“AI based Network Resource Management (2)”

Choong Seon Hong

Department of Computer Science and Engineering
Kyung Hee University, Republic of Korea.

KYUNG HEE UNIVERSITY

NETWORKING INTELLIGENT LAB

Outline

- Network Resource Management
- Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing
- Game Theory Approaches
- AI Based Approaches
- Conclusions

Network Resource Management

- Network Resources
  - Communication
  - Computation
  - Caching (Storage), etc…

- Lecture 1
  - Network slicing concept
  - Resource allocation with optimization (Network Slicing)

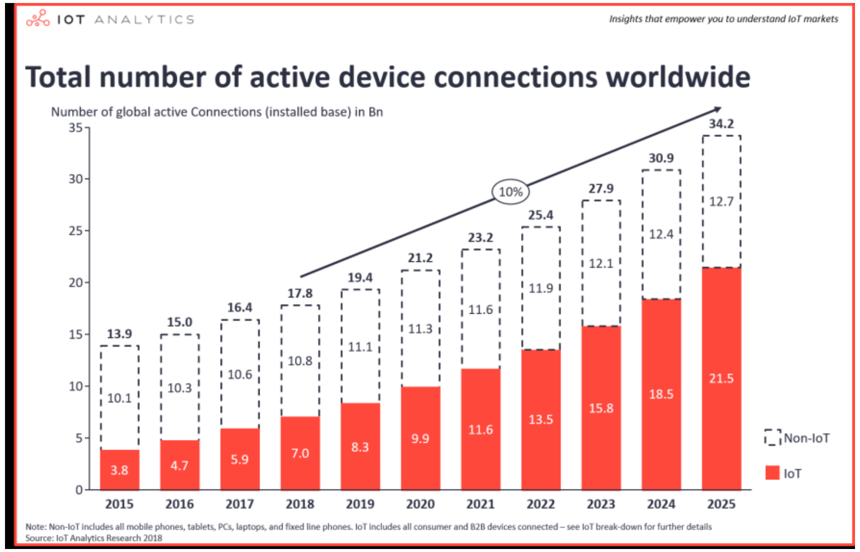
- **Lecture 2**
  - Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing
  - Game Theory Approaches
  - AI/ML Based Approaches

Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing

- Introduction
- Joint 4C in Big Data MEC
- Performance Evaluation

Introduction: Background

- **Global number of connected devices continue to increase at very rapid pace:**
  By year 2025, there will be 34.2 B with 21.5 B IoT devices (smartphones, tablets, laptops)



Source: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

- **Global mobile data traffic:**
  By year 2022, there will be 77 Exabytes per month of mobile data traffic

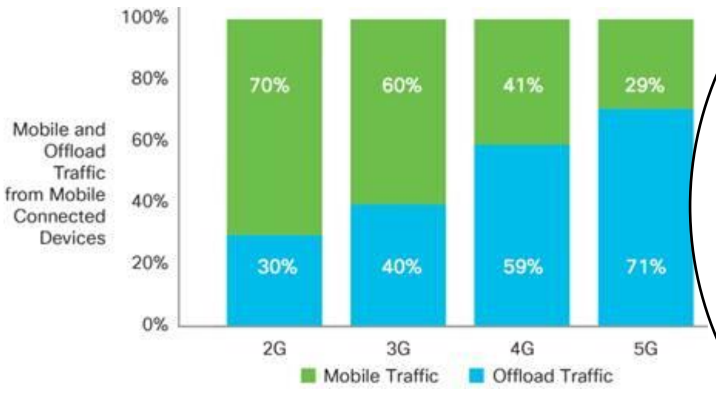
Therefore, wireless users' devices will be anywhere, anytime, and connected to anything

Source: Cisco VNI Mobile, 2019 (https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visu networking-index-vni/white-paper-c11-738429.html)

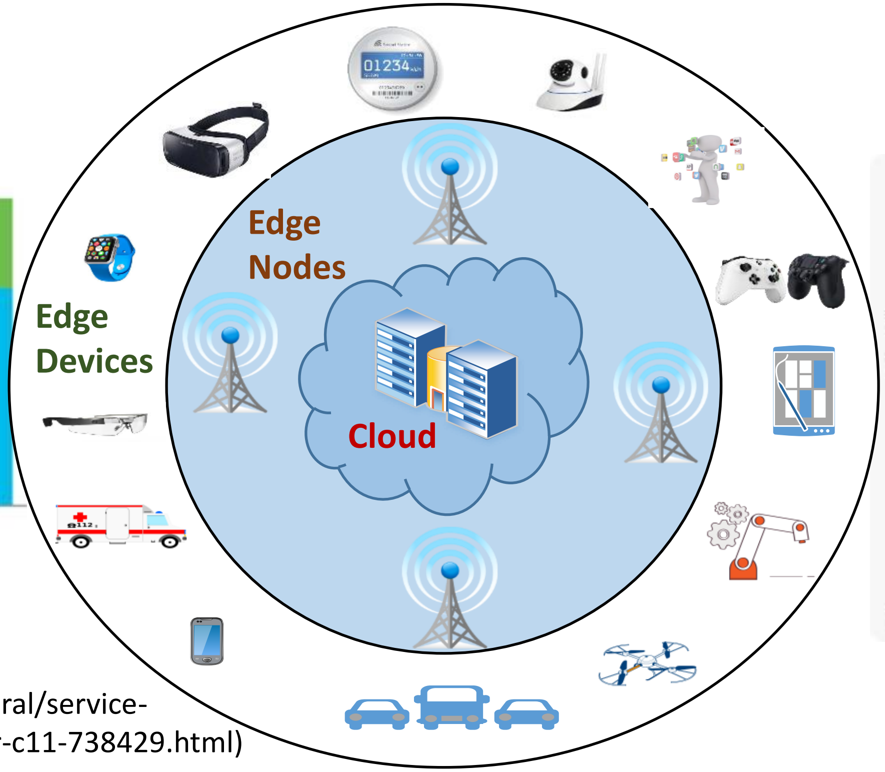
Introduction: Background

- Offloading: Offloading traffic will be 71% of mobile data traffic by year 2022
- Downloading: Data traffic from data centers to users will be 14.9% of global data center traffic by year 2021

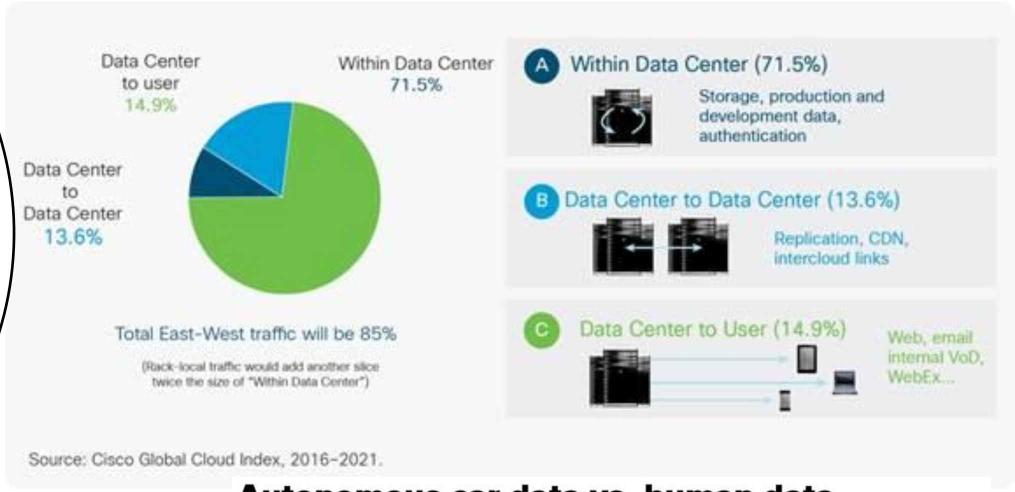
Offloading

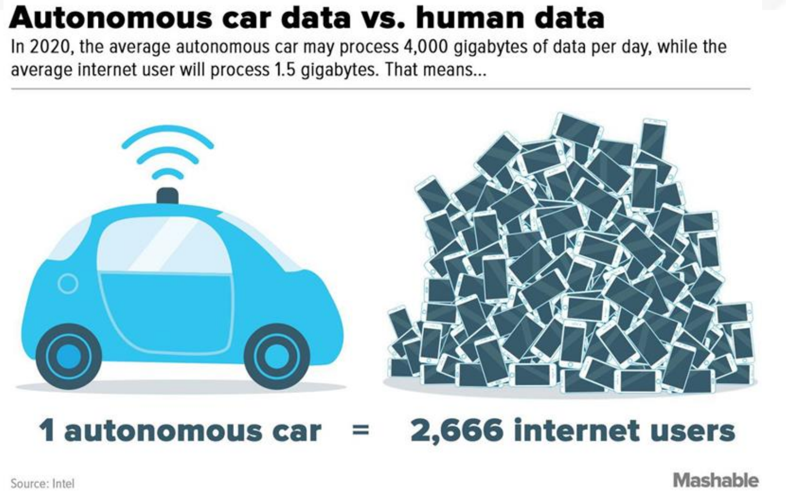


Source: Cisco VNI Mobile, 2019 (https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html)



Downloading



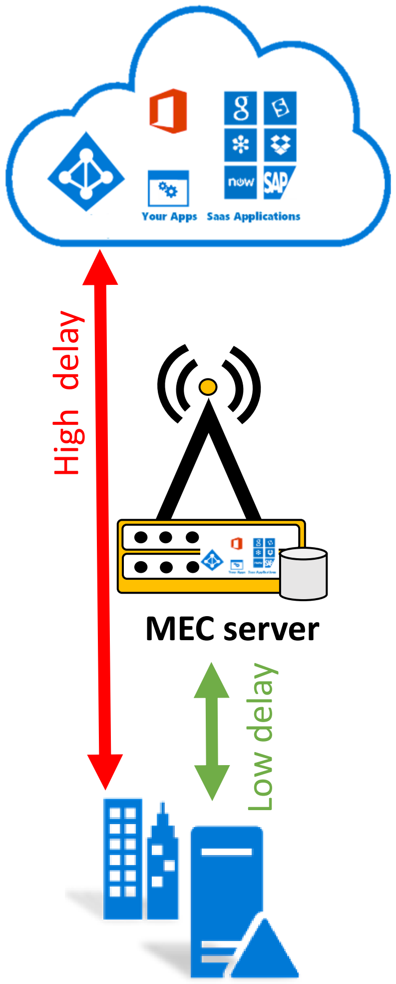


From the edge, there will be a tremendous growth of data traffic with different scale, distribution, diversity, and velocity fall into a big data framework

KYUNG HEE UNIVERSITY

Multi-access Edge Computing (MEC)

ETSI introduced Multi-access Edge Computing (MEC) as a suitable technology for providing cloud services to the edges in closed proximity to the users [1]

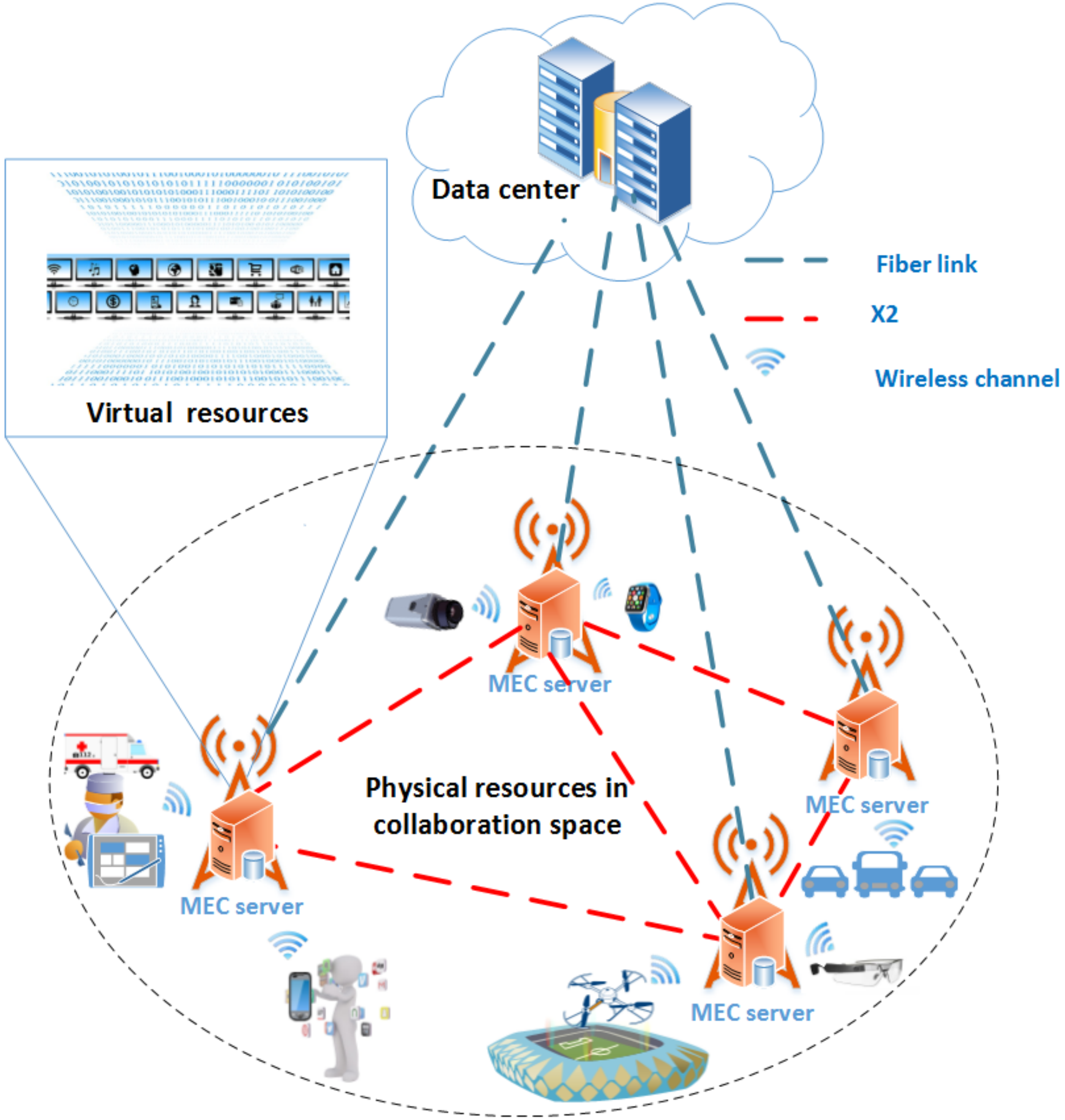


Challenges:

- However, when each MEC server operates independently, it cannot handle all computational and big data demands stemming from edge devices.
  How significantly reduce data exchange between edge devices and cloud?
- Edge devices offload tasks and corresponding data with varying rates, where data from multiple edge devices may reach MEC servers too rapidly with a finite or infinite flow, and needs to be processed immediately.
  How to handle such data for delay sensitive and mission critical applications?
- Integration of MEC with a mobile network environments raises a number of challenges related to the coordination of both MEC server and mobile network services.
  How to formulate a joint communication, computation, and caching for MEC?

[1]. MEC in 5G networks," ETSI White Paper No. 28, ISBN No. 979-10-92620-22-1,Jun. 2018.

Joint 4C in Big Data MEC



Solution: Collaboration space for Big Data MEC [1]

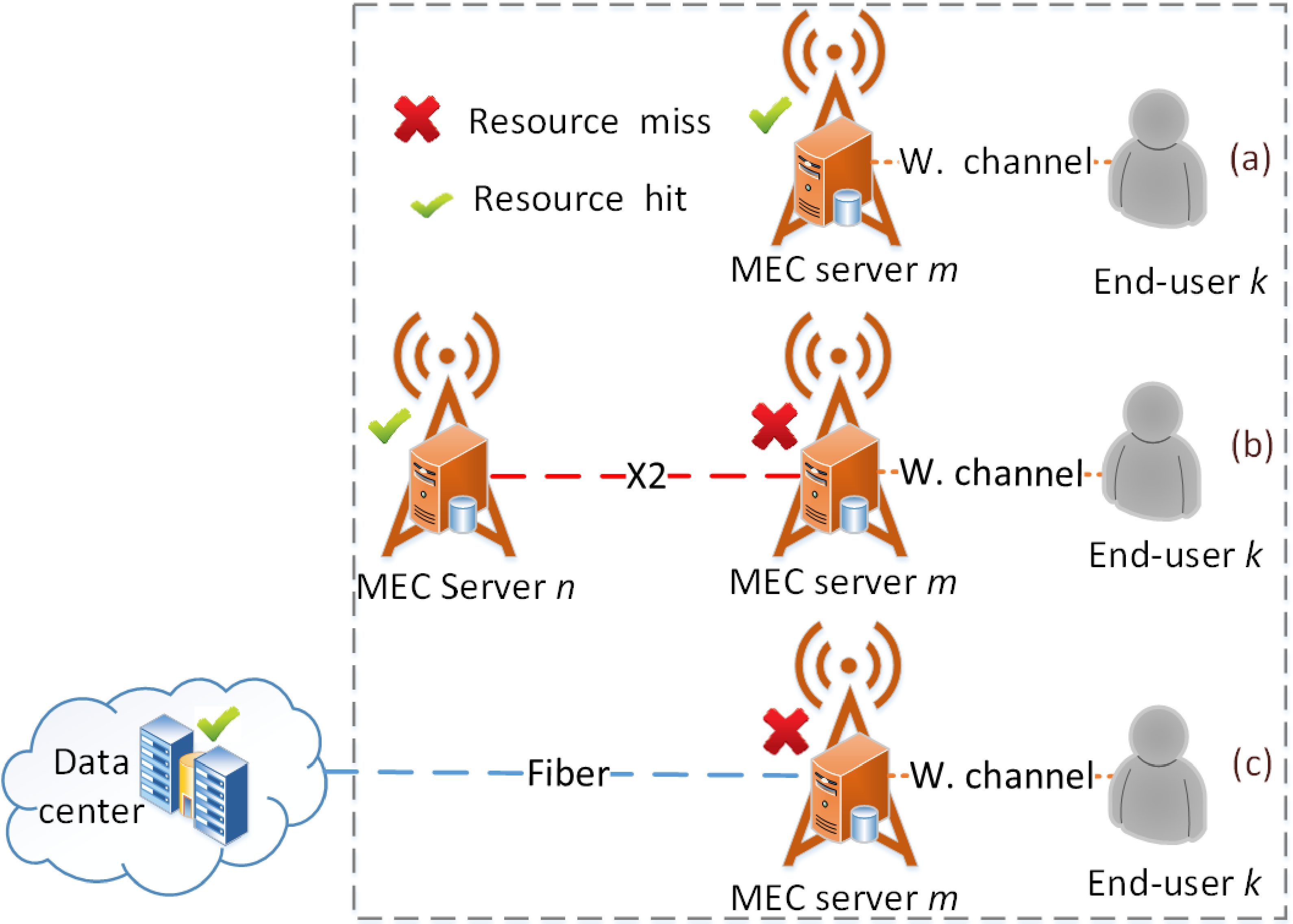
We propose joint computing, caching, communication, and control (4C) at the edge with MEC server collaboration for Big Data applications

