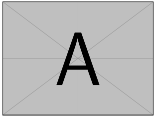
UAV-Assisted MEC Architecture for Collaborative Task Offloading in Urban IoT Environment

Maximizing User Satisfaction and Service Provider Profit

Subhrajit Barick and Chetna Singhal

IEEE Transactions on Network and Service Management
Vol. 22, No. 1, February 2025

October 30, 2025



Outline

Research Objectives

Primary Goals

- **Maximize User Satisfaction:** Ensure IoT devices meet strict latency requirements for computation-intensive tasks

Research Objectives

Primary Goals

- **Maximize User Satisfaction:** Ensure IoT devices meet strict latency requirements for computation-intensive tasks
- **Maximize Service Provider (SP) Profit:** Balance revenue from task offloading with operational costs

Research Objectives

Primary Goals

- **Maximize User Satisfaction:** Ensure IoT devices meet strict latency requirements for computation-intensive tasks
- **Maximize Service Provider (SP) Profit:** Balance revenue from task offloading with operational costs
- **Optimize Resource Utilization:** Joint optimization of IoT-UAV-ES association and UAV network topology

Research Objectives

Primary Goals

- **Maximize User Satisfaction:** Ensure IoT devices meet strict latency requirements for computation-intensive tasks
- **Maximize Service Provider (SP) Profit:** Balance revenue from task offloading with operational costs
- **Optimize Resource Utilization:** Joint optimization of IoT-UAV-ES association and UAV network topology
- **Minimize Infrastructure Cost:** Use minimum number of UAVs while maximizing service coverage

Research Objectives

Primary Goals

- **Maximize User Satisfaction:** Ensure IoT devices meet strict latency requirements for computation-intensive tasks
- **Maximize Service Provider (SP) Profit:** Balance revenue from task offloading with operational costs
- **Optimize Resource Utilization:** Joint optimization of IoT-UAV-ES association and UAV network topology
- **Minimize Infrastructure Cost:** Use minimum number of UAVs while maximizing service coverage

Research Objectives

Primary Goals

- **Maximize User Satisfaction:** Ensure IoT devices meet strict latency requirements for computation-intensive tasks
- **Maximize Service Provider (SP) Profit:** Balance revenue from task offloading with operational costs
- **Optimize Resource Utilization:** Joint optimization of IoT-UAV-ES association and UAV network topology
- **Minimize Infrastructure Cost:** Use minimum number of UAVs while maximizing service coverage

Key Innovation

First work to address three-way collaboration between IoTs, UAVs, and Edge Servers while considering both user and SP perspectives

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life

UAV Advantages

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure

UAV Advantages

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication

UAV Advantages

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

- ✓ Mobility and on-demand deployment

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

- ✓ Mobility and on-demand deployment
- ✓ Line-of-sight (LoS) communication

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

- ✓ Mobility and on-demand deployment
- ✓ Line-of-sight (LoS) communication
- ✓ Dynamic adaptation to user demands

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

- ✓ Mobility and on-demand deployment
- ✓ Line-of-sight (LoS) communication
- ✓ Dynamic adaptation to user demands
- ✓ Enhanced coverage in obstructed areas

Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

- ✓ Mobility and on-demand deployment
- ✓ Line-of-sight (LoS) communication
- ✓ Dynamic adaptation to user demands
- ✓ Enhanced coverage in obstructed areas

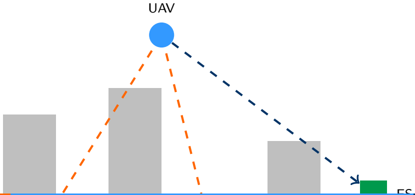
Motivation: Why UAV-Assisted MEC?

Challenges in Urban IoT

- IoT devices: limited processing resources and battery life
- Urban environments: high device density, complex infrastructure
- Physical barriers: Tall buildings hinder direct communication
- Fixed infrastructure: Costly and inflexible deployment

UAV Advantages

- ✓ Mobility and on-demand deployment
- ✓ Line-of-sight (LoS) communication
- ✓ Dynamic adaptation to user demands
- ✓ Enhanced coverage in obstructed areas



Literature Survey: State-of-the-Art

Research Area	Key Contributions	Limitations

Literature Survey: State-of-the-Art

Research Area	Key Contributions	Limitations
MEC Task Offloading	• Deep RL approaches • Service caching • Delay minimization	Assumes consistent connectivity; not suitable for urban obstacles

Literature Survey: State-of-the-Art

Research Area	Key Contributions	Limitations
MEC Task Offloading	• Deep RL approaches • Service caching • Delay minimization	Assumes consistent connectivity; not suitable for urban obstacles
Single UAV-MEC	• UAV as computing platform • Trajectory optimization • Resource allocation	Limited coverage area due to battery and payload constraints

Literature Survey: State-of-the-Art

Research Area	Key Contributions	Limitations
MEC Task Offloading	• Deep RL approaches • Service caching • Delay minimization	Assumes consistent connectivity; not suitable for urban obstacles
Single UAV-MEC	• UAV as computing platform • Trajectory optimization • Resource allocation	Limited coverage area due to battery and payload constraints
Multi-UAV MEC	• Multi-UAV coordination • UAV-BS collaboration • QoS improvements	Focus only on user perspective; neglect SP profit

