On-Demand Communication for Asynchronous Multi-Agent Bandits

Yu-Zhen Janice Chen
University of Massachusetts Amherst
yuzhenchen@cs.umass.edu

Lin Yang
Nanjing University
linyang@nju.edu.cn

Xuchuang Wang, Xutong Liu
The Chinese University of Hong Kong
{xcwang, liuxt}@cse.cuhk.edu.hk

Mohammad Hajiesmaili
University of Massachusetts Amherst
hajiesmaili@cs.umass.edu

John C. S. Lui
The Chinese University of Hong Kong
cslui@cse.cuhk.edu.hk

Don Towsley
University of Massachusetts Amherst
towsley@cs.umass.edu

Abstract

This paper studies a cooperative multi-agent multi-armed stochastic bandit problem where agents operate *asynchronously* – agent pull times and rates are unknown, irregular, and heterogeneous – and face the same instance of a K-armed bandit problem. Agents can share reward information to speed up the learning process at additional communication costs. We propose `ODC`, an on-demand communication protocol that tailors the communication of each pair of agents based on their empirical pull times. `ODC` is efficient when the pull times of agents are highly heterogeneous, and its communication complexity depends on the empirical pull times of agents. `ODC` is a generic protocol that can be integrated into most cooperative bandit algorithms without degrading their performance. We then incorporate `ODC` into the natural extensions of `UCB` and `AAE` algorithms and propose two communication-efficient cooperative algorithms. Our analysis shows that both algorithms are near-optimal in regret.

1 INTRODUCTION

Asynchronous multi-agent multi-armed bandit (MAMAB) settings arise naturally in several applications. For instance, in online advertising with multiple heterogeneous servers, server processing capabilities and speeds are often different. Furthermore, the times that servers receive recommendation requests are often unknown and irregular. Another example is clinical trials with multiple labs in collaboration, where trial times depend on client visit times, which vary from lab to lab. In other large-scale distributed learning scenarios, such as IoT devices cooperating to learn an underlying environment, agents can be asynchronous in nature due to task arrangements or hardware limits.

This paper studies a MAMAB setting where agents with *unknown asynchronous* decision times cooperate to improve their learning performance. Concretely, we consider a system where a group of M agents, $\mathcal{A} = \{1, ..., M\}$, cooperate to solve the same instance of a K-armed bandit problem. An agent repeatedly chooses an arm from the arm set to pull and receives a stochastic reward from it. Agents have different numbers of decision rounds (pull times) at arbitrary unknown times. Each agent aims to minimize its individual *regret* – the cumulative difference between the reward received by the agent and the expected reward of the best arm in the arm set. Agents cooperate by sharing

16th European Workshop on Reinforcement Learning (EWRL 2023).

reward information with each other, and their goal is to together minimize the *group regret* – the total amount of individual regret among the M agents. Cooperation among agents, however, comes with an additional *communication* cost, which can be expensive for some applications when agents are geographically dispersed or have limited power/bandwidth resources for communication.

Prior studies (Yang et al., 2021, 2022) have shown that it is possible to achieve near-optimal group regret by immediately broadcasting rewards. In an asynchronous setting where agents have different pull speeds, immediate broadcasts can incur unnecessary communication costs. With immediate broadcast communication, an agent can receive multiple reward-sharing messages from another agent between its two decision rounds; these messages could have been accumulated (buffered) by that agent and sent all at once, incurring lower communication overhead. Hence, for a group of asynchronous agents, tailoring the message exchange protocol between each pair of agents can yield better communication efficiency.

This paper aims to reduce communication costs over that of the immediate broadcast communication protocol (`IBC`) while achieving the same order of regret. The lack of synchronization between agents, however, poses a challenge on determining the timing of communication. Specifically, agents are uncertain of other agents' learning progress at any time due to the arbitrary asynchronicity of agent pull times and hence need to trade-off communication costs to learn this information for better cooperation. One might apply the idea of coordinated cooperative learning, e.g., the leader-follower framework, which has proven to be efficient in prior studies (Kolla et al., 2018; Dubey et al., 2020; Wang et al., 2020) of the synchronous MAMAB problem. However, unknown and irregular agent pull speeds hinder the application of coordinated cooperative learning. This can lead to a scenario where agents chosen to be in charge of exploration, leaders, are slow (have small pull rates), and agents chosen to perform exploitation are fast (have large pull rates), which can incur high regret. Another alternative is customizing spontaneous communication between agents, where each agent deliberately chooses its communication frequency to other agents according to their pull rates. However, efficient implementation of customized spontaneous communication is not possible since agents do not have prior knowledge of the pull times of others.

Contributions. This paper develops *On-Demand* Communication (`ODC`), an efficient protocol for the asynchronous cooperative MAMAB model, where unique technical challenges are introduced by the unknown, irregular, and different decision times of agents. By the design of `ODC`, we address the challenge of reducing the number of communications among asynchronous agents. Specifically, `ODC` reduces the number of communications by tailoring the times communications occur between each pair of agents based on their empirical pull times. More importantly, `ODC` is generic and can be used with most cooperative bandit algorithms. We propose two decentralized MAMAB algorithms, `UCB-ODC` and `AAE-ODC`, which combine `ODC` with natural extensions of `UCB` and `AAE` algorithms respectively. Our analysis shows that both `UCB-ODC` and `AAE-ODC` achieve near-optimal group regret upper bounds of $O(\sum_{i:\Delta_i>0} \log(N)/\Delta_i)$, where $N \equiv \sum_{j\in\mathcal{A}} N_j$ is the total number of decision rounds of all agents, N_j is the total number of decision rounds of agent j, and Δ_i is the suboptimality gap of arm i.

Under `ODC`, communication complexity, i.e., the total number of messages sent among agents, depends on the specific decision times of agents. We show that the communication complexity of `ODC` is $O(\sum_{j,j'\in\mathcal{A}} \min\{N_j, N_{j'}\})$, which depends on the agents with the fewest decision rounds. This communication complexity is much smaller than that of the immediate broadcast communication protocol (`IBC`), $O(MN)$, when agent pull times are highly heterogeneous. Moreover, following prior ideas on the synchronous MAMAB setting, one has the option to tune message transmission rates under `ODC` by allowing messages to vary in size to further reduce the communication complexity. For example, if the number of observations in a message is doubled after each communication, the communication complexity of `ODC` becomes $O(\sum_{j,j'\in\mathcal{A}} \min\{\log N_j, \log N_{j'}\})$. In this way, our asynchronous policy can recover the state-of-the-art logarithmic communication complexity when applied to the synchronous MAMAB setting.

Our experimental results verify our theoretical observations and demonstrate that `ODC` is especially advantageous when agent pull speeds are highly diversified, and when there exist many slow agents.

Prior Work. We review the most relevant work here and refer to Appendix A for extended literature review. The most relevant work considers asynchronous bandit agents cooperating in a fully decentralized manner (Yang et al., 2021, 2022; Sankararaman et al., 2019; Féraud et al., 2019). The

model in Yang et al. (2021, 2022) assumes each agent periodically make decisions at different *known* frequencies. Our paper assumes that pulling times are unknown and irregular. Sankararaman et al. (2019) study a gossip protocol, i.e., an agent can only communicate with one other agent at each time. Last, Féraud et al. (2019) studies the scenario where the goal is to identify the best arm instead of minimizing regret. More broadly there is extensive prior work on MAMAB with synchronous agents either in a fully decentralized setting, e.g., (Szorenyi et al., 2013; Chawla et al., 2020; Landgren et al., 2016; Buccapatnam et al., 2015; Martínez-Rubio et al., 2019; Madhushani et al., 2021; Cesa-Bianchi et al., 2016), or using coordinated cooperative approach (Shi et al., 2021a; Wang et al., 2019, 2020; Bar-On and Mansour, 2019; Chakraborty et al., 2017; Dubey et al., 2020; Kolla et al., 2018). In the synchronous MAMAB setting, the batch approach (a.k.a., doubling epoch, phase, buffer) (Perchet et al., 2016; Gao et al., 2019) has been used to achieve logarithmic communication complexity, e.g, by Agarwal et al. (2021); Shi et al. (2021b); Boursier and Perchet (2019). There are also works on asynchronous multi-agent learning in related fields such as federated linear bandit (Li and Wang, 2022; He et al., 2022) and online convex optimization with full information or semi-bandit feedback (Cesa-Bianchi et al., 2020; Jiang et al., 2021; Joulani et al., 2019; Bedi et al., 2019; Della Vecchia and Cesari, 2021).

2 ASYNCHRONOUS MULTI-AGENT BANDITS

We study an asynchronous version of the cooperative multi-agent multi-armed bandit (MAMAB) problem with a set $\mathcal{A} = \{1, ..., M\}$ of M independent agents and a set $\mathcal{K} = \{1, ..., K\}$ of K arms. Each arm $i \in \mathcal{K}$ is associated with a mutually independent sequence of i.i.d. rewards, taken to be Bernoulli with mean $0 \leq \mu(i) \leq 1$. Let $i^* = \arg\max_{i \in \mathcal{K}} \mu(i)$ denote the optimal arm. Define the suboptimality gap of arm i as $\Delta_i \equiv \mu(i^*) - \mu(i)$ and let $\Delta \equiv \min_{i \in \mathcal{K} \setminus \{i^*\}} \Delta_i$ denote the smallest suboptimality gap in the arm set.

Agents operate *asynchronously*. Let N_j be the total number of decisions made by agent j; agent $j \in \mathcal{A}$ pulls arms at time slots $t_1^j, t_2^j, ...,$ and $t_{N_j}^j$, where both N_j and the time slots are not known by any agent including agent j. We make no assumptions about when agents pull arms and the total number of pulls they make. One agent may pull many arms within an arbitrary interval, while another agent might not pull any arm. Furthermore, agents are allowed to join, leave, and re-join the system at arbitrary times. Let $T \equiv \max_{j \in \mathcal{A}} t_{N_j}^j$ denote the learning horizon of the entire group of agents and $N \equiv \sum_{j \in \mathcal{A}} N_j$ denote the total number of decisions among all agents over the time horizon.

We consider the problem where there are no *collisions*; i.e., agent always receives a Bernoulli reward with mean $\mu(i)$ from arm $i \in \mathcal{K}$, irrespective of the actions of other agents. Each agent $j \in \mathcal{A}$ pulls one arm at time $t \in \{t_1^j, t_2^j, ..., t_{N_j}^j\}$ with the goal of minimizing its cumulative regret. The expected cumulative regret of a single agent j is defined as $\mathbb{E}[R_{N_j}^j] = \mu(i^*)N_j - \mathbb{E}\big[\sum_{t \in \{t_1^j, t_2^j, ..., t_{N_j}^j\}} x_t(I_t^j)\big]$, where $I_t^j \in \mathcal{K}$ is the arm pulled by agent j at time t, reward $x_t(I_t^j)$ is taken from Bernoulli distribution with value 0 or 1, and the expectation is taken over the randomness of agent's decisions and arm rewards. We denote the number of times agent j pulls arm i by time t as $n_j^t(i)$, and the number of decisions agent j makes by time t as n_j^t. We assume that every agent can reliably communicate with every other agent to share their observations. Let $\hat{n}_j^t(i)$ denote the empirical number of observations of arm i that agent j has at time t, either by pulling the arm, or obtained from other agents, and let $\hat{n}_j^t$ denote the total empirical number of observations agent j has at time t. The objective of the cooperative MAMAB problem is to minimize expected *group regret*, defined as $\mathbb{E}[R] = \sum_{j \in \mathcal{A}} \mathbb{E}[R_{N_j}^j]$, while maintaining low communication overhead. Let C denote the total number of messages sent by agents in horizon T, as in Wang et al. (2020); Yang et al. (2021, 2022). We precisely define the information included in a message in Definition 1 in §3.1.

3 ALGORITHM DESIGN

In §3.1, we first elaborate the design and provide intuition behind the On-Demand Communication (`ODC`) protocol. Then, we incorporate `ODC` into bandit algorithms and propose two communication-efficient cooperative bandit algorithms: `UCB-ODC` (§3.2) and `AAE-ODC` (§3.3).

Algorithm 1 ODC for Agent j

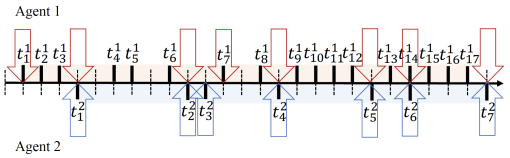
Initialization: exchange demands $E^{j\to j'} \leftarrow$ `True`, $\forall j' \in \mathcal{A}$, buffers $b_n^{j\to j'}(i) \leftarrow 0, b_\mu^{j\to j'}(i) \leftarrow 0, \forall j' \in \mathcal{A}, \forall i \in \mathcal{K}$, num. of communications $c^{j\to j'} \leftarrow 1, \forall j' \in \mathcal{A}$, buffer thresholds $f(c^{j\to j'}) \leftarrow f(1), \forall j' \in \mathcal{A}$
for $t = 1...T$ **do**
  if t is a decision time slot of agent j, i.e., $t \in \{t_1^j, ..., t_{N_j}^j\}$ **then** ▷ `decision making`
    Run an underlying bandit algorithm: pull arm I_t^j according to e.g., `UCB`, or `AAE`, and receive instantaneous reward $x_t(I_t^j)$
    Update para. of underlying bandit algorithm, e.g., empirical mean rewards, num. of observations
    for each agent $j' \in \mathcal{A}$ **do**
      Update the buffer for agent j': $b_n^{j\to j'}(I_t^j) \leftarrow b_n^{j\to j'}(I_t^j) + 1$, $b_\mu^{j\to j'}(I_t^j) \leftarrow b_\mu^{j\to j'}(I_t^j) + x_t(I_t^j)$
      if $E^{j\to j'}$ is `True` and $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then** ▷ `information sharing`
        Share buffered info. with j', i.e., send a message as defined in Def. 1, Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$
        Set exchange demand $E^{j\to j'} \leftarrow$ `False` and renew the buffer for agent j'
        Update buffer threshold $f(c^{j\to j'})$, e.g., $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
      end if
    end for
  end if
  for each new message received from any agent $j' \in \mathcal{A}$ **do** ▷ `message processing`
    Update para. of underlying bandit algorithm, e.g., empirical mean rewards, num. of observations
    if agent j has buffered $f(c^{j\to j'})$ observations for j', i.e., $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
      Share info. by sending a msg. as defined in Def. 1 to j'
      Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$, renew buffer for j'
      Update buffer threshold $f(c^{j\to j'})$, e.g., $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
    else
      Set exchange demand $E^{j\to j'} \leftarrow$ `True`
    end if
  end for
end for

3.1 ODC: On-Demand Communication Protocol

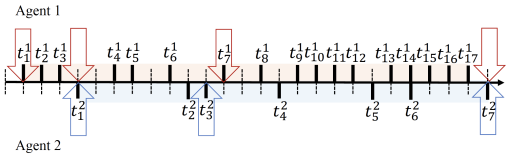
We present the On-Demand Communication (`ODC`) protocol (`ODC`) summarized in Algorithm 1. The core idea of `ODC` is to leverage the fact that agents pull arms at different rates to reduce communication complexity while achieving the same order of regret achieved by algorithms that immediately share rewards. Consider a scenario with a *fast* and a *slow* agent. By *fast*, we mean the agent pulls many arms while a *slow* agent pulls very few arms during the same time horizon. If agents immediately share their observations, the fast agent incurs a large communication overhead by sending multiple messages between two consecutive decision rounds of the slow agent. In fact, the fast agent can reduce communication overhead while achieving the same regret if it aggregates the instantaneous rewards during the slow agent's non-decision period and sends the information all at once prior to the slow agent's next decision round. Hence, different agent pull rates motivate a new communication protocol that reduces communication complexity by scheduling communication times for each pair of agents according to their pull rates.

Given the above motivation, one idea is to allow each agent to receive other observations at a rate proportional to its pull rate. In our asynchronous MAMAB model, it is challenging to tailor communication timings because agent pull times are irregular and unknown. A straightforward way to achieve this is to allow agents to request observations from other agents prior to pulling arms. However, requests introduce extra communication overhead, i.e., fast agents may make too many requests to slow agents before they obtain new reward information to share.

The idea implemented in `ODC` is to treat each observation sharing message as an exchange demand. Specifically, we let each agent j maintain a set of binary valued exchange demand variables $(E^{j\to 1}, E^{j\to 2}, ..., E^{j\to M})$. Once agent j receives a message from agent j', it sets exchange demand $E^{j\to j'}$ to `True`. Then, when agent j acquires new information it responds back to agent j' and resets $E^{j\to j'}$ to `False`. If agent j acquires new information while $E^{j\to j'}$ is `False`, agent j *buffers* it for agent j' while waiting for the exchange demand to be set to `True`. Specifically, the buffer maintained for agent j' records the following information for each arm i: (1) the number of observations of i that agent j acquires by pulling it since the last time agent j sent a message to



(a) ODC with static buffer thresholds, i.e., buffer thresholds are updated according to $f(1) \leftarrow 1$, $f(c) \leftarrow f(c-1)$



(b) ODC with doubling buffer thresholds, i.e., thresholds are updated according to $f(1) \leftarrow 1$, $f(c) \leftarrow 2f(c-1)$

Figure 1: Examples of two agents with arbitrary decision times

agent j', denoted as $b_n^{j \to j'}(i)$, (2) the cumulative reward over the observations of arm i that agent j acquires from pulling it since the last round agent j sends a message to j', denoted as $b_\mu^{j \to j'}(i)$. After agent j sends the buffered information in a message to agent j', it renews the buffer by resetting $b_n^{j \to j'}(i), b_\mu^{j \to j'}(i), \forall i \in \mathcal{K}$, to zero.

