# Optimal Joint Offloading and Wireless Scheduling for Parallel Computing with Deadlines

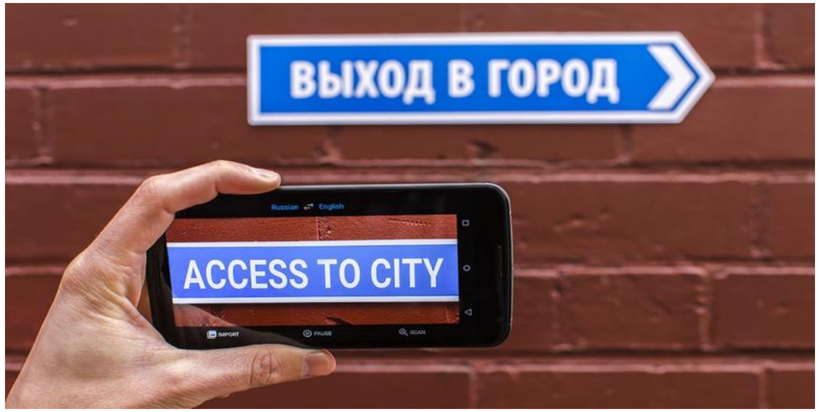
Xudong Qin* Weijian Xu† Bin Li*

*Dept. of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Rhode Island, USA

†Information Engineering College, Jimei University, Xiamen, China

# Real-time Mobile Applications


Real-time video analysis



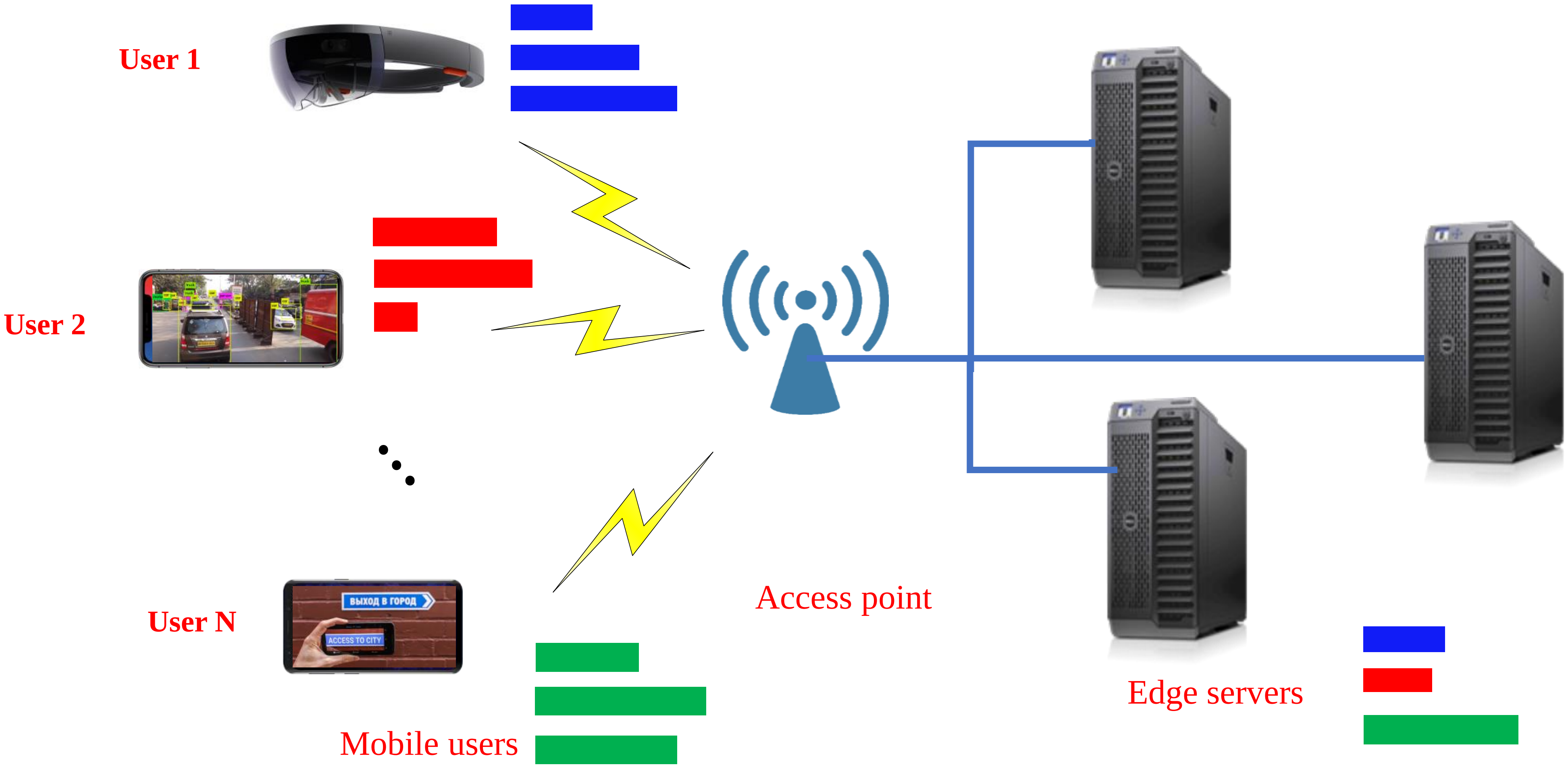

