# Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems

# This paper

Written by:-

Pengfei Wang , Student Member, IEEE,

Chao Yao , Member, IEEE,

Zijie Zheng , Student Member, IEEE,

Guangyu Sun, Member, IEEE

,Lingyang Song , Senior Member, IEEE

Presented by:-

Kapil Kumar Israni 17104011

Ayush Nagar 17104012

Akshara Nigam 17104018

Group 4

# Introduction and Motivation

With the advent of technology, the idea of interconnecting various devices intelligently has broadened the attention of both industrial & academic fields. This bloom is called Internet of Things(IoT). Generally, the IoT is defined as the network of interconnected devices embedded with electronics and sensors. The potentialities offered by the IoT enable the development and the automation of a huge number of applications in the fields of transportation, healthcare, smart environment etc. With such an exponential growth and demand it is predicted that there will be more than 50 billion IoT devices connected to the Internet by 2020.

Countless connected IoT devices will generate the massive data continuously, resulting in two main challenges :-

1) Large amount of raw data and computing tasks need to be processed, while the computing capacity of each IoT device is limited.

2) A huge volume of data needs to be transmitted through the network with a low latency to fulfill the requirements of the real-time tasks while both the wireless and the wired transmission resources are inadequate in the networks.

With these challenges in mind the motivation of the paper is to Jointly allocate the computational tasks by distributing it among the mobile edge computing systems and also reducing the latency of response and reducing the load at the IoT device.

# (A) Basic Concept in Cloud and Edge

The idea of introduction of Cloud computing in IoT devices was to take use of the strong computing capacity in the data centers to process the data delivered from these device. However, it is not scalable and efficient for these devices as they face many challenges:-

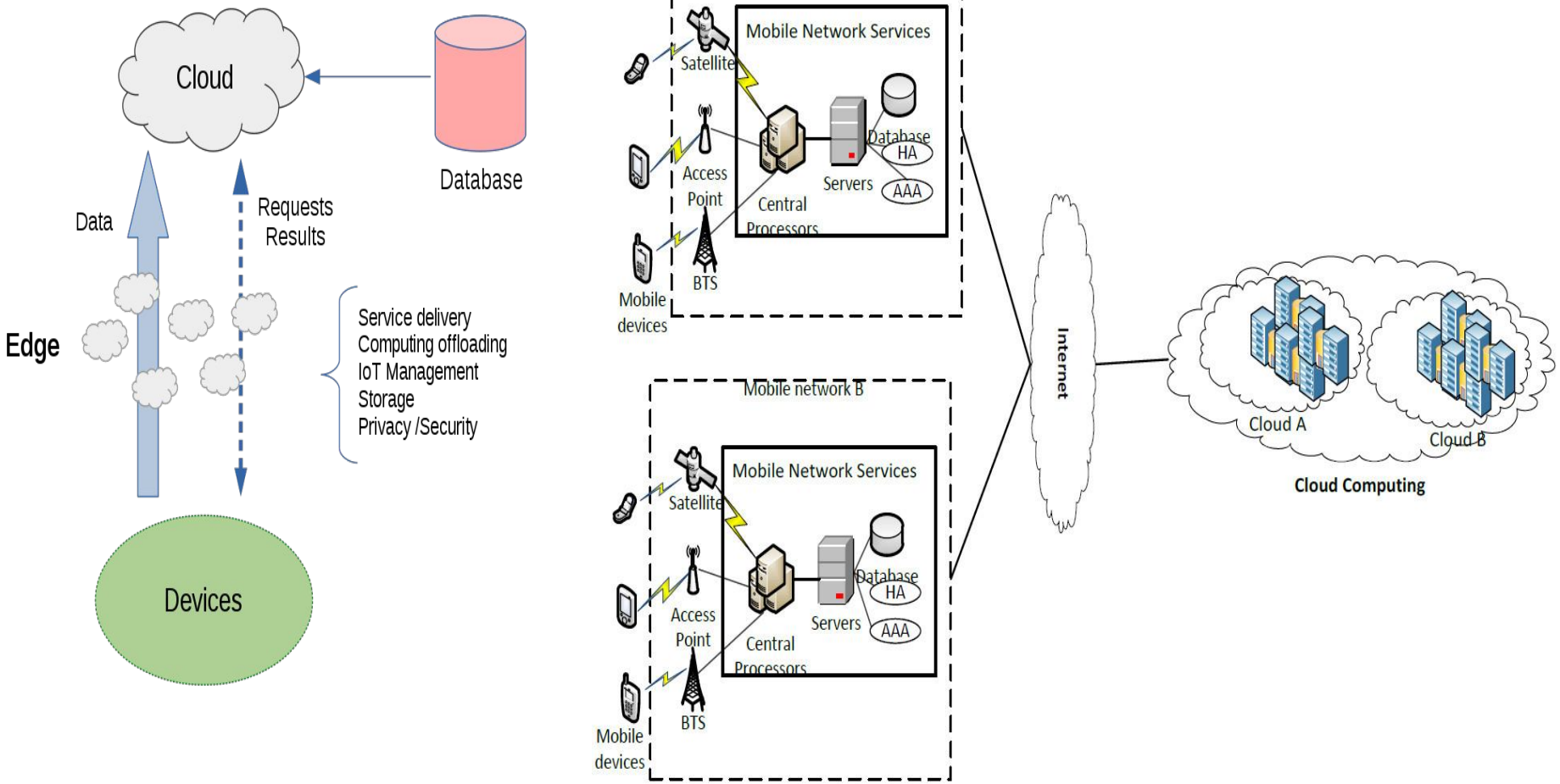
1. Cloud computing usually needs a long link to deliver large quantities of raw data from the IoT devices to the cloud center(CC), which results in huge transmission pressure over the limited frequency bandwidth. This may not fulfill some IoT applications. The latency of the cloud computing is too large to transmit huge amount (maybe in Gigabytes).

2. The transmission time between the IoT devices and the Cloud.

The Mobile Edge Computing (MEC) has been, defined as providing Internet service environment and enabling the computation to be performed at the edge of the mobile network, where the term “edge” refers to any computing and network resources between data sources and CCs. The edge has the computing capacity, offering an opportunity to offload part of the computing tasks from the CC to the edge, which can evidently help to reduce the transmission time.

Cloudlet can also be used, where the computing tasks are sent to the nearest deployed servers rather than the remote CC so that the transmission delay is significantly reduced.

Thus MEC shortens the response time in cases of self assisted driving cars and provides low latency and location awareness.


Cloud
Database
Data
Requests
Results
Edge
Service delivery
Computing offloading
IoT Management
Storage
Privacy /Security
Devices
Mobile network A
Mobile Network Services
Satellite
Access
Point
BTS
Mobile
devices
Central
Processors
Servers
Database
HA
AAA
Mobile network B
Internet
Cloud A
Cloud B
Cloud Computing

# (B) Coupled Task Assignment, Transmission & Computing Resource Allocation

If solely relying on the cloud computation, it results in the additional long transmission time and huge pressure, on other hand the processing capacity of  only MEC is still  a restriction for those IoT application with huge data volume.

The three aspects, task assignment, together with transmission resource allocation and computing resource allocation jointly form the MDP Approach. It schedules the computation tasks based on the queueing and the transmitting or processing execution state.

The task assignment, transmission, and computing resource allocation are coupled with close relationship, for the task assignment decision is directly influenced by the transmission and computing resource allocation, constrained by both the transmission latency and the computation latency. Unfortunately, the existing works in cloud computing or edge computing do not jointly consider the three closely correlated aspects, where only one or two aspects are taken into account. For eg:

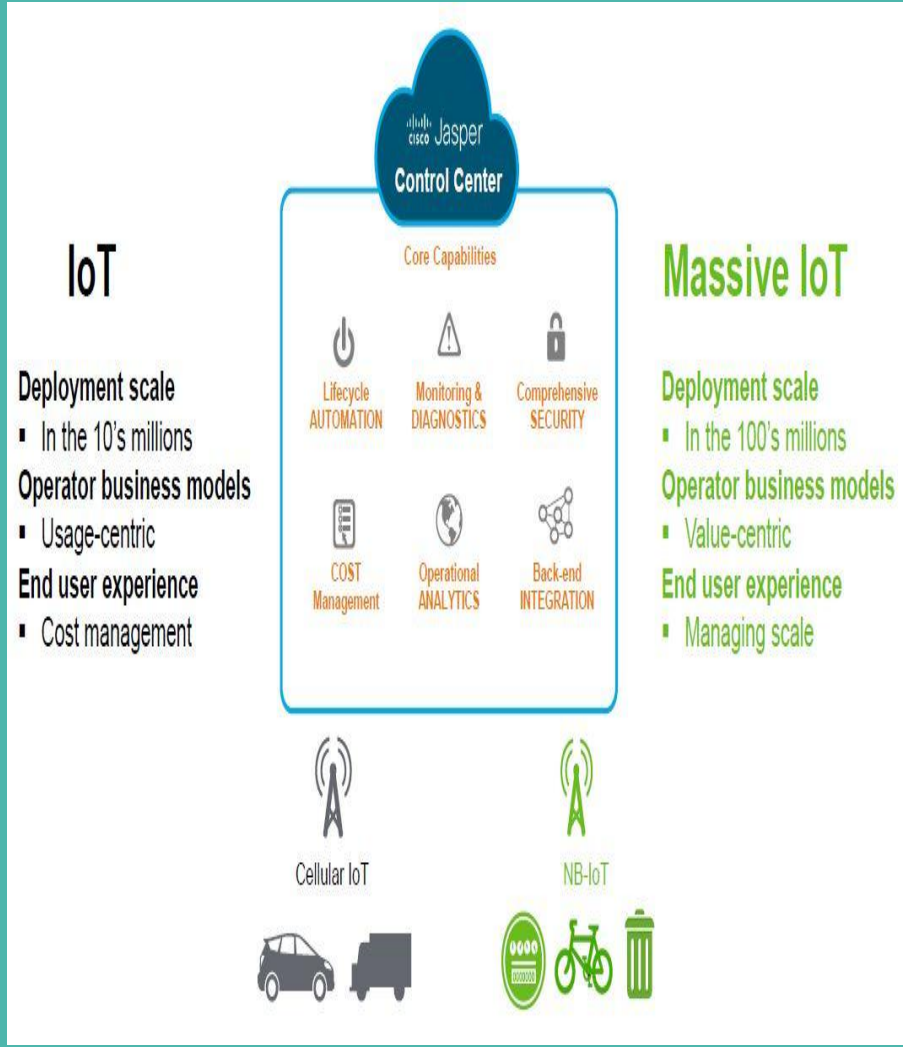
- The transmission resource allocation for multiuser mobile edge computational offloading constrained by the computation latency is studied by **C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," IEEE Trans. Wireless Communication** and  **X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing,"**and **Guo et al** discussed it in the ultradense IoT networks.
- **Ko et al** and **S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis",** analyzed the transmission latency and computation latency separately with different mobile device density, taken the task assignment and computing rate control into consideration.

