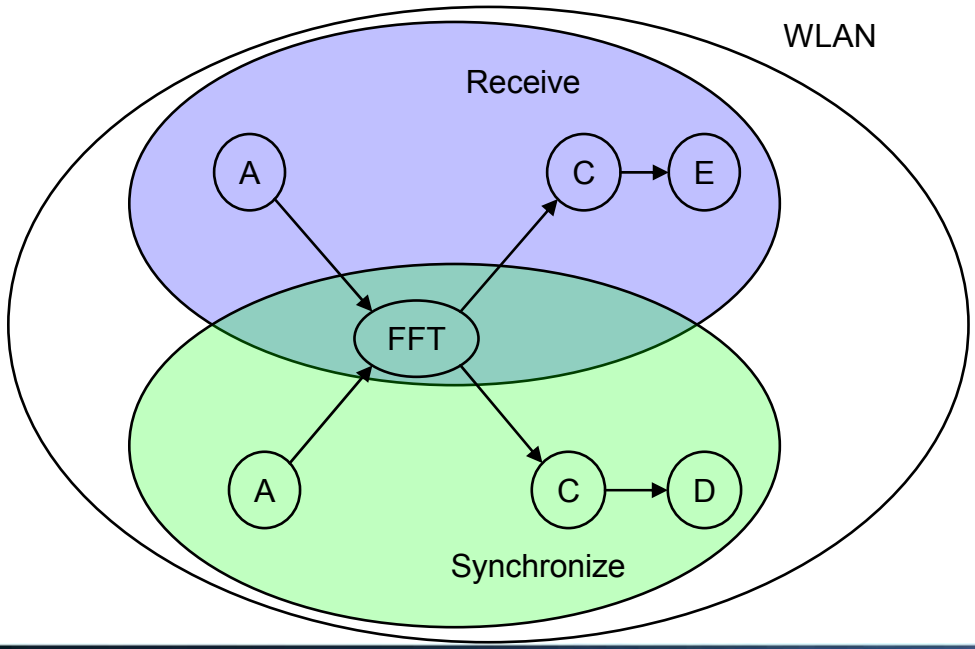
modes may share tasks

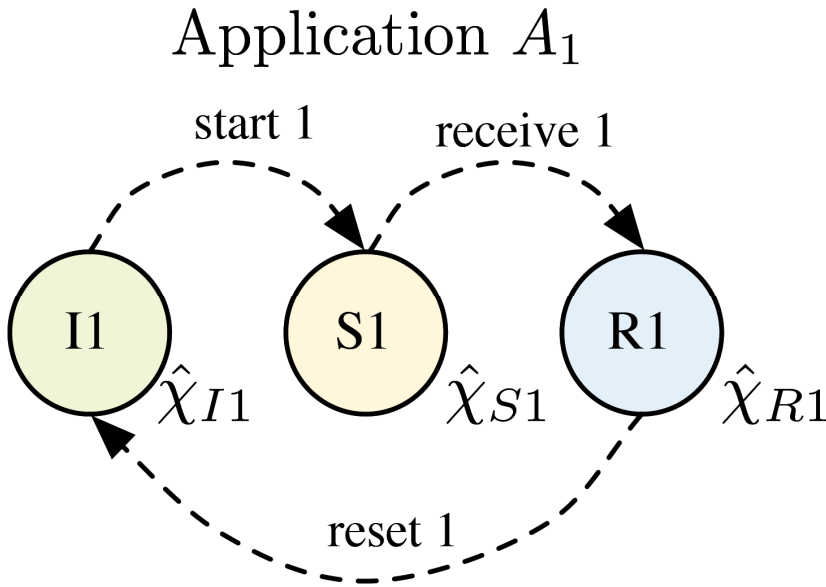


modes cannot execute concurrently

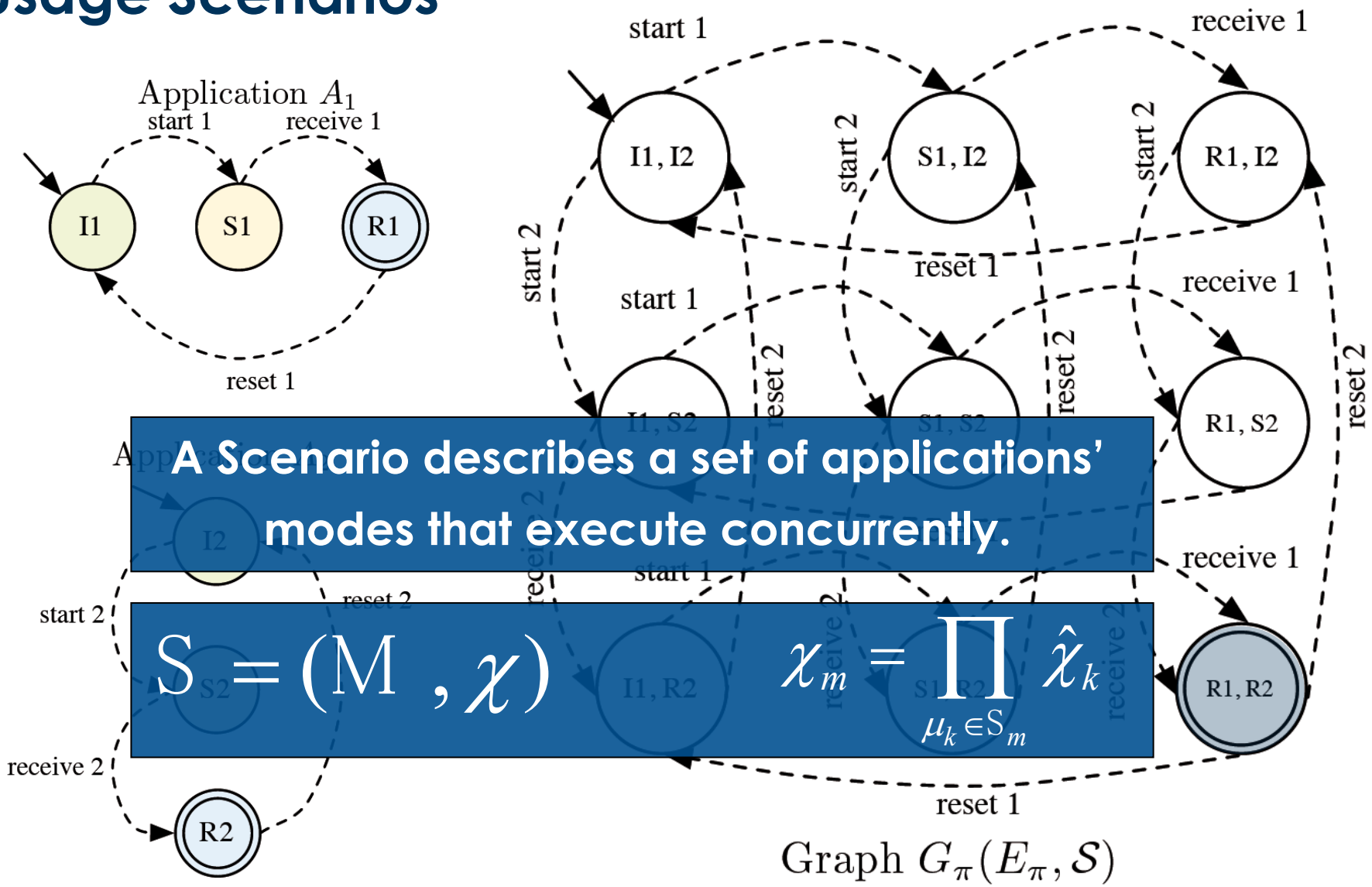
Motivational Example

- Example WLAN:
  - Modes have probabilities