KYUNG HEE UNIVERSITY

Anselme Ndikumana, Nguyen H. Tran, Tai Manh Ho, Zhu Han, Walid Saad, Dusit Niyato, Choong Seon Hong , "Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing," IEEE Transactions on Mobile Computing, Vol.19, Issue 6, pp.1359-1374, Jun. 2020

NETWORKING INTELLIGENT LAB

Joint 4C in Big Data MEC (continued)



Collaboration space

We introduce overlapping k-mean method for collaboration space (OKM-CS) in MEC that enables collaboration among MEC servers, which is not only based on distance measurements, but also based on available resources

Anselme Ndikumana, Nguyen H. Tran, Tai Manh Ho, Zhu Han, Walid Saad, Dusit Niyato, Choong Seon Hong , "Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing," IEEE Transactions on Mobile Computing, Vol.19, Issue 6, pp.1359-1374, Jun. 2020

Joint 4C in Big Data MEC (continued)

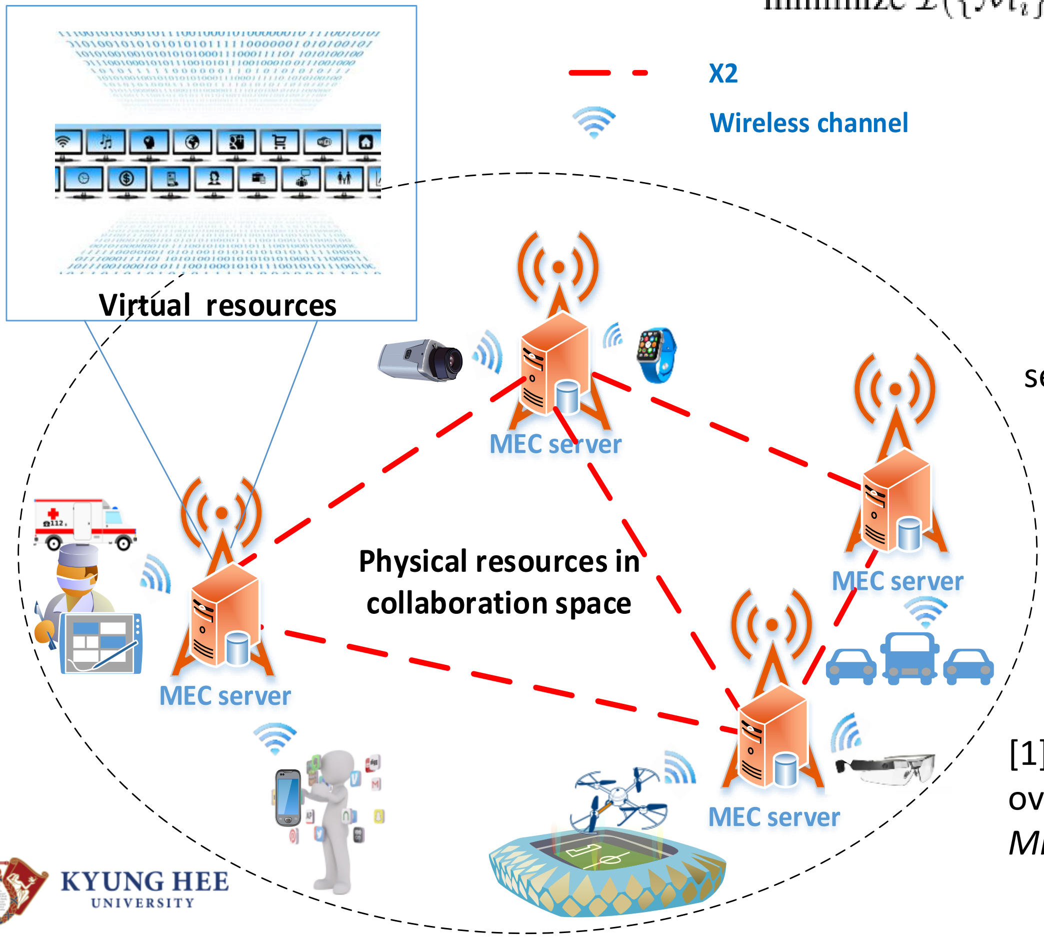
- We formulate the joint 4C in big data MEC as an optimization problem in [1] that aims at maximizing bandwidth saving while minimizing delay, subject to the local computation capabilities of user devices, and MEC resource constraints

- In order to solve the formulated problem, which is non-convex, we propose a proximal upper-bound problem of the original problem and apply the block successive upper bound minimization (BSUM) [2] for solving it.

1. Anselme Ndikumana, Nguyen H. Tran, Tai Manh Ho, Zhu Han, Walid Saad, Dusit Niyato, Choong Seon Hong , "Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing," IEEE Transactions on Mobile Computing, Vol.19, Issue 6, pp.1359-1374, Jun. 2020
2. M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, “A unified algorithmic framework for block-structured optimization involving big datag,” IEEE Signal Processing Magazine, vol. 33, no. 1, pp. 57–77, 25 Dec. 2015
3. O. Boswarva et al., “Sitefinder mobile phone base station database,” Edinburgh DataShare , the University of Edinburgh, UK , Feb. 2017.

Joint 4C Collaborative Big Data MEC

- ✓ In order to satisfy edge devices' demands, MEC servers located in the same area need to collaborate
- ✓ We proposed collaboration space formation by using **<u>Overlapping k-Means</u>** [1]
- ✓ In each collaboration space, based on available resources, MEC servers can exchange data, tasks, resource utilization information



Average of centroids

$$\text{minimize } \mathcal{I}(\{\mathcal{M}_i\}_{i=1}^{r}) = \sum_{i=1}^{r} \sum_{m \in \mathcal{M}_i} \|\boxed{m} - \boxed{\Phi(m)}\|^2$$

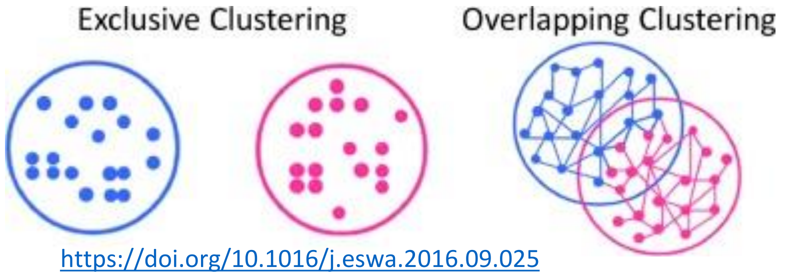
Base station location

$$\Phi(m) = \frac{\sum_{m_{c_i} \in \mathcal{A}_i^m} \boxed{m_{c_i}}}{\boxed{|\mathcal{A}_i^m|}},$$

Centroids of i

set of all centroids m_c

m: Base station
r: Number of collaboration spaces
$\Phi(m)$: Average of centroids



https://doi.org/10.1016/j.eswa.2016.09.025

Algorithm 1. OKM for Collaboration Space (OKM-CS)

1: **Input:** $\mathcal{M}$: A set of BSs with their coordinates, t_m: Maximum number of iterations, $\epsilon > 0$;
2: **Output:** $\{\mathcal{M}_i^{(t+1)}\}_{i=1}^r$: Final cluster coverage of BSs;
3: Choose r and initial clusters with $\{m_{c_i}^{(0)}\}_{i=1}^r$ centroid;
4: For each BS m, compute the assignment $\mathcal{A}_i^{m(0)}$ by assigning bs m to centroid $\{m_{c_i}^{(0)}\}_{i=1}^r$, and derive initial coverage $\{\mathcal{M}_i^{(0)}\}_{i=1}^r$, such that $\mathcal{M}_i^{(0)} = \{m | m_{c_i}^{(0)} \in \mathcal{A}_i^{m(0)}\}$;
5: Initialize $t = 0$;
6: For each cluster $\mathcal{M}_i^{(t)}$, compute the new centroid, $m_{c_i}^{(t+1)}$ by grouping $\mathcal{M}_i^{(t)}$;
7: For each BS m and assignment $\mathcal{A}_i^{m(t)}$, compute new assignment $\mathcal{A}_i^{m(t+1)}$ by assigning bs m to centroid $\{m_{c_i}^{(t+1)}\}_{i=1}^r$ and derive new coverage $\{\mathcal{M}_i^{(t+1)}\}_{i=1}^r$;
8: If Equation (1) does not converge or $t_m > t$ or $\mathcal{I}(\{\mathcal{M}_i^{(t)}\}_{i=1}^r) - \mathcal{I}(\{\mathcal{M}_i^{(t+1)}\}_{i=1}^r) > \epsilon$, set $t = t + 1$, restart from Step 6. Otherwise, stop and consider $\{\mathcal{M}_i^{(t+1)}\}_{i=1}^r$ as the final clusters.

[1]. Whang, Joyce Jiyoung, Inderjit S. Dhillon, and David F. Gleich. "Non-exhaustive, overlapping k-means." *Proceedings of the 2015 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2015.

Joint 4C Model for Collaborative Big Data MEC

Communication Model

To offload task and data from edge device to the MEC server, the network will incur a communication cost

Scenario (a)

- Offload a task from edge device to the nearest MEC server

$$x_k^m = \begin{cases} 1, & \text{if task } T_k \text{ from edge device } k \text{ is offloaded to BS } m, \\ 0, & \text{otherwise.} \end{cases}$$

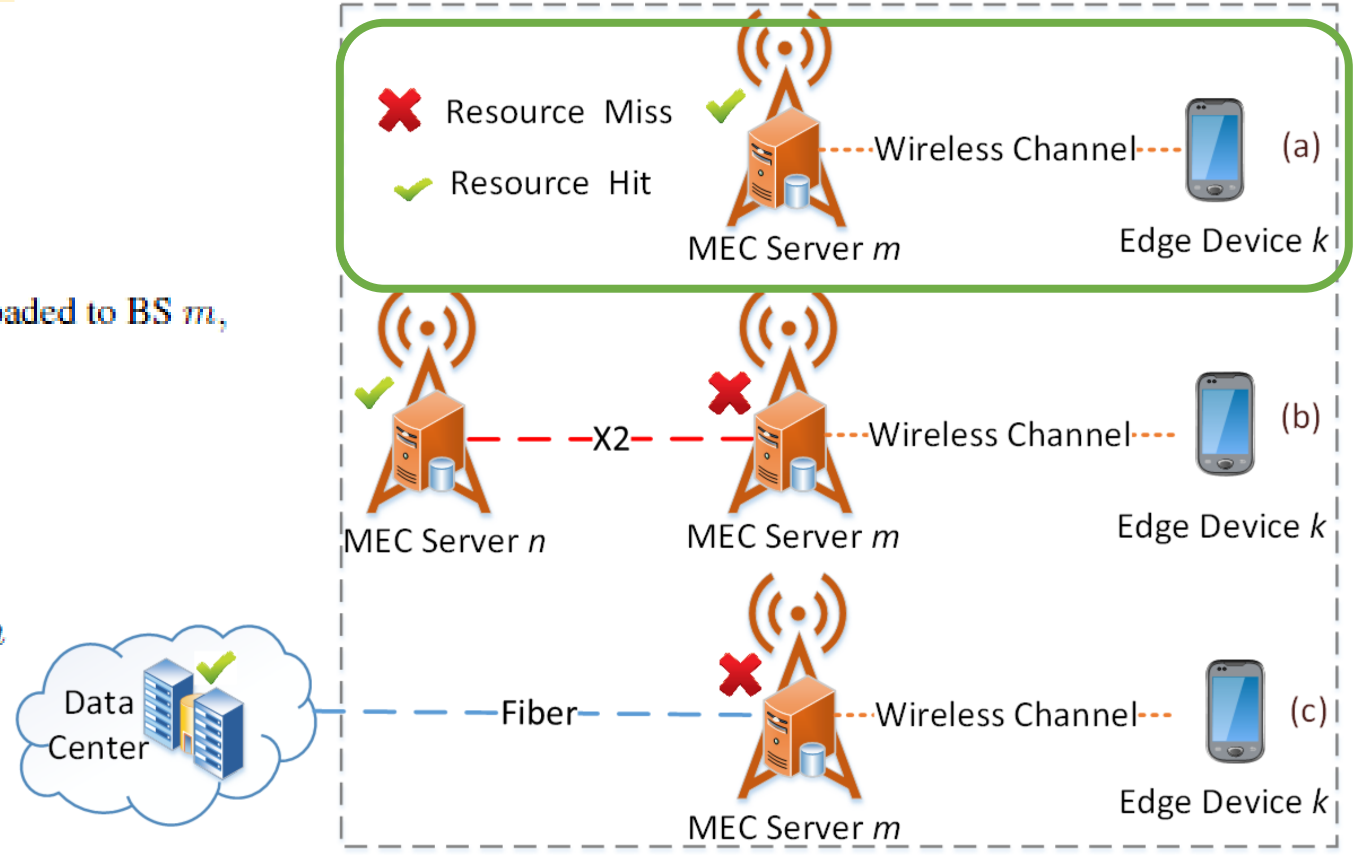
- The spectrum efficiency and instantaneous data rate:

$$\gamma_k^m = \log_2\left(1 + \frac{\rho_k |G_k^m|^2}{\sigma_k^2}\right), \ \forall k \in \mathcal{K},\ m$$

Instantaneous data rate $R_k^m = x_k^m \underbrace{\boxed{a_k^m}}_{\text{allocation}} \underbrace{\boxed{B_m}}_{\text{bandwidth}} \gamma_k^m, \forall k \in \mathcal{K},\ m \in \mathcal{M},$

- Transmission delay for offloading a task

$$\tau_k^{k \to m} = \frac{x_k^m s(d_k)}{R_k^m}, \ \forall k \in \mathcal{K}_m$$



Collaboration space for MEC with three typical scenarios

$\lvert G_k^m \rvert^2$	: Channel gain
ρ_k	: Transmission power
T_k	:Task
$s(d_k)$	: Size of input data

Communication Model

To offload task and data from a user to the MEC server, the network will incur a communication cost

Scenario (b)

- When the nearest MEC server m has insufficient resources, it forwards a request to another BS n

$$y_k^{m\to n} = \begin{cases} 1, & \text{if an offloaded task } T_k \text{ from edge device } k \text{ is forwarded from BS } m \\ & \text{to a nearest neighbor BS } n, \\ 0, & \text{otherwise.} \end{cases}$$

- The offloading delay between BS m and BS n

$$\tau_k^{m\to n} = \frac{\sum_{k\in\mathcal{K}_m} y_k^{m\to n} s(d_k)}{\Gamma_m^n}, \ \forall m,\ n \in \mathcal{M}$$

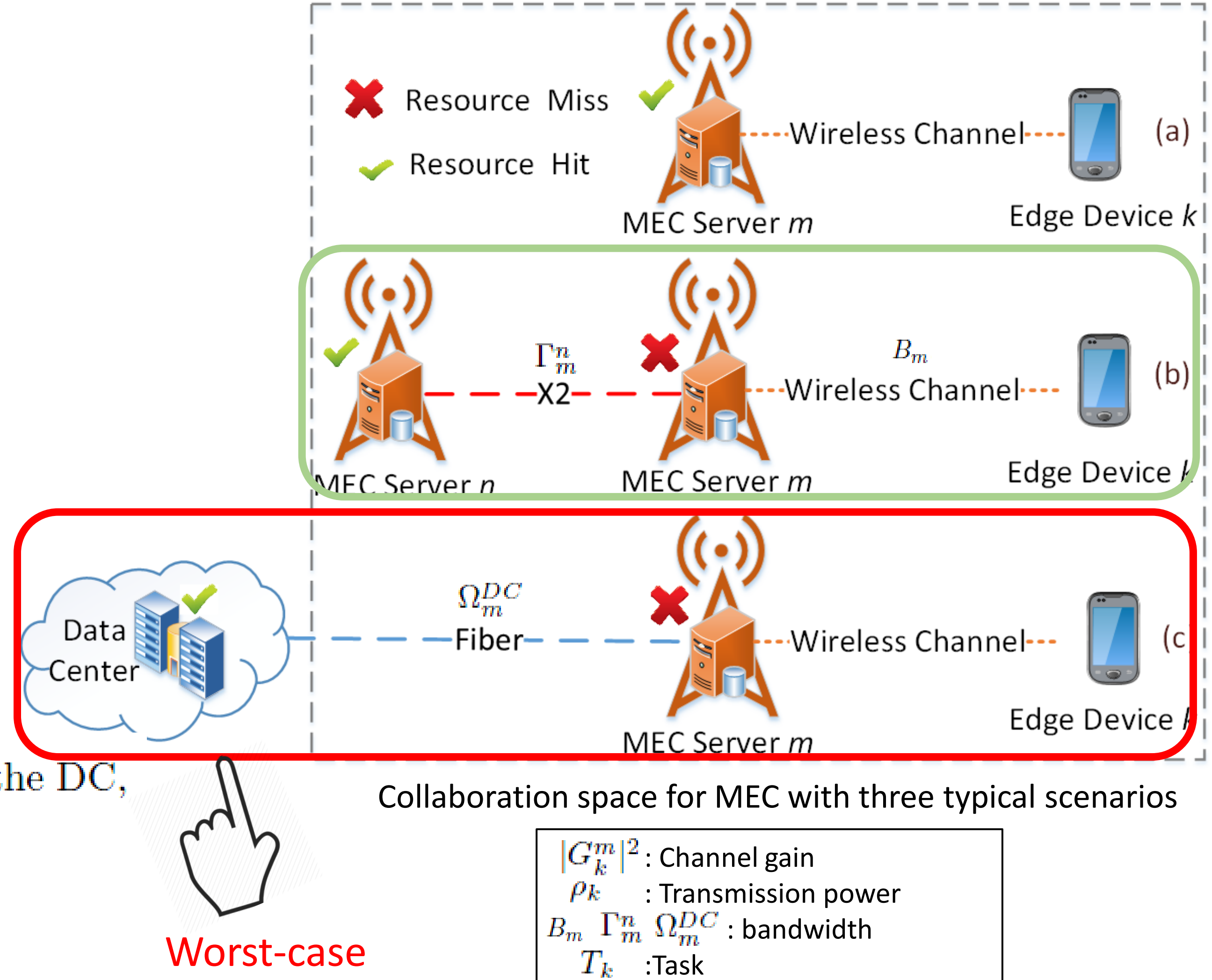
Scenario (c)

