Towards Green Transportation: Fast Vehicle Velocity Optimization for Fuel Efficiency

Chenxi Qiu*, **Haiying Shen**†, Ankur Sarker†, Vivekgautham Soundararaj‡, Mac Devine§, Andy Rindos§ and Egan Ford§

*College of Information Science and Technology, Pennsylvania State University
†Department of Computer Science, University of Virginia
‡Department of Electrical and Computer Engineering, Clemson University
§IBM Research

UNIVERSITY of VIRGINIA

Outline

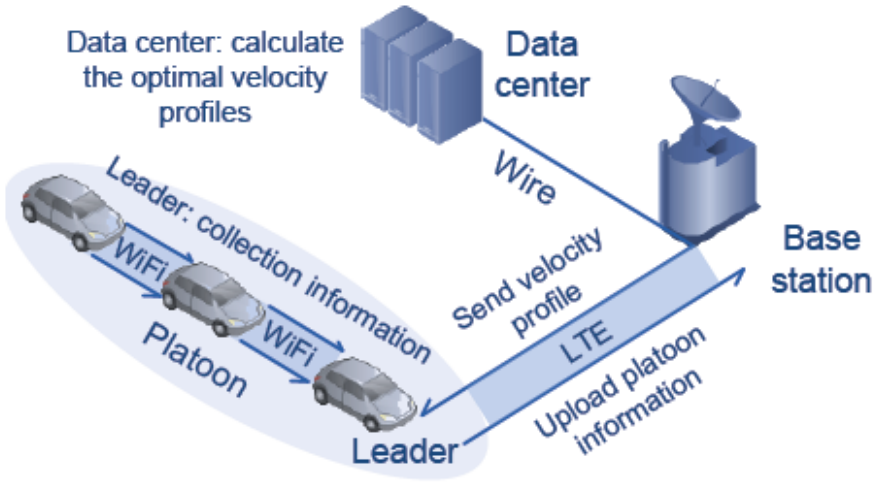
- Introduction
- System Design
- Performance Evaluation
- Conclusion

Introduction

The hierarchical structure of the vehicle network.

Two parts:

1. Congestion-avoidance information uploading
2. Traffic light considered parallel optimal velocity profile calculation



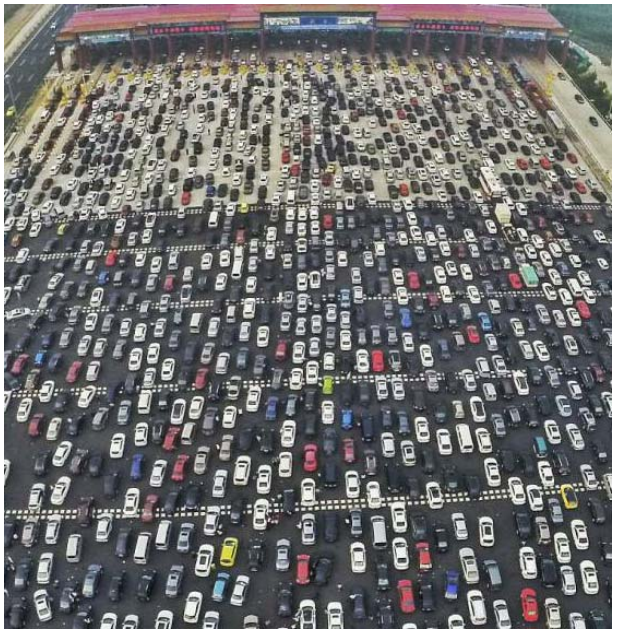
The hierarchical structure of the vehicle network.

