# *A Systematic Approach to Safe Coordination of Dynamic Participants in Real-time Distributed Systems*

Mong Leng Sin

DSO National Laboratories

26th Feburary 2016

Distributed Systems Group
Trinity College, Dublin

# Contents

- Comheolaíocht
  - Modheolaíocht Comhordú (Science of Coordination)
- Motivation and Overview

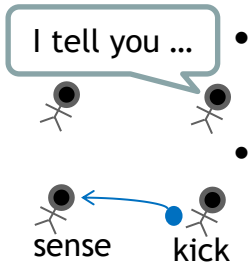
- Our 3-step design
  - System modeling and specification
  - System analysis
  - Protocol derivation
- Evaluation

Slides with green background are not in submitted paper

# Coordination

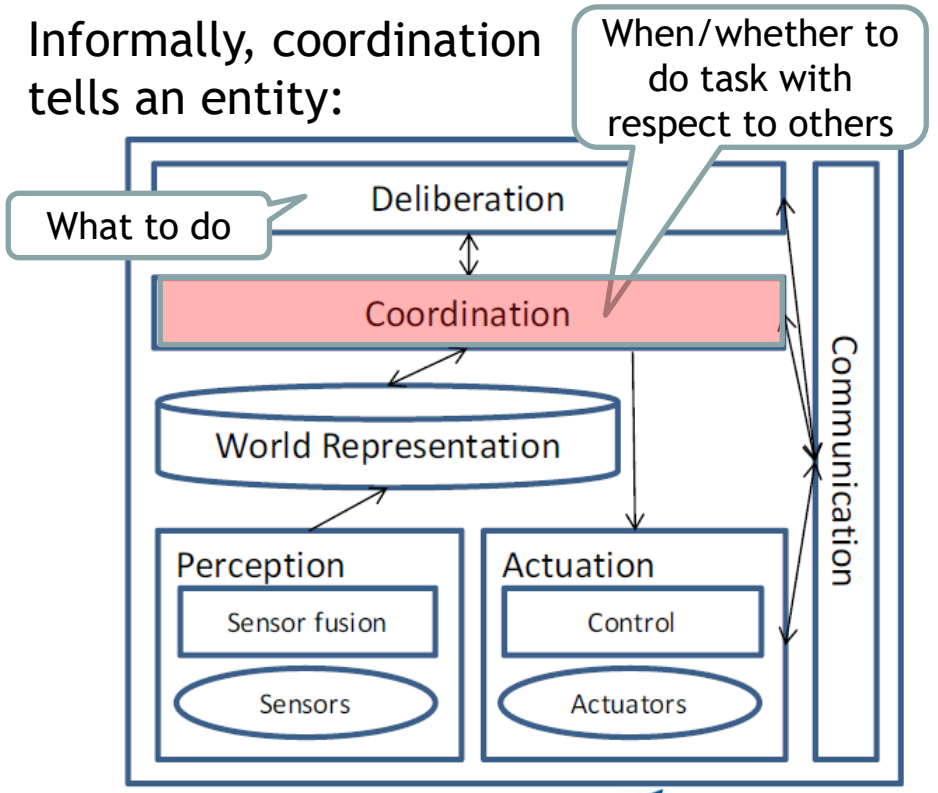
*"the management of interactions both amongst entities, and between entities and their environment, towards the production of a result"* [Bouroche, 2007]

- Interaction management
  - Direct communication (amongst entities)
  - Entities actions/Indirect communication (entities and their environment)




- Production of result
  - Safety: True at all times
    E.g., must not crash
  - Goal: Eventually true
    E.g., arrival at destination

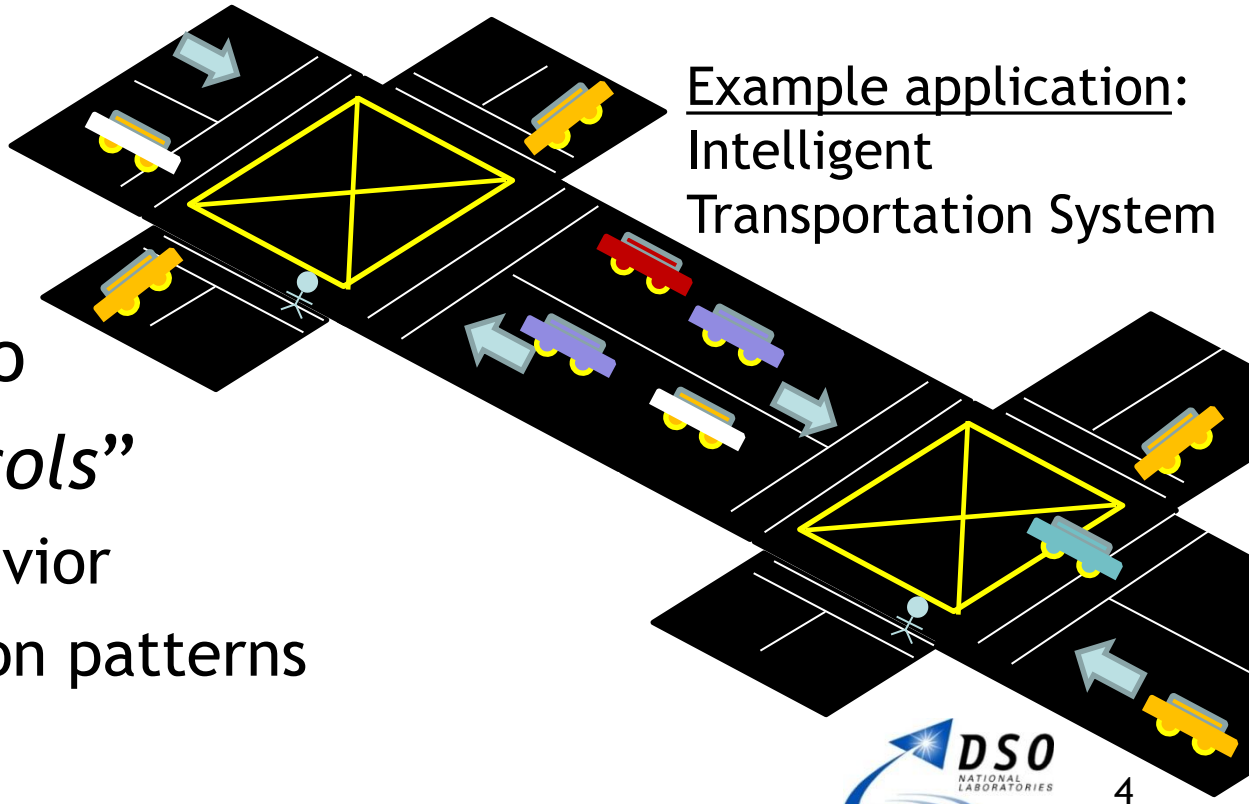
Informally, coordination tells an entity:




