

# Chapter 5

## Real-Time Entities and Images

May-05



### Overview

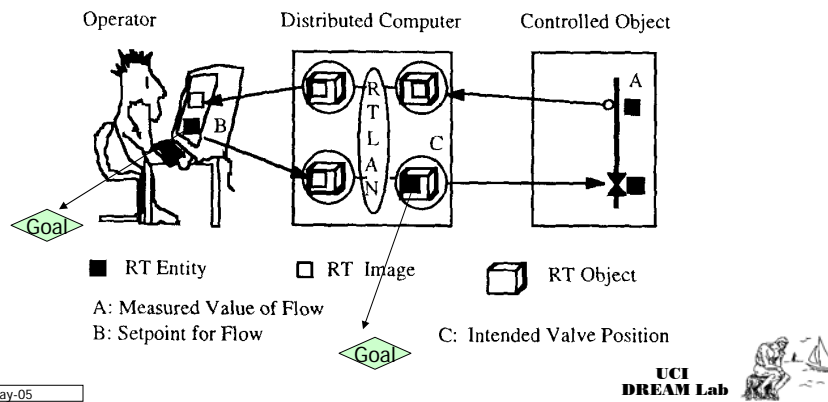
- Real-Time Entity, Image, and Object
- Temporal Accuracy
- Parametric and Phase-Sensitive Observation
- Permanent Messages and Action Delays
- Replica Consistency

May-05



## Real-Time Entities

- **State variable** in the environment
  - Flow of a liquid in a pipe
- Each has its **sphere of control (SOC)**
  - Pipe flow in SOC of the controller



May-05

## Real-Time Entities

- Discrete and Continuous RT Entities
  - Garage Door can be open or closed
  - There are intermediate states also which should not be taken seriously.

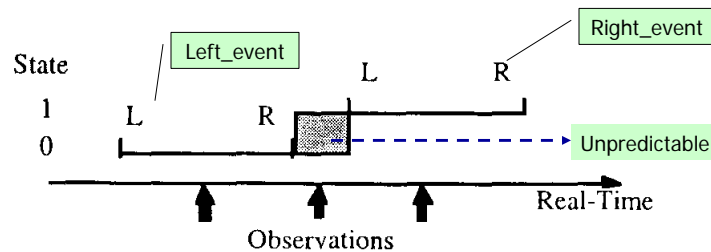
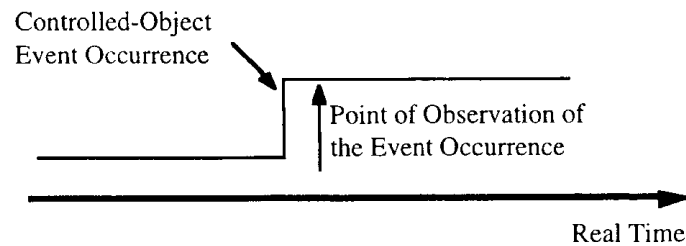


Figure 5.2: Discrete RT entity.

May-05

## Observations

- Observation:  $\langle \text{Name}, t_{\text{obs}}, \text{Value} \rangle$
- **Global timestamps** of observation acts are useful
- **State** – absolute value
- **Event** – relative change



May-05

UCI  
DREAM Lab



## Real-Time Images and Objects

- **RT Image**
  - Current picture of an RT Entity
  - Time-dependent
- **RT Object**
  - Container that holds an RT Image and has a real-time clock
  - Synchronous RT Object is updated only by periodic clock ticks
  - Distributed RT Objects – Global Time

May-05

UCI  
DREAM Lab



## Temporal Accuracy

- Relationship between a RT Entity and Image/Object
- $d_{acc}$  := A simplistic notion of the Temporal accuracy interval  
Basically the amount of time that lapses after a RT Entity is at a certain value until that value becomes no longer safe to rely on.
  - Meant to be an interval during which the state change in the RT entity can be ignored ?
- An RT image is **temporally accurate** at the present time  $t_i$  if
$$\exists t_j \in RH_i: Value (RT\ image\ at\ t_i) = Value (RT\ entity\ at\ t_j)$$
  - Again, a simplistic notion of which practical utility may be very narrow !
- A safer and more meaningful notion may be :  
Temporal accuracy := Max over  $\forall i [ z(t_i) - z(t_j) ]$

May-05



## Temporal Accuracy

- The **simplistic** notion of the Temporal accuracy interval  $d_{acc}$

$$z(t_{obs}) \leq z(t_{use}) \leq (z(t_{obs}) + d_{acc})$$

Where  $t_{use}$  denotes the point in time when the result of a computation using an RT image is applied to the env't.

