Toward Smarter Transportation Systems with Autonomous Driving

Tsz-Chiu Au

chiu.au@txstate.edu

TEXAS STATE UNIVERSITY®

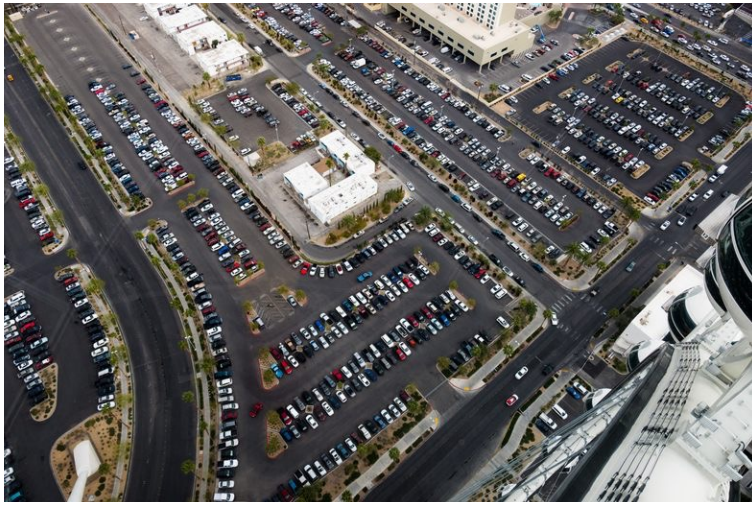
The Rise of Self-Driving Cars: A New Era in Transportation

Can autonomous driving help solve the most critical challenges facing modern transportation systems?

Two Major Challenges in Modern Transportation Systems

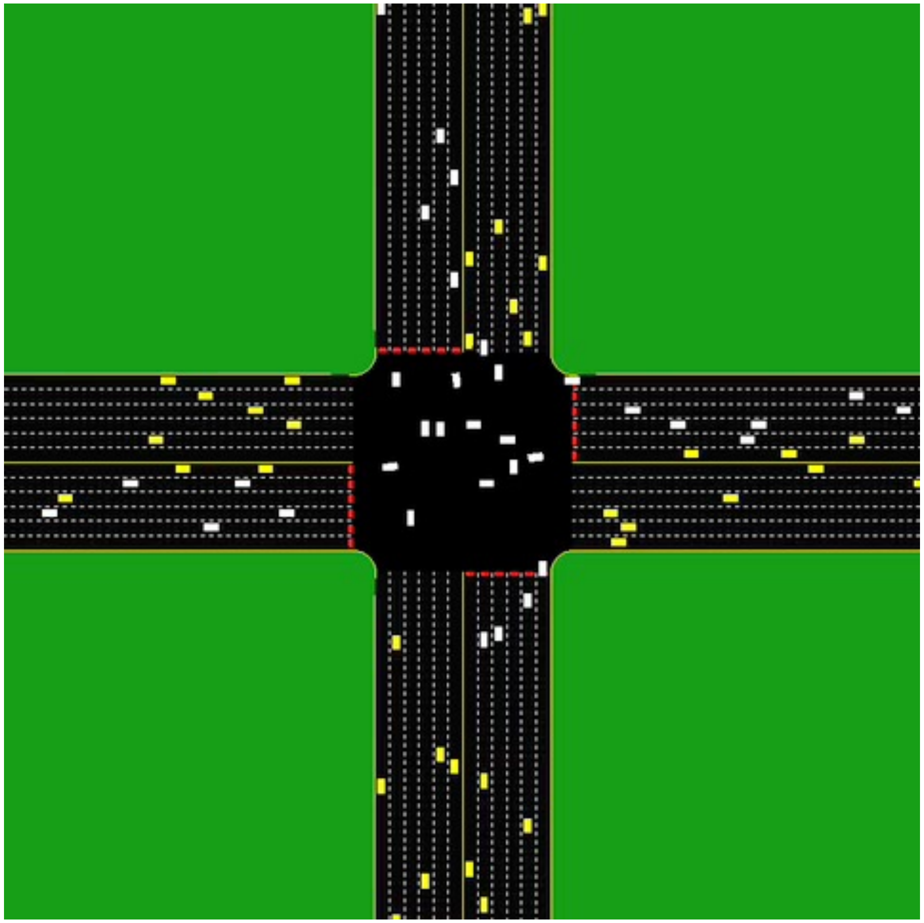
Traffic Jams at Intersections

Limited Parking Space



Part 1: Traffic Jams at Intersections

Autonomous Intersection Management (AIM)

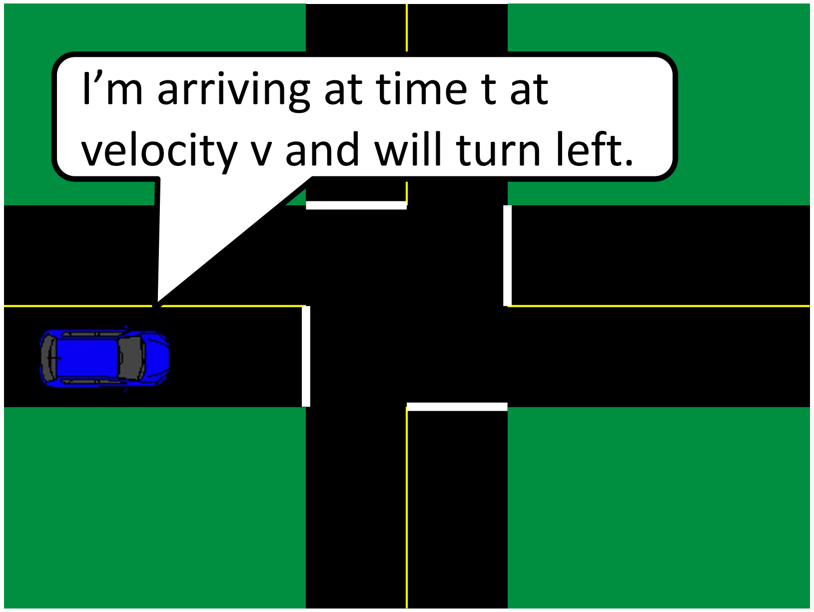


- Traffic signals and stop signs are designed for human drivers.
- Unlike human-driven vehicles, autonomous vehicles can coordinate with each other wirelessly to enter intersections.
- **Autonomous Intersection Management (AIM)**
  - ➔ Reduce traffic delay to almost zero.
  - ➔ Less fuel consumption
  - ➔ Less emissions
  - ➔ Sustainable society

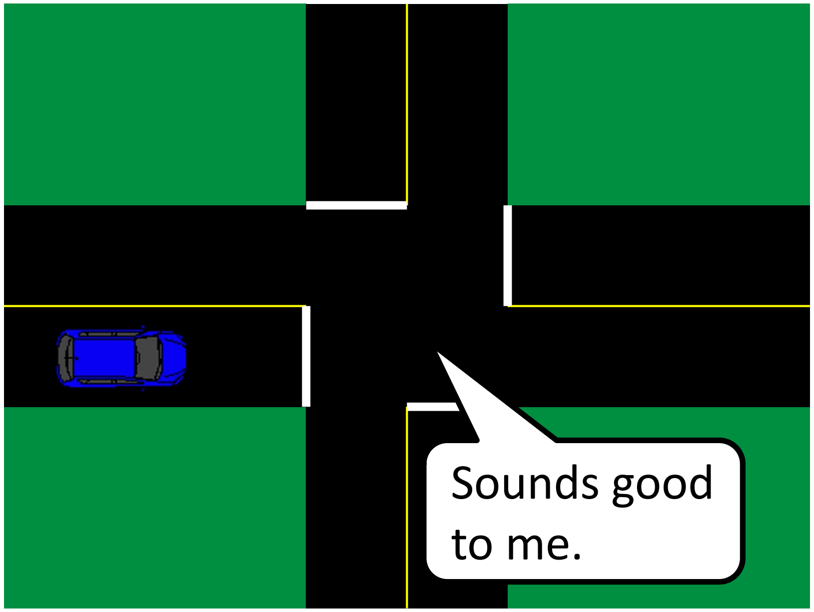
Intersection Manager

- Vehicles call ahead to **reserve regions of space-time** on their trajectories in the intersection.
- The **intersection manager** approves or denies the request.

The autonomous vehicle said:

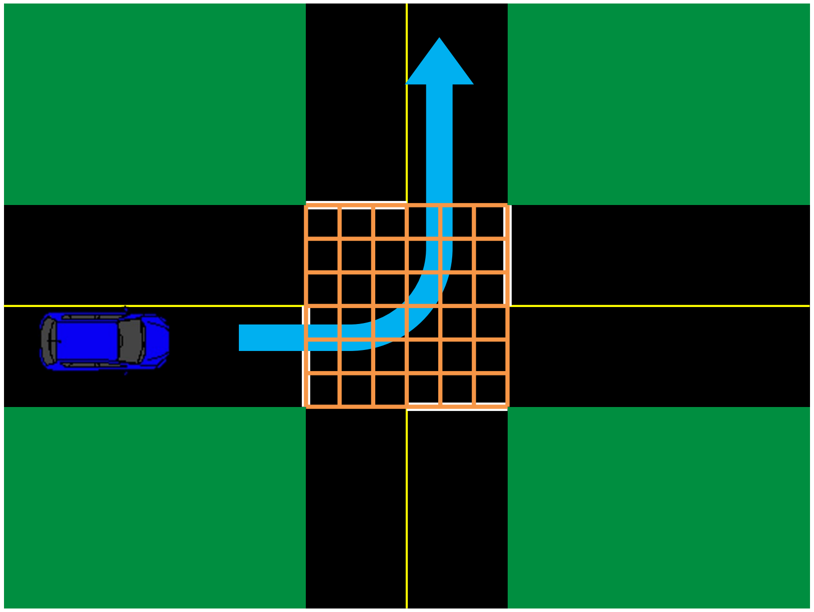


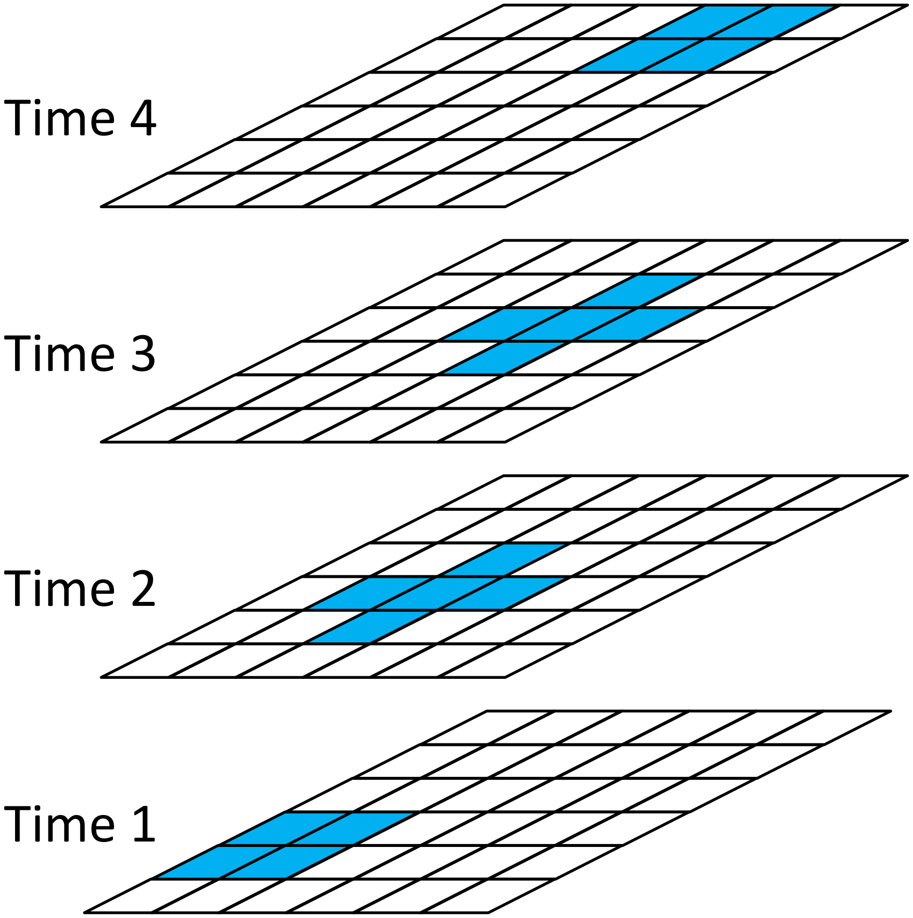
The intersection manager said:



The Reservation System

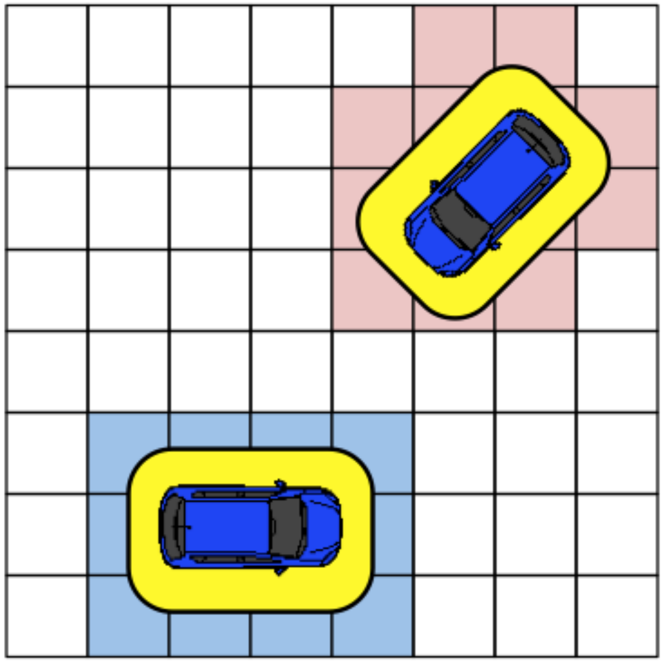
- Reserve the set of space-time tiles on the trajectory.



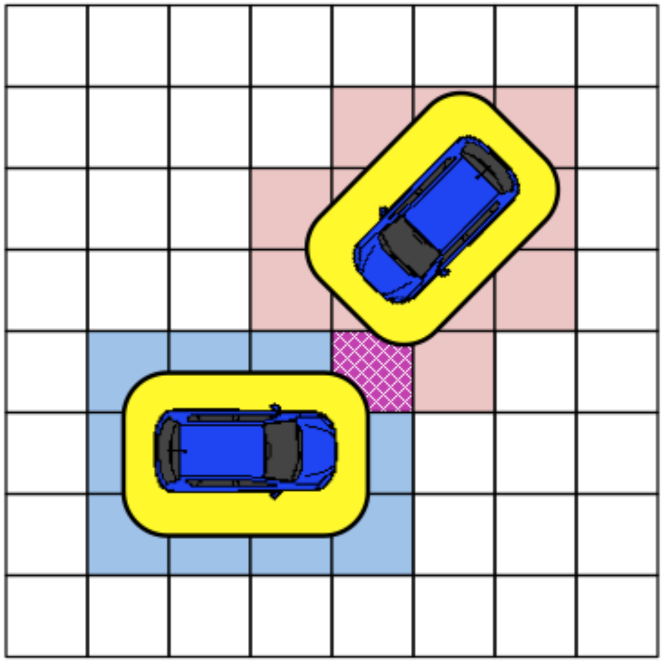


Grid-based Collision Detection

- The intersection manager approves the reservation requests of other vehicles only if their trajectories do not intersect with the reserved space-time tiles.



