

A Compositional Approach for Schedulability Analysis of Distributed Avionics Systems

Pujie Han

Thessaloniki, 15 April 2018



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY



AALBORG UNIVERSITY



Background



Approach



Modeling



Case study



Background



Approach

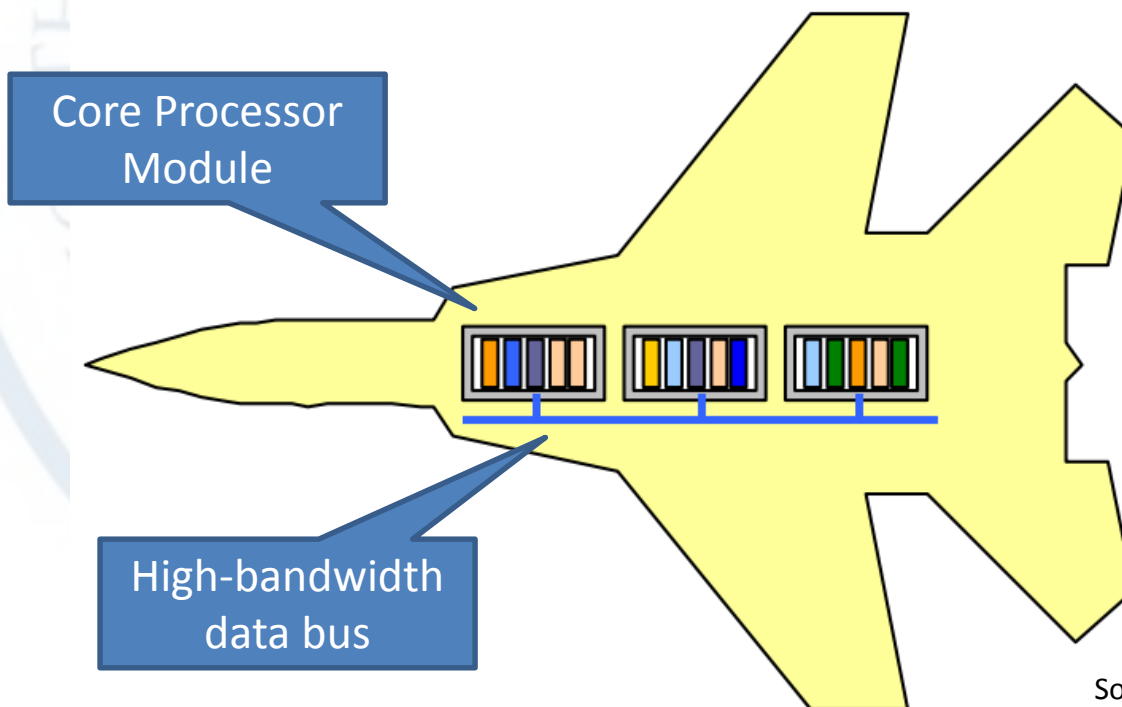


Modeling



Case study

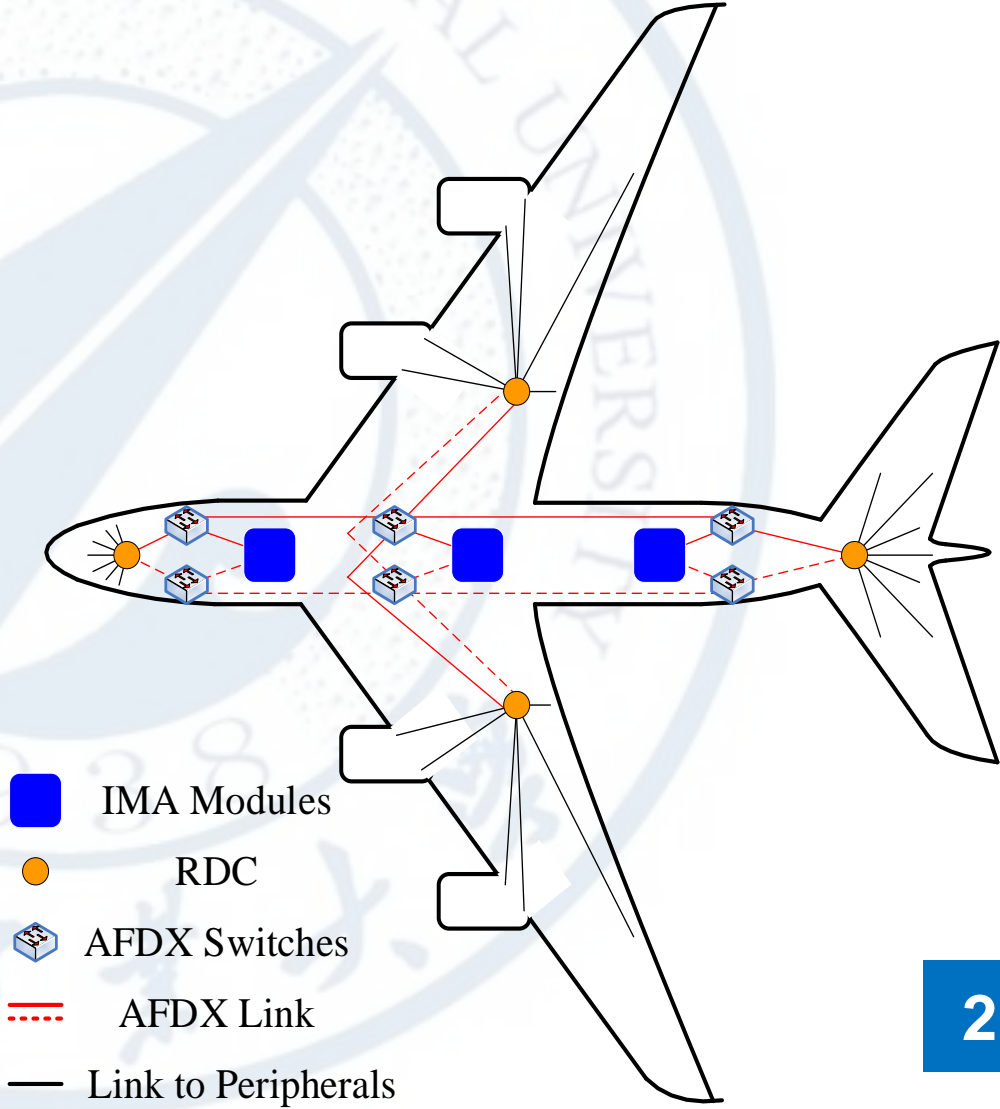
- **Integrated Modular Avionics (IMA)**
 - One function = Software downloaded to the modules
 - Generalized integrated processing modules
 - A unified high-bandwidth network



Source: ASAAC part I

Distributed Integrated Modular Avionics (DIMA)

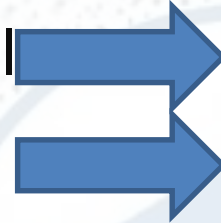
- **Features**
 - IMA but distributed intelligence
 - I/O close to actuators and sensors
 - Computation close to actuators and sensors
 - COTS computers and I/O units as Modules
 - Separation into integration areas
- **More complex schedulability analysis**



- **Analytical Methods**

- Resource Model

- Task Model



Supply

Demand

Response Time Analysis

Schedulability

- **Expressiveness of analytical model**

- Limited to simplified system behavior

- Only real-time computation constraints

- **Conservative assumptions**

- Too many "pessimistic" worst case assumptions in modeling phase and response time analysis

- Waste of computation and communication resources

- **Timing Anomalies:** local worst-case \neq global worst-case.