Common components within an entity

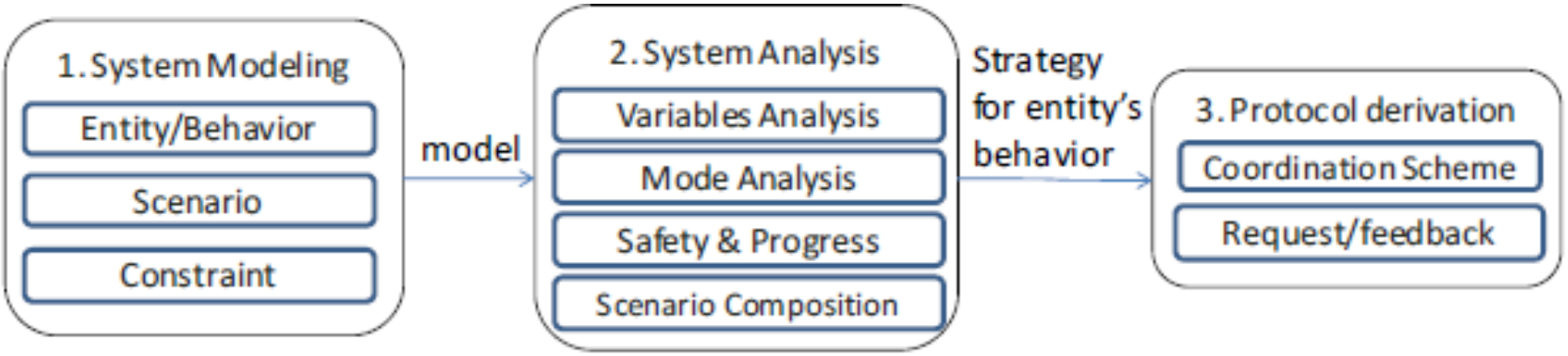
# Motivation and Overview

- Distributed physical entities
- Coordination; access shared resources
- Real-time; in a dynamic environment
- Scalable (wrt)
- Reliable (to)



Example application: Intelligent Transportation System

- An approach to design "*Protocols*"
  - Entities' behavior
  - Communication patterns

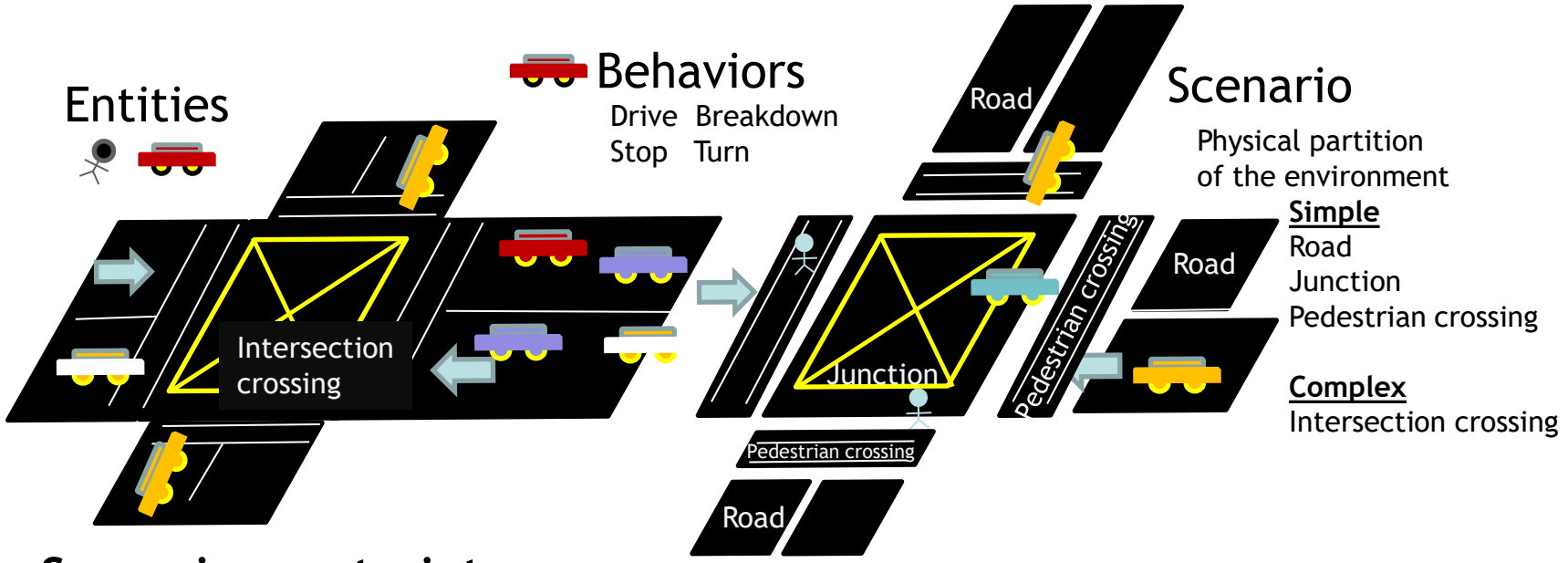



System modeling and specification

System analysis

Protocol derivation

# OUR 3 STEP DESIGN



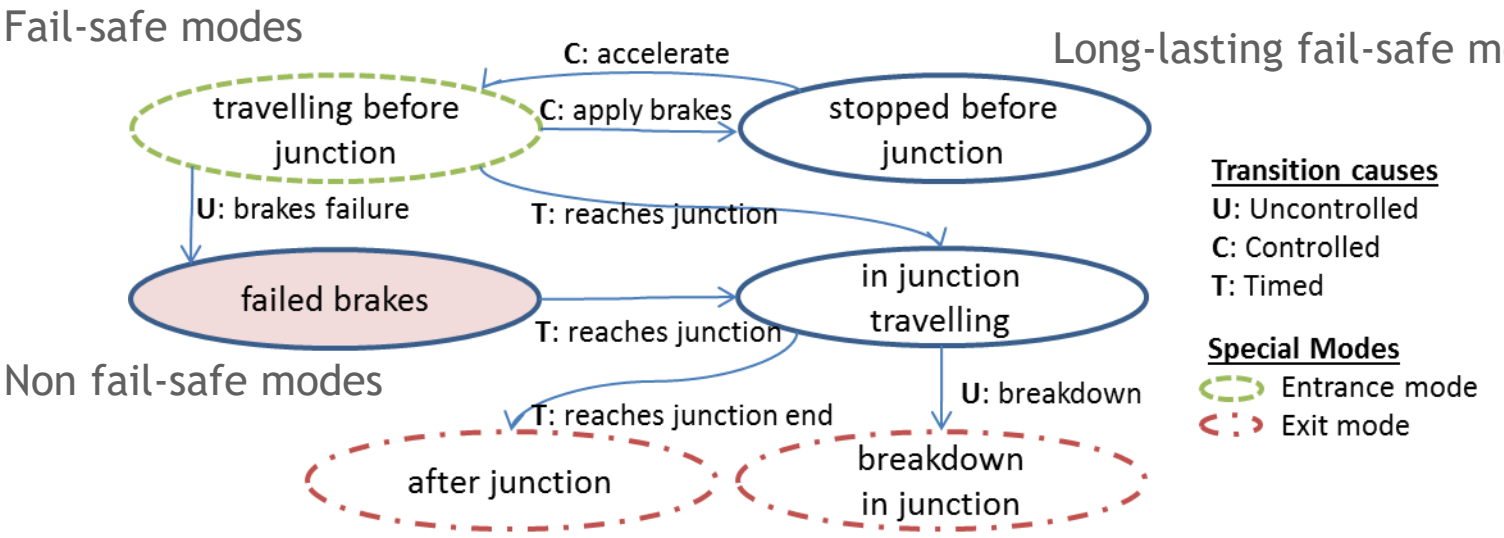

## Scenario constraints

- Scenario abstraction : Different-time event ordering
- Scenario setting : Sequence event ordering
- Entrance/Exit/Safety : Pre-condition/Post-condition/Invariant

