Resource Management for Mobile Edge Computing (MEC)

Jun ZHANG

Email: eejzhang@ust.hk

Collaborators

Yuyi Mao (PhD), Yinghao Yu (PhD), Juan Liu (Postdoc)

Khaled B. Letaief

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Outline

- **Introduction**
- **Resource Management for MEC**
  - Two-timescale computation offloading
  - MEC meets energy harvesting
  - Joint communication and computational resource management
  - Stochastic resource management for MEC
- **Key Takeaways**

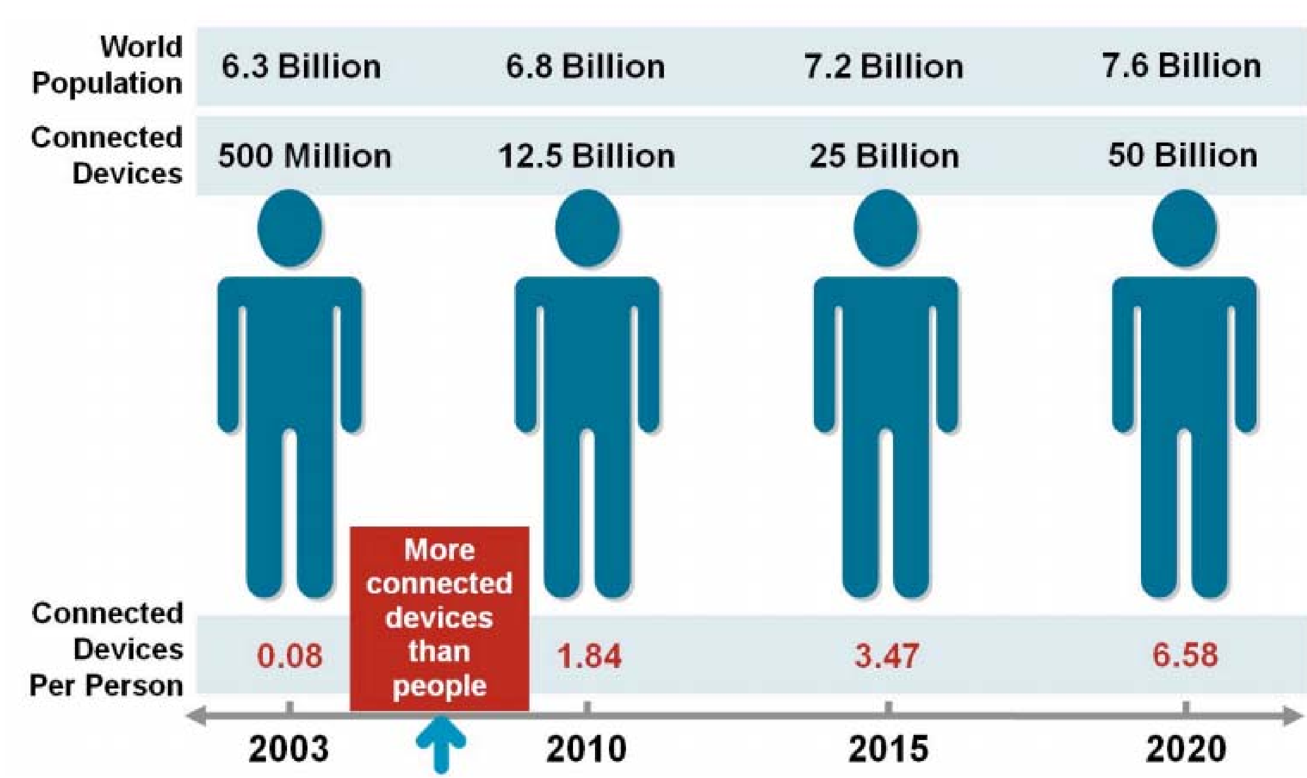
A Survey on MEC

- Y. Mao, C. You, **J. Zhang**, K. Huang, and K. B. Letaief, “A survey for mobile edge computing: The communication perspective,” submitted to *IEEE Commun. Surveys Tuts.*, under revision.
- Available: https://arxiv.org/pdf/1701.01090.pdf

- My other research interests
  - **Dense Cooperative Networks**
  - **Wireless Caching**
  - **Cloud Computing**
  - **Big Data Analytics**

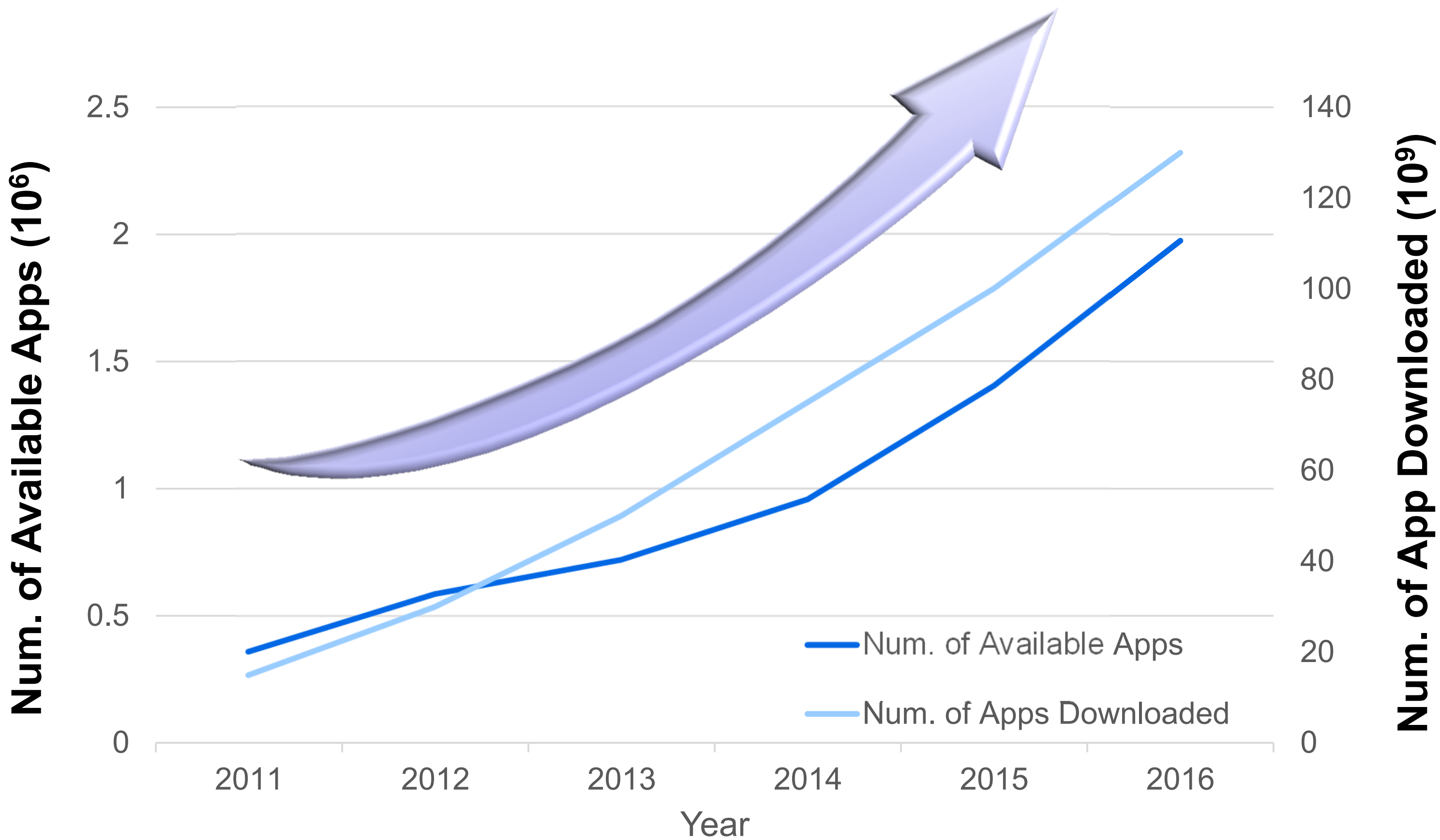
- For more information
  - http://www.ece.ust.hk/~eejzhang/

Era of Massive Connectivity



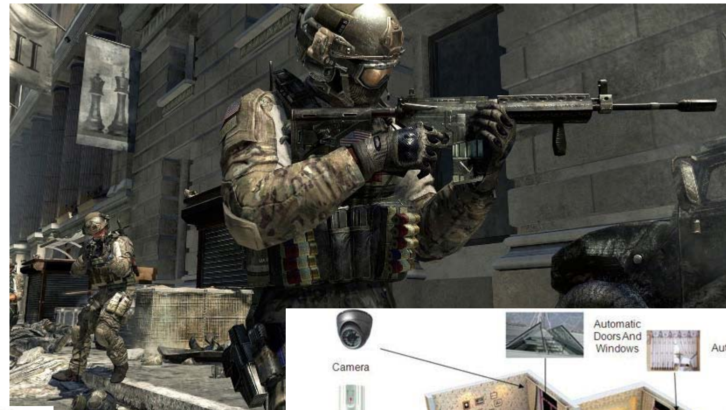
[Source: Cisco]

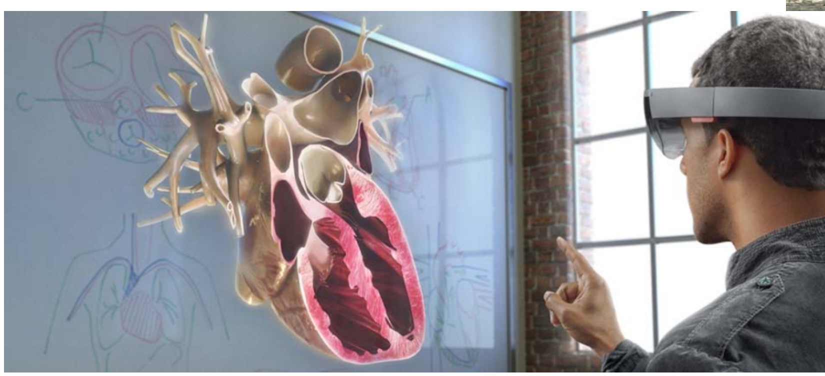
Growth of Mobile Applications Markets

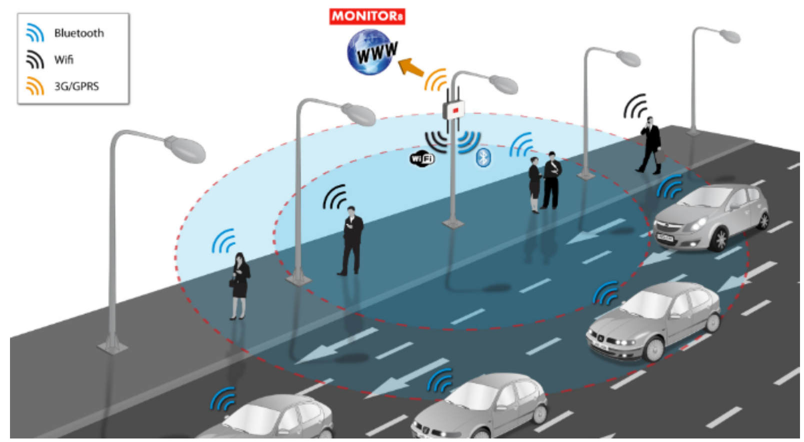


[Source: Statista]

