Peak Shaving: Applying limited-preemptive scheduling to peak load reduction in smart buildings

Fotso Sado Alex Yvan: Informatik M.Sc

Herr Lukas Barth: Betreuer

Motivation

- Limit the peak load in a smart building
- Manage the activation of electric loads by a real-time scheduler
- Minimize the number of preemption

Picture: http://www.rcrwireless.com/wp-content/uploads/2016/07/32383041_l.jpg

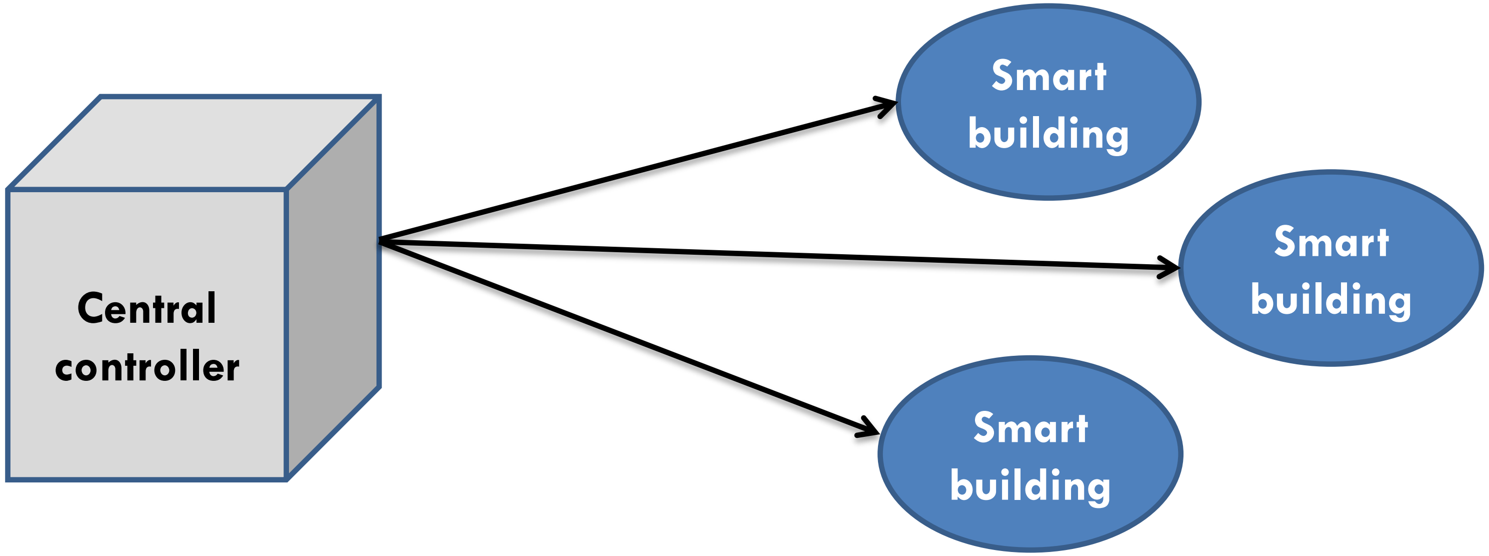
Outline

- Introduction
- Full-preemptive system
- Disadvantages of a Full-preemptive system
- Limited preemtive real-time scheduling
- Simulation result
- Conclusions

Introduction

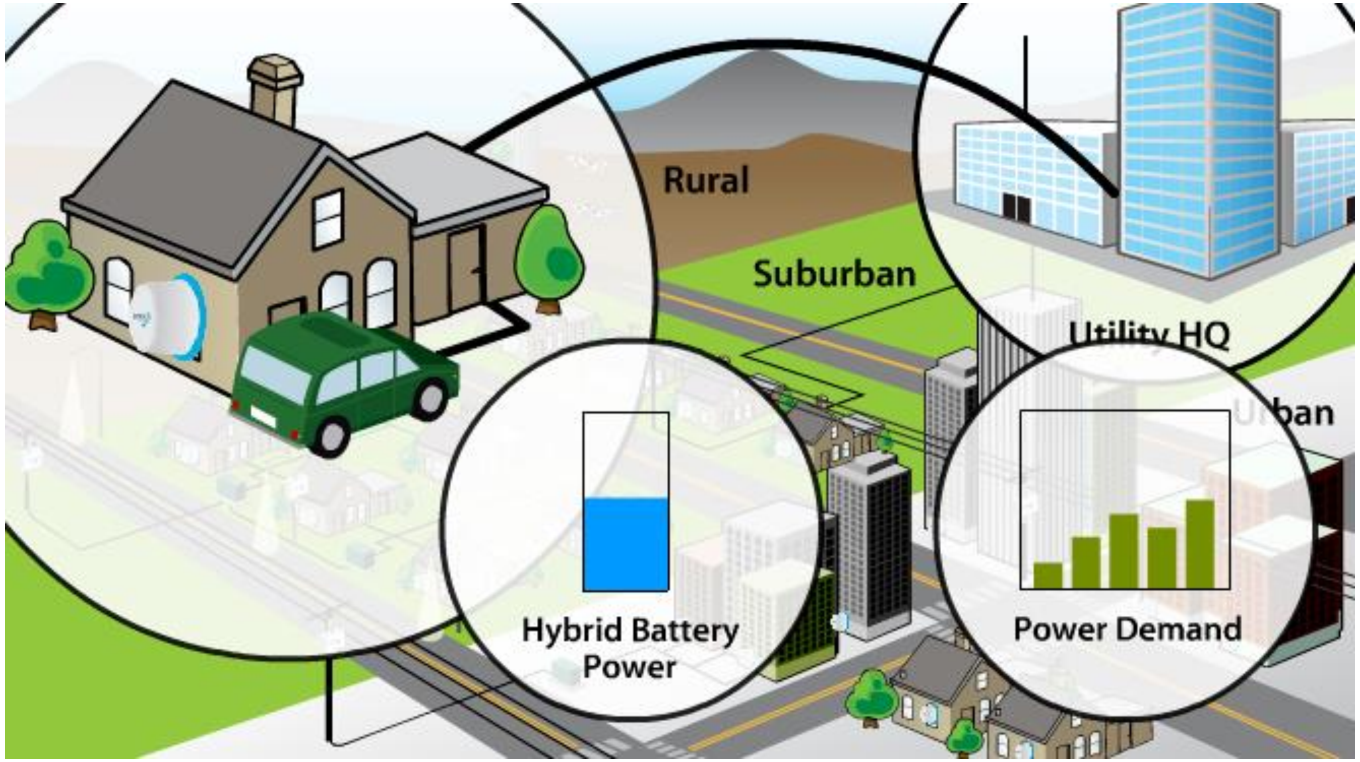
Demand-Side Management (DSM)

- A central controller is used in order to regulate the power demand in a set of smart building
  - Some appliance are stopped during certain time and restarted after
  - Thus decreases the power demand of entire building set
  - Avoids the deficiency of energy at supplier



