Cooperative Bandit Algorithms With Optimal Regret and Communication Costs

Lin Yang, Xuchuang Wang, *Member, IEEE*, Haoxu Chen, *Member, IEEE*, Mohammad H. Hajiesmaili, *Member, IEEE*, Lijun Zhang, *Senior Member, IEEE*, John C. S. Lui, *Fellow, IEEE*, and Don Towsley, *Life Fellow, IEEE*

***Abstract*—In the cooperative multi-armed bandits problem, multiple agents cooperatively play the same multi-armed bandit game. The goal is to develop bandit algorithms with optimal group and individual regrets and low communication among agents. Despite extensive prior research, existing algorithms either cannot achieve time-independent communication costs or fail to achieve optimal individual regrets. We present a simple yet effective communication policy for cooperative bandits where the core algorithmic ideas fundamentally differ from prior work. That is, the proposed communication policy carefully determines the sharing frequency of agents' local observations so that a certain quality of reward estimates is always maintained compared to a full cooperation policy. By deriving a separate lower bound on communication costs of cooperative algorithms, we show that our algorithms achieve the ultimate optimality goal: optimal group and individual regrets and time-independent communication costs.**

***Index Terms*—Distributed learning, multi-Agent multi-armed bandits, regret, communication cost.**

I. Introduction

RECENTLY, there has been a surge of interest in online learning in distributed settings, where a set of agents in a wired/wireless network perform individual learning algorithms to complete a common task and can cooperate with each other to improve the performance of the learning process. Distributed online learning is naturally motivated by a broad range of applications in networks where computational resources are geographically distributed, and machines have to communicate with each other to complete a common task cooperatively. Examples include nodes in a network, servers in a data center, and drones in a swarm (a comprehensive description of those applications is provided in Section VII). In distributed online learning settings, agents take actions over time and receive sequential samples associated with the selected actions. While the agents can cooperate to speed up the learning process, it comes at the expense of communication overhead in sharing sequential samples with others. Hence, distributed online learning problems involve a natural trade-off between learning performance and communication overheads.

This paper studies Cooperative Multi-Agent Multi-Armed Bandit (CMA2B) problems where multiple agents tackle the same instance of a bandit problem. In the standard setting of CMA2B, a set of M independent agents present over a time horizon each pulls an arm at each time from a common set of K arms. Associated with the arms are mutually independent sequences of i.i.d. $[0,1]$-valued rewards with mean $0 \leqslant \mu(k) \leqslant 1$, for arm $k \in \mathcal{K}$. Each agent has access to all arms: agents are allowed to pull and receive a reward from any arm without any reward degradation when pulling the same arm. The goal of each agent is to learn the best arm, with performance characterized by group regret and maximum individual regret according to different application scenarios, where regret is cumulative reward differences between constantly pulling the optimal arms and pulling arms by learning algorithms. In addition to regret, another important metric is the communication overhead that the agents incur in cooperative learning.

The above CMA2B problem is a natural extension of the basic MAB problem [1], [7] in a cooperative multi-agent setting, with extensive recent literature, to name a few [6], [9], [10], [11], [17], [24], [26], [29], [30], [31], [39], [40], [41], [47], [49]. In terms of solution design, the prior work could be categorized into one of two paradigms leader-follower, where a leader agent coordinates the learning process, and distributed, where there is no central coordinator.

In the leader-follower paradigm [3], [9], [14], [32], [35], [36], [37], [41], [42], [44], a leader agent coordinates the learning process among all agents. The state-of-the-art result in this paradigm is the DPE2 algorithm proposed in [41], which achieves the optimal group regret with a constant

Received 15 January 2025; revised 1 September 2025 and 27 January 2026; accepted 28 January 2026; approved by IEEE TRANSACTIONS ON NETWORKING Editor L. Ying. Date of publication 26 February 2026; date of current version 3 March 2026. The work of Lin Yang was supported in part by NSFC under Grant 62306138, in part by Jiangsu NSF under Grant BK20230784, in part by Suzhou's "Jiebang Guashuai" Project for Key Core Technologies under Grant SYG2024134, and in part by the Innovation Program of State Key Laboratory for Novel Software Technology at Nanjing University under Grant ZZKT2024B15 and Grant ZZKT2025B25. The work of John C. S. Lui was supported in part by Research Grants Council (RGC) under Grant GRF-14202925. *(Corresponding author: Xuchuang Wang.)*

Lin Yang and Haoxu Chen are with the School of Intelligence Science and Technology, Nanjing University, Suzhou Campus, Suzhou 215163, China, and also with the National Key Laboratory for Novel Software Technology, Nanjing 210023, China (e-mail: linyang@nju.edu.cn; 522023710001@smail.nju.edu.cn).

Xuchuang Wang, Mohammad H. Hajiesmaili, and Don Towsley are with the Manning College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01003 USA (e-mail: xuchuangw@gmail.com; hajiesmaili@cs.umass.edu; towsley@cs.umass.edu).

Lijun Zhang is with the National Key Laboratory for Novel Software Technology, Nanjing 210023, China, and also with the School of Artificial Intelligence, Nanjing University, Nanjing 210023, China (e-mail: zhanglj@lamda.nju.edu.cn).

John C. S. Lui is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong 999077 (e-mail: cslui@cse.cuhk.edu.hk).

Digital Object Identifier 10.1109/TON.2026.3662342

TABLE I
A COMPARISON SUMMARY OF PRIOR LITERATURE AND THIS WORK

Algorithm	Group regret	Individual regret	Communication cost
DPE2 (leader-follower) [41]	$O(\sum_k \Delta_k^{-1} \log T)$	$O(\sum_k \Delta_k^{-1} \log T)$	$O(K^2 M^2 \Delta_{\min}^{-2})$
GosInE [10]	$O((\sum_k \Delta_k^{-1} + 2M) \log T)$	$O(((\sum_k \Delta_k^{-1}/M) + 2) \log T)$	$O(\log T)$
ComEx [30]	$O(\sum_k \Delta_k^{-1} \log T)$	$O(\sum_k \Delta_k^{-1} \log T)$	$O(KM \log T)$
Dec_UCB [49]	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(MT)$
UCB-TCOM [43]	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(KM \log \log T)$
BatchedMAB [23]	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(KM \log \Delta_{\min}^{-1})$
DoE-bandit	$O(\sum_k \Delta_k^{-1} \log T)$	$O((\sum_k \Delta_k^{-1}/M) \log T)$	$O(M \sum_k \log \Delta_k^{-1})$
Communication Lower Bound	–	–	$\Omega(\max\{\sum_k \log \Delta_k^{-1}, M\})$

number of communication overheads,[1] Yet, DPE2 (and all other leader-follower-based algorithms) relies on a structure where the leader solely pays the exploration costs and incurs almost all the regret in the system. Hence, by nature, this paradigm fails to achieve good individual regret since all regret is imposed on the leader agent. It is worth noting that in many practical applications, the regret of each agent is crucial for a system's overall performance. For example, in a drone swarm, the failure/misbehavior of a single drone, e.g., it crashes into other drones, can dramatically degrade the overall system performance; or in network measurement, the slowest inference engine determines how fast network parameters, e.g., traffic flows and channel bandwidths, are learned.

An alternative approach is to remove the leader as the central coordinator and design distributed cooperative algorithms without relying on a leader-follower structure. Although there has been success in achieving optimal group and individual regrets for distributed online learning algorithms, they still fail to achieve low communication overheads, such as found in leader-follower-based algorithms. Early works in this space, for example, [8], [46], [47] adopted immediate broadcasting as their communication scheme, incurring a high communication cost of $O(T)$. More recent work [10], [31], [44], improved the communication overhead of cooperative algorithms to $O(\log T)$ by optimizing the use of the communication budget. The state of the art in this line of work is the UCB-TCOM algorithm [43] that achieves the optimal individual regret of $O(K/M \log T)$ with a communication cost of $O(KM \log \log T)$. Despite the above efforts, prior to this work, no existing algorithms, based either on the leader-follower scheme or not, achieve optimal group and individual regret with constant communication costs.

In addition to the literature on distributed bandits, works on batched bandits [16], [18], [22], [23], [33], [48] also relate to CMA2B. In batched bandits, the time horizon is separated into several batches of time slots, and the reward observations of pulling arms during each batch are revealed at the end of the batch. This scheme is similar to distributed bandits, where the observations of other agents after the last communication are only revealed at these agents' next communication. Therefore, the batched bandits algorithm can adapt to our multi-agent bandits setting. The current state-of-the-art batched algorithm, BatchedMAB [23], requires $O(K \log \Delta_{\min}^{-1})$ batches to achieve the near-optimal problem-dependent regret bound. That is, transferring their algorithms to the distributed setting leads to $O(KM \log \Delta_{\min}^{-1})$ communication costs. In contrast, our work shows that a lower constant communication cost $O\left(M \sum_{k:\Delta_k>0} \log \Delta_k^{-1}\right)$ is enough to guarantee optimal individual and group regrets, and we prove a lower communication bound showing that this communication cost is tight in terms of all factors.

Contributions. This paper presents DoE-bandit, the first distributed algorithm that achieves the optimal group and maximum individual regrets with optimal communication costs (Theorem 2). Specifically, DoE-bandit achieves an $O\left(\sum_k \Delta_k^{-1} \log T\right)$ group regret and an $O\left(\sum_k (\Delta_k^{-1}/M) \log T\right)$ maximum individual regret, where Δ_k is the gap between the reward of the optimal arm k^* and the arm k. In addition, DoE-bandit achieves a constant communication cost of $O\left(M \sum_k \log \Delta_k^{-1}\right)$. To show the optimality of the communication cost, we propose a novel gap-dependent communication lower bound $\Omega\left(\max\left\{\sum_k \log \Delta_k^{-1}, M\right\}\right)$ for any CMA2B algorithm that achieves near-optimal group and individual regrets (Theorem 1). This lower bound shows that the communication cost of DoE-bandit is tight in terms of all factors. A summary of our results and the most relevant prior works is given in Table I.

To achieve the above results, DoE-bandit leverages a communication policy called Distributed Online Estimation (DoE), which is a novel algorithmic contribution of this work. The key idea behind DoE is to determine the synchronization frequency of sharing local empirical estimates on arms with other agents so that the quality of estimates is maintained compared to full cooperation policy. Full cooperation is equivalent to having a *centralized estimator* with access to all samples from all agents. With limited cooperation, individual agents have access to their locally observed samples, which can cause intrinsic deviations from the centralized estimator. DoE measures the deviations precisely and uses them as an indicator to trigger a communication round. That is, DoE may urge agents to communicate with others to synchronize the estimates on the mean of arms once they realize that the deviation is large. By controlling the deviation in a proper margin, DoE guarantees the optimal learning performance,

[1]Constant communication cost in this paper means it is independent of time horizon T.

i.e., the one achievable by the centralized estimator, for individual agents with low communication cost. By plugging DoE into an elimination-based bandit algorithm, we derive DoE-bandit that improves the state-of-the-art results for the CMA2B problem. Last, we report experiments demonstrating the improved performance of DoE-bandit compared to all benchmark algorithms listed in Table I.

The paper is organized as follows: Section II provides a detailed formulation of the CMA2B problems. Section III introduces the algorithm DoE and integrates it into a bandit algorithm. Section IV presents a novel communication lower bound and upper bounds for communication costs, group regret, and individual regret, all of which are near-optimal. Section VI presents simulations that validate the efficacy of DoE in terms of communication costs, group regret, and individual regret. Section V offers a detailed proof of the regret analysis. Section VII discusses potential applications of CMA2B. Section VIII concludes the paper.

