Introduction to Intelligent Vehicles
[6. Intersection Management]

Chung-Wei Lin

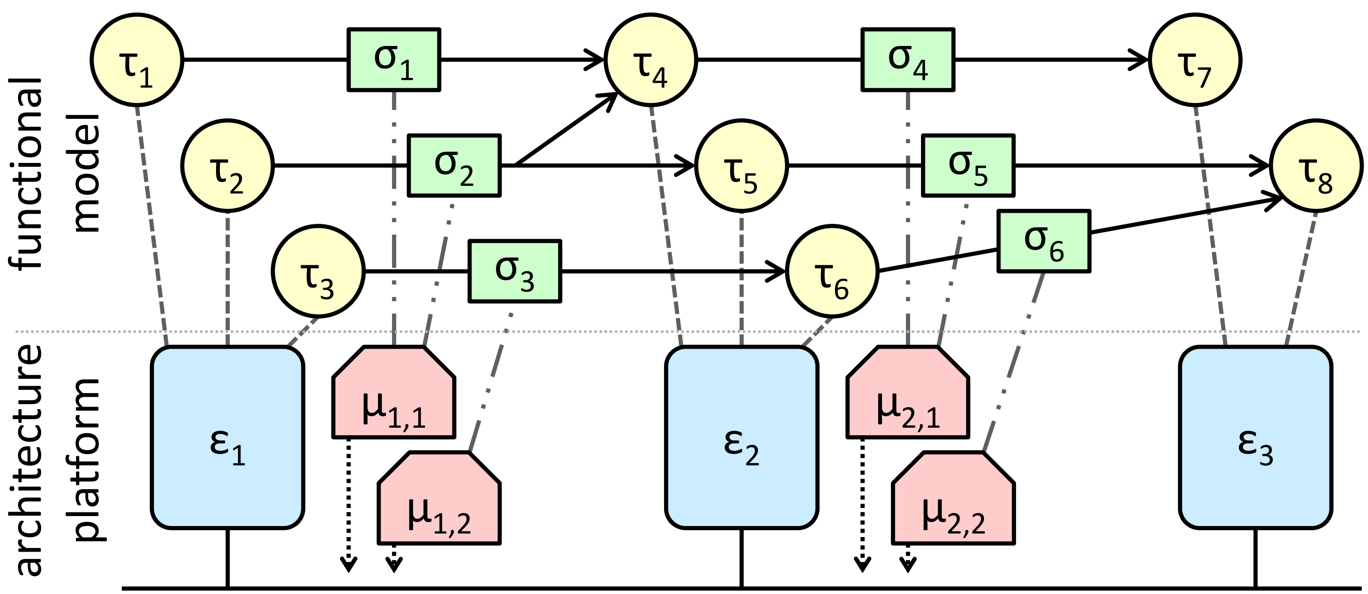
cwlin@csie.ntu.edu.tw

CSIE Department

National Taiwan University

Where Are We Now?

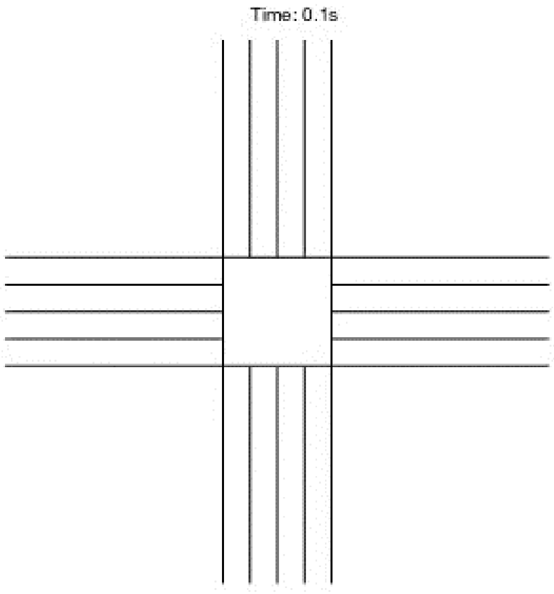
- Four parts in sequence
  - [Part 1] Preliminary
  - [Part 2] Applications
  - [Part 3] Intelligent Technology
  - [Part 4] Advanced Topics



Intersection Management

- What is intersection management?
  - Decide who goes first
- Why is intersection management helpful?
  - Make the intersection safer and traffic smoother and more efficient
- When is intersection management working?
  - Anytime?
- Where is intersection management working?
  - Centralized vs. distributed
- Who develops intersection management?
  - Basic ones from governments (or their suppliers)
  - Advanced ones have not been realized
- How does intersection management work?

Ideal Intersection

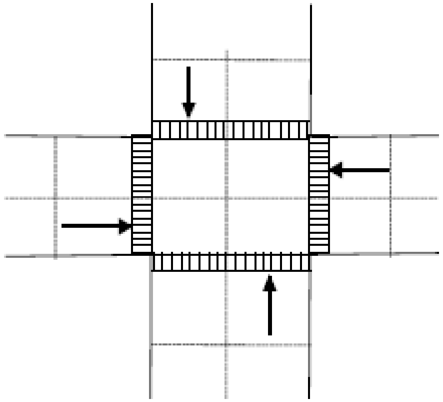


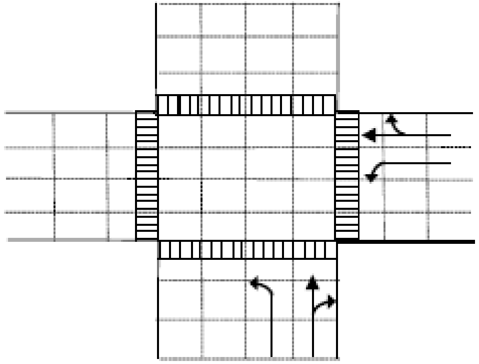
Outline

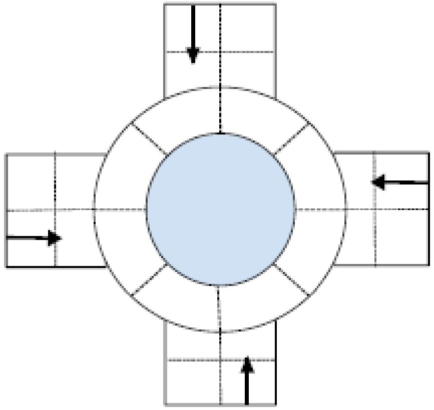
- ❑ **Modeling**
  - ➢ Note: not all following models will be used at the same time
- ❑ Controlling Lengths of Traffic Lights
- ❑ Intelligent Intersection Management
- ❑ Imperfect Communication
- ❑ Centralized and Distributed Approaches
- ❑ Graph-Based Approach
- ❑ Non-Cooperative Environment
- ❑ Special Case: Lane Merging

Tiles (Cells)

- Usually, higher granularity, e.g., smaller tiles, more detailed management and higher complexity
  - Example: an intersection with 20 tiles
  - Example: an intersection with 64 tiles
  - Example: a roundabout with 24 tiles



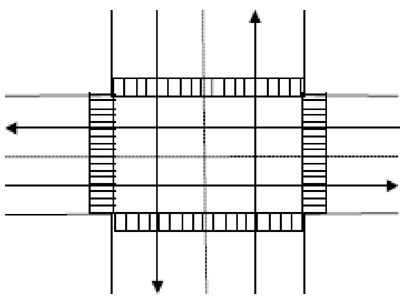


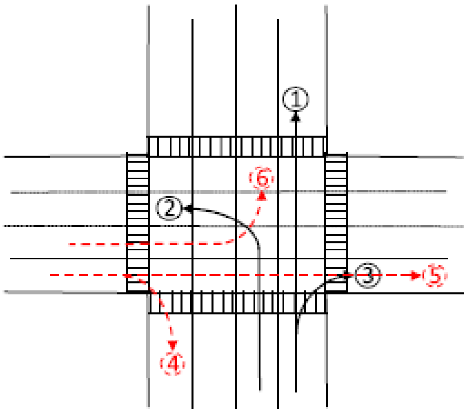


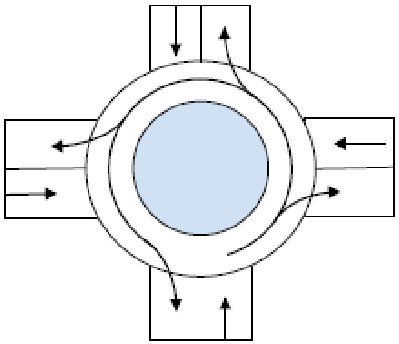
Chen and Englund, "Cooperative intersection management: A survey," IEEE Transactions on Intelligent Transportation Systems, 2016.

Trajectories

- The trajectories for vehicles from different directions with different intentions follows "pre-defined" routes
  - Example: 1 trajectory for each direction of the intersection
  - Example: 3 trajectories for each direction of the intersection
  - Example: 4 trajectories for each direction of the roundabout



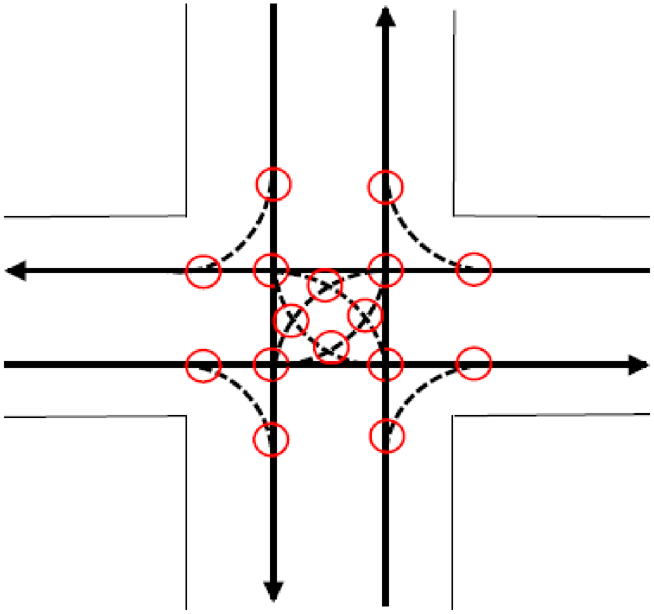




Chen and Englund, "Cooperative intersection management: A survey," IEEE Transactions on Intelligent Transportation Systems, 2016.