- **Related Work by Model Checking**
 - **Reachability Analyses of Formal Models**
 - Nonschedulability conditions encoded into Error states
 - Advanced Petri Nets, Linear Hybrid Automata (LHA), Timed Automata (TA), Stopwatch Automata (SWA)
 - Expressive to express more complex behavior
 - State space explosion
 - **Compositional Analyses**
 - Exploit the nature of temporal isolation of partitions
 - Reduce the complexity of reachability analyses.

Isolated computation and communication analysis

- System=Computer modules + Their underlying network
 - Independent hierarchical scheduling systems
 - Network delay in the worst case.
- Challenges
 - Interactions between avionics computers are increasing
 - Each subsystem can be distributed across the whole aircraft
 - Network delay cannot be ignored in schedulability analysis
 - All communications are integrated into a unified network.

Index



Background



Approach



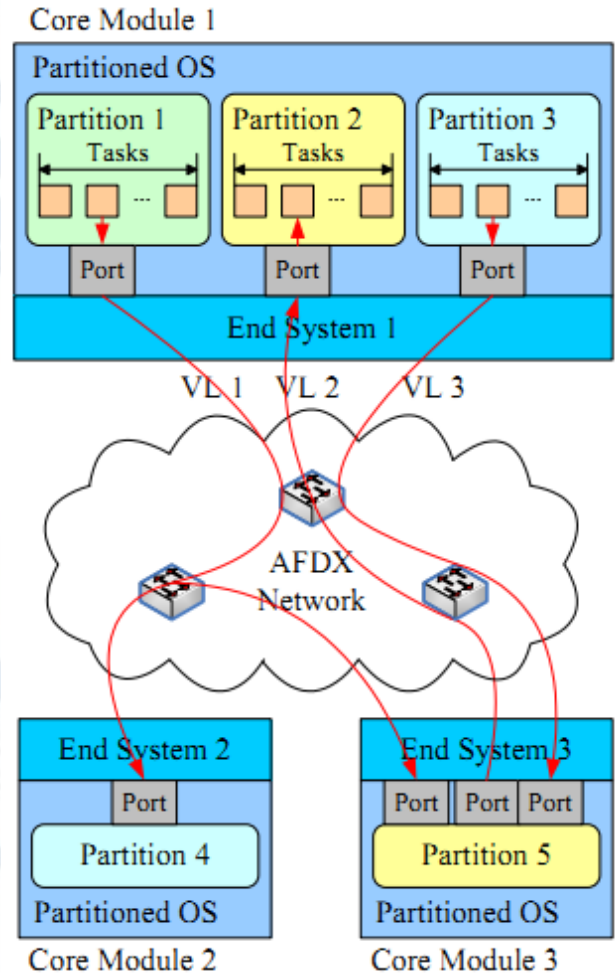
Modeling



Case study

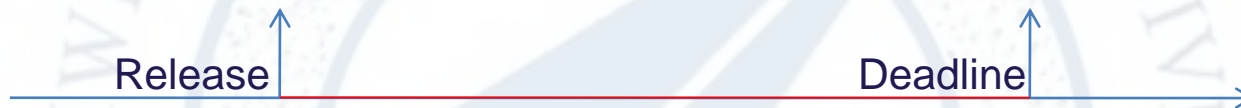
● We consider such a DIMA core system:

- ARINC-653 processing modules
- A unified AFDX network
- Two-level hierarchical scheduling
- Concrete task behavior
- Task synchronization
- Inter-partition communication via ARINC-653 ports

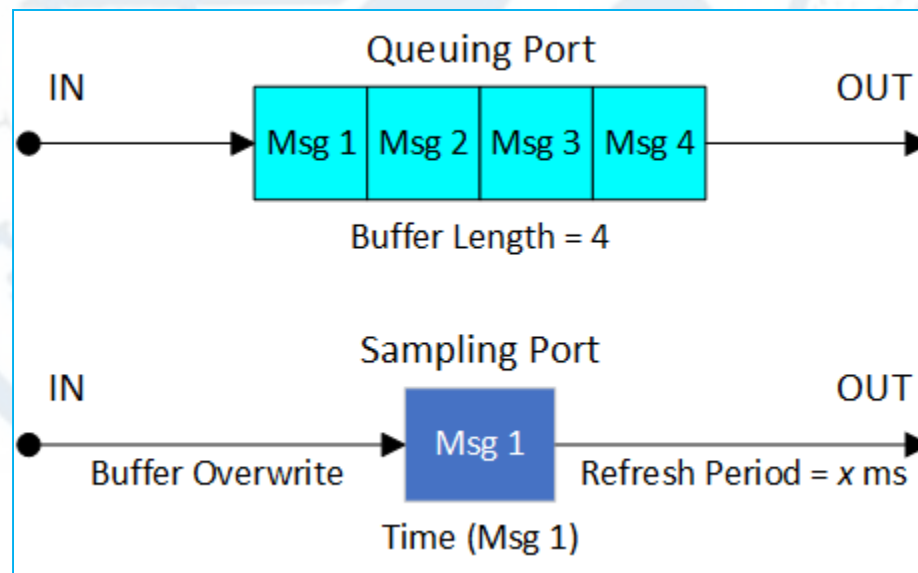


- **Schedulability Properties**

- Deadline of each real-time task



- Communication constraints



Our Approach Adopts:

■ Modeling in UPPAAL

- Stopwatch Automata

- Cover the major features of a DIMA core system

■ Global View

- Includes both computation and communication.

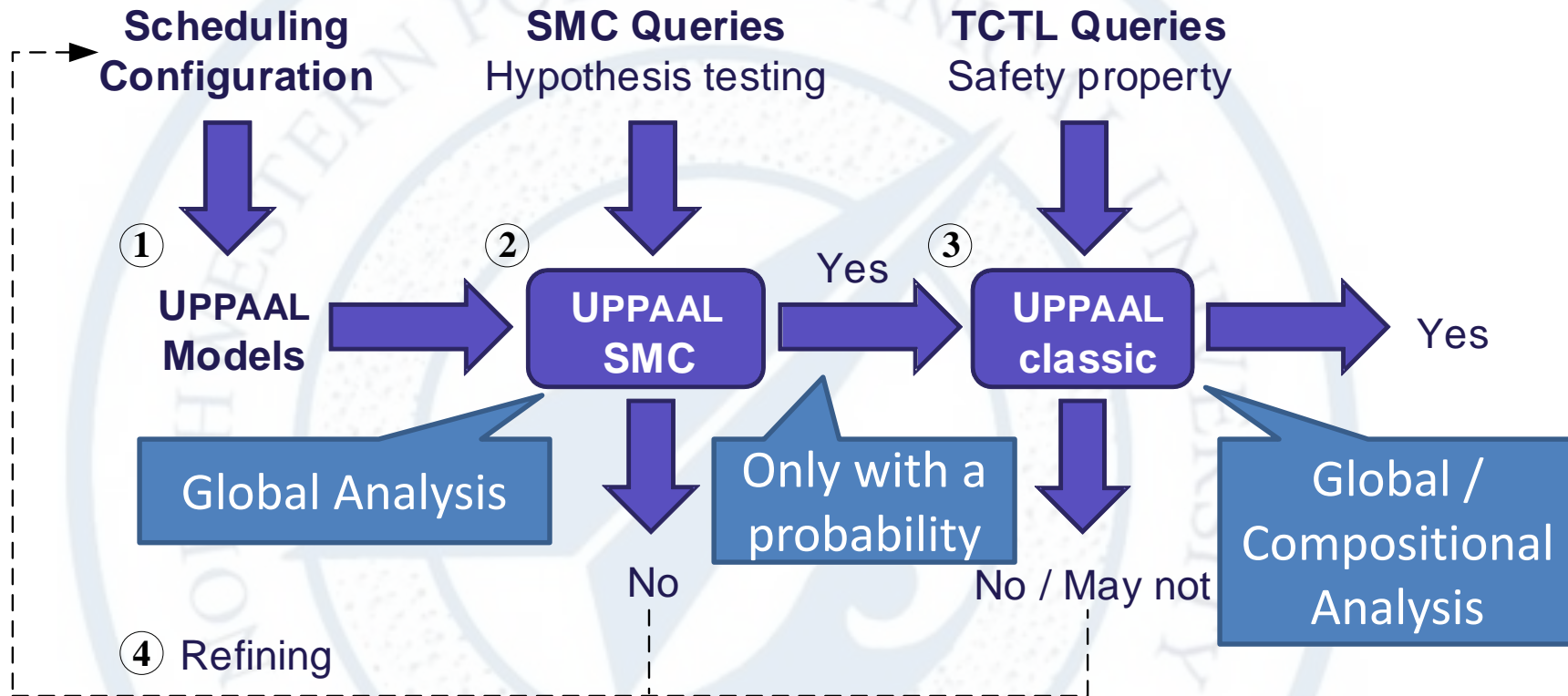
■ Alleviating the State Space Explosion

- Combination of classic and statistical model checking
- Compositional Method.