Emerging Applications







- Computation-intensive
- Data-intensive
- Delay-sensitive

Evolution of Mobile Phones – A Mismatch

Nokia 8250 (2000)

250 phonebook +
60 call records +
50 notes

Sony Ericsson W800 (2005)

34MB storage +
SD card

iPhone (2007)

ARM11 620MHz
128MB DRAM
Up to 16GB storage

Samsung Galaxy S2 (2011)

ARM Cortex-A9
1.2GHz (dual-core)
1GB DRAM
4 GB Storage + SD card

iPhone 6s (2015)

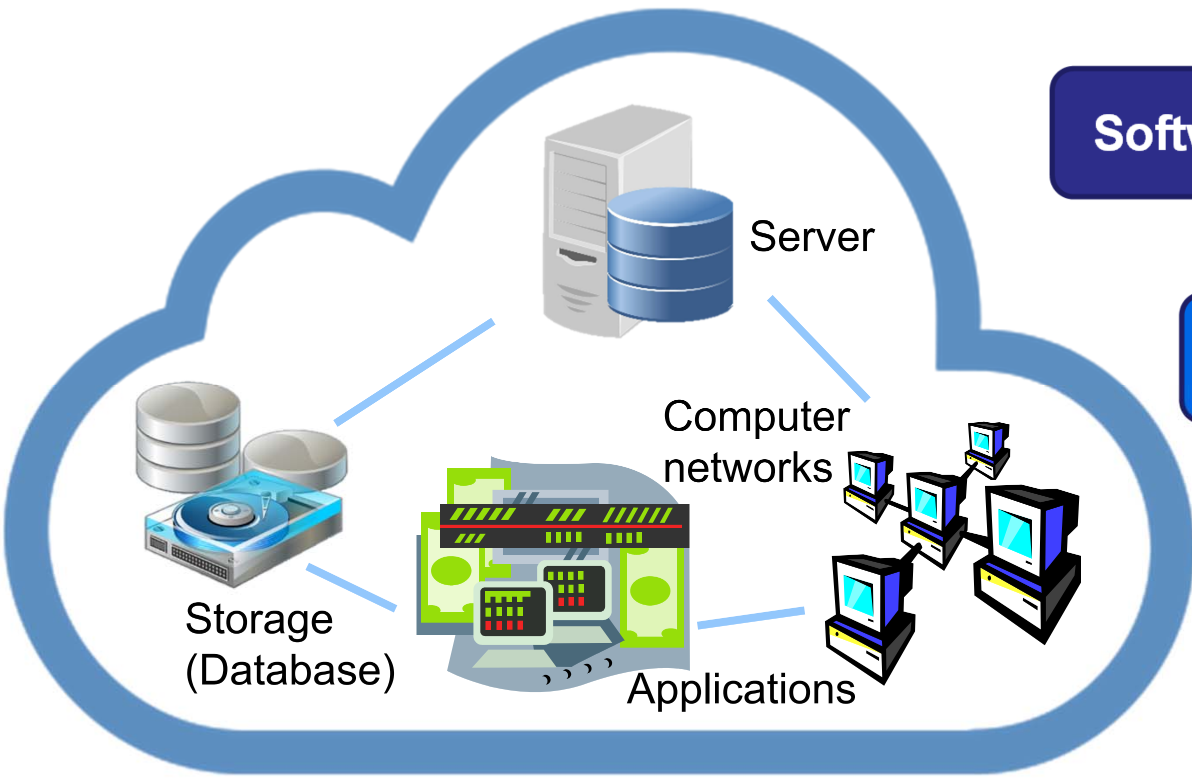
Apple A9+M9
1.8GHz (Tri-core)
2GB DRAM
Up to 128 GB Storage

…

830 mAh → 950 mAh → 1400 mAh → 1650 mAh → 1710 mAh

Old Paradigm for Mobile Computing (I)

- Cloud computing
  - *"Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand"*



Software as a Service (SaaS)

(End users)

Platform as a Service (PaaS)

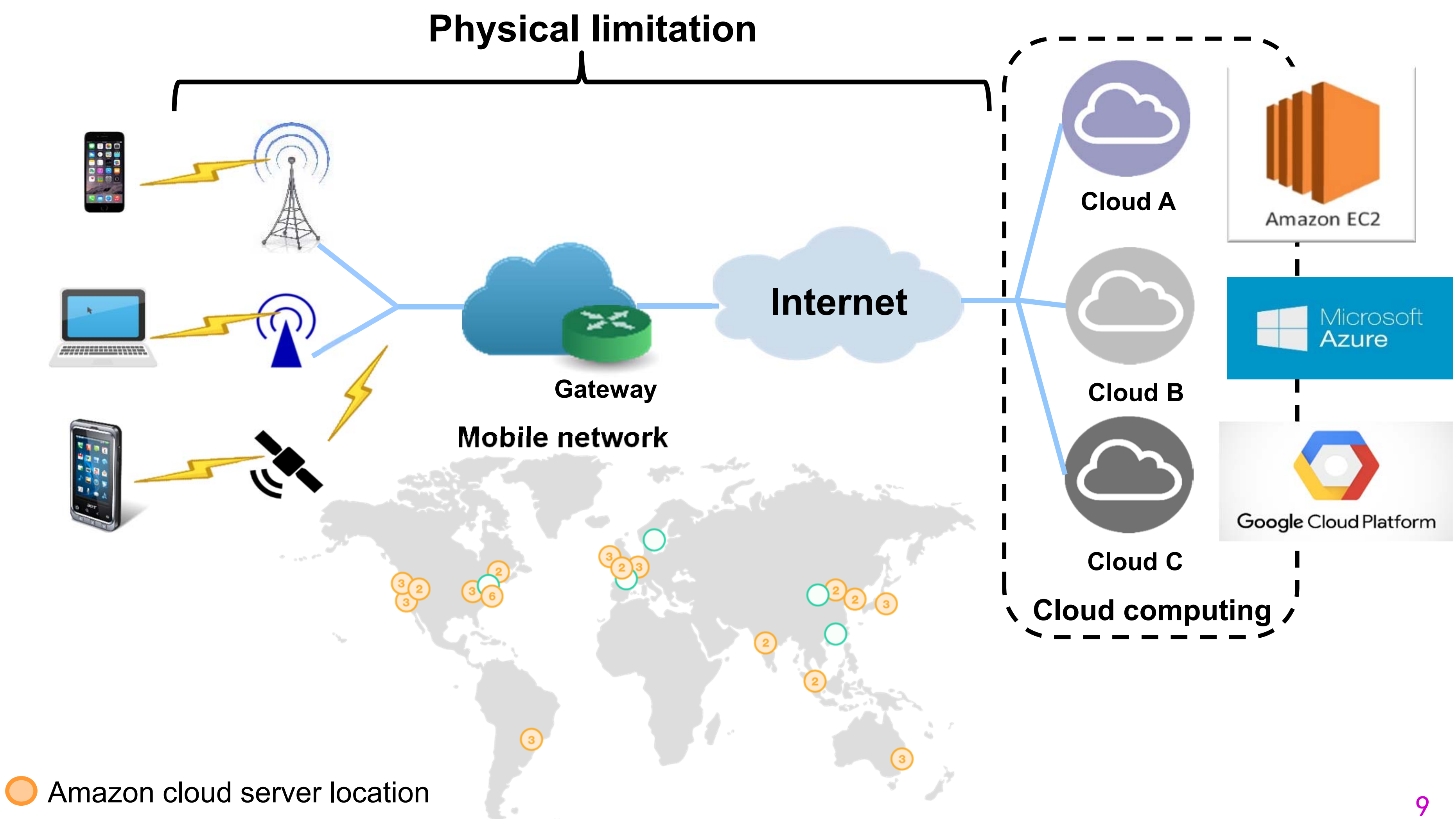
(Applications developers)

Infrastructure as a Service (IaaS)

(Network architects)

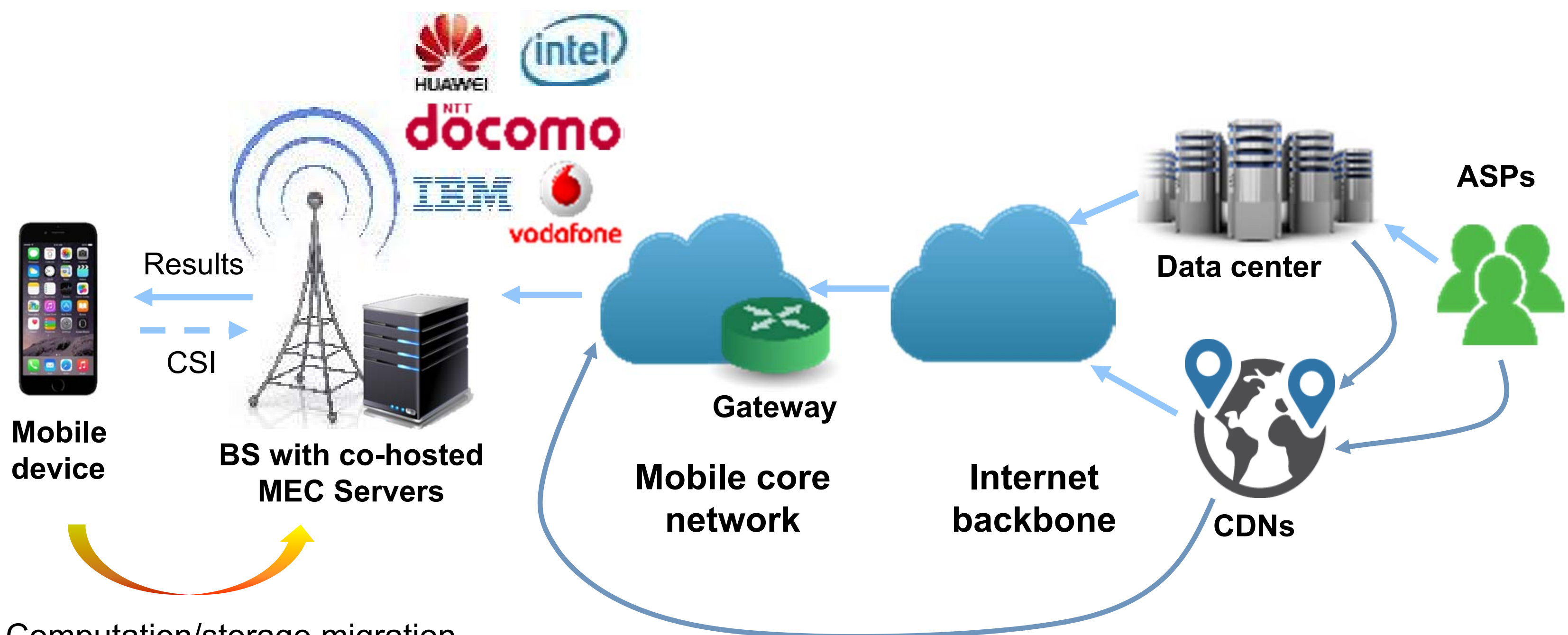
Old Paradigm for Mobile Computing (II)

