Predictably Flexible Real-time Systems

- a Scheduling Perspective

Gerhard Fohler
Mälardalen University, Sweden
www.idt.mdh.se/gfr

Real-time Computing Systems

Classic

- airplanes
- industrial production
- medical devices

but also

- cars
- phones
- consumer electronics

other terms

- „embedded systems“
  - computer integrated in system function
  - focus on hardware
  - usually also real-time
- „real-time“ – e.g. stock market
  - transmission delay not perceived by user
  - *not real-time system per se*

Roadmap

- activation paradigms
  - determinism vs. flexibility
  - activation paradigms and their implications
- combined approach
- applications
  - real-time and multimedia

Roadmap

- activation paradigms
  - determinism vs. flexibility
  - activation paradigms and their implications
- combined approach
- applications
  - real-time and multimedia

Activation Paradigms

- activation of activities - tasks
  - when are events recognized?
  - who initiated activities?
  - when are decisions taken?

- event triggered – ET
  - event initiates activities in system immediately

- time triggered – TT
  - activities initiated at predefined points in time

Properties – Time Triggered

- offline scheduling
- scheduling table
- slots – time triggered activitation of dispatcher
- runtime dispatcher executes decision in table

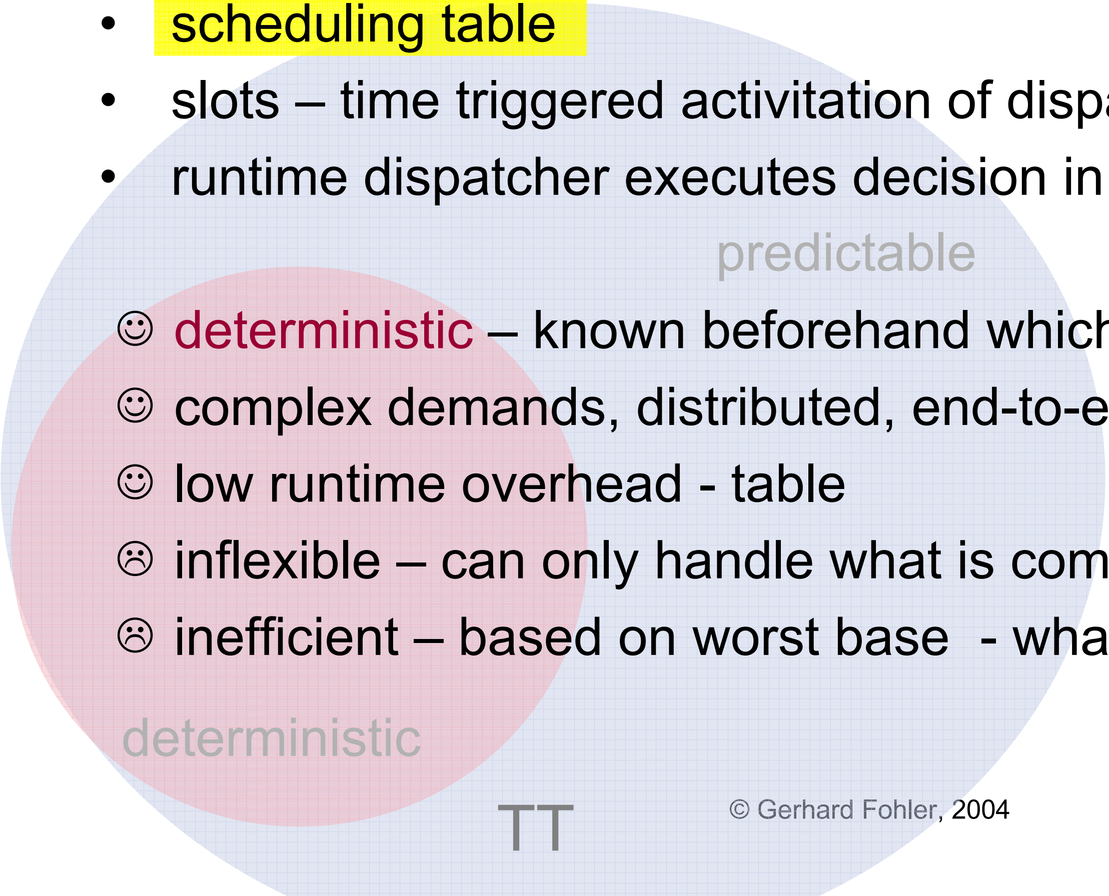
☺ deterministic – known beforehand which activity running when

☺ complex demands, distributed, end-to-end, jitter, …

☺ low runtime overhead - table

☹ inflexible – can only handle what is completely known before

☹ inefficient – based on worst base - what if it does not occur?



