Real-Time System Modeling 2

RT Entities, Images, Objects

slide credits: H. Kopetz, P. Puschner

Real-time (RT) Entity

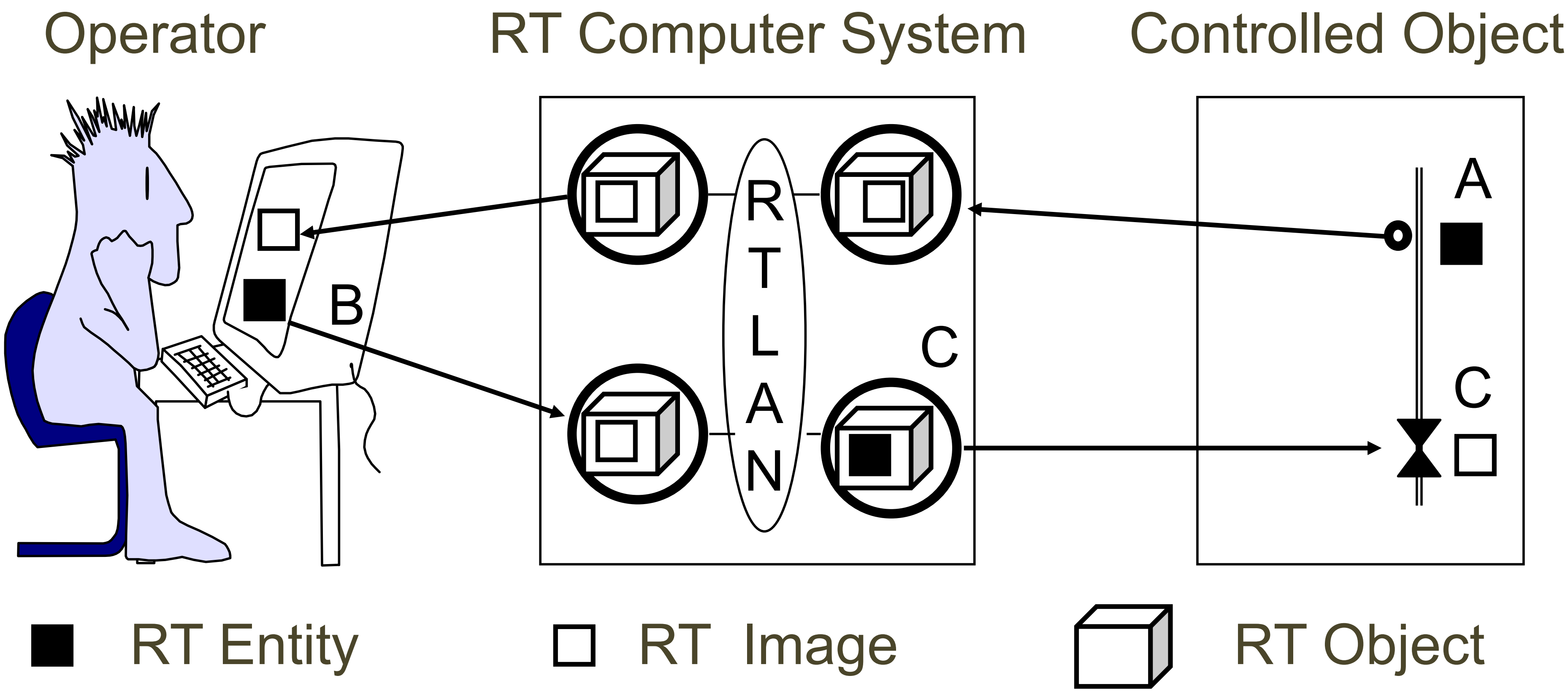
Real-Time (RT) Entity:

- state variable of relevance for a given purpose
- located either in the environment or in the computer system
- changes its state as a function of real-time
- may be *continuous* or *discrete*

Examples of RT Entities:

- flow in a pipe
- position of a switch
- setpoint selected by an operator
- intended position of an actuator

RT Entities, RT Images, RT Objects



How do we get an image?

Is the image valid at a certain time?

Attributes of RT-Entities

Static attributes

- Name (meaning)
- Type
- Value Domain
- Maximum Rate of Change

Dynamic attributes

- Actual value at a particular point in time

Sphere of Control

Every RT-Entity is in the Sphere of Control (SOC) of a subsystem that has the authority to set the value of the RT-entity:

RT-Entity	SOC
Setpoint for flow	Operator
Actual flow	Controlled object
Intended valve position	Computer

Outside its SOC an RT-entity can only be observed, but not modified.

Observation

Captures information about the state of an RT-entity

Observation = <*Name*, *Time*, *Value*>

Name: name of the RT-entity

Time: point in real-time when the observation was made

Value: value of the RT-entity

Observations are transported in messages.

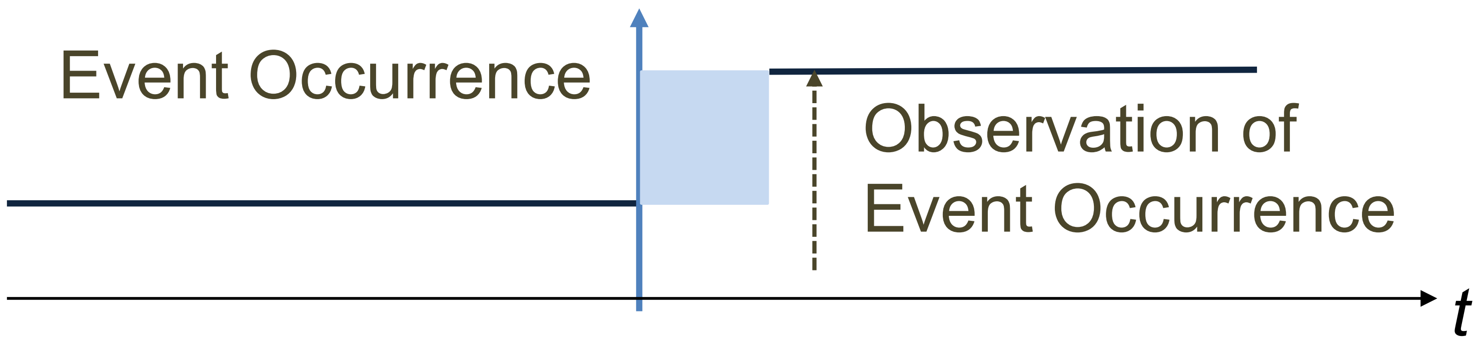
Continuous and Discrete RT-Entities

Continuous RT-Entity

- The set of values is always defined.

