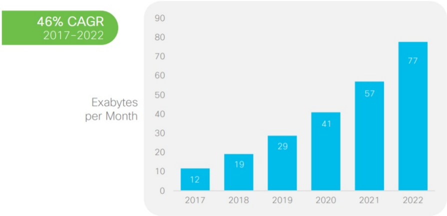
Task Placement and Resource Allocation in Edge Computing Systems

Slađana Jošilo

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden

Stockholm, May 27, 2020

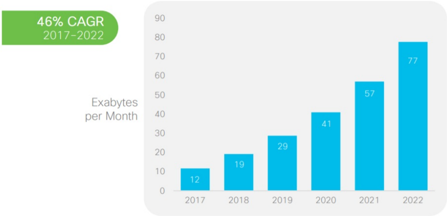
Global Mobile Data Traffic Explosion



Source: Cisco VNI Global Mobile Data Traffic Forecast, 2017-2022

2017 - 2022: Sevenfold increase

Global Mobile Data Traffic Explosion



Source: Cisco VNI Global Mobile Data Traffic Forecast, 2017-2022

2017 - 2022: Sevenfold increase

Key Drivers for Data Explosion

- ↑ number of mobile connections: 8.6 bil. in 2017 - 12.3 bil. in 2022
- ↑ mobile network speeds: 8.7 Mbps in 2017 - 28.5 Mbps in 2022
- ↑ demand for a variety of applications

Application Requirements vs. Device Capabilities

Applications

- Computationally intensive tasks: machine learning applications
- Delay sensitive tasks: real-time control applications

Application Requirements vs. Device Capabilities

Applications

- Computationally intensive tasks: machine learning applications
- Delay sensitive tasks: real-time control applications

Devices

- Battery powered $\rightarrow$ low energy consumption requirements
- Computationally constrained
  - Energy consumption requirements vs. clock speed of the processor
  - Requirements for light and small devices

Application Requirements vs. Device Capabilities

Applications

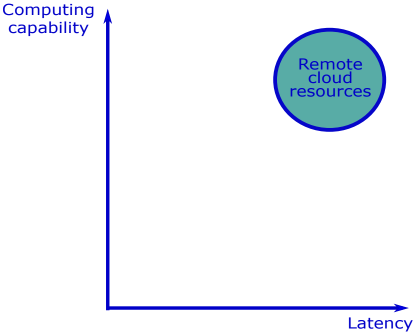
- Computationally intensive tasks: machine learning applications
- Delay sensitive tasks: real-time control applications

Devices

- Battery powered → low energy consumption requirements
- Computationally constrained
  - Energy consumption requirements vs. clock speed of the processor
  - Requirements for light and small devices

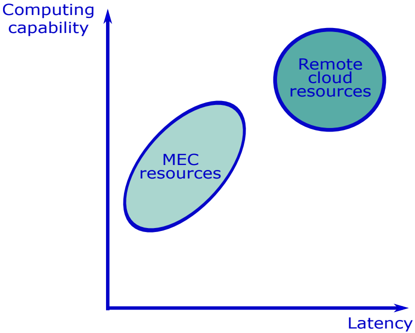
How to close the gap between the application requirements and device capabilities?

Computation Offloading



Remote Cloud Computing

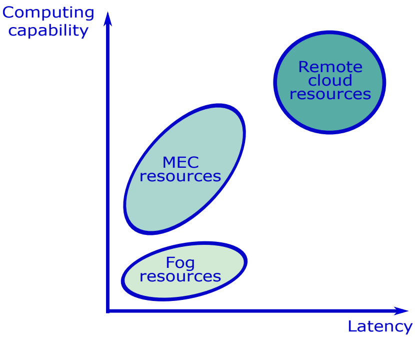
Computation Offloading



Remote Cloud Computing

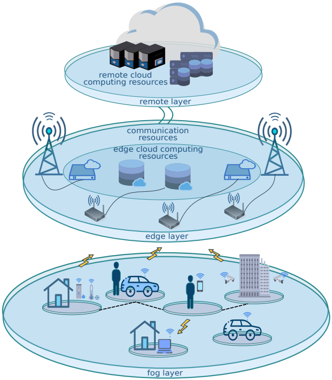
Mobile Edge Computing (MEC)

Computation Offloading



Remote Cloud Computing
Mobile Edge Computing (MEC)
Fog Computing

Edge Computing Systems

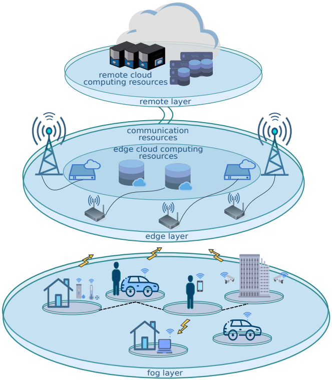


Remote Cloud Computing

Mobile Edge Computing (MEC)

Fog Computing

Edge Computing Systems



Remote Cloud Computing

Mobile Edge Computing (MEC)

Fog Computing

Edge System Resources

- Computing resources (remote clouds, edge clouds, fog devices)
- Communication resources (wireless and wireline)

Major Challenge

- Task placement and management of communication and computing resources
  - Response time requirements
  - Energy consumption requirements

Major Challenge

- Task placement and management of communication and computing resources
  - Response time requirements
  - Energy consumption requirements

- Algorithms for placing tasks and allocating resources
  - Scalability
  - Limited information availability
  - Cater for autonomous devices $\Rightarrow$ decentralized decisions
  - Guaranteed system performance

Outline

1. Task Placement in Edge Computing Systems
    - Completion Time Minimization
        - Collaborative offloading supported by a cloud server (Paper A)
    - Completion Time and Energy Consumption Minimization
        - Scheduling of tasks over time slots, communication and computing resources (Paper B and Paper C)
2. Task Placement and Resource Management in Edge Computing Systems
    - Completion Time Minimization
        - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

Outline

1. Task Placement in Edge Computing Systems
   - Completion Time Minimization
     - (Paper A)
   - Con ation
     - and computing
2. Task Pla Computing Systems
   - Con
     - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

1. Worst case problem complexity?

Outline

1. Task Placement in Edge Computing Systems
   - Completion Time Minimization
     - (Paper A)
   - Con ation
     - and computing
2. Task Pla Computing Systems
   - Con
     - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

1. Worst case problem complexity?
2. Equilibrium for autonomous devices?

Outline

1. Task Placement in Edge Computing Systems
   - Completion Time Minimization
     - (Paper A)
   - Con ation
     - and computing

1. Worst case problem complexity?
2. Equilibrium for autonomous devices?
3. Decentralized algorithms?

2. Task Plac Computing Systems
   - Con
     - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

Outline

1. Task Placement in Edge Computing Systems
   - Completion Time Minimization
     - (Paper A)
   - Con ation
     - and computing

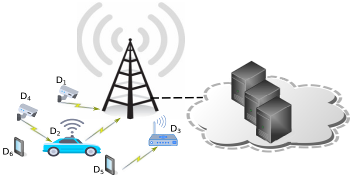
2. Task Pla Computing Systems
   - Con
     - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

1. Worst case problem complexity?
2. Equilibrium for autonomous devices?
3. Decentralized algorithms?
4. What is the complexity of the algorithm?
5. How good are the system performance?
6. How to achieve low signaling overhead?

Outline

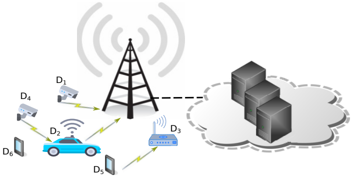
1. Task Placement in Edge Computing Systems
   - Completion Time Minimization
     - **Collaborative offloading supported by a cloud server (Paper A)**
   - Completion Time and Energy Consumption Minimization
     - Scheduling of tasks over time slots, communication and computing resources (Paper B and Paper C)
2. Task Placement and Resource Management in Edge Computing Systems
   - Completion Time Minimization
     - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

Collaborative Edge Computing



- Cloud server
- Set of WDs $\mathcal{N} = \{1, 2, ..., N\}$

Collaborative Edge Computing

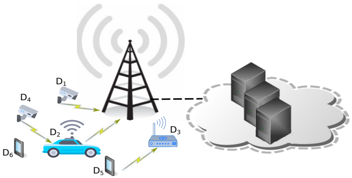


- Cloud server
- Set of WDs $\mathcal{N} = \{1, 2, ..., N\}$

Computational Tasks

- WD i generates a sequence $(t_{i,1}, t_{i,2}, \ldots)$ of tasks
  - Poisson task arrival process with arrival intensity λ_i
  - Mean size of the input data $\overline{D}_i$
  - Mean computational complexity $\overline{L}_i$

Collaborative Edge Computing

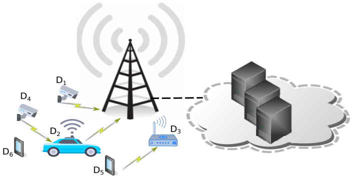


- Cloud server
- Set of WDs $\mathcal{N} = \{1, 2, ..., N\}$

Computational Tasks

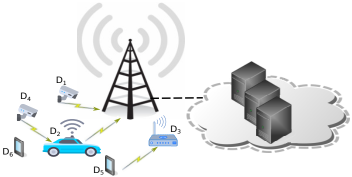
- WD i generates a sequence $(t_{i,1}, t_{i,2}, \ldots)$ of tasks
  - Poisson task arrival process with arrival intensity λ_i
  - Mean size of the input data $\overline{D}_i$
  - Mean computational complexity $\overline{L}_i$
- Decision of WD i for task $t_{i,k}$
  - Local computing with probability $p_{i,i}(k)$
  - Offloading to WD j with probability $p_{i,j}(k)$
  - Offloading to the cloud with probability $p_{i,c}(k)$

