High-Mobility Applications of Sensor-Actuator Networks

MICS Phase 2
Requirements and Challenges of Applications

EPFL-IC-ISC-LCA4
Michal Piorkowski,
Matthias Grossglauser

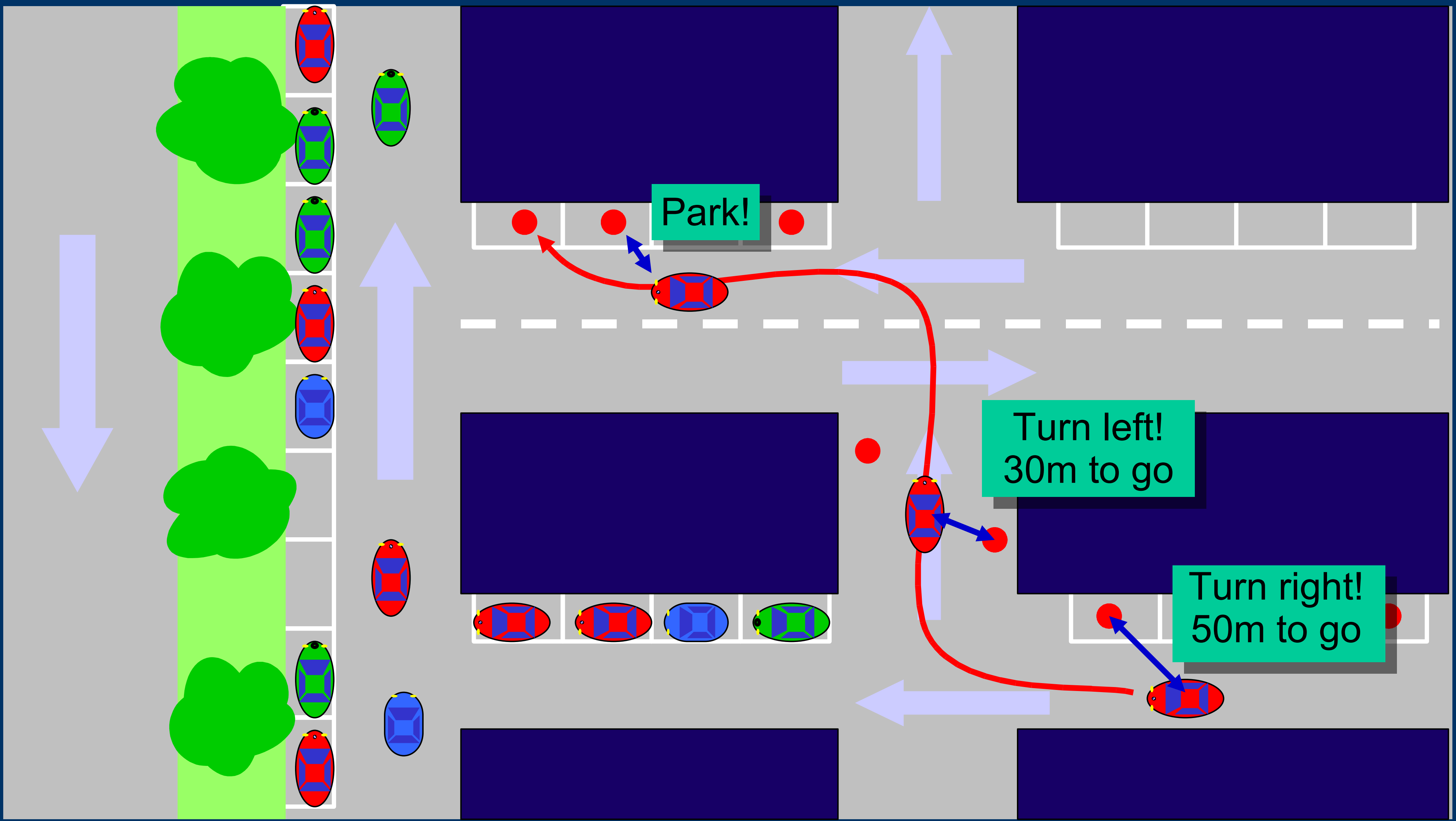
Background & Motivation

- Why a high-mobility application?
  - Arises in many practical settings: transportation, ubiquitous computing, supply chains, environmental monitoring
  - Complements other projects more focused on static settings
- Why a sensor-actuator application?
  - Information consumed inside network to make decisions
  - Control loop -> locality