- When the resources are not available in the whole collaboration space, BS m forwards the request to DC

$$y_k^{m\to DC} = \begin{cases} 1, & \text{if } T_k \text{ is offloaded from BS } m \text{ to the DC,} \\ 0, & \text{otherwise.} \end{cases}$$

- The offloading delay between BS m and DC

$$\tau_k^{m\to DC} = \frac{\sum_{k\in\mathcal{K}_m} y_k^{m\to DC} s(d_k)}{\Omega_m^{DC}}, \ \forall m$$



Collaboration space for MEC with three typical scenarios

$|G_k^m|^2$: Channel gain
ρ_k : Transmission power
B_m Γ_m^n Ω_m^{DC} : bandwidth
T_k :Task
$s(d_k)$: Size of input data

Joint 4C Model for Collaborative Big Data MEC

Computation Model

Scenario 1: Local Computation at User Device

- Energy consumption of CPU computation: $E_k = s(d_k)\boxed{\nu}\boxed{\tilde{z}_k}P_k^2,\ k \in \mathcal{K}$
- The execution latency for task: $l_k = \frac{s(d_k)\tilde{z}_k}{\boxed{P_k}}$

Computation workload

constant parameter(related to CPU H/W)

Total computation capacity

Computation deadline

- When $l_k > \tilde{\tau}_k,\ \tilde{z}_k > P_k$, or $E_k > \tilde{E}_k$, edge device can keep the computational task until the resources become available for local computation via its device. Otherwise, edge device needs to offload task to MEC server

Computation energy

$$\alpha_k = \begin{cases} 0, & \text{if } \tilde{z}_k > P_k, \text{ or } l_k > \tilde{\tau}_k, \text{ or } \boxed{E_k} > \boxed{\tilde{E}_k} \\ 1, & \text{otherwise.} \end{cases}$$

Available energy in user device k

Edge device status parameter

Average waiting time

$$\tau_k^{\text{loc}} = \begin{cases} l_k, & \text{if } \alpha_k = 1, \text{ and } x_k^m = 0, \\ l_k + \varphi_k, & \text{if } \alpha_k = 0, \text{ and } x_k^m = 0, \\ 0, & \text{if } \alpha_k = 0, \text{ and } x_k^m = 1, \end{cases}$$

Offload

Scenario 2: Computation at MEC Server

Sub-scenario (a)

- Offloaded task to MEC server:

$$y_k^{k\to m} = \begin{cases} 1, & \text{whem BS } m \text{ computes offloaded task } T_k \\ & \text{by edge device } k\,, \\ 0, & \text{otherwise.} \end{cases}$$

$$\sum_{k\in\mathcal{K}_m} x_k^m p_{km} y_k^{k\to m} \le P_m,\ \forall m \in \mathcal{M}.$$

- Computation allocation:

$$p_{km} = P_m \frac{\tilde{z}_k}{\sum_{g\in\mathcal{K}_m} \tilde{z}_g},\ \forall k \in \mathcal{K}_m,\ m \in \mathcal{M}.$$

Joint 4C model for collaborative big data MEC

Computation Model

Scenario 2: Computation at MEC Server

- The execution latency: $$l_{km} = \frac{s(d_k)\tilde{z}_k}{p_{km}}.$$ $$\boxed{\tau^e_{km}} = \tau_k^{k\to m} + l_{km},\ \forall k \in \mathcal{K}_m,\ m \in \mathcal{M}.$$

Total executing time of offloaded task

Sub-scenario (b)

- When $\tilde{z}_k > p_{km}$ or $\tau^e_{km} > \tilde{\tau}_k$, MEC server m does not have enough computational resources to meet the computation deadline. Then, it forwards a request to another BS n in collaboration space

$$\tau^e_{kmn} = \tau_k^{k\to m} + \tau_k^{m\to n} + l_{kn},\ \forall k \in \mathcal{K}_m,\ \text{and } m, n \in \mathcal{M}.$$

Sub-scenario (c)

- When the resources are not available in the whole collaboration space, BS m forwards the request to DC

$$\tau^e_{kmDC} = \tau_k^{k\to m} + \tau_k^{m\to DC} + l_{kDC},\ \forall k \in \mathcal{K}_m,\ \text{and } m \in \mathcal{M}$$

Control for communication and computation at MEC server

- **Coordination:** The constraints to ensure that task is executed at only one location

$$(1 - x_k^m) + x_k^m\Big(y_k^{k\to m} + \sum_{n\in\mathcal{M}} y_k^{m\to n} + y_k^{m\to DC}\Big) = 1, \qquad (23)$$

$$\max\{y_k^{k\to m}, y_k^{m\to n}, y_k^{m\to DC}, \forall n\} \leq x_k^m,\ \forall k \in \mathcal{K}_m. \qquad (24)$$

Joint 4C model for collaborative big data MEC

Caching Model at MEC server

$$w_m^k = \begin{cases} 1, & \text{if MEC server } m \in \mathcal{M} \text{ caches the data } d_k, \\ 0, & \text{otherwise.} \end{cases}$$

Total cache capacity at MEC

$$\left(\sum_{k \in \mathcal{K}_m} y_k^{k \to m} + \sum_{n \neq m \in \mathcal{M}} \sum_{k \in \mathcal{K}_n} y_k^{n \to m} \right) w_m^k s(d_k) \leq C_m, \quad \forall m \in \mathcal{M}.$$

Control Model for communication, computation, and caching at MEC server

- We propose a distributed optimization control model that coordinates and integrates the communication, computation, and caching models
- We use a cache rewards that aims to maximize the backhaul bandwidth saving by reducing the data exchange between MEC servers and remote DC, i.e., increasing the cache hits:

Request arrival rate

Alleviated backhaul bandwidth: $$\Psi(\boxed{x}, \boxed{y}, \boxed{w}) = \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}_m} s(d_k) \boxed{\lambda_m^{d_k}} x_k^m \Big(y_k^{k \to m} w_m^k + \sum_{n \in \mathcal{M}} y_k^{m \to n} w_n^k \Big), \quad (27)$$

Offload(User), offload(MEC), cache

- We use total offloading and computation delay that aims to minimize delay

Total delay $$\Theta(x, y) = \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}_m} (1 - x_k^m) \tau_k^{\text{loc}} + x_k^m \tau_k^{\text{off}}.$$

KYUNG HEE UNIVERSITY

NETWORKING INTELLIGENT LAB

Problem Formulation and Solution

- We formulate the joint 4C in collaborative big-data MEC as an optimization problem that *jointly minimizes both bandwidth consumption and latency* as follows:

Total delay

Alleviated backhaul bandwidth

$$\min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}} \ \Theta(\boldsymbol{x},\boldsymbol{y}) - \eta\Psi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) \quad (29)$$

subject to:

Communication

$$\sum_{k\in\mathcal{K}_m} x_k^m a_k^m \leq 1, \ \forall m \in \mathcal{M}, \quad (29a)$$

Computation

$$\sum_{k\in\mathcal{K}_m} x_k^m p_{km} y_k^{k\to m} \leq P_m, \forall m \in \mathcal{M}, \quad (29b)$$

$$x_k^m \Big(\sum_{k\in\mathcal{K}_m} y^{k\to m} + \sum_{n\neq m\in\mathcal{M}} \sum_{k\in\mathcal{K}_n} y_k^{n\to m} \Big) w_m^k s(d_k) \leq C_m, \quad (29c)$$

$$(1-x_k^m) + x_k^m \Big(y_k^{k\to m} + \sum_{n\in\mathcal{M}} y_k^{m\to n} + y_k^{m\to DC} \Big) = 1, \quad (29d)$$

$$\max\{y_k^{k\to m}, y_k^{m\to n}, y_k^{m\to DC}, \forall n\} \leq x_k^m, \quad (29e)$$

Caching

Coordination

Solution:
Using Block Successive Upper-bound Minimization (BSUM), we proposed Distributed optimization control algorithm for 4C in big data MEC

$$\mathcal{B}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}) := \Theta(\boldsymbol{x},\boldsymbol{y}) - \eta\Psi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}).$$

Objective function | Total delay | Saved backhaul bandwidth

Proximal upper-bound function

$$\mathcal{B}_j(\boldsymbol{x}_j, \boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}, \boldsymbol{w}^{(t)}) := \mathcal{B}(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{w}}) + \frac{\varrho_j}{2}\|(\boldsymbol{x}_j - \tilde{\boldsymbol{x}})\|^2.$$

The solution is updated by solving:

Offload (User)
$$\boldsymbol{x}_j^{(t+1)} \in \min_{\boldsymbol{x}_j\in\mathcal{X}} \mathcal{B}_j(\boldsymbol{x}_j, \boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}, \boldsymbol{w}^{(t)}), \quad (36)$$

Offload (MEC)
$$\boldsymbol{y}_j^{(t+1)} \in \min_{\boldsymbol{y}_j\in\mathcal{Y}} \mathcal{B}_j(\boldsymbol{y}_j, \boldsymbol{y}^{(t)}, \boxed{\boldsymbol{x}^{(t+1)}}, \boldsymbol{w}^{(t)}), \quad (37)$$

cache
$$\boldsymbol{w}_j^{(t+1)} \in \min_{\boldsymbol{w}_j\in\mathcal{W}} \mathcal{B}_j(\boldsymbol{w}_j, \boldsymbol{w}^{(t)}, \boxed{\boldsymbol{x}^{(t+1)}}, \boxed{\boldsymbol{y}^{(t+1)}}). \quad (38)$$

Proposed Algorithm

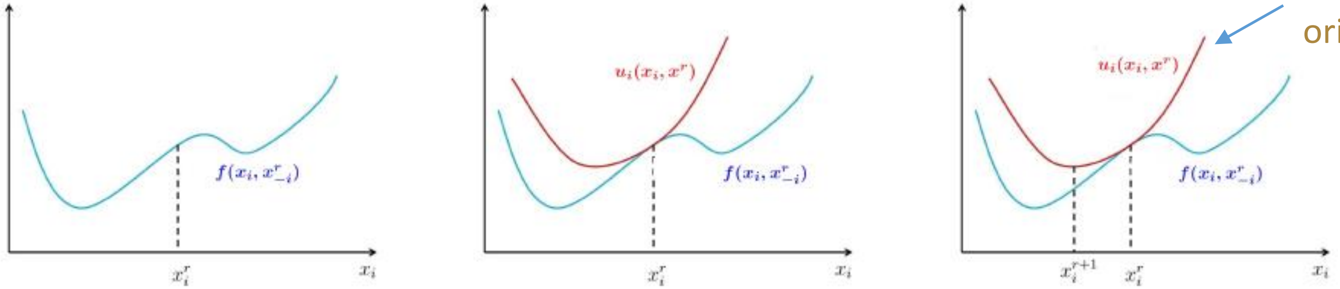
Algorithm 3. Distributed Optimization Control Algorithm (BSUM-based) for 4C in big Data MEC

1: **Input:** T: A vector of demands; B_m, P_m, and C_m: communication, computational and caching resources;
2: **Output:** $\boldsymbol{x}^*$, $\boldsymbol{y}^*$, $\boldsymbol{w}^*$, $\boldsymbol{c}$: A vector of cache allocation, $\boldsymbol{p}$: A vector of computation allocation, and $\boldsymbol{R}$: A vector of communication resources allocation;
3: Each user device $k \in \mathcal{K}$ chooses the offloading decision x_k^m;
4: If $x_k^m = 1$, user device $k \in \mathcal{K}$ sends its demand T_k to BS $m \in \mathcal{M}$;
5: For each T_k received at BS $m \in \mathcal{M}$, check RAT update;
6: Initialize $t = 0$, $\epsilon > 0$;
7: Find initial feasible points $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{w}^{(0)})$;
8: **repeat**
9: Choose index set $\mathcal{J}$;
10: Let $\boldsymbol{x}_j^{(t+1)} \in \min_{\boldsymbol{x}_j \in \mathcal{X}} \mathcal{B}_j(\boldsymbol{x}_j, \boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}, \boldsymbol{w}^{(t)})$;
11: Set $\boldsymbol{x}_k^{t+1} = \boldsymbol{x}_k^t, \forall k \notin \mathcal{J}$;
12: Go to Step 4, find $\boldsymbol{y}_j^{(t+1)}$, $\boldsymbol{w}_j^{(t+1)}$ by solving (37) and (38);
13: $t = t + 1$;
14: **until** $\| \frac{\mathcal{B}_j^{(t)} - \mathcal{B}_j^{(t+1)}}{\mathcal{B}_j^{(t)}} \| \leq \epsilon$;
15: Generate a binary solution of $\boldsymbol{x}_j^{(t+1)}$, $\boldsymbol{y}_j^{(t+1)}$, $\boldsymbol{w}_j^{(t+1)}$ and obtain $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{R}$ by using rounding technique (39) and solving $\mathcal{B}_j + \xi\Delta$;
16: Then, calculate β. If $\beta \leq 1$, consider $\boldsymbol{x}^* = \boldsymbol{x}_j^{(t+1)}$, $\boldsymbol{y}^* = \boldsymbol{y}_j^{(t+1)}$, and $\boldsymbol{w}^* = \boldsymbol{w}_j^{(t+1)}$ as a solution;
17: Update RAT, and send RAT update in collaboration space.

RAT : resource allocation table

BSUM overview[1]:

surrogate function of the original objective



(a) (b) (c)

[1]. Hong, Mingyi, et al. "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing." IEEE Signal Processing Magazine 33.1 (2016): 57-77 (Google citation: 393)

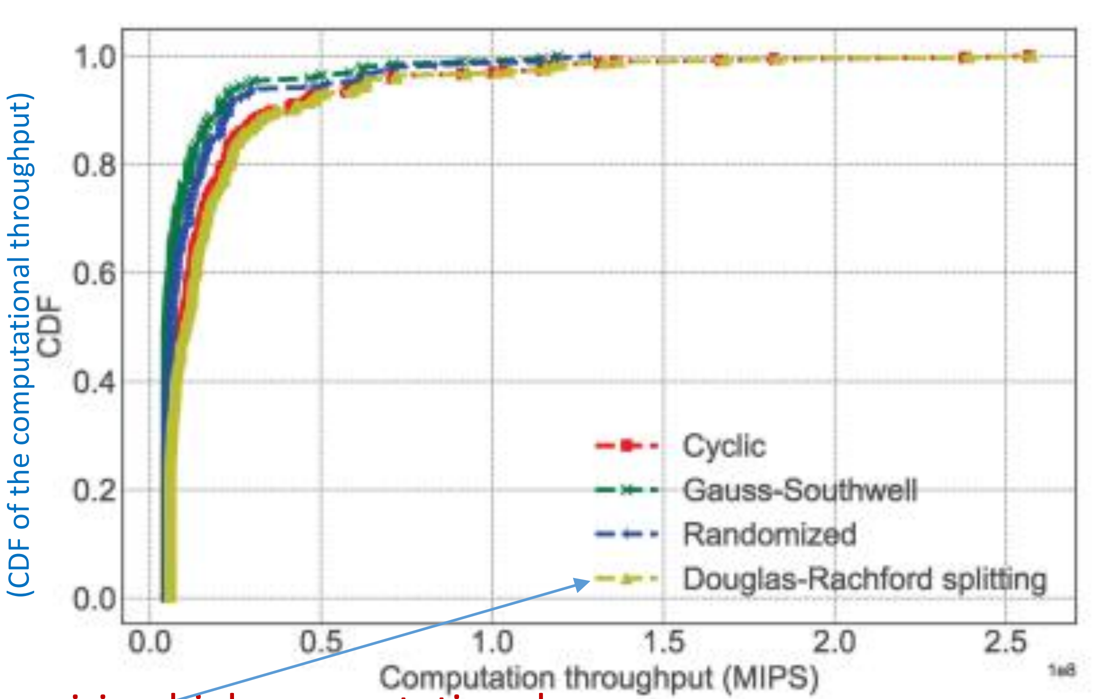
KYUNG HEE UNIVERSITY

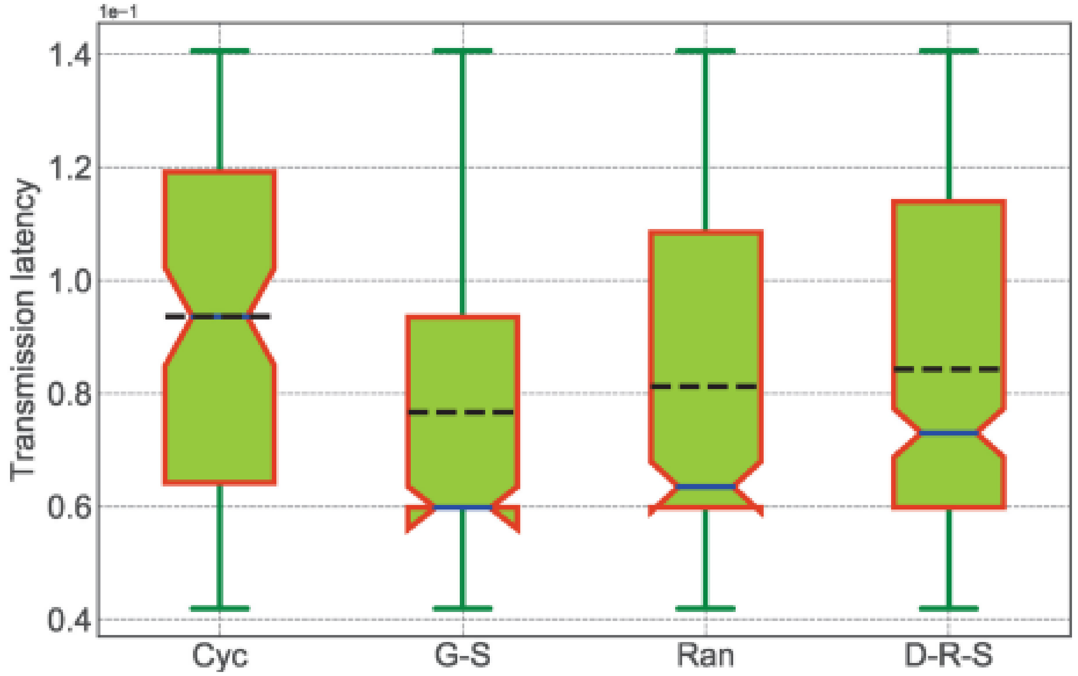
NETWORKING INTELLIGENT LAB

- For forming collaboration spaces, we use the Sitefinder dataset from Edinburgh DataShare [3]
- We randomly select one MNO, which has 12,777 BSs, through use of the Overlapping K-mean Method for Collaboration Space (OKM-CS) algorithm, where we group these BSs into 1,000 collaborations spaces
- Among 1,000 collaboration spaces, we randomly select one collaboration space, which has 12 BSs, and we associate each BS with 1 MEC server

1. O. Boswarva et al., "Sitefinder mobile phone base station database," Edinburgh DataShare , the University of Edinburgh, UK , Feb. 2017.