Thus on combining the three aspects it solves some problems relating to huge data and limited computation restrictions.

# (C) Contributions

- Edgeflow: a multi layer data flow processing system.
- Combined the strong computing power of CC and the close distance advantage of the MEC to integrally utilize the computing capacity throughout the whole network i.e.
    - Top layer: CC
    - Middle layer: MEC Server
    - Bottom Layer: IoT EDs (Edge devices)
- Challenges to design such unified schema:-
    - Task assignment on different layers and different node are highly correlated with computing and transmission resource allocation.
    - Volume of data varies with time and IoT application.
    - Task assignment strategies and resource allocation schema need to be adjusted according to the data generation speed.
    - Due to limited computing capacity and transmission resources of network, the system may be blocked when large amount of data pours into the network which leads to complicated cases.

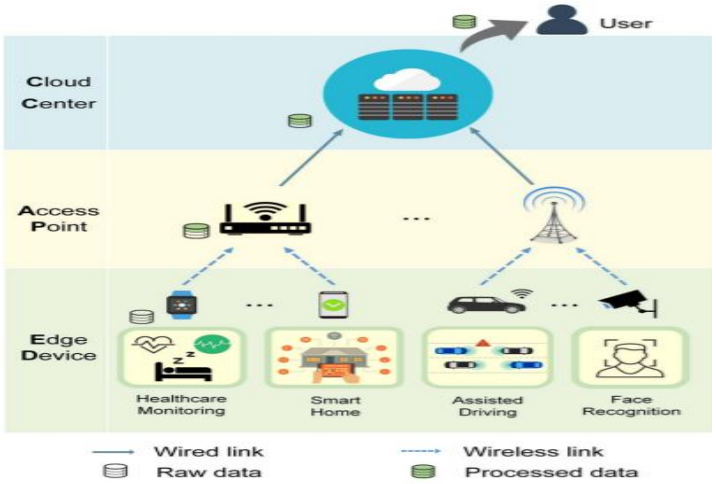
- Realizing data processing task in EdgeFlow
  - Optimally assigned the task on middle layer.
  - Wireless transmission between EDs and MEC servers.
  - Wired transmission between the server and the CC.
  - Proved system will be classified in two states:- non-blocking system and blocking system.
  - Proposing a latency minimization algorithm for a non blocking system.
  - Recovery time is through min-max problem for blocking system.



# Proposed Model

**Assumptions :-** Consider a general communication network, with one CC, n APs, and m EDs. The EDs connect with the AP via wireless communication, while the APs connect with the CC through the wired links. Also it is assumed that each ED can connect at most one AP, and each AP can connect at most one CC. Each node in the network possesses a certain amount of computing capacity. The raw data is generated at the EDs in the IoT applications and the results of the data processing is collected at the CC.

The downlink of the network is not considered in this scenario, since it is not necessary for the IoT. In IoT applications, focus is on the collection and processing of the data generated at the by these devices  i.e., the uplink. The processing of the raw data can be performed at any layer from the EDs to the CC. Moreover, once the data is processed at the edge of the network, i.e., at the ED or AP, only the results with the smaller size rather than the raw data is forwarded to the CC.




ig. 1. Three-layer EdgeFlow architecture.

# (A) Edge Device(ED)

The EDs on the bottom layer are responsible for generating the raw data, which includes the IoT devices, like, the mobile phones, cameras, bluetooth devices and various other sensors. The ED processes part of the raw data, and delivers the rest of the raw data, together with the processing, to the AP via wireless link.

$\lambda^{ji}_{ED}$ **:-** Denotes the data generation speed of ED i connected with AP j **( $1 \le j \le N, 1 \le i \le M$ )**

**ρ :-** Denotes the compression ratio after the data processing.

$s^{ji}_{ED}$ **:-** Represents task division percentage on The ED i connected with AP j **( $0 \le s^{ji}_{ED} \le 1$ )** **(1)**

$\Theta^{j,i}_{ED}$ **:-** Represents the computing capability of ED

$\phi^{j,i}_{ED}$ **:-** Represents the transmitting capability of ED

The computing data volume is limited by its computing capacity **:-** $s^{ji}_{ED} * \lambda^{ji}_{ED} <= \Theta^{j,i}_{ED}$ **(2)**

The transmitting data volume is limited by the wireless transmitting capacity of ED i, which is closely related with the wireless transmission resources allocated by AP j where the first part is the **processing results**, and the second part represents the **remaining raw data to transmit**.

$$\rho\lambda_{\mathrm{ED}}^{j,i}s_{\mathrm{ED}}^{j,i}+\lambda_{\mathrm{ED}}^{j,i}\left(1-s_{\mathrm{ED}}^{j,i}\right)\le\phi_{\mathrm{ED}}^{j,i} \qquad (3)$$

**$\phi^j_{AP}$ :-** The **total transmitting data volume** of all EDs connected with AP j is linearly constrained by the wireless transmission resources of AP j , this can be expressed as :-

$$\sum_{i=1}^{M} \phi_{\mathrm{ED}}^{j,i} \leq \phi_{\mathrm{AP}}^{j}. \qquad (4)$$

**Remark :-** The above constraint can be described as some wireless resources that influence the wireless data rate in a linear manner, for example, the spectrum and time resources. The power and the antenna resources cannot be modeled and discussed in the similar way, which are left in the future works.

# (B) Access Point (AP)

Being the middle layer of the three-layer Edge Flow model, APs, including base stations, WiFi and so on, receive the raw data and the products from the connected EDs. After processing part of the receiving raw data, the AP forwards the rest raw data together with the processing results of both the AP and EDs to the CC.

Considering that the raw data generated at ED i is transmitted to AP j, the equivalent raw data arriving speed at AP jcan be calculated by:-

$$\lambda_{\mathrm{AP}}^{j,i} = \phi_{\mathrm{ED}}^{j,i} \cdot \frac{1 - s_{\mathrm{ED}}^{j,i}}{1 - s_{\mathrm{ED}}^{j,i} + \rho s_{\mathrm{ED}}^{j,i}}$$

- $\phi_{\mathrm{ED}}^{j,i}$: Total data arriving speed to AP j from ED i,
- $s_{\mathrm{ED}}^{j,i}$: The raw data arriving speed. The raw data volume transmitted to AP j from ED i
- $\rho s_{\mathrm{ED}}^{j,i}$: The processed data volume

Accordingly, the processed data transmitted from ED i to AP j can be expressed by:-

$$\beta_{\mathrm{AP}}^{j,i} = \phi_{\mathrm{ED}}^{j,i} \cdot \frac{\rho s_{\mathrm{ED}}^{j,i}}{1 - s_{\mathrm{ED}}^{j,i} + \rho s_{\mathrm{ED}}^{j,i}}.$$

The AP can also process part of the received raw data :-

$$0 \le s_{\mathrm{AP}}^{j,i} \le 1.$$

$s_{\mathrm{AP}}^{j,i}$: The task division percentage of AP j for the data from ED i ($1 \le j \le N$, $1 \le i \le M$),

The computing capacity and the wired transmitting capacity of AP j for the task from ED i per unit time is denoted $\Theta^{j,i}_{AP}$ and $\phi^{j,i}_{AP}$, respectively.

The computing data volume of AP j for the task from ED i is limited by the computing capacity allocated to the ED:- $\lambda^{j,i}_{\mathrm{AP}} s^{j,i}_{\mathrm{AP}} \leq \theta^{j,i}_{\mathrm{AP}}.$

Moreover, the total computing capacity allocated to different EDs is no larger than the computing capacity of AP j, expressed by :-

$$\sum_{i=1}^{M} \theta^{j,i}_{\mathrm{AP}} \leq \theta^{j}_{\mathrm{AP}}$$

The transmitting data volume of AP j is limited by its wired transmitting capacity, which is closely related with the wired transmission resources allocated by the CC is given by :-

$$\rho\lambda^{j,i}_{\mathrm{AP}} s^{j,i}_{\mathrm{AP}} + \lambda^{j,i}_{\mathrm{AP}}\left(1 - s^{j,i}_{\mathrm{AP}}\right) + \beta^{j,i}_{\mathrm{AP}} \leq \phi^{j,i}_{\mathrm{AP}}.$$

Processed data of AP

Remaining Raw data to transit.

