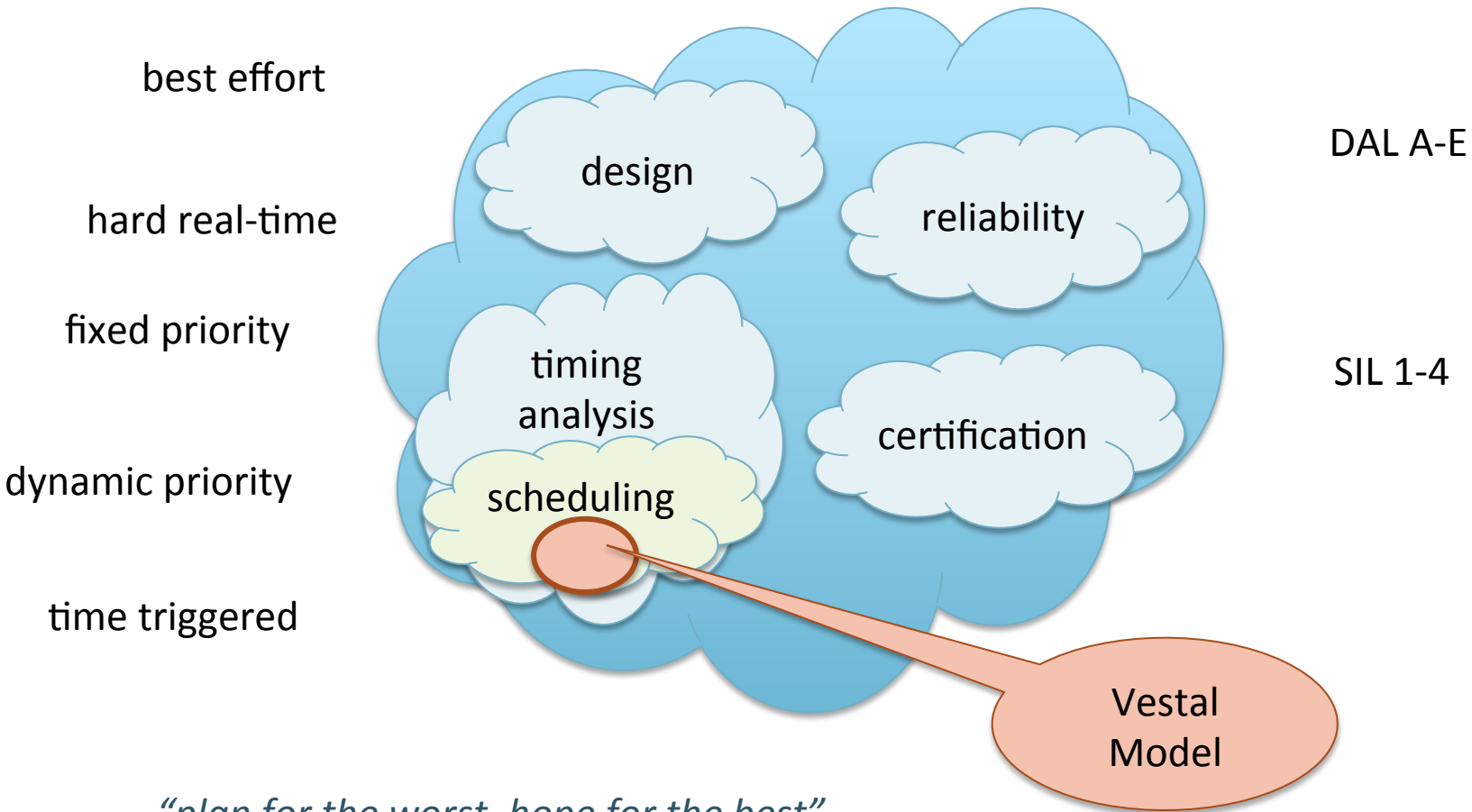


MIXED CRITICALITY SCHEDULING IN TIME-TRIGGERED LEGACY SYSTEMS

Jens Theis, Gerhard Fohler
TU Kaiserslautern



**MIXED CRITICALITY
SCHEDULING
IN TIME-TRIGGERED
LEGACY SYSTEMS**



**MIXED CRITICALITY
SCHEDULING***
IN TIME-TRIGGERED
LEGACY SYSTEMS

*Vestal model

MIXED CRITICALITY
SCHEDULING
IN **TIME-TRIGGERED**
LEGACY SYSTEMS

TT is from MARS

- MA(intainable) R(eal-time) S(ystem)
- TU Vienna, Hermann Kopetz
- MARS predecessor to TTP

TTTech

*“In a time-triggered system,
all activities are initiated by
the progression of real-time.”*

time trig·ger·ed ['taim 'trigərd]

activities initiated at predefined points in time

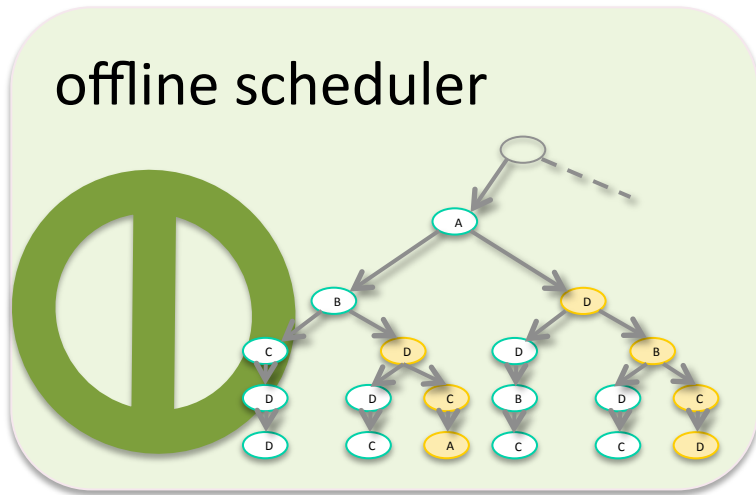
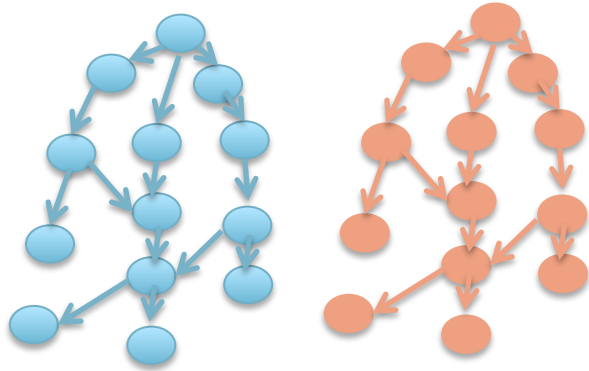
everything planned before system is deployed

How?

offline scheduling - scheduling table

- complex constraints
- retries possible
- slots – time triggered activation of dispatcher
 - period of dispatcher minimum granularity in system
- runtime dispatcher executes decision in table

Offline Workflow



Which cost?

everything planned before system is deployed

- need to know everything
 - all environmental situations
...and time of occurrence
 - all task parameters
...including arrival times
 - all system parameters
...for entire lifetime
- very high cost
- no flexibility

Which benefit for that price?

everything planned before system is deployed

- *know everything* before runtime!
- offline schedule (table)
 - complex constraints, distributed, end-to-end, jitter, ...
 - schedulability test “constructive proof”
- low runtime overhead
- simple fault-tolerance, e.g., replica determinism
- straightforward extension of constraints
- reduced pessimism of schedulability test
- ...

