Predictive Delay-Aware Network Selection in Data Offloading

Haoran Yu*, Man Hon Cheung*, Longbo Huang†, and Jianwei Huang*
*Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
†Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

***Abstract*—To address the increasingly severe congestion problem in cellular networks, mobile operators are actively considering offloading the cellular traffic to other complementary networks. In this paper, we study the online network selection problem in operator-initiated data offloading with multiple mobile users, taking into account the operation cost, queueing delay, and traffic load in different access networks (e.g., cellular macrocell, femtocell, and Wi-Fi networks). We first design a *Delay-Aware Network Selection* (DNS) algorithm based on the Lyapunov optimization technique. The DNS algorithm yields an operation cost within $O\left(\frac{1}{V}\right)$ bound of the optimal value, and guarantees an $O(V)$ traffic delay for any control parameter $V > 0$. Next, we incorporate the prediction of users' mobilities and traffic arrivals into the network selection. Specifically, we assume that the users' locations and traffic arrivals in the next few time slots can be estimated accurately, and propose a *Predictive Delay-Aware Network Selection* (P-DNS) algorithm to utilize this information based on a novel frame-based design. We characterize the performance bounds of P-DNS in terms of cost-delay tradeoff theoretically. To further reduce the computational complexity, we propose a *Greedy Predictive Delay-Aware Network Selection* (GP-DNS) algorithm, where the operator solves the network selection problem approximately and iteratively. Numerical results show that GP-DNS improves the cost-delay performance over DNS, and reduces the queueing delay by roughly 40% with the same operation cost.**

I. Introduction

Cellular networks worldwide have been facing an unprecedented growth in mobile data traffic. As predicted by Ericsson, the global mobile data traffic will increase by nearly 10-fold between 2013 and 2019 [1]. To tackle such an explosive growth in traffic volume, *mobile data offloading*, where the traffic originally targeted for cellular network is delivered over other complementary networks (such as Wi-Fi [2] or femtocell [3]), is a cost-efficient solution to alleviate the increasingly severe congestion problem in the cellular networks [2].

There are two main approaches in mobile data offloading, namely user-initiated and operator-initiated offloading. In *user-initiated* offloading, each mobile user decides on which network (e.g., cellular or Wi-Fi) its device would connect to. In *operator-initiated* offloading, the *network operator* monitors the network condition, and decides on whether to offload the traffic of some users from the cellular network to the other complementary networks. In fact, recent developments in Hotspot 2.0 of Wi-Fi Alliance (WFA) and the access network discovery and selection function (ANDSF) in the 3rd Generation Partnership Project (3GPP) standard have reflected the desire of the operators to implement the operator-initiated offloading, where an operator has more control on the network choices, the quality of experience of its subscribers, and its own revenue. The detailed network selection policy, however, is not specified in the Hotspot 2.0 and ANDSF standards, and may be implemented differently by different operators. This will be the focus of our study in this paper.

In this work, we consider the network selection problem in the operator-initiated offloading scenario, where the traffic arrivals and locations of mobile users vary over time. We take into account the *operation cost* (e.g., the backhaul, energy, and management cost [3], [4]), *queueing delay*, and *traffic load* in different networks when optimizing the network selection. Specifically, the operator, who has deployed several access networks (e.g., cellular macrocell, femtocell, and Wi-Fi networks), usually prefers to serve its users in a network with the lowest operation cost. However, since some networks (especially femtocell and Wi-Fi networks) do not have ubiquitous coverage, and the policy of serving users in low cost networks may lead to a large delay for users who do not move around very often. Hence, the operator needs to dynamically select network for each user based on the network availabilities, the user mobility, and the QoS requirement. We will focus on designing an efficient *online* network selection policy, which relies on limited or no information of the future, satisfies total traffic demands of the users, and balances both the operation cost and traffic delay.

In the first part of this paper, we apply the Lyapunov optimization framework [5] to design an online *Delay-Aware Network Selection* (DNS) algorithm, which does not require any prior statistical knowledge of the traffic arrivals and positions of the users. DNS yields an operation cost that can be pushed arbitrarily close to the optimal value, at the expense of an increase in the average user queueing delay.

Motivated by the recent advancement of accurate estimation of user mobility [6] and traffic demands [7], in the second part of this paper, we further improve the performance of DNS by incorporating the prediction of users' locations and traffic arrivals into the network selection. Intuitively, with an accurate predication of network information in the next few time slots,

The work of H. Yu, M. H. Cheung, and J. Huang was supported by the General Research Funds (Project Number CUHK 412713 and CUHK 412511) established under the University Grant Committee of the Hong Kong Special Administrative Region, China. The work of L. Huang was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003, 61303195, and the China youth 1000-talent grant.