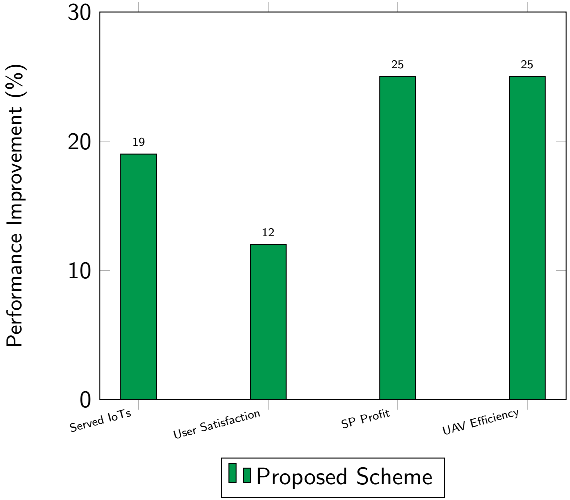
Literature Survey: State-of-the-Art

Research Area	Key Contributions	Limitations
MEC Task Offloading	• Deep RL approaches • Service caching • Delay minimization	Assumes consistent connectivity; not suitable for urban obstacles
Single UAV-MEC	• UAV as computing platform • Trajectory optimization • Resource allocation	Limited coverage area due to battery and payload constraints
Multi-UAV MEC	• Multi-UAV coordination • UAV-BS collaboration • QoS improvements	Focus only on user perspective; neglect SP profit
Matching Theory	• Two-sided matching • Stable allocations	No three-sided matching with cyclic preferences

Literature Survey: State-of-the-Art

Research Area	Key Contributions	Limitations
MEC Task Offloading	• Deep RL approaches • Service caching • Delay minimization	Assumes consistent connectivity; not suitable for urban obstacles
Single UAV-MEC	• UAV as computing platform • Trajectory optimization • Resource allocation	Limited coverage area due to battery and payload constraints
Multi-UAV MEC	• Multi-UAV coordination • UAV-BS collaboration • QoS improvements	Focus only on user perspective; neglect SP profit
Matching Theory	• Two-sided matching • Stable allocations	No three-sided matching with cyclic preferences

Proposed Approach vs. State-of-the-Art

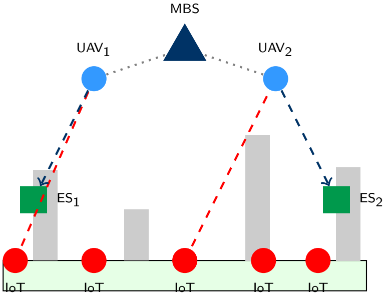


Key Achievements

System Model: Architecture Overview

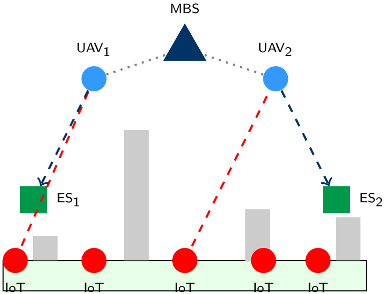
System Components:

- *M* IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$



System Components:

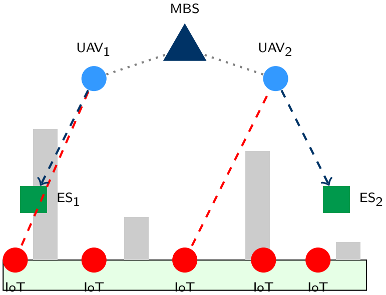
- M IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)



System Model: Architecture Overview

System Components:

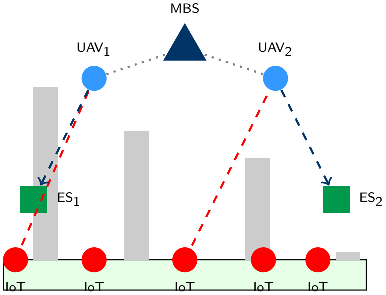
- *M* IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit



System Model: Architecture Overview

System Components:

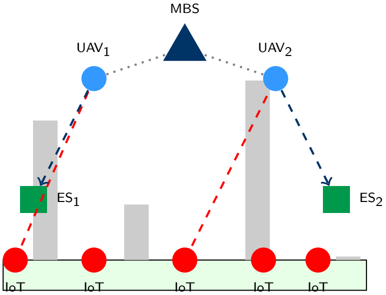
- *M* IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit
  - τ_m: Max tolerable delay



System Model: Architecture Overview

System Components:

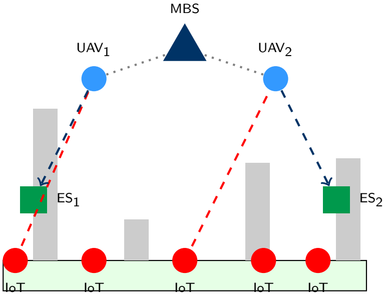
- M IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit
  - τ_m: Max tolerable delay
- K UAVs as relay nodes at altitude H



System Model: Architecture Overview

System Components:

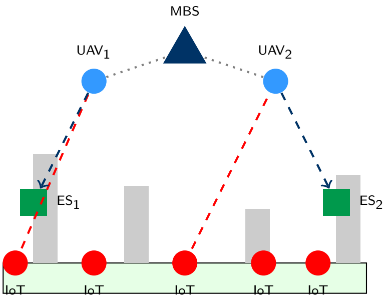
- M IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit
  - τ_m: Max tolerable delay
- K UAVs as relay nodes at altitude H
- L Edge Servers (ES) with heterogeneous computing capacity



System Model: Architecture Overview

System Components:

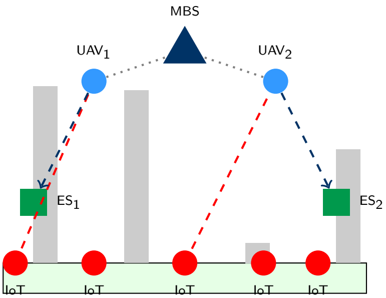
- M IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit
  - τ_m: Max tolerable delay
- K UAVs as relay nodes at altitude H
- L Edge Servers (ES) with heterogeneous computing capacity
- 1 Macro Base Station (MBS)



System Model: Architecture Overview

System Components:

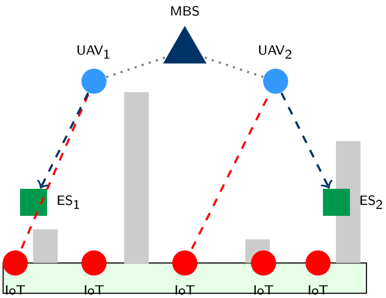
- *M* IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit
  - τ_m: Max tolerable delay
- *K* UAVs as relay nodes at altitude H
- *L* Edge Servers (ES) with heterogeneous computing capacity
- 1 Macro Base Station (MBS)



System Model: Architecture Overview

System Components:

- M IoT devices with tasks $T_m = (D_m, C_m, \tau_m)$
  - D_m: Data size (bits)
  - C_m: CPU cycles/bit
  - τ_m: Max tolerable delay
- K UAVs as relay nodes at altitude H
- L Edge Servers (ES) with heterogeneous computing capacity
- 1 Macro Base Station (MBS)



Communication Model:

- Probabilistic LoS path loss
- PRB allocation: N_U per UAV
- SINR threshold: $\Gamma \geq \Gamma_{th}$

Problem Formulation: Objective Function

Service Provider Profit Model

$$\text{Maximize } P_{SP} = \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{l=1}^{L} \delta_{m,k,l} \cdot P_{m,k,l} - K \cdot C_{UAV}$$

where:

- $\delta_{m,k,l} \in \{0, 1\}$: Offloading decision (IoT $m \to$ UAV $k \to$ ES l)
- $P_{m,k,l} = R_{m,k,l} - C_{m,k,l}$: Profit per successful offload
- $R_{m,k,l} = v \cdot \frac{D_m}{\tau_m}$: Revenue (proportional to data rate)
- $C_{m,k,l} = w \cdot \frac{D_m}{t_l^{proc}}$: ES leasing cost
- $K \cdot C_{UAV}$: UAV operational cost

Problem Formulation: Objective Function

Service Provider Profit Model

$$\text{Maximize } P_{SP} = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L} \delta_{m,k,l} \cdot P_{m,k,l} - K \cdot C_{UAV}$$

where:

- $\delta_{m,k,l} \in \{0,1\}$: Offloading decision (IoT $m \rightarrow$ UAV $k \rightarrow$ ES l)
- $P_{m,k,l} = R_{m,k,l} - C_{m,k,l}$: Profit per successful offload
- $R_{m,k,l} = v \cdot \frac{D_m}{\tau_m}$: Revenue (proportional to data rate)
- $C_{m,k,l} = w \cdot \frac{D_m}{t_l^{proc}}$: ES leasing cost
- $K \cdot C_{UAV}$: UAV operational cost

Total Delay Constraint

$$t_{m,k,l} = \underbrace{t_{m,k,l}^{trans}}_{\text{Transmission delay}} + \underbrace{t_l^{proc}}_{\text{Processing delay}} \leq \tau_m$$

Problem Formulation: Optimization Problem

Joint Optimization Problem (P1)

$$\max_{\Delta, K, X_U, Y_U} \quad P_{SP}$$

$$\text{subject to:} \quad \sum_{k=1}^{K}\sum_{l=1}^{L}\delta_{m,k,l} \leq 1, \quad \forall m \in \{1,\ldots,M\} \quad \text{(Each IoT paired once)}$$

$$\sum_{m=1}^{M}\sum_{l=1}^{L}\delta_{m,k,l} \leq N_U, \quad \forall k \in \{1,\ldots,K\} \quad \text{(UAV capacity)}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\delta_{m,k,l} \leq N_S, \quad \forall l \in \{1,\ldots,L\} \quad \text{(ES capacity)}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L}\delta_{m,k,l} \geq \zeta M \quad \text{(Min service demand)}$$

$$\Gamma_{m,k}, \Gamma_{k,l} \geq \Gamma_{th} \quad \text{(SINR threshold)}$$

Problem Formulation: Optimization Problem

Joint Optimization Problem (P1)

$$\max_{\Delta, K, X_U, Y_U} \quad P_{SP}$$

$$\text{subject to:} \quad \sum_{k=1}^{K}\sum_{l=1}^{L} \delta_{m,k,l} \le 1, \quad \forall m \in \{1, \ldots, M\} \quad \text{(Each IoT paired once)}$$

$$\sum_{m=1}^{M}\sum_{l=1}^{L} \delta_{m,k,l} \le N_U, \quad \forall k \in \{1, \ldots, K\} \quad \text{(UAV capacity)}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K} \delta_{m,k,l} \le N_S, \quad \forall l \in \{1, \ldots, L\} \quad \text{(ES capacity)}$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L} \delta_{m,k,l} \ge \zeta M \quad \text{(Min service demand)}$$

$$\Gamma_{m,k}, \Gamma_{k,l} \ge \Gamma_{th} \quad \text{(SINR threshold)}$$

Problem Decomposition Strategy

Joint Optimization Problem (P1)
NP-Hard: Offloading + Topology

Subproblem 1 (P2)
Offload Strategy Optimization

Subproblem 2 (P4)
UAV Topology Optimization

Solution: R-TMSC Algorithm

(Three-sided Matching)

Solution: K-means Iterative Algorithm

(Clustering-based positioning)

Iterative Joint Optimization

Proposed Solution: R-TMSC Algorithm

Three-Sided Matching with Cyclic Preferences

Key Insight: Agents have cyclic preference structure

1. IoT devices prefer UAVs with **better SINR**

Proposed Solution: R-TMSC Algorithm

Three-Sided Matching with Cyclic Preferences

Key Insight: Agents have cyclic preference structure

1. IoT devices prefer UAVs with **better SINR**
2. UAVs prefer Edge Servers with **lower computing capacity** (cost-effective for SP)