II. Problem Description

Consider a multi-agent stochastic bandit setting with a set $\mathcal{M} := \{1, \ldots, M\}$ of independent agents existing over the entire time period from 1 to T, and a set $\mathcal{K} := \{1, 2, \ldots, K\}$ of arms. Associated with each arm are mutually independent sequences of i.i.d $[0, 1]$-valued (e.g., Bernoulli) rewards with unknown means $0 \leqslant \mu(k) \leqslant 1$ for arms $k \in \mathcal{K}$. Agent $m \in \mathcal{M}$ has full access to the set of arms. Agents are allowed to pull and receive a reward from any arm k from $\mathcal{K}$. For ease of presentation, we focus on a basic model formulation where agents reside on a complete graph, incur no communication delays, and communication is lossless. The basic model and communication policy proposed in this paper can be extended to account for these practical additions.

In bandit learning, the goal of each agent m is to learn the best arm as fast as possible and minimize the (pseudo) regret in $T \in \mathbb{N}^+$ decision rounds. The expected regret of an agent m is formally defined as $\mathbb{E}[R_T^{(m)}] := \mu(k^*)T - \mathbb{E}\left[\sum_{t=1}^{T} x_t(I_t^{(m)})\right]$, where k^* is the optimal arm, $I_t^{(m)}$ is the action taken by agent m at round t, and $x_t(I_t^{(m)})$ is the realized reward. Also, the expectation is taken over the randomness of stochastic rewards and the algorithms. In a multi-agent setting, the total performance is measured by the total expected regret of all agents, defined as

$$\mathbb{E}\left[R_T\right] := \sum_{m \in \mathcal{M}} \mathbb{E}\left[R_T^{(m)}\right].$$

In addition to the group regret, which characterizes overall performance, the individual performance of each agent is also important. To capture this individual performance, we measure the maximum individual regret defined as follows,

$$\mathbb{E}\left[\bar{R}_T\right] := \mathbb{E}\left[\max_{m \in \mathcal{M}} R_T^{(m)}\right].$$

Similar to other distributed learning problems, CMA2B encourages distributed agents to cooperate with each other by sharing information through *messages*, which include reward observations, reward averages, or arm indices. We assume any message is communicated within a single time slot. The total number of messages communicated among agents quantifies the communication cost. We denote the expected total communication cost in T rounds among M agents as follows,

$$\mathbb{E}[C_T] = \sum_{t=1}^{T} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} \mathbb{E}[c_t^{(m)}(k)],$$

where $c_t^{(m)}(k) := \mathbb{1}\{\text{agent } m \text{ communicates about arm } k \text{ at time slot } t\}$. The communication cost definition assumes that each message only contains the information of one arm, and if the agents want to share information of regarding multiple arms, they need multiple messages, one for each arm. We choose this dentition in order to show the tightness of our communication cost analysis at the arm level, and our algorithm design and theoretical analysis can be adapted to the case that one message aggregates information about multiple arms [11], [43], [46]. On the other hand, the total number of bits of communication is also of interest in the literature [37], [41], [44]. One can convert the communication cost in our definition to the total number of bits of communication. We sketch one approach for converting communication cost to number of bits, which only introduces time-independent prefactors to our communication cost bounds.

III. Algorithm

This section presents an algorithm that adds a Distributed Online Estimation (DoE) subroutine to each learning agent m and enables them to approximate the estimate of the optimal centralized algorithm having all samples when estimating the parameter of a common i.i.d. process. We introduce the DoE algorithms in Section III-A and then integrate it to a bandit algorithm in Section III-B. The communication policy DoE is generic and could be applied to a broad range of cooperative online learning settings.

A. Distributed Online Estimation Algorithm (DoE)

To facilitate the presentation of the high-level idea of DoE, let us focus on a simplified setting that involves only one arm k whose reward mean $\mu(k)$ is unknown to the distributed agents, where agents sample the process simultaneously in each slot. Since each agent possesses the same number of pulls, we denote $n_t(k)$ as the number of samples available to each agent up to time t. The idea of DoE is to synchronize the estimates of distributed agents when the local estimates deviate substantially from the centralized one with all samples. By properly configuring DoE, each individual agent needs to efficiently control the deviation of its local estimates while incurring low communication costs.

More specifically, during the running time, DoE adopts a threshold policy to decide whether to trigger a communication round for agents to synchronize their estimates with all samples in the system. To decide whether to start a communication round, each agent maintains the so-called *Common Mean* (CM) for the mean over all system-wide available samples in the last communication round, and simply compare CM with *Auxiliary Local Estimates* (ALE, details shown in (1)). The

value of CM, denoted as $\hat{\mu}_{\mathtt{com},t}(k)$, is calculated by averaging all samples up to the last communication round, so, its value is updated only once at each communication round and remains unchanged in the subsequent non-communication rounds. At specific time slots, each agent checks whether the gap between CM and ALE has exceeded a certain threshold value.

In DoE, all agents share a common threshold value denoted as $\mathtt{ECR}_t(k)$, which can be time-varying with the number of available samples $n_t(k)$. If the gap between ALE and CM is larger than the threshold value, a new communication round is triggered to synchronize the estimates. By doing so, the sum of new samples from other agents will be collected, a new common mean is calculated, and then the agent broadcasts the new CM to all others.

The threshold value $\mathtt{ECR}_t(k)$ plays a key role in controlling estimate deviations and communication overheads. Intuitively, when the ALEs of each individual agent center around the common mean, the actual estimates of all agents center around CM as well. Thus, no communication is needed. Otherwise, a communication round is triggered to synchronize the estimates of all agents. Hence, the threshold value determines how far the estimates deviate from each other during the non-communication rounds; the smaller the threshold value, the smaller the deviations, and the closer the local estimates of agents approach the global mean over all samples. On the other hand, with smaller threshold values $\mathtt{ECR}_t(k)$, agents communicate more frequently with each other. Hence, the trade-off of estimation performance versus communication overheads is associated with $\mathtt{ECR}_t(k)$.

Next, we present the technical details of the DoE algorithm and show how to construct the estimate interval for each agent by using local estimates.

Constructing the Auxiliary Local Estimates (ALE). At a non-communication round t, an agent only accesses partial external samples from others. Below we introduce how an agent builds up the Auxiliary Local Estimate with missing samples from others. Note that $n_t(k)$ is the number of samples that an agent has made for arm k under the exploration phase up to time slot t. Let $t_{\mathtt{last}}$ denote the last round before t that the Condition in Line 6 holds, and $X_t^{(m)}(k)$ be the sum of rewards from $n_t(k)$ samples of agent m at time slot t for arm k. For agent m, there are $n_t(k) - n_{t_{\mathtt{last}}}(k)$ missing samples from any other agents. In DoE, agent m uses local samples in the same time slot to compensate the missing samples from other agents to construct ALE, denoted by $\hat{\mu}_{\mathtt{ALE},t}^{(m)}(k)$. That is

$$\hat{\mu}_{\mathtt{ALE},t}^{(m)}(k) = \frac{\sum_{m'=1}^{M}(X_{t_{\mathtt{last}}}^{(m')}(k) + X_t^{(m)}(k) - X_{t_{\mathtt{last}}}^{(m)}(k))}{Mn_t(k)} \quad (1)$$

where the term $X_t^{(m)}(k) - X_{t_{\mathtt{last}}}^{(m)}(k)$ serves as the compensation for the missing samples from other agents $m' \neq m$ from $t_{\mathtt{last}}$ to t. In DoE, ALE mimics the estimate of the estimator, which possesses all $Mn_t(k)$ samples and serves as an index through which the agents decide when to communicate. The ALE estimates may involve a larger estimation error. Hence, in addition to ALE, each agent m calculates the local estimate $\hat{\mu}_t^{(m)}(k)$ to be used in a bandit algorithm using the following equation.

$$\hat{\mu}_t^{(m)}(k) = \frac{\sum_{m'=1}^{M} X_{t_{\mathtt{last}}}^{(m')}(k) + (X_t^{(m)}(k) - X_{t_{\mathtt{last}}}^{(m)}(k))}{Mn_{t_{\mathtt{last}}}(k) + n_t(k) - n_{t_{\mathtt{last}}}(k)} \quad (2)$$

Algorithm 1 DoE: An Algorithm for Estimating the Mean of Arm k by Agent m, Subscript t Is Dropped

1: **Parameters:** $\beta > 1$;
2: **Initialization:** $\hat{\mu}_{\mathtt{ALE}}^{(m)}(k)$, $n(k) \leftarrow 0$ $\hat{\mu}_{\mathtt{com}}(k) \leftarrow 0$, $\mathtt{ECR}_{\mathtt{last}} \leftarrow 0$; $X^{(m')}(k) \leftarrow 0$, $X_{\mathtt{last}}^{(m')}(k) \leftarrow \infty$, $\forall m' \in \mathcal{M}, \mathtt{ECR}_t(k) \leftarrow 0$
3: **for** each round t when the agent gets a new sample **do**
4: $n(k) \leftarrow n(k) + 1$
5: Update $X^{(m)}(k)$ with the new sample
6: **if** $\beta\mathtt{ECR}(k) \leqslant \mathtt{ECR}_{\mathtt{last}}$ **then**
7: $\mathtt{ECR}_{\mathtt{last}} \leftarrow \mathtt{ECR}(k)$
8: **if** $|\hat{\mu}_{\mathtt{ALE}}^{(m)}(k) - \hat{\mu}_{\mathtt{com}}(k)| > \mathtt{ECR}(k)$ **then**
9: //Communicate to synchronize the estimates
10: Collect $X^{(m')}(k)$ from other agents and calculate the new $\hat{\mu}_{\mathtt{com}}(k)$
11: Broadcast the new $\hat{\mu}_{\mathtt{com}}(k)$ to other agents to update their estimates on arm k
12: $X_{\mathtt{last}}^{(m')}(k) \leftarrow X^{(m')}(k)$ for all $m' \in \mathcal{M}$
13: Update $\hat{\mu}_{\mathtt{ALE}}^{(m)}(k)$ according to Eq. (1) and $\hat{\mu}^{(m)}(k)$ according to Eq. (2)

Communication Policy of DoE. Now with the definition of ALE, we present the communication policy of DoE. The pseudocode of DoE is summarized in Algorithm 1. To decide a communication round, an agent m checks the values of $\hat{\mu}_{\mathtt{ALE},t}^{(m)}(k)$ and $\hat{\mu}_{\mathtt{com},t}(k) := \left(\sum_{m=1}^{M} X_{t_{\mathtt{last}}}^{(m)}(k)\right)/(Mn_{t_{\mathtt{last}}}(k))$ every time the specified threshold value $\mathtt{ECR}(k)$ reduces to $1/\beta(\beta > 1)$ times of the original value $\mathtt{ECR}_{\mathtt{last}}$ (Lines 6, 7). In DoE, β determines how frequently the algorithm checks those values. Once the deviation of the local estimate $\hat{\mu}_{\mathtt{ALE},t}^{(m)}(k)$ from the common mean $\hat{\mu}_{\mathtt{com},t}(k)$ is larger than $\mathtt{ECR}(k)$ (Line 8), agent m calls for triggering of a new communication round. In a communication round triggered by agent m, the sum of missing samples from the last communication round $t_{\mathtt{last}}$ from each other agent will be collected to calculate a new common mean. Then, this new common mean will be broadcast to all other agents.

Our analysis in Lemma 1 provided in Section V shows that DoE provides a provable performance guarantee for the single-arm-estimation problem (in the form of a confidence interval) with a tunable trade-off between the estimation quality and communication overheads. With a richer communication budget, the estimation performance of DoE approaches that of the optimal estimator with full access to the samples. Since DoE can provide an explicit confidence interval for the mean to be estimated, it is straightforward to plug DoE into bandit algorithms, as exemplified in the next section.

