FEDERATED REINFORCEMENT LEARNING FOR CONTENT DISSEMINATION IN UAV NETWORKS

By

Amit Kumar Bhuyan

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical and Computer Engineering – Doctor of Philosophy

2025

ABSTRACT

In disaster scenarios with compromised communication infrastructure, Unmanned Aerial Vehicles (UAVs) can provide *ad hoc* connectivity for resilient information dissemination. This thesis develops a hierarchical UAV-assisted framework of federated multi-armed bandit learning for post-disaster content dissemination. The developed framework incorporates a two-tier UAV hierarchy consisting of Anchor UAVs (A-UAVs) with high-cost backhaul connectivity, and more mobile Micro-Ferrying UAVs (MF-UAVs) without backhaul links. Such a hierarchy allows for strategic offloading of storage-intensive tasks to A-UAVs, while leveraging the mobility of MF-UAVs to dynamically ferry content across disconnected user clusters. By integrating trajectory-aware selective caching strategies into UAV operations, the framework aligns aerial mobility patterns with evolving spatio-temporal content demands. Algorithmic innovation of the framework stems from a federated bandit and stateless reinforcement learning paradigm, which enables UAVs to collaboratively learn content popularity profiles, and adapt caching policies based on localized user request patterns. Unlike centralized methods, the federated approach preserves data locality and minimizes inter-UAV communication overhead, which is critical in bandwidth- and energy-constrained post-disaster environments. The multi-armed bandit learning mechanism utilizes a multi-dimensional reward feedback architecture that captures content relevance, inter-UAV delivery latency, and caching diversity across disjointed and isolated user communities. The thesis also explores the interplay between UAV energy budgets, caching capacities, and mission-critical delivery constraints such as quality-of-service expectations in terms of tolerable access delay. To summarize, the research in this thesis bridges multi-agent learning with mission-oriented aerial networking towards developing solutions for smart content dissemination in networks with sparse connectivity.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Subir Biswas, whose guidance, insight, and support have been central to my academic journey. His mentorship has profoundly shaped my research and growth. I am also sincerely thankful to my dissertation committee members, Dr. Nihar Mahapatra, Dr. Shaunak Bopardikar, and Dr. Sandeep Kulkarni, for their valuable insights, constructive feedback, and time throughout this process.

I have been fortunate to share my PhD journey with outstanding colleagues and friends. I am especially grateful to my lab colleague and close friend, Hrishikesh Dutta, whose companionship, support, and stimulating conversations greatly enriched my doctoral experience. My sincere appreciation also goes to Gao Kang, whose friendship and encouragement have been truly meaningful. I also thank Yida Yang for his genuine camaraderie, helpfulness and support at every step. I extend my thanks to Avirup Roy and Tianxiang Zhang for fostering a positive lab environment. I would also like to acknowledge some old members and new additions to our lab group, Daniel McDermott, Grace Michael, Tushig Bolorchuluun, Fengyang Shang, Santhosh Manoharan, and Reeve Fernandes, whose presence and enthusiasm enhanced our team's dynamic.

I am grateful for the consistent administrative support from the College of Engineering and the Department of Electrical and Computer Engineering at Michigan State University, specifically Dr. Katy Colbry, Dr. Tim Hogan, Dr. Nelson Sepulveda, Dr. Yiming Deng, and staff members Lisa Clark, Michelle Stewart, Casie Medina, Michele Pursell, Brian Wright, Laurene Rashid, Meagan Kroll, and Jessica Pung.

I also want to acknowledge past mentors and collaborators who helped shape my early academic path, including Dr. D. L. Woodard, Dr. J. B. Harley, Dr. A. Zare, Dr. S. N. Merchant, Dr. J. H.

Nirmal, Dr. K. Samudravijaya, Dr. S. K. Kopparapu, and friends Vivek Patkar and Apoorva Kayal for their meaningful involvement during formative stages of my research career.

To my friends from childhood to adulthood, thank you for always being there, lending an ear, offering advice, and providing much-needed laughter along the way.

I owe my greatest gratitude to my family, especially to my mother, Kabita Rani Bhuyan (Amma), whose unconditional love, enduring strength, and sacrifice have provided the foundation upon which I stand today. I am profoundly thankful to my sister, Reshma, whose steadfast support, quiet resilience, and constant belief in me have remained a dependable source of strength throughout this journey. I extend my thanks to my brother-in-law, Babandeep, for his constant encouragement.