    - Avg. 10% of time synchronizing
    - Avg. 70% of time receiving
    - Avg. 20% of time idling
  - 1 shared task (FFT)





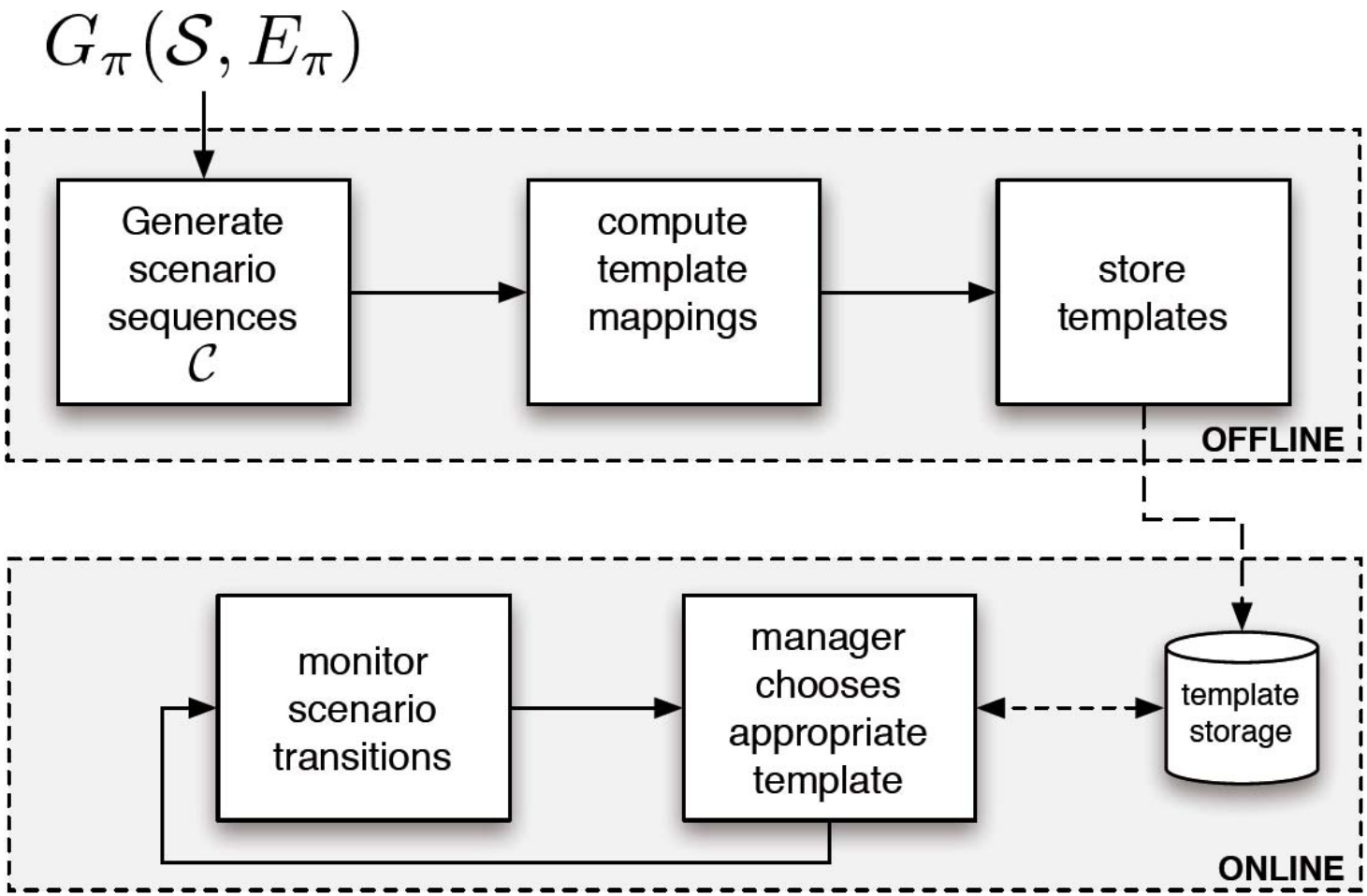
Usage Scenarios



A Scenario describes a set of applications' modes that execute concurrently.