Communication Model

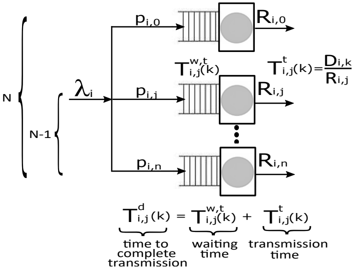


- OFDMA dedicated mode of communication
  - Assignment of subcarriers to pairs of communicating nodes
  - $R_{i,j}$: transmission rate from WD i to node j

Communication Model

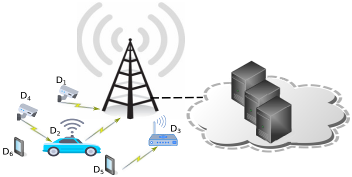


- OFDMA dedicated mode of communication
  - Assignment of subcarriers to pairs of communicating nodes
  - $R_{i,j}$: transmission rate from WD i to node j



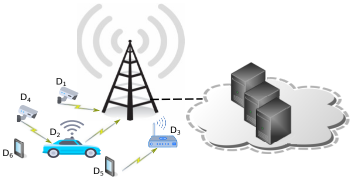
- Each WD has N transmission queues (FIFO order)

Computing Model

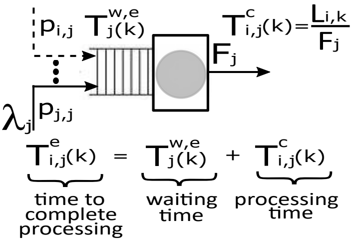


- F_i: computing capability of WD i
- F^c: computing capability of the cloud

Computing Model



- F_i: computing capability of WD i
- F^c: computing capability of the cloud



- Each WD has one execution queue (FIFO order)

Cost Model

Mean Completion Time

$$C_i = \lim_{K\to\infty} \frac{1}{K}\left[\sum_{k=1}^{K}\left(p_{i,i}(k)T^e_{i,i}(k) + \sum_{j\in\mathcal{N}\setminus\{i\}\cup\{c\}} p_{i,j}(k)\big(T^d_{i,j}(k) + T^e_{i,j}(k)\big)\right)\right]$$

Cost Model

Mean Completion Time

$$C_i = \lim_{K\to\infty} \frac{1}{K}\left[\sum_{k=1}^{K}\left(p_{i,i}(k)T^e_{i,i}(k) + \sum_{j\in\mathcal{N}\setminus\{i\}\cup\{c\}} p_{i,j}(k)\big(T^d_{i,j}(k)+T^e_{i,j}(k)\big)\right)\right]$$

Dynamic Non-Cooperative Game

- Closest to stochastic game with countably infinite state space

Cost Model

Mean Completion Time

$$C_i = \lim_{K\to\infty} \frac{1}{K}\left[\sum_{k=1}^{K}\left(p_{i,i}(k)T^e_{i,i}(k) + \sum_{j\in\mathcal{N}\setminus\{i\}\cup\{c\}} p_{i,j}(k)\big(T^d_{i,j}(k)+T^e_{i,j}(k)\big)\right)\right]$$

Dynamic Non-Cooperative Game

- Closest to stochastic game with countably infinite state space
- Existence results for Markov perfect equilibria are not known

System in Steady State

Communication Model

- Each transmission queue modeled as an M/G/1 system
- $\overline{T^d}_{i,j}$: mean time needed to deliver data $\overline{D}_i$ from WD i to node j

System in Steady State

Communication Model

- Each transmission queue modeled as an M/G/1 system
- $\overline{T^d}_{i,j}$: mean time needed to deliver data $\overline{D}_i$ from WD i to node j

Computing Model

- Execution queue of each WD modeled as an M/G/1 system
- Execution queue of the cloud modeled as an M/G/∞ system
- $\overline{T^e}_{i,j}$: mean time needed to execute $\overline{L}_i$ cycles at node j

System in Steady State

Communication Model

- Each transmission queue modeled as an M/G/1 system
- $\overline{T^d}_{i,j}$: mean time needed to deliver data $\overline{D}_i$ from WD i to node j

Computing Model

- Execution queue of each WD modeled as an M/G/1 system
- Execution queue of the cloud modeled as an M/G/∞ system
- $\overline{T^e}_{i,j}$: mean time needed to execute $\overline{L}_i$ cycles at node j

Cost Model

$$C_i(p_i, p_{-i}) = p_{i,i}\overline{T^e}_{i,i} + \sum_{j \in \mathcal{N} \setminus \{i\} \cup \{c\}} p_{i,j}\left(\overline{T^d}_{i,j} + \overline{T^e}_{i,j}\right)$$

Equilibrium Existence in Static Mixed Strategies

- The game has at least one equilibrium in static mixed strategies

- Proof based on using variational inequality theory
- Computing relies on average system parameters:
  - Average task arrival intensities
  - Average transmission rates
  - First and second moments of the task size distribution
  - First and second moments of the task complexity distribution

Decentralized Algorithms for Allocating Tasks

Static Mixed Nash Equilibrium (SM-NE) Algorithm

- Every WD allocates tasks based on the computed static mixed strategy equilibrium
- Relies on the average system performance $\Rightarrow$ low signaling overhead

Decentralized Algorithms for Allocating Tasks

Static Mixed Nash Equilibrium (SM-NE) Algorithm

- Every WD allocates tasks based on the computed static mixed strategy equilibrium
- Relies on the average system performance ⇒ low signaling overhead

Myopic Best Response (MBR)

- Every WD allocates tasks based on a myopic best response strategy
- Relies on the instantaneous states of the system ⇒ high signaling overhead

Performance Gain w.r.t Local Computing

Evaluation scenario

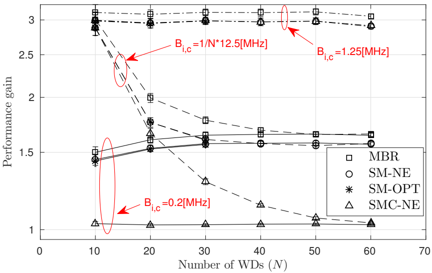
- $\lambda_i \sim \mathcal{U}(0.01, 0.03)$ tasks/s $F_i \sim \mathcal{U}(1,4)$ $Gcycles$, $F^c = 64$ $Gcycles$,
- Tasks: $D_i \sim \mathcal{U}(0.1, 3.4)$ Mb , $L_i \sim \mathcal{U}(0.2, 1)$ $Gcycles$

Performance Gain w.r.t Local Computing

Evaluation scenario

- $\lambda_i \sim \mathcal{U}(0.01, 0.03)$ tasks/s $F_i \sim \mathcal{U}(1,4)$ *Gcycles*, $F^c = 64$ *Gcycles*,
- Tasks: $D_i \sim \mathcal{U}(0.1, 3.4)$ Mb , $L_i \sim \mathcal{U}(0.2, 1)$ *Gcycles*

Performance gain (algorithm A) = $\frac{\text{sum of costs of WDs when computing locally}}{\text{sum of costs of WDs when using algorithhm A}}$



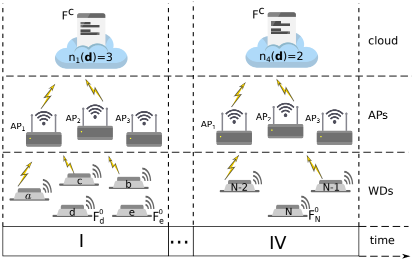
- Higher WD to cloud bandwidth ⇒ higher *performance gain*
- D2D offloading based on average system parameters performs close to D2D offloading based on the global knowledge
- SM-NE algorithm performs close to the SM-OPT algorithm

Outline

1. Task Placement in Edge Computing Systems
    - Completion Time Minimization
        - Collaborative offloading supported by a cloud server (Paper A)
    - Completion Time and Energy Consumption Minimization
        - **Scheduling of tasks over time slots, communication and computing resources (Paper B and Paper C)**

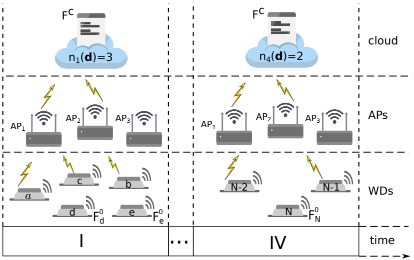
2. Task Placement and Resource Management in Edge Computing Systems
    - Completion Time Minimization
        - Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)

MEC System with Periodic Tasks



- Edge cloud
- Set of APs $\mathcal{A} = \{1, 2, ..., A\}$
- Set of WDs $\mathcal{N} = \{1, 2, ..., N\}$
- Set of time slots $\mathcal{T} = \{1, 2, \ldots, T\}$