Introduction

Green Transportation via velocity optimization

1. Road transportation is considered as one of the major sources of gas emissions
   - e.g., more than 13 million vehicles were sold in China in 2014

2. Vehicle optimization is one of the most efficient methods to reduce the fuel consumption
   - outputs the vehicle velocity profile
   - to prevent accelerations, which generate a higher fuel consumption



Introduction

Problems in previous methods

1. Expensive computation devices
   - e.g., road side unit
2. Only consider the local information
   - the next stop sign or traffic signal
   - ignore the global information
3. Optimal calculation is computation expensive
   - e.g., the next stop sign or traffic signal

Solution: vehicular cloud

1. All vehicles first upload their information to the cloud through base stations (BSs)
2. Based on the uploaded information stored in the cloud, the cloud derives the optimal velocity profile using DP

Related work

Vehicle networks

1. Ensure the packet delivery
   - [Son, AINA 2013] , [Shevade, CoNext 2011]
2. Avoid collision
   - [Chen, Mobicom 2013]

Speed optimization

1. Approach a traffic light at green whenever possible
   - [Asadi, Trans. CST 2011]
2. DP-based algorithm
   - [Ozatay, DCSS 2012]
3. Algorithms based on traffic and topographic information
   - [Park, FISTS 2011]

Introduction

Challenges of vehicular cloud

1. High queuing delay for information uploading at BSs caused by a large number of vehicles
2. The neglect of the traffic light and high computation delay for velocity profile calculation at the cloud

Our method: FastVO

1. Congestion-avoidance information uploading
   - to decrease the information uploading delay through decreasing the amount of information to be uploaded to the BSs
2. TrafficLight-considered Spark-based optimal velocity profile calculation
   - to quickly calculate the optimal velocity profile for each vehicle and also consider the traffic lights in DP

System Design

Congestion-avoidance information uploading

The objective: to reduce the amount of information that needs to upload to the cloud in order to reduce the information uploading delay.

1. Each vehicle in a group needs to report to the leader vehicle its vehicle ID and location through either single-hop or multi-hop routing
2. The leader vehicle maintains a group ID list and the group length.
3. A set of vehicles form a vehicle group iff they satisfy the following two criterions: all the vehicles are
   - connected with the leader vehicle through single-hop or multi-hop routing
   - running in the same lane
4. The leader vehicle periodically upload the leader coordinate, the velocity, and the group length to the cloud
5. The cloud can estimate the velocity and coordinates of all other vehicles in the group.

System Design

TrafficLight-considered Sparked-based optimal velocity profile calculation

Problem Statement:

Objective: to minimize the fuel consumption: $\min \sum_{t=0}^{T-1} c_{v_t, v_{t+1}}$

Constraints:

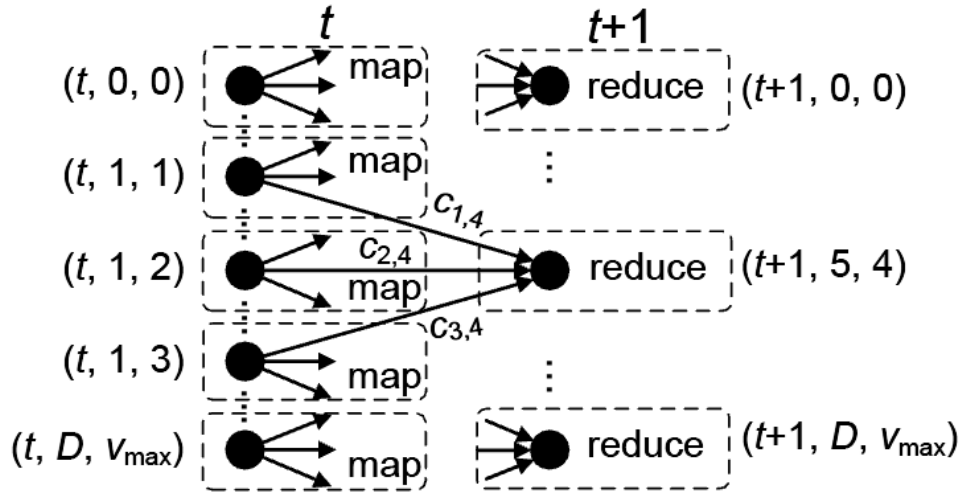
1. Speed limit constraint: $v_{\min}(d_t) \leq v_t \leq v_{\max}(d_t).$
2. Acceleration limit constraint: $a_{\min} \leq v_{t+1} - v_t \leq a_{\max}, \ \forall t = 0, 1, 2, \ldots$
3. Boundary constraint (its velocity is 0 when it starts and arrives at the destination): $v_0 = v_c \text{ and } v_T = 0.$
4. Stop sign constraint: $v_t = 0, \text{ if } v_{t-1} \neq 0 \text{ and } d_t \in \mathscr{D}_{\text{stop}}$
5. Red traffic light constraint: $v_t = 0, \text{ when } d_t = d_{\text{sig}}^l \text{ and } t \in \mathscr{T}_{\text{red}}^l.$