Performance Evaluation





Requiring high computational resources

- BSUM selection rules:
  - Cyc: Cyclic
  - G-S: Gauss Southwell
  - Ran: Randomized
- D-R-S: Douglas Rachford splitting

- BSUM and D-R-S algorithms enable to decompose our problem into small sub-problems, and address each sub-problem separately

Choong Seon Hong, et al. "Joint Communication, Computation, Caching, and Control in Big Data Multi-access Edge Computing." arXiv preprint arXiv:1803.11512, 2018 (Submitted to IEEE Transactions on Mobile Computing).

Game Theory Approaches

- Introduction
- Use Case : Network Slicing: Dynamic Isolation Provisioning and Energy Efficiency

Introduction: History of Game Theory

- **John von Neuman** (1903-1957) co-authored, Theory of Games and Economic Behavior, with Oskar Morgenstern in 1940s, establishing game theory as a field
- **John Nash** (1928-) developed a key concept of game theory (Nash equilibrium) which initiated many subsequent results and studies
- Since 1970s, game-theoretic methods have come to dominate microeconomic theory and other fields

Nobel prizes

- Nobel prize in Economic Sciences 1994 awarded to **Nash, Harsanyi** (Bayesian games) and **Selten** (Subgame perfect equilibrium)
- 2005, **Auman** and **Schelling** got the Nobel prize for having enhanced our understanding of cooperation and conflict through game theory
- 2007, **Leonid Hurwicz, Eric Maskin** and **Roger Myerson** won Nobel Prize for having laid the foundations of mechanism design theory

John von Neumann
(December 28,1903 – February 8,1957)

John Forbes Nash, Jr.
(born June 13, 1928)
Winner of Nobel Prize in Economics (1994)

Zhu Han, Dusit Niyato, Walid Saad, Tamer Başar, Are Hjørungnes, "Game Theory in Wireless and Communication Networks: Theory, Models, and Applications," Cambridge University Press, 2011

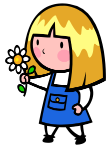
Game Theory: Mathematical models and techniques developed in economics to analyze interactive decision processes, predict the outcomes of interactions, and identify optimal strategies .

- Game theory techniques were adopted to solve many protocol design issues (e.g., resource allocation, power control, cooperation enforcement) in wireless networks
- Difference to control: against other players as well as nature
- Fundamental component of game theory is the notion of a game
  - A game is described by a set of rational *players*, the *strategies* associated with the players, and the *payoffs* for the players. A rational player has his own interest, and therefore, will act by choosing an available strategy to achieve his interest.
  - A player is assumed to be able to evaluate exactly or probabilistically, the outcome or payoff (usually measured by the utility) of the game which **depends not only on his action but also on other players' actions.**

2012 Nobel Prize in Economic Science.

Adam

Geeta, Heiki, Irina, Fran



Fran

Bob

Irina, Fran, Heiki, Geeta

Geeta

Carl > Adam

We reach a stable marriage!

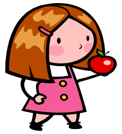
Carl

Geeta, Fran, Heiki, Irina

Heiki

David

Irina, Heiki, Geeta, Fran



Irina

David > Bob

- **Hierarchy** among the players exists
  - The player that imposes its own strategy upon others is called the **leader**
  - The other players who react to the leader's strategy are called **followers**

Definition 20 *In a two-person finite game with Player 1 as the leader, a strategy $s_1^* \in \mathcal{S}_1$ is called a* Stackelberg equilibrium strategy *for the leader, if*

$$\min_{s_2 \in \mathcal{R}_2(s_1^*)} u_1(s_1^*, s_2) = \max_{s_1 \in \mathcal{S}_1} \min_{s_2 \in \mathcal{R}_2(s_1)} u_1(s_1, s_2) \triangleq u_1^*. \tag{3.35}$$

The quantity u_1^* is the Stackelberg utility *of the leader. The same definition applies for the case where player 2 is the leader by simply swapping the subscripts 1 and 2.*

- Every two-person finite game admits a Stackelberg strategy for the leader
- Whenever the follower has a single optimal response for every strategy of the leader, then the leader can, at the Stackelberg solution, perform **at least as good as at the Nash equilibrium**

- Stackelberg games are **not** limited to the single-leader single-follower case

- In a single-leader multi-follower case, the Stackelberg equilibrium is basically composed of an **optimal** policy for the leader with respect to a **Nash equilibrium** of the followers
  - It is often desirable to have a **unique** Nash equilibrium for the followers game, so as to make the Stackelberg solution tractable
  - **Example application**: Pricing for Internet Service Providers

- Multi-leader multi-follower Stackelberg games
  - At the Stackelberg equilibrium, both leaders and followers are in a Nash equilibrium (the Nash equilibria are correlated)
  - Hard to solve when the followers game has many equilibria

Example: Buyer/Seller Game (Two Level)

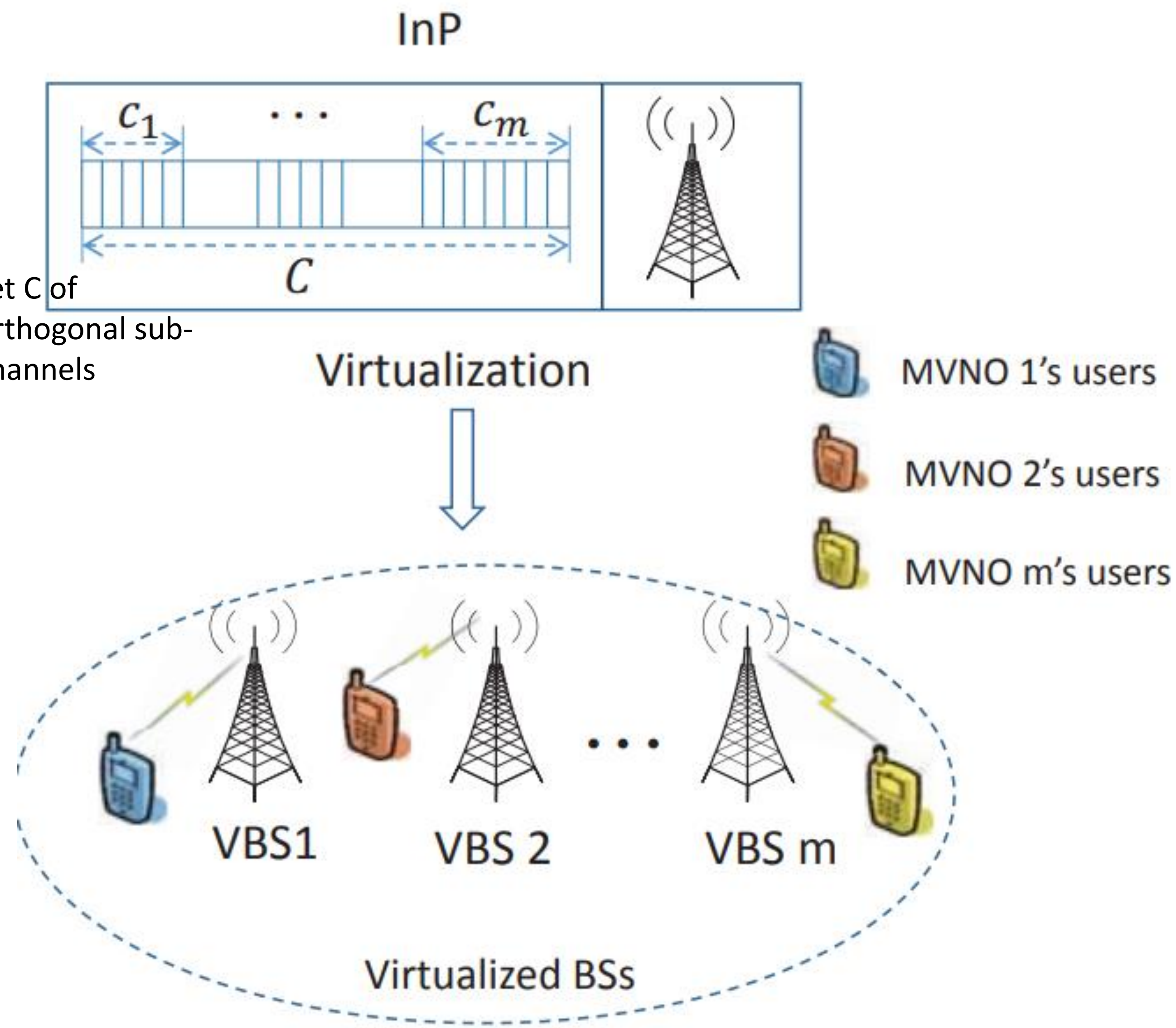
- **Buyer/Seller (Leader/Follower) Game**
  - Sender (buyer) buying the services from the relays to improve its performance, such as the transmission rate
  - Relays (sellers) selling service, such as power, by setting prices
  - Tradeoffs: Price too high, sender buying from others; price too low, profit low; sender decides to buy whose and how much to spend
  - Procedures: Convergence to the optimal equilibrium
  - Example: Power Control and Relay Section for Cooperative Transmission

$1000
Per Power

$800
Per Power

Use Case : Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach

- System Model
- Problem Formulation
- Simulation Results

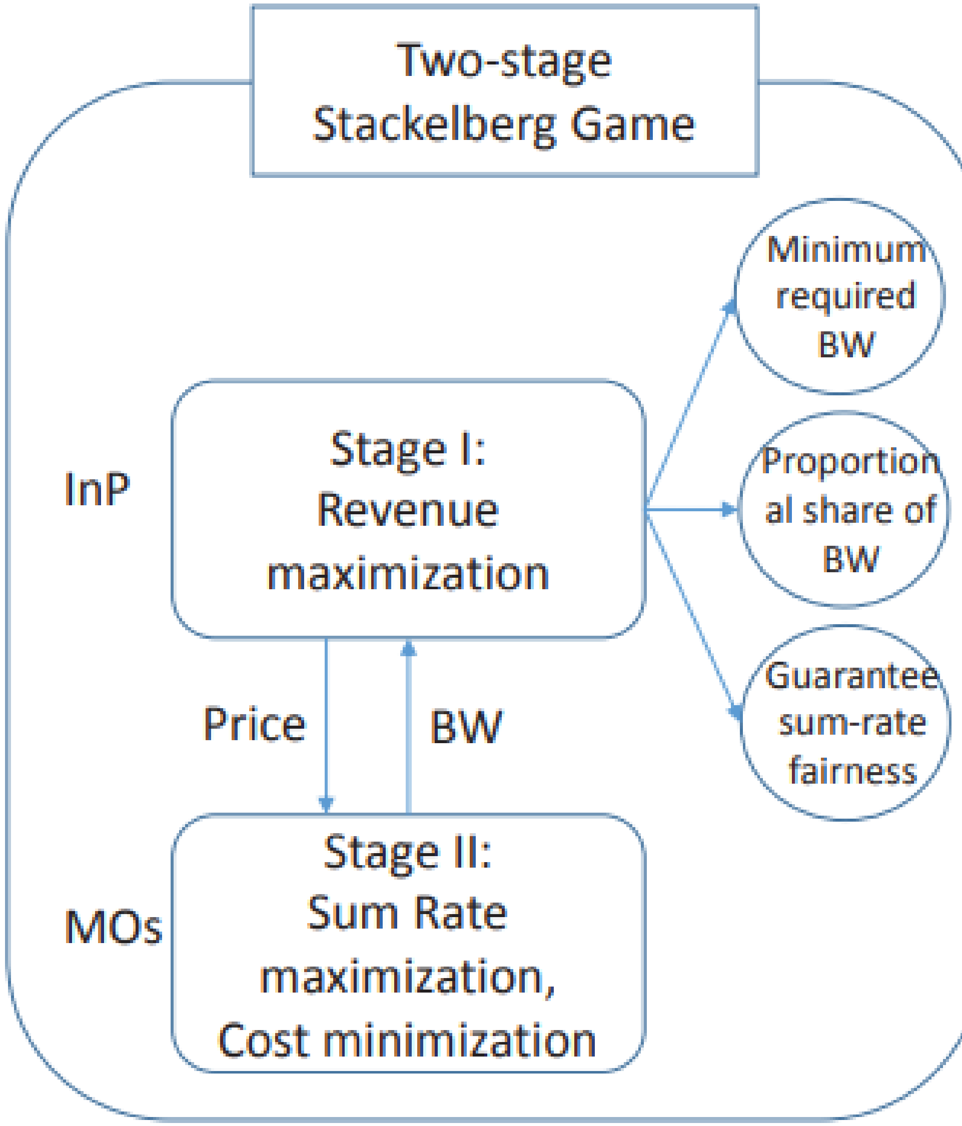


set C of orthogonal sub-channels

Formulate the resource allocation problem for the wireless network virtualization as a **hierarchical two stage Stackelberg game with InP plays the leader** role and **MVNOs act as followers**.

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

Stackelberg Game



Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

- Stage II: MVNO model - Followers Game

achievable data rates

normalized transmit power

large-scale channel power gain

$$R_{m,k} = c_{m,k}\omega_0 \ln\left(1 + \frac{P_m g_{m,k}}{c_{m,k}\omega_0 n_0}\right), \quad (1)$$

amount of bandwidth

background noise

$$\sum_{k=1}^{K_m} R_{m,k} \leq \bar{R}_m, \ \forall m \in \mathcal{M}. \quad (2)$$

pre-agreed bandwidth of slice allocated to MVNO m

<u>Net Utility function of MVNO m</u>

price per unit of bandwidth charged by InP

$$\mathcal{U}_m(\boldsymbol{c}_m, p_m) = \sum_{k=1}^{K_m} R_{m,k} - p_m \sum_{k=1}^{K_m} c_{m,k} \quad (3)$$

$$\mathbf{P_{MVNO}}: \quad \underset{\boldsymbol{c}_m}{\text{maximize}} \ \ \mathcal{U}_m(\boldsymbol{c}_m, p_m) \quad (4)$$

The optimization problem of MVNO m

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

- Stage I: InP model - Leader Game

$$c_m = \sum_{k=1}^{K_m} c_{m,k}, \forall m$$

represents the total bandwidth sold by InP to the MVNO m

Revenue function of the InP $$\mathcal{R}(\boldsymbol{c}, \boldsymbol{p}) = \sum_{m=1}^{M} c_m p_m, \quad (5)$$

$$\mathbf{P_{InP}}: \quad \underset{\boldsymbol{p}}{\text{maximize}} \quad \mathcal{R}(\boldsymbol{c}, \boldsymbol{p}) \quad (6)$$

$$\text{subject to} \quad c_m \geq \rho_m^{min} C, \ \forall m \quad (7)$$

minimum required BW for each MVNO

$$\sum_{m=1}^{M} c_m \leq C, \quad (8)$$

proportional share of BW among different MVNOs

$$\sum_{k=1}^{K_m} R_{m,k} \leq \bar{R}_m, \ \forall m, \quad (9)$$

service contract constraint.

$$0 \leq p_m \leq p^{\max}, \ \forall m, \quad (10)$$

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

- **Optimal solution for Stage I :** the optimal solution of the Stage-I based on the optimal solution of Stage II.

$$\mathbf{P}'_{\mathbf{InP}}: \quad \max_{\boldsymbol{p}}. \quad \sum_{m=1}^{M} p_m \sum_{k=1}^{K_m} G_{m,k} e^{-\left(\frac{\omega_0 + p_m}{\omega_0}\right)} \tag{14}$$

$$\text{s.t.} \quad \sum_{k=1}^{K_m} G_{m,k} e^{-\left(\frac{\omega_0 + p_m}{\omega_0}\right)} \geq \rho_m^{min} C, \ \forall m \tag{15}$$

$$\sum_{m=1}^{M} \sum_{k=1}^{K_m} G_{m,k} e^{-\left(\frac{\omega_0 + p_m}{\omega_0}\right)} \leq C, \tag{16}$$

$$\sum_{k=1}^{K_m} G_{m,k} (\omega_0 + p_m) e^{-\left(\frac{\omega_0 + p_m}{\omega_0}\right)} \leq \bar{R}_m, \ \forall m, \tag{17}$$

$$0 \leq p_m \leq p^{\max}, \ \forall m, \tag{18}$$

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

Problem Formulation: Convert to Lagrangian form

$$L(\boldsymbol{p}, \lambda, \mu, \nu) = \sum_{n=1}^{M} L_m(p_m, \lambda_m, \mu, \nu_m), \tag{19}$$

where λ_m, μ_m and ν_m are Lagrange multipliers and

$$\begin{aligned} L_m(p_m, \lambda_m, \mu, \nu_m) &= p_m \sum_{k=1}^{K_m} G_{m,k} e^{-(\frac{\omega_0 + p_m}{\omega_0})} \\ &+ \lambda_m \sum_{k=1}^{K_m} G_{m,k} e^{-(\frac{\omega_0 + p_m}{\omega_0})} - \mu \sum_{k=1}^{K_m} G_{m,k} e^{-(\frac{\omega_0 + p_m}{\omega_0})} \\ &- \nu_m \sum_{k=1}^{K_m} G_{m,k} (\omega_0 + p_m) e^{-(\frac{\omega_0 + p_m}{\omega_0})} - \delta_m p_m. \end{aligned} \tag{20}$$

The dual problem is then given as

$$\begin{aligned} &\max. \quad D(\lambda, \mu, \nu) \\ &\text{s.t} \quad \lambda, \mu, \nu \geq 0, \end{aligned} \tag{21}$$

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

Problem Formulation: Convert to Lagrangian form

Lagrangian multiplier

$$\lambda_m^{(t+1)} = \left[\lambda_m^{(t)} - \kappa_\lambda^{(t)} \left(\sum_{k=1}^{K_m} G_{m,k} e^{-\left(\frac{\omega_0 + p_m^{(t)}}{\omega_0}\right)} - \rho_m^{min} C\right)\right]^+, \tag{22}$$

$$\mu^{(t+1)} = \left[\mu^{(t)} + \kappa_\mu^{(t)} \left(\sum_{m=1}^{M} \sum_{k=1}^{K_m} G_{m,k} e^{-\left(\frac{\omega_0 + p_m^{(t)}}{\omega_0}\right)} - C\right)\right]^+, \tag{23}$$

$$\nu_m^{(t+1)} = \left[\nu_m^{(t)} + \kappa_\nu^{(t)} \left(\sum_{k=1}^{K_m} G_{m,k} (\omega_0 + p_m^{(t)}) e^{-\left(\frac{\omega_0 + p_m^{(t)}}{\omega_0}\right)} - \bar{R}_m\right)\right]^+ \tag{24}$$

$$\delta_m^{(t+1)} = \left[\delta_m^{(t)} + \kappa_\delta^{(t)} \left(p_m^{(t)} - p^{\max}\right)\right]^+, \tag{25}$$