# STEP 1: SYSTEM MODELING & SPECIFICATION

# Entrance/Exit/Safety Constraints

- Safety constraint
  - The condition that must be satisfied at all times
  - The scenario **invariant**
- Entrance constraint
  - Must be satisfied just before an entity enters the scenario
  - The scenario **pre-condition**
  - E.g., vehicle enters a road one at a time and at a speed such that it has sufficient time to stop without collision to the vehicle in front
- Exit constraint
  - Must be satisfied just before an entity leaves the scenario
  - The scenario **post-condition**
- Captures the safety-conditions across scenarios



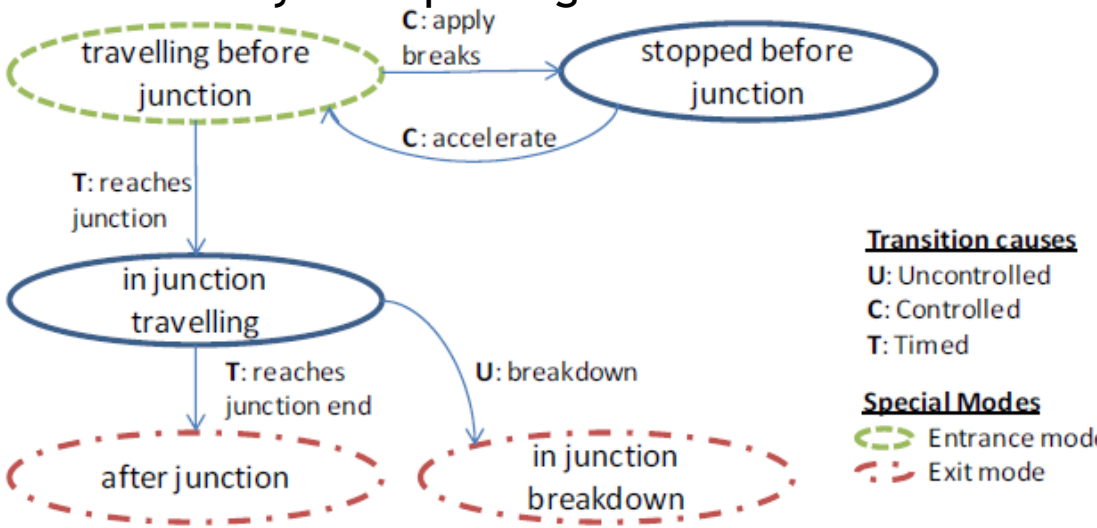

- Modes & Mode transition diagram **(Finite state machine)**
- Strategy: does an entity has a fail-safe {condition-behavior} that allows it to achieve its goal safely? What is this strategy?
- Composition: extension of safety analysis across scenarios

# STEP 2: SYSTEM ANALYSIS

# Mode transition diagram

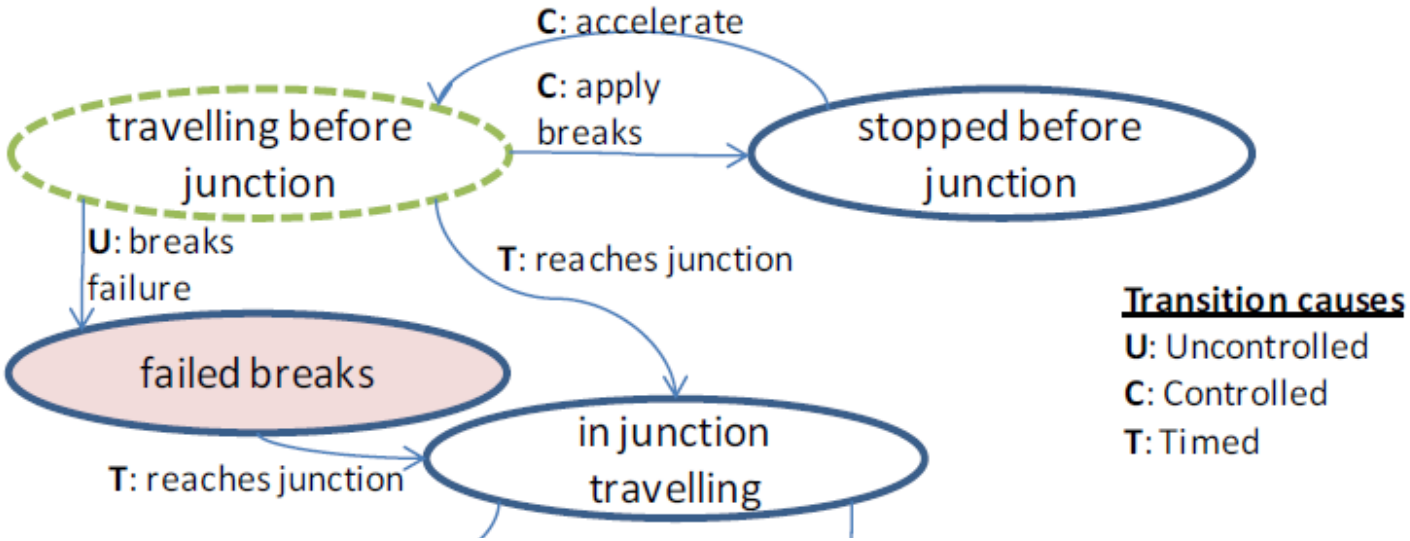
A ***mode transition*** from a mode *x*, to another mode, *y*, happens when an entity state changes states from $s_0$ to $s_1$, such that $s_0 \in x$ and $s_1 \in y$. An entities' ***mode transition diagram*** describes all modes and possible mode transitions.

- Transition causes
  - **Controllable**: caused by entity's deliberate actions; an entity can choose (not) to perform those actions that causes the transition
  - **Uncontrollable**: some external events causes the transition
    - The external event could causes the entity to perform some actions which are uncontrollable. E.g., burst vehicle's tire
  - **Timed**: subset of uncontrollable transitions, timed transition are transition caused by time passing



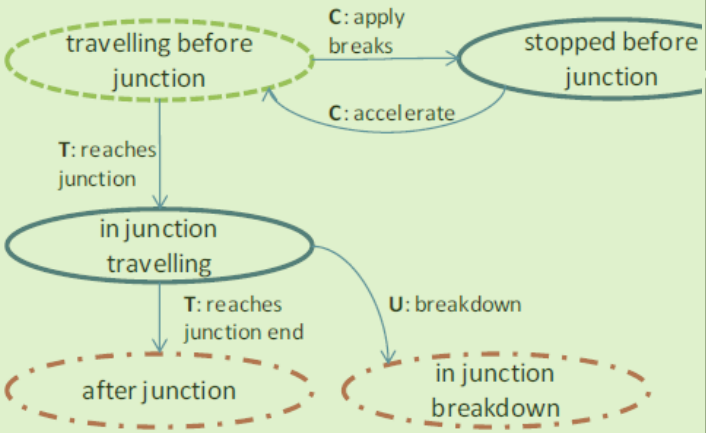
# (Non) Deterministic Transition

- Can an entity perform a transition deterministically?



