Intelligent Mobile Edge Computing (MEC): From Ground To Sky

Dr. Kezhi Wang

Senior Lecturer
kezhi.wang@northumbria.ac.uk

Department of Computer and Information Sciences
Northumbria University, UK

Intelligent Mobile Edge Computing (MEC)

We will introduce MEC later. Just think MEC is similar to cloud computing for now.

Two questions:

- How *MEC* can help users, e.g., IoT devices, sensors, user equipment or mobile phones to <u>have intelligence</u>.
- How *machine learning* can help MEC make better decisions, i.e., <u>better serve users.</u>

Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV (drone) -based MEC
- Hybrid MEC systems
- Conclusions

Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV-based MEC
- Hybrid MEC systems
- Conclusions

Why Mobile Edge Computing (1/2)

- ✓ Mobile devices are becoming more and more popular
- ✓ They can run attractive applications (resource-intensive)
- ✓ Mobile devices - Limited resources in terms of battery, CPU, storage

Not enough battery life?

CPU not fast enough?

YouTube

Resource-hungry applications

- ✓ Machine learning model (training) requires a large amount of computing resource.

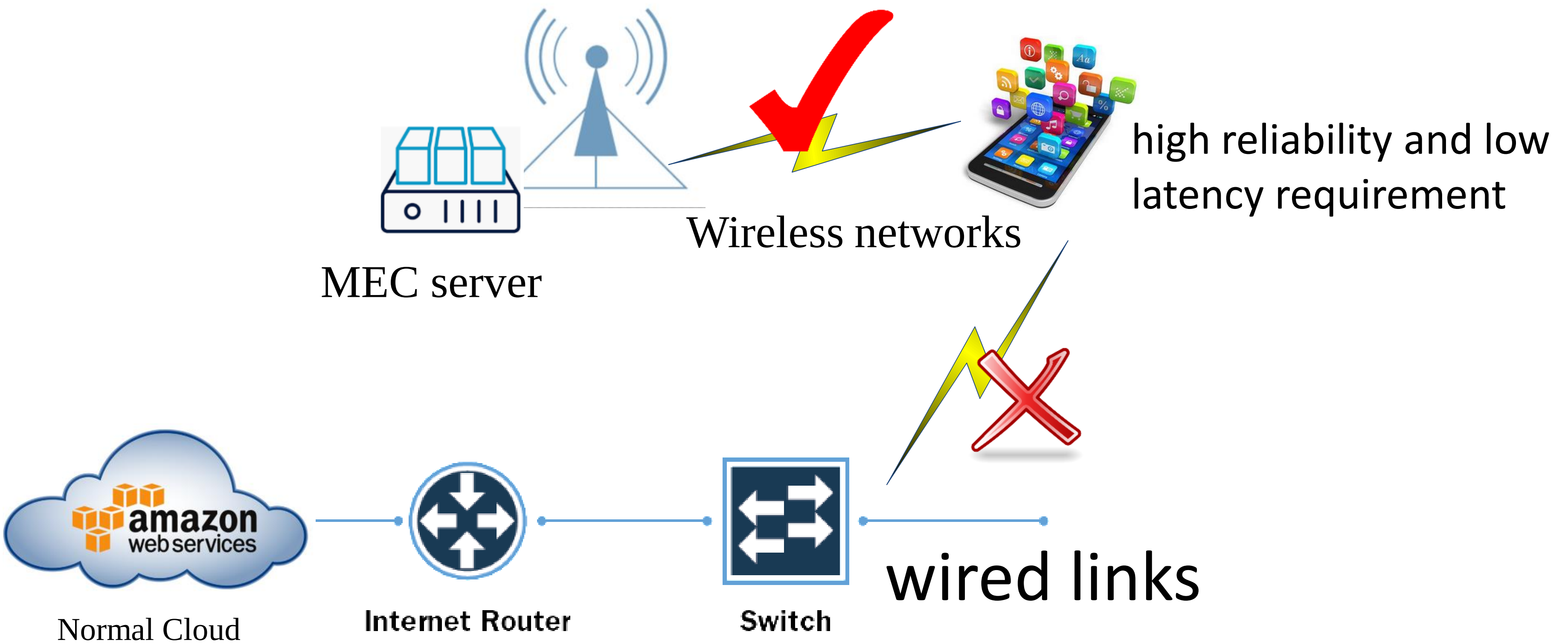
Why Mobile Edge Computing (2/2)

- ✓ Offload the computational intensive tasks to MEC server
- ✓ Save local battery
- ✓ User's experience will be increased, MEC much faster

- ✓ MEC: network architecture to deploy the computing resource at the network edge.

Where to deploy MEC (1/5)?

- ✓ MEC - at the network edge - respond to devices' requests very fast
- ✓ Different from the normal cloud- Amazon cloud (Centralized)
- ✓ Normal cloud - packet loss and latency – not for mobile application

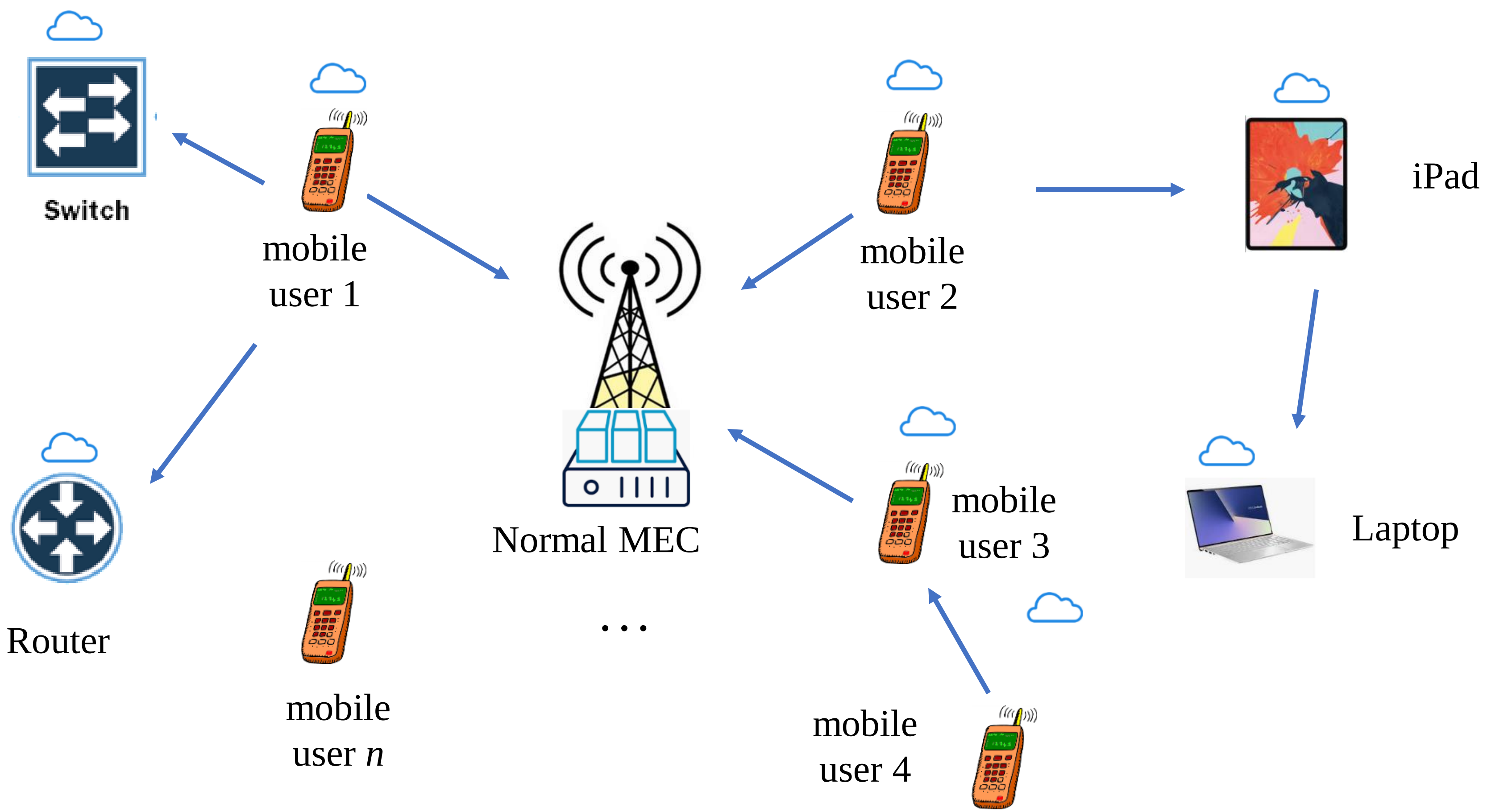


Where to deploy MEC (2/5)?

- ✓ MEC (<u>Decentralized</u>) - close to user- immediately respond to user's demand via wireless networks

Where to deploy MEC (3/5)?