MEC System with Periodic Tasks

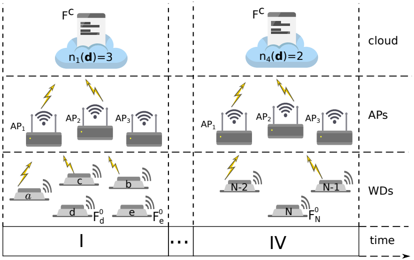


- Edge cloud
- Set of APs $\mathcal{A} = \{1, 2, ..., A\}$
- Set of WDs $\mathcal{N} = \{1, 2, ..., N\}$
- Set of time slots $\mathcal{T} = \{1, 2, \ldots, T\}$

Computational Tasks

- Task $< D_i, L_i >$ of WD i
  - Size of the input data D_i
  - Computational complexity L_i

MEC System with Periodic Tasks

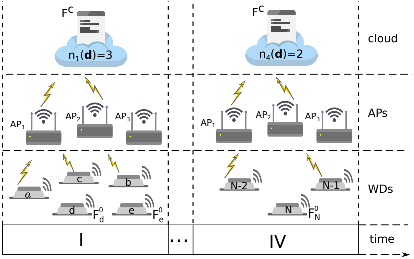


- Edge cloud
- Set of APs $\mathcal{A} = \{1, 2, ..., A\}$
- Set of WDs $\mathcal{N} = \{1, 2, ..., N\}$
- Set of time slots $\mathcal{T} = \{1, 2, \ldots, T\}$

Computational Tasks

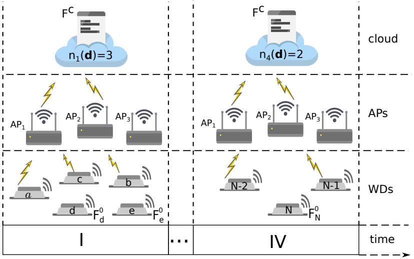
- Task $< D_i, L_i >$ of WD i
  - Size of the input data D_i
  - Computational complexity L_i
- Decision of WD i: $d_i = \begin{cases} (t, 0), & \text{local computing in time slot } t \\ (t, a), & \text{offloading via AP } a \text{ in time slot } t \end{cases}$
- Set of decisions for all WDs is a *strategy profile* $\mathbf{d}$

Communication Model



- $P_{i,a}$: transmit power of WD i on AP a
- $R_{i,a}$: PHY rate of WD i on AP a
- $n_{(t,a)}(\mathbf{d})$: number of WDs that offload in time slot t via AP a

Communication Model



- $P_{i,a}$: transmit power of WD i on AP a
- $R_{i,a}$: PHY rate of WD i on AP a
- $n_{(t,a)}(\mathbf{d})$: number of WDs that offload in time slot t via AP a

Cloud offloading through AP a in time slot t

$f_a(n_{(t,a)}(\mathbf{d}))$: non-increasing function of $n_{(t,a)}(\mathbf{d})$

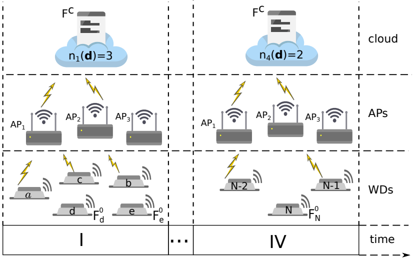
- Transmission time $T_{i,(t,a)}(\mathbf{d})$

$$T_{i,(t,a)}(\mathbf{d}) = \frac{D_i}{R_{i,a} \times f_a(n_{(t,a)}(\mathbf{d}))}$$

- Energy consumption $E^c_{i,(t,a)}(\mathbf{d})$

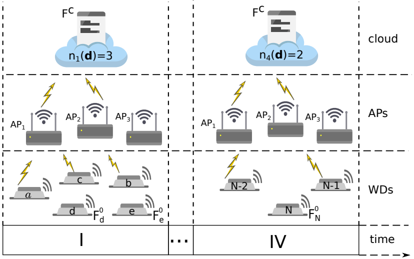
$$E^c_{i,(t,a)}(\mathbf{d}) = \frac{P_{i,a} D_i}{R_{i,a} \times f_a(n_{(t,a)}(\mathbf{d}))}$$

Computing Model



- F_i: computing capability of WD i
- F^c: computing capability of the cloud
- $n_t(\mathbf{d})$: total number of WDs that offload in time slot t

Computing Model



- F_i: computing capability of WD i
- F^c: computing capability of the cloud
- $n_t(\mathbf{d})$: total number of WDs that offload in time slot t

Local computing

- Task execution time

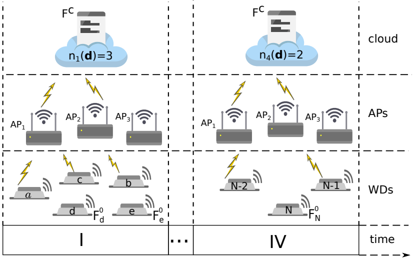
$$T_i^0 = \frac{L_i}{F_i}$$

- Energy consumption

v_i: energy consumption of WD i per CPU cycle

$$E_i^0 = v_i L_i$$

Computing Model



- F_i: computing capability of WD i
- F^c: computing capability of the cloud
- $n_t(\mathbf{d})$: total number of WDs that offload in time slot t

Local computing

- Task execution time

$$T_i^0 = \frac{L_i}{F_i}$$

- Energy consumption

$$E_i^0 = v_i L_i$$

Cloud offloading in time slot t

- Task execution time in time slot t

$$T_{i,t}^{c,exe}(\mathbf{d}) = \frac{L_i}{F^c \times f_i(n_t(\mathbf{d}))}$$

$f_i(n_t(\mathbf{d}))$: non-increasing function of $n_t(\mathbf{d})$

Local Computing Cost

$$C_i^0 = \gamma_i^T \underbrace{T_i^0}_{\text{delay}} + \gamma_i^E \underbrace{E_i^0}_{\text{energy}}$$

Cloud Offloading Cost

$$C^c_{i,(t,a)}(\mathbf{d}) = \gamma_i^T \underbrace{\Big(\overbrace{T_{i,(t,a)}(\mathbf{d})}^{\text{transmission time}} + \overbrace{T^{c,exe}_{i,t}(\mathbf{d})}^{\text{execution time}} \Big)}_{\text{delay}} + \gamma_i^E \underbrace{E^c_{i,(t,a)}(\mathbf{d})}_{\text{energy to offload}}$$

Local Computing Cost

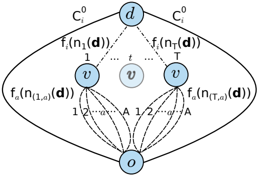
$$C_i^0 = \gamma_i^T \underbrace{T_i^0}_{\text{delay}} + \gamma_i^E \underbrace{E_i^0}_{\text{energy}}$$

Cloud Offloading Cost

$$C_{i,(t,a)}^c(\mathbf{d}) = \gamma_i^T \underbrace{\Big(\overbrace{T_{i,(t,a)}(\mathbf{d})}^{\text{transmission time}} + \overbrace{T_{i,t}^{c,exe}(\mathbf{d})}^{\text{execution time}} \Big)}_{\text{delay}} + \gamma_i^E \underbrace{E_{i,(t,a)}^c(\mathbf{d})}_{\text{energy to offload}}$$

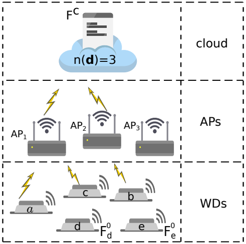
Selfish Computation Offloading

- Interactions between WDs modeled as a strategic game $\Gamma = <\mathcal{N}, (\mathfrak{D}_i)_i, (C_i)_i>$



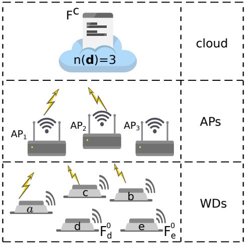
- Player specific network congestion game
  - Existence of Nash equilibria (NE) is not known in general

Single Time Slot and Elastic Cloud (Paper B)



- Decision of WD i: $d_i = \begin{cases} 0, & \text{local computing} \\ a, & \text{offloading via AP } a \end{cases}$

Single Time Slot and Elastic Cloud (Paper B)

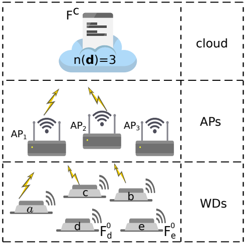


- Decision of WD i: $d_i = \begin{cases} 0, & \text{local computing} \\ a, & \text{offloading via AP } a \end{cases}$

NE Existence

- NE exist in the case of an elastic cloud and a single time slot
  - Proof based on generalized ordinal potential function

Single Time Slot and Elastic Cloud (Paper B)



- Decision of WD i: $d_i = \begin{cases} 0, & \text{local computing} \\ a, & \text{offloading via AP } a \end{cases}$

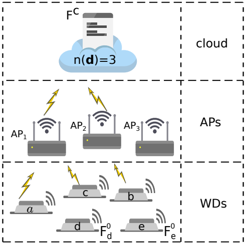
NE Existence

- NE exist in the case of an elastic cloud and a single time slot
  - Proof based on generalized ordinal potential function

ImprovementPath (IP) Algorithm

- Starts from an arbitrary initial strategy profile
- One WD at a time is allowed to perform an improvement step