Processed data delivered from ED's

All the three parts need to be transmitted to the CC, which is limited by the allocated wired transmitting capacity $\phi^{j,i}_{AP}$ of AP j. Moreover, the total transmitting data volume of all APs is limited by the wired transmission resources of the CC, denoted by $\phi_{CC}$, which can be expressed by :-

$$\sum_{j=1}^{N}\sum_{i=1}^{M} \phi^{j,i}_{\mathrm{AP}} \leq \phi_{\mathrm{CC}}.$$

# (C) Cloud Center (CC)

The CC collects the data from APs via wired links. All raw data delivered to the CC is processed and the whole results are forwarded to the user who generates the task.

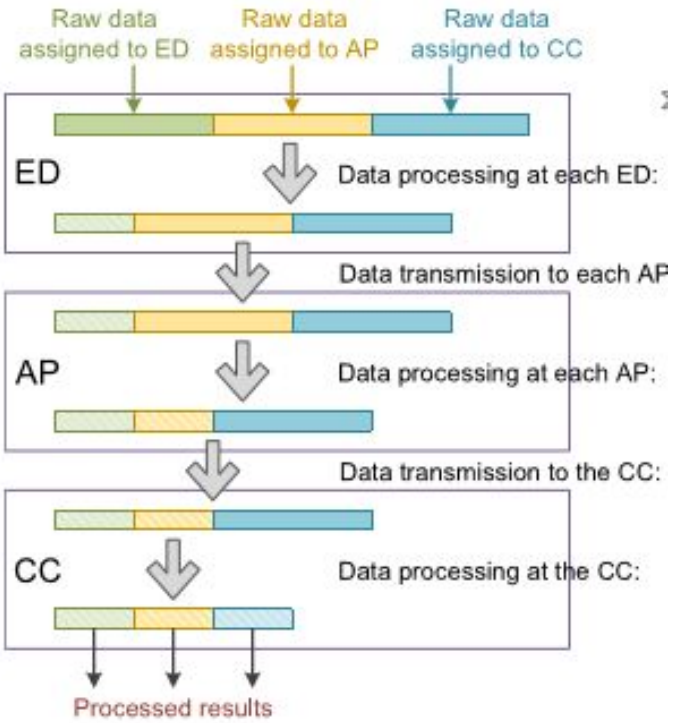
The equivalent raw data arriving speed at the CC forwarded by AP j for the task from ED i can be calculated by

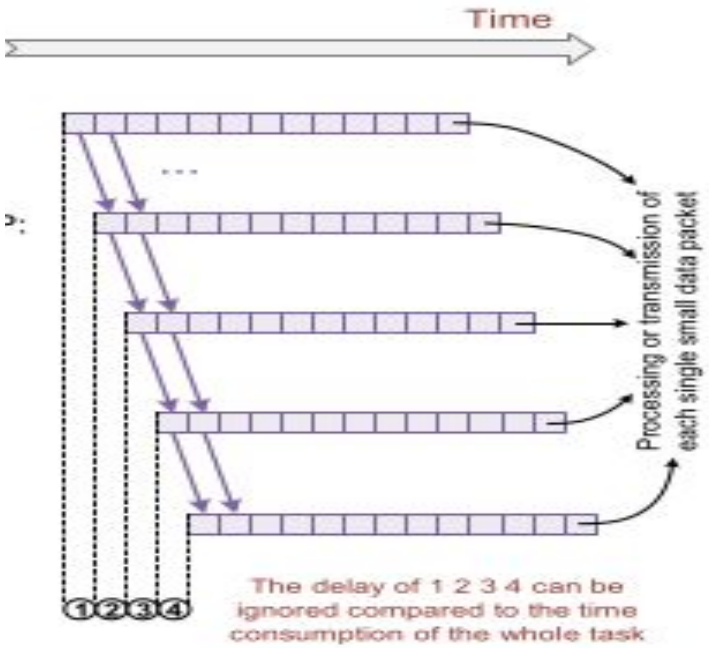
$$\lambda_{\mathrm{CC}}^{j,i} = \phi_{\mathrm{AP}}^{j,i} \cdot \frac{\left(1 - s_{\mathrm{AP}}^{j,i}\right)}{1 - s_{\mathrm{AP}}^{j,i} + \rho s_{\mathrm{AP}}^{j,i} + \frac{\rho s_{\mathrm{ED}}^{j,i}}{1 - s_{\mathrm{ED}}^{j,i}}}.$$

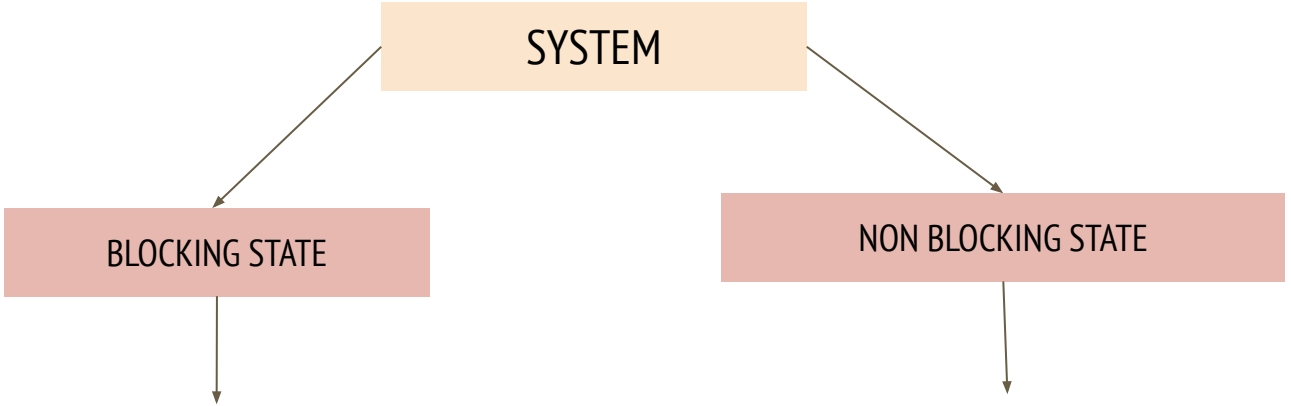
The arriving data at the Cloud Center has 3 main parts :

1. The remaining raw data.
2. The processing results of the APs.
3. The processing results of the EDs.

# Data Processing Pipeline Diagram & Time Delay






The blocking state is the state that no matter how the system adjusts its computing and transmission resources allocation or adjusts the task division on every node, the data will accumulate in the buffer of at least one node.

The Edge Flow is non- blocking if and only if the computing capacity of each layer doesn't surpass the data generation speed or that the transmission capacity does not exceed for each layer that is the ED , AP and CC.

# Non Blocking Conditions

- **Nonblocking Conditions of the ED Layer** : The ED layer blocks when the offloaded data volume surpasses the computing capacity of the ED, or the transmitting capacity of the AP is insufficient to support the data transmission from all EDs.Considering the case that all EDs fully use their computing capacity, expressed by $\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i} = \theta_{\mathrm{ED}}^{j,i},\ \forall 1 \le j \le N, 1 \le i \le M$ . The Following equations be satisfied for Non Blocking Condition :

I) $\rho\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i} + \lambda_{\mathrm{ED}}^{j,i}\left(1 - s_{\mathrm{ED}}^{j,i}\right) \le \phi_{\mathrm{ED}}^{j,i}$ II) $\sum_{i=1}^{M} \phi_{\mathrm{ED}}^{j,i} \le \phi_{\mathrm{AP}}^{j}.$ III) $\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i} = \theta_{\mathrm{ED}}^{j,i},\ \forall 1 \le j \le N, 1 \le i \le M$

- **Nonblocking Conditions of the AP Layer:** The AP layer blocks when the offloaded data volume surpasses the computing capacity of the AP, or the transmitting capacity between the AP layer and the CC is insufficient for the data transmission from all APs. Considering the case that all APs fully use their computing capacity, expressed by $\lambda_{\mathrm{AP}}^{j,i} s_{\mathrm{AP}}^{j,i} = \theta_{\mathrm{AP}}^{j,i},\ \forall 1 \le j \le N, 1 \le i \le M$ The Following equations be satisfied for Non Blocking Condition :

I) $\sum_{i=1}^{M} \theta_{\mathrm{AP}}^{j,i} \le \theta_{\mathrm{AP}}^{j}$ II) $\rho\lambda_{\mathrm{AP}}^{j,i} s_{\mathrm{AP}}^{j,i} + \lambda_{\mathrm{AP}}^{j,i}\left(1 - s_{\mathrm{AP}}^{j,i}\right) + \beta_{\mathrm{AP}}^{j,i} \le \phi_{\mathrm{AP}}^{j,i}.$ III) $\sum_{j=1}^{N}\sum_{i=1}^{M} \phi_{\mathrm{AP}}^{j,i} \le \phi_{\mathrm{CC}}.$

IV) $\lambda_{\mathrm{AP}}^{j,i} s_{\mathrm{AP}}^{j,i} = \theta_{\mathrm{AP}}^{j,i},\ \forall 1 \le j \le N, 1 \le i \le M$

- **Nonblocking Conditions of the CC Layer:** The CC need to process all remaining raw data transmitted from APs. Hence, the nonblocking conditions of computing at the CC is : $\lambda_{\mathrm{CC}}^{j,i} \le \theta_{\mathrm{CC}}^{j,i}$