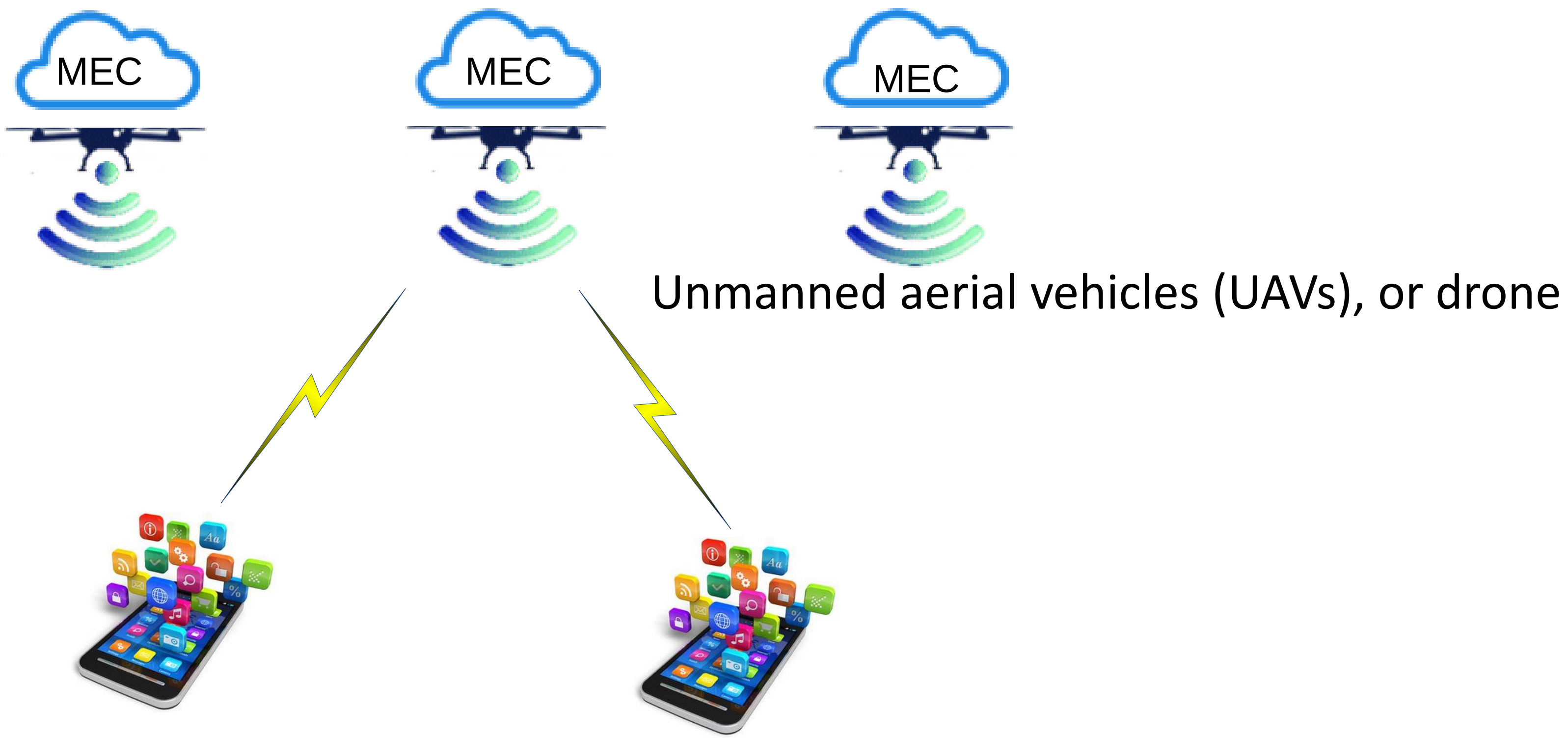
✓ Every device with computing resource maybe contribute to MEC



Where to deploy MEC (4/5)?

✓ UAV-enabled MEC (UAV carries small servers), with the feature of

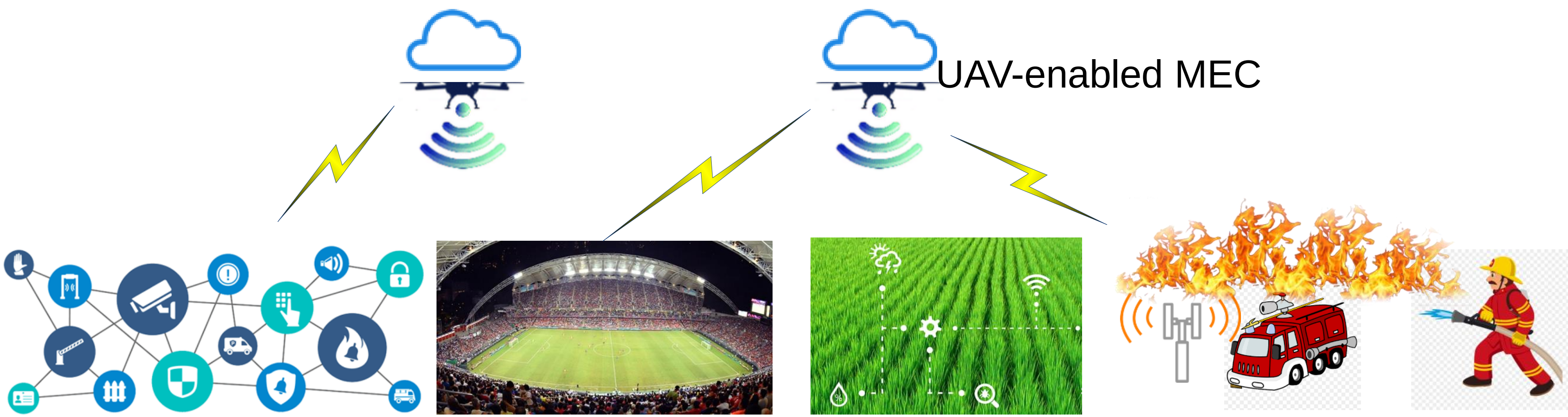
- Flexibility, mobility and autonomy
- Offers 3-Dimensional (3D) deployment
- Strong possibility of being able to engage in line-of-sight (LoS) communications



Where to deploy MEC (5/5)?

✓ UAV-enabled MEC, is particular useful in

- On-demand hotspot areas (e.g., large-scale users, base station is not powerful enough)
- Temporary activities (e.g., public event and football match)
- Emergency events (e.g., earthquake and large fires)



IoT Applications

Football March

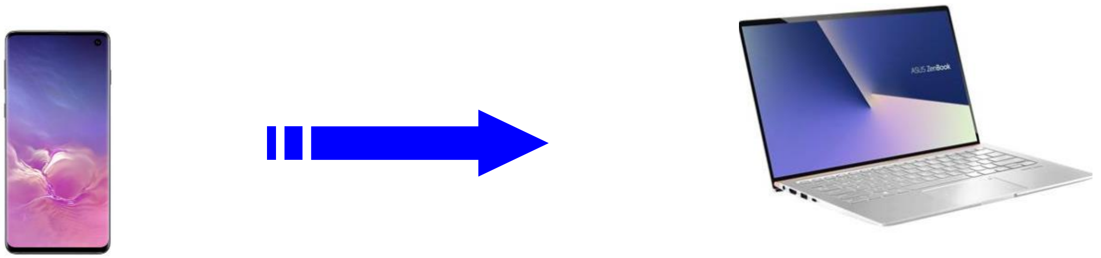
Smart Farm

Emergency Situations

Who may benefit from MEC (1/3)

✓ Ourselves

- We can offload tasks to MEC
- We may sell available computing resource within our device to other users to make money.



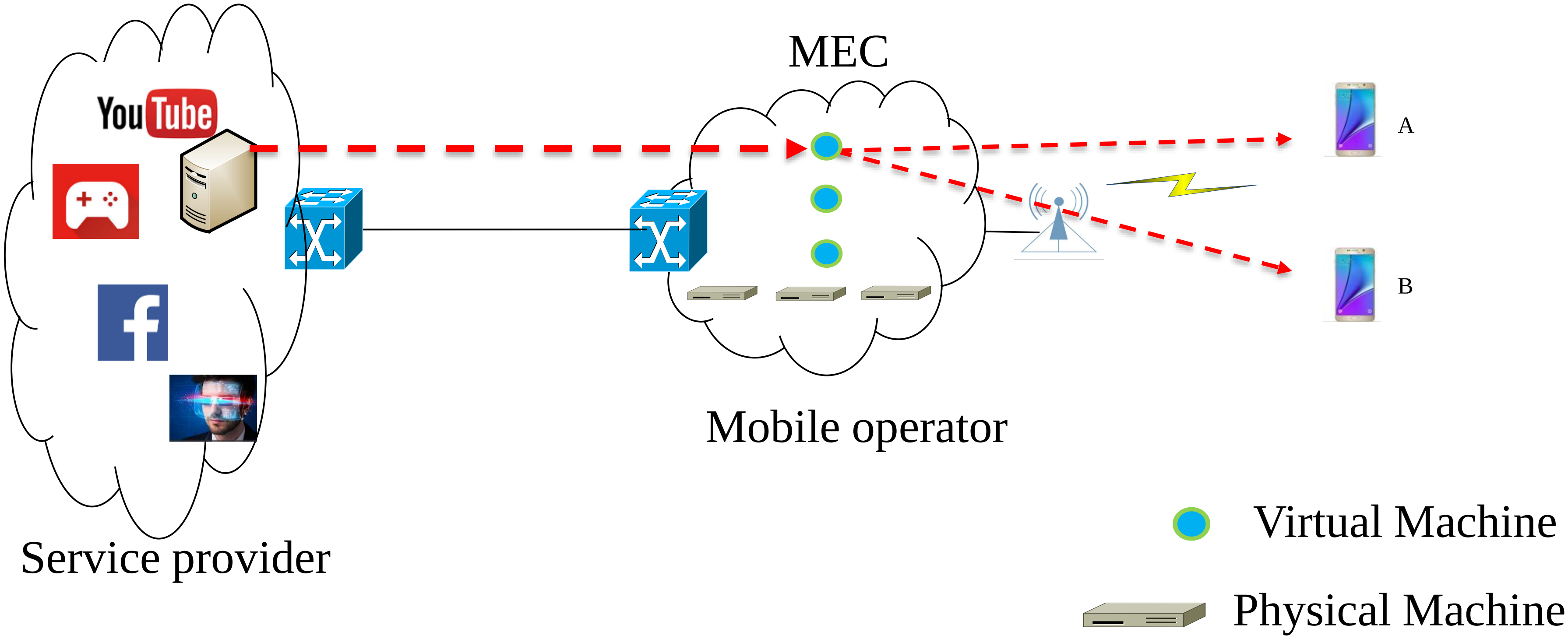
Who may benefit from MEC (2/3)

