

Time-free Collision Prevention for a Group of Mobile Robots:

The Additional Pinch of Pragmatism

Xavier DÉFAGO

¹⁾ *School of Information Science,
Japan Adv. Inst. of Science & Tech. (JAIST)*

²⁾ *PRESTO, Japan Science & Tech. Agency (JST)*

with:

Rami Yared, Julien Cartigny, Matthias Wiesmann

Context & Motivation

- **Context**

- Group of mobile robots

- **Objective**

- Prevent robot collisions

- **Guidelines**

- Decentralized
- Asynchronous communication
- Asynchronous positioning system
- Isolate synchronous & RT assumptions

Context

- **Equipment**

- 4 Pioneer-3 robots
 - Laptop
 - Wireless (WiFi; bluetooth)
 - Sonar (180°, 6-7m)



- **Observation**

- Ranges:
 - Sonar: 7m
 - Bluetooth: 10m
 - WiFi: 100m

System Model

- **Robots**
 - Have footprint
 - No vision
- **Positioning System**
 - Robots get **own position**
 - Robot can query asynchronously
- **Communication**
 - Asynchronous (mostly)
 - Two models: full, ad hoc

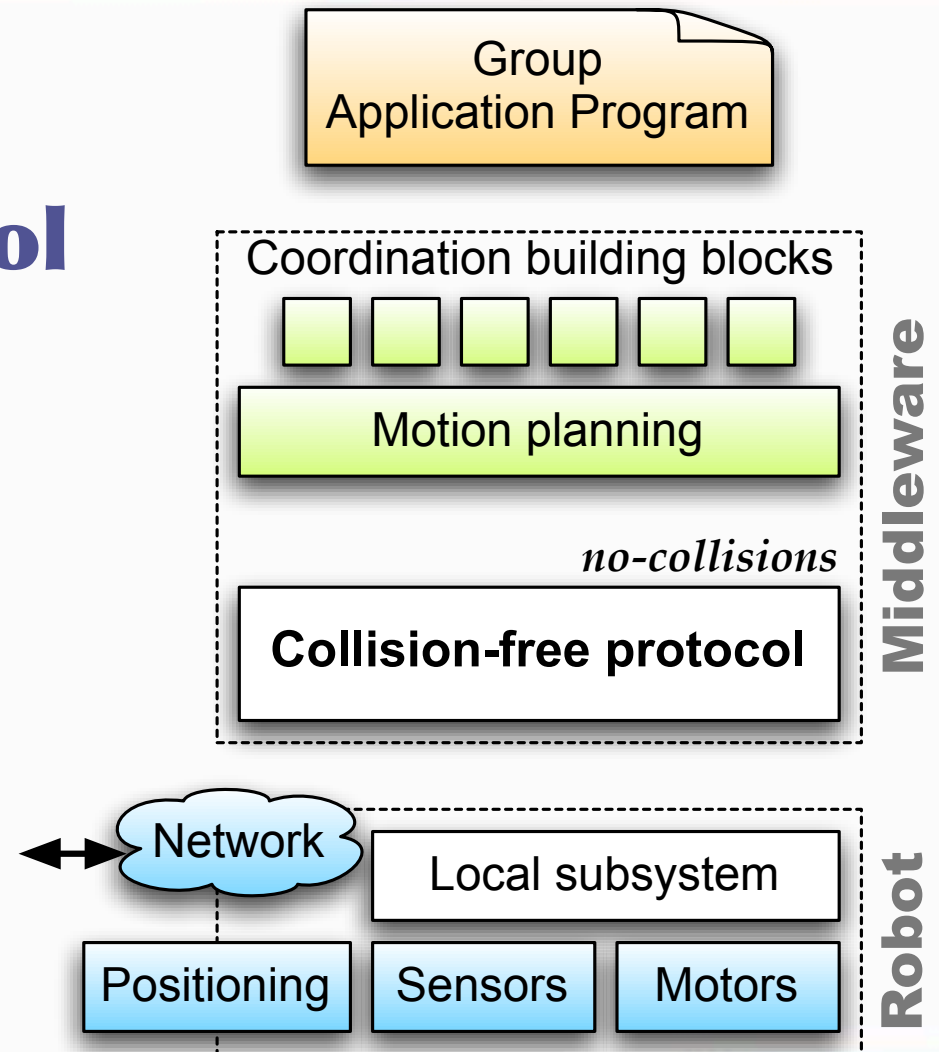
Architecture

- **Collision-free protocol**

- Ensure no-collision
- Fail-safe behavior

- **Local subsystem**

- Individ. robot movements
- Detect inert obstacles
- Use sonars



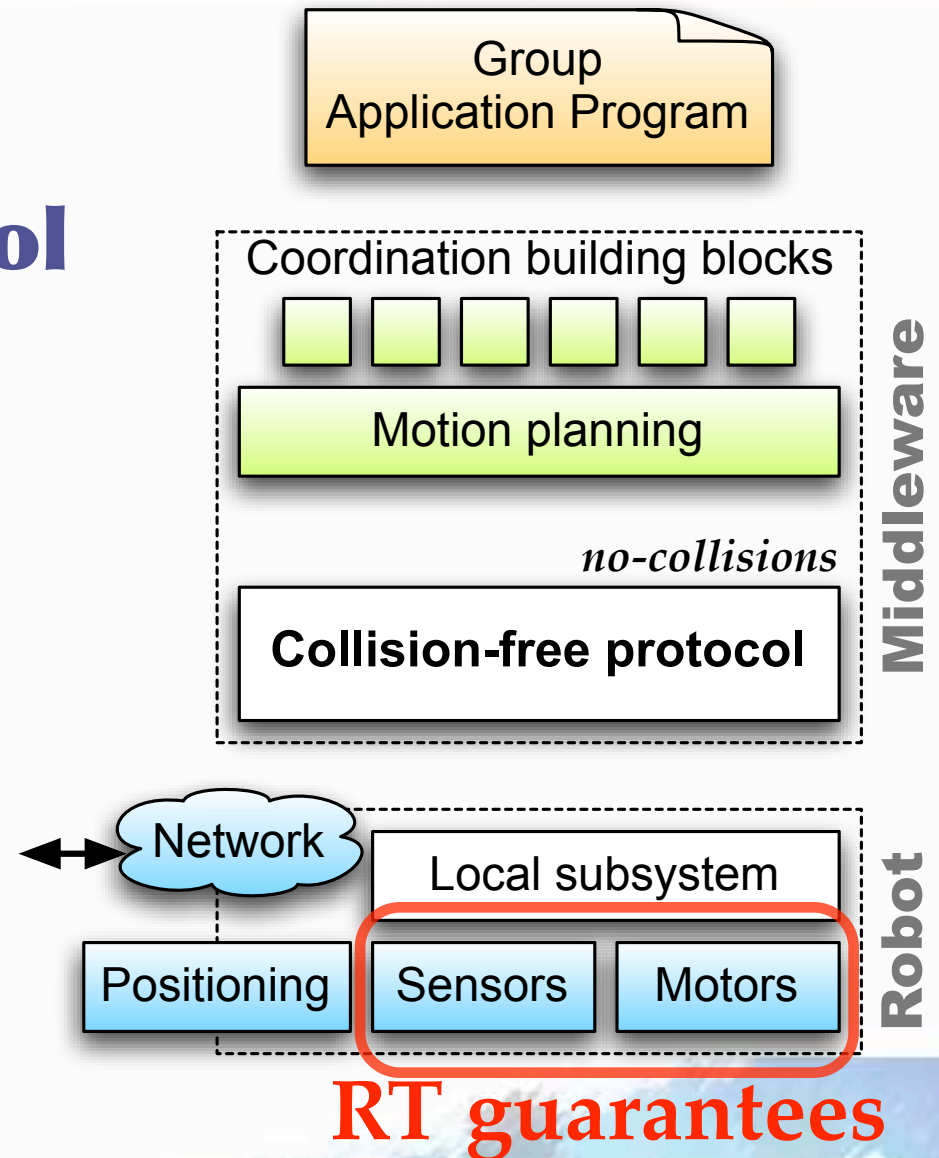
Architecture

- **Collision-free protocol**

- Ensure no-collision
- Fail-safe behavior

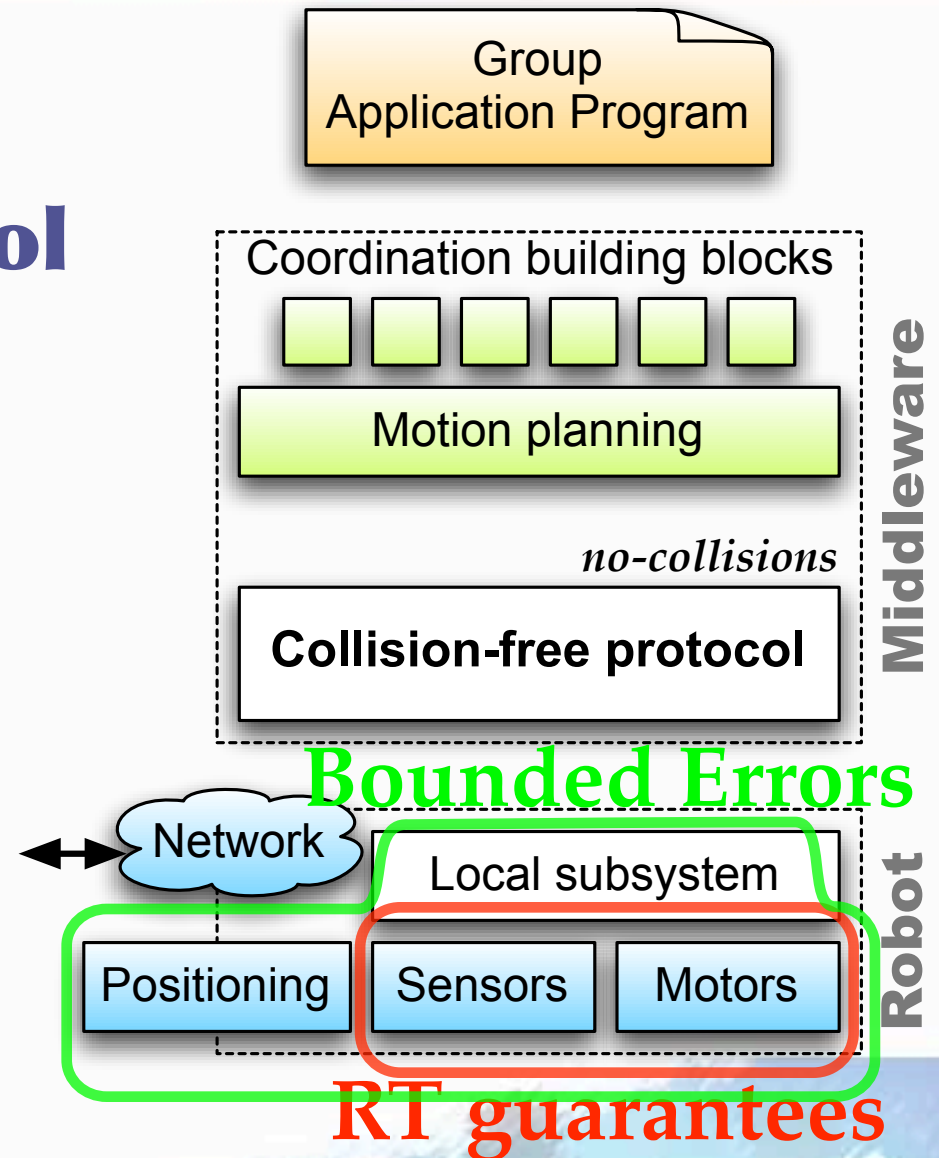
- **Local subsystem**

- Individ. robot movements
- Detect inert obstacles
- Use sonars



Architecture

- **Collision-free protocol**
 - Ensure no-collision
 - Fail-safe behavior
- **Local subsystem**
 - Individ. robot movements
 - Detect inert obstacles
 - Use sonars