Collision Zones (Conflict Regions)

- Intersections of trajectories

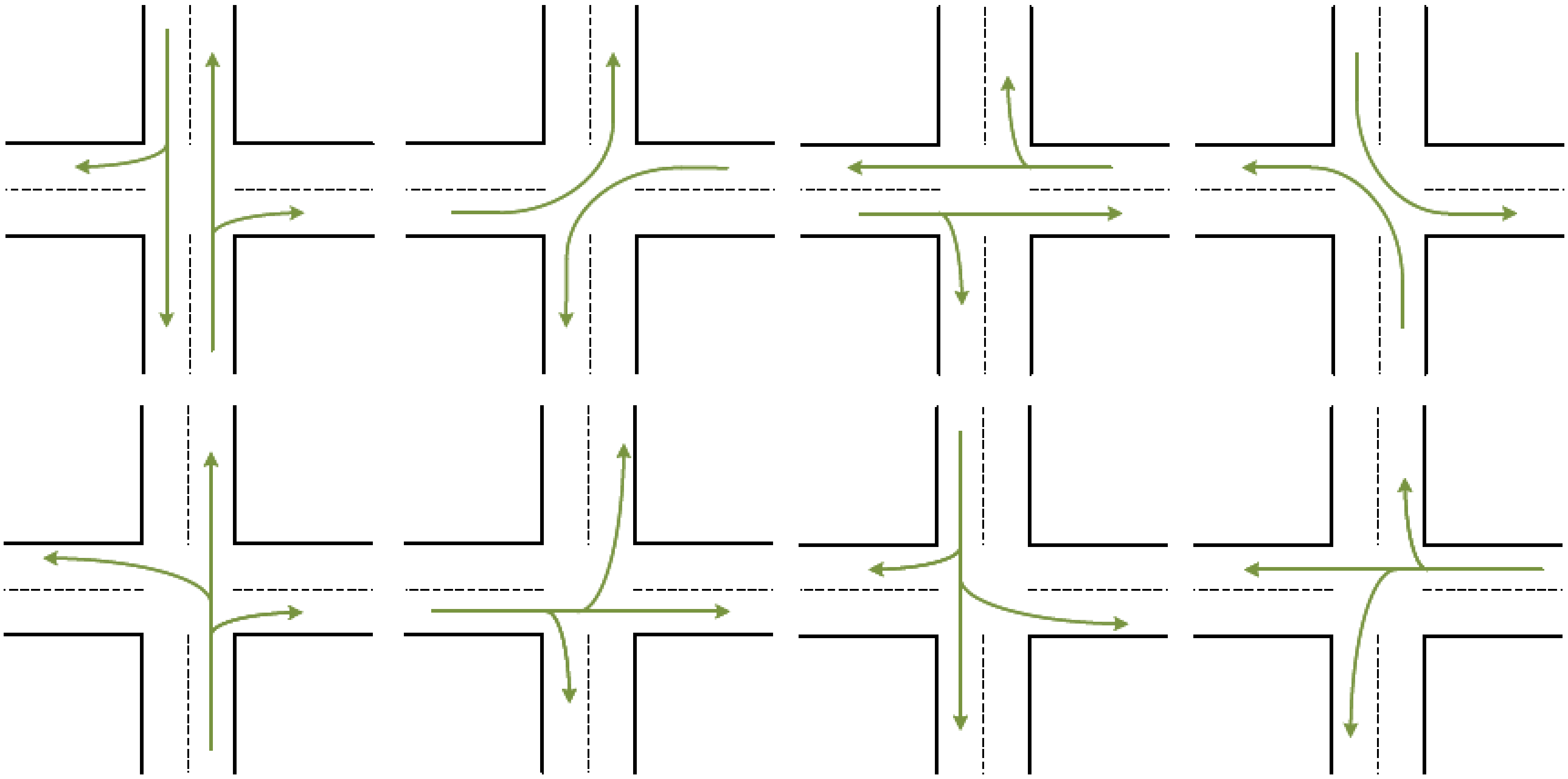


- The fundamental goal
  - We should not let two vehicles occupy a collision zone or a tile at the same time

Chen and Englund, "Cooperative intersection management: A survey," IEEE Transactions on Intelligent Transportation Systems, 2016.

Phases

- Example
  - 8 phases of an intersection



Outline

- ❑ Modeling
- ❑ **Controlling Lengths of Traffic Lights**
- ❑ Intelligent Intersection Management
- ❑ Imperfect Communication
- ❑ Centralized and Distributed Approaches
- ❑ Graph-Based Approach
- ❑ Non-Cooperative Environment
- ❑ Special Case: Lane Merging

Controlling Lengths of Traffic Lights

- Fixed time control
- Coordinated control
  - Continuous green lights
- Adaptive control
  - Design-time approach
    - Based on traffic data and history
  - Real-time approach
    - Based on sensor observation or communication

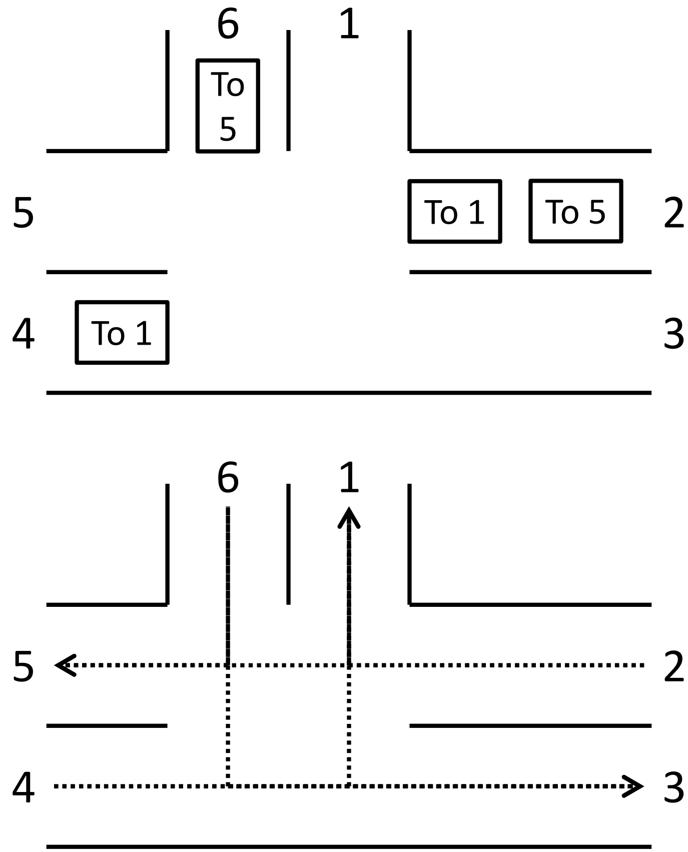
Back-Pressure Control: Method

- Traffic flow is similar to water flow
- Basic notation
  - λ_i: lane i
  - Q_i: queue length of lane i
  - P_i: pressure of lane i
    - $P_i = Q_i$
  - $D_{i,j}$: [0,1] there is a vehicle waiting at lane i to leave from lane i for lane j
  - $P_{i,j}$: pressure from lane i to lane j
    - $P_{i,j} = D_{i,j} \max(P_i - P_j, 0)$
- Each time slot, pick a phase which can maximize $\sum_{(i,j)} V_{i,j} P_{i,j}$
  - $V_{i,j}$: the maximum number of vehicles that can go from lane i to lane j in the phase during the time slot

Back-Pressure Control: Example

Computation

- $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$
- Q = {0, 2, 0, 1, 0, 1}
- P = {0, 2, 0, 1, 0, 1}
- $D_{2,1} = D_{2,5} = D_{4,1} = D_{6,5} = 1$
- $P_{2,1} = P_{2,5} = 2$ and $P_{4,1} = P_{6,5} = 1$
- Assume $V_{i,j}$ is V or 0
  - Why 0?
- (Check animations for phases)
- $\sum_{(i,j)} V_{i,j}\, P_{i,j} = 4V$ for Phase 1
- $\sum_{(i,j)} V_{i,j}\, P_{i,j} = 2V$ for Phase 2
- $\sum_{(i,j)} V_{i,j}\, P_{i,j} = 3V$ for Phase 3
- Pick Phase 1



Back-Pressure Control: Extensions

- Capacity-aware back-pressure control
  - Remove the assumption of infinite capacity
  - Improve the fairness (for low density traffic)
  - Redefine P_i as another more complicated function
- Adaptive max-pressure control
  - Model the network demand with a constant average rate
  - Provide some stability and performance guarantees
  - Redefine P_i as another more complicated function

Controlling Lengths of Traffic Lights

- If the lengths of green, yellow, and red lights can be very short

Outline

- ❑ Modeling
- ❑ Controlling Lengths of Traffic Lights
- ❑ **Intelligent Intersection Management**
- ❑ Imperfect Communication
- ❑ Centralized and Distributed Approaches
- ❑ Graph-Based Approach
- ❑ Non-Cooperative Environment
- ❑ Special Case: Lane Merging

Assumptions

- All vehicles are connected and autonomous
  - If not connected
    - Need road-side units to collect traffic information, e.g., vehicles coming
    - Need traffic lights to provide instructions
  - If not autonomous
    - Need to consider the control capability of human drivers

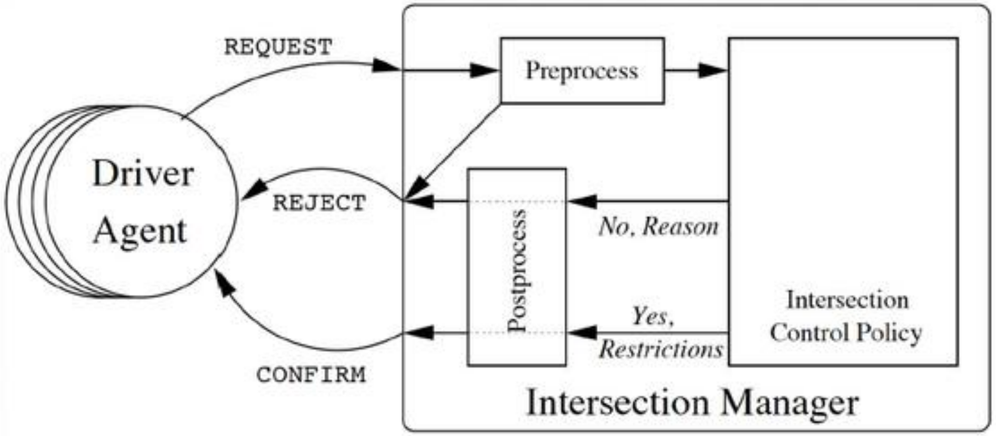
Goals

- Safety
- Traffic efficiency
- Deadlock and starvation avoidance
- Low communication and computation complexity
- Incremental deployability
- Protocol standardization

Dresner and Stone, "A multiagent approach to autonomous intersection management," Journal of Artificial Intelligence Research, 2008.

Intelligent Intersection Management

- ❑ **Vehicle**
  - ➢ Send a request to the intersection manager about its intention
  - ➢ Do not enter the intersection before confirmation
- ❑ **Intersection manager**
  - ➢ Resolve conflicts through scheduling policies
  - ➢ Allocate tiles (cells) to vehicles for every time step
  - ➢ Send confirmations or rejections to vehicles



Dresner and Stone, "A multiagent approach to autonomous intersection management," Journal of Artificial Intelligence Research, 2008.

Vehicle Behavior

- Message types
  - REQUEST to make a reservation
  - CHANGE-REQUEST to change a reservation
    - REQUEST and CHANGE-REQUEST include all the relevant properties of the vehicle
  - CANCEL to cancel an existing reservation
  - DONE after crossing the intersection
    - CANCEL and DONE include the IDs of the vehicle and the reservation
- Not enter the intersection if there is no confirmation from the manager

Dresner and Stone, "A multiagent approach to autonomous intersection management," Journal of Artificial Intelligence Research, 2008.