✓ Mobile operators, e.g., BT, EE, Vodafone

- MEC can enable them go **beyond** from just the pipe providers, but they could also be cloud service operators, like Amazon
- Operators can provide **better** cloud services than Amazon, as they holds both computing resource information and wireless channel status (wireless networks are controlled by operator ☺).
- Operators can jointly **leverage** (optimize) both communication and computing resource

Who may benefit from MEC (3/3)

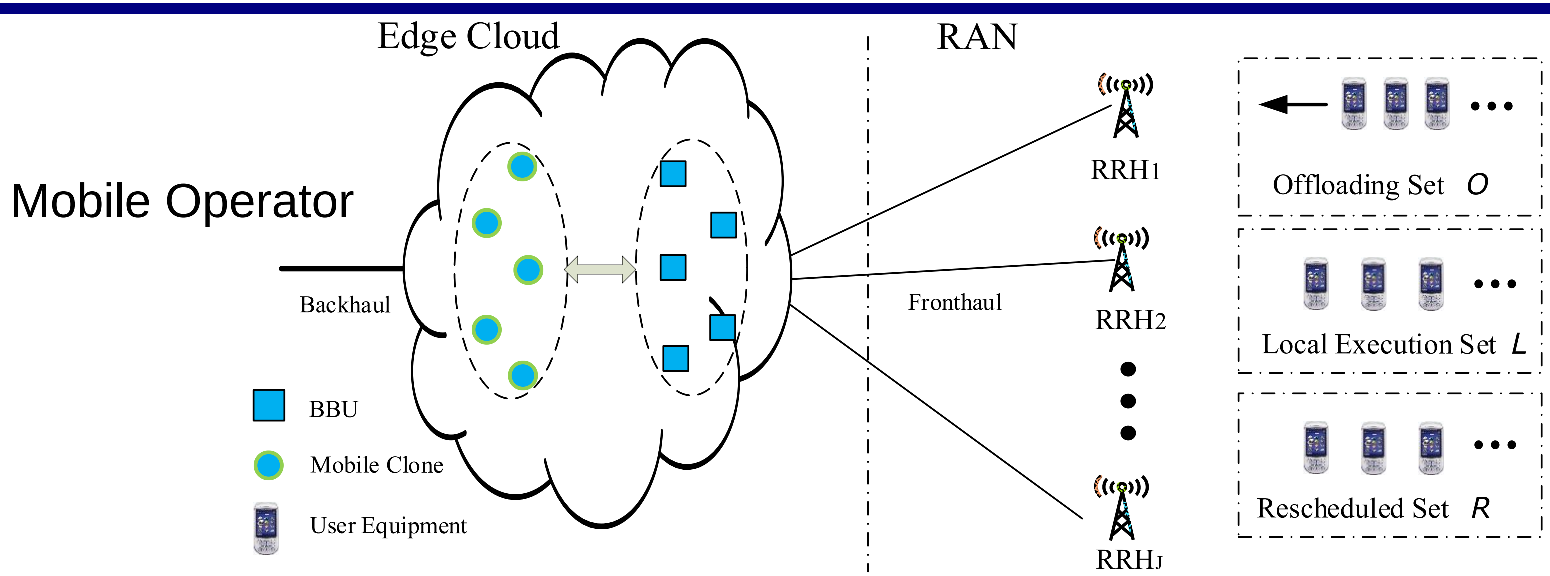
✓ Service Provider, e.g., Google, YouTube, Facebook
- Build up their service on top of the MEC
- Bring their service closer to the user, reduce latency.
- e.g. video transcoding according to the quality of the networks, 4K or 8K video?



Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV-based MEC
- Hybrid MEC systems
- Conclusions

Ground-based MEC



<u>Cloud Radio Access Network (C-RAN)</u>

BBU (Baseband Unit): Software-based signal processing unit, e.g., decoding, encoding

RRH (Remote Radio Head): like antennas.

Mobile Clone (like MEC server): For user to offload the computing tasks.

---

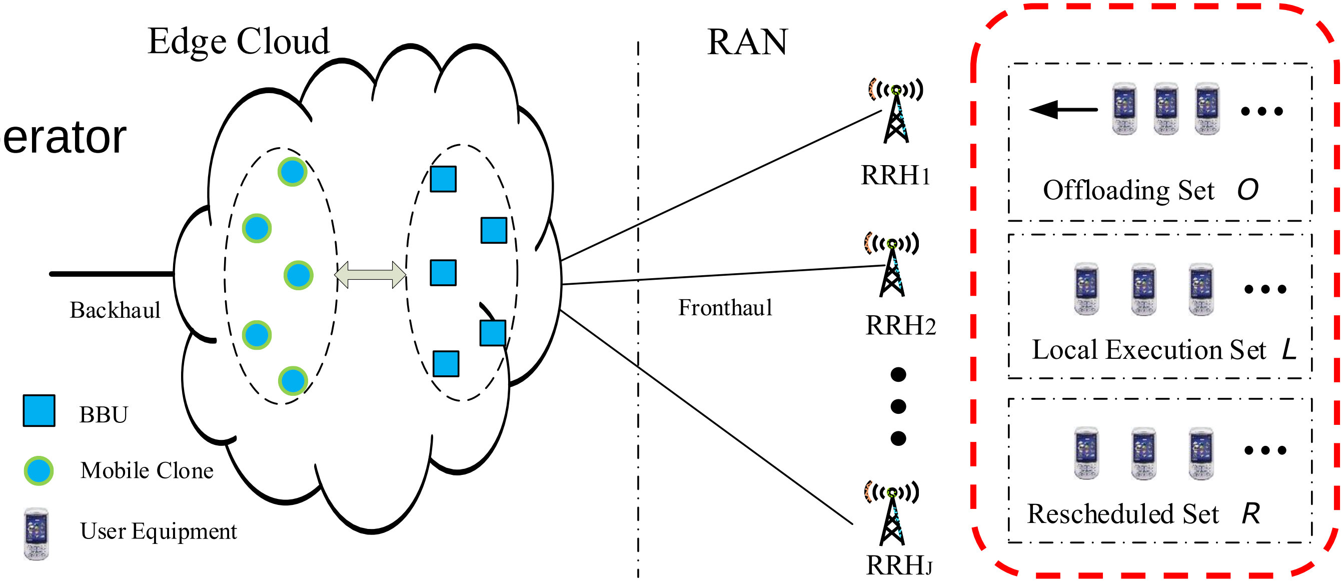
[1] <u>K. Wang</u>, et al, Unified Offloading Decision Making and Resource Allocation in ME-RAN, IEEE TVT, 2019.
[2] <u>K. Wang</u>, et al, Joint Energy Minimization and Resource Allocation in C-RAN with Mobile Cloud, IEEE TCC, 2018
[3] X. Wang, <u>K. Wang</u>, et al., "Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network," IEEE TPDS, 2018

How to reduce the energy consumption of all the

User equipments (UEs)

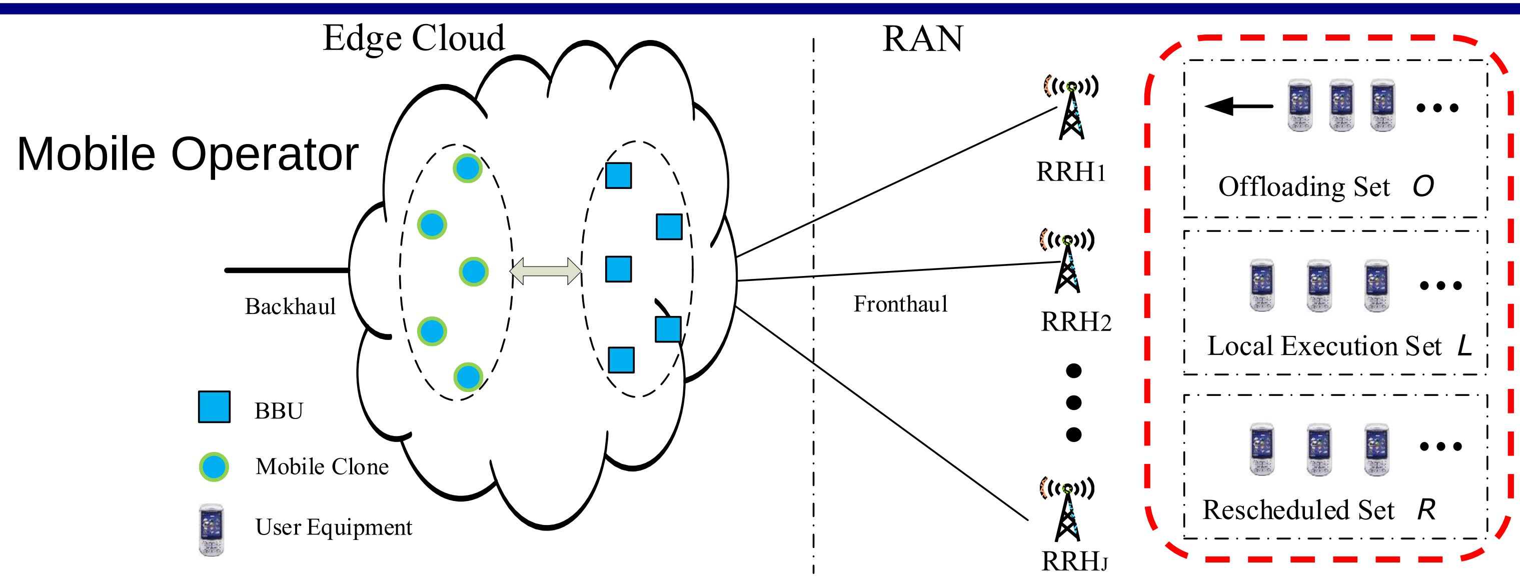
Challenges

Mobile Operator



- UEs offload the tasks.
- Resources in edge cloud are limited, not all the UEs can offload – access control.
- Not all UEs want to offload (comparing the <u>data offloading energy</u> with <u>local computing energy</u>)
- <u>Data offloading energy</u> depending on other users (who may bring interference)
- Not all UEs can complete the task locally – <u>**Offloading priority**</u>
- Reschedule set (minimize the number of UEs in this set)