Properties – Event Triggered

- online scheduling, priority driven
- event activates scheduler which takes decision
- priority rules + test
  - earliest deadline first (dynamic priority)
  - fixed priority

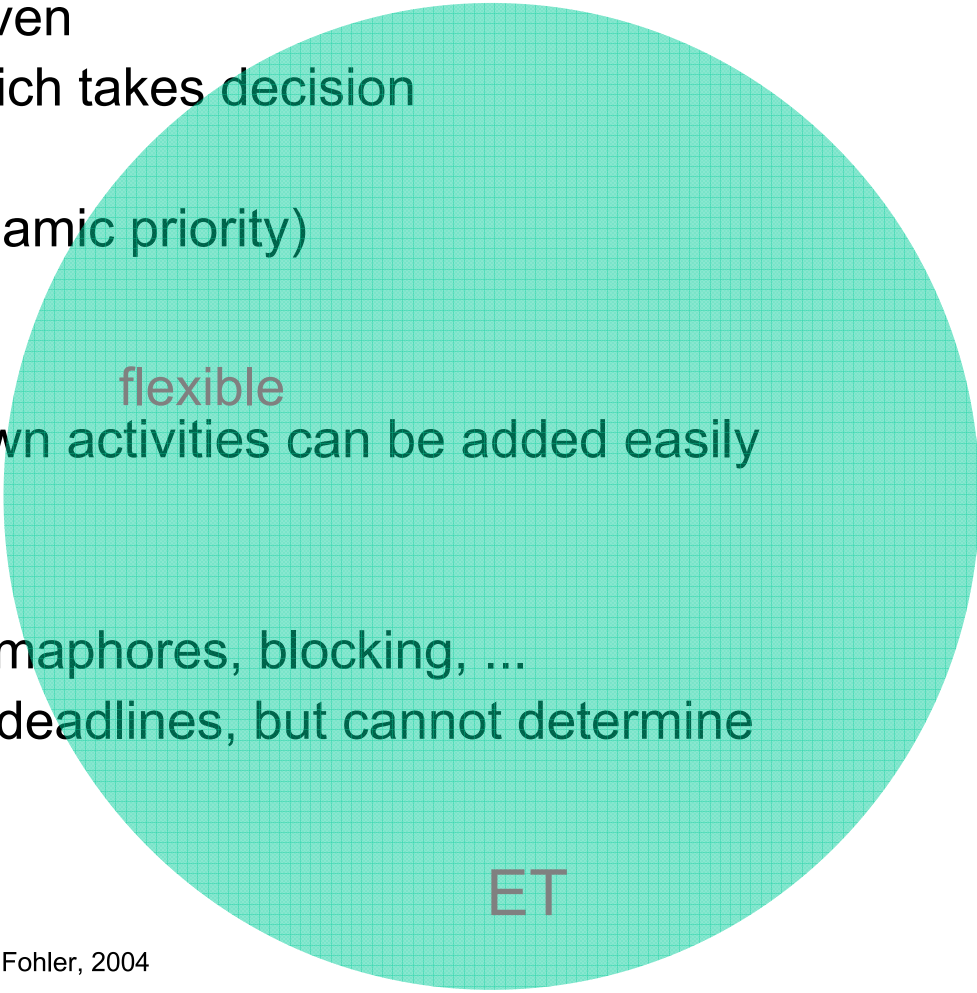
☺ flexible – not completely known activities can be added easily

☺ widely used

☹ only simple constraints

☹ high runtime overhead for semaphores, blocking, ...

☹ limited predictability – keeps deadlines, but cannot determine when exactly



© Gerhard Fohler, 2004

Effects on Design

- activation paradigm is central design decision
- “either – or” decision
  - advantages of one method at expense of those of other
  - demands outside paradigm need to be “squeezed in”
- system wide implications
  - same properties (e.g., cost) for *all* activities
  - mostly highest level

monolithic approaches - “power plant” approaches

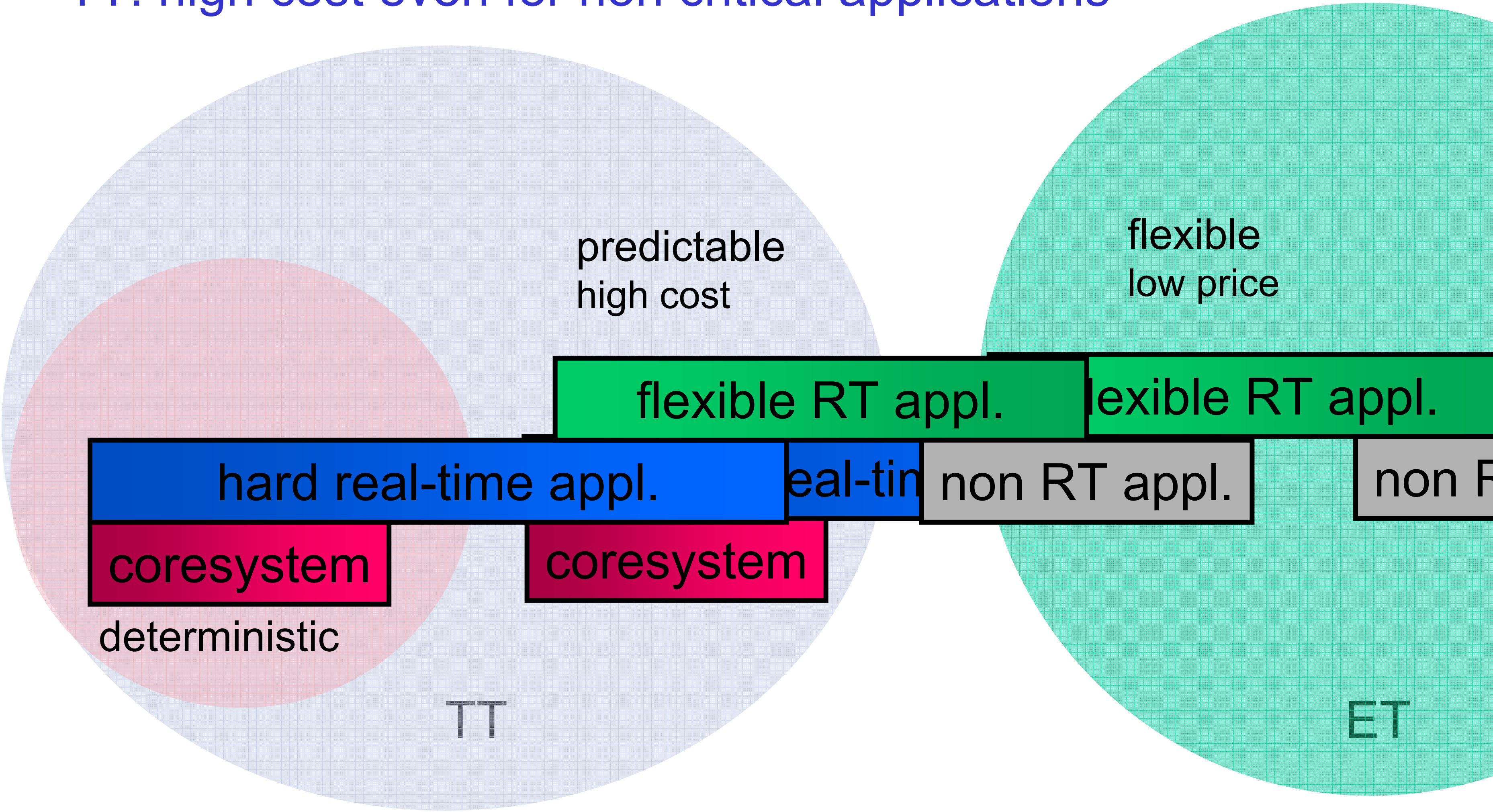
- single system for single application
- single paradigm for single class of demands
- high effort and cost