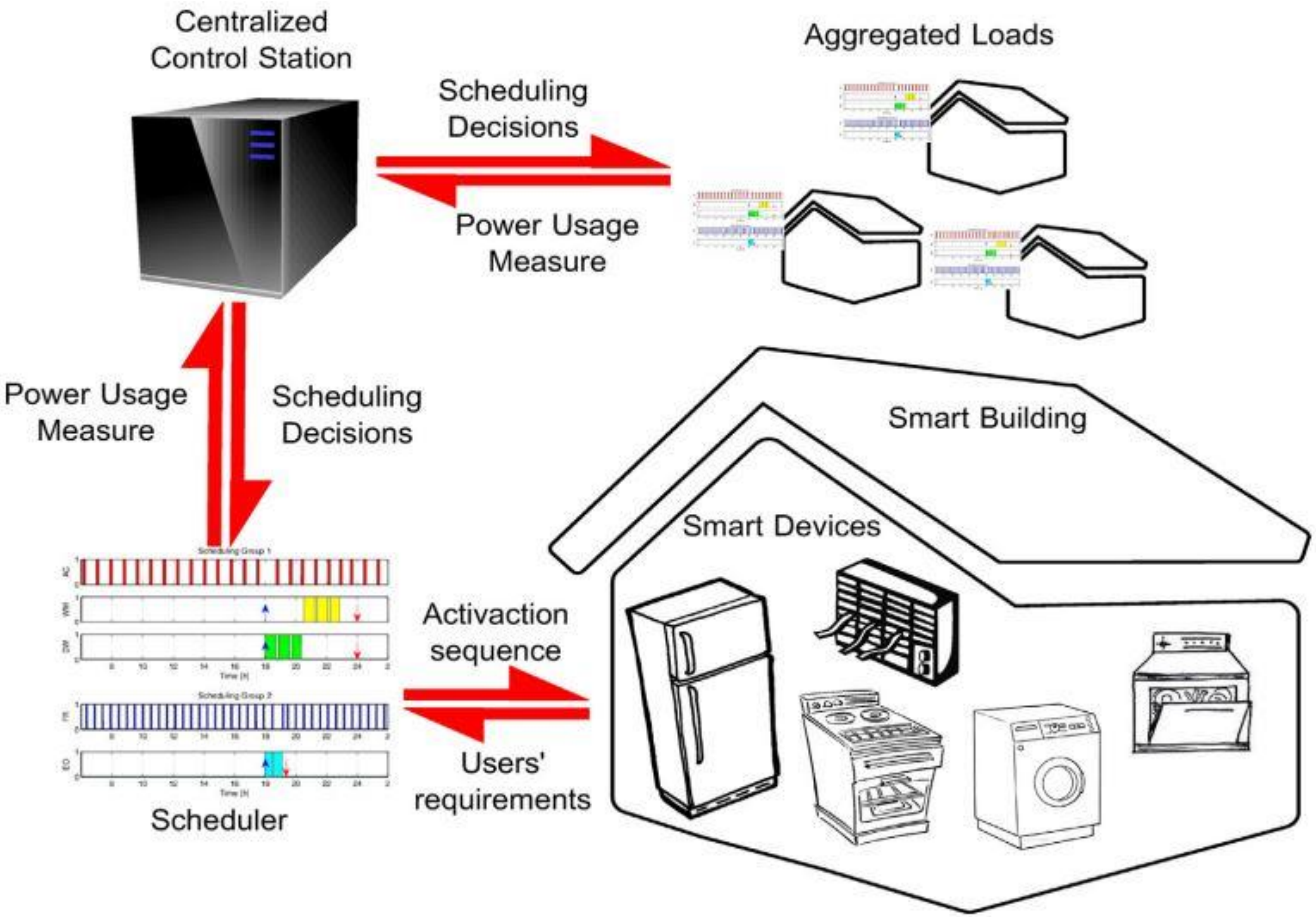
Demand-Side Management (DSM)

- A hybridbattery principe



Picture: http://climateinc.org/wp-content/uploads/2010/03/smart-grid.jpg

DSM with real-time scheduling



Picture: D. Caprino, M. Dova, and T. Facchinetti (2014) [3]

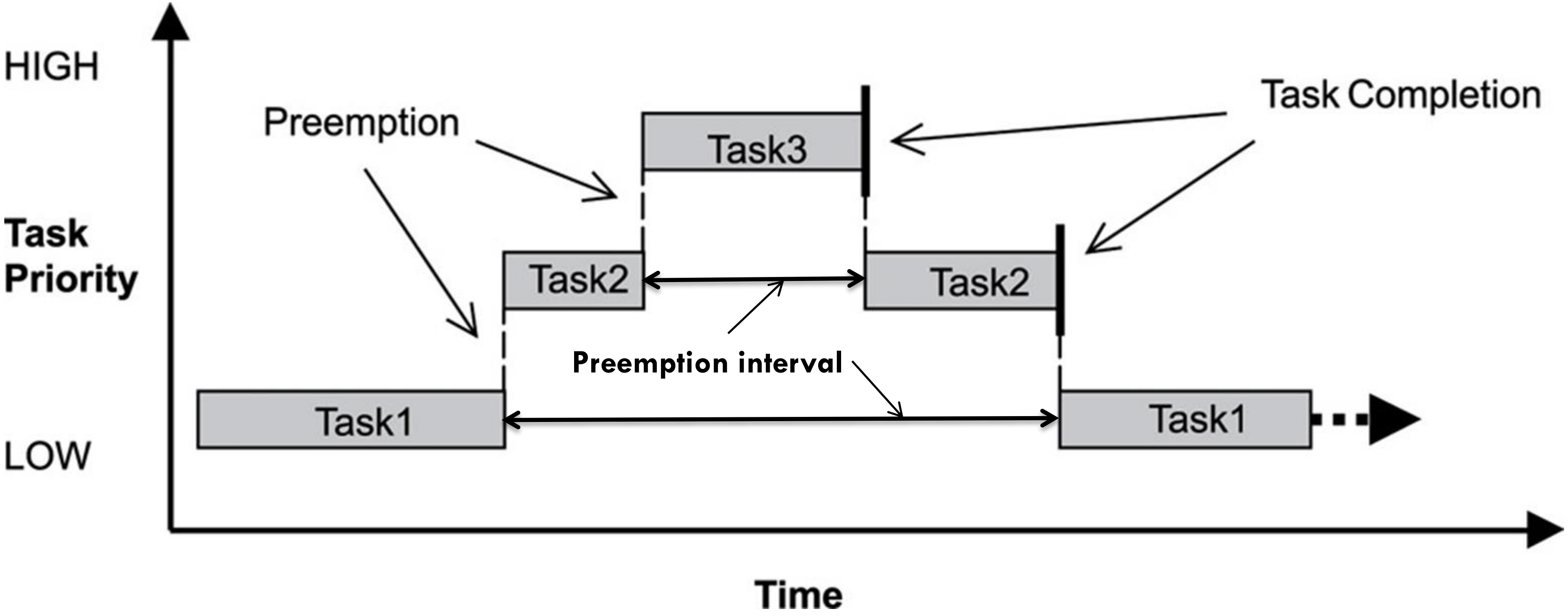
Fig. 1. The Home Energy Management system based on real-time control techniques.

Full-preemptive system

Definition

- **Full-preemptive system** is a model where every task can be interrupted at any time to allow the execution of a higher priority task
- **Preemption interval** is the time interval between the interruption of a tasks and its resumption

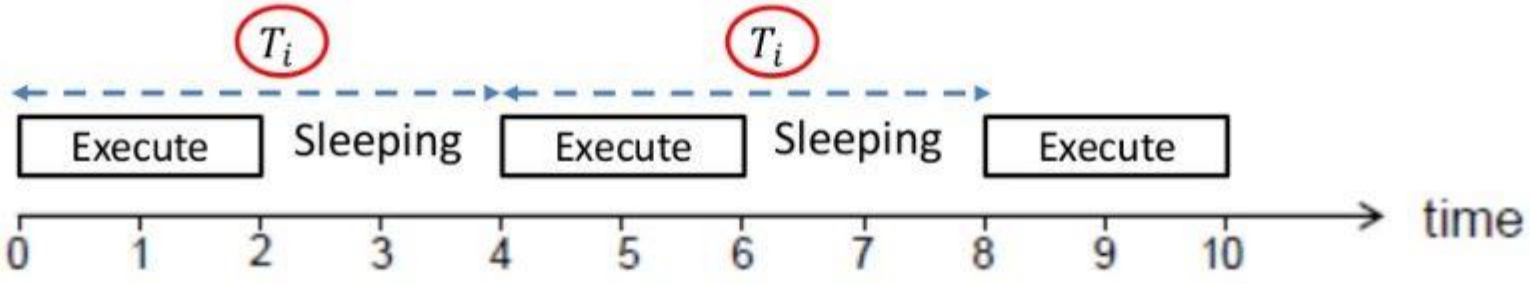
Illustration



Picture: http://www.embeddedlinux.org.cn/rtconforembsys/5107final/images/other-0405_0.jpg

Periodic activation

- **Periodic Task**: repeats in periodic intervals. E.g thermal loads.



Picture: http://de.slideshare.net/knowdiff/sara-afshar-scheduling-and-resource-sharing-in-multiprocessor-realtime-systems

T_i = minimum inter arrival time

Periodic activation

- A **thermal load** is a device whose task is to regulate the temperature of a given environment.

