CarSpeak: A Content-Centric Network for Autonomous Driving

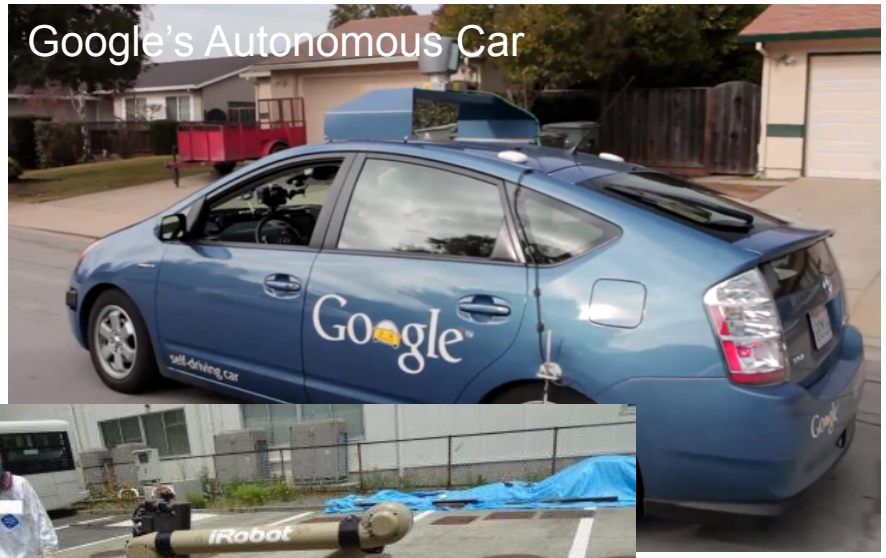
Swarun Kumar, Lixin Shi, Stephanie Gil, Nabeel Ahmed, Dina Katabi and Daniela Rus, MIT

SIGCOMM 2012

Presented by Fabián

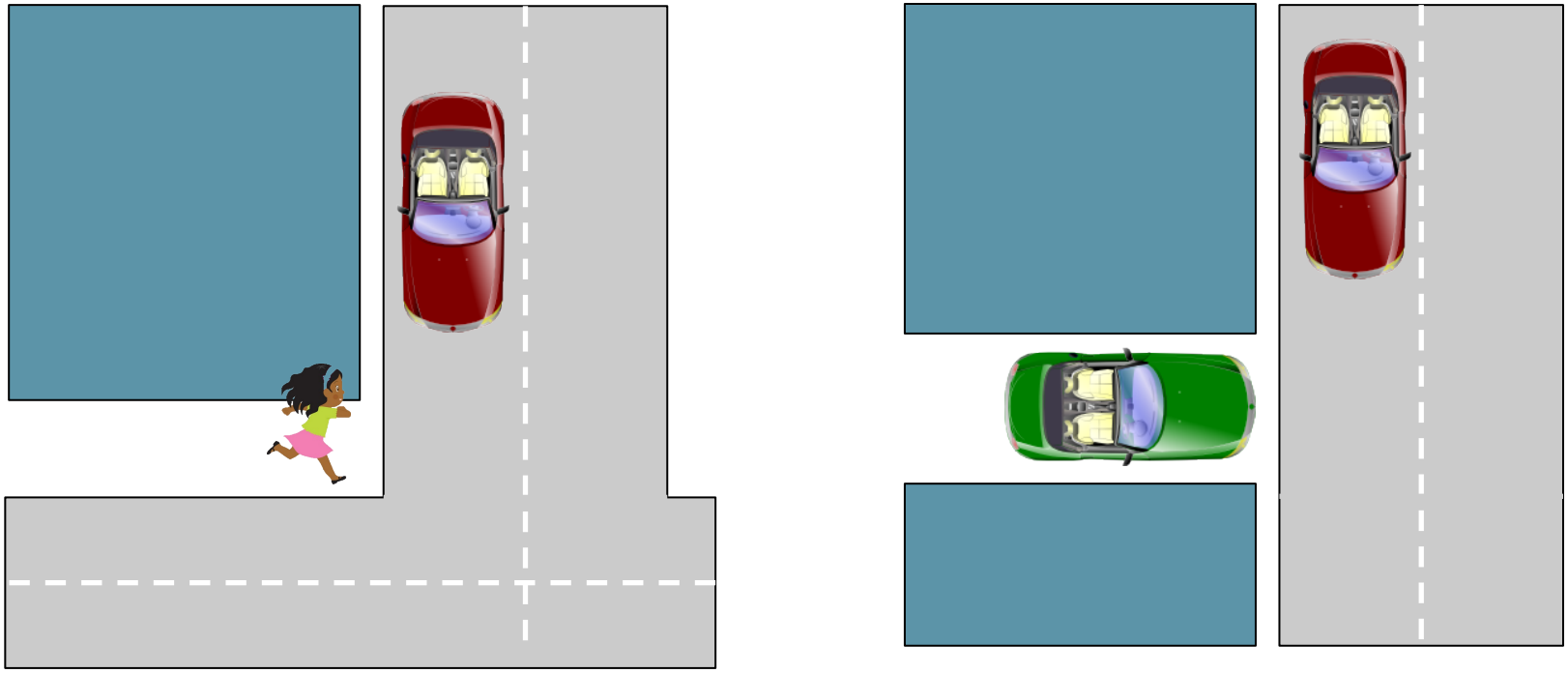
Slides based on the authors' presentation

Much interest in autonomous vehicles



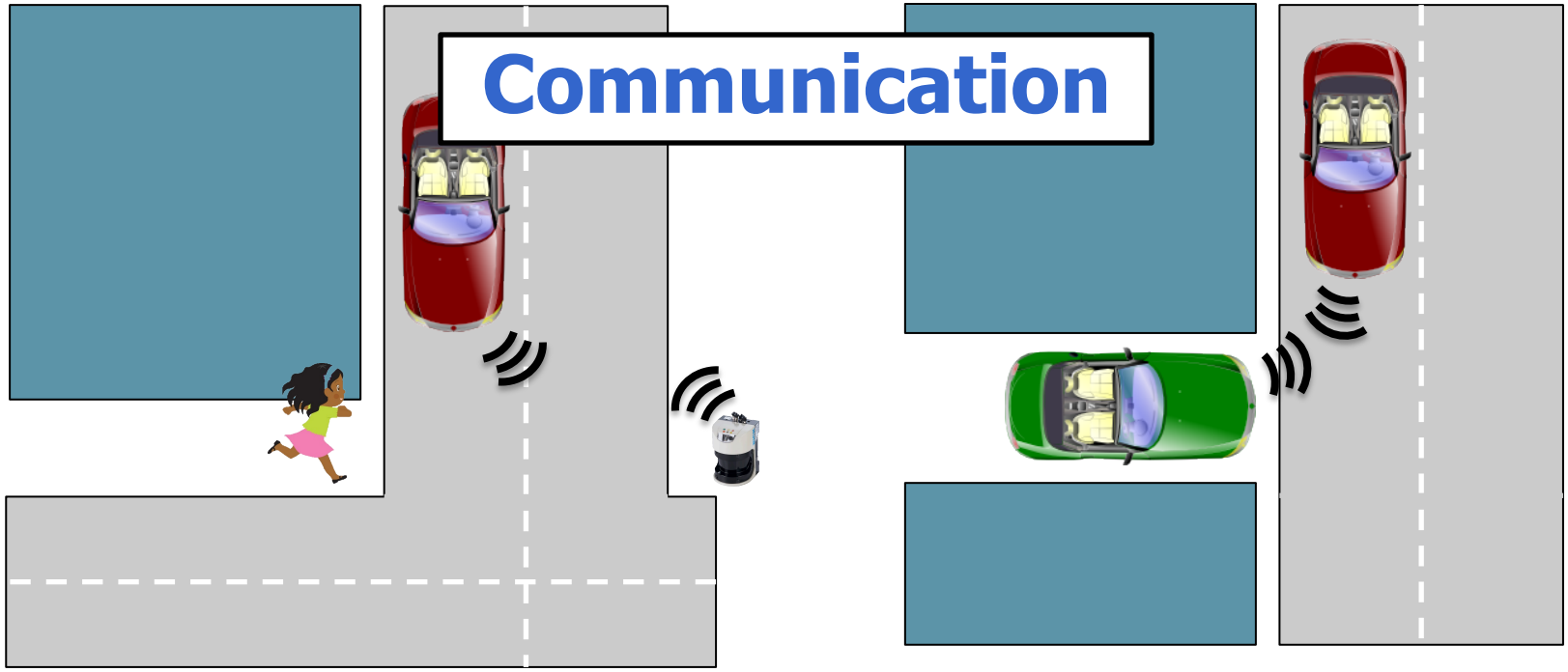
Lower traffic congestion, higher fuel efficiency, improved productivity, …

Challenge: Safely Detecting Hidden Objects



- Sensors on a car see only line of sight objects
- Hidden objects affect autonomous cars
  - "Google's autonomous car requires occasional human intervention to prevent accident"

How can they detect hidden objects?