Algorithm 1 Dual based Resource Allocation

1: **input**: $\epsilon > 0$
2: **initialize**: $t = 0$; $p_m^{(0)}$; $\lambda_m^{(0)}, \mu^{(0)}, \nu_m^{(0)} \geq 0$; $\kappa_\lambda^{(0)}, \kappa_\mu^{(0)}, \kappa_\nu^{(0)} > 0$
3: **repeat**
4: $t \leftarrow t + 1$;
5: Update $\lambda_m^{(t+1)}, \mu^{(t+1)}, \nu_m^{(t+1)}$ according to (22-24);
6: Update $p_m^{(t+1)}$ according to
7: **until** $|p_m^{(t+1)} - p_m^{(t)}| \leq \epsilon$;
8: Each MVNO calculates $c_{m,k}^*$ according to (13), and rounds $c_{m,k}^*$ according to (27);

$$p_m^{(t+1)} = \left[\frac{(\omega_0 \nu_m^{(t)} + \mu^{(t)} - \lambda_m^{(t)})\Lambda^{(t)}}{(1 - \nu_m^{(t)})\Lambda^{(t)} - \delta_m^{(t)}} \right]^+, \qquad (26)$$

Optimal required BW for each MVNO

$$\bar{c}_{m,k}^* = \lfloor c_{m,k}^* \rfloor, k = 1, ..., K_m, m = 1, ..., M, \qquad (27)$$

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

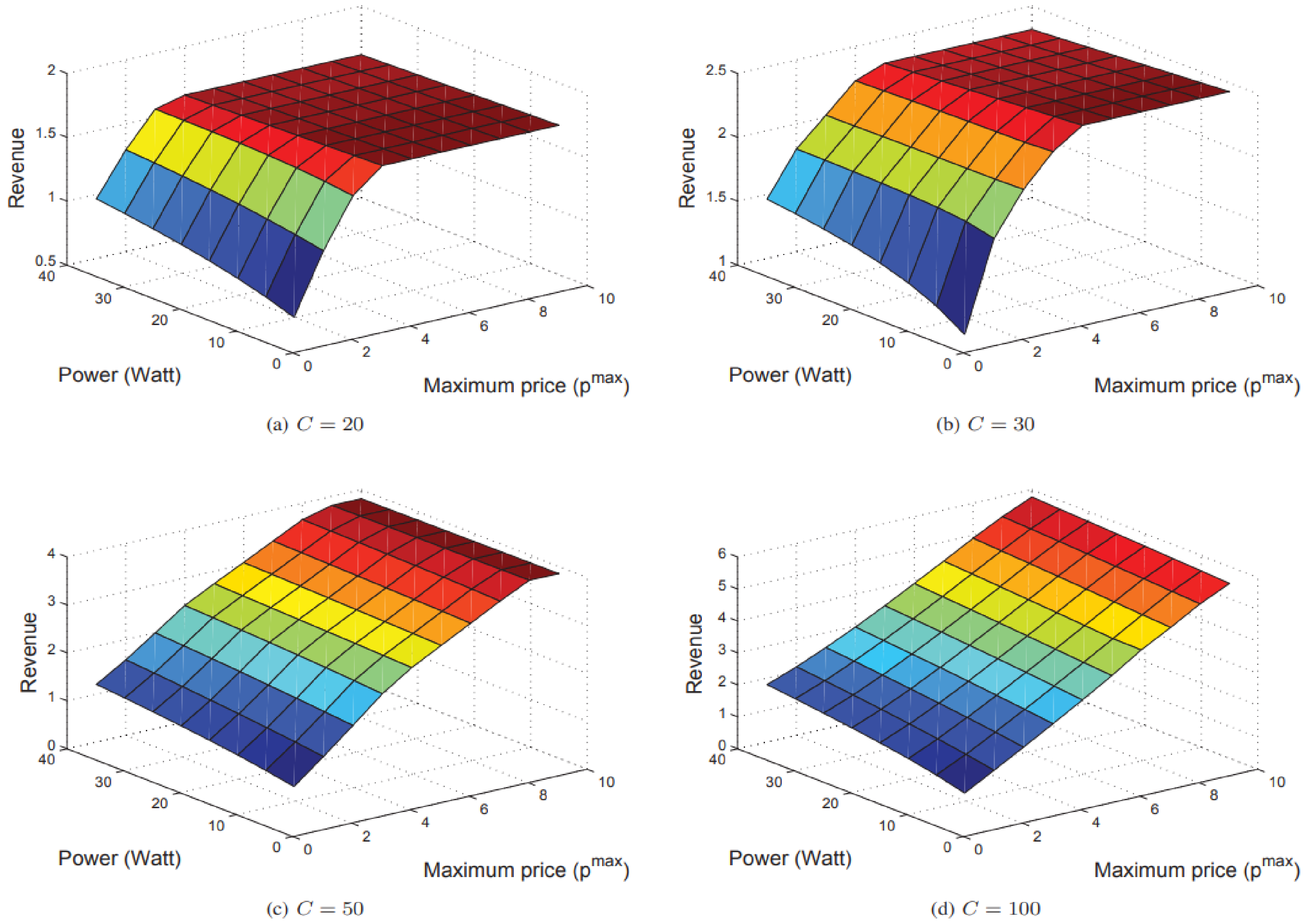


Fig. 4: Revenue versus power (Watt) and maximum price ($p^{\max}$) for different number of subchannels C.

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

In terms of price paid by the MVNOs

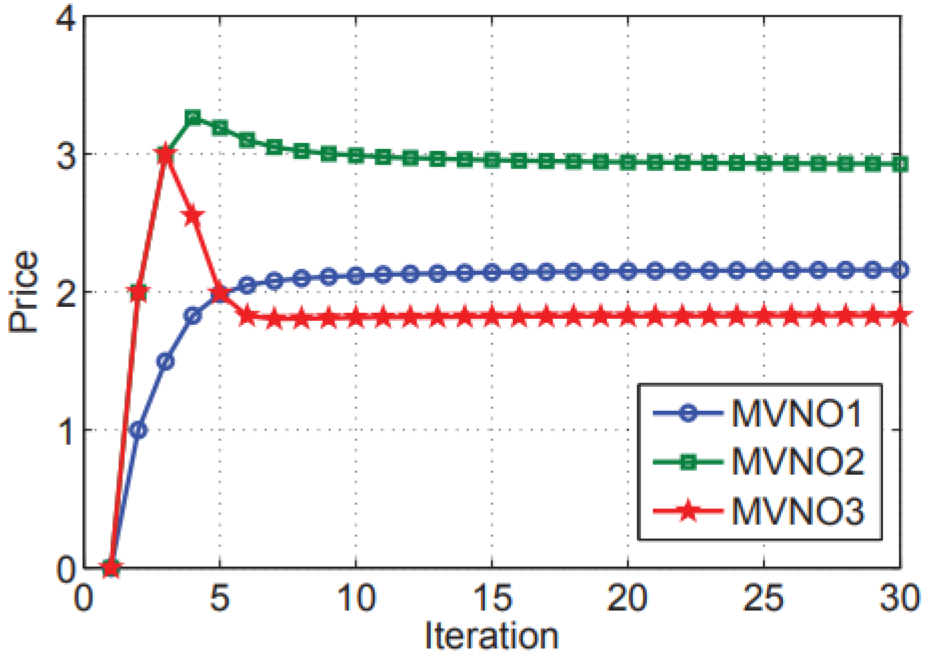


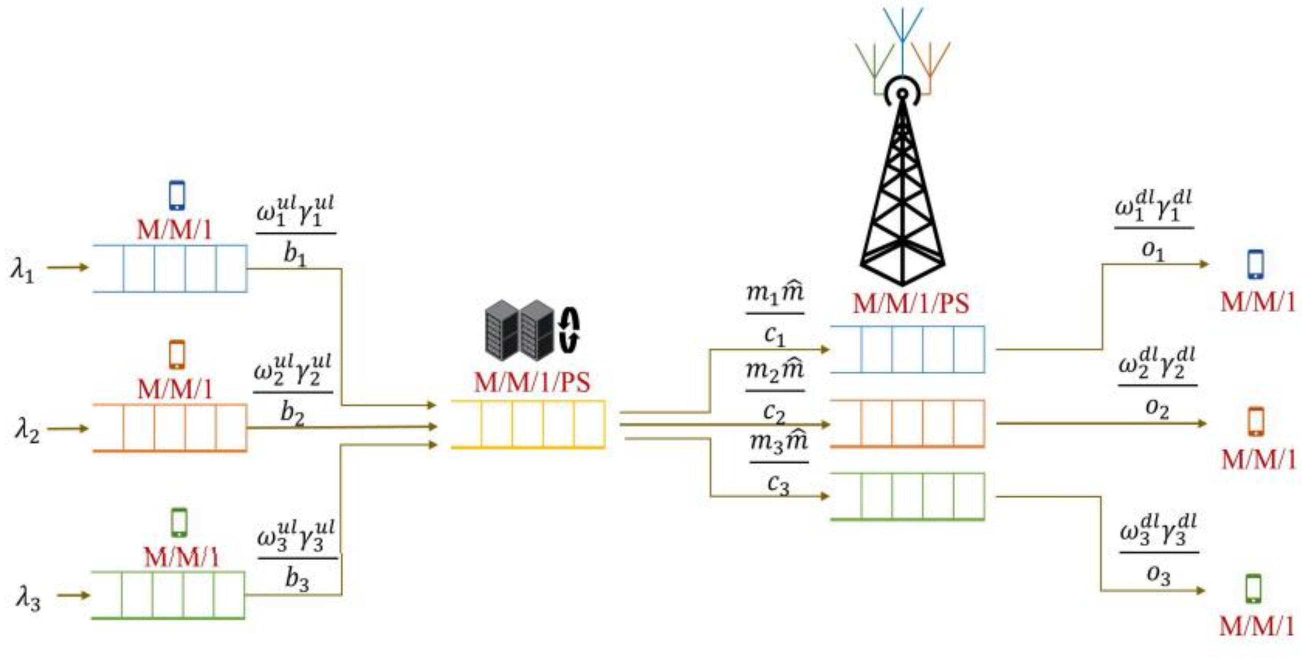
Fig. 3: Convergence of Algorithm 1

Tai Manh Ho, Nguyen H. Tran, S.M Ahsan Kazmi, Choong Seon Hong, "Dynamic Pricing for Resource Allocation in Wireless Network Virtualization: A Stackelberg Game Approach", The International Conference on Information Networking (ICOIN 2017), Jan. 11-13, 2017, Da Nang, Vietnam

Use Case 2: Radio and Computing Resource Allocation in Co-located Edge Computing: A Generalized Nash Equilibrium Model

- System Model
- Problem Formulation
- Simulation Results

Use case 2: System Model



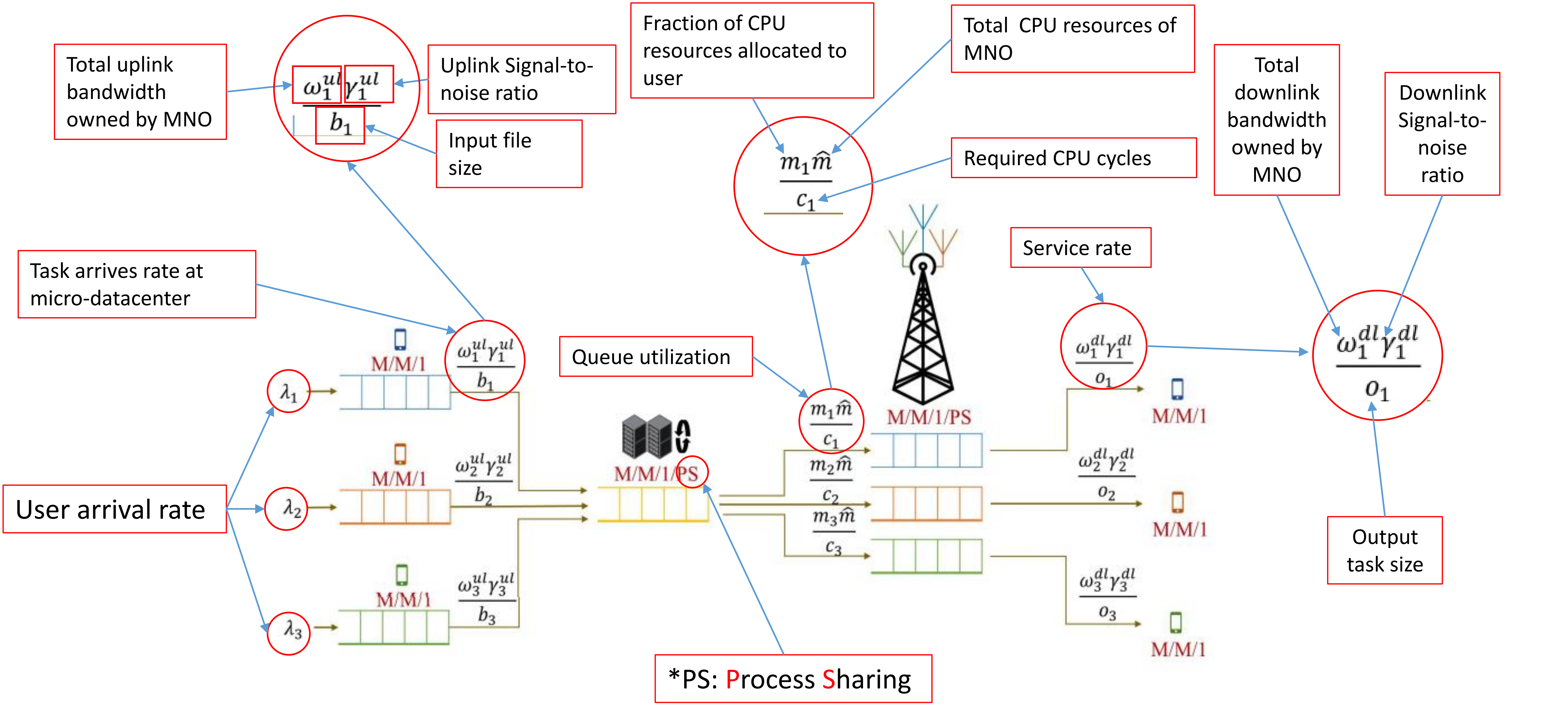
- Consider a single-cell tower model with N MNOs and a CRP are co-located
- Each MNO has a set of U users
- The challenging problem is joint uplink, downlink, and computing resources allocation problem
- The task offloading is modeled as a network of queues where the end-to-end latency is calculated based on the performance of the queue network

Goal: Generalized Nash Equilibrium Problem (GNEP) to capture the conflicting interests in the resource allocation among MNOs and CRP

*CRP: Computing Resource Provider
*MNO: Mobile Network Operator

Chit Wutyee Zaw, Choong Seon Hong, et al. "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model." *IEEE Transactions on Mobile Computing* (2021).

Use case 2: System Model



Chit Wutyee Zaw, Choong Seon Hong, et al. "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model." *IEEE Transactions on Mobile Computing* (2021).

The goal of the CRP is to minimize the total energy cost:

Base-load or static power consumption

Dynamic power consumption

Uplink bandwidth

Downlink bandwidth

$$\mathrm{P}_{\mathrm{CRP}}(\boxed{\mathbf{W}^{\mathrm{ul}}}, \boxed{\mathbf{W}^{\mathrm{dl}}}) : \quad \underset{\mathbf{m}}{\text{minimize}} \quad \boxed{\Theta_{\mathrm{CRP}}}(\mathbf{m}, \mathbf{W}^{\mathrm{ul}})$$

$$\Theta_{\mathrm{CRP}} = \left[\boxed{(1-\alpha)\psi P_{\mathrm{MEC}}} + \boxed{\alpha P_{\mathrm{MEC}}}\right] \Delta t.$$

subject to

$$t_u^{\mathrm{ul}} + t_u^{p} + t_u^{\mathrm{dl}} \leq \Delta t, \qquad \forall u$$

Uplink + processing +downlink latency less than or equal to time constraint

The end-to-end Latency Constraint

$$\sum_{j=1}^{N} \sum_{u \in \mathcal{U}_j} m_u \leq 1.$$

Resource Constraints: total CPU resources allocated to users must also be less than or equal to 1 (100%)

$$\upsilon_u \leq 1-\epsilon, \qquad \forall u \in \mathcal{U}_j, \forall j \in \mathcal{J},$$

$$\rho_j \leq 1-\epsilon, \qquad \forall j \in \mathcal{J},$$

$$\psi \leq 1-\epsilon.$$

Utilization of the uplink transmission queue

Utilization of the downlink transmission queue

Utilization of the processing queue

Any small, positive number (0..1)

Queue Utilization Constraints: The utilization of the queues must be less than 1 to maintain the stability of the queues

Chit Wutyee Zaw, Choong Seon Hong, et al. "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model." *IEEE Transactions on Mobile Computing* (2021).

Use case 2: Problem Formulation

MNO objective is to minimize the total energy cost by considering the expected completion time, uplink and downlink budget, the stability of the uplink, downlink and computing resource queues

Power consumption for signal processing, decoding

Power consumption for encoding, transmitting signals to users

$$\mathrm{P}_{\mathrm{MNO}_j}(\mathbf{W}^{\mathrm{ul}}, \mathbf{m}) : \underset{\mathbf{W}_j^{\mathrm{ul}}, \mathbf{W}_j^{\mathrm{dl}}}{\text{minimize}} \quad \Theta_{\mathrm{MNO}_j}(\mathbf{W}_j^{\mathrm{ul}}, \mathbf{W}_j^{\mathrm{dl}}, \mathbf{m}_j)$$

$$\Theta_{\mathrm{MNO}_j} = [\boxed{(1-\alpha)[\delta^{\mathrm{ul}}\mu_j + \delta^{\mathrm{dl}}\rho_j]P_{\mathrm{BS}}} + \boxed{\alpha P_{\mathrm{BS}}}]\Delta t,$$

subject to

$$t_u^{\mathrm{ul}} + t_u^{p} + t_u^{\mathrm{dl}} \leq \Delta t, \qquad \forall u.$$

The end-to-end Latency Constraint

$$\sum_{u \in \mathcal{U}_j} \omega_u^{\mathrm{ul}} \leq 1,$$

$$\sum_{u \in \mathcal{U}_j} \omega_u^{\mathrm{dl}} \leq 1.$$

Resource Constraints: total bandwidth allocated to users must also be less than or equal to 1 (100%)

J: MNO

-j: except MNO j

$$\tau_{-j}^{\mathrm{ul}} := \sum_{i=1, i \neq j}^{J} \sum_{u \in \mathcal{U}_i} \frac{\omega_u^{\mathrm{ul}} \gamma_u^{\mathrm{ul}}}{b_u}$$

$$\tau_{-j}^{\mathrm{m}} := \sum_{i=1, i \neq j}^{J} \sum_{u \in \mathcal{U}_i} \frac{m_u \hat{m}}{c_u}.$$

$$\frac{\sum_{u \in \mathcal{U}_j} \frac{\omega_u^{\mathrm{ul}} \gamma_u^{\mathrm{ul}}}{b_u} + \boxed{\tau_{-j}^{\mathrm{ul}}}}{\sum_{u \in \mathcal{U}_j} \frac{m_u \hat{m}}{c_u} + \boxed{\tau_{-j}^{\mathrm{m}}}} \leq 1 - \epsilon,$$

Queue Utilization Constraints:
The utilization of the queues must be less than 1 to maintain the stability of the queues

Chit Wutyee Zaw, Choong Seon Hong, et al. "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model." *IEEE Transactions on Mobile Computing* (2021).