- Mobile cloud computing (MCC)



A New Paradigm – Mobile Edge Computing

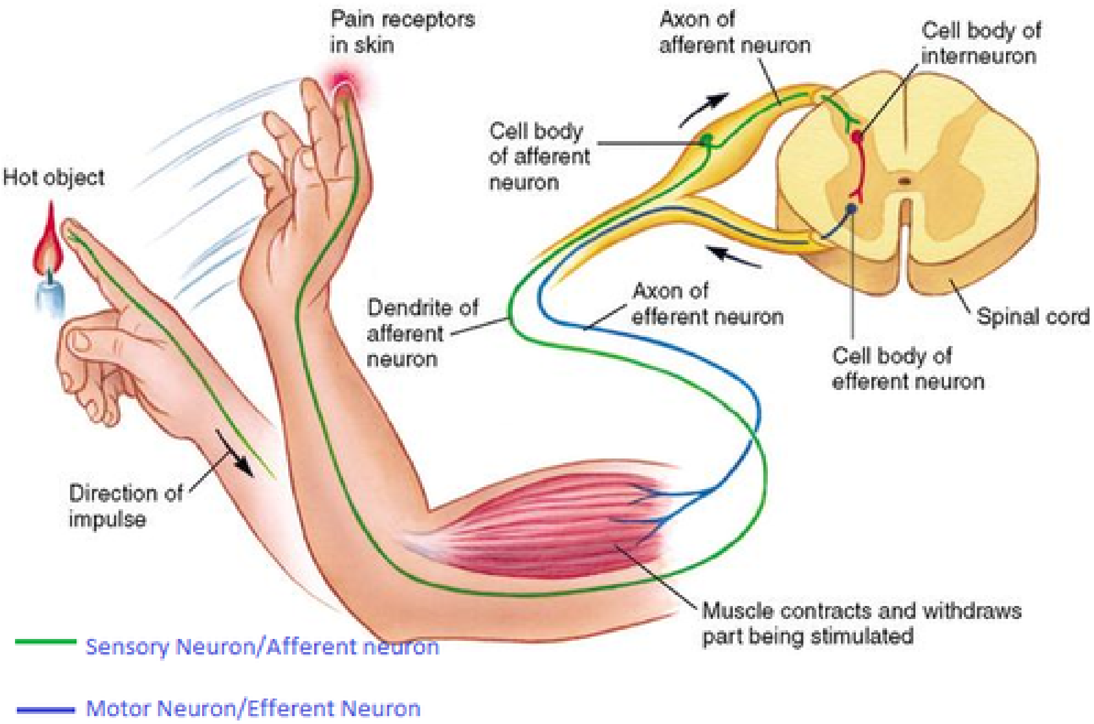
- Mobile edge computing (MEC)
  - Cloud computing capability and IT services within RAN [ETSI'14]



[ETSI'14] M. Patel et al., "Mobile-edge computing – Introductory technical white paper," *ETSI White Paper*, Sep. 2014.

MEC vs. Human Nervous System

- **Example**: Reflex arcs help the body respond to things like pain stimulus by creating a shorter neural pathway than the one going all the way to the brain



Mobile Edge Computing

	Mobile Edge Computing	Mobile Cloud Computing
Hardware	Small-scale data centers	Large-scale data centers
Server location	Co-located with wireless gateways, WiFi routers and BSs	Installed at dedicated buildings
Deployment	Lightweight configuration and planning	Sophisticated configuration and planning
Backhaul Usage	Infrequency use, alleviate congestion	Frequent use, likely to cause congestion
Distance to Users	Tens to hundreds of meters	Across the country boarders

- Ultra-low latency
- Mobile energy reduction
- Context-awareness
- Privacy/Security enhancement

Resource Management for MEC

- Computation offloading
  - Which tasks should be offloaded to the MEC server?
    - Effective transmissions for the offloading tasks
    - Based on **task characteristics** and **wireless channel conditions**
  - [Huang'12], [Zhang'13], [Baraossa'14], [Chen'15], etc.

- Joint radio and computational resource allocation
  - Maximize resource utilization
    - Properly allocate the available resources for each client
    - Joint management of both types of resource
    - Nested with the computation offloading decisions
  - [Baraossa'13], [Lorenzo'13], [Sardellitti'16], [You'17], etc.

In This Talk

- More on modeling/formulation, less on solution/algorithm
  - Identify key differences and challenges in MEC
- Systems
  - From single-user to multi-user systems
- Main objectives
  - Save energy
  - Reduce latency
- Emphasis on stochastic models
  - Less investigated before

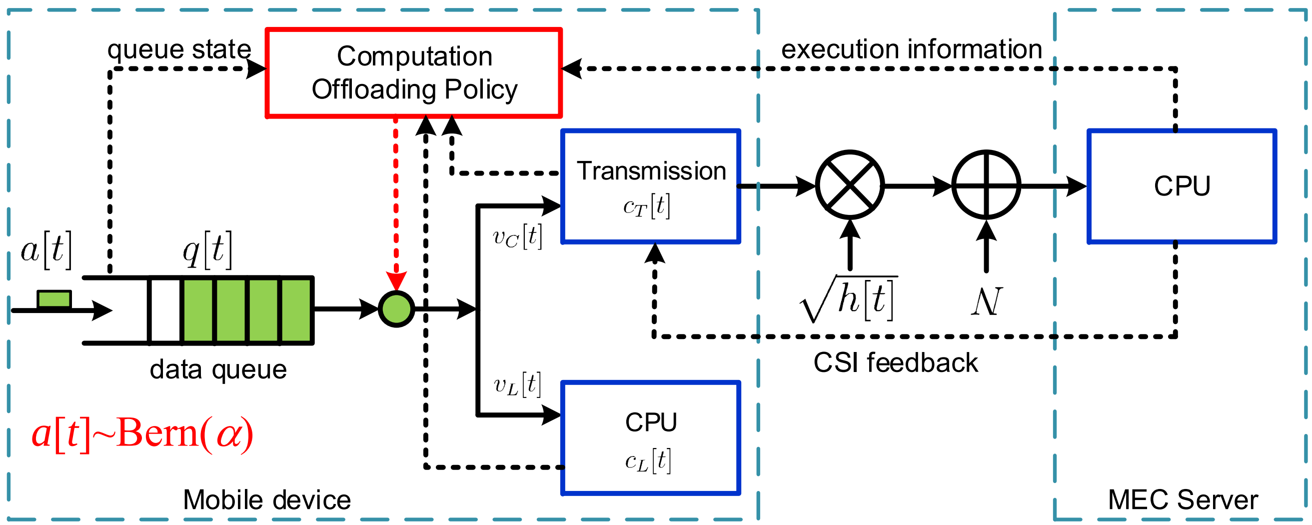
Two-Timescale Computation Offloading

Motivation

- Limitations of previous works
  - Most existing works assume the offloading processes can be **completed within a channel block**
  - Execution time of typical applications ~ tens of milliseconds
  - Duration of a channel block ~ a few milliseconds
- Two-timescale computation offloading
  - The larger timescale: *Whether to offload a task or not?*
  - The smaller timescale: *Transmission adaptation to CSI*
- Major challenge
  - The remote execution latency is unknown when making the offloading decision

System Model

- An MEC system with a mobile device and an MEC server



- Queueing model: $q[t+1] = \min\{(q[t] - v_L[t] - v_C[t])^+ + a[t], Q\}, t = 1, \cdots$
- Offloading decisions: $\mathcal{V} = \{v_C[t], v_L[t] \mid (0,1), (1,0), (1,1), (0,0)\}$
- Local execution: f_{loc} (Hz) (N time slots are needed)
- Task input data are encapsulated into M equal-size data packets
  - CSIT, ON/OFF transmit power control (*success trans. prob.* β)
  - One packet is transmitted at each time slot

Optimization Problem Formulation

- Power-constrained delay minimization
  - System state: $\boldsymbol{\tau}[t] = (q[t], c_T[t], c_L(t))$
  - Stochastic task scheduling policy: $\{g_{\boldsymbol{\tau}}^k\}$

$c_L(t)$: Processing state of the local CPU
$c_T(t)$: Processing state of the trans. Unit
$\{\pi_\tau\}$: Steady state probability

$$\min_{\{g_{\boldsymbol{\tau}}^k\}} \quad \overline{T} = \frac{1}{\alpha}\sum_{i=0}^{Q} i \sum_{m=0}^{M}\sum_{n=0}^{N-1}\pi_{(i,m,n)} + \eta N + (1-\eta)\, t_c \longleftarrow t_c = t_{tx} + N_{cloud} + t_{rx}$$

Queuing delay — Local — Remote

$$t_{tx} = M\sum_{j=1}^{\infty} j\,(1-\beta)^{(j-1)}\,\beta$$

$$\text{s.t.} \begin{cases} \overline{P} = \sum_{\boldsymbol{\tau}} \pi_{\boldsymbol{\tau}}\left(\mu_{\boldsymbol{\tau}} P_{loc} + \mu_{\boldsymbol{\tau}}^{tx} P_{tx}\right) \le \overline{P}_{\max} \\ \sum_{\boldsymbol{\tau}'} \chi_{\boldsymbol{\tau}',\boldsymbol{\tau}}\pi_{\boldsymbol{\tau}'} = \pi_{\pi}, \boldsymbol{\tau}\in\mathcal{S} \\ \sum_{i=0}^{Q}\sum_{m=0}^{M}\sum_{n=0}^{N-1}\pi_{(i,m,n)} = 1 \\ \sum_{k=1}^{4} g_{(i,m,n)}^k = 1, \forall i,m,n \\ g_{(i,m,n)}^k \ge 0, \forall i,m,n,k \end{cases}$$

← Average power constraint

Highly non-convex!

