Optimal Learning-Based Network Protocol Selection

Xiaoxi Zhang
Carnegie Mellon University
xiaoxiz2@andrew.cmu.edu

Siqi Chen
University of Colorado Boulder
Siqi.Chen@colorado.edu

Youngbin Im
University of Colorado Boulder
Youngbin.Im@colorado.edu

Maria Gorlotova
Duke University
maria.gorlatova@duke.edu

Sangtae Ha
University of Colorado Boulder
Sangtae.Ha@colorado.edu

Carlee Joe-Wong
Carnegie Mellon University
cjoewong@andrew.cmu.edu

ABSTRACT

Today's Internet must support applications with increasingly dynamic and heterogeneous connectivity requirements, such as video streaming, virtual reality (VR), and the Internet of Things. Yet current network management practices generally rely on pre-specified flow configurations, which may not be able to cope with dynamic flow priorities or changing network conditions, e.g., on volatile wireless links. In this work, we instead propose a model-free learning approach to find the optimal network policies for current flow requirements. This approach is attractive as comprehensive models do not exist for how different protocol choices affect flow performance, which may further be affected by dynamically changing network conditions. However, it raises new technical challenges: policy configurations can affect the performance of multiple flows sharing the network resources, and this flow coupling cannot be readily handled by existing online learning algorithms. In this work, we extend multi-armed bandit frameworks to propose new online learning algorithms for protocol selection with provably sublinear regret under certain conditions. We validate the algorithm performance through testbed experiments, demonstrating the superiority and scalability of our proposed algorithms.

1. 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]. 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, or require manual intervention to change the preset network policies [3], e.g., network functions virtualization (NFV) and the RAN (radio access network) Intelligent Controller [4]. Comprehensive models for flow performance under different policies generally do not exist, and it is unclear which policies can actually optimize over all possible scenarios and combinations of flow requirements.

In this work, we recognize that pre-specified policies are likely insufficient to handle all possible scenarios, and we focus on the selection of protocols for flows at 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, in an unknown manner that depends on prevailing network conditions. We develop algorithms that learn this unknown relationship between protocol choices and flow performance and demonstrate that they learn the optimal assignment of protocols to flows over time.

1.1 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 define "performance" as the total flow completion time, but our framework generalizes to other objectives. In doing so, we solve several research challenges:

Unpredictable, time-varying network conditions. The exact relationship between protocol selection and flow performance depends on the presence of background traffic and the overall network conditions: e.g., on an unreliable wireless link, conventional TCP may not perform well [5]; and background flows may take up an unknown amount of bandwidth.

Competition between different flows. When different protocols may be used for different flows, the choice of protocol for one flow affects the performance of other flows on the same link. This coupling between flows that may arrive at different times is particularly hard to manage given the absence of models for how protocol choices affect flow performance and the nonlinearity of our objective function.

A large set of protocol choices. The number of possible protocol assignments to flows can be exponential with the number of possible protocol choices when we allow different protocol decisions for different flows. Therefore, we cannot sample all possible protocol assignments.

We take the first steps towards meeting these challenges with a new extension of the multi-armed bandit (MAB) framework [6]. As flows arrive and depart, we select "arms" (protocol combinations) and decode each arm's "reward" from the transmission rates achieved by all present flows. We use the historical rewards from prior flows to learn a set of protocol

candidates and then execute an online protocol selection to minimize the aggregate flow completion times. While MAB algorithms can learn arbitrary distributions of arm rewards, they do not address our challenges: an exponential number of protocol combinations for different flows, a highly nonlinear objective function depending on past protocol selections, or constraints (bandwidth capacities) on the achievable reward that may be unknown or adversarial.

2. PROBLEM FORMULATION

We consider a communication network described by a graph $G(\mathscr{V},\mathscr{L})$, with links $\mathscr{L}$ and terminals (nodes) $\mathscr{V}$, which correspond to the locations of routers in the network. We divide time into discrete slots, each of which lasts ε seconds ($\varepsilon > 0$); in practice, ε would depend on the type of protocol used. Each link l, which may be a virtual link in an overlay network, has a bandwidth capacity B_{lt} (bps) in time slot t, which is not known before time t. Suppose n flow requests (*e.g.*, downloads of files or videos to be streamed offline) each arrive at the network at time t_i over the lifetime of the system, T. Each flow i has a source and destination in $\mathscr{V}$, a fixed path $\mathscr{P}_i$ that is 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 $\mathscr{P}_i$. We consider joint path and protocol selection in our technical report [7].

