Towards Automated Network Management: Learning the Optimal Protocol Selection

Xiaoxi Zhang*, Siqi Chen†, Youngbin Im‡, Maria Gorlatova§, Sangtae Ha†, Carlee Joe-Wong*
*Department of Electrical and Computer Engineering, Carnegie Mellon University, USA
†Department of Computer Science, University of Colorado Boulder, USA
‡School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, South Korea
§Department of Electrical and Computer Engineering, Duke University, USA
Email: *{xiaoxiz2,cjoewong}@andrew.cmu.edu, †{siqi.chen,sangtae.ha}@colorado.edu
‡ovinee@gmail.com, §maria.gorlatova@duke.edu

***Abstract*—Today's Internet must support applications with increasingly dynamic and heterogeneous connectivity requirements, such as video streaming and the Internet of Things. Yet current network management practices generally rely on pre-specified flow configurations, which cannot cover all possible scenarios. In this work, we instead propose a model-free learning approach to automatically optimize the policies for heterogeneous network flows. This approach is attractive as no existing comprehensive models quantify how different policy choices affect flow performance under dynamically changing network conditions. We extend multi-armed bandit frameworks to propose new online learning algorithms for protocol selection, addressing the challenge of policy configurations affecting the performance of multiple flows sharing the same network resources. This performance coupling limits the scalability and optimality of existing online learning algorithms. We theoretically prove that our algorithm achieves a sublinear regret and demonstrate its optimality and scalability through data-driven simulations.**

***Index Terms*—Network protocol selection, completion time minimization, multi-armed bandit, online algorithm design**

I. Introduction

The Internet today is diversifying in terms of both the applications and devices that it aims to support, as well as the means for doing so. Applications like virtual reality, for instance, require increasingly low latencies [1], while the Internet-of-Things has dramatically expanded the range of devices connected to the Internet [2]. Fifth-generation (5G) wireless networks are simultaneously predicted to integrate several different access frequencies in an effort to boost capacity and coverage [3]. Yet this heterogeneity comes with challenges: it is far from clear how the network can enforce heterogeneous application requirements when the applications share limited bandwidth on heterogeneous network links.

Current network management practices generally rely on static pre-configurations, e.g., pre-specifying the routing algorithms used to determine flow paths. Initiatives like network functions virtualization (NFV) and the RAN (radio access network) Intelligent Controller [4] aim to enable more flexible policies, but they still require manual intervention to change the preset network policies [5]. In this work, we recognize that pre-specified policies are likely insufficient to handle all possible scenarios. While a vast array of possible policies at various layers of the stack can be used for different types of networks, in this work we focus on the selection of protocols for flows on each link of a given network. We suppose that a given set of protocols is available and that each flow's achieved performance depends on the protocols chosen for all flows on shared links and the unknown network condition. We develop new, model-free algorithms that learn this unknown relationship between protocol choices and optimize the aggregate flow performance over time. Table I summarizes the network topology and protocol choices considered in each use case.

TABLE I: Outline of formulation use cases.

Use case	Topology	Example protocols
5G	Single link	Control channel size
MAC	Single link	CSMA/CD, CSMA/CA
Network slicing	Arbitrary	Slice reservation priorities
Transport layer	Arbitrary	TCP CUBIC, TCP Reno

Our contributions. We derive and validate, analytically and empirically, the *first algorithms that can learn the assignment of protocols to flows that maximizes aggregate flow performance*. We take the first steps towards meeting these challenges with a new extension of the multi-armed bandit (MAB) framework [6]. We view the selection of each set of protocols for all flows in the network as an "arm" to be pulled and decode each arm's "reward" from the transmission rates achieved by all present flows to learn a set of protocol candidates. We then execute an online protocol selection to minimize the aggregate flow completion times, which extends the existing MAB algorithms by: reducing the exponential number of arms (combinations of protocols) to be polynomial with the network size and independent with the number of co-existing flows; and accounting for (1) a completion time objective, which is a highly nonlinear function of the protocol performance that also depends on past protocol selections, and (2) constraints (bandwidth capacities) on the achievable reward that may be unknown or adversarial.