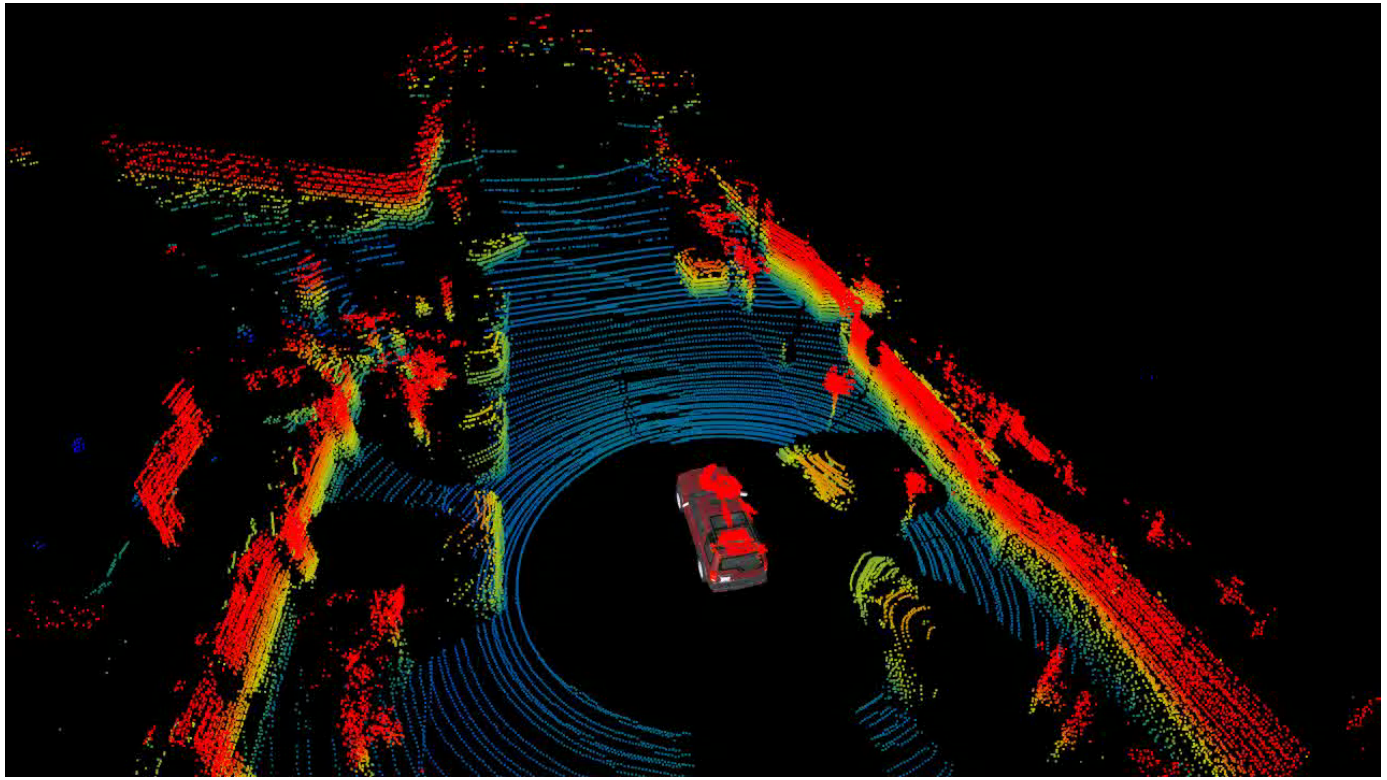
- Expand field of view beyond line of sight
- Also valuable for human drivers – can react faster to objects they couldn't see

Simply use past work on VANETs?

- VANETs typically oblivious to application
  - Efficient routing
  - Reliable message delivery
  - … (etc)
- But, not integrated with specific applications

Needs tight integration with communication

- Data is huge and time critical
- Communication should focus on information most critical to the application



Needs tight integration with communication

- Don’t know who has the desired content
  - In typical networks, you know your destination
  - Instead, autonomous car seeks sensor data from part of the road, e.g. an intersection
  - It doesn’t know which car has this information
  - Focus on content as opposed to accessing a particular destination

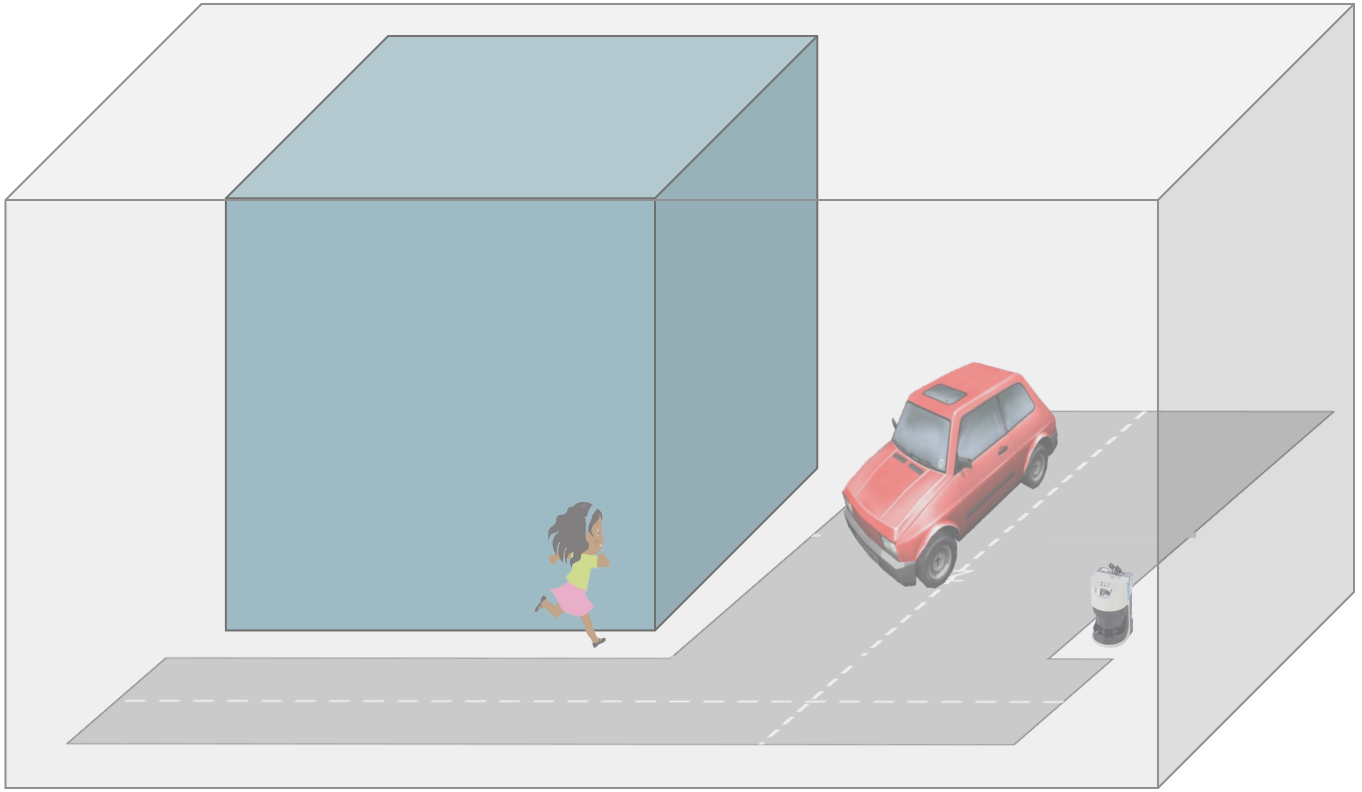
CarSpeak

- Integrates communication with path planning and navigation in autonomous vehicles
- Has a content centric design
  - Content, i.e. parts of road, is a first class citizen
- New MAC design where content, not senders, contend for the medium
- Implemented & evaluated on real vehicles

- Key issues
  - What is “content” and how do we represent it?
  - How do we disseminate this content?

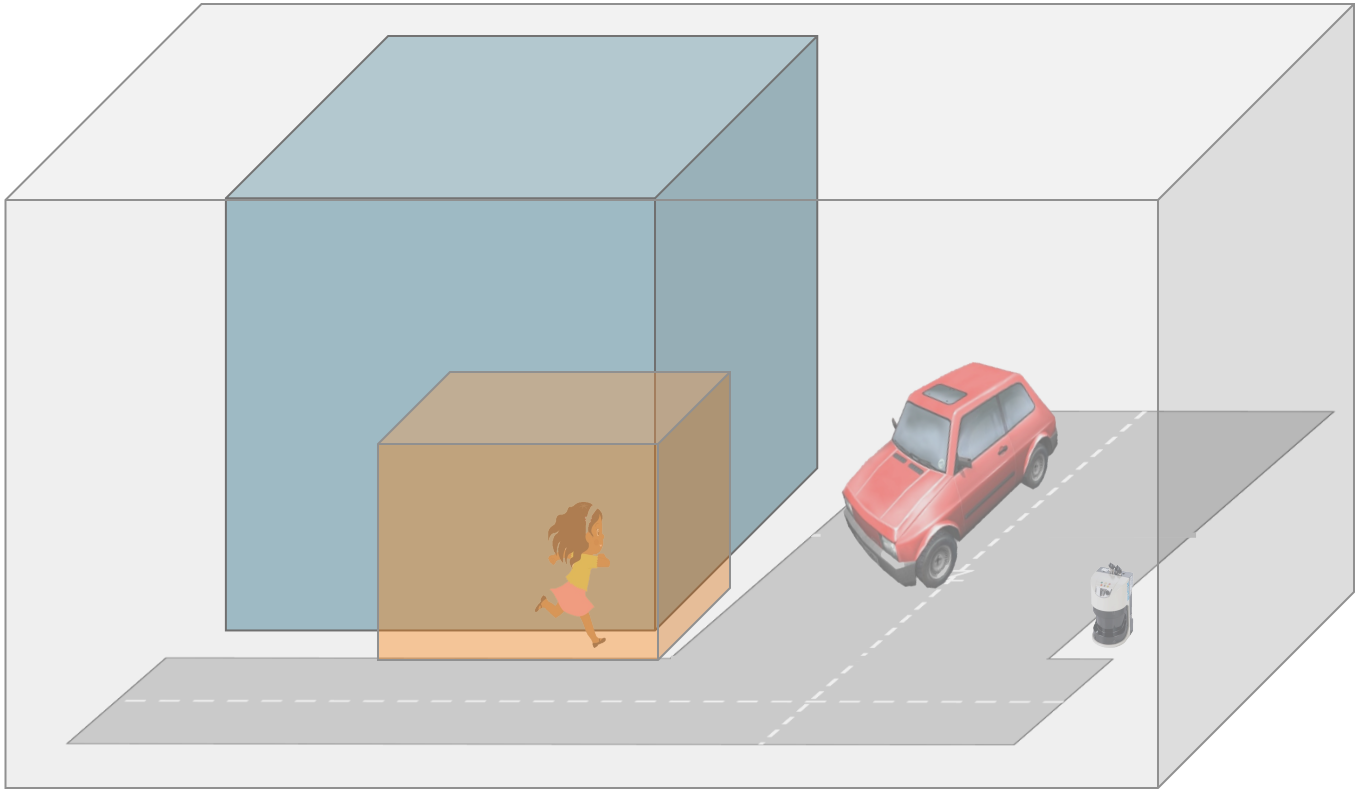
Content and content representation

- Content is sensor data from cubes in the environment
- How do we represent these cubes in environment?



