Mobile Edge Computing and Communications from the Sky

Prof. Kun Yang
University of Essex, Colchester, UK

IEEE ComSoc Distinguished Lecturer Series
9 December 2020 (online)
Agenda

- Mobile Edge Computing and Communications (MECC)
- Unmanned Aerial Vehicles (UAV)
- MECC+UAV

- Q&A
Tactile Internet

Our current Internet can support text, voice, video (triple play) quite well. But not touch (haptic) or smell!

Hearing: 100ms Vision: 10ms Touch: 1ms

RTT=1ms \[1\text{ms} \times (3 \times 10^8 \text{m/s}) = 300\text{km}\]
### Three Driving Forces

<table>
<thead>
<tr>
<th>Service Providers</th>
<th>Mobile Operators</th>
<th>Mobile End Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tactile Internet: Ultra-low latency (1ms)</td>
<td>Scarce spectrum mismatches increasing traffic</td>
<td>Mobile phones running out battery soon</td>
</tr>
<tr>
<td>IoT Massive Connectivity</td>
<td>Huge energy consumption</td>
<td>Real-time responses</td>
</tr>
<tr>
<td>Data privacy</td>
<td>Small cell issues</td>
<td>Limited capacity of mobile terminals in terms of processing, memory, etc.</td>
</tr>
<tr>
<td>New services: 3D/VR/AR, interactive gaming, etc.</td>
<td>Versatile vendor protocols</td>
<td></td>
</tr>
</tbody>
</table>

- **Scarce spectrum mismatches increasing traffic**
- **Huge energy consumption**
- **Real-time responses**
- **Limited capacity of mobile terminals in terms of processing, memory, etc.**

![Diagram showing Base Station Density and battery level of a mobile phone.](image)
Overall Solution: 3C at Edge

Mobile Edge Computing MEC, Edge Intelligence

Pre-coding, M-MIMO, pre-caching, etc. – comm. requirements

Ultra-low latency (<1ms)
Ultra reliability (Outage rate: 3s/year)

Communication Support
Task uploading and result downloading

Communication

Supports

Cloud-enabled Computation

Computation

Assists

Task computation

System Architecture:

Virtualization of both networks and terminals

- Mobile Clone (e.g., Android VM)
- Task offload
- D2C/C2C
- Edge Intelligence

- RRH: Remote Radio Head
- BBU: Base-band Unit
- C-RAN: Cloud Radio Access Network

- Message format and protocols
- Joint resource allocation
Computation Capacity vs Communication Quality

To finish signal processing within 3ms, at least 2.7GHz CPU processing capacity is needed*

**Scenario-specific experimental formula:** 
\[ P_{u,t} = \left( 30A_{u,t} + 10A_{u,t}^2 + 20 \frac{M_{u,t}}{6} C_{u,t} L_{u,t} \right) \cdot \frac{R_{u,t}}{50} \]

Unit: GOPS


Experiments

LTE RTT: 8ms  
=5ms (signal transmission) +3ms (signal processing)
Computation on SDN Networks

\[ P^O = 1 - Q \left( \frac{1}{f^S - p \cdot \lambda - \mu} - \rho \right) \]

Outage probability
Computation on C-RAN and Mobile Clone

\[ r_i = B_i \log(1 + SINR_i) \]

\[ r_i = F \left( f_i^T, SINR_i, B_i \right) \]

Consider there are $\mathcal{N} = \{1, 2, \ldots, N\}$ UEs and $\mathcal{L} = \{1, 2, \ldots, L\}$ RRHs, each of which has $K \geq 1$ antennas.

The mobile cloud is co-located with the BBU and is responsible for computational intensive tasks offloaded by UEs.

BBU is in charge of returning the execution results to the UE via RRHs.

Assume each of UE $i$ has the computational intensive task $U_i = (F_i, D_i)$, $i = 1, 2, \ldots, N$ to be accomplished in the mobile cloud.