ARINC-653 hierarchical scheduling
Multiple real-time task types
Resource sharing
Inter-partition communications
AFDX / FC-AE network

SMC, a simulation-based approach, avoid an exhaustive search of the state-space.

Verify different parts of the system **separately**, conclude about the **whole** system.



- Encoding system into UPPAAL SWA models
- Fast falsification by UPPAAL SMC
- Strict schedulability verification by UPPAAL classic MC
- Refinement of the system configuration.

- **Schedulability testing in UPPAAL SMC**

- Cannot guarantee schedulability but can quickly falsify non-schedulable schemes.
- Hypothesis testing:

$$\Pr[\leq M](\langle \rangle \text{ErrorLocation}) \leq \theta$$

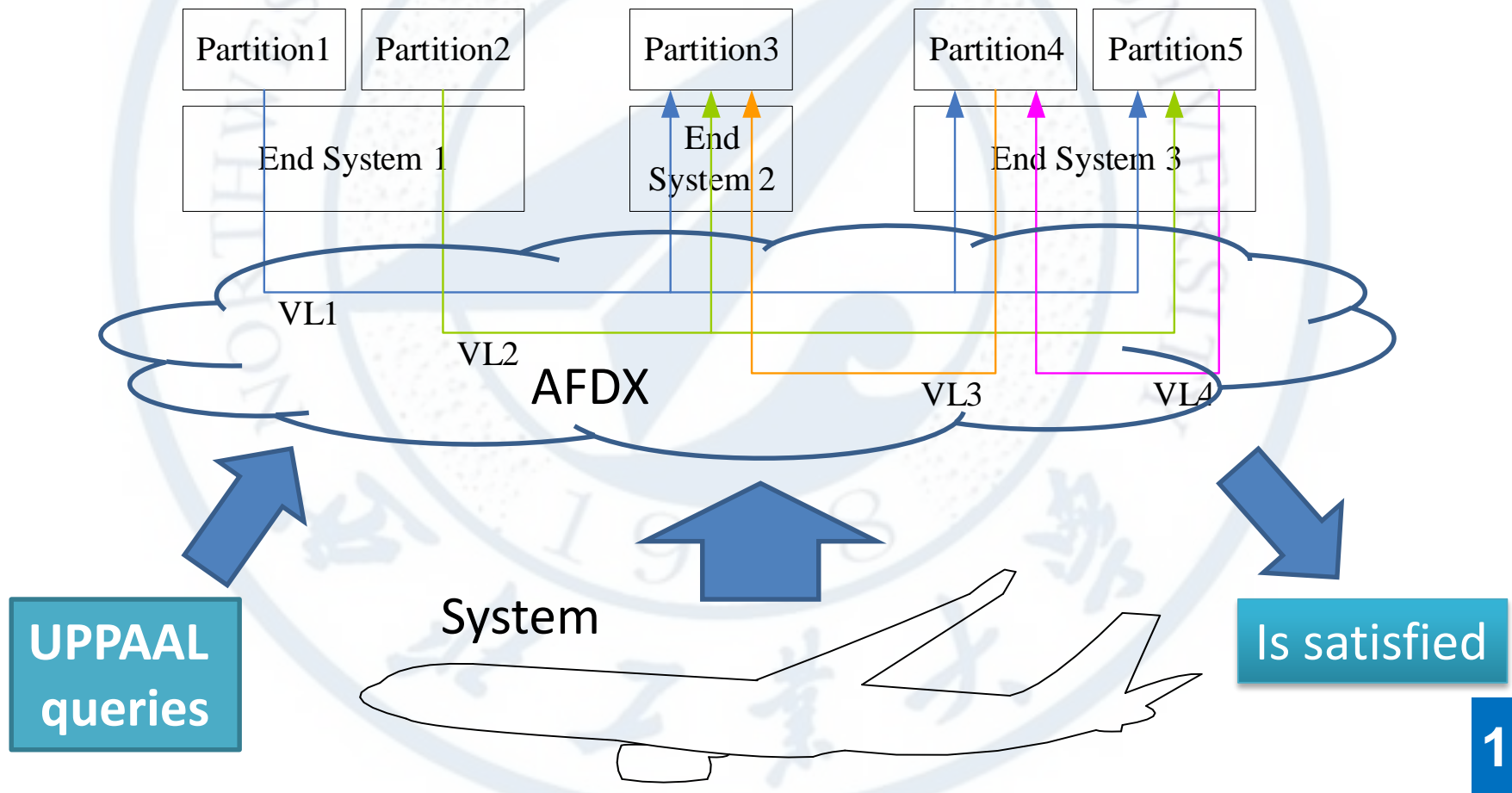
- **Schedulability Verification in Classic UPPAAL**

- Guarantee schedulability but face state-space explosion.
- Safety property:

$$A[] \text{ not ErrorLocation}$$

Global Analysis

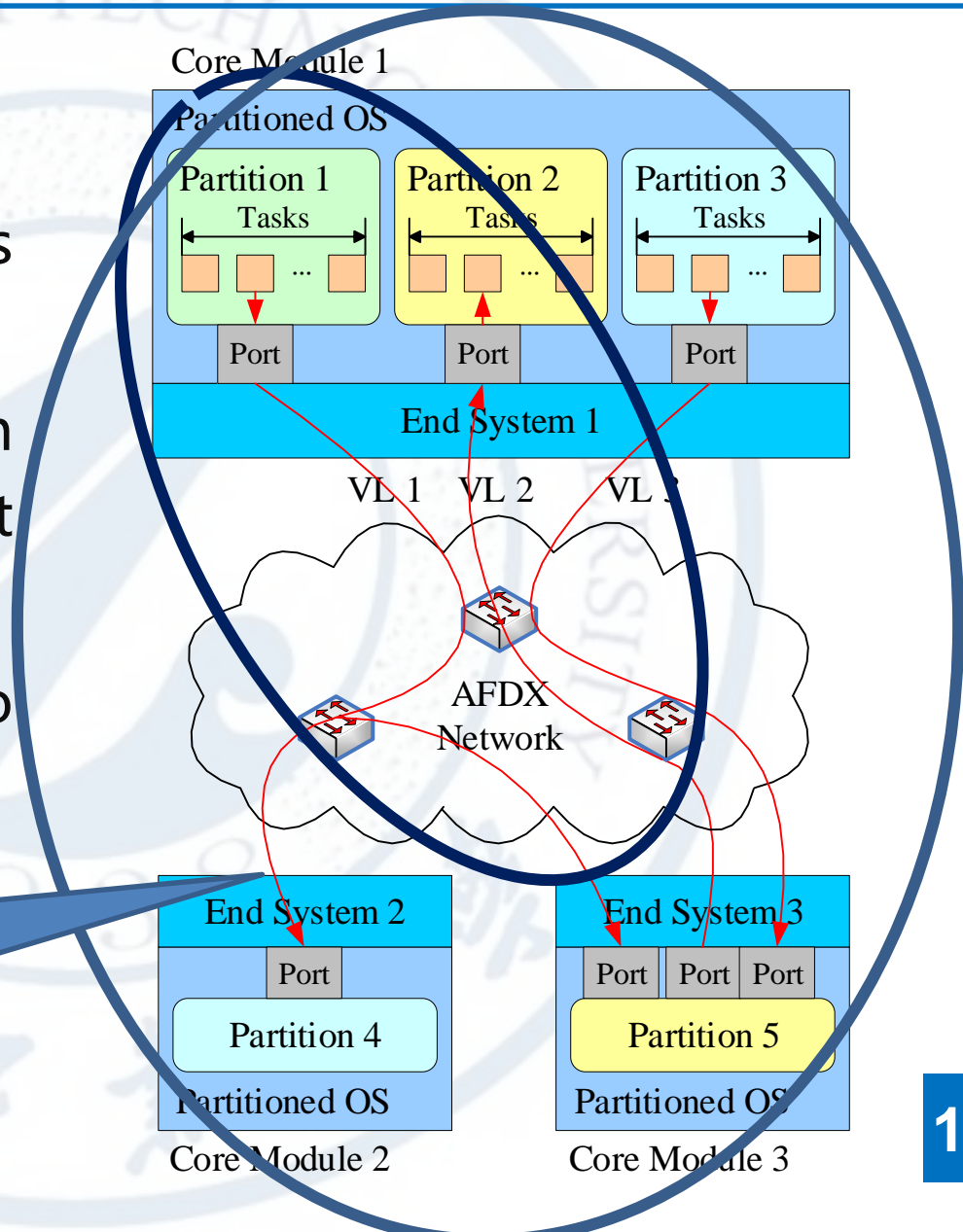
- Applied to the system with small size (Normally < 10 tasks)



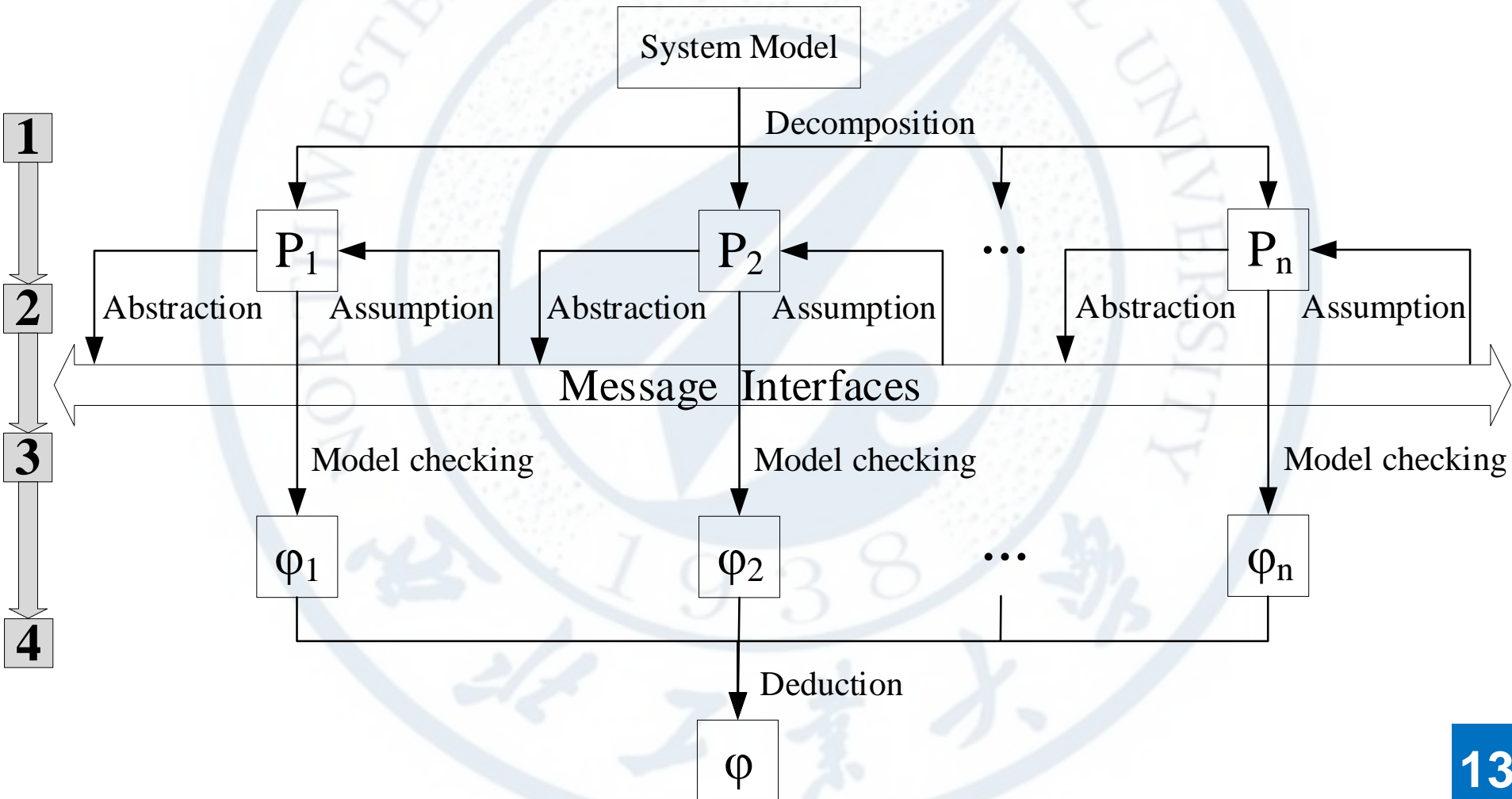
Compositional Analysis

- Used for larger systems (Normally > 10 tasks)
- Check each partition including its environment individually
- Combine local results to derive the global property.

How to decouple **communication dependency** from other partitions?



Assume-Guarantee Reasoning



- **Decomposition**

- **Global property**

$\varphi: A[]$ not ErrorLocation

- **Original Goal**

$$P_1 \parallel P_2 \parallel \dots \parallel P_n \models \varphi$$

- **Divided into n properties of Partition P_i**

$\varphi_i: A[]$ not ErrorLocation_i

where φ can be written as the conjunction $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n$

- **We now have n goals**

$$P_1 \parallel P_2 \parallel \dots \parallel P_n \models \varphi_i, i \in \{1, 2, \dots, n\}.$$

■ Construction of Message Interfaces

■ Message Interface

- An abstract model that describes the external message-sending behavior of a partition

■ Abstraction Relation \ll

- $A_{i,j}$: Partition P_j sends messages to P_i

$$P_j \ll A_{i,j}$$

■ How to Construct Message Interfaces ?

- 1. An intricate automaton that covers **all** the message types ?
- 2. Modeling **each** message in one automaton & **Composition**

■ Construction of Message Interfaces

■ Message Interface A^k_j

- P_j sends msg_k to other partitions:

$$P_j \preceq A^k_j$$

■ Abstraction Compositionality of Message Interfaces

- For any $k \in K$, if $P_j \preceq A^k_j$, the composition of A^k_j satisfies

$$P_j \preceq \coprod_{k \in K} A^k_j$$

■ Abstraction of a Partition

- P_j sends all msg_k , $k \in K$ to P_i , and P_j can be replaced with

$$A_{i,j} = \coprod_{k \in K} A^k_j, \text{ where } P_j \preceq A_{i,j}$$

■ Construction of Message Interfaces

■ Compositionality of Message Interfaces

$$P_1 \leq A_1, P_2 \leq A_2 \Rightarrow P_1 \parallel P_2 \leq A_1 \parallel A_2$$

■ Assumptions of the Environment of a Partition

- Composite model that describes the environment of P_i :

$$\left\|_{j=1, j \neq i}^n A_{i,j}\right.$$

■ Abstraction Relation in the Analysis of a Partition

$$P_1 \parallel P_2 \parallel \dots \parallel P_n \preceq P_i \parallel \left(\left\|_{j=1, j \neq i}^n A_{i,j}\right. \right)$$

The size of abstract model is smaller.