II. Problem Formulation

We consider a communication network described by a graph $G(\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ represents the nodes, corresponding to the

978-1-7281-2700-2/19/$31.00 2019 © IEEE

locations of routers in the network, and $\mathcal{L}$ represents the links. We divide time into discrete slots, each of which lasts $\epsilon > 0$ seconds. Each link l has a bandwidth capacity B_{lt} (bps) in time slot t, which is not known before time t. These links may be virtual links in an overlay network, with unknown or dynamic physical topologies. In a wireless setting, we can view a "link" as a wireless link between UEs and a base station. Network slice configurations and transport protocols, on the other hand, generally are not changed at physical links, but can be changed at some intermediate points, e.g., when flows enter a new network domain or via network proxies that forward packets on behalf of the sender for the next link such as Dropbox does in their datacenter networks [7]. Suppose n flow requests arrive at time t_i over the lifetime of the system, T. Each flow i has a fixed path $\mathcal{P}_i$ exogenously determined at the time of its arrival and a fixed size π_i, *i.e.,* the amount of data (in bytes), to transfer along $\mathcal{P}_i$.

Our algorithm chooses one out of M protocols for each flow i on its path, $\mathcal{P}_i$, so as to minimize the flows' overall completion time. Let x_{ilm} represent whether protocol $m \in [M]$ is chosen ($x_{ilm} = 1$) for flow i on link l, or not ($x_{ilm} = 0$). Let $r_{ilt}(\mathbf{x}_l)$ denote the transmission rate achieved on link l for flow i at time slot t, which is revealed after we choose the protocols for flow i. As different flows may compete for the bandwidth on one or more links, $r_{ilt}(\mathbf{x}_l)$ is a function of the decisions x_{jlt} for all alive flows j on l at t. Let $\delta_{il}(\mathbf{x}_l)$ denote the transmission delay on each link l of flow i. Then $\delta_{il}(\mathbf{x}_l) = \max_{t_i \le t \le +\infty} t \cdot \mathbf{1}(\sum_{t_i \le t' \le t} \epsilon r_{ilt'}(\mathbf{x}_l) \le \pi_i)$, where $\mathbf{1}(X)$ equals 1 if X is true; and 0 otherwise. We then let $\tau_i(\mathbf{x}) = t_i + \max_{l \in \mathcal{P}_i} \delta_{il}(\mathbf{x}_l) +$ propagation delay denote the completion time of flow i, *i.e.,* its arrival time plus the total delay of serving flow i. Since our protocol choices do not affect the propagation delay, our goal is to minimize $\max_{l \in \mathcal{P}_i} \delta_{il}(\mathbf{x}_l)$, and our optimization problem is then:

$$\text{minimize} \sum_{i \in [n]} \max_{l \in \mathcal{P}_i} \delta_{il}(\mathbf{x}_l) \tag{1}$$

$$\text{subject to:} \sum_{i \in \mathcal{A}_{lt}} r_{ilt}(\mathbf{x}_l) \le B_{lt}, \ \forall l \in \mathcal{P}_i, i \in [n], t \in [T] \tag{2}$$

$$\sum_{t_i \le t \le \tau_i(\mathbf{x})} r_{ilt}(\mathbf{x}_l) \ge \pi_i, \ \forall l \in \mathcal{P}_i, i \in [n] \tag{3}$$

$$\sum_{m \in [M]} x_{ilm} = 1, \ \forall l \in \mathcal{P}_i, i \in [n] \tag{4}$$

This formulation reflects our research challenges: $r_{ilt}(\mathbf{x}_l)$ is a function of the protocol choices of all alive flows at t, accounting for competition between flows, and both r_{ilt} and B_{lt} are unknown (B_{lt} can be adversarial: arbitrarily chosen by the environment). Thus, the problem is complex and difficult to solve. In some cases, prior models exist for r_{ilt}, e.g., network utility maximization (NUM) frameworks for certain MAC and TCP protocol variants [8], [9]. However, such models are not available for all protocols and may require knowing the bandwidth capacity, so we assume the r_{ilt} are unknown random variables that may depend on B_{lt}.