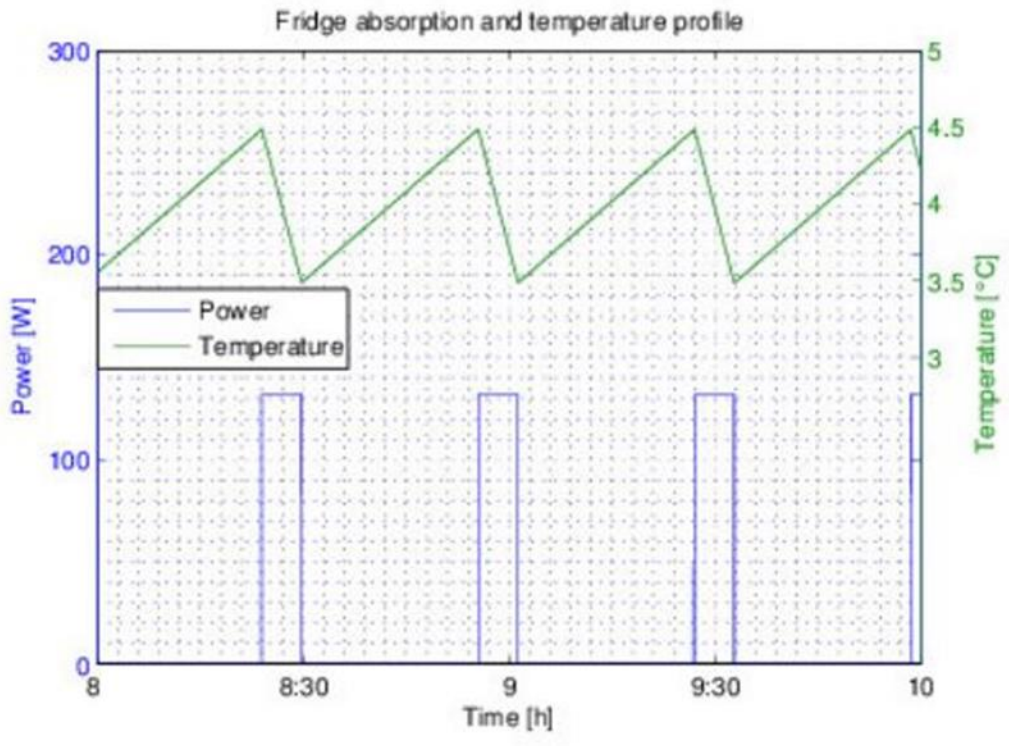
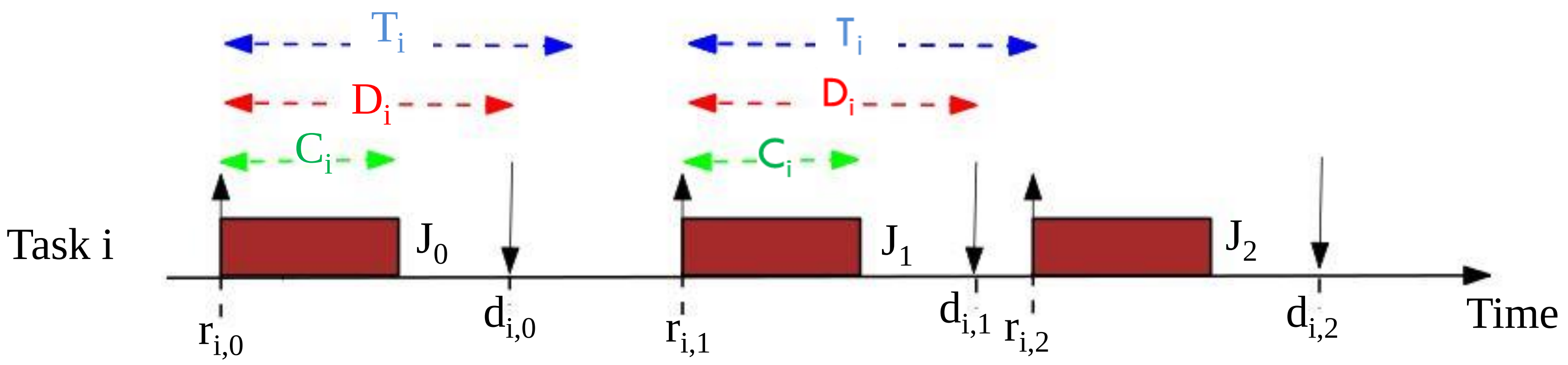


Fig. 1. Power and temperature profiles of a refrigerator. The set-point of the temperature is $4°\text{C} \pm 1$. The consumed power is 132 W.

Sporadic Tasks (as task model)

- The tasks are sporadic, when they
  - Can arrive at the system anytime, but with defined minimum inter-arrival times between two consecutive activations.
  - And are characterized by
    - Minimum inter-arrival time $\mathbf{T_i}$
    - Relative deadline $\mathbf{D_i \leq T_i}$
    - Activation time $\mathbf{C_i \leq D_i}$
    - Request time $\mathbf{r_{i,k}}$ whose $\mathbf{r_{i,k+1} \geq r_{i,k} + T_i}$

Sporadic Tasks (as task model)



- Absolute deadline $\mathbf{d_{i,j}}$ of a job. $\mathbf{d_{i,k} = r_{i,k} + D_i}$
- Job $\mathbf{J_i}$ is an activation of one task
- A sporadic task can generate infinitely many different legal sequences of Jobs

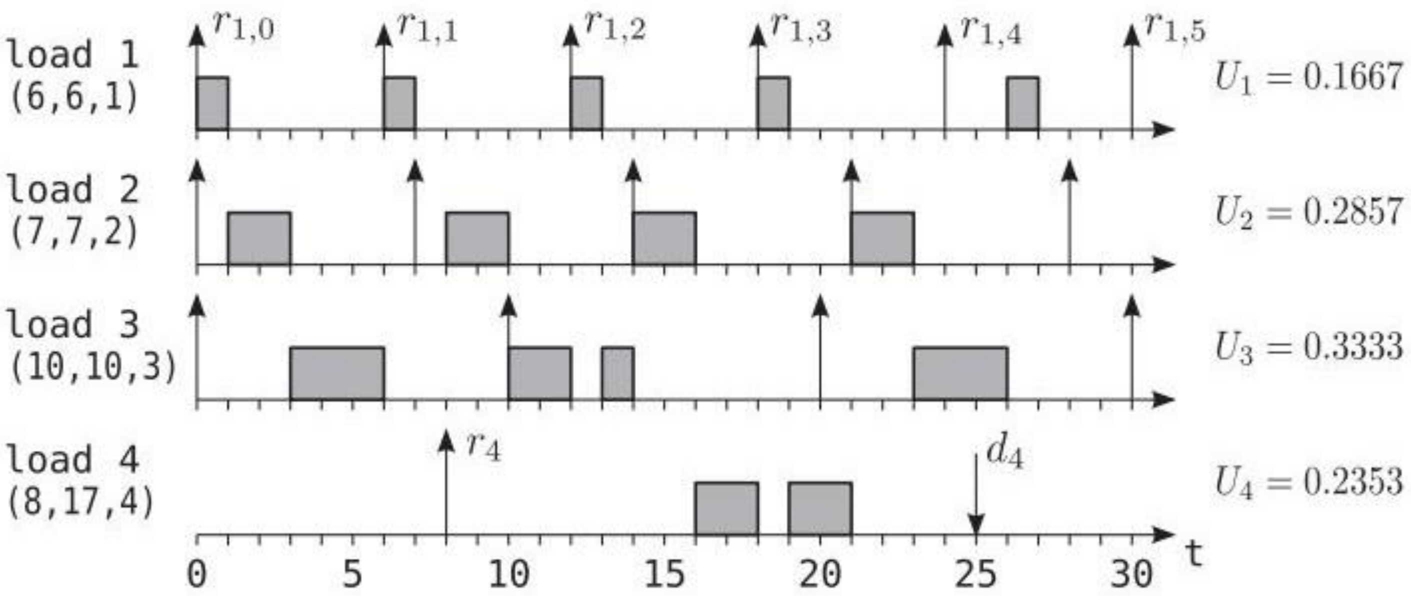
Advantages of using real-time scheduling techniques

- Efficiency
- Predictability
- Robustness
- Reusability
- Scalability
- Automation

Dynamic scheduling algorithm

- The **Earliest Deadline First** (EDF) algorithm dynamically assigns priorities according to deadline and is implemented only on a uniprocessor
  - Tasks having close deadline receive a high priority
  - Tasks having far deadline receive a low priority
- **Process**:
  - At anytime t, active the task whose deadline is closest to t
  - A task does not need to be periodic
  - But it needs to announce its deadline to the scheduler when it becomes activable

Dynamic scheduling algorithm



Picture: D. Caprino et al. (2014) [4]

