Delay constrained Energy Optimization for Edge Cloud Offloading

Shreya Tayade*, Peter Rost†, Andreas Maeder† and Hans D. Schotten*
*University of Kaiserslautern, Institute for Wireless Communications and Navigation, Kaiserslautern, Germany
Email: {tayade, schotten}@eit.uni-kl.de
†Nokia Bell Labs, Munich, Germany
Email: {peter.m.rost, andreas.maeder}@nokia-bell-labs.com

***Abstract*—Resource limited user-devices may offload computation to a cloud server, in order to reduce power consumption and lower the execution time. However, to communicate to the cloud server over a wireless channel, additional energy is consumed for transmitting the data. Also a delay is introduced for offloading the data and receiving the response. Therefore, an optimal decision needs to be made that would reduce the energy consumption, while simultaneously satisfying the delay constraint. In this paper, we obtain an optimal closed form solution for these decision variables in a multi-user scenario. Furthermore, we optimally allocate the cloud server resources to the user devices, and evaluate the minimum delay that the system can provide, for a given bandwidth and number of user devices.**

I. Introduction

Edge cloud offloading is a promising technique that enables resource-limited user devices to execute computationally extensive tasks. Cloud offloading has been broadly studied recently [1]–[4]. From the user device perspective, to optimally offload the computation, two necessary conditions have to be satisfied: a) Energy consumption of device should be minimum, and b) the offloaded task should be processed within a given latency constraints. However, although energy can be saved by offloading the computation, an additional energy is consumed for transmitting the data to the cloud. Furthermore, an additional transmitting and receiving delay is introduced for offloading the computation to the cloud. As their exist a trade-off between processing locally and offloading, we evaluate the optimal offloading decision.

Many optimal offloading strategies have been proposed to reduce the energy consumption of user devices [5]–[8]. In [5], [6], the energy efficiency of a user device is increased by dynamically scheduling data transmission and link selection, as per the channel condition. Also, a delay constrained, energy minimizing offloading techniques have been proposed in [9]–[11]. In [9], [10], the authors partition a single task, and offload the individual partitions to the distributed cloud servers, ensuring that the execution is completed within a given deadline. However, the work in [5], [6], [9], [10] does not consider the multi-user effects on the cloud server, while taking an offloading decision.

Furthermore, offloading computation also implies an additional cost of communication and cloud resources. The offloading decision is highly influenced by the availability of these resources [12]. Resources like bandwidth, cloud server capacity should be sufficient to satisfy the system requirements of ultra-low latency, and serve computational needs of all the user devices. At the same time, these resources must be used efficiently, which motivates the tradeoff analysis between these resources and the imposed delay requirements. [13], [14] deal with joint optimization of the communication and computational resources for cloud offloading. However, no analysis on the trade-off between these resources and the achieved delay performance was presented.

In this paper, a delay constrained energy optimization algorithm to optimally offload the computation to a cloud-server is designed. A closed form solution is provided for an optimal offloading decision. Also, the cloud resources are optimally allocated among multiple users. Furthermore, the delay performance of system is analyzed for the given bandwidth. In Section II, we describe the system model. The energy optimization problem and the closed form solution is presented in Section III. Finally, the results and conclusion are discussed in Section IV and V respectively.

II. System Model

Consider N uniformly distributed user devices in a circular area of radius R. An edge-cloud server is located at the base station in the center of the cell. The processor of an edge cloud server has a maximum computational capacity of C_{s}. The processors deployed in the user device have a maximum computational capacity of C_{u}, where, $C_{\mathrm{s}} \gg C_{\mathrm{u}}$. The user devices can successfully process all the data within the given time constraints. However, to minimize the energy consumption and reduce latency, user devices offload a share of data processing to the edge cloud server. The edge cloud server processes the data and sends the outcome of the computation to the user-device via the downlink channel as shown in Fig. 1. The uplink and downlink channel are known to the base-station for each user $i \in [1; N]$.