Discrete RT-Entity

- Discrete value set, constant during time intervals ➪ state
- Change events cannot be observed ➪ observe new state
- Between intervals, the value is undefined



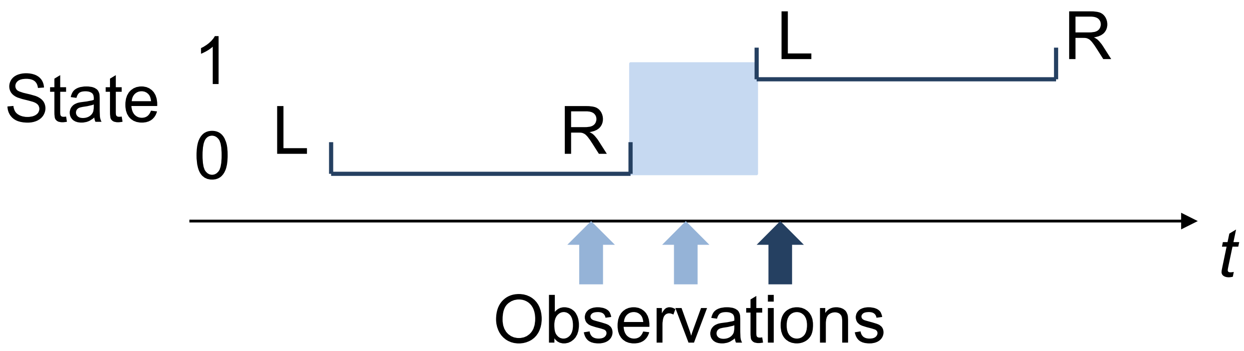
State and Event Observation

State observation

- Absolute value, contains the state of the RT-entity.
- Observation time: point in time when the RT-entity was sampled.

Event observation

- Value characterizes difference between “old” and “new” state.
- Observation time: point in time of the L-event of the “new state”.



RT Image and RT Object

RT-Image

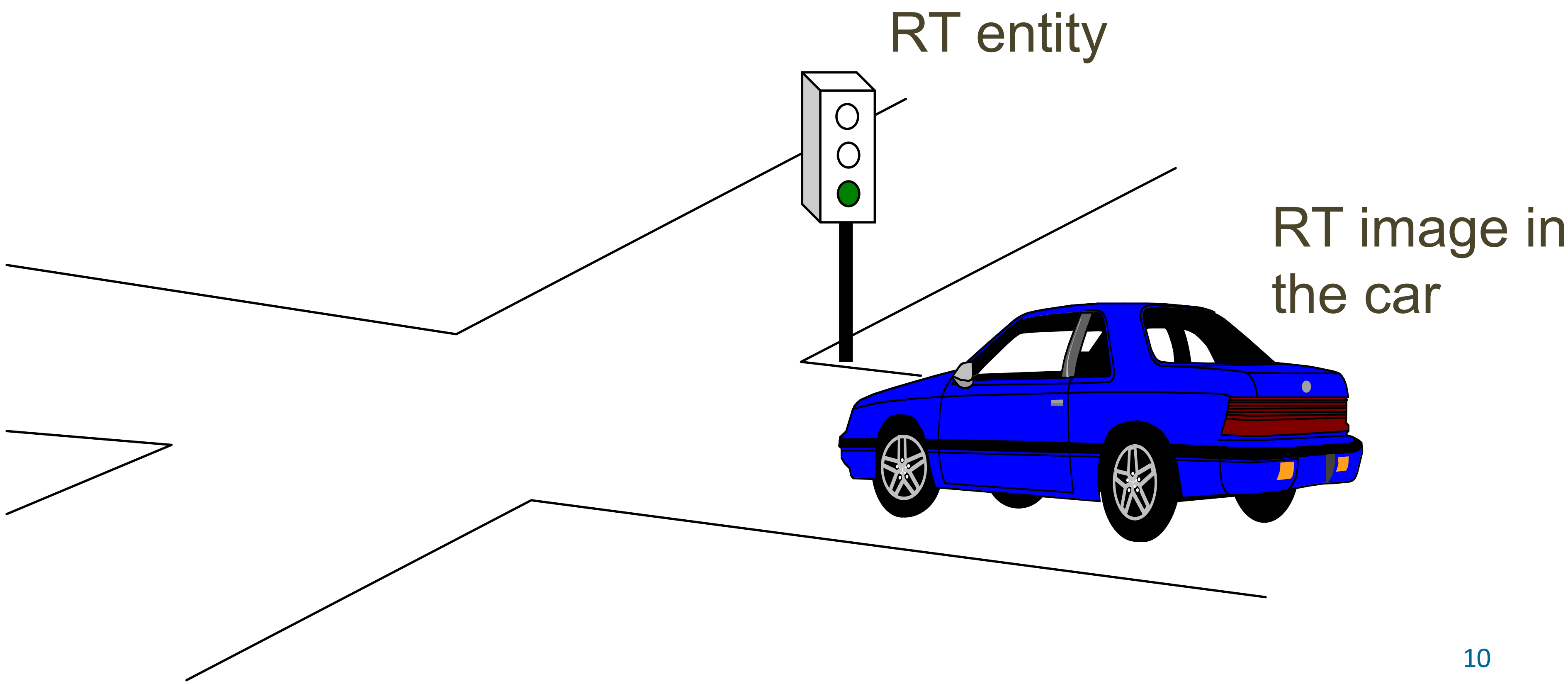
- picture of an RT-entity,
- valid at a given point in time, if it is an accurate representation of the corresponding RT-entity, in value and time,
- can be based on an observation or on state estimation,
- can be stored inside a computer or outside, in an actuator.