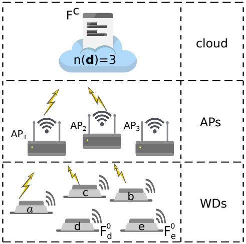
Single Time Slot and Non-Elastic Cloud (Paper B)



- Not a potential game - proof by constructing a cycle

$$(1,2,1,0,0) \xrightarrow[c]{} (1,2,2,0,0) \xrightarrow[b]{} (1,0,2,0,0) \xrightarrow[d]{}$$
$$(1,0,2,2,0) \xrightarrow[e]{} (1,0,2,2,2) \xrightarrow[c]{} (1,0,1,2,2) \xrightarrow[b]{}$$
$$(1,3,1,2,2) \xrightarrow[e]{} (1,3,1,2,0) \xrightarrow[d]{} (1,3,1,0,0) \xrightarrow[b]{}$$
$$(1,2,1,0,0)$$

Single Time Slot and Non-Elastic Cloud (Paper B)



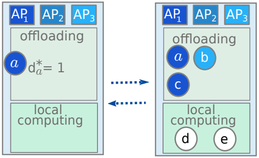
- Not a potential game - proof by constructing a cycle

$$
\begin{aligned}
&(1,2,1,0,0) \xrightarrow[c]{} (1,2,2,0,0) \xrightarrow[b]{} (1,0,2,0,0) \xrightarrow[d]{} \\
&(1,0,2,2,0) \xrightarrow[e]{} (1,0,2,2,2) \xrightarrow[c]{} (1,0,1,2,2) \xrightarrow[b]{} \\
&(1,3,1,2,2) \xrightarrow[e]{} (1,3,1,2,0) \xrightarrow[d]{} (1,3,1,0,0) \xrightarrow[b]{} \\
&(1,2,1,0,0)
\end{aligned}
$$

NE existence

- The game admits a pure NE
  - Constructive proof - Join and Play Best Reply (JP-BR) algorithm

- Induction phase - starting from an empty system, WDs enter the game one at a time and play BR



- Update phase - WDs are allowed to update their BR one at a time

Multiple Time Slots (Paper C)

- JP-BR may not converge to a NE

Multiple Time Slots (Paper C)

- JP-BR may not converge to a NE
- NE exists in the case of multiple time slots

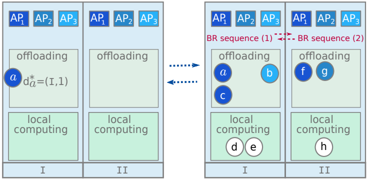
Multiple Time Slots (Paper C)

- JP-BR may not converge to a NE
- NE exists in the case of multiple time slots

Coordinated Myopic Alternating Best (MB) Algorithm

- WDs enter the game one at a time and implement BR over all time slots

- Induction phase - starting from an empty system, WDs enter the game one at a time and play BR



- Update phase - two types of BR sequences are played alternatingly
  (1) WDs are not allowed to replace previous deviators
  (2) WDs are only allowed to replace previous deviators

Proposed Algorithms - Main Results

Computability

- Single time slot: JP-BR algorithm computes a NE of a game in $\mathcal{O}(N^2 \times A)$
- Multiple time slots: MB algorithm computes a NE of a game in $\mathcal{O}(N^2 \times T \times A)$ steps

Proposed Algorithms - Main Results

Computability

- Single time slot: JP-BR algorithm computes a NE of a game in $\mathcal{O}(N^2 \times A)$
- Multiple time slots: MB algorithm computes a NE of a game in $\mathcal{O}(N^2 \times T \times A)$ steps

Price of Anarchy (PoA) Bounds

- Upper bound on the PoA for the computation offloading game:
  - $N \leq T$: $PoA = 1$
  - $N > T$: $PoA \leq N + 1$
- Provides bound on approximation ratio

Performance Gain w.r.t Local Computing

Evaluation scenario

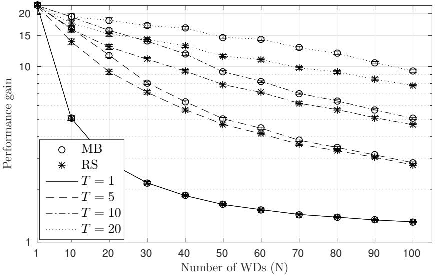
- $A = 4$ APs, $F^c = 100$ *Gcycles*, $F^c_{i,t}(n_t(\mathbf{d})) = \frac{F^c}{n_t(\mathbf{d})}$, $F_i \sim \mathcal{U}(0.5, 1)$ *Gcycles*
- Tasks: $D_i \sim \mathcal{U}(0.42, 2)$ Mb , $L_i \sim \mathcal{U}(0.1, 0.8)$ *Gcycles*

Performance Gain w.r.t Local Computing

Evaluation scenario

- $A = 4$ APs, $F^c = 100$ $Gcycles$, $F^c_{i,t}(n_t(\mathbf{d})) = \frac{F^c}{n_t(\mathbf{d})}$, $F_i \sim \mathcal{U}(0.5, 1)$ $Gcycles$
- Tasks: $D_i \sim \mathcal{U}(0.42, 2)$ Mb , $L_i \sim \mathcal{U}(0.1, 0.8)$ $Gcycles$

Performance gain (algorithm A) = $\dfrac{\text{sum of costs of WDs when computing locally}}{\text{sum of costs of WDs when using algorithhm A}}$



- *Performance gain* decreases with the number of WDs for both algorithms
- *Performance gain* of the MB algorithm is higher than that of the RS algorithm for $T > 1$ $\Longrightarrow$ coordination is important

Computational Complexity

Evaluation scenario

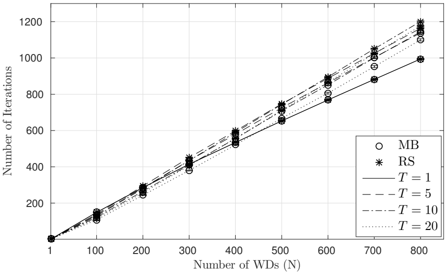
- $A = 4$ APs, $F^c = 100$ $Gcycles$, $F^c_{i,t}(n_t(\mathbf{d})) = \frac{F^c}{n_t(\mathbf{d})}$, $F_i \sim \mathcal{U}(0.5, 1)$ $Gcycles$
- Tasks: $D_i \sim \mathcal{U}(0.42, 2)$ Mb , $L_i \sim \mathcal{U}(0.1, 0.8)$ $Gcycles$

Computational Complexity

Evaluation scenario

- $A = 4$ APs, $F^c = 100$ $Gcycles$, $F^c_{i,t}(n_t(\mathbf{d})) = \frac{F^c}{n_t(\mathbf{d})}$, $F_i \sim \mathcal{U}(0.5, 1)$ $Gcycles$
- Tasks: $D_i \sim \mathcal{U}(0.42, 2)$ Mb , $L_i \sim \mathcal{U}(0.1, 0.8)$ $Gcycles$

Computational complexity



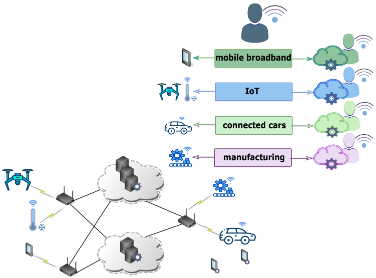
- Number of iterations scales approximately linearly with the number of WDs for both algorithms

Outline

1. Task Placement in Edge Computing Systems
   - Completion Time Minimization
     - Collaborative offloading supported by a cloud server (Paper A)
   - Completion Time and Energy Consumption Minimization
     - Scheduling of tasks over time slots, communication and computing resources (Paper B and Paper C)

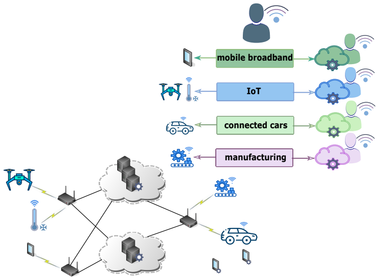
2. Task Placement and Resource Management in Edge Computing Systems
   - Completion Time Minimization
     - **Scheduling of tasks over network slices, communication and computing resources (Paper D and Paper E)**

Edge Computing Under Network Slicing



- Set $\mathcal{A}$ of access points (APs)
- Set $\mathcal{C}$ of edge clouds (ECs)
- Set $\mathcal{N}$ of wireless devices (WDs)
- Set $\mathcal{S}$ of network slices

Edge Computing Under Network Slicing

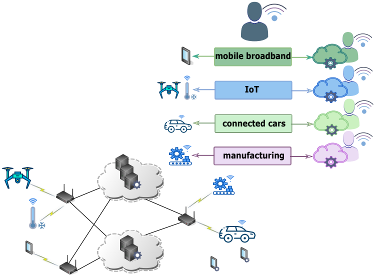


- Set $\mathcal{A}$ of access points (APs)
- Set $\mathcal{C}$ of edge clouds (ECs)
- Set $\mathcal{N}$ of wireless devices (WDs)
- Set $\mathcal{S}$ of network slices

Computational Tasks

- Task of WD i, $< D_i, L_i >$
  - size of the input data D_i
  - computational complexity L_i

Edge Computing Under Network Slicing



- Set $\mathcal{A}$ of access points (APs)
- Set $\mathcal{C}$ of edge clouds (ECs)
- Set $\mathcal{N}$ of wireless devices (WDs)
- Set $\mathcal{S}$ of network slices

Computational Tasks

$$d_i \in \mathfrak{D}_i, \mathfrak{D}_i = \{i\} \cup \{(a,c,s) | a \in \mathcal{A}, c \in \mathcal{C}, s \in \mathcal{S}\}$$

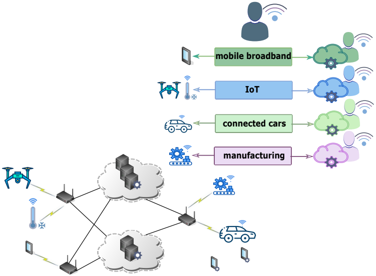
- Task of WD i, $< D_i, L_i >$
  - size of the input data D_i
  - computational complexity L_i
- Decision d_i of WD $i \in \mathcal{N}$:

d_i

local computing

offloading: in which slice s, through which AP a and to which EC c?

