EMP: Edge-assisted Multi-vehicle Perception

Xumiao Zhang, Anlan Zhang†, Jiachen Sun, Xiao Zhu,
Yihua Guo‡, Feng Qian†, Z. Morley Mao

University of Michigan  †University of Minnesota – Twin Cities  ‡Uber Technologies, Inc.
Connected and Autonomous Vehicle

**Sensors**

- **LiDAR** = Light Detection And Ranging
  - LiDAR: point cloud
  - Object position, type, ...
  - Object future path
  - CAV future path
  - Throttling
  - Braking

**Perception**

**Prediction**

**Planning**

**Control**

- Camera: 2D image
- LiDAR: point cloud ...

---

Image source: https://www.extremetech.com/computing/305691-the-future-of-sensors-for-self-driving-cars-all-roads-all-conditions
https://steemit.com/technology/@rnjena/low-cost-solid-state-2d-lidar
Limitations of On-board Sensors

- They are vulnerable to occlusion.
Limitations of On-board Sensors

- They are vulnerable to occlusion.
- The farther an object is, the fewer details they can capture.

*Ego-vehicle: the vehicle collecting sensor data and perceiving the environment*
Limitations of On-board Sensors

- They are vulnerable to occlusion.
- The farther an object is, the fewer details they can capture.

* Ego-vehicle: the vehicle collecting sensor data and perceiving the environment
Benefits of Sensor Data Sharing

- Different vehicles perceive information from various locations
  - *objects occluded in the views of some vehicles can be easily perceived by others.*
- Driving scenarios where vehicles can benefit from sensor data sharing:
  1. Blind Spots
  2. Unprotected Left Turn
  3. Broken Down Vehicle
Limitations of Existing Solutions

- Sharing processed data [1,2]
  - *Limited data granularity: missed detections will still be missed after sharing*
    - Combining sensor data can lead to a higher resolution
  - *Lack of generality*
    - Raw data has a fundamental and universal format, compatible with various applications

Limitations of Existing Solutions

- Vehicle-to-vehicle sharing [1,2,3]
  - *Additional network overhead for sharing with different vehicles*
    - N vehicles → N-1 copies, N*(N-1) bandwidth consumption
  - *Additional computational overhead for processing data from others*
    - CAV hardware is originally equipped for processing single-vehicle data

(1) Number of Vehicles = 2

(2) Number of Vehicles ≥ 3

Need for an Edge-assisted System

- Offloading heavy computational tasks to an edge
  - *Edge: computing resources close to vehicles, providing low network latency*
  - *Advantages of using an edge*
    - Less network overhead: vehicles only need to share their sensor data to the edge
    - More computational resources: compared to a vehicle’s on-board hardware
Challenges

1. Bulky size of raw sensor data
2. Increased latency to process aggregated data
3. Network resource variability
   ○ Vehicles have different available bandwidths.*
   ○ Wireless networks fluctuate under high mobility.
4. Asynchronous data arrival

* Available bandwidth: the maximum throughput that an end host can achieve during data transfer
EMP (Edge-assisted Multi-vehicle Perception)

Vehicles
- Sensor
  - Navigation info
  - Point Cloud
  - Ground Removal
  - Partitioning
  - Compression

Preprocess

Planning and Control

Data

Control

Edge
- Vehicle Database
  - [location, bandwidth]
- Uploading Scheduler
- REAP
- Partitioning Decisions
- Perception Module
  - Decompression
  - View Merging
  - Object Detection
  - Detection Results

Adaptation Module
Point Cloud Partitioning

- Partitions the whole area into non-overlapping regions
  - Key idea: assigns each point to the closest vehicle
  - Voronoi diagram: partitioned by the perpendicular bisectors of connections between every two neighboring vehicles.

Point clouds of two vehicles (bird-eye view)
Point Cloud Partitioning

- Naive partitioning of point cloud through Voronoi diagram

What if A’s bandwidth is much lower than B’s?
Bandwidth-aware Partitioning

- Partition based on the vehicle locations and the estimated bandwidths
  - Key idea: uploaded area positively correlated to the estimated bandwidths
  - Power diagram (weighted Voronoi diagram)

Weights: \( r_1 \propto BW_A \), \( r_2 \propto BW_C \)
\[ R^2 = d_1^2 - r_1^2 = d_2^2 - r_2^2 \]

What if A’s bandwidth becomes lower than B’s?
Adaptation to Bandwidth Fluctuation