■ Model Checking

■ Schedulability Verification

- Check n subproblems by model checking in UPPAAL:

$$P_i \parallel \left(\prod_{j=1, j \neq i}^n A_{i,j} \right) \models \varphi_i \quad i \in \{1, 2, \dots, n\}.$$

■ Verification of Abstraction Relations

- For any message interface A , create a **test automaton** A^T
- Check if the Error locations of A^T are reachable in UPPAAL:

$$(P \parallel A^T \models \neg E \Leftrightarrow A^T.Error) \implies P \preceq A$$

- Deduction

- Property Preservation

$$P \preceq Q \wedge Q \models \varphi \Rightarrow P \models \varphi$$

- Apply the assume-guarantee reasoning rule:

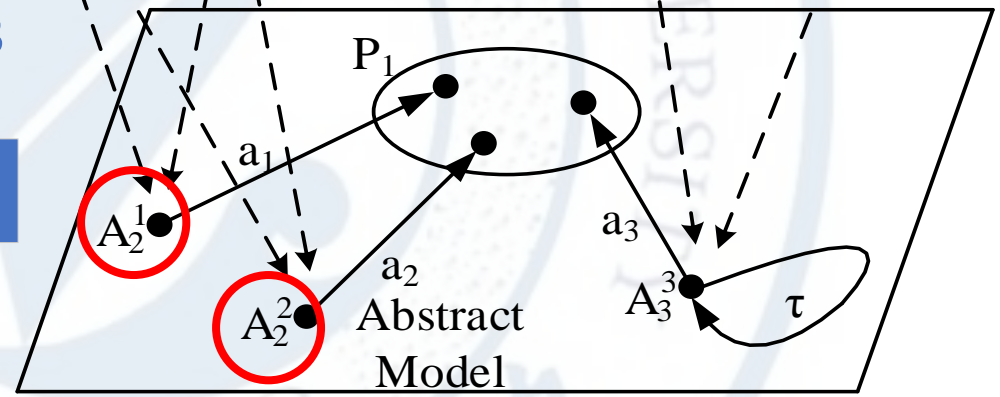
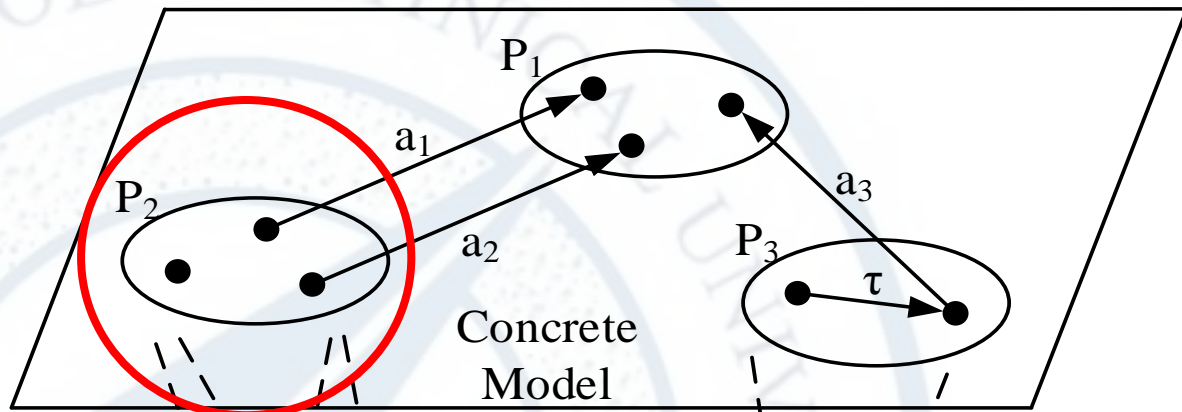
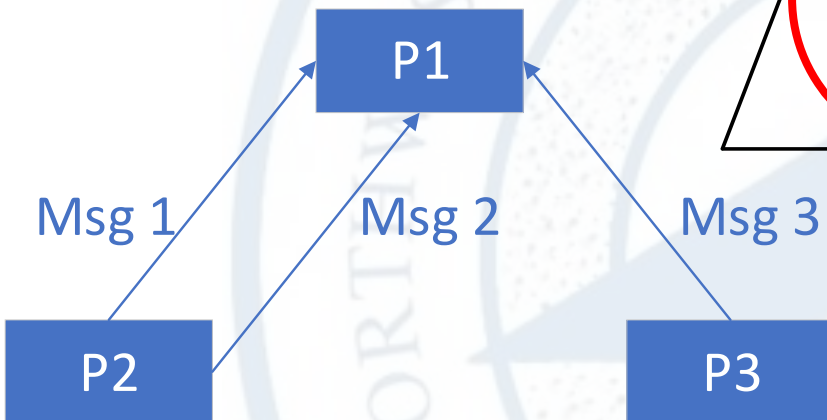
$$P_1 \parallel P_2 \parallel \cdots \parallel P_n \preceq P_i \parallel \left(\parallel_{j=1, j \neq i}^n A_{i,j} \right)$$

$$\bigwedge_{\{j \mid j > i\}} P_j \preceq A_{i,j}$$

$$P_i \parallel \left(\parallel_{j=1, j \neq i}^n A_{i,j} \right) \models \varphi_i$$

$$P_1 \parallel P_2 \parallel \cdots \parallel P_n \models \varphi_i$$

Example



$$P_2 \preceq A_2^1, P_2 \preceq A_2^2, P_3 \preceq A_3^3$$

$$P_2 \preceq A_2^1 \parallel A_2^2, P_3 \preceq A_3^3$$

$$P_1 \parallel P_2 \parallel P_3 \preceq P_1 \parallel A_2^1 \parallel A_2^2 \parallel A_3^3$$