Manager Behavior

- Message types
  - CONFIRM after approving a REQUEST or CHANGE-REQUEST
    - ID of the reservation
    - Start time
    - Start and departure lanes
    - A list of constraints for the vehicle's acceleration in the intersection
  - REJECT to reject a REQUEST or CHANGE-REQUEST
  - ACKNOWLEDGE to respond a CANCEL or DONE
  - EMERGENCY-STOP when detecting a major problem
- Control policies
  - First come (definition?), first served
  - "Virtual" stop sign
  - "Virtual" traffic light

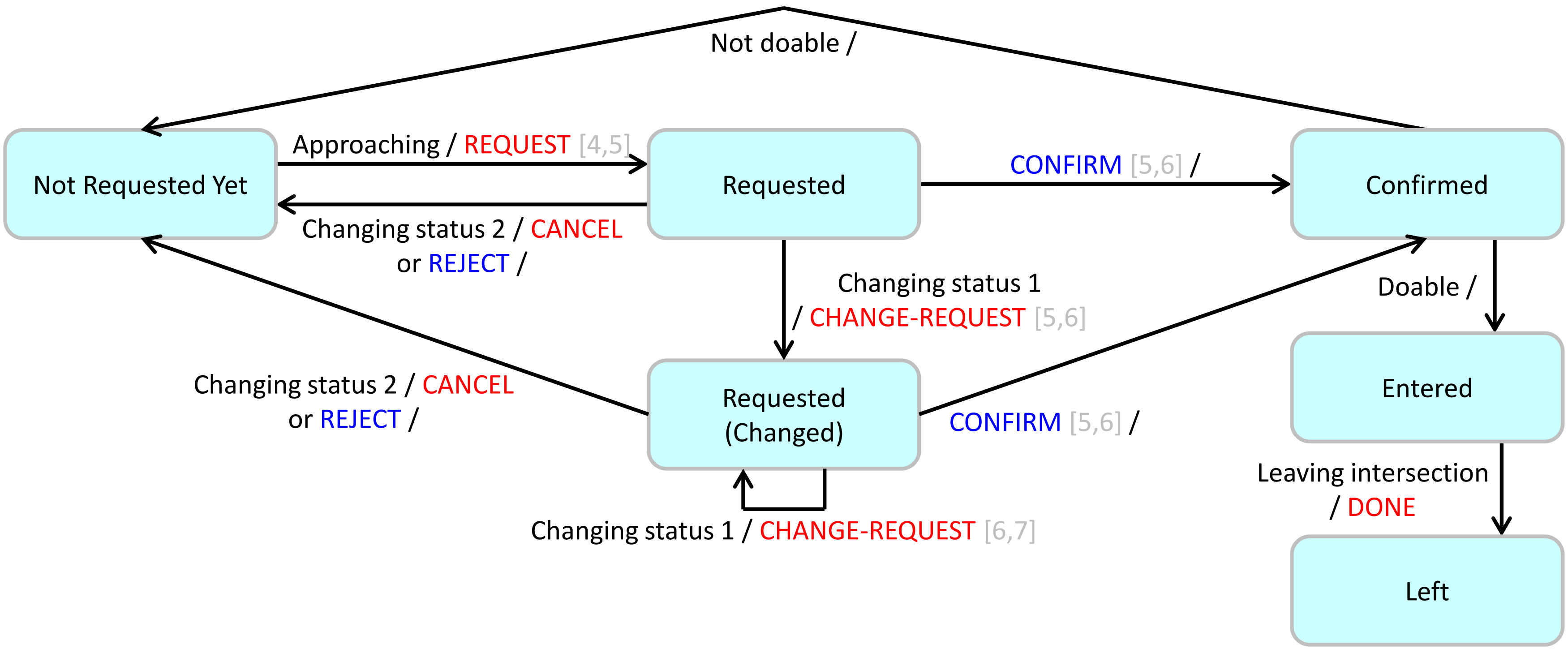
Dresner and Stone, "A multiagent approach to autonomous intersection management," Journal of Artificial Intelligence Research, 2008.

Outline

- ❑ Modeling
- ❑ Controlling Lengths of Traffic Lights
- ❑ Intelligent Intersection Management
- ❑ **<u>Imperfect Communication</u>**
- ❑ Centralized and Distributed Approaches
- ❑ Graph-Based Approach
- ❑ Non-Cooperative Environment
- ❑ Special Case: Lane Merging

Modeling Vehicle Behavior

- [4s,5s]: example time of feasible "entering" the intersection
  - This is a logical view --- the number can be estimated by the manager
- Reasons of slowing down
  - Sense the intersection in front and not receive a confirmation
  - Sense other vehicles in front



Imperfect Communication: Loss

- Informal analysis
  - Case 1/2: What if a REQUEST/CHANGE-REQUEST is lost?
  - Case 3: What if a CANCEL is lost?
  - Case 4: What if a DONE is lost?
  - Case 5: What if a CONFIRM is lost?
  - Case 6: What if a REJECT is lost?
  - Case 7: What if an ACKNOWLEDGE is lost?
  - Case 8: What if an EMERGENCY-STOP is lost?
- Ideally, we should have "formal analysis"
- Having "timeouts" is important
  - Re-request a reservation (Cases 1, 2, 5, 6)
  - Logically remove a vehicle after physically checking the intersection? (Case 4)

Imperfect Communication: Delay

- Informal analysis
  - Case 1/2: What if a REQUEST/CHANGE-REQUEST is delayed?
  - Case 3: What if a CANCEL is delayed?
  - Case 4: What if a DONE is delayed?
  - Case 5: What if a CONFIRM is delayed?
  - Case 6: What if a REJECT is delayed?
  - Case 7: What if an ACKNOWLEDGE is delayed?
  - Case 8: What if an EMERGENCY-STOP is delayed?
- Combinations of message loss and delay
  - How do you know which vehicle is in front?
- Having "time stamps" can help
  - Synchronization protocol?

Imperfect Communication: More

- Schedule only when 1st vehicle requests (otherwise?)
  - We assume that each vehicle knows if it is 1st vehicle and provides this information to the manager

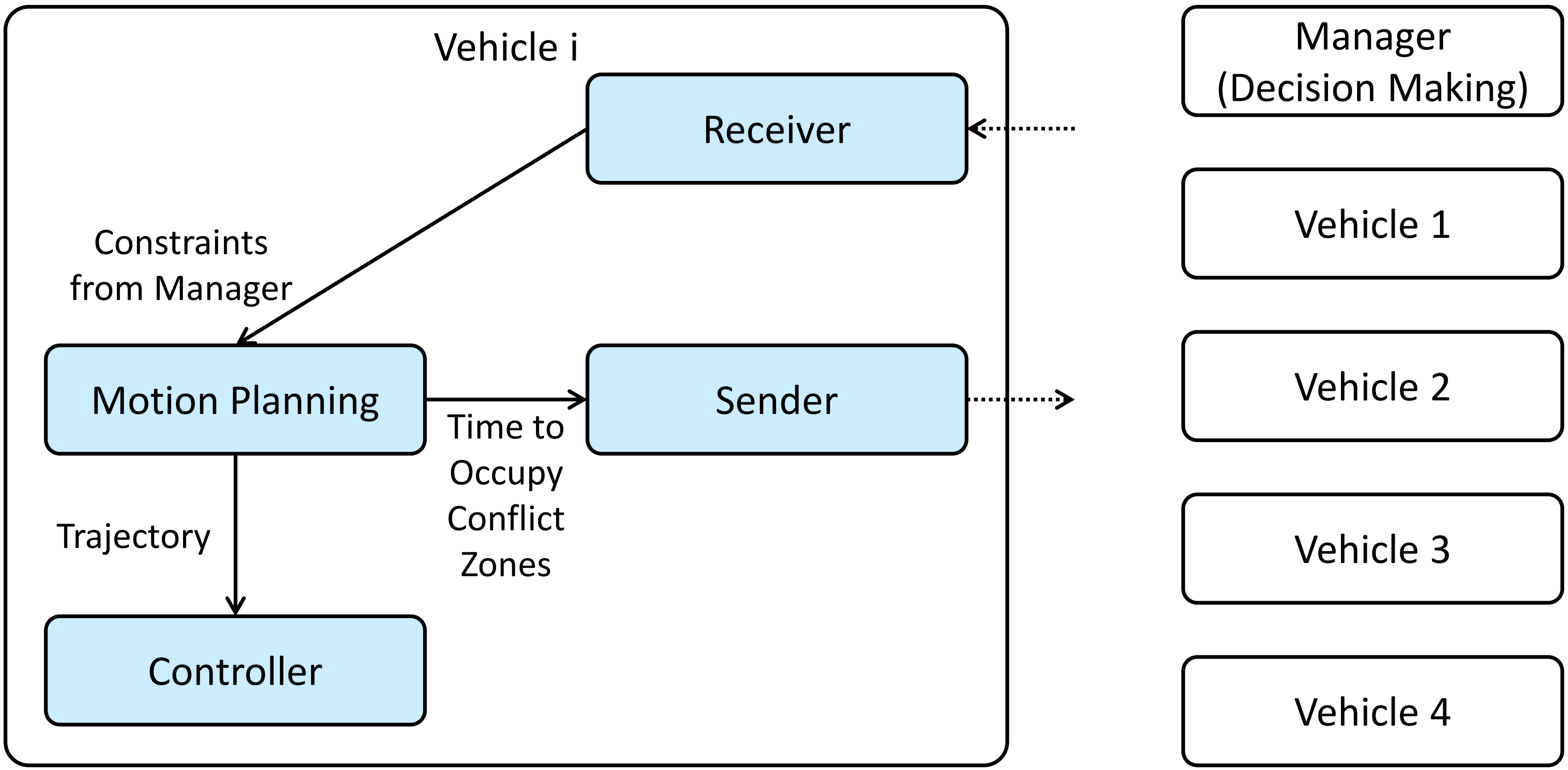
X		Missing X-Request	Missing X-Confirmation
1st Vehicle	Manager	Schedule nothing (1st vehicle missing)	Schedule all vehicles
	1st Vehicle	Slow down, timeout, re-request	Slow down, timeout, re-request
	2nd Vehicle	Slow down, timeout, re-request	Slow down, cannot make it, re-request
	3rd Vehicle	Slow down, timeout, re-request	Slow down, cannot make it, re-request
2nd Vehicle	Manager	Schedule 1st and 3rd vehicles	Schedule all vehicles
	1st Vehicle	Normal	Normal
	2nd Vehicle	Slow down, timeout, re-request	Slow down, timeout, re-request
	3rd Vehicle	Slow down, cannot make it, re-request	Slow down, cannot make it, re-request
3rd Vehicle	Manager	Schedule 1st and 2nd vehicles	Schedule all vehicles
	1st Vehicle	Normal	Normal
	2nd Vehicle	Normal	Normal
	3rd Vehicle	Slow down, timeout, re-request	Slow down, timeout, re-request