B. Integrating DoE to a Bandit Learning Algorithm

In this section, we present a distributed bandit algorithm named DoE-bandit that uses DoE as the underlying

Algorithm 2 DoE-bandit for Agent m; Subscript t Is Dropped

1: **Parameters:** $\alpha > 0$, $\beta > 1$, $2 \leqslant \Gamma \in \mathbb{N}^+$; $\texttt{ECR}_n$, $n = 1, 2, \ldots$
2: **Initialization:** $\hat{\mu}^{(m)}_{\texttt{com}}(k) \leftarrow 0$; $\texttt{ECR}_{\texttt{last}}(k)$; $n(k) \leftarrow 0$, $\hat{\mu}^{(m)}_{\texttt{ALE}}(k)$, $\forall i$; $\texttt{ECR}_n \leftarrow \alpha \texttt{CR}(Mn, \delta_t)$, $\forall n \in \mathbb{N}^+$
3: Pull each arm in the candidate arm set for one time, $n(k) \leftarrow 1$
4: **for** each round t **do**
5: $k^{\max} \leftarrow \arg\max_{k \in \mathcal{C}} \hat{\mu}_{t_{\texttt{last}}}(k)$
6: **if** an arm is eliminated by some other agent **then**
7: Update the candidate set by eliminating arm k
8: **if** $t \mod \Gamma = 0$ **then**
9: Choose arm from $\mathcal{C}$ with a round-robin manner
10: **else**
11: Pull arm $k^{\max}$
12: Execute Lines 4- 14 of DoE (Algorithm 1) for communicating estimates on arm k and Execute Line 15 for updating the estimates on arm k
13: Update the candidate set via Eq. (3)
14: Notify other agents if an arm is eliminated

communication policy. We summarize the pseudocode of DoE-bandit in Algorithm 2.

DoE-bandit is based on active arm elimination, which is a classic approach to address the well-known tradeoff between exploration (acquiring new information) and exploitation (optimizing based on available information) in bandit problems. In this approach, the learner constructs a *candidate set* for the arms, which are likely to be optimal, and exploration is allowed only from the arms in the candidate set. When exploring the candidate set, the algorithm periodically pulls an arm and dynamically eliminates the arms which are unlikely to be optimal. To improve empirical performance, DoE-bandit adopts a modified exploration strategy for the arms in the candidate set. Specifically, the algorithm introduces a hyperparameter, denoted as Γ, which controls the proportion of time slots dedicated to exploring the arm with the empirically best performance (denoted as $k^{\max}$). This approach ensures that the algorithm minimizes the number of times sub-optimal arms are pulled, particularly during the initial stages of the process.

To integrate DoE with the bandit algorithm, we initiate multiple instances of DoE run by DoE-bandit, each of which tackles the estimation of a single arm. To implement the DoE subroutine, each agent notifies others once an arm is eliminated (Line 16 in Algorithm 2), and pull arm $k^{\max}$ or arms in $\mathcal{C}_t$ by justifying whether $t \mod \Gamma$ is equal to zero. (Line 10), and DoE is able to keep track of the total number of samples in the system by $Mn_t(k)$, where $n_t(k)$ is the sample count in round-robin sampling. The above rules imply that all agents have a common candidate set, which is denoted by $\mathcal{C}_t$.

Constructing the candidate set. To construct the candidate set, DoE-bandit determines an explicit confidence interval for the arm reward means. Denote $\texttt{CR}(n, \delta_t)$ as the radius of the confidence interval for the $[0,1]$-valued reward process with n samples and confidence level $1 - \delta_t$, defined as, $\texttt{CR}(n, \delta_t) := \sqrt{\log \delta_t^{-1}/2n}$, where δ_t specifies the violation probability that the true mean lies outside the confidence interval with radius CR. As we mentioned, the threshold value, $\texttt{ECR}_t(k)$, in DoE determines the deviation of the estimates in individual agents from the optimal one with all samples. Hence, to ensure distributed agents achieving the same order of the convergence rate as the optimal one, we set $\texttt{ECR}_t(k)$ according to the confidence interval with the total of $Mn_t(k)$ samples; therefore, called estimated confidence radius (ECR). By setting $\texttt{ECR}_t(k) = \alpha \texttt{CR}(Mn_t(k), \delta_t)$ where $\alpha > 0$, DoE yields a confidence interval for the mean of arm k, whose radius is $(2\alpha\beta + \beta)\texttt{CR}(Mn_t(k), \delta_t)$ (Lemma 1). With the above result, an arm k is eliminated by agent m from the candidate set $\mathcal{C}_t$ at time t if there exist an arm $k' \in \mathcal{C}_t$ such that $\hat{\mu}^{(m)}_t(k) + (2\alpha + \beta)\texttt{CR}(Mn_t(k), \delta_t) < \hat{\mu}^{(m)}_t(k') - (2\alpha + \beta)\texttt{CR}(Mn_t(k'), \delta_t)$. That is, the candidate arm set $\mathcal{C}$ is updated as follows,

$$\mathcal{C} \leftarrow \big\{k \in \mathcal{C} : \max_{k' \in \mathcal{C}} \hat{\mu}^{(m)}_t(k') < \hat{\mu}^{(m)}_t(k) + 2(2\alpha + \beta)\texttt{CR}(Mn_t(k), \delta_t)\big\}. \quad (3)$$

IV. Regret and Communication Cost of DoE-Bandit

In this section, we summarize the theoretical results. We start with a novel communication lower bound for CMA2B in Section IV-A. Then, we analyze the DoE-bandit algorithm, with both regret and communication cost provided. With those results, we show that DoE-bandit attains near-optimal guarantees in group and individual regrets as well as communication costs.

A. Communication and Regret Bounds

In this section, we present a communication lower bound for the CMA2B model. Our focus is on investigating the necessary number of communications (lower bound) for any CMA2B algorithm attaining near-optimal group and individual regrets. The result is provided in Theorem 1 as follows:

Theorem 1: (Communication Lower Bound) For any algorithm that achieves the near-optimal group regret $O\left(\sum_k \Delta_k^{-1} \log T\right)$ and individual regret $O\left(\left(\sum_k \Delta_k^{-1}/M\right) \log T\right)$ and for any CMA2B instance, the algorithm spends a number of communications that is lower bounded as follows,

$$\mathbb{E}[C_T] \geqslant \Omega\left(\max\left\{\sum\nolimits_{k:\Delta_k>0} \log \Delta_k^{-1}, M\right\}\right).$$

Theorem 2 summarizes the results for DoE-bandit.

Theorem 2: Set $\texttt{ECR}_t(k) = \alpha \min\{1, \texttt{CR}(Mn_t(k), \delta_t)\}$ with $\alpha > 0, \beta > 1, 0 < \Gamma < 1$, then DoE-bandit achieves:

1) (Group Regret)

$$\mathbb{E}[R_T] \leqslant \sum_{k:\Delta_k>0} \frac{\left(2 + \frac{6}{\Gamma}\right)(2\alpha\beta + \beta)^2 \log \delta_T^{-1}}{\Delta_k} + \sum_{k:\Delta_k>0} \left(3 - \frac{2}{\Gamma}\right) M\Delta_k + 2KM^3 \sum_{t \leqslant s} t\delta_t^{1/2}, \quad (4)$$

2) (Individual Regret)

$$\mathbb{E}\left[\bar{R}_T\right] \leqslant \sum_{k:\Delta_k>0} \frac{\left(2+\frac{6}{\Gamma}\right)(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} + \sum_{k:\Delta_k>0} \left(3-\frac{2}{\Gamma}\right)\Delta_k + 2KM^2 \sum_{t\leqslant s} t\delta_t^{1/2} \tag{5}$$

3) (Communication Cost)

$$\mathbb{E}[C_T] \leqslant \sum_{k:\Delta_k>0} 6M \log_\beta \left(\frac{4(2\alpha+1)}{\alpha\Delta_k}\right) + 2KM^3T \sum_{t\leqslant T} t\delta_t^{1/2}. \tag{6}$$

Corollary 1: With the same parameters as Theorem 2 and setting $\delta_t \leftarrow 1/(K^2M^6T^2t^6)$,[2] `DoE-bandit` achieves the following performance:

(Group Regret)

$$\mathbb{E}\left[R_T\right] \leqslant O\left(\sum\nolimits_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log T}{\Delta_k}\right), \tag{7}$$

(Maximum Individual Regret)

$$\mathbb{E}\left[\bar{R}_T\right] \leqslant O\left(\sum\nolimits_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log T}{M\Delta_k}\right), \tag{8}$$

(Communication Cost)

$$\mathbb{E}[C_T] \leqslant O\left(\sum\nolimits_{k:\Delta_k>0} M \log_\beta \left(\frac{2\alpha+1}{\alpha\Delta_k}\right)\right). \tag{9}$$

B. Discussions

1) Optimality in All Three Metrics: Corollary 1's (7) and (8) show that we can recover a $O\left(\sum_{k:\Delta_k>0}(1/\Delta_k)\log T\right)$ group regret and $O\left(\sum_{k:\Delta_k>0}(1/\Delta_k)\log T/M\right)$ individual regret for the distributed bandit problem, implying that the proposed algorithm attains both the (order-) optimal group and maximum individual regrets [43]. In the meantime, compared with the communication lower bound $\Omega\left(\max\left\{\sum_{k:\Delta_k>0}\log\Delta_k^{-1}, M\right\}\right)$ in Theorem 1, the communication upper bound of `DoE-bandit` in (9), i.e., $O\left(M\sum_{k:\Delta_k>0}\log\Delta_k^{-1}\right)$, is optimal in terms of both the agent number M and the summation $\sum_{k:\Delta_k>0}\log\Delta_k^{-1}$.

2) Influence of α and β: Corollary 1's (9) shows that communication overheads influence the estimation quality through parameters α and β. Generally speaking, β specifies the frequency that `DoE` checks the deviation of individual estimates, directly upper bounding the communication overheads for `DoE-bandit`. Hence, β seems to have a larger influence in the communication overheads bound than α. On the other hand, α specifies the radius of the estimate interval `CR` as well as the threshold for the estimate deviation `ECR`, which triggers an actual communication demand.

[2]One limitation of our result is that the selection of δ_t for achieving these near-optimal results requires the knowledge of time horizon T. This is due to the elimination policy used in our multi-agent algorithm design.

3) Extension to Other i.I.D. Processes: In `DoE-bandit`, the communication overhead on a suboptimal arm k is approximately $O(\log_\beta(\texttt{ECR}_1/\texttt{ECR}_T))$. The threshold value $\texttt{ECR}_t$ is set based on that of the confidence interval with all samples (up to a tunable parameter α). For a Bernoulli process, the mean always lies in $[0,1]$. Hence, we can set $\texttt{ECR}_1 = 1$, which results in $O(\log_\beta(1/\texttt{ECR}_T))$ communication overheads. By slight modification, the `DoE-bandit` algorithm can tackle other i.i.d. processes with similar results obtained. For an i.i.d. process with an unbounded mean, such as the Gaussian process, the `DoE-bandit` may choose to start a communication round only when the size of the confidence interval shrinks to $O(\sqrt{M})$. This will not degrade the regret results guaranteed in Theorem 2, since the algorithm only has to spend on average $O(\log T)$ samples in shrinking the confidence intervals of all arms, with an increase of $O(K\log T)$ regret. On the other hand, the communication overhead is only $O(\log(\sqrt{M}/\delta_t))$, since $\texttt{ECR}_1$ can be set to $O(\sqrt{M})$.

4) Influence of Delays: With minor modifications to `DoE` (detailed in Section A), the `DoE-bandit` algorithm can effectively handle deterministic communication delays in a networked setting where agents communicate solely with their immediate neighbors via message-passing. Despite these changes, the algorithm maintains near-optimal group and individual regret bounds, with only a constant additional cost that grows linearly with the communication delays (see the corresponding theorem in Section A).

5) Bit-Wise Communication Upper Bound: In this subsection, we present a concise discussion showing that the total communication bits of `DoE-bandit` amount to $O\left(M^2\sum_k\log\Delta_k^{-1}\right)$, increasing a multiplicative M factor compared to the communication cost upper bound in Theorem 2, yet still surpassing `DPE2`'s communication cost of $O(K^2M^2\Delta_{\min}^{-2})$.