TT and certification

everything planned before system is deployed

TT popular with certification authorities
e.g. avionics

ET requires “for all” proof
all situations, even ones never happening

TT “single case” proof
can look at single scheduling table
test space reduced dramatically



TT and certification

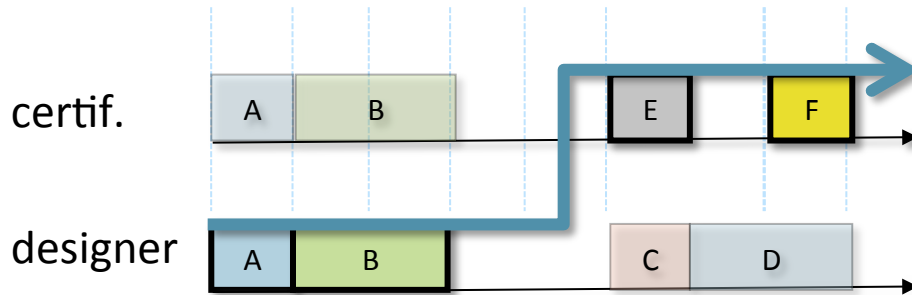
activities initiated at predefined points in time

strong run-time control
temporal enforcement

example fire drill
(vintage footage from 1991)
[RTSS:11]



TT mixed crit. – mode changes



switch between tables
- mode changes

- [Baruah:RTSS11]:
proof of concept

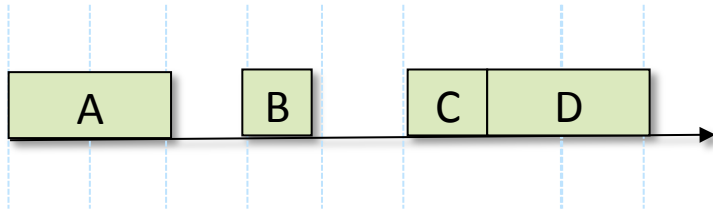
- [Theis:WMC13]:
next talk

require

new scheduling tables
recertification

MIXED CRITICALITY
SCHEDULING
IN TIME-TRIGGERED
LEGACY SYSTEMS

TT mixed crit. – legacy



single, existing
scheduling table

how to add more criticalities?

offline:

include high criticality tasks
in (unchanged) scheduling table

runtime:

