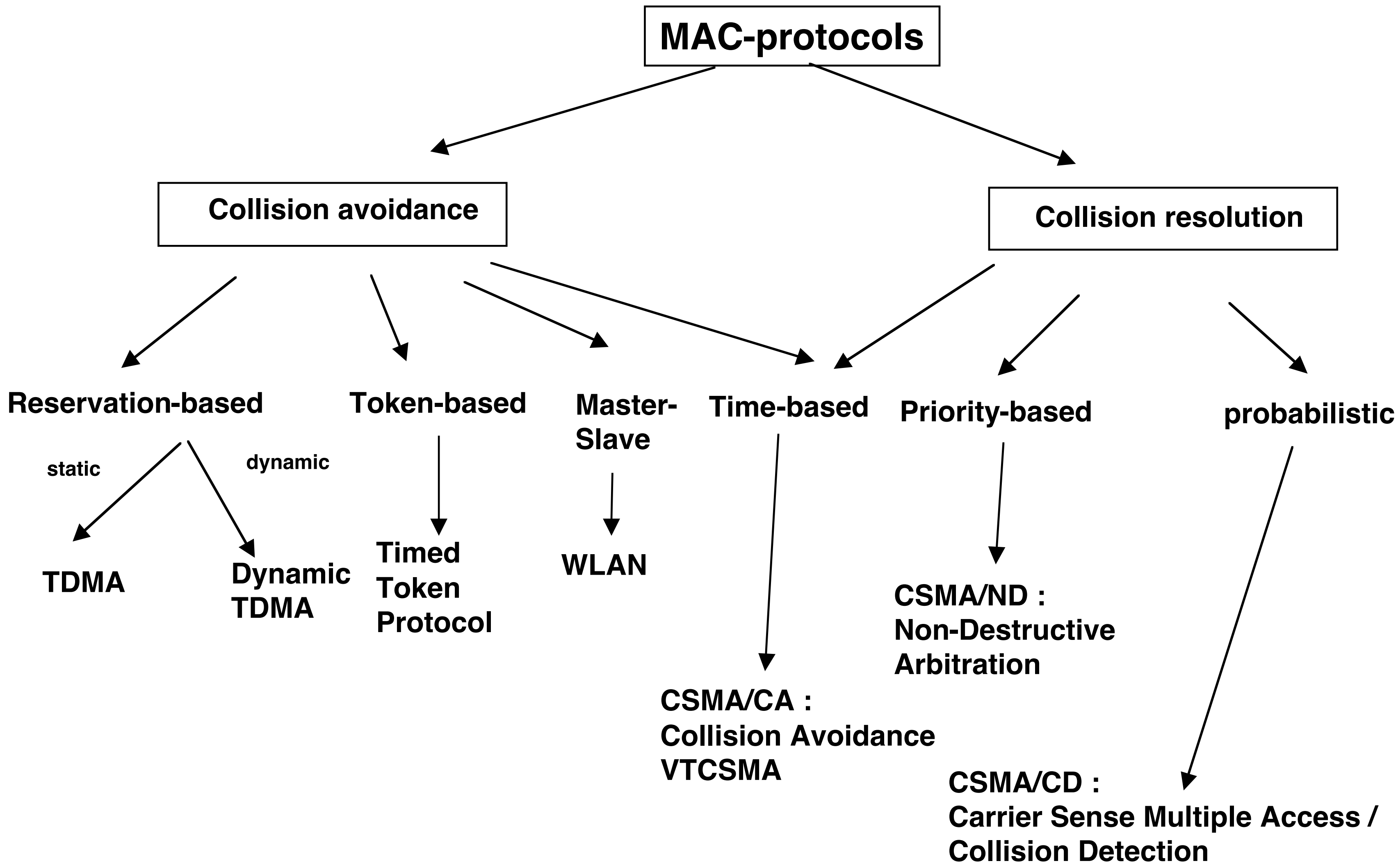