Real-time language translation

Low energy consumption

Low latency requirements

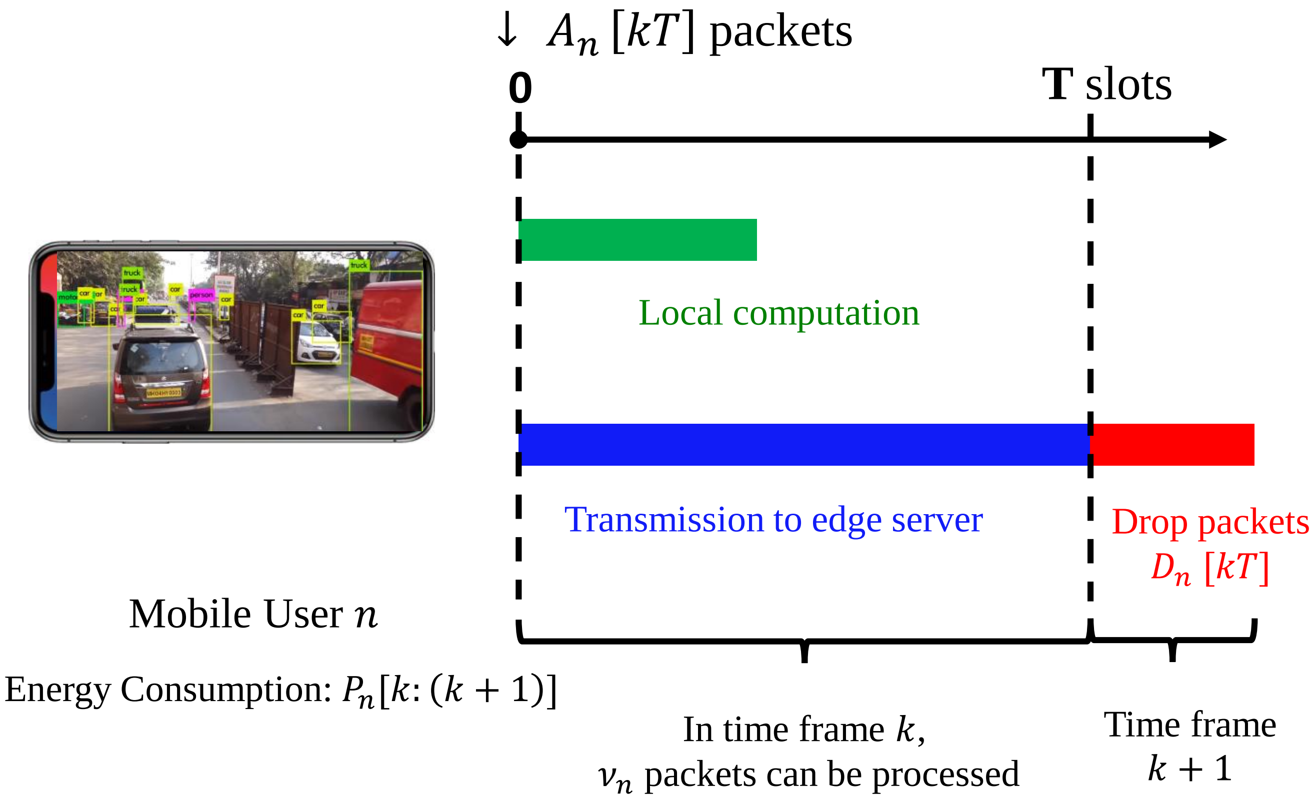
Intensive computation requirements

# System Model



# System Model (Cont')

Each user $n$ has dynamic and heterogeneous computing demands with strict $T$ time slots deadline.




$$\min \quad \limsup_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1}\sum_{n=1}^{N} \mathbb{E}[P_n[k{:}(k+1)]]$$