To my family here, words fall short. My heartfelt appreciation and deepest thanks go to my wife, Kristyn (Mrs. Bhuyan), who is the greatest inspiration and pillar of strength in my life, without whose unwavering love, support, thoughtful encouragement, and understanding this achievement would not have been possible. I am immensely thankful for my daughters, Daisy and Charlotte, whose joyful presence provides hope and purpose every day.

Finally, to all my immediate and extended family members, your continuous encouragement and love have been invaluable throughout this journey.

TABLE OF CONTENTS

Chapter 1: Introduction

1.1 Background and Motivation

Catastrophic events, including natural disasters like earthquakes and floods, as well as human-induced crises such as wars, have profound impacts on both the physical landscape and critical human-made infrastructures, notably the communication systems. In the aftermath of such events, the collapse of conventional communication networks can significantly hinder disaster response and relief efforts [1], [2]. Such situations often leave communities isolated from crucial information flow regarding disaster dynamics, relief operations, weather conditions, and rehabilitation efforts. Access to such information is essential, sometimes even lifesaving, for the affected communities [3], [4].

The thesis aims towards scenarios in which a disaster/war-stricken population is forced into multiple clusters of isolated communities with diverse information needs. The diversity, or heterogeneity, of these needs reflects in the varying popularity of requested content across communities, influenced by their proximity to the disaster and the users' geo-temporal context. For instance, a community close to a fire might prioritize information about nearby fire stations, whereas one farther away focuses on transportation to evacuate. Additionally, the expectation of Quality-of-Service, measured by tolerable access delay (TAD) [5], [6], [7] for different content, varies based on the urgency and type of information needed.

Numerous studies have explored the deployment of Device-to-Device (D2D) communication [8], [9] and Ad Hoc networks [10], [11] as solutions to bridge gaps in communication infrastructure, although with limitations in cases of total infrastructure collapse. Proposed Delay Tolerant Networks (DTNs) facilitate content transfer across fragmented communities [12], [13],

[14], [15], [16] that addresses routing delays but often neglects challenges in data caching and the effects of device mobility on caching efficiency.

Moreover, ensuring the consistent availability of content, given the wide variances in user request patterns, poses an unresolved challenge. Some strategies employ function approximation to predict request dynamics, a method dependent on extensive data collection that may not suit the urgent nature of information dissemination in crises.

The thesis underscores the potential of using Unmanned Aerial Vehicles (UAVs) as alternative platforms for content delivery that leverages their mobility against the constraints of limited storage, energy, and flight duration [17], [18], [19]. These insights aim to refine the understanding and management of content dissemination in disaster-affected areas which emphasizes the need for innovative solutions to ensure timely and reliable information access amidst challenging conditions.

1.2 UAVs and Micro-UAVs for Content Provisioning

This research introduces an advanced family of caching solutions towards employing trajectory-aware Unmanned Aerial Vehicles (UAVs) to facilitate the flow of information in areas where traditional communication infrastructures are compromised due to disasters or conflicts. This solution adopts a novel two-tier system which leverages anchor-UAVs (A-UAVs) equipped with high-cost vertical connectivity, such as satellite links [20], [21], [22], and a network of micro-ferrying-UAVs (MF-UAVs) [23], [24], [25] that operate without these connections. The MF-UAVs are pivotal in transferring and disseminating content among A-UAVs that ensures widespread content accessibility across fragmented communities by bypassing the need for direct vertical connections.

The main goal is to develop sophisticated strategies for content caching and downloading that are tailored to the unique storage limitations of the A-UAVs and MF-UAVs, the diverse demands for content among the communities, and the strategic distribution of content requests. A significant focus is placed on analyzing how the trajectories of MF-UAV fleets impact the availability of content to these segmented groups. The intention is to enhance content reach within these communities by overcoming the obstacles of restricted connectivity.

By leveraging a dual-layered strategy, this thesis aims to discover highly efficient methods for distributing content. It factors in the varying demands for information, the sizes of MF-UAV fleets, and their storage capabilities to maximize content access for isolated groups. This approach not only addresses the immediate need for reliable information in crisis situations but also sets a new benchmark for content delivery mechanisms in challenging environments.