# Non Blocking State: Latency Minimization

In the non blocking state, there is  a general objective in edge computing system: to minimize the system latency. The latency of a task is defined as the sum of the computing time and transmitting time from the ED layer to the CC.

$$L_{\mathrm{ED}}^{j,i} = \frac{1}{\theta_{\mathrm{ED}}^{j,i}} + \frac{\rho}{\phi_{\mathrm{ED}}^{j,i}} + \frac{\rho}{\phi_{\mathrm{AP}}^{j,i}}.$$

Latency for the data

Processing time for the data at the ED

Transmission time between the ED and AP

Transmission time between the AP and CC

$$L_{\mathrm{AP}}^{j,i} = \frac{1}{\phi_{\mathrm{ED}}^{j,i}} + \frac{1}{\theta_{\mathrm{AP}}^{j,i}} + \frac{\rho}{\phi_{\mathrm{AP}}^{j,i}}.$$

Latency for the data

Transmission time between the ED and AP

Processing time for the data at the AP

Transmission time between the AP and CC.

$$L_{\mathrm{CC}}^{j,i} = \frac{1}{\phi_{\mathrm{ED}}^{j,i}} + \frac{1}{\phi_{\mathrm{AP}}^{j,i}} + \frac{1}{\theta_{\mathrm{CC}}^{j,i}}.$$

Latency for the data

Transmission time between the ED and AP

Transmission time between the AP and CC

Processing time for the data at the CC

The system latency is the total latency of all tasks created at the EDs per unit time.

Considering the task produced at ED i, the data generation speed of which is $\lambda_{j,i}$ ED, the task division percentage at ED i, AP j and the CC is $s^{ji}_{ED}$, $s^{ji}_{AP}$ and $s^{ji}_{CC} = 1 - s^{ji}_{ED} - s^{ji}_{Ap}$, respectively. Therefore, the system latency of all links can be formulated as :-

$$L = \sum_{j=1}^{N} \sum_{i=1}^{M} \left( \lambda_{\mathrm{ED}}^{j,i} \cdot \left[ s_{\mathrm{ED}}^{j,i} L_{\mathrm{ED}}^{j,i} + s_{\mathrm{AP}}^{j,i} L_{\mathrm{AP}}^{j,i} + s_{\mathrm{CC}}^{j,i} L_{\mathrm{CC}}^{j,i} \right] \right)$$

Hence, the total latency minimization problem in the nonblocking state can be formulated as :-

$$\min_{\mathbf{s},\boldsymbol{\theta},\boldsymbol{\phi}} \; L$$

Where ,

$$\begin{aligned} L(\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{\phi}) = & \sum_{j=1}^{N} \sum_{i=1}^{M} \lambda_{\mathrm{ED}}^{j,i} \cdot \left( \frac{s_{\mathrm{ED}}^{j,i}}{\theta_{\mathrm{ED}}^{j,i}} + \frac{s_{\mathrm{AP}}^{j,i}}{\theta_{\mathrm{AP}}^{j,i}} + \frac{s_{\mathrm{CC}}^{j,i}}{\theta_{\mathrm{CC}}^{j,i}} \right) \\ & + \sum_{j=1}^{N} \sum_{i=1}^{M} \lambda_{\mathrm{ED}}^{j,i} \cdot \frac{\rho s_{\mathrm{ED}}^{j,i} + s_{\mathrm{AP}}^{j,i} + s_{\mathrm{CC}}^{j,i}}{\phi_{\mathrm{ED}}^{j,i}} \\ & + \sum_{j=1}^{N} \sum_{i=1}^{M} \lambda_{\mathrm{ED}}^{j,i} \cdot \frac{\rho s_{\mathrm{ED}}^{j,i} + \rho s_{\mathrm{AP}}^{j,i} + s_{\mathrm{CC}}^{j,i}}{\phi_{\mathrm{AP}}^{j,i}}. \end{aligned}$$

# Blocking State: Recovery Time Minimization

In the blocking state, the historical data has already accumulated in the buffer, and thus, the new generated data cannot be processed until accumulated data is processed. The primary target is naturally set to process the historical data in the buffer and let the network recover to be non blocking state as far as possible when the data generation speed slows down.The aim is to minimize the time of clearing the buffer from the perspective of the whole system.

The system latency in the blocking state is meaningless since the data is no longer the real-time data.

There are five stages for the data from the time it generates at the EDs to the time it arrives at the CC :

1. Processing at each ED : The processing time of ED i connected with AP j can be expressed by $$T_{\mathrm{ED}}^{j,i} = \frac{\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i}}{\theta_{\mathrm{ED}}^{j,i}}.$$

2. Transmitting to Each AP :The data to be transmitted to the AP includes the data processed at the ED and the remaining raw data.

   The transmitting time from ED i to AP j is denoted by $$t_{\mathrm{ED}}^{j,i} = \frac{\rho \lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i} + \lambda_{\mathrm{ED}}^{j,i}\left(1 - s_{\mathrm{ED}}^{j,i}\right)}{\phi_{\mathrm{ED}}^{j,i}}.$$

3. Processing at Each AP : The processing time of AP j is denoted by $$T_{\mathrm{AP}}^{j} = \frac{\left(\sum_{i=1}^{M} \lambda_{\mathrm{AP}}^{j,i}\right) s_{\mathrm{AP}}^{j}}{\theta_{\mathrm{AP}}^{j}}$$

4. Transmitting to the CC: The data to be transmitted from the AP to the CC consists of the data processed at the ED and AP as well as the remaining raw data. The transmitting time from AP j to the CC is represented by :

$$t_{\mathrm{AP}}^{j} = \frac{\left(\sum_{i=1}^{M} \lambda_{\mathrm{AP}}^{j,i}\right)\left(1 - s_{\mathrm{AP}}^{j} + \rho s_{\mathrm{AP}}^{j}\right) + \sum_{i=1}^{M} \beta_{\mathrm{AP}}^{j,i}}{\phi_{\mathrm{AP}}^{j}}.$$

5. Processing at the CC: The processing time of the CC is denoted by : $$T_{\mathrm{CC}} = \frac{\sum_{j=1}^{N} \sum_{i=1}^{M} \lambda_{\mathrm{CC}}^{j,i}}{\theta_{\mathrm{CC}}}.$$

> The recovery time is the longest time among the processing time at the EDs, the APs and the CC, as well as the transmission time to the APs and the CC. The recovery time can be expressed by

$$T_r = \max_{1 \le i \le M, 1 \le j \le N} \left\{ T_{\mathrm{ED}}^{j,i}, t_{\mathrm{ED}}^{j,i}, T_{\mathrm{AP}}^{j}, t_{\mathrm{AP}}^{j}, T_{\mathrm{CC}} \right\}.$$

> The recovery time minimization problem can be formulated as follows: $\min_{\mathbf{s},\boldsymbol{\theta},\boldsymbol{\phi}} \quad T_r$

> If, the recovery time is larger than a unit time, which indicates the system is blocking.Also, the allocated computing capacity or wireless transmission resources to EDs cannot surpass the those of the AP, and the allocated wired transmission resources to APs cannot surpass that of the CC.

# Non blocking State: Latency Minimization Algorithm

When the Edge Flow system is in the non blocking state, we aim to minimize the system latency, The latency can be rewritten as

$$L(s,\theta,\phi)=\sum_{j=1}^{N}\sum_{i=1}^{M}\lambda_{\mathrm{ED}}^{j,i}\cdot\left(\frac{s_{\mathrm{ED}}^{j,i}}{\theta_{\mathrm{ED}}^{j,i}}+\frac{s_{\mathrm{AP}}^{j,i}}{\theta_{\mathrm{AP}}^{j,i}}+\frac{s_{\mathrm{CC}}^{j,i}}{\theta_{\mathrm{CC}}^{j,i}}\right)+\sum_{j=1}^{N}\sum_{i=1}^{M}\lambda_{\mathrm{ED}}^{j,i}\cdot\frac{\rho s_{\mathrm{ED}}^{j,i}+s_{\mathrm{AP}}^{j,i}+s_{\mathrm{CC}}^{j,i}}{\phi_{\mathrm{ED}}^{j,i}}+\sum_{j=1}^{N}\sum_{i=1}^{M}\lambda_{\mathrm{ED}}^{j,i}\cdot\frac{\rho s_{\mathrm{ED}}^{j,i}+\rho s_{\mathrm{AP}}^{j,i}+s_{\mathrm{CC}}^{j,i}}{\phi_{\mathrm{AP}}^{j,i}}.$$

By utilizing the Cauchy–Schwarz inequality ,the above can be divided into several subproblem with the task assignment strategy s, computing capacity allocation θ, and transmitting capacity allocation φ separated. We consider that no spare computing capacity or transmission resource is left

$$L(s,\theta,\phi)\geq L_m(s)=\frac{\left[\sum_{j=1}^{N}\sum_{i=1}^{M}\left(\sqrt{\lambda_{\mathrm{ED}}^{j,i}s_{\mathrm{ED}}^{j,i}}+\sqrt{\lambda_{\mathrm{ED}}^{j,i}s_{\mathrm{AP}}^{j,i}}+\sqrt{\lambda_{\mathrm{ED}}^{j,i}s_{\mathrm{CC}}^{j,i}}\right)\right]^2}{\Theta_{\mathrm{total}}}+\frac{\left(\sum_{j=1}^{N}\sum_{i=1}^{M}\sqrt{\lambda_{\mathrm{ED}}^{j,i}\left(\rho s_{\mathrm{ED}}^{j,i}+s_{\mathrm{AP}}^{j,i}+s_{\mathrm{CC}}^{j,i}\right)}\right)^2}{\Phi_{\mathrm{wireless}}}+\frac{\left(\sum_{j=1}^{N}\sum_{i=1}^{M}\sqrt{\lambda_{\mathrm{ED}}^{j,i}\left(\rho s_{\mathrm{ED}}^{j,i}+\rho s_{\mathrm{AP}}^{j,i}+s_{\mathrm{CC}}^{j,i}\right)}\right)^2}{\Phi_{\mathrm{wired}}}$$

where $\Theta_{\mathrm{total}}=\sum_{j=1}^{N}\sum_{i=1}^{M}\theta_{\mathrm{ED}}^{j,i}+\sum_{j=1}^{N}\theta_{\mathrm{AP}}^{j}+\theta_{\mathrm{CC}}$ , $\Phi_{\mathrm{wireless}}=\sum_{j=1}^{N}\phi_{\mathrm{AP}}^{j}$ AND $\Phi_{\mathrm{wired}}=\phi_{\mathrm{CC}}$.