Protocol selection. Different types of protocols may be selected by different entities in a network, whether set on a link-by-link basis by intermediate routers for all flows on the same link (e.g., MAC protocols), or by each flow's source for its entire path (e.g., TCP/UDP protocols). To accommodate both per-link and per-flow protocol selection, we suppose we have a total of M protocol choices on each link for each flow.[1] We then define a binary indicator variable x_{ilm}, to represent whether protocol $m \in [M]$ is chosen ($x_{ilm} = 1$) for flow i on link l, or not ($x_{ilm} = 0$). Since flows can use only one protocol on each link at a time, our protocol decisions have to satisfy the constraint $\sum_{m\in[M]} x_{ilm} = 1$, $\forall l \in \mathscr{P}_i, i \in [n]$. Choosing the same protocol for all flows on a given link can be enforced via additional (linear) constraints on the protocol choices $\{x_{ilm}\}$. Our goal is to select the protocols for each flow so as to minimize the flows' overall completion time.

Table 1 summarizes the network topology and protocol choices for four different use cases. By a "link" in the network, we mean a subset of the network over which a single protocol is chosen for each flow. For instance, a "link" is a wireless link between UEs and a base station in a wireless setting, or a domain within a network slice that is configured [8] to enforce slice performance requirements. Transport protocols can also be deployed on a link-by-link basis in overlay networks via network proxies that forward packets according to the best transport-layer protocol for the next link, e.g., as Dropbox and Google do in their datacenter networks [9, 10] or as may be necessary for edge computing involving wired and wireless network links [11, 12].

Total completion time minimization. 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 share one or more links and compete for the bandwidth on each link, $r_{ilt}(\mathbf{x}_l)$ is a function of decisions for all alive flows on l at t. Let $\delta_{il}(\mathbf{x}_l)$ denote the transmission delay on each link l of flow i. We have $\delta_{il}(\mathbf{x}_l) = \max_{t_i\le t\le+\infty} t \cdot \mathbf{1}(\sum_{t_i\le t'\le t} \varepsilon r_{ilt'}(\mathbf{x}_l) \le \pi_i)$, where $\mathbf{1}(X)$ equals 1 if X is true; and 0 otherwise. Further, let ψ_i denote the propagation delay on link l for each flow and $\tau_i(\mathbf{x})$ denote the completion time of flow i, *i.e.,* its arrival time plus the total delay of serving flow i. We have

$$\tau_i(\mathbf{x}) = t_i + \max_{l\in\mathscr{P}_i} \delta_{il}(\mathbf{x}_l) + \sum_{l\in\mathscr{P}_i} \psi_l \tag{1}$$

Since our protocol choices do not affect the propagation delay, minimizing (1) is equivalent to minimizing $\max_{l\in\mathscr{P}_i} \delta_{il}(\mathbf{x}_l)$. Let $\mathscr{A}_t$ ($\mathscr{A}_{lt}$) denote the set of flows that have arrived but not yet finished by time t (on link l), *i.e.,* flows i that satisfy $t_i \le t \le t_i + \tau_i(\mathbf{x})$. We refer to these as "alive" flows. Our goal is to minimize the total completion time of n flows, subject to a bandwidth capacity constraint on each link in each time slot. Our online optimization problem is then

$$\text{minimize} \quad \sum_{i\in[n]} \tau_i(\mathbf{x}) \tag{2}$$

$$\text{subject to:} \sum_{i\in\mathscr{A}_{lt}} r_{ilt}(\mathbf{x}_l) \le B_{lt},\ \forall l \in \mathscr{P}_i, i \in [n], t \in [T] \tag{3}$$

$$\sum_{t_i\le t\le\tau_i(\mathbf{x})} r_{ilt}(\mathbf{x}_l) \ge \pi_i,\ \forall l \in \mathscr{P}_i, i \in [n] \tag{4}$$

This problem is difficult to solve due to unknown bandwidth capacities B_{lt} and rate functions r_{ilt}. We aim to first understand the fluctuations of r_{ilt} and then adopt a learning approach to adapt our protocol decisions in real time.

Table 1: Outline of formulation use cases.