Ideally…

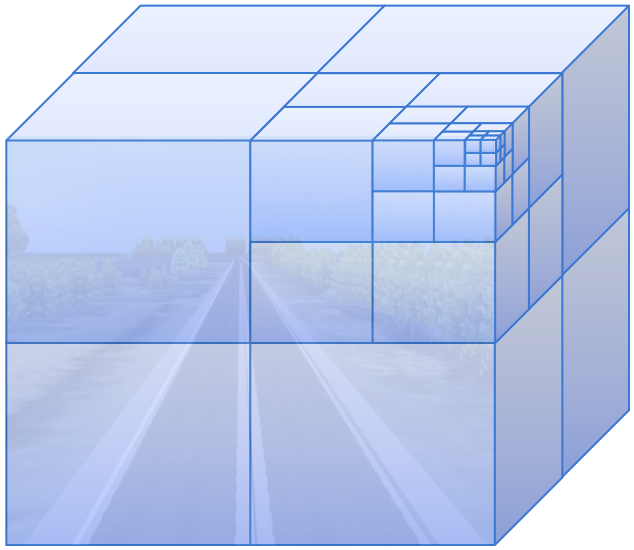
- Obtain low resolution view of environment
- Zoom in for higher res view of a smaller part of env

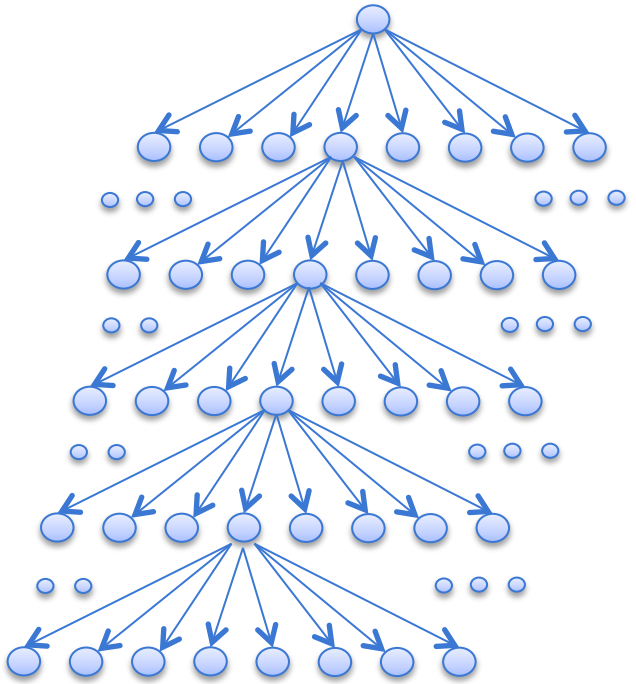


- Need recursive representation that makes best use of available wireless bandwidth

Representing content in CarSpeak

- Consider large cube encompassing environment
- Recursively divide into 8 smaller cubes

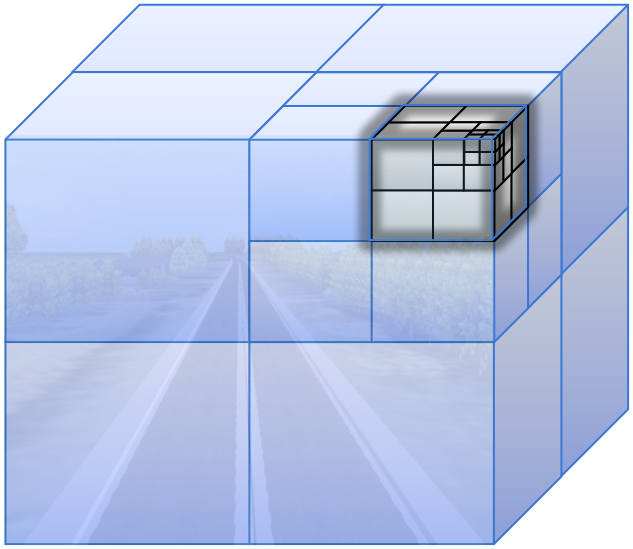




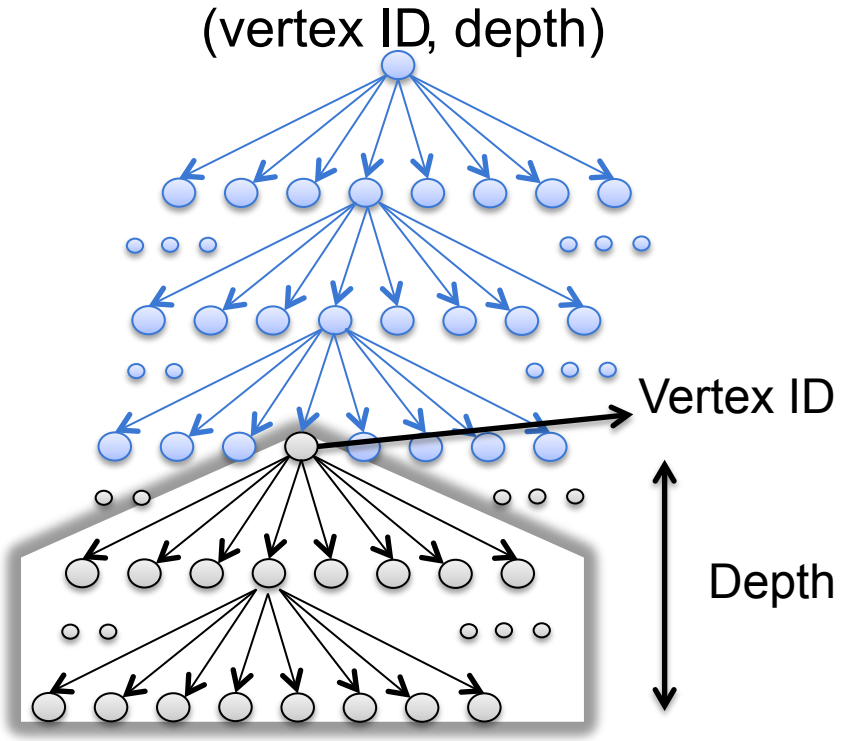
Tree Representation

Representing content in CarSpeak

- Consider large cube encompassing environment
- Recursively divide into 8 smaller cubes
- Car needs cube at resolution



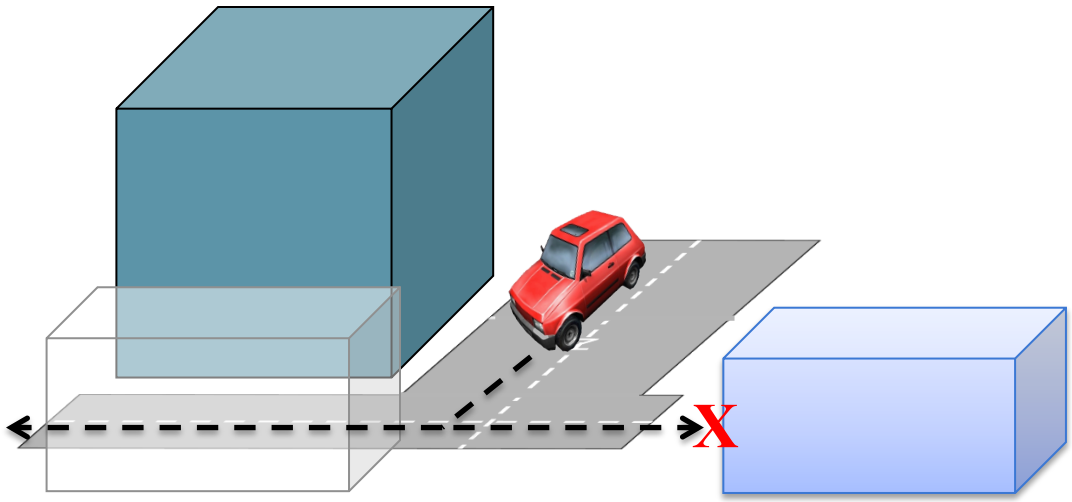
Cubes



Tree Representation

What info do autonomous cars need?

