國立交通大學
National Chiao Tung University

TERM PROJECT

Iris Hui-Ru Jiang
Fall 2017

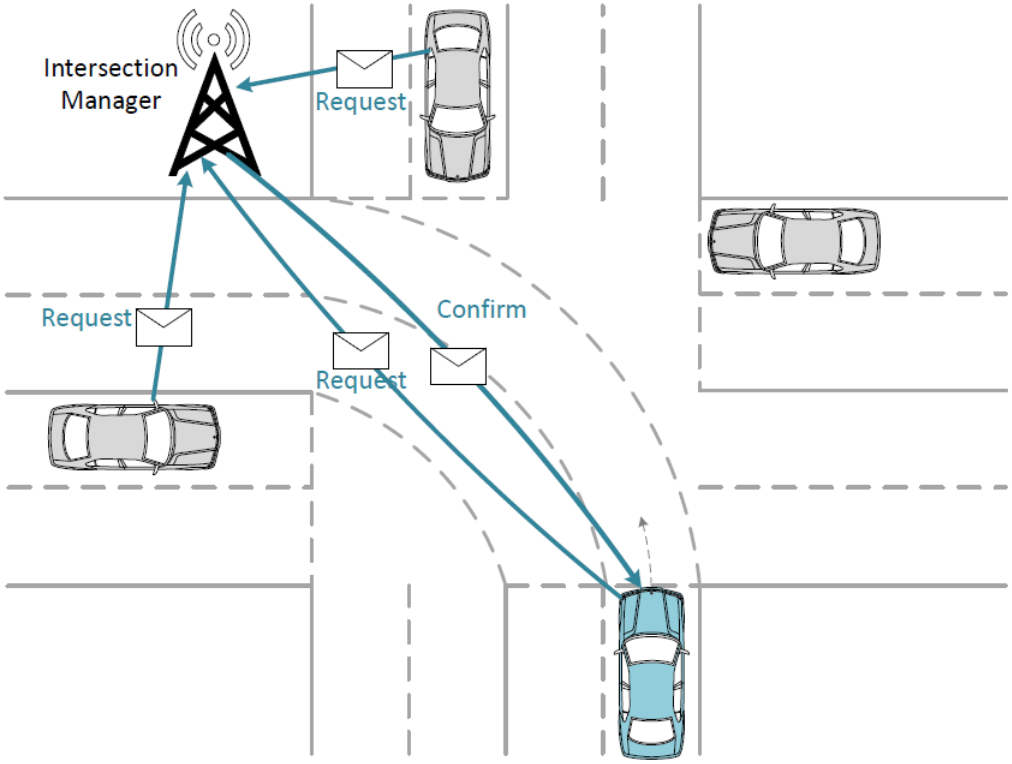
Department of Electronics Engineering
National Chiao Tung University