Use case	Topology	Example protocols	Selection type
5G	Single link	Control channel size	Per-flow, Per-link
MAC	Single link	CSMA/CD, CSMA/CA	Per-link
Network slicing	Arbitrary	Slice resource reservation, priority class	Per-flow, per-link over domains
Transport layer	Arbitrary	TCP CUBIC, TCP Reno, UDP	Per-flow, per-link w/ proxies

3. ONLINE PROTOCOL SELECTION

Our key insight in solving the optimization problem (2) – (4) is to understand and exploit the relationship among transmission rates, joint protocol decisions of flows, and bandwidth capacities. We formulate a general stochastic model to characterize the rates r_{ilt} in Section 3.1 and then adapt a multi-armed bandit approach to learn the rates and select the protocols in Sections 3.2 and 3.3.

3.1 Stochastic Protocol Interactions

As modeled in (6) below, the transmission rate achieved

[1] We assume for notational simplicity that the protocol choices are the same for each link; our results still hold if they are not.

by each flow is proportional to the *weight* of its corresponding protocol, divided by the total weights of other flows' protocols. The total transmission rate on a link will be its bandwidth capacity B_{lt} scaled by a utilization ratio $u(\mathbf{x}_l, B_{lt})$ ($<= 1$); e_m represents an indicator vector of length M with a one in the m^{th} entry. Hence, $\vec{x}_{ilt} = e_m$ when protocol m is chosen for link l in t. 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 $\mathscr{D}_l^w$ over t, e.g., due to fluctuations in wireless signal strength or routing in an overlay network.

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

$$\text{where: } u(\mathbf{x}_l, B_{lt}) \leq 1, \sum_{m \in [M]} x_{ilmt} = 1, \ \forall l, i, t \tag{6}$$

Based on this model, we choose protocols by learning the distribution of each weight vector. We simplify $u(\mathbf{x}_l, B_{lt})$ to be identical for all $\mathbf{x}_l$ and B_{lt} but allow B_{lt} to be adversarially chosen (*arbitrarily variant over time and across links*) in Section 3.2 and adapt our algorithm for non-identical $u(\mathbf{x}_l, B_{lt})$ and i.i.d. B_{lt} at the end of Section 3.3.

3.2 Selection with Uniform Utilization

If a large flow and several small flows share a link, selecting a protocol for the big flow such that it occupies most of the bandwidth may starve the small flows. Moreover, since minimizing the total flow time (2) is equivalent to minimizing the number of alive flows each time, the offline optimum would always make the flow with the shortest remaining time finish first. Guided by this intuition, 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, i.e., the *shortest alive flow* on the link. As shown in Alg. 1, our algorithm consists of two parts for each time t: 1) **learning**: independently predicting the weight vector on each link based on historical samples; and 2) **protocol selection**: on each link, choosing the protocol with the biggest estimated weight for the *shortest alive flow* and the one with the smallest estimated weight for all the other alive flows. The protocol selection is triggered at each time slot, accounting for all flows in the network at that time. Results of selecting protocols only when a new flow arrives or existing flow completes are given in [7].

Distributed MAB algorithm for predictions. Inspired by classic MAB algorithms that can predict the arm with the highest expected reward, we predict the protocols with the highest and lowest expected weights on each link. If we simply call a protocol decision vector for all flows an "arm," we obtain $M^{|\mathscr{L}| \times |\mathscr{A}_t|}$ arms at each time t, which is too large to effectively sample. However, if we consider each individual protocol decision on each link as an arm and the corresponding weight as the reward, we may not be able to observe an "effective" reward in each time. The reason is that *the rates do not translate into weights for protocols that are present at different times under different capacities*. To address these concerns, we let each link learn the protocol performance independently. In fact, with our greedy strategy of protocol selection, we only need to observe the weight ratio of each *pair* of protocols on each link to predict the best and worst protocols in expectation, reducing the arm space. Let $\eta_{lt}(m, m')$ represent the ratio of the average flow rates achieved by protocols m and m' on link l in time t, i.e.,

$$\eta_{lt}(m, m') = \frac{\text{average flow rate on } l \text{ under } m' \text{ at } t}{\text{average flow rate on } l \text{ under } m \text{ at } t}. \tag{7}$$

Our learning strategy is thus to run a Lower Confidence Bound (LCB) algorithm independently on each link to estimate $\mathbb{E}[\eta_{lt}(m, m')] =: \mathbb{E}\left[\frac{w_{lt}(e_{m'})}{w_{lt}(e_m)}\right]$. Let $x_{lt}(m, m')$ equal 1 if we choose the protocol pair (m, m'); and 0 otherwise. The LCB of $\mathbb{E}[\eta_{lt}(m, m')]$, denoted by $\eta_{lt}^{LCB}(m, m')$, is defined as:

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

$$R_{lt}(m, m') = \sqrt{\frac{\alpha \log t}{\sum_{t'=1}^{t} x_{lt'}(m, m')}} \tag{8}$$