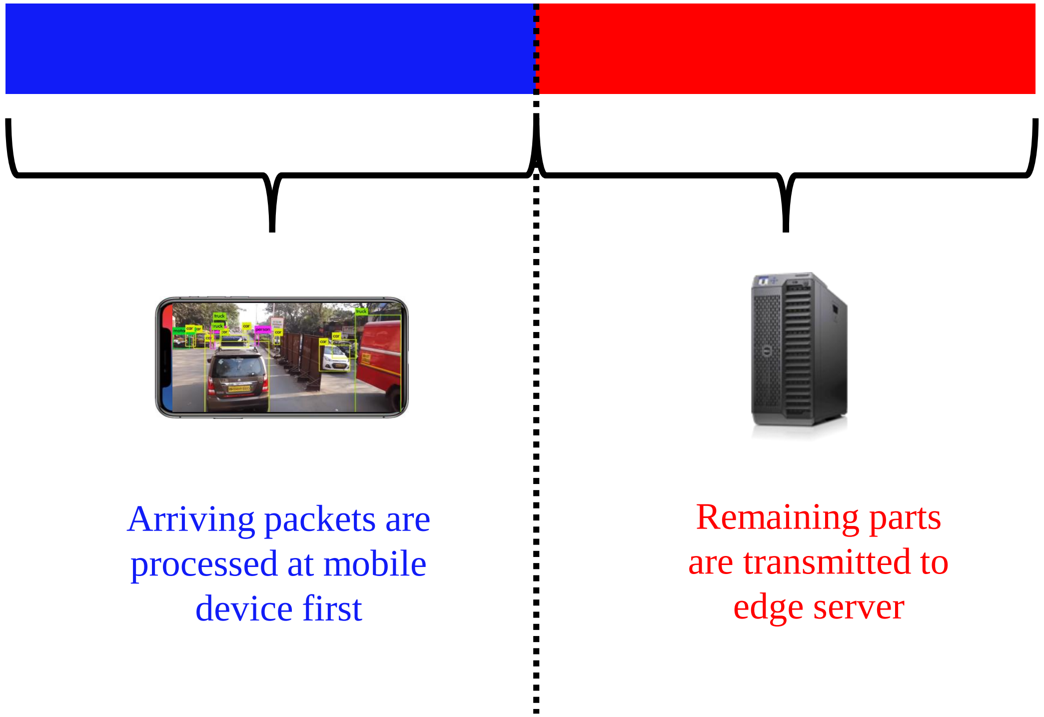
$$\text{s.t.} \quad \lambda_n(1-\rho_n) \le \nu_n, \qquad \forall n,k,$$

$$A_n^{(L)}[kT] + A_n^{(E)}[kT] = A_n[kT], \qquad \forall n,k,$$

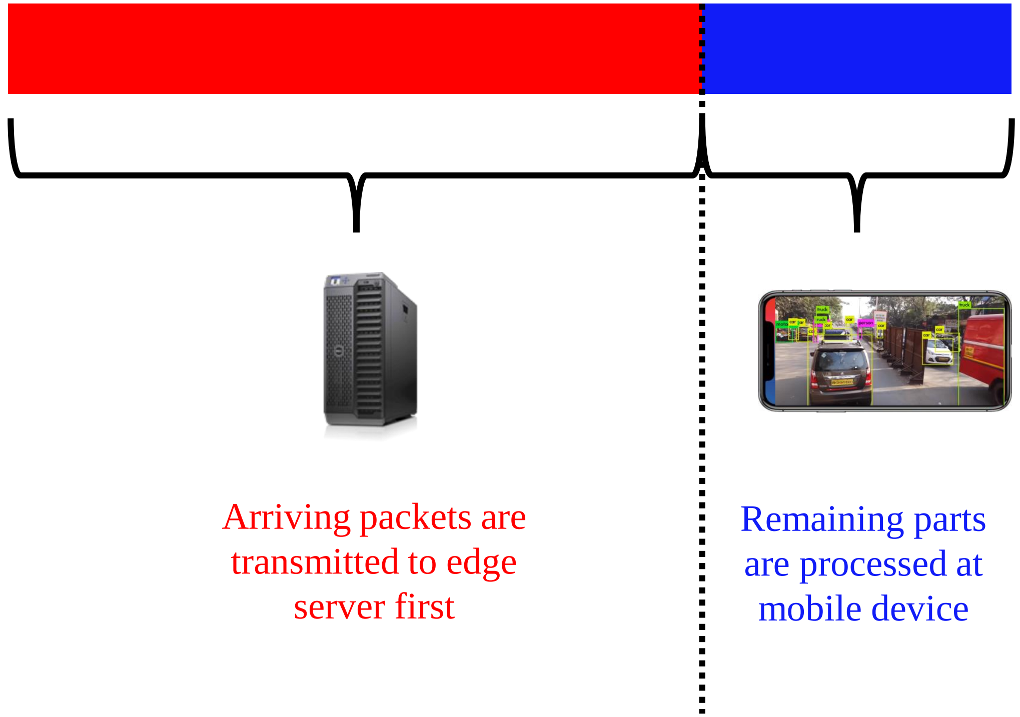
where $\mathbb{E}\big[A_n[kT]\big] = \lambda_n$, $\rho_n$ is the maximal allowable drop rate for user $n$, $\nu_n$ is the total number of packets that can be processed in frame $k$. $A_n^{(L)}[kT]$ are the packets that perform local computation and $A_n^{(E)}[kT]$ are the transmission part.