Definition 1 *A message sent from agent j to j' is a set of K tuples: $\{(b_n^{j \to j'}(i), b_\mu^{j \to j'}(i)), \forall i \in \mathcal{K}\}$.*[1]

ODC also implements thresholds on the buffer sizes[2], i.e., a buffer must contain at least as many observations as the buffer threshold, that together with exchange demands determine whether an agent should send a message to another agent. Specifically, each agent j maintains a set of positive integer valued variables $(c^{j \to 1}, ..., c^{j \to M})$ denoting the number of communications from agent j to other agents. The buffer threshold, $f(c^{j \to j'})$, is a positive and monotonically increasing function of the number of communications $c^{j \to j'}$. After agent j sends a message to agent j' when the communication counter is $c^{j \to j'}$, the communication counter is incremented by one and the buffer threshold for the next communication is $f(c^{j \to j'} + 1)$. Possible candidates for the buffer threshold function include $f(c) = a, c = 1, 2, \ldots$, where a is a positive integer, and $f(c) = a^{c-1}, c = 1, 2, \ldots$, where $a > 1$ is a positive integer. The first example produces a constant buffer threshold and the second allows buffer threshold to increase exponentially each time a message is sent. Under ODC, an agent j sends a message to agent j' if $E^{j \to j'}$ is `True` and agent j has buffered at least $f(c^{j \to j'})$ observations for agent j'.

In Figure 1, we provide simple examples of two agents with arbitrary decision time. To illustrate how ODC (Algorithm 1) works, we describe the communication schedule for the example in Figure 1b. Agent 1 first sends a message to agent 2 at time t_1^1, sets exchange demand $E^{1 \to 2}$ to `False`, sets $c^{1 \to 2}$ to 2, and updates $f(c^{1 \to 2})$ to 2. At t_2^1 and t_3^1, agent 1 pulls arms and buffers the obtained observations because $E^{1 \to 2}$ is `False`. At t_1^2, agent 1 receives a message from agent 2, replies with a message containing the observations obtained at t_2^1, t_3^1, renews the buffer, sets $c^{1 \to 2}$ to 3, and updates $f(c^{1 \to 2})$ to 4. At t_4^1, t_5^1, t_6^1, agent 1 pulls arms and buffers the obtained observations. At t_3^2, agent 1 receives a message from agent 2; instead of replying with a message, agent 1 sets the exchange demand $E^{1 \to 2}$ to `True` at this time as it only buffered three observations while the buffer threshold $f(c^{1 \to 2})$ is 4. At t_7^1, agent 1 obtains an observation and satisfies the buffer threshold; thus, it sends a message to agent 2, renews the buffer, sets $E^{1 \to 2}$ to `False`, sets $c^{1 \to 2}$ to 4, and updates $f(c^{1 \to 2})$ to 8.

Last, we note that ODC can handle agent arrivals and departures. Agent notifies the others when it departs or goes offline. When agent j receives a departure notice from agent j', it sets exchange demand $E^{j \to j'}$ to `False`, so it will buffer information for j'. When an agent (re)joins the system, it notifies all other agents. When agent j receives a join notice from agent j', it (re)initializes exchange demand $E^{j \to j'}$ to `True` so that the observations buffered during agent j''s leaving can be sent to it to (re)start cooperation.

3.2 UCB-ODC: Cooperative UCB with ODC

In this section, we present UCB-ODC, a fully decentralized cooperative MAMAB algorithm that samples according to a natural extension of the Upper Confidence Bound (UCB) algorithm and uses

[1] If an agent buffers n observations, the number of observations and cumulative reward of each arm take value in $\{0, ..., n\}$ and require $\log(n+1)$ bits.

[2] One may apply the batch/epoch setting techniques in synchronous MAMAB literature to the buffer threshold setting here.

ODC for communications. Under UCB-ODC, agent j computes an empirical mean reward, $\hat{\mu}(i, \hat{n}_j^t(i))$, over $\hat{n}_j^t(i)$ observations of agent j for each arm $i \in \mathcal{K}$. Note that the value of $\hat{n}_j^t(i)$ not only consists of instantaneous rewards agent j received from pulling arm i, but it also includes information agent j received from other agents. Under UCB-ODC, agent j also maintains a confidence interval for arm i centered on its empirical mean value, $\hat{\mu}(i, \hat{n}_j^t(i))$, with width defined as

$$\texttt{CI}_j^t(i) \equiv \sqrt{\alpha \log(1/\delta_j^t)/(2\hat{n}_j^t(i))}, \tag{1}$$

where α and δ_j^t are algorithm parameters. With probability at least $1 - (\delta_j^t)^\alpha$, the true reward mean, $\mu(i)$, lies in its confidence interval, i.e., $\mu(i) \in [\hat{\mu}(i, \hat{n}_j^t(i)) - \texttt{CI}_j^t(i), \hat{\mu}(i, \hat{n}_j^t(i)) + \texttt{CI}_j^t(i)]$. Further discussion and analysis of the confidence interval can be found in Bubeck and Cesa-Bianchi (2012).

Under UCB-ODC, agent j selects the arm with the largest upper confidence bound at each decision round, i.e.,

$$I_t^j \equiv \arg\max_{i \in \mathcal{K}} \hat{\mu}(i, \hat{n}_j^t(i)) + \texttt{CI}_j^t(i), \ t \in \{t_1^j, ...t_{N_j}^j\}.$$

Upon receiving an instantaneous reward for the selected arm I_t^j, UCB-ODC updates the reward mean estimate and confidence interval of I_t^j. In the meantime, agent j follows ODC checking exchange demands, buffer thresholds, and accordingly buffering the information or sending messages. The pseudocode of UCB-ODC is in Appendix E.

3.3 AAE-ODC: Cooperative AAE with ODC

We propose AAE-ODC, which combines ODC with a natural extension of the Active Arm Elimination (AAE) algorithm (Even-Dar et al., 2006). Agents executing AAE-ODC together maintain a dynamic *candidate set* to keep track of arms likely to be the optimal arm, where the candidate set is updated using the confidence intervals as defined in (1). Specifically, the candidate set initially contains all arms. When agents observe or receive rewards, they recompute the confidence intervals of arms; if an arm's confidence interval completely falls below that of any other arm, it is removed from the candidate set as it is unlikely to be the optimal arm. Formally, arm i is removed from the candidate set at time t if for any agent j:

$$\exists i' \in \mathcal{K} \text{ s.t. } \hat{\mu}_j^t(i) + \texttt{CI}_j^t(i) < \hat{\mu}_j^t(i') - \texttt{CI}_j^t(i').$$

Once an arm is eliminated by an agent, the agent broadcasts the index of the eliminated arm, so that other agents can keep updating the candidate set. At each decision round, agent j pulls from the candidate set the arm that agent j has fewest observations. Once the candidate set size reduces to one, agents have completed the exploration task having identified optimal arm with high probability, and they do not need information from one another anymore. Hence, agents under AAE-ODC stop their communication once the candidate set size shrinks to one. The cooperation policy of AAE-ODC follows the ODC protocol and is summarized in the pseudocode of AAE-ODC in Appendix F.

4 ANALYSIS OF REGRET AND COMMUNICATION COMPLEXITY

When agents are asynchronous and pull arms in arbitrary time slots, the performance of a cooperative bandit algorithm depends on how much agents can cooperate with each other. A unique technical challenge in the regret analysis of UCB-ODC and AAE-ODC is bounding the additional number of times agents pull suboptimal arms due to delayed observation sharing while waiting for exchange demands between asynchronous agents or waiting for buffer thresholds to be satisfied. To facilitate the regret analysis, we let τ_i denote the time slot such that $\mathscr{N}(i) + M \geq \sum_{j \in \mathcal{A}} n_j^{\tau_i}(i) > \mathscr{N}(i) \geq \sum_{j \in \mathcal{A}} n_j^{\tau_i - 1}(i)$, for each suboptimal arm $i \in \mathcal{K} \backslash \{i^*\}$, where for UCB-ODC, $\mathscr{N}(i) = (2\alpha \log N)/\Delta_i^2$, and for AAE-ODC, $\mathscr{N}(i) = (16\alpha \log N)/\Delta_i^2$.

4.1 Regret Results

Theorem 1 (Expected Group Regret under ODC) *With $\alpha \geq 3$ and buffer thresholds being updated according to a positive and monotonically increasing function f, we have:*

(a) *with* $\delta_j^t = 1/N$*, the expected group regret of* UCB-ODC *satisfies*

$$\mathbb{E}[R] \leq 3KM + \sum_{i \in \mathcal{K}: \Delta_i > 0} \Big(\frac{2\alpha \log N}{\Delta_i} + \sum_{j \in \mathcal{A}} F_i^j \Delta_i \Big), \tag{2}$$

where F_i^j *is a non-negative variable defined as*[3] $F_i^j = \min \Big\{ \Big(\sum_{j' \in \mathcal{A} \setminus \{j\}} f(c_{\tau_i}^{j' \to j}) \Big), \frac{2\alpha \log N}{\Delta_i^2} \Big\}$;
(b) *with* $\delta_j^t = 1/N^2$*, the expected group regret of* AAE-ODC *satisfies*

$$\mathbb{E}[R] \leq 3KM + \sum_{i \in \mathcal{K}: \Delta_i > 0} \Big(\frac{16\alpha \log N}{\Delta_i} + \sum_{j \in \mathcal{A}} G_i^j \Delta_i \Big), \tag{3}$$

where G_i^j *is a non-negative variable defined as* $G_i^j = \min \Big\{ \Big(\sum_{j' \in \mathcal{A} \setminus \{j\}} f(c_{\tau_i}^{j' \to j}) \Big), \frac{16\alpha \log N}{\Delta_i^2} \Big\}$.

The proofs of Theorem 1(a) and 1(b) deal with each suboptimal arm i, and upper bounds the extra number of times each agent pulls arm i after time τ_i. A formal proof is given in Appendix B. In the following, we highlight important properties of the regret results under ODC.

Remark 1 *(Regret characterization by total number of decision rounds.) We observe that the expected regret in Theorem 1 is characterized by N, the total number of decision rounds among all agents in the learning horizon. This is in contrast to the synchronous agent setting where regret is usually presented as a function of the learning horizon T and the total number of agents M. When agent pull rates significantly differ from each other, Theorem 1 provides a much tighter regret bound than those derived for synchronous settings, as N can be much smaller than $M \times T$.*

Remark 2 *(Regret optimality.) When buffer thresholds are set to a small constant a, i.e., $f(c^{j \to j'}) = a, c^{j \to j'} = 1, 2, \ldots, \forall j, j'$, then $F_i^j, \forall i, j$ (resp. $G_i^j, \forall i, j$) is bounded by constant Ma, and* UCB-ODC *(resp.* AAE-ODC*) achieves a provable optimal regret upper bound. To show this, we derive the following lower bound on group regret by adopting the proof techniques for the asymptotic lower bound for single-agent bandits, e.g., (Bubeck and Cesa-Bianchi, 2012, Theorem 2.2):*

$$\mathbb{E}[R] = \Omega\Big(\sum\nolimits_{i \in \mathcal{K}: \Delta_i > 0} \frac{\log N}{\Delta_i} \Big). \tag{4}$$

The proof of (4) is given in Appendix C. Then, since F_i^j (resp. G_i^j) is a constant $\forall i, j$, one observes that the regret of UCB-ODC *in* (2) *(resp. of* AAE-ODC *in* (3)*) is near-optimal compared to the lower bound (4), up to two constant terms, i.e., $3KM$ and $\sum_{i,j} F_i^j$ (resp. $\sum_{i,j} G_i^j$).*

Remark 3 *(Impact of buffer thresholds.) The setting of buffer thresholds influences the trade-off between communication complexity and group regret. Remark 2 shows that* UCB-ODC *and* AAE-ODC *have near-optimal regrets if buffer thresholds are set to be small (compared to $\log(N)/\Delta^2$). If buffer thresholds are simply set to be always large, one can reduce the communication complexity while incurring higher regret. Depending on specific scenarios, e.g., as in Remark 4, buffer thresholds can be wisely set to achieve low communication while not degrading the regret much.*

Remark 4 *(Performance in synchronous setting.) When applied to a MAMAB setting with synchronous agents where every agent makes a decision at every time slot, our asynchronous algorithm* AAE-ODC *recovers a near-optimal regret $O(\sum_{i \in \mathcal{K}: \Delta_i > 0} \log(N)/\Delta_i)$ with logarithmic communication complexity by setting a doubling buffer threshold whose size is proportional to the number of arms remaining in the candidate set, $\mathcal{C}$, i.e., $f(c) = |\mathcal{C}| 2^{c-1}$, $c = 1, 2, \ldots$. We show this in Appendix B.3 and discuss the recovery of logarithmic communication complexity in Remark 8.*

4.2 Communication Complexity

Remark 5 (Communication Complexity under IBC**)** *The communication complexity of MAMAB algorithms using immediate broadcasting communication (*IBC*) is $C = \sum_{j \in \mathcal{A}} \sum_{j' \in \mathcal{A} \setminus \{j\}} N_j$.*

[3] We drop t from notations $c_t^{j \to j'}$ and $E_t^{j \to j'}$ in algorithm presentations for brevity. The precise notations are used in analysis.

Table 1: Summary of Results (all regret bounds are problem-dependent and we omit the $1/\Delta$ factor)

	Pull Times	Buffer Thres.	Group Regret	Communication #
UCB-ODC	Async., Sync.	Constant	$O(K \log N)$	$O(\sum_{j,j' \in \mathcal{A}} \min\{N_j, N_{j'}\})$
AAE-ODC	Async., Sync.	Constant	$O(K \log N)$	$O(\sum_{j,j' \in \mathcal{A}} K \min\{\log N, N_j, N_{j'}\}/\Delta^2)$
AAE-ODC	Sync.	Doubling	$O(K \log N)$	$O(\sum_{j,j' \in \mathcal{A}} \log[K \min\{\log N, N_j, N_{j'}\}/\Delta^2])$

Theorem 2 (Communication Complexities under ODC) *When buffer thresholds are updated according to a positive and monotonically increasing function f, the communication complexities of* UCB-ODC *and* AAE-ODC *satisfy:*

$$C \leq \sum\nolimits_{j \in \mathcal{A}} \sum\nolimits_{j' \in \mathcal{A} \setminus \{j\}} \min\{C_j, C_{j'}\} + 1, \tag{5}$$

where C_j is the largest integer in set $\{1, ..., N_j\}$ such that
(a) *for* UCB-ODC

$$\Big(\sum_{c=1}^{C_j} f(c) \Big) \leq N_j;$$

(b) *for* AAE-ODC

$$\Big(\sum_{c=1}^{C_j} f(c) \Big) \leq \min \Big\{ 2K + \sum_{i \in \mathcal{K}} \frac{16\alpha \log N}{\max\{\Delta_i^2, \Delta^2\}}, N_j \Big\}.$$

Proofs of Theorem 2(a) and 2(b) are in Appendix D.

Corollary 1 *When buffer thresholds $f(c) = a, c = 1, 2, \ldots, a$ is a positive integer, we have:*
(a) *the communication complexity of* UCB-ODC *is* $O\Big(\sum\limits_{j,j' \in \mathcal{A}} \min \Big\{ \lfloor \frac{N_j}{a} \rfloor, \lfloor \frac{N_{j'}}{a} \rfloor \Big\} \Big)$;
(b) *the communication complexity of* AAE-ODC *is* $O\Big(\sum\limits_{j,j' \in \mathcal{A}} \min \Big\{ \Big\lfloor \frac{K \log N}{a\Delta^2} \Big\rfloor, \lfloor \frac{N_j}{a} \rfloor, \lfloor \frac{N_{j'}}{a} \rfloor \Big\} \Big)$.

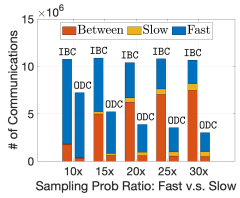
Corollary 2 *When buffer thresholds $f(c) = a^{c-1}, c = 1, 2, \ldots, a > 1$ is a positive integer, we have:*
(a) *the communication complexity of* UCB-ODC *is* $O\Big(\sum\limits_{j,j' \in \mathcal{A}} \min \big\{ \lfloor \log_a(N_j) \rfloor, \lfloor \log_a(N_{j'}) \rfloor \big\} \Big)$;
(b) *the communication complexity of* AAE-ODC *is* $O\Big(\sum\limits_{j,j' \in \mathcal{A}} \min \Big\{ \Big\lfloor \log_a \Big(\frac{K \log N}{a\Delta^2} \Big) \Big\rfloor, \lfloor \log_a(N_j) \rfloor, \lfloor \log_a(N_{j'}) \rfloor \Big\} \Big)$.