$$P_2 \preceq A_2^1 \wedge P_2 \preceq A_2^2 \wedge P_3 \preceq A_3^3$$

$$P_1 \parallel A_2^1 \parallel A_2^2 \parallel A_3^3 \models \varphi_1$$

$$P_1 \parallel P_2 \parallel P_3 \models \varphi_1$$

Abstraction (C) Property Preservation

Index



Background



Approach



Modeling

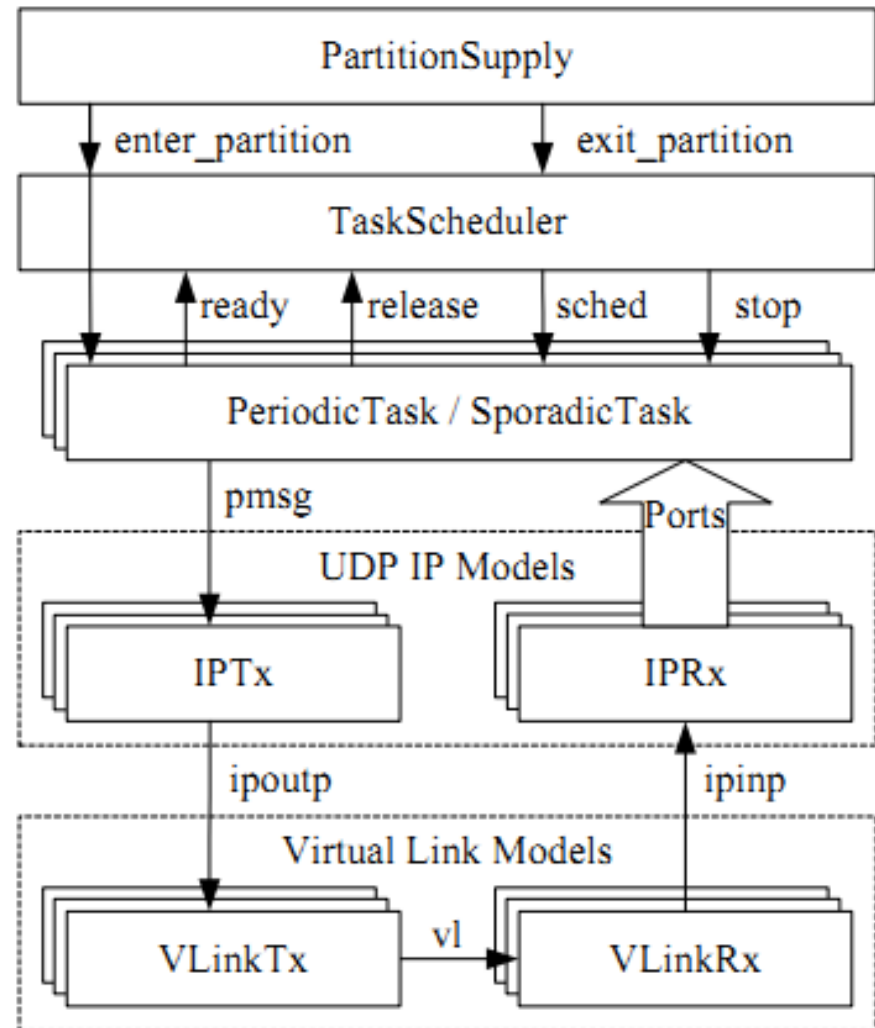


Case study

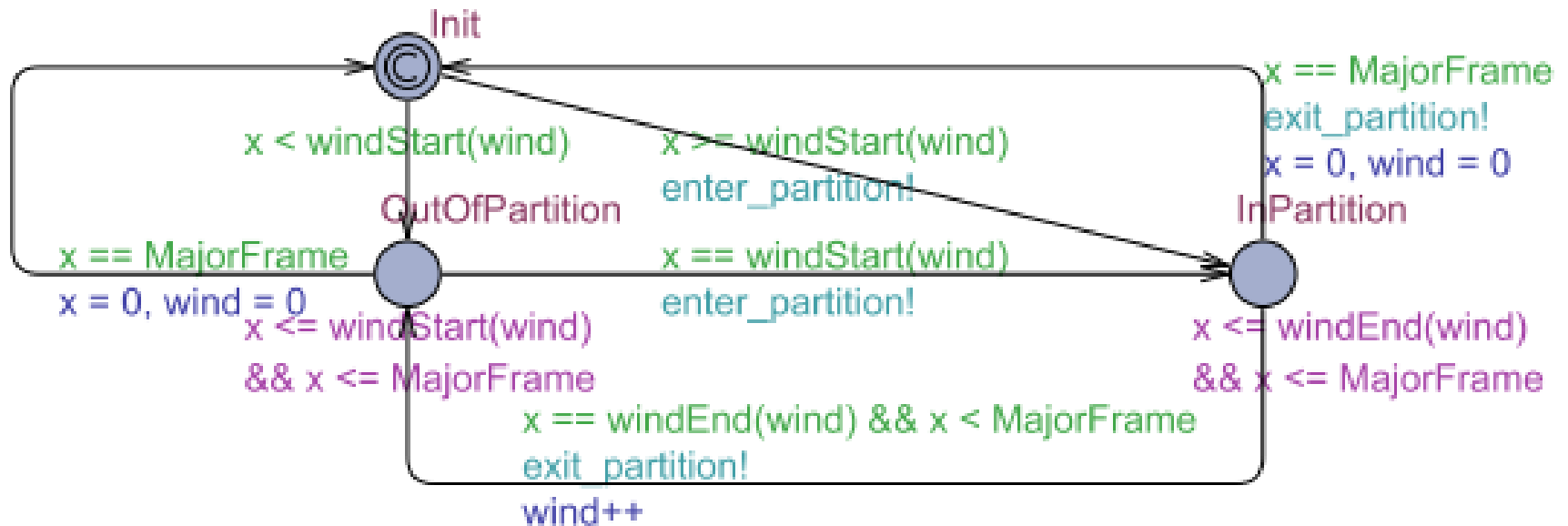
Timed stopwatch automata in UPPAAL



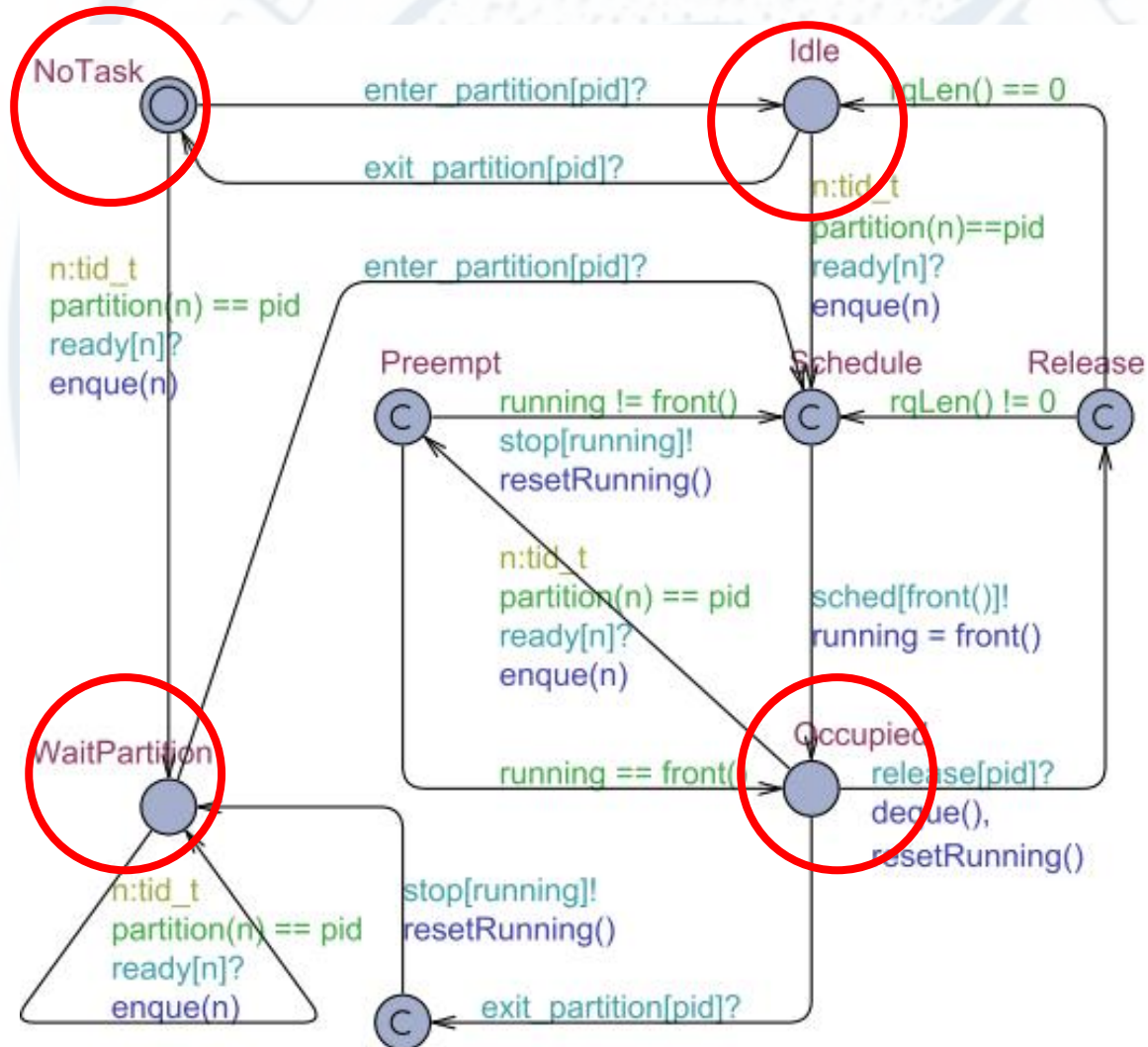
- **Scheduling layer**
 - PartitionScheduler
 - TaskScheduler
- **Task layer**
 - PeriodicTask
 - SporadicTask
- **Communication layer**
 - IPTx, IPRx
 - VLinkTx, VLinkRx