$$(z(t_{use}) - z(t_{obs})) \leq d_{acc}$$

May-05



## Temporal Accuracy

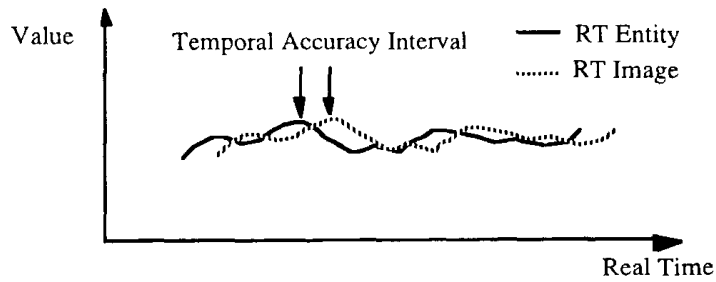


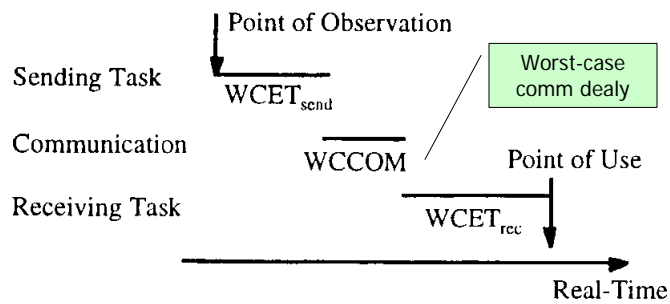
Figure 5.5: Time lag between RT entity and RT image.

May-05

UCI  
DREAM Lab 

## Temporal Accuracy

- Phase-Aligned Transaction
  - contains tightly integrated steps (sending, communication, and receiving)



$$(t_{use} - t_{obs}) = WCET_{send} + WCCOM + WCET_{rec}$$

May-05

UCI  
DREAM Lab 

## Simplistic Temporal Accuracy Intervals

RT Image within Computer	Max.Change	Accuracy	$d_{acc}$
Position of piston within cylinder	6000 rpm	0.1°	3 $\mu$ sec
Position of accelerator pedal	100%/sec	1 %	10 msec
Engine load	50 % / sec	1 %	20 msec
Temperature of the oil and the coolant	10 % /minute	1 %	6 seconds

**Table 5.1:** Temporal accuracy intervals in engine control.

May-05

UCI  
DREAM Lab

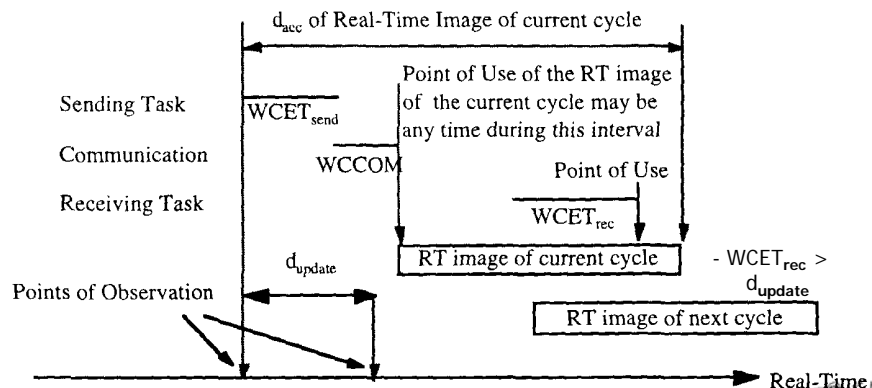


## Temporal Accuracy

- **Parametric or Phase-insensitive RT image:**
  - Does not depend on when the observation is used by the receiver within the cycle

State change in the RT entity during the long  $d_{acc}$  is insignificant.

$$d_{acc} > (d_{update} + WCET_{send} + WCCOM + WCET_{rec}),$$



May-05

UCI  
DREAM Lab



## Temporal Accuracy

- Phase-sensitive RT Image – does depend on the time of using the image.

$$d_{acc} \leq (d_{update} + WCET_{send} + WCCOM + WCET_{rec}) \text{ and}$$

$$d_{acc} > (WCET_{send} + WCCOM + WCET_{rec})$$

- \* Every phase-sensitive RT image imposes an additional constraint on the scheduling of the real-time task that uses this RT image.

May-05



## Temporal Accuracy – State Estimation

- Extends temporal accuracy by prediction, e.g., extrapolation.  
E.G., The crankshaft in an engine rotates with a rotational speed of 3000 RPM, i.e., 18 degrees per millisecond.  
If  $t_{use} - t_{obs} = 500$  microsec  $\rightarrow$   
Update the RT image by 9 degrees to arrive at an estimate of the position.
- 1<sup>st</sup> order estimate of the state of the RT entity at  $t_{use}$

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

May-05



## Temporal Accuracy

- The sender may perform a state estimation for the interval  $[t_{\text{obs}}, t_{\text{arr}}]$  and the receiver may perform a state estimation of the interval  $[t_{\text{arr}}, t_{\text{use}}]$ .