The above inequality is transformed into an equation if and only if the following conditions are satisfied:

$$\frac{\theta_{\mathrm{ED}}^{j,i}}{\theta_{\mathrm{ED}}^{j',i'}} = \frac{\sqrt{\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i}}}{\sqrt{\lambda_{\mathrm{ED}}^{j',i'} s_{\mathrm{ED}}^{j',i'}}} \quad , \quad \theta_{\mathrm{ED}}^{j,i} : \theta_{\mathrm{AP}}^{j,i} : \theta_{\mathrm{CC}}^{j,i} = \sqrt{\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{ED}}^{j,i}} : \sqrt{\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{AP}}^{j,i}} : \sqrt{\lambda_{\mathrm{ED}}^{j,i} s_{\mathrm{CC}}^{j,i}} \quad , \quad \frac{\phi_{\mathrm{ED}}^{j,i}}{\phi_{\mathrm{ED}}^{j',i'}} = \frac{\sqrt{\lambda_{\mathrm{ED}}^{j,i}\left(\rho s_{\mathrm{ED}}^{j,i} + s_{\mathrm{AP}}^{j,i} + s_{\mathrm{CC}}^{j,i}\right)}}{\sqrt{\lambda_{\mathrm{ED}}^{j',i'}\left(\rho s_{\mathrm{ED}}^{j',i'} + s_{\mathrm{AP}}^{j',i'} + s_{\mathrm{CC}}^{j',i'}\right)}}$$

$$\frac{\phi_{\mathrm{AP}}^{j,i}}{\phi_{\mathrm{AP}}^{j',i'}} = \frac{\sqrt{\lambda_{\mathrm{ED}}^{j,i}\left(\rho s_{\mathrm{ED}}^{j,i} + \rho s_{\mathrm{AP}}^{j,i} + s_{\mathrm{CC}}^{j,i}\right)}}{\sqrt{\lambda_{\mathrm{ED}}^{j',i'}\left(\rho s_{\mathrm{ED}}^{j',i'} + \rho s_{\mathrm{AP}}^{j',i'} + s_{\mathrm{CC}}^{j',i'}\right)}} \quad \text{for} \quad \forall 1 \le j, j' \le N, 1 \le i, i' \le M.$$

The above equations imply that the computing capacity division and transmission resources allocation, θ and φ, can be derived once the task assignment division percentage, s, is determined.

Hence, the latency minimization is converted to : $\min_{s} \; L_m(s)$

Where Lm(s) ,

$$L = \sum_{j=1}^{N} \sum_{i=1}^{M} \left( \lambda_{\mathrm{ED}}^{j,i} \cdot \left[ s_{\mathrm{ED}}^{j,i} L_{\mathrm{ED}}^{j,i} + s_{\mathrm{AP}}^{j,i} L_{\mathrm{AP}}^{j,i} + s_{\mathrm{CC}}^{j,i} L_{\mathrm{CC}}^{j,i} \right] \right).$$

# Blocking State : Recovery Time Minimization Algorithm Design

If we consider that the processed data volume of all APs and EDs equals the data volume assigned to them, the system blocks If the data volume to be transmitted of any AP or ED surpasses its transmitting capacity, or the amount of remaining raw data to be forwarded to the CC surpasses the computing capacity of the CC.

The key idea of minimising the recovery time is to make its computing time equal to transmitting time on the blocking layer.This includes minimization of recovery time using min-max problem at the bottom most layer (ED) to the top most layer(CC).

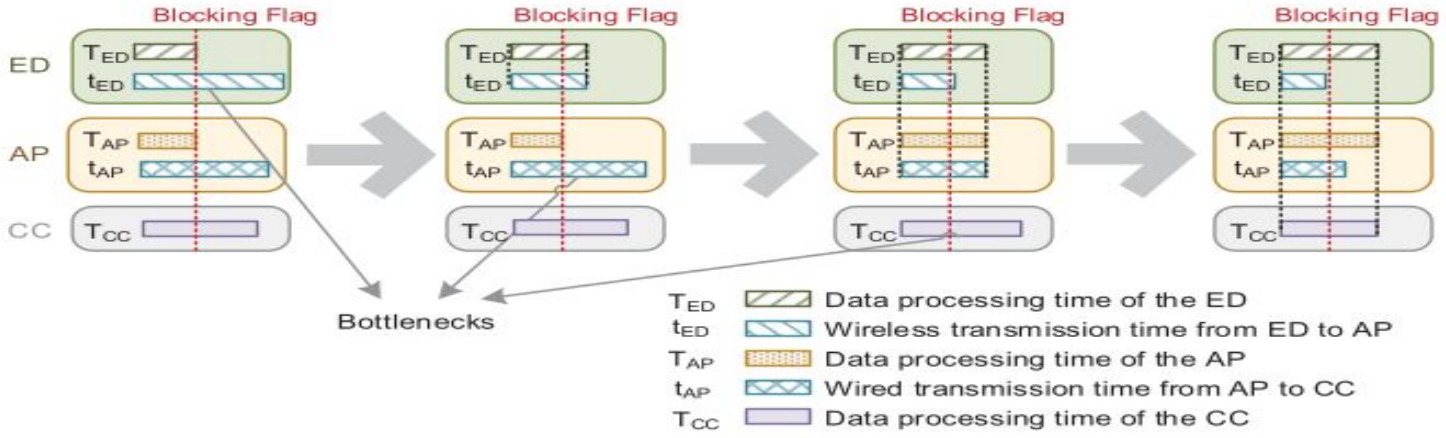



Fig. 4. Illustration of task assignment strategy in the *one ED-one AP-CC* system.

# (A) Minimizing Recovery Time on the ED Layer

When the ED layer blocks, it represents the wireless transmission resources of the AP are insufficient to transmit the processing results as well as the remaining raw data of its connected EDs,Then there is a need to adjust the task assignment strategy for the ED layer and the transmission resource  allocation of the AP to it's connected EDs.

Considering the case that the block appears between AP  j  and it connects EDs, the task division percentage is given by

$$s_{\mathrm{ED}}^{j,1} = \alpha \in [0, 1] \qquad T_{\mathrm{ED}}^{j,i} = T_{\mathrm{ED}}^{j,i'}, \quad \forall 1 \leq i, i' \leq M. \tag{1}$$

and the total wireless transmission resources of AP j is $\varphi^{j}_{AP}$ . Computing time of EDs and $AP_j$ are equals as  given in expression above.

Processing time of the data at ED

Data generating speed of ED

$$s_{\mathrm{ED}}^{j,i} = \begin{cases} \frac{\theta_{\mathrm{ED}}^{j,i} \lambda_{\mathrm{ED}}^{j,1}}{\theta_{\mathrm{ED}}^{j,1} \lambda_{\mathrm{ED}}^{j,i}} s_{\mathrm{ED}}^{j,1} = k_i \alpha, & 0 \leq \alpha \leq \frac{1}{k_i} \\ 1, & \frac{1}{k_i} < \alpha \leq 1. \end{cases} \tag{2}$$

Task assignment % of ED 2<=i<=M

As Assumed

Computing time of ED

Transmitting time of ED

$$T_{\mathrm{ED}}^{j,i} = t_{\mathrm{ED}}^{j,i}, \quad \forall 1 \leq i \leq M. \tag{3}$$

Hence

Transition time between ED and AP

Task assignment % of ED

$$\phi_{\mathrm{ED}}^{j,i} = \theta_{\mathrm{ED}}^{j,i} \frac{1 - s_{\mathrm{ED}}^{j,i} + \rho s_{\mathrm{ED}}^{j,i}}{s_{\mathrm{ED}}^{j,i}} = \theta_{\mathrm{ED}}^{j,i} f\left(s_{\mathrm{ED}}^{j,i}\right). \tag{4}$$

Processing time of data at ED

the total wireless transmission resources of AP j are fixed, that is

$$\sum_{i=1}^{M} \phi_{\mathrm{ED}}^{j,i} = \phi_{\mathrm{AP}}^{j}. \tag{5}$$

Therefore, the task assignment strategy and resources allocation scheme can be obtained by solving the simultaneous of (1), (3), (4), and (5).

# (B) Minimizing Recovery Time on the AP Layer

When the AP layer blocks, it represents the wired transmission resources of the CC are insufficient to transmit the processing results as well as the remaining raw data of all APs,Then there is a need to adjust the task assignment strategy for the ED and AP layer and the transmission resource  allocation of the AP and CC.