Accept

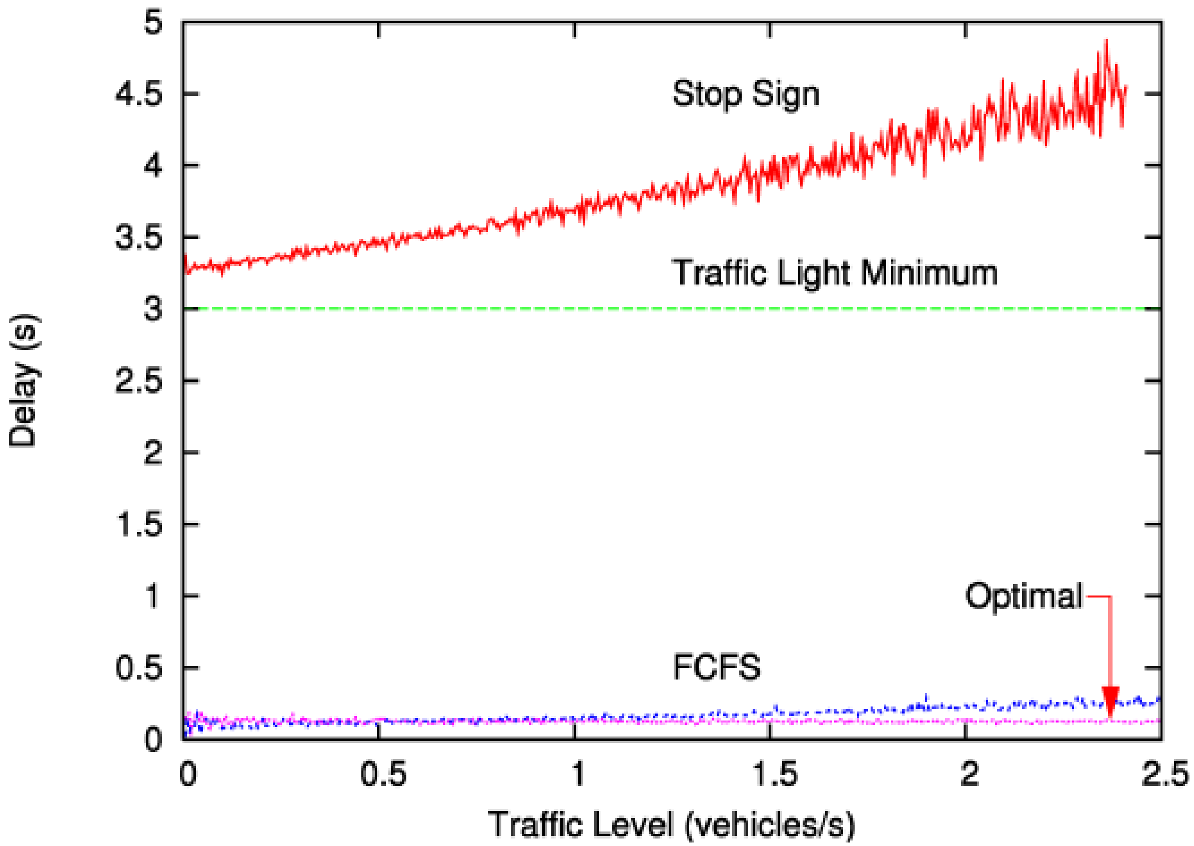


Reject

How does AIM work?

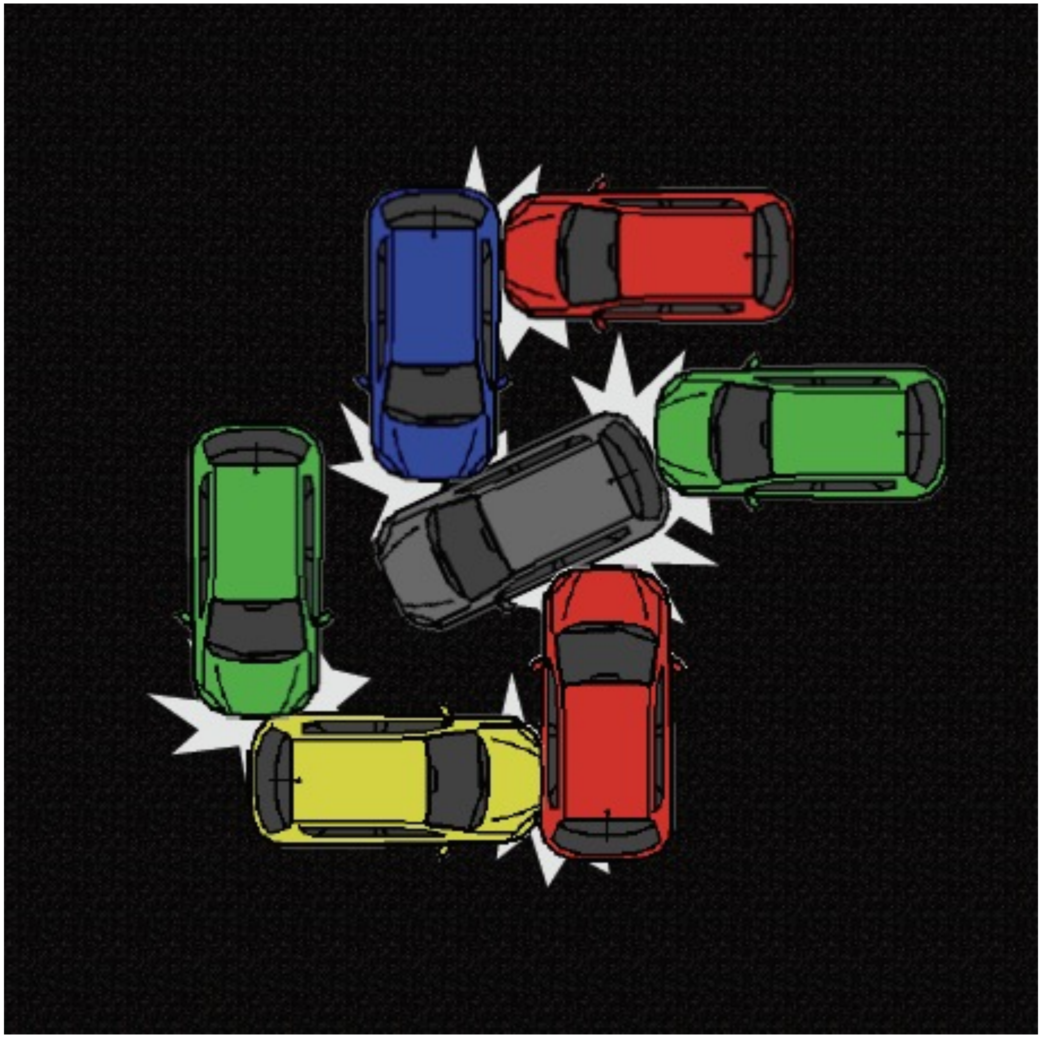
How does AIM work?

The Performance of AIM



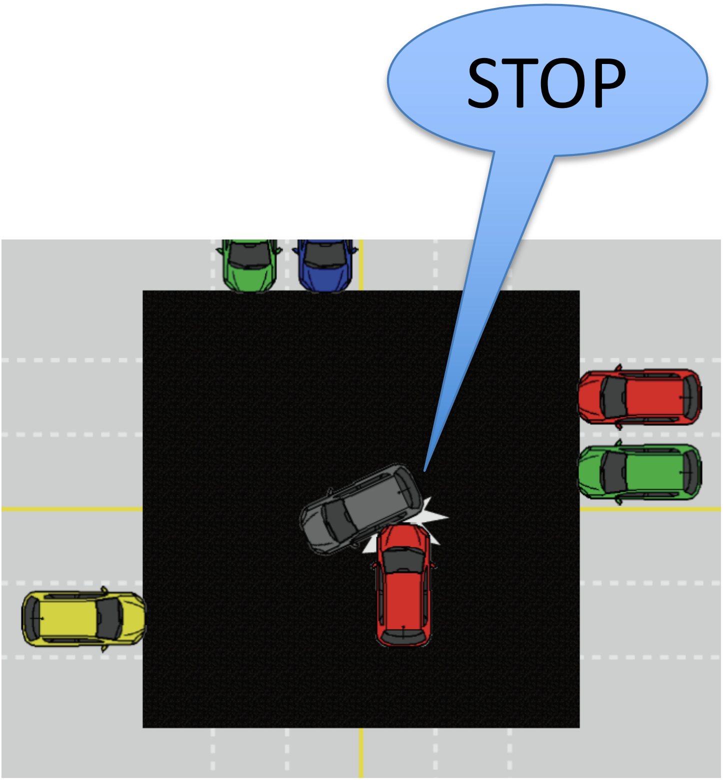
- The traffic delay was reduced to almost zero at most traffic levels!
  - » Traffic level – the number of vehicles per second per lane.
  - » Traffic Delay – the increased travel time due to the traffic.

Is the protocol safe?



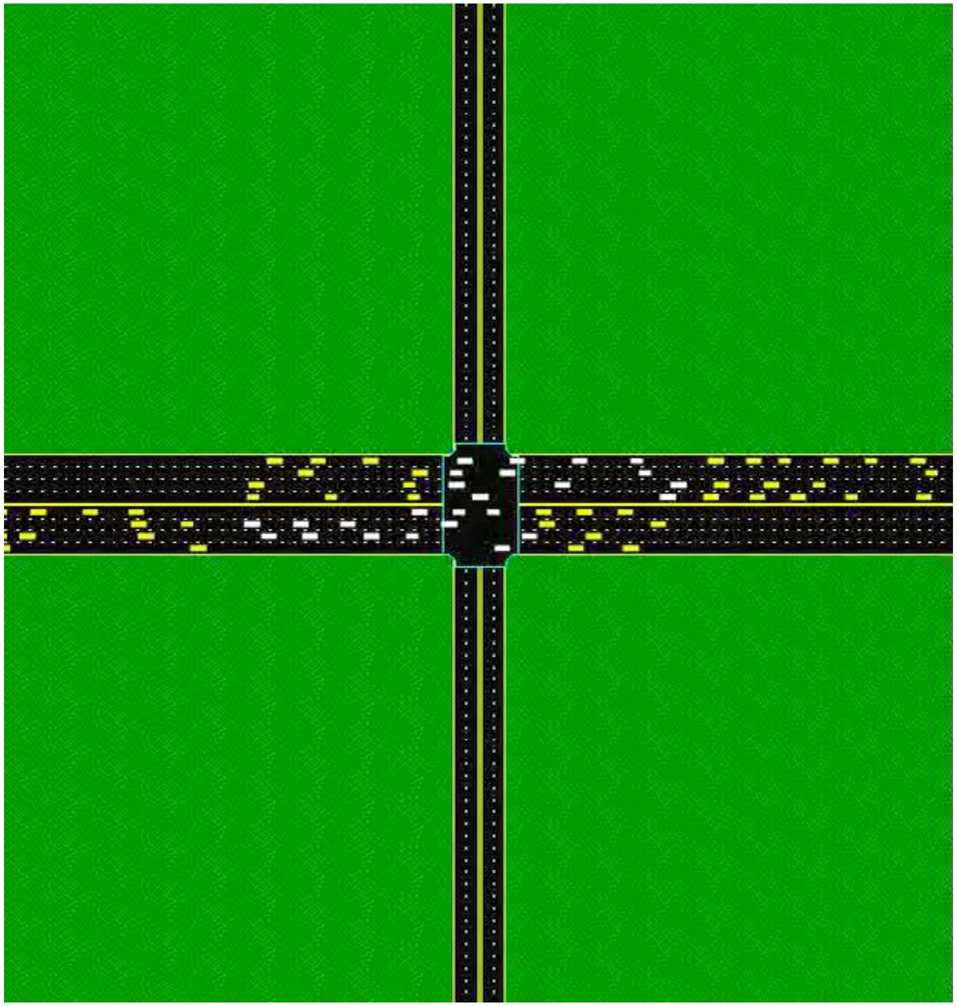
Safety of AIM

- If all autonomous vehicles follow the AIM protocol
  - no collision.
- If some messages were dropped
  - the vehicle won't enter the intersection.
    - AIM is a **fail-safe** protocol.
- If a vehicle loses control in the intersection
  - rely on **evasion planning**.



Unbalanced Traffic

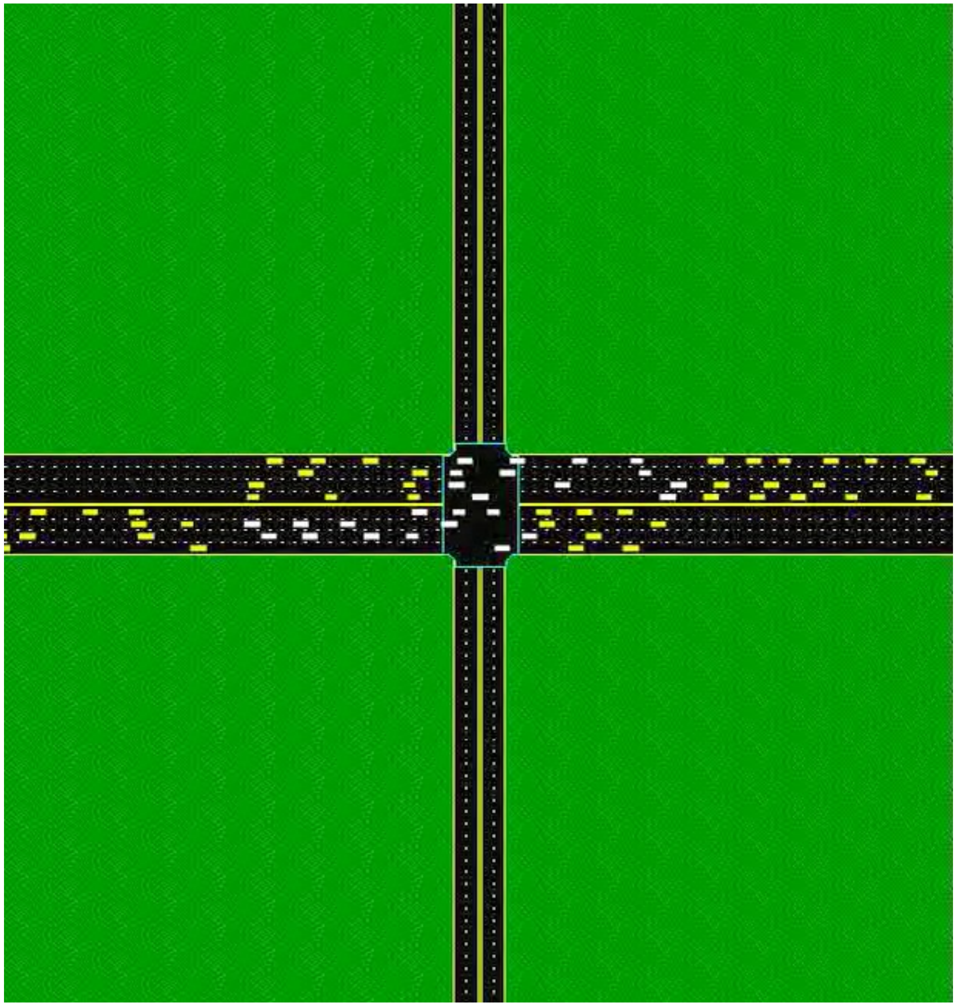
- **Unbalanced traffic**: traffic in one direction is much heavier than traffic in the other direction.
  - » AIM performs poorly.
  - » Worst case: **Starvation**
- Traffic signals can handle unbalanced traffic easily.
  - » But poor performance.