Upon closer examination of `DoE`'s communication bits concerning a fixed arm k, we identify that the primary cost stems from an agent m initiating communication, gathering other agents' observed reward sums $X^{(m)}(k)$, and subsequently disseminating the updated mean $\hat{\mu}_{\text{com}}(k)$ (Lines 10 and 11 in Algorithm 1). We propose a refined communication protocol inspired by [37, §3.3], along with a proof sketch demonstrating that the total communication bits under this protocol remain $O(M)$. With an upper bound of $O(M)$ bits for a single communication, we consequently establish that the overall communication bits of `DoE` amount to $O\left(M^2\sum_k\log\Delta_k^{-1}\right)$.

The communication procedure is the same as `DoE`, expect for that Lines 10 and 11 of `DoE` are replaced as follows,

10: Collect the difference of quantized empirical mean since the last communication $\tilde{\delta}_t^{(m')}(k) = \tilde{\mu}_t^{(m')}(k) - \tilde{\mu}_\tau^{(m')}(k)$, where $\tilde{\mu}_t^{(m')}(k)$ is a quantized version of $\hat{\mu}_t^{(m')}(k)$ using $O(\lceil\log n_t^{(m')}(k)\rceil)$ bits and τ is the last communication time slot for arm k, from all other agents $m' \neq m$.

11: Agent m broadcasts all agents' $\tilde{\delta}_t^{(m')}(k)$ to all other agents, and with this new information, all agents update the approximated global empirical mean $\tilde{\mu}_{\text{com}}(k)$, $\hat{\mu}_{\text{ALE}}^{(m)}(k)$, and $\hat{\mu}_t^{(m)}(k)$.

First, we illustrate that communicating $\tilde{\delta}_t^{(m)}(k)$ requires only a constant number of bits in expectation. This is due to the proximity between empirical means $\hat{\mu}_t^{(m)}(k)$ and $\hat{\mu}_\tau^{(m)}(k)$ (as well as the difference between their quantized versions), which results in a small $\tilde{\delta}_t^{(m)}(k)$ in expectation. Utilizing a derivation akin to [37, Appendix E.1], one can ascertain that the communication overhead for $\tilde{\delta}_t^{(m)}(k)$ is $O(1)$ in expectation. Given that Lines 10 and 11 involve all M agents, this communication process merely necessitates $O(M)$ bits in expectation.

On the other hand, one also needs to establish that the quantization error from $\hat{\mu}$ to $\tilde{\mu}$ does not fundamentally impact the final regret bound. This is because the quantization is adaptive to the sample size, resulting in an error smaller than the current confidence radius, thereby introducing only a constant prefactor to the ultimate regret bound.

V. Detailed Proofs

We provide detailed proofs of the theoretical results in this section, including both upper and lower bounds on regrets and communication costs.

A. Proof for Communication Lower Bound (Theorem 1)

We separately prove the two terms in the maximum operation at the lower bound. While the $\Omega(M)$ communication lower bound is derived by extending the results of [44, Theorem 2], the $\Omega\left(\sum_{k:\Delta_k>0}\log\Delta_k^{-1}\right)$ lower bound's proof is novel. We denote $n_t^{(m)}(k)$ as the *accessible observations* for arm k of agent m at time slot t. It includes agent m's own local observations as well as these received from other agents via communication on or before time slot t. The proof is based on three claims: Claim (1) guarantees that the first communication about arm k happens on or before the accessible observations $n_t^{(m)}(k)$ reaching $G\log T$ for a constant G for all agents; Claim (2) shows that, after a previous communication about arm k, the agent must make the next communication about arm k on or before the number of the accessible observations $n_t^{(m)}(k)$ is doubled; Claim (3) (i.e., formula (10) in the full proof) states that, as the algorithm achieves near-optimal regrets, the accessible observations for arm k should reach $\Omega(\Delta_k^{-2}\log T)$ by the end of the CMA2B game. Putting these three claims together, we have $2^{\mathbb{E}[C_T(k)]}G\log T \geqslant \Omega\left(\Delta_k^{-2}\log T\right)$, where $\mathbb{E}[C_T(k)]$ is the expected total number of communications about arm k. Rearranging the inequality yields $\mathbb{E}[C_T(k)] \geqslant \Omega(\log\Delta_k^{-1})$. Summing the communication costs over all suboptimal arms yields the lower bound.

Proof: [Proof of Theorem 1]

We first prove the $\Omega\left(\sum_{k:\Delta_k>0}\log\Delta_k^{-1}\right)$ *communication lower bound.* Let us fix a suboptimal arm k. To facilitate the proof presentation, we denote $n_t^{(m)}(k)$ as the *accessible* number of observations for arm k of agent m at time t, where the "accessible" observations includes the agent's own local observations as well as other agents' observations received through communications (on or before time slot t). We first prove two key claims by contradiction as follows,

Claim 1: The initial communication concerning arm k must occur on or before the time slot when the accessible number of observations of arm k of all agents reach $G\log T$ for a universal constant G.

Claim 2: If the algorithm communicates regarding arm k at a specific time slot t, it must have a further communication on arm k either on or before the time slot when the expected accessible number of observations of arm k of all agents reach twice the count recorded at time slot t.

Proof: [Proof of Claim 1] If at the initial communication, all agents' accessible observations of arm k are all $\Omega(G\log T)$ and noticing that all of these observations are local, the total number of pulls on arm k at time t would be $\Omega(MG\log T)$, where the M factor contradicts the near-optimal group regret (without M) that the algorithm achieves. □

Proof: [Proof of Claim 2] In the communication time slot t, we assume that agents have a "strongest" communication (as we are proving a lower bound), meaning that every agent broadcasts and receives all observations for arm k up to time slot t, and, therefore, all of their number of accessible observations $n_t^{(m)}(k)$ are equal. So, we have $n_t^{(m)}(k) = n_t(k)$ for all agents $m \in \mathcal{M}$. We note that since the algorithm achieves near-optimal regrets, we have that $\mathbb{E}[n_t(k)] = \Omega\left(\frac{\log t}{\Delta_k^2}\right)$. After time slot t, if there is no communication till time slot t' such that $n_{t'}^{(m)}(k) \geqslant 2n_t(k)$ for all agent m, then all agents more respectively pull arm k for $n_{t'}^{(m)}(k) - n_t^{(m)}(k) \geqslant n_t(k)$. Then, the total number of pulls of arm k between time slots t and t' is at least $\Omega(Mn_t(k)) = \Omega\left(M\frac{\log t}{\Delta_k^2}\right)$. This M factor on the number of pulling times of arm k contradicts the near-optimal group regret that the algorithm achieves. Therefore, there must exist a communication on arm k between time slots t and t'. □

With Claims 1 and 2 and assuming that the algorithm makes $\mathbb{E}[C_T(k)]$ number of communications on arm k, then the total number of pulls on arm k is at most $2^{\mathbb{E}[C_T(k)]}G\log T$. Since the algorithm also achieves the near-optimal regret upper bound, we have

$$2^{\mathbb{E}[C_T(k)]}G\log T \geqslant \Theta\left(\frac{\log T}{\Delta_k^2}\right),$$

which yields the communication lower bound for arm k as follows,

$$\mathbb{E}[C_T(k)] \geqslant \Omega\left(\log\left(\frac{1}{G\Delta_k^2}\right)\right) = \Omega(\log\Delta_k^{-1}). \tag{10}$$

Therefore, summing over all suboptimal arms yields the overall communication lower bound $\Omega\left(\sum_{k:\Delta_k>0}\log\Delta_k^{-1}\right)$.

Next, we prove another communication lower bound $\Omega(M)$, which, together with the above bound, concludes the proof. Blow, we prove the bound via contradiction. That is, we start from assuming the communication is less than cM where c is a constant.

Denote $Y^{(m)}$ as the number of integers or real numbers that agent m sends or receives throughout a run. $Y^{(m)}$ is a random

variable. Since expected communication cost is less than cM, we have

$$\sum_{m=1}^{M} \mathbb{E}[Y^{(m)}] \leqslant cM.$$

Denote $\mathcal{S}$ as the set of $M/2$ agents with smaller $\mathbb{E}[Y^{(m)}]$. The expected communication cost of any agent $m \in \mathcal{S}$ is at most $2c$. For any agent $m \in \mathcal{S}$, we have $\mathbb{P}(Y^{(m)} \geqslant 1) \leqslant \mathbb{E}[Y^{(m)}] \leqslant 2c$, where the first inequality is by Markov's inequality. That is, for any of these agents, the probability of communicating with some agent is less than $2c$. Suppose that agent m is such an agent. Then, we can map the communication protocol to a single-agent algorithm by simulating the learning process of agent m.

The simulation proceeds as follows: We engage with a single-agent bandit over a time horizon of T, executing the code corresponding to agent m within the specified protocol. In the absence of any communication requirements, we advance to the subsequent line in agent m's code. However, if the code initiates message transmission or awaits a message, we conclude the execution. Throughout the remaining time steps, we employ a single-agent optimal algorithm, specifically the one employed to achieve the optimal regret upper bound, denoted as R_T^*.

Then, if agent m's code has δ probability of involving in communication, and if agent m's regret $R_T^{(m)} \leqslant A$ (in its original distributed algorithm design), via this reduction, we can obtain an algorithm for single-agent MAB with expected regret

$$R_T \leqslant A + \delta \cdot R_T^*.$$

By the regret lower bound result of [25, Theorem 1], we have

That is,

$$\begin{aligned}\liminf_{T\to\infty} \frac{A}{\log T} &\geqslant (1-\delta) \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)} \\ &\geqslant (1-2c) \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)}. \end{aligned} \tag{11}$$

If setting $c = 0.0005$, then the regret of any agent m in set $\mathcal{S}$ fulfills the above lower bound. So, the total regret is at least

$$\sum_{m\in\mathcal{S}} \liminf_{T\to\infty} \frac{A}{\log T} \geqslant \left(\frac{1}{2} - c\right) \cdot M \sum_{k>1} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu_1)},$$

which contradicts the near-optimal regret upper bound (without the linear dependence on M). Therefore, a CMA2B algorithm with near-optimal regret requires at least $\Omega(M)$ communications. □

B. Proof of Regret Upper Bounds in Theorem 2

We sketch the high-level proof idea of Theorem 2. The regret mainly consist of two types, where Type-I represents the condition that arms are in the confidence interval and Type-II means that arms are outside the confidence interval. Type-I is also divided into two parts, where part I is the regret generated in the exploration and part II is the regret generated in the exploitation. In Section III, we tailor the elimination-based strategy in DoE-bandit such that all agents pull arms in a synchronized manner. Hence, all agents track the number of pulls of any arm k by $Mn_t(k)$. In this way, CMA2B involves multiple distributed online estimation problems, each of which can be solved by DoE separately. By Lemma 1, we show that the DoE subroutine builds up a confidence interval for the mean reward of an arm. By applying Lemma 1 to the standard analysis of our bandit algorithm, we prove the regret bound in Theorem 2. To prove the communication cost, we highlight the fact that agents communicate the mean of arm k only when the radius of the confidence interval provided by DoE is larger than $\Delta_k/2$ (such that the investigated arm remains in the candidate set). Combining with the rules of DoE that agents communicate only when the radius of the confidence interval reduces to $1/\beta$ of the previous, we prove the bound on the communication cost.

1) Step 1: Upper Bound the Estimation Error of DoE: By running the DoE subroutine, the bandit learning algorithm can build up a confidence interval for the mean reward of an arm. In Lemma 1, we provide the estimation performance of DoE in estimating the mean of an arm. Lemma 1 shows the upper bound of estimation error of DoE is proportional to the radius of the confidence interval with system-wide samples.