Algorithm 1: Online Protocol Selection via Learning Bandwidth Competition – OPSBC

Input: $G(\mathcal{V}, \mathcal{L})$, n, α
Output: $\mathbf{x}$
Initialize: $\mathbf{x} = \mathbf{0}$, $\eta = \mathbf{1}$, $t = 0$
while *time slot* $1 \le t \le T$ *starts* **do**
  for *each link* $l \in \mathcal{L}$ **do**
    Update $\tilde{\pi}_{ilt} =$ (remaining size of each alive flow);
    Update $i^* = \text{argmin}_{i \in \mathcal{A}_{lt}} \tilde{\pi}_{ilt}$;
    Choose $(m^*, m^b) = \text{argmin}_{(m,m')} \eta_{lt}(m, m')$;
    Update $x_{i^* l m^* t} = 1$;
    Update $x_{ilm^b t} = 1, \forall i \ne i^*$;
    Update $\eta_{lt}(m, m')$ and $\eta_{lt}^{LCB}(m^*, m^b)$ using (7) ;
  end
end

III. Online Protocol Selection

Our key insight in solving the optimization problem (1) – (3) is to understand and exploit the relationship among transmission rates, joint protocol decisions of flows, and bandwidth capacities. We first define a model of how the link bandwidth is divided between coexisting flows, depending on the protocols they use, and the resulting flow transmission rates. As shown in (5) below, the transmission rate achieved by each flow will be proportional to the *weight* of its corresponding protocol, divided by the total weights of other flows' protocols. We assume that on each link l, the weight vector $(w_{lt}(e_1), \cdots, w_{lt}(e_M))$ is i.i.d. drawn from an unknown distribution $\mathcal{D}_l^w$ over all times, e.g., due to fluctuations in wireless signal strength or routing in an overlay network.

$$r_{ilt}(\mathbf{x}_l) = \frac{w_{lt}(\vec{x}_{ilt}) B_{lt}}{\sum_{i' \in \mathcal{A}_{lt}} w_{lt}(\vec{x}_{i'lt})} \tag{5}$$

$$\text{where: } u(\mathbf{x}_l, B_{lt}) \le 1, \forall \vec{x}_l, l \in \mathcal{P}_i, i \in [n], t \in [T]$$

Based on this model, we choose protocols by learning the distribution of each weight vector, as shown in Algorithm 1.

Algorithm intuition. If the distributions of the weight vectors and the current capacities are known, the problem (1–3) becomes a pure online decision making problem, with unknown arrival times and sizes of future flows and future bandwidth capacities. Moreover, minimizing the total flow time (1) is equivalent to minimizing the number of alive flows at each time. Therefore, the offline optimum would make the flow with the shortest remaining time finish first so as to reduce the number of alive flows. Guided by these intuitions, we propose to greedily choose the protocols on each link at each time so as to minimize the remaining time of the flow with the smallest amount of un-transferred data on the link.

Distributed MAB algorithm for predictions. Inspired by classic MAB algorithms, we predict the protocols with the highest and lowest expected weights on each link. However, our protocol decisions do not directly map to arms. If we call a protocol decision vector for all flows an "arm," we obtain

$M^{|\mathcal{L}|\times|\mathcal{A}_t|}$ arms at each time t, which is too large to effectively sample. Moreover, *we cannot directly observe the protocol weights by recording the rates r_{ilt}, due to the time-varying bandwidth capacity.* For instance, we might observe $10Mbps$ and $20Mbps$ achieved by protocols 1 and 2 at the first time and $50Mbps$ and $500Mbps$ achieved by protocols 3 and 4 at the next time. However, we cannot interpret $(\frac{10}{580}, \frac{20}{580}, \frac{50}{580}, \frac{500}{580})$ as the rewards (weight samples) of these four arms: *the rates do not translate into weights for protocols that are present at different times under different capacities*. To address this, we discover that, on each link in each time, we only need to observe the weight ratio of each *pair* of protocols to predict the best and worst protocols in expectation. We first define:

$$\eta_{lt}(m,m') = \frac{\text{average flow rate on } l \text{ under } m' \text{ at } t}{\text{average flow rate on } l \text{ under } m \text{ at } t}. \quad (6)$$