In what follows, we highlight the significance of the communication complexity results of ODC.

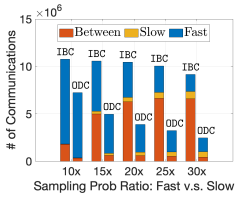
Remark 6 *(Communication complexity characterization by number of decision rounds.) A major contribution of Theorem 2 is the generalization of the communication complexity analysis to the asynchronous-agent setting where the upper bound depends on the number of decision rounds of the agents instead of the length of time horizon. More specifically, the communication complexity of* ODC *implicitly depends on the total number of decisions by all agents, N. In general when agent pull rates differ significantly from each other, N is much smaller than $M \times T$, and Theorem 2 provides a much tighter upper bound than previous results relying on T.*

Remark 7 *(Performance with asynchronous agents.)* ODC *is able to deal with heterogeneous pull rates, since communication can be tailored for on-demand transmissions. Especially, under ODC, the number of communications between each pair of agents depends on the slower agent; while, under IBC, the number of communications between each pair of agents is dominated by the faster agent. For example, consider a two-agent system where agent j^{fast} is a fast agent that pulls arms much more often than a slow agent j^{slow}, i.e., $N_{j^{fast}} \gg N_{j^{slow}}$. By Theorem 2, the number of messages sent by* ODC *is at most $2N_{j^{slow}} + 2$, while, by Remark 5, the number of messages sent under immediate communication can be as large as $N_{j^{fast}} + N_{j^{slow}}$.*

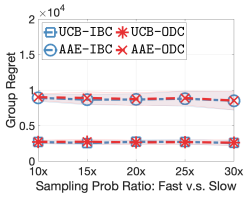
Remark 8 *(Recovery of logarithmic communication complexity in synchronous setting.) When our asynchronous* ODC *protocol is applied to a MAMAB setting with synchronous agents, i.e., $N_j = T, \forall j \in \mathcal{A}$, Corollary 2 implies that we can recover a $O(M^2 \log T)$ communication complexity by doubling buffer threshold after each message transmission.*



(a) UCB Communication

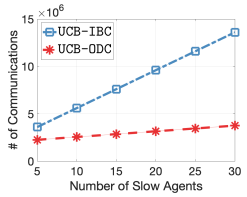


(b) AAE Communication

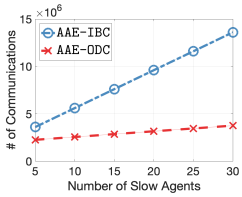


(c) UCB and AAE Regret

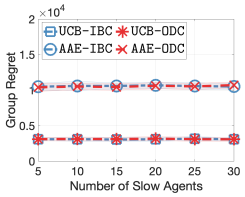
Figure 2: Experiment 1 — impact of the heterogeneity of agent speeds. Forty agents with fixed mean sampling probability and increasing sampling probability ratio between fast and slow agents.



(a) UCB Communication



(b) AAE Communication



(c) UCB and AAE Regret

Figure 3: Experiment 2 — impact of the number of slow agents. Increasing the number of slow agents while fixing the expected total number of decisions in the entire system.

Remark 9 *(Double logarithmic communication complexity of* `AAE-ODC`*) As shown in Theorem 2(b), the communication complexity of* `AAE-ODC` *depends logarithmically on the total number of decision rounds among all agents,* N*, and therefore depends logarithmically on the learning horizon,* T*, when the suboptimality gaps are large, e.g.,* $\Delta_i \gg 1/\sqrt{N_j}, \forall j \in \mathcal{A}, i \in \mathcal{K} \setminus \{i^*\}$*. This is because* `AAE-ODC` *stops communication once the exploration task completes, i.e., once the candidate set size becomes one. Corollary 2(b) further shows that double logarithmic communication complexity can be achieved if the buffer thresholds of* `AAE-ODC` *are set to be doubling.*

Note that `ODC` also works for the scenario that communication has a deterministic delay and that each agent only has access to a *local* subset of the K arms, as in Yang et al. (2022); Chawla et al. (2020); Yang et al. (2021). We provide regret and communication complexity analysis of both algorithms for such scenario in Appendix G and H.

5 NUMERICAL EXPERIMENTS

In this section, we first study the impact of differences in agent pull rates (Experiment 1) and number of slow agents in the system (Experiment 2) on communication complexity and group regret. We compare `UCB-ODC` and `AAE-ODC` with their counterparts that use immediate broadcast communication, labeled `UCB-IBC` and `AAE-IBC` [4] in the first two experiments with buffer thresholds set to one to better demonstrate the insights of `ODC`. In Experiment 3, we study the four above algorithms when buffer thresholds are allowed to double after each communication.

Experimental Setup. In our experiments, there are $M = 40$ agents each with $K = 16$ arms with Bernoulli rewards whose means are uniformly and randomly taken from *Ad-Clicks* (Avito, 2015). We set $\alpha = 3$ for all algorithms and report values averaged over 30 independent trials. We report the average cumulative group regret after $T = 80,000$ time slots for three experimental scenarios.

Experiment 1. In this experiment, we study the impact of differences in agent pull rates on communication complexity and group regret. Specifically, we fix the expected total number of decisions in the entire system by fixing the mean sampling probabilities among 40 agents, and we increase the sampling probability ratio between fast and slow agents from $10\times$ to $30\times$ with a step size of 5. Specifically we fix the sampling probability of each slow agent at 0.01 and vary that of each fast agent from 0.1 to 0.3 with step size 0.05. Note that we maintain the mean sampling probability

[4] `UCB-IBC` and `AAE-IBC` are essentially the same as `CO-UCB` and `CO-AAE` respectively proposed in Yang et al. (2022).

among agents at 0.085. Hence the number of fast (resp. slow) agents decreases (resp. increases) as the sampling probability ratio increases.

Figures 2a and 2b report the total amount of communication under the immediate broadcast communication (IBC) and the ODC protocols. We distinguish three types of communications: (1) between fast and slow agents (orange), (2) among slow agents (yellow), and (3) among fast agents (blue). Figures 2a and 2b show that ODC reduces the amount of communications in all three categories but most notably between fast and slow agents. The amount of communication between fast and slow agents increases under IBC while remaining relatively constant under ODC as the sampling probability ratio between fast and slow agents increases. This demonstrates that ODC is communication efficient when there is large difference in the pull rates of fast and slow agents. Figure 2c shows that UCB-ODC and AAE-ODC exhibit similar group regrets to those of UCB-IBC and AAE-IBC respectively.

Experiment 2. Next, we study the impact of the number of slow agents on communication complexity and group regret while fixing the expected total number of decisions in the entire system. We fix the number of fast agents at 5 and increase the number of slow agents from 5 to 30 with steps of 5. The sampling probability of a fast agent is always 0.8. As the expected total number of decisions is fixed at $300,000$, the sampling probability of slow agents decreases from 0.2 to 0.034 as the number of slow agents increases.

Figure 3 reports the results of Experiment 2. Figures 3a and 3b show that the number of communications of UCB-IBC and AAE-IBC increase significantly as the number of slow agents increases even though the expected total number of decisions does not change; while the amount of communication of UCB-ODC and AAE-ODC do not change much as the number of slow agents increases. Figure 3c shows that UCB-ODC and AAE-ODC still achieve similar group regrets as UCB-IBC and AAE-IBC respectively even though they require fewer message exchanges for cooperation.

Experiment 3. Finally, we study the performance of the four policies, UCB-IBC, UCB-ODC, AAE-IBC, and AAE-ODC, all with doubling buffer thresholds, which we denote as UCB-IBC-D, UCB-ODC-D, AAE-IBC-D, and AAE-ODC-D, in a system with one fast agent with sampling probability one and nine slow agents with sampling probabilities 0.001. Table 2 summarizes the results, which again verifies our theoretical observation that ODC reduces the communications between asynchronous agents while achieving similar group regrets as IBC.

Table 2: Experiment 3

	Communication	Group Regret
UCB-IBC-D	629 ± 2	1474 ± 89
UCB-ODC-D	563 ± 6	1514 ± 114
AAE-IBC-D	629 ± 2	2679 ± 254
AAE-ODC-D	564 ± 5	2672 ± 225

We present supplementary experiments and provide more insights on simulation results in Appendix I. Specifically, in Appendix I.1 we numerically study the performance of ODC with constant or doubling buffer threshold; in Appendix I.2 we present the individual regrets in Experiment 1 and Experiment 2; in Appendix I.3 we numerically study the performance of ODC under different types of asynchronicity; in Appendix I.4 we demonstrate when AAE-ODC would incur fewer communications than UCB-ODC.

6 CONCLUSION AND FUTURE DIRECTION

This paper presented a communication protocol for efficient cooperation in asynchronous multi-agent bandits settings. The communication protocol explicitly adjust the amount of cooperation in proportion to agent pull rates and could be integrated into an underlying bandit algorithm. We combined the proposed communication protocol with two bandit algorithms and analyzed their performance in terms of regret and communication complexities.

A limitation of this work is that we assume all messages are sent through reliable communication, e.g., TCP protocol. ODC suffers potential performance degradation when it is used under unreliable communication, e.g., UDP protocol. Specifically, ODC suffers performance degradation if there is packet loss in communication. For example, after agent j sends a message to agent j' and sets $E^{j \to j'} \leftarrow$ `False`, if this sharing message is lost without reaching agent j', then the cooperation between agent j and j' will end. This is because from both agents' perspectives, each other's exchange demands are both `False`. Designing a loss-tolerant communication protocol for asynchronous MAMAB is an interesting open problem.

References

Agarwal, M., Aggarwal, V., and Azizzadenesheli, K. (2021). Multi-agent multi-armed bandits with limited communication. *arXiv preprint arXiv:2102.08462*.

Avito (2015). *Avito Context Ad Clicks*. `https://www.kaggle.com/c/avito-context-ad-clicks`.

Bar-On, Y. and Mansour, Y. (2019). Individual regret in cooperative nonstochastic multi-armed bandits. *Advances in Neural Information Processing Systems*, 32.

Bedi, A. S., Koppel, A., and Rajawat, K. (2019). Asynchronous online learning in multi-agent systems with proximity constraints. *IEEE Transactions on Signal and Information Processing over Networks*, 5(3):479–494.

Besson, L. and Kaufmann, E. (2018). Multi-player bandits revisited. In *Algorithmic Learning Theory*, pages 56–92. PMLR.

Bistritz, I. and Bambos, N. (2020). Cooperative multi-player bandit optimization. *Advances in Neural Information Processing Systems*, 33:2016–2027.

Bistritz, I. and Leshem, A. (2018). Distributed multi-player bandits-a game of thrones approach. *Advances in Neural Information Processing Systems*, 31.

Boursier, E. and Perchet, V. (2019). Sic-mmab: synchronisation involves communication in multi-player multi-armed bandits. *Advances in Neural Information Processing Systems*, 32.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. In *Foundations and Trends® in Machine Learning*, pages 1–122.

Bubeck, S., Li, Y., Peres, Y., and Sellke, M. (2020). Non-stochastic multi-player multi-armed bandits: Optimal rate with collision information, sublinear without. In *Conference on Learning Theory*, pages 961–987. PMLR.

Buccapatnam, S., Tan, J., and Zhang, L. (2015). Information sharing in distributed stochastic bandits. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2605–2613. IEEE.

Cesa-Bianchi, N., Cesari, T., and Monteleoni, C. (2020). Cooperative online learning: Keeping your neighbors updated. In *Algorithmic Learning Theory*, pages 234–250. PMLR.

Cesa-Bianchi, N., Gentile, C., Mansour, Y., and Minora, A. (2016). Delay and cooperation in nonstochastic bandits. In *Conference on Learning Theory*, pages 605–622. PMLR.

Chakraborty, M., Chua, K. Y. P., Das, S., and Juba, B. (2017). Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *IJCAI*, pages 164–170.

Chawla, R., Sankararaman, A., Ganesh, A., and Shakkottai, S. (2020). The gossiping insert-eliminate algorithm for multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3471–3481. PMLR.

Della Vecchia, R. and Cesari, T. (2021). An efficient algorithm for cooperative semi-bandits. In *Algorithmic Learning Theory*, pages 529–552. PMLR.

Dubey, A. et al. (2020). Cooperative multi-agent bandits with heavy tails. In *International Conference on Machine Learning*, pages 2730–2739. PMLR.

Dubey, A. and Pentland, A. (2020). Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33:6003–6014.

Even-Dar, E., Mannor, S., Mansour, Y., and Mahadevan, S. (2006). Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6).

Féraud, R., Alami, R., and Laroche, R. (2019). Decentralized exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 1901–1909. PMLR.

Gao, Z., Han, Y., Ren, Z., and Zhou, Z. (2019). Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems*, 32.

He, J., Wang, T., Min, Y., and Gu, Q. (2022). A simple and provably efficient algorithm for asynchronous federated contextual linear bandits. *Advances in neural information processing systems*.

Hillel, E., Karnin, Z. S., Koren, T., Lempel, R., and Somekh, O. (2013). Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems*, 26.

Hossain, S., Micha, E., and Shah, N. (2021). Fair algorithms for multi-agent multi-armed bandits. *Advances in Neural Information Processing Systems*, 34.

Jiang, J., Zhang, W., Gu, J., and Zhu, W. (2021). Asynchronous decentralized online learning. *Advances in Neural Information Processing Systems*, 34.

Joulani, P., György, A., and Szepesvári, C. (2019). Think out of the" box": Generically-constrained asynchronous composite optimization and hedging. *Advances in Neural Information Processing Systems*, 32.

Kolla, R. K., Jagannathan, K., and Gopalan, A. (2018). Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking*, 26(4):1782–1795.

Landgren, P., Srivastava, V., and Leonard, N. E. (2016). Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 167–172. IEEE.

Li, C. and Wang, H. (2022). Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6529–6553. PMLR.

Madhushani, U., Dubey, A., Leonard, N., and Pentland, A. (2021). One more step towards reality: Cooperative bandits with imperfect communication. *Advances in Neural Information Processing Systems*, 34.

Martínez-Rubio, D., Kanade, V., and Rebeschini, P. (2019). Decentralized cooperative stochastic bandits. *Advances in Neural Information Processing Systems*, 32.

Perchet, V., Rigollet, P., Chassang, S., and Snowberg, E. (2016). Batched bandit problems. *The Annals of Statistics*, pages 660–681.

Sankararaman, A., Ganesh, A., and Shakkottai, S. (2019). Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35.

Shi, C., Shen, C., and Yang, J. (2021a). Federated multi-armed bandits with personalization. In *International Conference on Artificial Intelligence and Statistics*, pages 2917–2925. PMLR.

Shi, C., Xiong, W., Shen, C., and Yang, J. (2021b). Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. *Advances in Neural Information Processing Systems*, 34.

Szorenyi, B., Busa-Fekete, R., Hegedus, I., Ormándi, R., Jelasity, M., and Kégl, B. (2013). Gossip-based distributed stochastic bandit algorithms. In *International Conference on Machine Learning*, pages 19–27. PMLR.

Wang, P.-A., Proutiere, A., Ariu, K., Jedra, Y., and Russo, A. (2020). Optimal algorithms for multiplayer multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 4120–4129. PMLR.

Wang, Y., Hu, J., Chen, X., and Wang, L. (2019). Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*.

Yang, L., Chen, Y.-z. J., Hajiesmaili, M., Lui, J. C., and Don, T. (2022). Distributed bandits with heterogeneous agents. In *IEEE International Conference on Computer Communications (INFOCOM)*.

Yang, L., Chen, Y.-Z. J., Pasteris, S., Hajiesmaili, M., Lui, J., and Towsley, D. (2021). Cooperative stochastic bandits with asynchronous agents and constrained feedback. *Advances in Neural Information Processing Systems*, 34:8885–8897.

A LITERATURE REVIEW

Collision or no collision. One of the extensively studied MAMAB settings is the *collision* scenario (Wang et al., 2020; Boursier and Perchet, 2019; Shi et al., 2021b; Bistritz and Leshem, 2018; Bubeck et al., 2020; Besson and Kaufmann, 2018), where agents receive zero or degraded rewards if they pull the same arm simultaneously. This setting well models the opportunistic spectrum access applications with multiple users, where the objective is to choose the best channels while avoiding users communicate through the same channel at the same time. On the other hand, the MAMAB setting with *no collision* (Shi et al., 2021a; Wang et al., 2019, 2020; Bar-On and Mansour, 2019; Chakraborty et al., 2017; Dubey et al., 2020; Szorenyi et al., 2013; Chawla et al., 2020; Landgren et al., 2016; Buccapatnam et al., 2015; Martínez-Rubio et al., 2019; Bistritz and Bambos, 2020; Madhushani et al., 2021; Chakraborty et al., 2017; Cesa-Bianchi et al., 2016; Hillel et al., 2013; Dubey et al., 2020; Yang et al., 2021, 2022; Sankararaman et al., 2019; Féraud et al., 2019) has also attracted increasing research interest. In the MAMAB setting with no collision, agents receive independent rewards without any degradation even when they pull the same arm. This setting is more suitable for modeling applications like recommender systems, clinical trials, robotic taget searching, etc. In this paper, we focus on the *no collision* setting.