Objectives



- Minimize the energy consumption of all the UEs
- Meet the QoS requirement (task deadline) of UEs
- Meet the communication and computation resource constraints

Set (Offloading, Local Execution, Rescheduled)

Resource allocation from the edge cloud

Problem formulation and solutions

Mixed-integer programming

- 1: If user choose to offload
- 0: If user decides not to offload

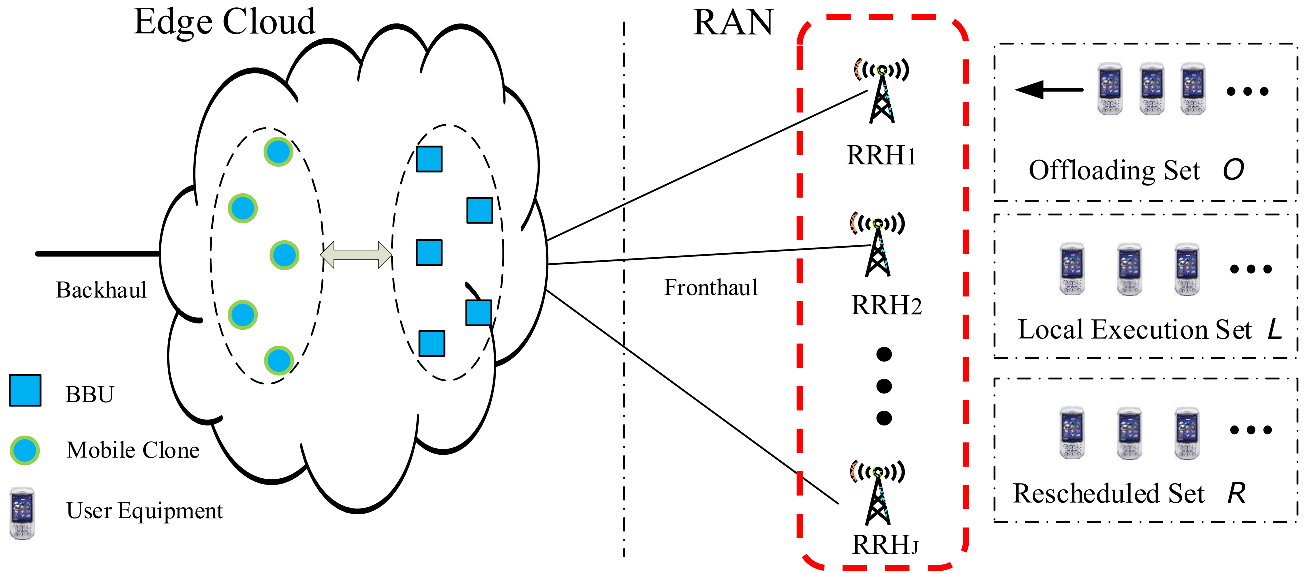
- Transformed to **<u>Second-order cone program (SOCP)</u>** **(Convex Optimization)**
- We can address it successfully with the help of some iterations.

- Machine learning based solution is also possible

How to reduce the energy consumption of

Mobile Operator

Strategy



Switch on / off RRH (sleep mode)

[4] Y. Luo, J. Yang, W. Xu, K. Wang, et al, Resource Allocation Using Gradient Boosting Aided Deep Q Network for IoT in C-RANs, available in arXiv:1910.13084, 2019

Challenges

- How to switch on/off RRHs in real-time
  - 0 for RRH off
  - 1 for RRH on
  - Integer programming
  - Non-convex problem
  - Branch-and-bound solutions
  - Exhaustive search: high complexity and time consuming

⬇

Deep Q-network (DQN) to generate a policy to control RRHs

- DQN is one type of reinforcement learning (RL)

- Apply an agent to interact with the environment at different states and select the optimal actions that can maximize the accumulated reward.

- DQN was designed to solve problem **with discrete variables (e.g., 0 or 1).**

Deep Q-network (DQN)

Deep Q-network (DQN):

- **States**: user requests + status of RRH
- **Action**: To change state of RRH (switch on and switch off)
- **Reward**: Total power / energy consumption of all the UEs

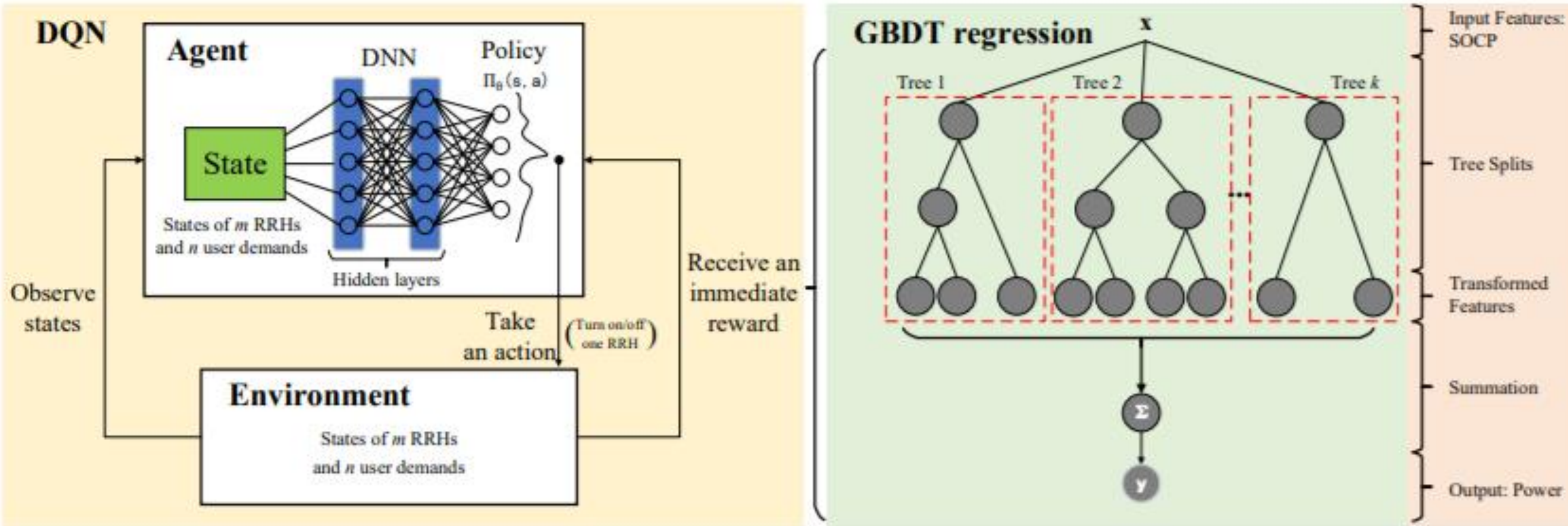
However: the **reward** is hard to get, as we have to solve Second-order cone program **(SOCP),** which is time-consuming and computational-expensive in general.