A. Data model

Uplink data model: Every user device processes data from L sensors. The data consists of M data elements that are represented by S bits each as shown in Fig. 1, e. g., surveillance by drones, where, L cameras send images to the user device for processing. Each image is of pixel size M, where each pixel is represented by S bits. Therefore, the total

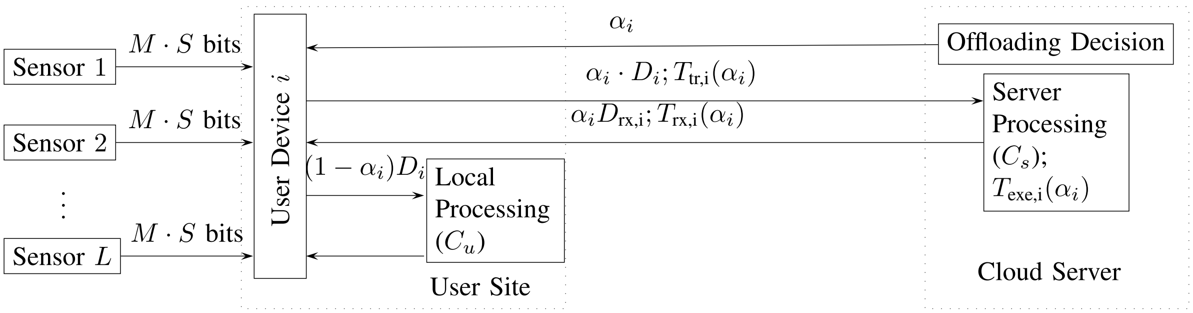

Figure 1. System model

data bits that the user device needs to process is $D_i = L \cdot M \cdot S$. An algorithm to process this data has a complexity class given by the function $f_i(M)$. $f_i(M)$ represents the amount of computational cycles required, with respect to the number of data elements. The decision variable α_i denotes the share of data that should be offloaded, where $0 \leq \alpha_i \leq 1$. Hence, $\alpha_i \cdot D_i$ data bits are transmitted to an edge cloud server for processing. The processing algorithm is distributed to user device and edge cloud if $0 < \alpha_i < 1$.

Downlink data model: Once the edge cloud server has completed all the data processing for the i^{th} user device, it sends back the result to user device via the downlink channel. If S_{rx} represents the number of bits, required to convey the result of a single sensor to the user device, the total data from cloud server to i^{th} user device is $\alpha_i D_{\text{rx,i}}$, where $D_{\text{rx,i}} = L \cdot S_{\text{rx}}$.

B. Delay model

Let $T_{\text{tr,i}}$ be the total time for transmitting the data bits to the edge cloud server from the i^{th} user. The available bandwidth B is distributed equally among all the user devices, i. e., $B_i = B/N$ per user device. If $\alpha_i D_i$ are the total data bits transmitted over the channel with maximum spectral efficiency R_i, the total transmission time is given as

$$T_{\text{tr,i}}(\alpha_i) = \frac{\alpha_i D_i}{B_i R_i}. \tag{1}$$

After transmission, the sensor data is processed in the edge cloud server.

The execution time $T_{\text{exe,i}}$ to process the data, depends upon the computational load $C_{\text{serv,i}}$ introduced on the cloud, and the available cloud server capacity C_s. The computation load at the cloud server from i^{th} user device is given as $C_{\text{serv,i}} = L \cdot \eta_s \cdot f_i(M)$, where η_s is the number of CPU cycles required to process a single data element, for an algorithm of complexity $f_i(\cdot)$. Therefore, the execution time is given as:

$$T_{\text{exe,i}}(\alpha_i) = \frac{\alpha_i C_{\text{serv,i}}}{\rho_i C_s}, \tag{2}$$

where ρ_i represents the percentage of cloud resource allocated to the i^{th} user device, and it holds $\sum_i^N \rho_i \leq 1$; $\forall i = 1 \dots N; \rho_i \geq 0$.

We further define $T_{\text{rx,i}}$ to be the time required to receive the processed result from the cloud server to the i-th user device, i. e.,

$$T_{\text{rx,i}}(\alpha_i) = \frac{\alpha_i D_{\text{rx,i}}}{B_{\text{rx,i}} \cdot R_{\text{rx,i}}}, \tag{3}$$

where, $B_{\text{rx,i}}$ and $R_{\text{rx,i}}$ are the allocated bandwidth and maximum spectral efficiency respectively, for the i^{th} user device in the downlink.

In order to fulfill the latency requirements, the total delay experienced by the user for transmitting, processing and receiving should be less than the maximum delay T_{max}:

$$\forall i = 1 \dots N : T_{\text{tr,i}}(\alpha_i) + T_{\text{exe,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i) \leq T_{\text{max}}. \tag{4}$$

For the sake of simplicity and due to the limited space of this paper, queuing delay is not considered in the model but a pre-reservation of computation resources at the cloud-server is assumed (as it would apply in hard real-time operating systems).