Special thanks go to Dr. Chung-Wei Lin

TOYOTA InfoTechnology Center

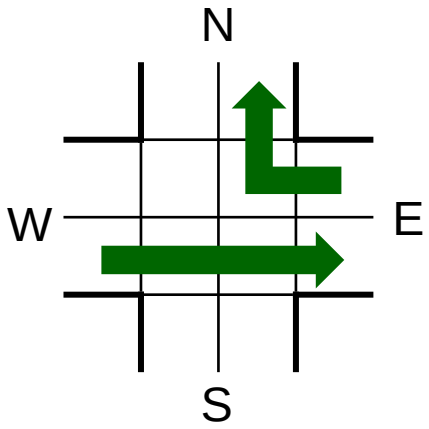
Intersection Manager

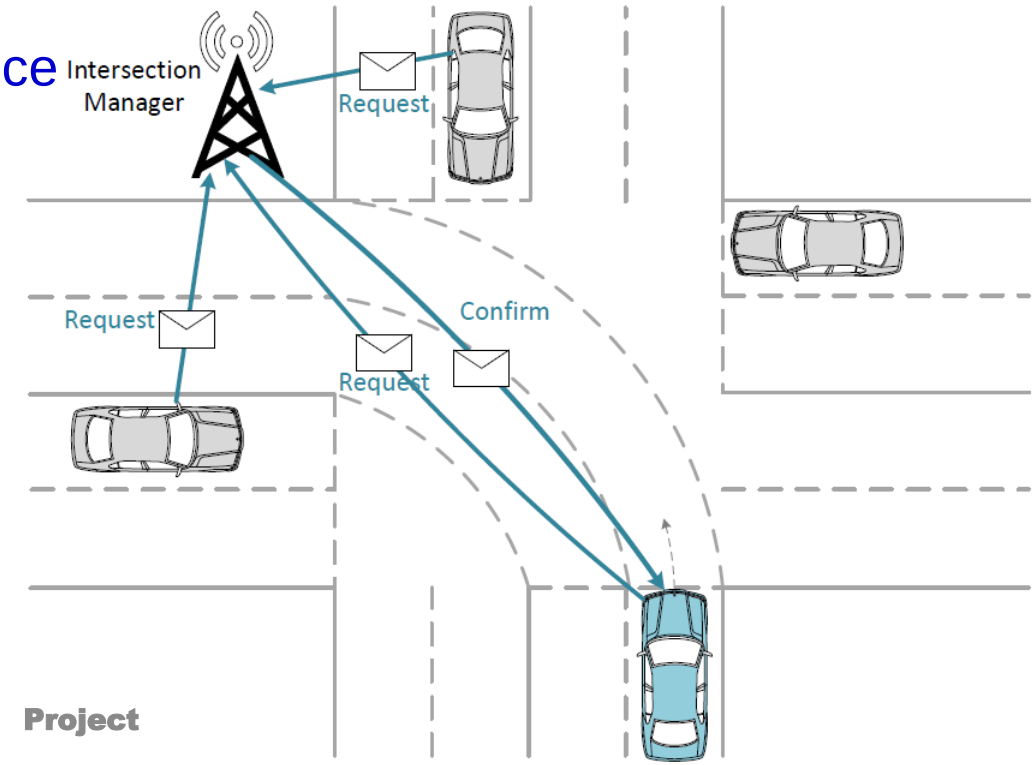
- Manage autonomous vehicles
  - Communication and scheduling



Intersection Manager (1/4)

- At each direction {N, E, S, W}, every 5s, there will be 0 or 1 car approaching the intersection.
- Each car has a destination direction {N, E, S, W}, and it will not be the same as the source direction. (No U turn!)
- Assumptions:
  - Arrival times given in advance
  - One lane for each way
  - The length of a round is 5s
  - Right-hand drive



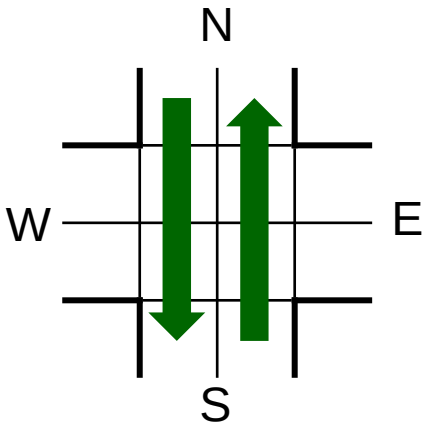


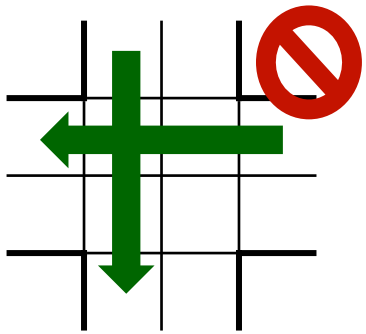
Intersection Manager (2/4)