Can not be real-time

Motivating us to use an **Approximation Method**, i.e. gradient boosting decision tree (GBDT) to approximate the solutions of **SOCP.**

(GBDT is one type of machine learning and it is good at approximating complex functions**)**

Overall Architecture (DQN+GBDT)



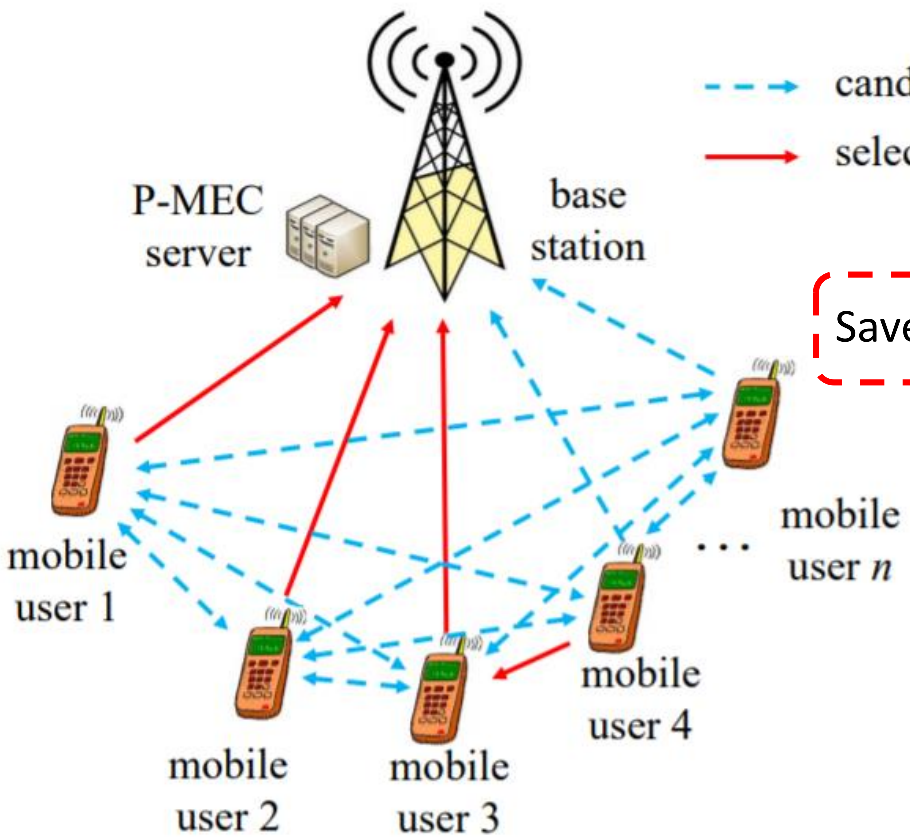
<u>Offline</u>: we first apply SOCP to generate minions of samples/solutions to train GBDT, which can provide immediate reward to DQN

<u>Online</u>: decision making and online training

Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV-based MEC
- Hybrid MEC systems
- Conclusions

Cooperative MEC



- ✓ Local device
- ✓ MEC server
- ✓ One of cooperative devices

Save energy and increase the task completion rate

Objectives

- Offloading decision
  - Whether to offload or not
  - **Where to offload**

  (search space increased)
- Resource allocation

---

[5] P. Huang, Y. Wang, K. Wang, et al, A Bilevel Optimization Approach for Joint Offloading Decision and Resource Allocation in Cooperative Mobile Edge Computing, IEEE Transactions on Cybernetics, 2019
[6] Y. Pan, C. Pan, K. Wang, et al, Cost Minimization for Cooperative Computation Framework in MEC Networks, submitted to IEEE TWC (under review), 2019

Cooperative MEC

$$\min_{\mathbf{s},\mathbf{f}} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} s_{ij} E_{ij}^{c} + \sum_{j=0, j\neq i}^{n} s_{ij} E_{ij}^{t} \right)$$

C1: $\sum_{j=0}^{n} s_{ij} = 1, \forall i \in \mathcal{N}$

C2: $\sum_{i=1}^{n} s_{ij} \leq 1, \forall j \in \mathcal{M} \backslash \{0\}$

C3: $\sum_{i=1}^{n} s_{ij} f_{i} \leq F_{j}, \forall j \in \mathcal{M}$

C4: $f_{ij} > 0, \forall s_{ij} = 1, i \in \mathcal{N}, j \in \mathcal{M}$

C5: $f_{ij} = 0, \forall s_{ij} = 0, i \in \mathcal{N}, j \in \mathcal{M}$

C6: $T_i \leq T_{i,max}, \forall i \in \mathcal{N}$

The <u>offloading decision **s**</u> is an integer variable and <u>the resources **f**</u> is continuous variable

It is a <u>mixed-variable</u> optimization

✓ Challenges

- Resource allocation strongly depends on the result of offloading decision
- It is not possible to evaluate the performance of offloading decision until resource allocation has been determined

Transformation

- Transforming it to a bi-level optimization
  - ✓ The **upper level** optimization aims to find the optimal offloading decision
  - ✓ The **lower level** optimization to find resource allocation under a given offloading decision

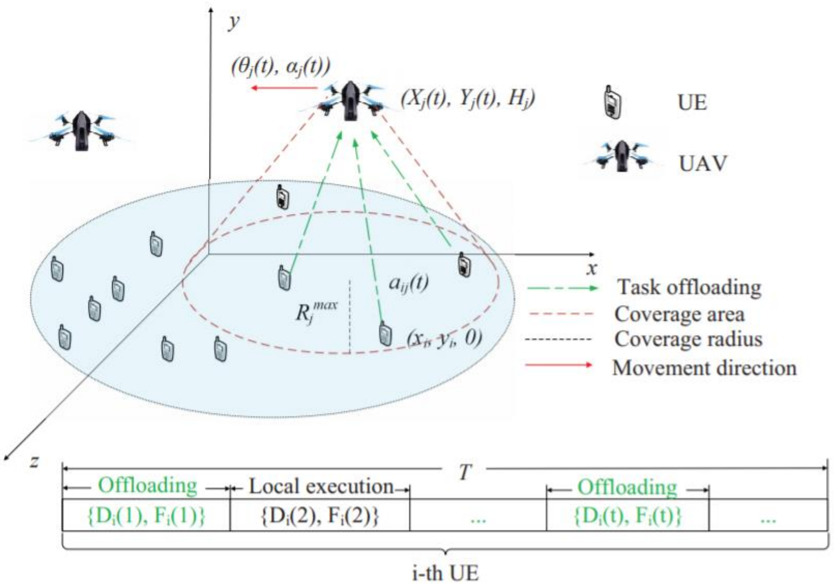
Algorithm 1 General Framework of BiJOR

```
gen = 0;
Determine the candidate execution mode sets M_1, ..., M_n for each task using Algorithm 2;
while gen < Gen_max do
    Construct np offloading decisions S = {s_1, ..., s_np} using Algorithm 3;
    Calculate the optimal resource allocations F = {f_1, ..., f_np} under the given offloading decisions;
    Evaluate the energy consumption of each offloading decision s with respect to the optimal resource allocation f;
    Perform local search on the iteration-best solution {s^ib, f^ib} using Algorithm 4;
    Update global pheromone;
    gen = gen + 1;
end while
return the optimal offloading decision and the corresponding optimal resource allocation
```

Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV-based MEC
- Hybrid MEC systems
- Conclusions

UAV-enabled MEC



UAV: Flying mobile edge computing

Objective:

- ✓ Minimize energy consumption of all the UEs
- ✓ Each UAV has limited resource
- ✓ Each UAV has limited coverage
- ✓ Each task has deadline

Jointly Optimize

- User association (which user offload to which UAV at which time slot)
- Resource allocation
- **Trajectory of each UAV**

Flying directions and distance

[7] L. Wang, K. Wang, et al., Deep Reinforcement Learning Based Dynamic Trajectory Control for UAV-assisted Mobile Edge Computing, submitted to IEEE JSAC, available in arXiv:1911.03887, 2019

Problem formulation

$$\mathcal{P}1: \min_{\boldsymbol{U},\boldsymbol{A},\boldsymbol{F}} \sum_{i=1}^{N}\sum_{j=0}^{M}\sum_{t=1}^{T} a_{ij}(t)E_{ij}(t)$$

Energy minimization of all the users

subject to:

$$a_{ij}(t) = \{0,1\}, \forall i \in \mathcal{N}, j \in \mathcal{M}', t \in \mathcal{T},$$

Offload or not at which time slot

$$\sum_{j=0}^{M} a_{ij}(t) = 1, \forall i \in \mathcal{N}, t \in \mathcal{T},$$

Each UE can only be served by at most one UAV or itself

$$0 \le \theta_j(t) \le 2\pi, \forall j \in \mathcal{M}, t \in \mathcal{T},$$

$$0 \le d_j(t) \le d_j^{\max}, \forall j \in \mathcal{M}, t \in \mathcal{T},$$

Moving direction and distance of each UAV

$$\sum_{i=1}^{N} a_{ij}(t) \le V_j^{\max}, \forall j \in \mathcal{M}, t \in \mathcal{T},$$

$$a_{ij}(t)R_{ij}(t) \le R_j^{\max}, \forall i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T},$$

Resource and coverage constraints

$$T_{ij}(t) \le T^{\max}, \ \forall i \in \mathcal{N}, j \in \mathcal{M}', t \in \mathcal{T},$$

QoS (time) requirement

$$\sum_{i=1}^{N} a_{ij}(t) f_{ij}^{\mathrm{C}}(t) \le f_j^{\max}(t), \ \forall j \in \mathcal{M}, t \in \mathcal{T}.$$

Computing resource constraints

Challenges

- ✓ Each UAV may **take off** from different locations
- ✓ Make the real-time decision
- ✓ We may not use traditional convex optimization based solutions
  - Requires iterations – high complexity and time consuming
  - Susceptible to the taking off points

  (Changing taking off locations of each UAV, we have to re-run the optimizations)

⇓

Do we have solutions which can adapt to any taking off points of each UAV?