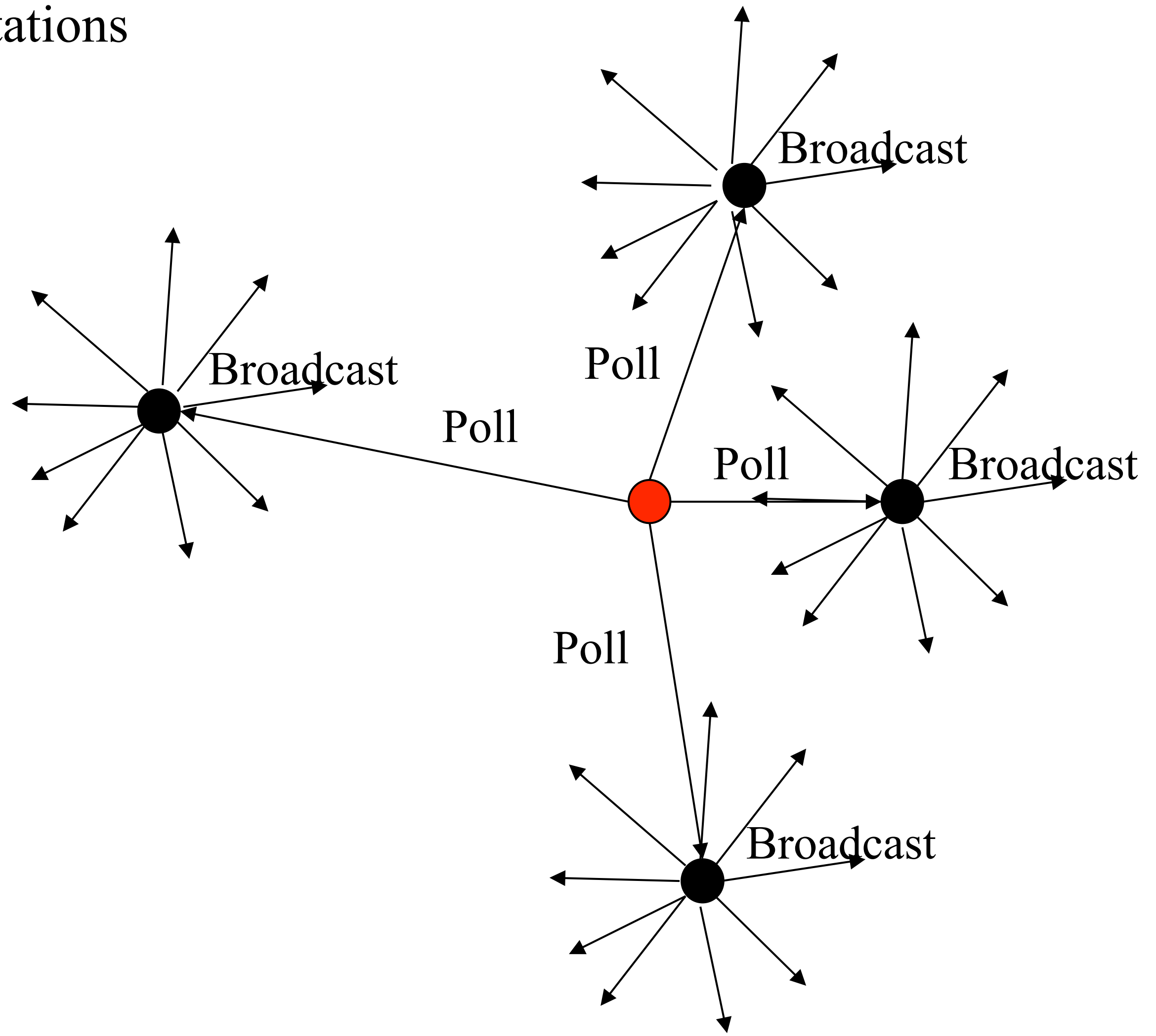
Real-Time (Paradigms) (70)