Unbalanced Traffic (Cont.)

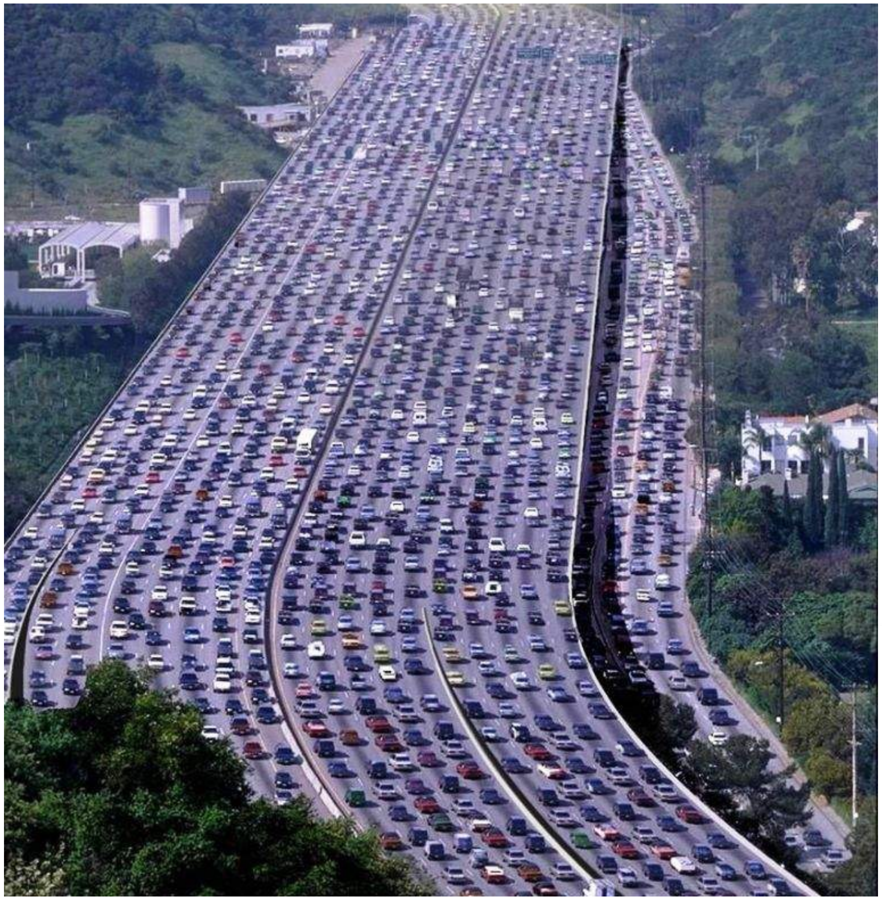
- **Solution:** give vehicles a *higher priority* to enter an intersection when they have been waiting for a long time.
  - » need a priority queue.

Can vehicles get stuck in a road network when multiple intersections are managed by AIM?



Traffic Jam in Road Networks

- How long can vehicles get stuck in traffic?

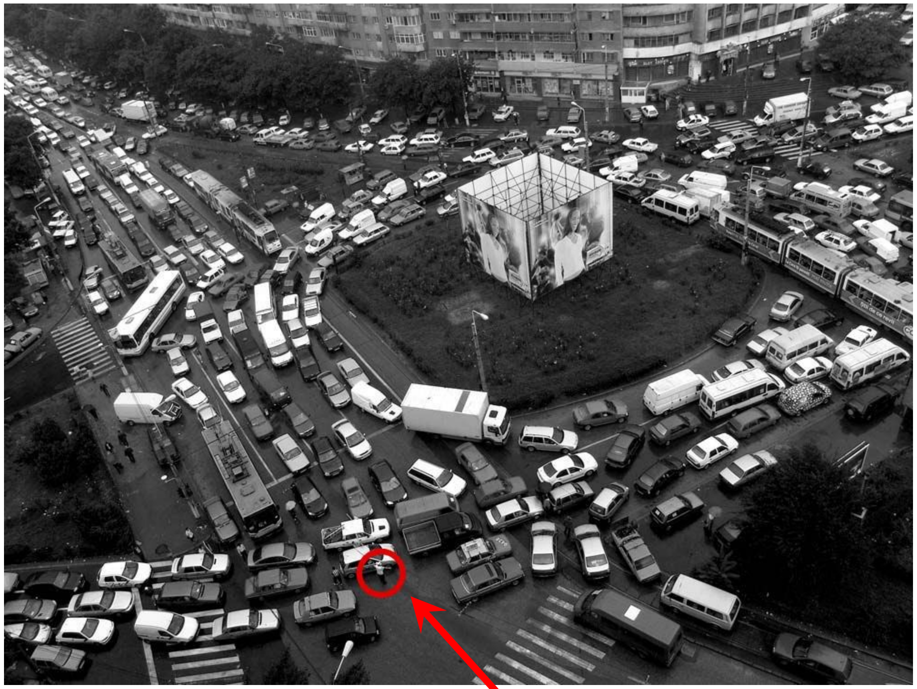


The 2010 Great Chinese Gridlock

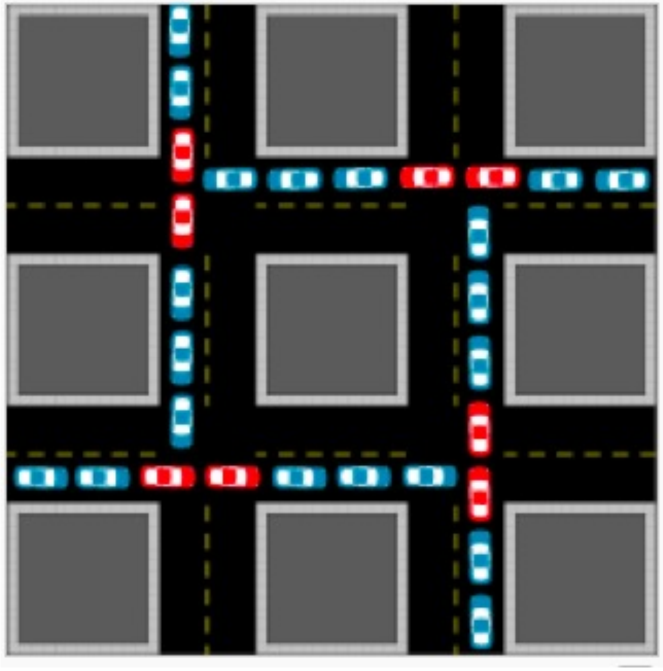
- 60 miles jam lasted for 11 days

Gridlock

- Vehicles in the Great Chinese Gridlock are not “locked”.
- In some gridlock, however, vehicles can be **stuck forever**.
- A bigger issue for autonomous vehicles than human drivers



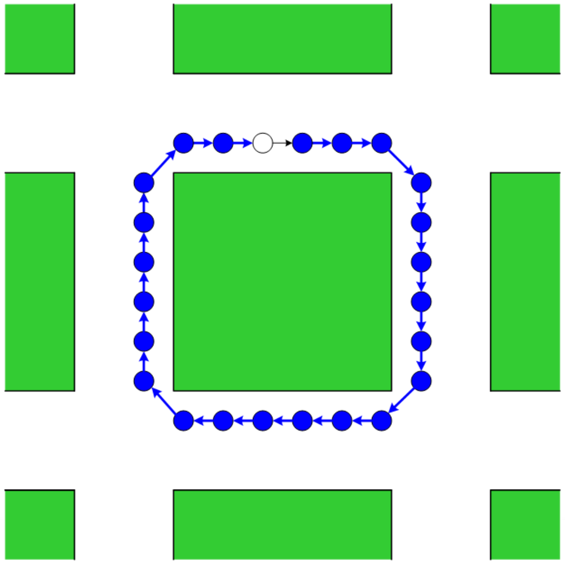
Traffic police



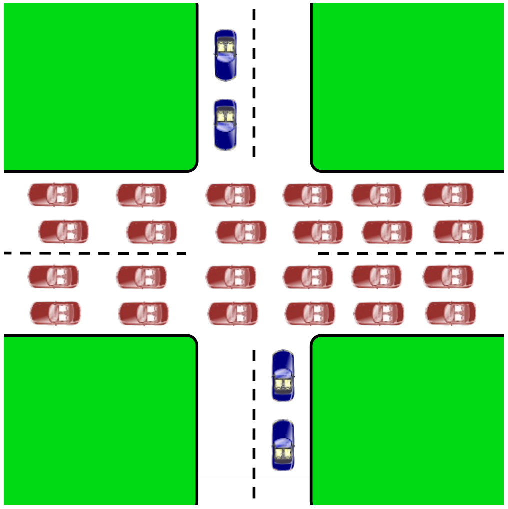
Beyond Gridlock

- The lack of gridlock does *not* guarantee no vehicle get stuck.

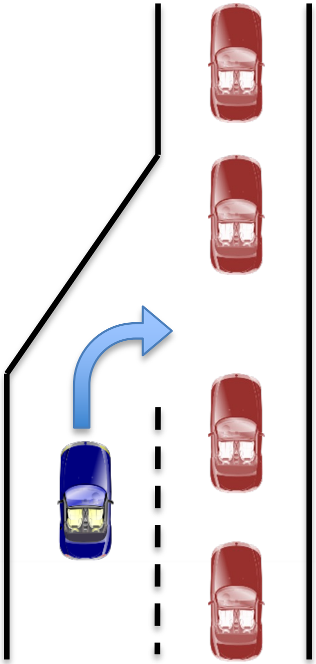
Gridlock-like situations



Poor Intersection Control

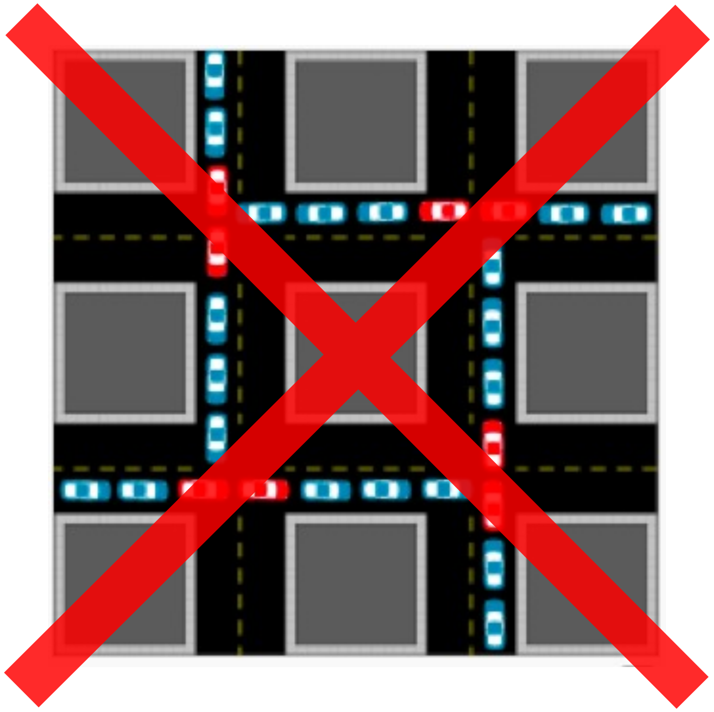


Unfair Competition



Liveness of Road Networks

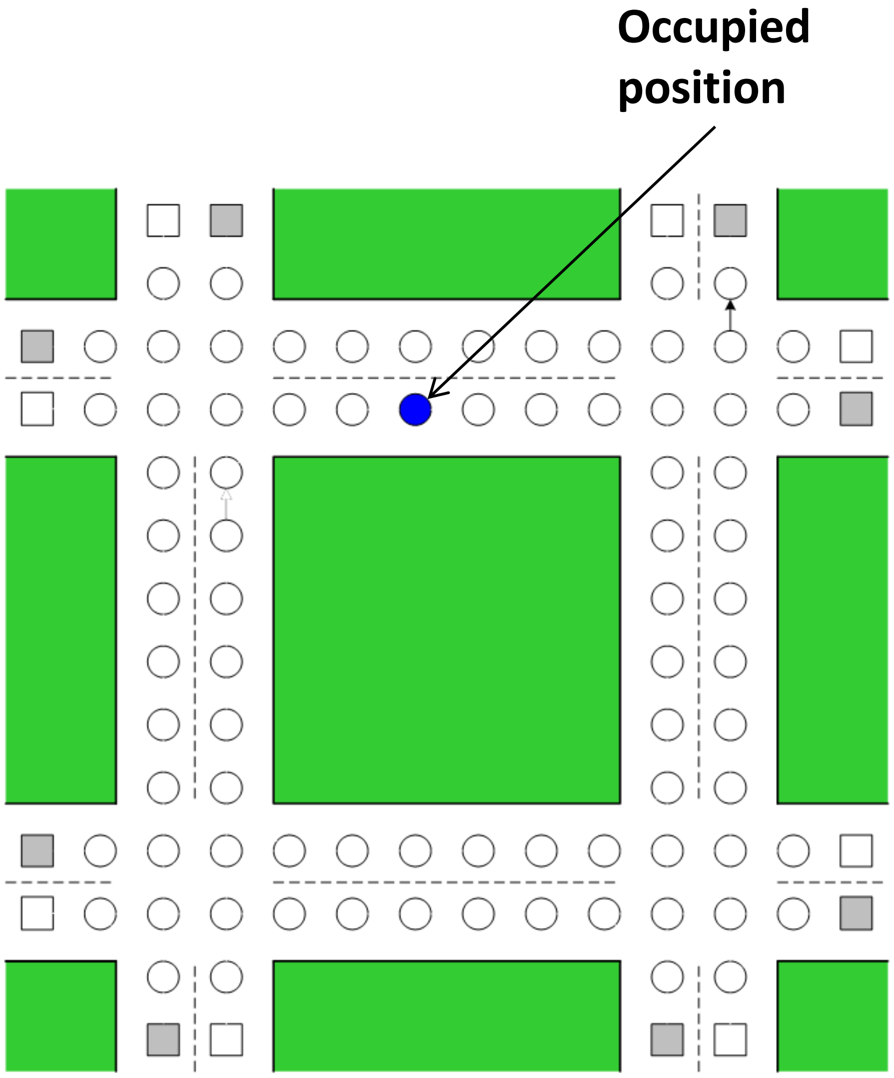
Liveness – every vehicle that entered a road network can eventually leave the road network.



What is the sufficient condition for liveness of a road network?