C. Device-centric energy consumption model

The energy consumption model for the user devices is based on the model presented in [12].

Energy consumption for local processing: The total energy consumed by the i^{th} user device to locally process $(1 - \alpha_i)D_i$ bits, is given as

$$E_{\text{u,i}}(\alpha_i) = (1 - \alpha_i) \cdot \epsilon_i \cdot C_{\text{u,i}} \tag{5}$$

where ϵ_i is the average amount of energy consumed by the user device for a single computation cycle, and $C_{\text{u,i}}$ is the computation load generated in terms of computation cycles on the user device [12]. The computational load is given as $C_{\text{u,i}} = L \cdot \eta_i f_i(M)$ where L is the number of sensors, and η_i is a processor specific proportionality constant. η_i represents the number of computation cycles required to process a single data element ($M = 1$) for an algorithm of complexity f_i.

Energy consumption for offloading: The energy consumed to transmit $\alpha_i D_i$ data bits to the cloud with spectral efficiency R_i is given as

$$E_{\text{tr,i}}(\alpha_i) = \frac{(2^{R_i} - 1)}{G} \cdot \left[\frac{d_i}{d_o}\right]^{\beta} \cdot N_0 B_i \cdot T_{\text{tr,i}}(\alpha_i) \tag{6}$$

where β is the path-loss exponent, d_i is the distance between user device and base station, d_o is the reference distance, N_0 is the noise power spectral density, and G is attenuation constant for free-space path-loss. Using $T_{\text{tr,i}}$ in (1), we get

$$E_{\text{tr,i}}(\alpha_i) = \frac{\left(2^{R_i} - 1\right)}{G} \cdot \left[\frac{d_i}{d_o}\right]^{\beta} \cdot N_0 B_i \cdot \frac{\alpha_i D_i}{B_i R_i}. \tag{7}$$

The total energy consumption at the i^{th} user device is

$$E_{\text{sum,i}}(\alpha_i) = E_{\text{u,i}}(\alpha_i) + E_{\text{tr,i}}(\alpha_i). \tag{8}$$

III. Offloading Optimization

A. Problem formulation

Our objective is the derivation of an optimal offloading strategy that minimizes the energy consumption of the user device, while simultaneously ensuring that the total delay is below the threshold T_{max}, i. e.,

$$\mathcal{A}' = \text{arg} \min_{\mathcal{A} \in \mathbb{R}^N} \sum_{i=1}^{N} E_{\text{sum,i}}(\alpha_i) \tag{9}$$

$$\text{s.t} \quad \sum_{i}^{N} \frac{\alpha_i C_{\text{serv,i}}}{T_{\text{exe,i}}(\alpha_i)} \leq C_s \tag{10}$$

$$\mathcal{A} = \{\alpha_1, \alpha_2, \dots \alpha_N\} \tag{11}$$

$$0 \leq \alpha_i \leq 1 \quad \forall i = 1 \dots N \tag{12}$$

In the given optimization problem, the decision vector $\mathcal{A}'$ is evaluated such that the total energy consumption of the user device is minimized. As the objective is to reduce the energy consumption of user devices, the energy consumed at the cloud server is not considered in the optimization problem. The second constraint for optimization ensures that the total processing rate required by the user devices should be less than the total server capacity. It reflects that the sum of the allocated shares of cloud resources ρ_i must not exceed 1, i. e. $\sum_i^N \rho_i \leq 1$. α_i is the offloading decision parameter for the i^{th} user. If α_i = 0, the user device do not offload, while if α_i = 1, all data processing is performed by the cloud server.

In order to serve more user devices on the cloud, it is necessary to efficiently allocate the cloud computational resources, ρ_i. The idea is to allocate more cloud server resources to the user devices that experience larger communication ($T_{\text{tr,i}} + T_{\text{rx,i}}$) delay, so as to reduce their execution time. As mentioned in Section II-B, the total delay for offloading should be less than the maximum delay threshold T_{max}. Therefore, the maximum execution time permitted by the user device is

$$T_{\text{exe,i}}(\alpha_i) \leq T_{\text{max}} - \left(T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i)\right). \tag{13}$$

By substituting (13) in (10), the solution to the optimization problem is evaluated. As the optimization problem is convex, the solution is obtained by applying Lagrangian's duality theorem and KKT conditions.