Taxonomy of Medium Access Control - Protocols:



IEEE 802.11 contention free access

A central access point grants access to the medium by polling the stations



Remaining problems

- Messages can be lost, even worse
- Some stations may receive a message, some others may not
- Stations can crash
- Stations can be out of reach
- No timing guarantees are given

➔ Must make specific fault assumptions for giving any kind of guarantees

Fault Assumptions

- Messages are either lost or delivered within a fixed time bound
- Message losses are bounded by an Omission Degree OD
- Stations may fail (silently)
- Stations may leave/enter the reach of other stations
- The access point can be considered to be stable

Reliable real-time communication can be achieved by using redundancy to tolerate faults

Static vs. Dynamic Redundancy

Static redundancy - Message diffusion

principle: every message is transmitted OD+1 times

good: simple, no need to detect message losses

bad: large overhead

Dynamic redundancy - Acknowledge/retransmit

principle: every message is only retransmitted if a message loss occurs (maximum OD retransmissions)

good: small overhead for retransmissions

bad: acknowledgements for detecting message loss induce extra overhead

Acknowledgment scheme is crucial

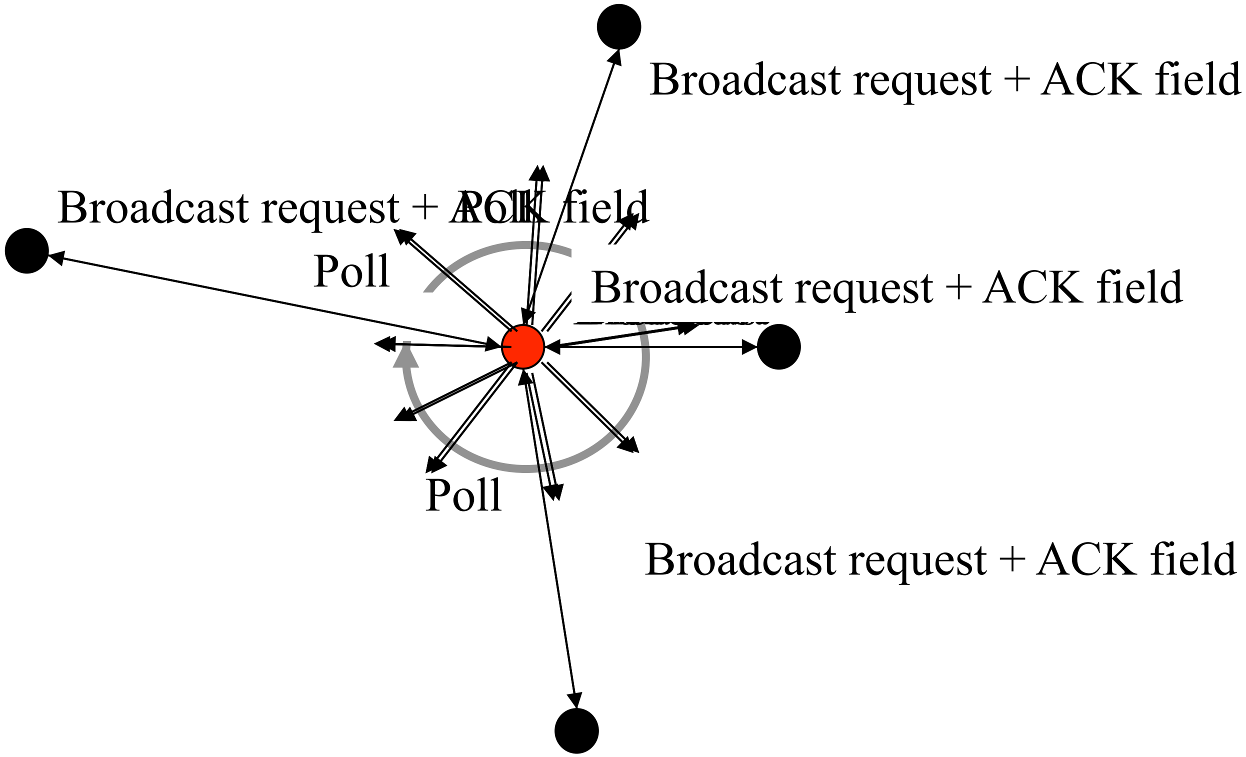
Key ideas of the RGC protocol

- Broadcast messages are routed through the access point
  - Membership problem due to limited reach and mobility solved
  - ordering problem solved

- Efficient acknowledgement scheme
  - communication is organized in rounds of length n
  - one ACK field (n bits) acknowledges all messages of the preceding round
  - ACK field is piggy-backed to the broadcast request message
  - if necessary, the access points retransmits the message of the preceding round (at most OD retransmissions).

no extra acknowledgment messages needed !