Cooperate with or without a coordinator. Regarding cooperation methods in cooperative MAMAB, there are two broad categories of prior work: (1) *cooperation with coordinator* (Shi et al., 2021a; Wang et al., 2019, 2020; Bar-On and Mansour, 2019; Chakraborty et al., 2017; Dubey et al., 2020), which utilizes a central server or elects leaders among agents to coordinate the learning process. (2) *cooperation without coordinator* (Szorenyi et al., 2013; Chawla et al., 2020; Landgren et al., 2016; Buccapatnam et al., 2015; Martínez-Rubio et al., 2019; Bistritz and Bambos, 2020; Madhushani et al., 2021; Chakraborty et al., 2017; Cesa-Bianchi et al., 2016; Hillel et al., 2013; Dubey et al., 2020; Yang et al., 2021, 2022; Sankararaman et al., 2019; Féraud et al., 2019), which addresses a decentralized learning scenario where agents communicate with each other to improve their learning performance. In this work, we consider the *cooperation without coordinator* (decentralized) approach in an asynchronous MAMAB setting where agent pull times and speeds being unknown, irregular, and different, hinders the application of a coordination approach.

Reward assumptions. Similar to standard bandit problem, various reward assumptions are studied in decentralized cooperative MAMAB model. For example, Szorenyi et al. (2013); Chawla et al. (2020); Landgren et al. (2016); Buccapatnam et al. (2015); Martínez-Rubio et al. (2019) consider stochastic bandit, Dubey et al. (2020) studies stochastic bandit with heavy tails, and Cesa-Bianchi et al. (2016) considers non-stochastic bandit. In this work, we consider arms with stochastic rewards and assume they have Bernoulli distributions.

Homogeneous or heterogeneous arm sets. In decentralized cooperative MAMAB model, agents can have homogeneous arm sets or heterogeneous arm sets. Homogeneous arm sets setting (Szorenyi et al., 2013; Landgren et al., 2016; Buccapatnam et al., 2015; Martínez-Rubio et al., 2019), i.e., same set of arms is available to each agent, is more extensively studied. Regarding heterogeneous arm sets scenario, there are two different notions of heterogeneous arm sets as far as we notice. One refer to the scenario where agents have access to the same set of arms but each agent receive different expected reward from the same arm, e.g., in Hossain et al. (2021). This setting models the opportunistic spectrum access application and mobile sensor environment estimating application, where the geographical location of agents influence the rewards they receive from the same arm. The other definition of heterogeneous arm sets models the scenario that agents receive same expected rewards from the same arm but each agent only have access to a subset of all the arms, e.g, in Yang et al. (2022); Chawla et al. (2020); Yang et al. (2021). In this work, we mainly consider homogeneous arm sets setting. We provide extension of our results to account for agents pulling from different but overlapping subsets of arms in Appendix.

Synchronous or asynchronous agents. In decentralized cooperative MAMAB model, agents can operate synchronously or asynchronously. The synchronous setting (Szorenyi et al., 2013; Chawla et al., 2020; Landgren et al., 2016; Buccapatnam et al., 2015; Martínez-Rubio et al., 2019) is more extensively studied. In the synchronous setting, there is a common clock among all agents, and every agent pulls an arm at every time slot. In the synchronous MAMAB setting, the batch approach (a.k.a., doubling epoch, phase, buffer) (Perchet et al., 2016; Gao et al., 2019) has been used to achieve logarithmic communication complexity, e.g, by Agarwal et al. (2021); Shi et al. (2021b); Boursier and Perchet (2019). Yang et al. (2021, 2022); Sankararaman et al. (2019); Féraud et al. (2019) addresses MAMAB with asynchronous agents. The model in Yang et al. (2021, 2022) assumes each agent periodically make decisions at different *known* frequencies. Sankararaman et al. (2019) assumes each agent is equipped with a Poisson clock and agent pull when its clock rings. Féraud et al. (2019) assumes there is a distribution determining which agent becomes active at each time slot. Our paper assumes that pulling times are unknown, irregular, and not necessarily stochastic. Last, asynchronous multi-agent learning has also been studied in related fields such as online (convex) optimization with full information or semi-bandit feedback (Cesa-Bianchi et al., 2020; Jiang et al., 2021; Joulani et al., 2019; Bedi et al., 2019; Della Vecchia and Cesari, 2021).

Communication schemes. Many different types of communication has been studied in decentralized cooperative MAMAB literature. For example, Szorenyi et al. (2013) considers peer-to-peer network and let each agent communicate to only a fixed number of agents at each round. Chawla et al. (2020); Sankararaman et al. (2019) considers a gossip-style communication, where agents are assumed located on a graph and agents can only communicate with their neighbors. The gossip-style communication can be used to model the scenario that users in a social network explore restaurants and make recommendations to their friends. In Buccapatnam et al. (2015); Yang et al. (2021, 2022), each agent is allowed to immediately broadcast the rewards to all other agents. In this work, we consider that each agent is allowed to communicate with every other agent and design a communication protocol that is efficient when agents operate asynchronously. The proposed on-demand communication protocol, `ODC`, is a fundamentally different idea from previously considered communication protocols in decentralized cooperative MAMAB literature, such as immediate broadcasting, peer-to-peer (Dubey and Pentland, 2020), consensus-based (Martínez-Rubio et al., 2019), and gossip-style (Sankararaman et al., 2019) communication, under which agents spontaneously transmit information.

B PROOF OF THEOREM 1

B.1 Proof of Theorem 1(a)

To proceed with the proof of expected group regret of `UCB-ODC`, we first state some intermediary lemmas and then use the lemmas to upper bound the group regret. The first two lemmas are regarding two types of decisions, namely Type-I and Type-II.

Definition 2 *At any decision round t, the decision of agent j is a Type-I decision if the following equation holds*

$$\mu(i) \in [\hat{\mu}_j^t(i) - \mathtt{CI}_j^t(i), \hat{\mu}_j^t(i) + \mathtt{CI}_j^t(i)], \quad \forall i \in \mathcal{K}; \tag{6}$$

otherwise the decision is a Type-II decision.

Lemma 1 *At any decision round t, an agent j makes a Type-I decision with a probability at least $1 - 2KN_t\delta_j^{t\alpha}$, where $N_t = \sum_{j\in\mathcal{A}} n_j^t$.*

Proof of Lemma 1: Note that for any arm i with n observations, by Hoeffding's inequality and union bound, we have

$$\mathbb{P}\left(\left|\mu(i) - \hat{\mu}(i,n)\right| > \sqrt{\frac{\alpha \log \delta^{-1}}{2n}}\right) \leq 2\delta^{\alpha}.$$

Thus, the probability that the true mean value of arm i is not in the confidence interval when agent j makes a decision at time t is at most $2N_t{\delta_j^t}^\alpha$, where $N_t = \sum_{j\in\mathcal{A}} n_j^t$, as shown in the following.

$$\mathbb{P}\left(\left|\mu(i) - \hat{\mu}(i, \hat{n}_j^t(i))\right| > \sqrt{\frac{\alpha \log {\delta_j^t}^{-1}}{2\hat{n}_j^t(i)}}\right)$$
$$\leq \sum_{s=1}^{N_t} \mathbb{P}\left(\left|\mu(i) - \hat{\mu}(i,n)\right| > \sqrt{\frac{\alpha \log {\delta_j^t}^{-1}}{2n}} \middle| n = s\right) \mathbb{P}(n=s) \leq 2N_t{\delta_j^t}^\alpha.$$

Hence, the probability that (6) holds for all arm $i \in \mathcal{K}$ is lower bounded by $1 - \sum_{i\in\mathcal{K}} 2N_t{\delta_j^t}^\alpha = 1 - 2KN_t{\delta_j^t}^\alpha$. □

By Lemma 1, with probability at most $2KN_t{\delta_j^t}^\alpha$, an agent makes a Type-II decision at a decision round. With $\delta_j^t = 1/N$, the expected number of Type-II decisions made by all agents over the entire time horizon, denoted by $\mathbb{E}[Q_{\text{II}}]$ is upper bounded by

$$\mathbb{E}[Q_{\text{II}}] \leq \sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} 2KN_{t_l}{\delta_j^{t_l}}^\alpha = \sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} 2K\frac{N_{t_l}}{N^\alpha} \leq \sum_{j\in\mathcal{A}} \sum_{l=1}^{N_j} 2K\frac{1}{N^{\alpha-1}}$$
$$\overset{\text{(a)}}{\leq} \sum_{j\in\mathcal{A}} \frac{2K}{\alpha-2}\left(1 - \frac{1}{N^{\alpha-2}}\right) \equiv q_{\text{II}} \overset{\text{(b)}}{\leq} 2KM. \tag{7}$$

In Eq. (7), (a) holds because $l \leq N$ for each l and (b) holds if $\alpha \geq 3$.

Lemma 2 *If agent $j \in \mathcal{A}$ makes a Type-I decision and pulls suboptimal arm $i \in \mathcal{K}$ by the* `UCB-ODC` *algorithm, at that decision round t we have*

$$\hat{n}_j^t(i) \leq \frac{2\alpha \log(1/\delta_j^t)}{\Delta_i^2}.$$

Proof of Lemma 2: If agent $j \in \mathcal{A}$ makes a Type-I decision and pulls suboptimal arm $i \in \mathcal{K}$ at time t by the `UCB-ODC` algorithm, we have

$$2\texttt{CI}_j^t(i) \geq \Delta_i. \tag{8}$$

Because otherwise,

$$\hat{\mu}(i^*) + \texttt{CI}_j^t(i^*) \geq \mu(i^*) = \mu(i) + \Delta_i > \mu(i) + 2\texttt{CI}_j^t(i) > \hat{\mu}(i) + \texttt{CI}_j^t(i),$$

contradicting the fact that arm i is pulled by `UCB-ODC` as it has the highest UCB. Rewrite (8) using the definition of $\texttt{CI}_j^t(i)$ in (1), we have

$$\hat{n}_j^t(i) \leq \frac{2\alpha \log(1/\delta_j^t)}{\Delta_i^2}.$$

□

Recall that $n_j^t(i)$ denotes the number of times agent j has pulled arm i up to time t. In the cooperative learning process, there must exist a time slot τ_i for each subooptimal arm $i \in \mathcal{K} \setminus \{i^*\}$ such that

$$\frac{2\alpha \log N}{\Delta_i^2} + M \geq \sum_{j'\in\mathcal{A}} n_{j'}^{\tau_i}(i) > \frac{2\alpha \log N}{\Delta_i^2} \geq \sum_{j'\in\mathcal{A}} n_{j'}^{\tau_i - 1}(i). \tag{9}$$

The number of times arm i pulled after time τ_i are considered as extra number of pulls. These extra pulls are because of three possible causes: 1) Type-I decision due to delayed transmission for waiting for exchange demands, 2) Type-I decision due to delayed transmission for waiting for buffer thresholds to be satisfied, 3) Type-II decisions.

We first examine Type-I decision cases. Note that $\hat{n}_j^t(i)$ is the total number of observations of arm i agent j possessed at time t, including both the number of times agent j pulls arm i and some or all

number of times other agents in $\mathcal{A}$ pull arm i. We define $B_t^{j\to j'}(i)$ as the number of reward samples of arm i stored in agent j's buffer for agent j' (and not yet been sent) at time t, and define $B_t^{j\to j'}$ as the total number of observations stored in agent j's buffer for agent j' at time t. Consider an agent $j \in \mathcal{A}$ and a suboptimal arm i such that, at time τ_i,

$$\frac{2\alpha \log N}{\Delta_i^2} \geq \frac{2\alpha \log 1/\delta_j^{\tau_i}}{\Delta_i^2} \geq \hat{n}_j^{\tau_i}(i) = n_j^{\tau_i}(i) + \sum_{j'\in\mathcal{A}\setminus\{j\}} n_{j'}^{\tau_i}(i) - B_{\tau_i}^{j'\to j}(i) \tag{10}$$

$$\overset{(a)}{>} \frac{2\alpha \log N}{\Delta_i^2} - \sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i) \tag{11}$$

$$= \frac{2\alpha \log N}{\Delta_i^2} - \sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}} - \sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}} \tag{12}$$

$$\overset{(b)}{\geq} \frac{2\alpha \log N}{\Delta_i^2} - \sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}} - \sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}, \tag{13}$$

where inequality (a) is because of (9); inequality (b) is because, for agent $j' \in \mathcal{A}\setminus\{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{true}$, we have $f(c_{\tau_i}^{j'\to j}) \geq B_{\tau_i}^{j'\to j} \geq B_{\tau_i}^{j'\to j}(i) \geq 0$. According to Lemma 2, such an agent j makes Type-I decisions to pull arm i after time τ_i.

In the following, we bound the extra number of times agent j pulls arm i to make up for the delayed transmission from other agents j'. For an agent $j' \in \mathcal{A}\setminus\{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{false}$, if $B_{\tau_i}^{j'\to j}(i) < f(c_{\tau_i}^{j'\to j})$, agent j has to make at most $f(c_{\tau_i}^{j'\to j})$ extra pulls of i to make up for agent j''s delay; if $B_{\tau_i}^{j'\to j}(i) \geq f(c_{\tau_i}^{j'\to j})$, agent j can receive those observations from j' once agent j buffers $f(c_{\tau_i}^{j\to j'})$ observations for j' and sends a message to j'. Hence, because of the delayed transmission from agents $j' \in \mathcal{A}\setminus\{j\} : E_{\tau_i}^{j'\to j} = \texttt{false}$, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}\setminus\{j\}} f(\max\{c_{\tau_i}^{j'\to j}, c_{\tau_i}^{j\to j'}\})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}} \leq \sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}}, \tag{14}$$

where the inequality is because, by the definition of the ODC, for any pair of agents $j, j' \in \mathcal{A}$ at any time t, if $E_t^{j'\to j} = \texttt{false}$, $1 \geq c_t^{j'\to j} - c_t^{j\to j'} \geq 0$. On the other hand, agents $j' \in \mathcal{A}\setminus\{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{true}$ delay transmission of $\sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}$ observations of i to agent j at time τ_i due to waiting for the buffer thresholds to be satisfied. To make up for this type of delay, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}. \tag{15}$$

By (14) (15) and Lemma 2, agent j contributes at most F_i^j extra numbers of pullings of arm i after time τ_i, where

$$F_i^j = \min\Big\{\Big(\sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\Big), \frac{2\alpha \log N}{\Delta_i^2}\Big\}. \tag{16}$$

We now examine Type-II decision case. According to Lemma 1 and (7), the expected number of Type-II decisions made by all agents over the entire time horizon is upper bounded by $2KM$. Since, in our case, $\Delta_i \leq 1, \forall i \in \mathcal{K}$, the regret incurred by Type-II decisions is upper bounded by $2KM$.

The expected group regret can be bounded by

$$\mathbb{E}[R] = \sum_{j\in\mathcal{A}} \mathbb{E}[R_{N_j}^j] = \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{K}} \Delta_i \mathbb{E}[n_j^T(i)] = \sum_{i\in\mathcal{K}} \Delta_i \Big(\sum_{j\in\mathcal{A}}\sum_{\ell=1}^{N_j} \mathbb{P}[I_{t_\ell^j}^j = i]\Big) \tag{17}$$

$$\leq 2KM + \sum_{i\in\mathcal{K}} \Delta_i \Big(\frac{2\alpha \log N}{\Delta_i^2} + M + \sum_{j\in\mathcal{A}} F_i^j\Big) \leq 3KM + \sum_{i\in\mathcal{K}:\Delta_i>0} \Big(\frac{2\alpha \log N}{\Delta_i} + \sum_{j\in\mathcal{A}} F_i^j \Delta_i\Big). \tag{18}$$

This completes the proof of Theorem 1(a).

B.2 Proof of Theorem 1(b)

Similar to the analysis of regret of UCB-ODC in previous subsection, we utilize intermediary lemmas regarding Type-I and Type-II decisions to upper bound the group regret. Agent j makes a Type-I decision if (6) holds, otherwise it is a Type-II decision.