System Design

The Spatial-Temporal DP Algorithm

Decompose the ST-DP process to a set of independent parts;

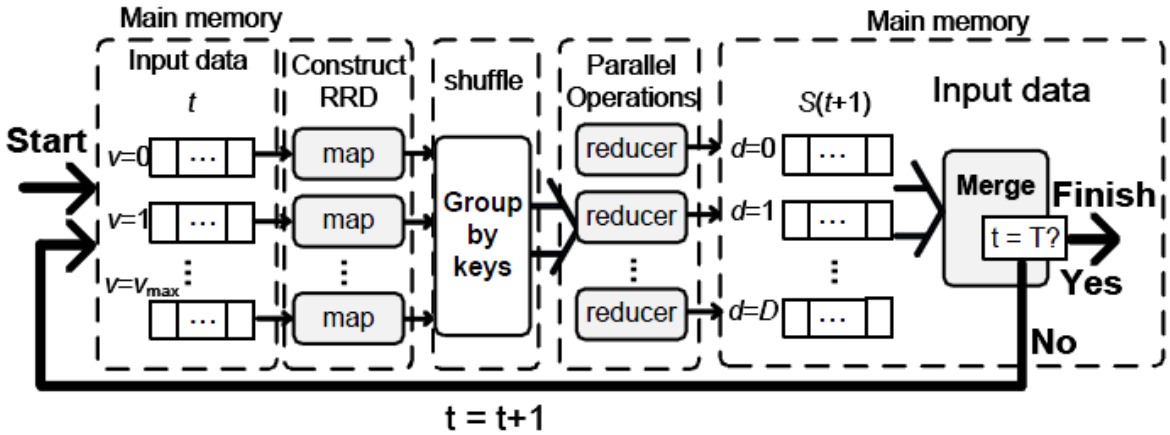
Design the map and reduce functions and the key for the output of the mapper;



ST-DP can be conducted as processing different task components in parallel (map function) and then combining the calculated results (reduce function), which are taken as the input in the next iteration if the program has not been finished.

System Design

Fast ST-DP Using Spark



The framework has two procedures: **map()** and **reduce()**

- Spark divides the input into chunks and then passes each chunk to a mapper. The key-value output pairs from each mapper are collected by a master controller and sorted by key.
- The keys are divided among all reducers, so all key-value pairs with the same key wind up at the same reducer. The reducer combines the data to produce a result, which will be taken as the input in the next iteration if the program has not been finished.