Edge Computing Under Network Slicing



- Set $\mathcal{A}$ of access points (APs)
- Set $\mathcal{C}$ of edge clouds (ECs)
- Set $\mathcal{N}$ of wireless devices (WDs)
- Set $\mathcal{S}$ of network slices

Computational Tasks

$d_i \in \mathfrak{D}_i, \mathfrak{D}_i = \{i\} \cup \{(a,c,s) | a \in \mathcal{A}, c \in \mathcal{C}, s \in \mathcal{S}\}$

- Task of WD i, $< D_i, L_i >$
  - size of the input data D_i
  - computational complexity L_i
- Decision d_i of WD $i \in \mathcal{N}$:

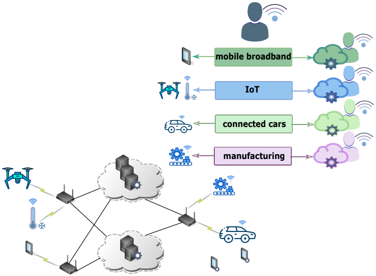
d_i

local computing

offloading: in which slice s, through which AP a and to which EC c?

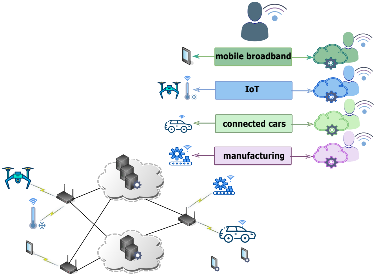
- Set of decisions for all WDs is a *strategy profile* **d**

Communication Model



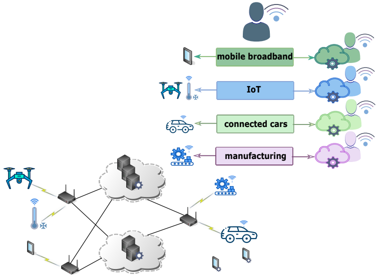
- Network level management:
  - Network operator shares resources among slices according to policy $\mathcal{P}_b$

Communication Model



- Network level management:
  - Network operator shares resources among slices according to policy $\mathcal{P}_b$
- Slice level management:
  - Each slice $s \in \mathcal{S}$ shares resources among WDs according to policy $\mathcal{P}_w^s$

Communication Model



- Network level management:
  - Network operator shares resources among slices according to policy $\mathcal{P}_b$
- Slice level management:
  - Each slice $s \in \mathcal{S}$ shares resources among WDs according to policy $\mathcal{P}_w^s$

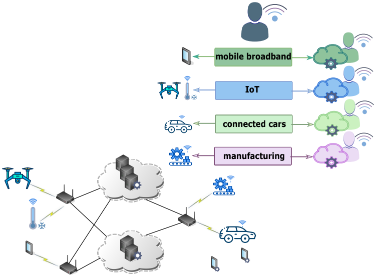
Task transmission time

- Uplink rate of WD i via AP a

$R_{i,a}$: PHY rate of WD i on AP a

$$\omega_{i,a}^s(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s) = b_a^s \, w_{i,a}^s \, R_{i,a}$$

Communication Model



- Network level management:
  - Network operator shares resources among slices according to policy $\mathcal{P}_b$
- Slice level management:
  - Each slice $s \in \mathcal{S}$ shares resources among WDs according to policy $\mathcal{P}_w^s$

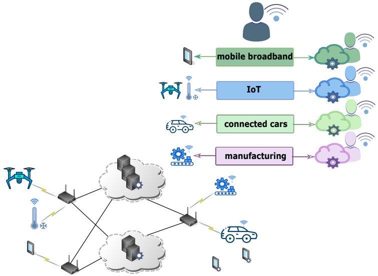
Task transmission time

b_a^s: bandwidth-slice provisioning coefficient (set by policy $\mathcal{P}_b$)

- Uplink rate of WD i via AP a

$$\omega_{i,a}^s(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s) = b_a^s \, w_{i,a}^s \, R_{i,a}$$

Communication Model



- Network level management:
  - Network operator shares resources among slices according to policy $\mathcal{P}_b$
- Slice level management:
  - Each slice $s \in \mathcal{S}$ shares resources among WDs according to policy $\mathcal{P}_w^s$

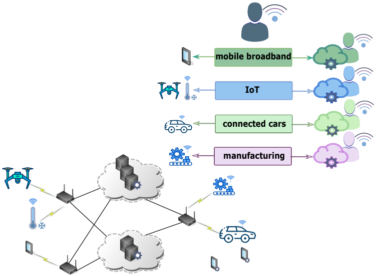
Task transmission time

$w_{i,a}^s$: uplink rate provisioning coefficient (set by policy $\mathcal{P}_w^s$)

- Uplink rate of WD i via AP a

$$\omega_{i,a}^s(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s) = b_a^s \, w_{i,a}^s \, R_{i,a}$$

Communication Model



- Network level management:
  - Network operator shares resources among slices according to policy $\mathcal{P}_b$
- Slice level management:
  - Each slice $s \in \mathcal{S}$ shares bandwidth among WDs according to policy $\mathcal{P}_w^s$

Task transmission time

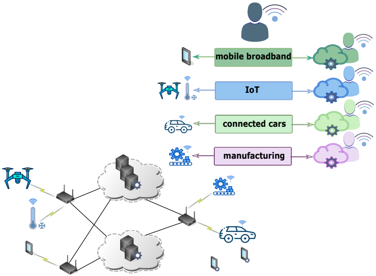
- Uplink rate of WD i via AP a

$$\omega_{i,a}^s(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s) = b_a^s \, w_{i,a}^s \, R_{i,a}$$

- Transmission time of WD i for offloading via AP a

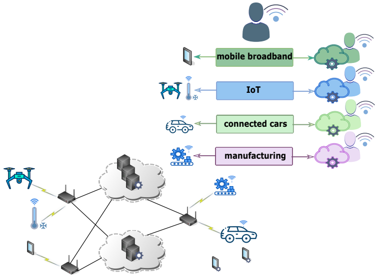
$$T_{i,a}^{tx,s}(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s) = \frac{D_i}{\omega_{i,a}^s(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s)}$$

Computation Model



- Local computing:
  - F_i: computing capability of WD i
- Computation offloading:
  - F_c^s: computing capability of cloud c in slice s
  - Slice s shares computing resources among WDs according to policy $\mathcal{P}_f^s$

Computation Model



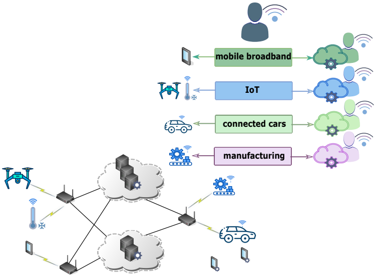
- Local computing:
  - F_i: computing capability of WD i
- Computation offloading:
  - F_c^s: computing capability of cloud c in slice s
  - Slice s shares computing resources among WDs according to policy $\mathcal{P}_f^s$

Local computing

- Task execution time

$$T_i^{exe} = \frac{L_i}{F_i}$$

Computation Model



- Local computing:
  - F_i: computing capability of WD i
- Computation offloading:
  - F_c^s: computing capability of cloud c in slice s
  - Slice s shares computing resources among WDs according to policy $\mathcal{P}_f^s$

$h_{i,s}$: goodness of slice s for WD i's task

Local computing

- Task execution time

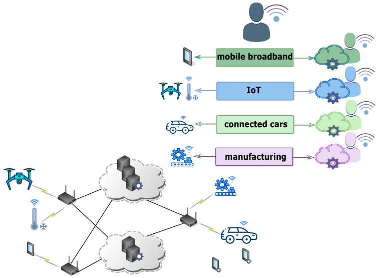
$$T_i^{exe} = \frac{L_i}{F_i}$$

Computation offloading

- Task execution time when offloading to cloud c in s

$$T_{i,c}^{ex,s}(\mathbf{d}, \mathcal{P}_f^s) = \frac{h_{i,s} L_i}{f_{i,c}^s F_c^s}$$

Computation Model



- Local computing:
  - F_i: computing capability of WD i
- Computation offloading:
  - F_c^s: computing capability of cloud c in slice s
  - Slice s shares computing resources among WDs according to policy $\mathcal{P}_f^s$