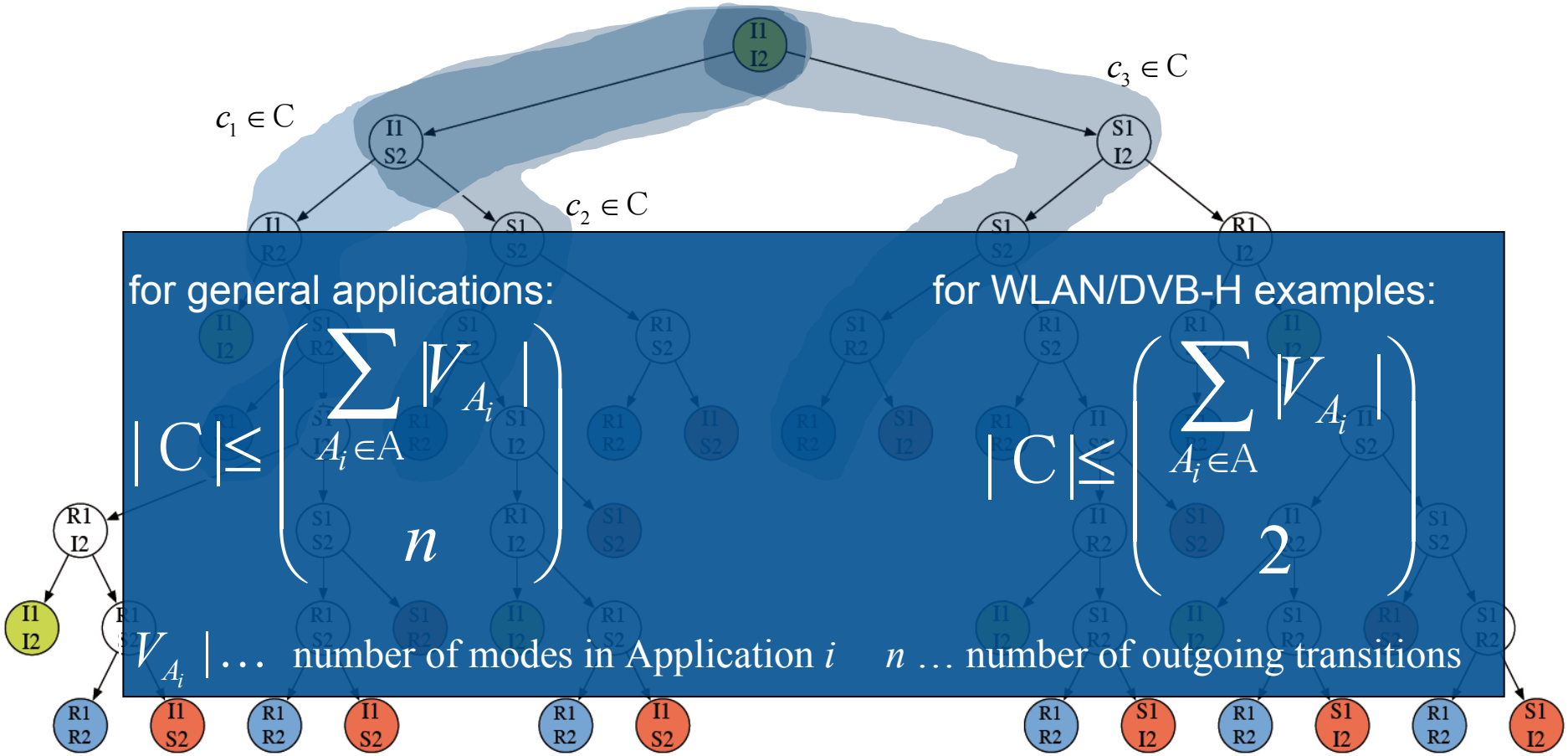
$$S = (M, \chi) \qquad \chi_m = \prod_{\mu_k \in S_m} \hat{\chi}_k$$