Task division % of each ED and AP

$$0 \le s_{\mathrm{ED}}^{j,i} \le 1, 0 \le s_{\mathrm{AP}}^{j} \le 1, \ \forall 1 \le j \le N, 1 \le i \le M. \quad (42)$$

(1)

To fully utilize computing capacity of all EDs and APs

$$T_{\mathrm{ED}}^{j,i} = T_{\mathrm{ED}}^{j',i'} = T_{\mathrm{AP}}^{j}, \ \forall 1 \le j, j' \le N, 1 \le i, i' \le M. \quad (43)$$

(2)

Computing time of EDs

Computing time of APs

Computing time of each AP

$$T_{\mathrm{AP}}^{j} = t_{\mathrm{AP}}^{j}, \ \forall 1 \le j, j' \le N. \quad (3)$$

Transmission time of each AP

$$\sum_{j=1}^{N} \phi_{\mathrm{AP}}^{j} = \phi_{\mathrm{CC}}. \quad (4)$$

Allocated wired transmission resources to AP

Allocated wired transmission resources of CC

# (C) Minimizing Recovery Time on the CC

When the CC layer blocks, it represents the computing capacity of CC are insufficient to the raw remaining data, then there is a need to adjust task assignment strategy  of the CC, AP and ED layer amd transmission resource allocation at CC and APs.

Similar, as analyzing the case of ED layer blocking,the task assignment strategy and resource allocation schema can be obtained by solving

$$0 \le s_{\mathrm{ED}}^{j,i} \le 1, 0 \le s_{\mathrm{AP}}^{j} \le 1, \ \forall 1 \le j \le N, 1 \le i \le M \quad (1)$$

$$T_{\mathrm{ED}}^{j,i} = T_{\mathrm{ED}}^{j',i'} = T_{\mathrm{AP}}^{j} = T_{\mathrm{CC}}, \forall 1 \le j, j' \le N, 1 \le i, i' \le M. \quad (2)$$

**Algorithm 2** Recovery Time Minimization Algorithm

**Input:** Computing capacity $\theta_{ED}^{j,i}, \theta_{AP}^{j}, \theta_{CC}$, wireless transmission resources of each AP $\phi_{AP}^{j}$, wired transmission resources $\phi_{CC}$, data generation speed $\boldsymbol{\lambda}$.

**Output:** Task assignment strategy $\boldsymbol{s}^*$, resources allocation scheme $\boldsymbol{\theta}^*, \boldsymbol{\phi}^*$.

**ED layer optimization:**

1: Fully utilize the computing capacity of all EDs.
2: **if** The ED layer blocks **then**
3: Make equal of the computing and transmitting time of all EDs connected with the same AP in the blocking area.
4: Update $\boldsymbol{s}^*$, $\boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$.

**AP layer optimization:**

1: Fully utilize the computing capacity of all APs and EDs.
2: **if** The AP layer blocks **then**
3: Make equal of the computing time of all APs and EDs.
4: Make equal of the computing and transmitting time of APs.
5: Update $\boldsymbol{s}^*$, $\boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$.

**CC layer optimization:**

1: Fully utilize the computing capacity of all APs, EDs and the CC.
2: **if** The CC layer blocks **then**
3: Make equal of the computing time of all APs, EDs and CC.
4: Update $\boldsymbol{s}^*$, $\boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^*$.

# Implementation Of The Experiment

The three layered edge flow system consisting of CC, AP's and the ED's are implemented on the basis of Linux System, USRP and Intel NUC.

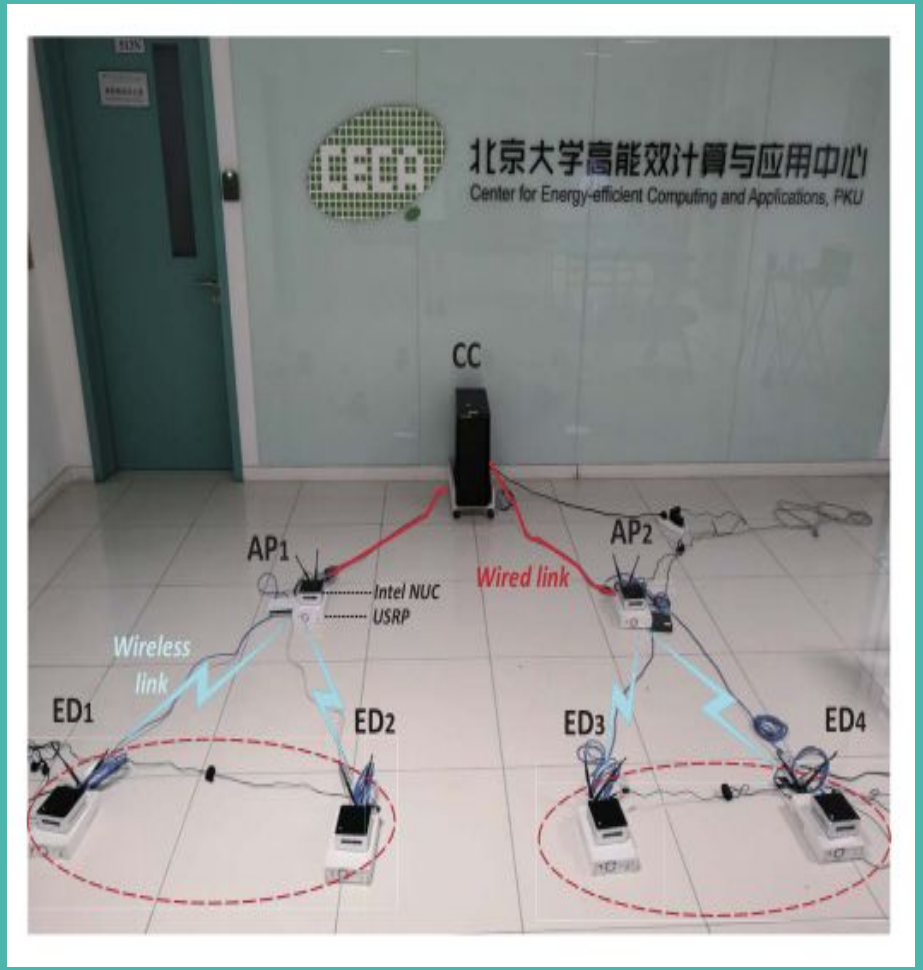
Next unit of computing (NUC) is a small-form-factor personal computer designed by Intel. The USRP is a range of software-defined radios. Most USRPs is composed with the hardware part as the radio front end, and the software part, GNU Radio, which is a free software toolkit that provides signal processing blocks of information.

One single server stands for the CC layer, and two NUC nodes communicate with it performing as two APs. With the USRP devices, each AP node connects to two ED nodes over the wireless links with limited resources.

It is based on Java and Python Environment.

Java environment :- For the task assignment and processing.

Python environment :- For the time division of multiple access resources management of the network.



**1) Network Initialization Module :-** It has 3 main functions:-

(a) Establish the network connection of the wireless channel
(b) Virtualize and manage the transmission resources for the wireless and wired links
(c) Provide the application program interface (API) to the upper framework.

**2) System Management Module :-** It includes the system initialization, logical graph establishment and management.The system is connected according to:-

(a) The communication links and
(b) Internet protocol address provided by the network initialization module.

It is also responsible for the node registration and resource configuration information updating. The nodes estimate their idle computing and transmission resources, and register on the CC. The CC can then create a logical graph of the nodes with their information of the available resources.

**3) Resource Management Module :-** The idle computing and transmission resources are evaluated in this module.

**4) Task Assignment Module :-** It targets at obtaining the task assignment strategy based on the idle resources of each node.

**5) Task Execution Module :-** It's job is the management of the task assignment strategy. According to the task assignment configuration file, which instructs the task assignment strategy of each node, the module offloads the task and manage the task queue of offloading and processing.

# Simulation And Experiment Results

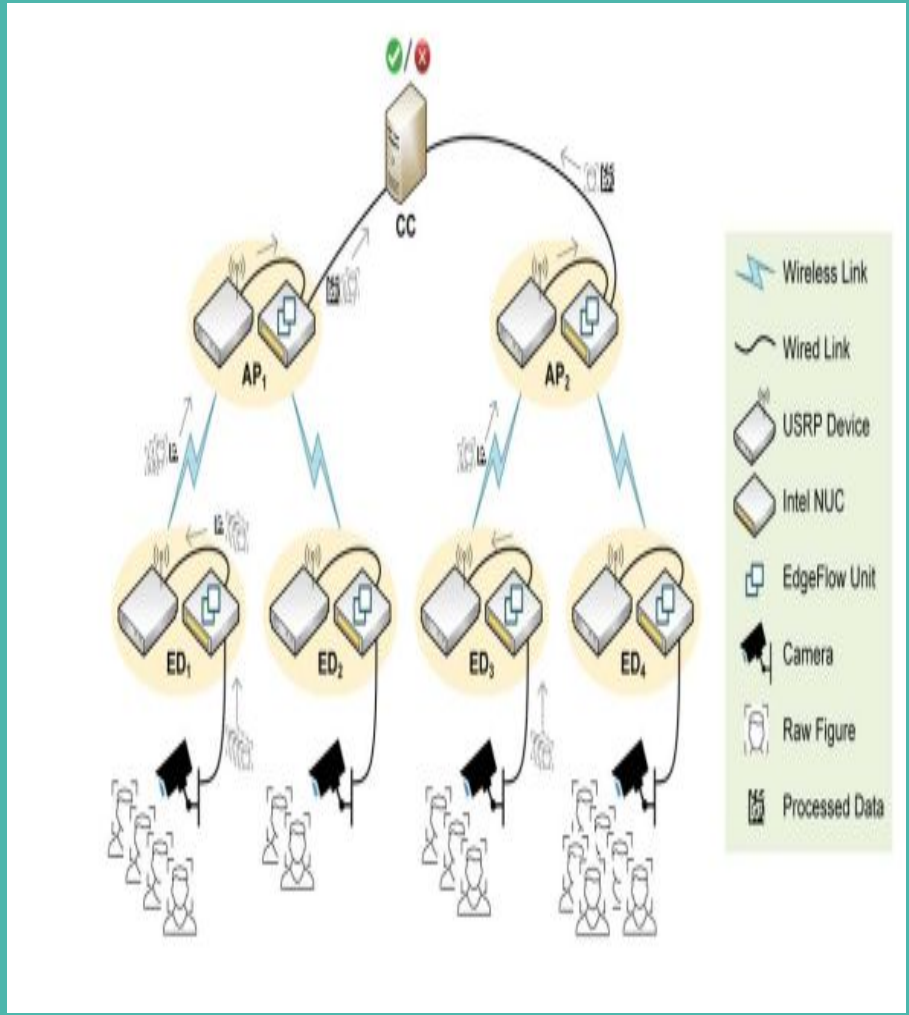
A typical example of Face Recognition System is used to simulate the algorithms and provide with the experimental results.

