

AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling

报告人：王琦 导师：李智勇





论文介绍:

题目: 《AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling》

期刊: 《IEEE Transactions on Vehicular Technology》

目标: 提出了AVE框架, 通过管理车辆上的空闲计算资源, 在动态车辆环境中提供计算服务 (不需要基础设施), 提高车辆的计算能力。

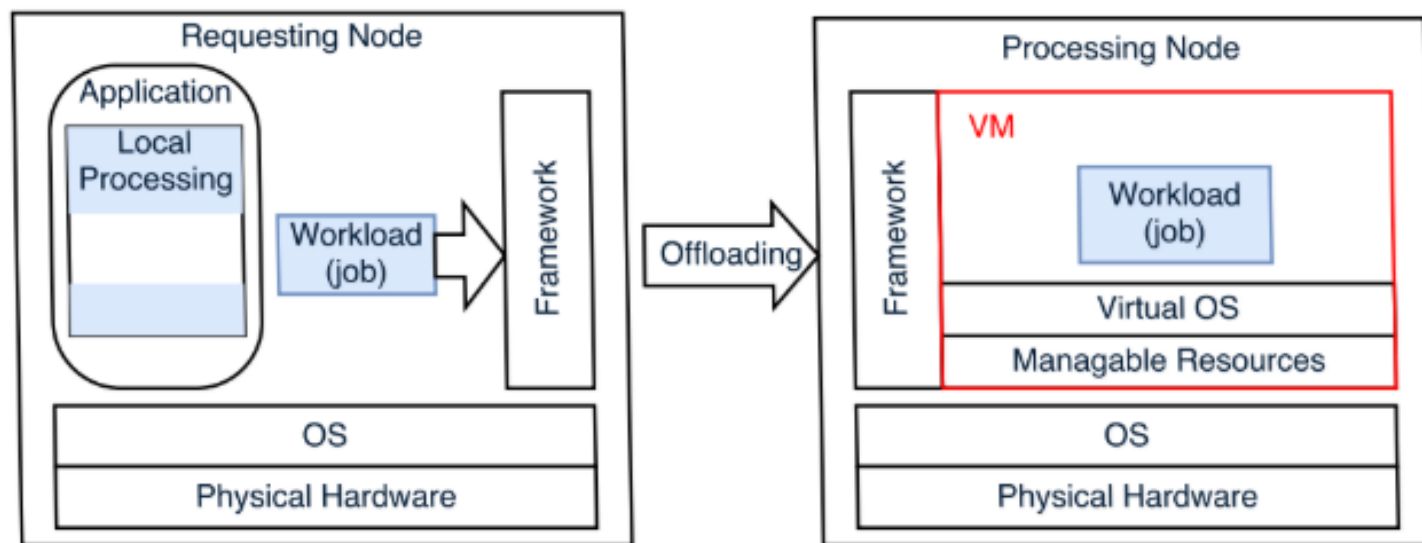
贡献:

- 1) 为AVE提出了一种**模块化架构**, 以支持任务卸载。
- 2) 设计了允许**卸载和调度的程序**, 而无需集中控制。
- 3) 基于**ACO的调度算法**, 通过快速收敛来解决NP-hard作业分配问题。
- 4) 使用**高速公路和城市情景**的实际交通数据进行了评估。

问题分析:

1. 车辆应用程序：关键应用程序(CA)，高优先级应用程序(HPA)和低优先级应用程序(LPA)。CA是车辆系统安全相关的核心应用，具有最高优先级，不算在AVE内。
2. 车辆边缘计算：
 - 1) 可共享资源：除了**CA**使用的计算资源之外，一些资源仍然可用。
 - 2) 运动模式：同一道路相同方向行驶的辆车之间的速度差小，链路持续时间长。
 - 3) 能源：车辆有大的电池存储，与驾驶期间消耗能量相比车载系统的能量很小。
3. 卸载和虚拟化：

请求节点将划分job并提交到AVE，AVE卸载到一个处理结点，处理节点的AVE会设置一个VM处理job。



AVE框架结构:

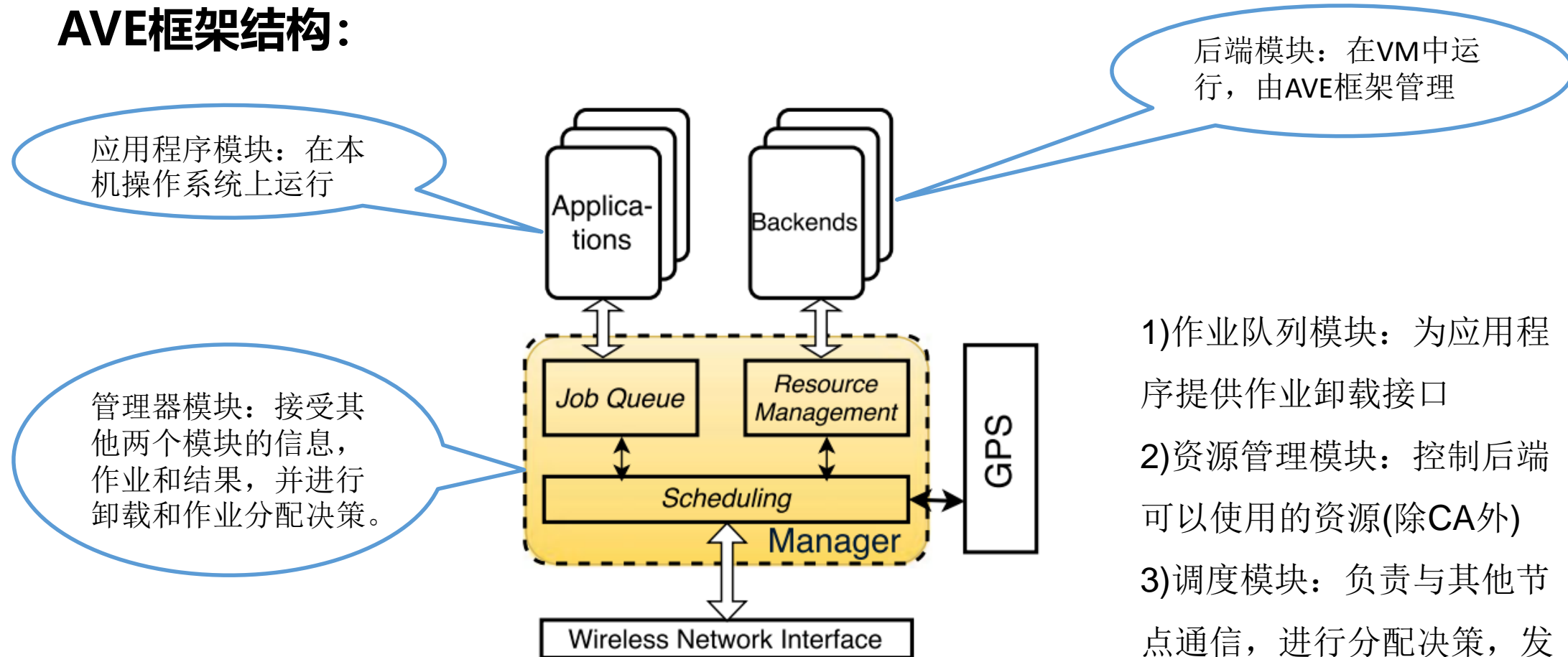


Fig. 2. Illustration of the architecture of proposed framework.