Methodology Overview

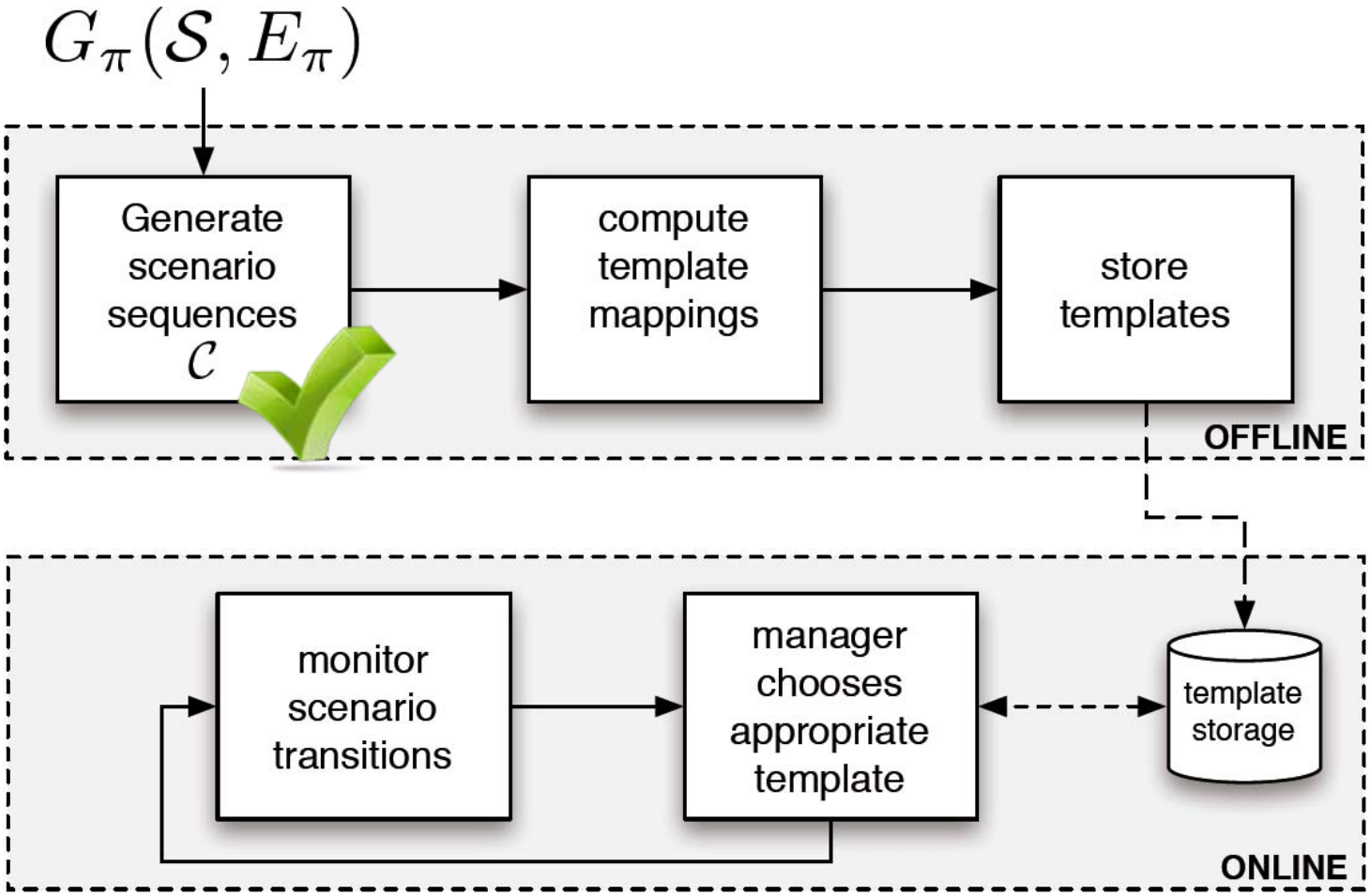


Scenario Sequences

- feasible sequence of loop free scenario transitions



Methodology Overview



Mapping for a single Scenario Sequence

- gather active tasks in Scenario Sequence $c_1 \in C$
  - calculate the tasks execution probabilities
  - static mapping using the approach shown in RTAS 2009
- compute Hardware Platform
- Store mapping as template on the system
- consider different execution probabilities

Task/PU Allocation

min.

leakage power

dynamic power