⇓

Deep Reinforcement Learning (DRL)

Challenges

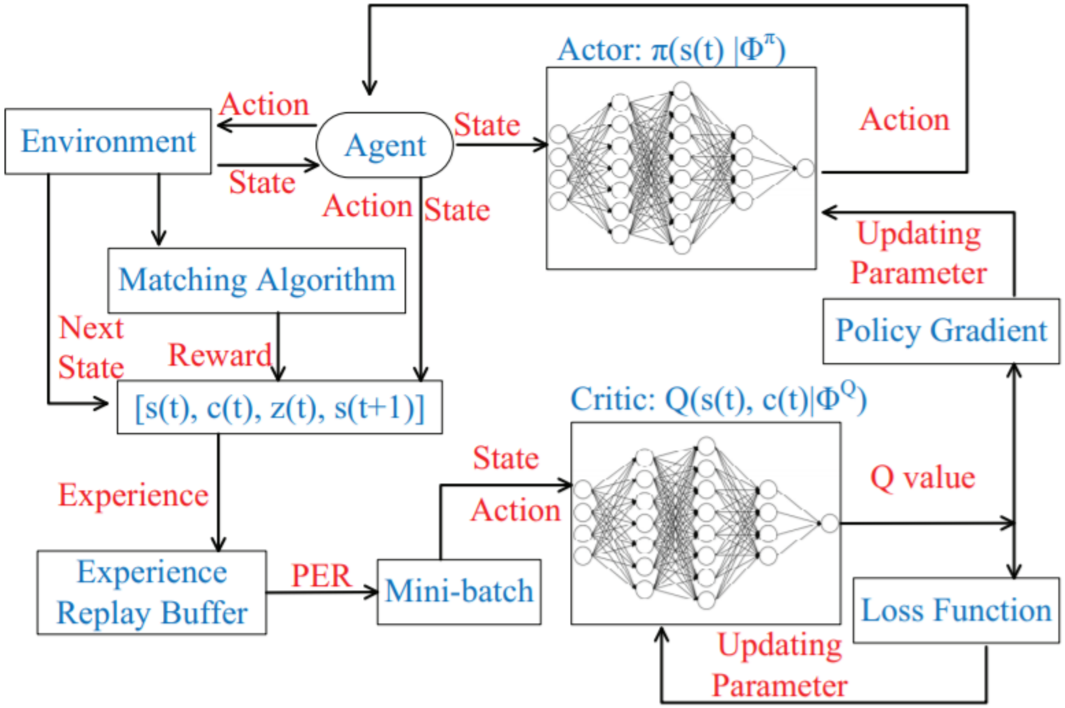
Deep Q networks (DQN) **can not** be applied here

- DQN was designed to solve the problem with discrete variables
- Our trajectory control problem – involves continuous variables

Deep deterministic policy gradient (DDPG) approach can be applied

- Continuous variables
- Actor network - deciding flying direction and distance of each UAV
- Critic network - evaluating actions generated by the actor network

DDPG based Framework



State: the set of the coordinates of all UAVs

Action: the set of the actions of all UAVs, including flying direction and distance

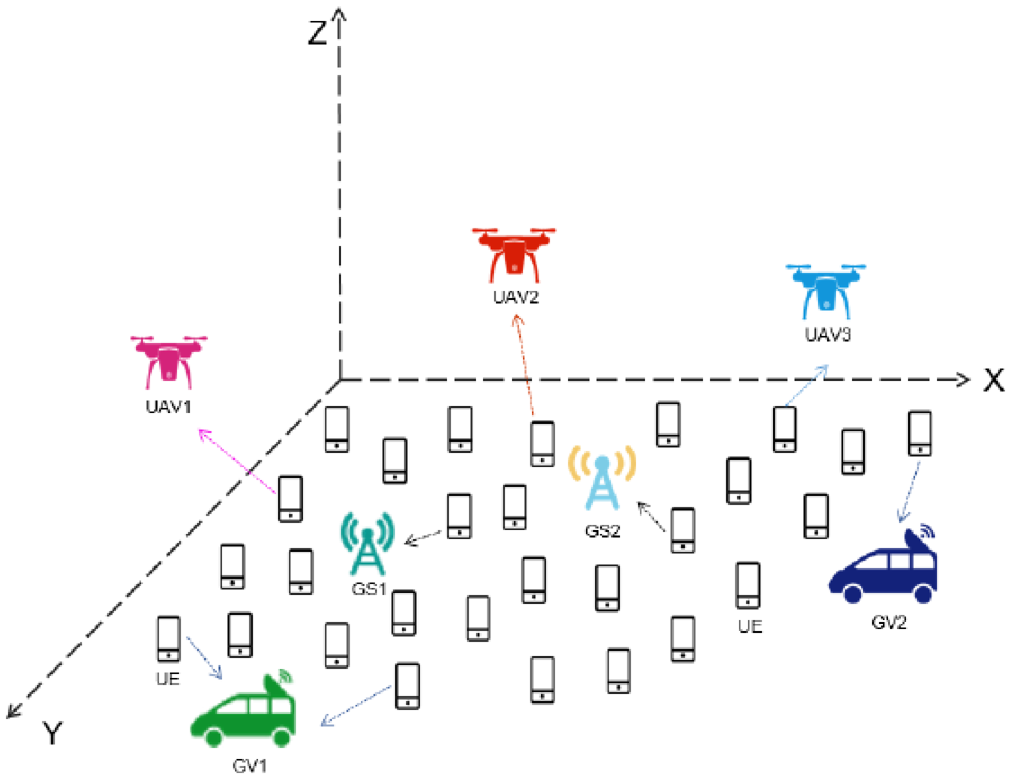
Reward: the minus of the overall energy consumption of all the UEs

- ✓ During off-line training stage, we will randomly generate many taking off point for each UAV, and then train the network to converge.
- ✓ Once training process has been completed, the solutions can be obtained quite fast, as only a few number of algebra calculations are needed.

Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV-based MEC
- Hybrid MEC systems
- Conclusions

Hybrid MEC systems



UE can offload the tasks to

- Ground stations (GSs)
- Ground vehicles (GVs)
- UAVs

Computing resource:

GS > GV > UAV

Moving speed:

UAV>GV>GS (can not move)

Price:

UAV > GV > GS

[9] F. Jiang, K. Wang, et al., "Deep Learning Based Joint Resource Scheduling Algorithms for Hybrid MEC Networks, IEEE IoT journal, 2019

Problem Formulation

Objective: minimize the energy consumption of all the users

Constraints:

- Limited resource in UAV, GS and GV
- Meet the QoS requirement of each task
- Dynamic environment (e.g., the number of UEs is changing)

↓

Jointly optimizing:

- Positions of GVs and UAVs,
- User association (integer variable)
- Resource allocation (continuous variable)

↓

Highly Dynamic and Mixed Integer Nonlinear Programming (HD-MINLP).

Challenges

Branch-and-bound algorithm
Convex based optimization

Time consuming and highly complex
Can not get real-time decision

Reinforcement Learning

difficult to converge (dynamic environment)

Deep Neural Network (DNN) based solutions

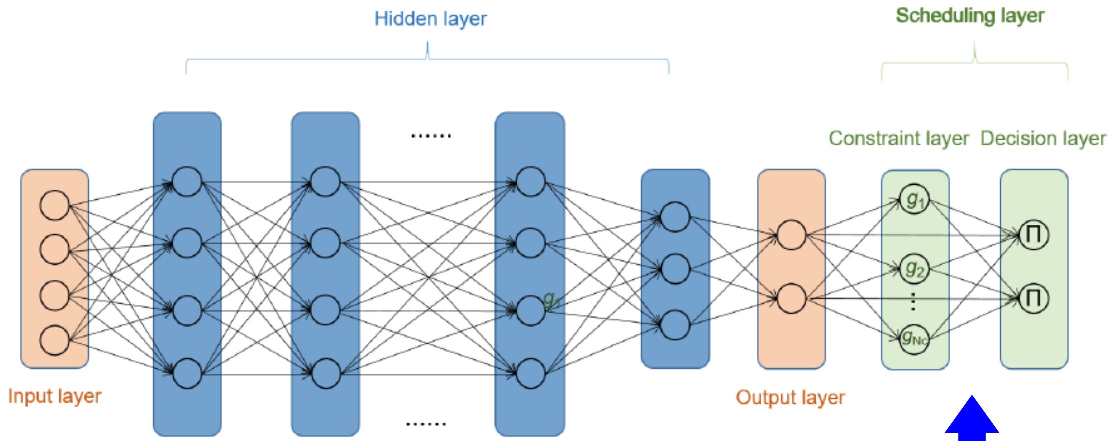
Labelled training data required

Proposed solutions

U-PSO (**membership vector** U-based Particle Swarm Optimization)

- Solve the optimizations and get the labelled sample for DNN for offline training
- The PSO will be carried out repeatedly until enough samples are collected.