Architecture

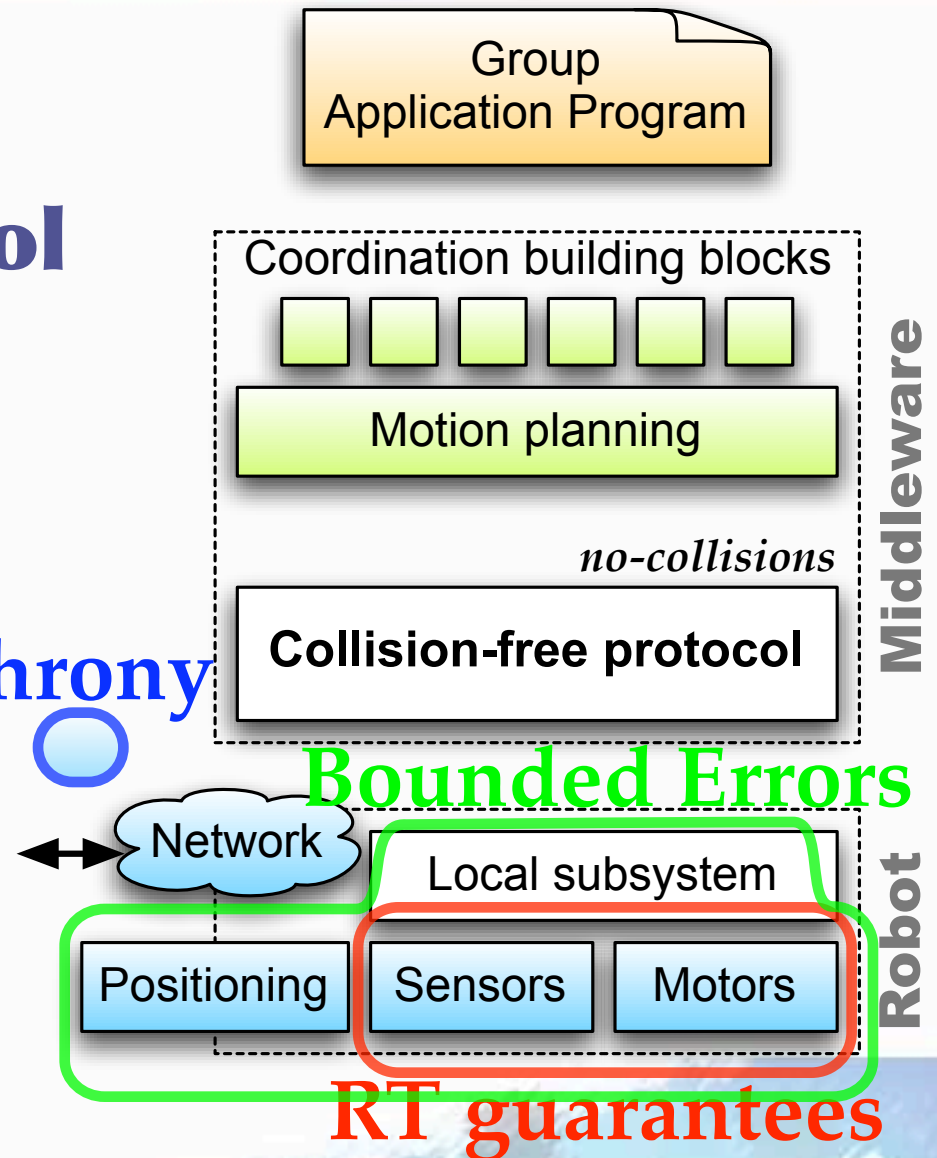
- **Collision-free protocol**

- Ensure no-collision
- Fail-safe behavior

- **Local subsystem**

- Individ. robot movements
- Detect inert obstacles
- Use sonars

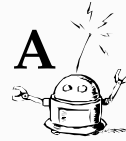
Synchrony



Path Reservation

- **Robot knows**

- own destination / path
- own location



- **Does NOT know**

- others' destinations
- others' location
- others' velocity
- communication delays