RT-Object

- "container" for RT-image or RT-entity in the computer system.
- has an associated real-time clock that ticks with granularity t_k (t_k must be in agreement with the dynamics of the RT-entity).

Temporal Accuracy of RT-Information

How long is the RT-image, based on the observation *"The traffic light is green"*, temporally accurate?



Temporal Accuracy

Temporal accuracy interval d_{acc}

determined by dynamics of observed RT entity

Recent history RH_i at time t_i:

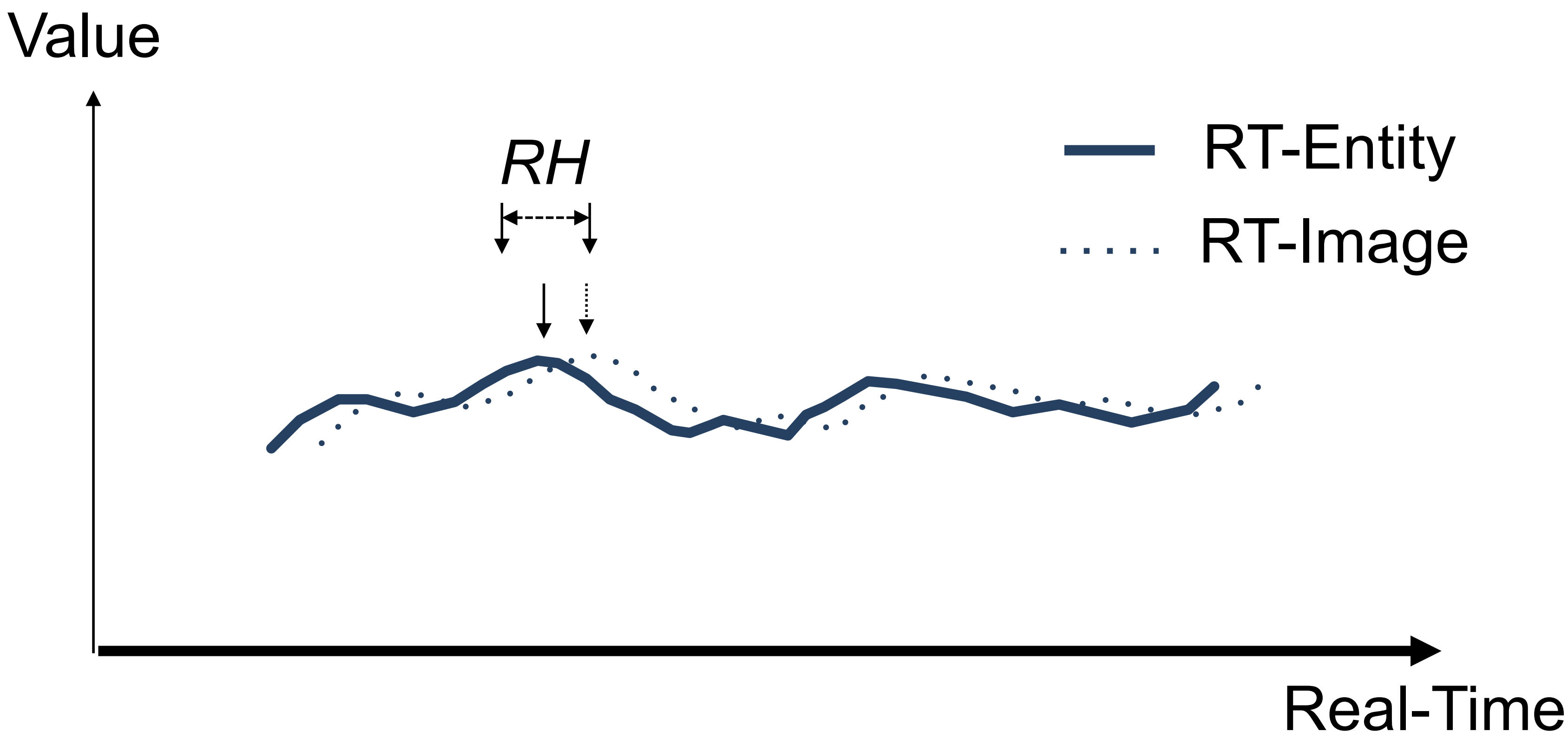
- Ordered set of time points $<t_{i-k}, \ldots, t_{i-1}, t_i>$
- Length of the recent history, $d_{acc} = t_i - t_{i-k}$

Assume that the RT-entity has been observed at every time point of the recent history.

The RT-image is temporally accurate at t_i if

$$\exists\, t_j \in RH_i : Value\,(RT\, image\, at\ t_i) = Value\,(RT\, entity\ at\ t_j)$$

Temporal Accuracy of RT-Objects



For an RT-object, updated by observations, there will always be a delay between the state of the RT-entity and that of the RT-object.