We then run a Lower Confidence Bound (LCB) algorithm independently on each link to estimate $\mathbb{E}[\eta_{lt}(m,m')]$, which equals $\mathbb{E}\left[\frac{w_{lt}(e_{m'})}{w_{lt}(e_m)}\right]$. In total, we have only $|\mathcal{L}| \times M^2$ arms, alleviating our research challenge of too many sets of protocol choices. Let $x_{lt}(m,m')$ be the indicator variable which equals 1 if we choose the protocol pair (m,m'); and 0 otherwise. Let $R_{lt}^{m,m'} = \sqrt{\frac{\alpha \log t}{\sum_{t'=1}^{t} x_{lt'}(m,m')}}$, the LCB of $\mathbb{E}[\eta_{lt}(m,m')]$, denoted by $\eta_{lt}(m,m')$, is defined as

$$\eta_{lt}(m,m') = \frac{\sum_{t'=1}^{t} \eta_{lt'}(m,m') x_{lt'}(m,m')}{\sum_{t'=1}^{t} x_{lt'}(m,m')} - R_{lt}^{m,m'} \quad (7)$$

Online protocol selection. At time slot t, on each link, we choose the protocol pair (m,m') that minimizes $\eta_{lt}(m,m')$, denoted as (m^*,m^b) (line 5 of Alg. 1) and assign m^* to the shortest alive flow (indexed by i^*) and m^b to all the other alive flows (lines 6 and 7). By doing so, we guarantee that the shortest flow gets the highest transmission rate if our predictions are accurate, namely $\eta_{lt}(m,m') = \mathbb{E}[\eta_{lt}(m,m')]$. To see this, note that line 5 guarantees that $\frac{w_{lt}(e_{m^b})}{w_{lt}(e_{m^*})} \leq \frac{w_{lt}(e_m)}{w_{lt}(e_{m'})}$ for all protocol pairs (m,m'), if $\eta_{lt}(m,m') = \mathbb{E}[\eta_{lt}(m,m')]$. Let $\mathbf{x}_{lt}^*$ and $\mathbf{x}_{lt}$ denote our protocol decision and any other feasible protocol decision at t, respectively. We can then show that the transmission rate achieved by the shortest flow i^* under our decisions at t is no smaller it would have been under any other decision at t, given the same decisions at other time slots:

$$\begin{aligned} r_{i^*lt}(\mathbf{x}_{lt}^*) &= B_{lt}\left(\frac{(|\mathcal{A}_{lt}|-1)w_{lt}(e_{m^b}) + w_{lt}(e_{m^*})}{w_{lt}(e_{m^*})}\right)^{-1} \\ \geq & B_{lt}\left(1 + \sum_{i\in\mathcal{A}_{lt}\setminus\{i^*\}} \frac{w_{lt}(\vec{x}_{ilt})}{w_{lt}(\vec{x}_{i^*lt})}\right)^{-1} = r_{i^*lt}(\mathbf{x}_{lt}), \ \forall \mathbf{x}_{lt} \quad (8) \end{aligned}$$

The inequality in (8) is due to the strategies in lines 5 – 7.

We evaluate our algorithm by upper-bounding its *Regret*, *i.e.,* the expected difference of the total completion time between that in our algorithm and the offline optimum. Here, the offline optimum is an omniscient algorithm that has perfect knowledge of: (1) the arrival times and sizes of all flows, (2) the bandwidth capacities on each link over all time slots, and (3) the distribution of each weight vector. Formally,

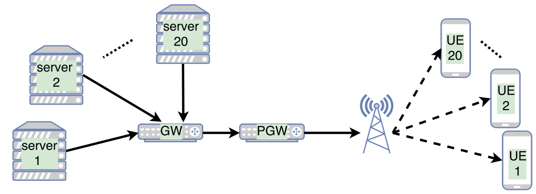

Fig. 1: Topology used in our ns-3 simulation.