Proposed Solution: R-TMSC Algorithm

Three-Sided Matching with Cyclic Preferences

Key Insight: Agents have cyclic preference structure

1. IoT devices prefer UAVs with **better SINR**
2. UAVs prefer Edge Servers with **lower computing capacity** (cost-effective for SP)
3. Edge Servers prefer IoT devices with **less complex tasks** ($D_m C_m$ smaller)

Proposed Solution: R-TMSC Algorithm

Three-Sided Matching with Cyclic Preferences

Key Insight: Agents have cyclic preference structure

1. IoT devices prefer UAVs with **better SINR**
2. UAVs prefer Edge Servers with **lower computing capacity** (cost-effective for SP)
3. Edge Servers prefer IoT devices with **less complex tasks** ($D_m C_m$ smaller)

Proposed Solution: R-TMSC Algorithm

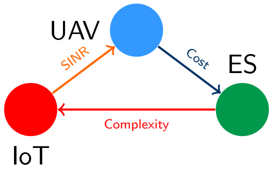
Three-Sided Matching with Cyclic Preferences

Key Insight: Agents have cyclic preference structure

1. IoT devices prefer UAVs with **better SINR**
2. UAVs prefer Edge Servers with **lower computing capacity** (cost-effective for SP)
3. Edge Servers prefer IoT devices with **less complex tasks** ($D_m C_m$ smaller)

Preference Lists:

- PL_{I_m}: UAVs sorted by $\Gamma_{m,k}$ (descending)
- PL_{U_k}: ESs sorted by q_l (ascending)
- PL_{S_l}: IoTs sorted by $D_m C_m$ (ascending)



R-TMSC Matching Algorithm

Algorithm 1 R-TMSC Offloading

Initialize $M_t = \emptyset$
while not converged **do**
  for each IoT I_m **do**
    Find $U^* = Z_{I_m} \cap Z^*_{I_m}$
    if $U^* \neq \emptyset$ **then**
      $U_k = \text{Head}(U^*)$
      $S^* = Z_{U_k} \cap \hat{Z}_{I_m}$
      $S_l = \text{Head}(S^*)$
      Update M_t with (I_m, U_k, S_l)
    end if
  end for
end while
return M_t (stable matching set)

Key Concepts:

- M_t: Set of matching triplets (I_m, U_k, S_l)

R-TMSC Matching Algorithm

Algorithm 2 R-TMSC Offloading

Initialize $M_t = \emptyset$
while not converged **do**
  for each IoT I_m **do**
    Find $U^* = Z_{I_m} \cap Z^*_{I_m}$
    if $U^* \neq \emptyset$ **then**
      $U_k = \text{Head}(U^*)$
      $S^* = Z_{U_k} \cap \hat{Z}_{I_m}$
      $S_l = \text{Head}(S^*)$
      Update M_t with (I_m, U_k, S_l)
    end if
  end for
end while
return M_t (stable matching set)

Key Concepts:

- M_t: Set of matching triplets (I_m, U_k, S_l)
- Z_{I_m}: UAVs preferred over current match

R-TMSC Matching Algorithm

Algorithm 3 R-TMSC Offloading

```
Initialize M_t = ∅
while not converged do
    for each IoT I_m do
        Find U* = Z_{I_m} ∩ Z*_{I_m}
        if U* ≠ ∅ then
            U_k = Head(U*)
            S* = Z_{U_k} ∩ Ẑ_{I_m}
            S_l = Head(S*)
            Update M_t with (I_m, U_k, S_l)
        end if
    end for
end while
return M_t (stable matching set)
```

Key Concepts:

- M_t: Set of matching triplets (I_m, U_k, S_l)
- Z_{I_m}: UAVs preferred over current match
- $Z^*_{I_m}$: UAVs with available capacity and suitable ES

R-TMSC Matching Algorithm

Algorithm 4 R-TMSC Offloading

Initialize $M_t = \emptyset$
while not converged **do**
  for each IoT I_m **do**
    Find $U^* = Z_{I_m} \cap Z^*_{I_m}$
    if $U^* \neq \emptyset$ **then**
      $U_k = \text{Head}(U^*)$
      $S^* = Z_{U_k} \cap \hat{Z}_{I_m}$
      $S_l = \text{Head}(S^*)$
      Update M_t with (I_m, U_k, S_l)
    end if
  end for
end while
return M_t (stable matching set)

Key Concepts:

- M_t: Set of matching triplets (I_m, U_k, S_l)
- Z_{I_m}: UAVs preferred over current match
- $Z^*_{I_m}$: UAVs with available capacity and suitable ES
- $\hat{Z}_{I_m}$: ESs that can accept task with delay constraint

R-TMSC Matching Algorithm

Algorithm 5 R-TMSC Offloading

Initialize $M_t = \emptyset$
while not converged **do**
  for each IoT I_m **do**
    Find $U^* = Z_{I_m} \cap Z^*_{I_m}$
    if $U^* \neq \emptyset$ **then**
      $U_k = \text{Head}(U^*)$
      $S^* = Z_{U_k} \cap \hat{Z}_{I_m}$
      $S_l = \text{Head}(S^*)$
      Update M_t with (I_m, U_k, S_l)
    end if
  end for
end while
return M_t (stable matching set)

Key Concepts:

- M_t: Set of matching triplets (I_m, U_k, S_l)
- Z_{I_m}: UAVs preferred over current match
- $Z^*_{I_m}$: UAVs with available capacity and suitable ES
- $\hat{Z}_{I_m}$: ESs that can accept task with delay constraint

