KICS The Korean Institute of Communications and Information Sciences

CrossMark

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 3 (2017) 33–37

www.elsevier.com/locate/icte

Multi-flow management for mobile data offloading☆,☆☆

Yeongjin Kim[a], Joohyun Lee[b,*], Jaeseong Jeong[c], Song Chong[a]

[a] *School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea*
[b] *Department of Electrical and Computer Engineering, Ohio State University, USA*
[c] *Ericsson Research, Stockholm, Sweden*

Received 25 July 2016; received in revised form 27 August 2016; accepted 29 August 2016
Available online 9 September 2016

Abstract

Cellular networks are facing explosive growth of mobile data traffic due to the proliferation of smart devices and traffic-intensive applications. As a cost-effective solution, delayed Wi-Fi offloading was introduced to shift the delay-tolerant traffic from cellular to Wi-Fi networks by trading additional delays. Our paper studies a multi-flow rate control problem where each flow has different traffic load and a deadline. To maximize user satisfaction defined as offloading efficiency minus disutility caused by deadline violation, we propose a dynamic programming-based rate control algorithm. Moreover, to reduce the computation and memory, we propose a simple threshold-based rate control algorithm.

Keywords: Delayed Wi-Fi offloading; Multi-flow management; Mobile data offloading

1. Introduction

As smart devices equipped with high processing capability and diverse applications become popular, mobile data traffic is growing exponentially in cellular networks. Cisco reported the exponential growth of mobile data traffic for past few years and forecasted that the total global mobile data traffic will increase 7-fold between 2013 and 2017. Traffic is video contents with high-definition (HD) resolution. This trend causes degradation in user experience, such as the huge amount of delay and energy consumption of cellular data transfer. Moreover, the burden of payment for cellular data usage also increases for mobile users saturation in cellular networks.

Wi-Fi offloading is an efficient solution to drastically decrease the cellular data traffic and it has several advantages for cellular network provider and mobile users: (i) Wi-Fi can be deployed by lower cost than a cellular base station and Wi-Fi APs are already spread in most of the casual spaces such as workplace and home, e.g., 270 million APs are deployed globally. (ii) Wi-Fi interface requires low transfer energy per bit that is only 5% of that of 3G interface [2] because Wi-Fi has a high data rate and low communication power within a short range. (iii) Mobile users are able to enjoy Internet access with low (or almost zero) monetary cost through Wi-Fi networks.

Even though Wi-Fi has many advantages, it has one serious drawback; because of short communication distance and unplanned deployments, Wi-Fi connectivity is intermittent which depends on the mobility pattern of mobile users. To make up for the limit, *Delayed Wi-Fi offloading* [3] was introduced which

* Corresponding author.
E-mail address: lee.7119@osu.edu (J. Lee).

Peer review under responsibility of The Korean Institute of Communications Information Sciences.

☆ The conference version of this paper was presented in ICOIN2016, Kota Kinabalu, Malaysia, Jan. 15, 2016 (Kim et al. 2016) [1].
This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0190-16-2017, Resilient/Fault-Tolerant Autonomic Networking Based on Physicality, Relationship and Service Semantic of IoT Devices).

☆☆ This paper has been handled by Prof. Seong-Lyun Kim.

http://dx.doi.org/10.1016/j.icte.2016.08.011

hold off the data transfer until Wi-Fi is connected for the delay-tolerant[1] applications. It enables mobile users to grab more Wi-Fi opportunities compared to on-the-spot offloading [3]. According to a survey dealt with delay-tolerance of mobile users under several type of applications [4], users can tolerate long delay (several minutes or hours) for software update, cloud backup and content download applications once proper incentives are supported. These applications incur larger data traffic than real-time applications such as web browsing, message and e-mail. In practice, there are several mobile applications utilizing delayed Wi-Fi offloading, such as *Dropbox* for cloud storage and *hoppin* for VOD (Video On Demand) services.