1.2.1 Advantages of employing UAVs

Using Unmanned Aerial Vehicles for content provisioning in scenarios where communication infrastructure is absent or damaged has several advantages:

a) *Rapid Deployment*: UAVs can be quickly deployed to areas lacking communication infrastructure that enables swift establishment of temporary communication networks. This is particularly beneficial in disaster-hit areas where existing infrastructure is destroyed or in remote regions that lack such facilities.

b) *Flexibility and Scalability*: UAVs offer flexible and scalable solutions for content delivery. They can be used in a variety of scenarios, ranging from small-scale deployments to cover specific areas to larger networks consisting of multiple UAVs that work together to cover wider regions.

c) *Cost-Effectiveness*: Compared to the construction of traditional communication infrastructure, UAVs represent a cost-effective solution, especially in hard-to-reach areas. They eliminate the need for physical infrastructure like towers and cables which reduces both the initial setup cost and ongoing maintenance expenses.

d) *Dynamic Network Topology*: UAVs can dynamically adjust their positions based on the demand for communication services which optimizes the network's performance and coverage. This adaptability ensures efficient use of resources and enhanced connectivity in areas where user density might fluctuate.

e) *Improved Accessibility*: By providing overhead communication links, UAVs can enhance the accessibility of content and internet services to remote and underserved communities. This helps in bridging the digital divide and promoting equal access to information.

f) *Enhanced Data Collection and Distribution*: UAVs equipped with sensors and cameras can collect and distribute a wide range of data, including live video feeds, which can be crucial for surveillance, environmental monitoring, and disaster management.

1.2.2 Challenges faced in UAV-based Content Provisioning

Deploying Unmanned Aerial Vehicles (UAVs) for content delivery in areas without traditional communication infrastructure encounters specific challenges like power management and operational efficiency. These UAVs face constraints like limited flight duration and operational range, which are critical factors in their ability to deliver content over extended distances or periods. The power requirements for various operations such as downloading and transmitting content, receiving content requests, and maintaining basic flight, are significant. Each of these operations depletes the UAV's battery, with the energy consumed during its flight and idle states [18], [26], [27], [28], [29].

A crucial aspect of using UAVs for content distribution is managing the balance between power consumption and operational effectiveness. The process of downloading and transmitting content to users, for instance, requires careful consideration of power use, especially since transmitting power needs are substantially higher than those for receiving requests. This discrepancy is largely due to the properties of signal transmission, where ensuring that a signal reaches the receiver with sufficient strength to be decoded correctly necessitates a higher energy output. This shows that increased power consumption with higher transmission power necessitates a careful balance between communication requirement and operational longevity of the UAVs. Enhancing one can significantly impact the other, potentially reducing the effectiveness of content distribution efforts.

Furthermore, to maintain unhindered content provisioning, the aim should be to maximize the on-time of the UAVs. However, there are inherent energy expenses associated with UAV like the ones cited above. One straight-forward approach is to minimize the communication energy expenditure which includes the content download, transmission and reception expenses. Such approach will depend heavily on the average data consumption rate of individual users. Service providers have reported an average monthly data consumption of 25-30 gigabytes per user [30], [31], [32]. With an average population density of 10^5 users/sq-mile [33], [34], the data requirement per day can reach up to 80-100 terabytes. Attempts to handle such requirements can be made by installing Non-Volatile Memory Express solid-state drive (NVMe SSD) memory cards [35], [36] in UAVs that can store contents with data size of aforesaid magnitude. Nevertheless, the communication energy expenditure can still deplete the battery of the UAVs while storing and replacing contents in these large memory devices. To exacerbate the situation, with increase in the data storage capacities of such memory cards, the communication energy expenditure scales which leads to even faster depletion of UAV battery.

This limitation necessitates the contents to be intelligently spread across UAVs. Also, the inability of the total storage capacity offered by the UAV network to store all contents required currently and, in the future, requires efficient content management strategies.