B. Solution

a) Optimal offloading decision:

Theorem 1. *If the rate of increase in energy consumption for local processing is higher than for offloading, i.e,* $\left[-E^{'}_{tr,i} - E^{'}_{u,i}\right] > 0$, *the optimal offloading decision for* i^{th} *user device is given by*

$$\alpha_i = \min \left[1, \frac{1}{\left[\frac{D_i}{B_i R_i} + \frac{D_{rx,i}}{B_{rx,i} R_{rx,i}}\right]} \left(T_{max} - \sqrt{\frac{\nu \gamma_i T_{max}}{(-E^{'}_{tr,i} - E^{'}_{u,i})}}\right)^{+}\right], \tag{14}$$

where, $E^{'}_{tr,i}(\alpha_i) = \frac{\partial E_{tr,i}(\alpha_i)}{\partial \alpha_i}$ *and* $E^{'}_{u,i}(\alpha_i) = \frac{\partial E_{u,i}(\alpha_i)}{\partial \alpha_i}$, ν *is the Lagrange parameter defining the threshold for admitting the user devices, and* γ_i *is the ratio of computational load to the cloud server capacity, given as* $\gamma_i = \frac{C_{serv,i}}{C_s}$.
If the computation load $C_{serv,i} \ll C_s$, $\gamma_i \rightarrow 0$ *and* $\nu = 0$, *the optimal offloading decision becomes*

$$\alpha_i = \min \left[1, \left(\frac{T_{max}}{\left[\frac{D_i}{B_i R_i} + \frac{D_{rx,i}}{B_{rx,i} R_{rx,i}}\right]}\right)^{+}\right] \tag{15}$$

Proof. The proof of the theorem is given in Appendix. □

Theorem 2. *In case of an overloaded system, i. e.,* $\nu \neq 0$, *as* $\alpha_i \geq 0$, *the lower bound on Lagrange parameter* ν *is*

$$\min_{i:\alpha_i > 0} \mathcal{B} \leq \nu, \tag{16}$$

where, $\mathcal{B} = \{\hat{\nu}_1, \hat{\nu}_2, \dots \hat{\nu}_N\}$, *and*

$$\hat{\nu}_i = \left[\left(T_{max} - \left[\frac{D_i}{B_i R_i} + \frac{D_{rx,i}}{B_{rx,i} R_{rx,i}}\right]\right)^2 \frac{(-E^{'}_{tr,i} - E^{'}_{u,i})}{\gamma_i T_{max}}\right]^{+} \tag{17}$$

Proof. The theorem follows from Theorem 1 and applying the condition $\alpha \leq 1$ in (14). □

b) Optimal cloud resource allocation: The relative share of cloud server capacity allocated to user device i is given as

$$\rho_i = \frac{\alpha_i C_{\text{serv,i}}}{C_s \cdot T_{\text{exe,i}}(\alpha_i)}. \tag{18}$$

Hence, ρ_i and $T_{\text{exe,i}}$ are a function of the optimal offloading decision α_i. If the communication delay is higher then the execution time should be lower and the assigned computational share ρ_i should be higher. In addition, the computational share ρ_i scales linearly with the required computational load $\alpha_i C_{\text{serv,i}}$.

C. Performance metrics

a) Offloading percentage: The offloading percentage is the ratio of total offloaded data processing for all user devices to the total data processing of the system, and it is given by

$$\Lambda = \frac{\sum\limits_{i=1}^{N} \alpha_i \cdot D_i}{\sum\limits_{i=1}^{N} D_i}. \tag{19}$$

b) Sum energy: The performance of the offloading strategy is evaluated by comparing the total optimized energy consumption for all N user devices to the total energy consumption for local processing. The total optimized energy consumption, i. e., $E_{\text{sum}}(\mathcal{A}')$, is given as

$$E_{\text{sum}}(\mathcal{A}') = \sum_{i=1}^{N} E_{\text{sum,i}}(\alpha_i) \tag{20}$$

where α_i is evaluated according to Theorem 1 and (14). The total energy consumption in the case that no user device offloads ($\forall i \in [1; \ldots N] : \alpha_i = 0$) is given by

$$E_{\text{sum}}(\underline{0}) = \sum_{i}^{N} E_{\text{u,i}}(0). \tag{21}$$

c) Cut-off delay T_c: The minimum threshold delay within which the system can process all the optimally offloaded data from N user devices, in presence of bandwidth B.