However, a mobile user is not able to wait for Wi-Fi opportunity all day long without any guarantee of future Wi-Fi contacts. Therefore, it is non-trivial, whether to use the cellular link or not, when Wi-Fi is unavailable. That becomes even more challenging if the mobile device serves multiple flows where each flow requires different amounts of traffic and deadline. Because the flows share the cellular and Wi-Fi network resources, some flows may not be finished within their own deadlines. It degrades satisfaction of the mobile user due to additional transmission time after the deadline. Therefore, a multi-flow rate control algorithm should balance between offloading efficiency[2] and disutility caused by deadline violations. Existing studies mainly focused on a single-flow management or multi-flow case without specified deadlines.

In this paper, we formulate the multi-flow rate control problem as a finite-horizon and discrete Markov decision problem. The objective is maximizing user satisfaction which is composed of offloading efficiency minus disutility caused by unfinished traffic after deadline. We focus on a download case because most of mobile data is download traffic,[3] but our results can be easily expended for the upload case. Then, we propose two rate control algorithms and evaluate them based on trace-driven simulations.

2. Proposed algorithm

In this section, we design a delayed Wi-Fi offloading system when multiple flows are coexisting with their own traffic loads and deadlines. We formulate an optimization problem to maximize user satisfaction which is a total volume of data traffic offloaded through Wi-Fi minus disutility caused by deadline violations. The formulation is based on a finite-horizon and discrete Markov decision problem motivated from [6] which solves a on/off problem in a single-flow case. Then, we propose two multi-flow rate control algorithms: (i) dynamic programming-based optimal algorithm and (ii) threshold-based heuristic algorithm.

2.1. System model

Flow and network model. We consider a scenario when a mobile user reserve m-download flows where each flow is indexed by $i \in \mathcal{I} = \{1, 2, \ldots, m\}$. Each flow i has to be completely transferred until its deadline T^i assigned by the user. We denote as $\boldsymbol{T} = (T^1, T^2, \ldots, T^m)$ a vector of deadline for the flow set $\mathcal{I}$ and it is arranged in an ascending order (i.e., $T^1 \leq T^2 \leq \cdots \leq T^m$). We model a time-slotted system, $t \in \mathcal{T} = \{1, 2, \ldots, T^m\}$ and at each time slot t, a remaining file size of flow i is denoted by f_t^i where $\boldsymbol{f}_t = (f_t^1, f_t^2, \ldots, f_t^m)$. $\boldsymbol{f}_1$ is an initial file size vector when $t = 1$ and $\boldsymbol{f}_t \in \boldsymbol{\mathcal{F}} = [\boldsymbol{0}, \boldsymbol{f}_1]$, for all $t \in \mathcal{T}$. We denote a network state at time t by $l_t \in \mathcal{L} = \{c, w\}$, where $l_t = c$ and $l_t = w$ denote cellular and Wi-Fi network, respectively. In this work, the network interface selection in heterogeneous networks is outside of our scope and we assume that the mobile device connects Wi-Fi network whenever it is possible. We model the network state transition over time as 2-state Markov chain[4] where $0 < p(l_{t+1}|l_t) < 1$, for all $l_t, l_{t+1} \in \mathcal{L}$. We denote a data rate for the network state l as r^l. We assume that the data rates of cellular and Wi-Fi networks are time and location independent for simplicity.

State transition model. We define a system state at time t $\boldsymbol{s}_t = (\boldsymbol{f}_t, l_t)$ as a tuple of the remaining file size vector and the network state. $\boldsymbol{s}_{t+1}$ only depends on the previous state $\boldsymbol{s}_t$ and an action $\boldsymbol{a}_t$. $\boldsymbol{a} = (a^1, a^2, \ldots, a^m)$ is defined as a rate control vector (called action vector) where a^i represents allocated data rate of flow i during 1-time slot. We define as $\boldsymbol{\mathcal{A}}_t(\boldsymbol{f}, l)$ a feasible set of action vector for a given state $\boldsymbol{s}$ at time t.