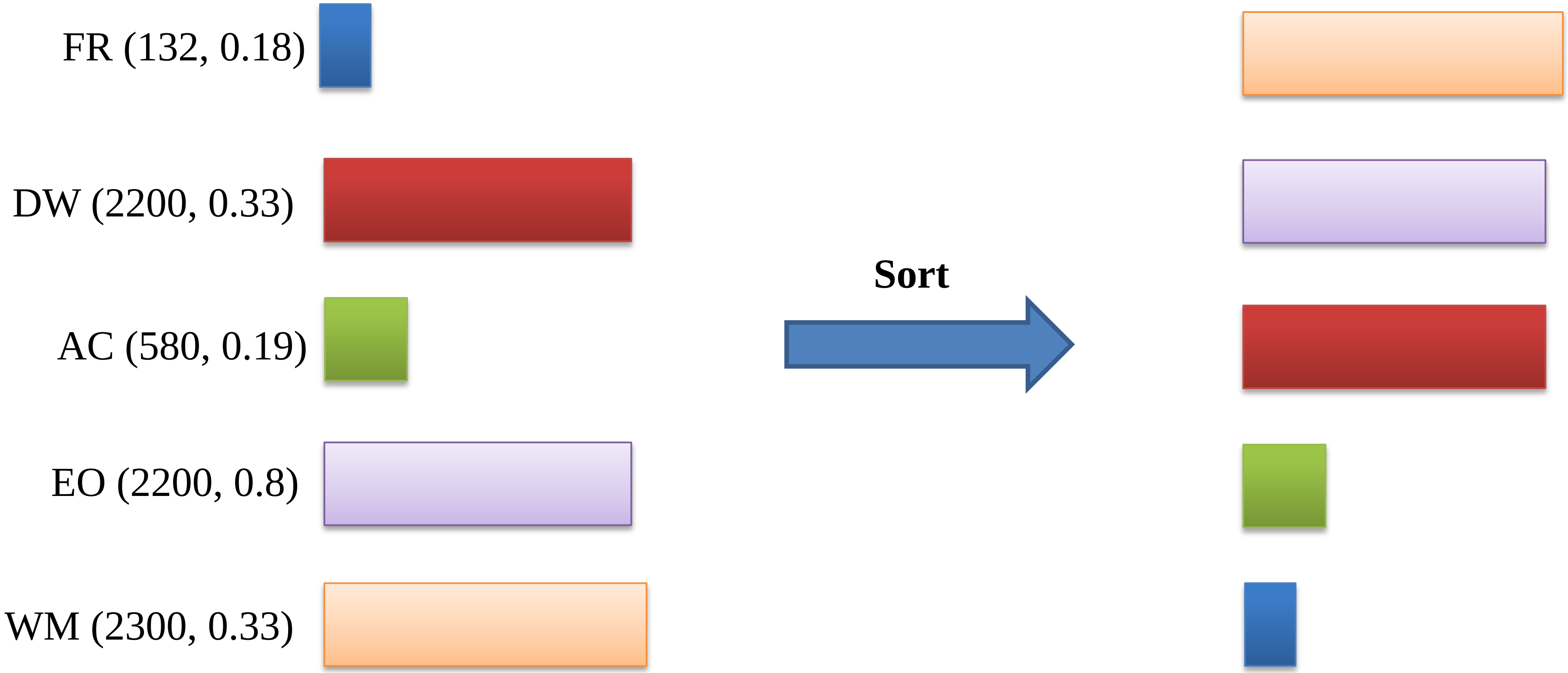
- EDF is **optimal** ⇔ the schedulability test is satisfied
  - System utilization $\mathbf{U = \sum_i U_i}$, $\mathbf{U \leq 1}$.
  - Task utilization $\mathbf{U_i = C_i/T_i}$
- When $\mathbf{U > 1}$ ⇒ simultaneous activaction of loads
  - The upper bound (on the peak load of power demand) is strong to determinate

Load partitioning algorithm

- Assigns load into groups ($\Omega_1,\ldots,\Omega_m$) called **scheduling groups**
- based on heuristic method
  - **First-Fit Decreasing Height (FFDH)**
- **FFDH**
  - Create least possible scheduling groups
  - Order the load in each scheduling groups according to decreasing power demand
  - Ensures satisfiability of the schedulability test in each scheduling group

Load partitioning algorithm

- FFDH **process**
  - Device ($\mathbf{P_i}$, $\mathbf{U_i}$), $\mathbf{P_i}$ is the nominal load power demand and represents the “**Height**“
  - **Phase 1: Sort**



FR (132, 0.18)

DW (2200, 0.33)

AC (580, 0.19)

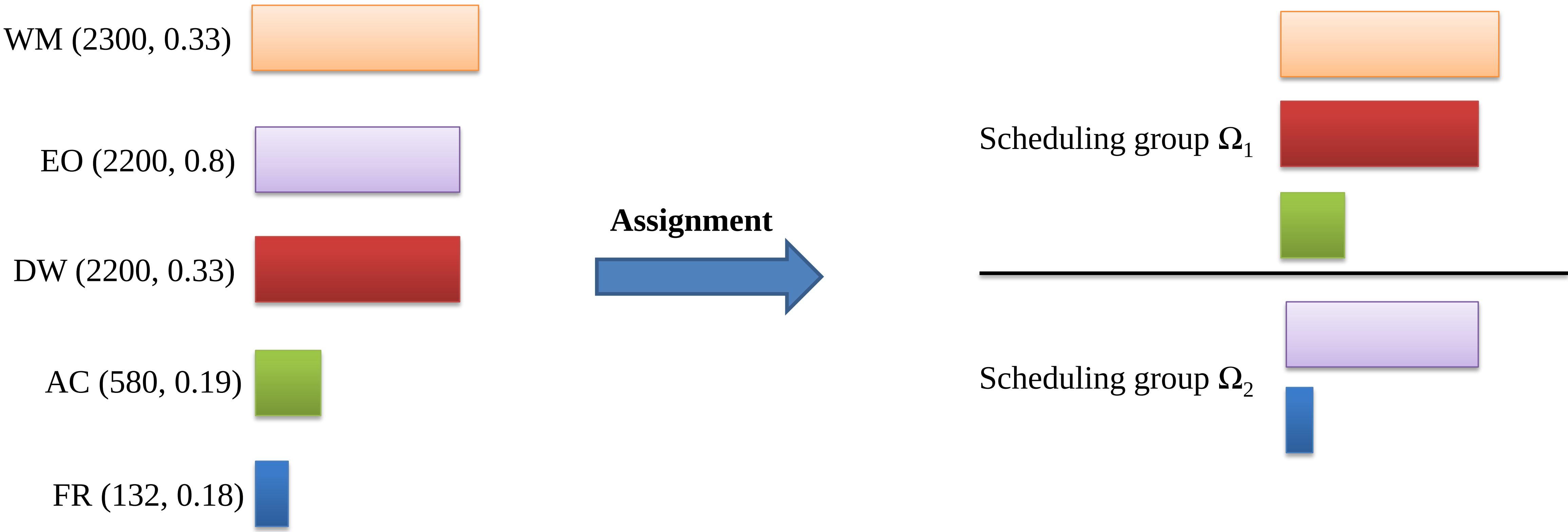
EO (2200, 0.8)

WM (2300, 0.33)

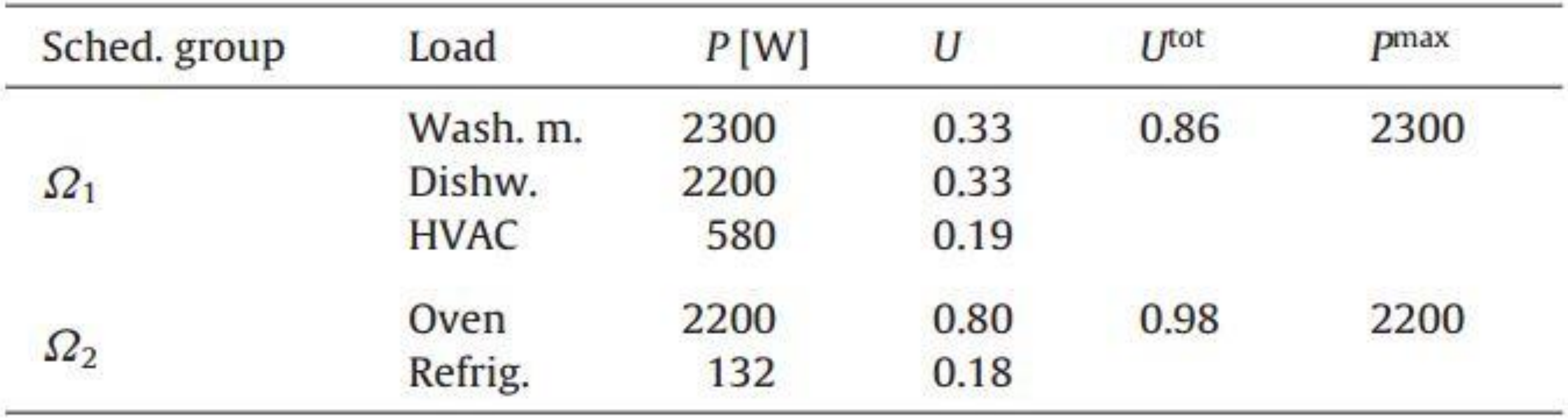
Sort

Load partitioning algorithm

- FFDH **process**
  - **Phase 2: Assignment**



Load partitioning algorithm



Sched. group	Load	P [W]	U	U^{tot}	P^{max}
Ω_1	Wash. m.	2300	0.33	0.86	2300
	Dishw.	2200	0.33		
	HVAC	580	0.19		
Ω_2	Oven	2200	0.80	0.98	2200
	Refrig.	132	0.18		