Discretized Road Network

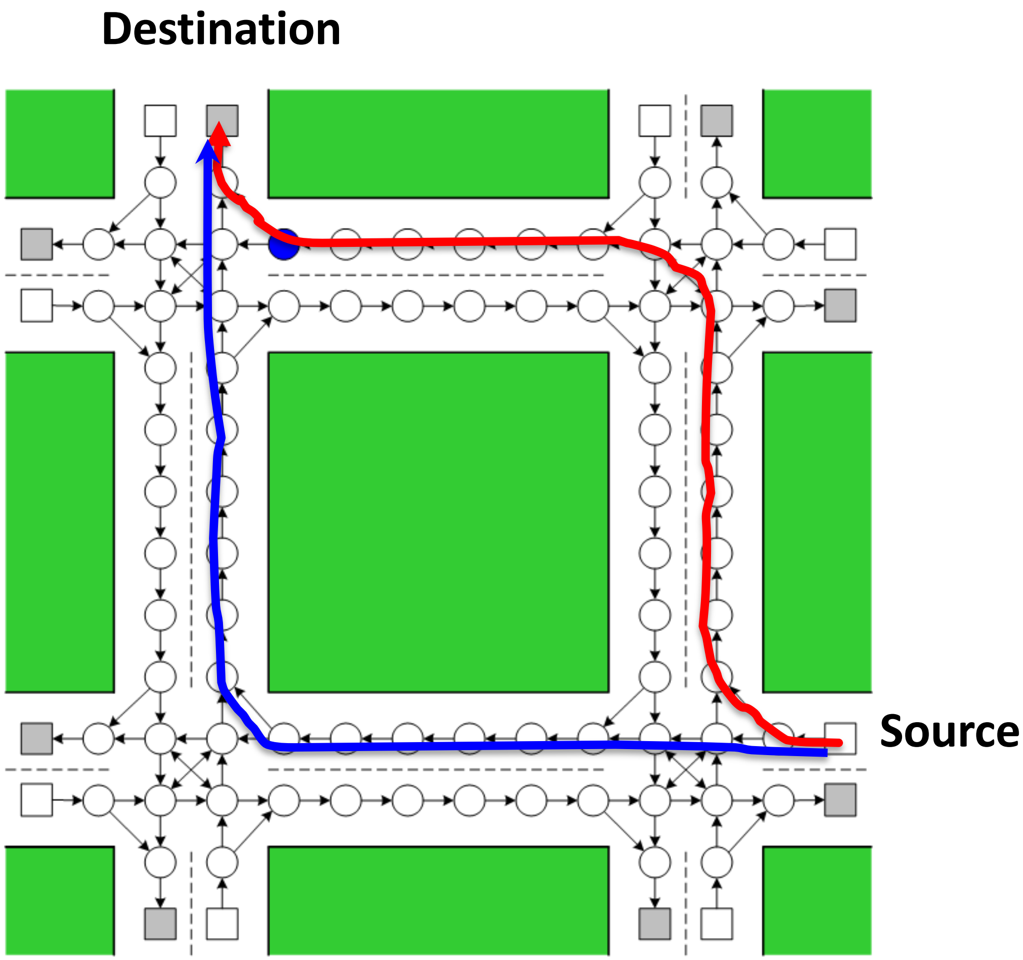
- A **simplified model** of transportation systems that includes
  - Vehicle movements
  - Competitions among vehicles.
  - Traffic control mechanisms
- Discretize the road surface into **positions**.
- Each position can be **occupied** by at most one vehicle at a time.



Occupied position

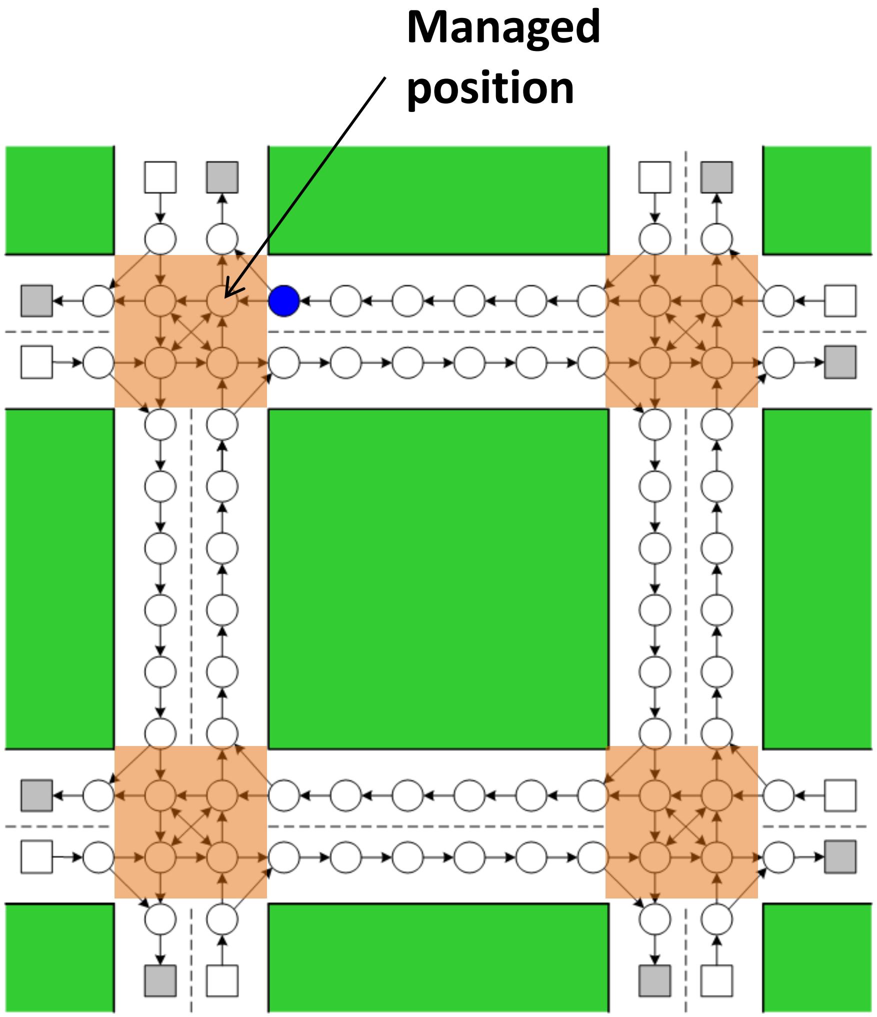
Vehicle Movements

- Vehicles enter the road network from the **sources** and leave the network from the **destinations**.
- When a vehicle enters the road network, it knows its destination.
- Each position has a set of **next positions**.
- Deterministic vehicle controller vs. stochastic vehicle controller.



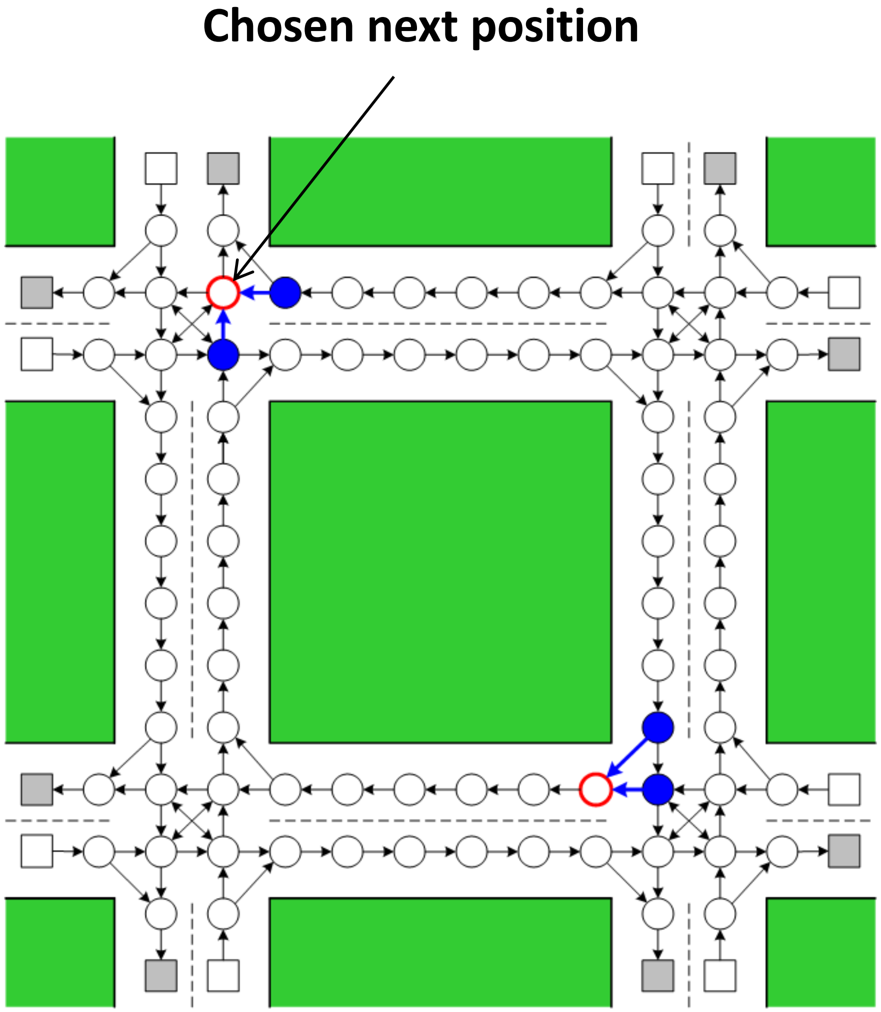
Traffic Control Mechanisms

- Positions inside the intersections are managed by **traffic control mechanisms.**
  - e.g., AIM
- They are called **managed positions.**
- Vehicles must obtain the **right of ways** of a managed position at time t in order to be allowed to enter the position at time t.
- The right of way is assigned by the traffic control mechanism of the managed position.
- All vehicles have the right of ways of *unmanaged* positions at all times.



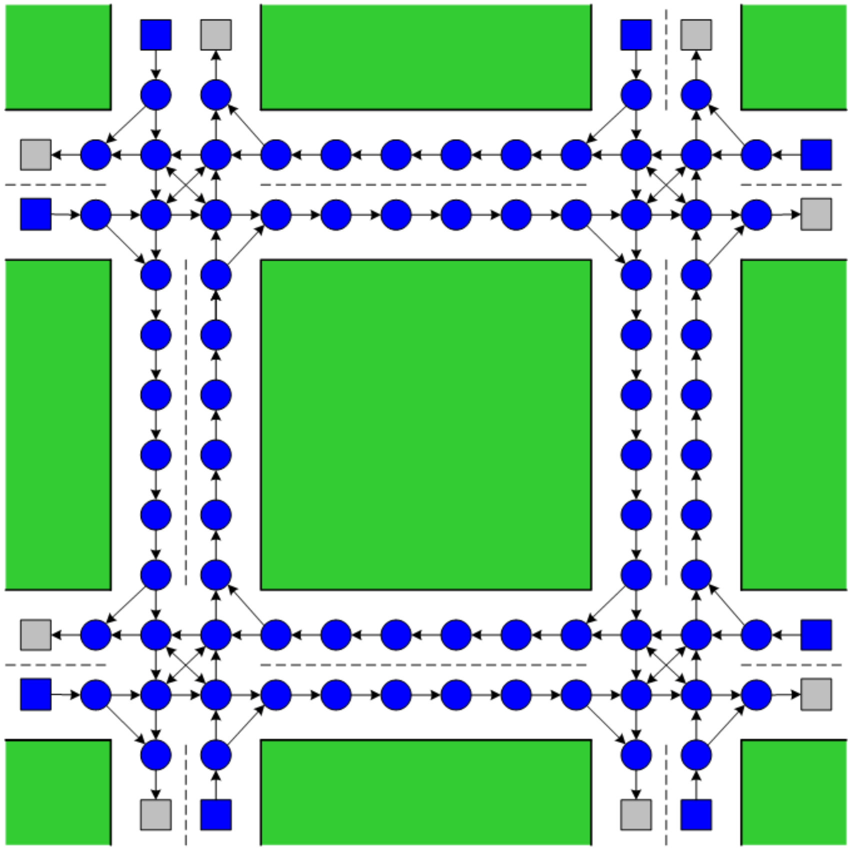
Coordination Mechanisms

- A vehicle can choose to move into the next position of which the vehicle has the right of way.
    - » **chosen next position**
- If two vehicles choose the same next position *p*, they are **competing** with each other.
- The **coordination mechanism** at *p* selects one of the competing vehicles to move into the position.
    - » The unselected vehicle remains at the same position.



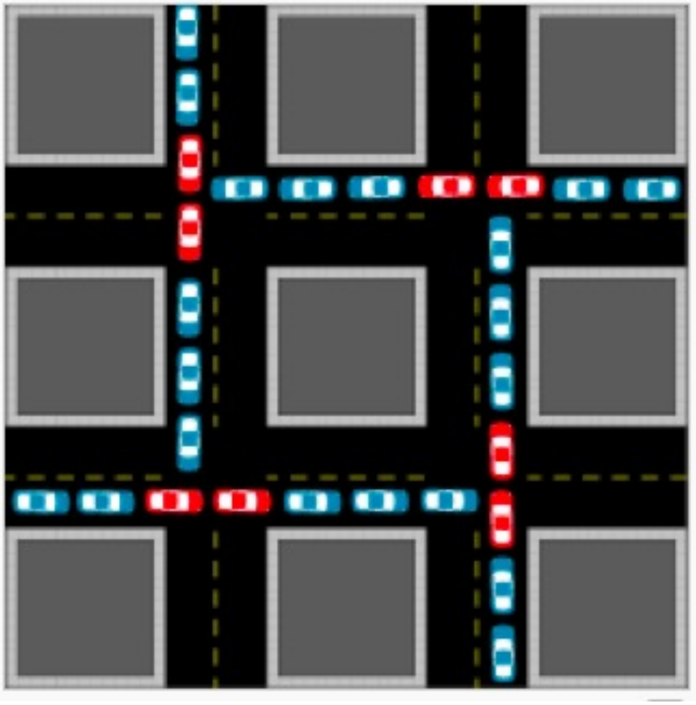
Nonexistence of Hard Gridlock

- A vehicle is **completely blocked** if *all* the next positions are occupied.
- A **hard gridlock** is a subgraph in which all positions are occupied, and all vehicles are completely blocked by each other.
- **Theorem 1**: If the body of a discretized road network is **strongly connected**, there can be *no* hard gridlock.



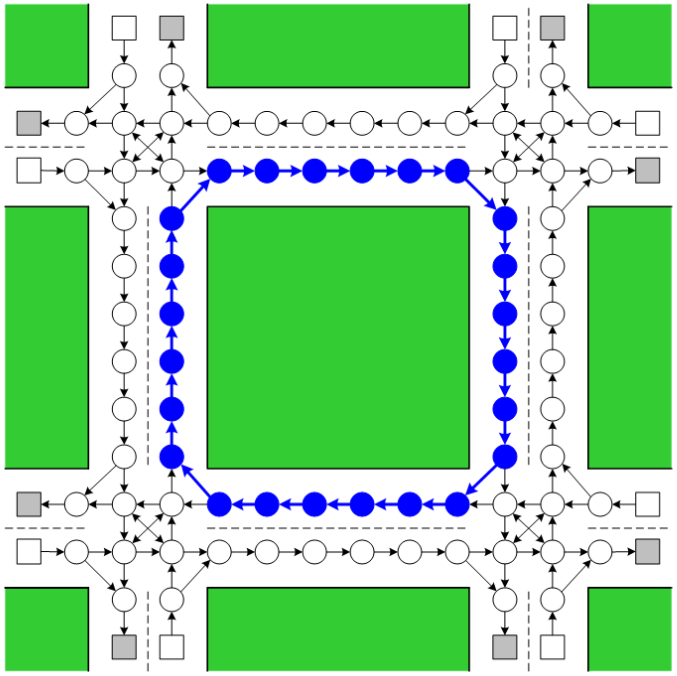
Soft Gridlock

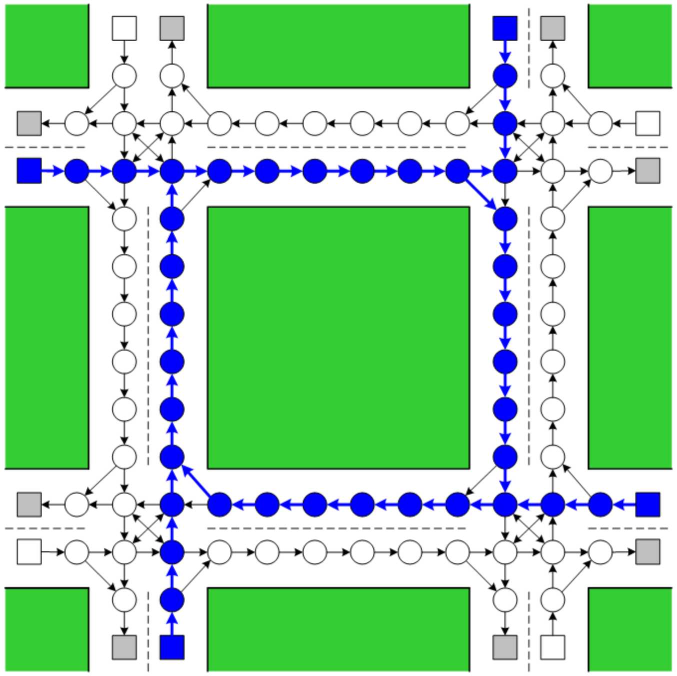
- Vehicles can get stuck in the traffic, even if some vehicles are not completely blocked.
- A vehicle is **essentially blocked** as long as all of its **relevant** next positions are occupied.
- In **soft gridlocks,** all vehicles are essentially blocked.