Novel Applications

mix of activities and demands

- core system with high demands
  - strict timing behavior
  - safety critical, fault tolerant
  - proven and tested for worst case
- hard real-time applications
  - temporal correctness, etc.
- flexible real-time applications
  - not completely known
  - some deadlines can be missed
- non real-time activities
  - must not disturb real-time activities

TT: high cost even for non critical applications



ET: not deterministic behavior of critical activities

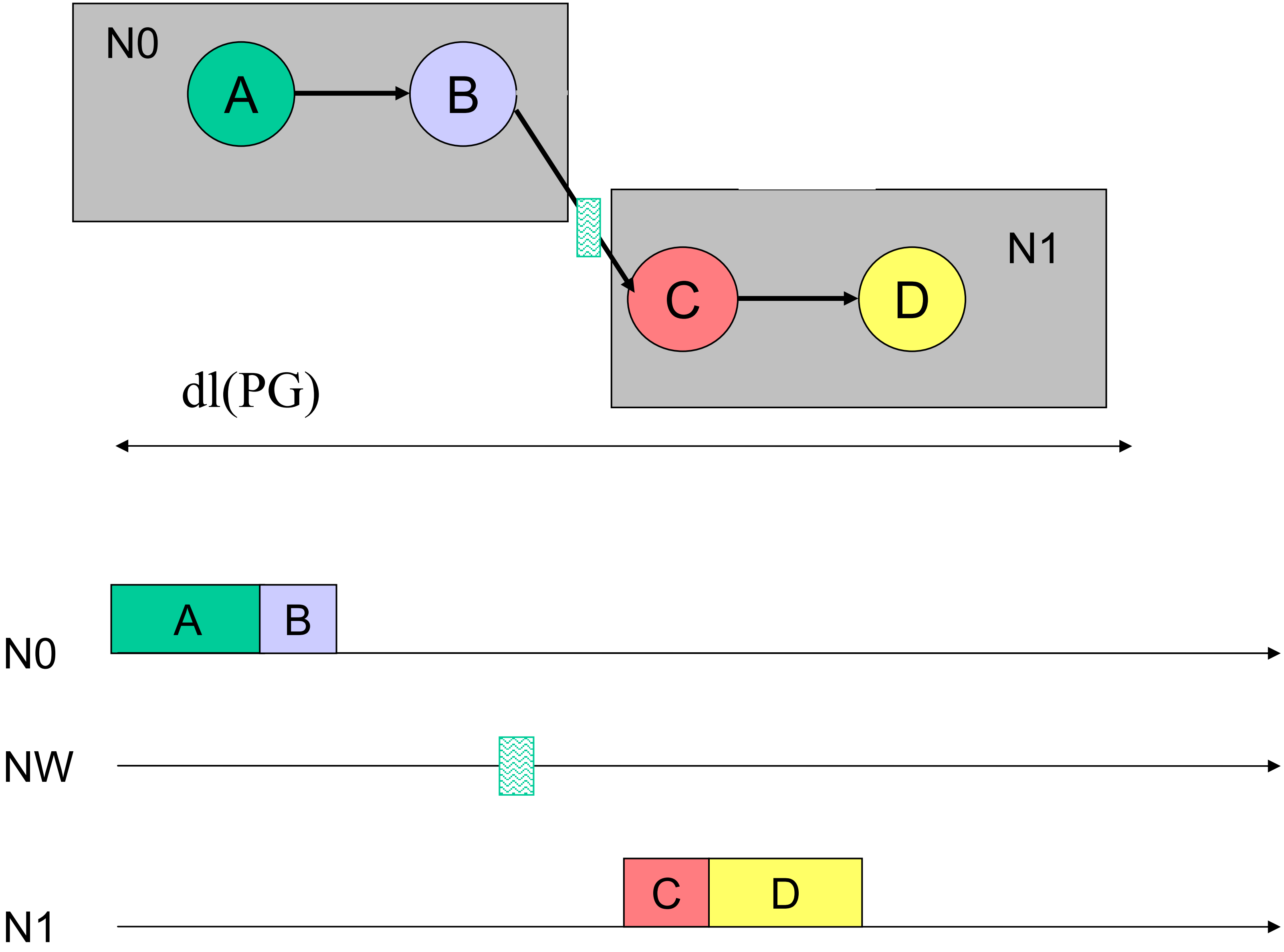
Roadmap

- activation paradigms
  - determinism vs. Flexibility
  - activation paradigms and their implications