provide (efficient) switch to high
criticality tasks

Slot Shifting - recap

originally

adding controlled flexibility
to TT scheduling tables

keep constraints of TT tasks

low runtime overhead



G. Buttazzo & I. Puaut,
The real-time quiz,
25th ECRTS

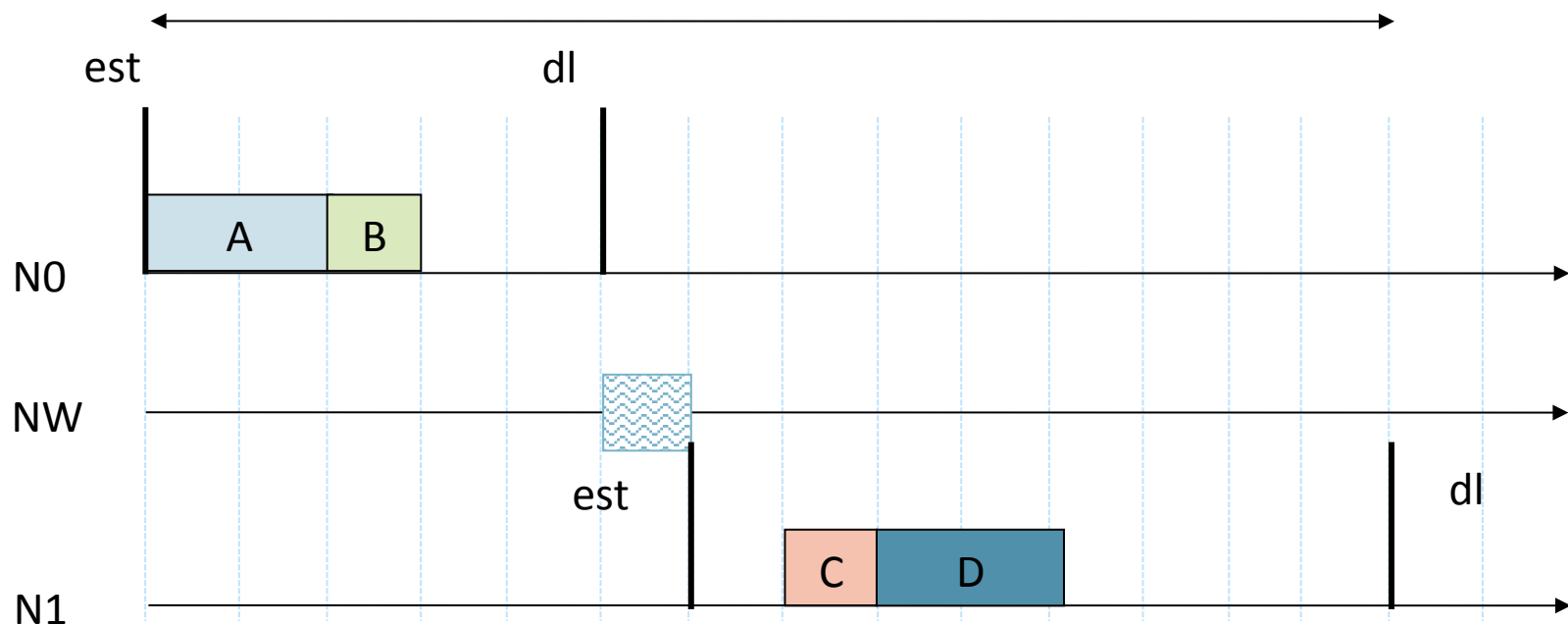
Slot Shifting - Offline

offline scheduler has to pick single solution; others possible

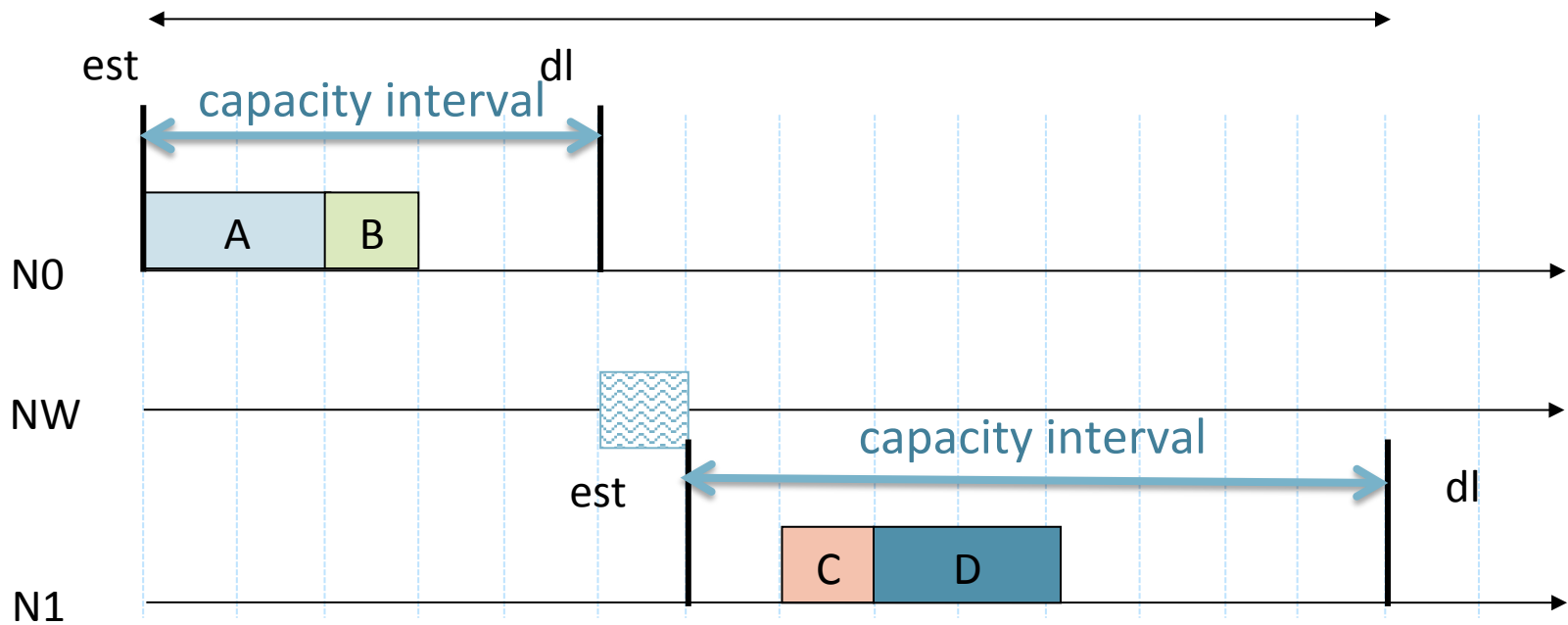
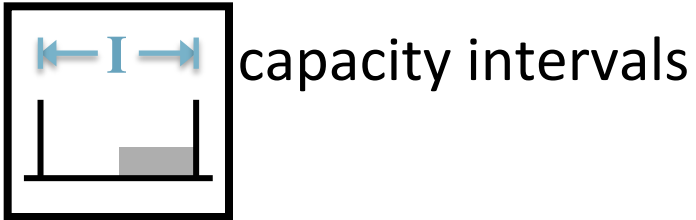
- flexibility in schedule, while feasible