- η: Proportion of tasks that are executed locally

$$\eta = \frac{\sum_{\boldsymbol{\tau}\in\mathcal{S}_1}\pi_{\boldsymbol{\tau}} g_{\boldsymbol{\tau}}^1 + \sum_{\boldsymbol{\tau}\in\mathcal{S}_3}\pi_{\boldsymbol{\tau}} g_{\boldsymbol{\tau}}^3}{\sum_{\boldsymbol{\tau}\in\mathcal{S}_1}\pi_{\boldsymbol{\tau}} g_{\boldsymbol{\tau}}^1 + \sum_{\boldsymbol{\tau}\in\mathcal{S}_2}\pi_{\boldsymbol{\tau}} g_{\boldsymbol{\tau}}^2 + 2\sum_{\boldsymbol{\tau}\in\mathcal{S}_3}\pi_{\boldsymbol{\tau}} g_{\boldsymbol{\tau}}^3}$$

Optimal Solution

- Introduce auxiliary variables

$$x_{\boldsymbol{\tau}}^{k} = \pi_{\boldsymbol{\tau}} g_{\boldsymbol{\tau}}^{k}, k = 1, \cdots, 4, \boldsymbol{\tau} \in \mathcal{S} \Rightarrow \pi_{\boldsymbol{\tau}} = \sum\nolimits_{k=1}^{4} x_{\boldsymbol{\tau}}^{k}$$

- Transformed problem

$$\min_{\boldsymbol{x}, \eta} \quad \overline{T} = \frac{1}{\alpha} \sum_{\boldsymbol{\tau} \in \mathcal{S}} \sum_{k=1}^{4} i \cdot x_{\boldsymbol{\tau}}^{k} + \eta N + (1-\eta) t_c$$

$$\text{s.t.} \begin{cases} \nu_{loc}(\boldsymbol{x}) P_{loc} + \beta \nu_{tx}(\boldsymbol{x}) P_{tx} \le \overline{P}_{\max} \\ \Gamma(\boldsymbol{x}, \eta) = (1-\eta) \sum_{\boldsymbol{\tau} \in \mathcal{S}_1} x_{\boldsymbol{\tau}}^{1} - \eta \sum_{\boldsymbol{\tau} \in \mathcal{S}_2} x_{\boldsymbol{\tau}}^{2} + (1-2\eta) \sum_{\boldsymbol{\tau} \in \mathcal{S}_3} x_{\boldsymbol{\tau}}^{3} = 0 \\ F_{\boldsymbol{\tau}}(\boldsymbol{x}) = 0, \forall \boldsymbol{\tau} \in \mathcal{S} \\ \sum_{i=0}^{Q} \sum_{m=0}^{M} \sum_{n=0}^{N-1} \sum_{k=1}^{4} x_{(i,m,n)}^{k} = 1 \\ x_{(i,m,n)}^{k} \ge 0, \forall i, m, n, k \\ \eta \in [0, 1] \end{cases}$$

  - ❖ Reduce to a linear programming problem for a fixed η
  - ❖ η^* can be found via a 1-dimensional search

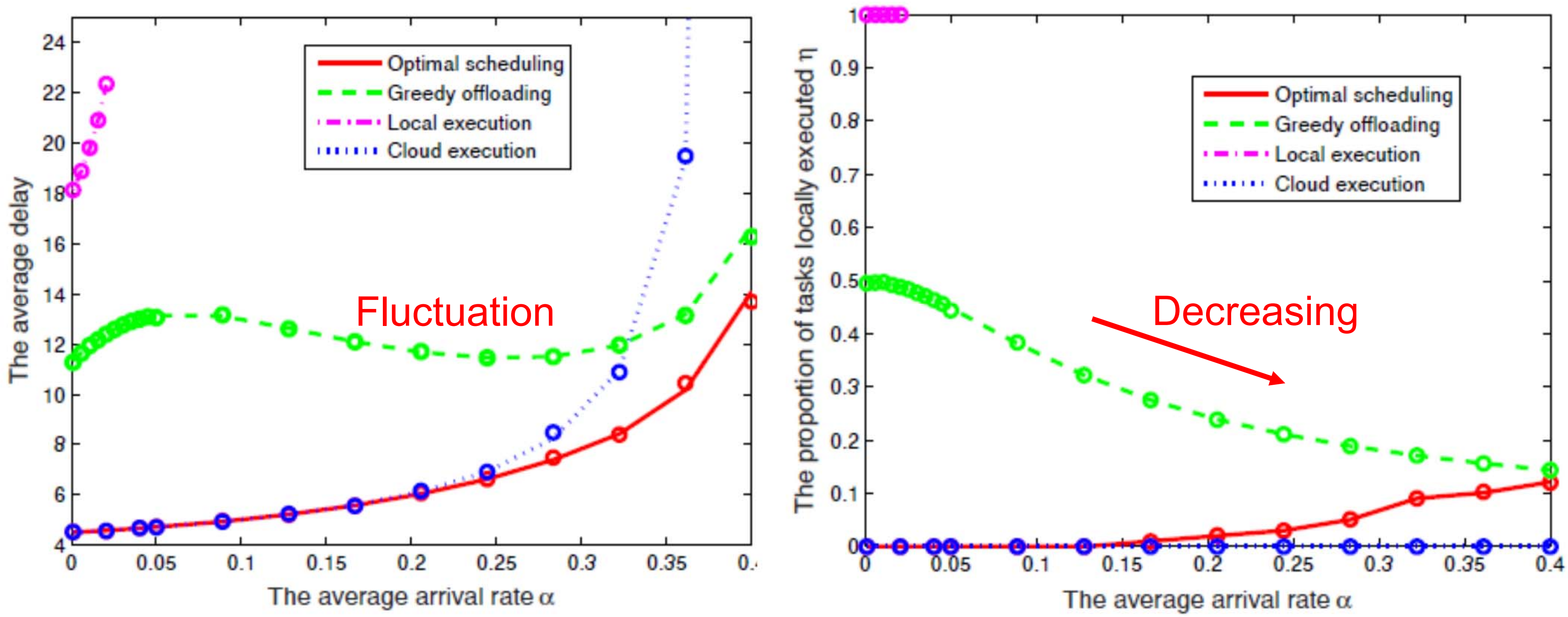
- Solution recovery

$$g_{\boldsymbol{\tau}}^{k*} = \frac{x_{\boldsymbol{\tau}}^{k*}}{\sum_{k=1}^{4} x_{\boldsymbol{\tau}}^{k*}}, \forall \boldsymbol{\tau} \in \mathcal{S}, k \in \{1, 2, 3, 4\}$$

Simulation Results

Greedy offloading: Schedule the waiting tasks to the local CPU and the Tx unit whenever they are idle

- Key parameters: N = 17, t_c = 3.5



❖ Behavior of the greedy offloading policy is greatly different

- Offloading is preferred as $N > t_c$
- $\alpha \uparrow$, queueing delay $\uparrow$ and execution delay $\downarrow$

Summary

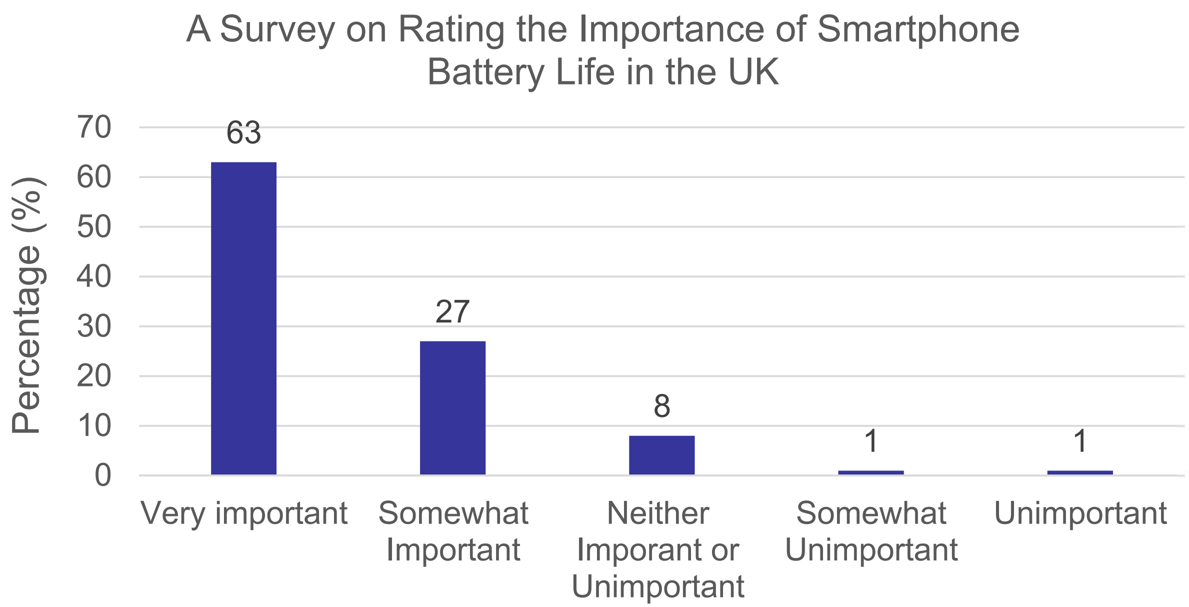
- The first work on two-timescale offloading
  - Stochastic task arrival
  - Multiple times slots for transmitting
- Very challenging problem
  - Greedy does not work
- Lots of room to follow
  - Consider more general MEC systems

MEC Meets Energy Harvesting

MEC With EH Devices (I)

Limitations

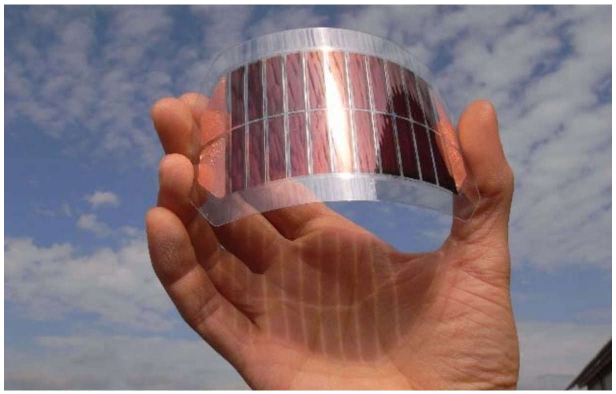
- MEC systems with battery-powered devices
  - Computation service interruption when battery energy runs out
- Battery lifetime
  - One of the most important features of smartphones

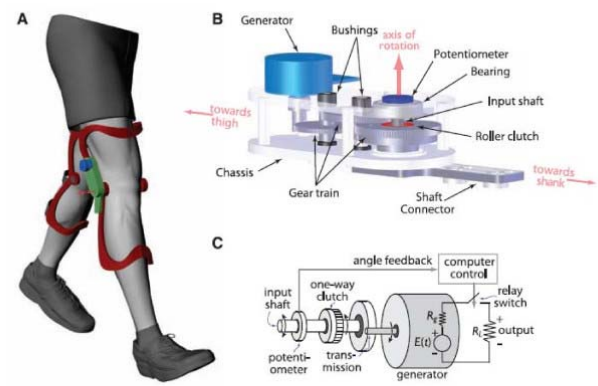


[Source: Statista]

MEC With EH Devices (II)

Solution: Energy harvesting (EH) mobile devices





- ❖ Potentially perpetual battery life
- ❖ Sustained and green computing

Challenges

- ❖ Intermittent availability of the renewable energy sources
  - Computation offloading policies should adapt to both **CSI** and **energy side information (ESI)**

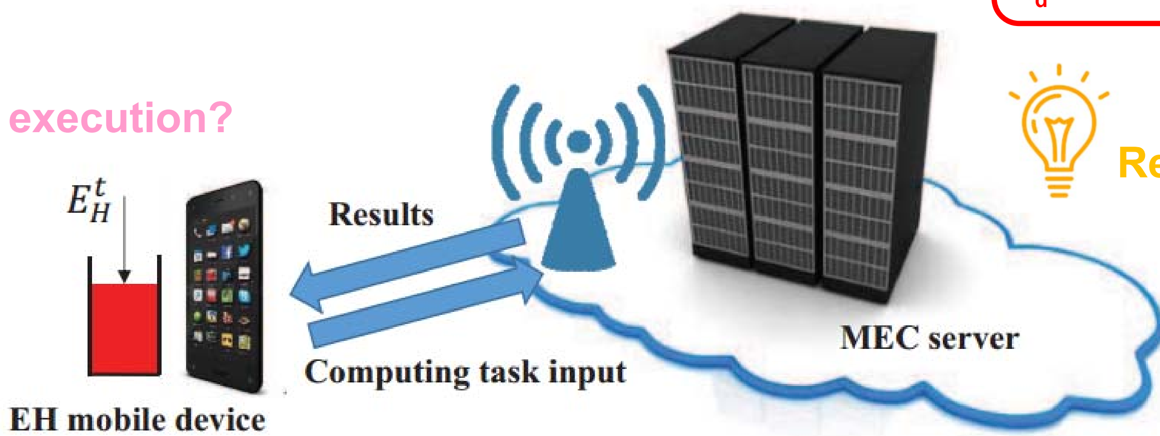
System Model

- MEC with EH devices

L: **Data size (bits)**
X: **Workload (cycles/bit) (*W* = *LX*)**
τ_d: Deadline

Local execution?