$$T_c(B, N) = \inf \left\{ T_{\max} > 0 \left| \frac{\partial \Lambda(T_{\max}, B, N)}{\partial T_{\max}} = 0 \right. \right\}$$

Algorithm 1 Optimal Cloud Offloading

```
Initialization:
    $\nu = 0$; $\alpha_i$ = 1, $\forall\, i = 1 \ldots N$;
    $\mathcal{B} = \{\hat{\nu}_1, \hat{\nu}_2, \ldots \hat{\nu}_N\}$
Check on energy consumption:
for User device i = 1:N do
    if $\left[-E'_{\text{tr,i}} - E'_{\text{u,i}}\right] > 0$ then
        $\alpha_i$ = 1                                    ▷ Offload
    else
        $\alpha_i$ = 0                                    ▷ Local processing
    end if
end for
Check load on the cloud server:
while $\left(\sum_{i}^{N} \frac{\alpha_i \gamma_i}{[T_{\max} - (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i))]} - 1\right) > 0$ do
    $\nu$ = $\min_{\forall i = 1 \ldots N; \alpha_i > 0} \{\mathcal{B} | \mathcal{B} > 0\}$
    Drop the user device that have highest communication delay and saves least energy, i. e. $\alpha_i = 0$
    Update:    $\mathcal{B}$, s.t $\mathcal{B} = \mathcal{B} - \{\nu\}$
    Assign $\alpha_i$ with new value of $\nu$, $\forall i = 1 \ldots N$
        $\alpha_i = \min \left[ 1, \frac{\left( T_{\max} - \sqrt{\frac{\nu \gamma_i T_{\max}}{(-E'_{\text{tr,i}} - E'_{\text{u,i}})}} \right)^{+}}{\left[ \frac{D_i}{B_i R_i} + \frac{D_{\text{rx,i}}}{B_{\text{rx,i}} R_{\text{rx,i}}} \right]} \right]$
end while
Output:    $\mathcal{A}'$ and $\nu$
```

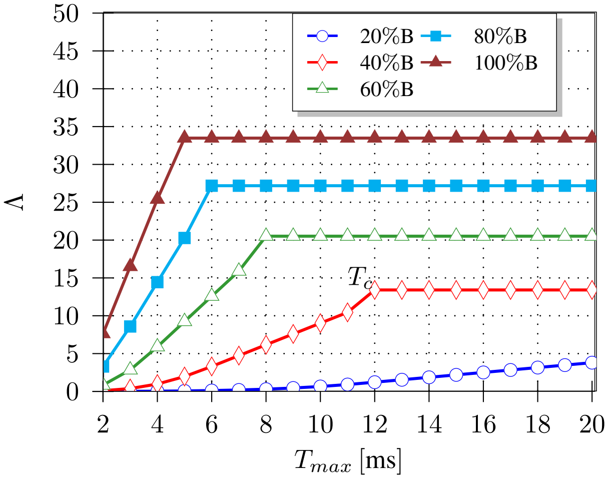

Figure 2. Offloading percentage

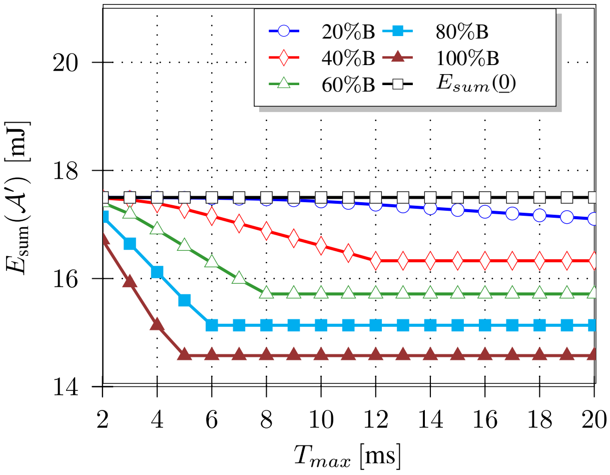

Figure 3. Energy consumption

Table I
SIMULATION PARAMETERS

Variable	Value	Variable	Value
C_s	200 MHz	N	50 Users
L	10	M	70 data elements
B, B_{rx}	20 MHz	S,S_{rx}	8 bits
d_0	200 m	R	800 m
β	2	R_i, R_{rx}	6 bps/Hz
ϵ_i	$5e{-}6$ mJ	η_i	100 cycles
$f_i(M)$	M	η_s	1 cycle

IV. RESULTS AND DISCUSSIONS

a) Optimal offloading strategy and energy consumption: Fig. 2 shows the effect on the average offloading percentage Λ, with an increasing delay $T_{\max}$, and for different bandwidth B available. The simulation parameters used for this evaluation are shown in Table I.