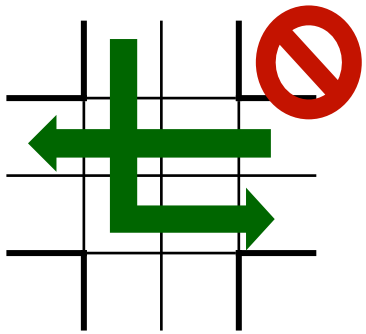
- Assume that the length of a round is 5s.
  - Within 1 second, the manager will schedule some cars to go through the intersection without "conflict paths" and inform those cars. It will schedule at most 1 car at each direction.
  - Those cars which get scheduled go through the intersection within 4 seconds.
- Those cars which do not get scheduled will wait the next round. However, a car may move forward if one of its front cars gets scheduled.
- Objective: Minimize the average waiting time of each car
  - No conflicts are allowed
- Note: if there is a car not scheduled at one direction, we can guarantee that there is a car at that direction (right before the stop line) in the next round.

Intersection Manager (3/4)

- Examples:
  - Two cars N-S and S-N arrive at 0s → Schedule both of them at 0s → Obj = 0
  - Two cars N-S and E-W arrive at 0s → Schedule them at 0s and 5s → Obj = 2.5s
  - Two cars N-S and E-W arrive at 0s + one car N-E at 5s → Schedule them at 0s, 5s, 5s → Obj = ((0-0)+(5-0)+(5-5))/3= 5/3s
    - One conflict!
  - Two cars N-S and E-W arrive at 0s + one car N-E at 5s → Schedule them at 5s, 0s, 10s → Obj = 10/3s







Intersection Manager (4/4)

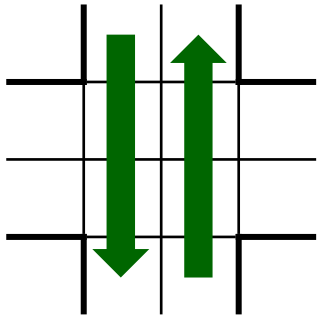
- Input
  - Each direction has a binary sequence: 0, 1, 1, …
  - Each “1” is associated with a source-destination direction: 1 (S-N)

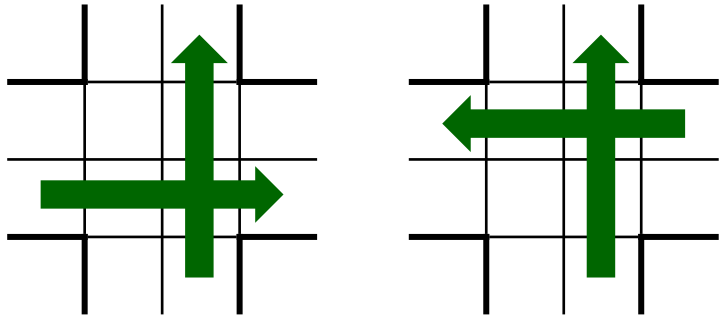
- Conflict vs. no conflict
  - Assume one lane for each way
  - S: straight
  - R: right turn
  - L: left turn
  - Check conflict table (consider symmetry/rotation)

Conflict Table (1/3)