The evaluations are accomplished by simulating a typical IoT scenario similar to the face recognition application, which is general in the IoT sensing network, such as smart cities. Based on the face information, the computing servers can provide the intelligent service to the users.

This is discussed under two subheadings as follows:-

(a) Experiment Scenario & Setup

(b) Simulation & Experimental Results



# (A) Experiment Scenario and Setup

Each ED is connected with a camera which collects the image data. The application aims to recognize the pedestrian faces and slice out the face part, which will be delivered to the CC. After the analysis of the face part, the CC will perform the appropriate action. The face recognition is based on openCV, which is a library of Python used for Computer Vision.

Each data file represents one image of the camera, and the data generation speed λ is the number of face images captured by the camera per unit time. Moreover, in this experiment, the data generation follows Poisson process

The system performance is evaluated by the following indicators.

1) **System Latency:-** The system latency represents the response time from the data generation of the task to the end of processing at the CC.

2) **Processing Rate:-** This is the average volume of data processed by the Edge Flow system per unit time.

3) **System Robustness:-** This reflects the average number of the unprocessed packages in the network, which represents the degree of blocking in the system.

| | |
|---|---|
| The CPU frequency of the ED device | $1 * 10^9$ Hz |
| The CPU frequency of the AP device | $3 * 10^9$ Hz |
| The CPU frequency of the CC device | $1 * 10^{10}$ Hz |
| Wireless transmission resources of each AP | 300 Kbps |
| Wired transmission resources | 1 Mbps |
| The transmission power of USRP devices | 20 dbm |
| The wireless transmission bandwidth | 5 Mhz |
| The volume of the data file | 60 Kbytes |
| Compression ratio $\rho$ | 10% |

# (B) Simulation and Experimental Results

To attest the effect of the proposed algorithm, it is compared with the following solutions:-

**Local Computing:-** Each ED processes all the tasks locally and delivers the results to the CC. This is suitable for the case that the task load is light and the ED is able to process the tasks timely.

**Cloudlet:-** The ED assigns the tasks to the corresponding AP. The AP will process all tasks and deliver the results to the CC. This is suitable for the case that the tasks are *resource-intensive* and the AP possesses abundant computing and transmission resources.

**Cloud Computing:-** The input data stream is forwarded to CC directly, and all the tasks is processed centrally at the CC. This is suitable for the case that the tasks are resource-intensive, while the edge nodes (including the APs and EDs) do not possesses enough resources to process the tasks.

**MDP**:- Based on MDP, a partial assignment scheme is designed based on the queueing state, the execution state, and the transmission state to minimize the task latency.

In the experiment, we observe the average task latency with different data generation speeds. More task data requires more computing and transmission resources. This experiment evaluates the effect of the proposed strategy given different task loads.

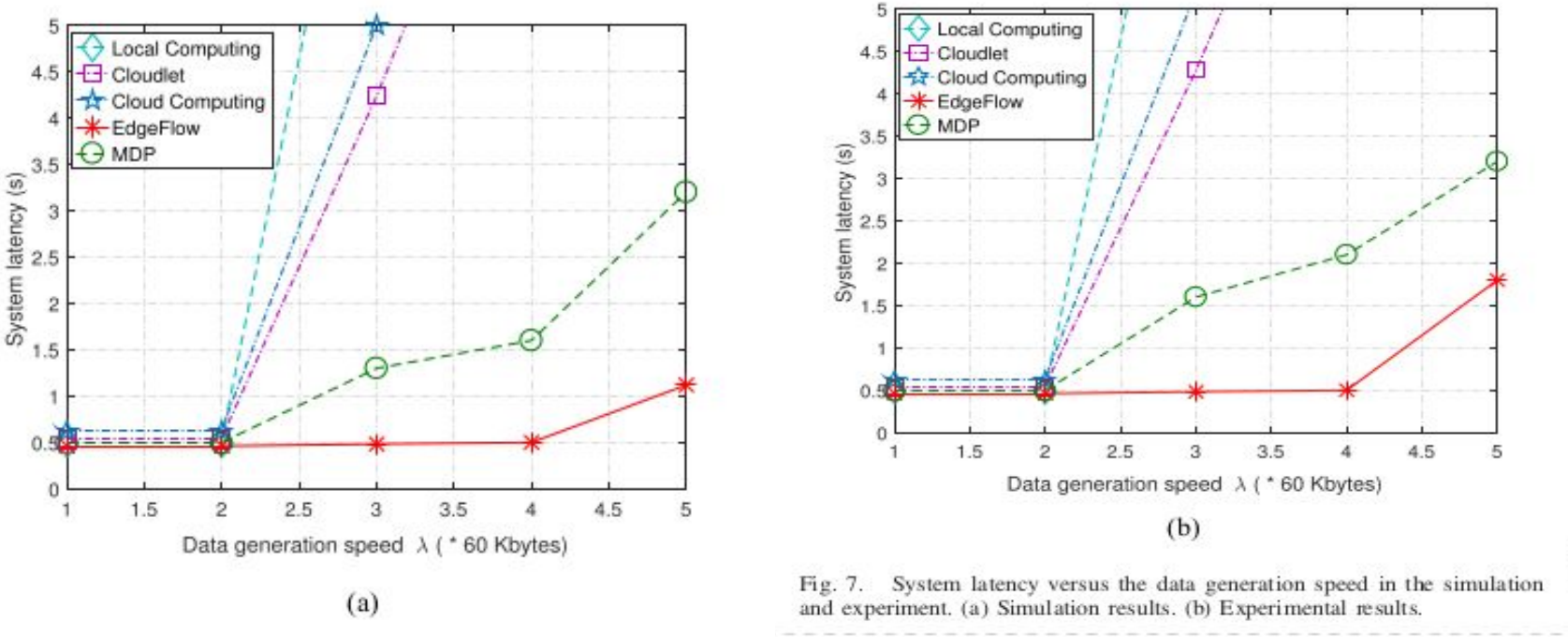



Fig. 7. System latency versus the data generation speed in the simulation and experiment. (a) Simulation results. (b) Experimental results.

On comparing the experimental and the actual results it is evident that the proposed algorithm is superior.Since the EdgeFlow system fully utilizes the computing capacity of the CC, APs, and EDs, the data processing ability is stronger than Cloudlet, local, or cloud computing, which only utilize part of the system computing capacity. Compared with other schemes, this EdgeFlow system is more tolerate to the data generation speed, that is, though the data generation speed variate, our EdgeFlow can still provide stable low-latency services.

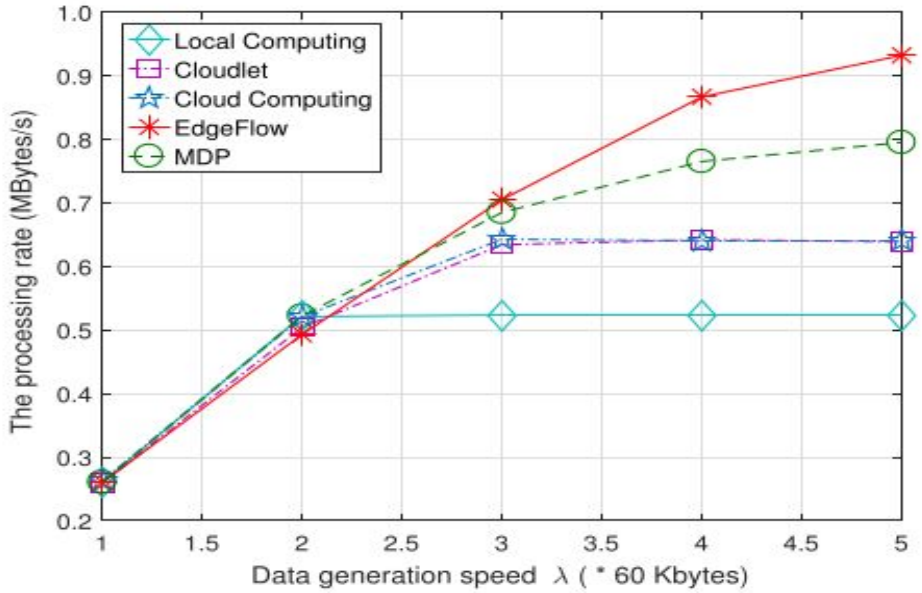



Fig. 8. Processing rate with the increase of the data generation speed on the actual experimental platform.