Temporal Accuracy and RT-Image Error

Delay between observation (at t_{obs}) and use (at t_{use}) of value v of an RT-entity causes an error of the RT-image:

$$error(v, t_{obs}, t_{use}) = v(t_{use}) - v(t_{obs}).$$

Approximation of worst-case error at the time of use of a temporally valid RT-image:

$$error_{max}(v) \leq \max\,(dv(t)/dt)\; d_{acc}$$

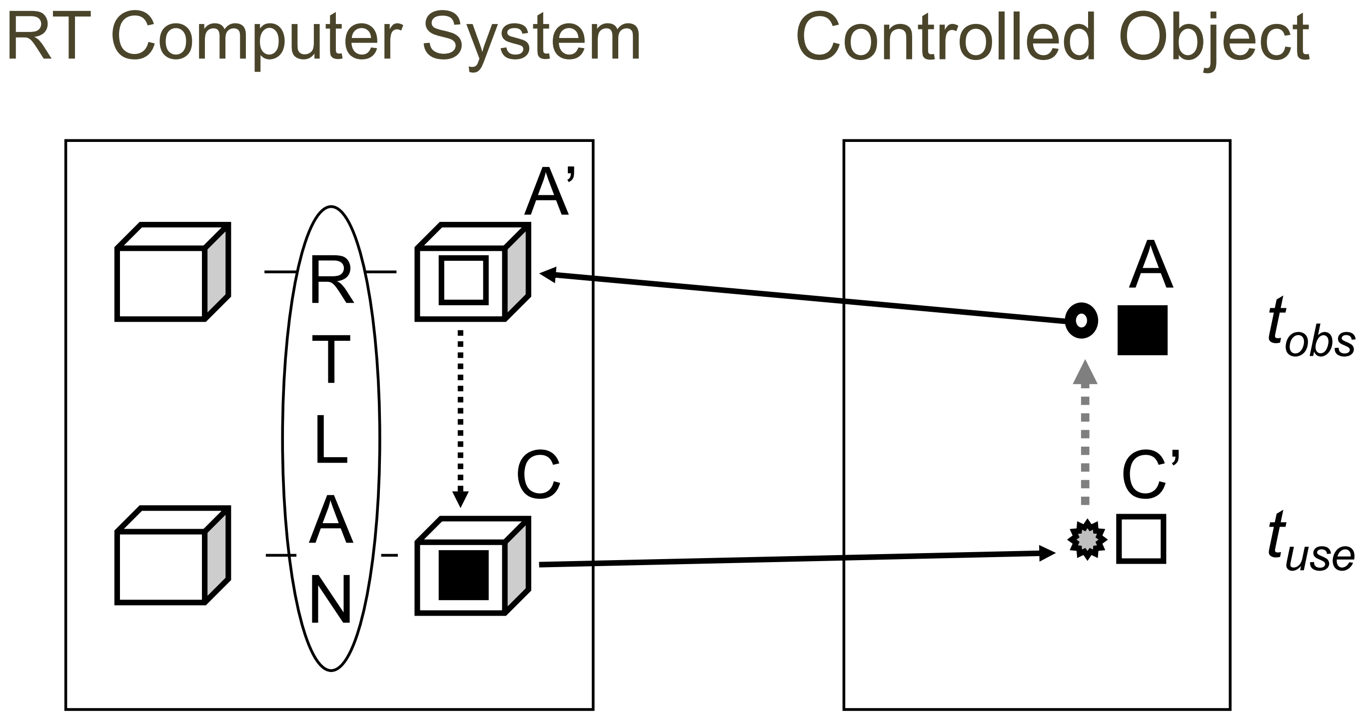
$error_{max}$ should be in the same order of magnitude as the worst-case measurement error in the value domain.

Example Temporal Accuracy Interval

The ignition time is a function of the following parameters:

RT Image	Max. change	Accuracy	d_{acc}
Piston position	6000 rpm	0.1 deg	3 μsec
Pedal position	100% / sec	1%	10 msec
Engine load	50% / sec	1%	20 msec
Oil temp.	10% / min	1%	6 sec