Lemma 1: Assume M agents independently sample an arm with an i.i.d. reward process with unknown mean $\mu(k)$, and $n_t(k)$ is the available samples for each agent up to time slot t. With $\beta > 1$ and $\mathtt{ECR}_t(k) = \alpha \mathtt{CR}(Mn_t(k), \delta_t)$, where $\delta_t \in (0,1)$ is a sequence of non-increasing parameters, then, for any t, with probability $1 - Mt\delta_t^{1/2}$, we have $|\hat{\mu}_t^{(m)}(k) - \mu(k)| \leqslant (2\alpha\beta + \beta)\mathtt{CR}(Mn_t(k), \delta_t)$.

Proof: We prove the lemma by analyzing the following three cases. Let s denote the last detection point, i.e., the last time slot (before t) that the condition in Line 6 of Algorithm 2 holds.

Case (1): the agent communicated at the last detection point s. In this case, the estimate $\hat{\mu}_t^{(m)}(k)$ is obtained by averaging $Mn_s(k) + n_t(k) - n_s(k)$ samples. Hence, the following equation holds with probability $1 - Mt\delta_t^{1/2}$,

$$\begin{aligned}|\hat{\mu}_t^{(m)}(k) - \mu(k)| &\overset{(a)}{\leqslant} \mathtt{CR}_{[0,1]}(Mn_s(k) + n_t(k) - n_s(k), \delta_t) \\ &\overset{(b)}{\leqslant} \mathtt{CR}_{[0,1]}(Mn_s(k), \delta_t) \overset{(c)}{\leqslant} \mathtt{CR}_{[0,1]}(Mn_s(k), \delta_s) \\ &\overset{(d)}{\leqslant} \beta \mathtt{CR}_{[0,1]}(Mn_t(k), \delta_t),\end{aligned}$$

where the inequality (a) is proved by Hoeffding's inequality and union bound (see below), inequality (b) is due to that the confidence radius CR increases with a smaller number of samples, inequality (c) is because δ_t is decreasing with respect to t and $s < t$, and the inequality (d) is due to that the condition in Line 6 is false at time slot t.

Then, we present the detailed steps for proving inequality (a) as Equation (29), shown at the bottom of page 11, where the equation (a1) is due to union bound, and inequality (a2) is by applying Hoeffding's inequality.

In this case, the result in Lemma 1 holds.

Case (2): there is no communication at s. Let A be the sum of samples obtained by agent m if communication *happened*

at s. We can infer Equation (12), as shown at the bottom of the page.

The equation is based on the fact that agent always has the local samples after s no matter there is communication at s.

Hence,

$$\begin{aligned}&\left|\hat{\mu}_t^{(m)}(k)-\frac{A}{Mn_s(k)+n_t(k)-n_s(k)}\right|\\&=\frac{\left|Mn_s(k)\hat{\mu}_{\mathrm{ALE},s}^{(m)}(k)-\sum_{m'=1}^{M}X_s^{(m')}(k)\right|}{(Mn_s(k)+n_t(k)-n_s(k))}\\&\leqslant\left|\hat{\mu}_{\mathrm{ALE},s}^{(m)}(k)-\frac{1}{Mn_s(k)}\sum_{m'=1}^{M}X_s^{(m')}(k)\right|\\&\overset{(a)}{\leqslant}2\alpha\,\mathrm{CR}_{[0,1]}(Mn_s(k),\delta_s(k))\\&\leqslant 2\alpha\,\mathrm{CR}_{[0,1]}(Mn_s(k),\delta_t(k))\\&\leqslant 2\alpha\beta\,\mathrm{CR}(Mn_t(k),\delta_t(k)),\end{aligned}$$

where inequality (a) is because the condition in Line 8 in Algorithm 1 does not hold at time slot s (since there is no communication at s). In this case, $\left|\hat{\mu}_{\mathrm{ALE},s}^{(m)}(k)-\hat{\mu}_{\mathrm{ALE},s}^{(m')}(k)\right|\leqslant 2\mathrm{ECR}_s(k)=2\alpha\mathrm{CR}_{[0,1]}(Mn_s(k),\delta_s)$ for any agents m and m'. Also, $\frac{1}{Mn_s(k)}\sum_{m'=1}^{M}X_s^{(m')}(k)$ averages all samples up to s, and hence its value lies between $\min_{m'}\hat{\mu}_{\mathrm{ALE},s}^{(m')}(k)$ and $\max_{m'}\hat{\mu}_{\mathrm{ALE},s}^{(m')}(k)$, which weight partial local samples with a factor M to replace missing ones. Combining the two facts yields inequality (a).

Since A contains the same set of samples as the $\hat{\mu}_t^{(m)}(k)$ in Case (1), the following equation also holds with probability $1-Mt\delta_t^{1/2}$:

$$\left|\frac{A}{Mn_s(k)+n_t(k)-n_s(k)}-\mu(k)\right|\leqslant\beta\mathrm{CR}_{[0,1]}(Mn_t(k),\delta_t).$$

Combining the above two equations yields that with probability $1-Mt\delta_t^{1/2}$,

$$\left|\hat{\mu}_t^{(m)}(k)-\mu(k)\right|\leqslant(2\alpha+1)\beta\mathrm{CR}_{[0,1]}(Mn_t(k),\delta_t).$$

As a result, we prove the Lemma 1. □

2) Step 2: Upper Bound the Exploration and Exploitation Regrets: According to the results in Lemma 1, letting $\mathrm{ECR}_t(k)=\alpha\min\{1,\mathrm{CR}_{[0,1]}(Mn_t(k),\delta_t)\}$, each agent can attain the order-optimal estimate (up to a constant factor $2\alpha\beta+\beta$) for the mean reward, which slightly degrades the performance of the bandit algorithm. We prove the regret of `DoE-bandit` by using the observation in Lemma 1.

The complete regret of `DoE-bandit` could be transformed into two parts: part I is the regret caused by elimination-based exploration, and part II is caused by inaccurate exploitation.

a) Regret Bound of Part I: By the regret decomposition in [27], the individual regret can be written as

$$\mathbb{E}[R_T^{(m)}]=\sum_{k=1}^{K}\Delta_k n_T(k).$$

The theorem follows by showing that $\mathbb{E}[n_T(k)]$ is not too large for suboptimal arms $k, k\neq k^\star$. From the update of the candidate arm set (equation (3)) and results in Lemma 1, the upper bound of the sample count is

$$n_T(k)\leqslant\frac{8(2\alpha\beta+\beta)^2\log\delta_T^{-1}}{M\Delta_k^2}+1. \tag{13}$$

This regret of agent m in part I is bounded by

$$\mathbb{E}[R_{T,I}^{(m)}]\leqslant\sum_{k:\Delta_k>0}\frac{8(2\alpha\beta+\beta)^2\log\delta_T^{-1}}{M\Delta_k}+\sum_{k:\Delta_k>0}\Delta_k. \tag{14}$$

b) Regret Bound of Part II: The decision-making in this part is mainly based on the experience gained in the exploration phase (Part I). Since choosing the optimal arm incurs no regret, we only need to consider the regret from selecting suboptimal arms. Let $T_t(k)$ denote the number of times arm k is pulled up to time t, independent of Part I. For convenience, we use $T_t(k)$ without the subscript t. We then define the 'good' event G_k as follows:

$$\begin{aligned}&G_k\\&=\{\mu_{k^\star}<\min_{t\in[T]}\mathrm{ECR}_t(k^\star)\}\cap\{\hat{\mu}_{u_k}(k)+\mathrm{ECR}_{u_k}(k)<\mu_{k^\star}\},\end{aligned} \tag{15}$$

where u_k represents that arm k will be played at most u_k times when event G_k occurs. If G_k occurs, then arm k will be played at most u_k times: $T_t(k)\leqslant u_k$. So G_k is the event when $\mu_{k^\star}$ is never underestimated by the upper confidence bound of the first arm, while at the same time the upper confidence bound for the mean of arm k after u_k observations are taken from this arm is below the pay-off of the optimal arm.

The sample count $T_t(k)$ is bounded by

$$\begin{aligned}\mathbb{E}[T_t(k)]&=\mathbb{E}[\mathbb{I}\{G_k\}T_t(k)]+\mathbb{E}[\mathbb{I}\{G_k^c\}T_t(k)]\\&\leqslant u_k+\mathbb{P}(G_k^c)T.\end{aligned} \tag{16}$$

Contrary to the 'Good' event G_k, the complement event G_k^c is defined as

$$G_k^c=\{\mu_{k^\star}\geqslant\min_{t\in[T]}\mathrm{ECR}_t(k^\star)\}\cup\{\hat{\mu}_t(k)+\mathrm{ECR}_t(k)\geqslant\mu_{k^\star}\}. \tag{17}$$

$$\begin{aligned}&\left|(Mn_s(k)+n_t(k)-n_s(k))\hat{\mu}_t^{(m)}(k)-A\right|\\&=\left|\left(Mn_s(k)\hat{\mu}_{\mathrm{ALE},s}^{(m)}(k)+\left(X_t^{(m)}(k)-X_s^{(m)}(k)\right)\right)-\left(\sum_{m'=1}^{M}X_s^{(m')}(k)+\left(X_t^{(m)}(k)-X_s^{(m)}(k)\right)\right)\right|\\&=\left|Mn_s(k)\hat{\mu}_{\mathrm{ALE},s}^{(m)}(k)-\sum_{m'=1}^{M}X_s^{(m')}(k)\right|.\end{aligned} \tag{12}$$

For the first term in equation (17), it is decomposed into

$$\{\mu_{k^\star} \geqslant \min_{t\in[T]} \hat{\mu}_t(k^\star) + \mathtt{ECR}_t(k^\star)\} \subset \bigcup_{s\in[T]} \{\mu_{k^\star} \geqslant \hat{\mu}_s(k^\star) + \mathtt{ECR}_s(k^\star)\},$$

i.e.,

$$\begin{aligned}&\mathbb{P}(\mu_{k^\star} \geqslant \min_{t\in[T]} \hat{\mu}_t(k^\star) + \mathtt{ECR}_t(k^\star)) \\ &\leqslant \mathbb{P}\left(\bigcup_{s\in[T]} \{\mu_{k^\star} \geqslant \hat{\mu}_s(k^\star) + \mathtt{ECR}_s(k^\star)\}\right) \\ &\leqslant \sum_{s=1}^{T} \mathbb{P}(\mu_{k^\star} \geqslant \hat{\mu}_s(k^\star) + \mathtt{ECR}_s(k^\star)) \leqslant \sum_{s=1}^{T} \delta_s. \end{aligned} \tag{18}$$

The next step is to bound the probability of the second set in equation (17). Assume that u_k is chosen large enough that

$$\Delta_k - \mathtt{ECR}_{u_k}(k) > c\Delta_k, \tag{19}$$

for some $c \in (0,1)$ to be chosen later. Then, since $\mu_{k^\star} = \mu_k + \Delta_k$, and using the Hoeffding's inequality, we have

$$\begin{aligned}&\mathbb{P}\{\hat{\mu}_{u_k}(k) + \mathtt{ECR}_{u_k}(k) \geqslant \mu_{k^\star}\} \\ &= \mathbb{P}\{\hat{\mu}_{u_k}(k) - \mu_i \geqslant \Delta_k - \mathtt{ECR}_{u_k}(k)\} \\ &\leqslant \mathbb{P}\{\hat{\mu}_{u_k}(k) - \mu_i \geqslant c\Delta_k\} \leqslant \exp\left(-\frac{u_k c^2 \Delta_k^2}{2}\right). \end{aligned} \tag{20}$$

Equation (18) gives the probability that arm $k^\star$ has been underestimated and Equation (20) gives the probability that the sub-optimal arm k is overestimated. Combining equations (18) and (20), we have

$$\mathbb{P}(G_k^c) \leqslant \sum_{s=1}^{T} \delta_s + \exp\left(-\frac{u_k c^2 \Delta_k^2}{2}\right). \tag{21}$$

According to equation (16), we have

$$\mathbb{E}[T_t(k)] \leqslant u_k + T\left(\sum_{s=1}^{T} \delta_s + \exp\left(-\frac{u_k c^2 \Delta_k^2}{2}\right)\right). \tag{22}$$