# A Motivating Example

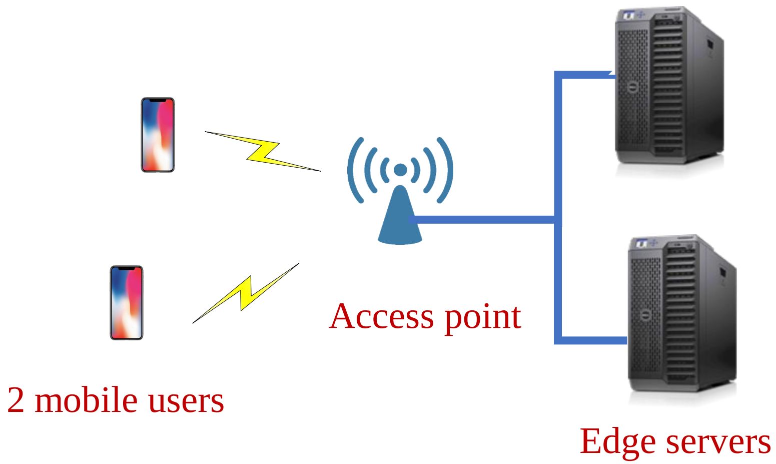
**Local-First Offloading and Scheduling (LFOS)** Algorithm




**Edge-First Offloading and Scheduling (EFOS)** Algorithm



# A Motivating Example (Cont’)



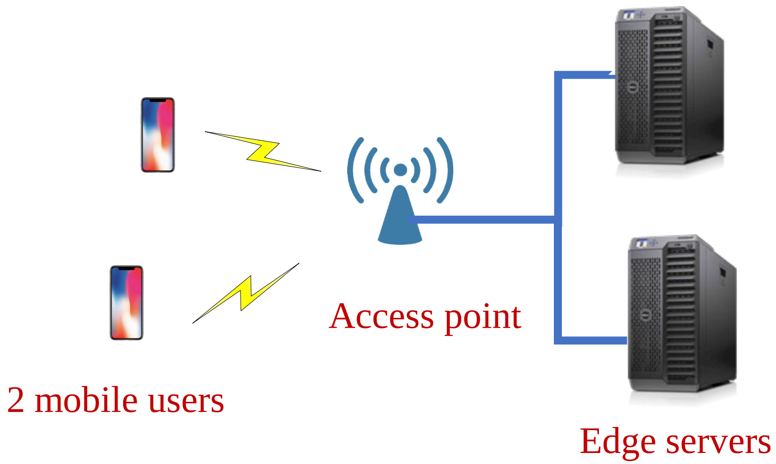

**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

- ❖ In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- ❖ In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- ❖ Only one user can transmit packet within one slot.

6 packets

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| LFOS | 1 | | | | | | |
| | 2 | | | | | | |
| EFOS | 1 | | | | | | |
| | 2 | | | | | | |
| Better Choice | 1 | | | | | | |
| | 2 | | | | | | |

# A Motivating Example (Cont’)




**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

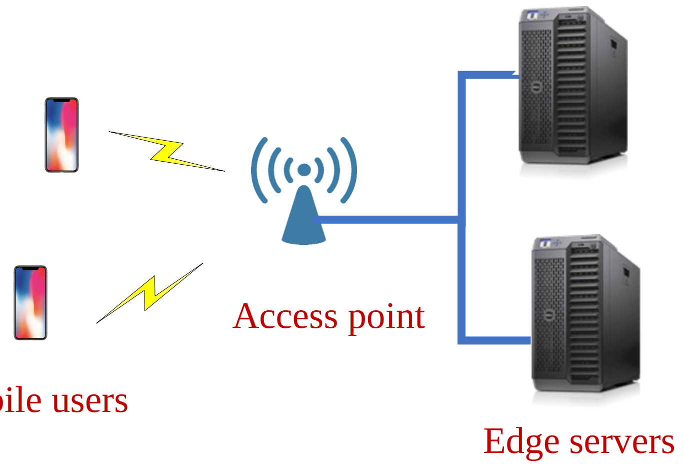
- ❖ In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- ❖ In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- ❖ Only one user can transmit packet within one slot.

Packets remaining

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| **LFOS** | 1 | 3, 11 | | | | | |
| | 2 | 5, 7 | | | | | |
| **EFOS** | 1 | 3, 11 | | | | | |
| | 2 | 5, 7 | | | | | |
| **Better Choice** | 1 | 4, 4 | | | | | |
| | 2 | | | | | | |

Energy consumption

# A Motivating Example (Cont')




**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

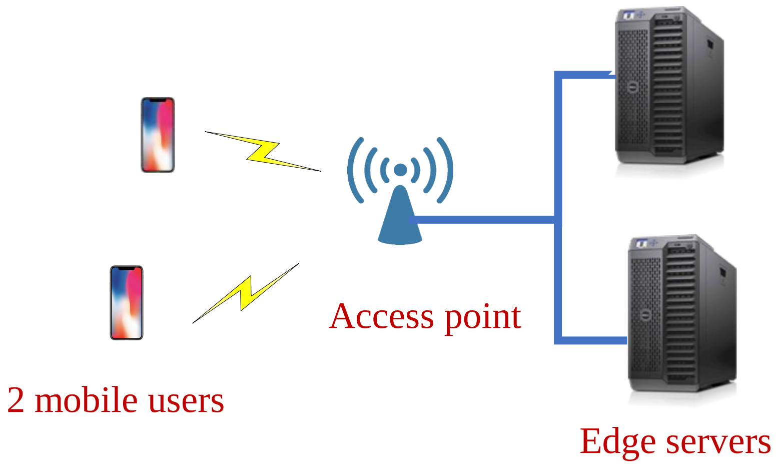
- In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- Only one user can transmit packets within one slot.