Lemma 3 *If agent $j \in \mathcal{A}$ makes a Type-I decision and pulls suboptimal arm $i \in \mathcal{K}$ by AAE-ODC algorithm, at that decision round t we have*

$$\hat{n}_j^t(i) \leq \frac{8\alpha \log(1/\delta_j^t)}{\Delta_i^2}. \tag{19}$$

Proof of Lemma 3: If agent $j \in \mathcal{A}$ makes a Type-I decision and pulls suboptimal arm $i \in \mathcal{K}$ at time t by AAE-ODC algorithm, we have

$$2\texttt{CI}_j^t(i^*) + 2\texttt{CI}_j^t(i) \geq \Delta_i. \tag{20}$$

Because otherwise,

$$\begin{aligned}\hat{\mu}(i^*) - \texttt{CI}_j^t(i^*) &= \hat{\mu}(i^*) + \texttt{CI}_j^t(i^*) - 2\texttt{CI}_j^t(i^*) \\ &\geq \mu(i^*) - 2\texttt{CI}_j^t(i^*) = \mu(i) + \Delta_i - 2\texttt{CI}_j^t(i^*) \\ &> \mu(i) + 2\texttt{CI}_j^t(i) \geq \hat{\mu}(i) + \texttt{CI}_j^t(i),\end{aligned} \tag{21}$$

contradicting the fact that arm i is pulled by AAE-ODC as it is in the candidate set (if (21) holds, arm i should not be in the candidate set). Since AAE-ODC pulls the arm with least observations in the candidate set, we have $\hat{n}_j^t(i) \leq \hat{n}_j^t(i^*)$ and thereby $\texttt{CI}_j^t(i) \geq \texttt{CI}_j^t(i^*)$. Rewrite $4\texttt{CI}_j^t(i) \geq \Delta_i$ using the definition of $\texttt{CI}_j^t(i)$ in (1), we obtain

$$\hat{n}_j^t(i) \leq \frac{8\alpha \log 1/\delta_j^t}{\Delta_i^2}.$$

□

We then upper bound the group regret of AAE-ODC by similar steps in previous subsection.

Let τ_i be the time slot for suboptimal arm i that

$$\frac{16\alpha \log N}{\Delta_i^2} + M \geq \sum_{j' \in \mathcal{A}} n_{j'}^{\tau_i}(i) > \frac{16\alpha \log N}{\Delta_i^2} \geq \sum_{j' \in \mathcal{A}} n_{j'}^{\tau_i - 1}(i). \tag{22}$$

Consider an agent j and a suboptimal arm i such that at time τ_i,

$$\frac{16\alpha \log N}{\Delta_i^2} \geq \frac{16\alpha \log 1/\delta_j^{\tau_i}}{\Delta_i^2} \geq \hat{n}_j^{\tau_i}(i) = n_j^{\tau_i}(i) + \sum_{j' \in \mathcal{A} \setminus \{j\}} n_{j'}^{\tau_i}(i) - B_{\tau_i}^{j' \to j}(i) \tag{23}$$

$$\begin{aligned}&\overset{(a)}{>} \frac{16\alpha \log N}{\Delta_i^2} - \sum_{j' \in \mathcal{A} \setminus \{j\}} B_{\tau_i}^{j' \to j}(i) \mathbb{1}_{E_{\tau_i}^{j' \to j} = \texttt{false}} - \sum_{j' \in \mathcal{A} \setminus \{j\}} B_{\tau_i}^{j' \to j}(i) \mathbb{1}_{E_{\tau_i}^{j' \to j} = \texttt{true}} \\ &\overset{(b)}{\geq} \frac{16\alpha \log N}{\Delta_i^2} - \sum_{j' \in \mathcal{A} \setminus \{j\}} B_{\tau_i}^{j' \to j}(i) \mathbb{1}_{E_{\tau_i}^{j' \to j} = \texttt{false}} - \sum_{j' \in \mathcal{A} \setminus \{j\}} f(c_{\tau_i}^{j' \to j}) \mathbb{1}_{E_{\tau_i}^{j' \to j} = \texttt{true}},\end{aligned} \tag{24}$$

where $B_t^{j \to j'}(i)$ denotes the number of reward samples of arm i stored in agent j's buffer for agent j' (and not yet been sent) at time t; $B_t^{j \to j'}$ denotes the total number of observations stored in agent j's buffer for agent j'; inequality (a) is because of (22); inequality (b) is because, for agent $j' \in \mathcal{A} \setminus \{j\}$ such that $E_{\tau_i}^{j' \to j} = \texttt{true}$, we have $f(c_{\tau_i}^{j' \to j}) \geq B_{\tau_i}^{j' \to j} \geq B_{\tau_i}^{j' \to j}(i) \geq 0$. According to Lemma 3, such an agent j makes Type-I decisions to pull arm i after time τ_i.

In the following, we bound the extra number of times agent j pulls arm i to make up for the delayed transmission from other agents j'. For an agent $j' \in \mathcal{A} \setminus \{j\}$ such that $E_{\tau_i}^{j' \to j} = \texttt{false}$,

if $B_{\tau_i}^{j'\to j}(i) < f(c_{\tau_i}^{j'\to j})$, agent j has to make at most $f(c_{\tau_i}^{j'\to j})$ extra pullings of i to make up for agent j''s delay; if $B_{\tau_i}^{j'\to j}(i) \geq f(c_{\tau_i}^{j'\to j})$, agent j can receive those observations from j' once agent j buffers $f(c_{\tau_i}^{j\to j'})$ observations for j' and send a message to j'. Hence, because of the delayed transmission from agents $j' \in \mathcal{A}\setminus\{j\} : E_{\tau_i}^{j'\to j} = \texttt{false}$, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}\setminus\{j\}} f(\max\{c_{\tau_i}^{j'\to j}, c_{\tau_i}^{j\to j'}\})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}} \leq \sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}}, \tag{25}$$

where the inequality is because, by the definition of the ODC, for any pair of agents $j, j' \in \mathcal{A}$ at any time t, if $E_t^{j'\to j} = \texttt{false}$, $1 \geq c_t^{j'\to j} - c_t^{j\to j'} \geq 0$. On the other hand, agents $j' \in \mathcal{A}\setminus\{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{true}$ delay transmission of $\sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}$ observations of i to agent j at time τ_i due to waiting for the buffer thresholds to be satisfied. To make up for this type of delay, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}. \tag{26}$$

By (25) (26)) and Lemma 3, agent j contributes at most G_i^j extra numbers of pulls of arm i after time τ_i, where

$$G_i^j = \min\Big\{\Big(\sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\Big), \frac{16\alpha\log N}{\Delta_i^2}\Big\}. \tag{27}$$

We now examine Type-II decision case. As AAE-ODC also selects arms based on confidence interval as defined in (1), Lemma 1 holds for AAE-ODC. According to Lemma 1, with probability at most $2KN_t\delta_j^{t\,\alpha}$, an agent makes a Type-II decision at a decision round. With $\delta_j^t = 1/N^2$, the regret incurred by Type-II decisions is upper bounded by $2KM$.

The expected group regret can be bounded by

$$\mathbb{E}[R] = \sum_{j\in\mathcal{A}} \mathbb{E}[R_{N_j}^j] = \sum_{j\in\mathcal{A}}\sum_{i\in\mathcal{K}} \Delta_i \mathbb{E}[n_j^T(i)] = \sum_{i\in\mathcal{K}} \Delta_i \Big(\sum_{j\in\mathcal{A}}\sum_{\ell=1}^{N_j} \mathbb{P}[I_{t_\ell^j}^j = i]\Big) \tag{28}$$

$$\leq 2KM + \sum_{i\in\mathcal{K}} \Delta_i \Big(\frac{16\alpha\log N}{\Delta_i^2} + M + \sum_{j\in\mathcal{A}} G_i^j\Big) \leq 3KM + \sum_{i\in\mathcal{K}:\Delta_i>0} \Big(\frac{16\alpha\log N}{\Delta_i} + \sum_{j\in\mathcal{A}} G_i^j\Delta_i\Big). \tag{29}$$

This completes the proof of Theorem 1(b).

B.3 Recovery of near-optimal regret in synchronous setting

When applied to a MAMAB setting with synchronous agents where every agent makes a decision at every time slot, our asynchronous algorithm AAE-ODC can recover a near-optimal regret

$$O\Big(\sum_{i\in\mathcal{K}:\Delta_i>0} \log N/\Delta_i\Big)$$

with the buffer thresholds set to be doubled and proportional to the number of arms remaining in the candidate set, $\mathcal{C}$, i.e., $f(c^{j\to j'}) = |\mathcal{C}| \times 2^{c^{j\to j'}-1}, \forall j, j' \in \mathcal{A}$, $c^{j\to j'} = 1, 2, \ldots$.

Note that, in a synchronous setting, the exchanges demands $E^{j\to j'}$ and $E^{j'\to j}$ are both always true. This is because both exchange demands are true at the beginning, and every time agent j sends a message to agent j', agent j' also sends a message to agent j as the buffer thresholds for all (pairs of) agents are the same in a synchronous setting.

Consider the time τ_i for a suboptimal arm i such that

$$\frac{16\alpha\log N}{\Delta_i^2} + M \geq \sum_{j'\in\mathcal{A}} n_{j'}^{\tau_i}(i) > \frac{16\alpha\log N}{\Delta_i^2} \geq \sum_{j'\in\mathcal{A}} n_{j'}^{\tau_i-1}(i). \tag{30}$$

Consider an agent j such that at time τ_i,

$$\frac{16\alpha \log N}{\Delta_i^2} \geq \frac{8\alpha \log 1/\delta_j^{\tau_i}}{\Delta_i^2} \geq \hat{n}_j^{\tau_i}(i) = n_j^{\tau_i}(i) + \sum_{j' \in \mathcal{A}\setminus\{j\}} n_{j'}^{\tau_i}(i) - B_{\tau_i}^{j' \to j}(i).$$

Under AAE-ODC, the maximum extra number of times agent j pulls arm i after time τ_i is at most $f(c_{\tau_i})$. Because after agent j makes $f(c_{\tau_i})$ number of observations it sends a message to all other agents and receives the outstanding observations of arm i, $\sum_{j' \in \mathcal{A}\setminus\{j\}} B_{\tau_i}^{j' \to j}(i)$.

Hence, the total amount of extra number of times agents pull arm i after time τ_i can be upper bounded by

$$\sum_{j \in \mathcal{A}} f(c_{\tau_i}) \overset{(a)}{\leq} \sum_{j \in \mathcal{A}} n_j^{\tau_i}(i) + K \overset{(b)}{\leq} \frac{16\alpha \log N}{\Delta_i^2} + M + K, \tag{31}$$

where inequality (a) is because, by setting the buffer thresholds to be doubled and proportional to the number of arms remaining in the candidate set, the current buffer threshold of an agent, $f(c_{\tau_i})$, is smaller than or equal to K plus the amount of observations that agent have ever made before τ_i, i.e., $f(c_{\tau_i}) \leq n_j^{\tau_i}(i) + K$; inequality (b) is because of the definition of τ_i in (30).

C ASYMPTOTIC GROUP REGRET LOWER BOUND FOR ASYNCHRONOUS MAMAB

The proof techniques for single-agent multi-armed bandit, e.g., in Bubeck and Cesa-Bianchi (2012), and those for synchronous multi-agent multi-armed bandit, e.g., in Dubey et al. (2020), can be applied to asynchronous multi-agent multi-armed bandit by slight modification. For completion of analysis, we provide the details as follows.

Let $\mathcal{E}_K$ denote the class of K-armed bandit where each arm has a Bernoulli reward distribution and there is no collision, i.e., reward realization of arms is not influenced by actions of agents. Let $\nu = (P_1, ..., P_K) \in \mathcal{E}_K$, $\nu' = (P'_1, ..., P'_K) \in \mathcal{E}_K$ be two K-armed bandit instances such that $P_i = P'_i, \forall i \in \mathcal{K}\setminus\{k\}$, where k is a suboptimal arm. Specifically, P'_k is chosen to be Bernoulli$(\mu_k+\lambda)$ and $\lambda > \Delta_k$. Let π denote a consistent cooperative policy for asynchronous M-agent multi-armed bandit. We have the following divergence decomposition:

$$\begin{aligned}
\mathbb{D}_{\text{KL}}(\mathbb{P}_{\nu\pi}, \mathbb{P}_{\nu'\pi}) &= \mathbb{E}_{\nu\pi}\left[\log \frac{d\mathbb{P}_{\nu\pi}}{d\mathbb{P}_{\nu'\pi}}\left(I_1^{j:t_1^j=1}, x_1(I_1^{j:t_1^j=1}), ..., I_T^{j:t_{n_j}^j=T}, x_T(I_T^{j:t_{n_j}^j=T})\right)\right] \\
&= \sum_{i \in \mathcal{K}} \mathbb{E}_{\nu\pi}\left[\sum_{j \in \mathcal{A}} n_j^T(i)\right] \mathbb{D}_{\text{KL}}(P_i, P'_i) = \mathbb{E}_{\nu\pi}\left[\sum_{j \in \mathcal{A}} n_j^T(k)\right] \mathbb{D}_{\text{KL}}(P_k, P'_k),
\end{aligned} \tag{32}$$

where $n_j^T(i)$ is the total number of times agent j pulls arm i, and $\mathbb{P}_{\nu\pi}$, $\mathbb{P}_{\nu'\pi}$ are the distributions of the action-reward history induced by the interaction of policy π with bandit instances ν and ν' respectively.

By high-probability Pinsker inequality, we have the following for any event A:

$$\mathbb{D}_{\text{KL}}(\mathbb{P}_{\nu\pi}, \mathbb{P}_{\nu'\pi}) \geq \log \frac{1}{2(\mathbb{P}_{\nu\pi}(A) + \mathbb{P}_{\nu'\pi}(A^c))}. \tag{33}$$

Let R and R' be the (group) regret obtained by policy π on bandit instances ν and ν' respectively given the asynchronous pulling times of agents $(t_1^j, t_2^j, ...t_{N_j}^j), \forall j \in \mathcal{A}$. By (32) (33) and by choosing $A = \left\{\sum_{j \in \mathcal{A}} n_j^T(k) \geq \frac{1}{2}\sum_{j \in \mathcal{A}} N_j = \frac{N}{2}\right\}$, we have

$$\begin{aligned}
R + R' &\geq \frac{N}{2}\Delta_k \mathbb{P}_{\nu\pi}(A) + \frac{N}{2}(\lambda - \Delta_k)\mathbb{P}_{\nu'\pi}(A^c) \geq \frac{N}{2}\min\{\Delta_k, \lambda - \Delta_k\}(\mathbb{P}_{\nu\pi}(A) + \mathbb{P}_{\nu'\pi}(A^c)) \\
&\geq \frac{N}{4}\min\{\Delta_k, \lambda - \Delta_k\}\exp\left(-\mathbb{D}_{\text{KL}}(\mathbb{P}_{\nu\pi}, \mathbb{P}_{\nu'\pi})\right) \\
&= \frac{N}{4}\min\{\Delta_k, \lambda - \Delta_k\} \times \exp\left(-\mathbb{E}_{\nu\pi}\left[\sum_{j \in \mathcal{A}} n_j^T(k)\right] \mathbb{D}_{\text{KL}}(P_k, P'_k)\right).
\end{aligned} \tag{34}$$

Rearranging (34) and taking limit inferior, we have

$$\begin{aligned}\liminf_{N\to\infty}\frac{\mathbb{E}_{\nu\pi}\left[\sum_{j\in\mathcal{A}} n_j^T(k)\right]}{\log(N)} &\geq \frac{1}{\mathbb{D}_{\mathrm{KL}}(P_k,P_k')}\liminf_{N\to\infty}\frac{\log(\frac{N\min\{\Delta_k,\lambda-\Delta_k\}}{R+R'})}{\log(N)}\\ &\geq \frac{1}{\mathbb{D}_{\mathrm{KL}}(P_k,P_k')}\left(1-\limsup_{N\to\infty}\frac{\log(R+R')}{\log(N)}\right).\end{aligned}$$

By the fact that π is consistent, we have some constants $\sigma>0$ and C_σ,

$$\liminf_{N\to\infty}\frac{\mathbb{E}_{\nu\pi}\left[\sum_{j\in\mathcal{A}} n_j^T(k)\right]}{\log(N)} \geq \frac{1}{\mathbb{D}_{\mathrm{KL}}(P_k,P_k')}\left(1-\limsup_{N\to\infty}\frac{a\log N+\log C_\sigma}{\log(N)}\right). \tag{35}$$

Plugging (35) into the definition of group regret, we have

$$\liminf_{N\to\infty}\frac{R}{\log(N)} \geq \liminf_{N\to\infty}\frac{\sum_i \mathbb{E}_{\nu\pi}\left[\sum_{j\in\mathcal{A}} n_j^T(k)\right]\Delta_i}{\log(N)} \geq \sum_i \frac{\Delta_i}{\mathbb{D}_{\mathrm{KL}}(P_i,P_i')}.$$

This completes the proof.

D PROOF OF THEOREM 2

D.1 Proof of Theorem 2(a)

We claim that, according to the rules of ODC, an agent $j\in\mathcal{A}$ would not send in total more than

$$\min\{C_j, C_{j'}\}+1$$

messages to agent $j'\in\mathcal{A}\setminus\{j\}$, where

$$C_j=\max\Big\{C\in\{1,...,N_j\}:\Big(\sum_{c=1}^{C} f(c)\Big)\leq N_j\Big\},$$

upper bounds the number of times agent j can fulfill the buffer thresholds when buffer thresholds are updated according to a monotonically increasing function f.

Suppose $C_j\leq C_{j'}$. Under ODC, the number of observations in buffer $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i)$ must be greater than or equal to the buffer threshold $f(c)$ when (right before) agent j sends the c-th message to agent j'. Hence, agent j can send in total at most C_j messages to agent j' when $C_j\leq C_{j'}$. Because if agent j sends in total more than C_j messages, e.g., C_j+1 messages, to agent j', that means at least one message transmission violates the rule of ODC as $\left(\sum_{c=1}^{C_j+1} f(c)\right)>N_j$.