Task \( i \) (each UE has only one task): \( U_i = (F_i, D_i), \ i = 1, 2, ..., N \)

- Time spent to complete this task: \( T_i^{C} = \frac{F_i}{f_i^{C}} \)

- Energy consumption: \( E_i^{C} = \kappa_i^{C} (f_i^{C})^\nu_i^{C} - 1 F_i \)

- Computation capacity constraint: \( f_i^{C} \leq f_{i, max}^{C} \)

- UE \( i \) data rate: \( r_i = B_i \log (1 + \text{SINR}_i) \)

- Time cost: \( T_i^{Tr} = \frac{D_i}{r_i} \)

- Energy consumption: \( E_i^{Tr} = p_i \cdot T_i^{Tr} = \frac{p_i D_i}{r_i} \), where \( p_i = \sum_{j \in C} |v_{ij}|^2 \), \( v_{ij} \) is the beamforming vector from \( j \)-th RRH to the \( i \)-th UE, \( C \) is the RRH cluster serving the UE.

- Power constraint for each RRH \( j \): \( \sum_{i=1}^{N} |v_{ij}|^2 \leq P_j, \ j = 1, 2, ..., L. \)
**QoS Requirement**

- **Fronthaul capability:**
  \[
  C_j = \sum_{i=1}^{N} |v_{ij}|^2 \cdot r_i \cdot |v_{ij}|^2 = \begin{cases} 
  0, & \text{if } |v_{ij}|^2 = 0 \\
  1, & \text{otherwise}
  \end{cases}
  \]

- **Fronthaul constraint:**
  \[ C_j \leq C_{j,\text{max}} \]

- **Total time cost:**
  \[ T_i = T_{i,\text{Tr}} + T_{i,C} \]

- **Time constraint:**
  \[ T_i \leq T_{i,\text{max}} \]

- **Total energy cost:**
  \[ E_i = E_{i,C} + \eta_i E_{i,\text{Tr}} \]
Joint Optimization Problem

\[ \text{minimize} \quad \sum_{i=1}^{N} E_i^C + \eta_i E_i^{Tr} \]

\[ \text{subject to} \quad f_i^C \leq f_{i,max}, \quad i = 1, 2, \ldots, N, \quad \text{Cloud computation constraints} \]

\[ \sum_{i=1}^{N} |v_{ij}|^2 \leq P_j, \quad j = 1, 2, \ldots, L. \quad \text{RRH Power constraints} \]

\[ \sum_{i=1}^{N} |v_{ij}|^2 \cdot r_i \leq C_{j,max}, \quad j = 1, 2, \ldots, L. \quad \text{Fronthaul constraints} \]

\[ r_i \geq R_{i,min}, \quad i = 1, 2, \ldots, N, \quad \text{Achievable rate constraints} \]

\[ T_i^C + T_i^{Tr} \leq T_{i,max}, \quad i = 1, 2, \ldots, N, \quad \text{QoS constraints} \]

- Non-convex problem
- Transform to weighted minimum mean square error (WMMSE) solution
- An iterative algorithm to solve this problem

Offload or Not?

Decision Making

\[
\min \sum_{u \in U} \{c_u - d_u\}^+ \\
\text{s.t.} \sum_{v \in V} x_{uv} + \sum_{b \in B} y_{ub} + \sum_{t \in \mathcal{L}} z_{ut} = 1 \quad \forall u \in U \\
\sum_{u \in U} x_{uv} \leq \gamma_v - \gamma_v^0, \quad \forall v \in V \\
\sum_{u \in U} y_{ub} \leq \gamma_b - \gamma_b^0, \quad \forall b \in B \\
x_{uv}, y_{ub}, z_{ut} = 0 \text{ or } 1, \quad \forall u \in U, v \in V, b \in B
\]  

Joint Assignment

Minimize # of failures

- non-linear integer programming problem
- Matching theory

More research

- Offloading + resource allocation


- Resource scheduling considering user requirement and fronthaul

**Testbed: openStack + USRPs**

- Amarisoft LTE 100 software Base Station for BBUs and EPC.
  - Android OS running on Mobile Devices (Mi’s RedNote).
- *openAirInterface* in open source
  - USRP X300/X310
  - Huaiwei HDD handset
- Android x86 OS running on Clone.
- Clones are hosted on an Openstack cloud.