Experiment

Simulation settings

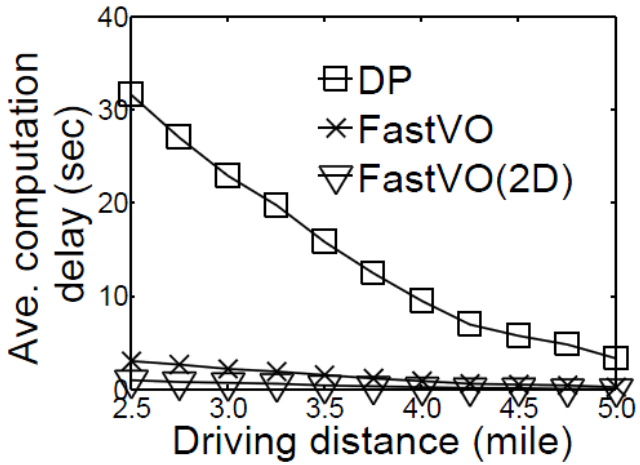
1. Both NS-2.33
2. Real vehicle mobility trace from San Francisco
    1. 30-day trace recorded by 536 taxis in San Francisco in May, 2008
    2. Each taxi driver has a tablet: timestamp, vehicle ID, GPS coordinate to a central server every 7 seconds.
    3. Randomly picked up a route with 5 miles length,
    4. Randomly picked up a vehicle and its 29 following vehicles in this route
3. The simulation takes 10 minutes
4. The transmission range of the traffic lights is 80 meters using IEEE 802.11p
5. Data center: Palmetto

Compared methods

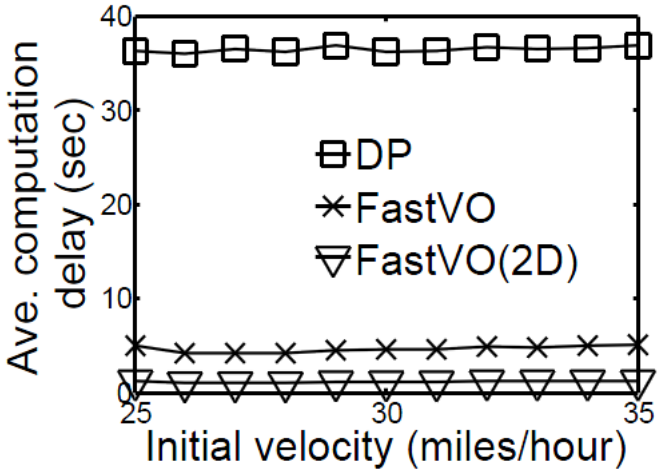
1. Predictive Cruise Control (PCC): each vehicle contacts each traffic light to get the traffic light information and calculates its optimal velocity profile
2. DP: use cloud to calculate the optimal velocity profile using DP
3. FastVO(2D): FastVO without considering the traffic lights

Experiment

Simulation results



(a) Different driving distances



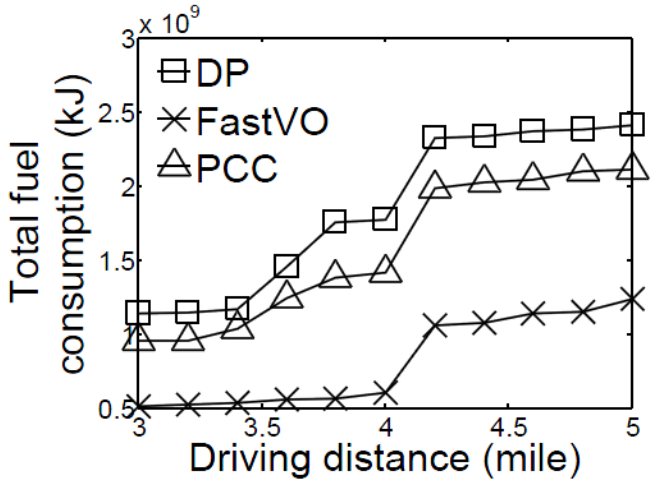
(b) Different initial velocities

Observation: the computation delay of DP is much higher than that of FastVO and FastVO(2D)