$$Regret = \mathbb{E}\left[\sum_{i\in[n]} \tau_i(\mathbf{x}_i)\right] - \min_{\mathbf{x}} \mathbb{E}\left[\sum_{i\in[n]} \tau_i(\mathbf{x})\right] \quad (9)$$

Let $\hat{\eta}_l^{max} = \max_{(m,m',t)} \mathbb{E}[\frac{w_{ltm}}{w_{ltm'}}]$, which represents the maximum expected ratio of the weights of any two protocols on link l over all t. Let η_{min} denote the minimum difference of $\mathbb{E}[\eta_{lt}(pm_1)]$ and $\mathbb{E}[\eta_{lt}(pm_1)]$ between any two protocol pairs pm_1 and pm_2 on any link, B_{max} and B_{min} denote the largest and smallest capacity over all links over all time, w_{max} and w_{min} denote the largest and smallest expected weights over all protocols and all links. We have the following theorem:

Theorem III.1. *If all the flows share the same path, the weight vector for each link is i.i.d., then the regret of our algorithm OPSBC will be upper-bounded by*

$$O\left(\frac{\epsilon B_{max} w_{max} |\mathcal{P}| M^2 \log T}{B_{min} w_{min} \eta_{min}^2}\right),$$

if we have $\frac{B_{lt}}{B_{l't}} \leq \frac{\hat{\eta}_l^{max}}{\hat{\eta}_{l'}^{max}} \leq \frac{B_{l't}}{B_{lt}}, \ \forall (l',l) : B_{l't} \geq B_{lt}, t \in [T]$.

IV. Experimental Validation

In this section, we validate our theoretical results. We simulate 500 flows arriving at the network according to a Poisson Process with an arrival rate of 0.8. Each flow takes a randomly chosen path in the network with size chosen uniformly within $[20, 60]$ (Mb). We compare our OPSBC algorithm's results with several heuristics: **HomoPS** always chooses the same protocol for all the alive flows on each link, which corresponds to using the same default protocol for all flows. In each time slot, **Random** randomly chooses a protocol pair for each flow on each link. **FixedPS** randomly chooses a protocol on all links for each flow at the first time slot without changing the decisions over time.

We test the algorithm on performance data from ns-3 [10] TCP simulations. Figure 1 shows the setup of our ns-3 [10] simulator. We assume that 20 mobile nodes each receive data from a dedicated server over an LTE network with a crab topology; the server-PGW links have 100Mbps capacity while the PGW-BS link has 20 Mbps capacity, making it the network bottleneck. Each node can use one of five transport protocols: UDP, TCP CUBIC, TCP NewReno, TCP Vegas, or TCP Westwood. Nodes running UDP saturate their flows at 1 Mbps. We run 20 flows for each protocol pair in 30 different scenarios (*e.g.,* with varying mobility of the mobile nodes) and measure their throughput and delay on each link. Figure 2

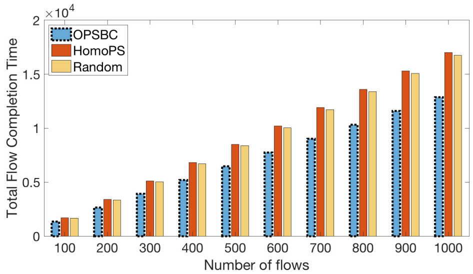

Fig. 2: OPSBC achieves lower flow-time on `ns-3` data traces.

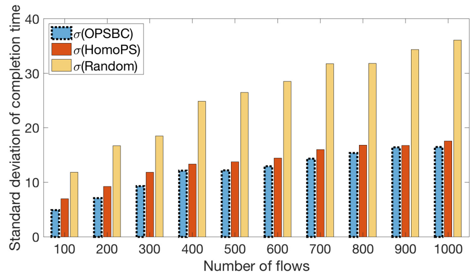

Fig. 3: OPSBC achieves a lower standard deviation of flow-time than heuristic algorithms on `ns-3` data traces.

compares the total flow-time of OPSBC (Alg. 1) to that of the **HomoPS** and **Random** heuristics. These two heuristics show a 30% larger total flow-time than our Alg. 1 when 1000 flows are processed, indicating that OPSBC outperforms both a priori fixed and random protocol selections. We also find the standard deviation of the total flow-time ($\sigma(\cdot)$) of **OPSBC**, **HomoPS** and **Random**. Figure 3 shows that **OPSBC** has the smallest standard deviations, indicating it consistently performs well over the randomly repeated experiments.