$$\sum_{S_m \in c_i} \sum_{p_j \in \mathrm{P}} \sum_{k=1}^{F_j} \chi_m \sigma_j Z_{m,j,k} + \sum_{t_i \in \mathrm{T}_m} \sum_{p_j \in \mathrm{P}} \sum_{k=1}^{F_j} u_{i,j} \delta_j \psi_i M_{i,j,k}$$

s.t.

$$\sum_{t_i \in \mathrm{T}_{new}} M_{i,j,k} u_{i,j} \le Z_{m,j,k}, \quad \forall p_j \in \mathrm{P}, \forall k = 1,2,\ldots,F_j$$

utilization constraint

$$\sum_{p_j \in P} \sum_{k=1}^{F_j} M_{i,j,k} = 1, \forall t_i \in \mathrm{T}_m,$$

$$M_{i,j,k} = \{0,1\} \forall t_i \in \mathrm{T}_{c_i}, p_j \in \mathrm{P}, k = 1,2,\ldots,F_j,$$

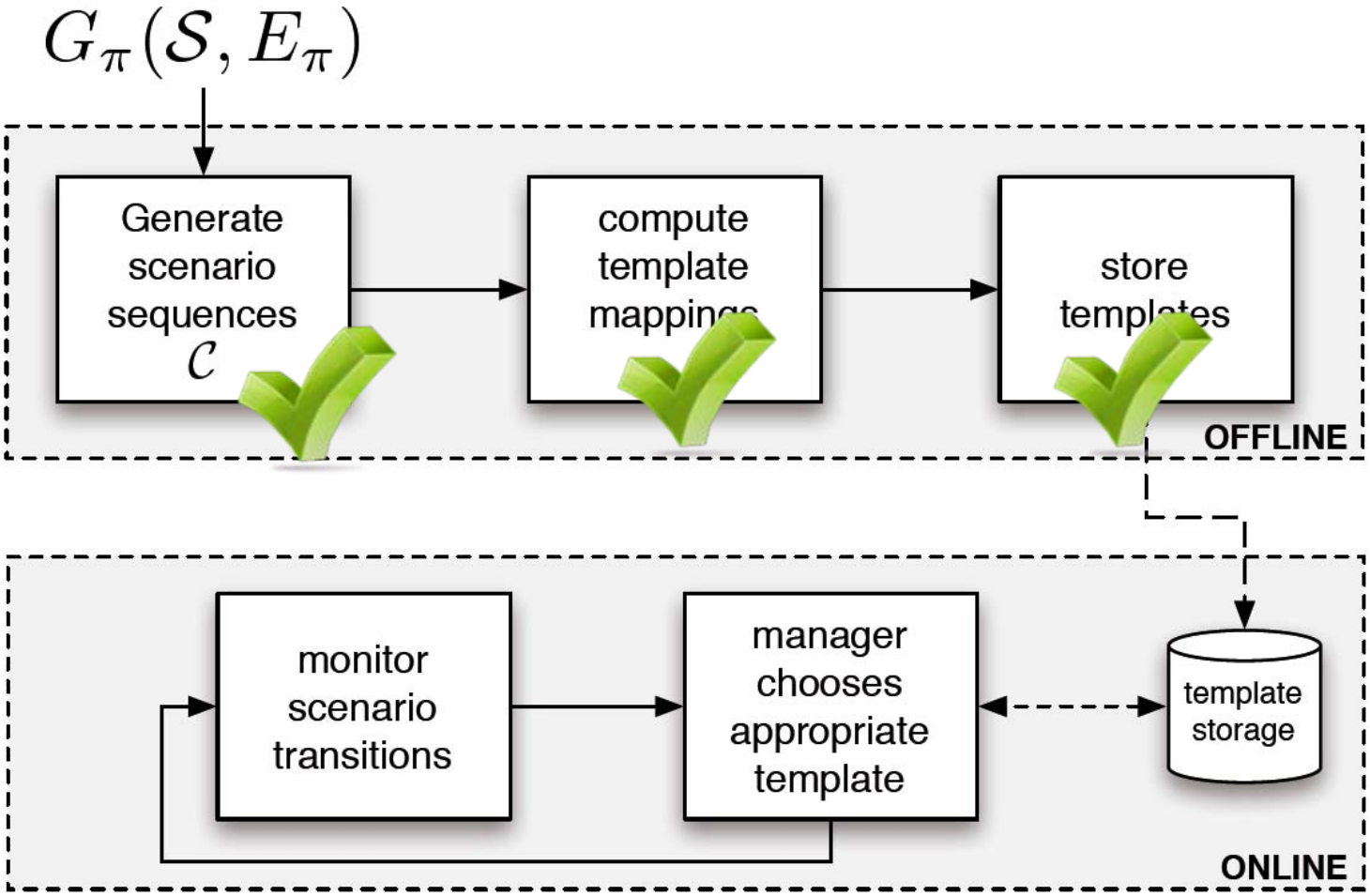
$$Z_{m,j,k} = \{0,1\}, p_j \in \mathrm{P}, k = 1,2,\ldots,F_j$$