Suppose $C_j>C_{j'}$. Under ODC, the exchange demand $E^{j\to j'}$ must be `true` when (right before) an agent j sends a message to agent j'. According to the rules of ODC, the exchange demand $E^{j\to j'}$ is set to `true` when: 1) during algorithm initialization, 2) agent j' sends agent j a message. Agent j' can send agent j at most $C_{j'}$ messages when $C_j>C_{j'}$ for otherwise it must violate the buffer thresholds rule of ODC. Hence, $E^{j\to j'}$ is set to `true` at most $C_{j'}+1$ times. Then, agent j can send agent j' in total at most $C_{j'}+1$ messages if agent j follows the exchange demand rule of ODC.

Now take into account the communications between all pairs of agents, we have the communication complexity:

$$C\leq \sum\nolimits_{j\in\mathcal{A}}\sum\nolimits_{j'\in\mathcal{A}\setminus\{j\}}\min\{C_j,C_{j'}\}+1.$$

This completes the proof.

D.2 Proof of Theorem 2(b)

We first upper bound the expected total numbers of Type-I decisions and Type-II decisions made by an agent before AAE-ODC stops communication, and then we follow similar steps in previous subsection to upper bound the communication complexity of AAE-ODC.

Note that agent j makes a Type-I decision if (6) holds, otherwise it is a Type-II decision. By Lemma 1, with $\delta_j^t = 1/N^2$, the expected number of Type-II decisions made by an agent in the entire learning horizon is upper bounded by $2K$. By Lemma 3, with $\delta_j^t = 1/N^2$, the expected number of Type-I decisions made by an agent j before the candidate set size reduces to one can be upper bounded by

$$\sum_{i \in \mathcal{K}} \frac{16\alpha \log N}{\max\{\Delta_i^2, \Delta^2\}}.$$

Note that, N_j is the total number of decisions made by agent j. Hence, the expected number of decisions an agent j makes before agents stop communicate with one another can be upper bounded by

$$\min\Big\{2K + \sum_{i \in \mathcal{K}} \frac{16\alpha \log N}{\max\{\Delta_i^2, \Delta^2\}}, N_j\Big\}.$$

Then, we claim that, under `AAE-ODC`, an agent $j \in \mathcal{A}$ would not send in total more than

$$\min\{C_j, C_{j'}\} + 1$$

messages to agent $j' \in \mathcal{A} \setminus \{j\}$, where

$$C_j = \max\Big\{C \in \{1, ..., N_j\} : \Big(\sum_{c=1}^{C} f(c)\Big) \leq \min\Big\{2K + \sum_{i \in \mathcal{K}} \frac{16\alpha \log N}{\max\{\Delta_i^2, \Delta^2\}}, N_j\Big\}\Big\},$$

upper bounds the number of times agent j executing `AAE-ODC` can fulfill the buffer thresholds when buffer thresholds are updated according to a monotonically increasing function f.

Suppose $C_j \leq C_{j'}$. Under `ODC`, the number of observations in buffer $\sum_{i \in \mathcal{K}} b_n^{j \to j'}(i)$ must be greater than or equal to the buffer threshold $f(c)$ when (right before) agent j sends the c-th message to agent j'. Hence, agent j executing `AAE-ODC` can send in total at most C_j messages to agent j' when $C_j \leq C_{j'}$. Because if agent j sends in total more than C_j messages, e.g., $C_j + 1$ messages, to agent j', that means at least one message transmission violates the rule of `ODC` or `AAE-ODC` as $\Big(\sum_{c=1}^{C_j+1} f(c)\Big) > \min\Big\{2K + \sum_{i \in \mathcal{K}} \frac{16\alpha \log N}{\max\{\Delta_i^2, \Delta^2\}}, N_j\Big\}$.

Suppose $C_j > C_{j'}$. Under `ODC`, the exchange demand $E^{j \to j'}$ must be `true` when (right before) an agent j sends a message to agent j'. According to the rules of `ODC`, the exchange demand $E^{j \to j'}$ is set to `true` when: 1) during algorithm initialization, 2) agent j' sends agent j a message. Agent j' can send agent j at most $C_{j'}$ messages when $C_j > C_{j'}$ for otherwise it must violate the buffer thresholds rule of `ODC`. Hence, $E^{j \to j'}$ is set to `true` at most $C_{j'} + 1$ times. Then, agent j can send agent j' in total at most $C_{j'} + 1$ messages if agent j follows the exchange demand rule of `ODC`.

Now take into account the communications between all pairs of agents, we have the communication complexity:

$$C \leq \sum\nolimits_{j \in \mathcal{A}} \sum\nolimits_{j' \in \mathcal{A} \setminus \{j\}} \min\{C_j, C_{j'}\} + 1.$$

This completes the proof.

E PSEUDO CODE OF `UCB-ODC`

We present `UCB-ODC` in Algorithm 2.

Algorithm 2 The UCB-ODC Algorithm for Agent j

Initialize: exchange demands $E^{j\to j'} \leftarrow$ `True`, $\forall j' \in \mathcal{A}\setminus\{j\}$, buffers $b_n^{j\to j'}(i) \leftarrow 0$, $b_\mu^{j\to j'}(i) \leftarrow 0$, $\forall j' \in \mathcal{A}\setminus\{j\}, i \in \mathcal{K}$, number of communications $c^{j\to j'} \leftarrow 1$, $\forall j' \in \mathcal{A}\setminus\{j\}$, buffer thresholds $f(c^{j\to j'}) \leftarrow f(1)$, $\forall j' \in \mathcal{A}\setminus\{j\}$, UCB parameters $\hat{n}_j(i) = 0$, $\hat{\mu}_j(i) = 0$, $\forall i \in \mathcal{K}$, $n_j = 0$, $\delta_j^t = 1/n_j$, $\alpha \geq 2$
for $t = 1...T$ **do**
  if t is a decision time slot of agent j, i.e., $t \in \{t_1^j, ..., t_{N_j}^j\}$ **then**
    Pull arm I_t^j with highest UCB, i.e., $I_t^j \equiv \arg\max_{i\in\mathcal{K}} \hat{\mu}(i) + \mathtt{CI}_j^t(i)$, and receive instantaneous reward $x_t(I_t^j)$
    Increase $\hat{n}_j(I_t^j)$ and n_j by 1, and update the empirical mean value, $\hat{\mu}(I_t^j)$, with instantaneous reward $x_t(I_t^j)$
    Reconstruct the UCBs based on the updated values of $\hat{n}_j(I_t)$, n_j, and $\hat{\mu}_j(I_t^j)$ by using Equation (1)
    for each agent $j' \in \mathcal{A}\setminus\{j\}$ **do**
      Update the buffer for agent j': $b_n^{j\to j'}(I_t^j) \leftarrow b_n^{j\to j'}(I_t^j) + 1$, $b_\mu^{j\to j'}(I_t^j) \leftarrow b_\mu^{j\to j'}(I_t^j) + x_t(I_t^j)$
      if $E^{j\to j'}$ is `True` and $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
        Share the buffered information with j', i.e., send a message as defined in Definition 1, Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$
        Set exchange demand $E^{j\to j'} \leftarrow$ `False` and renew the buffer for agent j'
        Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
      end if
    end for
  end if
  for each new message received from any agent $j' \in \mathcal{A}\setminus\{j\}$ **do**
    Increase $\hat{n}_j(i), \forall i \in \mathcal{K}$ and update empirical means, $\hat{\mu}_j(i), \forall i \in \mathcal{K}$, according to the information in the message
    Execute Line (6) to reconstruct UCBs
    if agent j has buffered $f(c^{j\to j'})$ observations for j', i.e., $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
      Share information by sending a message as defined in Definition 1 to j', Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$, renew buffer for j'
      Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
    else
      Set exchange demand $E^{j\to j'} \leftarrow$ `True`
    end if
  end for
end for

F PSEUDO CODE OF AAE-ODC

We present AAE-ODC in Algorithm 3.

Algorithm 3 The AAE-ODC Algorithm for Agent j

Initialize: exchange demands $E^{j\to j'} \leftarrow$ True, $\forall j' \in \mathcal{A}\setminus\{j\}$, buffers $b_n^{j\to j'}(i) \leftarrow 0$, $b_\mu^{j\to j'}(i) \leftarrow 0$, $\forall j' \in \mathcal{A}\setminus\{j\}, i \in \mathcal{K}$, number of communications $c^{j\to j'} \leftarrow 1$, $\forall j' \in \mathcal{A}\setminus\{j\}$, buffer thresholds $f(c^{j\to j'}) \leftarrow f(1), \forall j' \in \mathcal{A}\setminus\{j\}$, AAE parameters $\hat{n}_j(i) = 0$, $\hat{\mu}_j(i) = 0, \forall i \in \mathcal{K}$, $n_j = 0$, $\delta_j^t = 1/n_j$, $\alpha \geq 2$, self candidate set $\mathcal{C} = \{1, 2, ..., K\}$
for $t = 1...T$ **do**
  if t is a decision time slot of agent j, i.e., $t \in \{t_1^j, ..., t_{N_j}^j\}$ **then**
    Recompute confidence intervals $\mathtt{CI}_j^t(i), \forall i \in \mathcal{K}$ as defined in (1)
    for $i \in \mathcal{C}$ **do**
      if $|\mathcal{C}| > 1$ and $\exists i' \in \mathcal{K}$ s.t. $\hat{\mu}_j^t(i) + \mathtt{CI}_j^t(i) < \hat{\mu}_j^t(i') - \mathtt{CI}_j^t(i')$ **then**
        Eliminate arm i from the candidate set, i.e., $\mathcal{C} \leftarrow \mathcal{C}\setminus\{i\}$
        Broadcast index of arm i to all agents $j \in \mathcal{A}$
      end if
    end for
    Pull arm I_t^j from the candidate set $\mathcal{C}$ with the least observations, and receive instantaneous reward $x_t(I_t^j)$
    Increase $\hat{n}_j(I_t^j)$ and n_j by 1, and update empirical mean, $\hat{\mu}(I_t^j)$, with instantaneous reward $x_t(I_t^j)$
    if $|\mathcal{C}| > 1$ **then**
      for each agent $j' \in \mathcal{A}\setminus\{j\}$ **do**
        Update the buffer for agent j': $b_n^{j\to j'}(I_t^j) \leftarrow b_n^{j\to j'}(I_t^j) + 1$, $b_\mu^{j\to j'}(I_t^j) \leftarrow b_\mu^{j\to j'}(I_t^j) + x_t(I_t^j)$
        if $E^{j\to j'}$ is True and $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
          Share the buffered information with j', i.e., send a message as defined in Definition 1, Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$
          Set exchange demand $E^{j\to j'} \leftarrow$ False and renew the buffer for agent j'
          Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
        end if
      end for
    end if
  end if
  for each new message received from any agent $j' \in \mathcal{A}\setminus\{j\}$ **do**
    if it is an elimination notice of arm i **then**
      Eliminate arm i from the candidate set, i.e., $\mathcal{C} \leftarrow \mathcal{C}\setminus\{i\}$
    else
      Increase $\hat{n}_j(i), \forall i \in \mathcal{K}$ and update empirical means, $\hat{\mu}_j(i), \forall i \in \mathcal{K}$, according to the information in the message
      if agent j has buffered $f(c^{j\to j'})$ observations for j', i.e., $\sum_{i\in\mathcal{K}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
        Share information by sending a message as defined in Definition 1 to j', Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$, renew buffer for j'
        Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
      else
        Set exchange demand $E^{j\to j'} \leftarrow$ True
      end if
    end if
  end for
end for

G ACCOUNTING FOR COMMUNICATION DELAY

Suppose message transmission between agents suffers a deterministic delay, d. In the following, we discuss how the communication delays affect the group regrets and communication complexities of UCB-ODC and AAE-ODC.

For UCB-ODC (resp. AAE-ODC), we consider time slot τ_i for each suboptimal arm i such that (9) (resp. (22)) holds, and consider agent j such that, at time τ_i, (13) (resp. (24)) holds; by Lemma 2 (resp. Lemma 3), agent j makes Type-I decisions to pull arm i after time τ_i. In the following, we upper bound the extra number of times (under deterministic communication delays) agent j pulls arm i to make up for the delayed transmission of observations from other agents.

Recall that $B_t^{j\to j'}(i)$ denotes the number of reward samples of arm i stored in agent j's buffer for agent j' (and not yet been sent) at time t, and $B_t^{j\to j'}$ denotes the total number of observations stored in agent j's buffer for agent j' at time t. For an agent $j' \in \mathcal{A}\setminus\{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{false}$, if $B_{\tau_i}^{j'\to j}(i) < f(c_{\tau_i}^{j'\to j})$, agent j has to make at most $f(c_{\tau_i}^{j'\to j})$ extra pulls of i to make up for agent j''s delay. If $B_{\tau_i}^{j'\to j}(i) \geq f(c_{\tau_i}^{j'\to j})$, agent j can send a message to j' once agent j buffers $f(c_{\tau_i}^{j\to j'})$ observations for j'; the message takes d time slots to reach agent j', and the reply from agent j' with the outstanding observations takes d time slots to reach agent j. During the $2d$ time slots, agent j makes at most $2d$ pulls on arm i. Hence, because of the delayed transmission from agents $j' \in \mathcal{A}\setminus\{j\} : E_{\tau_i}^{j'\to j} = \texttt{false}$, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}\setminus\{j\}} 2d + f(\max\{c_{\tau_i}^{j'\to j}, c_{\tau_i}^{j\to j'}\})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}} \leq \sum_{j'\in\mathcal{A}\setminus\{j\}} 2d + f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}}, \quad (36)$$

where the inequality is because, by the definition of the ODC, for any pair of agents $j, j' \in \mathcal{A}$ at any time t, if $E_t^{j'\to j} = \texttt{false}$, $1 \geq c_t^{j'\to j} - c_t^{j\to j'} \geq 0$. On the other hand, agents $j' \in \mathcal{A}\setminus\{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{true}$ delay transmission of $\sum_{j'\in\mathcal{A}\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}$ observations of i to agent j at time τ_i due to waiting for the buffer thresholds to be satisfied. To make up for this type of delay, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}. \quad (37)$$

Therefore, the upper bound of expected group regret of UCB-ODC under deterministic communication delay d has the same form as (2) in Theorem 1(a) but with F_i^j defined as follows:

$$F_i^j = \min\Big\{\Big(\sum_{j'\in\mathcal{A}\setminus\{j\}} 2d + f(c_{\tau_i}^{j'\to j})\Big), \frac{2\alpha\log N}{\Delta_i^2}\Big\}. \quad (38)$$

The upper bound of expected group regret of AAE-ODC under deterministic communication delay d has the same form as (3) in Theorem 1(b) but with G_i^j defined as follows:

$$G_i^j = \min\Big\{\Big(\sum_{j'\in\mathcal{A}\setminus\{j\}} 2d + f(c_{\tau_i}^{j'\to j})\Big), \frac{16\alpha\log N}{\Delta_i^2}\Big\}. \quad (39)$$

Under ODC, any agent j needs the exchange demand $E^{j\to j'}$ to be set to true to be allowed to send a message to another agent j'. Communication delay would never increase the number of times exchange demands be set to true. Hence, the communication complexity upper bounds in Theorem 2(a) and Theorem 2(b) still hold for UCB-ODC and AAE-ODC respectively under deterministic communication delay d.

H ACCOUNTING FOR HETEROGENEOUS ARM SETS

Agents having different but overlapping arm sets is a practical scenario in MAMAB. In the following, we discuss how to generalize UCB-ODC and AAE-ODC to account for heterogeneous arm sets.

H.1 Model Formulation

We need additional notations for formulating heterogeneous arm set scenario. In this scenario, agents receive the same expected rewards from the same arms but each agent only has access to a *local* subset of the K arms, as in Yang et al. (2022); Chawla et al. (2020); Yang et al. (2021). Specifically, agent $j \in \mathcal{A}$ has access to a subset of arms $\mathcal{K}_j \subseteq \mathcal{K}$ *known* to every agent. We refer to arms in $\mathcal{K}_j$ as *local* arms of agent j. Let $K_j = |\mathcal{K}_j|$. Without loss of generality, we assume that at least two arm sets overlap; i.e., $\exists j, j' \in \mathcal{A}$ s.t. $\mathcal{K}_j \cap \mathcal{K}_{j'} \neq \emptyset$.