Operation of the protocol



Timing Analysis

- Polling/broadcast request messages can be lost
- Broadcast messages can be lost
- At most omission degree OD retransmissions required,
  (OD is dependent on the physical characteristics of the application environment)

➔ worst case delivery time can be computed

$(\Delta bc_{max} \approx 2 \times OD \times \Delta round)$

$(\Delta round := n \times 3\ t_m)$

Example 1: OD = 10, n = 4 stations, t_m = delay for a single message = 2,8 ms

---> worst case delivery time ≈ 680 ms

Example 2: OD = 15

---> worst case delivery time = 1016 ms

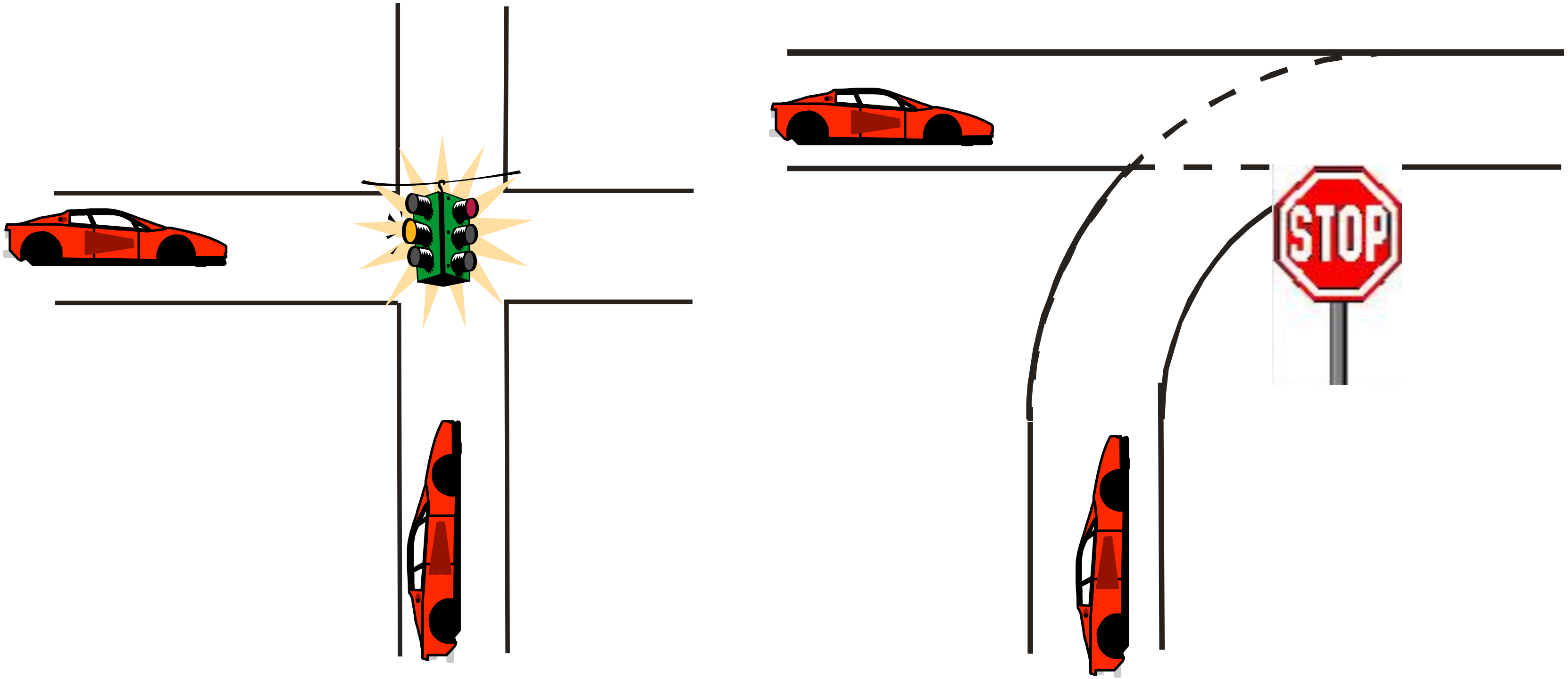
Trading Timing Guarantees against Reliability

