Reducing Tail Response Time of Vehicular Applications

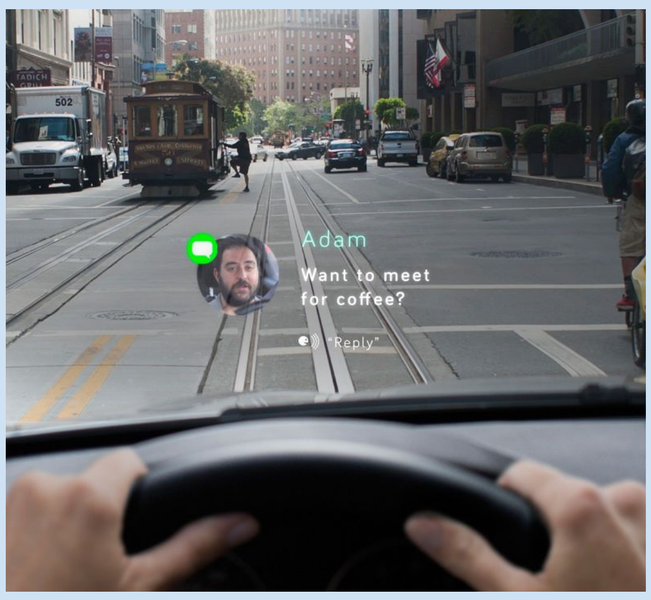
HyunJong (Joseph) Lee, Jason Flinn
{hyunjong, jflinn}@umich.edu

Motivation

- Emerging vehicular applications are sensor-rich (e.g., around-view cameras, telematics, GPS, …)
- Apps are latency-critical
  - Driver assistance
  - Augmented reality
  - Infotainment

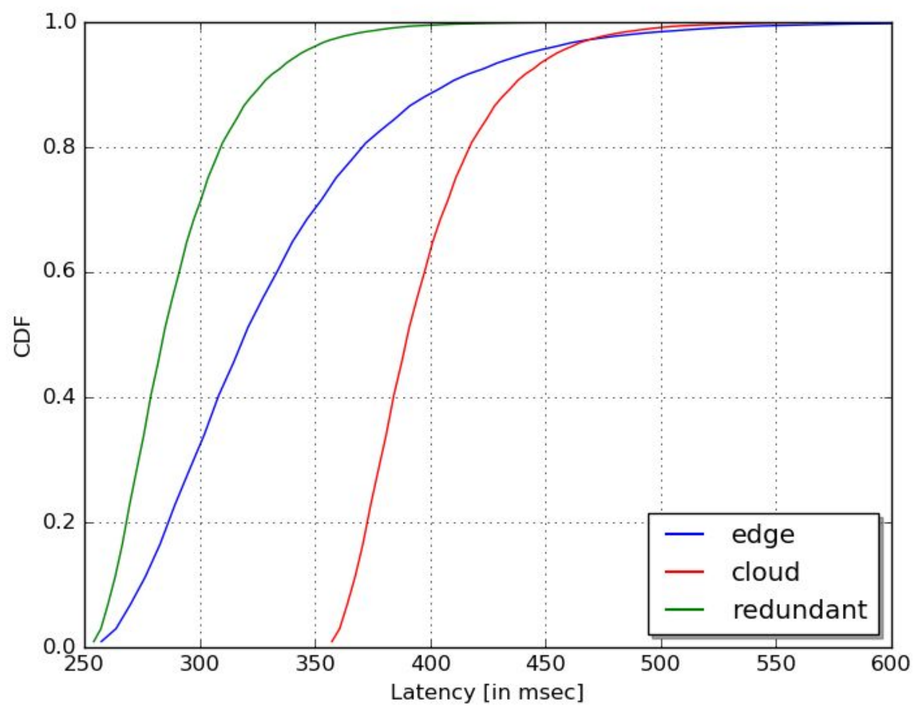
Problem

- Insufficient computation power in vehicles
- Cloud offload suffers from high network latency
- Response time varies due to
  - Vehicular mobility
  - Uneven provisioning in edge computing
  - Network variability



Solution

- Offload computation to edge-computing infrastructure located at
  - roadside WiFi hotspots or cellular infrastructure
  - smartphones in vehicles
- Principled application redundancy *when uncertainty exists*,
  ⇒ because mobility & variability are high
  - benefit (response time) is **critical**
  - cost ($$$ for data consumption) can be small
- Network redundancy in network stack, using MPTCP
  - implemented the redundancy as a scheduler
  - transparent to existing TCP standard
- Estimate error distribution with confidence interval via EMA
  - to detect tail network latency (jitter)
  - to forecast moving direction and do pre-provisioning VMs in edge infra.



API for expressing preferences

```
size_t dcm_send(int socket_fd, void *data, size_t data_len, uint32_t label_name);
int dcm_set_labels(int socket_fd, uint32_t label_name);
int dcm_set_constraints(int socket_fd, uint32_t constraint_name, int num_args, ...);
int dcm_set_policy(int socket_fd, uint32_t policy_name, void *arg, size_t arg_len);
```

Future Work

- Route prediction and pre-provisioning
- Smooth handover personal VMs from/to edge-computing nodes