AVE 工作流程

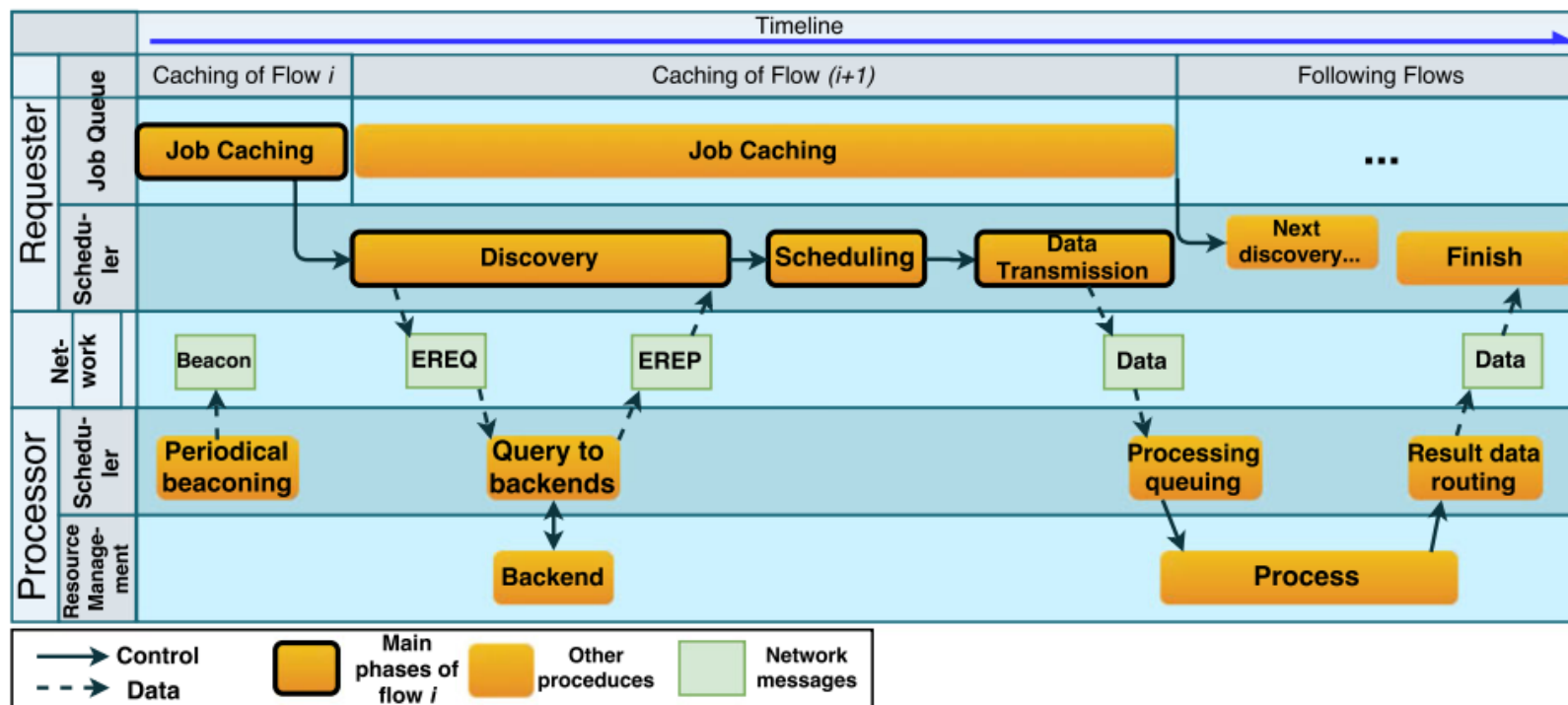


Fig. 3. Workflow of AVE. The orange boxes with borders correspond to the four main phases discussed in the following sections. Other minor procedures discussed in those sections are also shown.

AVE 工作流程：A. Beaconsing

每个车辆以一定的周期单独广播短beacon消息(默认值为10秒)。beacon消息不会被转发(只会发给一跳的节点)。 beacon消息包含以下信息：

- 1) 源车辆的ID,
- 2) 它的速度,
- 3) 它的空闲状态,
- 4) 源车辆的空闲1跳邻居的ID列表。

接收者接收到beacon时，计算自身与发送者之间的速度差为 $|v-v_0|$ ，如果该速度差大于预设值，则认为其与源节点的链接不稳定并忽略此消息。否则，接收方将其添NAI(邻居可用性指数)表中，NAI结构<车辆ID，空闲状态，跳数，到期时间>

$$NAI = \sum_{k=1}^2 (\text{number of } k\text{-hop idle neighbors}) \cdot \phi^k$$



AVE 工作流程：B. Job Caching

管理器模块运行队列来缓存 job，而不是在 job 到达时立即调度。缓存持续时间为 caching window。

当第一个作业在 t_{first} 时刻到达当前 window，设置结束时间为 $t_{\text{end}} = t_{\text{first}} + Q / (NAI + 1)$ 。Q 是在本地处理作业的排队时间，如果此请求者空闲，则 $Q = 0$ 。