Remote execution?



- Task $A(L, X, \tau_d)$ arrives at each time slot with probability $\rho \in [0,1]$
- Harvestable energy $\{E_H^t\}$, block fading channel $\{h^t\}$
- Local execution: DVFS, $f^t \le f_{\mathrm{CPU}}^{\max}$
- Computation offloading: Tx power control, $p^t \le p_{\mathrm{tx}}^{\max}$
- Powerful MEC server, short computation result

DVFS: Dynamic voltage and frequency scaling

Problem Formulation

Execution cost minimization problem

- Offloading indicator: $\boldsymbol{I}^t = [I_{\rm m}^t, I_{\rm s}^t, I_{\rm d}^t]$ (local, remote, drop)

$$\mathcal{D}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) = \mathbf{1}(\zeta^t = 1) \cdot \left(I_{\rm m}^t \sum_{w=1}^{W} (f_w^t)^{-1} + I_{\rm s}^t \frac{L}{r(h^t, p^t)} \right)$$

$$\mathcal{E}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) = I_{\rm m}^t \sum_{w=1}^{W} \kappa (f_w^t)^2 + I_{\rm s}^t \frac{p^t L}{r(h^t, p^t)}$$

ϕ: Penalize the task drop events

- Execution cost: $\text{cost}^t = \mathcal{D}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) + \phi \cdot \mathbf{1}(\zeta^t = 1, I_{\rm d}^t = 1)$

$$\mathcal{P}_1: \min_{\mathbf{I}^t, \boldsymbol{f}^t, p^t, \mathbf{e}^t} \lim_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\text{cost}^t]$$

$$\begin{aligned}
\text{s.t.} \quad & I_{\rm m}^t + I_{\rm s}^t + I_{\rm d}^t = 1, I_{\rm m}^t, I_{\rm s}^t, I_{\rm d}^t \in \{0,1\}, t \in \mathcal{T} && \text{Operation} \\
& 0 \le e^t \le E_H^t, t \in \mathcal{T} \\
& \mathcal{E}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) \le B^t < +\infty, t \in \mathcal{T} && \text{Energy causality} \\
& \mathcal{D}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) \le \tau_d, t \in \mathcal{T} && \text{Execution latency} \\
& I_{\rm m}^t + I_{\rm s}^t \le \zeta^t, t \in \mathcal{T} \\
& \mathcal{E}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) \le E_{\max}, t \in \mathcal{T} && \text{Max. battery output energy} \\
& 0 \le p^t \le p_{\rm tx}^{\max} \cdot \mathbf{1}(I_{\rm s}^t = 1), t \in \mathcal{T} \\
& 0 \le f_w^t \le f_{\rm CPU}^{\max} \cdot \mathbf{1}(I_{\rm m}^t = 1), w = 1, \cdots, W, t \in \mathcal{T}
\end{aligned}$$

ζ^t: Task arrival indicator

$$B^{t+1} = B^t - \mathcal{E}(\boldsymbol{I}^t, \boldsymbol{f}^t, p^t) + e^t, t \in \mathcal{T}$$

A high-dimensional infinite horizon MDP problem

The LODCO Algorithm (I)

- Proposition (f_w^t is the frequency for the w-th cycle)
  - f_w^t's are identical for a computation task ($f_w^t = f^t, \forall w$)
- The LODCO algorithm - **L**yapunov **o**ptimization-based **d**ynamic **c**omputation **o**ffloading
  - Solve a deterministic problem at each time slot

$$\min_{\boldsymbol{I}^t, f^t, p^t, e^t} \left(B^t - \theta\right)\left[e^t - \mathcal{E}\left(\boldsymbol{I}^t, f^t, p^t\right)\right] + V \cdot \mathrm{cost}^t$$

An UB of the Lyapunov drift-plus-penalty

$$\text{s.t.} \quad \text{All constraints in } \mathcal{P}_1 \text{ except the energy causality constraint}$$

$$\mathcal{E}\left(\boldsymbol{I}^t, f^t, p^t\right) \in \{0\} \cup \left[E_{\min}, E_{\max}\right], t \in \mathcal{T}$$

- Control parameter, V ($J^2 \cdot sec^{-1}$)
- Perturbation parameter, $\theta = \tilde{E}_{\max} + V\phi \cdot E_{\min}^{-1}$
- Battery output energy non-zero lower bound, $E_{\min}$

The LODCO Algorithm (II)

- Solving the per-time slot problem
  - Optimal energy harvesting

$$e^{t*} = E_H^t \cdot \mathbf{1}\{\tilde{B}^t \leq 0\}$$

  - Optimal computation offloading decisions

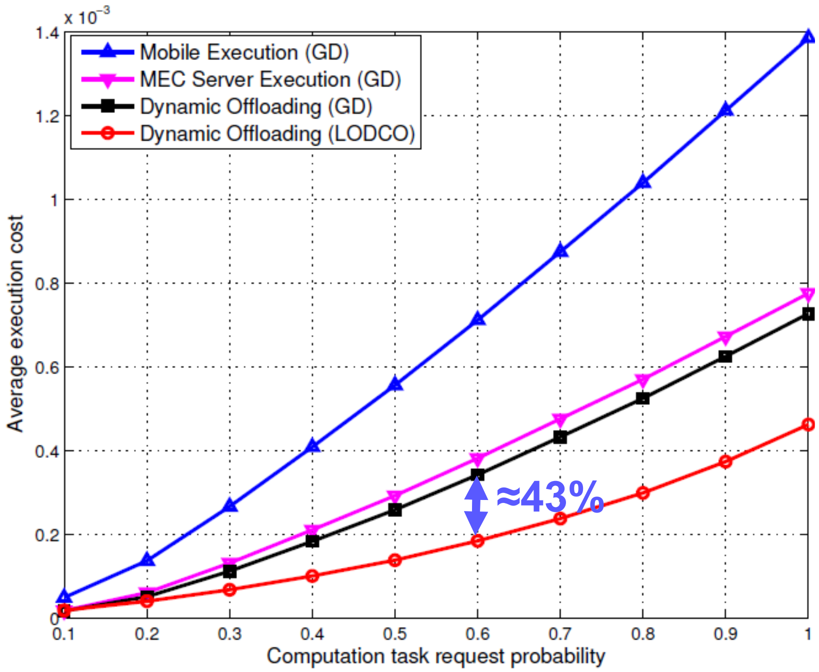
$$< \boldsymbol{I}^{t*}, f^{t*}, p^{t*} >= \arg \min_{<\boldsymbol{I}^t, f^t, p^t> \in \mathcal{F}_{\mathrm{CO}}^t} J_{\mathrm{CO}}^t \left(\boldsymbol{I}^t, f^t, p^t \right)$$

    - Evaluate the optimal values of the three computation modes
    - Semi-closed form solution is available

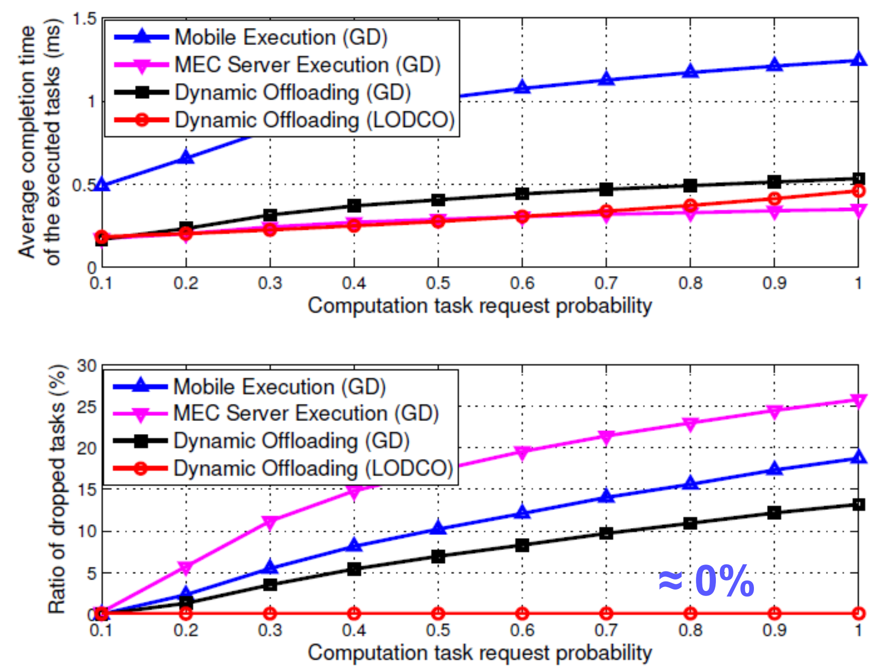
- Property of the LODCO algorithm
  - Satisfies the energy causality constraint
  - Achieves asymptotic optimality when $V \to +\infty, E_{\min} \to 0$

Simulation Results (I)