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

the operator is able to make a more precise network selection decision, and achieve an even better cost-delay performance than DNS. However, designing such a predictive network selection algorithm is challenging. First, the state space grows exponentially with the size of the information window. If we try to analyze such a problem by dynamic programming, the corresponding algorithm will become computationally intractable due to the *curse of dimensionality*. Second, the commonly used multi-slot Lyapunov optimization technique in [8] is not applicable here. The reason is that, it potentially increases the traffic delay, which is in conflict with our goal of considering the delay-aware offloading.

Instead, we design a *Predictive Delay-Aware Network Selection* (P-DNS) algorithm through a novel frame-based approach. Different from the previous Lyapunov optimization technique in [5], [8], we introduce a new controllable parameter θ into the algorithm design. By properly adjusting θ, we can balance the variance of queue length within each frame, and significantly improve the delay performance. We are able to explicitly characterize the performance bounds of P-DNS as functions of θ.

To further reduce the computational complexity of P-DNS, we propose a *Greedy Predictive Delay-Aware Network Selection* (GP-DNS) algorithm, where the operator solves the optimization problem in P-DNS approximately and iteratively. Our numerical results show that GP-DNS achieves a much better cost-delay tradeoff than DNS, and the improvement increases with the the prediction capability of the operator.

To the best of our knowledge, this is the first work that proposes a network selection policy for the operator in a stochastic multi-user data offloading scenario. The main contributions of our work are as follows:

- *Online operator-initiated offloading algorithms for multiple users*: We design online network selection algorithms with and without predictive information on the traffic arrivals and trajectories of the users.
- *Novel frame-based predictive scheduling analysis*: We characterize the operation cost and queueing delay tradeoff theoretically under our novel frame-based predictive network selection design.
- *Performance improvement with prediction*: Simulation results show that GP-DNS improves the cost-delay performance and reduces the queueing delay by roughly 40% over DNS with the same operation cost.

Prior works have considered different aspects of the general operations in data offloading. The work in [9] investigated the optimal offloading strategy for a particular user based on the tradeoff among the throughput, cost, and delay. Iosifidis *et al.* in [10] analyzed a general offloading market, where multiple cellular network operators compete in leasing the access points for data offloading. Cheung *et al.* in [11] applied the congestion game to study the equilibrium outcome of users' interaction in data offloading. However, to the best of our knowledge, there has not been any prior work on the study of network selection strategy in a stochastic multi-user data offloading scenario.

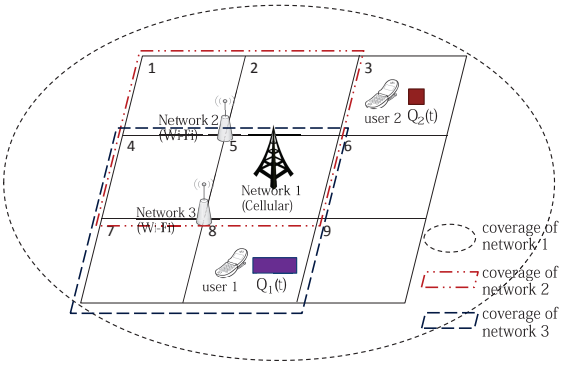

Fig. 1: An example of the system model, where user 1 and 2 are moving within the set $\mathcal{S} = \{1, 2, \ldots, 9\}$. Some locations are only covered by the cellular network (e.g., $\mathcal{N}_3 = \{1\}$), while some locations can access both the cellular and Wi-Fi networks (e.g., $\mathcal{N}_4 = \{1, 2, 3\}$, $\mathcal{N}_8 = \{1, 3\}$). We use $Q_1(t)$ and $Q_2(t)$ to denote user 1 and 2's queue backlog, respectively.

In terms of the *predictive scheduling*, Huang *et al.* in [12] proposed a predictive backpressure algorithm that predicts and pre-serves the traffic arrivals. Here we consider the prediction of both traffic arrivals and users' locations, and do not consider traffic pre-serving, hence propose a predictive algorithm that is completely different from that in [12].

The rest of the paper is organized as follows. In Section II, we introduce the system model. In Sections III and IV, we study the non-predictive and predictive network selection, respectively. We present the numerical results in Section V, and conclude the paper in Section VI.

II. System Model

We consider a slotted system, i.e. $t \in \{0, 1, \ldots\}$, where an operator serves L users in N networks. We use $\mathcal{L} = \{1, 2, \ldots, L\}$ to represent the set of users and use $\mathcal{N} = \{1, 2, \ldots, N\}$ to represent the set of networks. For each network $n \in \mathcal{N}$, we let μ_n be its capacity.[1] We assume that the availability of networks is location-dependent. Let $\mathcal{S} = \{1, 2, \ldots, S\}$ be the set of locations. We use $\mathcal{N}_s \subseteq \mathcal{N}$ to represent the set of available networks at location $s \in \mathcal{S}$.[2] We illustrate the system model through an example in Figure 1.