On analyzing the processing rate of the system with different data generation speed. When the generation speed $\lambda \leq 2$, the system does not reach the bottleneck of the computing capacity, and thus, all the schemes can process the input task timely. When the generation rate $\lambda \geq 3$, the Edge Flow system tries its best to guarantee the nonblocking condition by sharing the task overload among multiple layers, which effectively eases the accumulation of the unprocessed tasks and increases the processing rate. In other schemes, however, the data starts to accumulate in the buffer once the processing rate reaches saturation. For example, when $\lambda = 5$, the processing rate of EdgeFlow is 15% higher than the MDP scheme and 43% higher than the local computing scheme.

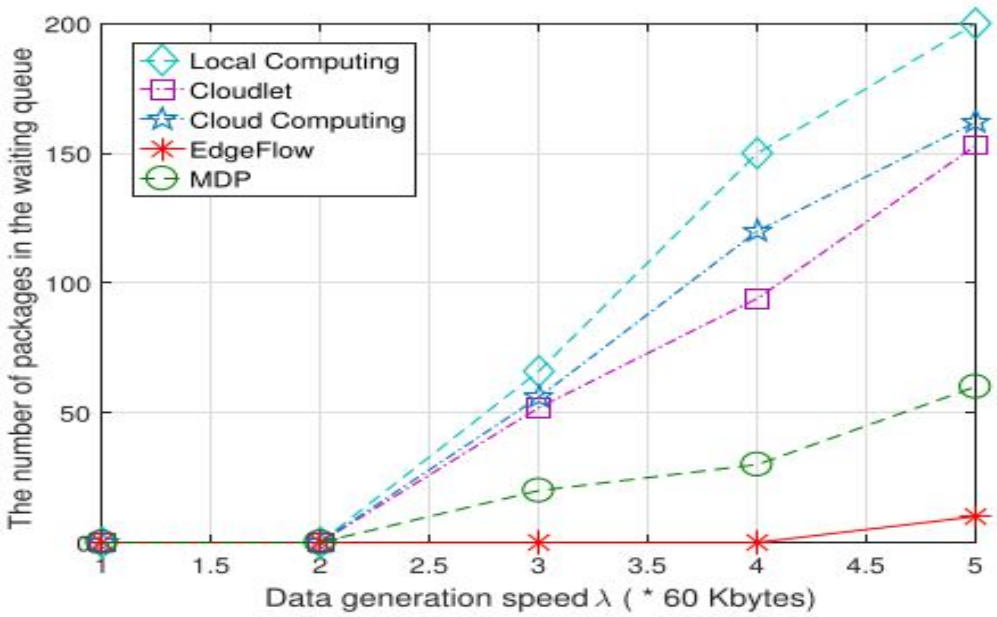
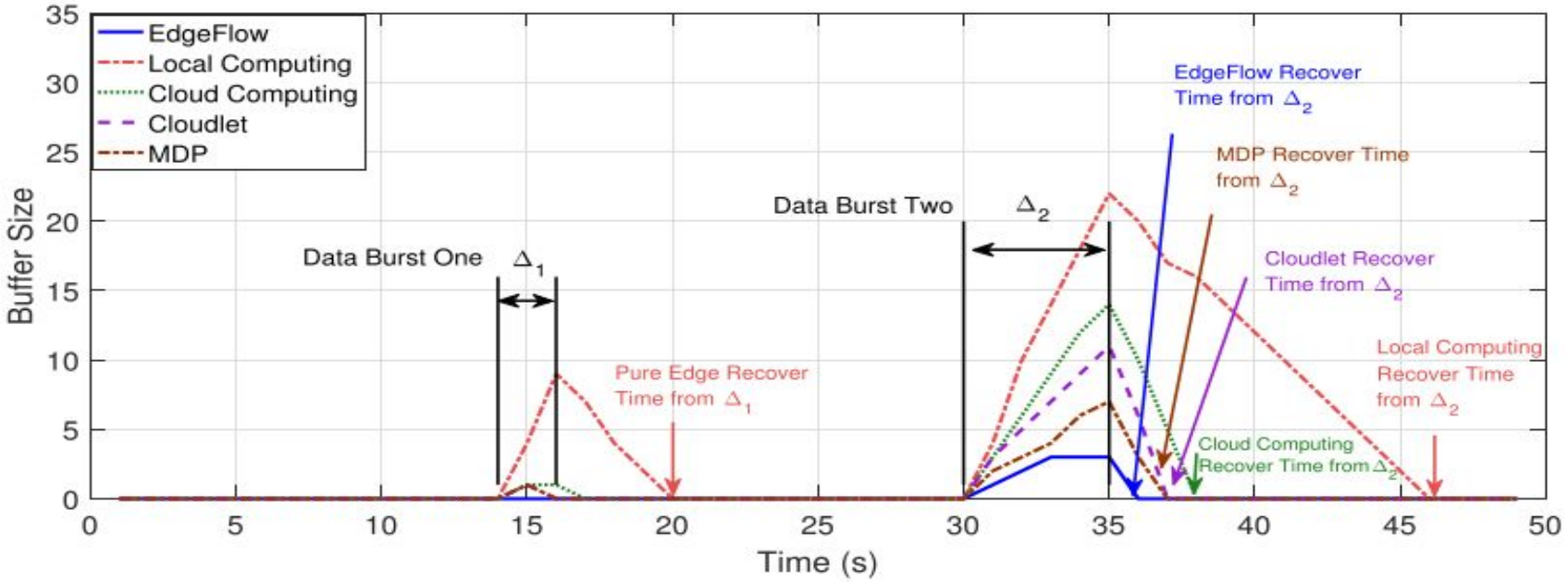



Fig. 9. System robustness with the different burden of tasks on the actual experimental platform.

On evaluation the system robustness for the tasks with heavy loads, which can be reflected by the number of unprocessed packages waiting in the buffer. The length of the waiting queue represents the degree of the blocking in the system. When $\lambda \leq 2$, the computing and transmission resources can still handle the input tasks, and thus all schemes do not result in the data accumulation. When $\lambda \geq 3$, the resources of the AP or CC cannot guarantee the stable operation of the system. The EdgeFlow system balances the computing and transmission resources in the multilayer network, which can maintain the stability of the system as much as possible. Other system, e.g., Cloudlet, offloads all computing tasks to the AP, which bring about heavy loads to wireless.




We analyze the system robustness in the perspective of the time for the system to recover from the blocking time after the data burst. At the time t = 14 s, the first burst lasting 2 s only causes the data accumulation for the local computing scheme, while the others are barely affected. After that, at the time t = 30 s, a bigger burst lasting 5 s arrives and affects all schemes. When dealing with data burst, the system suddenly turns from the nonblocking to the blocking state. Other schemes, i.e., the cloud computing, local computing, and Cloudlet, remain the same task assignment strategy, which is not suitable for the blocking state since clearing the accumulated data in the buffer becomes the primary mission. The MDP scheme adjust the task assignment strategy based on the queue state, which is hysteretic than the change of data generation speed. Compared with other schemes, EdgeFlow determines the optimal task assignment strategy based on the data generation speed and system state, and thus guarantees the smallest volume of accumulated data and the shortest recovery time, which reflects that the EdgeFlow system is most robust among these schemes, especially for the tasks with heavy loads.

# Conclusions

The proposed a multilayer data flow processing system called the EdgeFlow, which consists of the CC, APs, and the EDs. The EdgeFlow system can provide the low latency services for the IoT real-time applications via the integrated utilization of the computing capacity and transmission resource of both CC and edge nodes. The blocking and nonblocking states have been investigated and the quantitative boundary between the two states has been derived in

**Proposition:-** In the non blocking state, the system latency is minimized, while in the blocking state, the latency is meaningless for the accumulated data and the recovery time of the system is minimized. The multilayer collaborative task assignment and resource allocation strategies have been proposed In Algorithms 1 and 2 for both states to achieve the optimal solutions. The implementation of the EdgeFlow system is based on the USRPs, the Intel NUCs, and the Linux system for the typical IoT applications, face recognition. Experimental results have shown that our EdgeFlow system can obviously reduce the system latency and increase the data processing rate, especially in the case of high data generation speed. The system is able to stay in the nonblocking state by the dynamic task assignment strategy and the resources allocation when the data generation speed increases, and thus the volume of accumulated data in the buffer remains small

# Related Work

The three aspects are not considered jointly or the relationship between them is ignored. An MDP(Markov decision process) approach is proposed in [27], which schedules the computation tasks based on the queueing and the transmitting or processing execution state. The transmission resource allocation for multiuser mobile edge computational offloading constrained by the computation latency is studied in [28] and [29], and Guo et al. [30] discussed it in the ultradense IoT networks. However, the allocation of the computing resource is not taken into account. Ko et al. [31] analyzed the transmission latency and computation latency separatively with different mobile device density, taken the task assignment and computing rate control into consideration. The energy harvesting is studied in the computation latency constrained task assignment problem in [32], in order to minimizing the power consumption of the MEC server.


[27] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in Proc. IEEE ISIT, Barcelona, Spain, Jul. 2016, pp. 1451–1455.

[28] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," IEEE/ACM Trans. Netw., vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[29] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," IEEE Trans. Wireless Commun., vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[30] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultra-dense IoT networks," IEEE Internet Things J., to be published, doi: 10.1109/JIOT.2018.2838584.

[31] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," IEEE Trans. Wireless Commun., vol. 17, no. 8, pp. 5225–5240, Aug. 2018.