R-TMSC Matching Algorithm

Algorithm 6 R-TMSC Offloading

Initialize $M_t = \emptyset$
while not converged **do**
  for each IoT I_m **do**
    Find $U^* = Z_{I_m} \cap Z^*_{I_m}$
    if $U^* \neq \emptyset$ **then**
      $U_k = \text{Head}(U^*)$
      $S^* = Z_{U_k} \cap \hat{Z}_{I_m}$
      $S_l = \text{Head}(S^*)$
      Update M_t with (I_m, U_k, S_l)
    end if
  end for
end while
return M_t (stable matching set)

Key Concepts:

- M_t: Set of matching triplets (I_m, U_k, S_l)
- Z_{I_m}: UAVs preferred over current match
- $Z^*_{I_m}$: UAVs with available capacity and suitable ES
- $\hat{Z}_{I_m}$: ESs that can accept task with delay constraint

Properties:

- Finds stable matching
- Satisfies all constraints
- Maximizes number of served IoTs

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

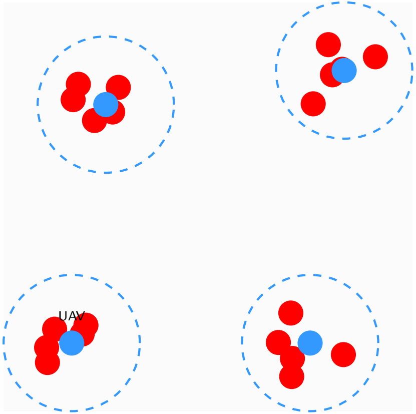
UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$



K-means Clustering Result

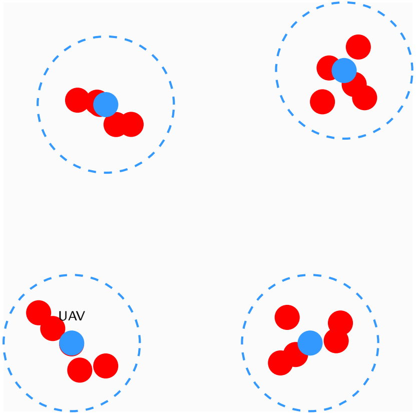
UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$
2. Cluster IoT locations using K-means



K-means Clustering Result

UAV Topology Optimization: K-means Approach

Objectives:

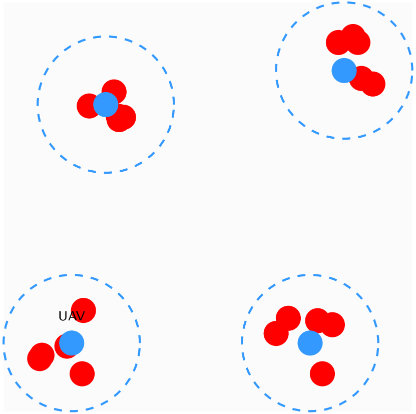
- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$
2. Cluster IoT locations using K-means
3. Position UAV at cluster centroid:

$$x_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} x_{I_m}$$

$$y_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} y_{I_m}$$



K-means Clustering Result

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

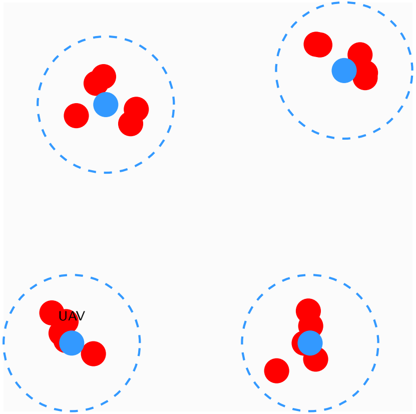
Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$
2. Cluster IoT locations using K-means
3. Position UAV at cluster centroid:

$$x_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} x_{I_m}$$

$$y_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} y_{I_m}$$

4. Update K based on service coverage



K-means Clustering Result

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

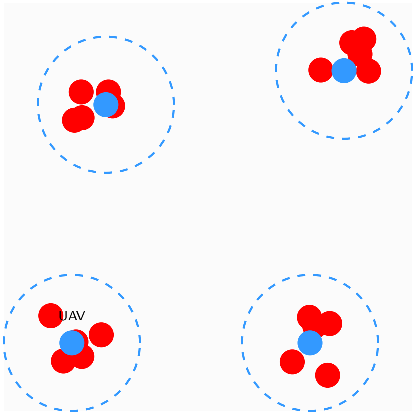
Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$
2. Cluster IoT locations using K-means
3. Position UAV at cluster centroid:

$$x_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} x_{I_m}$$

$$y_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} y_{I_m}$$

4. Update K based on service coverage



K-means Clustering Result

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

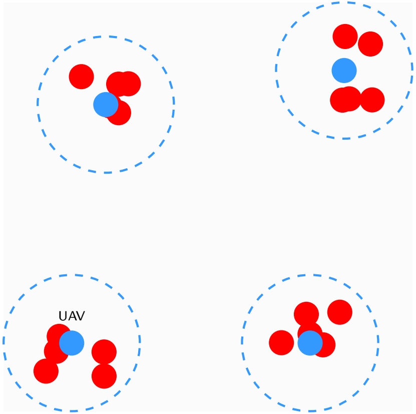
Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$
2. Cluster IoT locations using K-means
3. Position UAV at cluster centroid:

$$x_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} x_{I_m}$$

$$y_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} y_{I_m}$$

4. Update K based on service coverage
5. Iterate until convergence



K-means Clustering Result

UAV Topology Optimization: K-means Approach

Objectives:

- Determine minimum number of UAVs K
- Find optimal positions (x_{U_k}, y_{U_k})
- Ensure $\geq \zeta\%$ IoTs are served

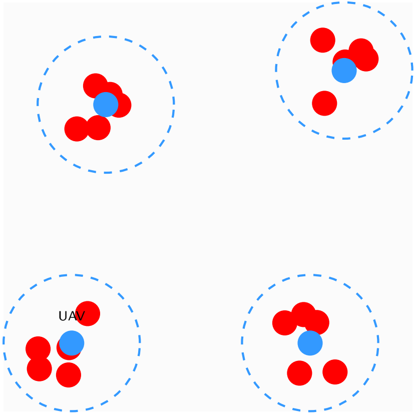
Iterative Process:

1. Start with $K_{initial} = \lceil M/N_U \rceil$
2. Cluster IoT locations using K-means
3. Position UAV at cluster centroid:

$$x_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} x_{I_m}$$

$$y_{U_k} = \frac{1}{N_{CL_k}} \sum_{I_m \in CL_k} y_{I_m}$$

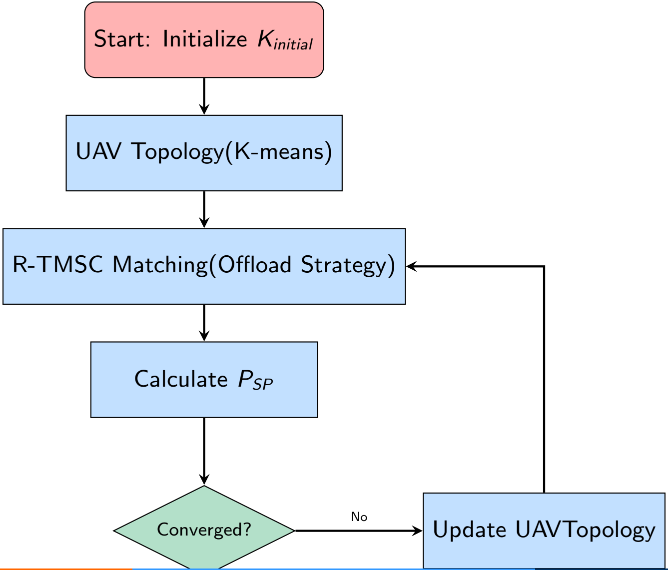
4. Update K based on service coverage
5. Iterate until convergence



K-means Clustering Result

Advantages:

Joint Optimization: Putting It All Together



Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring

Implementation Plan: System Architecture

Hardware Components:

- IoT Devices
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- UAV Fleet
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- Edge Servers
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- MBS Coordinator
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm

Implementation Plan: System Architecture

Hardware Components:

- IoT Devices
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- UAV Fleet
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- Edge Servers
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- MBS Coordinator
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm
   - K-means clustering

Implementation Plan: System Architecture

Hardware Components:

- IoT Devices
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- UAV Fleet
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- Edge Servers
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- MBS Coordinator
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm
   - K-means clustering
   - Preference list generation

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm
   - K-means clustering
   - Preference list generation
4. **Monitoring & Analytics**

Implementation Plan: System Architecture

Hardware Components:

- IoT Devices
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- UAV Fleet
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- Edge Servers
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- MBS Coordinator
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm
   - K-means clustering
   - Preference list generation
4. **Monitoring & Analytics**
   - Performance metrics

Implementation Plan: System Architecture

Hardware Components:

- **IoT Devices**
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- **UAV Fleet**
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- **Edge Servers**
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- **MBS Coordinator**
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm
   - K-means clustering
   - Preference list generation
4. **Monitoring & Analytics**
   - Performance metrics
   - User satisfaction tracking

Implementation Plan: System Architecture

Hardware Components:

- IoT Devices
  - Sensors, cameras, etc.
  - Generate computation tasks
  - Limited processing power
- UAV Fleet
  - Communication relay nodes
  - Fixed altitude deployment
  - Equipped with transceivers
- Edge Servers
  - Small cell 5G NR gNodeB
  - Heterogeneous computing capacity
  - Privately owned resources
- MBS Coordinator
  - Central control and coordination
  - Resource allocation
  - Algorithm execution

Software Modules:

1. **Communication Module**
   - LoS path loss calculation
   - SINR measurement
   - PRB allocation
2. **Task Management**
   - Task queue at UAVs
   - Buffer management
   - Delay monitoring
3. **Optimization Engine**
   - R-TMSC algorithm
   - K-means clustering
   - Preference list generation
4. **Monitoring & Analytics**
   - Performance metrics
   - User satisfaction tracking
   - Profit calculation

Implementation Plan: Deployment Steps

Phase 1: System Initialization

1. Deploy ESs at strategic urban locations with 5G infrastructure
2. Configure ES computing capacities $\{q_1, q_2, \ldots, q_L\}$
3. Set system parameters: N_U, N_S, Γ_{th}, ζ, altitude H
4. Initialize MBS coordination framework

Implementation Plan: Deployment Steps

Phase 1: System Initialization

1. Deploy ESs at strategic urban locations with 5G infrastructure
2. Configure ES computing capacities $\{q_1, q_2, \ldots, q_L\}$
3. Set system parameters: N_U, N_S, Γ_{th}, ζ, altitude H
4. Initialize MBS coordination framework

Phase 2: UAV Deployment

1. Collect IoT device position data $Q_I = \{Q_{I_m}, m = 1, \ldots, M\}$
2. Calculate initial UAV count: $K_{initial} = \lceil M/N_U \rceil$
3. Apply K-means clustering to determine UAV positions (x_{U_k}, y_{U_k}, H)
4. Deploy UAVs at calculated positions

Implementation Plan: Deployment Steps

Phase 1: System Initialization

1. Deploy ESs at strategic urban locations with 5G infrastructure
2. Configure ES computing capacities $\{q_1, q_2, \ldots, q_L\}$
3. Set system parameters: N_U, N_S, Γ_{th}, ζ, altitude H
4. Initialize MBS coordination framework

Phase 2: UAV Deployment

1. Collect IoT device position data $Q_I = \{Q_{I_m}, m = 1, \ldots, M\}$
2. Calculate initial UAV count: $K_{initial} = \lceil M/N_U \rceil$
3. Apply K-means clustering to determine UAV positions (x_{U_k}, y_{U_k}, H)
4. Deploy UAVs at calculated positions