- Problem: How to achieve better timing guarantees ?
- Observation: applications may afford to loose a (late) messages, if it is guaranteed that all stations reject the message in this case.
- Approach: Allow the application to limit the number of retransmission and guarantee agreement on consistent delivery

User defined resiliency degree

- Limit the number of retransmission by an application defined resiliency degree res(c) (maximum OD)
- If a message is not acknowledged by all stations after res(c) retransmissions, it is rejected.
- The access point puts its decision whether to reject/accept a message in an accept field that is piggy-backed with every broadcast message.

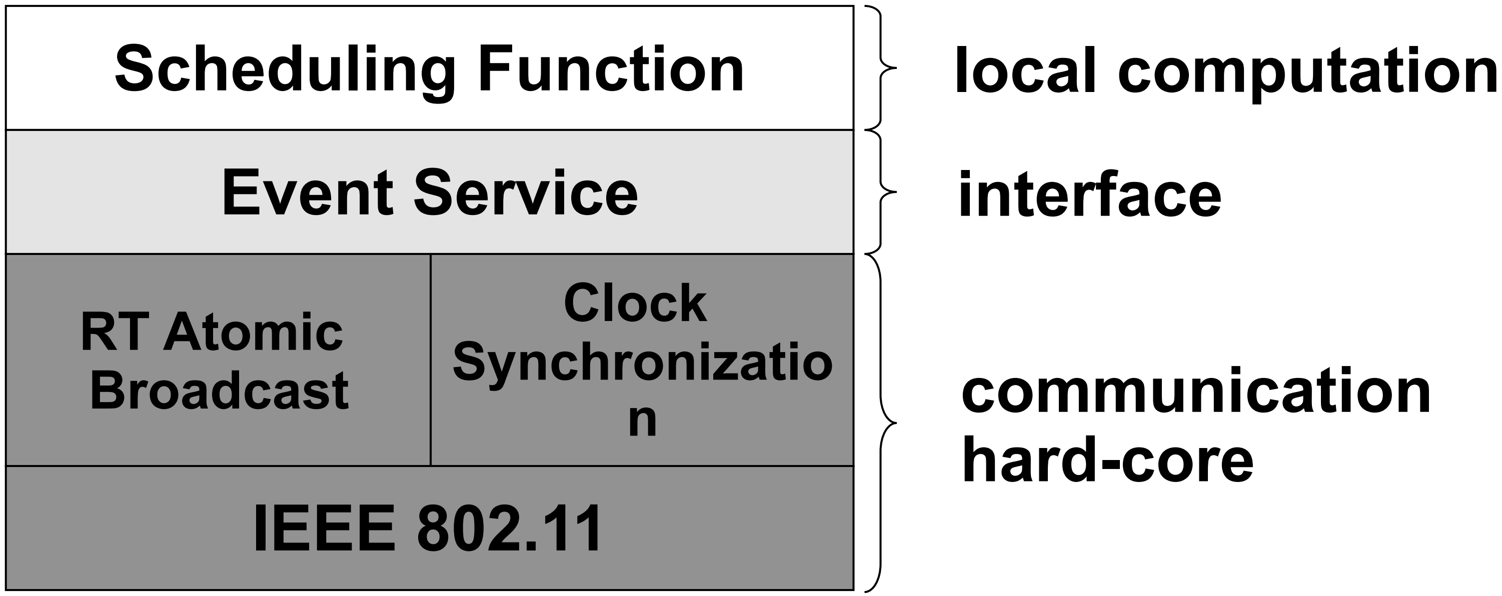
Problem Scenario



- vehicles are forced to stop, even if resource is free
- low throughput

⇒ apply resource scheduling instead