Ack: Chathura M.
Cloud-Edge Collaboration

- Task offloading
- Joint resource allocation
**Edge Intelligence (EI)**

**Key:** how to simplify and distribute conventional AI algorithms/techniques, which are typically used on big servers, so as to be applicable on small edge servers.
EI Technique Example #1: Early Exit


Have early exit points to end the original complex deep learning algorithms while satisfying: 1) computation and communication constraints of edge nodes; 2) task QoS (e.g., accuracy, response time)
EI Technique Example #2: Federated Learning
Agenda

- Mobile Edge Computing and Communications
- Unmanned Aerial Vehicles (UAV)
- MECC+UAV
- Q&A
An unmanned aerial vehicle (UAV) (commonly known as a drone) is an aircraft without a human pilot on board.

Unmanned aircraft system (UAS): includes a UAV, a ground-based controller, and a system of communications between the two.
# UAV Classification: by Size

<table>
<thead>
<tr>
<th>UAS Description</th>
<th>Weight (Pounds)</th>
<th>Overall Size (Feet)</th>
<th>Mission Altitude (Feet Above the Surface)</th>
<th>Mission Speed (Miles per Hour)</th>
<th>Mission Radius (Miles)</th>
<th>Mission Endurance (Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nano</td>
<td>&lt; 1</td>
<td>&lt;1</td>
<td>&lt;400</td>
<td>&lt;25</td>
<td>&lt;1</td>
<td>&lt;1</td>
</tr>
<tr>
<td>Micro</td>
<td>1 to 4.5</td>
<td>&lt;3</td>
<td>&lt;3,000</td>
<td>10 to 25</td>
<td>1 to 5</td>
<td>1</td>
</tr>
<tr>
<td>Small UAS</td>
<td>4.5 to 55</td>
<td>&lt;10</td>
<td>&lt;10,000</td>
<td>50 to 75</td>
<td>5 to 25</td>
<td>1 to 4</td>
</tr>
<tr>
<td>Ultralight Aircraft*</td>
<td>55 to 255</td>
<td>&lt;30</td>
<td>&lt;15,000</td>
<td>75 to 150</td>
<td>25 to 75</td>
<td>4 to 6</td>
</tr>
<tr>
<td>Light Sport Aircraft*</td>
<td>255 to 1320</td>
<td>&lt;45</td>
<td>&lt;18,000</td>
<td>75 to 150</td>
<td>50 to 100</td>
<td>6 to 12</td>
</tr>
<tr>
<td>Small Aircraft*</td>
<td>1,320 to 12,500</td>
<td>&lt;60</td>
<td>&lt;25,000</td>
<td>100 to 200</td>
<td>100 to 200</td>
<td>24 to 36</td>
</tr>
<tr>
<td>Medium Aircraft*</td>
<td>12,500 to 41,000</td>
<td>TBD</td>
<td>&lt;100,000</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

# UAV Classification: By Mechanism

<table>
<thead>
<tr>
<th>Mechanism</th>
<th><strong>Fixed-Wing</strong></th>
<th><strong>Rotary-Wing</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lift generated using wings with forward airspeed</td>
<td>Lift generated using blades revolving around a rotor shaft</td>
<td></td>
</tr>
<tr>
<td>Pros</td>
<td>Simpler structure, usually higher payload, higher speed</td>
<td>Can hover, able to move in any direction, vertical takeoff and landing</td>
</tr>
<tr>
<td>Cons</td>
<td>Need a runway or a launcher for takeoff and landing; need to maintain forward motion</td>
<td>Usually lower payload, lower speed, shorter range</td>
</tr>
</tbody>
</table>
UAV Classification: by Degrees of Autonomy