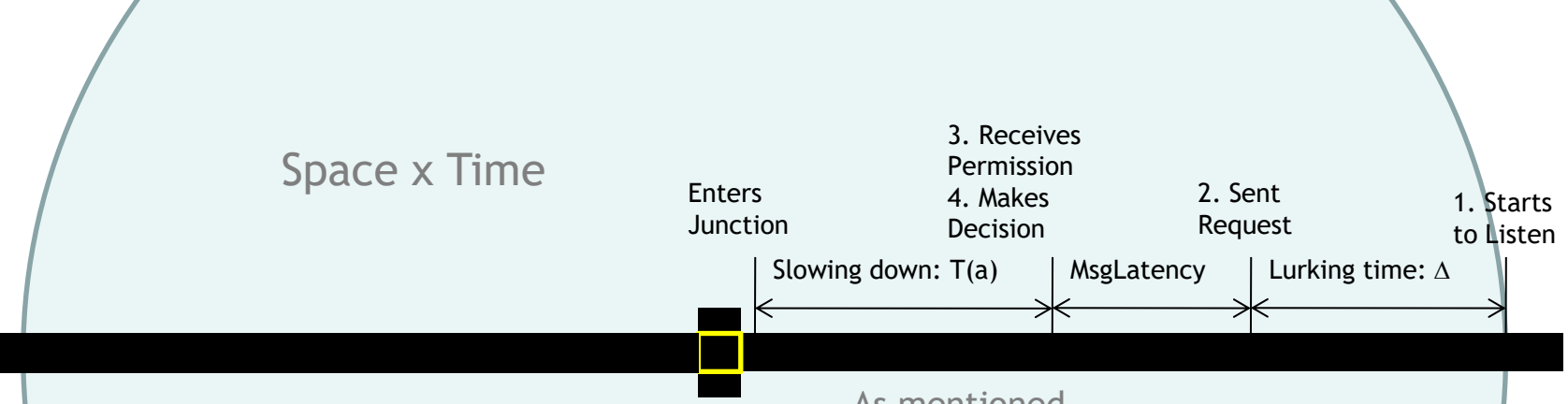

| Edge under examination | Other edges from the same vertex | Can entity transition along edge under examination deterministically? | Do the transition has deterministic transition time? |
|---|---|---|---|
| uncontrolled | some timed/uncontrolled | No | - |
| | all controlled | Yes | No (possibly infinite) |
| timed | some timed/uncontrolled | No | - |
| | all controlled | Yes | Yes |
| controlled | all controlled/timed | Yes | Yes |
| | some uncontrolled | No | - |

# Coordination Strategy




| Index | Mode | Result | Condition | Comments |
|---|---|---|---|---|
| 1 | traveling before junction | Safe | O:3 \| 2 | Entrance |
| 2 | stopped before junction | Safe | - | LLFSM |
| 3 | in junction traveling | Safe | I:1 \| 2 | non-fail-safe |
| 4 | after junction | Safe | - | Exit, LLFSM |
| 5 | in junction breakdown | Safe | - | Exit |

- Result
  - Safe
  - Unsafe
  - Deadend-1; no out-edge
  - Deadend-2; no safe out-edge
  - Pending; during calculation
- Condition
  - O: 3|2
    - Take out edge 3 after successful coordination, otherwise take out-edge 2
  - I: 1|2
    - Arrive from node 1, may be required to coordinate in mode 2.




As mentioned,

- Responsibility; not accessing the shared resources
- First-come/first-served; to decide winner in races

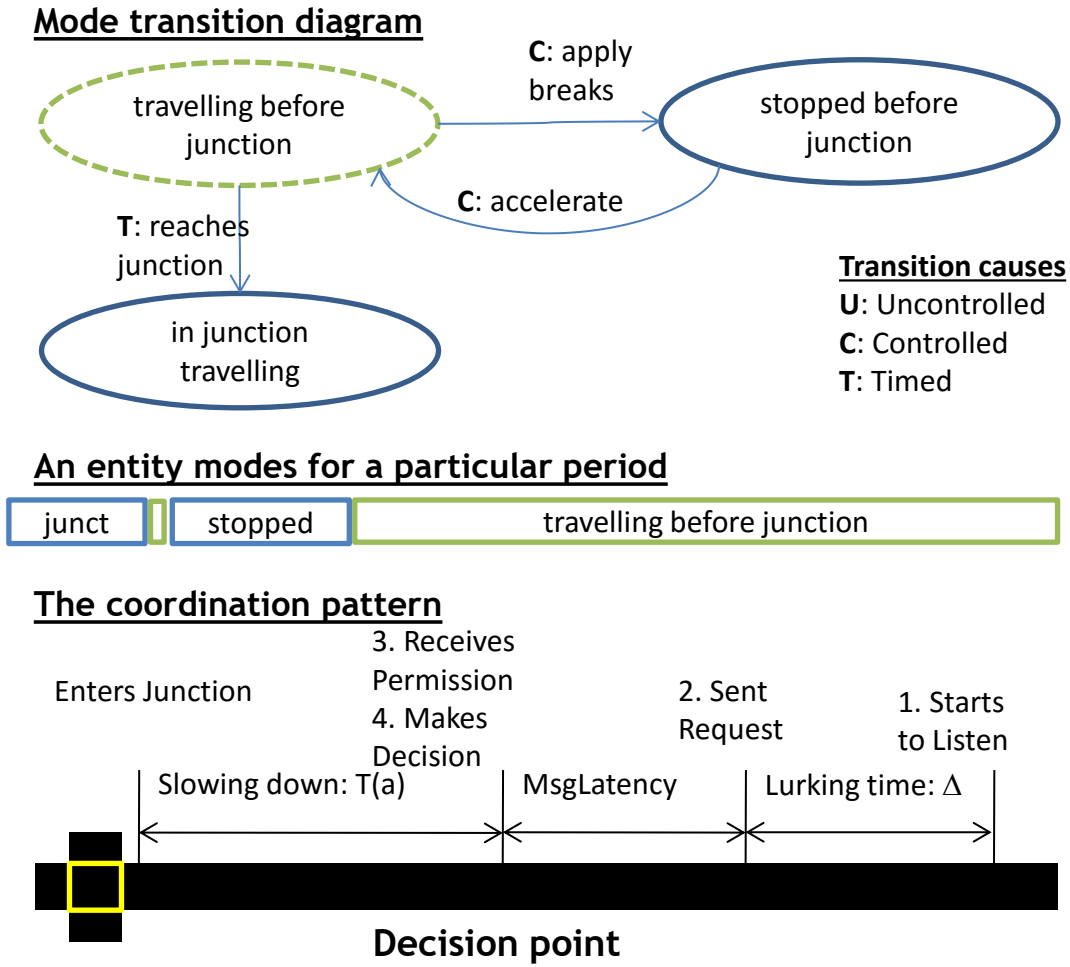
- To be scalable, entity only listens for some period and send requests to a local area

- Coordination pattern
- Request feedback protocol
  - Inference & score
  - Beta constraints
- Rescheduling