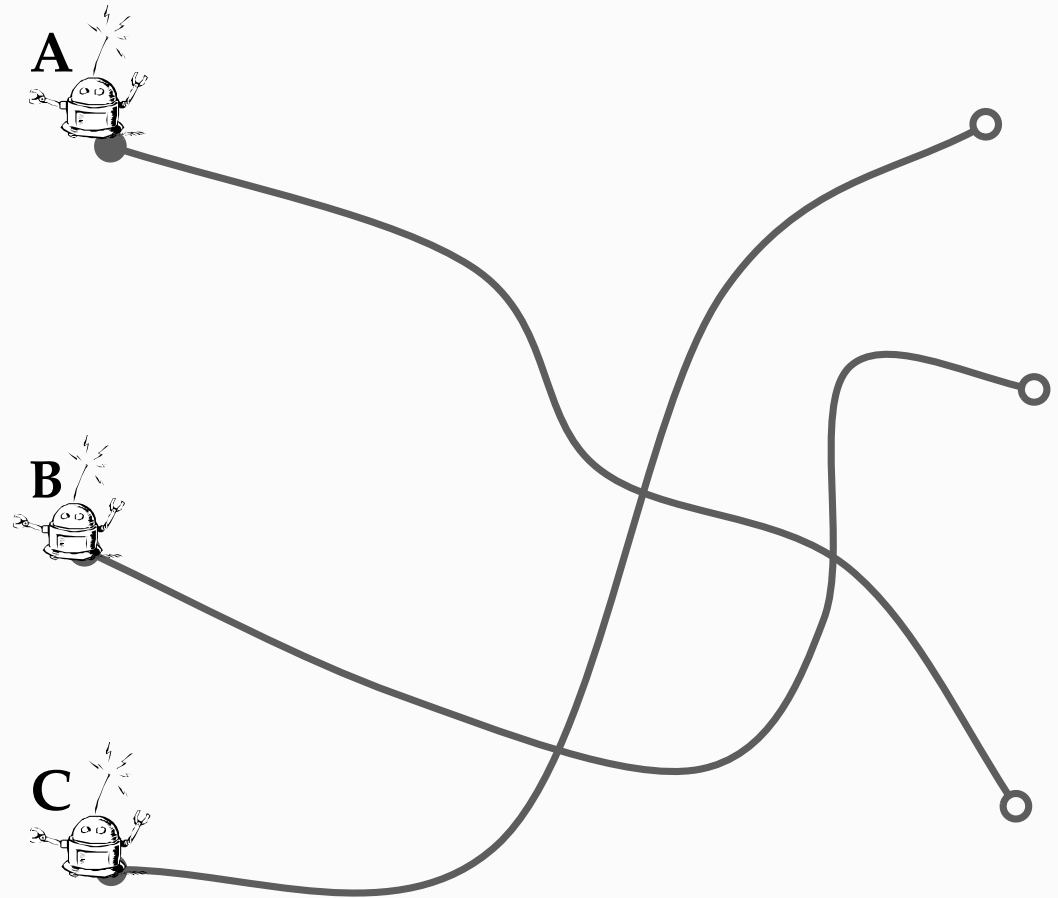
- **Reservation**

- *request*: request lock
- *release*: release lock



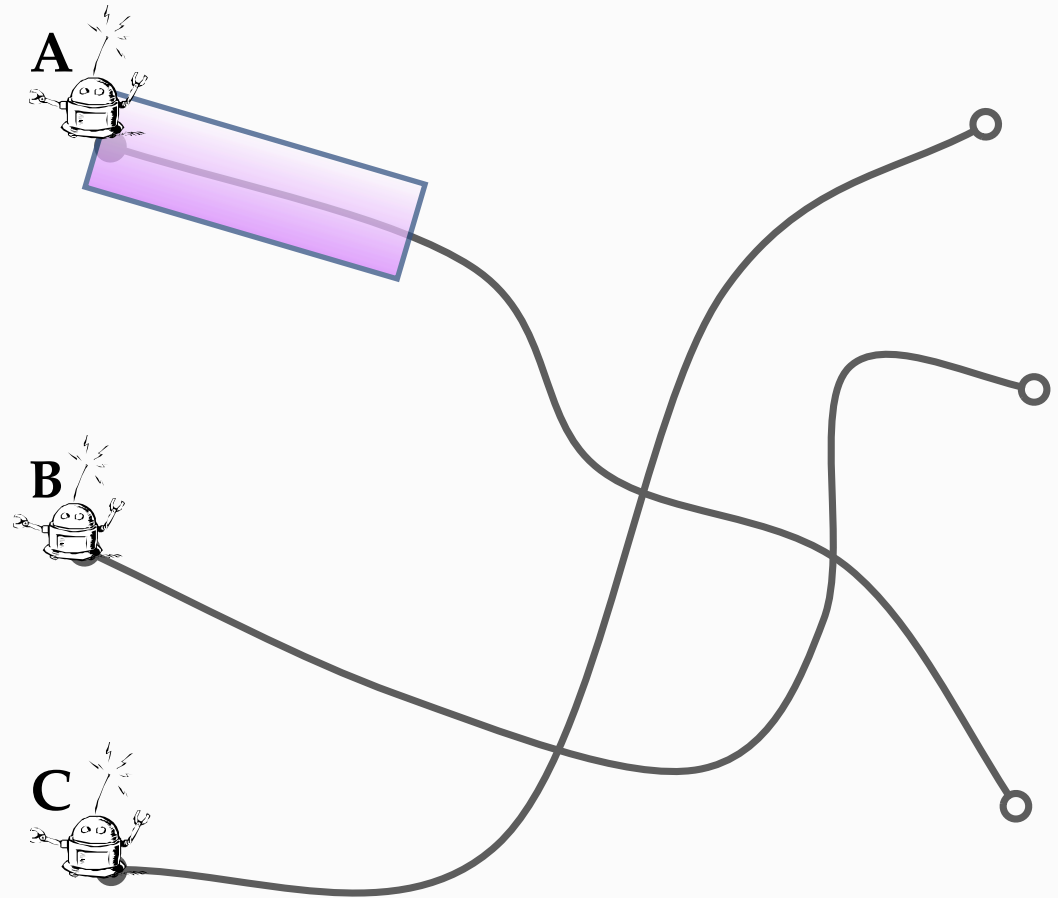
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



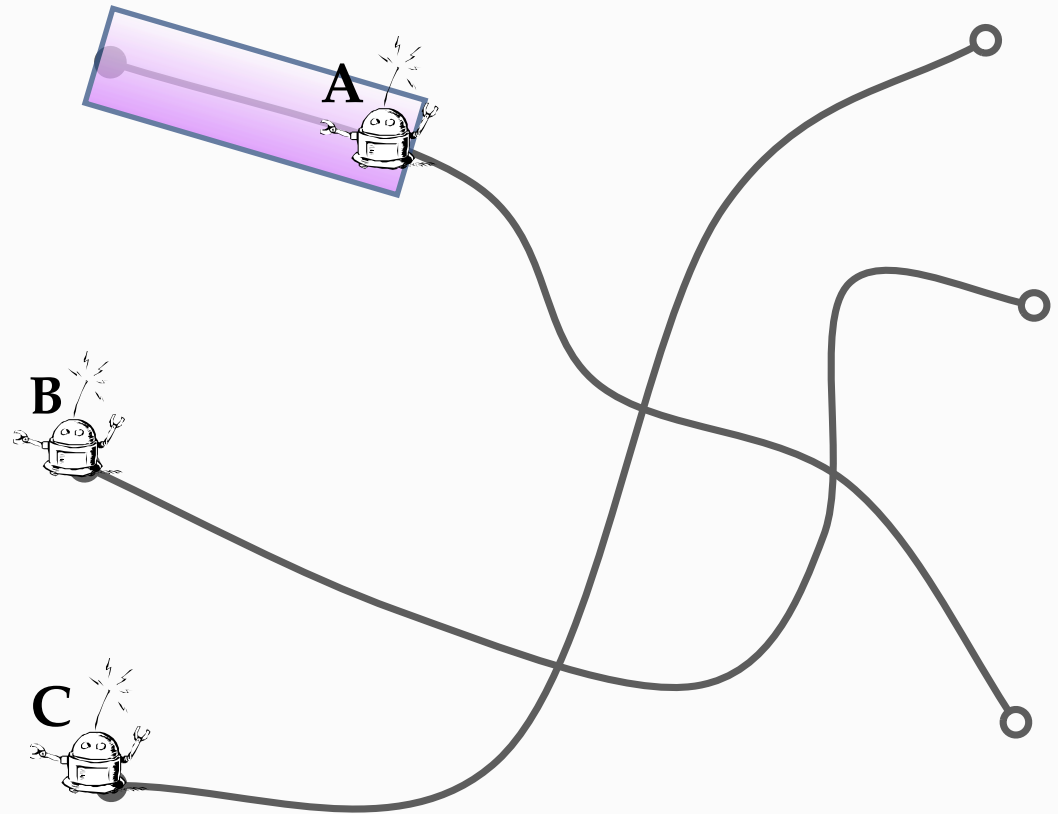
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