Definition 1 (*Feasible Set of Action Vector*)**.** For all rate control action vector $\boldsymbol{a} \in \boldsymbol{\mathcal{A}}_t(\boldsymbol{f}, l)$, it satisfies:

$$\sum_{i \in \mathcal{I}} a^i \leq r^l, \tag{1}$$

$$\boldsymbol{0} \leq \boldsymbol{a} \leq \boldsymbol{f}, \tag{2}$$

$$a^i = 0, \forall i \text{ s.t. } T^i < t, \tag{3}$$

where (1) represents that a sum rate of all flows is within the current network capacity; (2) represents that rate of flow i is not able to exceed remaining file size f^i for all $i \in \mathcal{I}$; (3) means that a flow whose deadline is already expired cannot be activated.

We denote by $p_t(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ a state transition probability that the system goes from the state $\boldsymbol{s}$ to $\boldsymbol{s}'$ when applying the action $\boldsymbol{a}$ at time t. The probability can be decomposed into two component, because the network state transition is time-independent and action-independent.

$$p_t(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) = p_t((\boldsymbol{f}', l')|(\boldsymbol{f}, l), \boldsymbol{a}) = p(l'|l)\, p_t(\boldsymbol{f}'|\boldsymbol{f}, \boldsymbol{a}),$$

$$\text{where } p_t(\boldsymbol{f}'|\boldsymbol{f}, \boldsymbol{a}) = \begin{cases} 1, & \text{if } \boldsymbol{f}' = \boldsymbol{f} - \boldsymbol{a}, \\ 0, & \text{otherwise.} \end{cases}$$

[1] They do not have instantaneous delay constraints.

[2] The amount of data traffic transferred through Wi-Fi network.

[3] The amount of download traffic is about 6 times more than that of upload traffic in cellular network [5].

[4] It is a discrete version of Poisson process of Wi-Fi contact and inter-contact.

2.2. Problem definition

We denote as $g(l, \boldsymbol{a}) = \|\boldsymbol{a}\|_1 I_{l=w}$ the data volume transferred through Wi-Fi for a given network state l and an action vector $\boldsymbol{a}$. When the deadlines of all flows are expired (i.e., $t = T^m + 1$) and the system state is $\boldsymbol{s}$, we define a disutility function of user as $c_{T^m+1}(\boldsymbol{s})$.

$$c_{T^m+1}(\boldsymbol{s}) = c_{T^m+1}(\boldsymbol{f}, l) = c(\boldsymbol{f}).$$

The disutility only depends on the unfinished file size vector $\boldsymbol{f}$ which degrades user satisfaction because of incurring additional download delay after deadlines. It is a non-decreasing function on $\boldsymbol{f}$ and user-dependent function correspond to sensitiveness on the additional delay. Also, there is no disutility when all the flows are finished within their deadlines ($c(\mathbf{0}) = 0$).

We denote a policy of user as $\boldsymbol{\pi} = (\boldsymbol{\delta}_t^{\boldsymbol{\pi}}(\boldsymbol{f}, l), \forall \boldsymbol{f} \in \boldsymbol{\mathcal{F}}, l \in \mathcal{L}, t \in \mathcal{T})$. The policy includes actions for all state $\boldsymbol{s}$ and time t, where $\boldsymbol{\delta}_t^{\pi}(\boldsymbol{f}, l) \in \boldsymbol{\mathcal{A}}_t(\boldsymbol{f}, l)$ and $\boldsymbol{\pi} \in \boldsymbol{\Pi}$. Then, we target our objective to maximize the expected satisfaction (i.e., total data volume transferred through Wi-Fi during $[1, T^m]$ minus disutility caused by deadline violations at $t = T^m + 1$) by controlling the rate control policy $\boldsymbol{\pi}$ as follows.