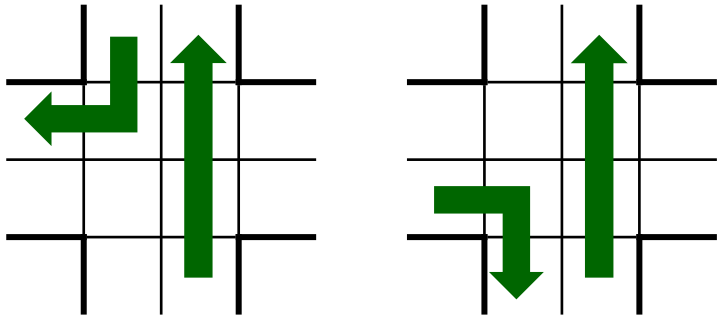
- **Conflict-free**
- **Conflict**

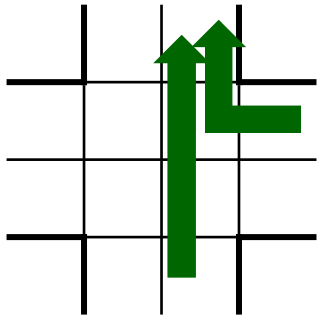
SS





SR

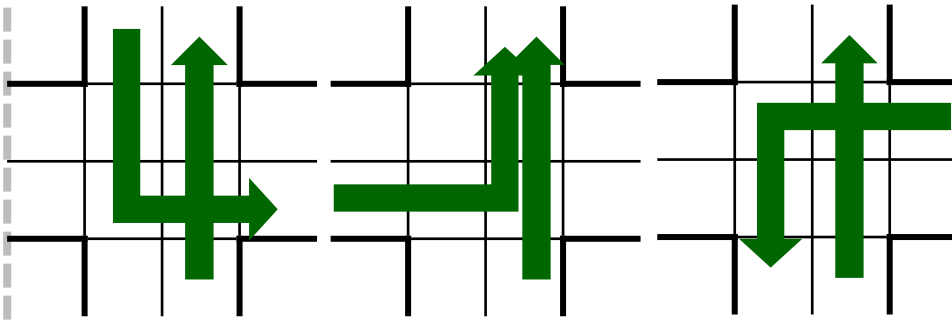




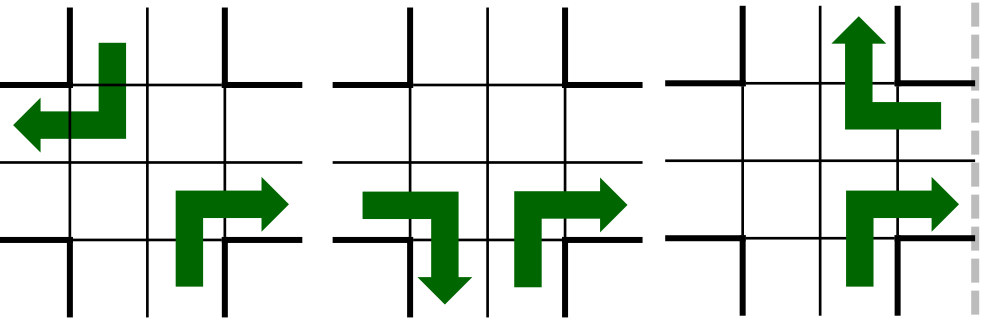
Conflict Table (2/3)

- **Conflict-free**
- **Conflict**

SL



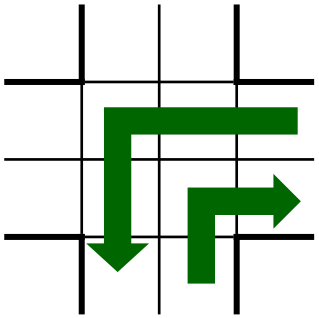
RR

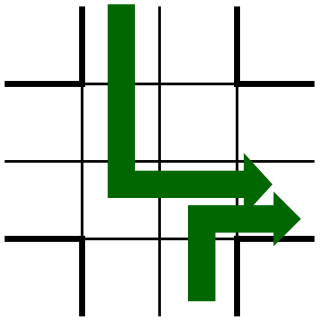


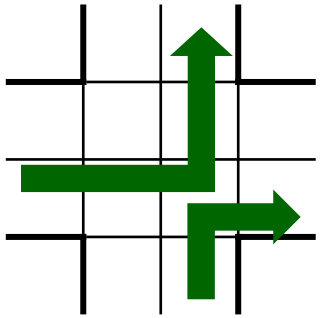
Conflict Table (3/3)