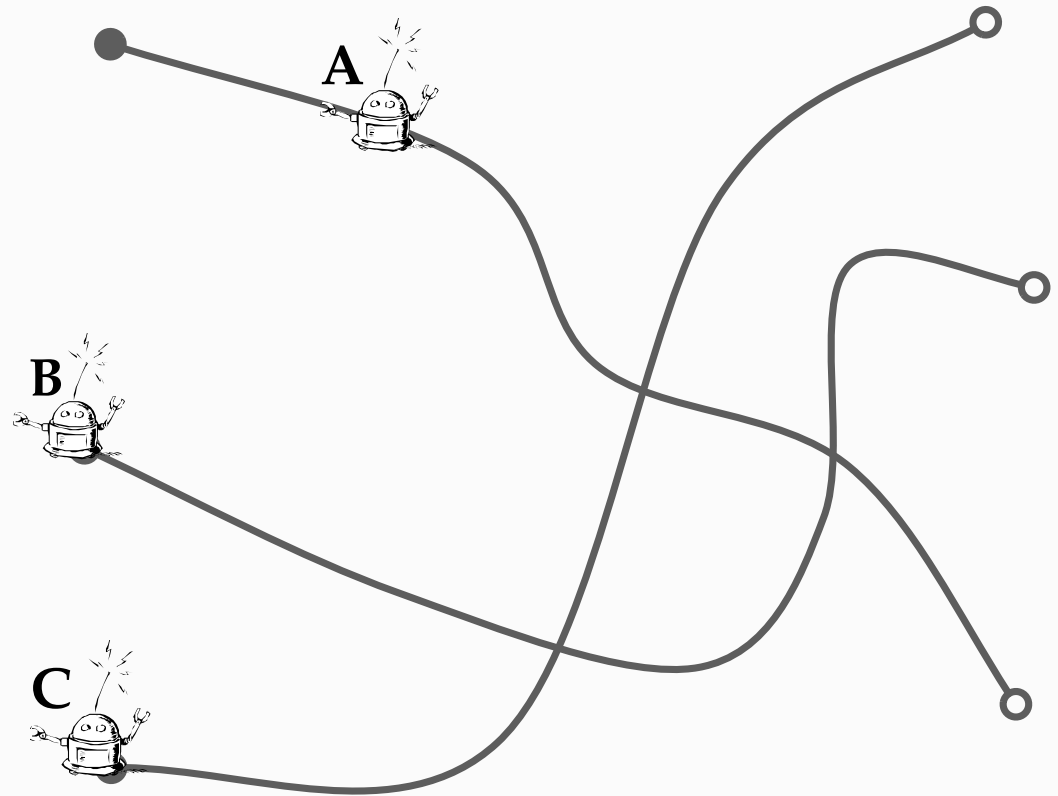
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



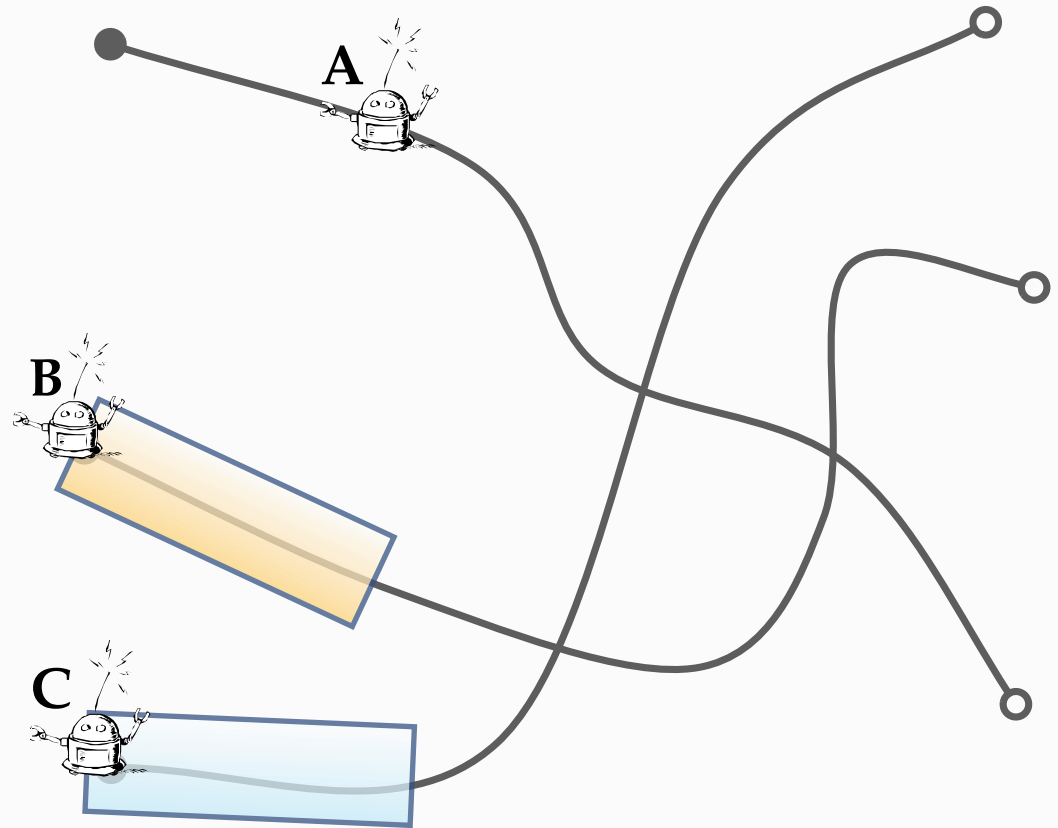
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