⌖ Performance evaluation



Execution cost vs. task arrival probability ρ

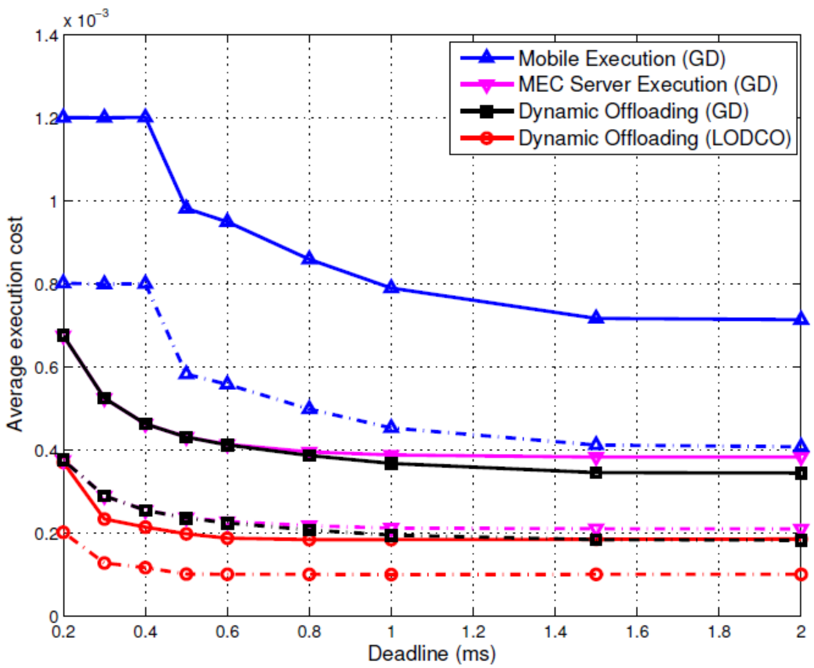


Average completion time/task drop radio vs. ρ

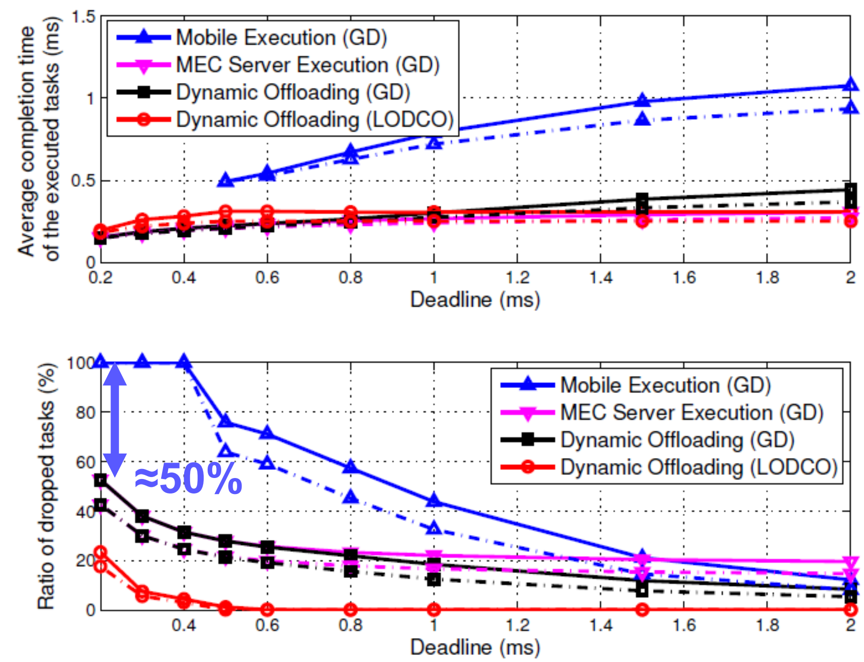
- ❖ Execution cost is greatly reduced by the LODCO algorithm
- ❖ Avoid task failures with minor delay performance degradation

Simulation Results (II)

Performance evaluation



Execution cost vs. deadline



Average completion time/task drop radio vs. deadline

- Benefits of MEC: ≈ 50% tasks can be executed even with the MEC server execution (GD) policy

Summary

- The first work on MEC with EH devices
  - Results showed such systems are promising
- Lyapunov optimization is a good tool
  - **L**yapunov **o**ptimization-based **d**ynamic **c**omputation **o**ffloading
- Extensions
  - More general MEC systems, e.g., multi-user and/or multi-server systems
  - Combine wireless power transfer with EH

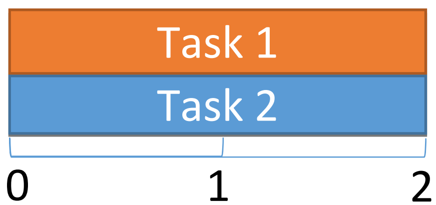
Joint Communication and Computational Resource Management

Motivation

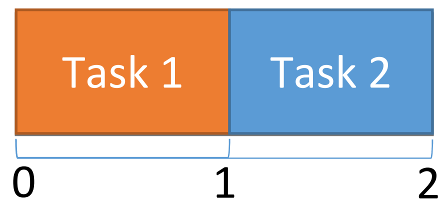
Limitations of previous works

- Idealized computation model of the MEC server
  - Infinite amount of computational resource
  - Constant execution time
- MEC server with limited computational resources
  - Frequency allocation

  1) May not be supported 2) Prolongs the execution time unnecessarily



Frequency Allocation: $t_1 = t_2 = 2$



Timeslot Allocation: $t_1 = 1$, $t_2 = 2$

Challenges

- Non-preemptive CPU scheduling is NP-hard [Jeffay'91]
- Nested with the offloading decision and radio resource allocation

System Model

- Single-cell OFDMA MEC systems
  - *M* users, each with task $(\mathbf{X}_i, \mathbf{D}_i, \mathbf{T}_i)$
  - Local execution $E_l^i = \kappa X_i^3 \frac{\mathbf{D}_i^3}{\mathbf{T}_i^2}, T_l^i = \mathbf{T}_i$
  - Remote execution
    - Uplink data rate

$$\mathbf{R}_i = B_N \sum_{j=1}^{N} \mathcal{W}_{i,j} \log_2\left(1 + \mathcal{P}(i,j)\,\mathcal{G}(i,j)\right)$$

    - Queuing and remote execution

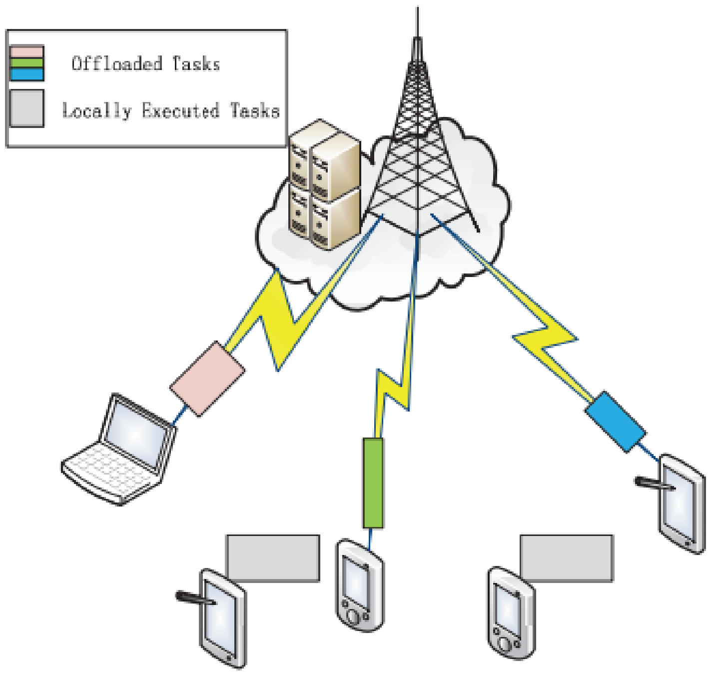
$$T_r^i = T_t^i + Q_c^i + T_c^i$$

Tx time Queuing time Remote execution time

$$Q_c^i = \sum_{j,\mathbf{q}_j<\mathbf{q}_i} \boldsymbol{\alpha}_j \cdot T_c^j$$

Execution sequence:
$\mathbf{q} = \{\mathbf{q}_i | \mathbf{q}_i \in \{1,2\ldots,M\}, \mathbf{q}_i \neq \mathbf{q}_j\}$

Offloading decision: $\boldsymbol{\alpha}_i$



Problem Formulation

- Total energy consumption minimization problem

$$\underset{\boldsymbol{\alpha},\mathcal{W},\mathcal{P},\mathbf{q}}{\text{minimize}} \quad \sum_{i=1}^{M} \left((1-\boldsymbol{\alpha}_i)\cdot E_l^i + \boldsymbol{\alpha}_i \cdot E_t^i \right)$$

subject to

$$\sum_{i=1}^{M} \mathcal{W}(i,j) \le 1, \quad \forall j \in \mathcal{C}$$

$$\mathbf{q}_i \ne \mathbf{q}_j, if\ i \ne j. \quad \forall i,\ j \in \mathcal{U}$$

Allocation

$$\mathbf{p}_i = \sum_{j=1}^{N} \mathcal{W}(i,j)\mathcal{P}(i,j) \le \mathbf{p}_i^m, \quad \forall i \in \mathcal{U}$$

Power constraint

$$\mathbf{R}_i = B_N \sum_{j=1}^{N} \mathcal{W}_{i,j} \log(1 + \mathcal{P}(i,j)\mathcal{G}(i,j)), \quad \forall i \in \mathcal{U}$$

$$E_l^i = \kappa X^3 \frac{\mathbf{D}_i^3}{\mathbf{T}_i^2}, \quad \forall i \in \mathcal{U}$$

$$E_r^i = \frac{\mathbf{D}_i(\mathbf{p}_i + \mathbf{p}_i^c)}{\mathbf{R}_i}, \quad \forall i \in \mathcal{U}$$

$$T_r^i = \frac{\mathbf{D}_i}{\mathbf{R}_i} + \sum_{j,\mathbf{q}_j < \mathbf{q}_i}^{M} \boldsymbol{\alpha}_j T_c^j + \boldsymbol{\alpha}_i T_c^i \le T_i, \quad \forall i \in \mathcal{U}.$$

Deadline requirement

$\mathcal{W}$: Subcarrier allocation
$\mathcal{P}$: Uplink power allocation

- NP-hard: Mixed-integer non-linear programming

Proposed Algorithms

- Case I: Negligible remote processing time
  - $\mathcal{P}, \boldsymbol{\alpha}$ can be easily determined once $\mathcal{W}$ is fixed
  - Minimum Set Allocation Algorithm
    - Main idea: Find the least number of subcarriers (minimum set) for each user that support its favorable offloading
    - The users that can save more energy have higher priorities