Local computing

- Task execution time

$$T_i^{exe} = \frac{L_i}{F_i}$$

Computation offloading

- Task execution time when offloading to cloud c in s

$$T_{i,c}^{ex,s}(\mathbf{d}, \mathcal{P}_f^s) = \frac{h_{i,s} L_i}{f_{i,c}^s F_c^s}$$

$f_{i,c}^s$: computing power provisioning coefficient (set by policy $\mathcal{P}_f^s$)

Cost - Task Completion Time

Cost - Task Completion Time

Computing cost of WD i

execution

$$C_i(\mathbf{d},\mathcal{P}_b,\mathcal{P}_w^s,\mathcal{P}_f^s)=\begin{cases}T_i^{ex}, & \text{local computing } d_i=i\\ T_{i,a}^{tx,s}(\mathbf{d},\mathcal{P}_b,\mathcal{P}_w^s)+T_{i,c}^{ex,s}(\mathbf{d},\mathcal{P}_f^s), & \text{offloading } d_i=(a,c,s)\end{cases}$$

Cost - Task Completion Time

Computing cost of WD i

transmission

execution

$$C_i(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s, \mathcal{P}_f^s) = \begin{cases} T_i^{ex}, & \text{local computing } d_i = i \\ T_{i,a}^{tx,s}(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w^s) + T_{i,c}^{ex,s}(\mathbf{d}, \mathcal{P}_f^s), & \text{offloading } d_i = (a, c, s) \end{cases}$$

Cost - Task Completion Time

Computing cost of WD i

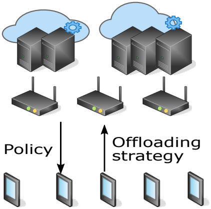
$$C_i(\mathbf{d},\mathcal{P}_b,\mathcal{P}_w^s,\mathcal{P}_f^s)=\begin{cases}T_i^{ex}, & \text{local computing } d_i=i\\ T_{i,a}^{tx,s}(\mathbf{d},\mathcal{P}_b,\mathcal{P}_w^s)+T_{i,c}^{ex,s}(\mathbf{d},\mathcal{P}_f^s), & \text{offloading } d_i=(a,c,s)\end{cases}$$

System cost

$$C(\mathbf{d},\mathcal{P}_b,\mathcal{P}_w,\mathcal{P}_f)=\sum_{i\in N}C_i(\mathbf{d},\mathcal{P}_b,\mathcal{P}_w^s,\mathcal{P}_f^s)$$

Single Network Slice (Paper D)

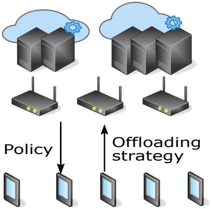
Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game

Single Network Slice (Paper D)

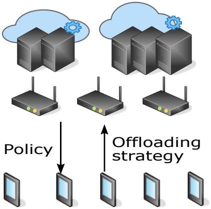
Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game
    - Cost minimizing (CM) operator $\mathfrak{A}_{CM} = \{(\mathbf{w}, \mathbf{f}) | \mathbf{w} \in \mathbb{R}_{\geq 0}^{A \times N}, \mathbf{f} \in \mathbb{R}_{\geq 0}^{C \times N}\}$

Single Network Slice (Paper D)

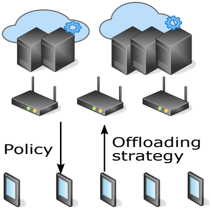
Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game
  - Cost minimizing (CM) operator $\mathfrak{A}_{CM} = \{(\mathbf{w}, \mathbf{f}) | \mathbf{w} \in \mathbb{R}_{\geq 0}^{A \times N}, \mathbf{f} \in \mathbb{R}_{\geq 0}^{C \times N}\}$
  - Time fair (TF) operator $\mathfrak{A}_{TF} = \{(\mathbf{w}, \mathbf{f}) | w_{i,a} = 1, f_{i,c} = 1, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}, \forall c \in \mathcal{C}\}$

Single Network Slice (Paper D)

Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game
  - Cost minimizing (CM) operator $\mathfrak{A}_{CM} = \{(\mathbf{w}, \mathbf{f}) | \mathbf{w} \in \mathbb{R}^{A \times N}_{\geq 0}, \mathbf{f} \in \mathbb{R}^{C \times N}_{\geq 0}\}$
  - Time fair (TF) operator $\mathfrak{A}_{TF} = \{(\mathbf{w}, \mathbf{f}) | w_{i,a} = 1, f_{i,c} = 1, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}, \forall c \in \mathcal{C}\}$

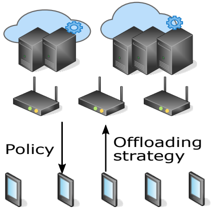
Objective of the operator $o \in \{CM, TF\}$

- Minimization of total cost

$$\min_{\{(\mathcal{P}_w, \mathcal{P}_f) | (\mathbf{w}, \mathbf{f}) \in \mathfrak{A}_o\}} C(\mathbf{d}, \mathcal{P}_w, \mathcal{P}_f)$$

Single Network Slice (Paper D)

Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game
  - Cost minimizing (CM) operator $\mathfrak{A}_{CM} = \{(\mathbf{w}, \mathbf{f}) | \mathbf{w} \in \mathbb{R}_{\geq 0}^{A \times N}, \mathbf{f} \in \mathbb{R}_{\geq 0}^{C \times N}\}$
  - Time fair (TF) operator $\mathfrak{A}_{TF} = \{(\mathbf{w}, \mathbf{f}) | w_{i,a} = 1, f_{i,c} = 1, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}, \forall c \in \mathcal{C}\}$

Objective of the operator $o \in \{CM, TF\}$

- Minimization of total cost

$$\min_{\{(\mathcal{P}_w, \mathcal{P}_f) | (\mathbf{w}, \mathbf{f}) \in \mathfrak{A}_o\}} C(\mathbf{d}, \mathcal{P}_w, \mathcal{P}_f)$$

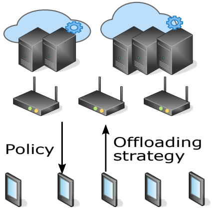
Objective of WDs

- Minimization of own cost

$$\min_{d_i \in \mathfrak{D}_i} C_i(\mathbf{d}, \mathcal{P}_w^*, \mathcal{P}_f^*)$$

Single Network Slice (Paper D)

Mobile Edge Computation Offloading Game (MEC-OG)



- Multi-leader common-follower Stackelberg game
  - Cost minimizing (CM) operator $\mathfrak{A}_{CM} = \{(\mathbf{w}, \mathbf{f}) | \mathbf{w} \in \mathbb{R}_{\geq 0}^{A \times N}, \mathbf{f} \in \mathbb{R}_{\geq 0}^{C \times N}\}$
  - Time fair (TF) operator $\mathfrak{A}_{TF} = \{(\mathbf{w}, \mathbf{f}) | w_{i,a} = 1, f_{i,c} = 1, \forall i \in \mathcal{N}, \forall a \in \mathcal{A}, \forall c \in \mathcal{C}\}$

Objective of the operator $o \in \{CM, TF\}$

- Minimization of total cost

$$\min_{\{(\mathcal{P}_w, \mathcal{P}_f) | (\mathbf{w}, \mathbf{f}) \in \mathfrak{A}_o\}} C(\mathbf{d}, \mathcal{P}_w, \mathcal{P}_f)$$

Objective of WDs

- Minimization of own cost

$$\min_{d_i \in \mathfrak{D}_i} C_i(\mathbf{d}, \mathcal{P}_w^*, \mathcal{P}_f^*)$$

Strategic game played by WDs

- Player-specific weighted congestion game Γ^{CM} under CM operator
- Player-specific congestion game Γ^{TF} under TF operator

Resource Allocation Policy of the CM operator

- Best response of the CM operator to strategy profile **d** chosen by WDs

$$w^*_{i,a}(\mathbf{d}) = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}, \forall i \in O_a(\mathbf{d}), \forall a \in \mathcal{A}$$

$$f^*_{i,c}(\mathbf{d}) = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}, \forall i \in O_c(\mathbf{d}), \forall c \in \mathcal{C}$$

Resource Allocation Policy of the CM operator

- Best response of the CM operator to strategy profile **d** chosen by WDs

$$w^*_{i,a}(\mathbf{d}) = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}, \forall i \in O_a(\mathbf{d}), \forall a \in \mathcal{A}$$

$$f^*_{i,c}(\mathbf{d}) = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}, \forall i \in O_c(\mathbf{d}), \forall c \in \mathcal{C}$$

Game Γ^{CM} under the optimal operator policy

- We transform Γ^{CM} into a congestion game Γ^* with resource dependent weights

Offloading cost: $C^c_{i,a}(\mathbf{d}) = \omega_{i,a}\sum_{j\in O_a(\mathbf{d})}\omega_{j,a} + \omega_{i,c}\sum_{j\in O_c(\mathbf{d})}\omega_{j,c}$

Weights: $\omega_{i,a} = \sqrt{\frac{D_i}{R_{i,a}}}, \omega_{i,c} = \sqrt{\frac{L_i}{F^c}}$

Resource Allocation Policy of the CM operator

- Best response of the CM operator to strategy profile $\mathbf{d}$ chosen by WDs

$$w^*_{i,a}(\mathbf{d}) = \frac{\sqrt{D_i/R_{i,a}}}{\sum_{j\in O_a(\mathbf{d})}\sqrt{D_j/R_{j,a}}}, \forall i \in O_a(\mathbf{d}), \forall a \in \mathcal{A}$$

$$f^*_{i,c}(\mathbf{d}) = \frac{\sqrt{L_i/F^c}}{\sum_{j\in O_c(\mathbf{d})}\sqrt{L_j/F^c}}, \forall i \in O_c(\mathbf{d}), \forall c \in \mathcal{C}$$

Game Γ^{CM} under the optimal operator policy

- We transform Γ^{CM} into a congestion game Γ^* with resource dependent weights

Offloading cost: $C^c_{i,a}(\mathbf{d}) = \omega_{i,a}\sum_{j\in O_a(\mathbf{d})}\omega_{j,a} + \omega_{i,c}\sum_{j\in O_c(\mathbf{d})}\omega_{j,c}$

Weights: $\omega_{i,a} = \sqrt{\frac{D_i}{R_{i,a}}}, \omega_{i,c} = \sqrt{\frac{L_i}{F^c}}$

- Does strategic game Γ^* have a Nash equilibrium (NE)?