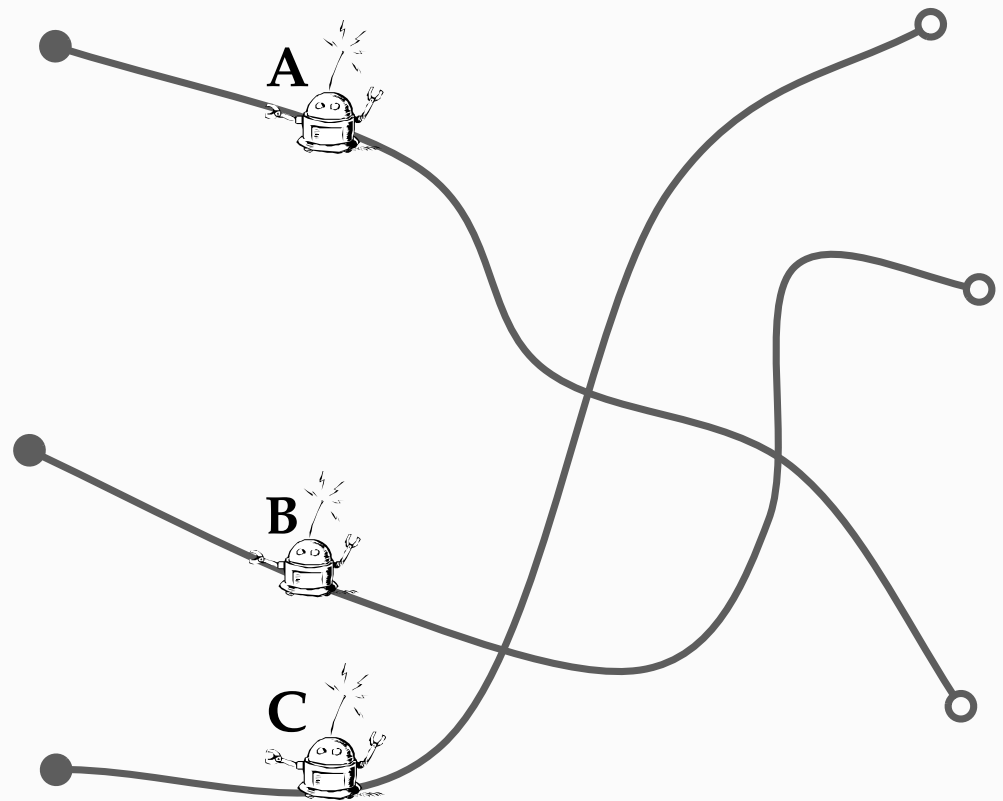
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



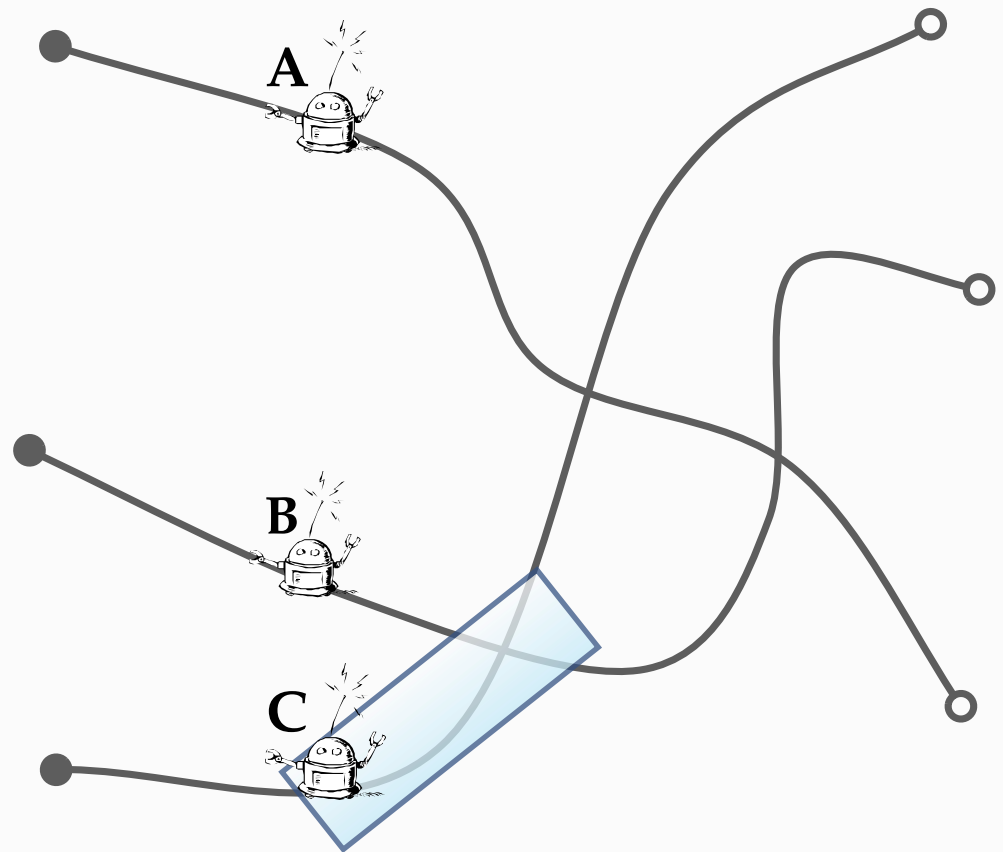
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