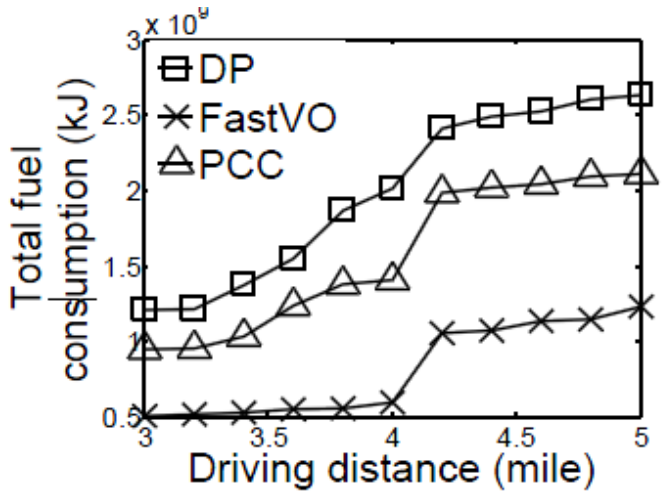
Reason: Parallel computing using Spark

Experiment

Simulation results



(a) Ideal environment



(b) Non-ideal environment

Observation: Fuel consumption follows: DP > PCC >> FastVO

Reason: DP has high consumption since it doesn't consider traffic light. PCC has high consumption since it only considers local information.

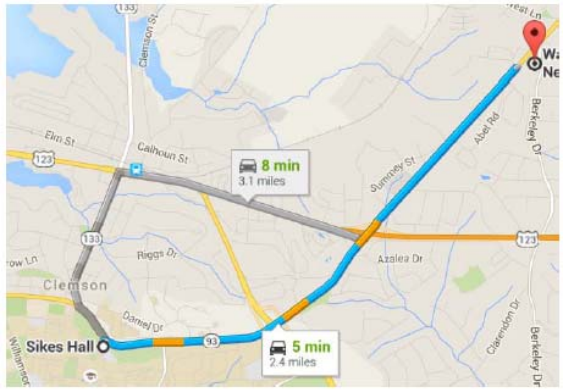
UNIVERSITY of VIRGINIA

Experiment

Real implementation settings

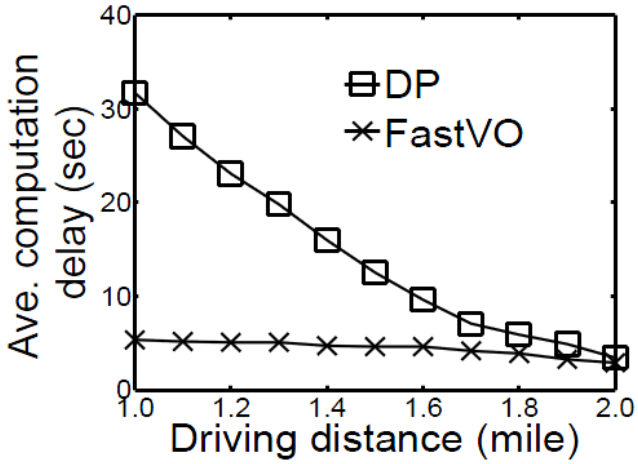
We built an Android application for mobile phones, equipped the phones in three cars, and drove these cars around a university campus to test the performance of different algorithms.

The route has a length of 2 miles, three traffic lights (located at 0.18 mile, 0.42 mile, and 0.63 mile from the source) and two stop signs (located at 0.56 mile and 0.74 mile from the source).

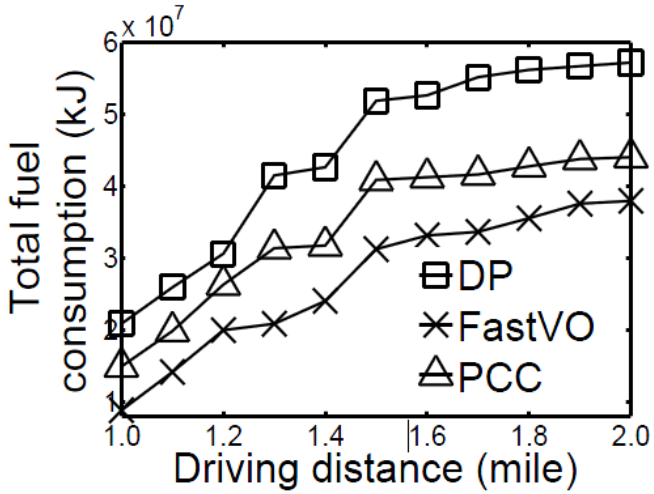


Experiment

Real implementation results



(a) Computation delay



(b) Fuel consumption

Observation:

1) The computation delay of DP is much higher than that of FastVO.
2) The fuel consumption follows: PCC > DP > FastVO

Consistent with the simulation results.

UNIVERSITY of VIRGINIA

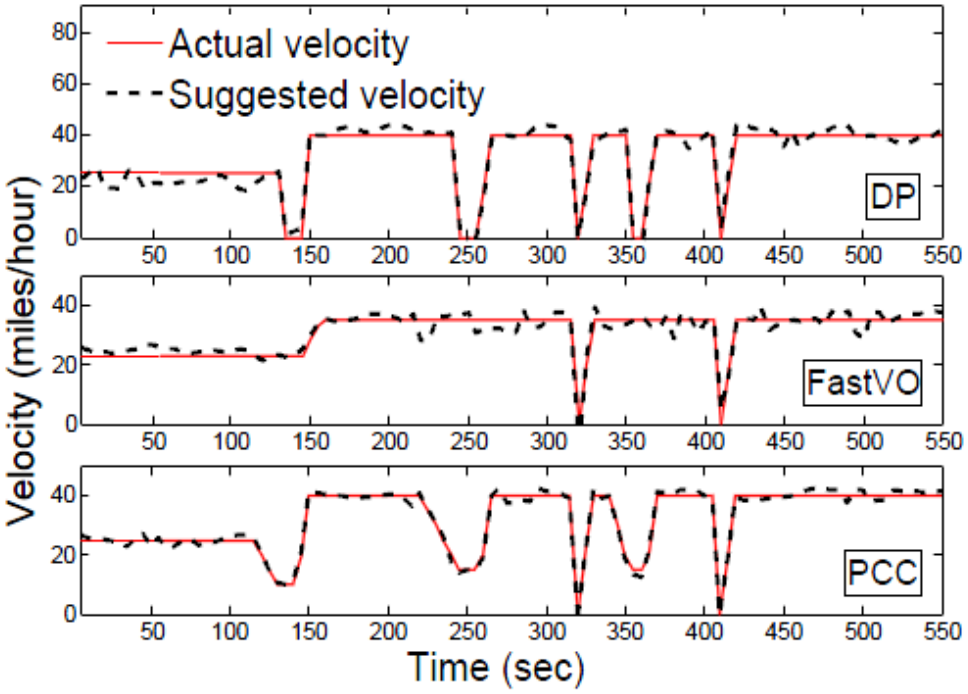
Experiment

Real implementation results

Compare the suggested velocity and the actual velocity of the leader vehicle in the three systems.

The results confirm that

1) vehicles cannot always follow the suggested velocity profiles in practice.
2) the FastVO has fewer accelerations compared with DP and PCC and hence generates less fuel consumption.



UNIVERSITY of VIRGINIA

Conclusions

1. We proposed to group vehicles within a certain range and let the leader vehicle in each group to upload the group information to the cloud, which then derives the velocity of each vehicle in the group.
2. We proposed spatial-temporal DP that additionally considers the traffic lights. We innovatively find that the DP process makes it well suited to run on Spark and then present how to run STDP on Spark.
3. We demonstrate the superiority of our method using both trace-driven simulation and real-world experiments.

Future work

Further take into account human and economic factors for velocity calculation

UNIVERSITY of VIRGINIA

Thank you!
Questions & Comments?

Dr. Haiying Shen

hs6ms@Virginia.edu

Associate Professor

Pervasive Communication Laboratory

University of Virginia