Phase 3: Preference List Generation

1. Measure SINR for all IoT-UAV pairs: $\Gamma_{m,k}$
2. Measure SINR for all UAV-ES pairs: $\Gamma_{k,l}$

Implementation Plan: Operational Workflow

Phase 4: Task Offloading Execution

1. IoT devices submit task requests $T_m = (D_m, C_m, \tau_m)$ to MBS
2. MBS executes R-TMSC algorithm to find stable matching set M_t
3. Generate offloading strategy $\Delta = \{\delta_{m,k,l}\}$ from M_t
4. Communicate assignments to IoTs, UAVs, and ESs
5. Execute task offloading: IoT $\rightarrow$ UAV $\rightarrow$ ES
6. Monitor delay constraints and SINR thresholds

Implementation Plan: Operational Workflow

Phase 4: Task Offloading Execution

1. IoT devices submit task requests $T_m = (D_m, C_m, \tau_m)$ to MBS
2. MBS executes R-TMSC algorithm to find stable matching set M_t
3. Generate offloading strategy $\Delta = \{\delta_{m,k,l}\}$ from M_t
4. Communicate assignments to IoTs, UAVs, and ESs
5. Execute task offloading: IoT $\rightarrow$ UAV $\rightarrow$ ES
6. Monitor delay constraints and SINR thresholds

Phase 5: Dynamic Optimization

1. Calculate current SP profit: P_{SP}
2. Evaluate service coverage: percentage of served IoTs
3. If service coverage $< \zeta$: Increase UAV count and reposition
4. If service coverage $> \zeta$: Try to reduce UAV count (cost optimization)
5. Update UAV topology using K-means iterative algorithm

Implementation Plan: Performance Monitoring

Real-time Metrics:

- Number of active IoT devices
- Served IoT percentage
- Average task completion delay
- SINR distribution
- UAV buffer utilization
- ES computational load

Implementation Plan: Performance Monitoring

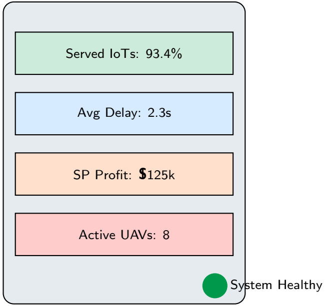
Real-time Metrics:

- Number of active IoT devices
- Served IoT percentage
- Average task completion delay
- SINR distribution
- UAV buffer utilization
- ES computational load

Business Metrics:

- Revenue from task offloading
- ES leasing costs
- UAV operational costs
- Overall SP profit
- User satisfaction scores

Monitoring Dashboard



Implementation Plan: Performance Monitoring

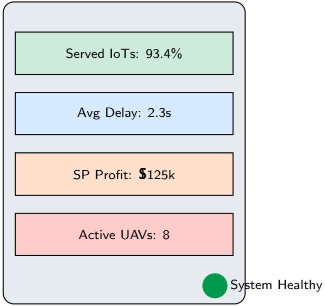
Real-time Metrics:

- Number of active IoT devices
- Served IoT percentage
- Average task completion delay
- SINR distribution
- UAV buffer utilization
- ES computational load

Business Metrics:

- Revenue from task offloading
- ES leasing costs
- UAV operational costs
- Overall SP profit
- User satisfaction scores

Monitoring Dashboard



Adaptive Actions:

- Trigger re-optimization on significant changes
- Alert on constraint violations
- Automatic UAV repositioning

Simulation Setup and Parameters

Environment Setup:

- Area: 1 km $\times$ 1 km urban region
- IoT distribution: Uniform random
- ES positions: Fixed strategic locations
- UAV altitude: 100-150m
- Communication: 5G NR at 2.4 GHz

System Parameters:

- IoT devices: $M \in [50, 300]$
- Edge Servers: $L \in [4, 12]$
- UAV capacity: $N_U = 20$ IoTs
- ES capacity: $N_S = 40$ tasks
- Service threshold: $\zeta = 90\%$

Task Characteristics:

- Data size: $D_m \in [0.5, 5]$ MB
- CPU cycles: $C_m \in [500, 1500]$ cycles/bit
- Max delay: $\tau_m \in [1, 10]$ seconds

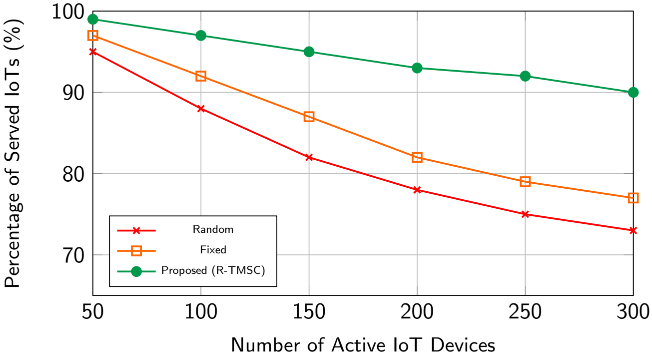
Pricing Model:

- Revenue param: $v = 0.1$ \$/Mbps
- Cost param: $w = 0.01$ \$/Mbps
- UAV cost: $C_{UAV} = 100$ \$/UAV

Benchmark Schemes:

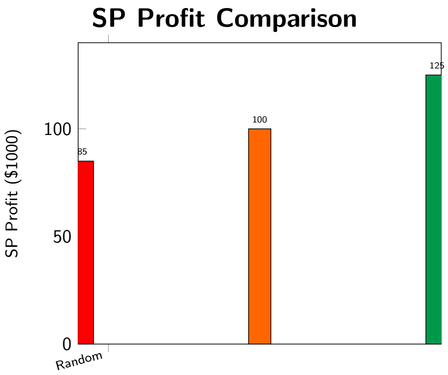
1. Fixed Offloading: Predefined associations
2. Random Offloading: Random pairing
3. Proposed R-TMSC: Our scheme