- Case II: Non-negligible remote processing time

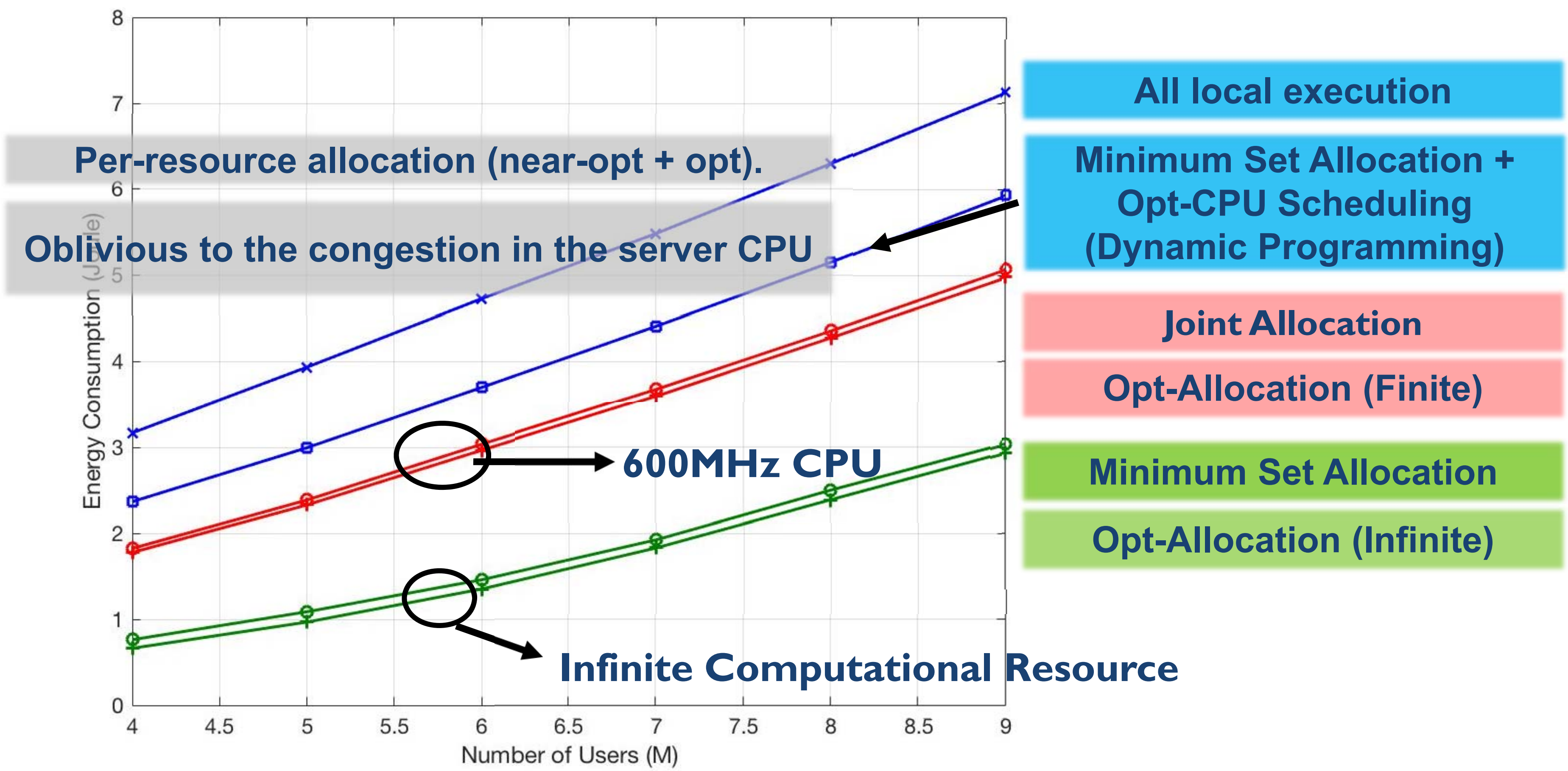
Joint Allocation Algorithm, O(M^2N)

1: Allocate the minimum set to the user who saves the *most energy with each unit of CPU time*, until the remaining subcarrier cannot support any user left to offload.

2: Allocate each of remaining subcarrier to the offloaded user gaining the *largest marginal energy saving* with it.

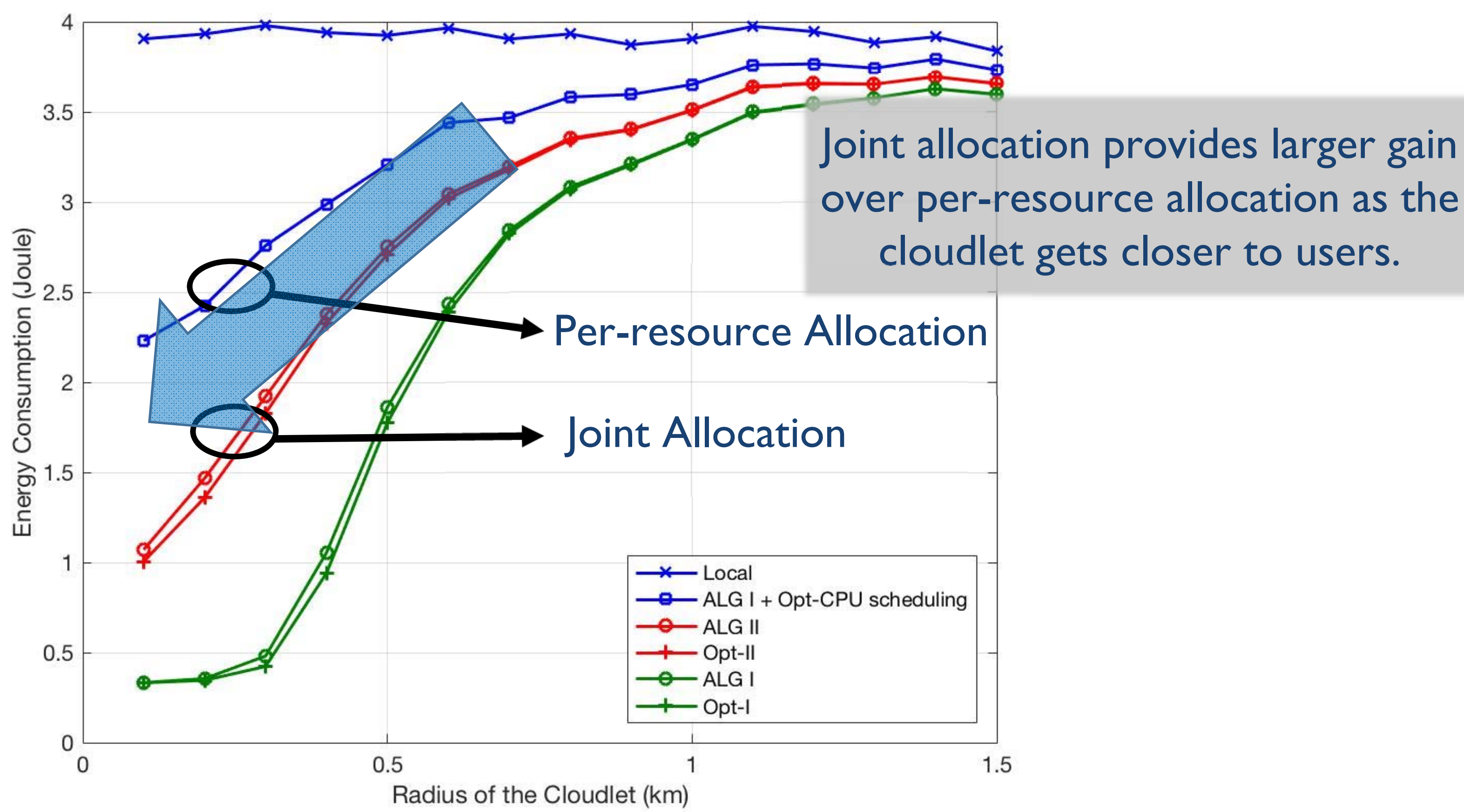
Simulation Results (I)

- Performance evaluation



Simulation Results (II)

- Coverage of the cloudlet



Summary

- Joint radio and computation resource management is necessary
  - Lower energy consumption
  - Better coverage of cloudlets
- Such problems are highly challenging
  - More efficient algorithms are needed
  - Difficult to extend to stochastic models

Stochastic Resource Management for MEC

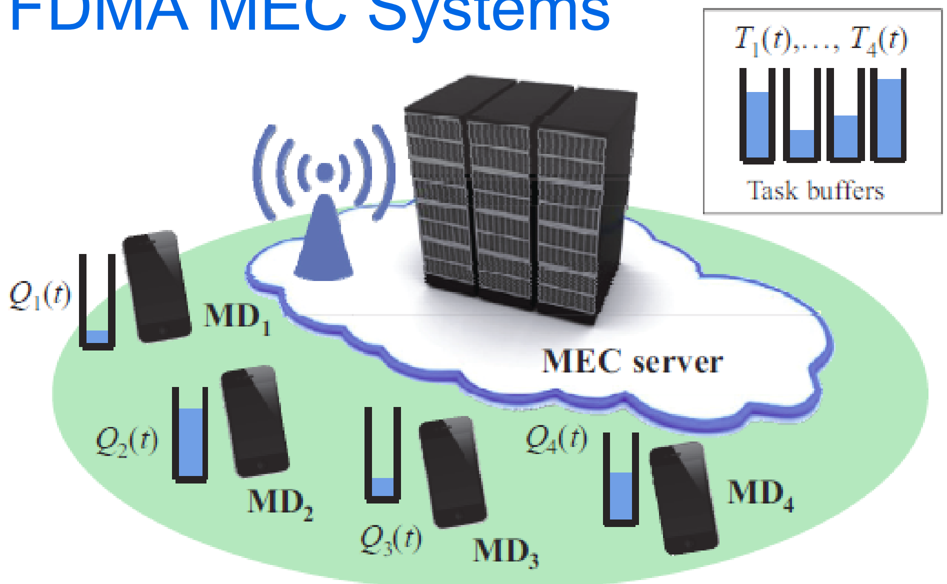
Motivation

- Limitations of previous works
  - Existing works mainly focus on delay-sensitive applications
  - Not applicable to delay-tolerant applications

- Challenges
  - Stochastic task models need to be incorporated
  - Temporal and spatial correlations on system operations
  - Joint management on both types of resources

System Model

- Multi-user FDMA MEC Systems



❖ Queuing model

- Mobile side: $Q_i(t+1) = (Q_i(t) - D_{\Sigma,i}(t))^+ + A_i(t)$ Task arrival (bits)
- Server side: $T_i(t+1) = (T_i(t) - D_{s,i}(t))^+ + \min\{(Q_i(t) - D_{l,i}(t))^+, D_{r,i}(t)\}$

$\propto f_{l,i}(t)$

Power-rate function CSI $\Gamma_i(t)$

❖ Mobile/server CPU speeds, $f_{l,i}(t)/f_{C,m}(t)$

❖ MEC scheduling decision, $D_{s,i}(t)$

❖ Transmit power and bandwidth allocation, $p_{\mathrm{tx},i}(t)$ and $\alpha_i(t)$

Problem Formulation

- Average weighted sum power consumption minimization

$$\mathcal{P}_2: \min_{\{\mathrm{X}(t)\}} \lim_{T\to+\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i\in\mathcal{N}} w_i\left(p_{\mathrm{tx},i}(t)+p_{l,i}(t)\right)+w_{N+1}p_{\mathrm{ser}}(t)\right]$$

$$p_{l,i}(t)=\kappa_{\mathrm{mob},k}f_{l,i}^3(t) \qquad p_{\mathrm{ser}}(t)=\sum_{m\in\mathcal{M}}\kappa_{\mathrm{ser},m}f_{C,m}^3(t)$$

s.t

$$0\le f_{l,i}(t)\le f_{i,\max}, i\in\mathcal{N}, t\in\mathcal{T}$$
$$0\le f_{C,m}(t)\le f_{C_m,\max}, m\in\mathcal{M}, t\in\mathcal{T}$$

CPU speed constraints

$$0\le p_{\mathrm{tx},i}(t)\le p_{i,\max}, i\in\mathcal{N}, t\in\mathcal{T}$$
$$\boldsymbol{\alpha}(t)\in\mathcal{A}, t\in\mathcal{T}$$

Tx power and bandwidth allocation constraints

$$\mathcal{A}=\{\boldsymbol{\alpha}|\alpha_i\ge\epsilon_A, \textstyle\sum_{i\in\mathcal{N}}\alpha_i\le 1\}, \epsilon_A\searrow 0^+$$

$$\sum_{i\in\mathcal{N}} D_{s,n}(t)L_n\le\sum_{m\in\mathcal{M}} f_{C,m}(t)\tau, t\in\mathcal{T}$$
$$D_{s,i}(t)\ge 0, i\in\mathcal{N}, t\in\mathcal{T}$$

Server scheduling constraints

$$\lim_{T\to+\infty}\frac{\mathbb{E}\left[|Q_i(T)|\right]}{T}=0, i\in\mathcal{N}$$
$$\lim_{T\to+\infty}\frac{\mathbb{E}\left[|T_i(T)|\right]}{T}=0, i\in\mathcal{N}$$

Mean rate stability

A challenging stochastic optimization problem!