Online protocol selection based on predictions. At time slot t, on each link, once we use $\eta_{lt}^{LCB}(m, m')$ to estimate $\mathbb{E}[\eta_{lt}(m, m')]$, we choose the protocol pair (m, m') that has the smallest value of $\eta_{lt}^{LCB}(m, m')$. Such a protocol pair is denoted as (m^*, m^b) (line 5 of Alg. 1). Our strategy is to assign m^* to the shortest alive flow (indexed by i^*) and m^b to all the other alive flows (lines 6 and 7). Such an assignment can guarantee the shortest flow gets the highest transmission rate if our predictions are accurate, namely $\eta_{lt}^{LCB}(m, m') = \mathbb{E}[\eta_{lt}(m, m')]$, which is proved in [7]. Finally, we apply our protocols to the alive flows, and observe the transmission rates of each flow. We than update the LCB of $\eta_{lt}(m, m')$ according to (8) for use in the next time slot.

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

Input: $G(\mathscr{V}, \mathscr{L})$, n, α
Output: $\mathbf{x}$
Initialize: $\mathbf{x} = \mathbf{0}$, $\eta^{LCB} = \mathbf{1}$, $t = 0$
1 **while** *time slot* $1 \leq t \leq T$ *starts* **do**
2 **for** *each link* $l \in \mathscr{L}$ **do**
3 Update $\tilde{\pi}_{ilt}$ (remaining size of each alive flow);
4 Update $i^* = \text{argmin}_{i \in \mathscr{A}_{lt}} \tilde{\pi}_{ilt}$ (shortest alive flow);
5 Choose $(m^*, m^b) = \text{argmin}_{(m,m')} \eta_{lt}^{LCB}(m, m')$;
 `/* Choose the best protocol pair */`
6 Update $x_{i^* l m^* t} = 1$;
 `/* Choose the dominant protocol in` (m^*, m^b) `for flow` i^* `*/`
7 Update $x_{ilm^b t} = 1, \forall i \neq i^*$;
 `/* Choose the inferior protocol in` (m^*, m^b) `for other flows */`
8 Update $\eta_{lt}(m, m')$ and $\eta_{lt}^{LCB}(m^*, m^b)$ using (8) ;
 `/* Receive feedback */`
9 **end**
10 **end**

THEOREM 3.1. *If all the flows share the same path and the weight vector for each link is i.i.d., the regret of our*

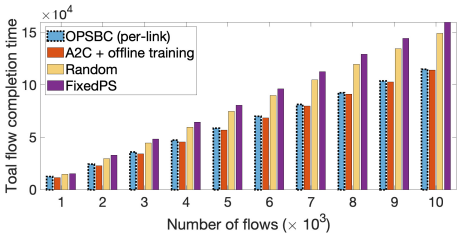

Figure 1: OPSBC achieves similar flow-time to that of A2C with offline training

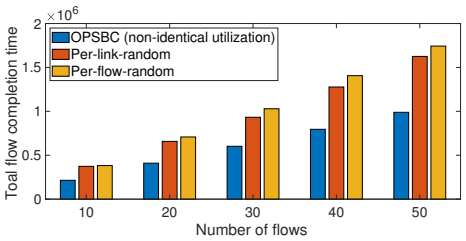

Figure 2: OPSBC for non-identical utilization achieves lower flow-time in experiments

algorithm will be upper-bounded by $O\left(\varepsilon|\mathscr{P}|M^2\log T\right)$, *if we have* $\frac{B_{lt}}{B_{l't}} \leq \frac{\hat{\eta}_l^{max}}{\hat{\eta}_{l'}^{max}} \leq \frac{B_{l't}}{B_{lt}}$, $\forall(l',l): B_{l't} \geq B_{lt}, t \in [T]$.

We prove Theorem 3.1 in our technical report [7].

3.3 Selection with Non-identical Utilization

While it is easy to adapt Algorithm 1 to handle non-identical $u(\mathbf{x}_l, B_{lt})$ [7] for per-link protocol selection with similar performance guarantees, it is much harder for per-flow selection. For instance, choosing the protocol with the highest relative weight over other protocols for the shortest alive flow may lead to a lower bandwidth utilization for all flows on the link and even a lower actual achieved rate for the shortest alive flow, depending on the protocols used by other flows. Due to space limit, the detailed algorithm and performance guarantees are deferred to our technical report [7]. The basic idea is to learn the *dominant throughput* of each pair of protocols, which is the transmission rate achieved by flows using the better protocol in any protocol pair.