(RTAS 2009)

Mapping for a single scenario sequence

- Optimal Solution using ILP
  - NP-hard
  - no approximation with constant factor in polynomial time
- Initial Mapping
  - assign task to most efficient processing element
- Remapping of tasks
  - save leakage power, by increasing average utilization
- For details: ***Power-Aware Mapping of Probabilistic Applications onto Heterogeneous MPSoC Platforms***, RTAS 2009

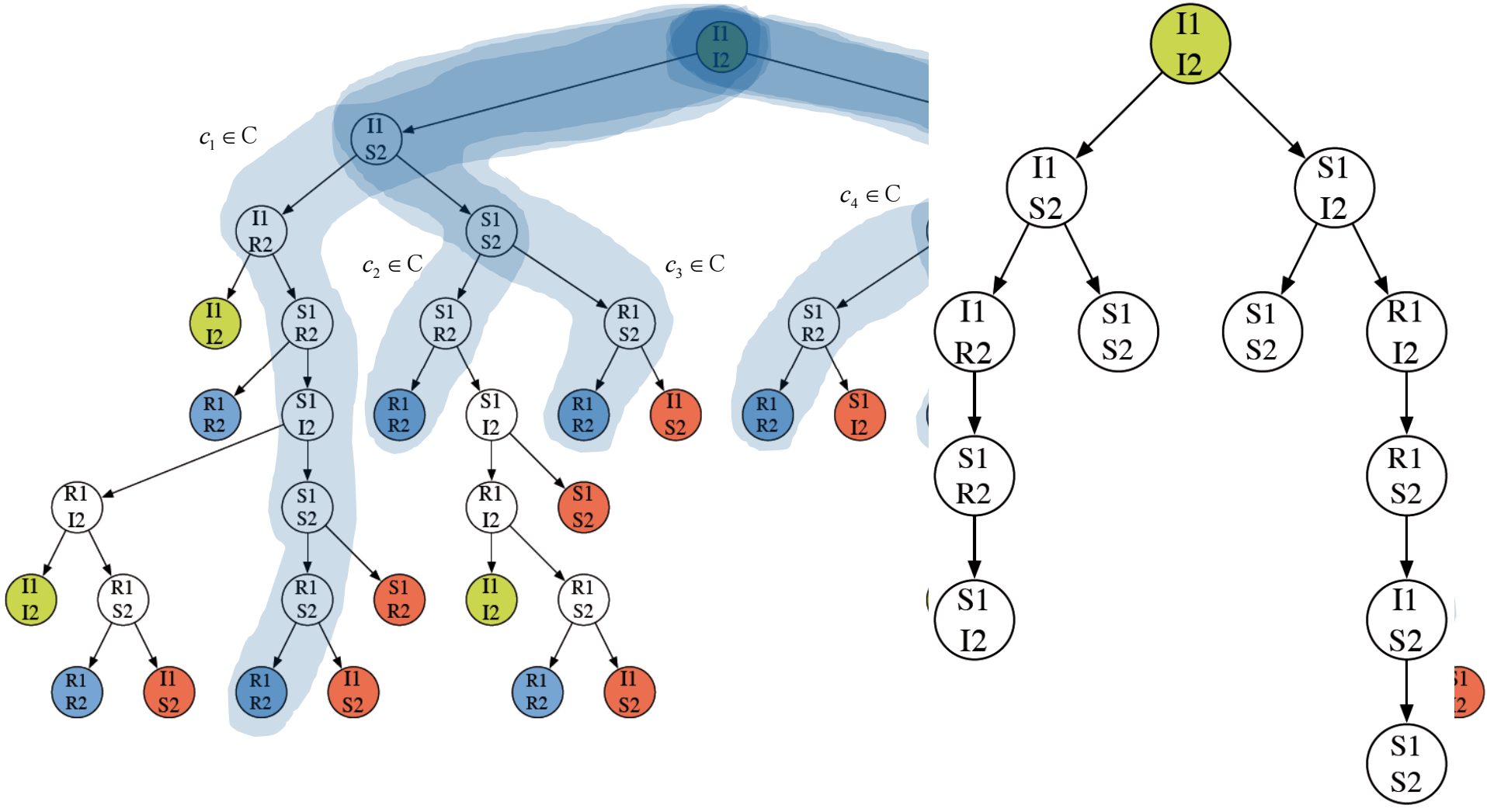
Methodology Overview



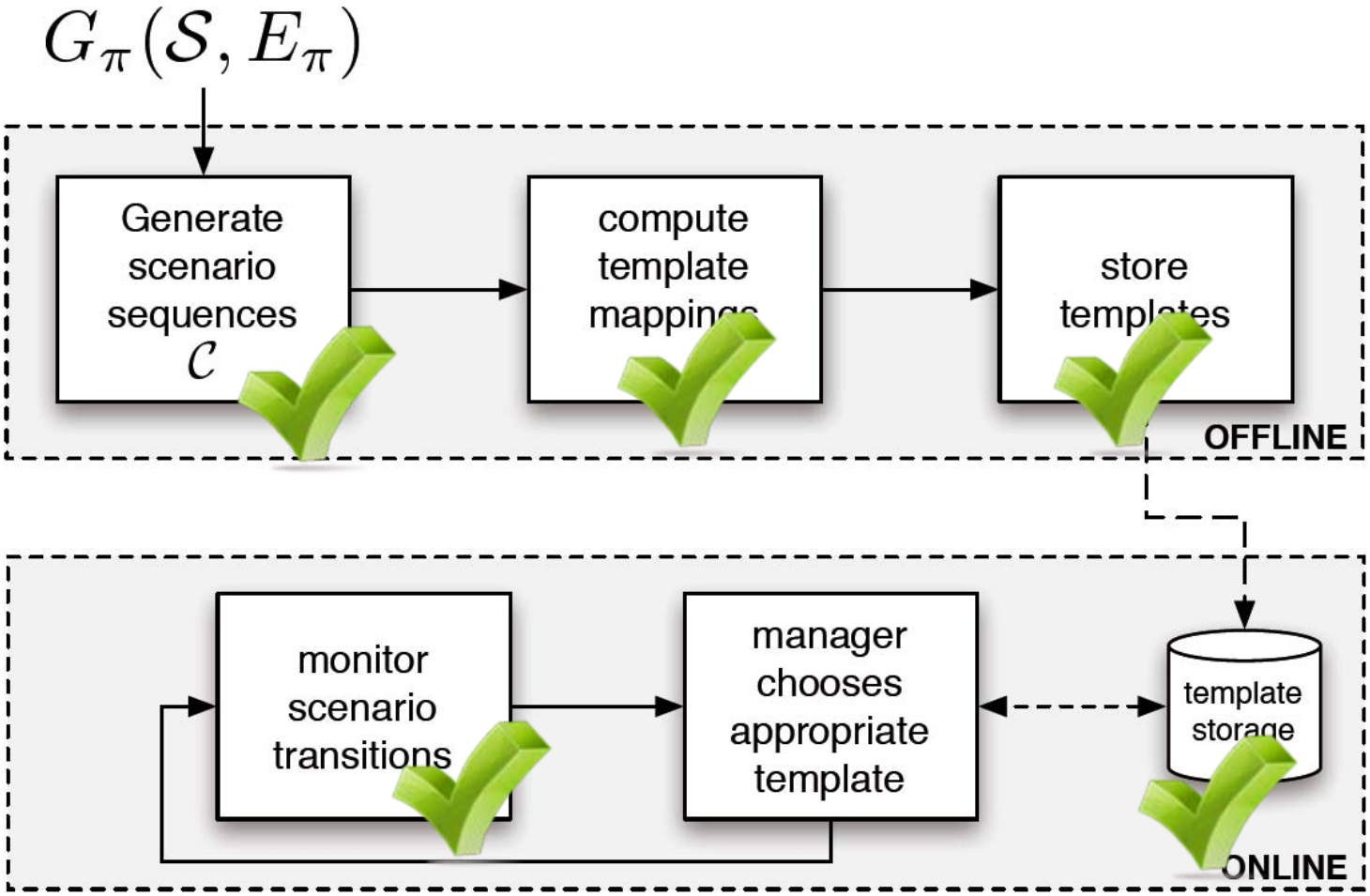
Dynamic and Adaptive Mapping

- template mappings are stored on the systems
  - for all feasible loop-free scenario sequences
  - for different representative execution probabilities of modes
- at run-time a manager monitors:
  - the scenario transitions
  - the scenarios execution pattern
- an appropriate template is applied