Proposed Solution (I)

- Challenges
  - Large amount of side information to be handled
  - Optimal decisions are temporally and spatially correlated
  - Joint radio and computational resource management

- Online resource management algorithm
  - Solve a deterministic optimization problem at each time slot

$$\min_{\mathrm{X}(t)} \; -\sum_{i\in\mathcal{N}} Q_i(t)\, D_{\Sigma,i}(t) - \sum_{i\in\mathcal{N}} T_i(t)\left(D_{s,i}(t) - D_{r,i}(t)\right) + V\cdot P_{\Sigma}(t)$$

$$\text{s.t} \quad \text{All constraints in } \mathcal{P}_2 \text{ except the stability constraints}$$

An UB of the Lyapunov drift-plus-penalty

- Control parameter: V (bits·W^{-1})
- Decomposable into 3 sub-problems

Proposed Solution (II)

- Optimal solution at each time slot
  - Optimal local CPU speed

$$f_i^\star(t) = \begin{cases} \min\{f_{i,\max}, \sqrt{\frac{Q_i(t)\tau}{3\kappa_{\mathrm{mob},i} w_i V L_i}}\}, & w_i > 0 \\ f_{i,\max}, & w_i = 0 \end{cases}, i \in \mathcal{N}$$

  - Optimal transmit power and BW allocation
    - Device offloads only when $Q_i(t) > T_i(t)$
    - Optimal solution for devices in $\tilde{\mathcal{N}}^{\mathrm{c}}(t)$ based on the **G-S method**
  - Optimal server CPU speed and scheduling decision
    - The device ($i_{\mathcal{N}}^{\max}$) with highest value of $T_i(t)/L_i$ will be served

$$f_{C,m}^\star(t) = \begin{cases} \min\{f_{C_m,\max}, \sqrt{\frac{T_{i_{\mathcal{N}}^{\max}}(t)\tau}{3\kappa_{\mathrm{ser},m} w_{N+1} V L_{i_{\mathcal{N}}^{\max}}}}\}, & w_{N+1} > 0 \\ f_{C_m,\max}, & w_{N+1} = 0 \end{cases}, m \in \mathcal{M}$$

- Delay-improved mechanism
  - Based on $\mathbf{X}^\star(t)$, modify $\mathbf{D}_s^\star(t)$ whenever $D_{s,i_{\mathcal{N}}^{\max}}^\star(t) > T_{i_{\mathcal{N}}^{\max},act}(t)$

Proposed Solution (III)

- Performance analysis
  - The average weighted sum power consumption satisfies

$$\overline{P}_{\Sigma}^{\star} \leq P_{\Sigma,\mathcal{P}_2}^{\mathrm{opt}} + \frac{C}{V}$$

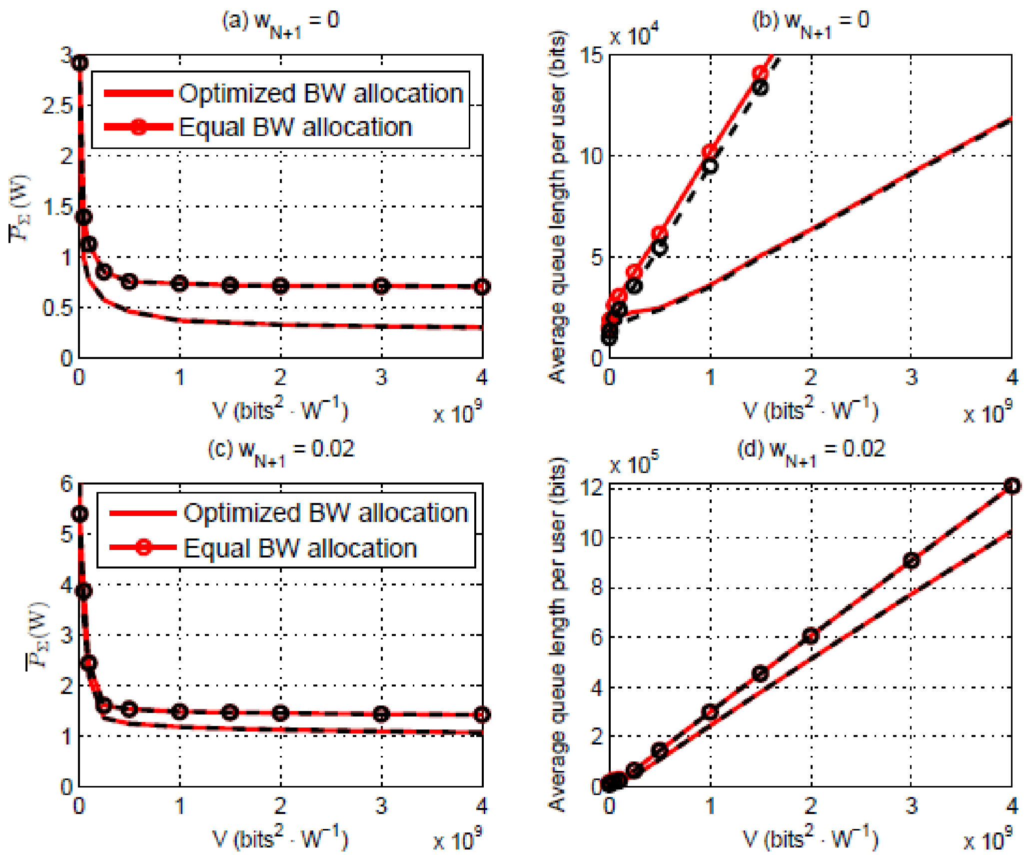
  - All queues are mean rate stable
  - Average sum queue length of the task buffer satisfies

$$\lim_{T\to+\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i\in\mathcal{N}} \left(Q_i\left(t\right) + T_i\left(t\right)\right)\right] \leq \frac{C + V \cdot \left(\Psi\left(\epsilon\right) - P_{\Sigma,\mathcal{P}_2}^{\mathrm{opt}}\right)}{\epsilon}$$

Power-delay tradeoff in multi-user MEC systems: [O(1/V), O(V)]

Simulation Results (I)

- Benchmark: Equal bandwidth allocation



$N = 5$, $\lambda_i = 4$ kbits/slot

Verify the [O(1/V), O(V)] power-delay tradeoff

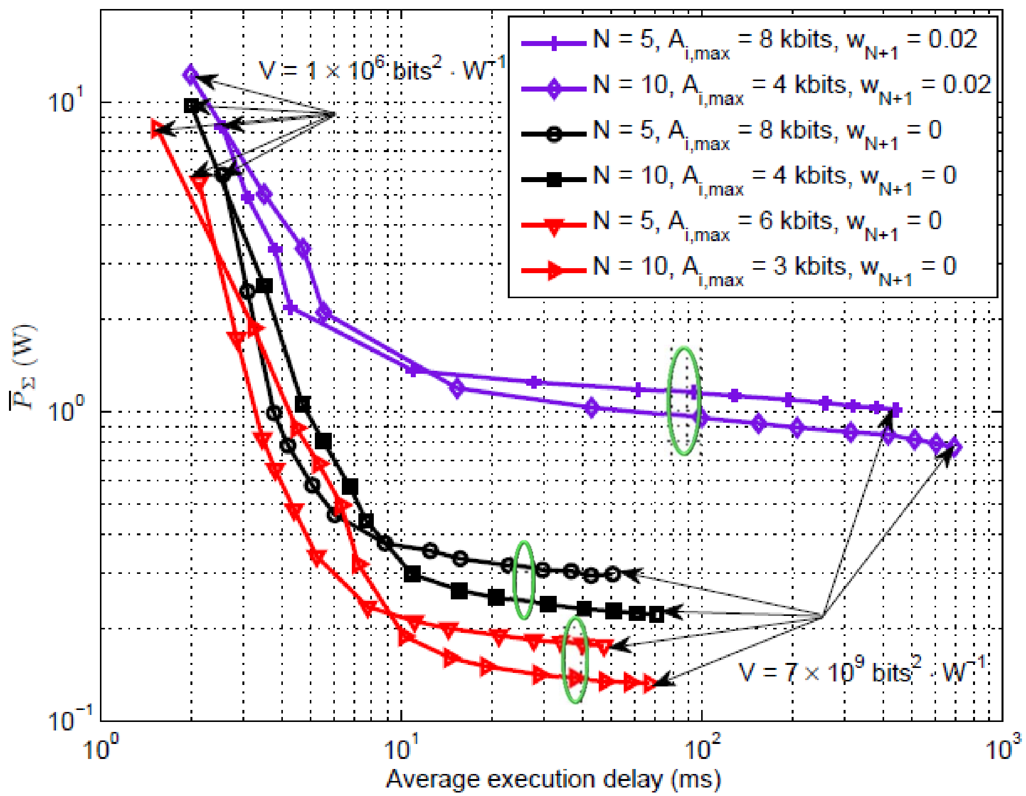
Benefits of joint resource management on power and delay performance for MEC

The delay-improved mechanism enhances the delay performance without extra power consumption

Performance of the delay-improved mechanism is shown by the dash curves.

Simulation Results (II)

- Performance evaluation



Increased MU diversity gain

Availability of extra local CPUs

❖ Number of devices ↑ & task arrival rate at each device ↓ leads to lower average weighted sum power consumption

Summary

- Joint radio and computation resource management is necessary
- Lyapunov optimization provides low-complexity online algorithm
  - Sub-problems require special efforts
  - Theoretical performance guarantee
  - Power-delay tradeoff
- Extensions
  - Fairness consideration among users
  - Distributed implementation

Key Takeaways

- Resource management for MEC
  - Joint management on radio and computational resource
  - Essential to incorporate the CSI and task characteristics
  - Stochastic models are important
  - Efficient and effective algorithms

- Interesting research directions
  - Mobility-aware resource management for MEC
  - Server cooperation in MEC
  - Dependency-aware offloading in MEC
  - MEC with coded distributed computing
  - …

References

- J. Liu, Y. Mao, **J. Zhang**, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Barcelona, Spain, Jul. 2016.
- Y. Mao, **J. Zhang**, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016.
- Y. Yu, **J. Zhang**, and K. B. Letaief, “Joint subcarrier and CPU time allocation for mobile edge computing,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016.
- Y. Mao, **J. Zhang**, S. H. Song, and K. B. Letaief, “Stochastic joint radio and computational resource management for mobile-edge computing systems,” *IEEE Trans. Wireless Commun.*, to appear.
- Y. Mao, **J. Zhang**, and K. B. Letaief, “Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems,” in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017.

Thank you!