DNN



DNN

- An additional scheduling layer to check the practical constraints
- Only input the membership information of one UE each time

Advantages

- **Efficient** than traditional DNN which requires to input information of all UE
- **Suitable** for dynamic scenarios (if the number of users change, we do not have to change the input structure, as we always only input the info of **one user**)

Outline

- Introduction
- Ground-based MEC
- Cooperative MEC
- UAV-based MEC
- Hybrid MEC systems
- Conclusions

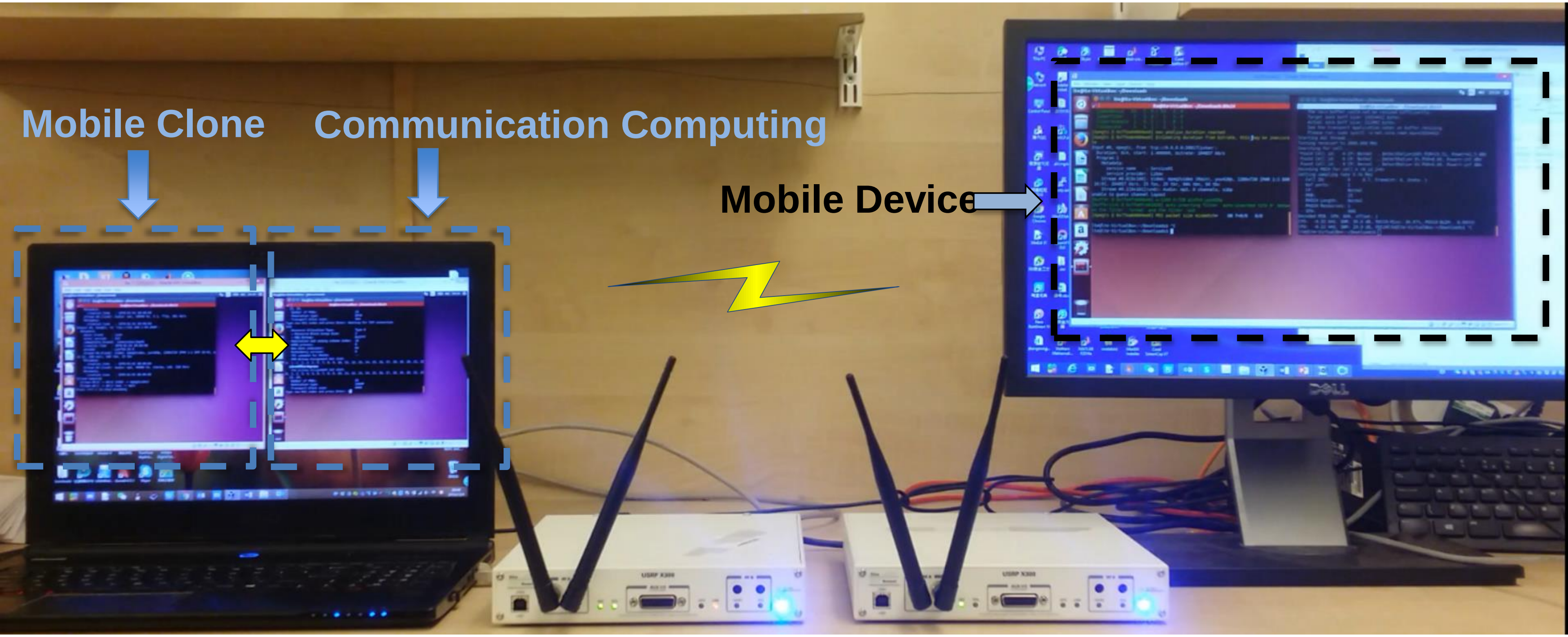
Conclusions

- Introduction to mobile edge computing (MEC)
- Different types of MEC, including-
  - Ground-based MEC
  - Cooperative MEC
  - UAV-assisted MEC
  - Hybrid MEC
- Solutions for resource allocation and user associations
  - Convex optimization
  - Machine learning
    - DQN – discrete variable
    - DDPG- continuous variable
    - DNN
    - GBDT – excellent approximation
- <u>Machine learning </u>is good at
    - -No model
    - -Model, optimization is very complex

Opportunities

- Many research challenges and opportunities
- Hardware implementation
- Real-data to train the model
- Real-time decision making in
  - Varying environment
  - Large-scale of users
- More general scenario
- Much more to be done……

Testbed

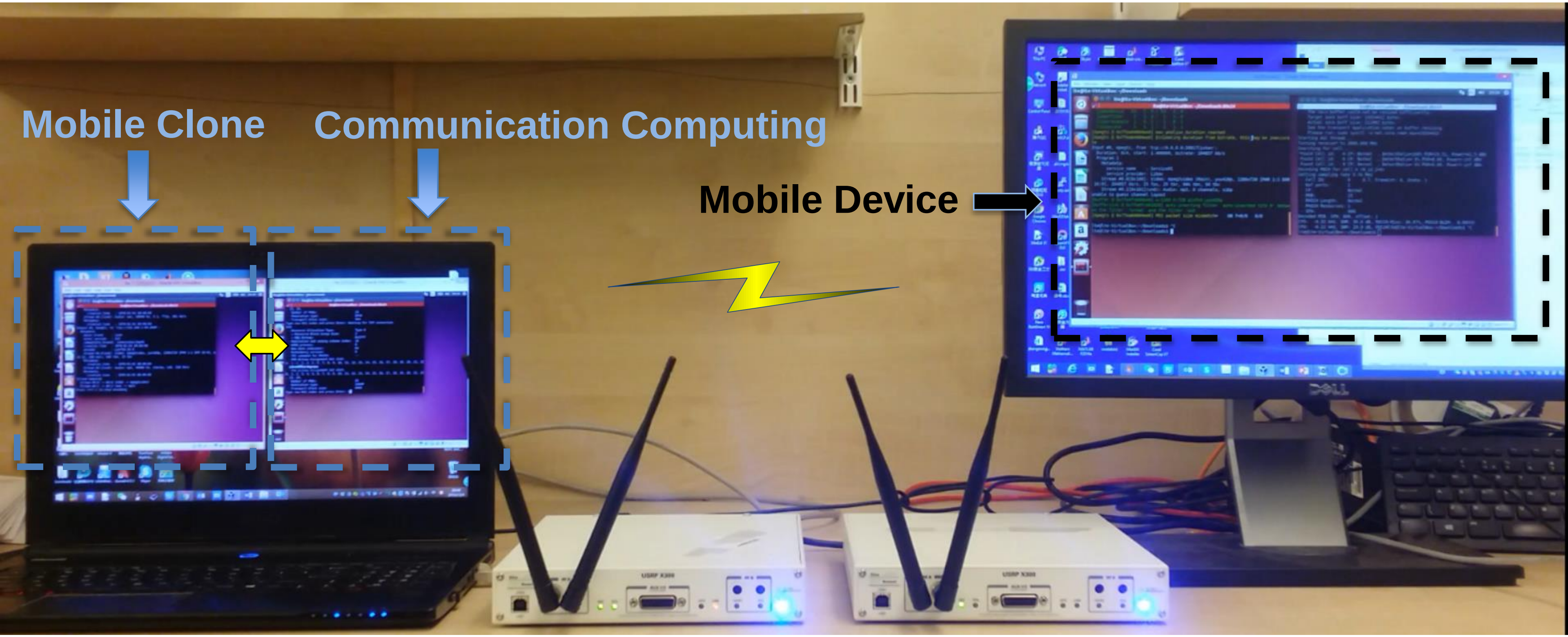


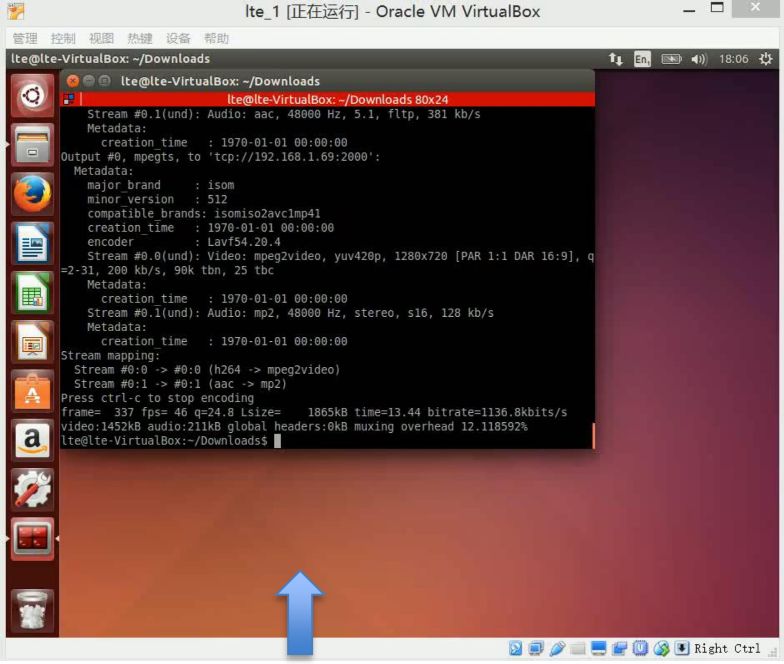
Edge Cloud **Software Defined Radio (SDR) USRP - RRH** **Mobile User**

Ismael Gomez-Miguelez, et al, srsLTE: An Open-Source Platform for LTE Evolution and Experimentation, arxiv.1602.04629, 2016

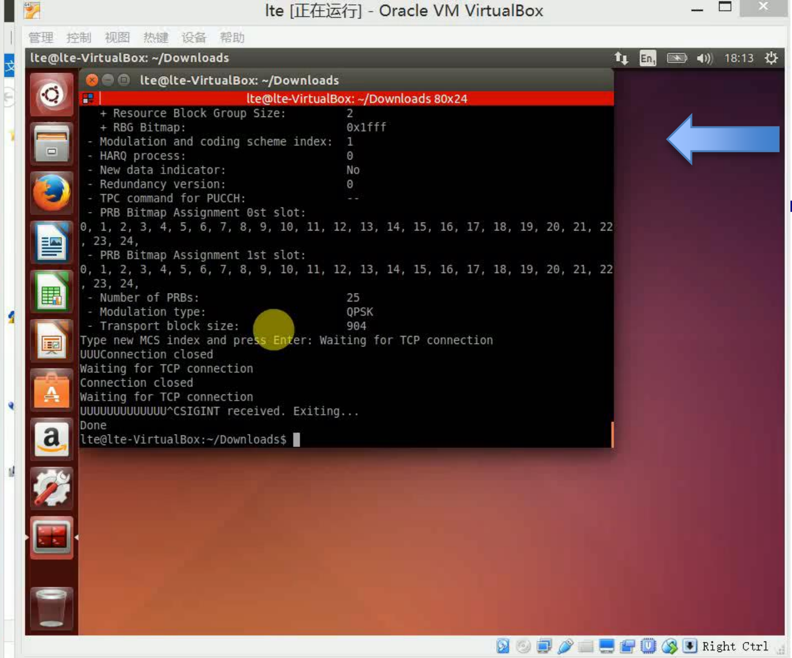
Demo

- More computing resource is allocated ⇨ higher quality of the video is received
  - Mobile clone to transcode and process the video
  - Video will be sent to mobile device via communication computing unit (BBU)