# STEP 3: PROTOCOL DERIVATION

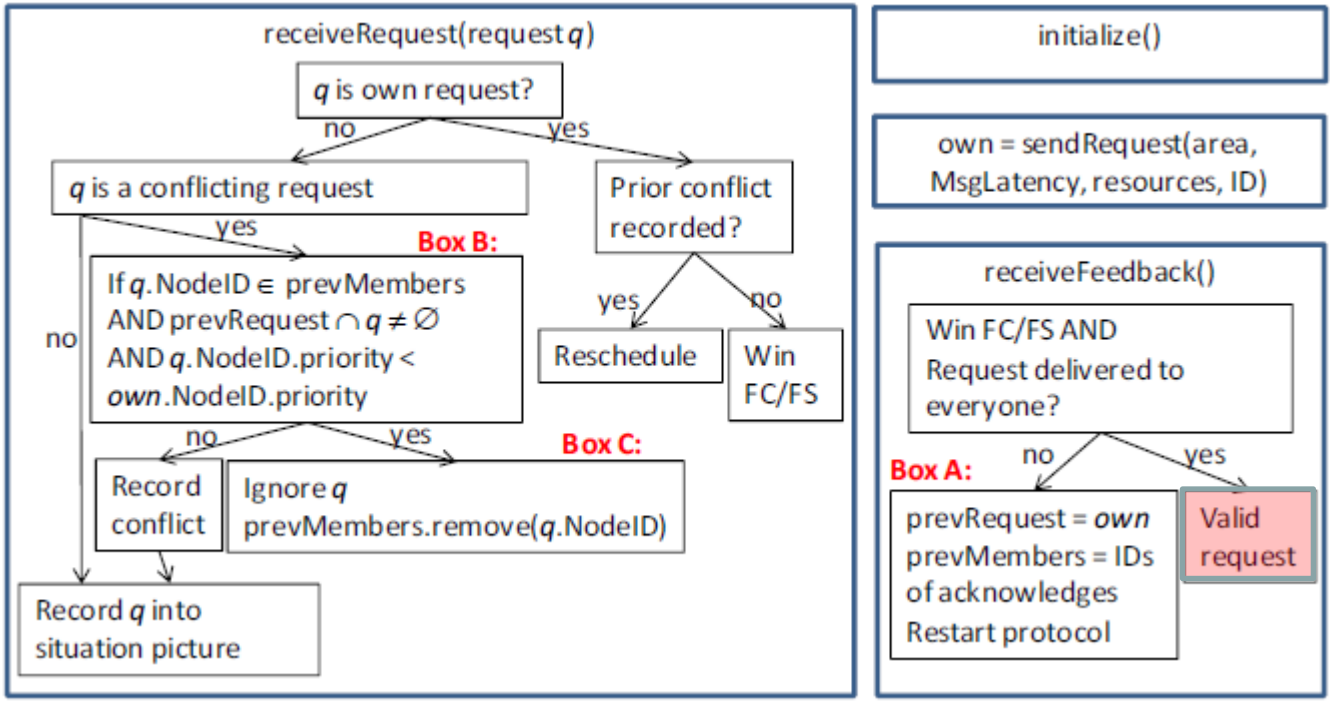
# Coordination Pattern (CwoRIS)

- The **decision point** is the last point in time where an entity must start changing its behavior to avoid accessing the shared resources.
- **Sending area**; an entity must deliver its request to every entity that might use some of the shared resources.
- **Lurking time**; an entity must lurk for a period in order to ascertain that no other entities is holding on to the resource it requires prior to its arrival.



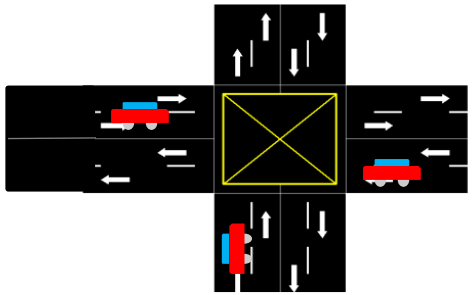

* Calculations of the location of decision point, sending area and lurking time are not presented, they are shown in the thesis.

# Request Feedback protocol (Inference)

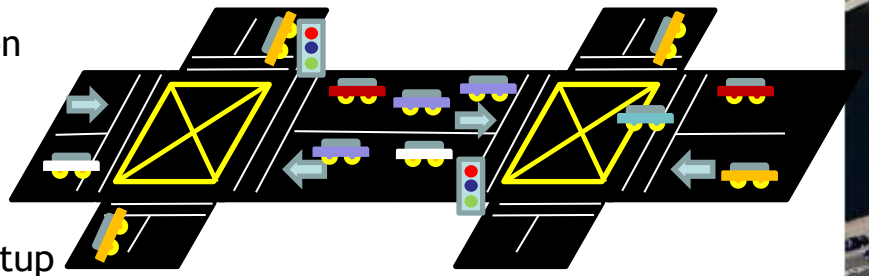



- Valid request (Win FC/FS & Request delivered to everyone)
- On detection of a race, an entity may ignore a request when
  - It has a previous conflicting request
  - That is delivered to the sender
  - The entity has higher priority than the sender

Simple cross junction



Liffey junction



* Picture from Google Map

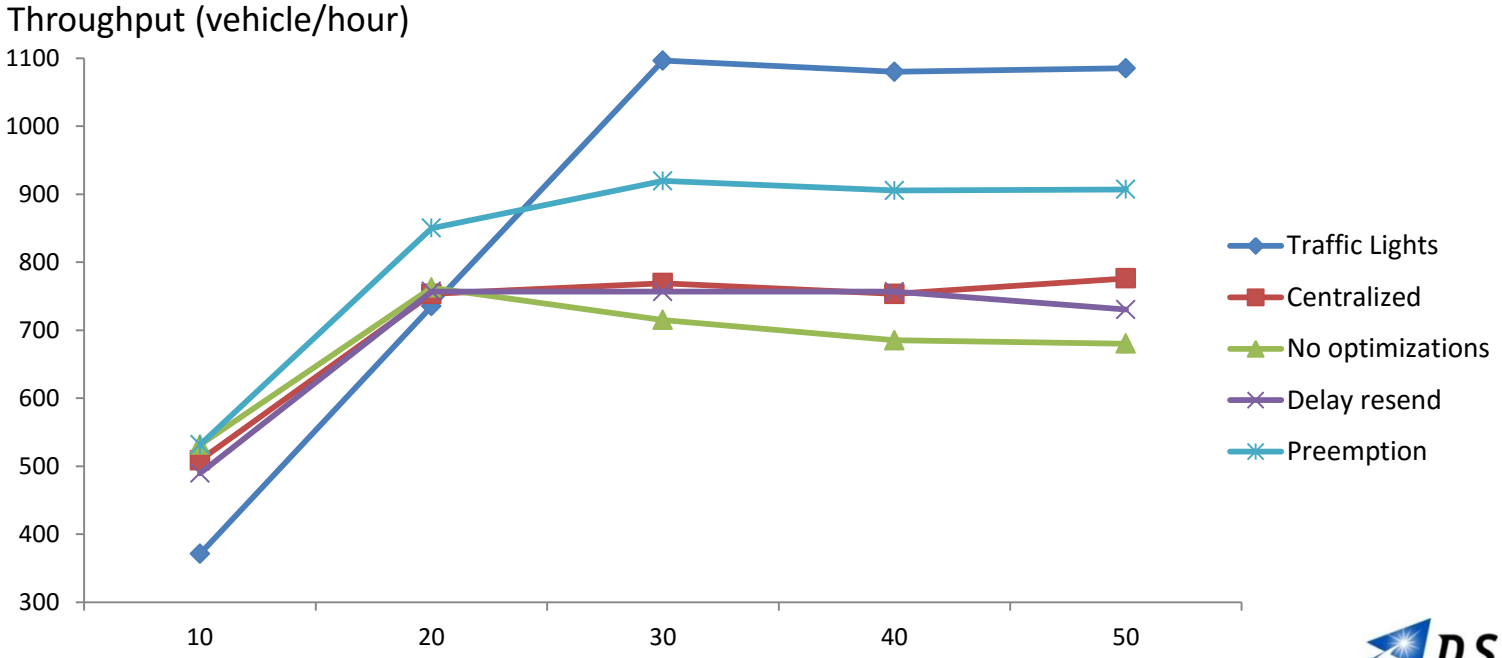
Simulation setup

- Player & Stage
- Maximum vehicle speed, 60km/h
- Road leading into junction is 280m (based on 2* maximum sending area; not all vehicles receive a sent request)
- Vehicles generated randomly at entrance of road, with random destination
  - Vehicle density parameter: the number of vehicles in the scenario at one time; used for measuring scalability