shift offline tasks

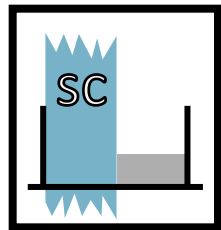
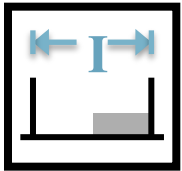
- how much?



Slot Shifting – Offline, ctd.

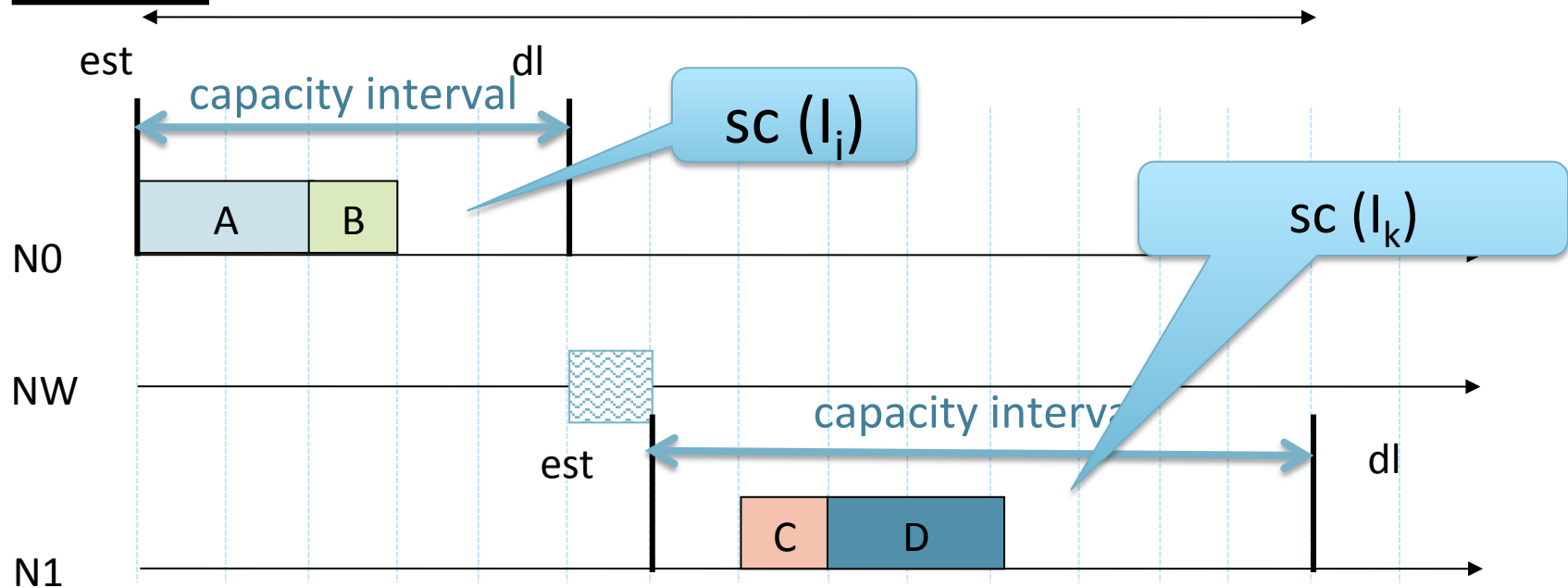


Slot Shifting – Offline, ctd.

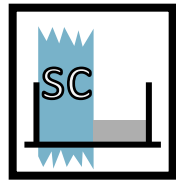
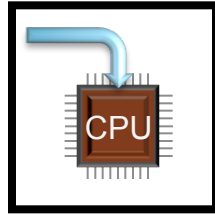
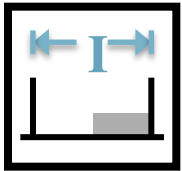


spare capacities

- include aperiodic tasks
 - guarantee tests
 - etc
- ...while keeping TT constraints