To optimally offload the data from the user devices, two crucial goals need to be achieved: a) save energy of the user

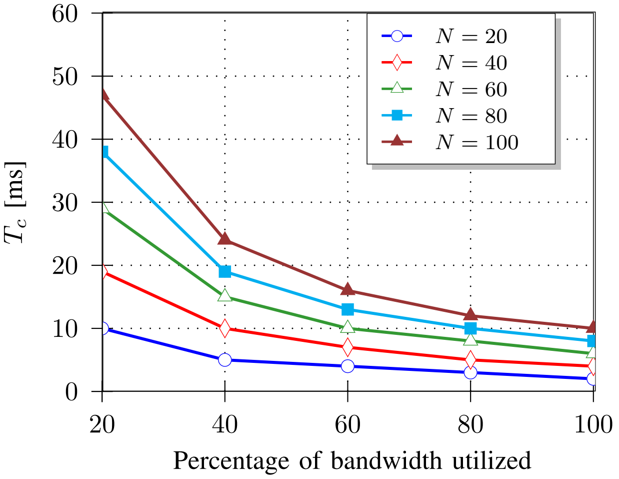

Figure 4. Bandwidth-delay trade-off

device, and b) finish the computation within T_{max}. Consider the case when 40%, of bandwidth B is available. At lower values of threshold delay, i. e. $T_{\text{max}} < 12ms$, the cloud server cannot process all the computational data within the given time constraint. Therefore, user devices only offload the data partially, ensuring that the offloaded data is computed within the specified delay. Therefore, more data is offloaded as the delay threshold is increased until the cut-off delay T_c is reached. A further increase in the delay threshold $T_{\text{max}} > 12ms$ do not impact the offloading percentage Λ, which remains constant and is limited below 15%. This indicates that the cloud server could process more data for $T_{\text{max}} > 12ms$ but due to the limited bandwidth, it is not energy-efficient to offload more data. Therefore, the user devices do not offload, even for the lenient time constraints. The optimal offloading percentage increases, if the percentage of available bandwidth is increased. Higher the bandwidth available, lower will be the data transmission time, and hence, the end-end delay. Therefore, at higher bandwidth ($60\%B$ and $80\%B$), the cut-off delay T_c is reduced.

Similar behavior is seen in Fig. 3. The energy consumption decreases, with an increase in the delay threshold T_{max}. This implies that energy consumption can be further reduced, if the cloud server can process more computation within the threshold delay. However, more computation could not be offloaded due to limited cloud server capacity and bandwidth. Therefore, at lower threshold delay, i. e., $T_{\text{max}} < T_c$, more energy can be saved by increasing the cloud server capacity. An increase in cloud server capacity will reduce the processing time, and hence will allow user devices to offload more computation. Whereas, if the threshold delay is higher, i. .e., $T_{\text{max}} > T_c$, the cloud server can effortlessly process more computation. An increase in cloud server capacity will have no impact on the energy consumption. Therefore, the energy consumption can only be further reduced by increasing the bandwidth.

b) Bandwidth-delay-user devices trade-off: The results discussed previously show that there exists a trade-off between bandwidth, B, number of users N that can offload their computation, and cut-off delay T_c. If the number of user device increases, more bandwidth is required, and processing at the cloud server will take longer time. Whereas, if the bandwidth increases, the cut-off delay decreases, and more user devices can offload. Fig. 4 demonstrates the behavior of this trade-off. The aim is to serve the maximum number of user devices with minimum bandwidth, at very low cut-off delay. Fig. 4 illustrates the rate at which the cut-off delay decreases with respect to bandwidth. The cut-off delay decreases exponentially with an increase in bandwidth. However, when the number of user devices is lower, bandwidth has less impact on the cut-off delay. This is due to the fact that the time to transmit and receive is significantly lower compared to the execution time.

V. Conclusion

In this paper, we presented a delay constrained and energy optimized cloud offloading framework. A closed form solution is obtained to optimally offload the computation from multiple user devices, depending on the availability of bandwidth and latency constraints. The optimal offloading strategy is to offload more computation to the cloud, if sufficient bandwidth and cloud server capacity is available. At higher threshold delay, the offloading decision is independent of the delay constraint. However, it depends upon the difference of the energy for transmitting and local processing. We also analyzed the trade-off between the bandwidth and cut-off delay considering the optimal offloading solution. In future, we plan to extend the framework to study the effects of channel fading and shadowing on the optimal offloading strategy.