Use case 2: Solution Approach

Algorithm 1 Penalized NEP Algorithm for the Resource Allocation Game

1: Choose the initial penalty parameters $\kappa_p^{\mathrm{UE}_u,0}, \forall u \in \mathcal{U}$, $\kappa_p^{\mathrm{BS}_j,0}, j = 1, \cdots, N$ and $\kappa_p^{\mathrm{MEC},0}$, $p = 0, \cdots, N$.
2: $k \leftarrow 0$.
3: Choose an initial point for $\mathbf{m}^k, \mathbf{W}^{\mathrm{ul},k}, \mathbf{W}^{\mathrm{dl},k}$ as in (30).
4: **repeat**
5: CRP solves the problem in (28).
6: Each MNO j solves the problem in (29).
7: **until** $[\mathbf{W}^{*,\mathrm{ul}}, \mathbf{W}^{*,\mathrm{dl}}, \mathbf{m}^*]$ is unchanged.
8: **if** $\tilde{h}_u(\omega_u^{*,\mathrm{ul}}, m_u^*, \omega_u^{*,\mathrm{dl}}) \leq 0, \forall u \in \mathcal{U}$, $f_j(\mathbf{W}_j^{*,\mathrm{dl}}, \mathbf{m}_j^*) \leq 0, j = 1, \cdots, N$ and $g(\mathbf{W}^{*,\mathrm{ul}}, \mathbf{m}^*) \leq 0$ **then**
9: $[\mathbf{W}^{*,\mathrm{ul}}, \mathbf{m}^*, \mathbf{W}^{*,\mathrm{dl}}]$ is a GNE.
10: **else**
11: Penalty parameters, $\kappa_p^{\mathrm{UE}_u,k+1}, \forall u \in \mathcal{U}$, $\kappa_p^{\mathrm{BS}_j,k+1}, j = 1, \cdots, N$ and $\kappa_p^{\mathrm{MEC},k+1}$, $p = 0, \cdots, N$, are updated as in (31), (32) and (33).
12: $k \leftarrow k + 1$.
13: **go to** line number 5.
14: **end if**

First, the initial penalty parameters and resource allocation are chosen

$$\omega_u^{\mathrm{ul},0} = \frac{\lambda_u b_u}{\gamma_u^{\mathrm{ul}}(1-\epsilon)}, \quad \omega_u^{\mathrm{dl},0} = \frac{o_u \sum_{u \in \mathcal{U}_j} \frac{m_u^0 \hat{m}}{c_u}}{\gamma_u^{\mathrm{dl}} |\mathcal{U}_j| (1-\epsilon)},$$
$$m_u^0 = \frac{c_u \sum_{j=1}^{J} \sum_{u \in \mathcal{U}_j} \frac{\omega_u^{\mathrm{ul},0} \gamma_u^{\mathrm{ul}}}{b_u}}{\hat{m}(1-\epsilon) \sum_{j=1}^{J} |\mathcal{U}_j|}. \qquad (30)$$

Each player solves its optimization problem until NE is found

Updates the penalty parameters

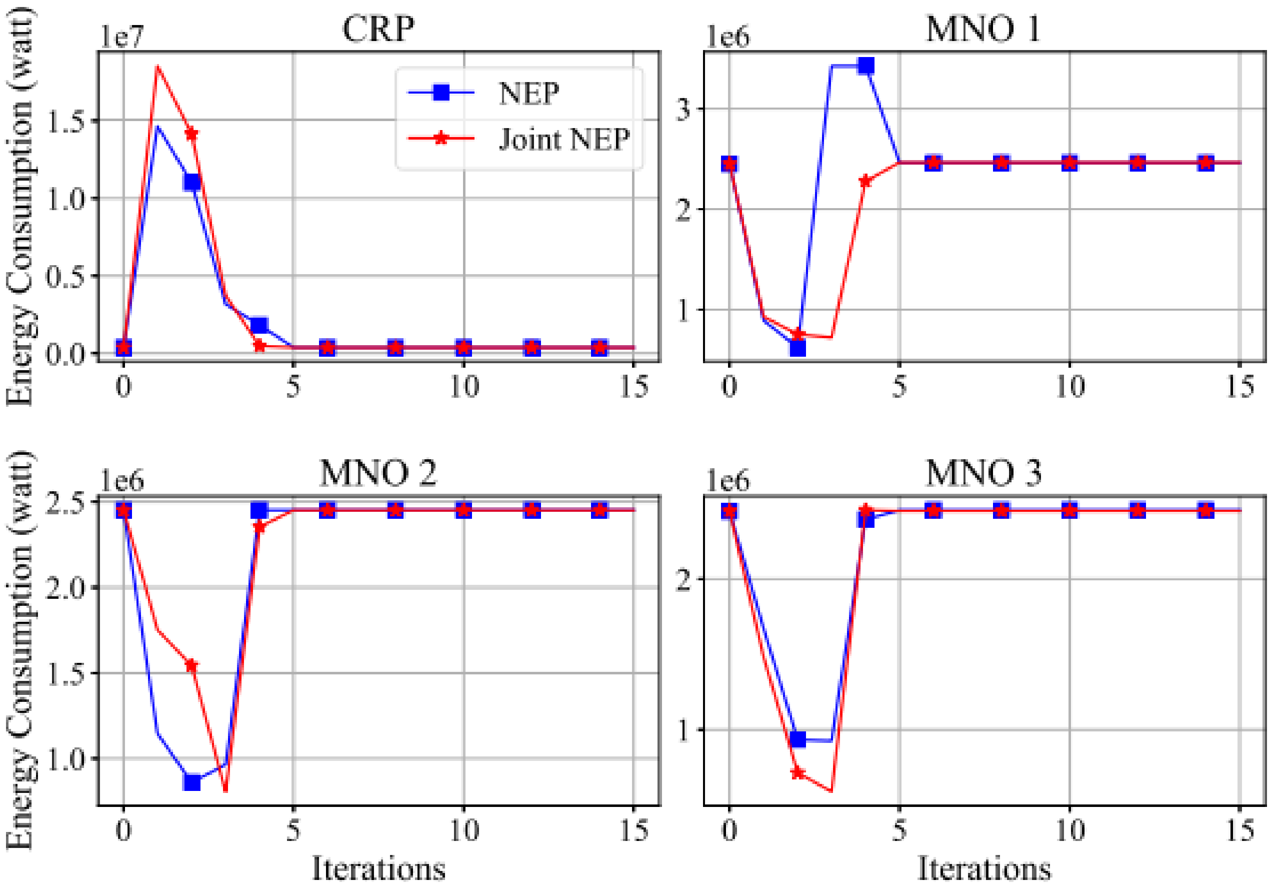
$$\kappa_p^{\mathrm{UE}_u,k+1} = \begin{cases} \kappa_p^{\mathrm{UE}_u,k} + \delta_p^{\mathrm{UE}_u,k} & \text{if } \tilde{h}_u(\omega_u^{\mathrm{ul}}, m_u, \omega_u^{\mathrm{dl}}) > 0, \\ \kappa_p^{\mathrm{UE}_u,k} & \text{if } \tilde{h}_u(\omega_u^{\mathrm{ul}}, m_u, \omega_u^{\mathrm{dl}}) \leq 0, \end{cases} \qquad (31)$$

$$\kappa_p^{\mathrm{BS}_j,k+1} = \begin{cases} \kappa_p^{\mathrm{BS}_j,k} + \delta_p^{\mathrm{BS}_j,k} & \text{if } f_j(\mathbf{W}_j^{*,\mathrm{dl}}, \mathbf{m}_j^*) > 0, \\ \kappa_p^{\mathrm{BS}_j,k} & \text{if } f_j(\mathbf{W}_j^{*,\mathrm{dl}}, \mathbf{m}_j^*) \leq 0, \end{cases} \qquad (32)$$

$$\kappa_p^{\mathrm{MEC},k+1} = \begin{cases} \kappa_p^{\mathrm{MEC},k} + \delta_p^{\mathrm{MEC},k} & \text{if } g(\mathbf{W}^{*,\mathrm{ul}}, \mathbf{m}^*) > 0, \\ \kappa_p^{\mathrm{MEC},k} & \text{if } g(\mathbf{W}^{*,\mathrm{ul}}, \mathbf{m}^*) \leq 0, \end{cases} \qquad (33)$$

Chit Wutyee Zaw, Choong Seon Hong, et al. "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model." *IEEE Transactions on Mobile Computing* (2021).

Use case 2: Numerical Results



Convergence of the proposed algorithm

Chit Wutyee Zaw, Choong Seon Hong, et al. "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model." *IEEE Transactions on Mobile Computing* (2021).

AI Based Approaches

- System Model
- Problem Formulation
- Solution Approach
- Simulation Results

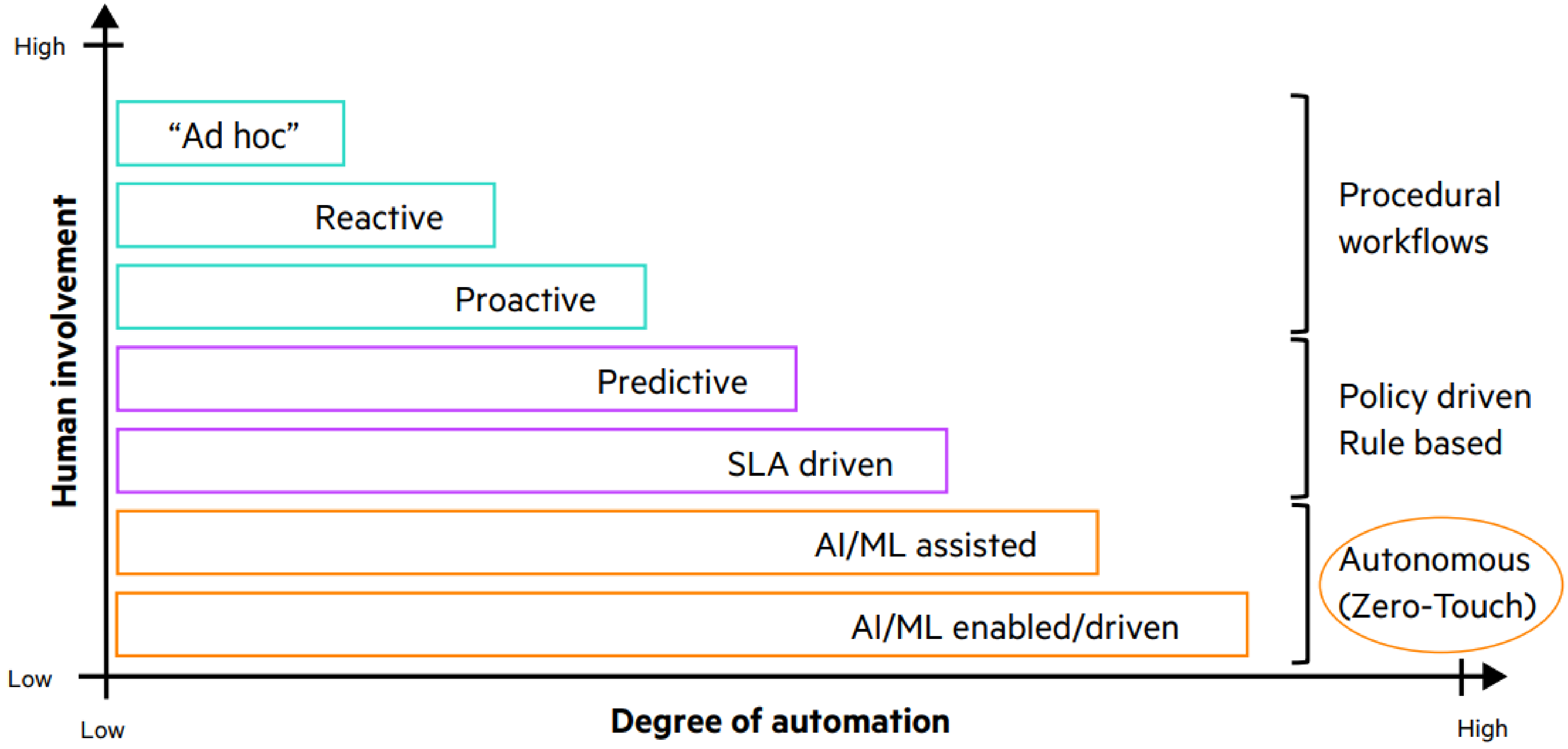
- The basic goal of an AI in 5G and beyond network is its ability to extract, predict, and characterize specific patterns from datasets

- To unleash the true potential of 5G and beyond networks:
  - Intelligent functions using AI across both the edge and core of the network are required along with the novel enabling technologies

- AI functions must be able to:
  - Adaptively exploit the wireless system resources
  - Generated data to optimize network operation
  - Guarantee the QoS in real time

- Such mobile edge and core intelligence can only be realized by integrating fundamental notions of artificial intelligence (AI) across the wireless infrastructure and end-user devices

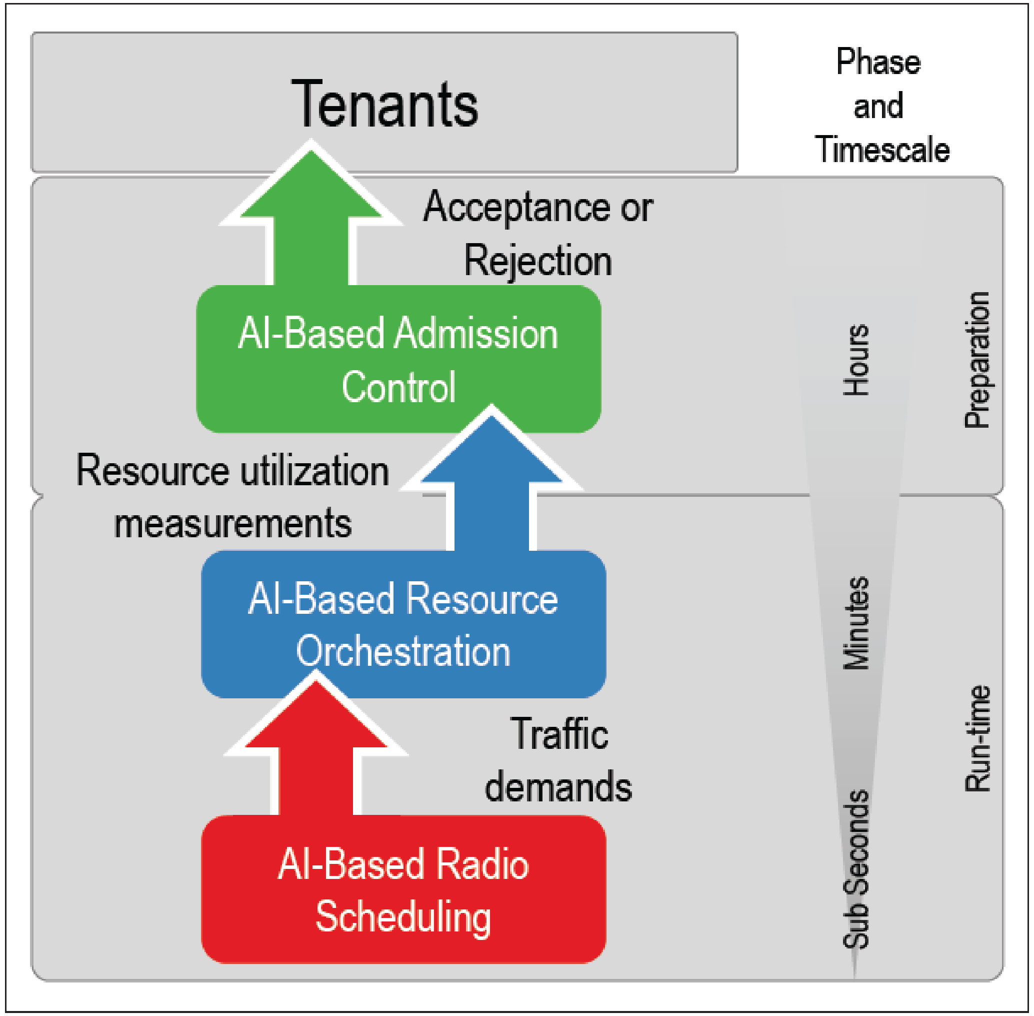
Introduction: How Close Are Operators To Zero-Touch Automation?