当满足以下两个条件时，缓存持续时间结束：

- 1) 前一个流程已完成数据传输，并且
- 2) 缓存作业的数量大于 $(NAI + 1)$ 或者时间超过 t_{end} 。



AVE 工作流程：C. Discovery

当缓存时间到期时，当前作业队列被锁定并且开始发现阶段。请求者首先广播边缘请求消息 (EREQ) 以发现潜在结点。EREQ<请求者ID, 速度, 缓存作业信息>。

接收者计算EREQ中的速度和自身速度差，太大则忽略，会将EREQ转发给接收者的单跳车辆(转发者)。然后资源管理器会计算作业的预计完成时间bid。

如果有bid存在，则返回边缘响应消息 (EREP)，EREP<响应者ID, bid>。(网络拓扑在如此短的时间内不会发生变化)。

为了避免来自两个不同请求者的请求冲突，每个接收器保留资源一小段时间(默认值为50毫秒)，在此期间它暂停响应请求。

AVE 工作流程：D. Scheduling

一组有限的作业和一组有限的节点，
每个作业将被发送到其中一个处理节点。

调度方法就是要得到**要发送作业的顺序**和**每个作业的目的节点**。

分配完成后，使用与EREQ相同的规则广播短通知消息。此通知包含在调度期间获得的分配结果。这是为了通知分配的节点停止发送EREP并等待已分配给它们的作业到达。

A sequence $[(i_1, j_1), (i_2, j_2), \dots, (i_n, j_n)]$, where $i_k \in P$ ($k = 1, 2, \dots, n$) and $\bigcup_{k=1}^n \{j_k\} = J$, must be found to

$$\text{maximize } \sum_{j \in J} \omega_j(t_j) \quad (1)$$

$$\text{s.t. } j_k = j_h, \text{ i.f.f. } k = h; \quad (C1)$$

$$c(i_k, j_k) = 1, \forall k = 1, 2, \dots, n; \quad (C2)$$

$$t_{j_k} = o_{j_k} + TR_{i_k, j_k}, \forall k = 1, 2, \dots, n; \quad (C3)$$

$$o_{j_k} = \max(p_{j_k}, q_{j_k}) + B_{i_k, j_k}, \forall k = 1, 2, \dots, n; \quad (C4)$$

$$p_{j_0} = 0; \quad (C5)$$

$$p_{j_k} = p_{j_{k-1}} + TS_{i_k, j_k}, \forall k = 1, 2, \dots, n; \quad (C6)$$

$$q_{j_k} = \max(\{o_{j_h} | i_h = i_k, h < k\} \cup \{Q_{i_k}\}), \\ \forall k = 1, 2, \dots, n. \quad (C7)$$