- combined approach
- applications
  - real-time and multimedia

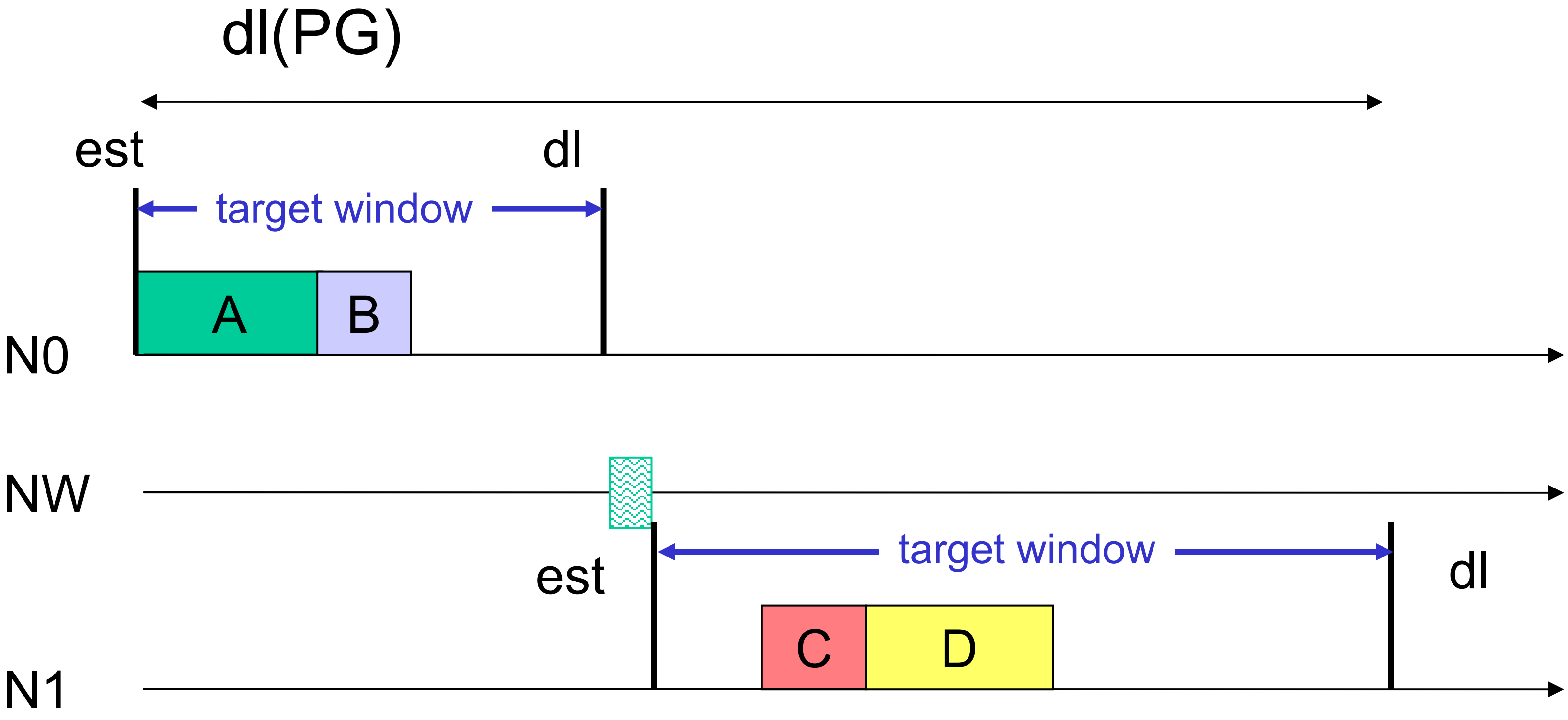
Offline Scheduling

- general, complex (temporal) constraints
- offline scheduler
  - resolves demands
  - constructs *single* solution meeting all demands
  - table for least common multiple of periods

- complete information needed beforehand
- no flexibility


N0
A
B
N1
C
D
dl(PG)
N0
A
B
NW
N1
C
D

- analysis of offline schedule and demands
- limit task executions - target windows
  - demands fulfilled, if tasks execute within target windows
  - starttime, deadline pairs
- ready for dynamic, priority driven scheduling; event triggered