■ Example: PartitionScheduler



■ Example: TaskScheduler



Index



Background



Approach

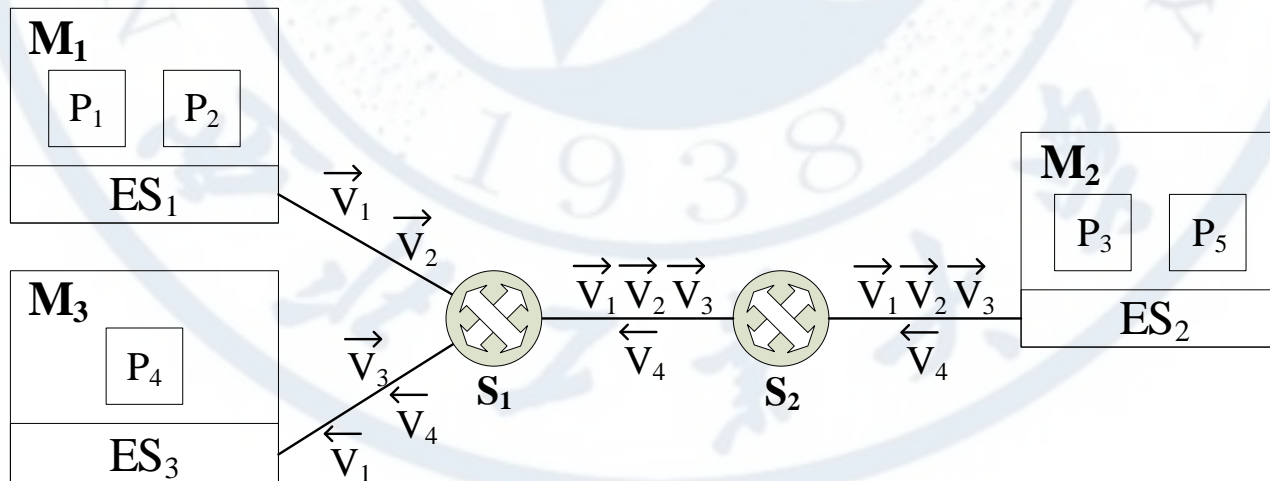


Modeling



Case study

- **Statistics of This Avionics System**
 - 3 Core Processing Modules
 - 5 ARINC-653 Partitions
 - 18 periodic tasks and 4 sporadic tasks
 - 4 AFDX Virtual Links
 - 2 Sampling Ports and 2 Queuing Ports



No.	Task	Release	Offset	Jitter	Deadline	Priority	Execution Chunks			
							Time	Mutex	Output	Input
P_1	Tsk_1^1	[25,25]	2	0	25	2	[0.8,1.3]	-	-	-
							[0.1,0.2]	-	-	-
	Tsk_2^1	[50,50]	3	0	50	3	[0.2,0.4]	-	Msg_1	-
	Tsk_3^1	[50,50]	3	0	50	4	[2.7,4.2]	-	-	-
	Tsk_4^1	[50,50]	0	0	50	5	[0.1,0.2]	Mux_1^1	-	-
							[0.6,0.9]	-	-	-
		[120,∞)	0	0	120	6	[0.1,0.2]	Mux_1^1	-	-
P_2	Tsk_1^2	[50,50]	0	0.5	50	2	[1.9,3.0]	-	-	-
	Tsk_2^2	[50,50]	2	0	50	3	[0.7,1.1]	-	Msg_2	-
	Tsk_3^2	[100,100]	0	0	100	4	[0.1,0.2]	Mux_1^2	-	-
							[0.8,1.3]	-	-	-
			[100,∞)	10	0	100	5	[0.2,0.3]	Mux_1^2	-
P_3	Tsk_1^3	[25,25]	0	0.5	25	2	[0.5,0.8]	-	-	Msg_1
	Tsk_2^3	[50,50]	0	0	50	3	[0.7,1.1]	-	-	Msg_2
	Tsk_3^3	[50,50]	0	0	50	4	[1.0,1.6]	-	-	Msg_3
	Tsk_4^3	[100,∞)	11	0	100	5	[0.7,1.0]	-	-	-
							[0.1,0.3]	-	-	-
P_4	Tsk_1^4	[25,25]	3	0.2	25	2	[0.7,1.2]	-	-	-
	Tsk_2^4	[50,50]	5	0	50	3	[1.2,1.9]	-	Msg_3	Msg_1
	Tsk_3^4	[50,50]	25	0	50	4	[0.1,0.2]	-	-	Msg_4
	Tsk_4^4	[100,100]	11	0	100	5	[0.7,1.1]	-	-	-
	Tsk_5^4	[200,200]	13	0	200	6	[3.7,5.8]	-	-	-
P_5	Tsk_1^5	[50,50]	0	0.3	50	1	[0.7,1.1]	-	-	Msg_1
	Tsk_2^5	[50,50]	2	0	50	2	[1.2,1.9]	-	Msg_4	Msg_2
	Tsk_3^5	[200,200]	0	0	200	3	[0.4,0.6]	-	-	-
							[0.2,0.3]	Mux_1^5	-	-
	Tsk_4^5	[200,∞)	14	0	200	4	[1.4,2.2]	-	-	-
							[0.1,0.2]	Mux_1^5	-	-

Global analysis

22 task processes

vs

Compositional analysis

 ≤ 5 task processes

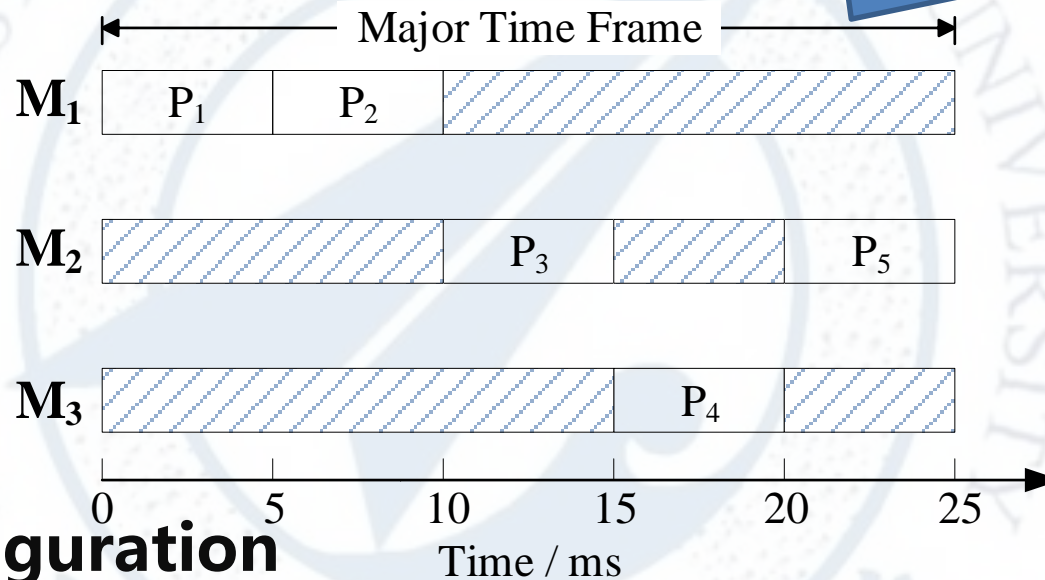
Source: 2013 Carnevali, Pinzuti & Vicario, Compositional verification for hierarchical scheduling of real-time systems.