# EVALUATION

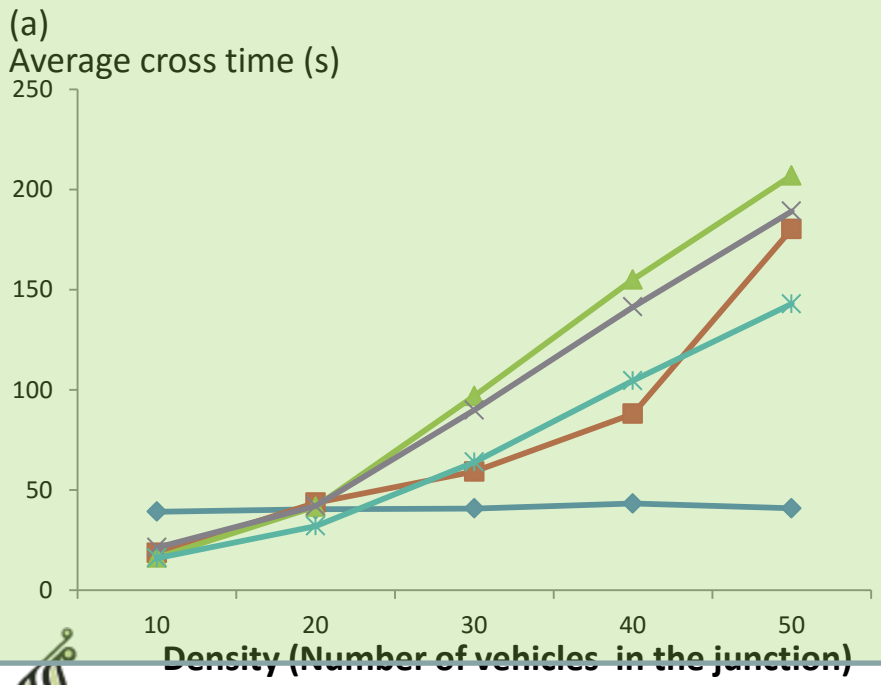
# Comparisons with other protocols

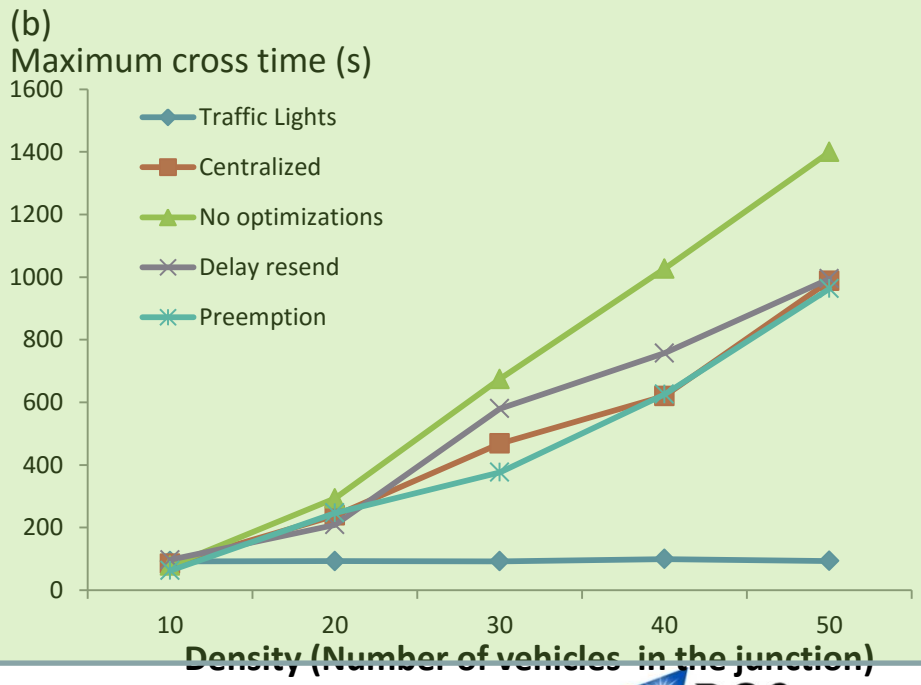
- Better throughput at low density
  - Traffic lights; natural platooning effect
- Better than centralized approach!
  - Right-turns given same priority in centralized approach; decentralized approach – a free-for-all market
  - Allocation fragmentation in centralized approach



# Comparisons with other protocols (2)

- Traffic light has best average/maximum crossing time
  - Platoon effect
  - Optimized-by-hand solution (the traffic lights time)
- Our preemption-based protocol is comparable to the centralized protocol