$$\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathbb{E}_{\boldsymbol{s}}^{\boldsymbol{\pi}} \Big[\sum_{t=1}^{T^m} g_t(l_t, \boldsymbol{\delta}_t^{\boldsymbol{\pi}}(\boldsymbol{s}_t^{\boldsymbol{\pi}})) - c(\boldsymbol{s}_{T^m+1}^{\boldsymbol{\pi}}) \Big],$$

where $\boldsymbol{s}_t^{\boldsymbol{\pi}} = (\boldsymbol{f}_t^{\boldsymbol{\pi}}, l_t)$ is the state at time t when the policy $\boldsymbol{\pi}$ is applied in the system.

2.3. Dynamic programming-based algorithm

We derive an optimal multi-flow rate control algorithm by using dynamic programming (DP) framework. We define as $v_t(\boldsymbol{s})$ the maximum expected user satisfaction during the interval $[t, T^m + 1]$ for a given state $\boldsymbol{s}$ immediately before the rate control at time t.

$$v_t(\boldsymbol{s}) = \max_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}_t(\boldsymbol{f}, l)} \{\psi_t(\boldsymbol{f}, l, \boldsymbol{a})\}, \tag{4}$$

where $\psi_t(\boldsymbol{f}, l, \boldsymbol{a})$ is the maximum expected satisfaction during $[t, T^m + 1]$ for a given state $\boldsymbol{s}$ and applying an action $\boldsymbol{a}$ at time t defined as follows.

$$\begin{aligned} &\psi_t(\boldsymbol{f}, l, \boldsymbol{a}) \\ &= g(l, \boldsymbol{a}) + \sum_{l' \in \mathcal{L}} \sum_{\boldsymbol{f}' \in \boldsymbol{\mathcal{F}}} p_t((\boldsymbol{f}', l') | (\boldsymbol{f}, l), \boldsymbol{a}) v_{t+1}(\boldsymbol{f}', l') \\ &= \|\boldsymbol{a}\|_1 I(l = 1) + \sum_{l' \in \mathcal{L}} p(l'|l) v_{t+1}(\boldsymbol{f} - \boldsymbol{a}, l'). \end{aligned} \tag{5}$$

The first term in (5) means the immediate satisfaction at time t for the state $\boldsymbol{s}$ and the action $\boldsymbol{a}$, and the second term means the expected future satisfaction during the interval $[t + 1, T^m + 1]$. For the special case, $v_{T^m+1}(\boldsymbol{s}) = -c(\boldsymbol{f})$ because there is no active flow and only remains disutility for the unfinished file size vector $\boldsymbol{f}$ at time $T^m + 1$.

Let $\boldsymbol{\pi}^*$ be an optimal policy the maximizes expected satisfaction during $[1, T^m + 1]$ that we target on. Then, the optimal action for the system state $(\boldsymbol{f}, l)$ at time t can be written as follows.

$$\boldsymbol{\delta}_t^{\boldsymbol{\pi}^*}(\boldsymbol{f}, l) = \operatorname*{argmax}_{\boldsymbol{a} \in \boldsymbol{\mathcal{A}}} \{\psi_t(\boldsymbol{f}, l, \boldsymbol{a})\}, \forall \boldsymbol{f} \in \boldsymbol{\mathcal{F}}, l \in \mathcal{L}, t \in \mathcal{T}, \tag{6}$$

where (4) and (6) can be computed by backward induction. They are recursively calculated from the boundary time $t = T^m + 1$ to the initial time $t = 1$. The optimality of our DP-based algorithm can be derived by following pp. 83 in [7] which is omitted here for the space limit. Then, our DP-based rate control algorithm can be described as follows.