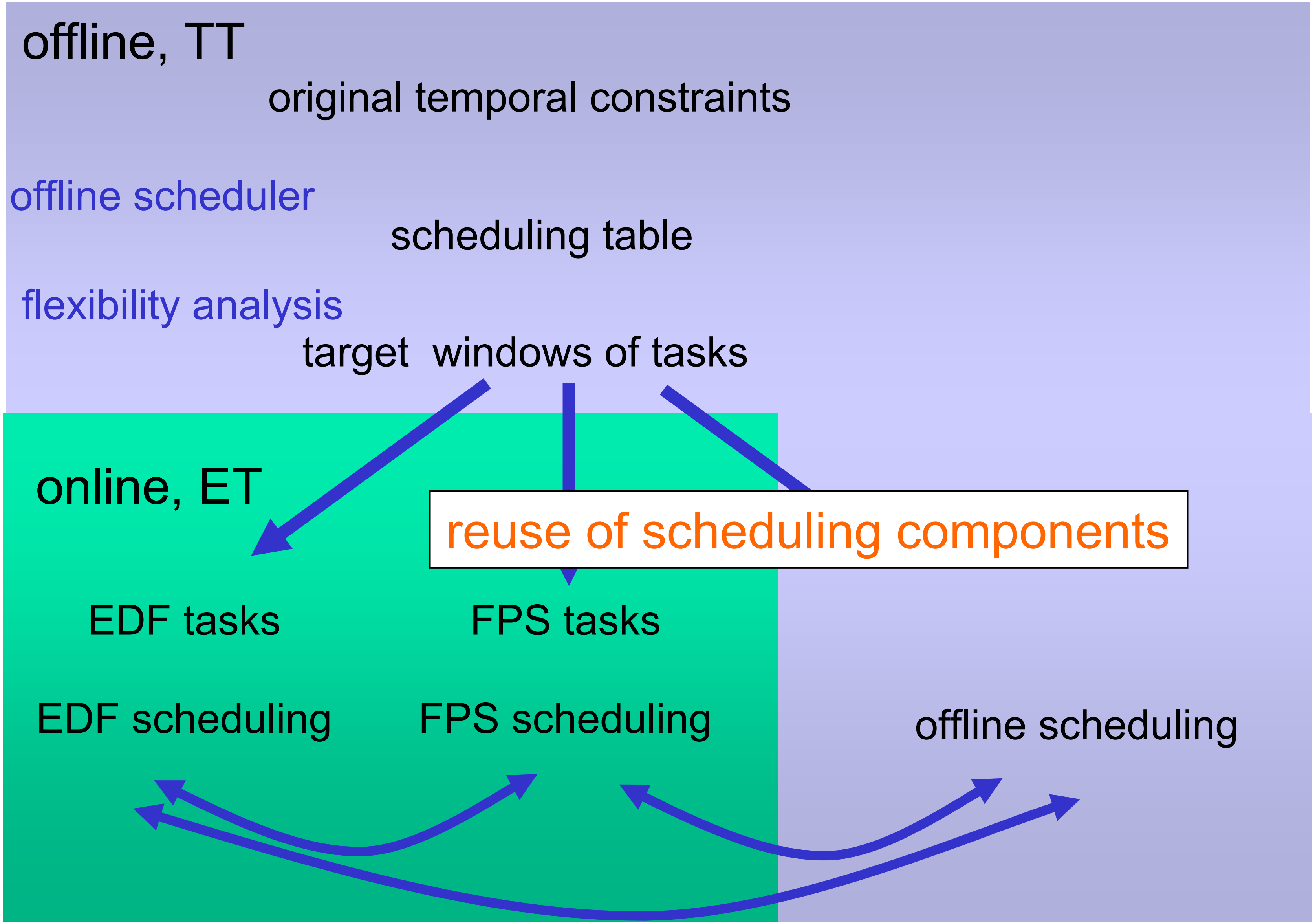
© Gerhard Fohler, 2004

Slot shifting

for completeness, all algorithms

- theoretical results proven (optimality etc.) (many formulae)
- implemented
- evaluated
- etc

		Periodic with constraints		Sporadic	Aperiodic	
		Simple	Complex		Firm	Soft
		• Periods • Deadlines • Start times	• End-to-end dl • Inst. separation • Distribution • Jitter etc.	Minimum separation between instances	• Deadlines • Guarantee	• No dl
Offline	Sch	x	x			
	Test			x		
Online	Sch	x	x	x	x	x
	Test				x	