A. Users' mobilities and traffic arrivals

Users randomly move across the locations with random traffic arrivals. Let $A_l(t)$ be the traffic arrival (measured in bits) of user $l \in \mathcal{L}$ at time slot t. We assume that there exists a constant $A_{\max}$ such that $0 \leq A_l(t) \leq A_{\max}$ for all $l \in \mathcal{L}, t \geq 0$. Let $S_l(t) \in \mathcal{S}$ be user l's location at time slot t. In our model, we assume that both $A_l(t)$ and $S_l(t)$ are random for all $l \in \mathcal{L}$ and t. Hence, we use $\omega_l(t) = (A_l(t), S_l(t))$ to denote the random event experienced by user l at time slot t, and use $\boldsymbol{\omega}(t) = (\omega_l(t), \forall l \in \mathcal{L})$ to denote the random events of the entire system at time slot t.

[1] We assume that the networks' capacities are fixed, while it is easy to generalize the results in this paper to the random capacity case through treating $\mu_n(t)$ as a random event as $\boldsymbol{\omega}(t)$ in Section II-A.

[2] In particular, the cellular network is assumed to cover all the locations. For example, if we use $n = 1$ to represent cellular network, we will have $1 \in \mathcal{N}_s$ for all $s \in \mathcal{S}$.

B. Network selection and transmission rate

At each time slot t, the operator makes the network selection decision for all the users. We denote the decision by $\boldsymbol{\alpha}(t) = (\alpha_l(t), \forall l \in \mathcal{L})$, where $\alpha_l(t)$ is the network that user l is connected to at time slot t. If the operator does not assign user l to any network, then $\alpha_l(t) = 0$. Since the availability of networks is location-dependent, we have

$$\alpha_l(t) \in \mathcal{N}_{S_l(t)} \cup \{0\}, \ \forall l \in \mathcal{L}, t \geq 0. \tag{1}$$

User l's transmission rate at time slot t is a function of the network selection vector $\boldsymbol{\alpha}(t)$, and we denote it by $r_l(\boldsymbol{\alpha}(t))$. We assume that $r_l(\boldsymbol{\alpha}(t))$ is upper bounded by a positive constant $r_{\max}$:

$$0 \leq r_l(\boldsymbol{\alpha}(t)) \leq r_{\max}, \ \forall \boldsymbol{\alpha}(t), l \in \mathcal{L}, \tag{2}$$

and satisfies

$$r_l(\boldsymbol{\alpha}(t)) = 0, \text{ if } \alpha_l(t) = 0. \tag{3}$$

That is, when user l is not assigned to any network, its transmission rate is zero.

We allow a general function of $r_l(\boldsymbol{\alpha}(t))$ that satisfies (2) and (3) in our analysis and algorithm design in Sections III and IV. As an example, if users are homogeneous and the capacity of a network is always evenly shared among all users connected to the network [11], then user l's transmission rate at time slot t is given by

$$r_l(\boldsymbol{\alpha}(t)) = \frac{\mu_{\alpha_l(t)}}{m_{\alpha_l(t)}(\boldsymbol{\alpha}(t))}, \tag{4}$$

where $m_n(\boldsymbol{\alpha}(t)) = |\{l' \in \mathcal{L} : \alpha_{l'}(t) = n\}|$ is the number of users connected to network $n \in \mathcal{N}$ from the entire coverage area of network n. Apparently, $r_l(\boldsymbol{\alpha}(t))$ satisfies (2) and (3).

C. Queueing dynamics

Each user l has a data queue, and $Q_l(t)$ is the queue length of unserved traffic at time slot t. Let $\boldsymbol{Q}(t) = (Q_l(t), \forall l \in \mathcal{L})$ be the queue backlog vector. We assume that all queues are initially empty, i.e.,

$$Q_l(0) = 0, \ \forall l \in \mathcal{L}. \tag{5}$$

The queue length evolves according to the traffic arrival rate and transmission rate as

$$Q_l(t+1) = [Q_l(t) - r_l(\boldsymbol{\alpha}(t))]^+ + A_l(t), \forall l \in \mathcal{L}, t \geq 0. \tag{6}$$

Here $[x]^+ = \max\{x, 0\}$ indicates that the actual amount of served packets cannot exceed the current backlog size.