Source: Intel

FIGURE 19. Infrastructure automation maturity

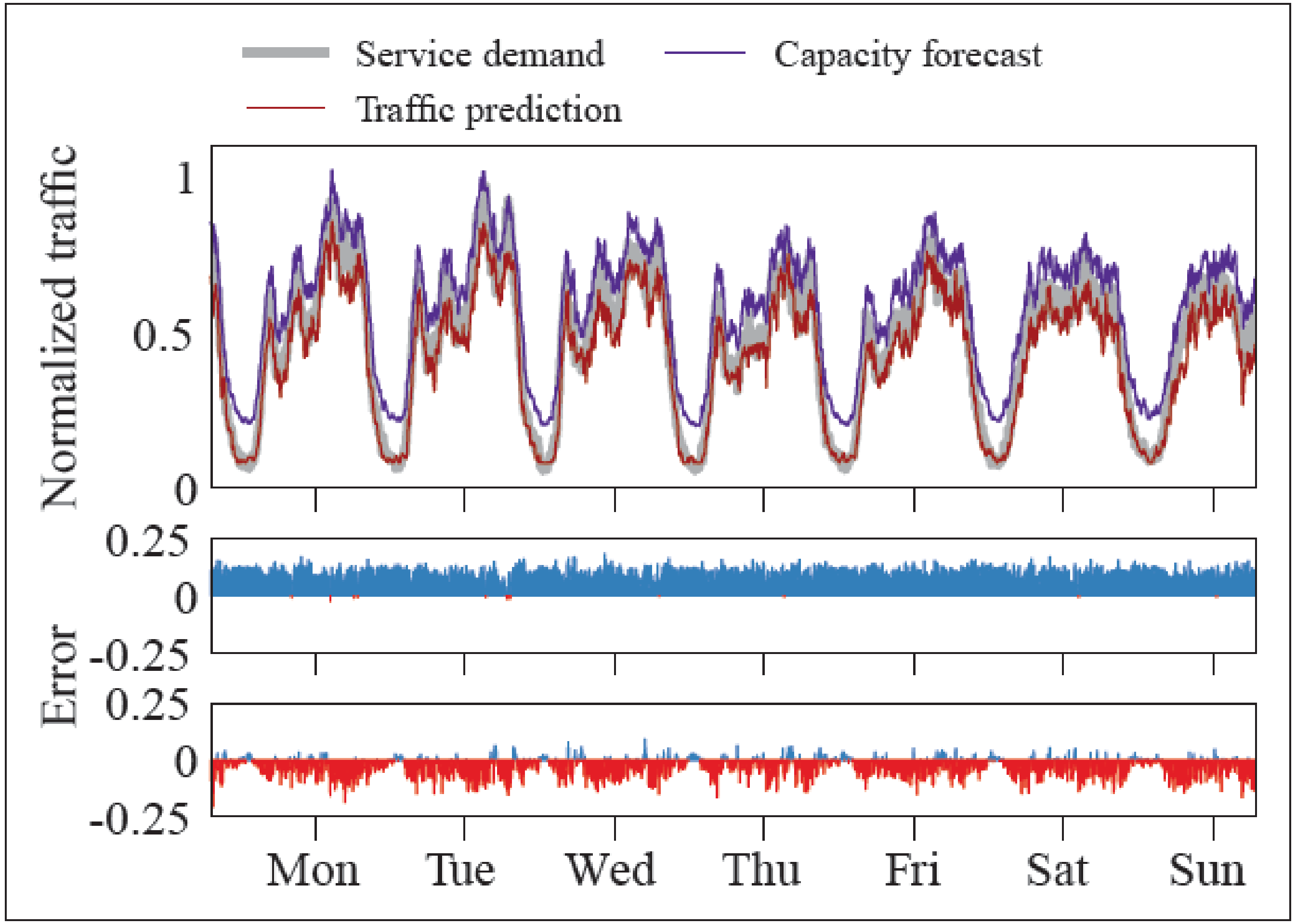
Source: 5G ORCHESTRATION AND AUTOMATION TOWARD ZERO-TOUCH SERVICE MANAGEMENT



Comprehensive network slicing framework. The diagram outlines the timescales and composition of the key slice management functions.

D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs and X. Costa-Perez, "Network Slicing Meets Artificial Intelligence: An AI-Based Framework for Slice Management," in IEEE Communications Magazine, vol. 58, no. 6, pp. 32-38, June 2020, doi: 10.1109/MCOM.001.1900653.

Predict the Demand



Top: predictions of a sample one-week demand, as produced by a legacy MAE traffic predictor and by a capacity forecasting model; middle: error incurred by the capacity forecasting model, which only generates overprovisioning; bottom: error incurred by the MAE traffic predictor, which leads to frequent service requirement violations.

D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs and X. Costa-Perez, "Network Slicing Meets Artificial Intelligence: An AI-Based Framework for Slice Management," in IEEE Communications Magazine, vol. 58, no. 6, pp. 32-38, June 2020, doi: 10.1109/MCOM.001.1900653.

Network Slicing Meets Artificial Intelligence

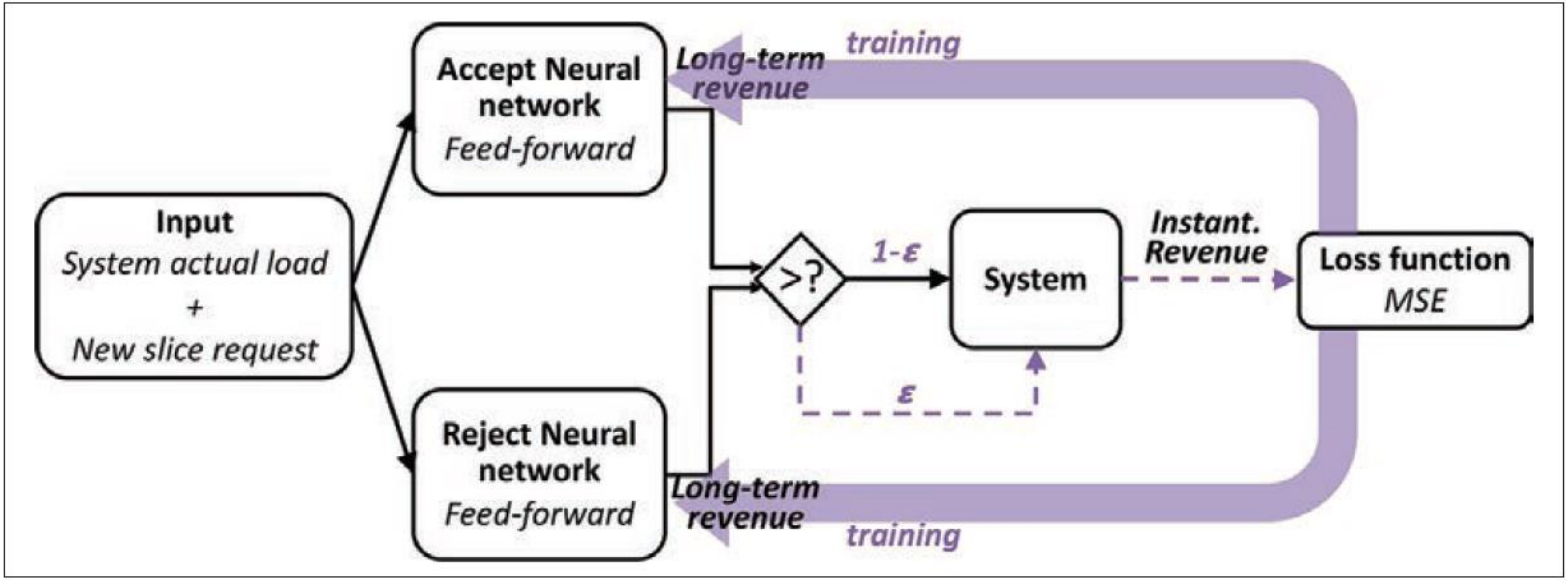


Figure 2. High-level design of AI-based slice admission control.

D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs and X. Costa-Perez, "Network Slicing Meets Artificial Intelligence: An AI-Based Framework for Slice Management," in IEEE Communications Magazine, vol. 58, no. 6, pp. 32-38, June 2020, doi: 10.1109/MCOM.001.1900653.

Various Machine Learning Approaches

- Reinforcement Learning (RL)
  - Q-Learning can efficiently approximate the optimal slice admission policy that maximizes the MNO's revenue [1]
  - RL algorithms can be designed model-free by appropriately selecting the reward functions, which makes them much more robust against imperfect estimations of the slicing statistics
- Deep Learning
  - As the most important part of modern artificial intelligence technologies, artificial neural networks (ANN) are known to be efficient in modeling non-linear systems.
  - This can be used to enhance RL methods into deep reinforcement learning (DRL) methods, such the deep Q-Learning method reported in [2].
  - Another common application of ANN is the model estimation and prediction of complex non-linear processes.
  - Encoder-decoder structured [3]cognitive network is proven capable to predict service capacity requirement in a data-driven fashion with high accuracy, which helps the slice orchestrator to make decisions in slice admission control and cross-slice resource allocation.

[1] A. Ayala-Romero et al., "vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," Proc. ACM MobiCom, Oct. 2019, pp. 1–16.
[2] D. Bega et al., "A Machine Learning Approach to 5G Infrastructure Market Optimization," IEEE Trans. Mobile Computing, vol. 19, no. 3, Feb. 2020, pp. 498–512.
[3] T. P. Lillicrap et al., "Continuous Control With Deep Reinforcement Learning," arXiv preprint arXiv:1509.02971, 2015.

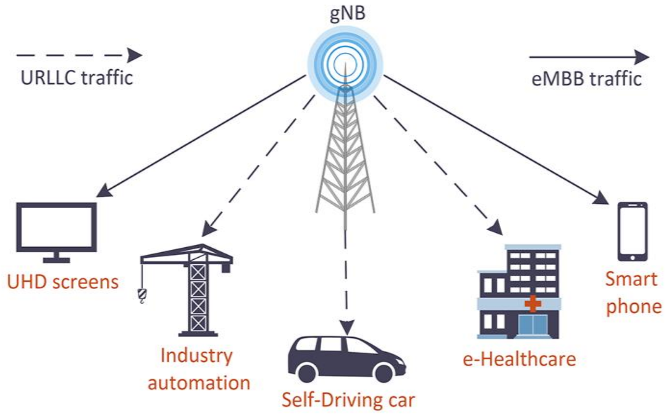
Use case : Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach

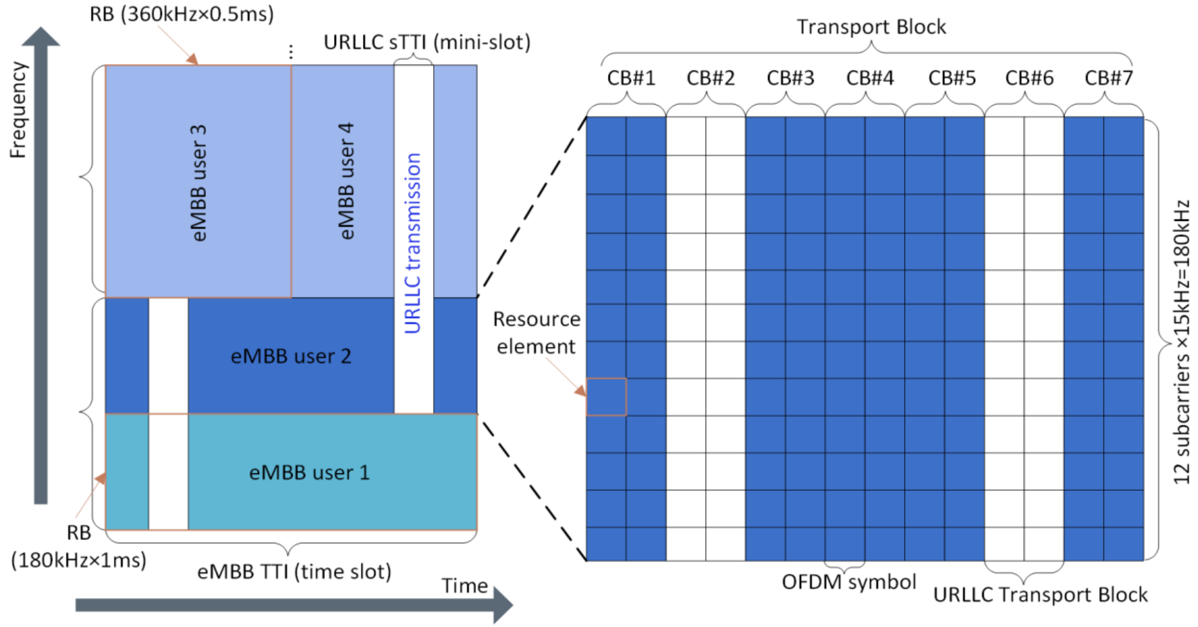
- AI for 5G Networks
- Network Slicing Meets Artificial Intelligence
- Evolution of Operations Functionality

System Model

Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach.

- This paper studies the resource slicing problem in a dynamic multiplexing scenario of two distinct 5G services, namely Ultra-Reliable Low Latency Communications (URLLC) and enhanced Mobile Broad Band (eMBB).
- While eMBB services focus on high data rates, URLLC is very strict in terms of latency and reliability.





We propose a system design in which eMBB traffic is transmitted over long TTIs while URLLC traffic is transmitted over short TTIs by puncturing the ongoing eMBB transmissions.
Transmitting the incoming URLLC traffic in the next short TTI ensures its latency requirement.

TTI : Transmission Time Interval, CB : Code Block

Madyan Alsenwi, Nguyen H. Tran, Mehdi Bennis, Shashi Raj Pandey, Anupam Kumar Bairagi, Choong Seon Hong , “Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach,” IEEE Transactions on Wireless Communications, Vol. 20, Issue 7, pp. 4585-4600, July 2021

Problem Formulation

We aim at:

1. Maximizing the eMBB data rate,
2. Satisfying the URLLC reliability constraint, and
3. reducing the impact of URLLC on eMBB transmissions.

The data rate of eMBB traffic is captured by the Shannon's capacity considering the impact of URLLC transmissions, while URLLC depends on the finite blocklength capacity model due to its small packets size nature.

*The objective function is formulated based on **Markowitz mean-variance** model to maximize the average eMBB data rate for a given level of risk.*

Problem Formulation

The variance part captures the dynamic characteristics of wireless channels

RBs allocation variable, Power allocation variable, puncturing variable

Weighting parameter

Data rate of eMBB user k at time slot t

$$\underset{\boldsymbol{x},\boldsymbol{p},\boldsymbol{z}}{\text{maximize}} \quad \sum_{k=1}^{K} \mathbb{E}_{\mathrm{h}}\left[\frac{1}{T}\sum_{t=0}^{T} r_k^e(t)\right] - \beta \mathrm{Var}_{\mathrm{h}}\left[r_k^e(t)\right] \tag{1a}$$

Data rate of URLLC user n at time slot t

URLLC packet size

$$\text{subject to} \quad \Pr\left[\sum_{n=1}^{N} r_n^u(t) \le \zeta L(t)\right] \le \epsilon^*, \tag{1b}$$

URLLC reliability

Total number of URLLC packets at a time slot t

$$\sum_{k=1}^{K}\sum_{b=1}^{B} p_{kb}(t) \le P_{\max}, \tag{1c}$$

$$\sum_{k=1}^{K} x_{kb}(t) \le 1, \quad \forall\, b \in \mathcal{B}, \tag{1d}$$

$$p_{kb}(t) \ge 0, \ \forall k \in \mathcal{K}, \ b \in \mathcal{B}, \tag{1e}$$

$$x_{kb}(t) \in \{0,1\}, \ \ \forall k \in \mathcal{K}, \ \ b \in \mathcal{B}, \tag{1f}$$

$$z_{kb}(t) \in \{0,1,\ldots,M\}, \ \ \forall k \in \mathcal{K}, \ b \in \mathcal{B}, \tag{1g}$$

We propose a two-phase-framework, including:

1. eMBB resource allocation phase.

RBs and transmission power are allocated to eMBB users by applying some **optimization techniques**.

2. URLLC scheduling phase.

we propose a **DRL-based algorithm** to schedule the URLLC transmissions over the ongoing eMBB transmissions.

1. eMBB Resource Allocation Phase:

We first simplify the objective function to a smoothing form and eliminate the complexity caused by the variance by using an equivalent **risk-averse utility function**.

We consider the exponential function that can capture both the mean and variance as defined in:

$$\mathcal{G}(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{z}) = \frac{1}{\mu} \log \mathbb{E}_{\mathrm{h}} \left[\exp\left(\mu \sum_{k=1}^{K} r_k^e(t) \right) \right], \quad (2)$$

$$\Rightarrow \quad \mathcal{G}(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{z}) = \mathbb{E}_{\mathrm{h}} \left[\sum_{k=1}^{K} r_k^e(t) \right] + \frac{\mu}{2} \mathrm{Var} \left[\sum_{k=1}^{K} r_k^e(t) \right] + \mathcal{O}(\mu^2). \quad (3)$$

μ controls the desired risk-sensitivity

1. eMBB Resource Allocation Phase (Cont.):

- We propose a Decomposition and Relaxation based Resource Allocation (DRRA) algorithm.
- The proposed DRRA algorithm decomposes the optimization problem into three subproblems: *1) eMBB RBs allocation, 2) eMBB power allocation, and 3) URLLC scheduling.*
- We replace the integer variable in the URLC scheduling problem, i.e., the number of punctured mini-slots, by a **continuous weighting variable** for each RB.
- Later, we calculate the number of punctured mini-slots from each RB by modeling it as a **binomial distribution** with parameters puncturing weight and number of mini-slots in each time slot.

Algorithm 1 : DRRA Algorithm for the eMBB/URLLC coexistence Problem

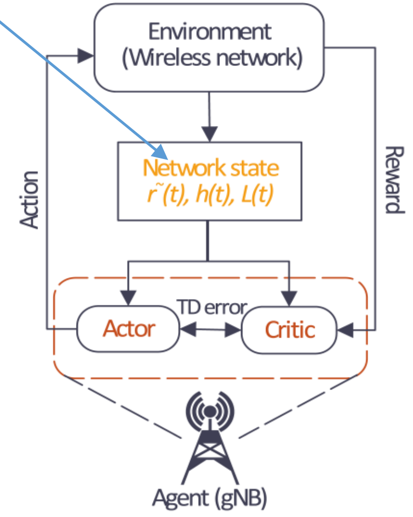
```
Initialization: Set i = 0, ϵ1, ϵ2, ϵ3 > 0, and find initial
    feasible solutions (x^(0), p^(0), w^(0));
Decompose P into P1, P2, and P3;
Relax P1 and P3 to a concave problems;
repeat
    Compute x^(i+1) from (15), (14) at given p^i, and z^i;
    Compute p^(i+1) from (17) at given i^(i+1), and z^i;
    Compute w^(i+1) from (23) at given x^(i+1), and p^(i+1);
    i = i + 1;
until ‖ x^(i+1) − x^i ‖ ≤ ϵ1, and ‖ p^(i+1) − p^i ‖ ≤ ϵ2,
    and ‖ w^(i+1) − w^i ‖ ≤ ϵ3;
Compute x* from (14) based on x^(i+1);
Set p* = p^(i+1) and z* = M × w^(i+1);
Then, set (x*, p*, z*) as the desired solution.
```

the data rate of eMBB/URLLC channel gain/total number of URLLC packets

e Temporal-Difference (TD)

2. URLLC Resource Scheduling Phase:

- The URLLC scheduling obtained by the DRRA algorithm may violate the URLLC reliability constraint at the worst-case conditions due to the relaxation applied to the probability constraint.
- In practice, URLLC traffic is random and sporadic; thus, it is necessary to dynamically and intelligently allocate resources to the URLLC traffic by interacting with the environment.
- Therefore, we propose a DRL-based algorithm to tackle the dynamic URLLC traffic and channel variations.