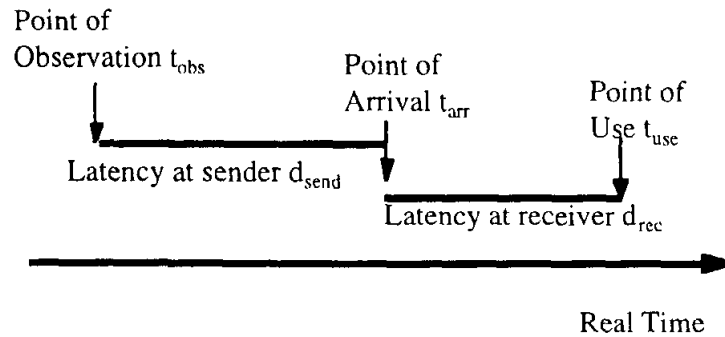


Figure 5.8: Latency at sender and receiver.

May-05

UCI  
DREAM Lab



## Permanent Message: Msg Ready for In-Order Processing

- A message at a node is permanent if all other messages sent before it would have already reached that node.

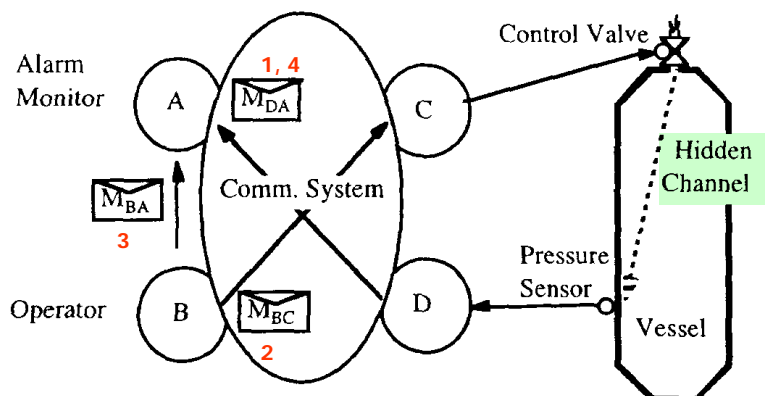


Figure 5.9: Hidden channel in the controlled object.

May-05

UCI  
DREAM Lab



## Permanent Message: Msg Ready for In-Order Processing

- $M_{DA}$  arrives at Alarm monitor before  $M_{BA}$  does.

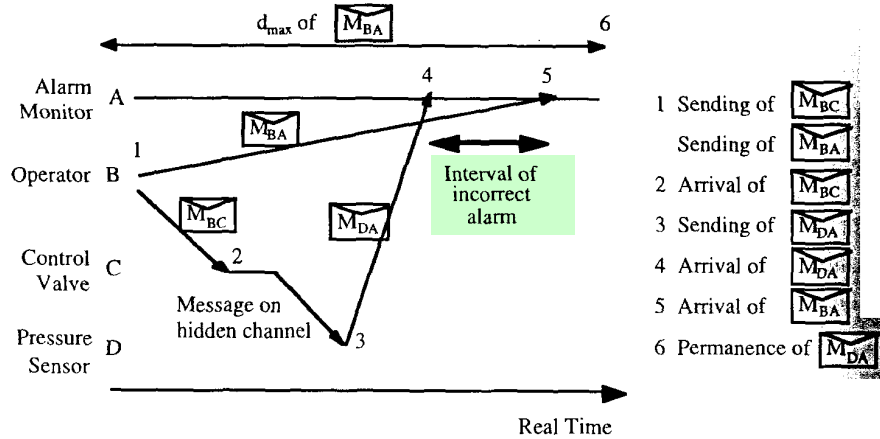


Figure 5.10: Permanence of messages.

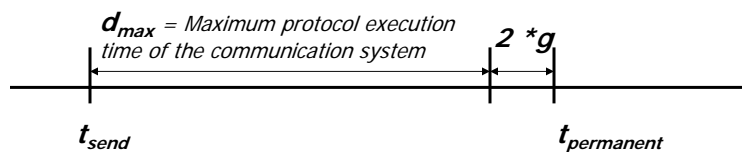


May-05

## Action Delay

- Action Delay := The time interval between the start of transmission of a given message and the point in time when this message becomes permanent at the receiver.
- The duration of the action delay depends on
  - the jitter of the communications system and
  - the temporal awareness of the receiver.
- Systems with a global time base

$$t_{\text{permanent}} = t_{\text{send}} + d_{\text{max}} + 2g$$



Here  $g$  is the granularity of the global time.