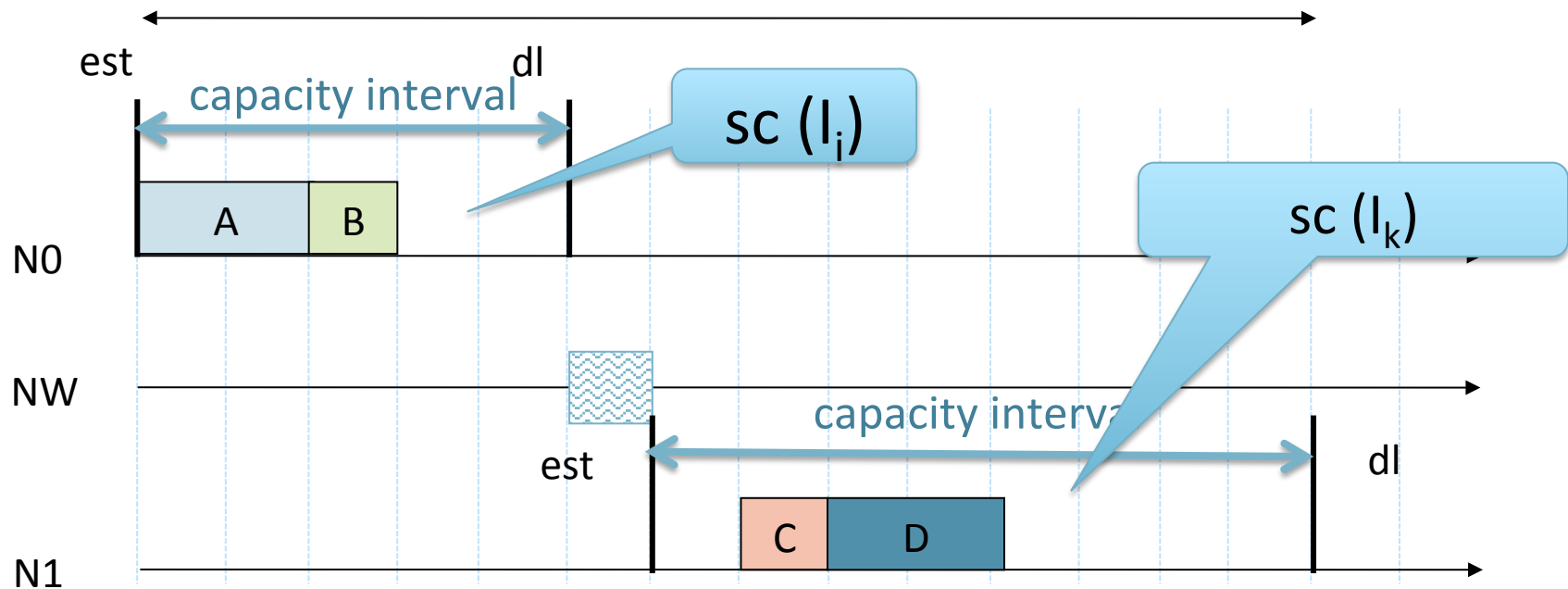
Slot Shifting – Online



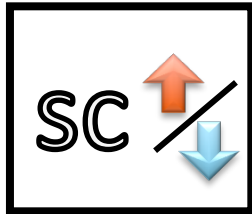
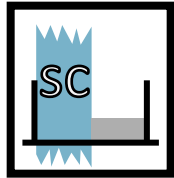
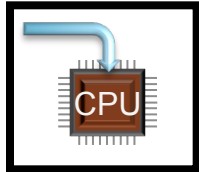
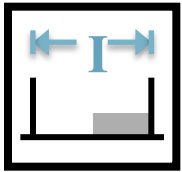
runtime scheduling

current $sc > 0$: EDF, aperiodic, ...

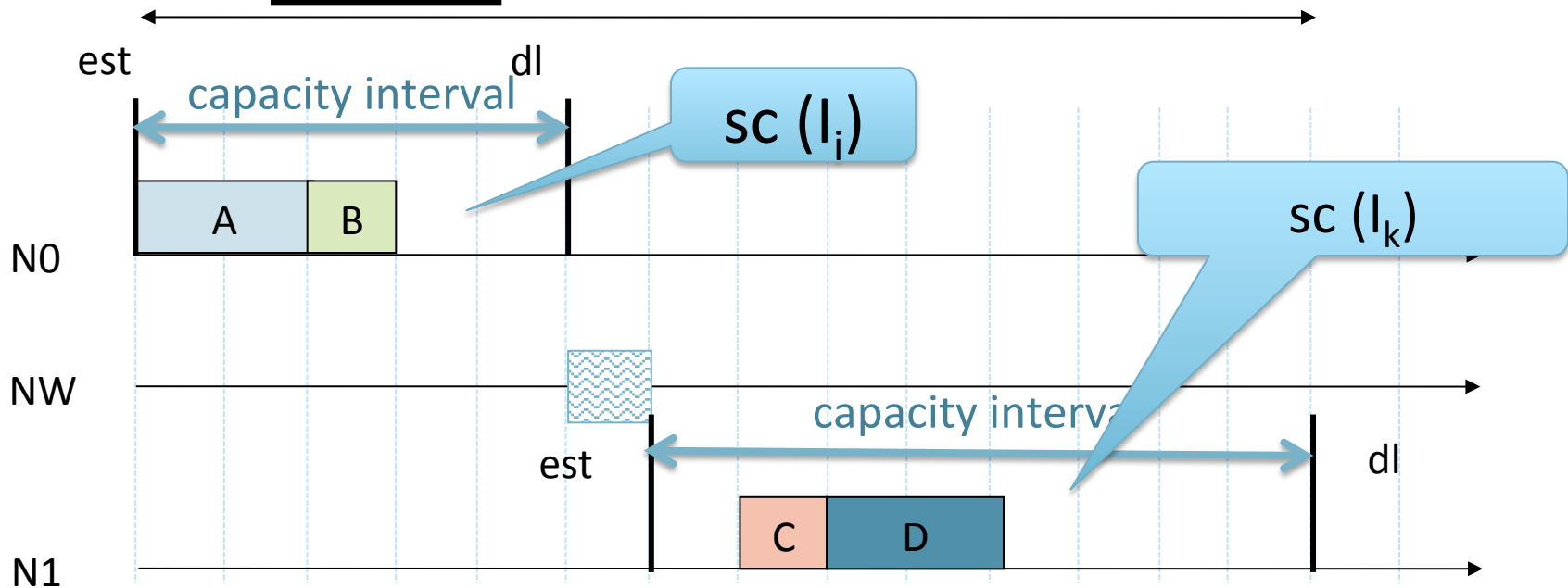
$sc = 0$: execute guar. task



Slot Shifting – Online, ctd.