offline, TT

original temporal constraints

offline scheduler

scheduling table

flexibility analysis

target windows of tasks

online, ET

reuse of scheduling components

EDF tasks

FPS tasks

EDF scheduling

FPS scheduling

offline scheduling

Predictable Flexibility

target windows control flexibility of task execution

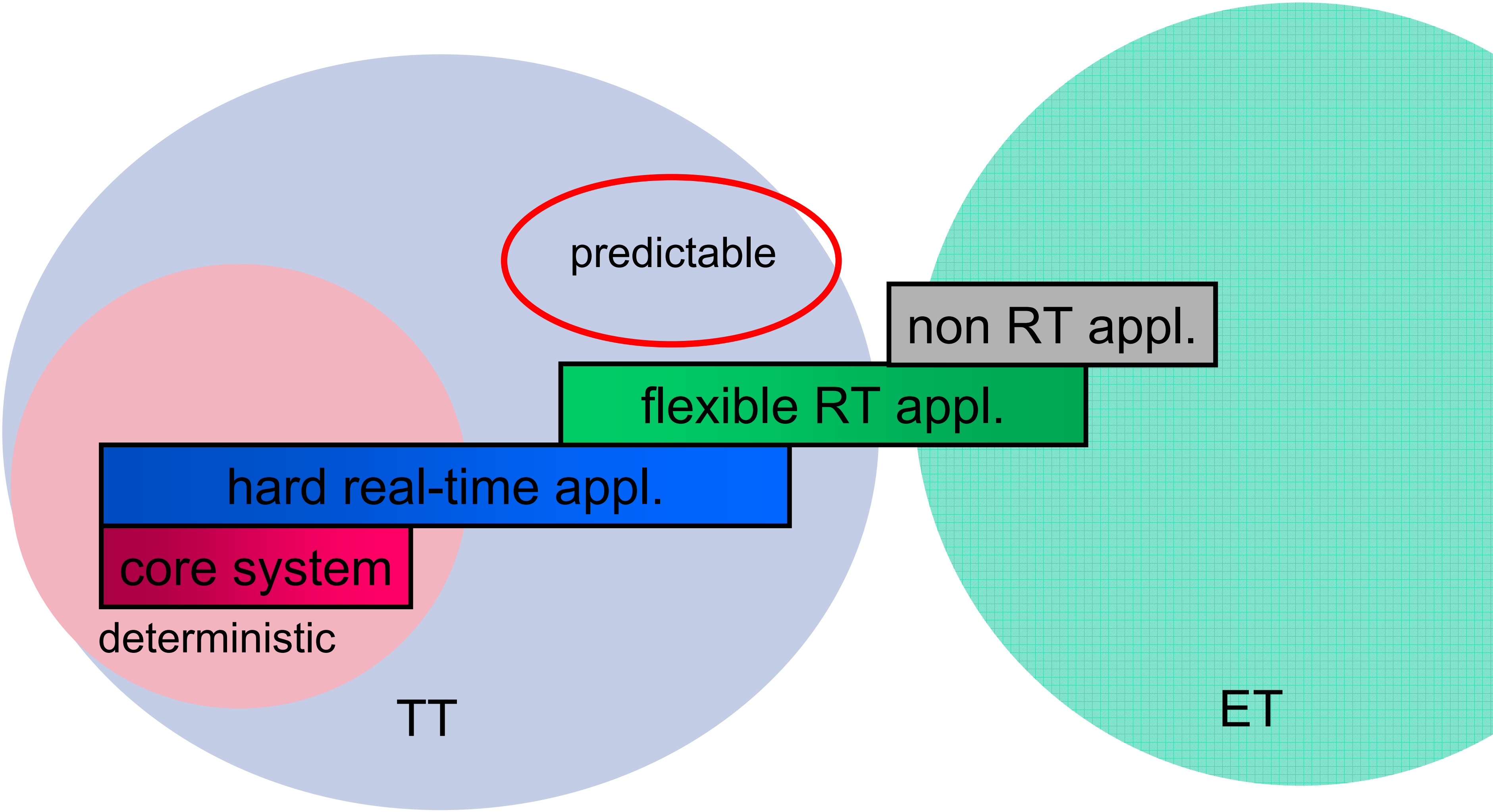
- target window = original task execution
  no flexibility, original schedule
- target window after flexibility analysis
  flexibility of execution while meeting demands
- reduced target windows
  reduced flexibility, e.g., for jitter control
- modifying target windows selects flexibility of tasks individually

Meeting Novel Application Demands

- core system
  offline scheduling
- hard real-time applications
  offline scheduling or online scheduling
- flexible real-time applications
  combined offline/online approach
- non real-time activities
  together with combined offline/online
- flexibility individually configured

- guaranteed tasks protected

Predictably flexible real-time systems



© Gerhard Fohler, 2004

Roadmap

- activation paradigms
  - determinism vs. flexibility
  - activation paradigms and their implications
- combined approach
- applications
  - real-time and multimedia