Appendix

In the following, we prove Theorem 1 using the method of Lagrange multipliers and Karush-Kuhn-Tucker (KKT) conditions. Substituting the maximum value of $T_{\text{exe}}(\alpha_i)$ in (10), and replacing $\left(\frac{C_{\text{serv,i}}}{C_s}\right) = \gamma_i$, the constraint (10) of optimization problem becomes $\sum_i^N \frac{\alpha_i\gamma_i}{[T_{\text{max}} - T_{\text{rx}}(\alpha_i) - T_{\text{tr}}(\alpha_i)]} \leq 1$. The objective function for the optimization is hence given as

$$\mathcal{L}(\alpha_i, \psi, \nu) = \sum_{i=1}^{N} E_{\text{sum,i}}(\alpha_i) + \nu\left(\sum_i^N \frac{\alpha_i\gamma_i}{[T_{\text{max}} - (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i))]} - 1\right) - \text{tr}\left[\mathbf{\Psi}\text{diag}(\alpha_i)\right] \tag{22}$$

where, ν and ψ are the Lagrange's multiplier. Take partial derivative with respect to α_i and equating $\frac{\partial\mathcal{L}}{\partial\alpha_i}$ to 0, we get,

$$\frac{\partial\mathcal{L}}{\partial\alpha_i} = E'_{\text{tr,i}} + E'_{\text{u,i}} + \nu \cdot \frac{\gamma_i T_{\text{max}}}{[T_{\text{max}} - T_{\text{rx,i}}(\alpha_i) - T_{\text{tr}}(\alpha_i)]^2} - \psi_i = 0 \tag{23}$$

$$\text{Where,}\quad E^{'}_{\text{tr,i}}(\alpha_i) = \frac{\partial E_{\text{tr,i}}(\alpha_i)}{\partial \alpha_i} \text{ and } E^{'}_{\text{u,i}}(\alpha_i) = \frac{\partial E_{\text{u,i}}(\alpha_i)}{\partial \alpha_i}. \tag{24}$$

$E^{'}_{\text{tr,i}}(\alpha_i)$ and $E^{'}_{\text{u,i}}(\alpha_i)$ are obtained by taking derivative of (7) and (5) respectively.

$$E^{'}_{\text{tr,i}} = \frac{2^{R_i}-1}{G} \cdot \left[\frac{d_i}{d_o}\right]^{\beta} \cdot N_0 \cdot \frac{D_i}{R_i} \tag{25}$$

$$E^{'}_{\text{u,i}} = -L \cdot \epsilon_i \eta_i f_i(M). \tag{26}$$

Convexity check: Take second derivative of $\mathcal{L}(\alpha_i, \nu, \psi_i)$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_i^2} = 2\nu\gamma_i T_{\text{max}}(T_{\text{max}} - [T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i)])^{-3} \tag{27}$$

$$\text{where,}\quad T_{\text{max}} > (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i)).$$

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_i \alpha_j} = 0, \forall i \neq j. \tag{28}$$

Substituting the values of (27) and (28) in Hessian matrix, shows that it is positive semi-definite, hence, the objective function is a convex function.

KKT Conditions:

$$\nu \left(\sum_i^N \frac{\alpha_i \gamma_i}{[T_{\text{max}} - (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i))]} - 1 \right) = 0 \tag{29}$$

$$\nu \geq 0; \psi_i \alpha_i = 0; \tag{30}$$

$$\psi_i \geq 0; \alpha_i \geq 0; \tag{31}$$

CASE I: Fully-loaded case ($\nu > 0; \psi_i = 0$)

If ν > 0, i.e $\left(\sum_i^N \frac{\alpha_i \gamma_i}{[T_{\text{max}} - (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i))]} - 1 \right)$ = 0. If ψ_i = 0, i.e α_i > 0. Substitute the values of $\psi_i = 0$ in (23), and rearranging to evaluate α_i, we get

$$\alpha_i = \frac{1}{\frac{D_i}{B_i R_i} + \frac{D_{\text{rx,i}}}{B_{\text{rx,i}} R_{\text{rx,i}}}} \left(T_{\text{max}} - \sqrt{\frac{\nu \gamma_i T_{\text{max}}}{(-E^{'}_{\text{tr,i}} - E^{'}_{\text{u,i}})}} \right) \tag{32}$$