Two types of Soft Gridlock

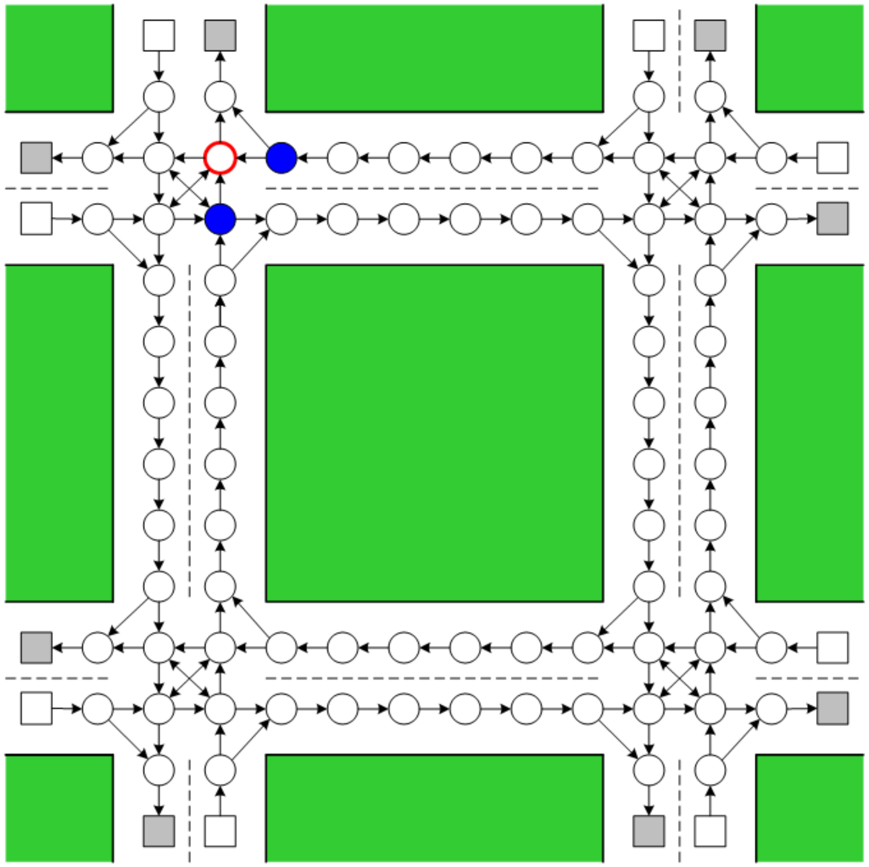
- **Soft Gridlock for Deterministic Controllers (SGDC)**
- **Soft Gridlock for Stochastic Controllers (SGSC)**





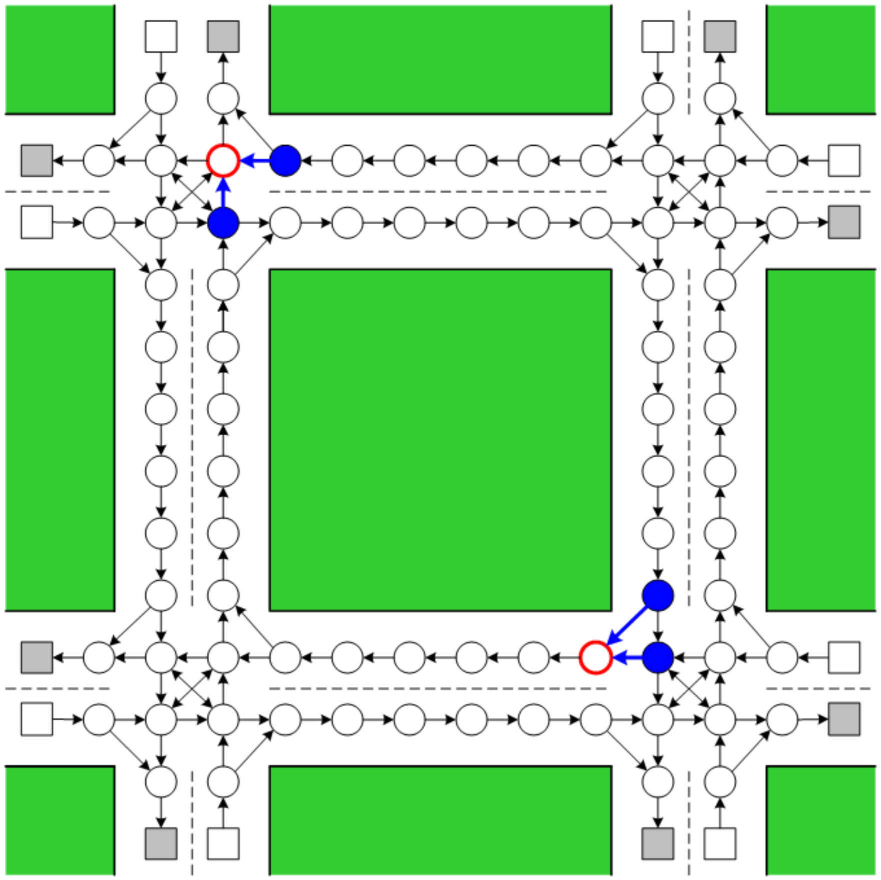
Open Traffic Control Mechanisms

- A traffic control mechanism Ψ is **open** if whenever a vehicle **repeatedly** chooses to move into an unoccupied position p managed by Ψ it will **eventually** get the right of way of p.



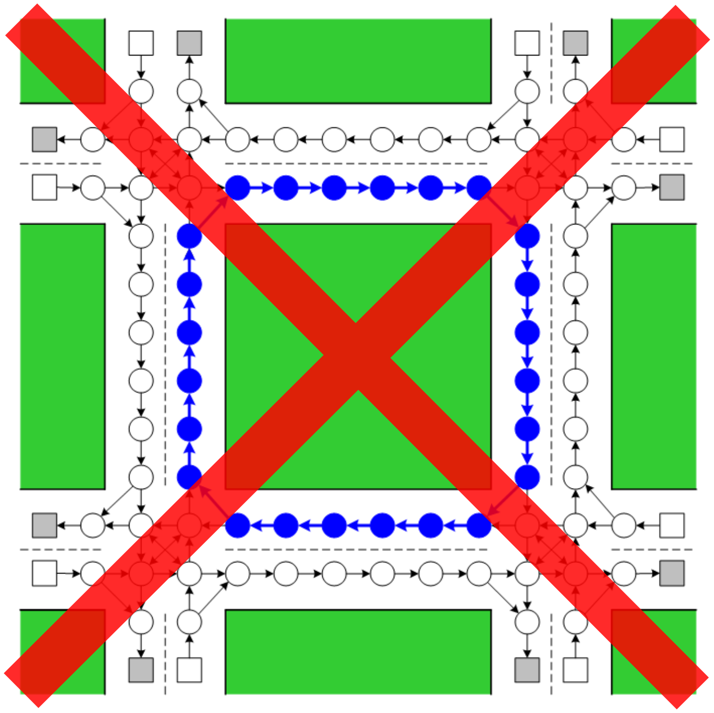
Fair Coordination Mechanisms

- A coordination mechanism is **fair** if whenever a vehicle **repeatedly** chooses a chosen next position *p* it will **eventually** move into *p*.



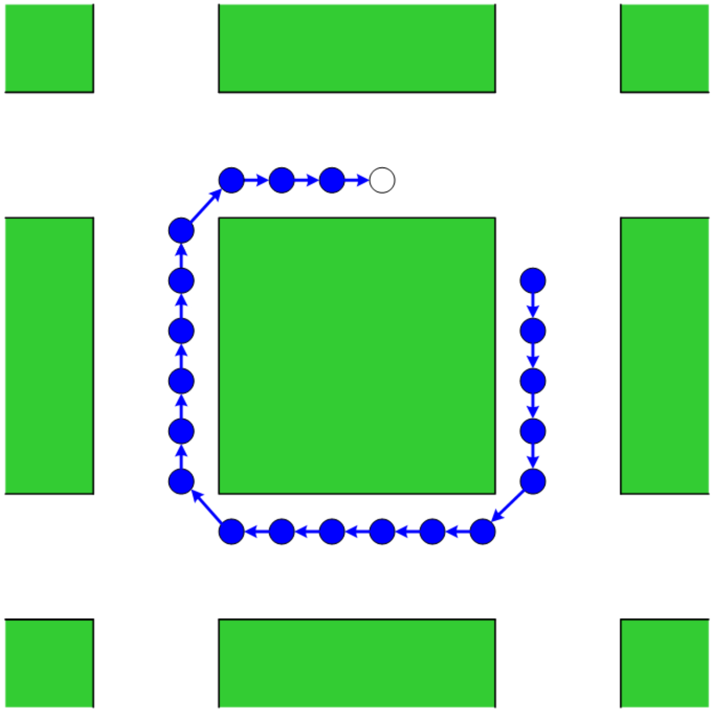
Sufficient Condition for Liveness for Deterministic Vehicle Controllers only

- **Theorem 2**: Every vehicle will eventually reach its destination if
  - All vehicle controllers are ***deterministic***.
  - All traffic control mechanisms are ***open***.
  - All coordination mechanisms are ***fair***.
  - There is ***no soft gridlock*** for deterministic controllers (SGDCs) at *any* time.



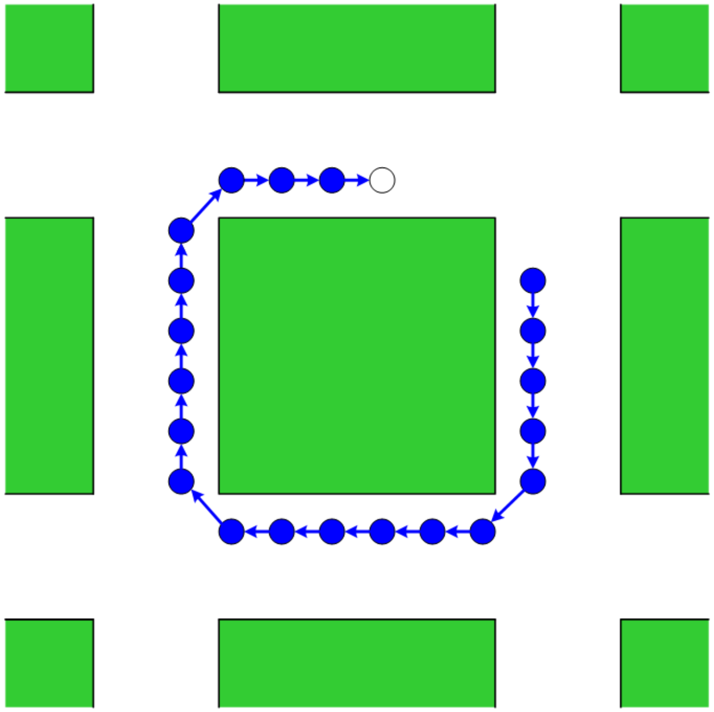
Outline of the Proof

- If the chosen next position *p* of a vehicle *v* is **unoccupied repeatedly**,
  - » *v* will eventually obtain the right of way of *p* if the traffic control mechanism (if any) is open.
  - » *v* will eventually move into *p* if the coordination mechanism at *p* is fair and *v* obtains the right of way of *p* repeatedly.
  - » Therefore, *v* will eventually move to *p*.



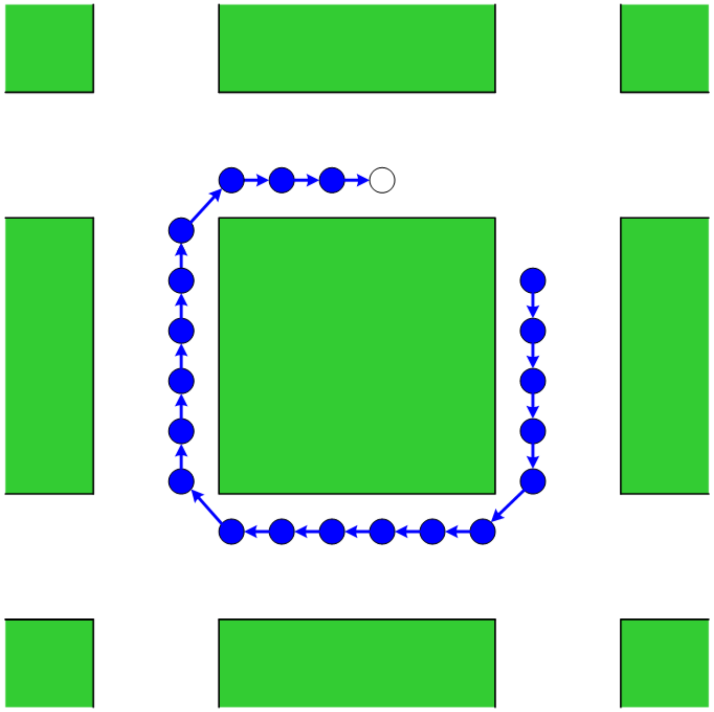
Outline of the Proof (cont.)

- When there are no SGDCs, all dependency lists are finite and at most $|P_{body}|+1$
- We partition the set of all dependency lists in all possible configurations according to their lengths.
- We prove by **backward induction** on the length of the dependency list that the last vehicle of a dependency list can eventually move to its chosen next position.
- **Base Case:** The last vehicle of a dependency list of length $|P_{body}|+1$ can always move to its chosen next position, which is a destination.
- Assume the last vehicle of a dependency list of length *greater than or equal to* k can eventually move to its chosen next position in all possible configurations.