Picture: D. Caprino et al. (2014) [4]

In short

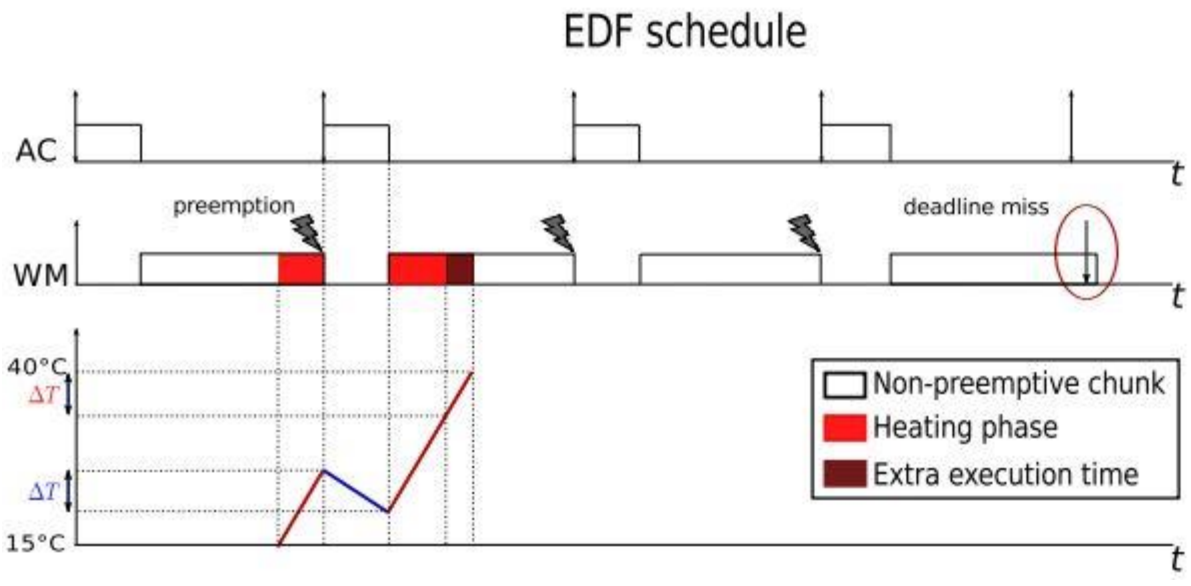
- EDF schedules tasks in each group independently from other groups
  - Only one load is active in each scheduling group at any given time
- Full preemptive system ensures that an upper bound of the peak load can be determinated
  - The maximum peak load is obtained when the loads having the highest power demand are simultaneously activated in all the scheduling groups
  - But can not guarantee the minimun peak load

Disadvantages of a Full-preemptive system

Effects of preemptions on load scheduling

- The tasks can miss their deadline due to Extra-time added to their execution time
  - Violation of timing constraints
- Impacts on lifetime of component (deterioration of appliances)

Effects of preemptions on thermal loads



- An **overhead** is an extra-time added to the “regular” activation time of the load

Effects of preemptions on thermal loads

- The extra-time is calculated as follows
  - $\boldsymbol{\alpha}^{\mathbf{on}}$ be the constant rate of an increasing temperature
  - $\boldsymbol{\alpha}^{\mathbf{off}}$ is constant rate of the decreasing temperature
  - The extra activation time $\Delta\mathbf{C}_{\mathbf{i,j}}$ of the j-th job of the i-th load is

$$\Delta C_{i,j} = \frac{\alpha_i^{\text{off}}}{\alpha_i^{\text{on}}}\, t_{i,j}^{\text{stop}}$$

  - $\mathbf{t}_{\mathbf{i,j}}^{\mathbf{stop}}$ is the sum of the durations of all preemption intervals occurring during the heating phase

LIMITED-PREEMPTIVE REAL-TIME SCHEDULING

Definition

- **Limited-preemptive scheduling** aims to decrease the number of preemptions occurring at run-time
- A **non-preemptive chunk** is a continuous time interval during which the running task is executed without being interrupted

Principle

- Calculate the non-preemptive chunks of each task
- Put the jobs of one task as far as possible within one of its non-preemptive chunks
- A higher priority can not interrupt the running task that is in one of its non-preemptive chunks, consequently
  - It is placed in an queue
  - And activate at the end of the non-preemptive chunk of the running task.

Calculation of non preemptive chunk

- Let $\mathbf{b_i}$ the duration of the largest non-preemptive chunk
- Calculate an upper bound $\mathbf{T^*}$ such that $d_k \leq \mathbf{T^*}$
- Calculate the **Demand Bound Function (DBF)**
- Calculate the value of the **SLACK($\mathbf{d_k}$) function** related to every deadline $d_1, \ldots, d_n$

Calculation of non preemptive chunk

- **Upper bound T***

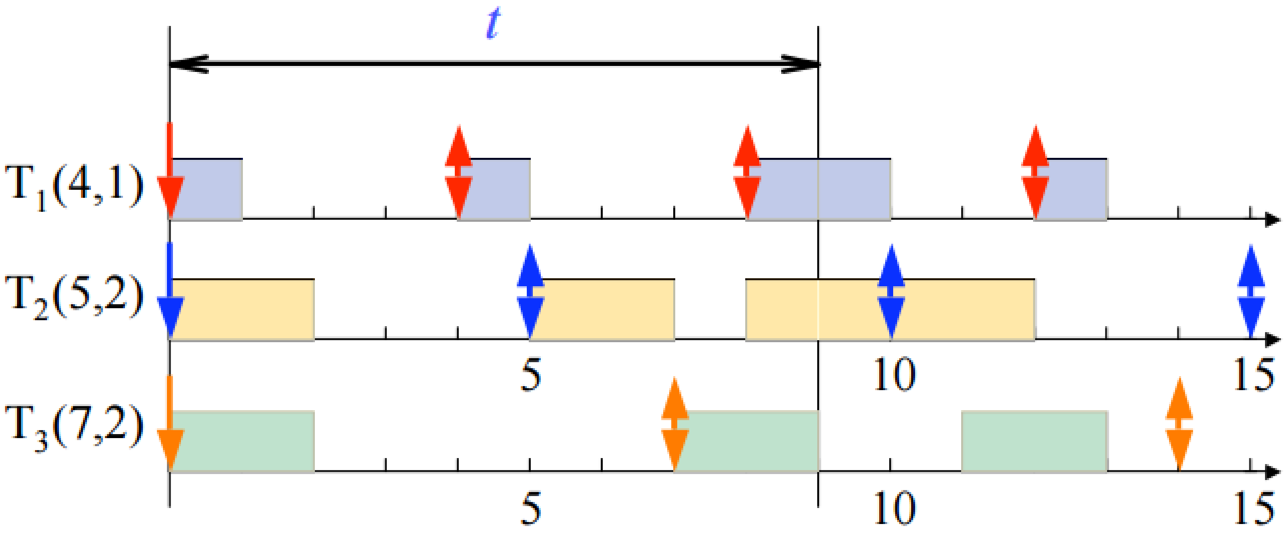
$$T^*(\tau) \overset{\text{def}}{=} \min\left[P, \max\left(d_{\max}, \frac{1}{1-U}\sum_{i=1}^{n} U_i \cdot (T_i - d_i)\right)\right]$$

- Some notation
  - Hyperperiod $P \overset{\text{def}}{=} \mathrm{lcm}(T_1, \ldots, T_n)$
  - Task system τ
  - System utilization U
  - Task Utilization U_i
  - Deadline d_i
  - Minimum inter arrival time T_i
  - Execution time C_i

Calculation of non preemptive chunk

- **Demand Bound Function (DBF)**
  - computes the large number of the execution time of all job that can be generate by the task τ_i whose request time und deadline are in a time interval of length **t**

$$\mathrm{DBF}(\tau_i, t) \overset{\mathrm{def}}{=} \max\left(0, \left(\left\lfloor \frac{t-d_i}{T_i} \right\rfloor + 1\right) \times C_i\right)$$



Task(min inter arrival time, execution time)