D. Operator's objective

The operator's operation cost at time slot t is a continuous function of the vector $\boldsymbol{R}(t) = (R_n(t), n \in \mathcal{N})$, where $R_n(t) \triangleq \sum_{l=1}^{L} \mathbb{1}_{\{\alpha_l(t)=n\}} r_l(\boldsymbol{\alpha}(t))$ is the total transmission rate[3] of network n at time slot t. The operation cost is non-decreasing in each entry $R_n(t)$. In order to simplify the notation and emphasize the dependence of the operation cost on the network selection $\boldsymbol{\alpha}(t)$, we use $c(\boldsymbol{\alpha}(t))$ to denote the operation cost at time slot t. We assume that there exists a constant $c_{\max}$ such that

$$0 \leq c(\boldsymbol{\alpha}(t)) \leq c_{\max}, \ \forall \boldsymbol{\alpha}(t). \tag{7}$$

We allow a general function of $c(\boldsymbol{\alpha}(t))$ that satisfies (7) in Sections III and IV. As an example, an explicit form of $c(\boldsymbol{\alpha}(t))$ is

$$c(\boldsymbol{\alpha}(t)) = \sum_{n=1}^{N} u_n R_n(t) = \sum_{l=1}^{L} u_{\alpha_l(t)} r_l(\boldsymbol{\alpha}(t)). \tag{8}$$

In this example, we assume that the operation cost of each network is linear in its total transmission rate, and we use u_n to denote the unit operation cost of network n.

The objective of the operator is to design an online network selection algorithm that minimizes the expected time average operation cost, while keeping the network stable. This can be formulated as the following optimization problem:

$$\min \quad \bar{c} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{c(\boldsymbol{\alpha}(\tau))\} \tag{9}$$

$$\text{subject to} \quad \overline{Q_l} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Q_l(\tau)\} < \infty, \ \forall l \in \mathcal{L}, \tag{10}$$

$$\text{variables} \quad \alpha_l(t) \in \mathcal{N}_{S_l(t)} \cup \{0\}, \ \forall l \in \mathcal{L}, t \geq 0, \tag{11}$$

where (10) is the stability constraint.

III. Network Selection without Prediction

A. Delay-aware network selection

In this section, we assume that there is no prediction of traffic arrivals and users' mobilities, and propose the following algorithm.

Delay-Aware Network Selection (DNS): At each time slot t, the operator:

- Chooses the network selection vector $\boldsymbol{\alpha}(t)$ that solves

$$\min \quad \left[-\sum_{l=1}^{L} Q_l(t) r_l(\boldsymbol{\alpha}(t))\right] + V c(\boldsymbol{\alpha}(t)) \tag{12}$$

$$\text{variables} \quad \alpha_l(t) \in \mathcal{N}_{S_l(t)} \cup \{0\}, \ \forall l \in \mathcal{L}. \tag{13}$$

- Updates the backlog vector $\boldsymbol{Q}(t+1)$ according to (6).

Here, V is a positive parameter. The intuition behind DNS can be understood through the following example. Consider the special case where $c(\boldsymbol{\alpha}(t))$ is defined as in (8), then (12) can be simplified as

$$\min \quad \sum_{l=1}^{L} r_l(\boldsymbol{\alpha}(t)) \left(V u_{\alpha_l(t)} - Q_l(t)\right). \tag{14}$$

Apparently, the operator will serve user l in network n only if network n's unit operation cost $u_n < \frac{Q_l(t)}{V}$. Hence, if user l's queue backlog $Q_l(t)$ is small, the operator will wait for a network with a low operation cost, e.g., Wi-Fi, to serve user l. Since $Q_l(t)$ is small, suspending its traffic does not

[3]Here $\mathbb{1}_{\{\cdot\}}$ is the indicator function, which equals 1 if the event in the brace is true, otherwise it's zero.

incur much delay. On the other hand, if $Q_l(t)$ is large, the big "pressure" will push the operator to serve user l immediately, even through a network with a high operation cost. Therefore, DNS is able to balance the operation cost and the traffic delay.

B. Performance analysis of DNS

For the ease of exposition, we analyze the performance of DNS by assuming that the random event $\boldsymbol{\omega}(t)$ is independent and identically distributed (i.i.d.) over slots. Notice that with the technique developed in [13], we can obtain similar results under Markovian randomness.

Let $\lambda_l = \mathbb{E}\{A_l(t)\}$ be the mean traffic arrival rate of user $l \in \mathcal{L}$, and denote $\boldsymbol{\omega}(t)$'s state space by $\Omega = \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_J\}$. Let $\pi_{\boldsymbol{\omega}_j}$ be the probability that $\boldsymbol{\omega}(t) = \boldsymbol{\omega}_j$, $j = 1, 2, \ldots, J$.