AVE 工作流程：E. Data Transmission

由于调度算法的运行时间很短，因此认为数据传输的路径和EREQ一样。

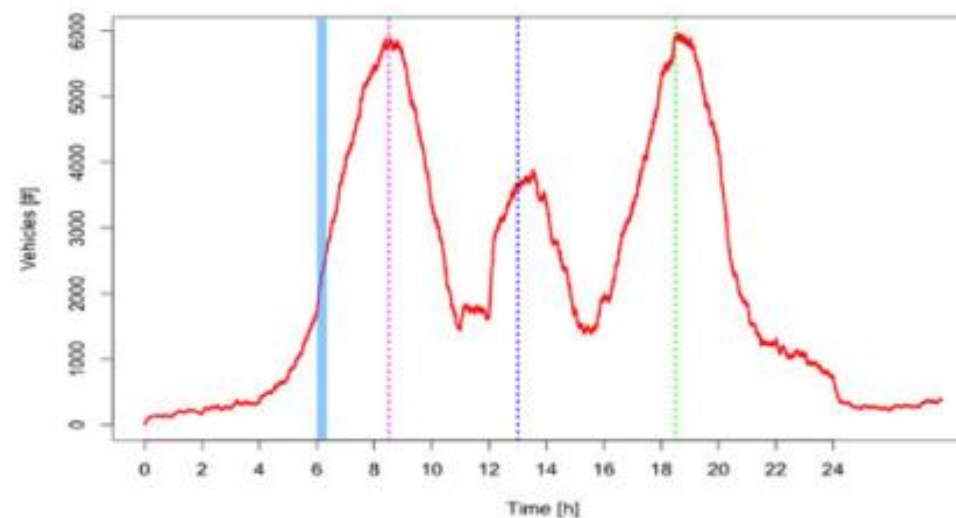
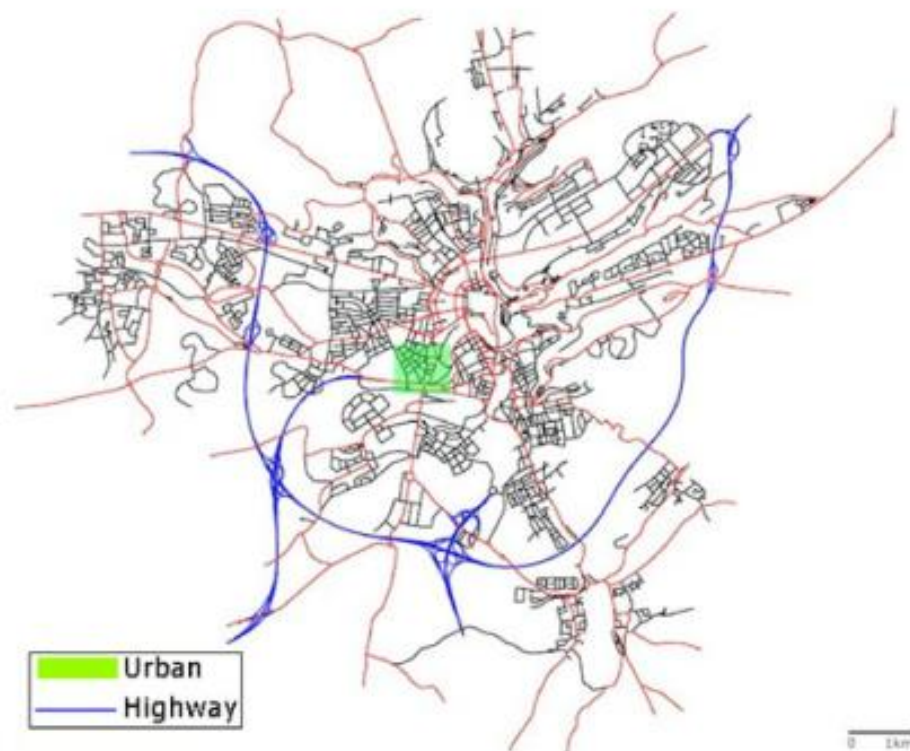
在接收到一个作业的所有数据后，处理器将该作业添加到其处理队列的末尾。排队的作业逐个处理。

作业处理完成后，结果将发送回请求者。

实验:

平台: SUMO、OMNet++、Veins

数据来源: 卢森堡市6:00-6:15时间段内的车辆。



实验:

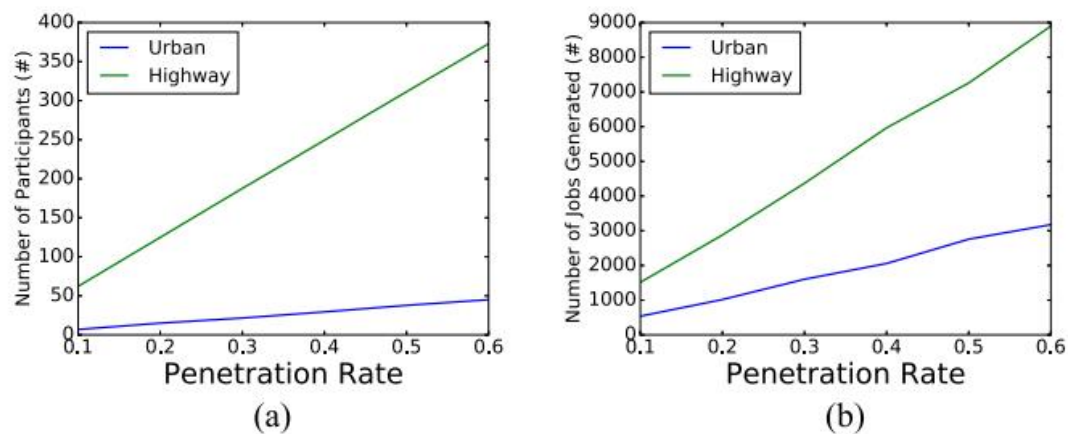


Fig. 9. Parameters of the evaluation. (a) Number of participating vehicles. (b) Number of generated jobs.