Picture: http://www.cis.upenn.edu/~lee/06cse480/lec-real-time-scheduling.pdf

Calculation of non preemptive chunk

- **SLACK(B_k) function**

$$\mathrm{SLACK}(B_1) = B_1 - \sum_{\tau_i \in \tau} \mathrm{DBF}(\tau_i, B_1)$$

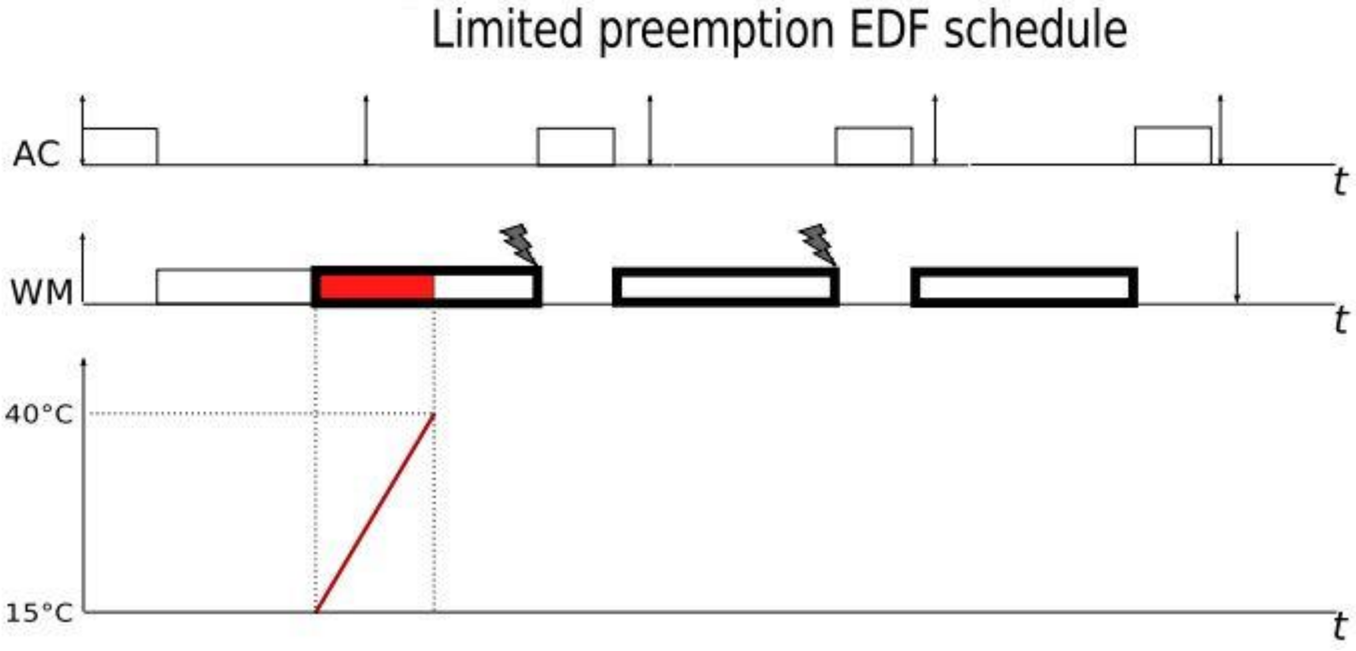
$$\mathrm{SLACK}(B_k) = \min\left(\mathrm{SLACK}(B_{k-1}), B_k - \sum_{\tau_i \in \tau} \mathrm{DBF}(\tau_i, B_k)\right)$$

- It is worth to notice that B_k represent a time-instant, $B_k \leq B_{k+1}$

Calculation of non preemptive chunk

- **Algorithm**
  - Input: task set τ ($\tau_1, \ldots, \tau_n$) and the time-instants B_1, B_2, …
  - Calculate SLACK(B_1)
  - For $k \leftarrow 2, 3, \ldots$ do
    - If $B_k > T^*(\tau)$ then return **feasible**
    - Calculate SLACK(B_k)
    - If SLACK(B_k) < 0 then return **infeasible**
    - If $B_k = d_j$ for some task τ_1, $1 \leq j \leq n$ then $b_j \leftarrow$ SLACK(B_k)