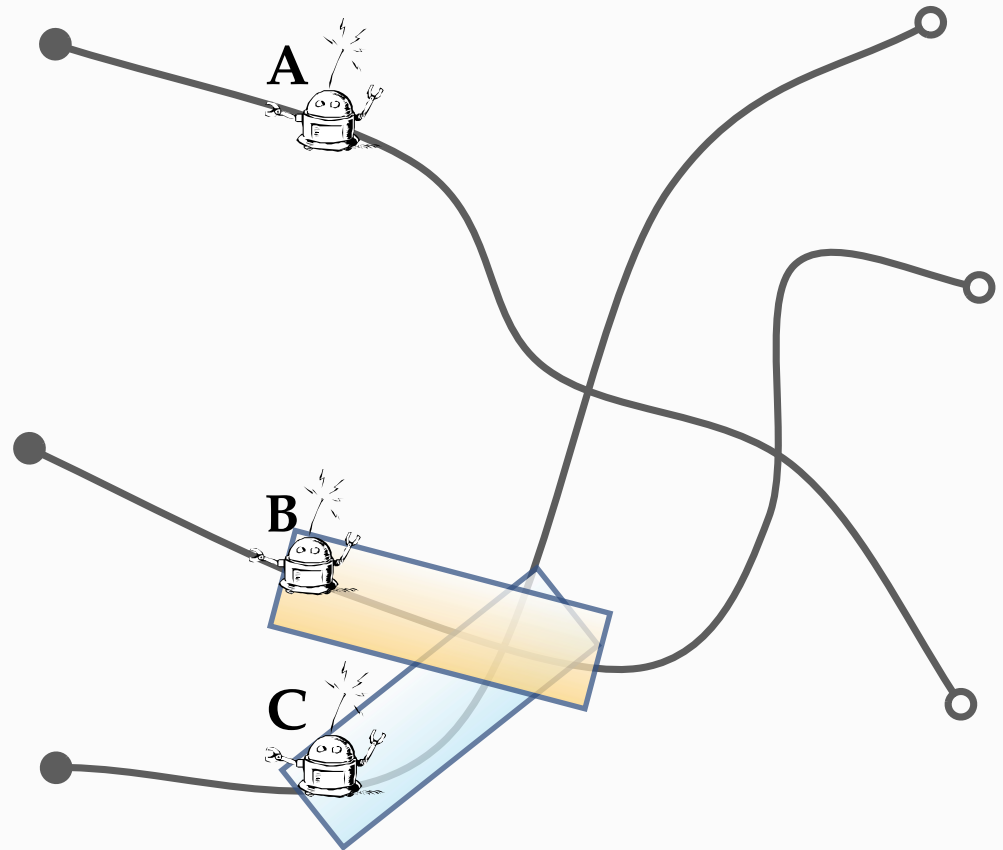
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



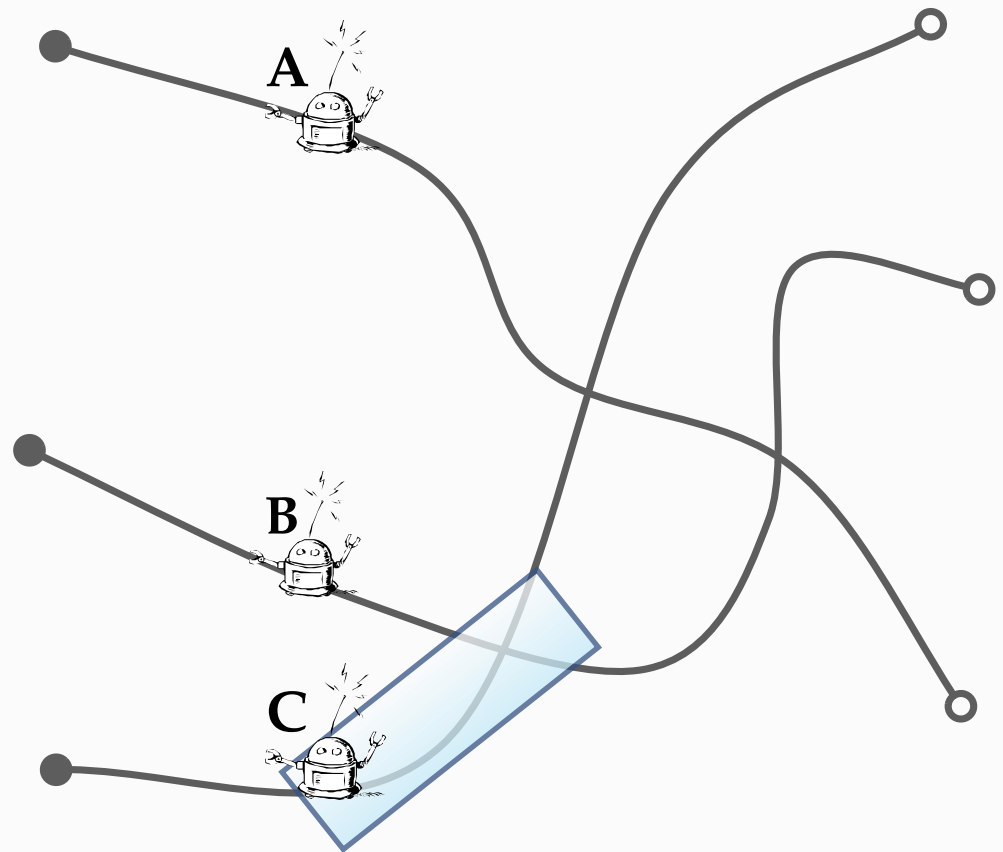
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