- Partition the data into multiple chunks with two additional boundaries
  - Consider Accurate/Overestimated/Underestimated bandwidth

(1) Vehicle A’s point cloud

(2) Vehicle C’s point cloud
Adaptation to Bandwidth Fluctuation

- Partition the data into multiple chunks with two additional boundaries
  - Consider Accurate/Overestimated/Underestimated bandwidth
  - Each vehicle sequentially uploads from chunk 1 to chunk 4

(1) Vehicle A’s point cloud

(2) Vehicle C’s point cloud
Upload Scheduling

- Upload finish conditions
  - $C_1$ & $C_2$

* $C_1$: chunk 1
Upload Scheduling

• Upload finish conditions
  • $C_1 \& C_2$
  • $C_1 + \text{neighbors’ } C_3$

* $C_1$: chunk 1
Upload Scheduling

- Upload finish conditions
  - $C_1$ & $C_2$
  - $C_1$ + neighbors’ $C_3$
  - neighbors’ $C_3$ & $C_4$

* $C_1$: chunk 1
Upload Scheduling

- Upload finish conditions
  - $C_1$ & $C_2$
  - $C_1$ + neighbors’ $C_3$
  - neighbors’ $C_3$ & $C_4$
- Check chunk delivery status upon receiving each chunk
View Merging

- A point cloud is generated from the perspective of the detecting vehicle
  - The origin is the LiDAR sensor mounted atop the vehicle.
  - Point clouds collected by different vehicles have different coordinate systems.
- The edge merges the views of different vehicles
Evaluation - Experimental Setup

- EMP prototype in Java: [https://github.com/Shawnxm/EMP](https://github.com/Shawnxm/EMP)
- Emulation testbed: EMP-edge instance + multiple EMP-vehicle instances
Evaluation - Experimental Setup

- Network conditions
  - *Trace collection*
    - Saturate the link with UDP data upload when driving at urban and rural areas
    - Measure the actual network throughput
  - *Network types*
    - LTE cellular networks (AT&T)
    - 60GHz WiFi networks (802.11ad, also considered in [1])
    - *Replay traces over Ethernet with Linux tc throttling the bandwidth*

Evaluation - Experimental Setup

• Sensor (LiDAR)
  • Modify an existing tool* for generating driving data in a video game (GTA V)
  • Collect the **first** multi-vehicle dataset with panoramic LiDAR point clouds

* DeepGTAV-PreSIL: https://github.com/Shawnxm/DeepGTAV-PreSIL/tree/modified_for_emp
System Scalability

- Compare the end-to-end latency of four schemes
  - EMP outperforms V2V sharing schemes by 49-65% in end-to-end overhead
  - Partitioning and scheduling effectively reduces latency

![Graph showing latency vs. number of vehicles for different schemes](image)
Evaluation - Experimental Setup

• Real-world driving test
  • *One machine runs the EMP-edge instance*
  • *Multiple vehicles each carries a laptop running EMP-vehicle instances*
System Scalability

- Real-world driving tests
  - *The latency does not inflate when increasing the number of vehicles*
  - *REAP helps reduce the processing delay*
Perception Enhancement

• Object detection accuracy
  • *Single-CAV (CAV) < Multi-CAV (EMP) < Combined (Edge+CAV)*
  • *REAP introduces negligible performance degradation while saving bandwidth*
Road Hazard Avoidance

• Blind Spots (camera images)
Road Hazard Avoidance

- **Blind Spots (visualized point clouds):** save 0.6s
  - *The blocked vehicle can be detected in both 2-vehicle setups*

\[
\text{Frame 0: 1-vehicle} \quad \text{Frame 0: 2-vehicle} \quad \text{Frame 0: 2-vehicle (REAP)}
\]

\[
*0.18 - (0.2 \text{ processing} - 0.063 \text{ inference} + 0.051 \text{ transmission}) = 0.6s*
\]
Conclusion

Thank you!

• Propose EMP, an edge-assisted multi-vehicle perception framework
• Develop robust algorithms for scalable, adaptive, and resource-efficient sensor data sharing under fluctuating network conditions
  • A point cloud partitioning algorithm with bandwidth adaptation
  • A graph-based upload scheduling algorithm
• Implement the first LiDAR-based cooperative perception system
  • Outperforms V2V sharing schemes by 49-65% in end-to-end overhead
  • Reduce network bandwidth by 36-43% by adaptively uploading sensor data
  • Demonstrates its benefits of improved perception in realistic driving scenarios