2009 Easwaran, Lee, Sokolsky & Vestal, A compositional scheduling framework for digital avionics systems

Partition Schedule

5 Disjoint Partition Windows

To make a comparison, keep the temporal order of the schedule in [2013 Carnevali] and [2009 Easwaran].



AFDX Configuration

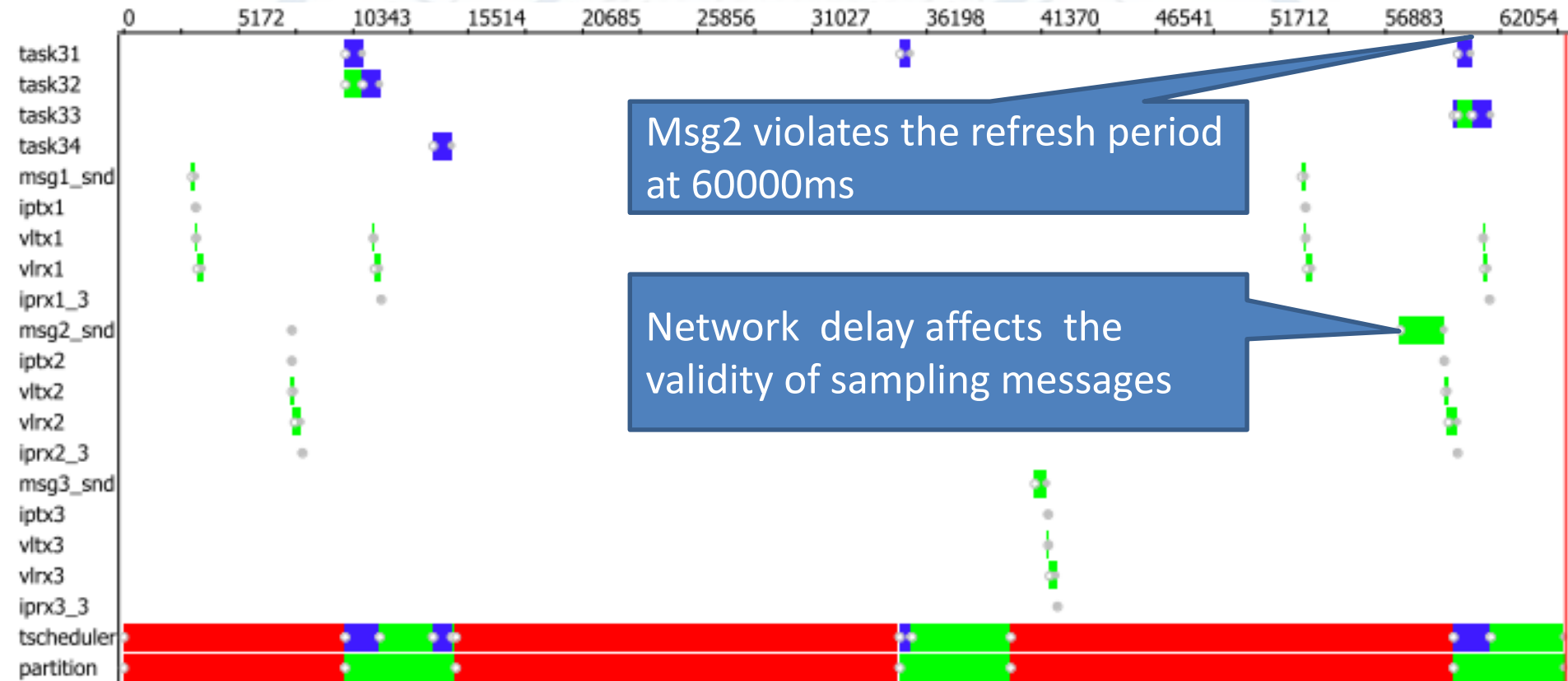
Message	Length	VL	BAG	L_{max}	Source	Destinations
Msg_1	306	V_1	8	200	P_1	P_3, P_4, P_5
Msg_2	953	V_2	16	1000	P_2	P_3, P_5
Msg_3	453	V_3	32	500	P_4	P_3
Msg_4	153	V_4	32	200	P_5	P_4

■ Experiment Results

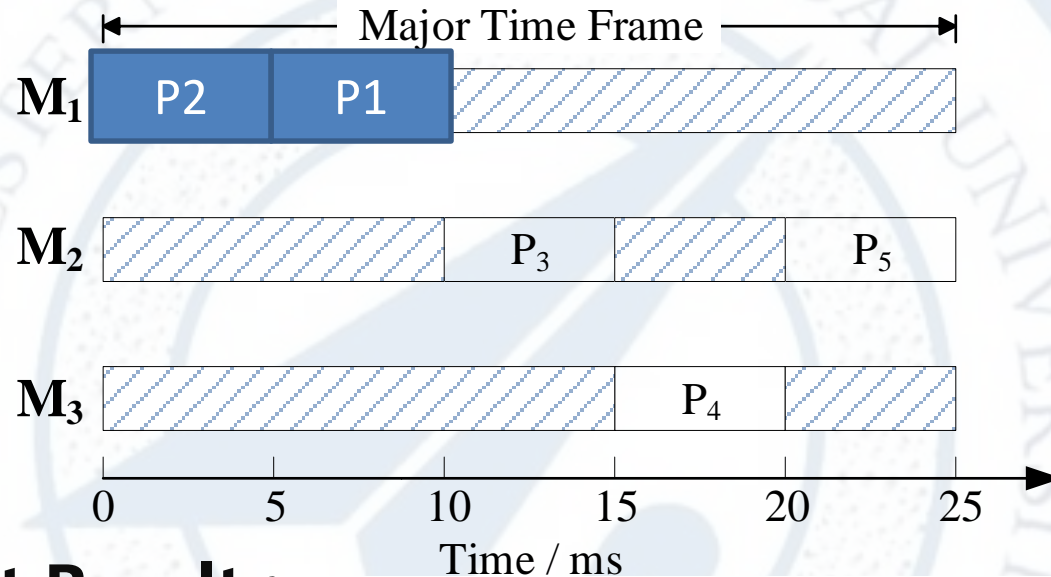
- The Experiment Results (Result), Execution Time (Time/sec.) and Memory Usage (Mem/MB)

MC				SMC		
No.	Result	Time	Mem	Result	Time	Mem
P_1	Yes	7.35	141			
P_2	Yes	1.02	45			
P_3	Maynot	57.84	563	No	2.67	53
P_4	Yes	0.83	45			
P_5	Yes	33.27	526			

■ A Counter Example



Improved Partition Schedule



Experiment Results

The Experiment Results (Result), Execution Time (Time/sec.) and Memory Usage (Mem/MB)

No.	MC			SMC		
	Result	Time	Mem	Result	Time	Mem
P ₁	Yes	6.07	101	Yes	77.58	53
P ₂	Yes	1.09	49			
P ₃	Yes	437.99	3150			
P ₄	Yes	0.88	43			
P ₅	Yes	179.25	2078			

This Approach :

- Modeling DIMA systems in UPPAAL
- Modeling and analysis in a global view
- Combination of classic and statistical model checking
- Application of compositional method.



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY



谢谢聆听！

Thanks for listening !



AALBORG UNIVERSITY