Packets remaining

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| **LFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | | | | |
| **EFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | | | | |
| **Better Choice** | 1 | 4, 4 | 2, 4 | | | | |
| | 2 | | | | | | |

Energy consumption

# A Motivating Example (Cont’)




**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

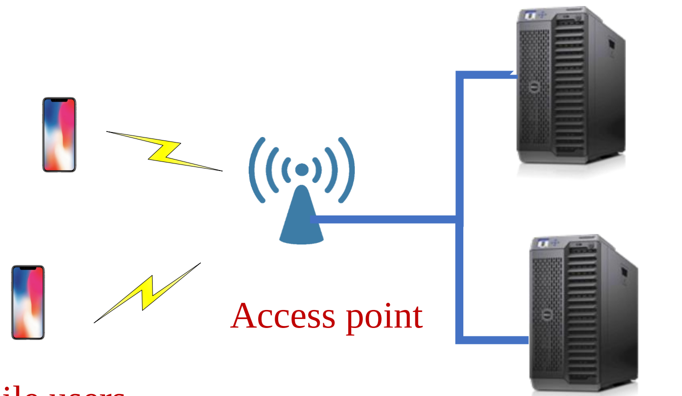
- In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- Only one user can transmit packet within one slot.

Packets remaining

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| **LFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | | | |
| **EFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | | | |
| **Better Choice** | 1 | 4, 4 | 2, 4 | 0, 4 | | | |
| | 2 | | | | | | |

Energy consumption

# A Motivating Example (Cont')




**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

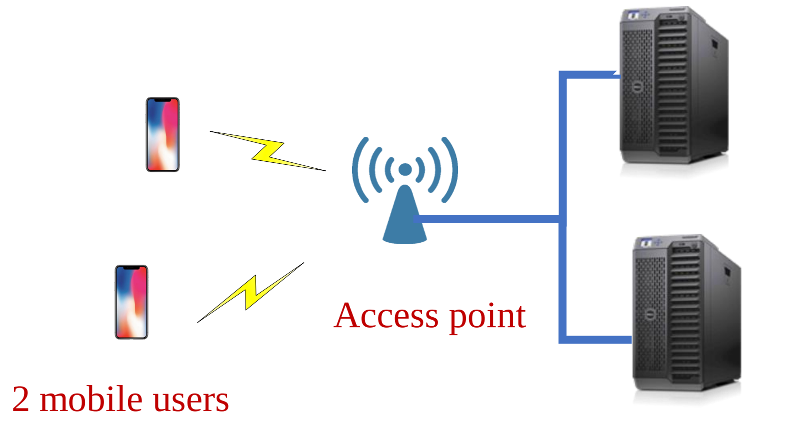
- In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- Only one user can transmit packet within one slot.

Packets remaining

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| **LFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 7 | | |
| **EFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 4 | | |
| **Better Choice** | 1 | 4, 4 | 2, 4 | 0, 4 | | | |
| | 2 | | | | 4, 4 | | |

Energy consumption

# A Motivating Example (Cont')




**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

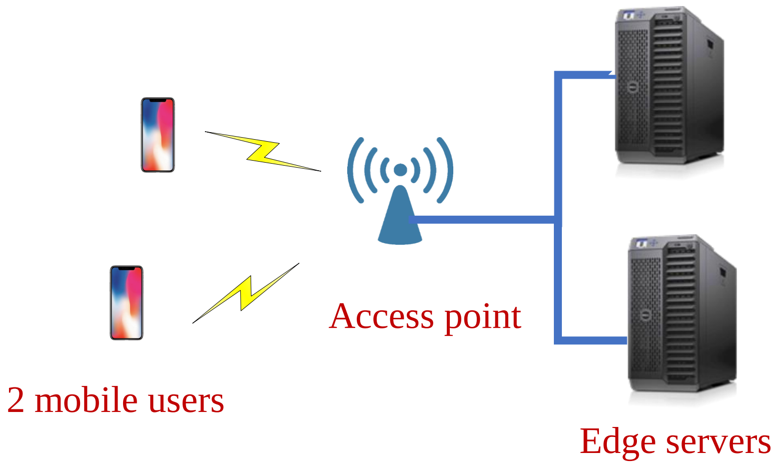
- In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- Only one user can transmit packet within one slot.

Packets remaining

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| LFOS | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 7 | | |
| EFOS | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 4 | | |
| Better Choice | 1 | 4, 4 | 2, 4 | 0, 4 | | | |
| | 2 | | | | 4, 4 | 2, 4 | |

Energy consumption

# A Motivating Example (Cont’)




**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

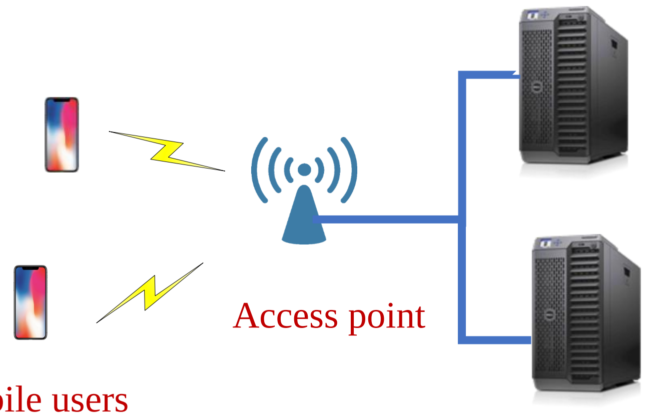
- In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- Only one user can transmit packet within one slot.