- To handle the slow convergence issue of the DRL, we propose a policy gradient based actor-critic learning (PGACL) algorithm that can learn policies by combining the policy learning and value learning with a good convergence rate.
- Moreover, at the initial start, **we leverage the URLLC scheduling results obtained by the DRRA algorithm in the eMBB resource allocation phase to train the PGACL algorithm and improve its convergence time.**
- Hence, combining the advantages of the DRRA and PGACL algorithms (DRRA-PGACL) provides a reliable and efficient resource allocation approach.

Proposed Solution Approach

- Considering the requirements of eMBB and URLLC services, we formulate the reward function as:

$$R(\boldsymbol{a}(t), \boldsymbol{s}(t)) = g(t) + \phi(t)\mathbb{E}\left[\sum_{n=1}^{N} r_n^u(t) - \zeta L(t)\right], \quad (4)$$

- The $\emptyset(t)$ is a time-varying weight that ensures the URLLC reliability over time slots where the network states change dynamically.
- The experience pool of the proposed PGACL algorithm is initialized according to the current optimal solution obtained by the DRRA algorithm.

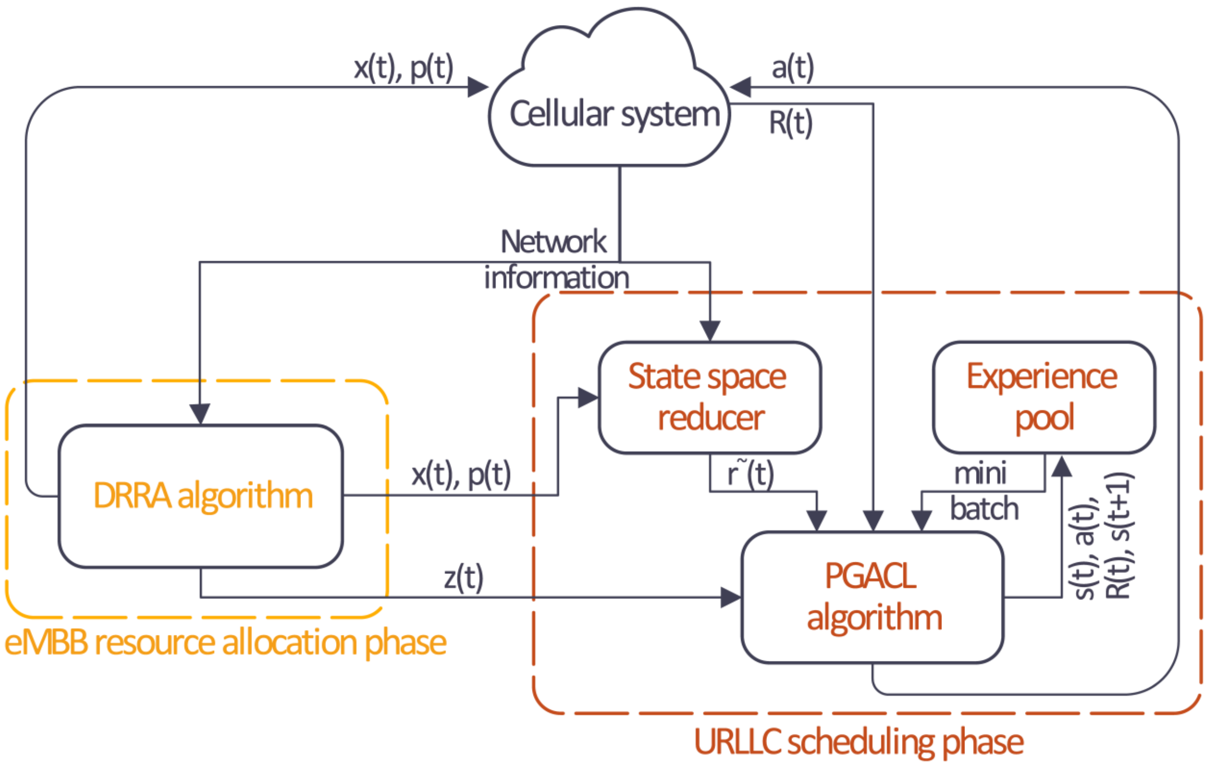


Figure 4: Block diagram of the proposed DRRA-PGACL framework.

x,p,z: RBs allocation variable, Power allocation variable, puncturing variable

1. Performance analysis of the DRRA algorithm

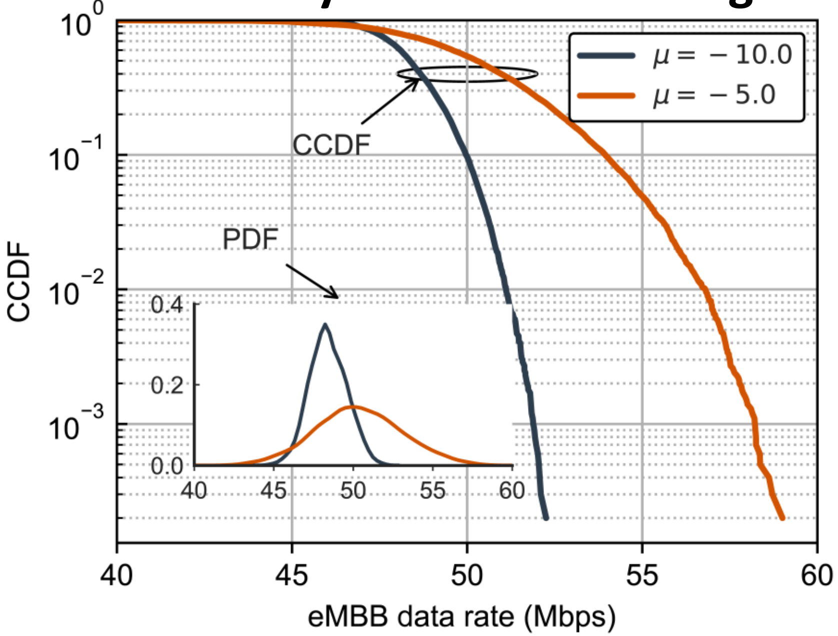
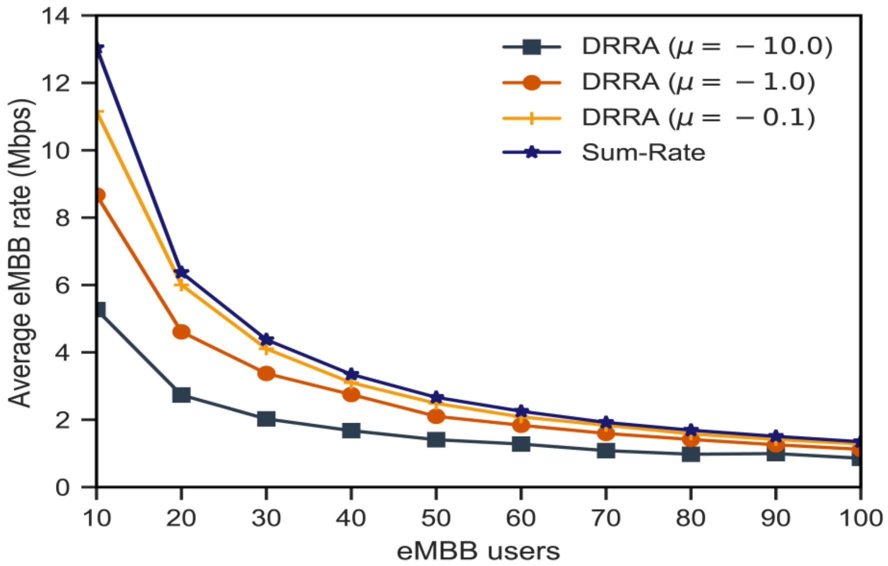
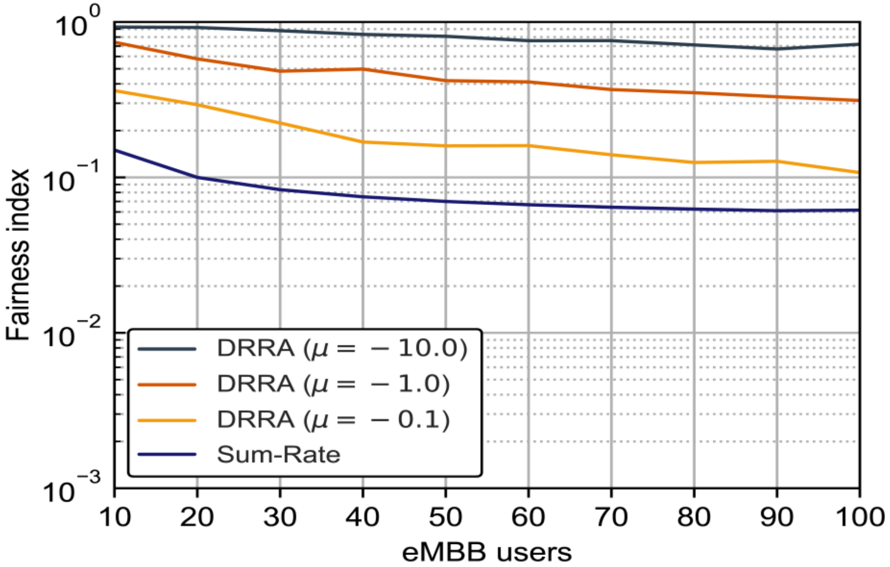


Figure 7: CCDF and PDF of the sum eMBB data rate for different values of μ





- Complementary cumulative distribution function (CCDF) and the probability density function (PDF) of the eMBB data rate calculated over time for different values of μ.
- Setting μ to higher negative values degrades the eMBB sum data rate while reducing its variance which leads to more stable and reliable eMBB transmissions over time.
- The average eMBB sum data rate is around 50 Mbps and it varies from 40 Mbps to 60 Mbps when μ = −5.0.
- Setting μ = −10.0 gives data rate between 45 Mbps to 52 Mbps resulting in a stable eMBB transmission.

2. **Convergence analysis of the PGACL algorithm and URLLC reliability analysis**.

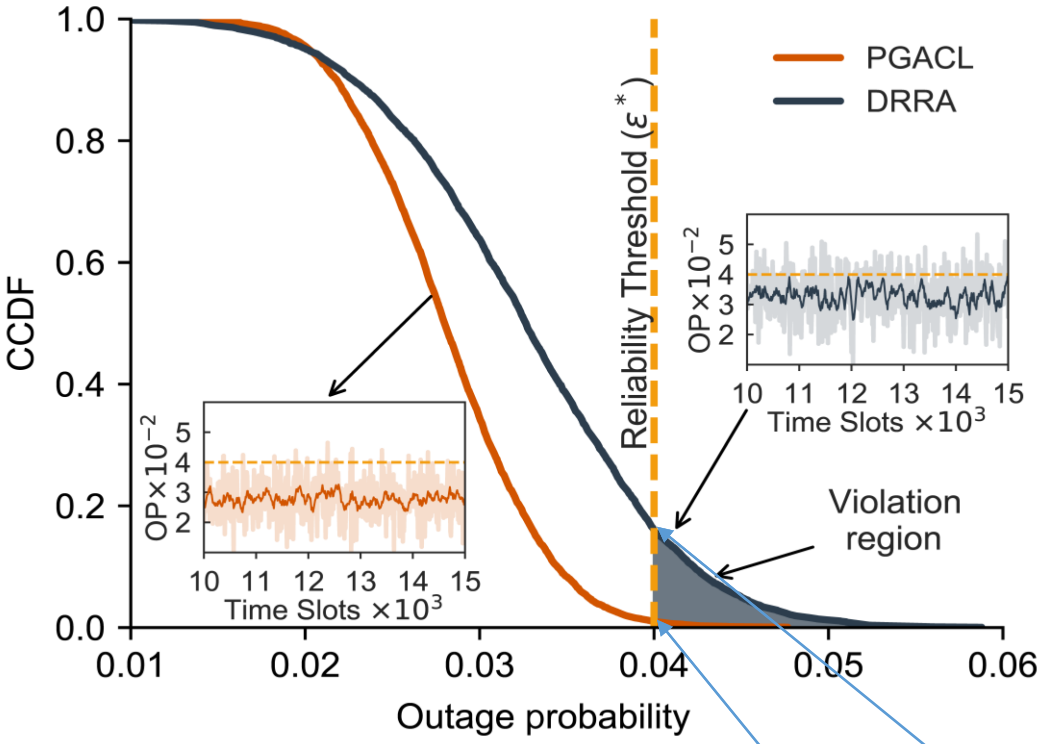
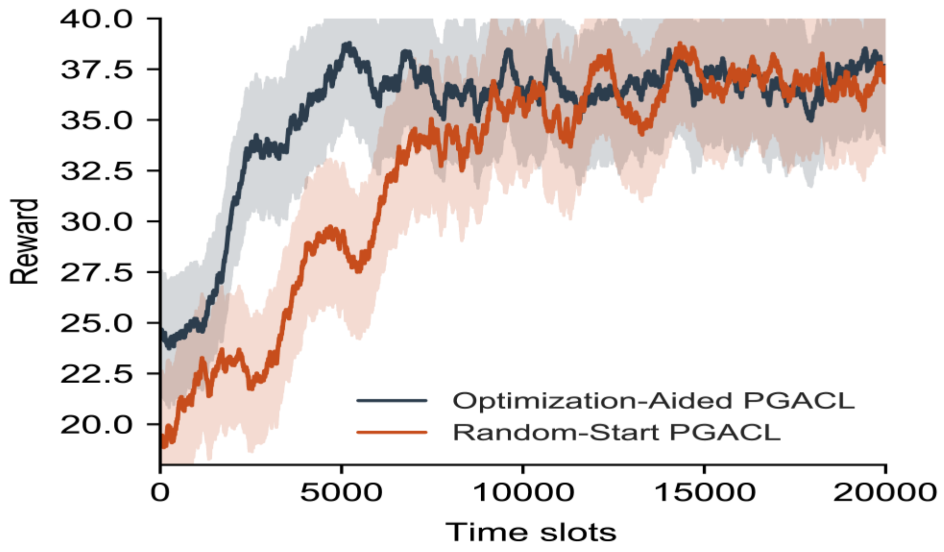


Figure 7: CCDF of the outage probability.

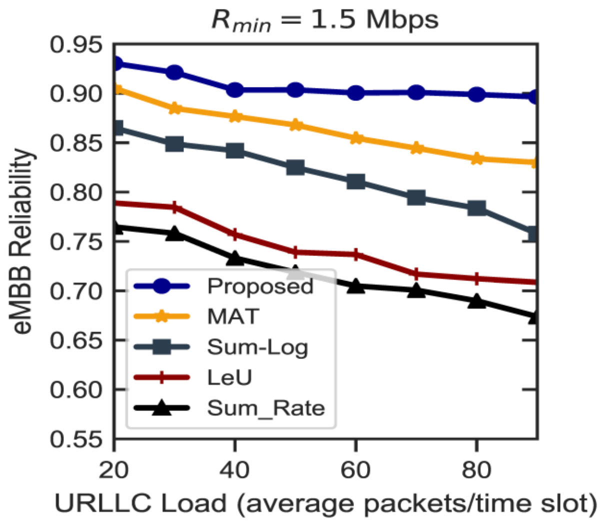


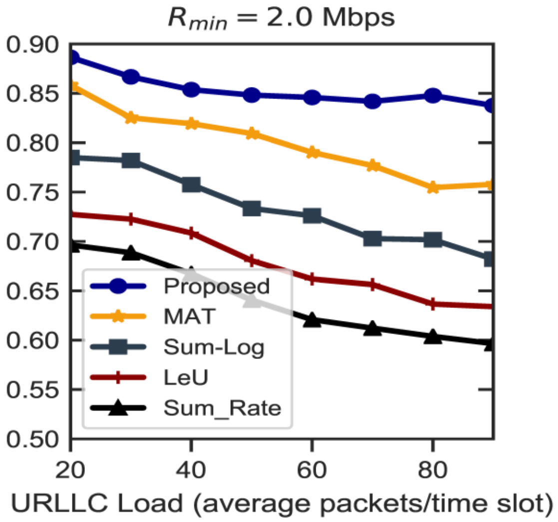
Incurring a worse performance at the beginning when initializing it with a random data and improves over time.

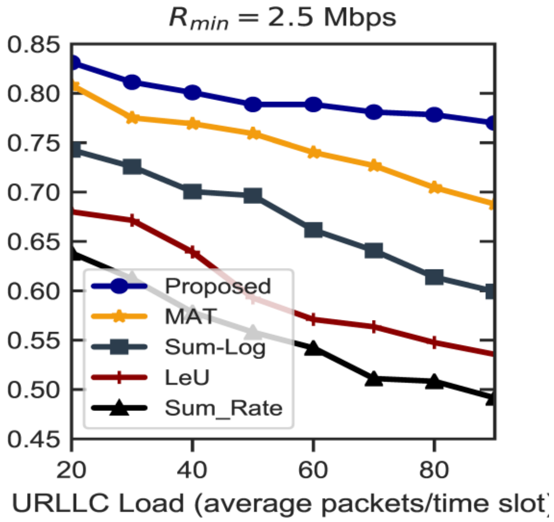
PGACL algorithm minimizes the tail-risk of the URLLC outage probability

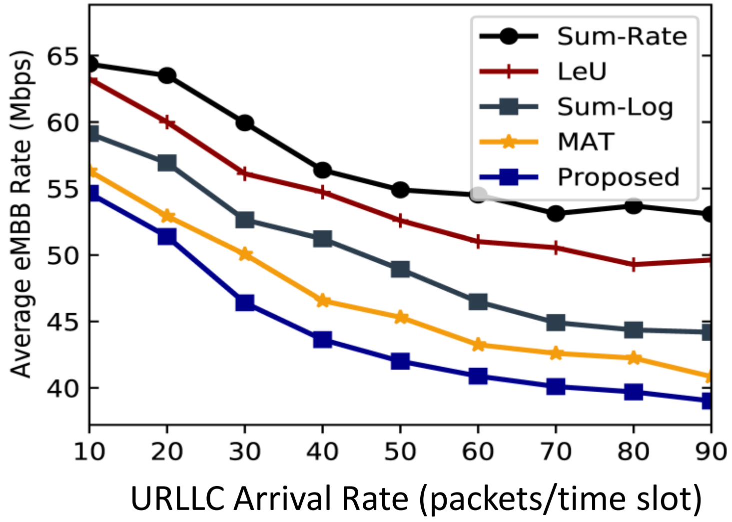
A violation probability around 0.18 when setting reliability threshold = 0.04

3. **eMBB performance analysis**:









Conclusion

Conclusion

- Deep Neural Networks are envisioned to fill this gap and serve as key predicting enabler to support the 5G networks
- Network Management coupled with AI will be defining the future of wireless networks

Challenges

- AI-Enhanced Optimization in More Complex Admission Control Scenario
- Cooperative Game with Distributed Learning

Thanks !!!

Q & A