Example of application



SIMULATION RESULTS

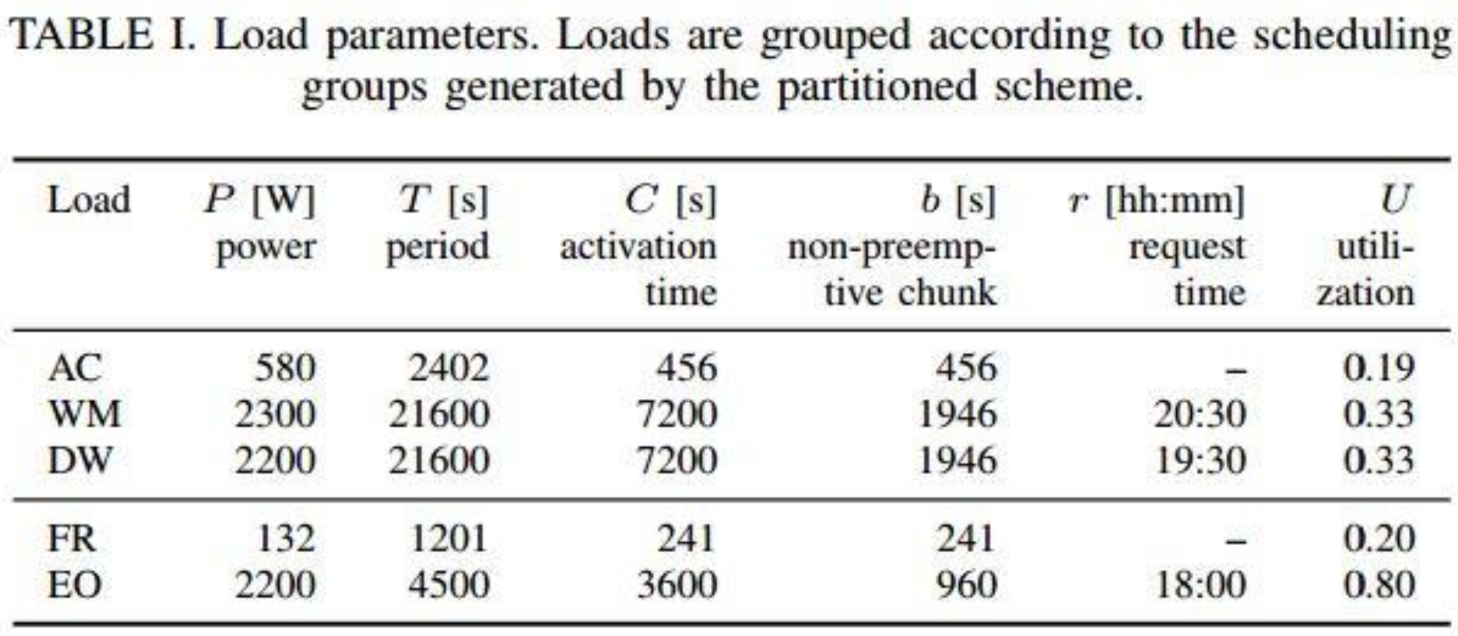
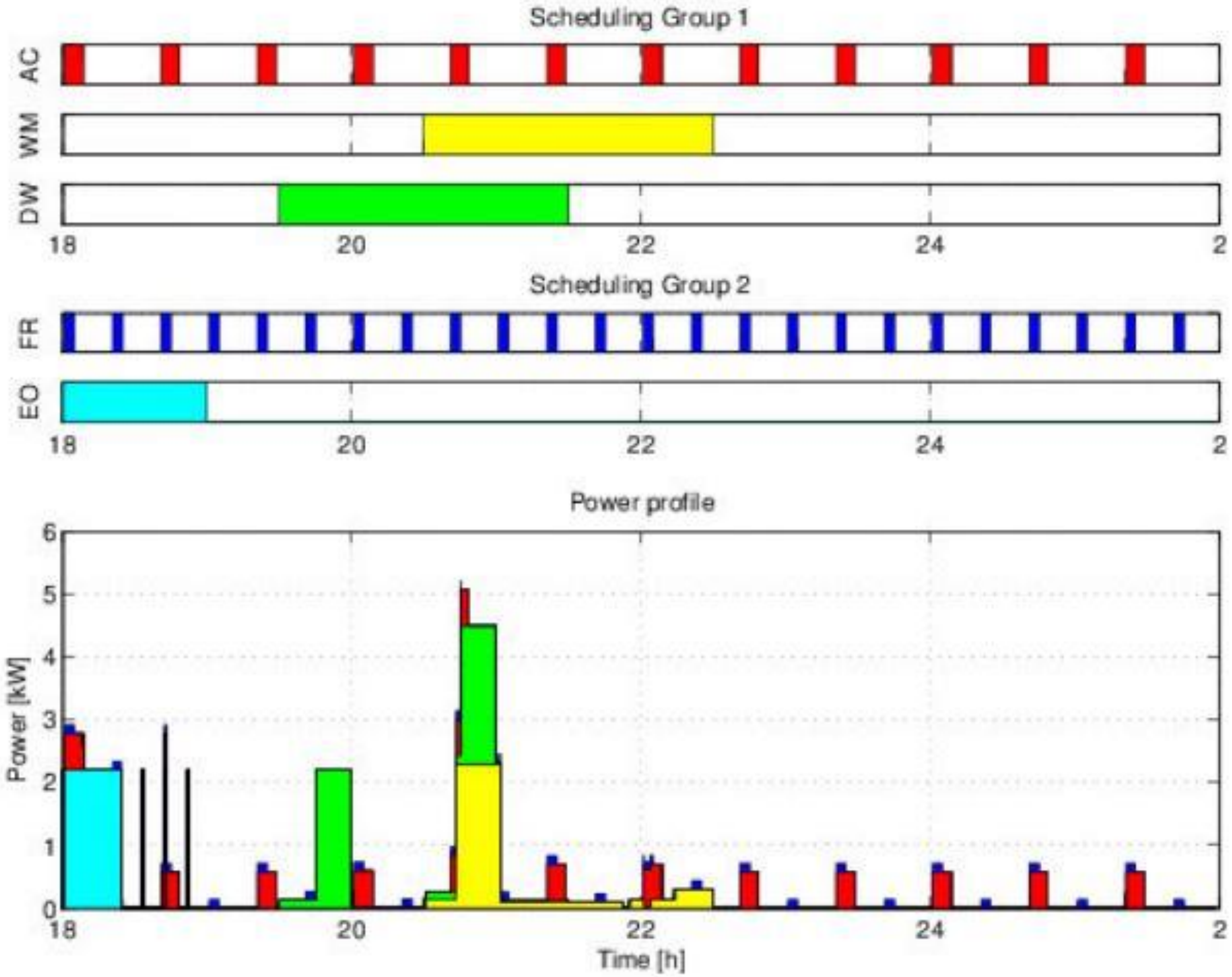
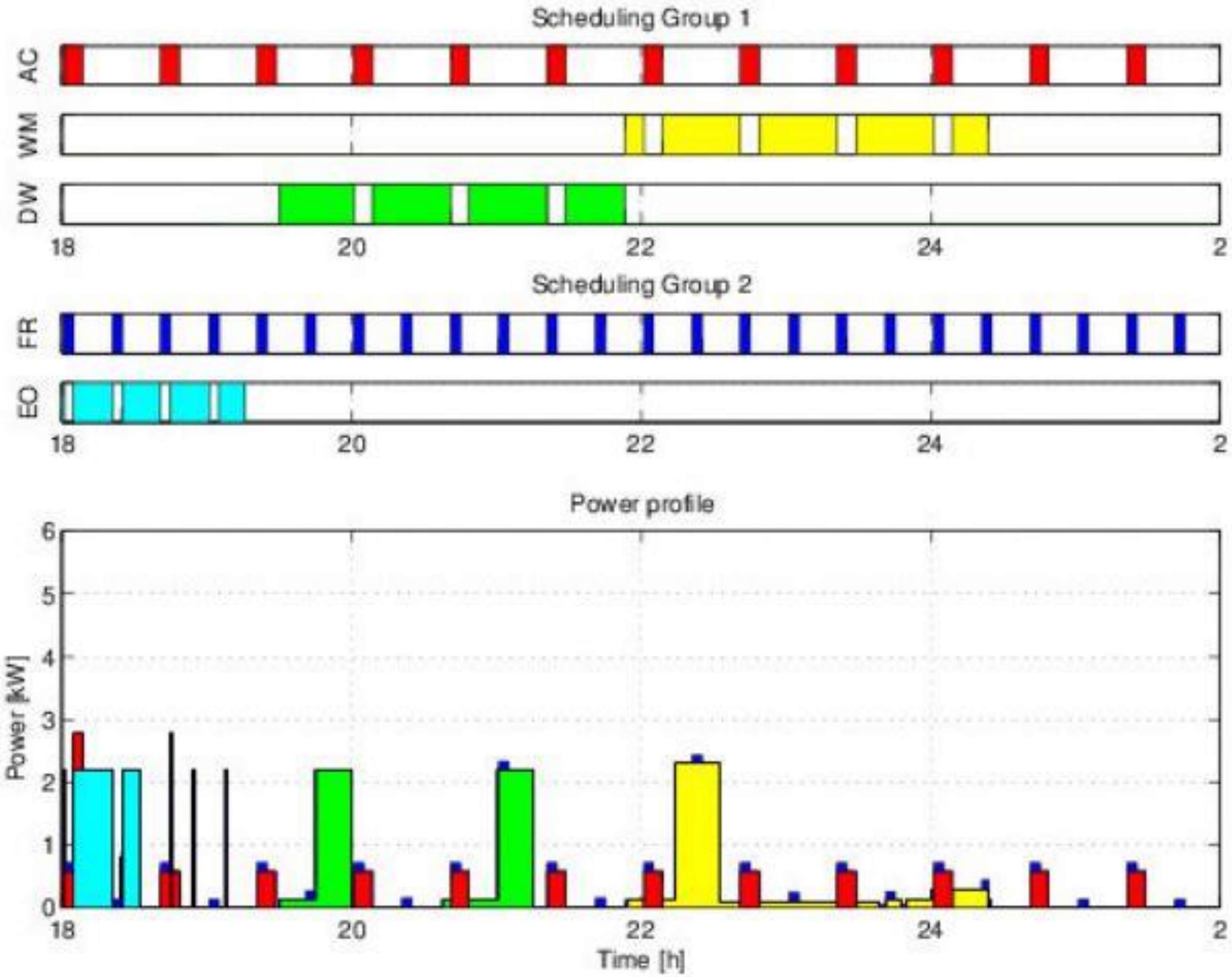


TABLE I. Load parameters. Loads are grouped according to the scheduling groups generated by the partitioned scheme.

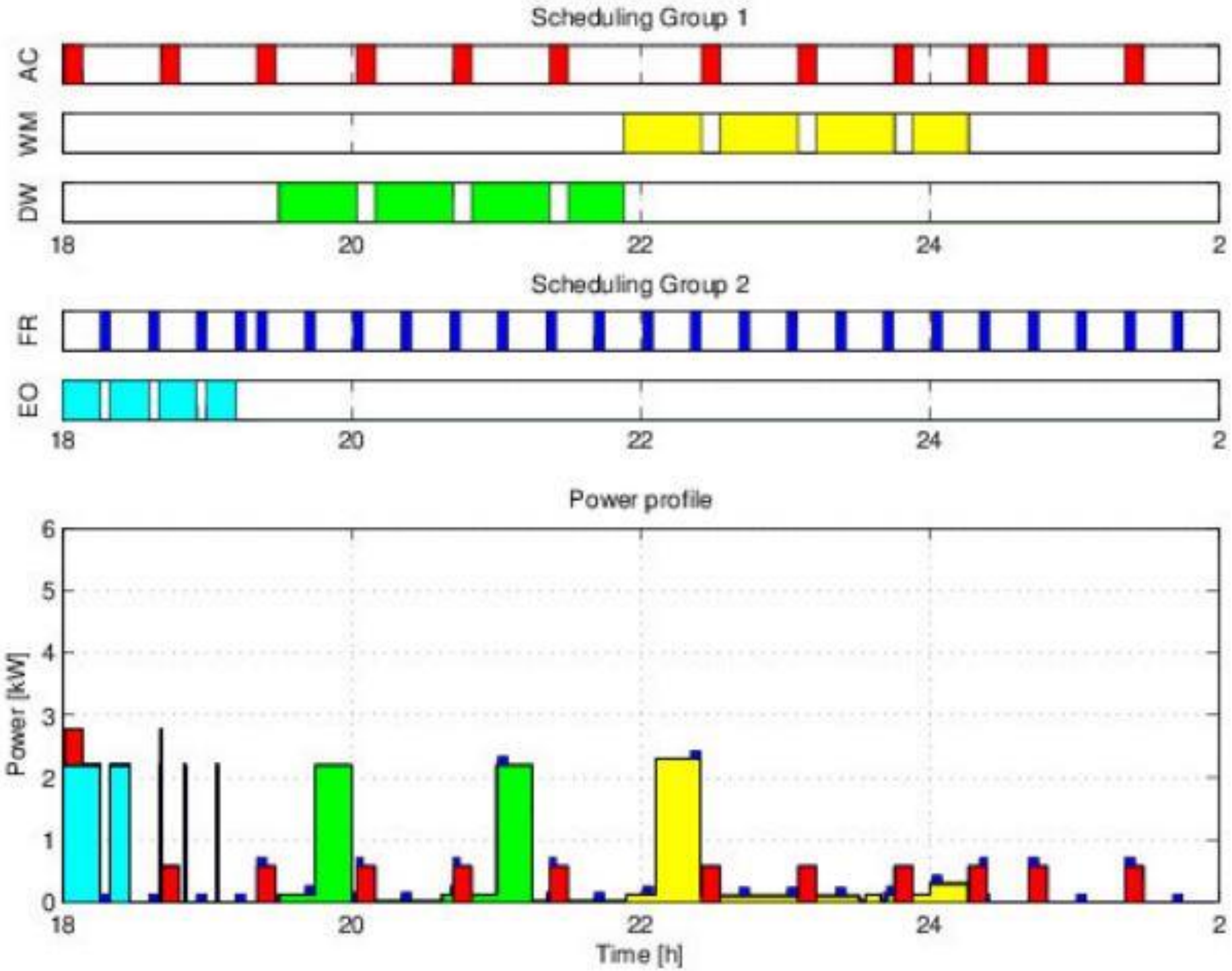
Load	P [W] power	T [s] period	C [s] activation time	b [s] non-preemptive chunk	r [hh:mm] request time	U utilization
AC	580	2402	456	456	–	0.19
WM	2300	21600	7200	1946	20:30	0.33
DW	2200	21600	7200	1946	19:30	0.33
FR	132	1201	241	241	–	0.20
EO	2200	4500	3600	960	18:00	0.80