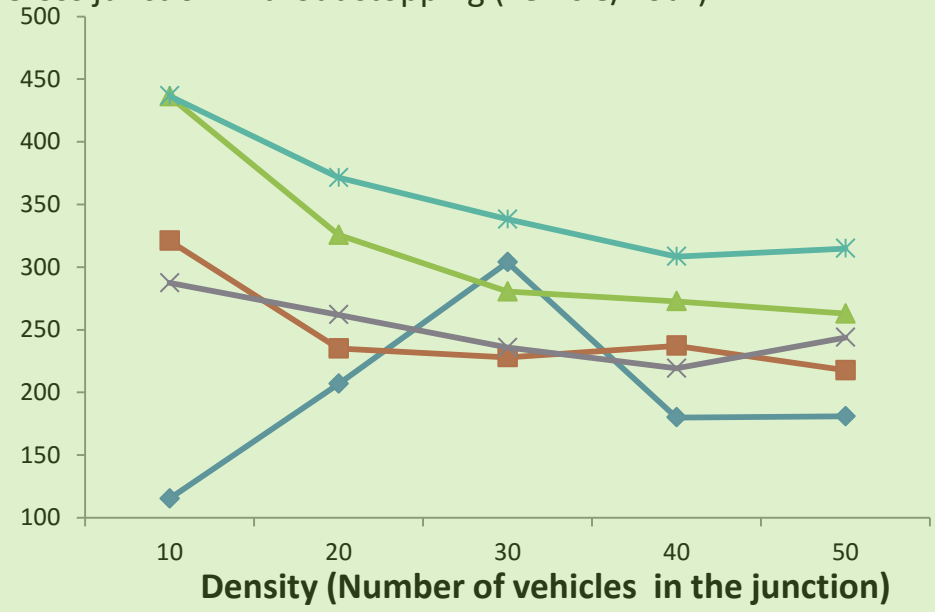


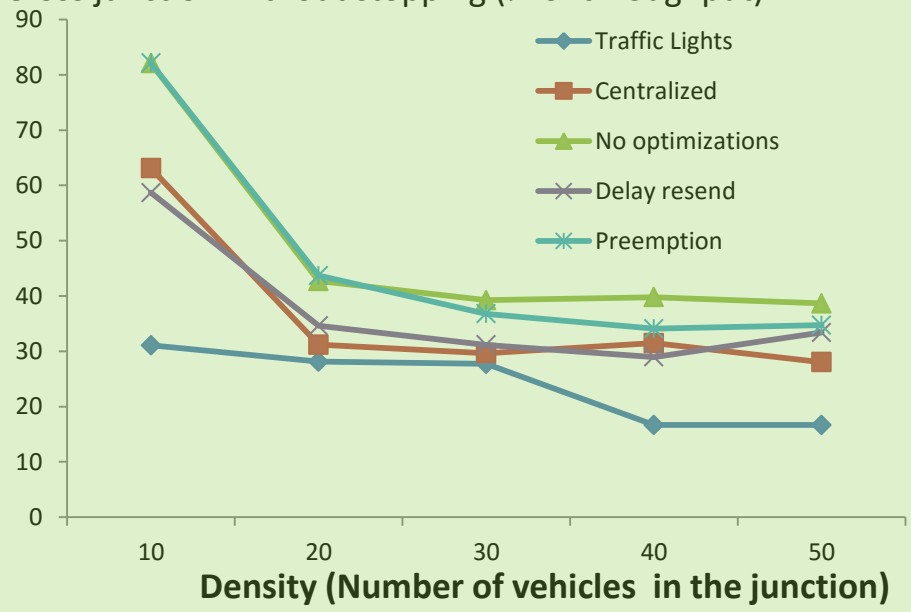
# Comparisons with other protocols (3)

- Using our protocol, most number and percentage of vehicles may cross the junction without stopping

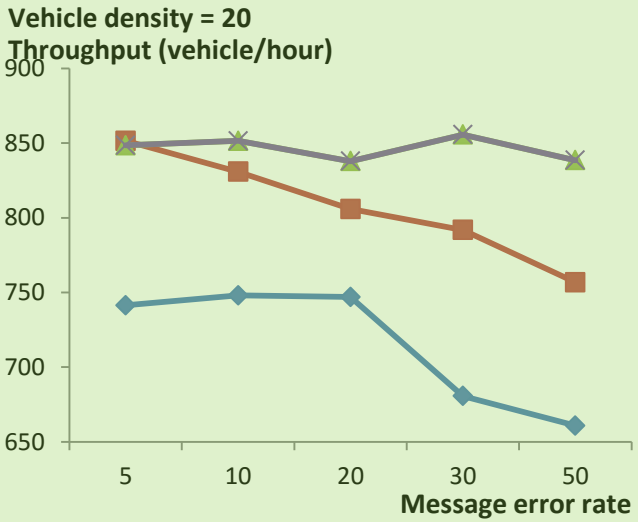
(a)

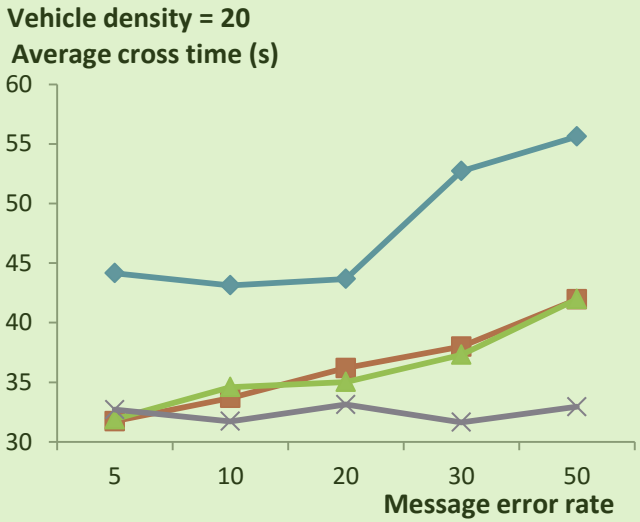



(b)