4. EXPERIMENTAL VALIDATION

In this section, we implement a line network with four nodes under synthetic data for per-link protocol selection and on Amazon EC2 testbed for per-flow selection. We report the averaged results of 100 repetitions of each experiment. Results for other network topologies can be found in [7].

For per-link protocol selection, we implement three benchmarks: **FixedPS** randomly chooses a protocol on all links for each flow at the first time slot without changing the decisions over time, **Random** randomly selects a protocol for each flow on each link each time, and **A2C + offline training** uses a reinforcement learning (RL) algorithm (A2C) with a pre-trained model. For A2C, we develop an environment to update the network characteristics as a Markov Decision Process, consisting of 20 distinct bandwidth capacities as states, 3 protocol choices as actions, and a distribution of reward representing the total transmission rate that is dependent with protocol and bandwidth capacity. Figure 1 shows that our **OPSBC (modified)** is comparable in achieving the completion time with A2C but does not require an offline training, which can benefit scenarios with stringent constraints on data and time required by the offline training.

We then implement a single-source-destination network with four nodes, each equipped with a TCP proxy, deployed on four Amazon EC2 VM instances with 50 Mbps capacity on each link. We generate 50 flows with sizes uniformly drawn from $[10, 20]$ (Mb) arriving in five batches spaced 3 seconds apart. On each link, **OPSBC** can choose TCP CUBIC, Vegas, Reno, or Westwood. Figure 2 shows the resulting completion times. Despite the overhead at the node proxies, **OPSBC** achieves a 66.18%, 39.27%, and 43.27% decrease in flow-time compared to using TCP CUBIC for all flows, randomly selecting a single protocol for all flows on each link (**Per-link-random**), and randomly selecting a protocol for each flow on each link (**Per-flow-random**), respectively.

5. RELATED WORK

Machine learning for protocol selection. Winstein *et al.* [13] design Remy, a program that can generate distributed congestion control algorithms in a multi-user network and achieve desired outcomes, *e.g.,* high throughput. In light of this offline learning application, we believe that online learning of the protocol performance as in this work can further facilitate protocol selection in dynamic network environments. Mao *et al.* [14] use deep reinforcement learning to allocate cloud resources with a goal of minimizing job slow-downs. In contrast, we provide theoretical guarantees of our model-free online learning algorithms' performance.

Online learning for network management. Chen et al. [15] design novel algorithms for online convex optimization problems with switching costs. Zhang *et al.* [16] integrate the online gradient descent method into online cloud resource provisioning with theoretical guarantees. However, these studies assume full feedback on all feasible solutions, which is unavailable in our model. Thus, we instead take an MAB approach, which only uses information from the chosen decisions, leading to the famous *exploration-and-exploitation* trade-off. MAB algorithms have been extended for many applications, e.g., ad-display optimization [17], dynamic channel access [18], and cloud resource pricing [19]. Their problems' structures are different from ours: the rewards do not directly translate into our completion time objective, but instead provide estimates of unknown inputs needed by an additional algorithm to find the optimal protocols.

6. DISCUSSION AND 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 model coexisting

flows' transmission rates under different protocols and 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 by selecting an optimal pair of protocols to use for all flows present in the network. Our asymptotic optimality is validated through theoretical analysis and experiments.

7. 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] 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.

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

[5] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, "Adaptive congestion control for unpredictable cellular networks," in *In Proc. of ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 509–522.

[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] "Towards automated network management: Learning the optimal protocol selection for network flows," Dropbox, 2019. [Online]. Available: https://www.dropbox.com/s/25ae0nykg2i39pt/TechReport_2019.pdf?dl=0

[8] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, May 2017.

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

[10] Google Cloud, "Overview of TCP forwarding," https://cloud.google.com/iap/docs/tcp-forwarding-overview, 2019.

[11] Y. Cai, F. R. Yu, and S. Bu, "Cloud computing meets mobile wireless communications in next generation cellular networks," *IEEE Network*, vol. 28, no. 6, pp. 54–59, 2014.

[12] G. Siracusano, R. Bifulco, S. Kuenzer, S. Salsano, N. B. Melazzi, and F. Huici, "On the fly tcp acceleration with miniproxy," in *Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization*. ACM, 2016, pp. 44–49.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] R. Combes, C. Jiang, and R. Srikant, "Bandits with budgets: Regret lower bounds and optimal algorithms," in *In Proc. of ACM SIGMETRICS*, 2015.

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

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