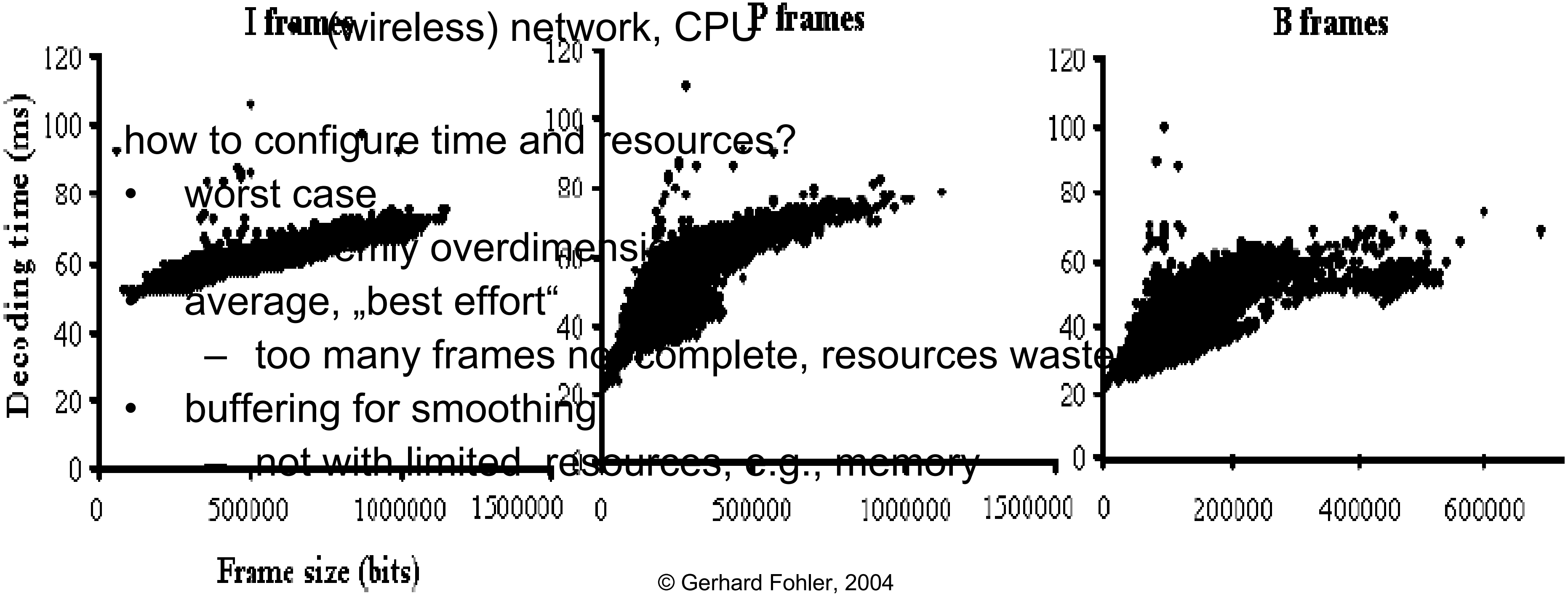
Video streaming

large variations

- size and structure of MPEG streams
  - strong compression - DVD, fast compression- live
  - availability of resources
    - (wireless) network, CPU

how to configure time and resources?

- worst case
  - inherently overdimensioned
- average, „best effort“
  - too many frames not complete, resources wasted
- buffering for smoothing
  - not with limited resources, e.g., memory



© Gerhard Fohler, 2004

Real-time Resource Management

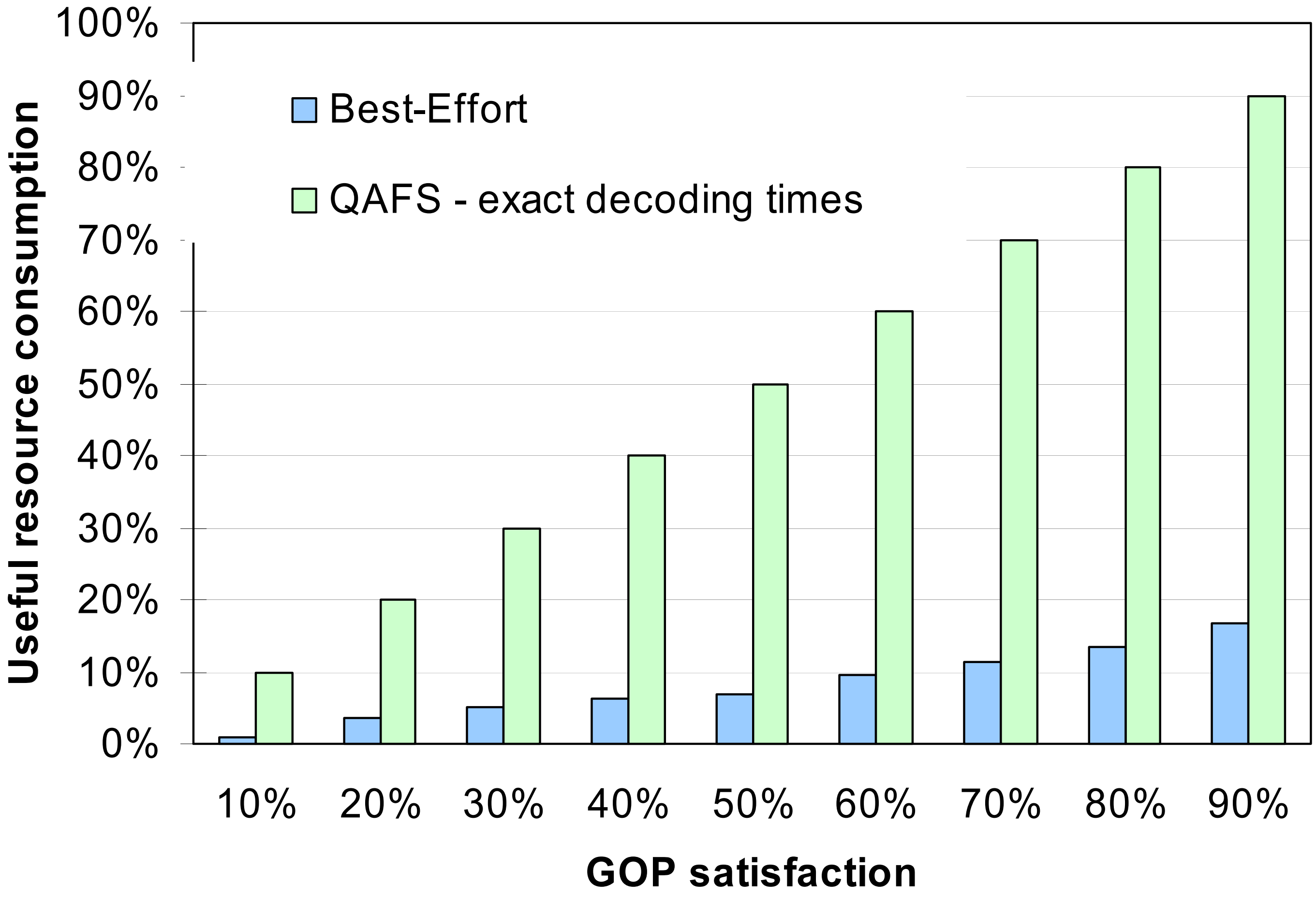
- guarantee of basic budgets (critical)
- stream adaption (flexible)
  - quality aware frameskipping
  - start decoding only
    of what can be completely decoded (real-time methods)
  - resources not wasted for incomplete frames
  - efficient resource use
  - Quality aware frame skipping - QAFS
- processing of additional frames, reclaiming