Dynamic Programming-based rate control algorithm

Pre-calculating,
Input: $\boldsymbol{f}_1, l_1, (p(l'|l), \forall l, l' \in \mathcal{L}), c(\cdot), r^c$ and r^w, **Output:** $\boldsymbol{\pi}^*$.
$v_{T^m+1}(\boldsymbol{f}, l) = -c(\boldsymbol{f}), \forall \boldsymbol{f} \in \boldsymbol{\mathcal{F}}, l \in \mathcal{L}$.
$t = T^m$;
while $(t > 0)$
  for $\boldsymbol{f} \in \boldsymbol{\mathcal{F}}$
    for $l \in \mathcal{L}$
      Calculate $\psi_t(\boldsymbol{f}, l, \boldsymbol{a}), \forall \boldsymbol{a} \in \boldsymbol{A}_t(\boldsymbol{f}, l)$ by (5).
      Find $\boldsymbol{\delta}_t^{\boldsymbol{\pi}^*}(\boldsymbol{f}, l)$ by (6) and $v_t(\boldsymbol{f}, l) = \psi_t(\boldsymbol{f}, l, \boldsymbol{\delta}_t^{\boldsymbol{\pi}^*}(\boldsymbol{f}, l))$.
    end for
  end for
  $t = t - 1$.
end while

At each time t,
Input: $\boldsymbol{f}_t, l_t$, **Output:** $\boldsymbol{a}_t$.
$\boldsymbol{a}_t = \boldsymbol{\delta}_t^{\boldsymbol{\pi}^*}(\boldsymbol{f}, l)$
Update $\boldsymbol{f}_{t+1}$ according to the action $\boldsymbol{a}_t$.

In spite of the optimality of DP-based solution, it has practical limitations; To find the optimal actions at time $t = 1$, we have to already find all the actions for system states during the interval $[1, T^m]$ and store them which requires high processing capability and huge memory. The complexity is $\mathcal{O}(|\mathcal{T}||\boldsymbol{\mathcal{F}}|)$, where $|\mathcal{T}|$ is the total number of time slots and $|\boldsymbol{\mathcal{F}}|$ is the number of file size states which exponentially increases to the number of flows m. Thus, we derive another threshold-based heuristic algorithm which requires low-processing and low-memory. Moreover, it does not require probabilistic information about future Wi-Fi contacts.

2.4. Threshold-based algorithm

We implement a threshold-based rate control algorithm based on following rationales: (i) It guarantees download completion of all flows within their deadlines with probability 1 when it is possible. (ii) It sequentially allocates the rates in an EDF (Earliest Deadline First) manner. It is because the urgent flow (i.e., a flow with short deadline) has less opportunities for data transfer than other flows. (iii) It tries to increase the offloading efficiency by avoiding cellular data usage as much as possible if there is enough time until the deadline. (iv) The success or failure of download completion within deadline for each flow should be estimated by considering loads (remaining file size, deadline) of all unfinished flows. Then, our threshold-based rate control algorithm can be described as follows.

Threshold-based rate control algorithm

At each time t,
Input: $\boldsymbol{f}_t$, $\boldsymbol{T}$, l_t, r^c and r^w, **Output:** $\boldsymbol{a}_t$.

```
a_t = 0
if l_t = c,
    i = m, τ = ∞.
    while (T^i ≥ t)
        τ = min(τ, T^i) − f_t^i / min(r^c, r^w), i = i − 1.
    end while
    r = (t + 1 − τ)r^c.
else r = r^w.
end if
if r > 0,
    i = min_{i∈I}{i|T^i ≥ t, f_t^i > 0}.
    while (r > 0||i ≤ m)
        a_t^i = min(r, f_t^i), r = r − a_t^i, f_t^i = f_t^i − a_t^i, i = i + 1.
    end while
end if
```

Update $\boldsymbol{f}_{t+1}$ according to the action $\boldsymbol{a}_t$.

When the network state is Wi-Fi, it fully utilizes the Wi-Fi bandwidth by maximally allocating flow rates in an EDF manner. On the other hand, when the network state is cellular, it minimally allocates flow rates in an EDF manner under satisfying deadline completion with the lowest data rate of network, $\min(r^c, r^w)$.