Outline

- ❑ Modeling
- ❑ Controlling Lengths of Traffic Lights
- ❑ Intelligent Intersection Management
- ❑ Imperfect Communication
- ❑ **Centralized and Distributed Approaches**
- ❑ Graph-Based Approach
- ❑ Non-Cooperative Environment
- ❑ Special Case: Lane Merging

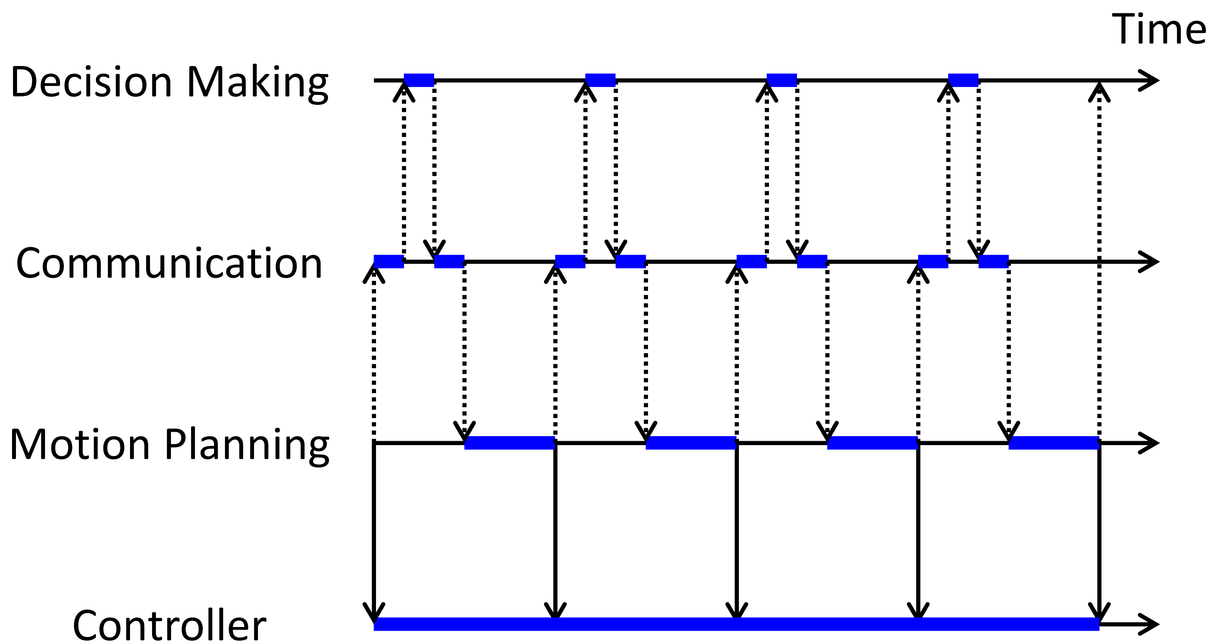
System Block Diagram: Centralized

- The manager is the decision maker



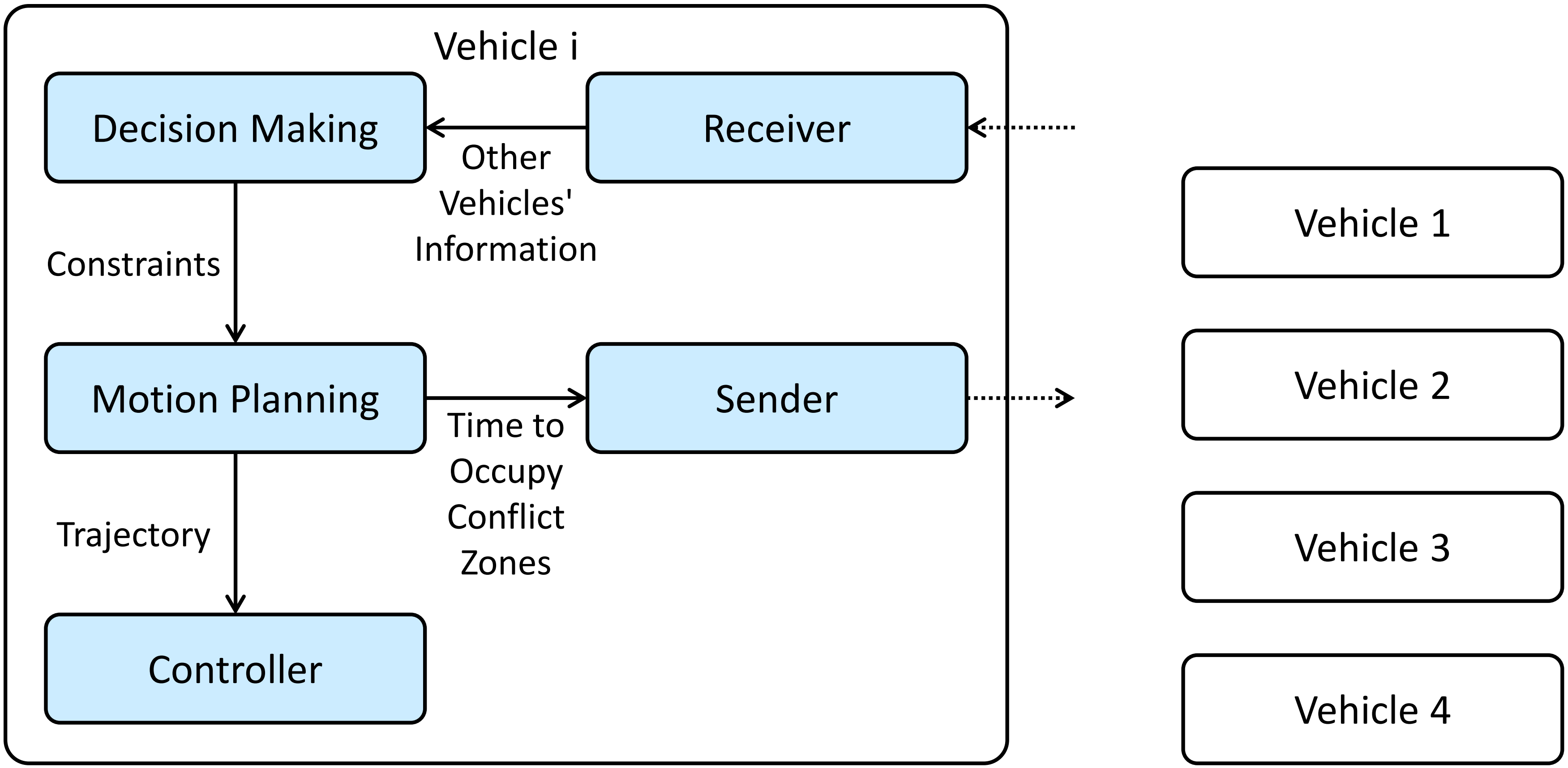
Flow of Execution: Centralized

- ❑ Each task should be completed in time



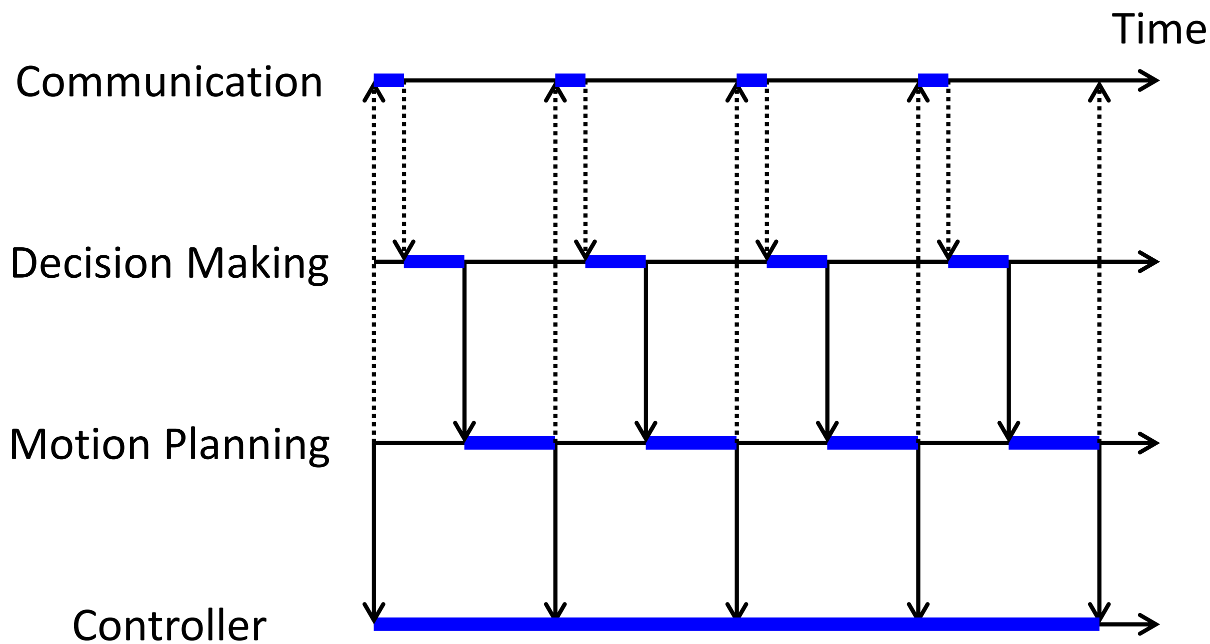
System Block Diagram: Distributed

- Each vehicle is a decision maker



Flow of Execution: Distributed

- Each task should be completed in time



Outline

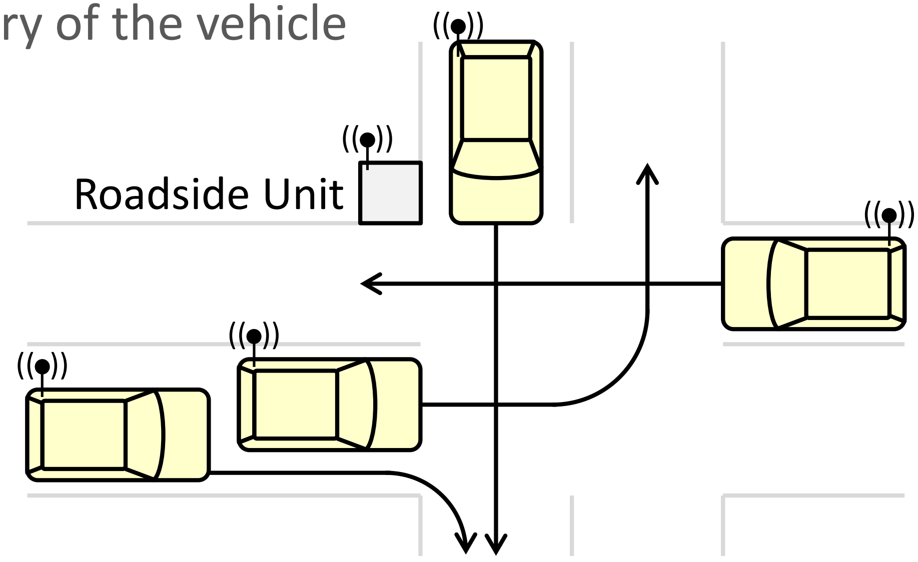
- Modeling
- Controlling Lengths of Traffic Lights
- Intelligent Intersection Management
- Imperfect Communication
- Centralized and Distributed Approaches
- **Graph-Based Approach**
  - **Modeling**
  - Design: Scheduling (Not Covered)
  - Analysis: Verification
- Non-Cooperative Environment
- Special Case: Lane Merging