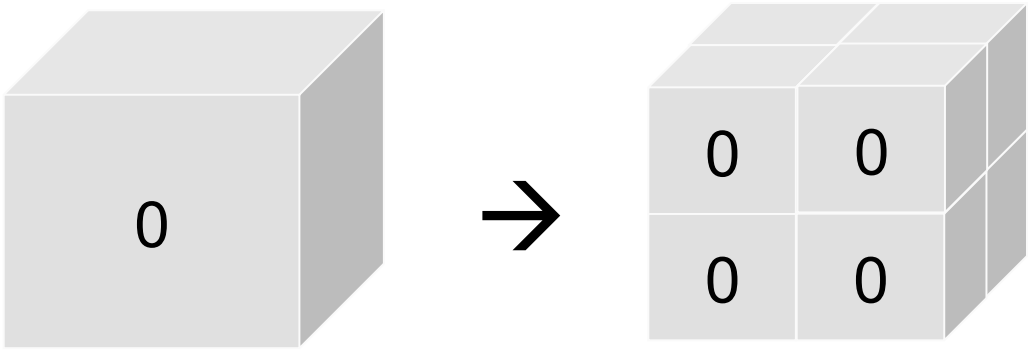
- Looks for obstacle free paths to destination
- Needs to know which parts of environment:
  - Are empty and safe to pass through
  - Are occupied and unsafe to pass through



What info do autonomous cars need?

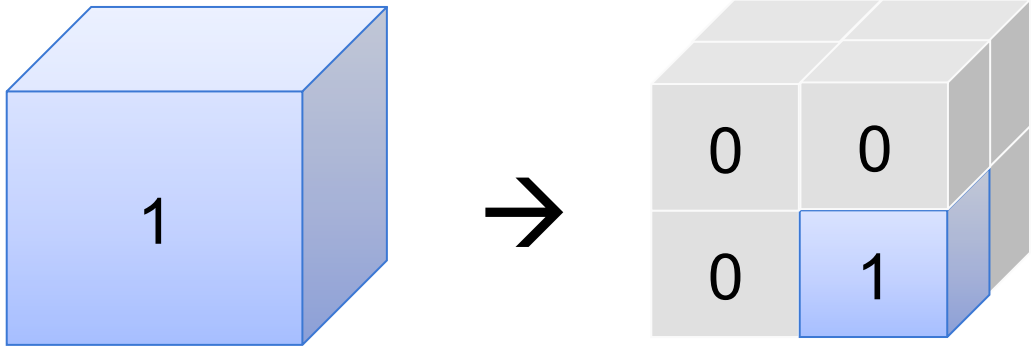
- Each cube has one bit: Empty (0) or Occupied (1)
- If cube is empty

  → all cubes inside are empty



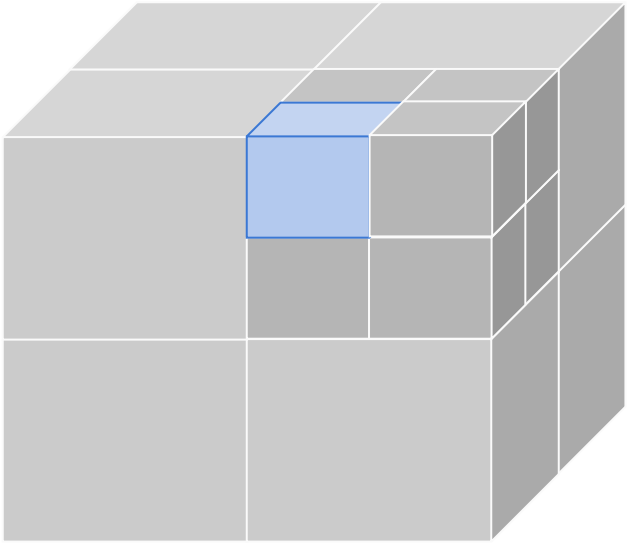
- If cube is occupied

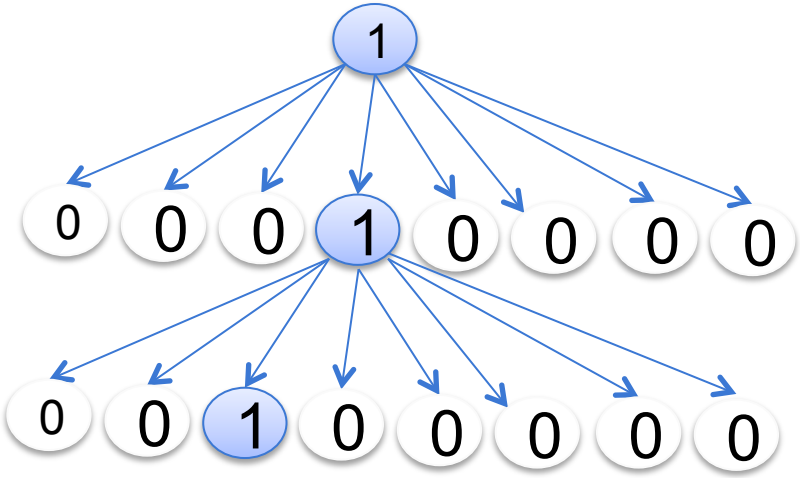
  → at least one cube inside is occupied



Compactly representing data

- Level 1 has 8 bits where 0-empty, 1-occupied
- None of 0 nodes need to be expanded
- Expand 1 node to see inside at more resolution





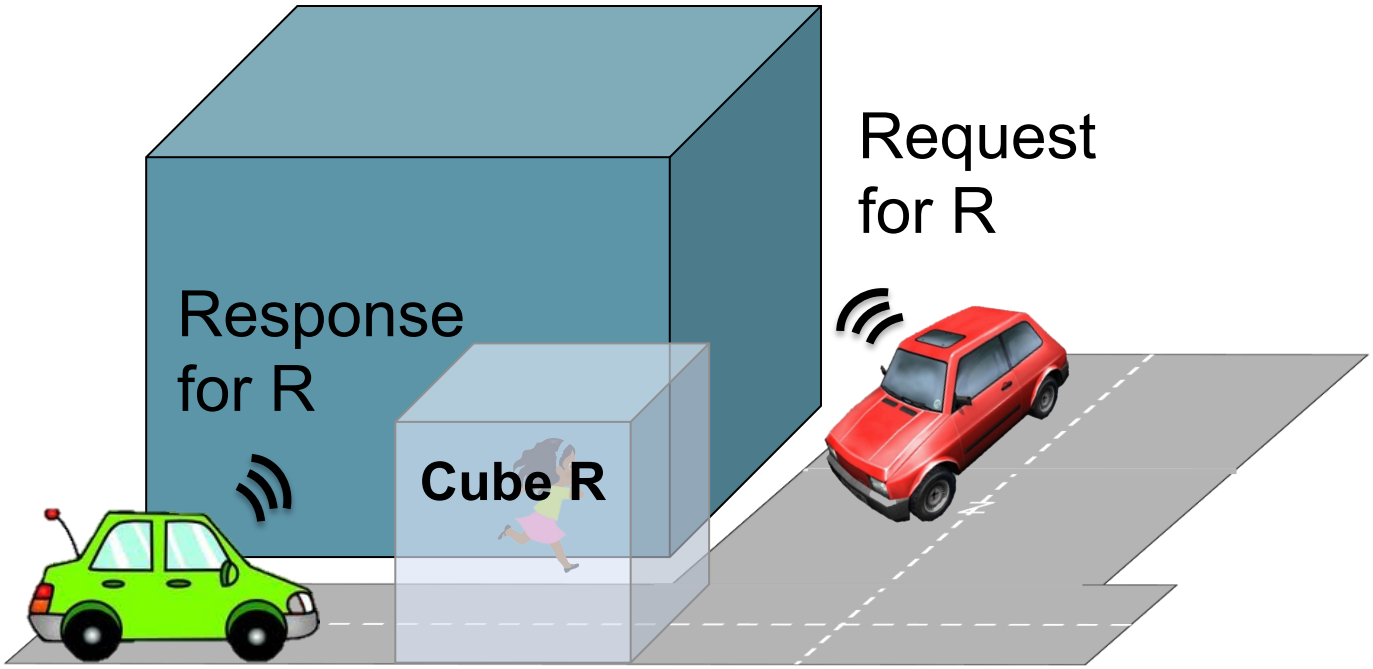
- Tree representation compresses data efficiently

How do we disseminate this content?

- Autonomous cars collect huge amount of data
  - Cannot flood medium with all their data

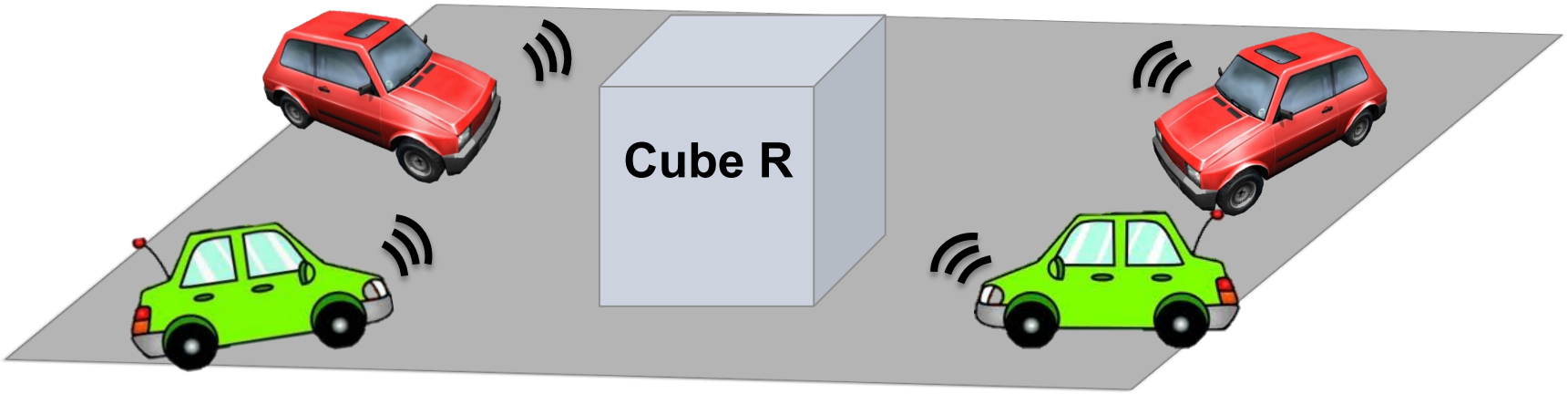
A request-response approach

- Car requests only data of interest
  - E.g. at blind spots, intersections, etc.
- Cars which sense the data, may respond



Challenge 1: Who should respond?