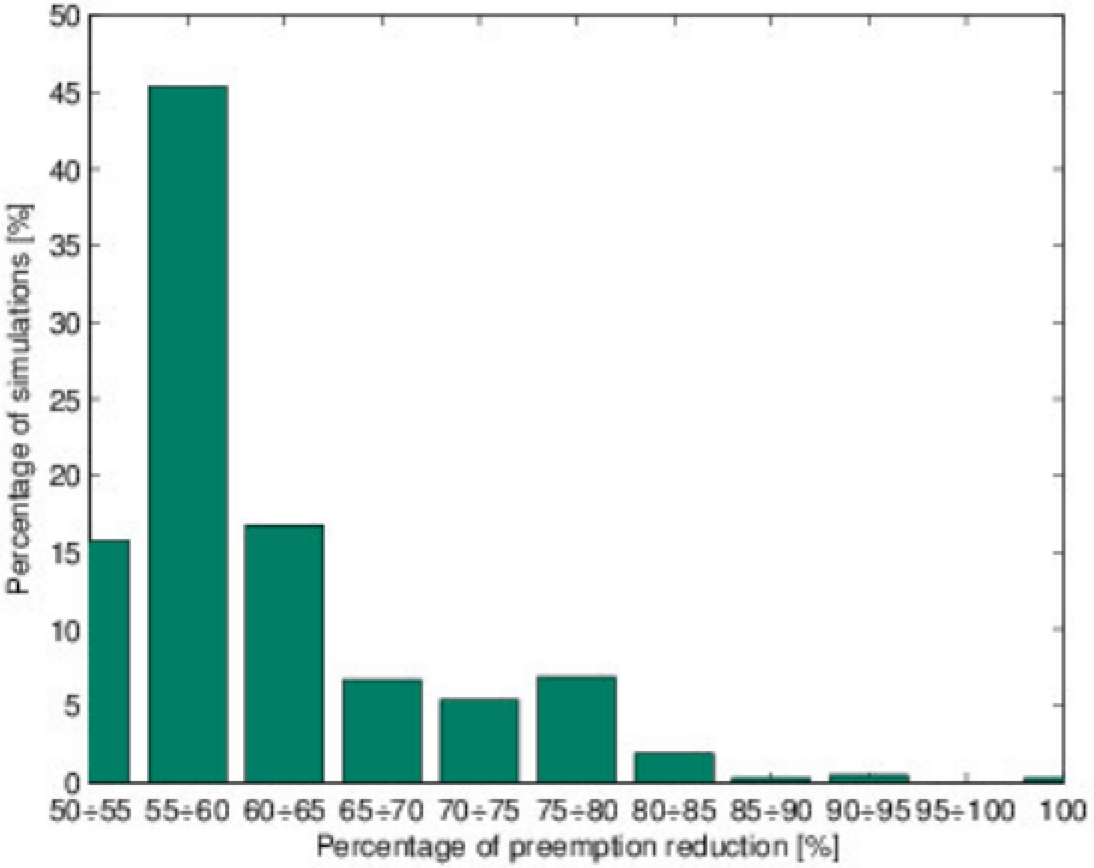
(a) No coordination



(b) Full-preemptive EDF



(c) Limited-preemptive EDF


0
5
10
15
20
25
30
35
40
45
50
Percentage of simulations [%]
50÷55
55÷60
60÷65
65÷70
70÷75
75÷80
80÷85
85÷90
90÷95
95÷100
100
Percentage of preemption reduction [%]

CONCLUSIONS

- The application of limited-preemptive scheduling allows to
    - Obtain a same peak load that in case of a full preemptive scheduling
    - Decrease the number of preemptions on each load
        - Thus reducing the negative impact of preemption on lifetime of appliances
    - Execute completely a task without interruption.
        - The task is placed completely in a non-preemptive chunk
    - The possibility to placed a non-preemptive chunk anywhere in the time does not cause failure of the system schedulability

References

[1] S. Zhou, Z. Wu, J. Li, and X.-P. Zhang, “Real-time energy control approach for smart home energy management system,” Electric Power Components and Systems, vol. 42, no. 3-4, pp. 315–326, 2014.

[2] P. Palensky and D. Dietrich, “Demand side management: Demand response, intelligent energy systems, and smart loads,” IEEE Transactions on Industrial Informatics, vol. 7, no. 3, pp. 381–388, August 2011.

[3] D. Caprino, M. L. Della Vedova, and T. Facchinetti, “Peak shaving through real-time scheduling of household appliances,” Energy and Buildings, vol. 75, no. 0, pp. 133–148, 2014.

[4] S. Baruah, “The limited-preemption uniprocessor scheduling of sporadic task systems,” in Proceedings of the Euromicro Conference on RealTime Systems (ECRTS), July 2005, pp. 137–144.

[5] G. Buttazzo, Hard real-time computing systems: predictable scheduling algorithms and applications, ser. Real-Time Systems Series. Springer, 2005.

[6] T. Facchinetti, “The many faces of real-time scheduling applied to power load management,” in Proceedings of the Third International Conference on Smart Grids, Green Communications and IT Energyaware Technologies (Energy), April 2013, pp. 59–64.

[7] T. Facchinetti and M. L. Della Vedova, “Real-time modeling for direct load control in cyber-physical power systems,” IEEE Transactions on Industrial Informatics, special issue on Information Technology in Automation, vol. 7, no. 4, pp. 689–698, November 2011.

[8] M. L. Della Vedova and T. Facchinetti, “Feedback scheduling of realtime physical systems with integrator dynamics,” in Proceedings of 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), September 2012, conference, pp. 1–8.

[9] M. Bertogna and S. Baruah, “Limited preemption edf scheduling of sporadic task systems,” Industrial Informatics, IEEE Transactions on, vol. 6, no. 4, pp. 579–591, 2010.

[10] A. van Staden, J. Zhang, and X. Xia, “A model predictive control strategy for load shifting in a water pumping scheme with maximum demand charges,” Applied Energy, vol. 88, no. 12, pp. 4785–4794, 2011.

[11] K. Huang, “Demand subscription services - an iterative dynamic programming for the substation suffering from capacity shortage,” IEEE Transactions on Power Systems, vol. 18, no. 2, pp. 947–953, 2003.

[12] A. Subramanian, M. Garcia, D. Callaway, K. Poolla, and P. Varaiya, “Real-time scheduling of distributed resources,” Smart Grid, IEEE Transactions on, vol. 4, no. 4, pp. 2122–2130, 2013.

[13] D. Setlhaolo, X. Xia, and J. Zhang, “Optimal scheduling of household appliances for demand response,” Electric Power Systems Research, vol. 116, pp. 24–28, 2014.

[14] S. Shao, M. Pipattanasomporn, and S. Rahman, “Development of physical-based demand response-enabled residential load models,” Power Systems, IEEE Transactions on, vol. 28, no. 2, pp. 607–614, 2013.

Vielen Dank für Ihre Aufmerksamkeit und wünsche Ihnen frohe Feste

☺