The following assumptions ensure that there exists a network selection algorithm that satisfies the stability constraint (10). We assume that there exists a set of vectors $\{\boldsymbol{\alpha}_k^{(\boldsymbol{\omega}_j)}\}_{j=1,2,\ldots,J}^{k=1,2,\ldots,L+2}$ with $\boldsymbol{\alpha}_k^{(\boldsymbol{\omega}_j)}$ satisfying (1) for all $\boldsymbol{\omega}_j \in \Omega$, and a set of variables $\{\varphi_k^{(\boldsymbol{\omega}_j)}\}_{j=1,2,\ldots,J}^{k=1,2,\ldots,L+2}$ with $\sum_k \varphi_k^{(\boldsymbol{\omega}_j)} = 1$ and $\varphi_k^{(\boldsymbol{\omega}_j)} \geq 0$ for all $\boldsymbol{\omega}_j \in \Omega$ and k, such that

$$\lambda_l - \sum_{\boldsymbol{\omega}_j} \pi_{\boldsymbol{\omega}_j} \sum_k \varphi_k^{(\boldsymbol{\omega}_j)} r_l\left(\boldsymbol{\alpha}_k^{(\boldsymbol{\omega}_j)}\right) \leq -\eta \qquad (15)$$

for some positive η and all $l \in \mathcal{L}$. (15) is commonly assumed in the network stability problems [13]. The intuition is that, we can find a stationary randomized algorithm (which chooses action $\boldsymbol{\alpha}_k^{(\boldsymbol{\omega}_j)}$ with probability $\varphi_k^{(\boldsymbol{\omega}_j)}$ when $\boldsymbol{\omega}(t) = \boldsymbol{\omega}_j$) such that for any $l \in \mathcal{L}$, the expected transmission rate of user l is greater than its mean traffic arrival rate λ_l.

We define c_{av}^{DNS} and Q_{av}^{DNS} as the long-term average operation cost and average queue length of DNS, respectively. Theorem 1 establishes the upper bounds of c_{av}^{DNS} and Q_{av}^{DNS}.

Theorem 1: DNS achieves:

$$c_{av}^{\text{DNS}} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{c(\boldsymbol{\alpha}(\tau))\} \leq c_{av}^* + \frac{B}{V}, \qquad (16)$$

$$Q_{av}^{\text{DNS}} \triangleq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{l=1}^{L} \mathbb{E}\{Q_l(\tau)\} \leq \frac{B+Vc_{\max}}{\eta}, \qquad (17)$$

where c_{av}^* is the optimal expected time average operation cost of problem (9)-(11) and $B = \frac{L\left(A_{\max}^2 + r_{\max}^2\right)}{2}$.

The proof is given in [14]. According to Little's law, the average queue length is proportional to the average delay. Hence, Theorem 1 implies that, by increasing parameter $V > 0$, the operator can push the operation cost arbitrarily close to c_{av}^*, at the expense of the increase in average traffic delay.

IV. Network Selection with Prediction

In this section, we incorporate the prediction of users' mobilities and traffic arrivals into the network selection decision. We firstly propose a Predictive Delay-Aware Network Selection (P-DNS) algorithm, where the operator predicts the future information, and makes the network selection decisions in a frame-based manner. Then we propose a Greedy Predictive Delay-Aware Network Selection (GP-DNS) algorithm to further reduce the computational complexity.

A. The frame-based prediction and network selection model

We consider a frame-based structure, where the k-th ($k \in \{0, 1, \ldots\}$) frame is defined as the time interval that contains slots $kT, kT+1, \ldots, kT+T-1$. We use T to denote the length of each frame, and define $\mathcal{T}_k$ as the set of all time slots within the k-th frame, i.e. $\mathcal{T}_k = \{kT, kT+1, \ldots, kT+T-1\}$.

We assume that at time slot $t = kT$, i.e. the beginning of the k-th frame, the operator accurately predicts $\{\boldsymbol{\omega}(\tau)\}$, $\tau \in \mathcal{T}_k$ (i.e. the knowledge of users' mobilities and traffic arrivals for the whole frame). With this information, the operator runs P-DNS or GP-DNS algorithm at $t = kT$ and makes the network selection decisions $\{\boldsymbol{\alpha}(\tau)\}$, $\tau \in \mathcal{T}_k$, for the entire k-th frame. The structure is shown in Fig. 2.

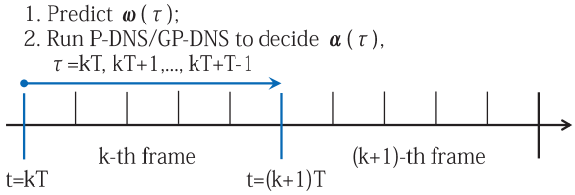

Fig. 2: The Frame-Based Structure.