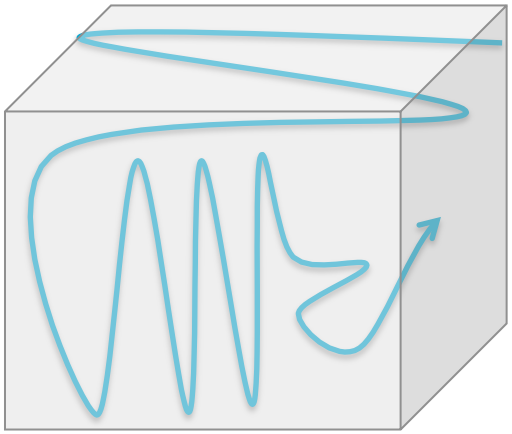
- Naïve solution 1: Simply let all cars respond
  - A lot of redundant data

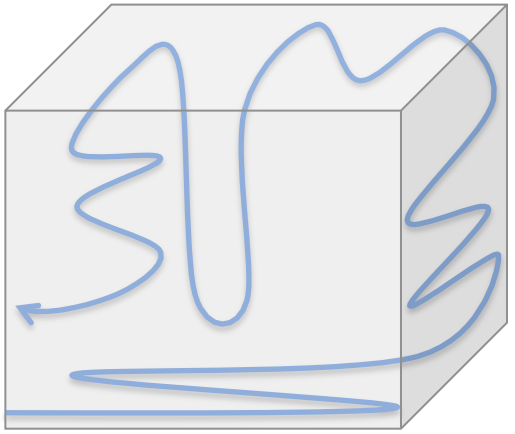


- Naïve solution 2: One car respond; others who hear it suppress their response
  - Responder leaves before sending all packets
  - Different cars have different perspectives
- **Need to balance data diversity with data overlap**

Solution: Random Walks

- Content of the cube (i.e., its subtree) is divided into packets
- Each car uses a different random walk to transmit packets



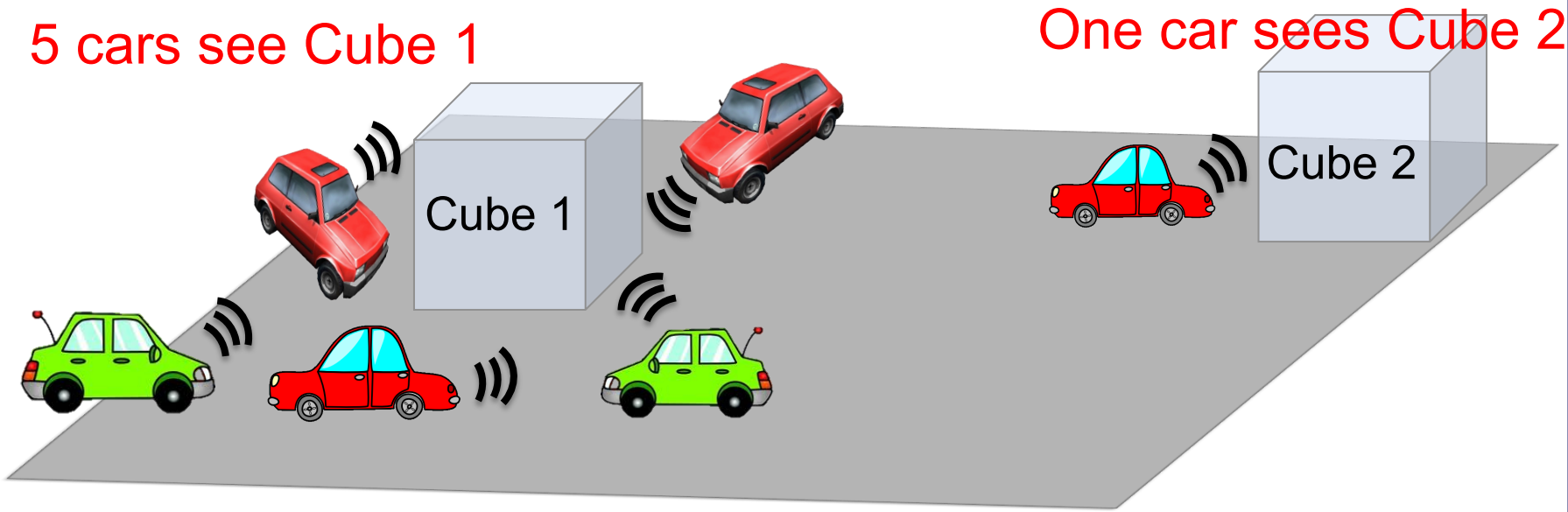


If one car transmits → Eventually finishes walk

If multiple cars transmit → Overlap is minimum

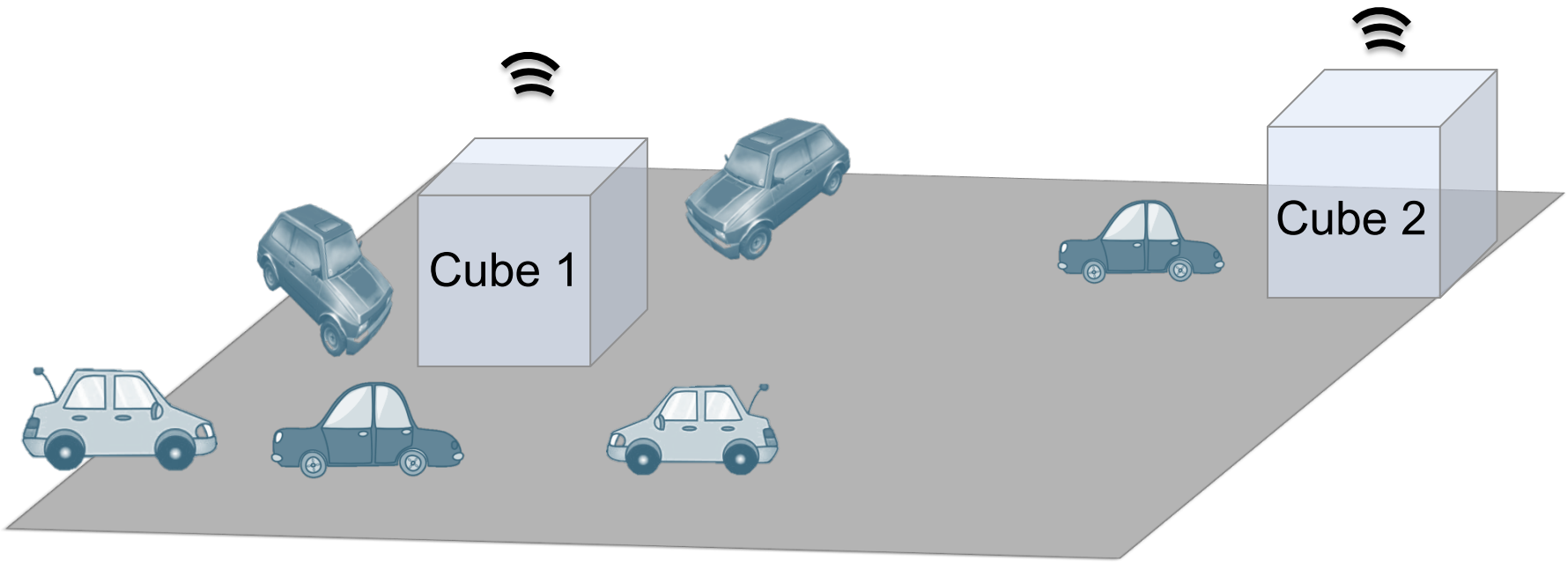
Challenge 2: Sharing the medium

- Shared medium by fairly by requested content
- 802.11 shares medium between senders
  → Cube 1's share = 5 x Cube 2's share
- Ideally, we want a MAC where →
  Cube 1 share = Cube 2's share



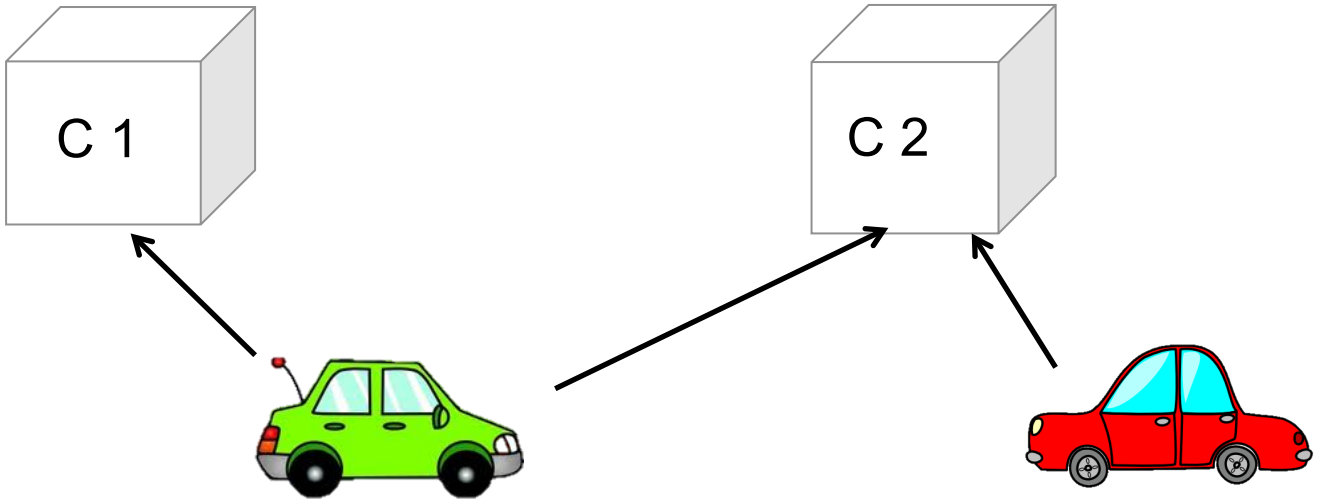
Solution 2: Content-based contention

- Replace sender-contention by content-contention
- Instead of senders, cubes contend for the medium → Requested cubes get equal share of medium
- But cubes are virtual entities → Cars viewing a cube contend on its behalf