MEC-OG with the CM Operator

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
    - Proof based on exact potential function

MEC-OG with the CM Operator

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Improve Local Computing (ILC) algorithm

- Starts from a strategy profile in which all WDs perform computation locally
- Lets WDs to start offloading in non-increasing order of their task complexities
  - Results in minimal number of iterations

MEC-OG with the CM Operator

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Improve Local Computing (ILC) algorithm

- Starts from a strategy profile in which all WDs perform computation locally
- Lets WDs to start offloading in non-increasing order of their task complexities
  - Results in minimal number of iterations

SPE existence

- The MEC-OG with the CM operator has a SPE $(\mathbf{d}^*, \mathcal{P}_w^*, \mathcal{P}_f^*)$

MEC-OG with the CM Operator

NE existence

- Game Γ^* has a NE $\mathbf{d}^*$
  - Proof based on exact potential function

Improve Local Computing (ILC) algorithm

- Starts from a strategy profile in which all WDs perform computation locally
- Lets WDs to start offloading in non-increasing order of their task complexities
  - Results in minimal number of iterations

SPE existence

- The MEC-OG with the CM operator has a SPE $(\mathbf{d}^*, \mathcal{P}_w^*, \mathcal{P}_f^*)$

Price of Anarchy (PoA) bound

- Ratio of worst case NE cost and minimal social cost $PoA \leq \frac{3+\sqrt{5}}{2} \approx 2.62$
- Provides bound on approximation ratio

MEC-OG with the TF Operator

NE existence

- Game Γ^{TF} is not a potential game (cycle from Paper B)
- Game Γ^{TF} admits a pure NE $\mathbf{d}^*$
  - Constructive proof

MEC-OG with the TF Operator

NE existence

- Game Γ^{TF} is not a potential game (cycle from Paper B)
- Game Γ^{TF} admits a pure NE $\mathbf{d}^*$
  - Constructive proof

Join and Play Asynchronous Updates (JPAU) algorithm

- Starts from empty system
- Adds WDs one at a time
  - Lets them play their best replies - in a certain order
- Computational complexity $\mathcal{O}(AN^3)$

MEC-OG with the TF Operator

NE existence

- Game Γ^{TF} is not a potential game (cycle from Paper B)
- Game Γ^{TF} admits a pure NE $\mathbf{d}^*$
    - Constructive proof

Join and Play Asynchronous Updates (JPAU) algorithm

- Starts from empty system
- Adds WDs one at a time
    - Lets them play their best replies - in a certain order
- Computational complexity $\mathcal{O}(AN^3)$

SPE existence

- The MEC-OG with the TF operator has a SPE

MEC-OG with the TF Operator

NE existence

- Game Γ^{TF} is not a potential game (cycle from Paper B)
- Game Γ^{TF} admits a pure NE $\mathbf{d}^*$
  - Constructive proof

Join and Play Asynchronous Updates (JPAU) algorithm

- Starts from empty system
- Adds WDs one at a time
  - Lets them play their best replies - in a certain order
- Computational complexity $\mathcal{O}(AN^3)$

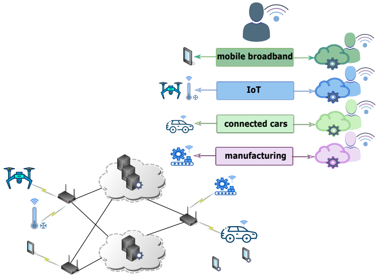
SPE existence

- The MEC-OG with the TF operator has a SPE

Price of Anarchy (PoA) bound

- $PoA \leq N+1$

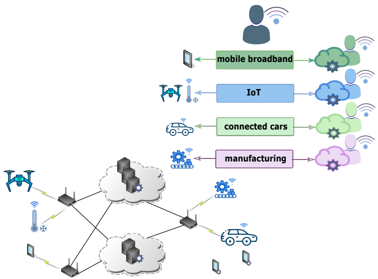
Multiple Network Slices (Paper E)



Joint Slice Selection and Edge Resource Management (JSS-ERM) problem:

- Find task placement $\mathbf{d}$ and policies $\mathcal{P}_b$, $\mathcal{P}_w^s$, $\mathcal{P}_f^s$ so as to minimize system cost $C(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w, \mathcal{P}_f)$

Multiple Network Slices (Paper E)



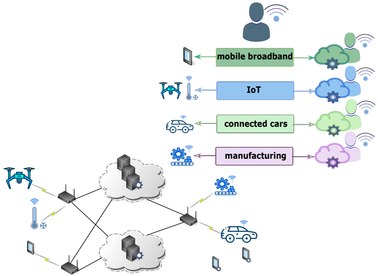
Joint Slice Selection and Edge Resource Management (JSS-ERM) problem:

- Find task placement $\mathbf{d}$ and policies $\mathcal{P}_b$, $\mathcal{P}_w^s$, $\mathcal{P}_f^s$ so as to minimize system cost $C(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w, \mathcal{P}_f)$

Computational Complexity

- JSS-ERM problem is NP-hard (already for a single slice case)
  - Reduction from the *minimum sum of squares* problem

Multiple Network Slices (Paper E)



Joint Slice Selection and Edge Resource Management (JSS-ERM) problem:

- Find task placement $\mathbf{d}$ and policies $\mathcal{P}_b$, $\mathcal{P}_w^s$, $\mathcal{P}_f^s$ so as to minimize system cost $C(\mathbf{d}, \mathcal{P}_b, \mathcal{P}_w, \mathcal{P}_f)$

Computational Complexity

- JSS-ERM problem is NP-hard (already for a single slice case)
  - Reduction from the *minimum sum of squares* problem
- Is there an approximate computationally efficient solution to the JSS-ERM problem?

Approximation scheme for the JSS-ERM problem

Decomposition based

- Step 1
  - Finding optimal intra-slice resource allocation policies $(\mathcal{P}_w^{s,*}, \mathcal{P}_f^{s,*})$
    - Closed-form expressions of the CM operator

Approximation scheme for the JSS-ERM problem

Decomposition based

- Step 1
  - Finding optimal intra-slice resource allocation policies $(\mathcal{P}_w^{s,*}, \mathcal{P}_f^{s,*})$
    - Closed-form expressions of the CM operator
- Step 2
  - Finding optimal inter-slice resource allocation policy $\mathcal{P}_b^*$
    - Closed-form expression for the inter-slice provisioning coefficients

$$b_a^{s,*} = \frac{\sum_{j \in O_{(a,s)}(\mathbf{d})} \sqrt{D_i/R_{i,a}}}{\sum_{s' \in \mathcal{S}} \sum_{j \in O(a,s')(\mathbf{d})} \sqrt{D_j/R_{j,a}}}, \forall a \in \mathcal{A}, \forall s \in \mathcal{S}$$

Approximation scheme for the JSS-ERM problem

Decomposition based

- Step 1
  - Finding optimal intra-slice resource allocation policies $(\mathcal{P}_w^{s,*}, \mathcal{P}_f^{s,*})$
    - Closed-form expressions of the CM operator
- Step 2
  - Finding optimal inter-slice resource allocation policy $\mathcal{P}_b^*$
    - Closed-form expression for the inter-slice provisioning coefficients

$$b_a^{s,*} = \frac{\sum_{j \in O_{(a,s)}(\mathbf{d})} \sqrt{D_i/R_{i,a}}}{\sum_{s' \in \mathcal{S}} \sum_{j \in O(a,s')(\mathbf{d})} \sqrt{D_j/R_{j,a}}}, \forall a \in \mathcal{A}, \forall s \in \mathcal{S}$$

- Step 3
  - Finding an equilibrium task placement vector $\mathbf{d}^*$
    Choose Offloading Slice (COS) algorithm
    - Starts from a strategy profile in which all WDs perform computation locally
    - Updates offloading decision of the WDs one at a time