V. Related Work

Machine learning has recently been adopted to design TCP congestion control protocols and help with network management. Winstein *et al.* [11] design Remy, a program that can generate distributed congestion control algorithms in a multi-user network. Mao *et al.* [12] use deep reinforcement learning to allocate cloud resources to minimize job slowdowns. In contrast, we provide *theoretical* guarantees of our online algorithms' solution optimality. We also compare our work to prior work on making decisions in network settings when data gradually become available over time. Chen et al. [13] design novel algorithms for online convex optimization problems with switching costs. Zhang *et al.* [14] integrate the online gradient descent method into online cloud resource provisioning. However, these studies assume full feedback on all feasible solutions. MAB approaches, such as the one we propose in this work, instead only use information from the chosen decisions and have been adopted in various scenarios, e.g., dynamic channel access [15] and cloud job scheduling [16]. Unlike these works, we do not directly optimize the rewards of our chosen protocol arms, but instead use them as estimates of unknown inputs needed by an additional algorithm to optimize the protocols.

VI. Conclusion

To cope with flows' increasingly heterogeneous requirements and changing network characteristics, we propose a dynamic network management framework that leverages existing network protocols. We use model-free online learning to support automatic protocol selection for each individual flow, so as to optimize the overall flow completion time. Motivated by the deficiency of existing models for flow performance under different protocol choices, we propose a model to characterize coexisting flows' transmission rates under different protocols. We then extend multi-armed bandit algorithms to learn the rate function and predict an optimal assignment of protocols to flows at each time. Taking these real-time predictions as input, we then propose a provably optimal online protocol selection scheme that can minimize the aggregate flow completion time. The asymptotic optimality of our learning and assignment algorithm is validated through theoretical analysis and experiments.

References

[1] C. Westphal, "IETF presentation: Challenges in networking to support augmented reality and virtual reality," Mar. 2017.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] C.-X. Wang, F. Haider, X. Gao, X.-H. You, Y. Yang, D. Yuan, H. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular architecture and key technologies for 5g wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, 2014.

[4] O-RAN Alliance, "O-ran: Towards an open andsmart ran," https://www.o-ran.org/, 2018.

[5] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[6] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121 – 2159, 2011.

[7] R. Bhargava, "Evolution of Dropbox's edge network," Dropbox, 2017, https://blogs.dropbox.com/tech/2017/06/evolution-of-dropboxs-edge-network/.

[8] S. H. Low, "A duality model of tcp and queue management algorithms," *IEEE/ACM Transactions On Networking*, vol. 11, no. 4, pp. 525–536, 2003.

[9] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models for the latency and steady-state throughput of tcp tahoe, reno, and sack," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, no. 6, pp. 959–971, 2003.

[10] "NS-3 network simulator," https://www.nsnam.org/, 2018.

[11] K. Winstein and H. Balakrishnan, "Tcp ex machina: computer-generated congestion control," in *In Proc. of ACM SIGCOMM*, vol. 43, no. 4. ACM, 2013, pp. 123–134.

[12] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *In Proc. of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 2016, pp. 50–56.

[13] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on the past," in *In Proc. of ACM SIGMETRICS*, 2016.

[14] X. Zhang, C. Wu, Z. Li, and F. C. Lau, "Proactive vnf provisioning with multi-timescale cloud resources: Fusing online learning and online optimization," in *In Proc. of IEEE INFOCOM*, 2017.

[15] Y. Liu and M. Liu, "An online approach to dynamic channel access and transmission scheduling," in *In Proc. of MobiHoc*, 2015.

[16] X. Zhang, C. Wu, Z. Huang, and zongpeng Li, "Occupation-oblivious pricing of cloud jobs via online learning," in *Proc. of INFOCOM*, 2018.