B. Predictive delay-aware network selection

We present P-DNS as follows:

Predictive Delay-Aware Network Selection (P-DNS): At time slot $t = kT$, $k \in \{0, 1, \ldots\}$, the operator:

- Chooses the network selection vectors $\{\boldsymbol{\alpha}^*(\tau)\}$, $\tau \in \mathcal{T}_k$, that solve problem (18)-(20).

$$\min \sum_{\tau=kT}^{kT+T-1} \left(\sum_{l=1}^{L} Q_l(\tau)(A_l(\tau) - r_l(\boldsymbol{\alpha}(\tau)) + \theta) + Vc(\boldsymbol{\alpha}(\tau)) \right) \qquad (18)$$

$$\text{subject to} \quad Q_l(\tau), \tau \in \mathcal{T}_k, \text{ evolves according to (6)}, \qquad (19)$$

$$\text{variables} \quad \alpha_l(\tau) \in \mathcal{N}_{S_l(\tau)} \cup \{0\}, \ \forall l \in \mathcal{L}, \tau \in \mathcal{T}_k. \qquad (20)$$

- Updates the vector $\boldsymbol{Q}(kT+T)$ according to (6).

Unlike DNS, P-DNS works in a frame-based manner. The basic idea of P-DNS is to balance the average operation cost and the average queue length of each frame. Besides V, we introduce another positive controllable parameter θ in P-DNS, which captures the variance of queue length within each frame. The intuition is that, with θ, the transmission rates of the earlier slots are assigned larger weights than those of the latter slots within the frame. As a result, when the other conditions are the same, serving users in the earlier slots within the frame is better than serving them in the latter slots. This helps to reduce the average queue length (or equivalently, average traffic delay) of each frame. A detailed example that illustrates such an intuition is given in [14].

C. Performance analysis of P-DNS

Similar to DNS, we characterize the performance of P-DNS under the assumption that $\boldsymbol{\omega}(t)$ is i.i.d. over slots and the condition in (15) is satisfied.

First, we define $C(\theta)$ as the optimal expected time average operation cost of problem (9)-(11) with the mean traffic arrival rate $\mathbb{E}\{A_l(t)\}$ increasing from λ_l to $\lambda_l + \theta$ for all $l \in \mathcal{L}$. Apparently, we have

$$\lim_{\theta \to 0} C(\theta) = c_{av}^*. \quad (21)$$

Let $c_{av}^{\text{P-DNS}}$ and $Q_{av}^{\text{P-DNS}}$ be the expected average operation cost and average queue length yielded by P-DNS, respectively. We state the following theorem (see [14] for proof).

Theorem 2: P-DNS achieves

$$c_{av}^{\text{P-DNS}} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{c(\boldsymbol{\alpha}(\tau))\} \leq C(\theta) + \frac{B}{V}, \quad (22)$$

$$Q_{av}^{\text{P-DNS}} \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{l=1}^{L} \mathbb{E}\{Q_l(\tau)\} \leq \frac{B + VC(\theta)}{\theta}, \quad (23)$$

for any $V > 0$ and $\theta \in (0, \eta]$.

Note that P-DNS achieves similar performance bounds as DNS. In particular, (a) when θ approaches 0, the bound for the operation cost achieved by P-DNS is equal to that of DNS in (16). That is,

$$\lim_{\theta \to 0} \left(C(\theta) + \frac{B}{V} \right) = c_{av}^* + \frac{B}{V}, \quad (24)$$

(b) When $\theta = \eta$, the average queue length of P-DNS satisfies

$$Q_{av}^{\text{P-DNS}} \leq \frac{B + VC(\eta)}{\eta} \leq \frac{B + Vc_{\max}}{\eta}, \quad (25)$$

where the right bound is the same as the one specified in (17).

Since P-DNS operates in a larger time scale and the decisions are made based on a joint consideration over the whole frame, the actual gaps between the two sides of the inequalities (22) and (23) are usually much larger than those in (16) and (17). In other words, the actual cost-delay tradeoff achieved by P-DNS can be much better than that by DNS.

D. Greedy predictive delay-aware network selection

We observe that in problem (18)-(20) the set of feasible solutions has a size that is exponentially large in T. This is due to the fact that the state space of the random events grows exponentially with the size of the information window T. To reduce the computational complexity of P-DNS, we propose a greedy algorithm, GP-DNS, where the operator solves problem (18)-(20) approximately.

The basic idea of the greedy algorithm is that, instead of globally searching for the optimal solution in problem (18)-(20), the operator iteratively updates the network selection vectors for different time slots until the values of all vectors converge. For example, when updating vector $\boldsymbol{\alpha}(t)$, $t \in \mathcal{T}_k$, the operator treats all other vectors $\boldsymbol{\alpha}(t')$, $t' \in \mathcal{T}_k$, $t' \neq t$, as given constants, and chooses the feasible $\boldsymbol{\alpha}(t)$ that minimizes the objective function in (18).