Online Mapping - Example



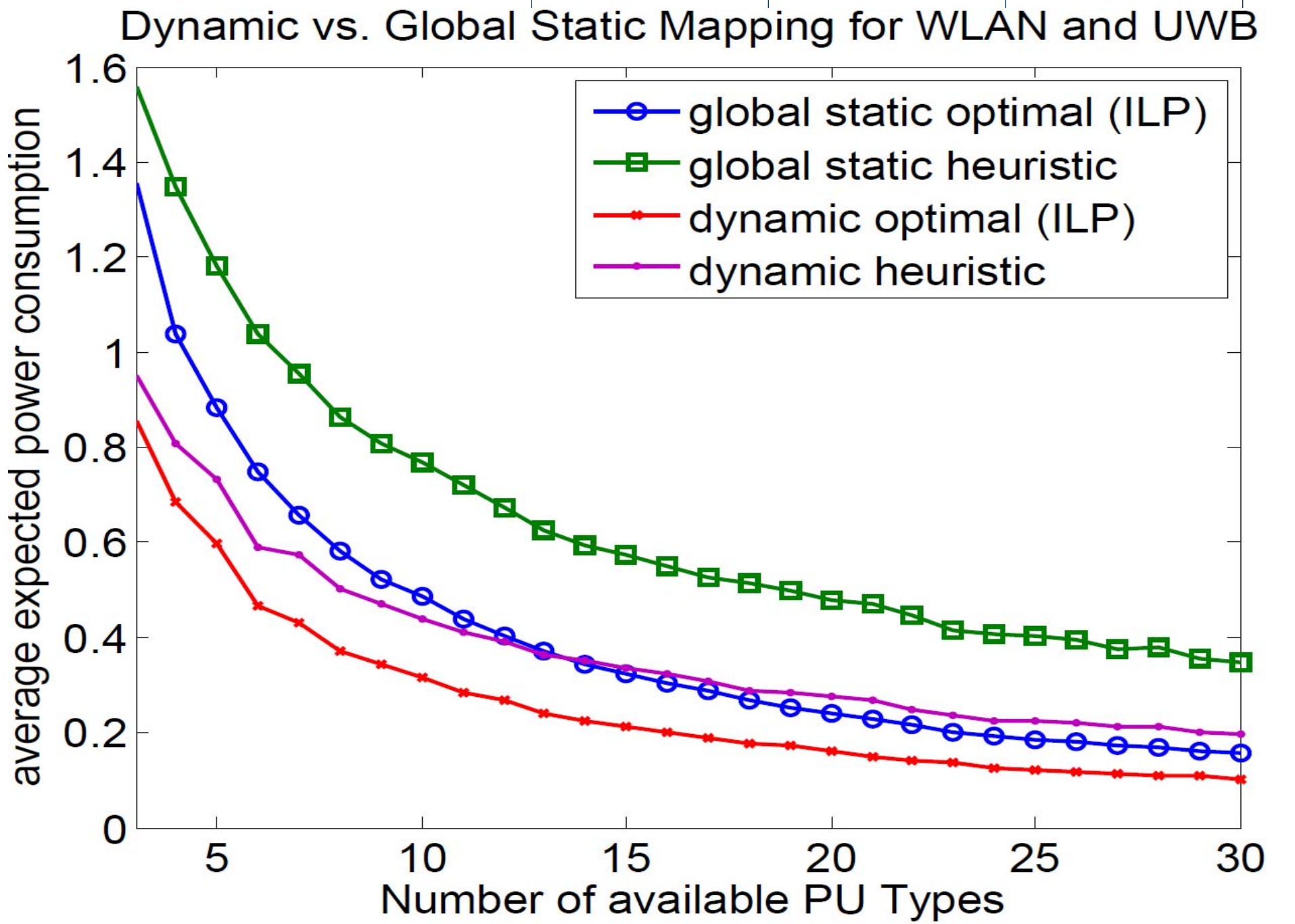
Methodology Overview

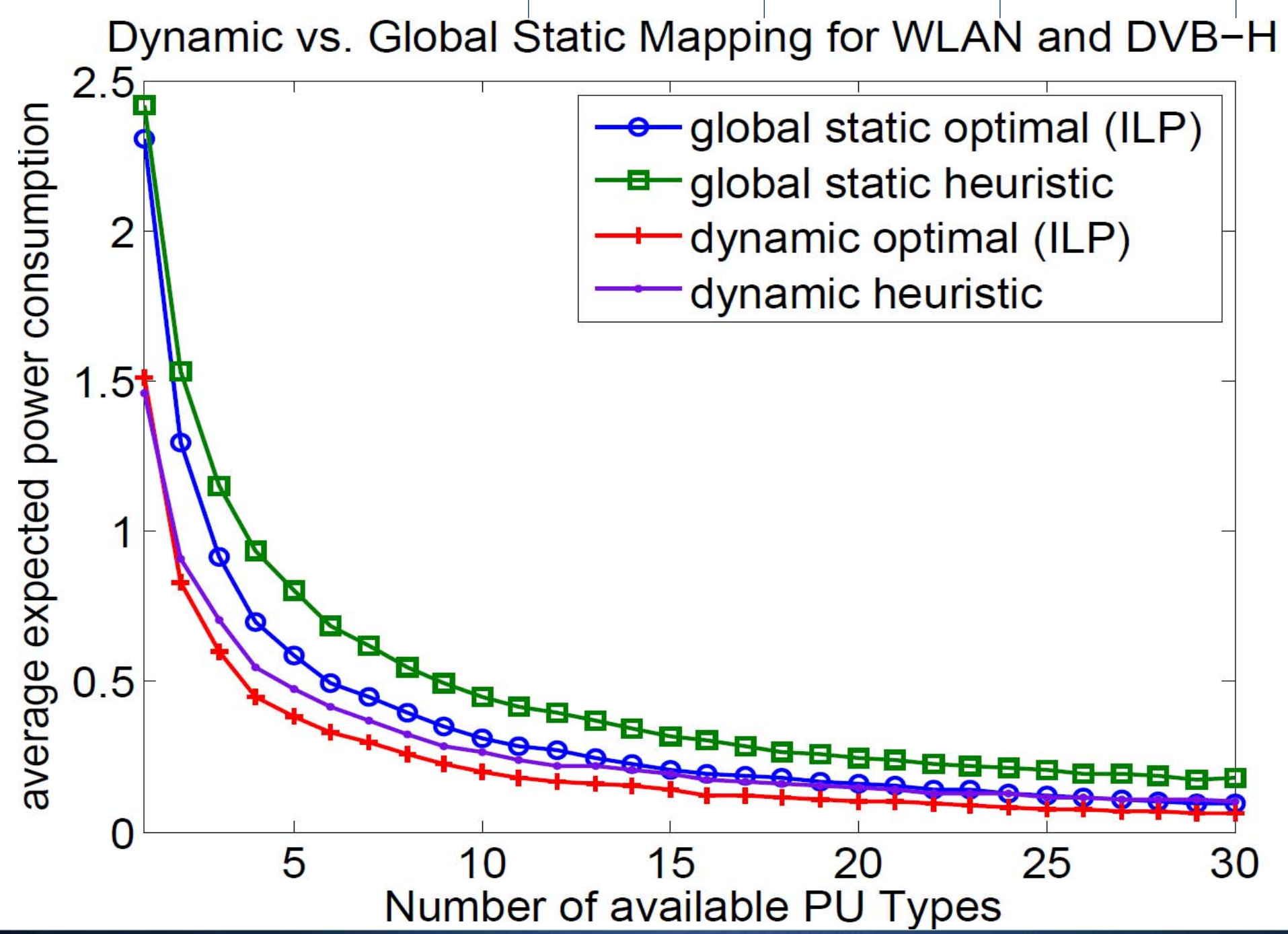


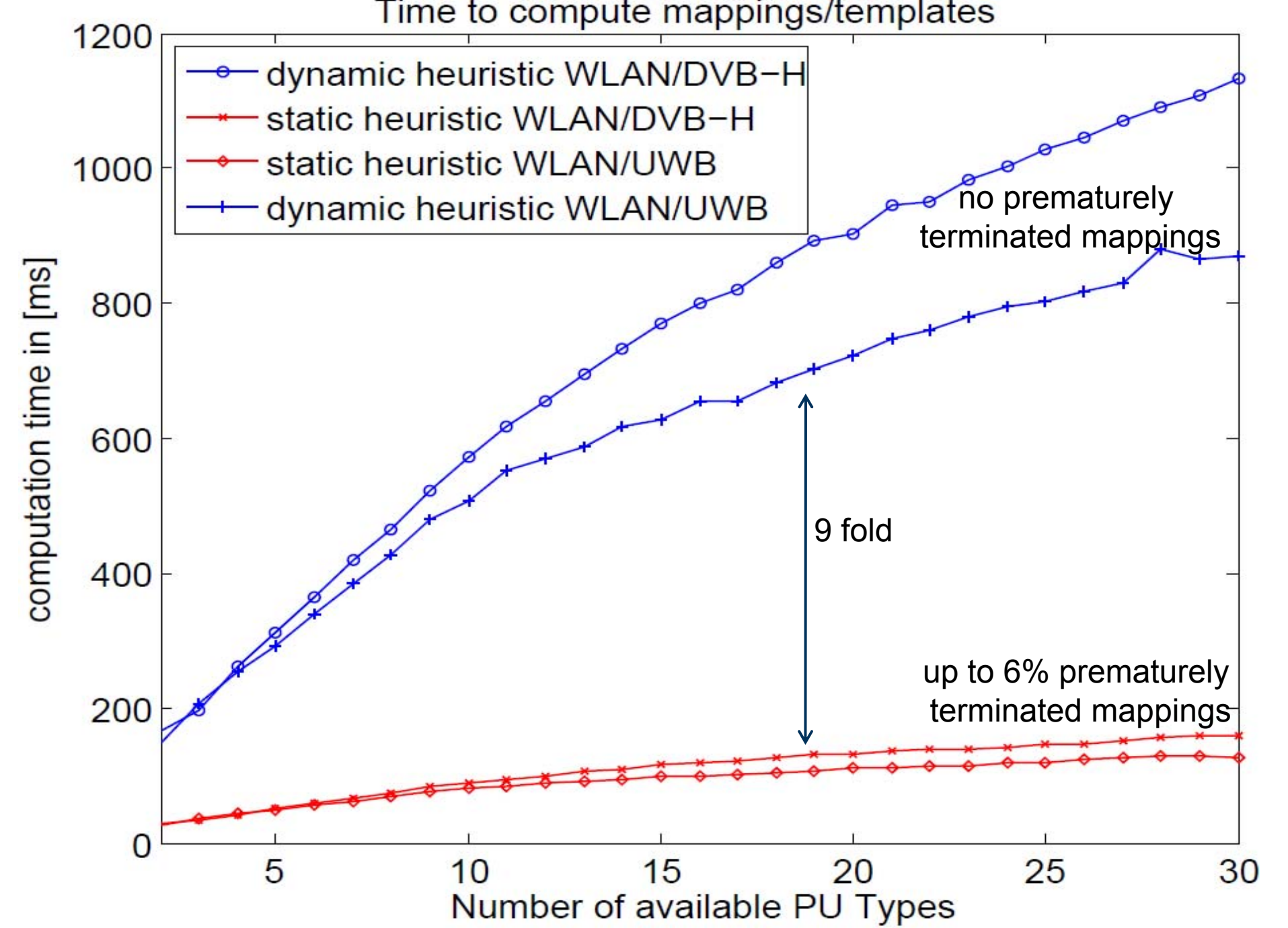
Experimental Results

- 2 Applications executed concurrently
  - Experiment 1: WLAN/DVB-H (34 Tasks)
  - Experiment 2: WLAN/UWB (28 Tasks)
- Each Application is composed of 2 Modes
- Results in 12 Scenario Sequences
- Experiments with 6 different probability distributions
- average expected power consumption compared to global static mapping (RTAS 2009)

Application groups	$\hat{\mathcal{A}}_1$ (DVB-H)		$\hat{\mathcal{A}}_2$ (WLAN)		$\hat{\mathcal{A}}_3$ (UWB)	
Applications	A_1 (**sync**)	A_2 (**receive**)	A_3 (**send**)	A_4 (**receive**)	A_5 (**send**)	A_6 (**receive**)
Number of tasks	4	7	10	23	3	5
Number of shared tasks	0	0	10	10	1	1
$\gamma_{i,j}$	random variables with $0 \leq \gamma_{i,j} \leq 0.5$					
χ	determined by profiling					


Dynamic vs. Global Static Mapping for WLAN and UWB
global static optimal (ILP)
global static heuristic
dynamic optimal (ILP)
dynamic heuristic
average expected power consumption
Number of available PU Types
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
5
10
15
20
25
30


Dynamic vs. Global Static Mapping for WLAN and DVB−H