Architecture

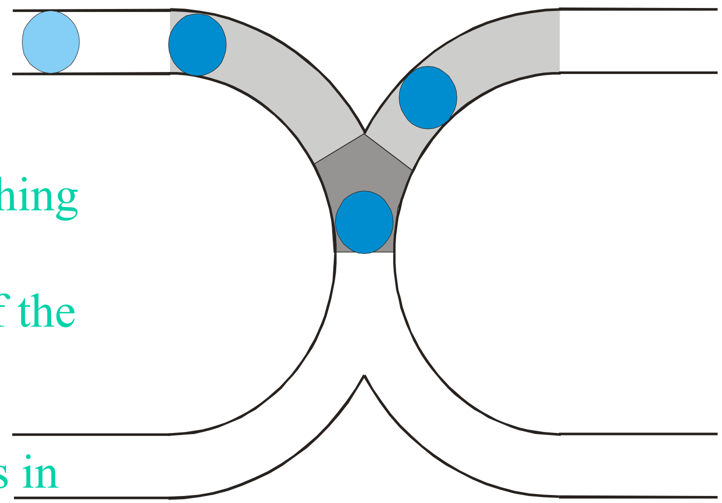


Scheduling Function - Model

System local informations

The *local state* $s_i.z(t)$ of a system s_i comprises all its scheduling relevant parameters

The scheduled enter time $s_i.t_{se}(t)$



Global informations

- The *group* $\mathbf{g}(t)$ is the vector of all systems that are within the approaching zone plus the one in the hot spot
- The *global state* $\mathbf{z}(t)$ is the vector of the local states of all systems in $\mathbf{g}(t)$
- The *plan* $\mathbf{p}(t)$ is the vector of the scheduled enter times of all systems in $\mathbf{g}(t)$

Domain

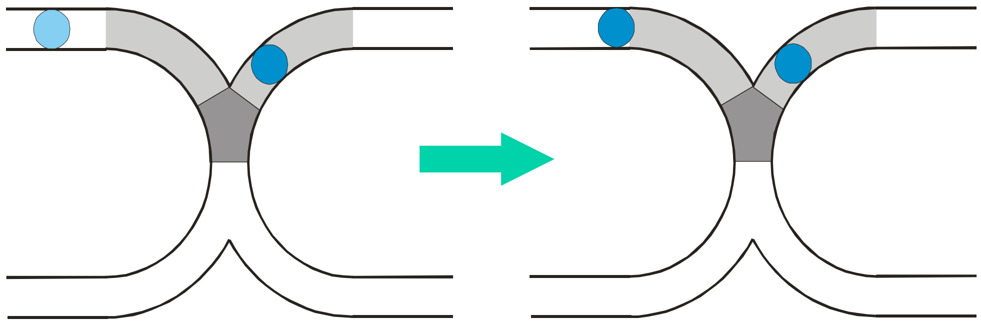
- The domain of the scheduling function is the set of all global states

Range

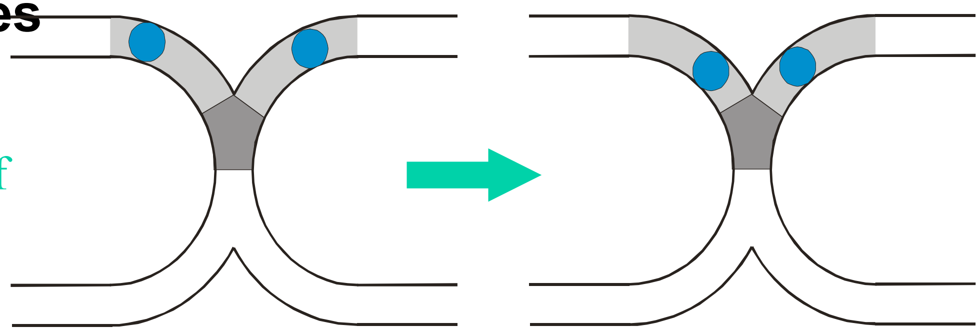
- The range of the scheduling function is the set of all plans