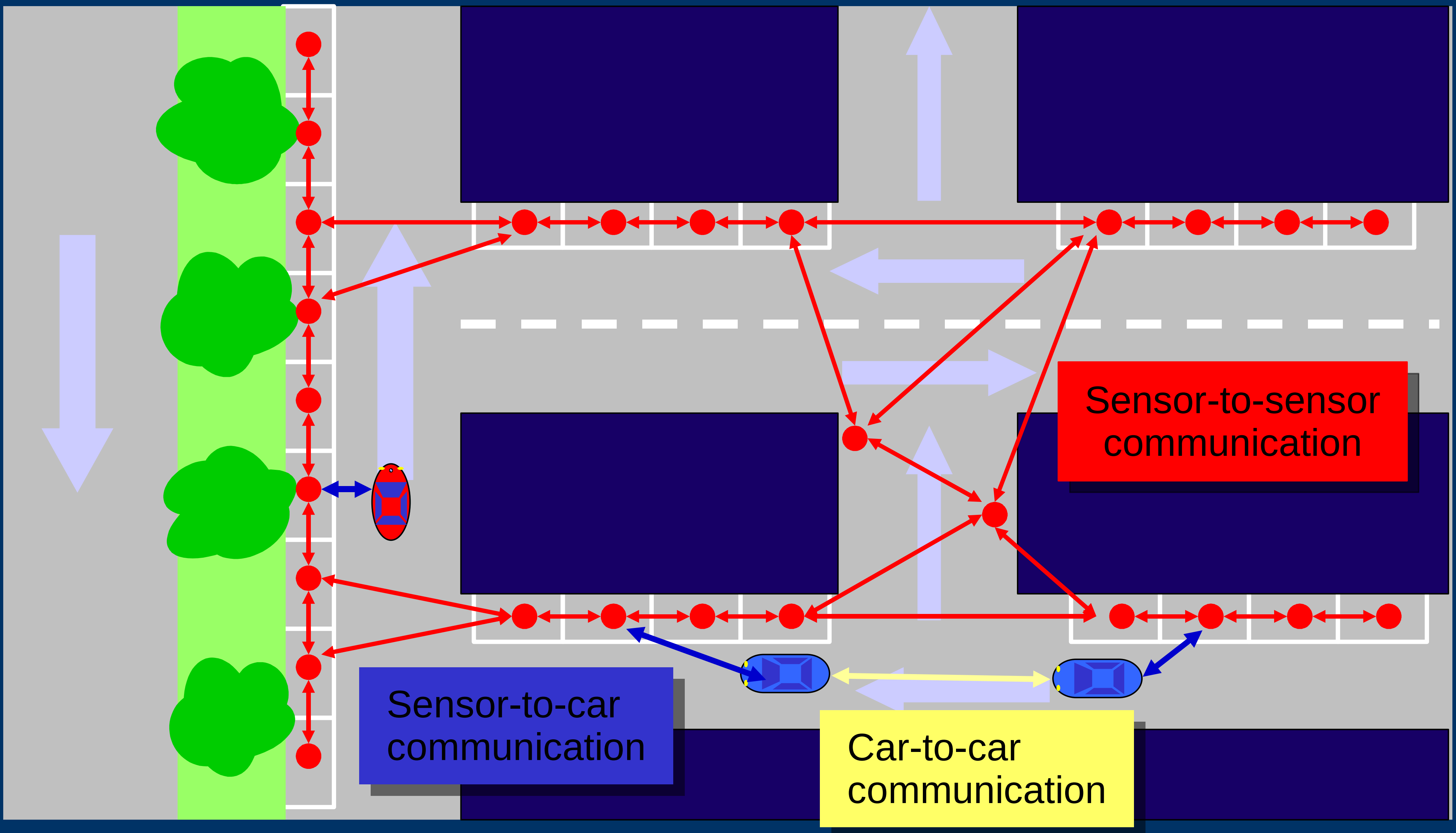
Example Applications

- SmartPark:
  - Guiding drivers to free parking spots
- SmartTaxi:
  - Guiding cabs to customers
- WatchMyStuff:
  - An inconspicuous personal belongings control
- PeopleNet:
  - Keeping friends altogether in the crowd