average expected power consumption
Number of available PU Types
global static optimal (ILP)
global static heuristic
dynamic optimal (ILP)
dynamic heuristic
0
0.5
1
1.5
2
2.5
5
10
15
20
25
30


Time to compute mappings/templates
dynamic heuristic WLAN/DVB−H
static heuristic WLAN/DVB−H
static heuristic WLAN/UWB
dynamic heuristic WLAN/UWB
no prematurely terminated mappings
9 fold
up to 6% prematurely terminated mappings
computation time in [ms]
Number of available PU Types
0
200
400
600
800
1000
1200
5
10
15
20
25
30

Memory Overhead for dynamic mapping

- 2 Applications in 3 modes each:
  - 12 Scenario Sequences
- Experiment 1: 34 Tasks, at most 34 Processing Units:
  - static mapping: ~ 150 byte
  - dynamic mapping: ~ 2.0 kb × number of execution prob.
- Experiment 2: 28 Tasks, at most 28 Processing Units:
  - static mapping: ~ 100 byte
  - dynamic mapping: ~ 1.5 kb × number of execution prob.

Conclusion

- Scenario and Scenario Sequences generated from application specification
- static template mapping computed for each
  - Scenario Sequence
  - execution probability
- Memory overhead bounded by number of scenario sequences
- Experimental Results show up to 45% reduction on average expected power consumption

Thank you for your Attention !