Mobile Clone

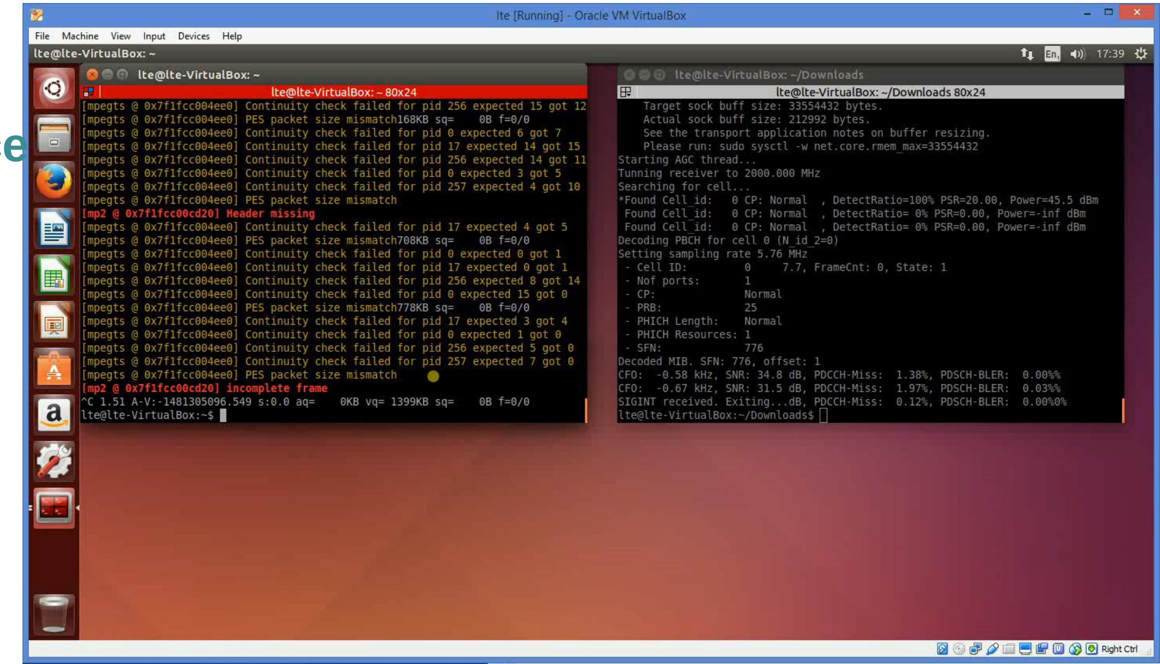


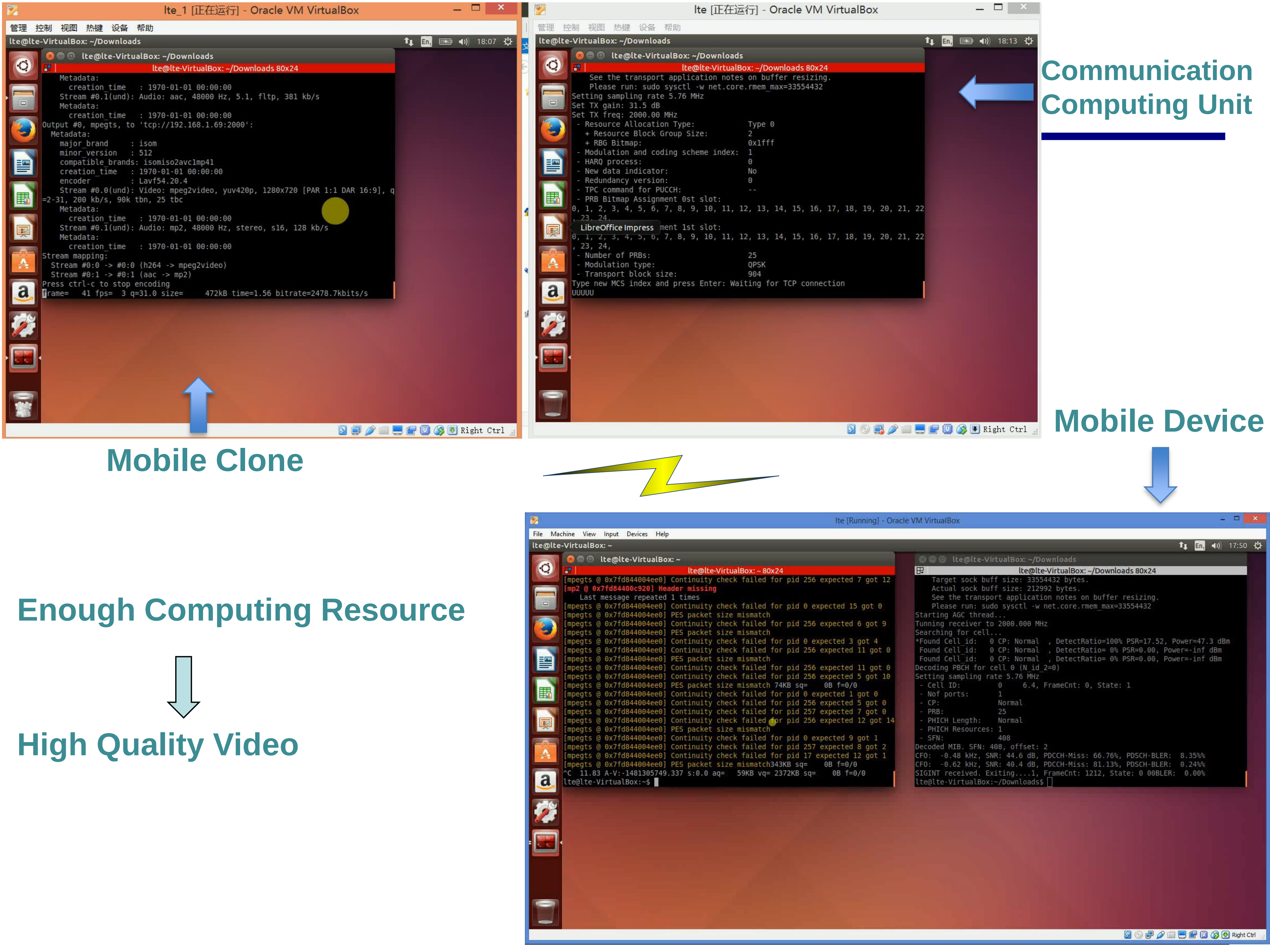
Communication Computing Unit

Mobile Device

Not Enough Computing Resource

Video can not play smoothly




lte_1 [正在运行] - Oracle VM VirtualBox
lte [正在运行] - Oracle VM VirtualBox
Communication
Computing Unit
Mobile Clone
Mobile Device
Enough Computing Resource
High Quality Video
lte [Running] - Oracle VM VirtualBox

Reference

[1] K. Wang, et al, ”Unified Offloading Decision Making and Resource Allocation in ME-RAN,” in IEEE TVT, 2019.

[2] K. Wang, et al, ”Joint Energy Minimization and Resource Allocation in C-RAN with Mobile Cloud,” in IEEE TCC, 2018

[3] X. Wang, K. Wang, et al., ”Dynamic Resource Scheduling in Mobile Edge Cloud with Cloud Radio Access Network,” in IEEE TPDS, 2018

[4] Y. Luo, J. Yang, W. Xu, K. Wang, et al, Resource Allocation Using Gradient Boosting Aided Deep Q Network for IoT in C-RANs, submitted, available in arXiv:1910.13084, 2019

[5] P. Huang, Y. Wang, K. Wang, et al, ”A Bilevel Optimization Approach for Joint Offloading Decision and Resource Allocation in Cooperative Mobile Edge Computing,” in IEEE T. CYBE, 2019.

[6] Y. Pan, C. Pan, K. Wang, et al, Cost Minimization for Cooperative Computation Framework in MEC Networks, submitted to IEEE TWC (under review), 2019

[7] L. Wang, K. Wang, et al., Deep Reinforcement Learning Based Dynamic Trajectory Control for UAV-assisted Mobile Edge Computing, submitted, available in arXiv:1911.03887, 2019

[8] F. Jiang, K. Wang, et al., ”Deep Learning Based Joint Resource Scheduling Algorithms for Hybrid MEC Networks, IEEE IoT, 2019

[9] P. Huang, Y. Wang, K. Wang, et al, ”Differential Evolution With a Variable Population Size for Deployment Optimization in a UAV-Assisted IoT Data Collection System,” in IEEE TETCI, 2019.

[10] Y. Wang, Z. Ru, K. Wang, et al ”Joint Deployment and Task Scheduling Optimization for Large-Scale Mobile Users in Multi-UAV-Enabled Mobile Edge Computing,” in IEEE T. CYBE, 2019.

[11] Z. Yang, C. Pan, K. Wang et al, ”Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks,” in IEEE TWC, 2019

[12] Y Zhou, C Pan, PL Yeoh, K. Wang, et al, “Secure Communications for UAV-Enabled Mobile Edge Computing Systems”, in IEEE TCOM, 2019.

Thank you very much!

Any questions