The greedy algorithm is proposed as follows:

Greedy Predictive Delay-Aware Network Selection (G-P-DNS): At time slot $t = kT$, $k \in \{0, 1, \ldots\}$, the operator:

Algorithm 1 Greedy Network Selection for the k-th frame

Initialization:

Set $i = 1$ and initialize the network selection vectors for the k-th frame, i.e. $\boldsymbol{\beta}^1(t) = \mathbf{0}$, $\forall t \in \mathcal{T}_k$;

Iteration:

1: **while** $i = 1$ or $\boldsymbol{\beta}^i(t) \neq \boldsymbol{\beta}^{i-1}(t)$ for any $t \in \mathcal{T}_k$ **do**
2: **for** $\tau = kT$ to $kT + T - 1$ **do**
3: Update the network selection vector $\boldsymbol{\beta}^{i+1}(\tau)$:
4: Number all feasible network selection vectors for time slot τ as $\boldsymbol{\alpha}^1(\tau), \boldsymbol{\alpha}^2(\tau), \ldots, \boldsymbol{\alpha}^M(\tau)$;
5: **for** $m = 1$ to M **do**
6: Calculate the value of (18) under the network selection vectors $\boldsymbol{\beta}^{i+1}(kT), \boldsymbol{\beta}^{i+1}(kT+1), \ldots, \boldsymbol{\beta}^{i+1}(\tau-1), \boldsymbol{\alpha}^m(\tau), \boldsymbol{\beta}^i(\tau+1), \boldsymbol{\beta}^i(\tau+2), \ldots, \boldsymbol{\beta}^i(kT+T-1)$;
7: **end for**
8: Choose $\boldsymbol{\beta}^{i+1}(\tau) = \boldsymbol{\alpha}'(\tau)$, where $\boldsymbol{\alpha}'(\tau)$ is the vector that minimizes (18) in line 6 (If multiple vectors result in the same minimum value, choose the vector with the smallest index m) ;
9: **end for**
10: $i \leftarrow i + 1$;
11: **end while**
12: $\boldsymbol{\alpha}^*(t) \leftarrow \boldsymbol{\beta}^i(t), \forall t \in \mathcal{T}_k$;
13: **return** $\boldsymbol{\alpha}^*(t), \forall t \in \mathcal{T}_k$.

- Chooses network selection vectors $\{\boldsymbol{\alpha}^*(\tau)\}$, $\tau \in \mathcal{T}_k$, according to Algorithm 1.
- Updates the vector $\boldsymbol{Q}(kT + T)$ according to (6).

The condition for ending the iteration (line 1) is that all network selection vectors converge, which is always achievable as shown in the following lemma (see [14] for proof).

Lemma 1: In Algorithm 1, for any $\boldsymbol{Q}(kT)$ and $\boldsymbol{\omega}(t)$, $t \in \mathcal{T}_k$, there always exists a finite I such that for any $i \geq I$ we have $\boldsymbol{\beta}^i(t) = \boldsymbol{\beta}^I(t), \forall t \in \mathcal{T}_k$.

V. Numerical Results

In this section, we compare the performance of DNS and GP-DNS in terms of the average operation cost and the average queue length. We also study the amount of data offloaded under these two algorithms.

We simulate DNS and GP-DNS in MATLAB with $|\mathcal{L}| = 4$ users, $|\mathcal{N}| = 8$ networks, and $|\mathcal{S}| = 64$ locations. In particular, we use network 1 to represent the cellular network, which has the highest data rate, 672 Mbps (4G HSPA+), and covers all the locations. The other networks are Wi-Fi networks, and the data rates are normally distributed random variables with means equal to 150 Mbps (IEEE 802.11n) and standard deviations equal to 50 Mbps. These Wi-Fi networks are randomly distributed spatially. Each Wi-Fi network covers at most four connected locations. We consider the transmission rate function $r_l(\boldsymbol{\alpha}(t))$ defined in (4) and operation cost function $c(\boldsymbol{\alpha}(t))$ defined in (8). Markovian dynamics is used to model users' traffic arrivals and locations. We run the experiment for 100000 slots and obtain the following results.

In Figure 3, we plot the average operation cost against the average queue length for DNS and GP-DNS. We obtain these cost-delay tradeoff curves by varying V. As V increases, the average operation costs of the network selection algorithms approach the minimum value, while the average queue lengths become larger. In (16), (17), (22) and (23), the upper bounds for the optimality gap decrease with V, while the upper bounds

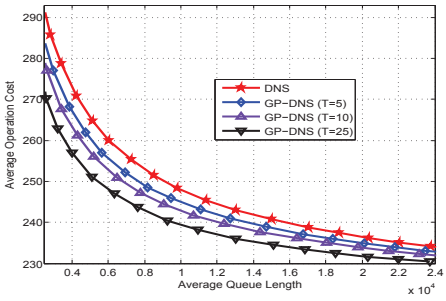

Fig. 3: Cost-Delay Tradeoff of DNS and GP-DNS.