Content-based Contention

- But how can a car compute its medium share?
- Each cube should get a share of ½
  - Green car share of the medium ¾
  - Red car share of the medium ¼



Content-based contention

- Car listens to how many cars respond for a particular cube
  - Share-per-cube = 1 / # cars responding
  - *Car's total share = (Σ share-per-cube) / # requested cubes*
- Set contention window using car's share on off-the-shelf cards

CarSpeak implementation

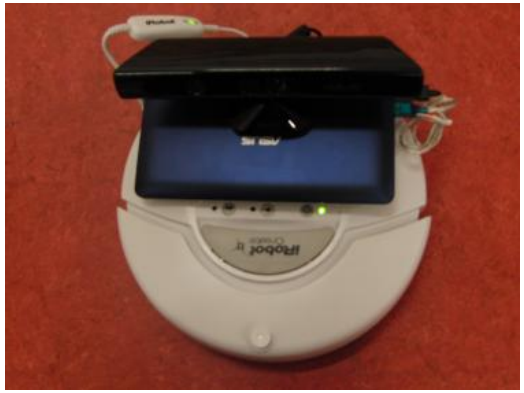
- Implemented in Robot OS (ROS)
- Integrated with MIT's Path Planner from DARPA challenge
- MAC implemented in the ath9k driver for Atheros WiFi cards

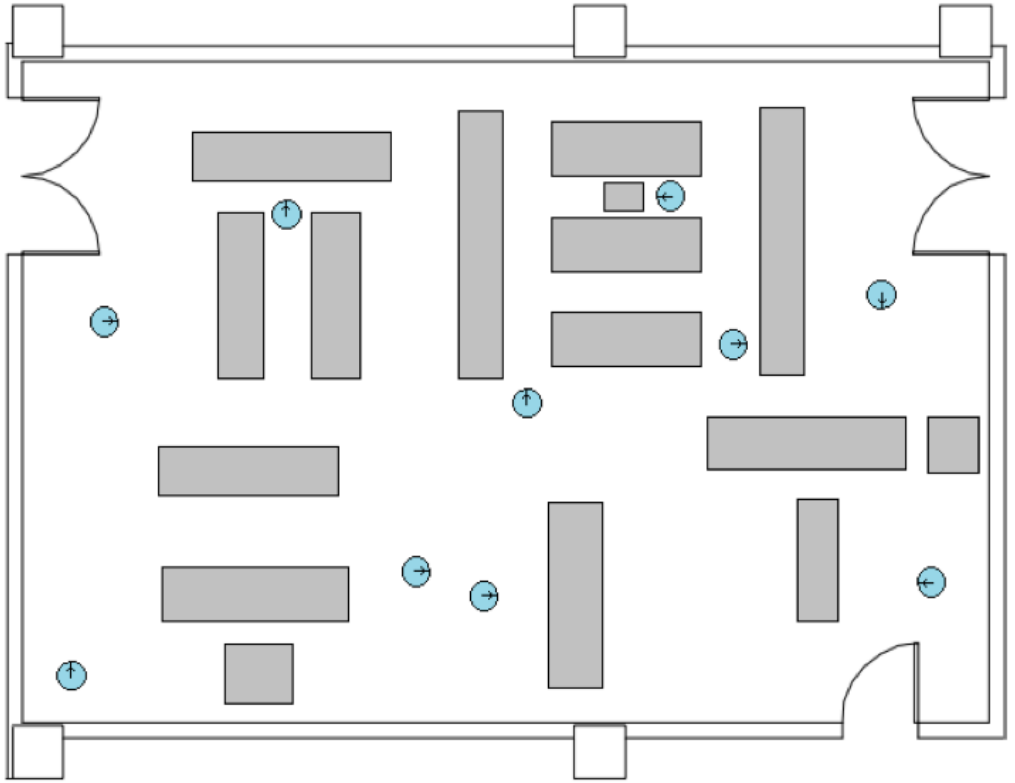
Compared schemes

- CarSpeak
- 802.11 (and 802.11+Naming)
- Experiment settings
  - Indoor testbed with robots
  - Outdoor testbed with autonomous car

Indoor testbed

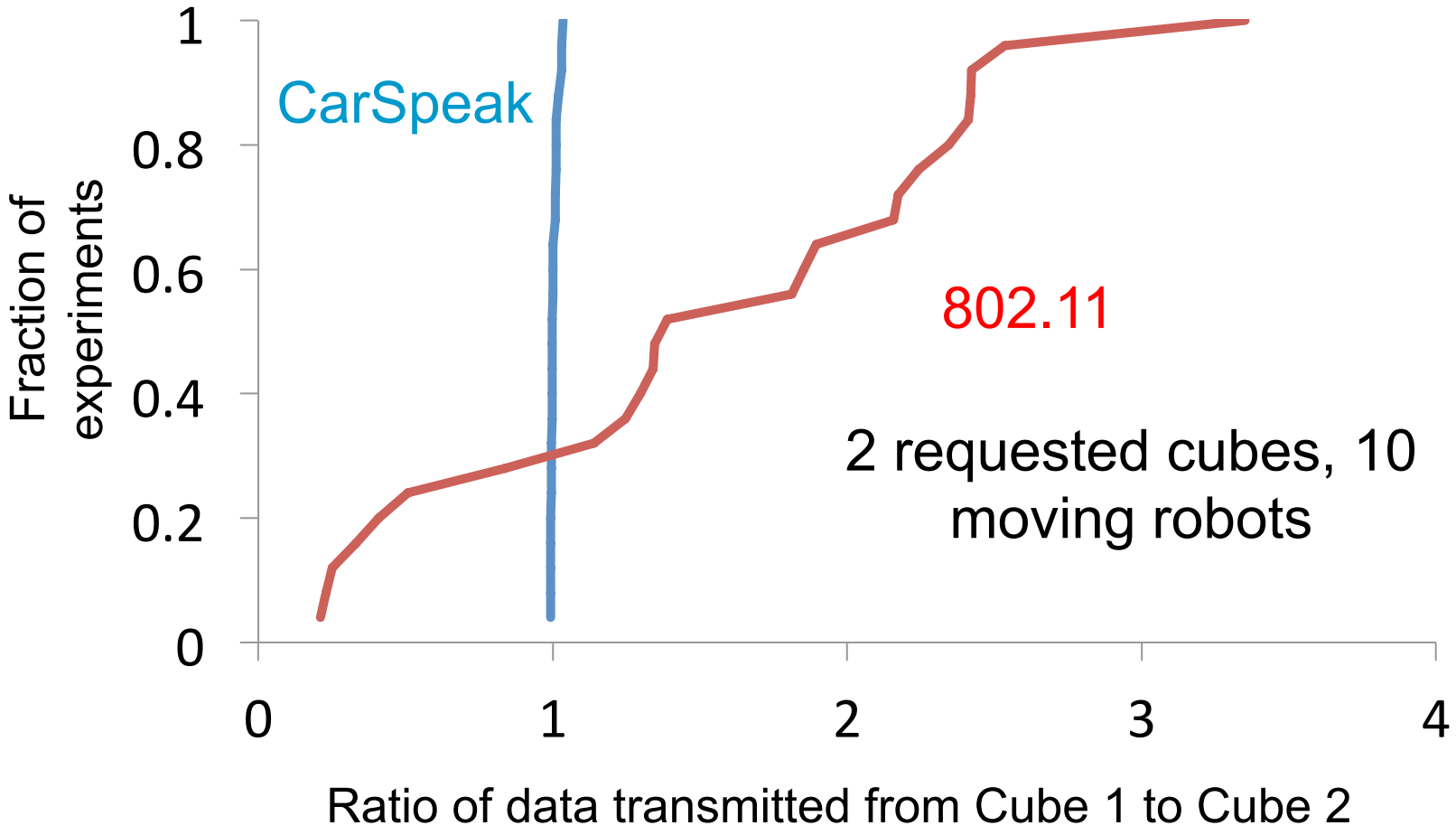
- 10 Roomba robots with Kinect
- Navigate environment with obstacles