Graph-Based Policy

- The decision making problem in intersection management can be transformed to a cycle-removal problem in a graph
- Generalization
  - Conflict-resolution problem
    - The goal: we should not let two vehicles occupy a conflict zone (or a tile) at the same time
  - Distributed intersection management
    - Does it make sense to have no manager?
    - If vehicles (even from different OEMs) agree how to remove a cycle, then "conflict-free" can be guaranteed
    - Cycle-removal algorithm is the "agreement" between vehicles (e.g., four-way stop sign) in a distributed setting
      - We have an existing system in …

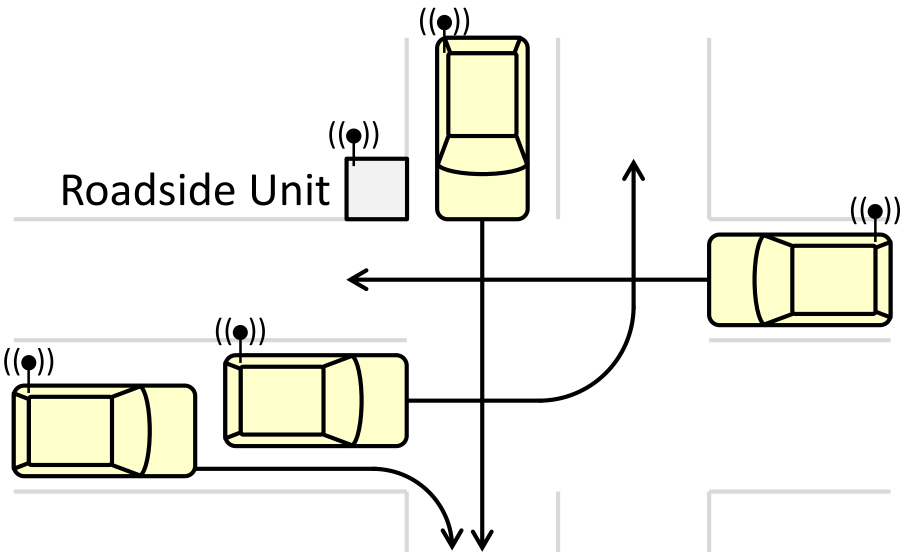
Intersection

- **Intersection**
  - There is one intersection
- **Intersection manager**
  - Receive the information from vehicles
  - Assign a time window to each vehicle at each location (conflict zone) on the trajectory of the vehicle



Vehicle

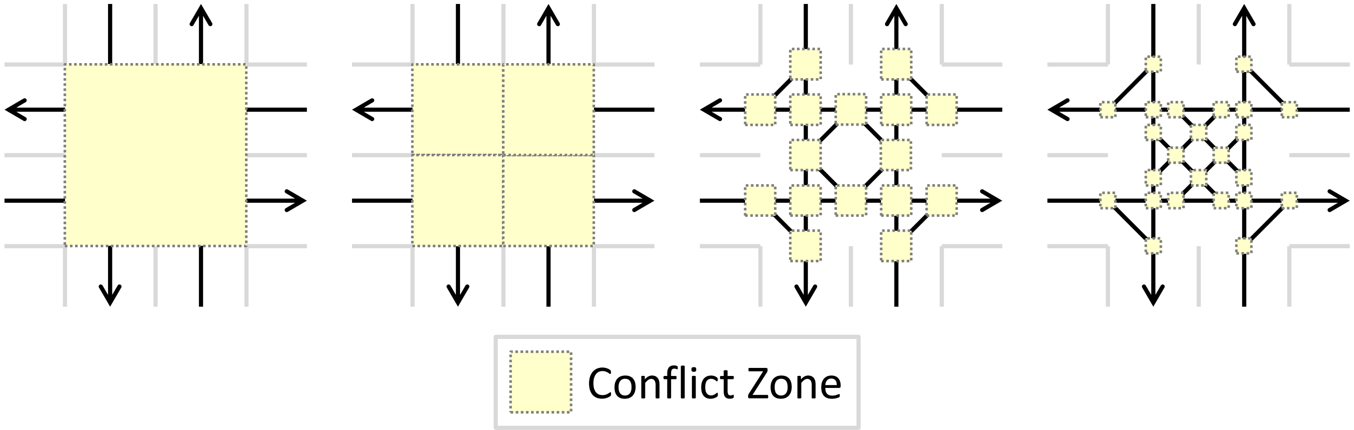
- Vehicle
  - All vehicles are connected and autonomous
  - Each vehicle has a fixed trajectory
  - Vehicle does not change lanes before and after the intersection



Conflict Zone

❑ Conflict zone

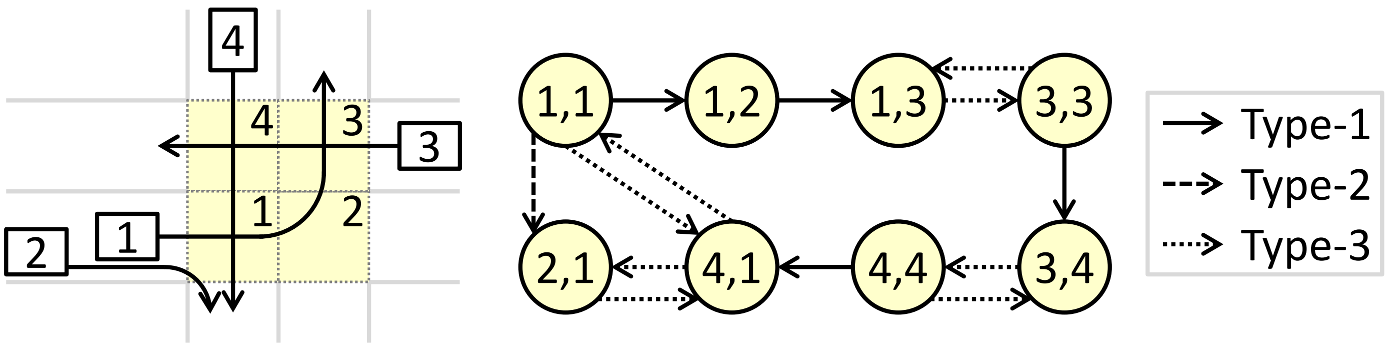
- A conflict zone is the crossing location of two trajectories
- Two vehicles cannot be at (occupy) the same conflict zone at the same time



Timing Conflict Graph

❑ Timing conflict graph

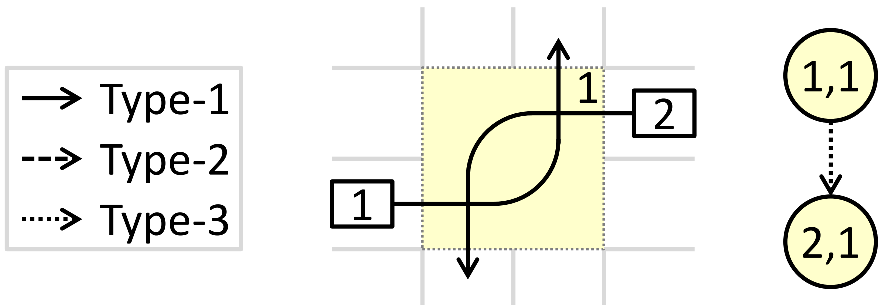
- A directed timing conflict graph G = (V, E) is constructed
- Vertex set $v_{i,j}$ is a subset of the Cartesian product of the sets of vehicles and conflict zones



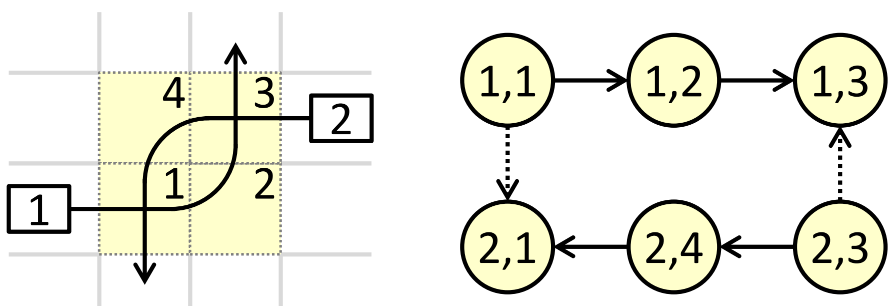
- Edge
  - Type-1: same vehicle's trajectory
  - Type-2: conflicts between different vehicles from the same lane
  - Type-3: conflicts between different vehicles from different lanes

Model Expressiveness

- If the intersection is modeled by only one conflict zone
  - Its expressiveness is limited, and the two vehicles cannot enter the intersection at the same time

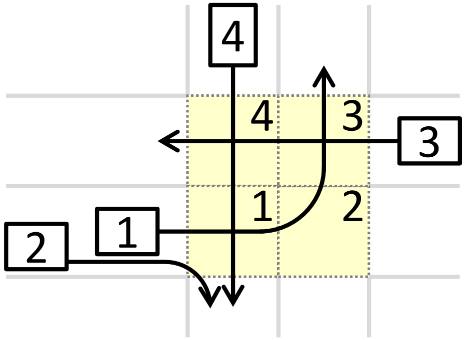


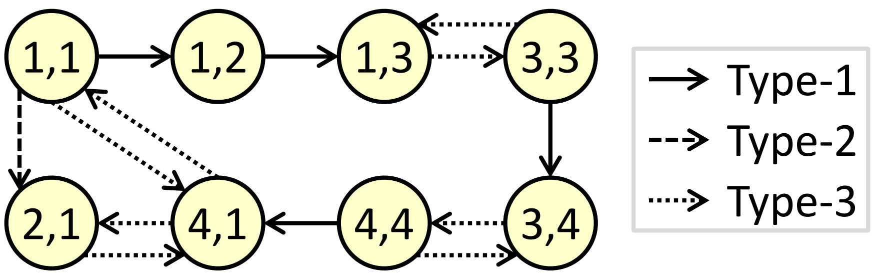
- If the intersection is modeled by four conflict zones
  - Two vehicles can enter the intersection at the same time



Goals

- A cycle-removal (on Type-3 edges) algorithm to
  - Minimize
    - The passing time of the last vehicle, or
    - The average delay of vehicles
  - Guarantee collision-freeness
    - Provided by the passing order and scheduling after removing cycles
  - Guarantee deadlock-freeness
    - Graph-based verification
    - Petri-Net-based verification