Remote control by a human operator

Controlled autonomously by onboard computers

UAV Swarm: formation and more intelligence
UAV Classifications: By Applications

- Military uses: Reconnaissance, attack, demining, target practice,…

Pictures are from https://www.dji.com/
### How to have my own UAVs?
**To Buy or To Build?**

- **3D-printed UAV**
  - U of Southampton

<table>
<thead>
<tr>
<th>Layer</th>
<th>Operations</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Firmware</strong></td>
<td>From machine code to processor execution, memory access</td>
<td>ArduCopter-v1, PX4</td>
</tr>
<tr>
<td><strong>Middleware</strong></td>
<td>Flight control, navigation, radio management</td>
<td>Cleanflight, ArduPilot</td>
</tr>
<tr>
<td><strong>Operating system</strong></td>
<td>Optic flow, obstacle avoidance, SLAM, decision-making</td>
<td>ROS, Nuttx, Linux distributions, Microsoft IOT</td>
</tr>
</tbody>
</table>
**UAV-enabled Wireless Communication**

UAV-aided ubiquitous coverage

- Ground gateway
- Core network
- Overloaded base station
- Malfunctioning base station

UAV-aided relaying

Wireless Communications/5G for UAVs

- **extended Mobile Broadband (faster)**
- **Vehicular Network**
- **massive Machine-Type Communications**
- **1 Million/km^2**
- **Ultra Reliable Low Latency Comm.**
- **Remote Surgery**

**Networks:**
- 5G
- 4G
- 1Gbps
- 20Gbps
- 100K
- 1ms
- 10ms
- 100K

**Applications:**
- VR/AR
- Smart City
- IoT
- Industrial IoT
- IoT Industrial
- Remote Surgery
- UAV/drones

**5G Benefits:**
- Faster speeds
- Low latency
- High capacity
Agenda

- Mobile Edge Computing and Communications
- Unmanned Aerial Vehicles (UAV)
- MECC+UAV

- Q&A
A UAV moves in a constant altitude $H$, TDD communication
- UAV hovers above $N$ IoT devices at $M$ different locations.
- With the help of wireless powering technique, IoT devices can be charged by UAV
- **Objective:** Minimize the energy consumption of the UAV

Inductive Wireless Charging (short distance)

- Smart phones run out of battery soon.

**Inductive coupling**
- Strict coils matching
- Short distance: mm-cm
- Frequency: Hz-MHz, suitable for small devices

**Magnetic resonant coupling**
- Distance: tens of cms; not very strict coil alignment
- Frequency: KHz-MHz
Radio Frequency Wireless Charging (medium distance)

Charging IoT devices

Energous WattUp RF wireless charging: pushing standard 2.0, can be as far as 4.6m

Space solar energy to ground
Comparison

From Prof. R. Zhang’s tutorial in IEEE Globecom 2016
Problem Formulation

- **Channel Model**: assume line of sight (LoS), perfect Doppler compensation

\[ h_{ij} = \frac{h_0}{d_{ij}^2} \quad r_{ij} = a_{ij} B \log_2 \left( 1 + \frac{p_i h_{ij}}{\sigma^2} \right), \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \]

- **Wireless Powering Model**: the harvested energy should be more than the uploading energy each IoT device consumes.

\[ E_{ij}^W = v_i h_{ij} P_{uav}^W t_{ij}^w \geq p_i t_{ij}^u, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \]

- **Computing Task Model**: UAV is required to provide sufficient computing resources for each IoT device

\[ \sum_{j=1}^{M} a_{ij} f_{ij} t_{ij}^c \geq F_i, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \]
Problem Formulation

\[
\begin{align*}
\text{minimize} & \quad \varphi(E^C + E^W) + \phi E^H \\
\text{s.t.} & \quad a_{ij} = \{0, 1\}, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \\
& \quad \sum_{j=1}^{M} a_{ij} = 1, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \\
& \quad 0 \leq f_{ij} \leq f_{\max}, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \\
& \quad v_i h_{ij} P_{\text{uav}}^W t_{ij}^w \geq p_i t_{ij}^v, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \\
& \quad a_{ij} (t_{ij}^w + t_{ij}^v + t_{ij}^c) \leq t_i^{gos}, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \\
& \quad T_j \geq \sum_{i=1}^{N} a_{ij} (t_{ij}^w + t_{ij}^v + t_{ij}^c), \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M}
\end{align*}
\]