Packets remaining

| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| LFOS | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 7 | | |
| EFOS | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 4 | | |
| Better Choice | 1 | 4, 4 | 2, 4 | 0, 4 | | | |
| | 2 | | | | 4, 4 | 2, 4 | 0, 4 |

Energy consumption

# A Motivating Example (Cont’)




| Policy | User | t=0 | t=1 | t=2 | t=3 | t=4 | t=5 |
|---|---|---|---|---|---|---|---|
| **LFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 7 | | |
| **EFOS** | 1 | 3, 11 | 0, 11 | | | | |
| | 2 | 5, 7 | 4, 7 | 1, 11 | 0, 4 | | |
| **Better Choice** | 1 | 4, 4 | 2, 4 | 0, 4 | | | |
| | 2 | | | | 4, 4 | 2, 4 | 0, 4 |

**System setup**: $T = 6$ slots, $N = 2$ users, at time $t = 0$, each user has 6 packets waiting to be processed.

- In each slot, a mobile device can process 1 packet with 7 watt energy consumption;
- In each slot, a mobile device can transmit 2 packets with 4 watt energy consumption;
- Only one user can transmit packet within one slot.

| Policy | LFOS | EFOS | Better choice |
|---|---|---|---|
| Average Energy consumption for each user(watt) | 4.5 | 4.25 | 2 |

A better choice can save energy consumption up to 55.6% compared to LFOS.

# Algorithm Design

We introduce a virtual queue $X_n[kT]$ to keep track of the amount of packets that are dropped.

Virtual queue $X_n[kT]$

Virtual queue arrival:
$D_n[kT]$ packets drop

Virtual queue service:
$B_n[kT]$ service is generated,
where $\mathbb{E}[B_n[kT]] = \rho_n \lambda_n$

Virtual queue dynamics:

$$X_n[(K+1)T] = (X_n[kT] + D_n[kT] - B_n[kT])^+$$

where $(x)^+ = \max\{x, 0\}$ for any real number $x$.

Then the average drop rate of user $n$ meets the requirement if its virtual queue is stable (c.f. [1, Definition 2.2] ).

[1] M. Neely, *Stochastic network optimization with application to communication and queueing systems.* Morgan & Claypool, 2010

# Joint Offloading and Scheduling Algorithm

| Joint Offloading and Scheduling (**JOS**) algorithm |
| --- |

max $\quad \sum_{n=1}^{N} F_n^{(L)}[kT] + \sum_{n=1}^{N} F_n^{(E)}[kT],$

Strongly coupled

where

Nonlinear
$$\begin{cases} F_n^{(L)}[kT] \triangleq X_n[kT] \min\left\{A_n^{(L)}[kT], T\mu_n\right\} - Me_n^{(L)} \min\left\{\left\lceil \frac{A_n^{(L)}[kT]}{\mu_n} \right\rceil, T\right\}, \\ F_n^{(E)}[kT] \triangleq X_n[kT] \min\{A_n^{(E)}[kT], C_n[kT] \sum_{t=kT}^{(k+1)T-1} S_n[t]\} - Me_n^{(E)} \sum_{t=kT}^{(k+1)T-1} S_n[t], \end{cases}$$

$M > 0$ is some parameter, $A_n^{(E)}[kT] + A_n^{(L)}[kT] = A_n[kT].$

***Proposition 1***: The JOS algorithm with any $M > 0$ achieves $O(^1/_M)$ close to the optimal energy consumption at the expense of the mean virtual queue-length growing with $O(M)$.

# Algorithm Implement Roadmap

In **JOS** algorithm, the offloading decisions and wireless scheduling decisions are strongly coupled, which make it hard to implement

↓

Consider one time slot deadline setup, we build decoupled joint offloading and scheduling (**DJOS**) algorithm for the case with one time slot deadline.

↓

Based on the insight of one time slot DJOS, we developed **DJOS** for the general case.

Decoupled Joint Offloading and Scheduling (**DJOS**) algorithm

Wireless scheduling decisions

Offloading decisions

# Wireless Scheduling Decisions (1 Time Slot)

**Step 1:**

Compute weight $W_n^{(L)}[t]$

Only local computation

Compute weight $W_n^{(L,E)}[t]$

Both local computation and wireless transmission

**Step 2:**

**User 1**  Calculate $\text{WeightGap}_1 = W_1^{(L,E)}[t] - W_1^{(L)}[t]$

**User 2**  Calculate $\text{WeightGap}_2 = W_2^{(L,E)}[t] - W_2^{(L)}[t]$

…… ……

**User *N***  Calculate $\text{WeightGap}_\text{N} = W_N^{(L,E)}[t] - W_N^{(L)}[t]$

**Step 3:**

Choose user $n^*[t] \in argmax_n \text{ WeightGap}_\text{n}$ for both wireless transmission and local computation, the rest users($n \neq n^*[t]$) only perform local computations