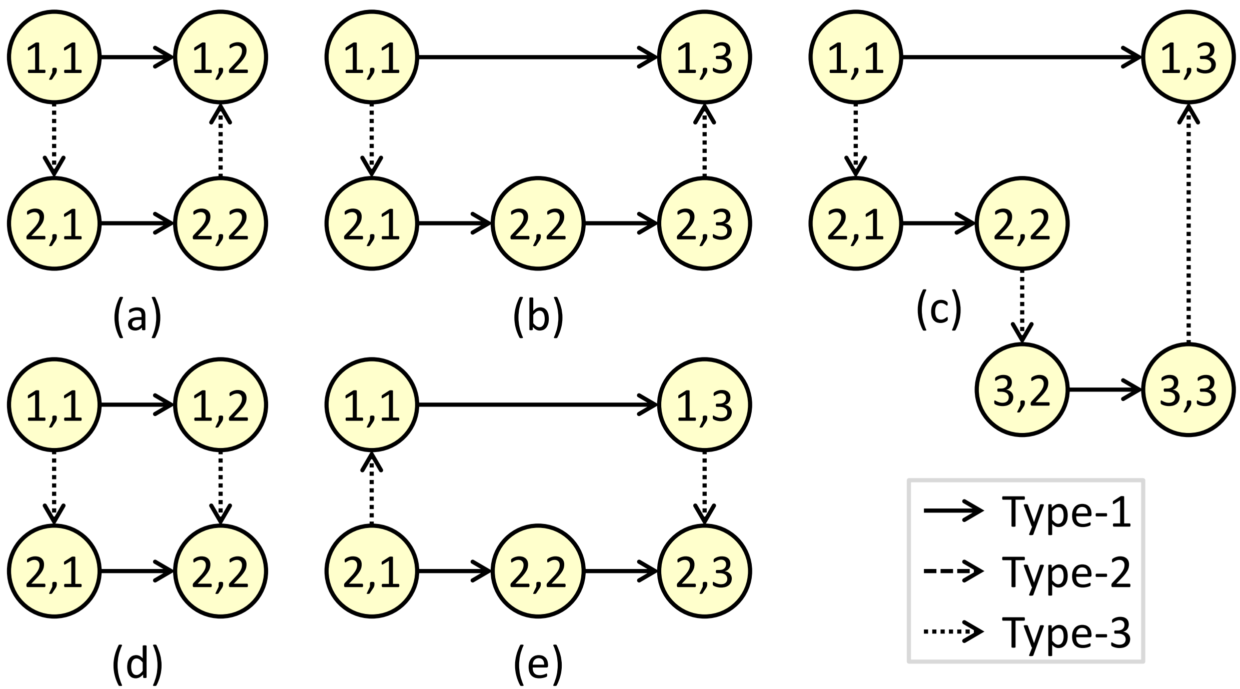


Outline

- ❑ Modeling
- ❑ Controlling Lengths of Traffic Lights
- ❑ Intelligent Intersection Management
- ❑ Imperfect Communication
- ❑ Centralized and Distributed Approaches
- ❑ **Graph-Based Approach**
  - ➢ Modeling
  - ➢ Design: Scheduling (Not Covered)
  - ➢ **Analysis: Verification**
- ❑ Non-Cooperative Environment
- ❑ Special Case: Lane Merging

Graph-Based Verification

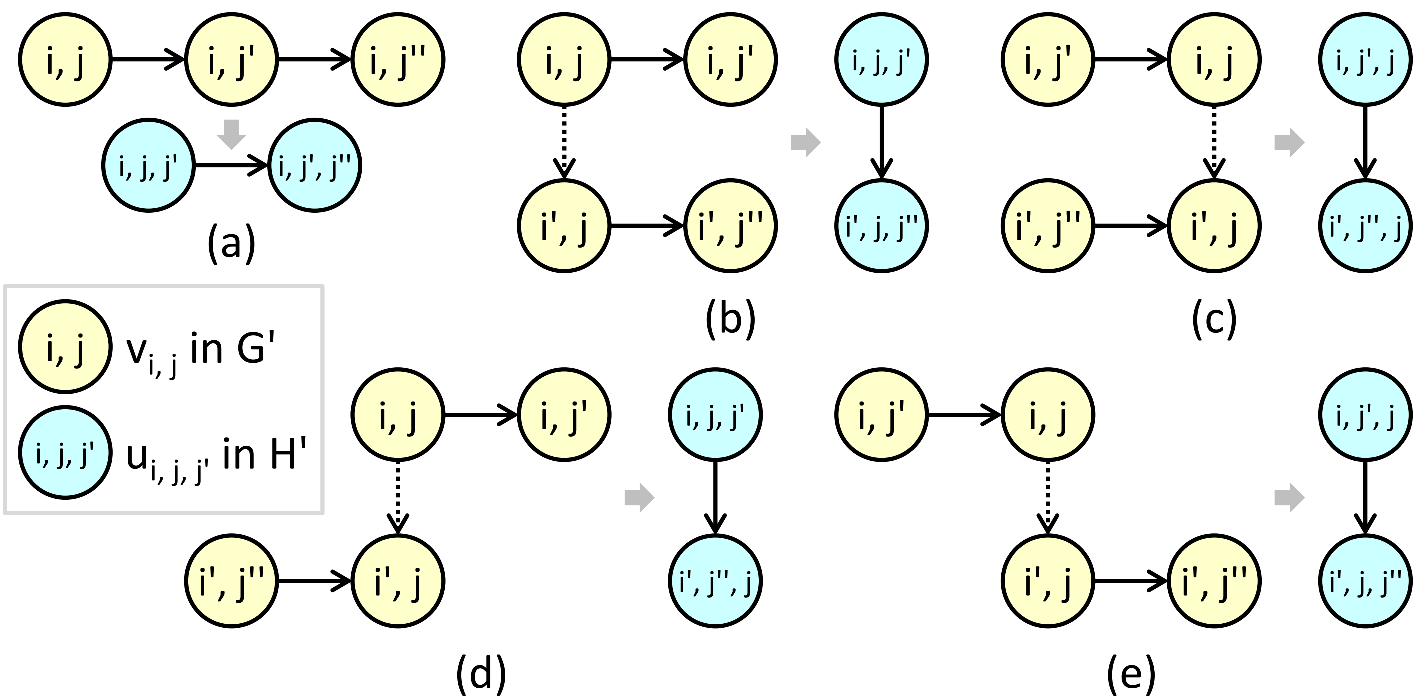
- Having no cycles in G does not guarantee deadlock-freeness
  - There are deadlocks in (a), (b), (c)
  - There are no deadlocks in (d), (e)



Graph-Based Verification

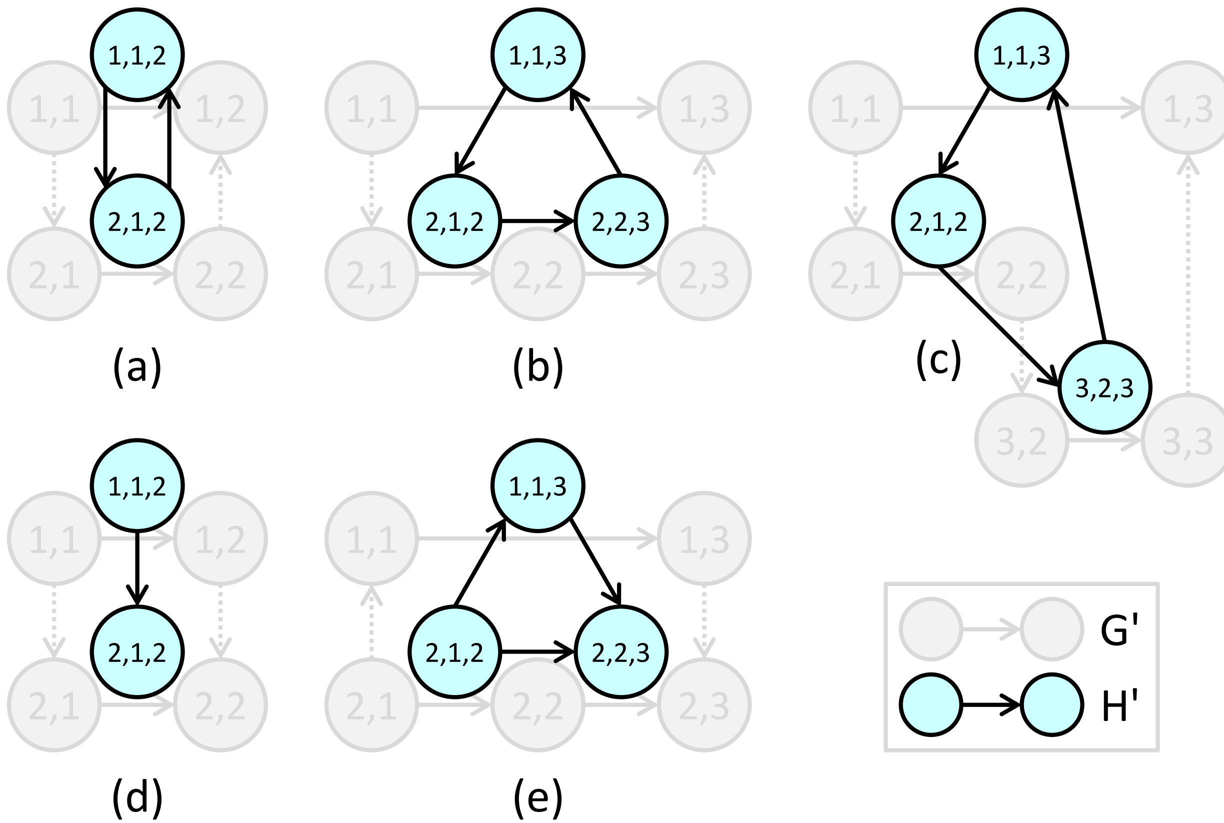
❑ Construction of resource conflict graph