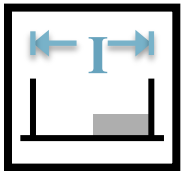


maintain spare capacities
after each slot

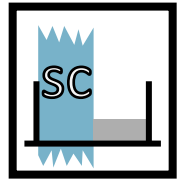


Mixed Criticality - Offline

do the high criticality tasks fit into the table?



analyze existing scheduling table

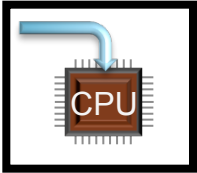


two sets of capacity intervals and spare cap.

- LO: all jobs with $C(LO)$
- HI: high-criticality jobs with $C(HI)$

if ok, can “squeeze” HI tasks into table

Mixed Criticality - Online

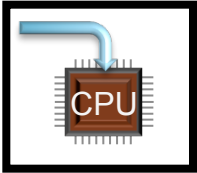


- ready queues R^{LO} , R^{HI} , R ($R^{LO} + R^{HI}$)
- EDF

switch criticality:

- if job overruns $C(LO)$
- use R^{HI} only

Mixed Criticality – Online, ctd.



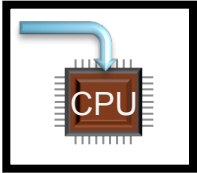
```
while ( $sc^{HI}(I_C) > 0$ )
```

```
// enough slots for HI
```

```
// do as in basic
```

- $sc^{LO}(I_C) > 0$
 - use $R(t)$, EDF
- $sc^{LO}(I_C) = 0$
 - use $R(t)$, have to select guar. task

Mixed Criticality – Online, ctd.



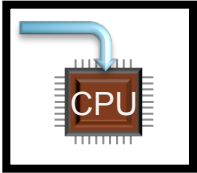
$$sc^{HI}(I_c) = 0$$

// need to start HI job

// for correct switch if overrun

- use $R^{HI}(t)$, EDF

Mixed Criticality – Online, ctd.



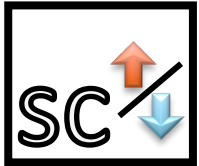
$$sc^{HI}(I_c) < 0$$

// something wrong

// will not happen with correct table

use as (offline) test for integration of HI jobs

Mixed Criticality - Online



spare capacity maintenance

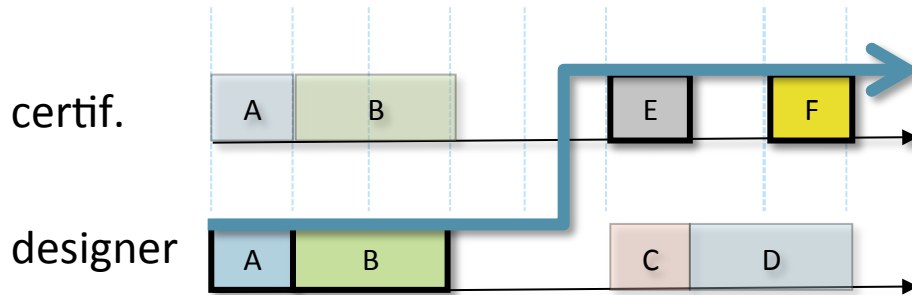
- after each slot, similar to basic
- consider sc^{LO} and sc^{HI}
- details in paper

Mixed Crit. – Legacy TT

- have existing, certified scheduling table
 - independent of offline scheduler
- can add high criticality tasks
 - without changing table
 - offline
- handle changes of criticality
 - at runtime

THE END

TT mixed crit. – mode changes



switch between tables
- mode changes

- [Baruah:RTSS11]:
proof of concept

- [Theis:WMC13]:
next talk

require

new scheduling tables
recertification