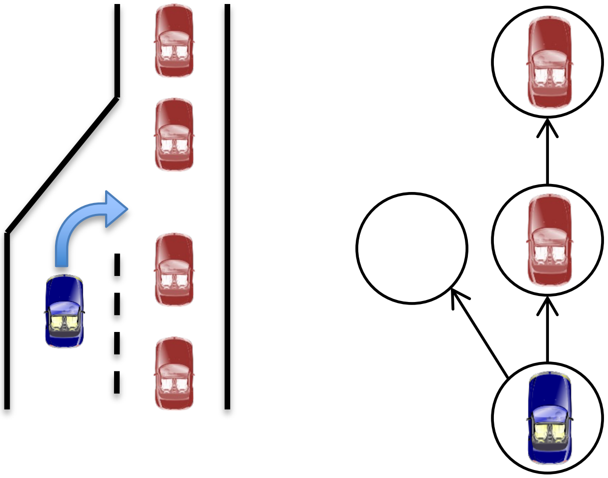
Outline of the Proof (cont.)

- The last vehicle *v* of a dependency list of length k-1 will try to move to its chosen next position *p*.
  - If *v* succeeds, *v* moves to *p*.
  - If *v* fails, another vehicle v’ moves to *p*.
    - This forms a dependency list longer than k-1.
    - By the assumption, *v’* and all vehicles after *v’* will move to its chosen next position.
    - Therefore, *p* will become unoccupied again, and *v* can try to move to *p* again.
    - After trying repeatedly, *v* will eventually succeed due to openness and fairness.
  - In both cases, *v* can eventually move to its chosen next position.
- By backward induction, a vehicle will eventually move into its chosen next position
  - A vehicle will eventually reach its destination.



Sufficient Condition for Liveness in the Presence of Stochastic Controllers

- **Theorem 3**: Every vehicle will eventually reach its destination if
  - There is ***no*** soft gridlock for stochastic controllers (SGSCs) at any time.
  - All traffic control mechanisms are ***open.***
  - All coordination mechanisms are ***fair.***
  - All vehicle controllers are ***progressive.***
    - The vehicle always moves on a path towards its destination.
  - All vehicle controllers are ***opportunistic***.
    - Whenever a relevant next position *p* is unoccupied repeatedly, the vehicle will eventually choose *p*.



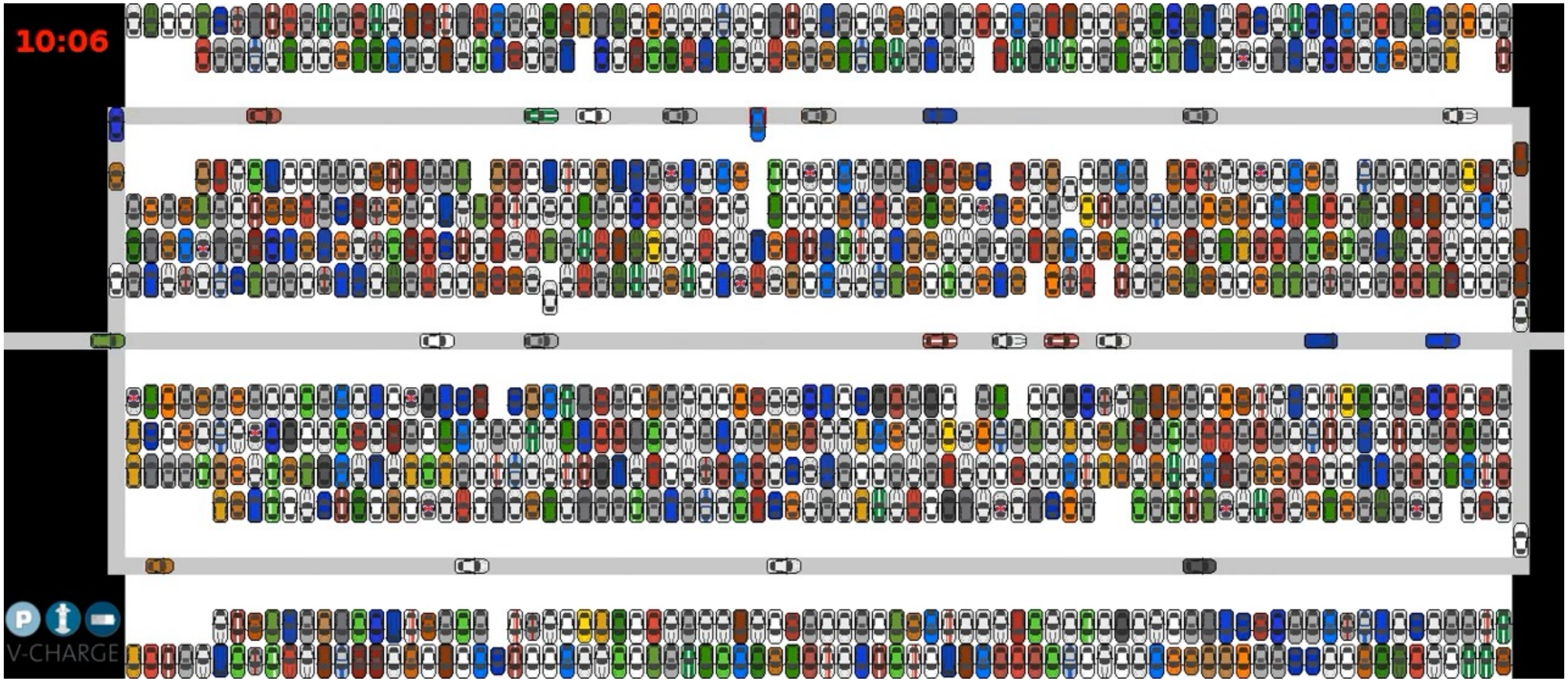
Summary of Part 1

- Liveness is an important property for transportation systems.
- When intersections are not managed by traffic signals and stop signs, liveness may not be guaranteed.
- The sufficient condition for liveness in simplified road networks suggests that
  - » To guarantee the liveness of *real* road networks, we may need to consider
    - New traffic control mechanisms that prevent soft gridlock.
    - New laws to enforce certain driving behavior (fairness in competitions, opportunistic, etc.)

Part 2: Limited Parking Space

High-Density Parking (HDP)

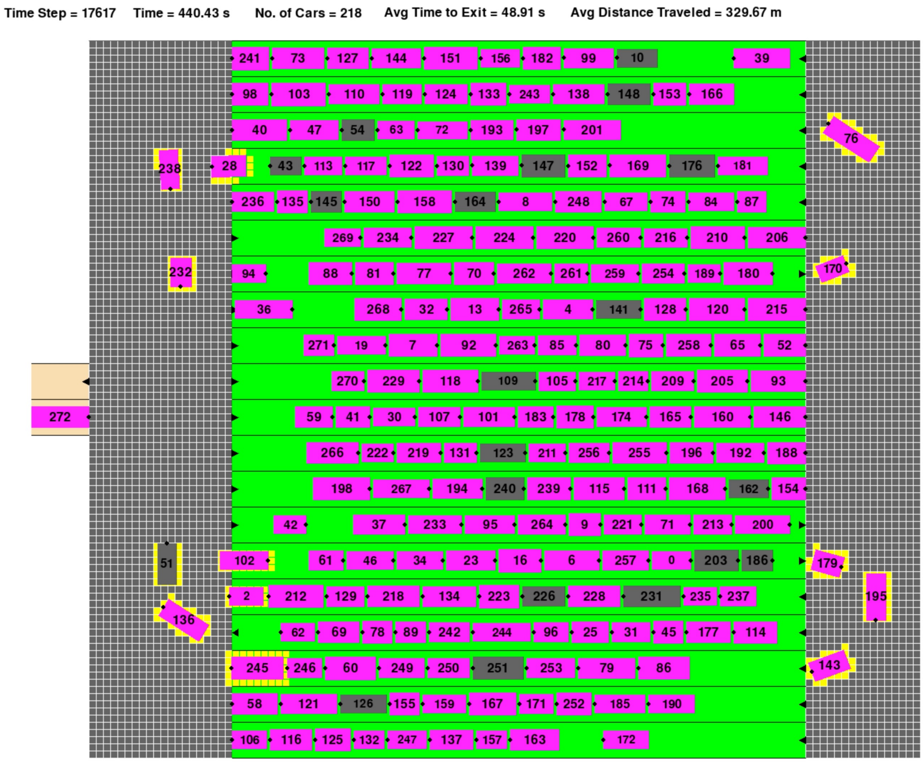
- Conventional parking lots reserve more than half of the space for driveways and sidewalks.
- **High-Density Parking** – utilize autonomous driving to reclaim these spaces by allowing vehicles to block each other.



Source: https://www.youtube.com/watch?v=pCzI-l8tsPY

High-Density Parking

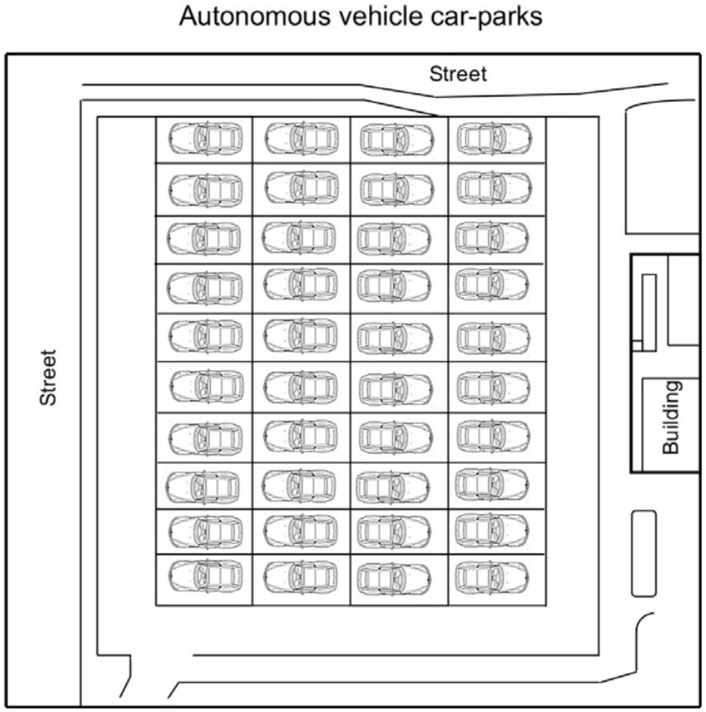
- HDP can increase the capacity of parking lots by an average of 62% (Nourinejad et al. 2018).
- Require fully automated driverless parking functions (SAE Level 4) only.

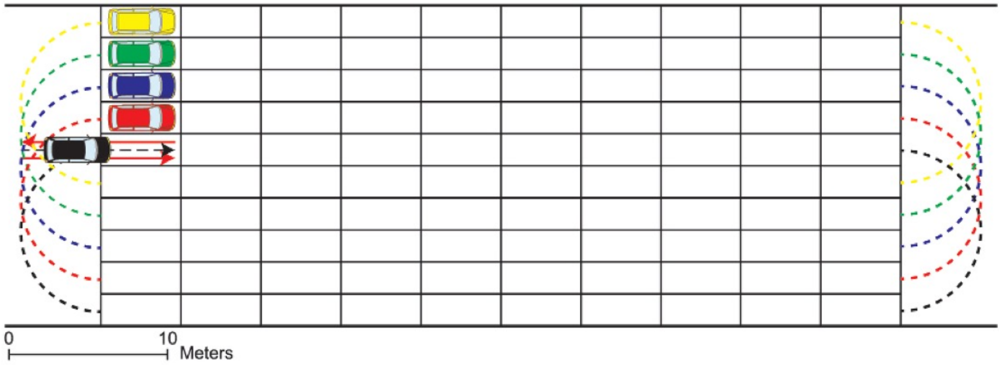


Time Step = 17617 Time = 440.43 s No. of Cars = 218 Avg Time to Exit = 48.91 s Avg Distance Traveled = 329.67 m

Existing Parking Strategies in HDP

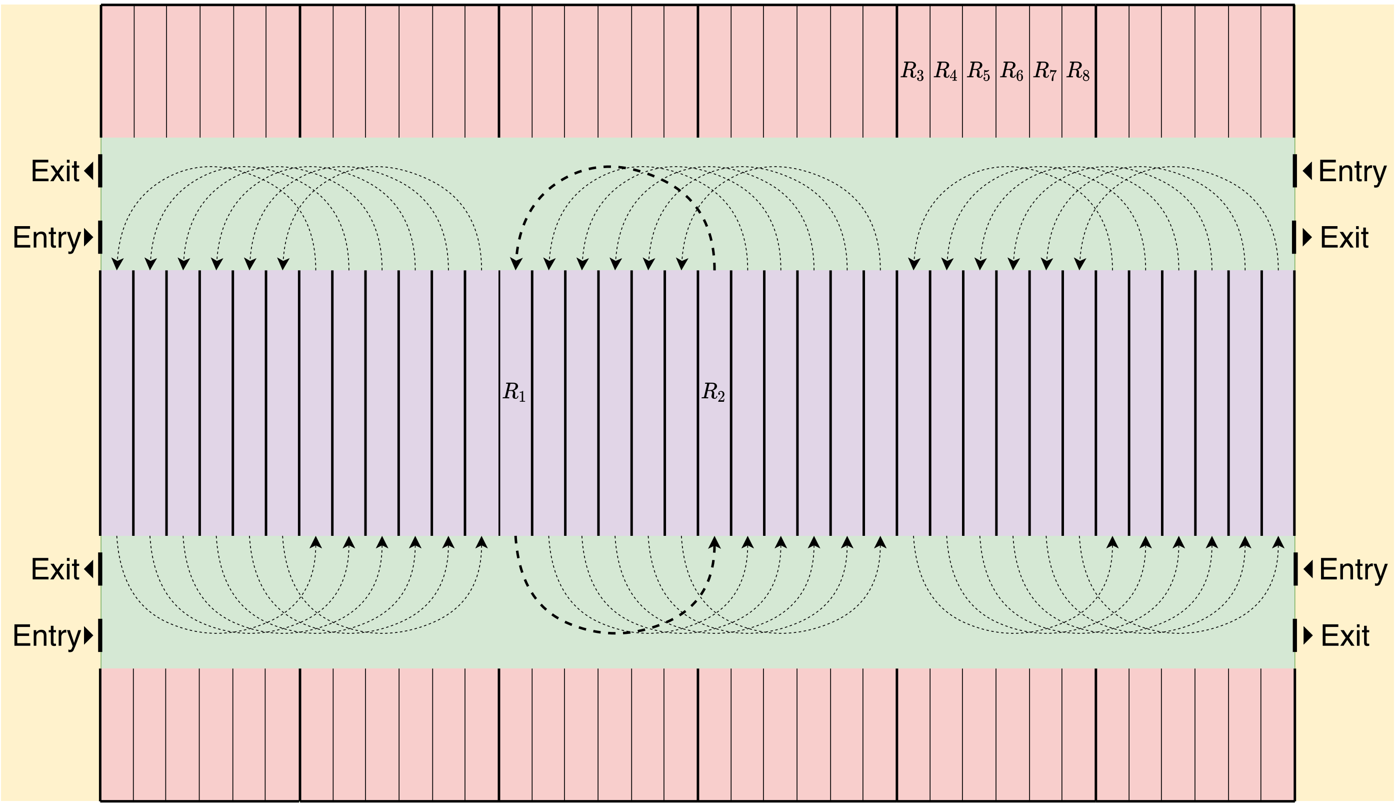
- **Parking Strategies**:
  - **Queues**: Ferreira et al. (2014)
  - **Stacks**: Timpner et al. (2015), P. M. d'Orey et al. (2016, 2017), Nourinejad (2018), Azevedo (2020).
- However, no existing work considers mixing different parking strategies.