➢ The directed resource conflict graph is constructed by following rules



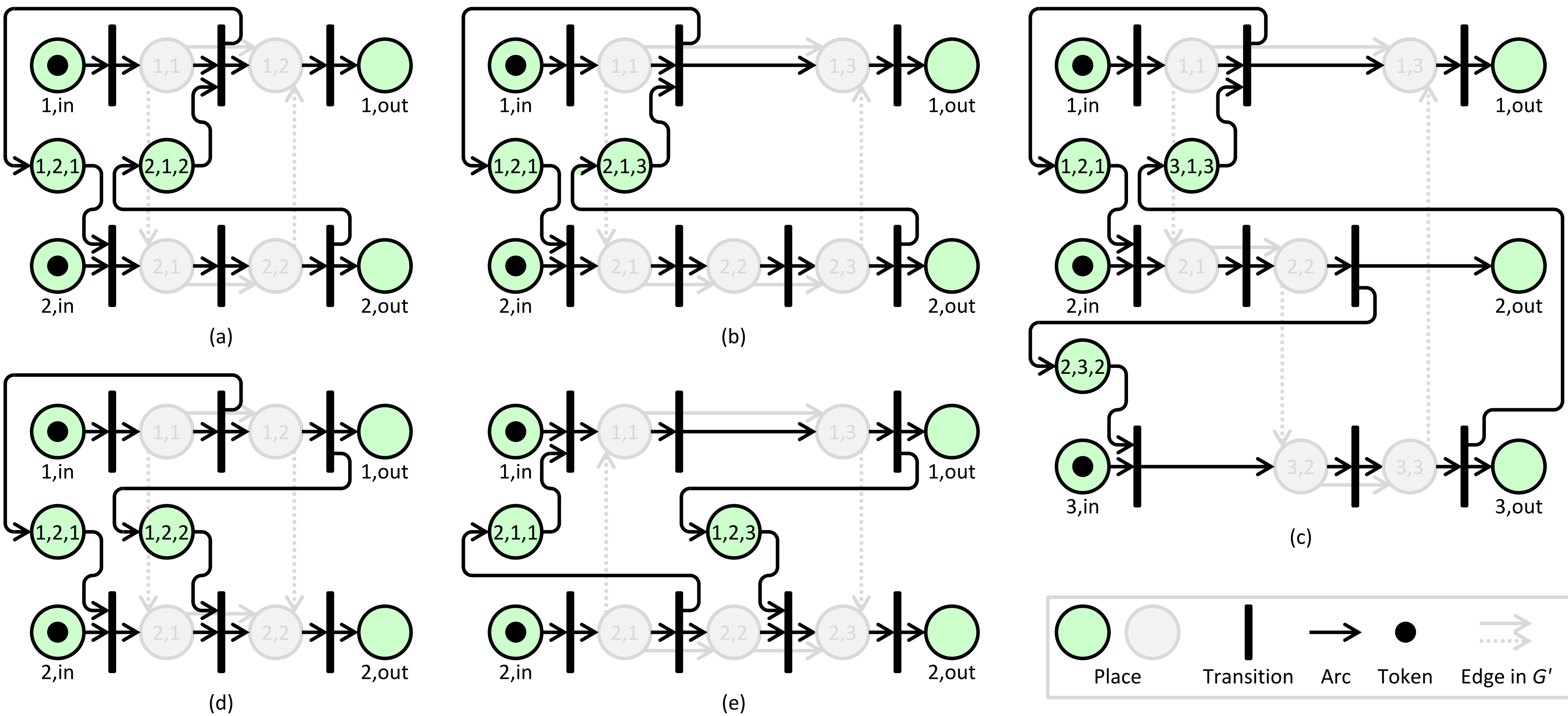
Graph-Based Verification

- Examples of resource conflict graph



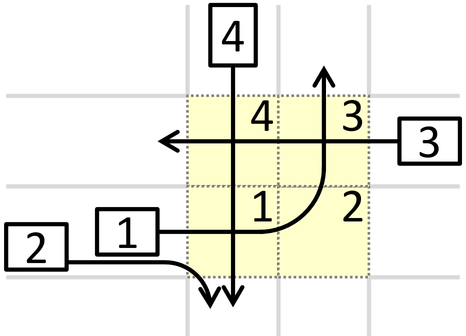
Petri-Net-Based Verification

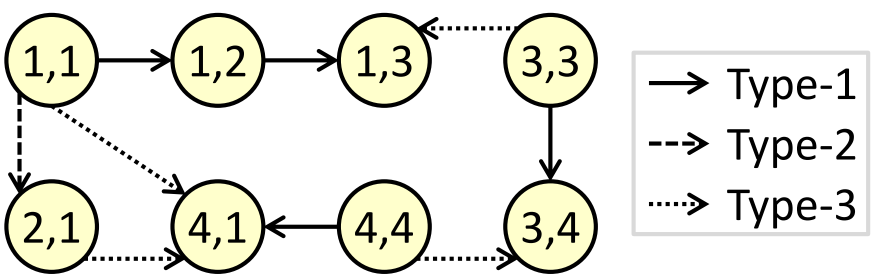
❑ Petri Net Examples



Summary

- A cycle-removal (on Type-3 edges) algorithm (not covered) to
  - Minimize
    - The passing time of the last vehicle, or
    - The average delay of vehicles
  - Guarantee collision-freeness
    - Provided by the passing order and scheduling after removing cycles
  - Guarantee deadlock-freeness
    - Graph-based verification
    - Petri-Net-based verification



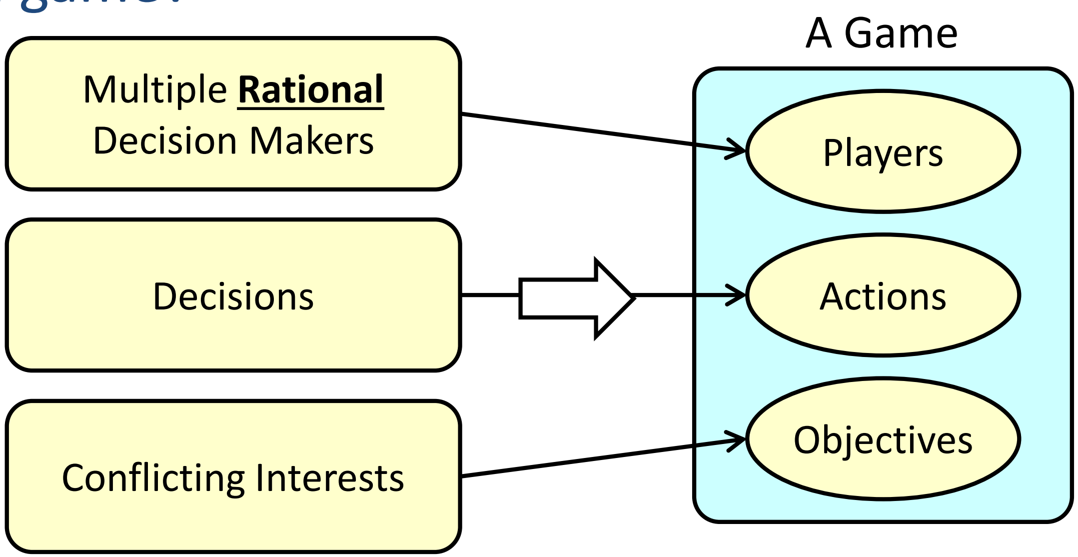


Outline

- ❑ Modeling
- ❑ Controlling Lengths of Traffic Lights
- ❑ Intelligent Intersection Management
- ❑ Imperfect Communication
- ❑ Centralized and Distributed Approaches
- ❑ Graph-Based Approach
- ❑ **Non-Cooperative Environment**
- ❑ Special Case: Lane Merging

Game Theory

❑ What is a game?



❑ Prisoner's dilemma

	Prisoner A Stays Silent	Prisoner A Betrays
Prisoner B Stays Silent	Both in Jail for 1 Year	A Goes Free B in Jail for 3 Years
Prisoner B Betrays	A in Jail 3 Years B Goes Free	Both in Jail for 2 Years (**Nash Equilibrium**)

Two-Vehicle Scenario (w/o Manager)

❑ Payoff matrix (waiting time) of vehicles A and B

- Case 1 (same priority)

Payoffs of A and B	Vehicle A Enters	Vehicle A Stops
Vehicle B Enters	(-3600, -3600)	(-4, 0)
Vehicle B Stops	(0, -4)	(-3, -3) (Average of -1 and -5)

- Case 2 (Vehicle A has a higher priority)

Payoffs of A and B	Vehicle A Enters	Vehicle A Stops
Vehicle B Enters	(-1800, -7200)	(-4, 0)
Vehicle B Stops	(0, -4)	(-1, -5)

- Where are the Nash Equilibria?
  - Are them realistic? → People are not rational

Three-Vehicle Scenario (with Manager)

❑ Three-vehicle strategic game

➢ Assume that the time needed to go through an intersection is 7

No Vehicle lies				
Vehicle	Actual Time	Reported Time	Allocated Time	Delay
A	5	5	5	0
B	10	10	12	2
C	12	12	19	7
System Performance				9

Vehicle C Lies				
Vehicle	Actual Time	Reported Time	Allocated Time	Delay
A	5	5	5	0
B	10	10	19	9
C	12 ⟷	**6**	12	0
System Performance				9

➢ Vehicle C does not worsen the overall system performance

- Total delays = 9

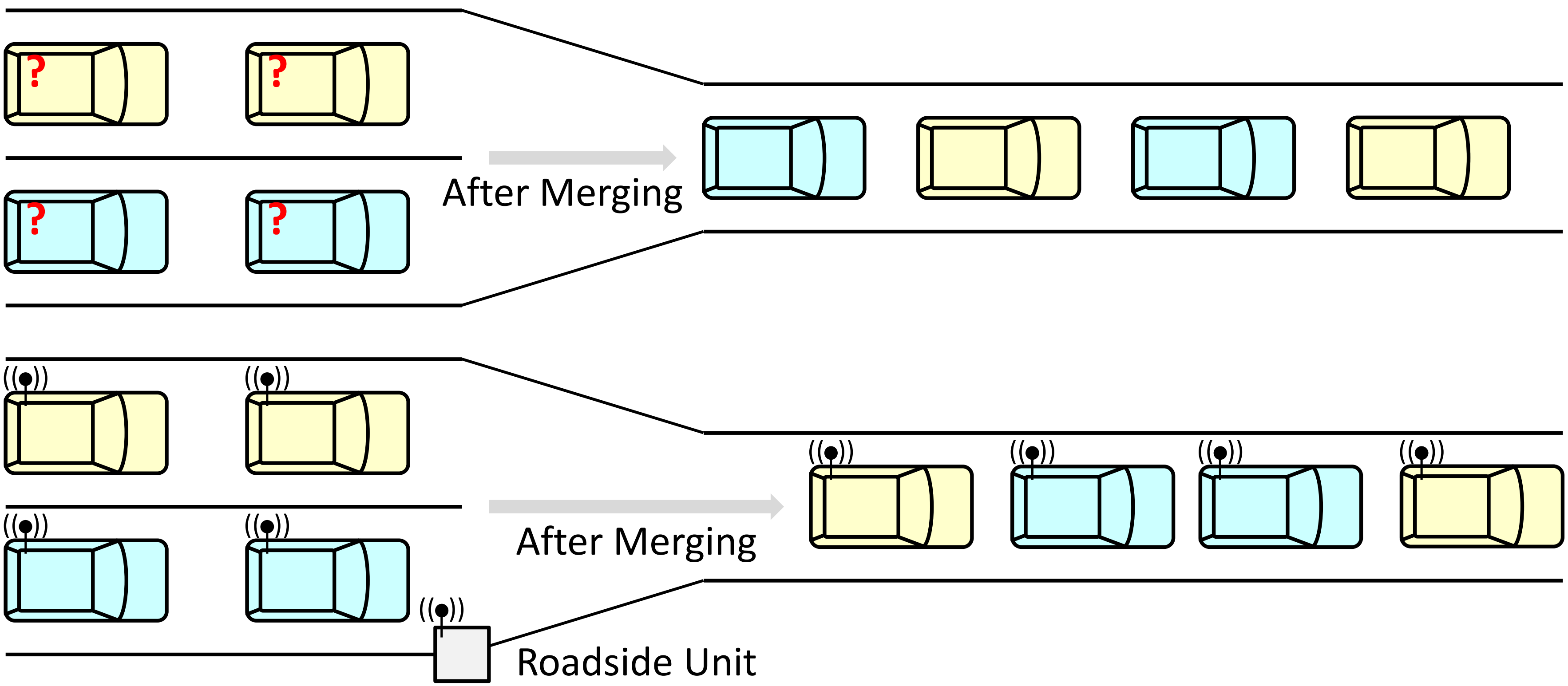
➢ However, Vehicle C can take advantage from it