To minimize the sample count, we choose a suitable u_k as

$$u_k = \left\lceil (2\alpha\beta+\beta)\frac{\log \delta_t^{-1}}{2M(1-c)^2\Delta_k^2} \right\rceil.$$

Assuming that $\delta_t = T^{-2}$, the bound of the sample count $T_t(k)$ is

$$T_t(k) \leqslant u_k + 1 + T^{1-\frac{2c^2}{(1-c)^2}}. \tag{23}$$

Defining $c = 0.5$, we have

$$T_t(k) \leqslant \frac{2(2\alpha\beta+\beta)^2 \log \delta_t^{-1}}{M\Delta_k^2} + 3. \tag{24}$$

This regret of agent m in part II is bounded by

$$\mathbb{E}[R_{T,II}^{(m)}] \leqslant \sum_{k:\Delta_k>0} \frac{2(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} + 3\sum_{k:\Delta_k>0} \Delta_k. \tag{25}$$

Combining equation (14) and equation (25) with suitable proportions, the totoal regret is

$$\begin{aligned}\mathbb{E}[R_T^{(m)}] &= \frac{1}{\Gamma}\mathbb{E}[R_{T,I}^{(m)}] + \left(1-\frac{1}{\Gamma}\right)\mathbb{E}[R_{T,II}^{(m)}] \\ &\overset{(a)}{\leqslant} \sum_{k:\Delta_k>0} \frac{\left(2+\frac{6}{\Gamma}\right)(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} \\ &\quad + \sum_{k:\Delta_k>0} \left(3-\frac{2}{\Gamma}\right)\Delta_k. \end{aligned} \tag{26}$$

When an arm is outside the confidence interval, it suffers regret which is bounded by $KM^2\sum_{t\leqslant s} t\delta_t^{1/2}$. Considering both regrets of Type-I and Type-II, we have

$$\begin{aligned}\mathbb{E}[R_T^{(m)}] &\leqslant \sum_{k:\Delta_k>0} \frac{\left(2+\frac{6}{\Gamma}\right)(2\alpha+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} \\ &\quad + \sum_{k:\Delta_k>0} \left(3-\frac{2}{\Gamma}\right)\Delta_k \\ &\quad + 2KM^2 \sum_{t\leqslant s} t\delta_t^{1/2}. \end{aligned} \tag{27}$$

Substitute equation (27) into the definitions of individual and group regrets, the proof is completed.

C. Proof of Communication Cost in Theorem 2

We analyze the communication overheads of `DoE-bandit` arm by arm. If there is a Type-II decision before τ, we use MT to upper bound the communication overheads. The expected communication complexity in this case is then

$$KM^3T\sum_{t\leqslant\tau} t\delta_t^{1/2}. \tag{28}$$

In the following, we focus on Type-I decisions. For any suboptimal arm k (with $\Delta_k > 0$), let τ_k be the last time that `DoE-bandit` pulls the arm k. At τ_k, we have

$$4(2\alpha\beta+\beta)\mathtt{CR}_{[0,1]}(Mn_{\tau_k}(k), \delta_t) \geqslant \Delta_k.$$

The above equation is proved in the proof of Theorem 2 at Section V-B. With $\mathtt{ECR}_{\tau_k}(k) = \alpha\mathtt{CR}_{[0,1]}(Mn_{\tau_k}(k), \delta_t)$, and

$$4(2\alpha\beta+\beta)\frac{1}{\alpha}\mathtt{ECR}_{\tau_k}(k) \geqslant \Delta_k.$$

Hence, up to time τ_k, the communications due to arm k is (recall $\mathtt{ECR}_1(k) = 1$)

$$\log_\beta \frac{\mathtt{ECR}_1(k)}{\mathtt{ECR}_{\tau_k}(k)} \leqslant \log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k}\right).$$

The expected number of communications by suboptimal arms is at most

$$\sum_{k:\Delta_k>0} \log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k}\right).$$

For the optimal arm, the number of communications (when there is no Type-II decision) can be upper bounded by the largest communication overheads of suboptimal arms. That is, the number of communications about the optimal arm is upper bounded by $O(\log_\beta(1/\Delta))$ where the Δ corresponds to

the smallest non-zero reward gap. That is because when there are multiple arms in the candidate set, the optimal arm with others in the candidate set is pulled in a round-robin manner and incurs the same communication overheads as others in the set; and when there is only one arm left in the candidate set, the DoE-bandit stops communication. So, to sum up, the total communication overheads is upper bounded by

$$\sum_{k:\Delta_k>0} \log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k}\right)+\log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta}\right) \leqslant 2\sum_{k:\Delta_k>0}\log_\beta\left(\frac{4(2\alpha\beta+\beta)}{\alpha\Delta_k}\right).$$

At each communication time, agents spend totally $3M$ messages in collecting messages and synchronize the estimates in each agent. In addition, DoE may update the candidate set in agents when an arm is eliminated, that costs another $M(K-1)$ messages. Therefore, combined with (28)), the expected communication overheads of DoE-bandit (the total number of messages) is upper bounded by (6)).

VI. Numerical Results

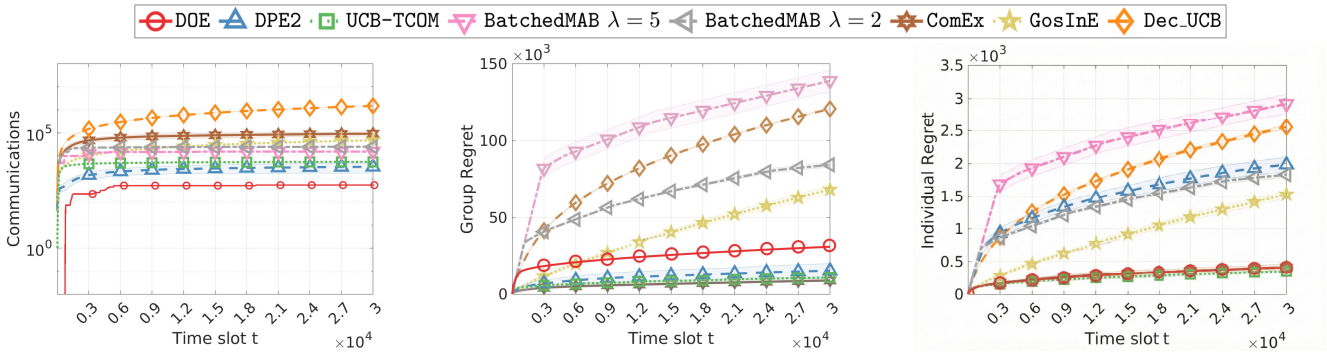
This section reports numerical experiments to corroborate the performance of DoE-bandit. They highlight the advantage of DoE-bandit based on group and individual regrets, and communication costs over state-of-the-art baselines.

Setups and Baselines. In DoE-bandit algorithm, we set parameters $\alpha=1,\beta=3,\Gamma=0.1$ and $\delta_t=1/T^2$. We run 50 trials of each experiment and plot the means as lines and their standard deviations as shaded regions. We compare the regret and communication costs of DoE-bandit to those of six baselines, ComEx [30], GosInE [10], Dec_UCB [49], DPE2 [41], UCB-TCOM [43]) and BatchedMAB [23] outlined in Table I. We note that some of the baseline algorithms are developed for a set of agents that are connected through an underlying graph topology. Hence, to make the comparison fair, we consider a complete graph for all algorithms so that any two agents can communicate. Among these baselines, the most relevant ones to ours are BatchedMAB and DPE2, as they also achieve constant communication costs. Especially, Batched has a parameter λ ($\geqslant 2$) that is used to tune its communication frequency. To make the comparison fair, we pick both $\lambda=2$ and $\lambda=5$ in the experiments. All other baselines' parameters follow their default choice.

Experimental Results. Figure 1 reports the comparison results in group regret, individual regret, and communication costs. Figure 1a shows that DoE-bandit achieves the smallest communication costs among all algorithms. The experiments are conducted with $K=100$ arms, $M=50$ agents, and $T=30$K, and each arm is associated with a Bernoulli distribution with mean randomly taken from the click-through-rate in Ad-Clicks [2]. Figure 1b reports the group regrets of algorithms, where DoE-bandit is not as good as DPE2, ComEx, and UCB-TCOM. This is because DoE-bandit is based on the arm-elimination policy and others are UCB-like algorithms. It is known that with the same order-wise regret performance, UCB algorithms are empirically better than elimination ones in general [19, §6]. Figure 1c reports the maximum individual agent regret. Our algorithm has a similar performance compared with the UCB-like algorithms. In Figure 1c, ttDPE2—one of the other algorithms with constant communication cost—suffers poor individual regret since DPE2 leverages a leader-follower structure, where the leader agent incurs high individual regret. For both $\lambda=2$ and $\lambda=5$ cases, BatchedMAB, the other baseline with constant communication, has relative bad group and individual regret performance.

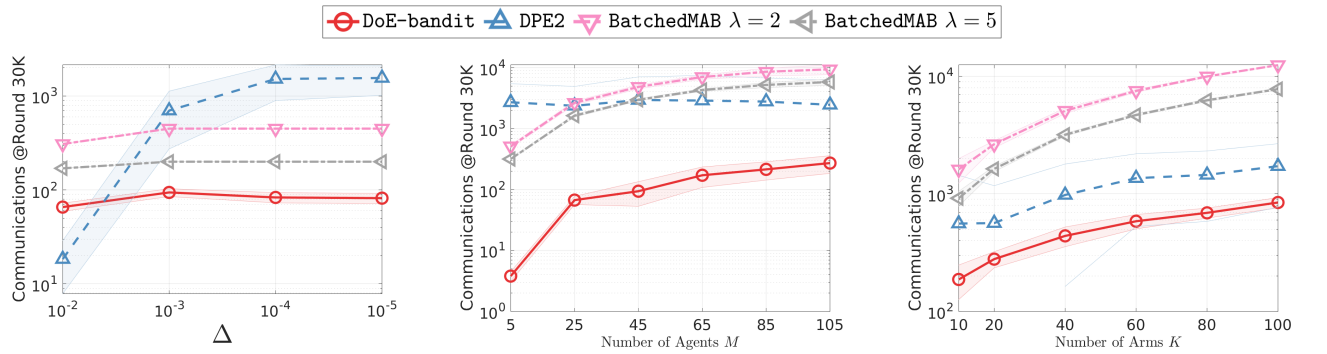
In Figure 2, we compare the communication cost of DoE-bandit to the constant communication cost alternatives DPE2 and BatchedMAB across various parameter settings. Three parameters are analyzed: (1) reward gap Δ between arms ($K=10$ with mean $\mu(k)=0.09+k\Delta$, $M=5$) in

$$\begin{aligned}
&\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)|\leqslant \mathtt{CR}_{[0,1]}(Mn_s(k)+n_t(k)-n_s(k),\delta_t)\right)\\
&=\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)|\leqslant\sqrt{\frac{\log\delta_t^{-1}}{2\left(Mn_s(k)+n_t(k)-n_s(k)\right)}}\right)\\
&=1-\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)|>\sqrt{\frac{\log\delta_t^{-1}}{2\left(Mn_s(k)+n_t(k)-n_s(k)\right)}}\right)\\
&\overset{(a1)}{=}1-\sum_{n=1}^{M\cdot t}\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)|>\sqrt{\frac{\log\delta_t^{-1}}{2n}}\,\middle|\, Mn_s(k)+n_t(k)-n_s(k)=n\right)\\
&\qquad\times\mathbb{P}\left(Mn_s(k)+n_t(k)-n_s(k)=n\right)\\
&\geqslant 1-\sum_{n=1}^{M\cdot t}\mathbb{P}\left(|\hat{\mu}_t^{(m)}(k)-\mu(k)|>\sqrt{\frac{\log\delta_t^{-1}}{2n}}\,\middle|\, Mn_s(k)+n_t(k)-n_s(k)=n\right)\\
&\overset{(a2)}{\geqslant} 1-\sum_{n=1}^{M\cdot t}\delta_t^{1/2}\geqslant 1-Mt\delta_t^{1/2},
\end{aligned}\tag{29}$$



(a) Communication cost (b) Group regret (c) Individual regret

Fig. 1. DoE-bandit (this work) vs. baseline algorithms listed in Table I.