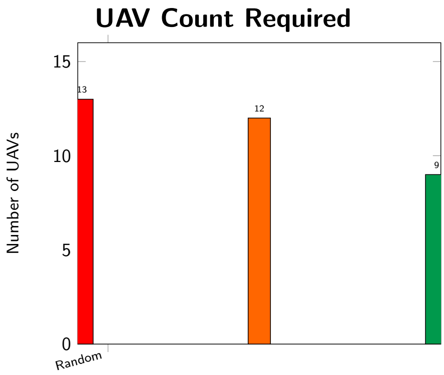
Results: Performance Comparison



- **Proposed scheme achieves 93.4% average served IoT percentage**
- **19% improvement** over Fixed and **28% improvement** over Random
- Maintains high performance even with increasing IoT density

Results: SP Profit and UAV Efficiency





Key Findings

- **25% higher SP profit** with proposed scheme
- **25% fewer UAVs required compared to benchmarks**

Conclusion and Future Work

Key Contributions

- Novel **UAV-assisted MEC architecture** for urban IoT environments
- First work addressing **three-sided collaboration** (IoT-UAV-ES) with **SP profit maximization**
- **R-TMSC algorithm** for stable matching based on cyclic preferences
- **K-means iterative algorithm** for optimal UAV topology
- Comprehensive evaluation showing **significant performance improvements**

Conclusion and Future Work

Key Contributions

- Novel **UAV-assisted MEC architecture** for urban IoT environments
- First work addressing **three-sided collaboration** (IoT-UAV-ES) with **SP profit maximization**
- **R-TMSC algorithm** for stable matching based on cyclic preferences
- **K-means iterative algorithm** for optimal UAV topology
- Comprehensive evaluation showing **significant performance improvements**

Performance Highlights

- ✓ 19% more IoTs served
- ✓ 25% higher SP profit
- ✓ 12% better user satisfaction
- ✓ 25% fewer UAVs required

Conclusion and Future Work

Key Contributions

- Novel **UAV-assisted MEC architecture** for urban IoT environments
- First work addressing **three-sided collaboration** (IoT-UAV-ES) with **SP profit maximization**
- **R-TMSC algorithm** for stable matching based on cyclic preferences
- **K-means iterative algorithm** for optimal UAV topology
- Comprehensive evaluation showing **significant performance improvements**

Performance Highlights

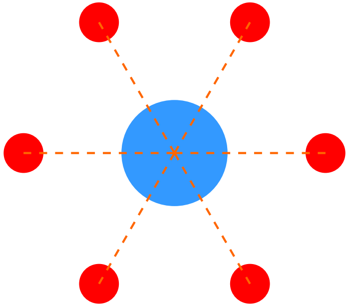
- ✓ 19% more IoTs served
- ✓ 25% higher SP profit
- ✓ 12% better user satisfaction
- ✓ 25% fewer UAVs required

Future Research Directions

- Extension to **dynamic IoT mobility scenarios**

Thank You!

Questions & Discussion



Backup Slides

Additional Technical Details

Mathematical Formulation: Detailed Communication Model

Path Loss Model

$$PL_{i,j} = \beta_0 d_{i,j}^2 \left(P_{i,j}^{LoS} \mu^{LoS} + P_{i,j}^{NLoS} \mu^{NLoS} \right)$$

where:

- $\beta_0 = (4\pi f/c)^2$: Free space path loss at reference distance
- $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + H^2}$: Euclidean distance
- $P_{i,j}^{LoS} = \frac{1}{1+a\exp(-b(\theta_{i,j}-a))}$: LoS probability
- $\theta_{i,j} = \sin^{-1}(H/d_{i,j})$: Elevation angle

SINR and Data Rate

$$\Gamma_{i,j} = \frac{P_i/PL_{i,j}}{\sum_{i' \neq i} P_{i'}/PL_{i',j} + N_0}, \quad r_{i,j} = B_{i,j} \log(1 + \Gamma_{i,j})$$

Algorithm Complexity Analysis

Algorithm Component	Complexity	Explanation
R-TMSC Matching	$O(n_1 \cdot MKL)$	n_1 iterations, matching triplets
K-means Topology	$O(n_2 \cdot K)$	n_2 iterations, update positions
Joint Optimization	$O(n \cdot [n_1 MKL + n_2 K])$	n outer iterations

Practical Performance

- Typical values: $n = 4$, $n_1 = 3$, $n_2 = 2$
- For $M = 200$, $K = 10$, $L = 8$: Total complexity $\approx O(19.2 \times 10^4)$
- Execution time: 60ms per outer iteration, 240ms total
- **Suitable for quasi-static scenarios** (minutes to hours)

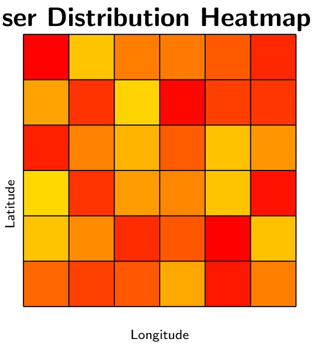
Real-World Dataset Validation

China Mobile User Dataset:

- Real user geolocation data
- Region: 22.5-23.3°N, 112.5-114.8°E
- Time window: 10 AM - 12 PM
- 15-minute time slots

Results:

- 10% higher served percentage vs. Fixed
- 28% higher served percentage vs. Random
- $1.2\times$ SP profit compared to Fixed
- $1.5\times$ SP profit compared to Random



Validates robustness in non-uniform distributions