Approximation scheme for the JSS-ERM problem

Decomposition based

- Step 1
  - Finding optimal intra-slice resource allocation policies $(\mathcal{P}_w^{s,*}, \mathcal{P}_f^{s,*})$
    - Closed-form expressions of the CM operator
- Step 2
  - Finding optimal inter-slice resource allocation policy $\mathcal{P}_b^*$
    - Closed-form expression for the inter-slice provisioning coefficients

$$b_a^{s,*} = \frac{\sum_{j \in O_{(a,s)}(\mathbf{d})} \sqrt{D_i/R_{i,a}}}{\sum_{s' \in \mathcal{S}} \sum_{j \in O(a,s')(\mathbf{d})} \sqrt{D_j/R_{j,a}}}, \forall a \in \mathcal{A}, \forall s \in \mathcal{S}$$

- Step 3
  - Finding an equilibrium task placement vector $\mathbf{d}^*$
    Choose Offloading Slice (COS) algorithm
    - Starts from a strategy profile in which all WDs perform computation locally
    - Updates offloading decision of the WDs one at a time
    - Terminates in $\mathcal{O}(N^2 \frac{C^{max}}{\epsilon} \log \frac{\sum_{i \in \mathcal{N}} T_i^{ex}}{\Psi^{min}})$ iterations

Approximation scheme for the JSS-ERM problem

Decomposition based

- Step 1
  - Finding optimal intra-slice resource allocation policies $(\mathcal{P}_w^{s,*}, \mathcal{P}_f^{s,*})$
    - Closed-form expressions of the CM operator
- Step 2
  - Finding optimal inter-slice resource allocation policy $\mathcal{P}_b^*$
    - Closed-form expression for the inter-slice provisioning coefficients

$$b_a^{s,*} = \frac{\sum_{j \in O_{(a,s)}(\mathbf{d})} \sqrt{D_i/R_{i,a}}}{\sum_{s' \in \mathcal{S}} \sum_{j \in O(a,s')(\mathbf{d})} \sqrt{D_j/R_{j,a}}}, \forall a \in \mathcal{A}, \forall s \in \mathcal{S}$$

- Step 3
  - Finding an equilibrium task placement vector $\mathbf{d}^*$
    Choose Offloading Slice (COS) algorithm
    - Starts from a strategy profile in which all WDs perform computation locally
    - Updates offloading decision of the WDs one at a time
    - Terminates in $\mathcal{O}(N^2 \frac{C^{max}}{\epsilon} \log \frac{\sum_{i \in \mathcal{N}} T_i^{ex}}{\Psi^{min}})$ iterations
- 2.62-approximation solution to the JSS-ERM problem

Performance Gain w.r.t Equal Sharing Across Slices

Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

Performance Gain w.r.t Equal Sharing Across Slices

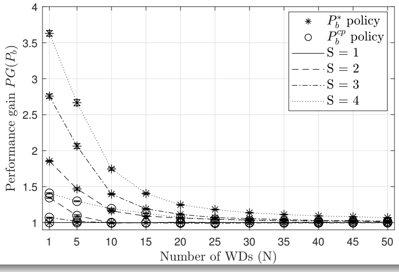
Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

System performance gain (PG)

- PG defined w.r.t. $\mathcal{P}_b^{eq}$ policy

$$PG(\mathcal{P}_b) = \frac{C(\mathbf{d}^*, \mathcal{P}_b^{eq}, \mathcal{P}_w^*, \mathcal{P}_f^*)}{C(\mathbf{d}^*, \mathcal{P}_b, \mathcal{P}_w^*, \mathcal{P}_f^*)}$$



Performance Gain w.r.t Equal Sharing Across Slices

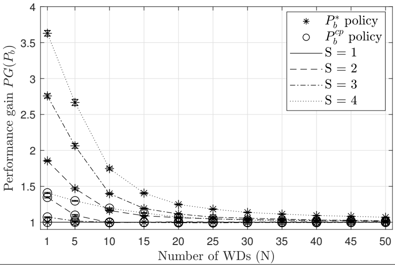
Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

System performance gain (PG)

- PG defined w.r.t. $\mathcal{P}_b^{eq}$ policy

$$PG(\mathcal{P}_b) = \frac{C(\mathbf{d}^*, \mathcal{P}_b^{eq}, \mathcal{P}_w^*, \mathcal{P}_f^*)}{C(\mathbf{d}^*, \mathcal{P}_b, \mathcal{P}_w^*, \mathcal{P}_f^*)}$$



- PG($\mathcal{P}_b^*$) = PG($\mathcal{P}_b^{cp}$) = 1 for $S = 1$

Performance Gain w.r.t Equal Sharing Across Slices

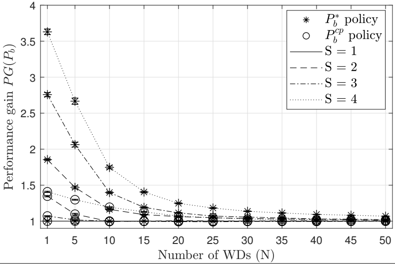
Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

System performance gain (PG)

- PG defined w.r.t. $\mathcal{P}_b^{eq}$ policy

$$PG(\mathcal{P}_b) = \frac{C(\mathbf{d}^*, \mathcal{P}_b^{eq}, \mathcal{P}_w^*, \mathcal{P}_f^*)}{C(\mathbf{d}^*, \mathcal{P}_b, \mathcal{P}_w^*, \mathcal{P}_f^*)}$$



- PG($\mathcal{P}_b^*$) = PG($\mathcal{P}_b^{cp}$) = 1 for $S = 1$
- PG($\mathcal{P}_b^*$) > 1 and PG($\mathcal{P}_b^{cp}$) > 1 for $S > 1$

Performance Gain w.r.t Equal Sharing Across Slices

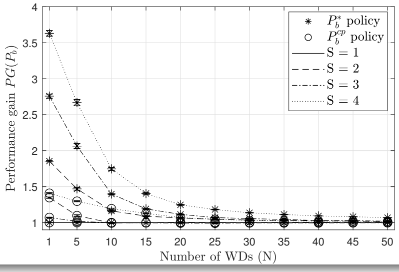
Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

System performance gain (PG)

- PG defined w.r.t. $\mathcal{P}_b^{eq}$ policy

$$PG(\mathcal{P}_b) = \frac{C(\mathbf{d}^*, \mathcal{P}_b^{eq}, \mathcal{P}_w^*, \mathcal{P}_f^*)}{C(\mathbf{d}^*, \mathcal{P}_b, \mathcal{P}_w^*, \mathcal{P}_f^*)}$$



- PG($\mathcal{P}_b^*$) = PG($\mathcal{P}_b^{cp}$) = 1 for $S = 1$
- PG($\mathcal{P}_b^*$) > 1 and PG($\mathcal{P}_b^{cp}$) > 1 for $S > 1$
- $\mathcal{P}_b^*$ achieves better PG than $\mathcal{P}_b^{cp}$ (up to 2.5 times greater)

Computational Complexity

Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

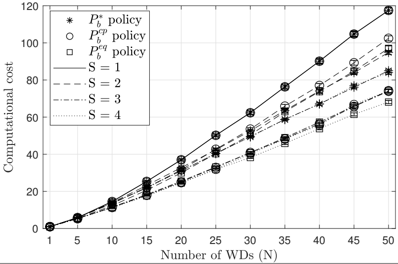
Computational Complexity

Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

Computational cost

- Number of updates needed for the COS algorithm to compute $\mathbf{d}^*$



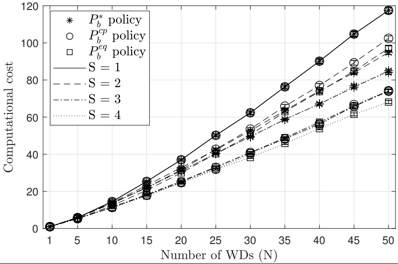
Computational Complexity

Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

Computational cost

- Number of updates needed for the COS algorithm to compute $\mathbf{d}^*$



- Computational cost scales approximately linearly with N

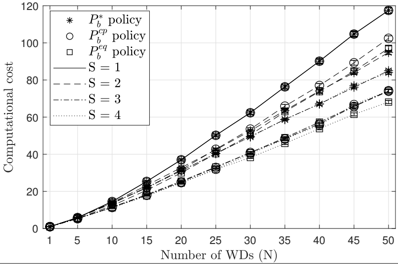
Computational Complexity

Evaluation scenario

- $A = 5$ heterogeneous APs, $C = 3$ heterogeneous ECs, heterogeneous slices
- WDs with heterogeneous tasks, PHY rates and computing capabilities
- Baseline policies: *equal sharing* policy $\mathcal{P}_b^{eq}$ and *cloud proportional* policy $\mathcal{P}_b^{cp}$

Computational cost

- Number of updates needed for the COS algorithm to compute $\mathbf{d}^*$



- Computational cost scales approximately linearly with N
- Computational cost decreases with S

Summary

DECENTRALIZED ALGORITHMS FOR EDGE COMPUTING RESOURCE MANAGEMENT

- Based on a game theoretical treatment of the problems
- Computationally efficient
- With performance guarantee

Summary

Decentralized Algorithms for Edge Computing Resource Management

- Based on a game theoretical treatment of the problems
- Computationally efficient
- With performance guarantee

Future Work

- Unknown information about WDs
- Unknown information about resource allocation policies
- Non-atomic models of computational tasks

Task Placement and Resource Allocation in Edge Computing Systems

Slađana Jošilo

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden

Stockholm, May 27, 2020