Outline

- Modeling
- Controlling Lengths of Traffic Lights
- Intelligent Intersection Management
- Imperfect Communication
- Centralized and Distributed Approaches
- Graph-Based Approach
- Non-Cooperative Environment
- **Special Case: Lane Merging**

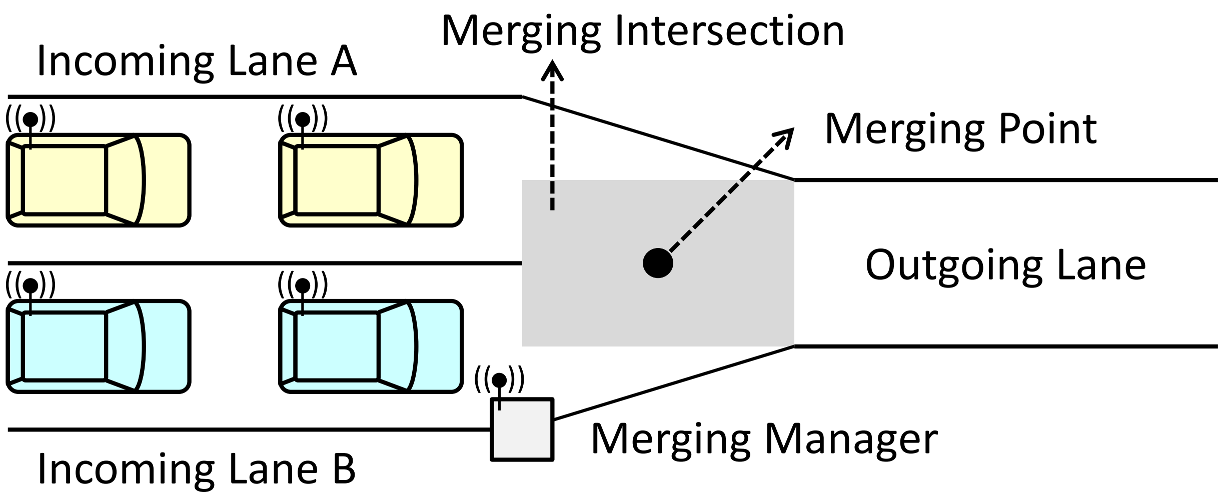
Lane Merging

- Vehicles on the same lane can utilize Cooperative Adaptive Cruise Control (CACC) to form a platoon
- Interleaving vehicles from different lanes increase the vehicle-following gaps (without immediate CACC)



Two-Lane Model

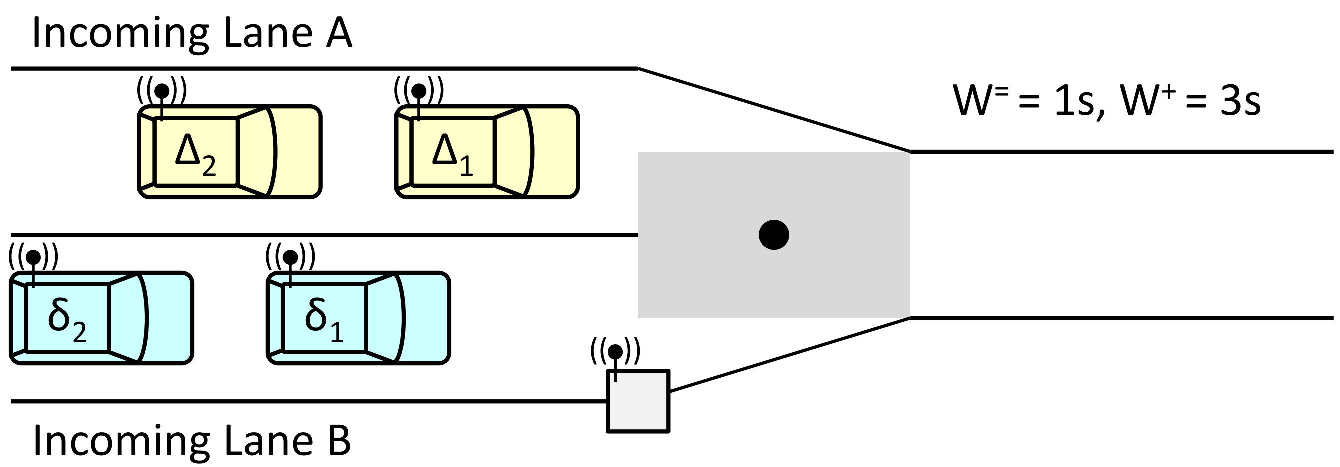
- A fundamental assumption is that vehicles from the same lane can maintain a smaller vehicle-following gaps



- $W^{=}$: the vehicle-following gap between two vehicles from the same lane
- W^{+}: the vehicle-following gap between two vehicles from different lanes

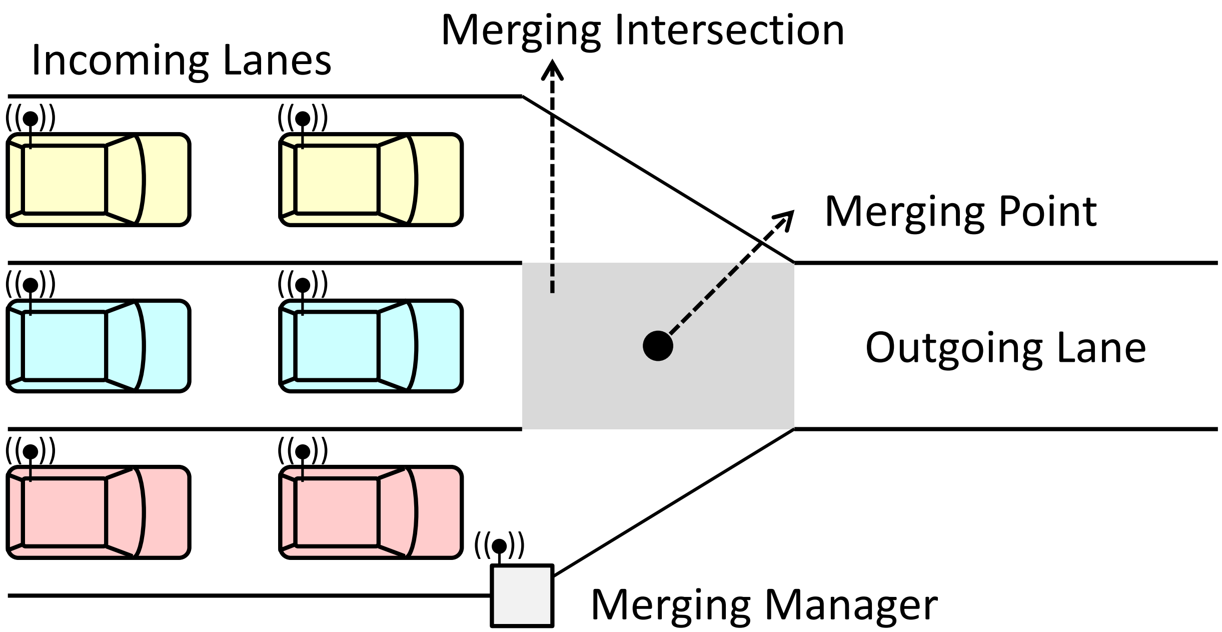
First-Come-First-Go (FCFG)

- The FCFG approach does not guarantee an optimal solution

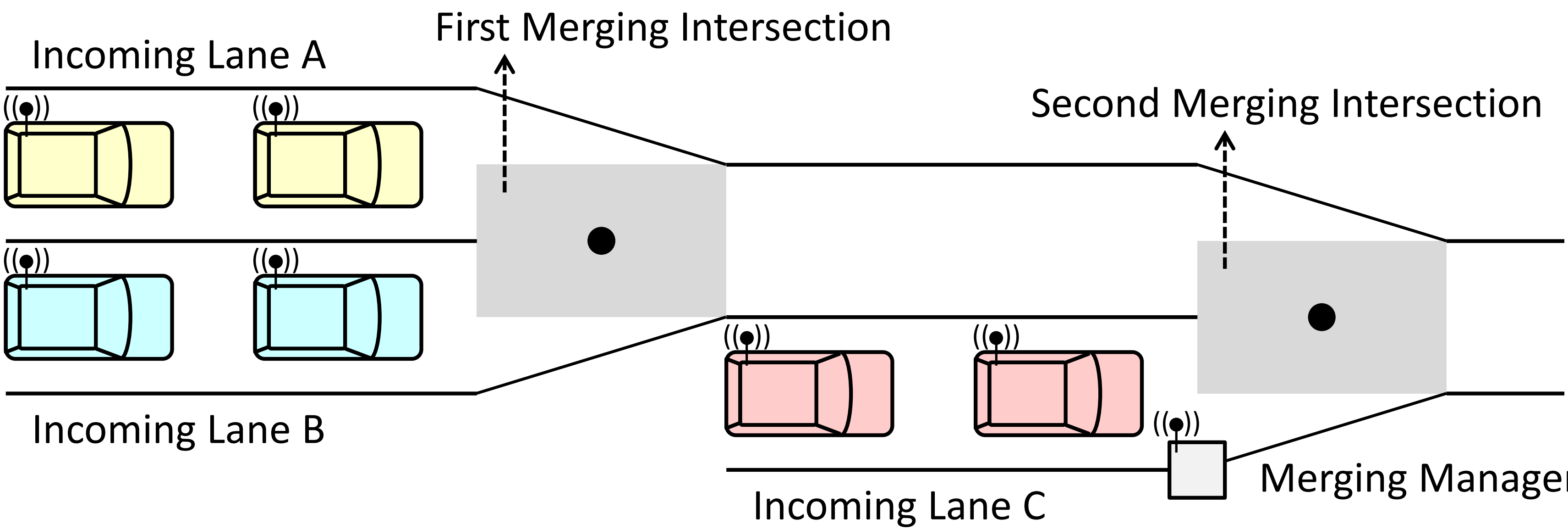


- Given earliest arrival time: (1, 2, 3, 4)
- FCFG scheduled entering time: (1, 4, 7, **10**)
- Optimal scheduled entering time: (1, 6, 3, **7**)
  - Dynamic programming!
  - T(i,j) is the scheduled entering time of the last vehicle among all vehicles until i-th vehicle from Lane A and j-th vehicle from Lane B

Three-Lane Model



Consecutive Two-Lane Model



Q&A