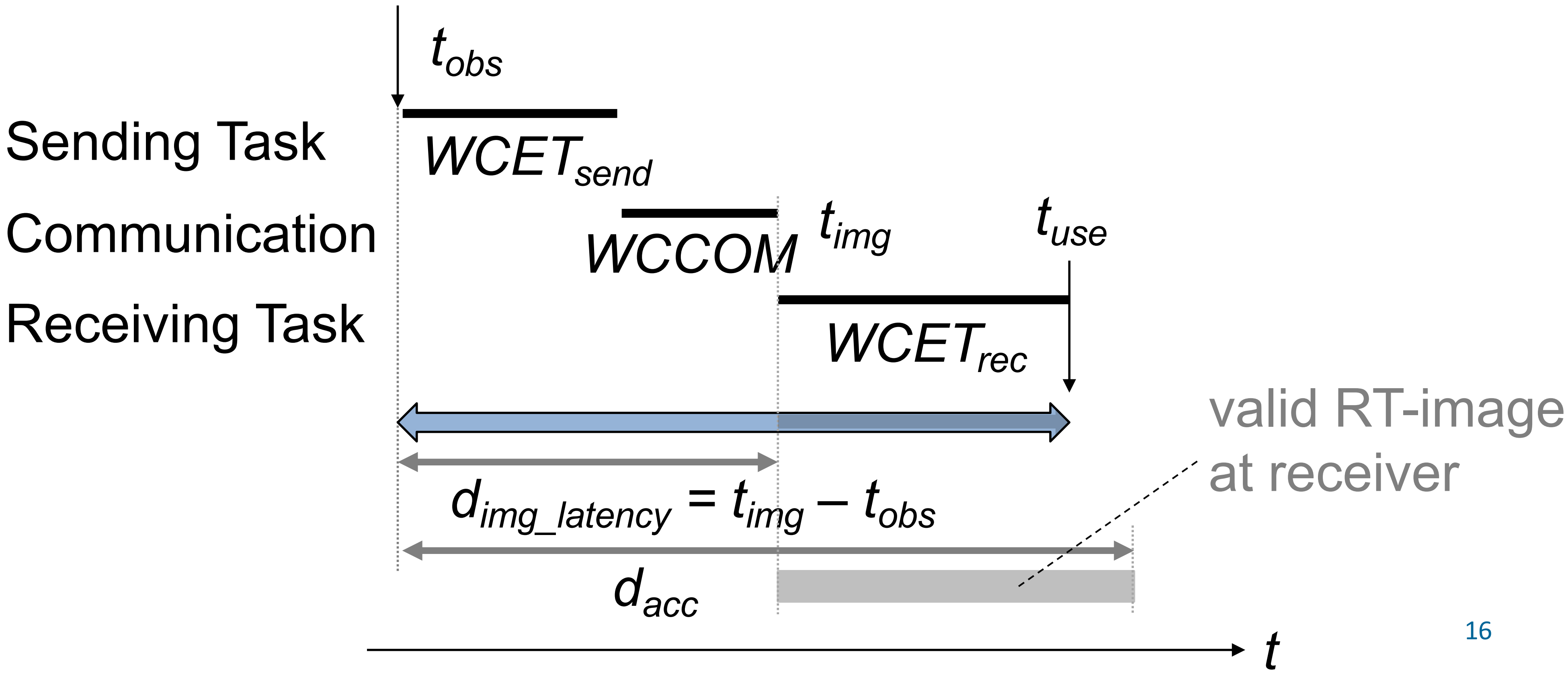
Temporal Accuracy and RT-Image Update



Synchronized Actions

If an RT entity changes its value quickly, d_{acc} must be short.

Phase-aligned transactions to guarantee: $t_{use} - t_{obs} \leq d_{acc}$



Periodic Updates and Accuracy

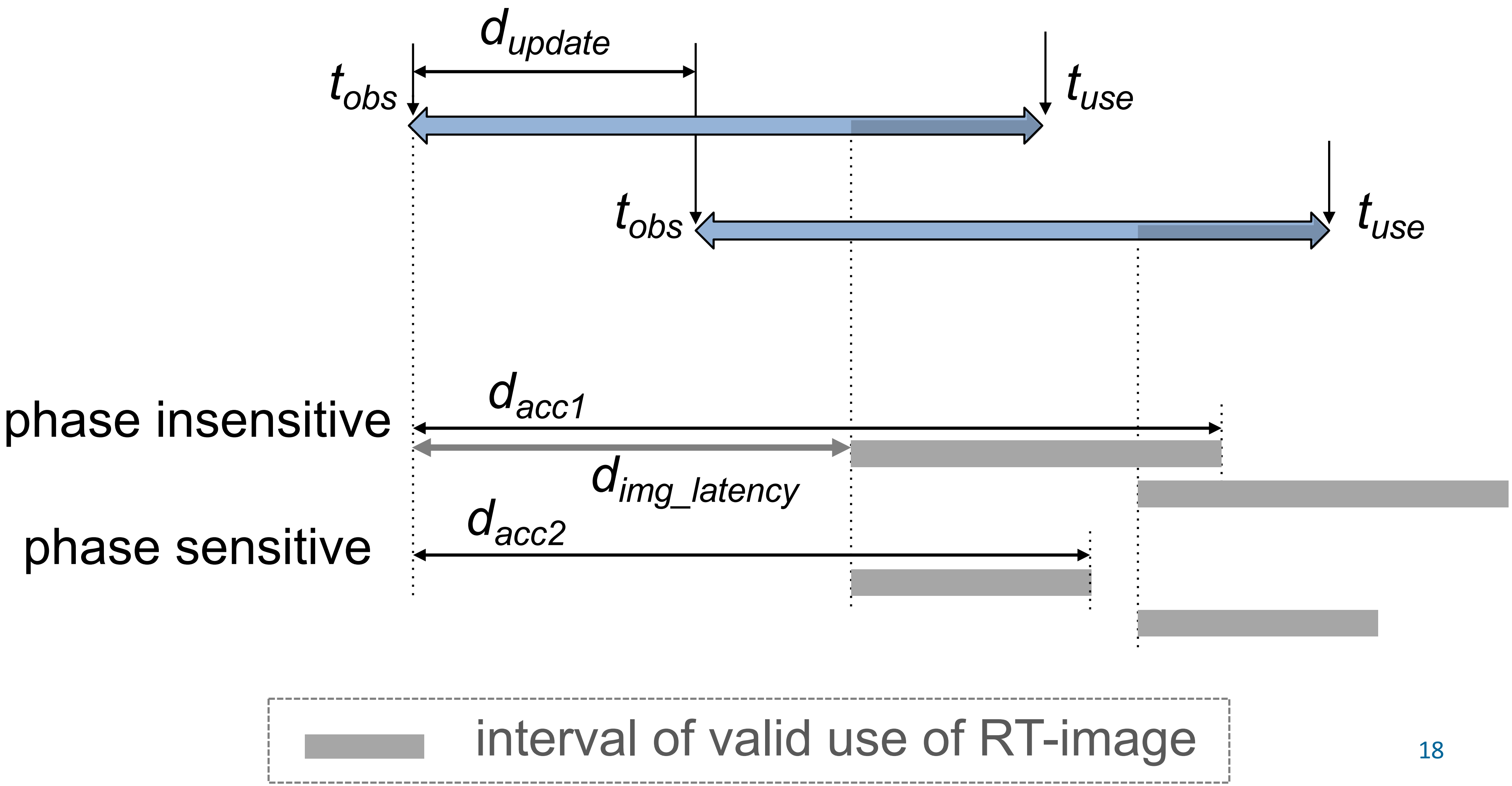
Assumptions

- Periodic update of RT-image, period d_{update}
- Short temporal accuracy
- Transaction is phase aligned

Question

- When can we use the RT-image?

Phase Sensitivity of RT-Images



Phase Sensitivity of RT-Images

Assume an RT image with internal image latency $d_{img\_latency}$ and update period d_{update}.