3. Trace-driven simulation

3.1. Simulation settings

In this section, we evaluate our DP-based and threshold-based algorithms through measurements and trace-driven simulations. We consider the case when from one to four flows coexists. We use the dataset of YouTube video size [8] to generate file size because it is fit with delay-tolerant applications. We re-scale the dataset to make the average file size be 100 MB. The deadline of each flow is determined by the sum of two components: (i) Essential delay to download the file with the lowest data rate. (ii) Additional delay that the mobile user can tolerate. Each flow picks its additional delay randomly from the set {10 min, 30 min, 1 h, 2 h}. For the network generation, we fix the data rates of cellular and Wi-Fi networks as time-averaged values based on our measurements. We measure throughputs of cellular network with HSPA, LTE and LTE-A technologies and Wi-Fi network (802.11g) deployed in KAIST campus. The average data rates are 3.3Mbps, 7.2Mbps, 13.3Mbps and 6.9Mbps, respectively. Moreover, we recruit 63 students in KAIST using Android smartphones to collect their Wi-Fi connectivity during 14 days.[5] Next, we define the disutility function as $c(\boldsymbol{f}) = \sum_{i=1}^{m} \lambda f^i$ which is proportional to the sum of unfinished file size of all flows where λ is a delay sensitiveness parameter.[6] As λ becomes larger, our dynamic programming-based scheduler strives to finish the flows earlier by trading Wi-Fi opportunities in the future.

[5] The trace is already introduced in our other work [9].

[6] The disutility function of each flow also can be different depending on the characteristic of application.

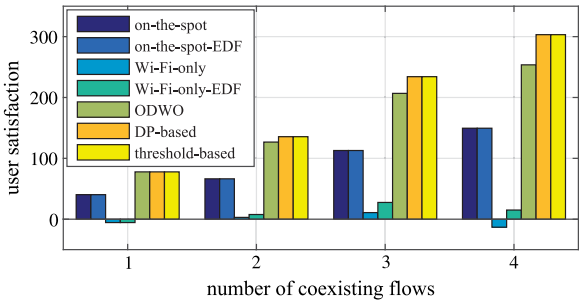

Fig. 1. Satisfaction of DP-based, threshold-based and existing algorithms when $(r^c, r^w) = (7.2, 6.9)$ Mbps, temporal coverage of Wi-Fi = 33% and $\lambda = 2$.

3.2. Comparison algorithms

In our simulations, we compare our proposed rate control algorithms with existing algorithms: *on-the-spot*, *on-the-spot-EDF*, *Wi-Fi-only*, *Wi-Fi-only-EDF* and *ODWO* (Optimal Delayed Wi-Fi Offloading). On-the-spot is a baseline algorithm that activates all the flows simultaneously. The flows equally share the current network resources regardless of network status. Wi-Fi-only activates all the flows simultaneously only when the network state is Wi-Fi. on-the-spot-EDF and Wi-Fi-only-EDF are the EDF versions of on-the-spot and Wi-Fi-only, respectively. ODWO [6] is an optimal flow on/off algorithm for the single-flow scenario.

3.3. Simulation results

Fig. 1 shows the user satisfactions (data volume transferred through Wi-Fi minus disutility) for our two rate control algorithms and existing ones when $(r^c, r^w) = (7.2, 6.9)$ Mbps, temporal coverage of Wi-Fi = 33% and $\lambda = 2$. On-the-spot performs almost zero disutility because it turns on the flows without any delaying. However, it achieves low offloading efficiency because of the loss of future Wi-Fi opportunities. on-the-spot-EDF achieves almost the same with on-the-spot case. The sequential and parallel transmission do not influence on the satisfaction in this case because on-the-spot has enough time to finish all the flows before their deadlines. On the other hand, Wi-Fi-only and Wi-Fi-only-EDF performs high offloading efficiency because they fully utilize the Wi-Fi opportunities. However, high disutility may occur because many unfinished flows exist after deadlines. Therefore, the temporal coverage of Wi-Fi should be carefully considered. Interestingly, there is huge satisfaction gap between Wi-Fi-only and Wi-Fi-only-EDF because they have not enough slots to download the files before their deadlines. ODWO achieves minimum 88% of the satisfaction gain compared to on-the-spot that comes from delaying the transmissions until Wi-Fi contact in some degrees. The on/off control of each flow is optimally determined in an individual sense, but not globally optimal due to a lack of regarding network resource contention among the flows. Our DP-based algorithm optimizes the multi-flow rate control by jointly considering Wi-Fi contact distribution, data rates, traffic load, and disutility function. Surprisingly, our threshold-based algorithm performs almost the same as DP-based algorithm