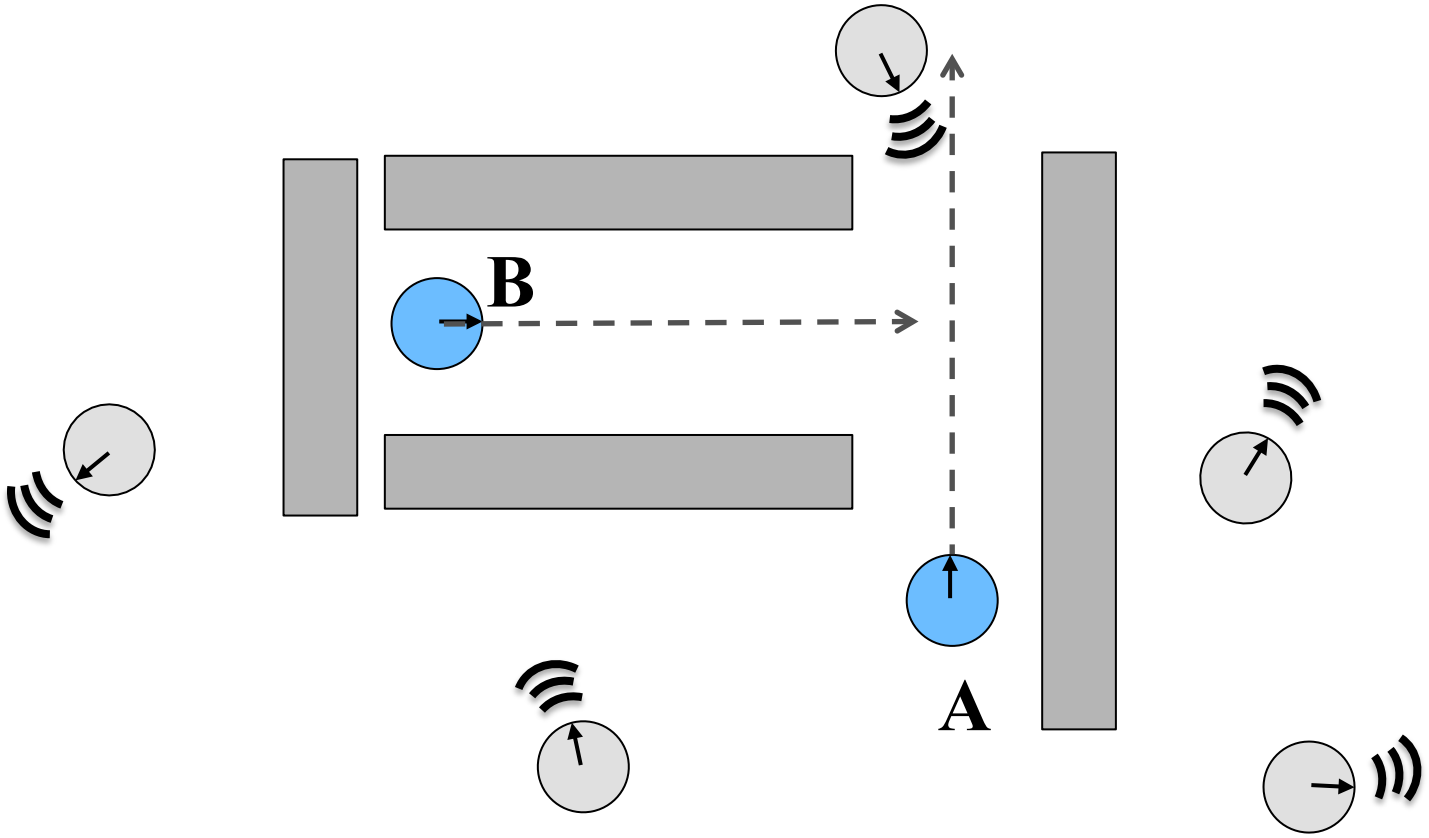
CarSpeak's MAC

CarSpeak's content-centric MAC divides the medium fairly between requested content

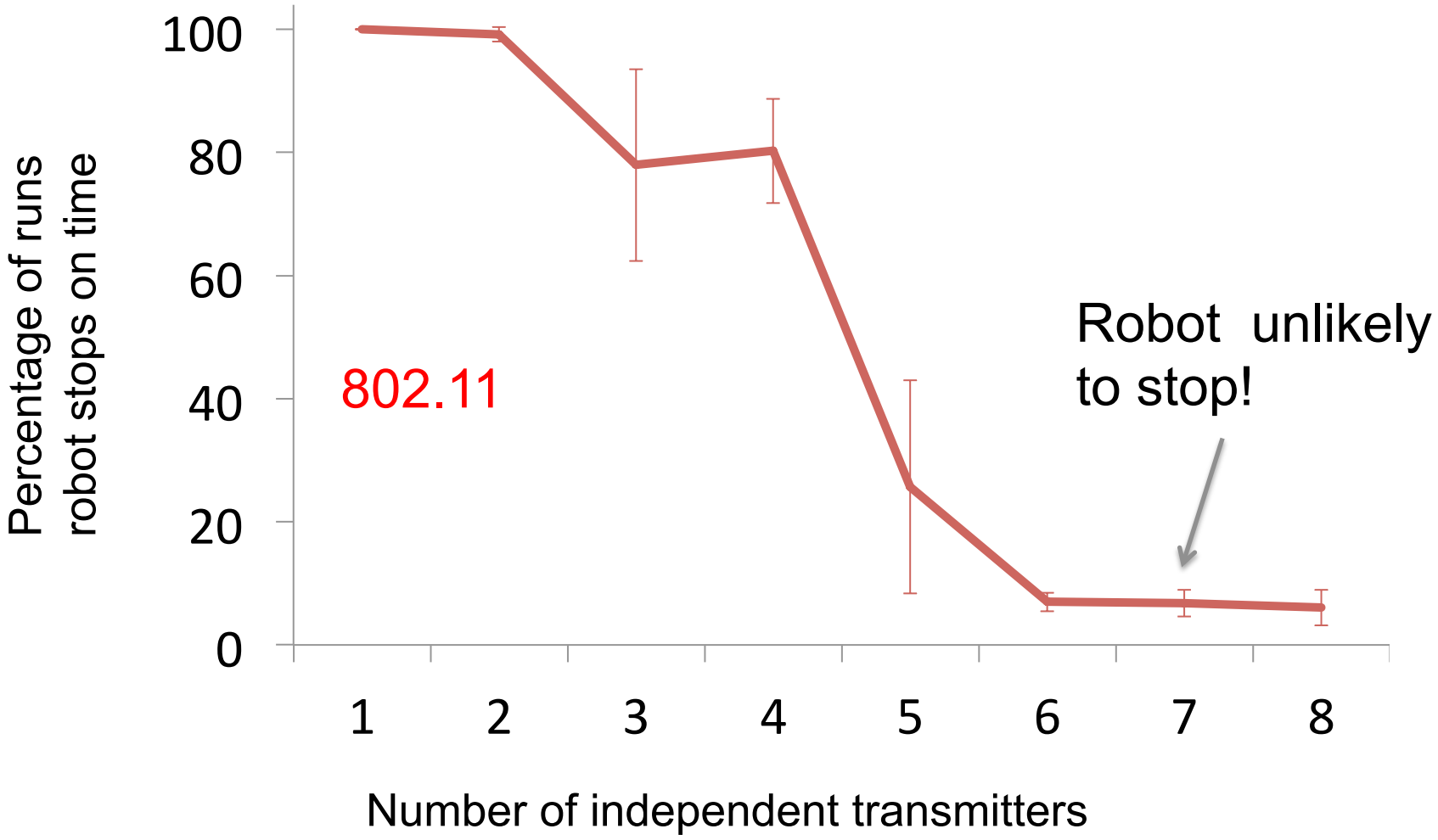


Reaction to objects in blind spots

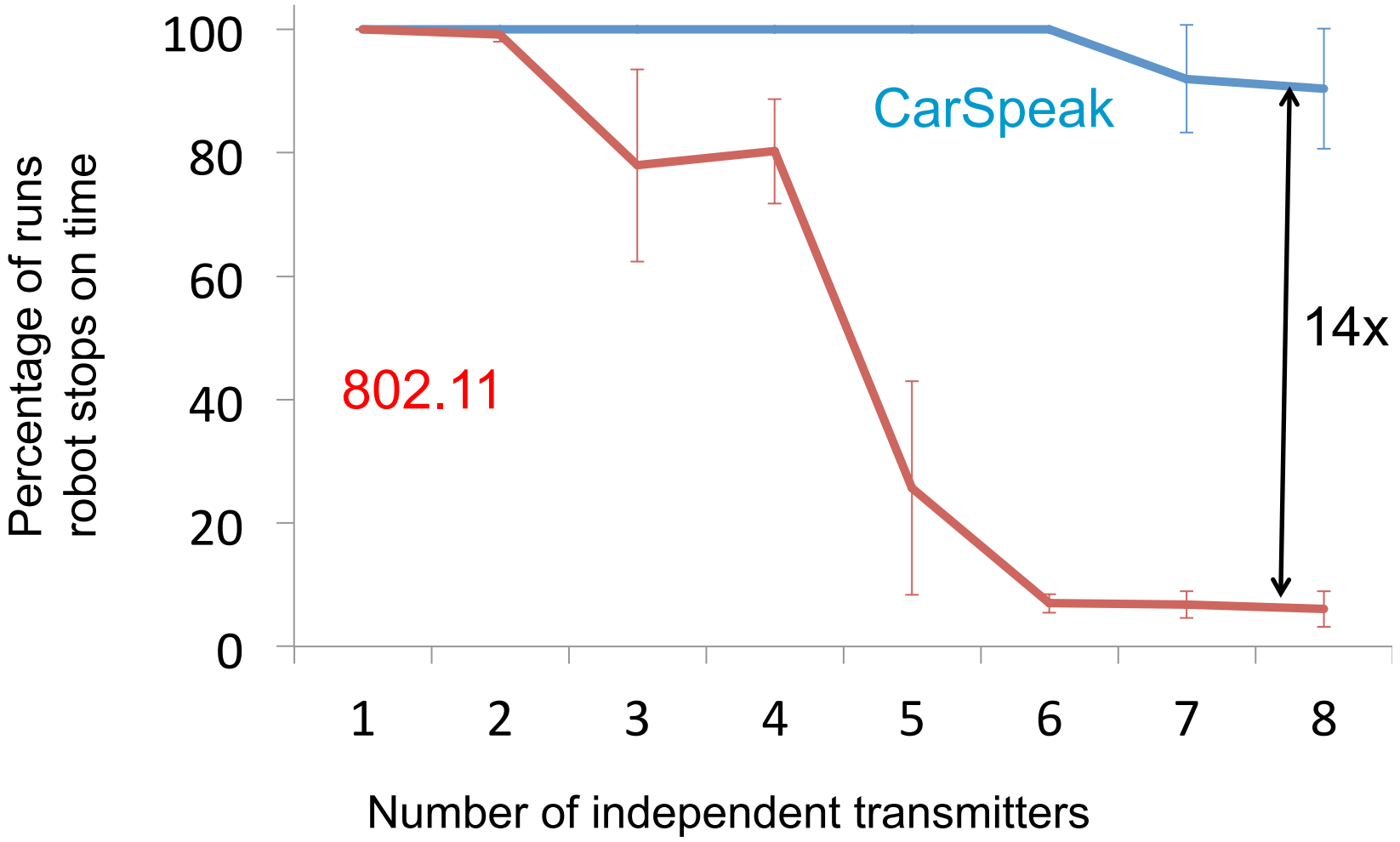
- A drives on road while B pulls out of occluded driveway
- Timely communication can help avoid collisions