Event Service - Model

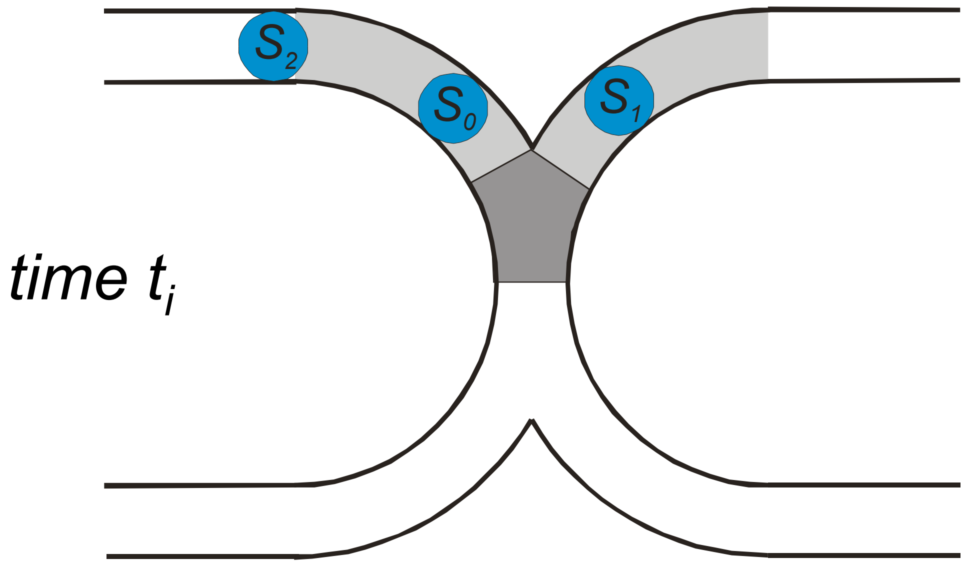
- **Discrete state changes**
  - unpredictable
  - modeled as events $e_i \in E := \{e_1, e_2, \ldots\}$
  - perceived totally ordered (at times t_1, t_2, ...)



- **Continuous state changes**
  - predictable
  - modeled as a function (F) of time and velocities

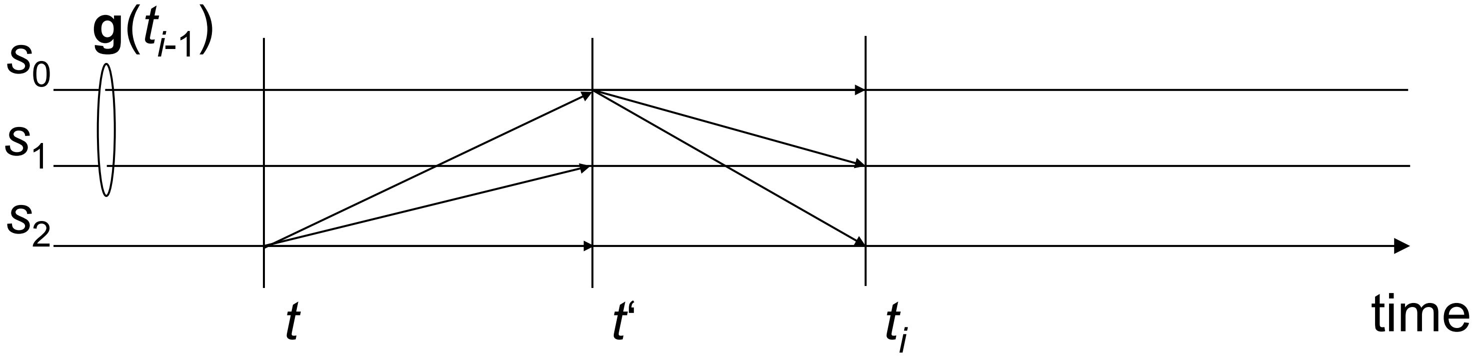


Global State Computation



- s_0 can compute $s_0.z(t_i)$ and $s_1.z(t_i)$ using F and $z(t_{i-1})$
- s_1 can compute $s_0.z(t_i)$ and $s_1.z(t_i)$ using F and $z(t_{i-1})$
- s_0 and s_1 must learn $s_2.z(t_i)$
- s_2 must learn $z(t_{i-1})$

Necessary Communication for Discrete Changes



- t: s_2 enters the approaching zone,
  broadcasts request message rqu($s_2.z(t)$, t)
- t': request is delivered,
  s_0 broadcasts in-message in($s_2, s_2.z(t), t, \mathbf{z}(t_{i-1}), t_{i-1}$)
- t_i: in-message is delivered,
  $\mathbf{z}(t_i)$ is computed and delivered to the scheduling function