Let $i_j^* = \arg\max_{i \in \mathcal{K}_j} \mu_i$ denote the local optimal arm of agent j. Let $\mathcal{A}_i$ denote the set of agents whose local arm set includes arm i, i.e., $\mathcal{A}_i \equiv \{j \in \mathcal{A} : i \in \mathcal{K}_j\}$. Let $\mathcal{A}_i^*$ denote the set of agents whose local optimal arm is i, i.e., $\mathcal{A}_i^* \equiv \{j \in \mathcal{A}_i : i = i_j^*\}$ and let $\mathcal{A}_{-i}^* = \mathcal{A}_i \setminus \mathcal{A}_i^*$. Note that $\mathcal{A}_i^*$ or $\mathcal{A}_{-i}^*$ may be empty. Let $\mathcal{A}^{(j)}$ denote the set of agents that share arms with agent j, i.e., $\mathcal{A}^{(j)} \equiv \cup_{i \in \mathcal{K}_j} \mathcal{A}_i \setminus \{j\}$. Let $M_i = |\mathcal{A}_i|$, $M_i^* = |\mathcal{A}_i^*|$, $M_{-i}^* = |\mathcal{A}_{-i}^*|$, and $M^{(j)} = |\mathcal{A}^{(j)}|$. Let $\Delta(i_j^*, i)$ denote the suboptimality gap of arm i in agent j's local arm set $\mathcal{K}_j$, $i \in \mathcal{K}_j$. We further denote the smallest suboptimality gap of arm i as $\tilde{\Delta}_i$ and denote the agent that contains $\tilde{\Delta}_i$ as $\tilde{j}_i$, i.e.,

$$\tilde{\Delta}_i \equiv \begin{cases} \min_{j \in \mathcal{A}_{-i}^*} \Delta(i_j^*, i), & \mathcal{A}_{-i}^* \neq \emptyset, \\ 0, & \text{otherwise}, \end{cases} \quad \text{and} \quad \tilde{j}_i \equiv \begin{cases} \arg\min_{j \in \mathcal{A}_{-i}^*} \Delta(i_j^*, i), & \mathcal{A}_{-i}^* \neq \emptyset, \\ 0, & \text{otherwise}. \end{cases}$$

The expected cumulative regret of each agent j becomes

$$\mathbb{E}[R_{N_j}^j] = \mu(i_j^*) N_j - \mathbb{E}[\sum\nolimits_{t \in \{t_1^j, t_2^j, \ldots, t_{N_j}^j\}} x_t(I_t^j)].$$

Expected group regret is $\mathbb{E}[R] = \sum_{j \in \mathcal{A}} \mathbb{E}[R_{N_j}^j]$.

H.2 Algorithm

We present the extension of `UCB-ODC` in Algorithm 4.

Algorithm 4 The `UCB-ODC` Algorithm for Agent j (with heterogeneous arm sets)

Input: other agents' local arm sets $(\mathcal{K}_1, ..., \mathcal{K}_M)$
Initialize: exchange demands $E^{j\to j'} \leftarrow$ `True`, $\forall j' \in \mathcal{A}^{(j)}$, buffers $b_n^{j\to j'}(i) \leftarrow 0$, $b_\mu^{j\to j'}(i) \leftarrow 0$, $\forall j' \in \mathcal{A}^{(j)}, i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$, number of communications $c^{j\to j'} \leftarrow 1$, $\forall j' \in \mathcal{A}^{(j)}$, buffer thresholds $f(c^{j\to j'}) \leftarrow f(1)$, $\forall j' \in \mathcal{A}^{(j)}$, `UCB` parameters $\hat{n}_j(i) = 0$, $\hat{\mu}_j(i) = 0$, $\forall i \in \mathcal{K}_j$, $n_j = 0$, $\delta_j^t = 1/n_j$, $\alpha \geq 2$
for $t = 1...T$ **do**
  if t is a decision time slot of agent j, i.e., $t \in \{t_1^j, ..., t_{N_j}^j\}$ **then**
    Pull arm I_t^j with highest UCB, i.e., $I_t^j \equiv \arg\max_{i\in\mathcal{K}_j} \hat{\mu}(i) + \texttt{CI}_j^t(i)$, and receive instantaneous reward $x_t(I_t^j)$
    Increase $\hat{n}_j(I_t^j)$ and n_j by 1, and update the empirical mean value, $\hat{\mu}(I_t^j)$, with instantaneous reward $x_t(I_t^j)$
    Reconstruct the UCBs based on the updated values of $\hat{n}_j(I_t)$, n_j, and $\hat{\mu}_j(I_t^j)$ by using Equation (1)
    for each agent $j' \in \mathcal{A}_{I_t^j}$ **do**
      Update the buffer for agent j': $b_n^{j\to j'}(I_t^j) \leftarrow b_n^{j\to j'}(I_t^j) + 1$, $b_\mu^{j\to j'}(I_t^j) \leftarrow b_\mu^{j\to j'}(I_t^j) + x_t(I_t^j)$
      if $E^{j\to j'}$ is `True` and $\sum_{i\in\mathcal{K}_j\cap\mathcal{K}_{j'}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
        Share the buffered information with j', i.e., send a message as defined in Definition 1, Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$
        Set exchange demand $E^{j\to j'} \leftarrow$ `False` and renew the buffer for agent j'
        Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
      end if
    end for
  end if
  for each new message received from any agent $j' \in \mathcal{A}^{(j)}$ **do**
    Increase $\hat{n}_j(i), \forall i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$ and update empirical means, $\hat{\mu}_j(i), \forall i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$, according to the message
    Execute Line (7) to reconstruct UCBs
    if agent j has buffered $f(c^{j\to j'})$ observations for j', i.e., $\sum_{i\in\mathcal{K}_j\cap\mathcal{K}_{j'}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
      Share information by sending a message as defined in Definition 1 to j', Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$, renew buffer for j'
      Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
    else
      Set exchange demand $E^{j\to j'} \leftarrow$ `True`
    end if
  end for
end for

We present the extension of AAE-ODC in Algorithm 5. Note that, in heterogeneous arm sets setting, each agent needs to maintain a *candidate set*; two agents stop communicating once both of their candidate set sizes reduce one.

Algorithm 5 The AAE-ODC Algorithm for Agent j (with heterogeneous arm sets)

Input: Other agents' local arm sets $(\mathcal{K}_1, ..., \mathcal{K}_M)$
Initialize: exchange demands $E^{j\to j'} \leftarrow$ True, $\forall j' \in \mathcal{A}^{(j)}$, buffers $b_n^{j\to j'}(i) \leftarrow 0$, $b_\mu^{j\to j'}(i) \leftarrow 0$, $\forall j' \in \mathcal{A}^{(j)}, i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$, number of communications $c^{j\to j'} \leftarrow 1$, $\forall j' \in \mathcal{A}^{(j)}$, buffer thresholds $f(c^{j\to j'}) \leftarrow f(1)$, $\forall j' \in \mathcal{A}^{(j)}$, AAE parameters $\hat{n}_j(i) = 0$, $\hat{\mu}_j(i) = 0$, $\forall i \in \mathcal{K}_j$, $n_j = 0$, $\delta_j^t = 1/n_j$, $\alpha \geq 2$, candidate sets $\mathcal{C}_j = \mathcal{K}_j$ and $\mathcal{C}_{j'} = \mathcal{K}_{j'}, \forall j' \in \mathcal{A}^{(j)}$
for $t = 1...T$ **do**
  if t is a decision time slot of agent j, i.e., $t \in \{t_1^j, ..., t_{N_j}^j\}$ **then**
    Recompute confidence intervals $\mathtt{CI}_j^t(i), \forall i \in \mathcal{K}_j$ as defined in (1)
    for $i \in \mathcal{C}_j$ **do**
      if $|\mathcal{C}_j| > 1$ and $\exists i' \in \mathcal{K}_j$ s.t. $\hat{\mu}_j^t(i) + \mathtt{CI}_j^t(i) < \hat{\mu}_j^t(i') - \mathtt{CI}_j^t(i')$ **then**
        Eliminate arm i from the candidate set, i.e., $\mathcal{C}_j \leftarrow \mathcal{C}_j \setminus \{i\}$
        Broadcast index of arm i to all agents $j \in \mathcal{A}_i$
      end if
    end for
    Pull arm I_t^j from the candidate set $\mathcal{C}_j$ with the least observations, and receive instantaneous reward $x_t(I_t^j)$
    Increase $\hat{n}_j(I_t^j)$ and n_j by 1, and update empirical mean, $\hat{\mu}(I_t^j)$, with instantaneous reward $x_t(I_t^j)$
    for each agent $j' \in \mathcal{A}_{I_t^j}$ that $|\mathcal{C}_{j'}| > 1$ or $|\mathcal{C}_j| > 1$ **do**
      Update the buffer for agent j': $b_n^{j\to j'}(I_t^j) \leftarrow b_n^{j\to j'}(I_t^j) + 1$, $b_\mu^{j\to j'}(I_t^j) \leftarrow b_\mu^{j\to j'}(I_t^j) + x_t(I_t^j)$
      if $E^{j\to j'}$ is True and $\sum_{i\in\mathcal{K}_j\cap\mathcal{K}_{j'}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
        Share the buffered information with j', i.e., send a message as defined in Definition 1, Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$
        Set exchange demand $E^{j\to j'} \leftarrow$ False and renew the buffer for agent j'
        Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
      end if
    end for
  end if
  for each new message received from any agent $j' \in \mathcal{A}^{(j)}$ **do**
    if it is an elimination notice of arm i from agent j' **then**
      Eliminate arm i from the candidate set, i.e., $\mathcal{C}_{j'} \leftarrow \mathcal{C}_{j'} \setminus \{i\}$
    else
      Increase $\hat{n}_j(i), \forall i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$ and update empirical means, $\hat{\mu}_j(i), \forall i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$, according to the message
      if agent j has buffered $f(c^{j\to j'})$ observations for j', i.e., $\sum_{i\in\mathcal{K}_j\cap\mathcal{K}_{j'}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$ **then**
        Share information by sending a message as defined in Definition 1 to j', Set $c^{j\to j'} \leftarrow c^{j\to j'} + 1$, renew buffer for j'
        Update buffer threshold $f(c^{j\to j'})$, e.g., double it $f(c^{j\to j'}) \leftarrow 2f(c^{j\to j'} - 1)$ or keep it the same
      else
        Set exchange demand $E^{j\to j'} \leftarrow$ True
      end if
    end if
  end for
end for

H.3 Analysis of Regret and Communication Complexity

Expected Group Regret of `UCB-ODC` under Heterogeneous Arm Sets. With algorithm parameters $\delta_j^t = 1/N$ and $\alpha \geq 3$, the expected group regret of *`UCB-ODC`* under heterogeneous arm sets satisfies

$$\mathbb{E}[R] \leq 3KM + \sum_{i \in \mathcal{K}: \tilde{\Delta}_i > 0} \left(\frac{4\alpha \log N}{\tilde{\Delta}_i} + \sum_{j \in \mathcal{A}_{-i}^*} \min \left\{ \Big(\sum_{j' \in \mathcal{A}_i \setminus \{j\}} f(c_{\tau_i}^{j' \to j}) \Big), \frac{2\alpha \log N}{\Delta^2(i_j^*, i)} \right\} \Delta(i_j^*, i) \right). \tag{40}$$

Recall that $\mathcal{A}_{-i}^*$ is the set of agent with arm i as a local suboptimal arm. Following similar arguments in the proof of Lemma 2, if agent $j \in \mathcal{A}_{-i}^*$ makes a Type-I decision and pulls arm $i \in \mathcal{K}_j$ by `UCB-ODC` algorithm at time t, we have that

$$\hat{n}_j^t(i) \leq \frac{2\alpha \log(1/\delta_j^t)}{\Delta^2(i_j^*, i)}. \tag{41}$$

Without loss of generality, we let $\mathcal{A}_{-i}^* = \{j_m : m = 1, 2, ..., M_{-i}\}$, where $\Delta(i_{j_1}^*, i) \geq \Delta(i_{j_2}^*.i) \geq \cdots \geq \Delta(i_{j_{M_{-i}}}^*, i)$ and $M_{-i} = |\mathcal{A}_{-i}^*|$. Agent $j_{M_{-i}}$ needs the most number of observations of arm i to differentiate it from its local optimal arm because $j_{M_{-i}}$ is agent with the smallest $\Delta(i_j^*, i)$ among all $j \in \mathcal{A}_{-i}^*$. Though $j_{M_{-i}}$ is the agent in $\mathcal{A}_{-i}^*$ that needs the most number of observations of arm i, each time agent $j_{M_{-i}}$ pulls arm i in fact incur the smallest regret than each time other agents in $\mathcal{A}_{-i}^*$ pull arm i because it has the smallest $\Delta(i_j^*, i)$ among all $j \in \mathcal{A}_{-i}^*$. When those agents $j_m \in \mathcal{A}_{-i}^*$ with largest $\Delta(i_{j_m}^*, i)$s make the most number of pulls of arm i, the largest regret on arm i is incurred with the same number of times arm i being pulled. With $\delta_j^t \geq 1/N, \forall j \in \mathcal{A}_{-i}$, and $A_m = \frac{2\alpha \log N}{\Delta^2(i_{j_m}^*, i)}$, we have

$$A_1 \Delta(i_{j_1}^*, i) + \sum_{m=1}^{M_{-i}-1} (A_{m+1} - A_m) \Delta(i_{j_{m+1}}^*, i) \tag{42}$$

$$= \sum_{m=1}^{M_{-i}-1} A_m (\Delta(i_{j_m}^*, i) - \Delta(i_{j_{m+1}}^*, i)) + A_{M_{-i}} \Delta(i_{j_{M_{-i}}}^*, i) \tag{43}$$

$$\leq \int_{\Delta(i_{j_{M_{-i}}}^*, i)}^{\Delta(i_{j_1}^*, i)} \frac{2\alpha \log N}{z^2} dz + \frac{2\alpha \log N}{\Delta(i_{j_{M_{-i}}}^*, i)} \leq \frac{4\alpha \log N}{\Delta(i_{j_{M_{-i}}}^*, i)} = \frac{4\alpha \log N}{\tilde{\Delta}_i}. \tag{44}$$

Consider time slot τ_i for each suboptimal arm i such that

$$\frac{2\alpha \log N}{\tilde{\Delta}_i^2} + M \geq \sum_{j' \in \mathcal{A}_i} n_{j'}^{\tau_i}(i) > \frac{2\alpha \log N}{\tilde{\Delta}_i^2} \geq \sum_{j' \in \mathcal{A}_i} n_{j'}^{\tau_i - 1}(i).$$

Consider agent $j \in \mathcal{A}_{-i}$ such that, at time τ_i,

$$\frac{2\alpha \log N}{\tilde{\Delta}_i^2} \geq \frac{2\alpha \log 1/\delta_j^{\tau_i}}{\Delta^2(i_j^*, i)} \geq \hat{n}_j^{\tau_i}(i) = n_j^{\tau_i}(i) + \sum_{j' \in \mathcal{A}_i \setminus \{j\}} n_{j'}^{\tau_i}(i) - B_{\tau_i}^{j' \to j}(i) \tag{45}$$

$$\geq \frac{2\alpha \log N}{\tilde{\Delta}_i^2} - \sum_{j' \in \mathcal{A}_i \setminus \{j\}} B_{\tau_i}^{j' \to j}(i) \mathbb{1}_{E_{\tau_i}^{j' \to j} = \texttt{false}} - \sum_{j' \in \mathcal{A}_i \setminus \{j\}} f(c_{\tau_i}^{j' \to j}) \mathbb{1}_{E_{\tau_i}^{j' \to j} = \texttt{true}}, \tag{46}$$

where $B_t^{j \to j'}(i)$ denotes the number of reward samples of arm i stored in agent j's buffer for agent j' (and not yet been sent) at time t; $B_t^{j \to j'}$ denotes the total number of observations stored in agent j's buffer for agent j'. By (41), such agent $j \in \mathcal{A}_{-i}$ makes Type-I decisions to pull arm i after time τ_i.

In the following, we bound the extra number of times agent $j \in \mathcal{A}_{-i}$ pulls arm i to make up for the delayed transmission from other agents $j' \in \mathcal{A}_i$. For an agent $j' \in \mathcal{A}_i \setminus \{j\}$ such that $E_{\tau_i}^{j' \to j} = \texttt{false}$, if $B_{\tau_i}^{j' \to j}(i) < f(c_{\tau_i}^{j' \to j})$, agent j has to make at most $f(c_{\tau_i}^{j' \to j})$ extra pulls of i to

make up for agent j''s delay; if $B_{\tau_i}^{j'\to j}(i) \geq f(c_{\tau_i}^{j'\to j})$, agent j can receive those observations from j' once agent j buffers $f(c_{\tau_i}^{j\to j'})$ observations for j' and sends a message to j'. Hence, because of the delayed transmission from agents $j' \in \mathcal{A}_i \setminus \{j\} : E_{\tau_i}^{j'\to j} = \texttt{false}$, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}_i\setminus\{j\}} f(\max\{c_{\tau_i}^{j'\to j}, c_{\tau_i}^{j\to j'}\})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}} \leq \sum_{j'\in\mathcal{A}_i\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{false}}, \tag{47}$$

where the inequality is because, by the definition of the ODC, for any pair of agents $j, j' \in \mathcal{A}$ at any time t, if $E_t^{j'\to j} = \texttt{false}$, $1 \geq c_t^{j'\to j} - c_t^{j\to j'} \geq 0$. On the other hand, agents $j' \in \mathcal{A}_i \setminus \{j\}$ such that $E_{\tau_i}^{j'\to j} = \texttt{true}$ delay transmission of $\sum_{j'\in\mathcal{A}_i\setminus\{j\}} B_{\tau_i}^{j'\to j}(i)\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}$ observations of i to agent j at time τ_i due to waiting for the buffer thresholds to be satisfied. To make up for this type of delay, agent j pulls arm i after time τ_i at most following number of times:

$$\sum_{j'\in\mathcal{A}_i\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\mathbb{1}_{E_{\tau_i}^{j'\to j}=\texttt{true}}. \tag{48}$$

Hence, agent $j \in \mathcal{A}_{-i}$ incur at most

$$\min\left\{\Big(\sum_{j'\in\mathcal{A}_i\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\Big), \frac{2\alpha\log N}{\Delta^2(i_j^*, i)}\right\}\Delta(i_j^*, i) \tag{49}$$

extra regret by pulling arm i after time τ_i.