An RT-image is *parametric* or *phase insensitive* if:

$$d_{acc} > d_{img\_latency} + d_{update}$$

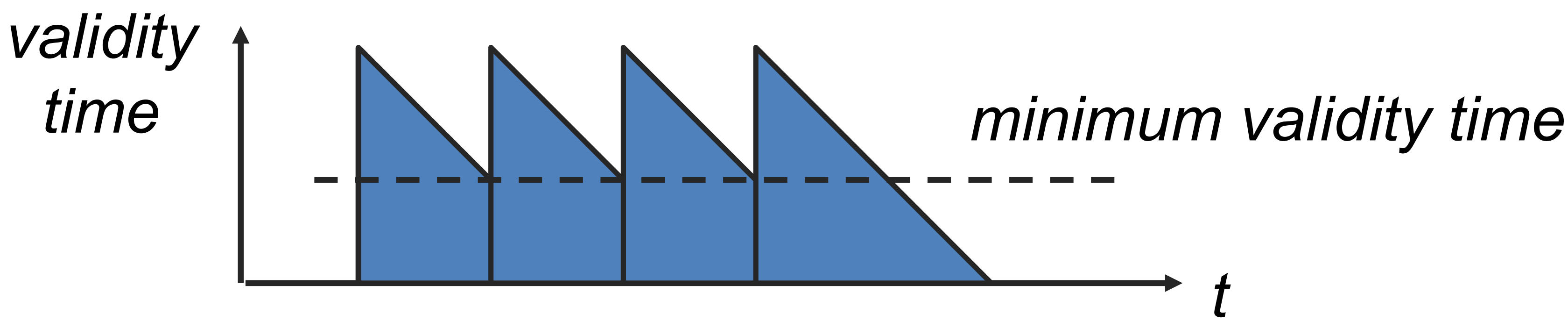
An RT-image is *phase sensitive* if:

$$d_{acc} \leq d_{img\_latency} + d_{update}$$

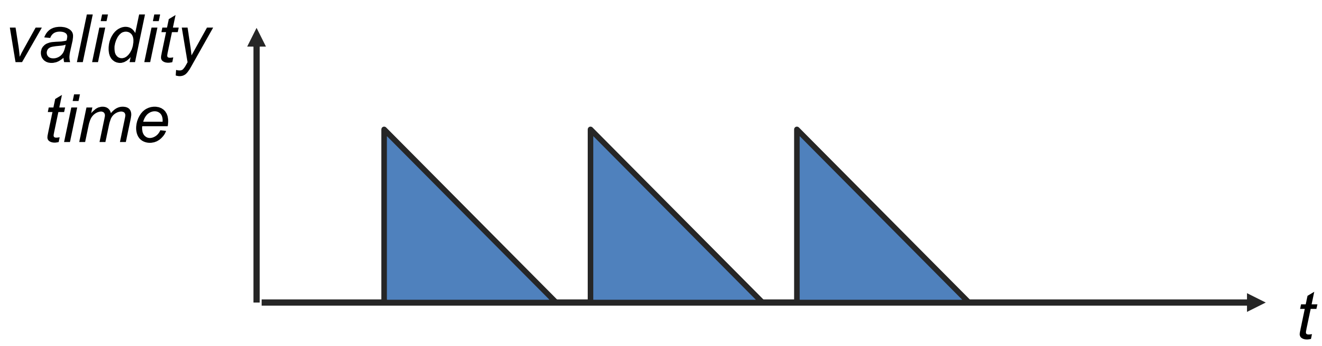
$$\text{and} \quad d_{acc} > d_{img\_latency}$$