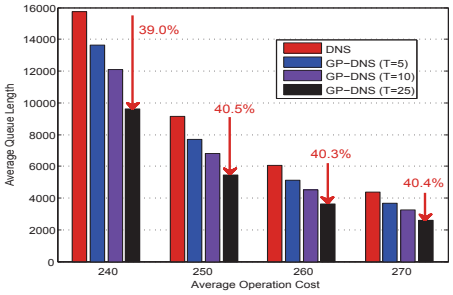

Fig. 4: Delay Reduction through Prediction.

for the average queue length increase with V, which are consistent with our observations here. Comparing DNS with GP-DNS, we observe that the curve of GP-DNS is always below that of DNS, which implies that GP-DNS achieves a better cost-delay tradeoff. Besides, such improvement increases with the operator's prediction capability.

In particular, we set the operation cost target as 240, 250, 260, and 270, find the corresponding average queue lengths yielded by the algorithms in Figure 3, and show their relation in Figure 4. In Figure 4, we plot the average operation cost against the average queue length under different algorithms. We observe that GP-DNS always yields a smaller queue length than DNS, which means that GP-DNS has a smaller average traffic delay than DNS. For example, when the operator pursues an operation cost of 270, GP-DNS with frame size T=25 saves 40.4% delay over DNS, as shown in the last group of bars. The reason is that, the predictive information notifies the operator on whether there will be networks with lower operation costs in the future slots, and whether it worths delaying the users' traffic. As a result, the network selection decisions made in the predictive case are more delay-efficient.

In Figure 5, we compare the volumes of the traffic offloaded to the Wi-Fi networks under DNS and GP-DNS, by plotting the average queue length against the amount of the traffic served in cellular/Wi-Fi network. We find that, when DNS and GP-DNS yield the same average queue length, GP-DNS always offloads more traffic than DNS. The reason is similar to the one we explain in Figure 4, where future information helps the operator to design a better network selection strategy and results in more "successful" offloading.

VI. Conclusions

In this paper, we studied the online network selection problem in the operator-initiated offloading with multiple

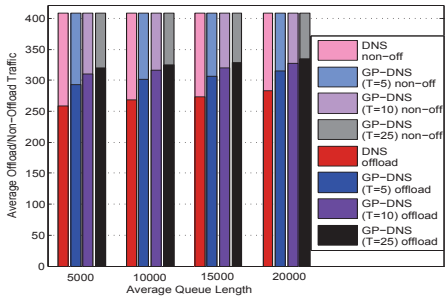

Fig. 5: Fraction of Offloaded Data under DNS and GP-DNS.

users. We first proposed the DNS algorithm, which achieves a close-to-optimal operation cost. We then proposed the P-DNS algorithm and the GP-DNS algorithm by incorporating the prediction of users' mobilities and traffic arrivals in a frame-based manner into the network selection. Simulation results showed that the predictive information helps the operator achieve a more efficient data offloading. We believe that the improvement gained by the predictive algorithms depends on the underlying system randomness (e.g., users' mobilities and traffic arrivals). We are interested in analytically characterizing such a relation in our future work.

References

[1] Ericsson, "Ericsson mobility report," Tech. Rep., November 2013.
[2] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can WiFi deliver?" *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 536–550, April 2013.
[3] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell networks: A survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, September 2008.
[4] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: Technical and business perspectives," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 104–112, April 2013.
[5] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.
[6] A. J. Nicholson and B. D. Noble, "Breadcrumbs: forecasting mobile connectivity," in *Proc. of ACM MobiCom*, San Francisco, CA, September 2008, pp. 46–57.
[7] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in *Proc. of IEEE INFOCOM*, Shanghai, China, April 2011, pp. 882–890.
[8] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. J. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. of IEEE INFOCOM*, Orlando, FL, March 2012, pp. 1431–1439.
[9] M. H. Cheung and J. Huang, "Optimal delayed Wi-Fi offloading," in *Proc. of IEEE WiOpt*, Tsukuba Science City, Japan, May 2013, pp. 564–571.
[10] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "An iterative double auction for mobile data offloading," in *Proc. of IEEE WiOpt*, Tsukuba Science City, Japan, May 2013, pp. 154–161.
[11] M. H. Cheung, R. Southwell, and J. Huang, "Congestion-aware network selection and data offloading," in *Proc. of IEEE CISS*, Princeton, NJ, March 2014, pp. 1–6.
[12] L. Huang, S. Zhang, M. Chen, and X. Liu, "When backpressure meets predictive scheduling," in *Proc. of ACM MobiHoc*, Philadelphia, PA, August 2014.
[13] L. Huang and M. J. Neely, "Max-weight achieves the exact $[O(1/V), O(V)]$ utility-delay tradeoff under markov dynamics," *arXiv preprint arXiv:1008.0200*, 2010.
[14] H. Yu, M. H. Cheung, L. Huang, and J. Huang, "Predictive delay-aware network selection in data offloading," http://jianwei.ie.cuhk.edu.hk/publication/ReportPDNS.pdf.