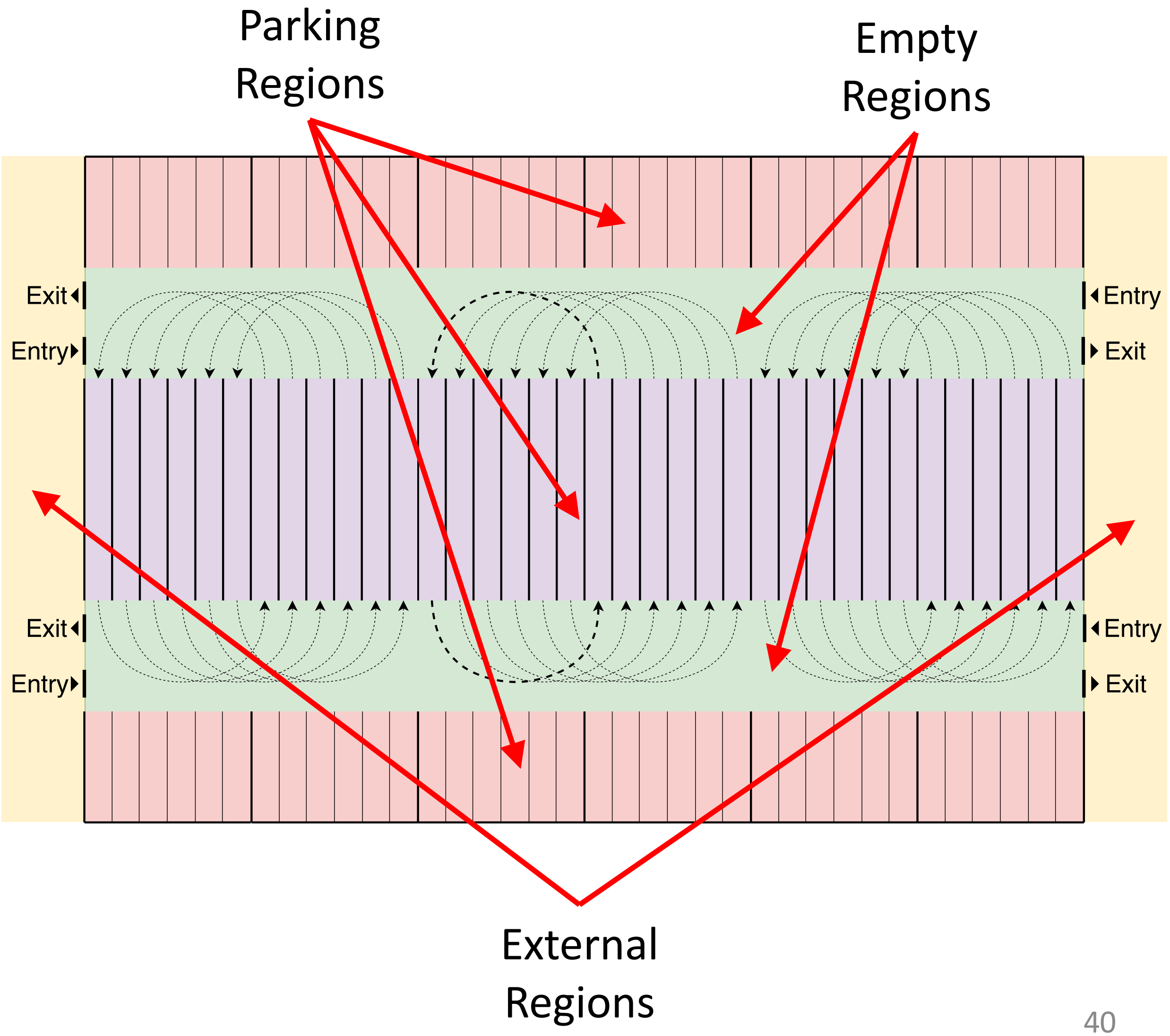
Autonomous Parking Lots

- **Autonomous parking lots (APLs)** – partition a parking lot into different regions, each of which has its own parking strategies and is managed by different agents.



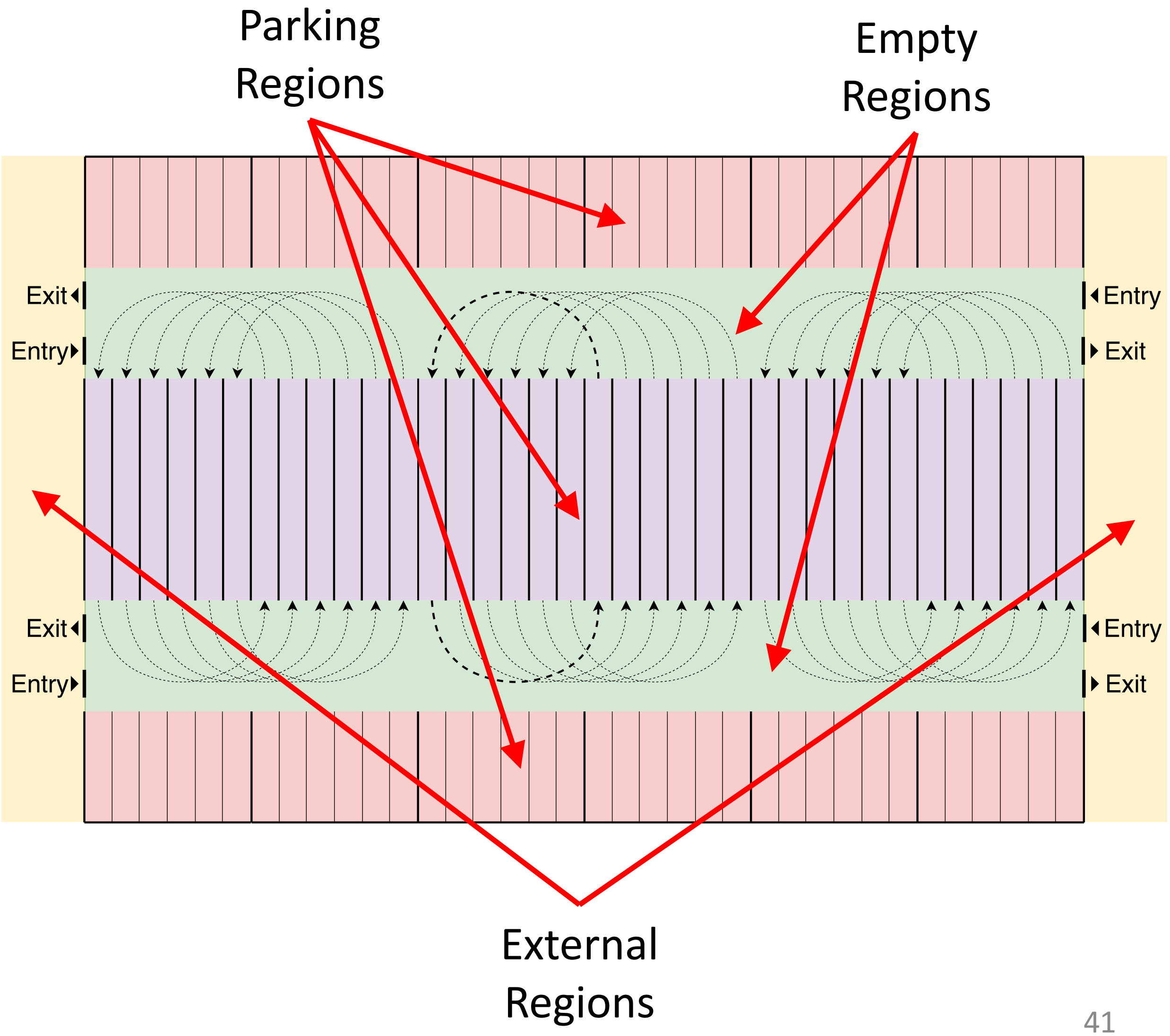
The Design of Autonomous Parking Lots

- **Parking Regions** are managed by agents called **parking managers**.
  - **Stack regions** are managed by **stack managers**.
  - **Queue regions** are managed by **queue managers**.



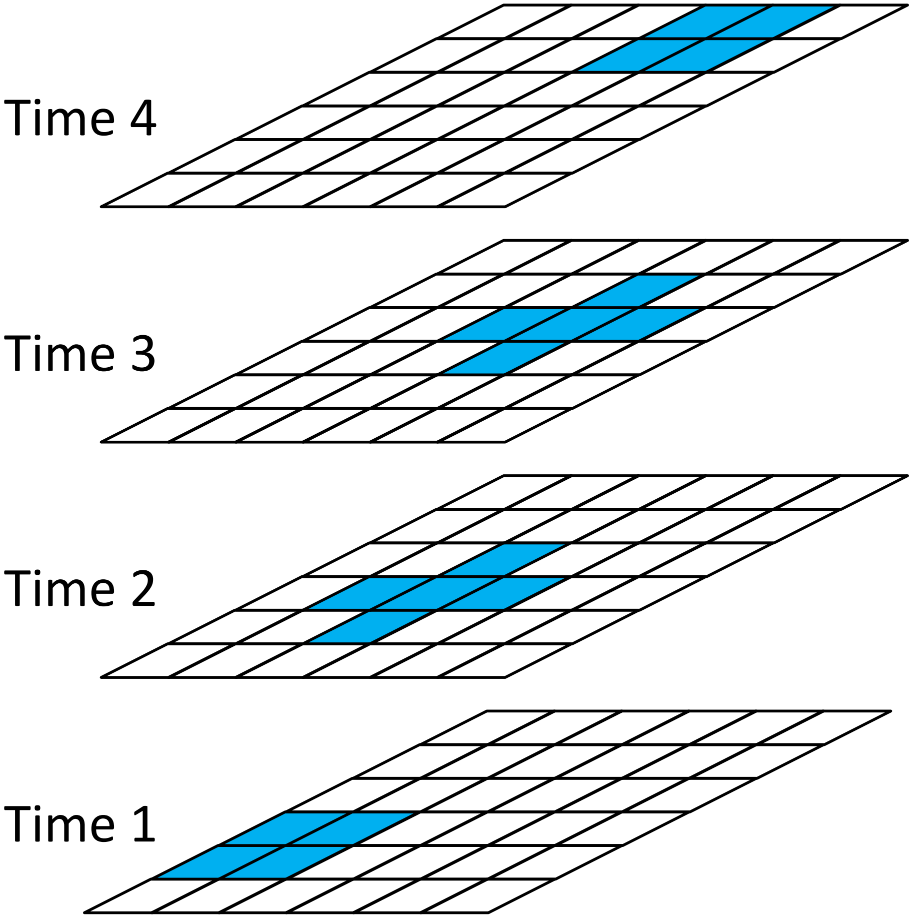
The Design of Autonomous Parking Lots

- **Empty regions** are managed by **reservation managers.**
- **External regions** are managed by **gate managers.**
- External regions and parking regions are connected to empty regions *only*, and vice versa.
- Each parking region must connect to at least one empty region that connects to an external region.



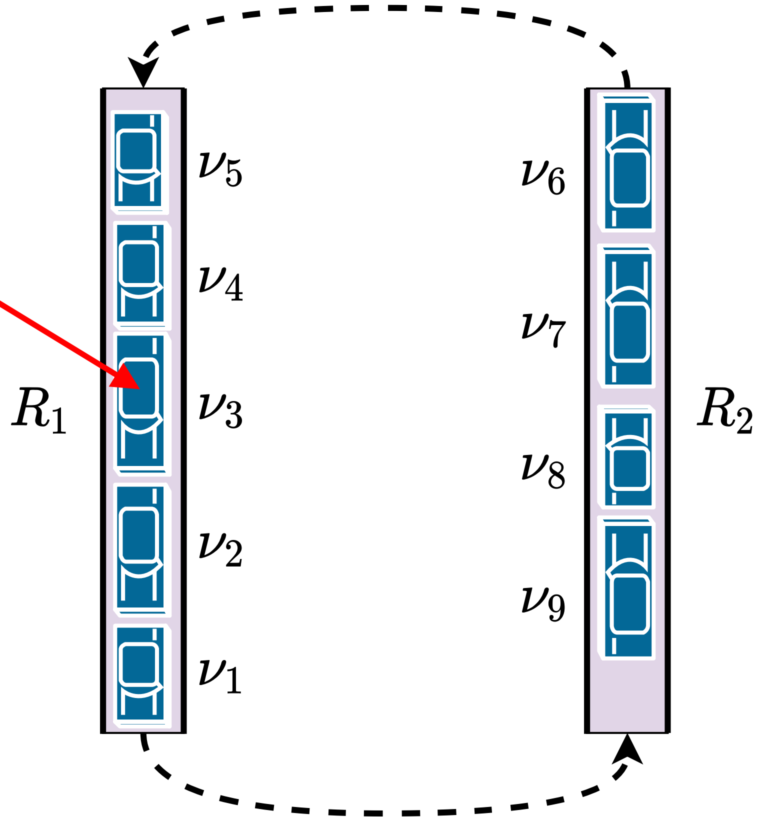
Gridlock in Autonomous Parking Lots

- **Gridlock** – some vehicles cannot leave the parking lot *forever*.
    - » Do *not* occur in existing HDP with stack/queue regions only.
    - » Can occur when *mixing* different parking strategies.
- Our solution:
    - » Use a **reservation system** to coordinate vehicles in empty regions.
    - » The **reservation manager** denies vehicles from parking in a parking region if certain conditions are violated.