As for Type-II decisions, Lemma 1 still holds under heterogeneous arm sets. Thus, the expected regret incurred under Type-II decisions can still be upper bounded by $2KM$.

Combining the regret upper bounds for Type-II and Type-I decisions, we obtain Eq. (40).

Expected Group Regret of AAE-ODC under Heterogeneous Arm Sets. With algorithm parameters $\delta_j^t = 1/N^2$ and $\alpha \geq 3$, the expected group regret of *AAE-ODC* under heterogeneous arm sets satisfies

$$\mathbb{E}[R] \leq (K+2)M + \sum_{i\in\mathcal{K}:\tilde{\Delta}_i>0}\left(\frac{32\alpha\log N}{\tilde{\Delta}_i} + \sum_{j\in\mathcal{A}_{-i}^*}\min\left\{\Big(\sum_{j'\in\mathcal{A}_i\setminus\{j\}} f(c_{\tau_i}^{j'\to j})\Big), \frac{16\alpha\log N}{\Delta^2(i_j^*, i)}\right\}\Delta(i_j^*, i)\right). \tag{50}$$

The analysis of the expected group regret of AAE-ODC under heterogeneous arm sets follows similar steps as the analysis for UCB-ODC.

Communication Complexity of UCB-ODC under Heterogeneous Arm Sets. When buffer thresholds are updated according to a positive and monotonically increasing function f, the communication complexity UCB-ODC under heterogeneous arm sets satisfies:

$$C \leq \sum\nolimits_{j\in\mathcal{A}}\sum\nolimits_{j'\in\mathcal{A}^{(j)}\setminus\{j\}} \min\{C_j, C_{j'}\} + 1, \tag{51}$$

where C_j is the largest integer in set $\{1, ..., N_j\}$ such that $\Big(\sum_{c=1}^{C_j} f(c)\Big) \leq N_j$.

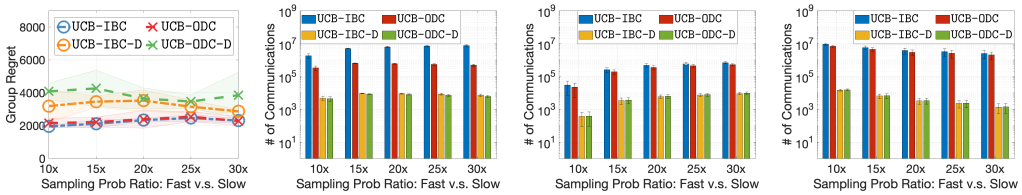
Under ODC, any agent j needs the exchange demand $E^{j\to j'}$ to be set to true to be allowed to send a message to another agent j'. Having heterogeneous arm sets would never increase the number of times exchange demands be set to true. Under heterogeneous arm sets, an agent j may make $f(c^{j\to j'})$ observations but still cannot fulfill the buffer threshold because some of those observations may not be of arm $i \in \mathcal{K}_j \cap \mathcal{K}_{j'}$ and we need $\sum_{i\in\mathcal{K}_j\cap\mathcal{K}_{j'}} b_n^{j\to j'}(i) \geq f(c^{j\to j'})$.

Communication Complexity of AAE-ODC under Heterogeneous Arm Sets. When buffer thresholds are updated according to a positive and monotonically increasing function f, the communication complexity AAE-ODC under heterogeneous arm sets satisfies:

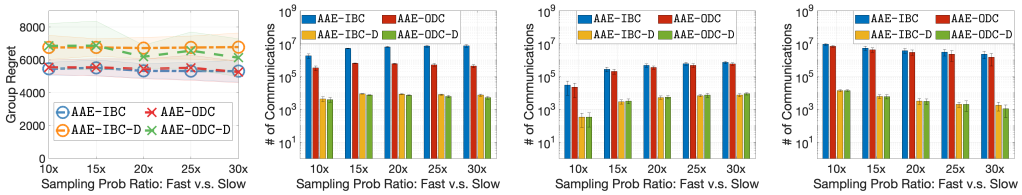
$$C \leq \sum\nolimits_{j\in\mathcal{A}}\sum\nolimits_{j'\in\mathcal{A}^{(j)}\setminus\{j\}} \min\{C_j, C_{j'}\} + 1, \tag{52}$$

where C_j is the largest integer in set $\{1, ..., N_j\}$ such that

$$\Big(\sum_{c=1}^{C_j} f(c)\Big) \leq \min\Big\{2K + \sum_{i\in\mathcal{K}_j}\frac{16\alpha\log N}{\max\{\Delta^2(i_j^*, i), \min_{i\in\mathcal{K}_j\setminus\{i_j^*\}}\Delta^2(i_j^*, i)\}}, N_j\Big\}. \tag{53}$$



(a) UCB Group Regret (b) UCB Comm Between (c) UCB Comm Slow (d) UCB Comm Fast



(e) AAE Group Regret (f) AAE Comm Between (g) AAE Comm Slow (h) AAE Comm Fast

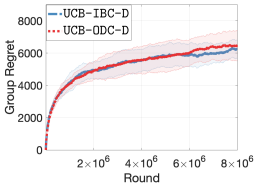
Figure 4: Experiment 1 — impact of the heterogeneity of agent speeds. Comparison between IBC and ODC with buffer thresholds set to one as well as IBC and ODC with buffer thresholds set to be doubling. For communication complexities, we present the numbers of communications between fast and slow agents, among slow agents, and among fast agents separately in different subfigures. Note that, in Subfigures (b)(c)(d) and (f)(g)(h), the Y axis is in Log scale.

I SUPPLEMENTARY EXPERIMENTAL RESULTS

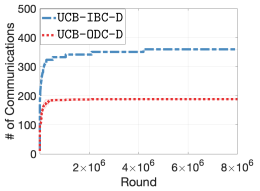
In this section, we present supplementary numerical experimental results to provide more insights about ODC protocol.

I.1 Performance of ODC with constant or doubling buffer thresholds

In the Experiment 1 and Experiment 2 results presented in Section 5, we observe that, when agent pull speeds are highly diversified and when there exist many slow agents, the on-demand rule of ODC saves communication overheads in contrast to IBC while achieving similar group regrets as IBC when both of them have constant buffer thresholds. In Figure 4 and Figure 6, we compare the performance of both IBC and ODC with both constant (size one) buffer thresholds (denoted as AAE-IBC, AAE-ODC, UCB-IBC, UCB-ODC) and doubling buffer thresholds (denoted as AAE-IBC-D, AAE-ODC-D, UCB-IBC-D, UCB-ODC-D) under Experiment 1 and Experiment 2 setups respectively.



(a) UCB Regret

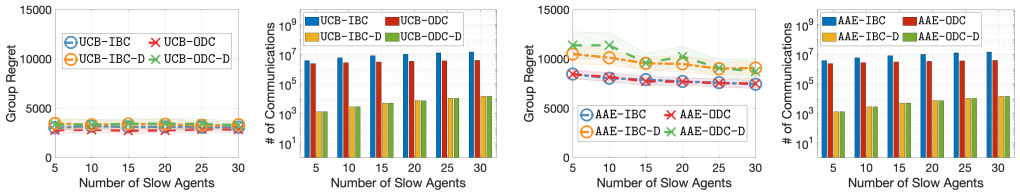


(b) UCB Comm

Figure 5: Experiment 4 — A system with agents that have exponentially large differences in their sampling probabilities

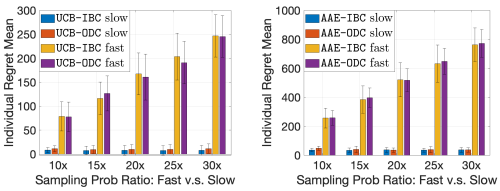
From Figures 4(a), 4(e) and Figures 6(a), 6(c), we observe that, with doubling buffer thresholds, both policies under IBC and under ODC have higher group regrets than those with constant buffer thresholds. From the communication complexities results in Figure 4 and Figure 6, we observe that, with doubling buffer thresholds, both policies under IBC and under ODC incur logarithmic communication overheads than those with constant buffer thresholds. With doubling buffer thresholds, policies under ODC incur slightly smaller communication overheads than policies under IBC but the improvements are not as significant as when their buffer thresholds are all set to be constant. This is because the ratio of sampling probabilities between fast and slow agents are at most 30 times in both experimental setups; hence, when doubling buffer thresholds is applied, the effect of the on-demand rule of ODC is diminished.

The advantage of the on-demand rule of ODC is obvious, even with doubling buffer thresholds applied, when the differences of pull rates are exponentially large, as shown in Experiment 4 (Figure 5). Figure 5 shows the results of simulations of a system with 10 agents, where there is a fast agent with sampling probability set to be always 1 and nine slow agents with sampling probabilities initially set to be 0.1 and halved after each message transmission. We report the cumulative group regret and number of communication over $T = 8,000,000$ rounds. Figure 5(b) shows that UCB-ODC-D



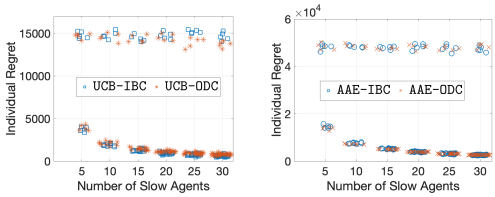
(a) UCB Group Regret (b) UCB Comm (c) AAE Group Regret (d) AAE Comm

Figure 6: Experiment 2 — impact of the number of slow agents in the system. Comparison between IBC and ODC with buffer thresholds set to one as well as IBC and ODC with buffer thresholds set to be doubling. Note that, in Subfigures (b) and (d), the Y axis is in Log scale.



(a) Individual Regret Mean (b) Individual Regret Mean

Figure 7: Experiment 1 — impact of the heterogeneity of agent speeds.



(a) Individual Regret (b) Individual Regret

Figure 8: Experiment 2 — impact of the number of slow agents in the system.

effectively saves communication overheads and Figure 5(a) shows that UCB-ODC-D still achieves similar group regrets as UCB-IBC-D.

I.2 Individual Regrets in Experiment 1 and Experiment 2

In asynchronous MAMAB setting, individual agent's expected regret varies as the pulling times and the total number of decision rounds of the agent, N_j, vary.

For Experiment 1 and Experiment 2 in Section 5, we add Figure 7 and Figure 8 respectively here to provide experimental observations about individual regrets. Specifically, Figure 7 (for Experiment 1) contains two bar charts to present the mean and variance of individual regrets in between *fast* agents and in between *slow* agents after $T = 80,000$ time slots. The height of a bar shows the individual regret mean of agents with same sampling probability and the error bar on each bar denotes mean plus/minus one standard deviation of the individual regrets of agents with same sampling probability. Figure 8 (for Experiment 2) contains two scatter charts on which each dot represents the individual regret of an agent.

Following are the experimental observations about individual regret.

In Experiment 1, we fix the sampling probability of each slow agent and vary the sampling probability of fast agents. In Figure 7, the individual regret mean among slow agents stays almost the same when the difference in sampling probabilities of fast and slow agents increases; the variance of individual regrets among slow agents also stays almost the same. The individual regret mean of fast agents increases as the sampling probability of fast agent increases while the variance of individual regrets among fast agents stays almost the same.

In Experiment 2, we fix the number of fast agents as well as the sampling probability of fast agents and increase the number of slow agents. In Figure 8, dots are clustered into two groups; the five dots with larger individual regrets are of the fast agents, and the other dots with smaller individual regrets are of the slow agents.

Figure 7 and Figure 8 show that ODC achieve similar regret performance as IBC not only in terms of group regret but also in terms of individual regrets.

I.3 Performance of ODC under different types of asynchronicity

In this subsection, we study the performance of ODC under three more variants of asynchronicity, which are different from the stochastic asynchronicity considered in Experiments 1, 2, and 3.

Experiment 5. In this experiment, we study the impact of agents going offline and online, which models wireless sensing devices with sleeping/active modes for power saving. Specifically, there are five slow agents each with sampling probability 0.2 and five fast agents each with sampling probability 0.8. Fast agents, while having high pull rates when they are online, may go offline for a long time. Specifically, fast agents stay online or offline both according to a geometric distribution with parameter 0.01 in this experiment. We report the number of communications and group regret after $T = 80,000$ time slots averaged over 30 independent trials in Table 3.

Table 3: Experiment 5

	Communication	Group Regret
UCB-IBC	$(2.1604 \pm 0.0247) \times 10^6$	2442 ± 267
UCB-ODC	$(1.0119 \pm 0.0135) \times 10^6$	2225 ± 232
AAE-IBC	$(2.1605 \pm 0.0209) \times 10^6$	6788 ± 412
AAE-ODC	$(1.0105 \pm 0.0165) \times 10^6$	6957 ± 446

Experiment 6. In this experiment, we study the impact of less learning horizons overlapping among agents. We have five slow agents each with sampling probability 0.1 and five fast agents each with sampling probability 0.7. In Experiment 6(a), we let the five slow agents go online from the very beginning and let the five fast agents go online at time slot $t = 40,000$. We do the other way around in Experiment 6(b) – we let the five fast agents go online from the very beginning and let the five slow agents go online at time slot $t = 40,000$. We report the number of communications and group regrets after $T = 80,000$ time slots averaged over 30 independent trials in Tables 4 and 5 respectively.

Table 4: Experiment 6(a)

	Communication	Group Regret
UCB-IBC	$(1.6196 \pm 0.0025) \times 10^6$	1931 ± 257
UCB-ODC	$(0.8332 \pm 0.0021) \times 10^6$	2052 ± 228
AAE-IBC	$(1.6199 \pm 0.0024) \times 10^6$	3906 ± 813
AAE-ODC	$(0.8325 \pm 0.0023) \times 10^6$	5021 ± 641

Table 5: Experiment 6(b)

	Communication	Group Regret
UCB-IBC	$(2.7000 \pm 0.0025) \times 10^6$	2803 ± 283
UCB-ODC	$(1.2804 \pm 0.0022) \times 10^6$	2568 ± 285
AAE-IBC	$(2.5073 \pm 0.4978) \times 10^6$	6264 ± 568
AAE-ODC	$(1.2797 \pm 0.0023) \times 10^6$	6741 ± 447

Experiment 7. In this experiment, we study the impact of non-stationary asynchronicity. Specifically, we have ten agents and the sampling probability of agent j follows a sine function, $\sin(\theta_j + t/30)$, where the phase shifts $\theta_j = j/5, j \in \{1, ..., 10\}$ are different for different agents. We report the number of communications and group regrets after $T = 80,000$ time slots averaged over 30 independent trials in Table 6.

Table 6: Experiment 7

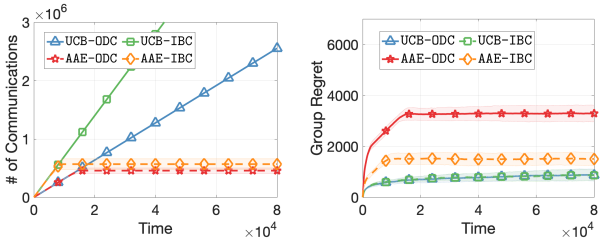
	Communication	Group Regret
UCB-IBC	$(2.2936 \pm 0.0020) \times 10^6$	2411 ± 296
UCB-ODC	$(1.5762 \pm 0.0021) \times 10^6$	2335 ± 224
AAE-IBC	$(2.2934 \pm 0.0023) \times 10^6$	7327 ± 341
AAE-ODC	$(1.5764 \pm 0.0026) \times 10^6$	7526 ± 422

Results of Experiment 5, 6, and 7 in Table 3, 4, 5, and 6 support our theoretical and experimental observations that ODC incurs less communication than IBC while achieving similar group regret, and further show that ODC is affective under various kinds of asynchronicity.

I.4 When would AAE-ODC incur fewer communications than UCB-ODC?

Note that our theoretical analysis suggests that AAE-ODC outperforms UCB-ODC in terms of communication complexity. However, this has not been clearly shown in previous experiments because, in previous experiments, the time horizon T is comparatively small for the arm reward suboptimality gap considered. In Figure 9, we present the number of communications and group regret (averaged

over 30 independent trials) incurred by AAE-ODC and UCB-ODC in the setting with $K = 16$ arms, 5 fast agents each with sampling probability 0.8, 10 slow agents each with sampling probability 0.1, and $T = 80\,000$. Note that this setting is same as one of the cases in Experiment 2, except that here we experiment with an easier arm reward instance (with larger suboptimality gap).



(a) Communication (b) Group Regret

Figure 9: AAE-ODC outperforms UCB-ODC in terms of communication complexity when the time horizon T is comparatively large for the arm reward instance.