\(A\) : IoTDs association; \(F\) : computing resources allocation; \(\tau\) : WPT time; \(T\) : UAV hovering duration

Note: UAV locations pre-defined and fixed.
The optimal wireless powering duration and hovering time are

$$t_{ij}^w = \frac{p_i t_{ij}^u}{v_i h_{ij} P_{uav}^W}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M}$$

$$T_j^* = \sum_{i=1}^{N} a_{ij}(t_{ij}^w + t_{ij}^u + t_{ij}^c), \forall j \in \mathcal{M}$$

The proposed iterative algorithm (i.e., block-coordinate descent method) can give a near-optimal solution to the Joint Resources Allocation problem.
To further minimize the hovering time of UAV, we propose the multiple-workflow structure for UAV-assisted IoT platform. This TDMA based workflow allows parallel transmissions and executions on different devices. The hovering time of UAV is minimized and the QoS of each IoTD is guaranteed at the same time.
## Problem Formulation

**Minimize** 
\[ \varphi(E^C + E^W) + \phi P^{uav}_u \sum_{j=1}^{M} c_{Kj}^c \]  

**subject to** 
\[ a_{ij}(t_{ij}^w + t_{ij}^u + t_{ij}^c) \leq t_{ij}^{gos}, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]

\[ 0 \leq a_{ij} \leq 1, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]

\[ \sum_{j=1}^{M} a_{ij} = 1, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]

\[ 0 \leq f_{ij} \leq f_{\text{max}}, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]

\[ t_{ij}^w \geq \frac{p_i t_{ij}^u}{v_i h_{ij} P^{w}_{uav}}, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]

\[ S_j \in \mathcal{S}_j, \quad \forall j \in \mathcal{M} \]

\[ \sum_{j=1}^{M} a_{ij} f_{ij} t_{ij}^c \geq F_i, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]

\[ \tau : \text{WPT time; } S : \text{service sequence of IoTDs} \]

\( c_{Kj}^c \) is the last computing task completion time in each j-th hovering place.
Hovering Time Minimization Algorithm

**Algorithm 3** Optimal UAV hovering time minimization algorithm

1: Given the optimal $t_{ij}^u$, $t_{ij}^c$ and $t_{ij}^w$, let $n = 0$, $m = 0$;
2: **Repeat**:
3: Find the minimal time $t_{k_{min}}$ in time sets $T_1$ and $T_2$;
4: If $t_{k_{min}} \in T_1$ then $S_{n+1} = k_{min}$, $n = n + 1$;
5: else if $t_{k_{min}} \in T_2$ then $S_{K-m} = k_{min}$, $m = m + 1$;
6: Remove $t_{k_{min}}^f$ from $T_1$ and $t_{k_{min}}^c$ from $T_2$;
7: **Until** $T_1$ and $T_2 = \emptyset$;
8: Update $n=1$ and $s_{1}^f = 0$;
9: **Repeat**:
10: If $t_{n+1}^f \geq t_{c}^c$ then $c_{n+1}^c = s_{n}^f + t_{n}^f + t_{n+1}^f + t_{n+1}^c$ and $s_{n+1}^f = s_{n}^f + t_{n}^f$;
11: If $t_{n+1}^f < t_{c}^c$ then $c_{n+1}^c = s_{n}^f + t_{n}^f + t_{n}^c + t_{n+1}^c$ and $s_{n+1}^f = s_{n}^f + t_{n}^f + t_{n}^c - t_{n+1}^f$;
12: $n = n + 1$;
13: **Until** $n = K$;
14: $T_j^* = c_{k_j}^c$;
15: **Return**: The optimal services sequence $S^*$ and the optimal UAV hovering durations $t^*$ for $P1$.