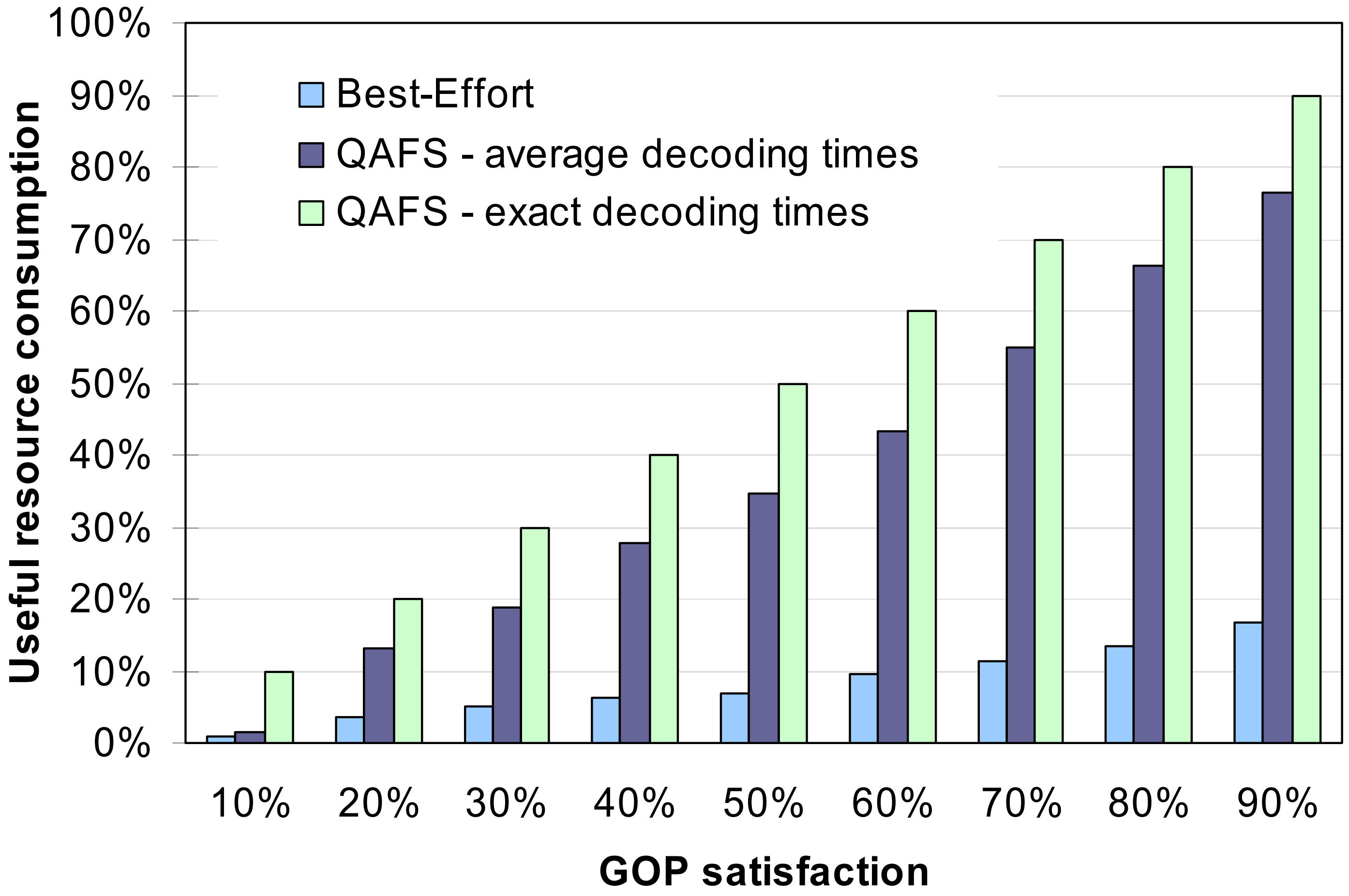
QAFS - Analysis

- offline time tr
- MPEG decod
- importance
- how efficiently
  MPEG decoding
  without disturbing time triggered tasks?

Which portion of used
resources contribute to
picture quality, are not
wasted

lines
ion of
needed
or
ces are
available


Best-Effort
QAFS - exact decoding times
Useful resource consumption
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
GOP satisfaction


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Useful resource consumption
Best-Effort
QAFS - average decoding times
QAFS - exact decoding times
10%
20%
30%
40%
50%
60%
70%
80%
90%
GOP satisfaction

Summary

- activation paradigms
  - time triggered
    - offline scheduling, determinism, no flexibility
  - event triggered
    - online scheduling, flexibility
  - predictable flexibility by combination of both
- design independent of paradigm, reuse
- management of individual activites instead of systemwide
- applications, video streaming

THE END