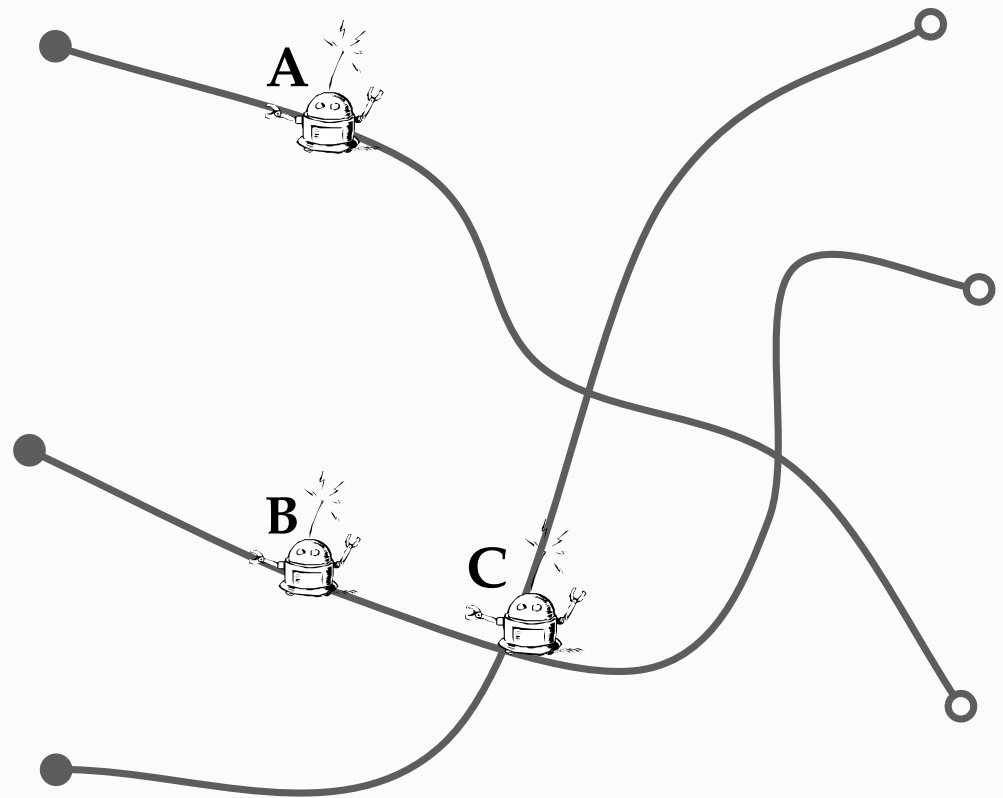
Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



Path Reservation

- **Robot knows**
 - own destination / path
 - own location
- **Does NOT know**
 - others' destinations
 - others' location
 - others' velocity
 - communication delays
- **Reservation**
 - *request*: request lock
 - *release*: release lock



Two Models

[YCDW06]

	Model 1	Model 2
Comm. range	Unlimited	within range D
Group	Static	Dynamic
Group knowledge	Full system	Partial; within range D
Synch. assumpt.	<>S failure detector	Neighborhood discovery
Scalability	Low	Very high
Fault-tolerance	YES	not yet
Deadlocks	Detect locally	Detect within range D

Model 1: Reservation

- **Idea**

- Use Total Order Broadcast
- Local conflict management (deterministic)

- **Advantages**

- Many algorithms (see ACM CS, December 2004)
- Well-known requirements; Fault-tolerant solutions (e.g., with unreliable FDs & maj. correct hosts)

- **Synchrony assumption**

- E.g., unreliable failure detectors
- FAIL => performance degradation

Model 2: Neighborhood Discovery

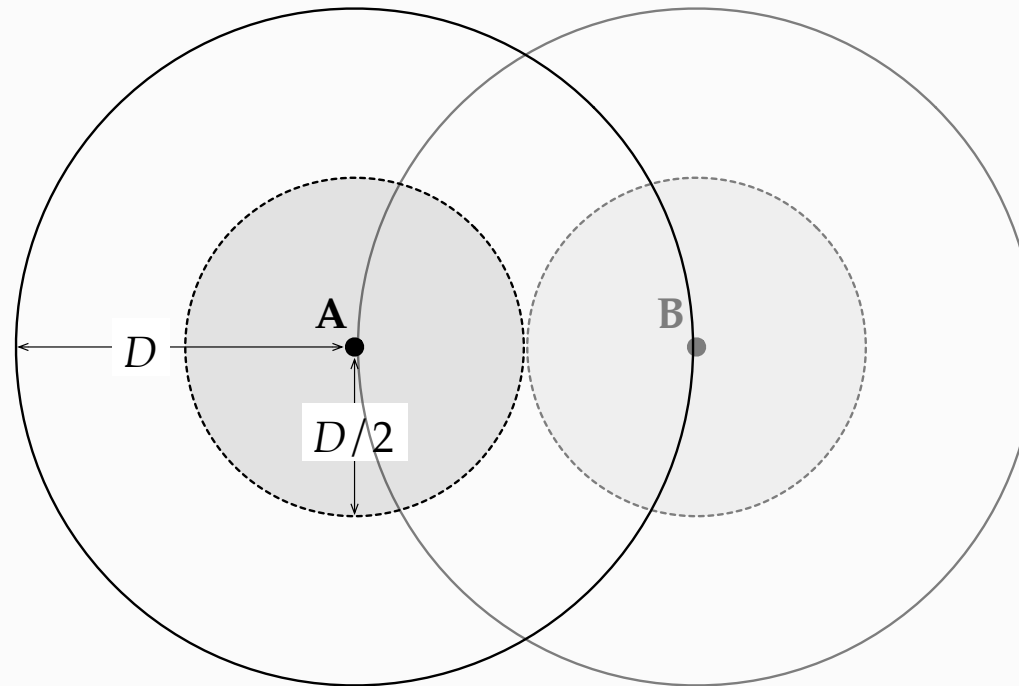
- **Primitive**

- Query by robot R at time t
- Return set $Neighbors(r,t)$

- **Query period**

- From *query* to *return*
- For each robot s' ; during query period:
 - s' in range $\Rightarrow s'$ in $Neighbors(r)$
 - s' not in range $\Rightarrow s'$ not in $Neighbors(r)$
 - s' partially in range \Rightarrow undetermined.

Model 2: Restrictions



- **Restriction**

- Reservations within $D/2$ (- errors)

- **Ensures**

- Cannot conflict without being “introduced”

Ongoing Directions

- **Protocol extension**

- pipelining / interleaving
- Improve FT for ad hoc model

- **Parameter dimensioning**

- robots density
- robot speed / acceleration / braking distance
- communication range / delays
- errors

- **Implementation**