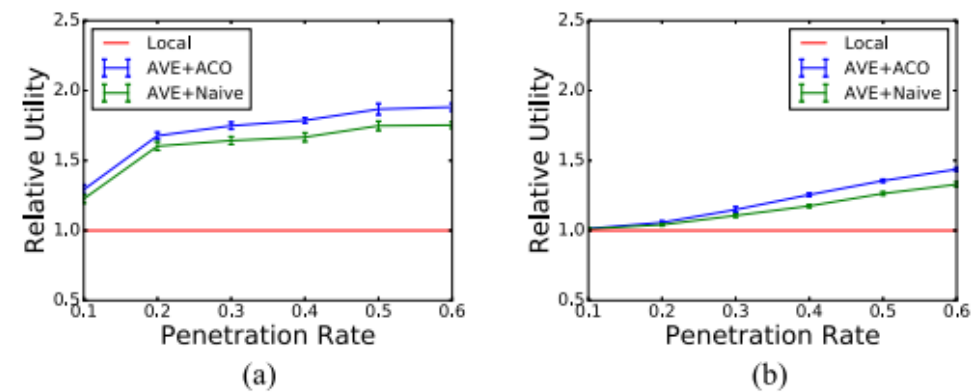


Fig. 10. Relative utility, normalized by dividing the utility gained by that in the purely local scheme. (a) Urban. (b) Highway.

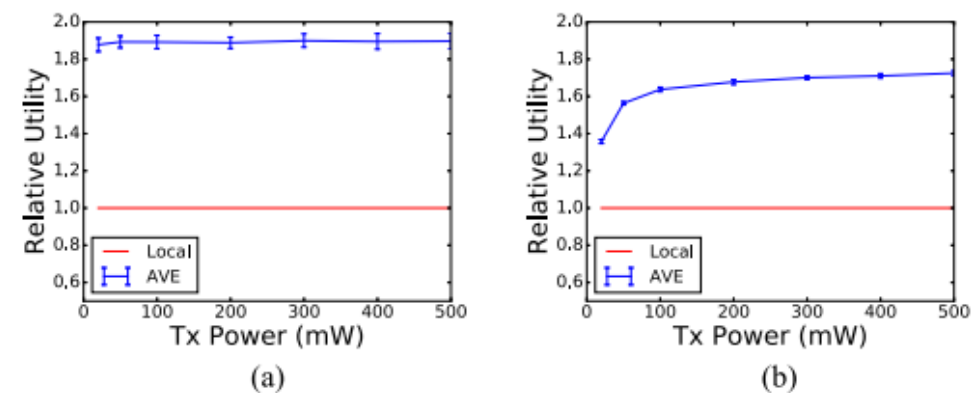


Fig. 11. Relative utility under different transmission power settings. The penetration rate is set to 0.5. (a) Urban. (b) Highway.

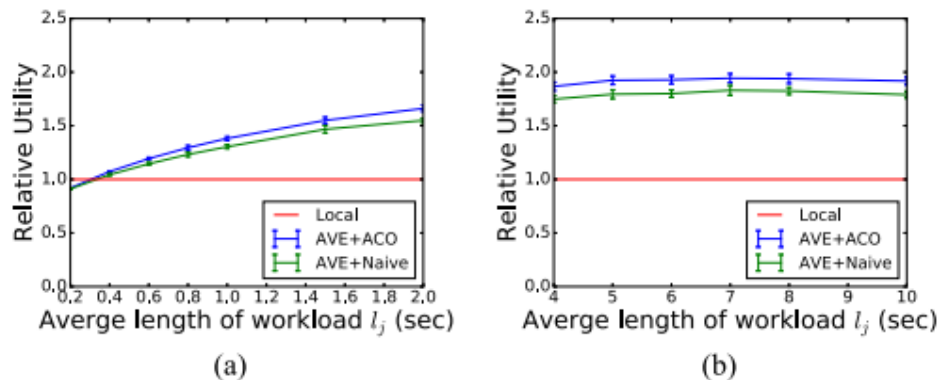


Fig. 12. Relative utility under different settings in the **urban** scenario. (a) Short workload. (b) Long workload.