Reaction to Objects in Blind Spots



Reaction to Objects in Blind Spots



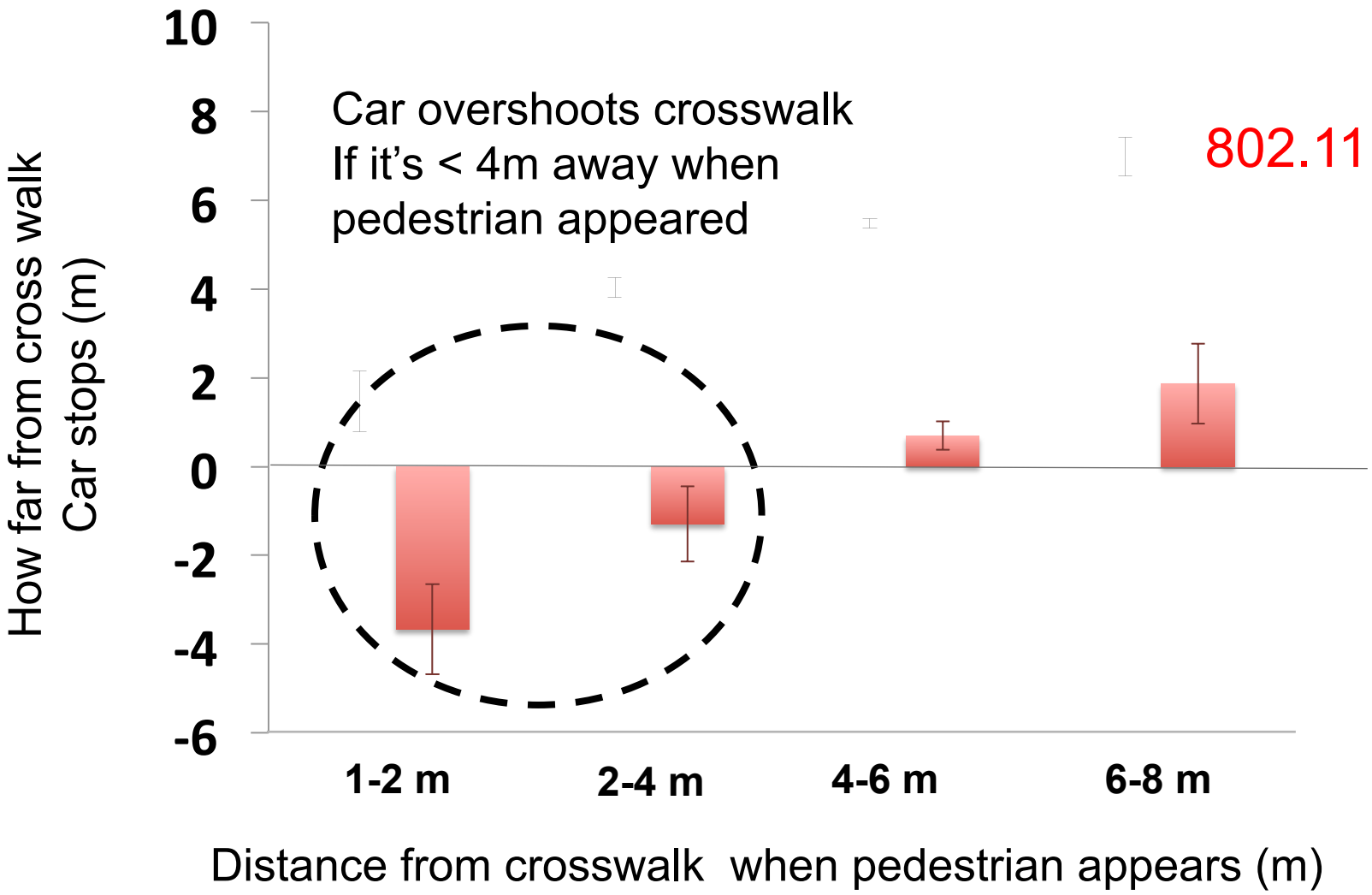
Outdoor testbed

- Instrumented Yamaha car with laser sensors
- Pedestrian crosswalk in campus-like environment

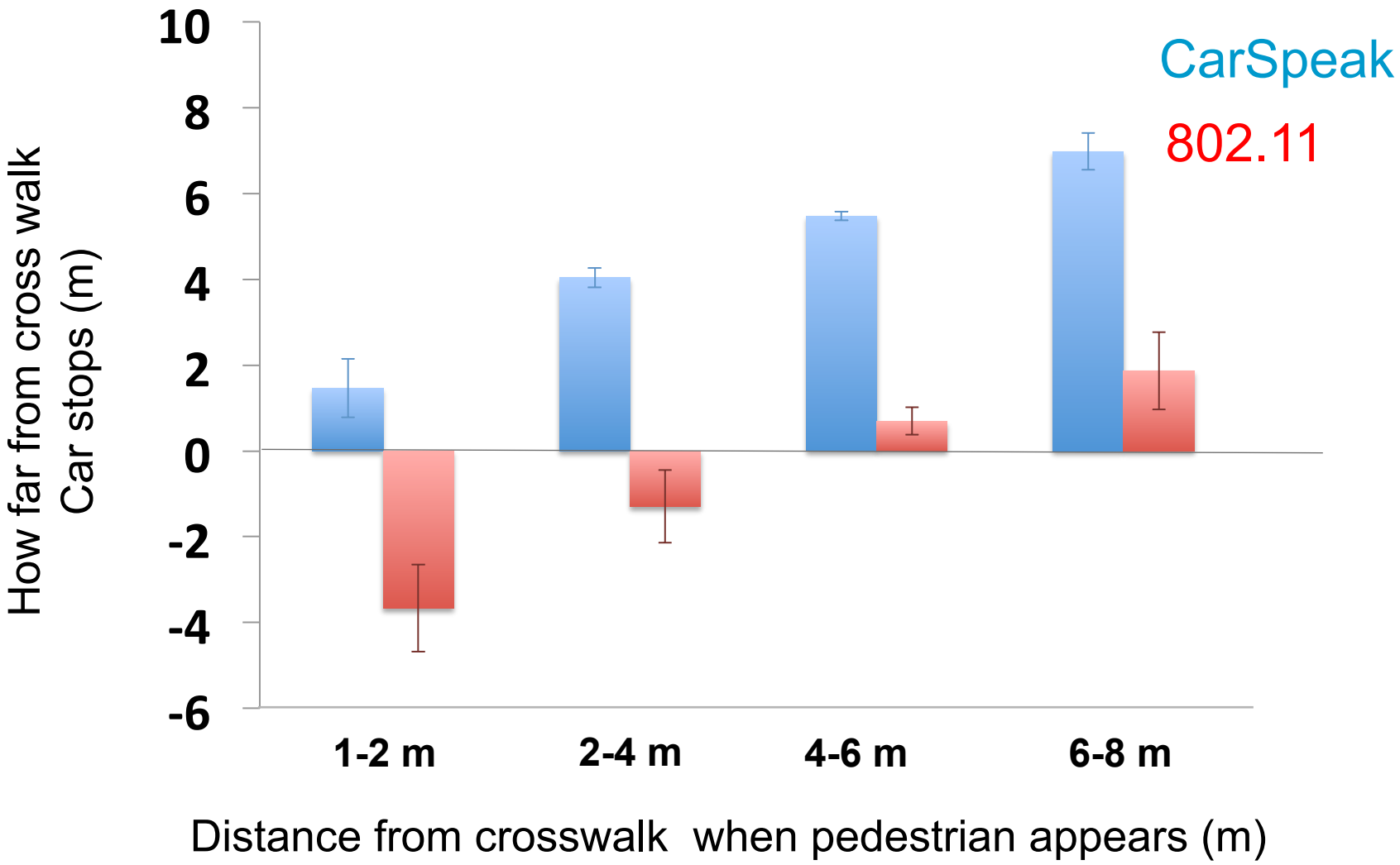
Detecting a pedestrian in blind spot

- Pedestrians emerge from lobby
- Lobby is a blind spot
- Infrastructure sensors can send view of lobby

Outdoor results



Outdoor results



CarSpeak in action

CarSpeak autonomous vehicle driving on the NUS campus in Singapore

Conclusion

- A communication system fully integrated with autonomous driving
- Content-Centric approach to data access & MAC
- Generally applies to collaborative robotics, virtual reality and virtual games