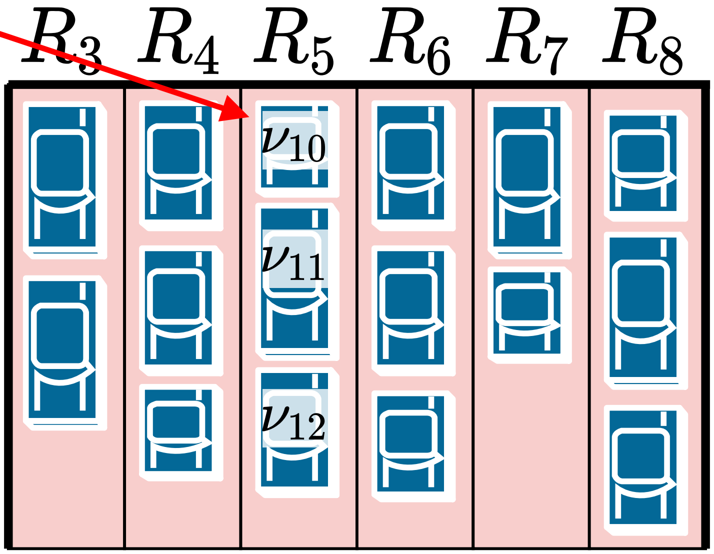
1-Cylability of Queue Groups

- In this example, v_3 can never leave the queue region.
- **Cyclability** – vehicles can fit into different queues of a queue region, no matter how they rotate.
  - However, it may require multiple vehicles to move simultaneously.
- **1-Cyclability** – vehicles can fit into different queues of a queue region, no matter how they rotate, *by moving one vehicle at a time.*
- A queue manager is **1-cyclable** if it maintains 1-cyclability at all times.
  - enforced by the 1-cyclability checking algorithm



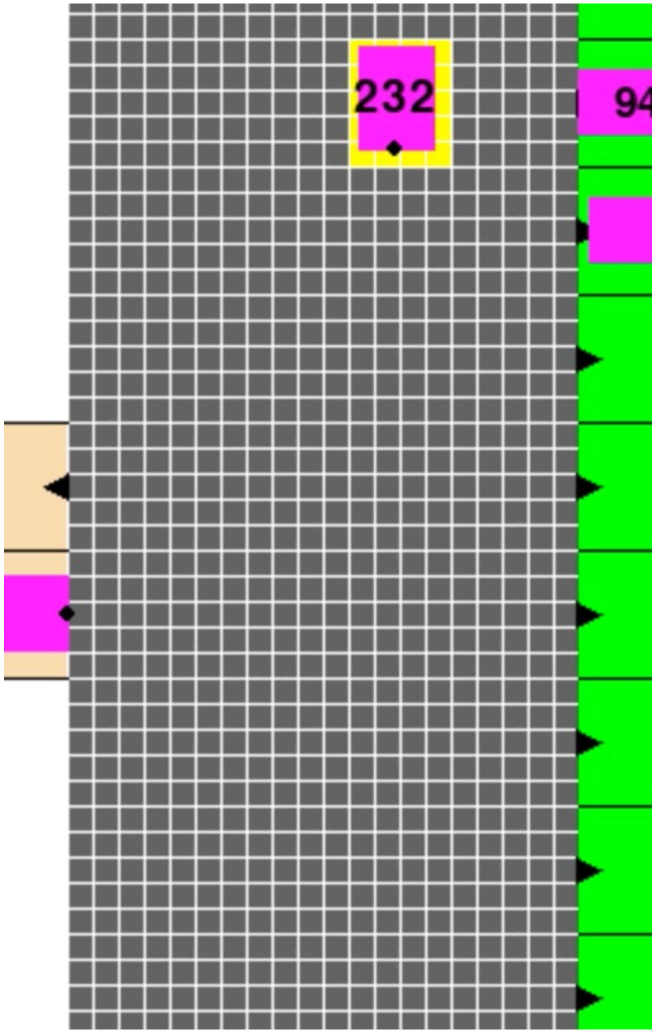
Greedily Relocatability of Stack Groups

- In this example, v_{10} can never leave the stack region.
- **Relocatability** – vehicles at the bottom of any stack can move to another stack by relocating the vehicles above it in the stack.
  - However, checking relocatability is an NP-hard problem (the bin packing problem).
- **Greedily relocatability** – always relocates a vehicle to a stack with the largest empty space.
- A stack manager is **greedily relocatable** if it maintains greedily relocatability at all times.



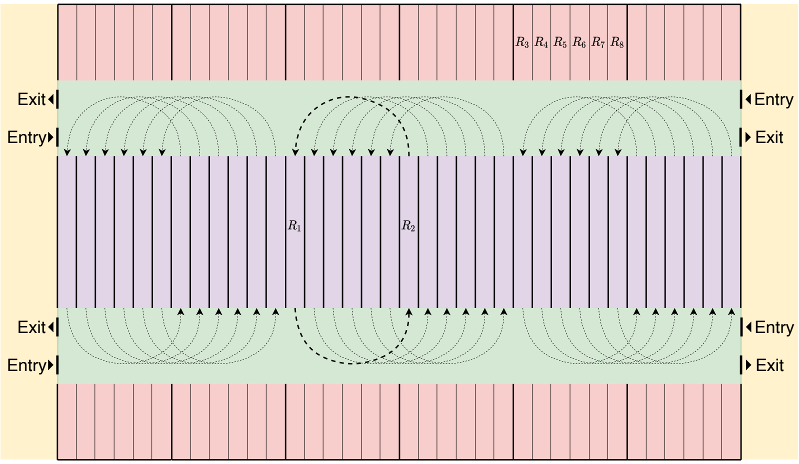
Fair Reservation Managers

- A reservation manager of an empty region *R* is **fair** if
  - When a parking manager adjacent to *R* repeatedly requests that a vehicle be sent to the empty region, the reservation manager will eventually approve the request.
- i.e., no request will be denied indefinitely.



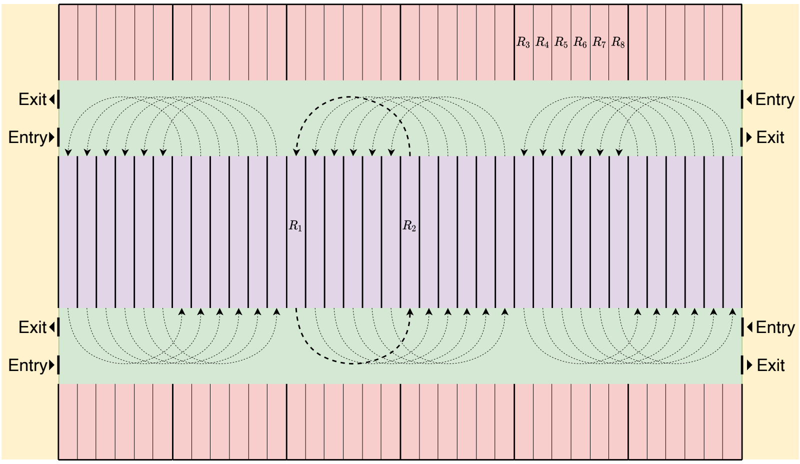
Sufficient Condition for Gridlock-free APLs

- **Theorem 4**: An autonomous parking lot whose parking regions are queue regions and stack regions is **gridlock-free** if
  1) All queue managers are 1-cyclable.
  2) All stack managers are greedily relocatable.
  3) All reservation managers are fair.
  4) All reservation managers deny any request that puts vehicles in a parking region that causes the parking manager neither 1-cyclable nor greedily relocatable.



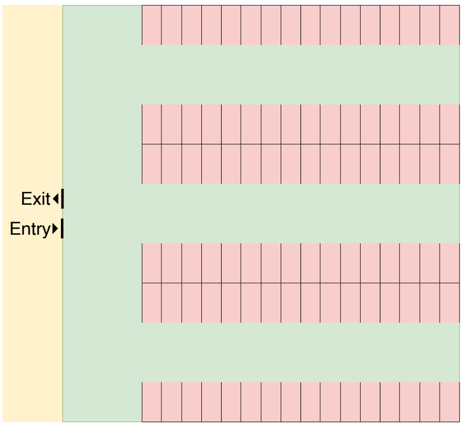
Sufficient Condition for Gridlock-free APLs

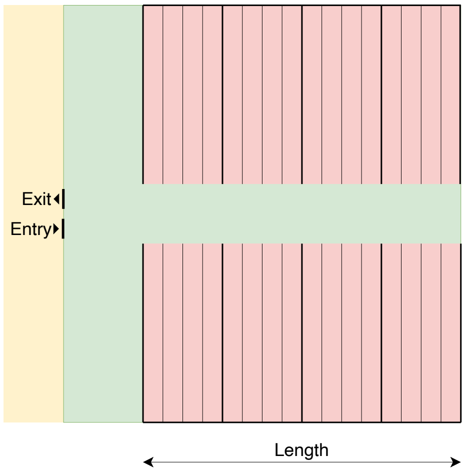
- **Sketch of Proof:**
  - All vehicles in an empty region *R* will eventually leave *R*.
    - Otherwise, the reservation manager won’t accept the reservation.
  - Both queue managers and stack managers can reorder any vehicle in their parking regions via its adjacent empty region.
    - Because they are always 1-cyclable or greedily relocatable, and reservation managers are fair.
  - Any vehicle in the parking regions can eventually enter the adjacent empty region that connects to an external region, therefore it can eventually leave the parking lot.

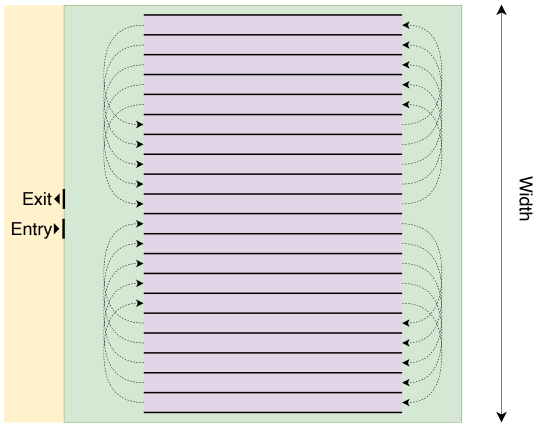


Experimental Evaluation

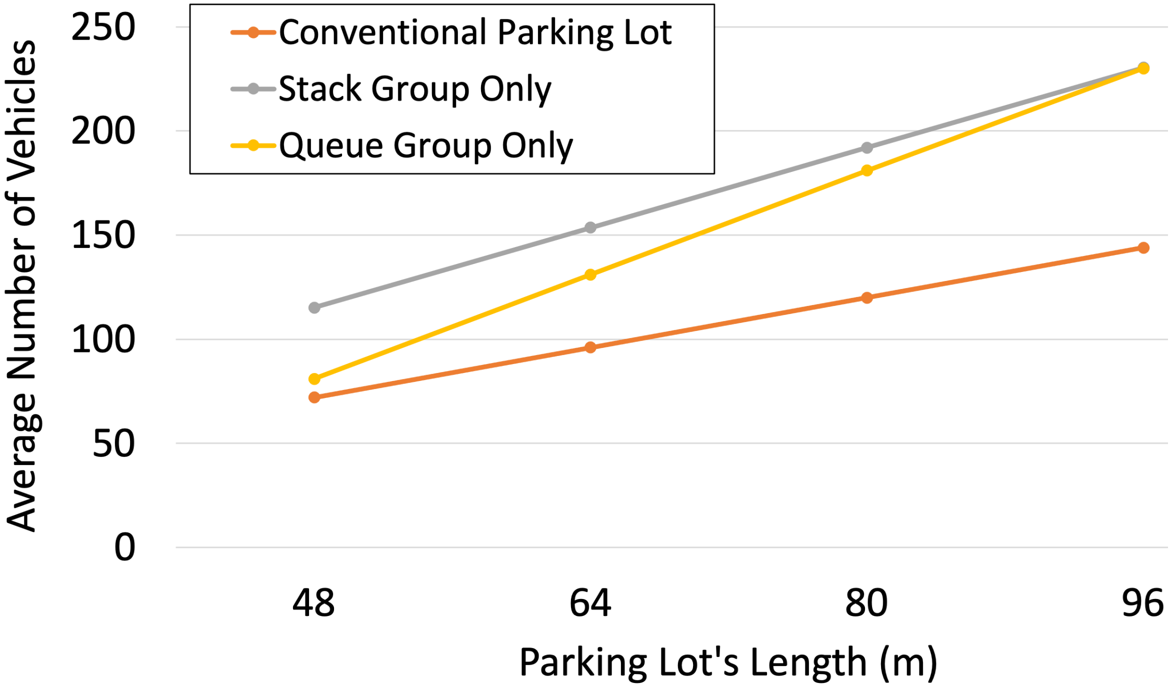
- **Hypothesis:** Autonomous parking lots are more space-efficient, but vehicles take a longer time to leave the parking lots.
- **Experimental setup:**
  - A simulator written in C++ and PyGame 2.0 (connected via gRPC).
  - The parking lot's width is 84 m, but the length of the parking lot varies from 48 m to 96 m.
  - The size of vehicles is random.
  - The vehicles' kinematic model is the unicycle model.


Conventional Parking Lot


Autonomous Parking Lot (Stack Group only)

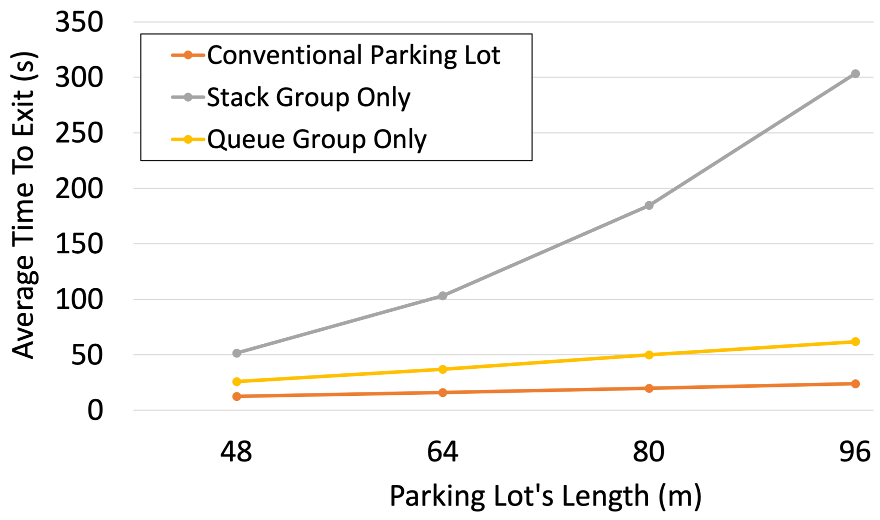

Autonomous Parking Lot (Queue Group only)

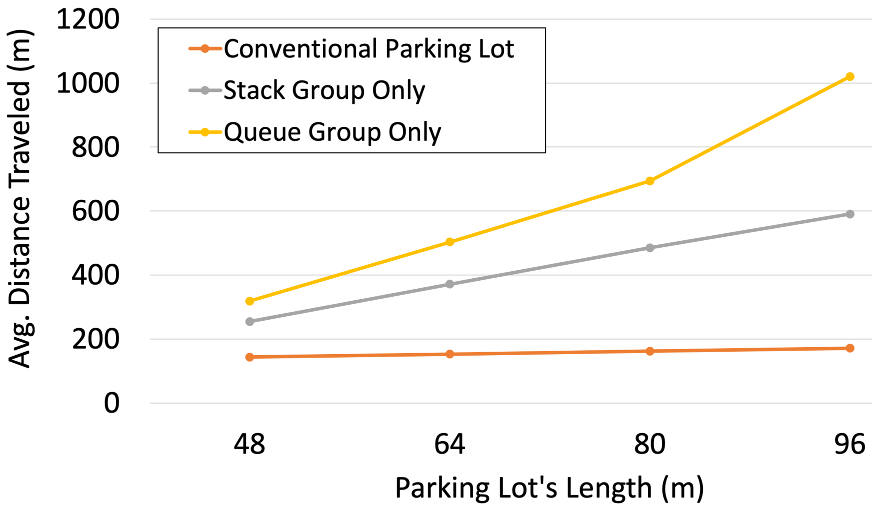
Experimental Results



- Autonomous parking lots are about 60% more space-efficient than conventional parking lots.
- The queue groups were not as space-efficient as the stack groups when the parking lot’s length is small
  - » As the parking lot’s length increases, the queue groups eventually become as space-efficient as the stack groups.

Experimental Results (cont.)





- Vehicles in the conventional parking lot took less time to exit the parking lot
  - Less movement inside the parking lot.
- Congestion occurred in stack groups.

Summary of Part 2

- **High-Density Parking (HDP)** utilizes autonomous driving to greatly increase the capacity of conventional parking lots.
    - » But existing works on HDP have not considered mixing different parking strategies and the gridlock issue.
- An **autonomous parking lot (APL)** uses reservation systems and managing agents to coordinate autonomous vehicles in HDP.
    - » guarantee that the parking lot is gridlock-free – no vehicle gets stuck and can never leave the parking lot when mixing different parking strategies.
- Our experiment confirmed an 60% increase in parking lots' capacity.

Conclusion and Future Work

Can autonomous driving help solve the most critical challenges facing modern transportation systems?

- Yes, it can.
- In the future,
  - design *a live and efficient* transportation network to cope with increasing demand for transport.
  - devise innovative parking solutions for autonomous vehicles.
  - utilize autonomous driving to unleash the full potential of highway systems.

Thank You!

Any question?