(a) Vary reward gap (b) Vary agent number (c) Vary arm number

Fig. 2. Communications: DoE-bandit vs. DPE2 and BatchedMAB.

Figure 2b; (2) number of agents M ($K = 20$) in Figure 2a; and (3) number of arms K ($M = 25$) in Figure 2c, where the reward means of Figures 2b and 2c are drawn from Ad-Clicks [2]. The log-y-axis represents cumulative communication costs at the end. DoE-bandit consistently outperforms DPE2 and BatchedMAB in all scenarios, except for large Δ, where DPE2 excels. Notably, as Δ decreases, DoE-bandit's communication costs remain stable, contrasting with DPE2, which experiences an increase. This is attributed to DoE-bandit's superior $O(KM \log \Delta^{-1})$ communication cost compared to DPE2's $O(K^2M^2\Delta^{-2})$. Lastly, communication costs for all algorithms rise with increasing M (Figure 2b) or K (Figure 2c), confirming the dependence on K and M in their bounds.

VII. Relevant Applications

In this section, we discuss how our algorithm effectively addresses challenges in various distributed systems including drone swarms, networks etc. In each case, our algorithm provides a robust solution that can improve overall & individual system performance and achieve low communication costs independent of time.

Applying CMA2B to Networking: In networking, the multi-armed bandit (MAB) framework has been widely applied, such as in link selection [4], [12], to optimize network performance in dynamic and uncertain environments. The core idea behind these applications is to treat each potential decision—such as the link a TCP flow selects in link selection—as an "arm" in a multi-armed bandit problem. The goal is to maximize a specific performance metric (e.g., throughput, latency, or packet loss) by selecting the best link over time, while balancing exploration (trying new links) and exploitation (favoring the best-performing link based on past experience).

As an example, we now demonstrate how CMA2B can be applied to link/channel selection in a network. Nodes within the network can choose from a variety of communication links or channels between devices, such as those between nodes in a wireless network or between a device and an access point. To optimize packet transmission performance—measured by metrics like delay and packet loss rate—the nodes aim to select the most appropriate link or channel for transmitting data flows. Once a link or channel is deployed, users receive performance metrics (similar to the arms in a multi-armed bandit), based on factors such as delay and packet loss rate. For example, the work in [20] built up a channel access model aided by cognitive radio.to solve the channel selection problem in Internet of Things Networks [20]. The model consists of K available channels, corresponding to the arms in MAB, and each frame is separated into time slots in the network. Each channel is assigned the availability probability p_k, which denotes the degree of channel congestion. Without knowing the prior knowledge on the availabilities,

the cognitive users adopt the MAB algorithms to learn the best channel from K channels to send data. Since multiple users may share common links in a network, it becomes possible for them to collaboratively learn the optimal link by sharing information about link performance, which can be modeled as the CMA2B problem. In this context, both communication cost and individual regret should be considered: communication costs relate to minimizing the communication overhead of sharing information (especially in a wireless network where the bandwidth is scarce), while individual regret ensures that each user has a fair experience when using the links.

In addition to link selection, there are several other applications. For instance, in Wi-Fi Access Point Selection [34], [45], devices use distributed MAB to choose the best AP from multiple available options. In Cellular Load Balancing [21], [28], MAB algorithms help balance load across base stations, enabling devices to switch between them based on performance data. In Edge and Cloud Computing [13], [15], distributed MAB can optimize resource allocation by selecting the best path or server based on latency or load. Due to page limitations, we will skip the details of these applications.

Applying CMA2B to Drone Swarm: In a drone swarm [5], [38] (or a fleet of drones), multiple drones work together to perform tasks such as surveillance, search-and-rescue, photography, or environmental monitoring. Each drone operates based on its own planned actions and local observations.

In general, the bandit-based methodology enables drones to explore and utilize more advanced majority strategies grounded in realistic probability distributions. These strategies account for factors such as geographic location, proximity to the base station, landmarks, and the drones' available energy. By leveraging a database (DB) of landmarks and applying machine learning techniques, as MAB, to drone swarms, this approach can enhance the overall decision-making process.

For example, various countermeasures can be utilized by a swarm of uncrewed aerial vehicles (UAVs) to evade ground-based attacks. However, when confronted with a high volume of threats, it becomes difficult for the swarm to identify the countermeasures that will most effectively enhance drone survivability. The multi-armed bandit framework offers a solution by guiding the swarm in selecting the optimal countermeasures. Various countermeasures that have differing levels of effectiveness against attacks can be taken as the arms, and assume that the outcome of whether a drone can evade a ground attack follows an independent and identically distributed (i.i.d.) distribution. To identify the most effective countermeasure, each drone selects and implements a countermeasure, then collects the outcome as a sample. In a swarm where drones are capable of wireless communication, they collaborate to learn the optimal countermeasure by sharing samples. The performance of the cooperative learning algorithm can be assessed using group regret. Additionally, the failure or malfunction of a single drone can significantly affect the performance of the entire swarm. To ensure that each drone can quickly and effectively learn the optimal policy, the cooperative learning algorithm is evaluated based on individual regret as well.

VIII. Conclusion

This paper presented DoE-bandit, a fully distributed algorithm for a multi-agent multi-armed bandits problem. DoE-bandit achieves the optimal group and individual regret with constant communication overhead. We also proposed a new communication lower bound matching the constant communication overhead. This implied DoE-bandit is near-optimal in all three metrics. The theoretical claims are verified by numerical experiments and show that DoE-bandit outperforms prior algorithms.

The core communication policy proposed in this paper could be further extended in multiple directions. To address the exploitation-exploration dilemma in bandit learning, DoE-bandit adopts an elimination-based strategy, but there also exists other strategy, such as Upper Confidence Bound and Thompson sampling. As Figure 1 shows UCB-like algorithm has better regret performance, it is interesting to develop an UCB/TS-based algorithm which achieves better practical performance with guaranteeing the same optimal theoretical results claimed in this work. Second, one can extend the work to capture more practical concerns, such as considering an underlying topology for agents, communication delays between agents, and lossy communication between agents.

Appendix

A. Appendix / Extension of DoE-Bandit With Time Delay

In this section, we extend the basic DoE-bandit algorithm to accommodate deterministic communication delays and transition from a broadcast setting — where each agent can communicate with all others — to a more realistic networked setting, where agents can only communicate with their immediate neighbors. In this communication model, we allow for message-passing, meaning that agents may forward received messages to their neighbors. We show that DoE-bandit, with only minor modifications, continues to achieve near-optimal group and individual regret bounds as stated in Theorem 2, incurring only a constant additional cost independent of T.

Let D denote the deterministic delay associated with message propagation, which is upper bounded by the diameter of the network. This delay model can be further generalized to allow for random delays, as long as the delay remains bounded by D. Due to this delay, local information from each agent is not immediately available to others. Consequently, the effective sample count for each arm increases only when the corresponding reward observations have been received by all agents. This ensures that agents can base their decisions on a consistent set of shared observations. Based on the intuition above, an extension of DoE-bandit is proposed, which is Algorithm 3. Compared with the original algorithm, we need to transmit all agents' sampling, as well as their sample counts, to help all agents learn the global information.

Theorem 3: When the communication arm set parameter $\alpha > 0$ and buffering-ratio $\beta > 1$, given all delays of communication is no greater than D, DoE-bandit attains a near-optimal group regret upper bound in terms of number of decision rounds T, arms K, and agents M, or formally,

Algorithm 3 DoE-delay: an algorithm for estimating the mean of arm k by agent m, subscript t is dropped (A fully distributed network)

Parameters: $\beta > 1$;
Initialization: $\hat{\mu}^{(m)}_{\texttt{ALE}}(k)$, $n(k) \leftarrow 0$ $\hat{\mu}_{\texttt{com}}(k) \leftarrow 0$, $\texttt{ECR}_{\texttt{last}} \leftarrow 0$; $X^{(m')}(k) \leftarrow 0$, $X^{(m')}_{\texttt{last}}(k) \leftarrow \infty$, $\forall m' \in \mathcal{M}_m, \texttt{ECR}_t(k) \leftarrow 0$
for each round t when the agent gets a new sample **do**
  $n(k) \leftarrow n(k) + 1$
  Update $X^{(m)}(k)$ with the new sample
  if $\beta\texttt{ECR}(k) \leqslant \texttt{ECR}_{\texttt{last}}$ **then**
    $\texttt{ECR}_{\texttt{last}} \leftarrow \texttt{ECR}(k)$
    if $|\hat{\mu}^{(m)}_{\texttt{ALE}}(k) - \hat{\mu}_{\texttt{com}}(k)| > \texttt{ECR}(k)$ **then**
      //Communicate to synchronize the estimates
      Exchange the sample counts of all arms and their cumulative rewards with agent m's neighbors
      Update the stack of the sample counts and cumulative rewards according to the new messages and update $\hat{\mu}_{\texttt{com}}(k)$
      $X^{(m')}_{\texttt{last}}(k) \leftarrow X^{(m')}(k)$ for all $m' \in \mathcal{M}$
  Update $\hat{\mu}^{(m)}_{\texttt{ALE}}(k)$ according to (1) and $\hat{\mu}^{(m)}(k)$ according to (2)

1) (Group Regret)

$$\mathbb{E}[R_T] \leqslant \sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{\Delta_k} + 2KM^3T \sum_{t\leqslant T} t\delta_t^{1/2} + DM \sum_{k:\Delta_k>0} \Delta_k,$$

2) (Individual Regret)

$$\mathbb{E}\left[\bar{R}_T\right] \leqslant \sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} + 2KM^2T \sum_{t\leqslant T} t\delta_t^{1/2} + D \sum_{k:\Delta_k>0} \Delta_k.$$

Proof: The proof procedure of Theorem 2 still applies to Theorem 3. The only change is that the upper bound of pull-times should be updated as follows,

$$\sum_{k:\Delta_k>0} n_t(k) \overset{(a)}{\leqslant} \sum_{k:\Delta_k>0} (n_t(k) + D) \overset{(b)}{\leqslant} \sum_{k:\Delta_k>0} \frac{8(2\alpha+1)^2\beta^2 \log \delta_t^{-1}}{M\Delta_k^2} + D, \tag{30}$$

where inequality (a) is because each agent will sample sub-optimal arms additional D times compared with the theoretical counters $n_t(k)$, and inequality (b) is similar to the proof of Theorem 2.

With the rest proof the same as Theorem 2's, the group regret with time delays is upper bounded as follows

$$\mathbb{E}[R_T] \leqslant \sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{\Delta_k} + 2KM^3T \sum_{t\leqslant T} t\delta_t^{1/2} + DM \sum_{k:\Delta_k>0} \Delta_k.$$

As the symmetry of DoE-bandit still holds, the individual regret upper bound immediately follows, □

$$\mathbb{E}\left[\bar{R}_T\right] \leqslant \sum_{k:\Delta_k>0} \frac{8(2\alpha\beta+\beta)^2 \log \delta_T^{-1}}{M\Delta_k} + 2KM^2T \sum_{t\leqslant T} t\delta_t^{1/2} + D \sum_{k:\Delta_k>0} \Delta_k.$$

By adopting similar parameter setups, we can recover the results reported in Corollary 1, except for an additional term of order $DM\sum_k \Delta_k$, which is independent of the T. In realistic systems, as $D \ll T$, the impact of delays does not dominate the overall results.

ACKNOWLEDGMENT

The authors are grateful for the help from the Interdisciplinary Research Center for Future Intelligent Chips (Chip-X) and Yachen Foundation.

REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, May 2002.
[2] (2015). *Avito Context Ad Clicks*. [Online]. Available: https://www.kaggle.com/c/avito-context-ad-clicks
[3] J. Yi and M. Vojnovic, "On regret-optimal cooperative nonstochastic multi-armed bandits," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, May 2023, pp. 1329–1335.
[4] E. Beres and R. Adve, "Selection cooperation in multi-source cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 118–127, Jan. 2008.
[5] L. Bertizzolo et al., "SwarmControl: An automated distributed control framework for self-optimizing drone networks," in *Proc. IEEE INFOCOM - IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 1768–1777.
[6] I. Bistritz and N. Bambos, "Cooperative multi-player bandit optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 2016–2027.
[7] S. Bubeck, "Bandits games and clustering foundations," Ph.D. thesis, Dept. Statist., Université des Sciences et Technologie de Lille-Lille I, Villeneuve-d'Ascq, France, 2010.
[8] S. Buccapatnam, J. Tan, and L. Zhang, "Information sharing in distributed stochastic bandits," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jun. 2015, pp. 2605–2613.
[9] M. Chakraborty, K. Y. P. Chua, S. Das, and B. Juba, "Coordinated versus decentralized exploration in multi-agent multi-armed bandits," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 164–170.
[10] R. Chawla, A. Sankararaman, A. Ganesh, and S. Shakkottai, "The gossiping insert-eliminate algorithm for multi-agent bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3471–3481.
[11] Y.-Z. J. Chen et al., "On-demand communication for asynchronous multi-agent bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2023, pp. 3903–3930.
[12] D. Deng, J. Xia, L. Fan, and X. Li, "Link selection in buffer-aided cooperative networks for green IoT," *IEEE Access*, vol. 8, pp. 30763–30771, 2020.
[13] S. Duan et al., "Distributed artificial intelligence empowered by end-edge-cloud computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 591–624, 1st Quart., 2023.
[14] A. Dubey and A. Pentland, "Cooperative multi-agent bandits with heavy tails," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2730–2739.
[15] H. El-Sayed et al., "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.
[16] H. Esfandiari, A. Karbasi, A. Mehrabian, and V. Mirrokni, "Regret bounds for batched bandits," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 7340–7348.
[17] R. Féraud, R. Alami, and R. Laroche, "Decentralized exploration in multi-armed bandits," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1901–1909.

[18] Z. Gao, Y. Han, Z. Ren, and Z. Zhou, "Batched multi-armed bandits problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 503–513.
[19] A. Garivier, E. Kaufmann, and T. Lattimore, "On explore-then-commit strategies," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 784–792.
[20] S. Hasegawa et al., "Multi-armed-bandit based channel selection algorithm for massive heterogeneous Internet of Things networks," *Appl. Sci.*, vol. 12, no. 15, p. 7424, Jul. 2022.
[21] Md. F. Hossain, K. S. Munasinghe, and A. Jamalipour, "Distributed inter-BS cooperation aided energy efficient load balancing for cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5929–5939, Nov. 2013.
[22] T. Jin, J. Tang, P. Xu, K. Huang, X. Xiao, and Q. Gu, "Almost optimal anytime algorithm for batched multi-armed bandits," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5065–5073.
[23] N. Karpov and Q. Zhang, "Communication-efficient collaborative regret minimization in multi-armed bandits," in *Proc. 38th AAAI Conf. Artif. Intell.*, 2024, pp. 13076–13084.
[24] R. K. Kolla, K. Jagannathan, and A. Gopalan, "Collaborative learning of stochastic bandits over a social network," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1782–1795, Aug. 2018.
[25] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, Mar. 1985.
[26] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision-making in multiarmed bandits: Frequentist and Bayesian algorithms," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 167–172.
[27] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
[28] X. Li, R. Zhang, and L. Hanzo, "Cooperative load balancing in hybrid visible light communications and WiFi," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1319–1329, Apr. 2015.
[29] J. Liu, H. Qiu, L. Yang, and M. Xu, "Distributed multi-agent bandits over Erdos–Rényi random networks," *arXiv:2510.22811*.
[30] U. Madhushani and N. Leonard, "When to call your neighbor? Strategic communication in cooperative stochastic bandits," 2021, *arXiv:2110.04396*.
[31] D. Martínez-Rubio, V. Kanade, and P. Rebeschini, "Decentralized cooperative stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2018, pp. 4529–4540.
[32] É. Boursier, E. Kaufmann, A. Mehrabian, and V. Perchet, "A practical algorithm for multiplayer bandits when arm means vary among players," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1211–1221.
[33] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg, "Batched bandit problems," *Ann. Statist.*, pp. 660–681, 2016.
[34] A. Raschella, F. Bouhafs, M. Seyedebrahimi, M. Mackay, and Q. Shi, "Quality of service oriented access point selection framework for large Wi-Fi networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 441–455, Jun. 2017.
[35] C. Shi and C. Shen, "Federated multi-armed bandits," in *Proc. 35th AAAI Conf. Artif. Intell. (AAAI)*, vol. 35, 2021, pp. 9603–9611.
[36] C. Shi, C. Shen, and J. Yang, "Federated multi-armed bandits with personalization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2917–2925.
[37] C. Shi, W. Xiong, C. Shen, and J. Yang, "Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 22392–22404.
[38] E. Soria, F. Schiano, and D. Floreano, "Distributed predictive drone swarms in cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 73–80, Jan. 2022.
[39] B. Szörényi, R. Busa-Fekete, I. Hegedus, R. Ormándi, M. Jelasity, and B. Kégl, "Gossip-based distributed stochastic bandit algorithms," in *Proc. Int. Conf. Mach. Learn.*, vol. 28, 2013, pp. 19–27.
[40] Y. Wan, T. Wei, M. Song, L. Zhang, and Z. Lijun, "Optimal and efficient algorithms for decentralized online convex optimization," *J. Mach. Learn. Res.*, vol. 26, no. 135, pp. 1–43, 2024.
[41] P.-A. Wang, A. Proutiere, K. Ariu, Y. Jedra, and A. Russo, "Optimal algorithms for multiplayer multi-armed bandits," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, vol. 108, 2020, pp. 4120–4129.
[42] X. Wang et al., "Asynchronous multi-agent bandits: Fully distributed vs. leader-coordinated algorithms," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 53, no. 1, pp. 1–39, Jun. 2025.
[43] X. Wang et al., "Achieve near-optimal individual regret & low communications in multi-agent bandits," in *Proc. Int. Conf. Learn. Represent.*, 2023.
[44] Y. Wang, J. Hu, X. Chen, and L. Wang, "Distributed bandit learning: Near-optimal regret with efficient communication," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–18.
[45] X. Wu, M. Safari, and H. Haas, "Access point selection for hybrid li-fi and Wi-Fi networks," *IEEE Trans. Commun.*, vol. 65, no. 12, pp. 5375–5385, Dec. 2017.
[46] L. Yang, Y. J. Chen, M. H. Hajiemaili, J. C. S. Lui, and D. Towsley, "Distributed bandits with heterogeneous agents," in *Proc. IEEE Conf. Comput. Commun.*, May 2022, pp. 200–209.
[47] L. Yang, Y.-Z. J. Chen, S. Pasteris, M. Hajiesmaili, J. C. S. Lui, and D. Towsley, "Cooperative stochastic bandits with asynchronous agents and constrained feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8885–8897.
[48] H. Zhang et al., "Federated multi-armed bandits with efficient bit-level communications," in *Proc. 39th Annu. Conf. Neural Inf. Process. Syst.*, 2025.
[49] J. Zhu, E. Mulle, C. S. Smith, A. Koppel, and J. Liu, "Decentralized upper confidence bound algorithms for homogeneous multi-agent multi-armed bandits," 2021, *arXiv:2111.10933*.

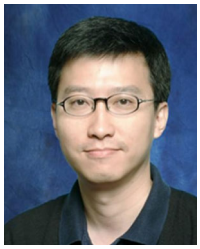
Lin Yang received the B.Eng. and M.Sc. degrees from the University of Science and Technology of China in 2012 and 2015, respectively, and the Ph.D. degree from The Chinese University of Hong Kong in 2018, (fortunately) supervised by Prof. Wing Shing Wong. He was a Post-Doctoral Research Associate with the Department of Computer Science, UMass Amherst, working with Prof. Don Towsley and Mohammad Hajiesmaili in 2022. He is currently an Assistant Professor with the National Key Laboratory for Novel Software Technology and the School of Intelligence Science and Technology, Nanjing University, Suzhou Campus. His research interests include learning theory, online optimization, and model and analysis for computing systems.

Xuchuang Wang (Member, IEEE) received the B.Eng. degree from the School of Electronic and Information Engineering, Xi'an Jiaotong University, in 2019, and the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, under the supervision of John C. S. Lui. He is currently a Post-Doctoral Researcher with the College of Information and Computer Sciences, University of Massachusetts Amherst, under the supervision of Don Towsley and Mohammad Hajiesmaili. His research interests include online learning and sequential decision making.

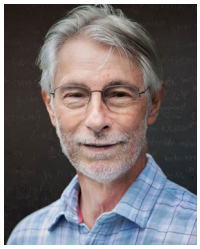
Haoxu Chen (Member, IEEE) received the bachelor's degree in artificial intelligence from Nanjing University in 2019. He is currently pursuing the master's degree with the National Key Laboratory for Novel Software Technology and the School of Intelligence Science and Technology, Nanjing University, Suzhou Campus, under the supervision of Assistant Professor Yang Lin. His research interests include online learning and sequential decision making.

Mohammad H. Hajiesmaili (Member, IEEE) received the Ph.D. degree from the University of Tehran. He was a Post-Doctoral Fellow with The Chinese University of Hong Kong and Johns Hopkins University. He is currently an Associate Professor with the Manning College of Information and Computer Sciences, University of Massachusetts Amherst. He directs the Sustainability, Optimization, Learning, and Algorithms Research (SOLAR) Lab, where the research focuses on developing optimization and machine learning tools to facilitate the decarbonization of digital and societal infrastructure. His awards and honors include the lead for theoretical and AI foundations of an NSF Expedition in computing on computational decarbonization, an NSF CAREER Award, and other awards from NSF, Amazon, Google, VMWare, and Adobe.

Lijun Zhang (Senior Member, IEEE) received the B.E. and Ph.D. degrees in software engineering and computer science from Zhejiang University, Hangzhou, China, in 2007 and 2012, respectively. He was a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA. He is currently a Professor with the National Key Laboratory for Novel Software Technology and the School of Artificial Intelligence (AI), Nanjing University, Nanjing, China. His research interests include machine learning and optimization.



John C. S. Lui (Fellow, IEEE) received the Ph.D. degree in computer science from UCLA. After his graduation, he joined the IBM Laboratory and participated in research and development projects on file systems and parallel I/O architectures. He later joined the CSE Department, The Chinese University of Hong Kong (CUHK). He is currently the Choh-Ming Li Chair Professor with the Department of Computer Science and Engineering (CSE), CUHK. His current research interests include online learning algorithms and applications (e.g., multi-armed bandits and reinforcement learning), machine learning on network sciences and networking systems, large-scale data analytics, network/system security, network economics, large-scale storage systems, and performance evaluation theory. He is an elected member of the IFIP WG 7.3, a Fellow of ACM, a Senior Research Fellow of the Croucher Foundation, and a Fellow of Hong Kong Academy of Engineering Sciences (HKAES). He has served at the IEEE Fellow Review Committees.



Don Towsley (Life Fellow, IEEE) received the Ph.D. degree in computer science from The University of Texas in 1975. He is currently a Distinguished Professor with the Manning College of Information and Computer Sciences, University of Massachusetts Amherst. His research interests include performance modeling and analysis, as well as quantum networking. He is a Fellow of ACM. He has received several achievement awards, including the 2007 IEEE Koji Kobayashi Award and the 2011 INFOCOM Achievement Award.