May-05

## Action Delay (cont.)

- Systems without a global time base

$$t_{\text{permanent}} = t_{\text{send}} + 2d_{\text{max}} - d_{\text{min}} + g_l$$

Here  $g_l$  is the granularity of the local time-base

- Proof

– Because the receiver doesn't know when the message was sent. =>

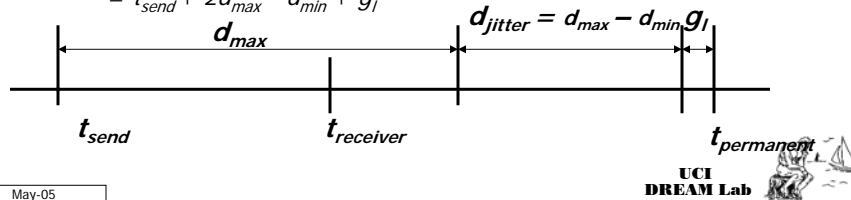
to be safe,  $t_{\text{permanent}} = t_{\text{receive}} + d_{\text{jitter}} + g_l$ ,

where  $d_{\text{jitter}}$  is the jitter for the protocol execution time,

$$d_{\text{jitter}} = d_{\text{max}} - d_{\text{min}},$$

while  $t_{\text{receive}}$  can be in reality as bad as  $t_{\text{send}} + d_{\text{max}}$ .

$$\begin{aligned} \Rightarrow t_{\text{permanent}} &= t_{\text{send}} + d_{\text{max}} + d_{\text{max}} - d_{\text{min}} + g_l \\ &= t_{\text{send}} + 2d_{\text{max}} - d_{\text{min}} + g_l \end{aligned}$$



May-05

## Idempotency

- A set of replicated messages is **idempotent** if the effect of receiving more than one copy of a message is the same as receiving only a single copy.
- Idempotent messages can arrive as multiple copies and still be valid.

E.G., "Position of valve at 45 degree".

May-05

UCI  
DREAM Lab

## Replica Consistency

- A set of **replicated RT objects** is **replica-consistent** if all the members of this set have the same externally visible h-state, and produce the same output messages at points in time that are at most an interval of **d** time units apart.
- A set of nodes is **replica-consistent** if all the nodes in this set contain the same externally visible h-state at their ground state, and produce the same output messages at points in time that are at most an interval of **d** time units apart.
- In a time-triggered system, an upper bound for "d" is given by the precision of the global time.
- So, replica consistency is maintained when all replicas have the **same output history** and the **same history of externally visible states**.
- Good for active redundancy and for testing

May-05

UCI  
DREAM Lab



## Replica Consistency

- Replica consistency is needed to
  - Implement fault-tolerance by active redundancy
  - Facilitate the system test
- A **major decision** point is a decision point in an algorithm that provides a choice between a set of significantly different courses of actions.

Channel	Decision	Action
Channel 1	Take off	Accelerate Engine
Channel 2	Abort	Stop Engine
Channel 3	Abort	Accelerate Engine (Fault)
Majority	Abort	Accelerate Engine

*The faulty channel wins!*

- \* Faulty channel (3) plays a decisive role since correct channels (1 and 2) are not replica-consistent !

May-05

UCI  
DREAM Lab



## Basic Causes of Replica Inconsistency

- Differing Inputs
- Deviation of Computational Progress Rate relative to Physical Time
- Oscillator Drift
- Preemptive Scheduling
- Non-deterministic Language Features
- Race Conditions
- Consistent Comparison Problem

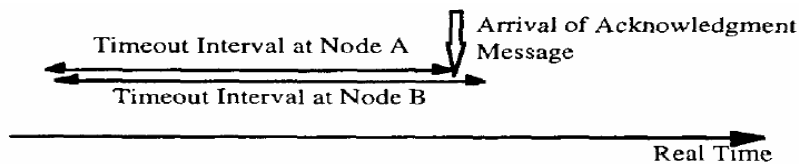
May-05

UCI  
DREAM Lab



## Building a Replica-Consistent System

- Cautious approaches for building a replica-consistent system
  - Sparse Time-base
    - Any reference to the local real-time clock can lead to replica inconsistency.



The effect of local time-outs at FTU

May-05

UCI  
DREAM Lab



## Building a Replica-Consistent System

---

- Agreement on Input

Whenever a redundant observation of an RT entity is performed, an agreement on the observed value and the observation time must be reached.

- Static Control Structure
- Deterministic Algorithms

May-05



## Leader-Follower Protocol: An alternative

---

- The leader takes all the major decisions.
- It forces the followers to be behind the leaders to take the same decisions.
- The protocol requires a fair amount of **inter-replica coordination**.
- It has a **window of vulnerability** between the decision time-point of the leader and the learning time-point of the followers.
  
- In general, **good to avoid inter-replica coordination**.
  - It compromises the independency of the replicas.
  - It requires additional time
  - It requires additional comm bandwidth.

May-05