$W_n^{(L,E)}[t]$ and $W_n^{(L)}[t]$ are defined as follows:

$$W_n^{(L,E)} \triangleq \max_{A_n^{(L)}, A_n^{(E)}} \left( (X_n[t] \min\left\{A_n^{(L)}, \mu_n\right\} - Me_n^{(L)} \mathbb{1}_{\left\{A_n^{(L)} > 0\right\}}) + (X_n[t] \min\left\{A_n^{(E)}, C_n[t]\right\} - Me_n^{(E)} \right)^+ )$$

$$W_n^{(L)} \triangleq \max_{A_n^{(L)}} ((X_n[t] \min\left\{A_n^{(L)}, \mu_n\right\} - Me_n^{(L)} \mathbb{1}_{\left\{A_n^{(L)} > 0\right\}}))$$

# Offloading Decisions(1 Time Slot)

For user $n^*$ that is allowed for both wireless transmission and local computations

| | $W_{n^*}^{(E)}[t] \geq 0$ | $W_{n^*[t]}^{(E)}[t] < 0$ |
|---|---|---|
| $W_{n^*}^{(L)}[t] \geq 0$ | $A_{n^*}^{(E)}[t] = \min\{A_{n^*}[t], C_{n^*}[t]\}$<br>$A_{n^*}^{(L)}[t] = A_{n^*}[t] - A_{n^*}^{(E)}[t]$ | $A_{n^*}^{(E)}[t] = 0$<br>$A_{n^*}^{(L)}[t] = A_{n^*}[t]$ |
| $W_{n^*}^{(L)}[t] < 0$ | $A_{n^*}^{(E)}[t] = A_{n^*}[t]$<br>$A_{n^*}^{(L)}[t] = 0$ | $A_{n^*}^{(E)}[t] = A_{n^*}[t]$<br>$A_{n^*}^{(L)}[t] = 0$ |

For users $n \neq n^*$ that are allowed for local computations only

| | |
|---|---|
| $W_n^{(L)}[t] \geq 0$ | $A_n^{(L)}[t] = \mathrm{A_n}[t]$ |
| $W_n^{(L)}[t] < 0$ | $A_n^{(L)}[t] = 0$ |

$W_{n^*}^{(E)}[t], W_{n^*}^{(L)}[t]$ and $W_n^{(L)}[t]$ are defined as follows:

$$W_{n^*}^{(E)}[t] \triangleq X_{n^*}[t] \min\{A_{n^*}[t], C_{n^*}[t]\} - Me_{n^*}^{(E)}$$

$$W_{n^*}^{(L)}[t] \triangleq X_{n^*}[t] \min\left\{\left(A_{n^*}^{(L)}[t] - C_{n^*}[t]\right)^+, \mu_{n^*}\right\} - Me_{n^*}^{(L)}$$

$$W_n^{(L)}[t] \triangleq X_n[t] \min\left\{A_n^{(L)}[t], \mu_n\right\} - Me_n^{(L)}$$

# General Case (T Time Slots)

**Wireless scheduling decisions :**

Wireless scheduling decisions are obtained by solving the following optimization problem:

$$\boldsymbol{y}^* \in \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{n=1}^{N} \left(W_n^{(y_n)(L,E)}[kT] - W_n^{(L)}[kT]\right)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} y_n = T$$

$O(N^T)$ complexity

where $\boldsymbol{y}^* = (y_n^*)_{n=1}^N$ are the wireless scheduling decisions, $y_n$ denote the maximum number of wireless transmissions that are allowed for user $n$ in frame $k$.

$T = 5$ slots

User 1: schedule 3 time slots

User 2: schedule 2 time slots

$y_1 = 3$ $\quad$ $y_2 = 2$

$y_1 + y_2 = T$

**Offloading Decision:**

The offloading decisions for each user are similar with the case with one time slot