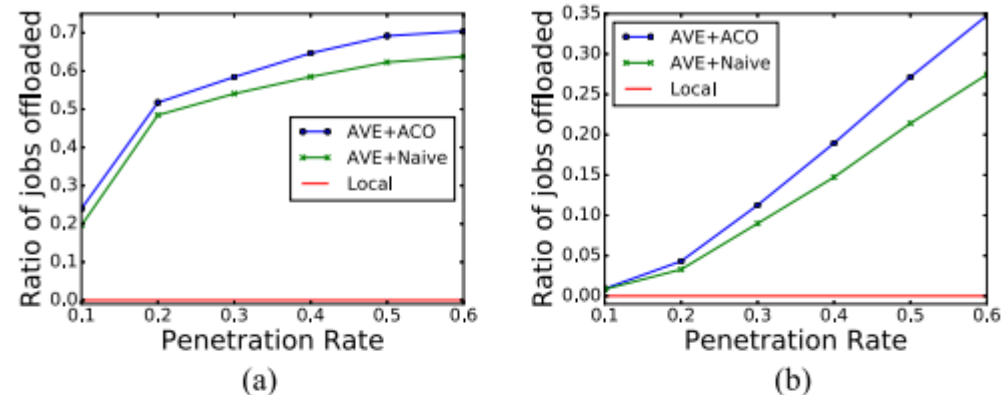
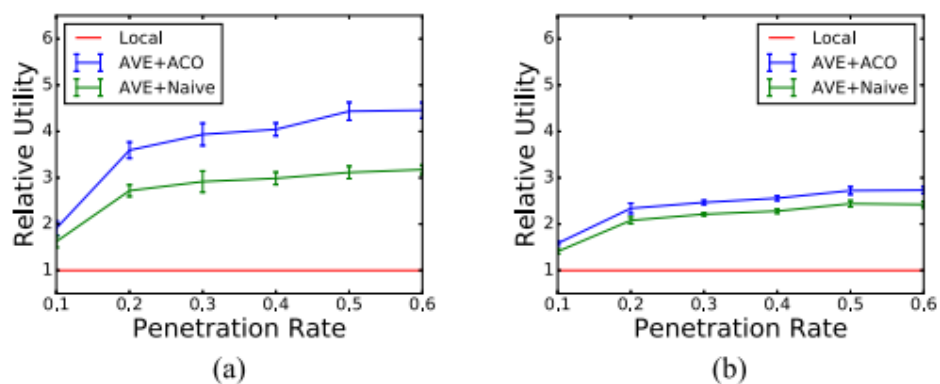


Fig. 14. Proportion of offloaded jobs. (a) Urban. (b) Highway.

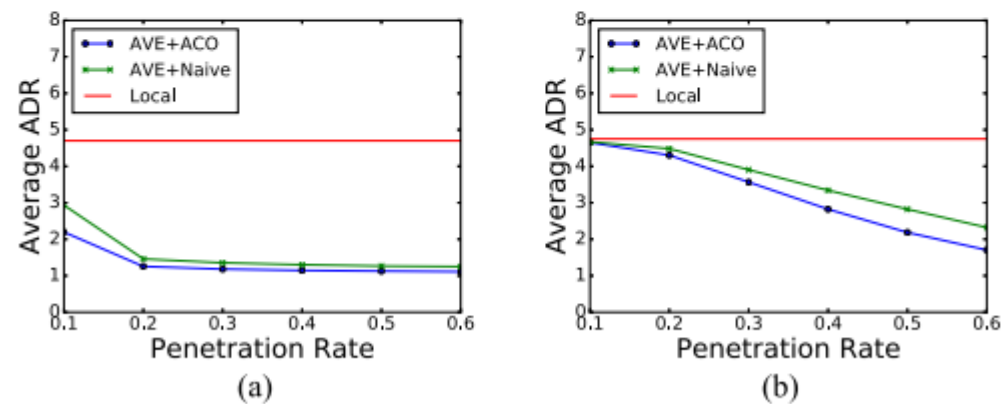


Fig. 15. ADR under different penetration rates. (a) Urban. (b) Highway.

谢谢

汇报人：王琦

导师：李智勇