- **Conflict-free**
- **Conflict**

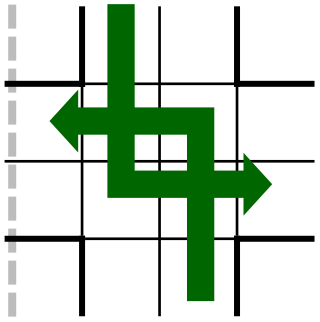
RL

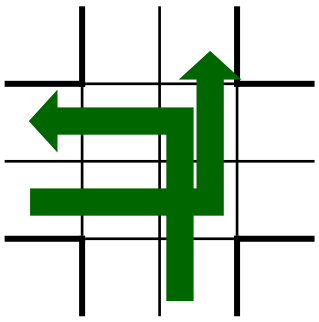


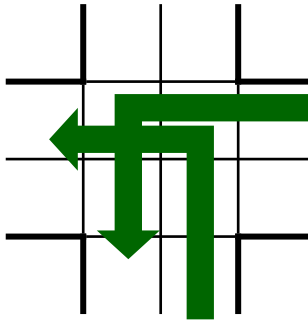




LL







Sample Input

- Four sequences: one for each direction (N, E, S, W)

- Input 1:
  - `N: 1S 1E 00`
  - `E: 1W 00 00`
  - `S: 00 00 00`
  - `W: 00 00 00`

- Input 2:
  - `N: 00 1E 00 00 1S 1E 00 00`
  - `E: 00 1S 1S 1N 00 00 1W 00`
  - `S: 1N 1E 1W 1W 1N 1E 1W 00`
  - `W: 1N 1E 1S 00 00 1N 00 1E`

Output Format

- Same as input format
  - Each direction: events listed in order but possibly with extra 00's

- Output 1:
  - `N: 00 1S 1E`
  - `E: 1W 00 00`
  - `S: 00 00 00`
  - `W: 00 00 00`

- Input 1:
  - `N: 1S 1E 00`
  - `E: 1W 00 00`
  - `S: 00 00 00`
  - `W: 00 00 00`

- Cost of Output 1:
  - `2/3`
  - The average number of waiting rounds
  - Difference between input and output sequences

More Cases…

Sample Input (More)

- Input 3:
  - `N: 1E 00 1E 00 1S 1E 00 00`
  - `E: 00 1S 1S 1S 1N 00 1W 00`
  - `S: 1E 1E 1E 1E 00 1E 1W 00`
  - `W: 1E 1E 00 00 1E 1N 00 1E`
- Input 4:
  - `N: 00 1E 00 00 1E 00 00 00`
  - `E: 00 00 00 1S 00 1N 1S 00`
  - `S: 1E 1W 1W 1W 1N 00 1N 1E`
  - `W: 1N 00 1N 00 00 1N 00 1E`
- Input 5:
  - `N: 1E 1W 1E 1S 1E 00 00 00`
  - `E: 1S 1S 1W 1N 1W 00 00 00`
  - `S: 1N 1E 1E 1W 1N 00 00 00`
  - `W: 1E 1N 1S 1E 1N 00 00 00`

Extensions

- From off-line to on-line
  - Consider k rounds at a time, k is a small number
- Multiple lanes in each direction
  - More flexibility
  - Update conflict table

Form your team today!

Project Proposal due Nov. 9

- List your team members
  - Everyone should submit one proposal
  - Members in one team submit the same one
- List your solutions for 5 sample inputs
  - Follow the output format

Term Project due Jan. 11

- Files:
  - Each team needs to submit the following materials:
  - (1) source codes
  - (2) executable binary
  - (3) a text readme file (readme.txt) stating how to build and use your programs.
- Scoring function:
  - Each case is individually evaluated by
  - (60%) The correctness of schedule: fully assignment without conflicts.
  - (15%) The solution quality
  - (15%) Runtime
  - (10%) Memory
  - Each term of quality, runtime, and memory is measured as follows:

  All teams are divided into 5 groups, top 20% teams get full points, the second 20% get 4/5 points, and so on.

Project Presentation on Jan. 8

We have gifts for best-presentation winners