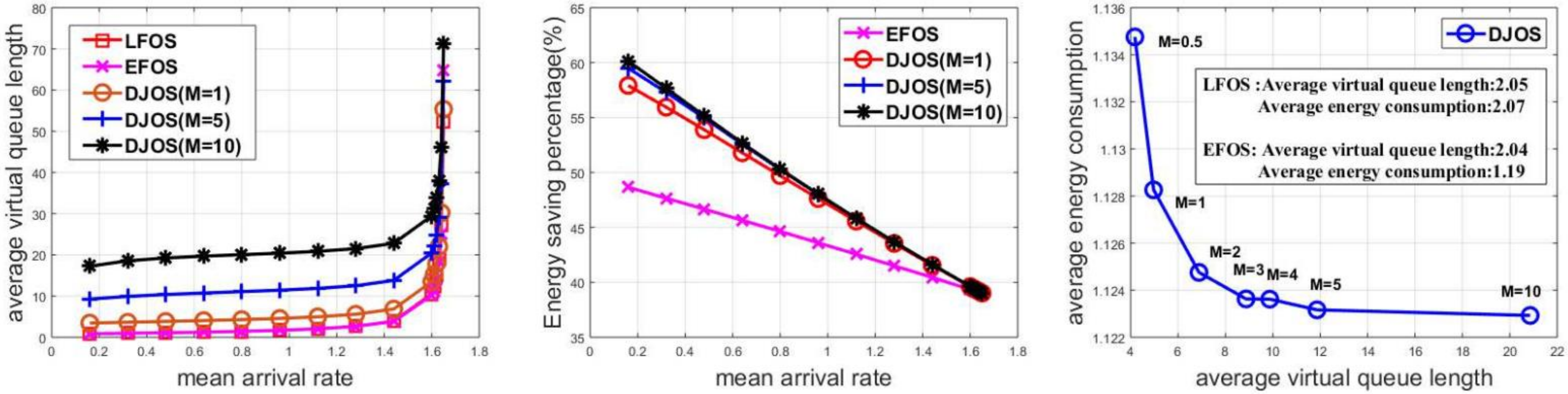
# Simulation Setup

- $N = 5$ users;
- Maximal allowable drop rate $\rho = 0.1$;
- All users suffer from i.i.d ON-OFF channel fading;
- Local computation energy consumption $e^{(L)} = 7$ watt;
- Wireless transmission energy consumption $e^{(E)} = 4$ watt;
- Local processing rate $\mu = 1$

| The case with one time slot deadline | The case with three time slot deadline |
| --- | --- |
| Transmission rate 5 when the channel is ON | Transmission rate 4 when the channel is ON |
| $X = \begin{cases} 5, & p = \lambda/5 \\ 0, & \text{otherwise} \end{cases}$ | $X = \begin{cases} 7, & p = \lambda/7 \\ 0, & \text{otherwise} \end{cases}$ |

Where $X$ denotes number of arriving packets and $p$ denotes the probability that have arrivals.

# Simulation Results (1 Time Slot Deadline)

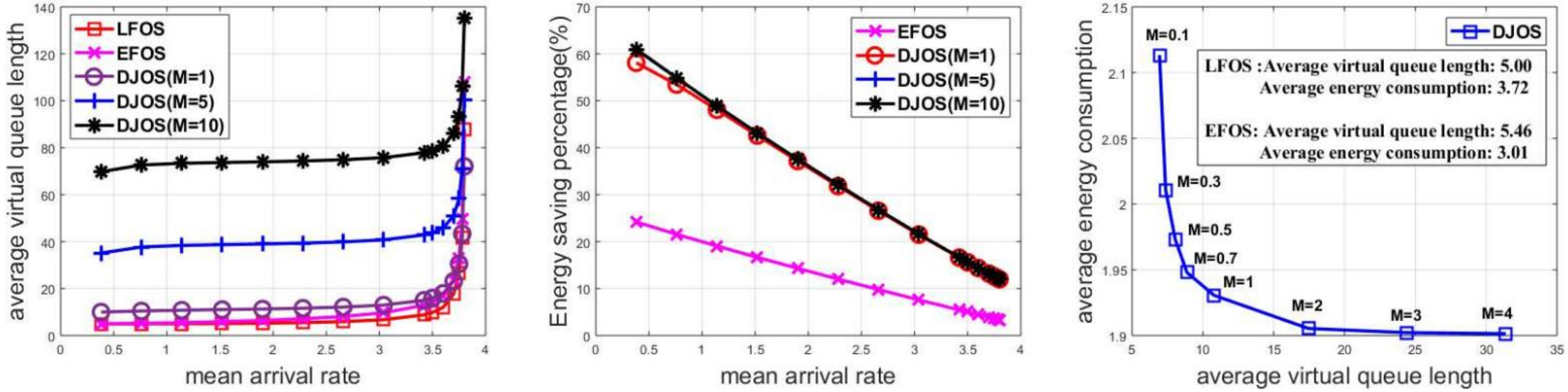



(a) Average virtual queue length　(b) Energy saving compared to LFOS　(c) Impact of $M$ in the DJOS

**The case with one time slot deadline**

- Figure (a) implies that all users satisfy the maximum allowable drop rate
- Figure (b) shows that our proposed DJOS algorithm significantly saves the energy compared to LFOS
- Figure (c) studies the impact of parameter M

# Simulation Results (3 Time Slots Deadline)




(a) Average virtual queue length (b) Energy saving compared to LFOS (c) Impact of $M$ in the DJOS

**The case with three time slots deadline**

- Figure (a) implies that all users satisfy the maximum allowable drop rate
- Figure (b) shows that our proposed DJOS algorithm significantly saves the energy compared to LFOS
- Figure (c) studies the impact of parameter M

# Conclusions & Future work

- We developed the joint offloading and scheduling (JOS) algorithm;
- We developed the decoupled joint offloading and scheduling (DJOS) algorithm for the case with one time slot deadline;
- We further developed the decoupled joint offloading and scheduling (DJOS) algorithm for general case;
- Low-complexity implementation for the decoupled joint offloading and scheduling (DJOS) algorithm.

Thank you!

Q&A