- The two-stage flow-shop problem can be solved by Johnson’s algorithm.
- Based on the traditional Johnson’s algorithm, we develop the novel hovering time minimization algorithm.
Even if the multi-workflow system is not scheduled, the hovering time of UAV is significantly reduced compared with single workflow system.
Algorithm 4 The proposed iterative algorithm for multi-workflow system

1: **Initialize:** $A^0$ and $S^0$. Let $r = 0$.
2: **Repeat:**
3: For given $\{A^r, S^r\}$, obtain the optimal solution denoted as $F^{r+1}$;
4: For given $\{S^r, F^{r+1}\}$, obtain the optimal solution denoted as $A^{r+1}$;
5: For given $\{A^{r+1}, F^{r+1}\}$, use Algorithm 3 to obtain the optimal solution denoted as $S^{r+1}$ and $T^{r+1}$;
6: Update $r = r + 1$.
7: **Until:** The fractional decrease of $E$ is below a threshold $\epsilon$ or a maximum number of iterations ($r_{\text{max}}$) is reached;
8: **Return:** The optimal IoTDS association $A^*$, computing resources allocation $F^*$, UAV hovering durations $T^*$ and the services sequence of the IoTDS $S^*$.

- The optimal wireless powering duration and hovering time are
  \[ t_{ij}^{w*} = \frac{p_i t_{ij}^{u}}{v_i h_{ij} P_{uav}^{W}}, \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \]
  \[ T_j^{*} = c_{Kj}^{c*}, \forall j \in \mathcal{M} \]

- The proposed iterative algorithm (i.e., block-coordinate descent method) can give a near-optimal solution to the Joint Resources Allocation problem.
Multiple UAVs

Two kinds of drones: the LAPs (UAVs) and a solar-powered HAP serving as energy charging stations for all the LAPs.

- $M$ LAPs hovers above $K$ Desired Regions (DRs). For example, the DRs in smart cities may include remote factories, farms and crowded buildings, etc.

Objective: Minimize the energy consumption of the system.

---

Problem Formulation

- **UAVs mobility constraints:**

  \[ q_j[S_j] = q_j[0], \ \forall j \in \mathcal{M} \]

  \[ \sum_{j=1}^{M} \sum_{t=1}^{S_j} a_j^k[t] = 1, \ \forall j \in \mathcal{M}, \ \forall k \in \mathcal{K}, \ \forall t \in \mathcal{T}_j \]

  \[ d_j[t] = ||q_j[t] - q_j[t - 1]||, \ \forall t \in \mathcal{T}_j, \ \forall j \in \mathcal{M} \]

  \[ \mathcal{T}_j = \{1, 2, ..., S_j\}, \ \forall j \in \mathcal{M} \]

  \[ q_j[0] \] : initial location of \( j \)-th UAV

  \[ q_j[S_j] \] : final location of \( j \)-th UAV

  \[ d_j[t] \] : distance between two destinations (significantly affected by the order \( t \))

- **Laser Powering Model:** The laser energy that each LAP receives from the HAP should be enough for its flight

  \[ \eta_j = \frac{P_j^r}{P_j^l} = e^{-\alpha_j(H-h)} \]

  \[ \eta_j P_j^l \tau_j \geq E_j, \ \forall j \in \mathcal{M} \]
**Problem Formulation**

\[ \mathcal{P}1: \text{minimize} \sum_{j=1}^{M} \eta_j P_j^l \tau_j \]

subject to:

- \( E_j \leq C_j, \ \forall j \in \mathcal{M} \)
- \( \eta_j P_j^l \tau_j \geq E_j, \ \forall j \in \mathcal{M} \)
- \( q_j[S_j] = q_j[0], \ \forall j \in \mathcal{M} \)
- \( \sum_{j=1}^{M} \sum_{t=1}^{S_j} a_j^k[t] = 1, \ \forall j \in \mathcal{M}, \ \forall k \in \mathcal{K}, \ \forall t \in \mathcal{T}_j \)
- \( d_j[t] = \|q_j[t] - q_j[t - 1]\|, \ \forall t \in \mathcal{T}_j, \ \forall j \in \mathcal{M} \)
- \( \mathcal{T}_j = \{1, 2, ..., S_j\}, \ \forall j \in \mathcal{M} \)

Minimize energy consumption of system/HAP

UAV battery capacity as powered by laser

Laser charging constraints

UAVs trajectory constraints

\( \mathcal{A} \): selection of DRs; \( \mathcal{S} \): the number of DRs selected by each LAP;
\( \tau \): Laser charging duration; \( \mathcal{Q} \): multi-LAP routes.
Trajectory Design

An Example of Trajectory Design for London
The trajectory design problem is a typical one deposit multiple traveling salesmen problem (TSP) with the time window. We rename it as Drones Traveling Problem (DTP).

Our proposed DTA only uses 5 iterations (fast convergence) to obtain the near-optimal solution whereas the normal Genetic Algorithm needs nearly 10000 iterations and still fails to obtain an acceptable solution.
Drones Traveling Algorithm

- Solving DTP is equivalent to obtaining the optimal direct graph $s^*$;

- We define the local optimum graph as the graph with no self-knot in each LAP cycle;

- We use the efficient and effective 2-opt algorithm to transform a graph into a local optimum graph;

- We develop an efficient method to jump from the local optimum graph to a better graph. The Jump operations include three simple operations: the Exchanging, Shifting and the Knot Removing.
Drones Traveling Algorithm

Algorithm 1: The Drones Traveling Algorithm (DTA) for \( \mathcal{P}1 \)

Input: \( A^0, Q^0, S^0 \): Random Solutions
Output: \( A^*, Q^*, S^*, \tau^* \): The Near-optimal Solutions

1. \( r \leftarrow 1 \); // the iterations of DTA
2. Generate a direct graph \( s^r \) using \( A^0, Q^0, S^0 \);
3. \( f(s^0) \leftarrow \infty \); // initialize the utility function
4. while \( f(s^r) < f(s^{r-1}) \) do
   5. Use the 2-opt algorithm to obtain the local optiaml direct graph \( s^r \);
   6. Jump from graph \( s^r \) to a new direct graph \( s^{r+1} \);
   7. \( r \leftarrow r + 1 \);
5. end
6. Use direct graph \( s^r \) to obtain \( A^*, Q^*, S^* \);
7. Obtain the optimal \( \tau^* \);
8. return: \( A^*, Q^*, S^*, \tau^* \).

- The utility function \( f(s) \) is the system energy consumption (UAV energy consumption):

\[
\sum_{j=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{S_j} P^H a_j^k [t] T^k + \sum_{j=1}^{M} \sum_{t=1}^{S_j} P^F d_j [t] v^{-1}
\]
Trajectory Design

(a) Greedy Algorithm
(b) Genetic Algorithm
(c) DTA
(d) DTA Efficiency
More Research


- 3D+caching

The Internet-Above-the-Clouds

Fig. 1. AANET topology and the corresponding logical topology. (a) PHY topology. (b) Logical topology.

Using Aircrafts as Base Stations?

(a) Heathrow airport

(b) European airspace

(c) North Atlantic

Aircraft mobility patterns

Future Air-Ground-Sea Integrated Networks

Air-Ground-Sea Integrated Networks

Mobile Internet

Internet of Things (IoT)

Smoke Signal

Signal Beacon

Marconi Wireless Communication
We have a long way to go ...

About 80% of the land and 95% of the sea in the world still has no wireless connections!
Summary

- Close collaboration between communication and computing is necessary
- UAV/aircraft-enabled mobile edge computing and communications are promising
Thanks for your attention!