CASE II: Underloaded (ν = 0; ψ_i = 0)

If ν = 0, implies that $\left(\sum_i^N \frac{\alpha_i \gamma_i}{[T_{\text{max}} - (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i))]} - 1 \right)$ < 0. It states that total required processing rate is less than the server capacity. It means cloud server can process all the computation data (i. e. $\alpha_i = 1$) from all user devices within the specified time. Substitute the values ν = 0 in (23) we get,

$$E'_{\text{tr,i}} + E'_{\text{u,i}} = \psi_i \tag{33}$$

Now, put ψ_i = 0 in (33). In this case user devices can offload to the cloud if the transmission energy is less than or equal to local processing energy.

CASE III: Overloaded (ν > 0, ψ_i > 0 i. e. $\alpha_i = 0$)

Substitute $\alpha_i = 0$ in (23)

$$E^{'}_{\text{tr,i}} + E^{'}_{\text{u,i}} + \frac{\nu \gamma_i T_{\text{max}}}{T_{\text{max}}} - \psi_i = 0$$

$$\psi_i = E^{'}_{\text{tr,i}} + E^{'}_{\text{u,i}} + \nu \gamma_i \tag{34}$$

The Lagrange multiplier $\nu > 0$ and $\gamma_i > 0$. Therefore, for $\psi_i > 0$, i. e. $\alpha_i = 0$, the rate at which transmission energy $E^{'}_{\text{tr,i}}$ increases, has to be greater than the rate of increase in energy consumption due to local processing $E^{'}_{\text{u,i}}$. However, in this case as $\nu > 0$, that means the cloud server is already overloaded. Hence, user device do not offload even if the transmission energy is less than the local processing energy consumption, unless the difference between $E^{'}_{\text{u,i}}$ and $E^{'}_{\text{tr,i}}$ exceeds the value $\nu \gamma_i$.

CASE IV: Underloaded (ν = 0, ψ_i > 0; α_i = 0); If ν = 0, then $\left(\sum_i^N \frac{\alpha_i \gamma_i}{[T_{\text{max}} - (T_{\text{tr,i}}(\alpha_i) + T_{\text{rx,i}}(\alpha_i))]} - 1 \right) < 0$; Now, substitute the values ν = 0 and α = 0 in equation (23), we get

$$E^{'}_{\text{tr,i}} + E^{'}_{\text{u,i}} = \psi_i \tag{35}$$

As ψ_i > 0, if α_i = 0, it means that if the transmission energy exceeds the in-device energy consumption, do not offload.

REFERENCES

[1] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy? Techniques to save energy for mobile systems."

[2] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," *Mob. Networks Appl.*, no. 1, pp. 129–140, feb 2013.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint arXiv:1701.01090*, 2017.

[4] Y. Cui, X. Ma, H. Wang, I. Stojmenovic, J. Liu, Y. Cui, X. Ma, ·. H. Wang, I. Stojmenovic, and J. Liu, "A Survey of Energy Efficient Wireless Transmission and Modeling in Mobile Cloud Computing," *Mob. Netw Appl*, vol. 18, pp. 148–155, 2013.

[5] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel," *IEEE Trans. Wirel. Commun.*, vol. 12, no. 9, pp. 4569–4581, sep 2013.

[6] Xudong Xiang, Chuang Lin, and Xin Chen, "Energy-Efficient Link Selection and Transmission Scheduling in Mobile Cloud Computing," *IEEE Wirel. Commun. Lett.*, vol. 3, no. 2, pp. 153–156, apr 2014.

[7] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[8] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 3, pp. 1397–1411, mar 2017.

[9] Y.-H. Kao and B. Krishnamachari, "Optimizing Mobile Computational Offloading with Delay Constraints."

[10] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, June 2012.

[11] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE Int. Conf. Commun.* IEEE, may 2017, pp. 1–6.

[12] S. Tayade, P. Rost, A. Maeder, and H. Schotten, "Device-centric Energy Optimization for Edge Cloud Offloading," in *Globecom 2017*. IEEE, 2017.

[13] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, June 2015.

[14] ——, "Distributed joint optimization of radio and computational resources for mobile cloud computing," in *2014 IEEE 3rd Int. Conf. Cloud Netw.* IEEE, oct 2014, pp. 211–216.