SmartPark: Close-up I



SmartPark: Close-up II



SmartPark: Requirements

- User:
  - Very simple to use
  - Low cost (off-the-shelf)
  - Unlicensed band
  - Only local communication with nearby nodes
  - No GPS, map database, interaction with central server, etc.
- System Administrator:
  - Low cost:
    - Self-contained, self-powered, small form factor
    - No GPS, “location-based service” infrastructure, map database, etc.
  - Robust:
    - Simple and cheap to overprovision for redundancy
  - Simple deployment:
    - Unlicensed spectrum: no red tape, no licensing fees
    - No or minimal configuration for new nodes
  - Simple maintenance:
    - No need for repairs, simply replace low-cost sensors
    - Sensor network autoconfigures & adapts

SmartPark: Challenges

- Randomness in mobility patterns
- Localization and tracking
- Information dissemination
- Dynamic end-to-end routing of messages
- Efficient and collision-free resource allocation
- Mobile node coordination
- Scalability, robustness and modularity

SmartPark: WSAN Approach

- Sensor nodes - triple use:
  - Sensing
    - Sensor embedded in pavement, parking meters, etc.
  - Communication/networking substrate
    - Only local communication between nodes, self-organizes into network
  - Navigation
    - No need for GPS, map database, etc.
- User device - contact nearby node to:
  - Get and disseminate updates on state of parking network
  - Communicate desire to find parking spot
  - Receive and forward navigational instructions
  - Allow sensor network to learn road topology
  - Extend sensor network connectivity through “mobility forwarding”

SmartPark: Architectures

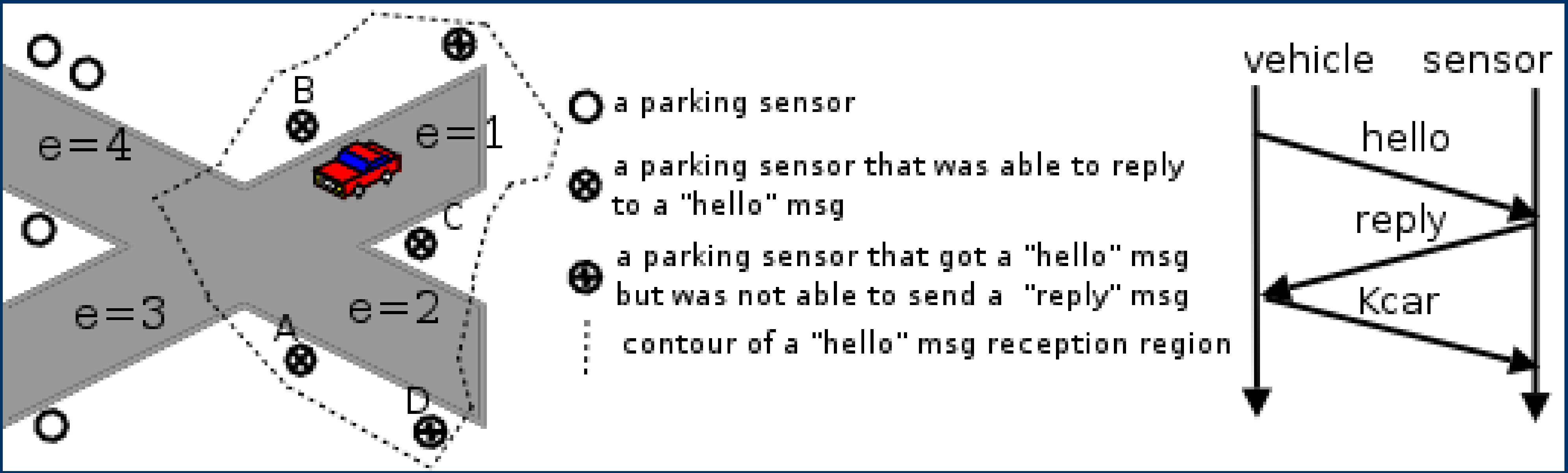
Architecture Type (only parking sensors)	**Cost**	**Lifetime**	**Deployment and Maintenance**
sensing, computation, communication	high	short	high
computation, communication	affordable	short	affordable
no parking sensor at all (system relies on the vehicles only)	low	∞	low

Requirements:

- small physical size
- low power consumption
- robust operations
- design and usage diversity
- programmability
- flexibility
- security

SmartPark: Current Status I

- MOONwalk - a constrained tracking algorithm
  - problem: radio channel uncertainty
  - constraint: car moves on directed graph
  - approach: use the road topology constraints and the following relation RSS ~ d^{-1}
  - experimental evaluation in in-house testbed



SmartPark: Current Status II

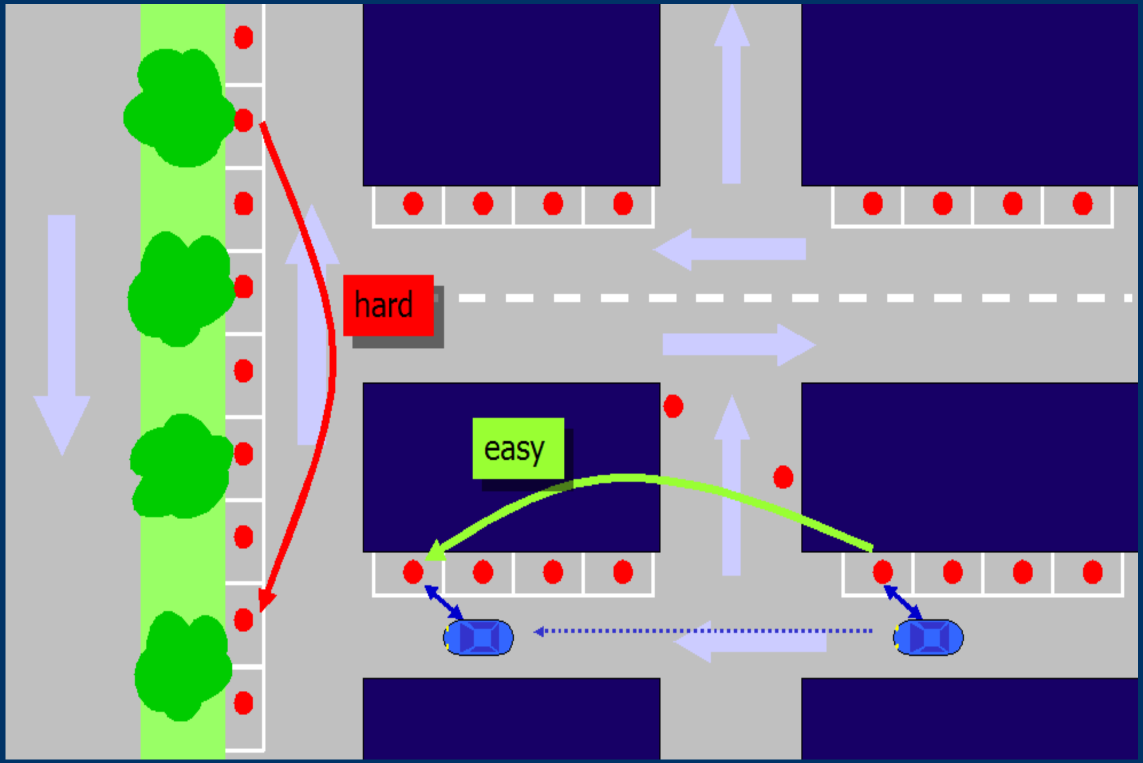
- Efficient Event Propagation
  - Goal: bring the information about resource availability to the users, so that they do not need to ask for it specifically
  - Approach: mobility diffusion at different time/space scales
    - Compute multiple resolutions over space and time, car mobility disseminates occupancy estimators
    - Space: block, neighborhood, city
    - Time: 30 sec for individual space, 2 min for block, 1h for city

Space X:
"available
30 sec ago"

SoHo area:
"1% available
over last hour"

SmartPark: Current Status III

- Learning the Road Topology
  - Goal: optimize decisions in terms of road topology, traffic conditions
    - Critical to make efficient space allocation & car routing decisions
    - E.g., take into account one-way streets, road construction sites
    - But avoid “explicit” approach: difficult, expensive, unreliable, non-adaptive to “statically code” road topology
- Approach: automatic inference of road topology from actual movements of cars
  - Use movements of user devices, tracked by sensor nodes
  - Adapt to changes on many time-scales: short (traffic jam), intermediate (construction site), long (road construction)



Questions?