Phase Sensitivity of RT-Images

Phase-insensitive RT-image



Phase-sensitive RT-image



State Estimation

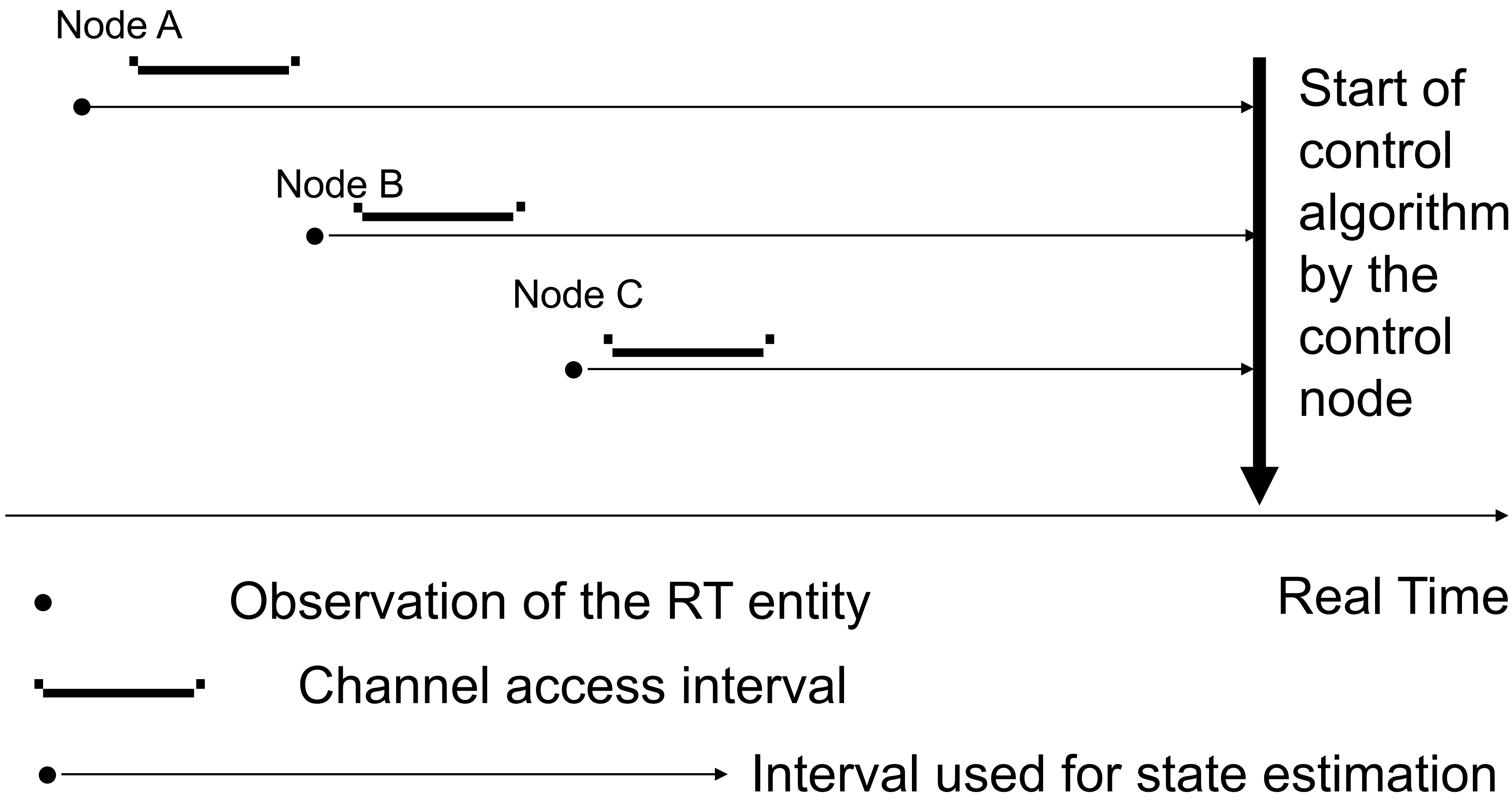
State estimation

- estimation of the current state of the RT-entity,
- periodically calculated within an RT-object,
- based on computational model of the dynamics of RT-entity.

$$v(t_{use}) \approx v(t_{obs}) + (t_{use} - t_{obs})\, dv/dt\, (t_{obs})$$

Tradeoff: computational resources/error vs. comm. resources.

State Estimation of Sensor Observations



Latency Jitter at Sender

Knowledge of sender latency (observation timestamp) improves control quality ⇔ receiver uses latency for state estimation.

Approaches

- Latency guarantee: sender guarantees latency between point of sampling and point of transmission.
- Timed messages: messages contain the observation time / interval between observation and transmission.

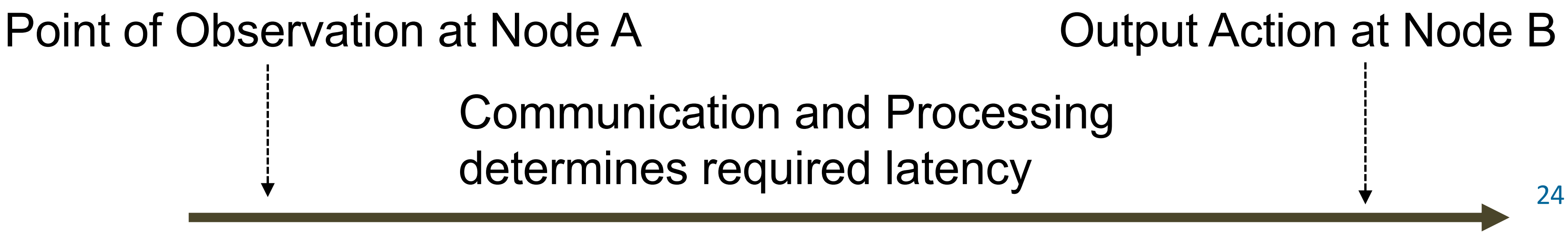
Timing Requirements for State Estimation

To compensate for the delay, a state estimation program needs

- the time of observation of an RT-entity,
- the planned time of actuation.

The quality of state estimation depends on the

- Precision of the clock synchronization,
- Latency and quality of latency measurement,
- Quality of state-estimation model.