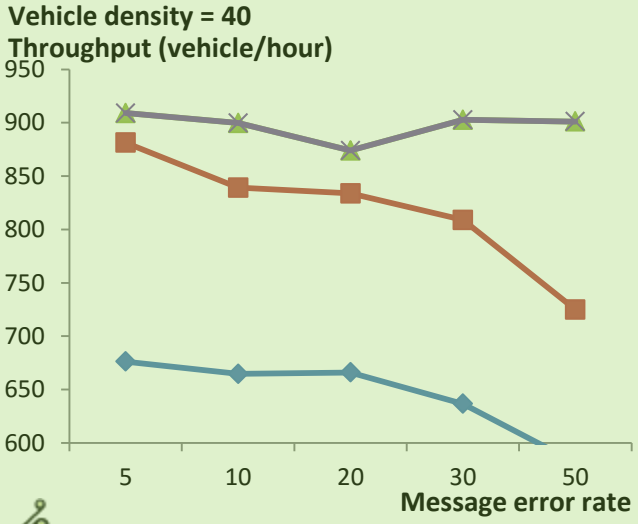
# Reliability – communication errors

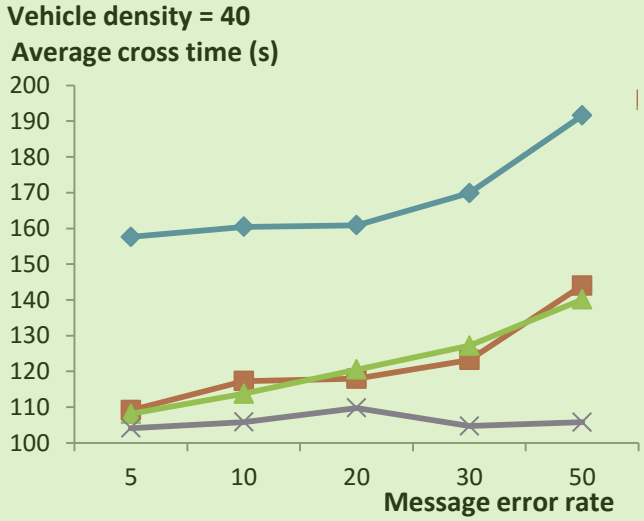


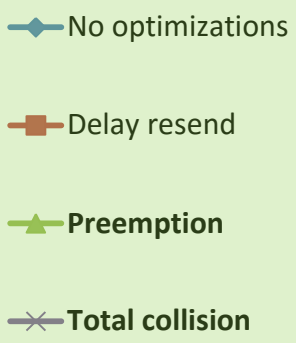




No crashes observed

Performance degrade when Message errors > 20%




DSG Distributed Systems Group Trinity College Dublin

DSO National Laboratories

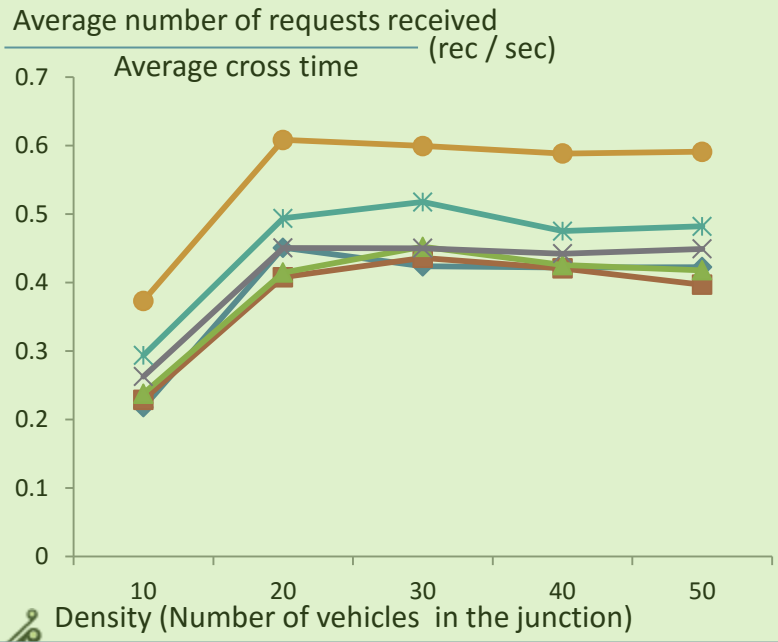
# Reliability - Vehicle breakdowns

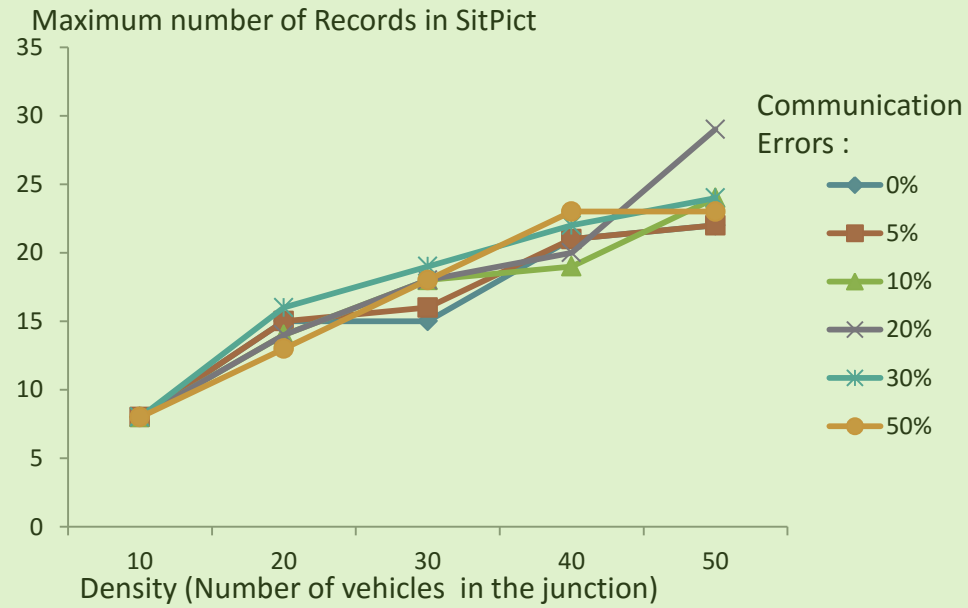
- Vehicle density = 20 (fastest moving vehicles)
- 24 Hours of simulated time * 4 runs

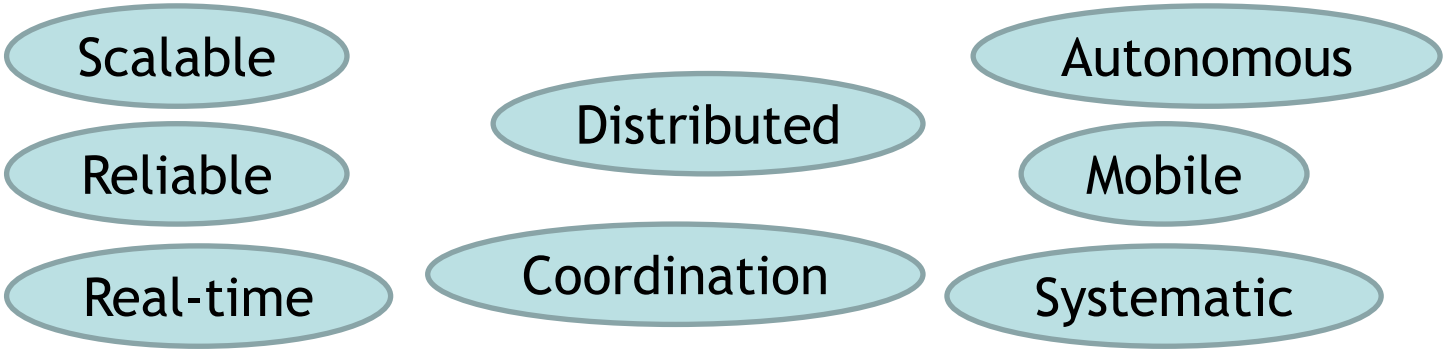
| Communication Error | Number of Vehicles | Average cross time | Maximun cross time | Breakdowns | Crashes |
|---|---|---|---|---|---|
| 0% | 19187 | 34.8 | 241.6 | 646 | 0 |
| 5% | 18869 | 36.3 | 254.6 | 662 | 0 |
| 10% | 18859 | 36.2 | 238 | 675 | 0 |
| 20% | 18356 | 38.4 | 249 | 667 | 0 |
| 30% | 18234 | 39.2 | 274.2 | 609 | 0 |
| 50% | 17187 | 44.9 | 325 | 569 | 0 |

# Scalability

- Number of request received
  - Bandwidth & computational power
- Number of records stored in Situation picture data structure
  - Memory & computational power




Scalable
Reliable
Real-time
Distributed
Coordination
Autonomous
Mobile
Systematic

# CONCLUSION

# References


- H. Attiya, A. Kogan, and J.L. Welch. Efficient and robust local mutual exclusion in mobile ad hoc networks. In IEEE Trans. on Mobile Comp., volume 9, Mar 2010.
- F. Borran, R. Prakash, and A. Schiper. Consensus in wireless ad hoc networks. Technical report, Technical Report LSR-REPORT-2008-001, EPFL, 2008b.
- M. Bouroche. Real-Time Coordination of Mobile Autonomous Entities. PhD thesis, Univ. of Dublin, Trinity College, 2007.
- G. Chockler, M. Demirbas, S. Gilbert, N. Lynch, C. Newport, and T. Nolte. Reconciling the theory and practice of (un)reliable wireless broadcast. In 4th Int'l Workshop on Assurance in Distributed Systems and Networks (ADSN) (ICDCS'05), pages 42–48, Jun 2005.
- W. Emmerich. *Engineering distributed objects*. Wiley, 2000.
- M.J. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of distributed consensus with one faulty process. Journal of the ACM (JACM), 32(2):374–382, 1985. ISSN 0004-5411.
- G. Hackmann, C. Gill, and G.C. Roman. Towards a real-time coordination model for mobile computing. In Proceedings of the 12th Monterey conference on Reliable systems on unreliable networked platforms, pages 184–202. Springer-Verlag, 2005.
- B. Hughes. Hard real-time communication for mobile ad hoc networks. PhD thesis, University of Dublin, Trinity College, 2006.
- R. Meier and V. Cahill. Steam: Event-based middleware for wireless ad hoc networks. In Proc. 22nd Int'l Conf. on Distributed Comp. Systems Workshops, 2002., pages 639–644. IEEE, 2002. ISBN 0769515886.
- G.C. Roman, R. Handorean, and R. Sen. Tuple space coordination across space and time. In Coordination Models and Languages, pages 266–280. Springer, 2006.
- S. Schemmer, E. Nett, and M. Mock. Reliable real-time cooperation of mobile autonomous systems. In Reliable Distributed Systems, 2001. Proceedings. 20th IEEE Symposium on, pages 238–246. IEEE, 2001.
- W. Wu, J. Cao, and J Yang. A fault tolerant mutual exclusion algorithm for mobile ad hoc networks. In Pervasive and Mobile Comp., volume 4, pages 139–160, Feb 2008.
- P. Veríssimo and A. Casimiro. The timely computing base. 1999.
- P.E. Veríssimo. Travelling through wormholes: a new look at distributed systems models. ACM SIGACT News, 37(1):66-81, 2006.