Table 1
Satisfaction gain of threshold-based algorithm compared to *ODWO*.

Number of flows	1	2	3	4
Satisfaction gain (%)	0.0	7.1	13.3	19.7

with low complexity in our simulation settings. The threshold-based is a conservative algorithm which always considers worst-case of network states in the future to meet the deadlines. Therefore, it performs almost close to optimal for the case when the disutility term is large enough compared to offloading efficiency term (e.g., when λ is high enough). These results demonstrate our rationales for implementing threshold-based algorithm are well-fitted for multi-flow management in mobile environment.

Table 1 shows the satisfaction gain of threshold-based algorithm compared to ODWO for different number of coexisting flows under the same simulation parameters. The gain increases as the number of flows increases. Therefore, the rate control taking into account resource coupling among flows becomes more important as the user uses many delay-tolerant applications, simultaneously.

4. Conclusion

In this paper, so as to increase the user satisfaction, we proposed two multi-flow rate controllers with specified deadlines in a delayed Wi-Fi offloading system. To increase the Wi-Fi utilization and reduce the disutility caused by deadline violations, we considered network resource contention among multiple flows and network transition between cellular and Wi-Fi networks. Through trace-driven simulations, we showed that our dynamic programming-based and threshold-based rate control algorithms drastically increase the user satisfaction compared to existing flow management algorithms that do not consider the network resource contention among flows.

References

[1] Y. Kim, J. Lee, J. Jeong, S. Chong, Multi-flow rate control in delayed Wi-Fi offloading systems, in: Proc. of ICOIN, Kota Kinabalu, Malaysia, 2016, pp. 274–279.
[2] G. Ananthanarayanan, I. Stoica, Blue-fi: Enhancing wi-fi performance using bluetooth signals, in: Proc. of ACM MobiSys, Kraków, Poland, 2009, pp. 249–262.
[3] K. Lee, J. Lee, Y. Yi, I. Rhee, S. Chong, Mobile data offloading: How much can wifi deliver? IEEE/ACM Trans. Netw. 21 (2) (2013) 536–550.
[4] S. Ha, S. Sen, C. Joe-Wong, Y. Im, M. Chiang, Tube: Time-dependent pricing for mobile data, in: Proc. of ACM SIGCOMM, Helsinki, Finland, 2012, pp. 247–258.
[5] N. Ding, D. Wagner, X. Chen, Y.C. Hu, A. Rice, Characterizing and modeling the impact of wireless signal strength on smartphone battery drain, in: Proc. of the ACM SIGMETRICS, Pittsburgh, PA, USA, 2013, pp. 29–40.
[6] M.H. Cheung, J. Huang, DAWN: Delay-aware wi-fi offloading and network selection, IEEE J. Sel. Areas Commun. 33 (6) (2015) 1214–1223.
[7] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, 2005.
[8] Dataset for statistics and social network of YouTube videos. URL http://netsg.cs.sfu.ca/youtubedata/.
[9] J. Jeong, Y. Yi, J.-W. Cho, D. Eun, S. Chong, Energy-efficient wi-fi sensing policy under generalized mobility patterns with aging, IEEE Trans. Netw. 24 (4) (2016) 2416–2428.