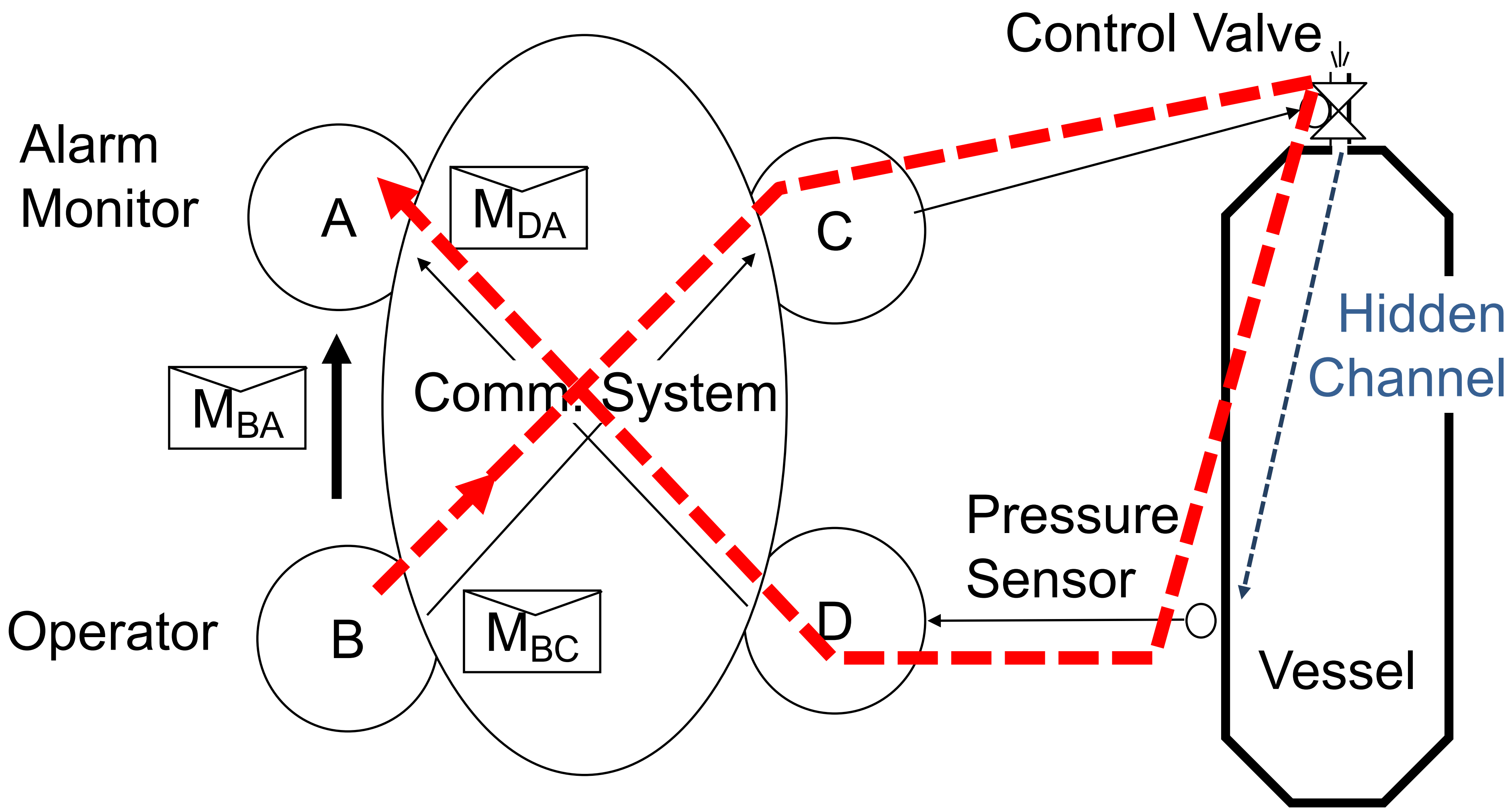


Permanence

A message M_i becomes *permanent* at object O as soon as all messages M_{i-1}, M_{i-2}, … that have been sent to O before M_i (in temporal order) have arrived at O.

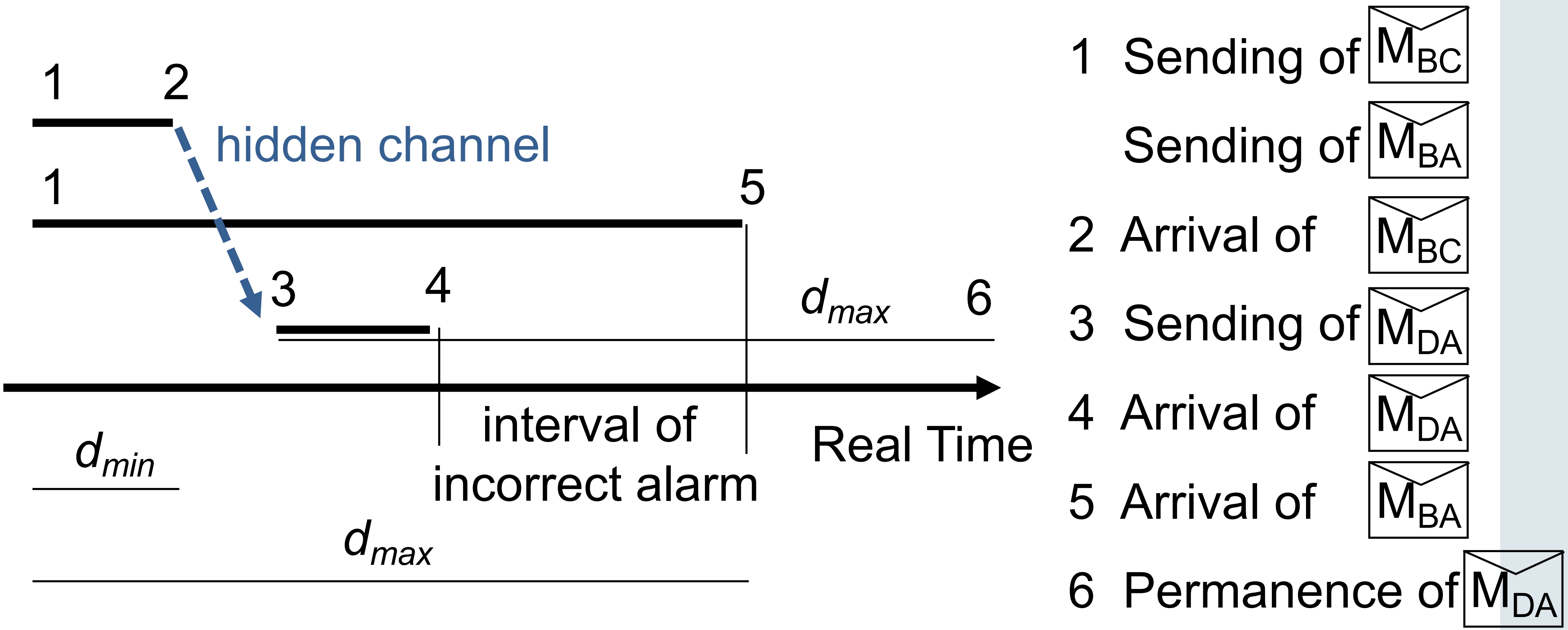
Actions taken on non-permanent messages may cause errors or inconsistencies!

Permanence



Control Valve
Alarm
Monitor
A
M_{DA}
C
M_{BA}
Comm. System
Hidden
Channel
Pressure
Sensor
Operator
B
M_{BC}
D
Vessel

Permanence


1
2
hidden channel
1
5
3
4
dmax
6
interval of
incorrect alarm
Real Time
dmin
dmax
1 Sending of MBC
Sending of MBA
2 Arrival of MBC
3 Sending of MDA
4 Arrival of MDA
5 Arrival of MBA
6 Permanence of MDA

Action Delay

Interval between the point in time when a message is sent by the sender and the point in time when the receiver knows that the message is permanent.

Distributed RT systems without global time base:

maximum action delay: $d_{max} + \varepsilon = 2d_{max} - d_{min}$

Systems with global time (timestamped messages):

action delay: $d_{max} + 2g$

Distributed real time system: maximum protocol execution time, not “median” protocol execution time determines responsiveness!

Accuracy vs. Action Delay

In a properly designed RT system

$$\text{Action Delay} < d_{acc}$$

- Accuracy (d_{acc}) is an application specific parameter.
- The action delay is an implementation-specific parameter.

What happens if this condition is violated?

⇨ Then we need state estimation!

Points to Remember

- RT-entity vs. RT-image
- RT-object
- Observation vs. state estimation
- Temporal accuracy
- Action delay