1.3 Advantages of Proactive Caching over Traditional Caching Techniques

Proactive caching is an advanced strategy that enhances data storage and access beyond conventional methods like LRU, FIFO, and LFU [37], [38], [39], particularly vital in the context of UAVs with their limited storage and battery constraints. Unlike traditional caching, which often relies on removing the oldest or least used content, proactive caching anticipates and prepares for future demand by intelligently predicting which content will be needed soon. This foresight allows UAVs to prioritize and store only the most relevant information which optimizes resource use and improves content delivery efficiency. Key to this approach is leveraging data on content popularity and request rates that enables these systems to dynamically adjust their cache to meet anticipated needs. This ensures that high-priority content is always ready for users, thereby maximizing cache space efficiency and minimizing latency.

1.4 Cooperative Federated Reinforcement Learning-based Techniques

Building on these principles by leveraging machine learning (ML) techniques such as reinforcement learning (RL) [40], [41] and Multi-Armed Bandit (MAB) algorithms [42], [43], [44], UAVs can develop sophisticated caching policies that adapt in real-time without prior knowledge of content popularity or request rates. These ML strategies enable UAVs to balance between exploring new caching approaches and exploiting effective existing ones which in turn optimizes for long-term benefits like lower latency and higher content accessibility. This adaptive framework allows UAVs to dynamically tailor caching decisions to fluctuating network conditions and user demands by using trial and error to refine strategies based on direct environmental feedback.

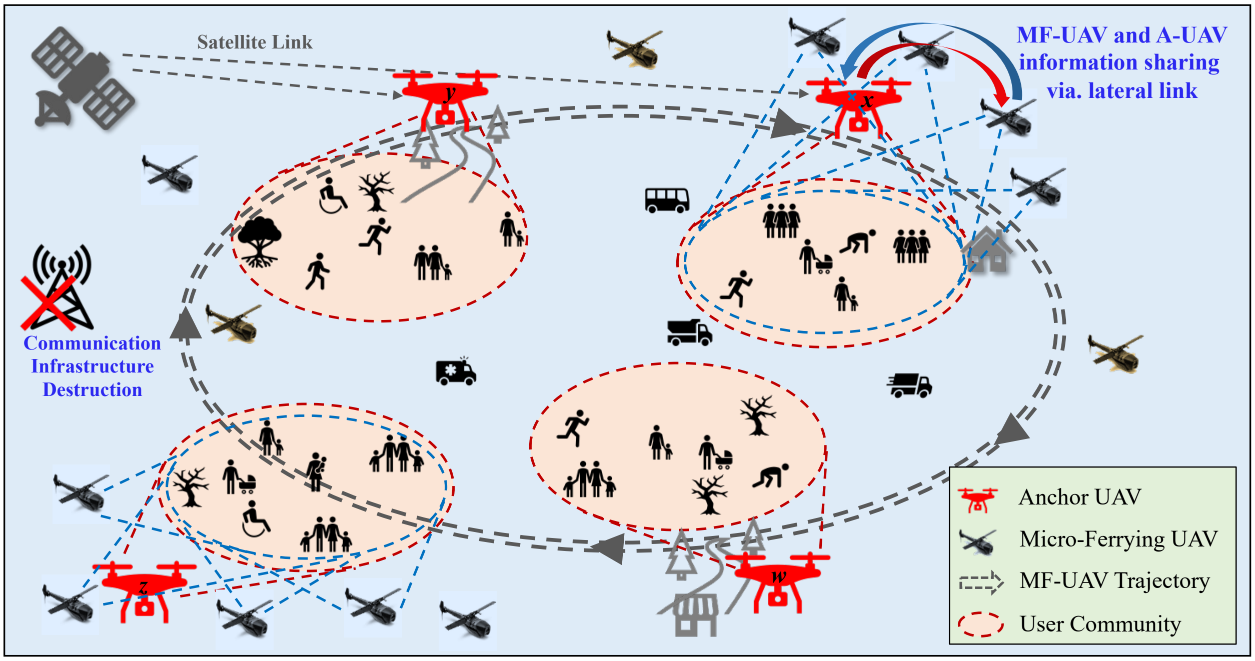

Satellite Link
MF-UAV and A-UAV information sharing via. lateral link
y
x
Communication Infrastructure Destruction
z
w
Anchor UAV
Micro-Ferrying UAV
MF-UAV Trajectory
User Community

Figure 1.1. Coordinated UAV system for content dissemination in environments without communication infrastructure

Furthermore, the adoption of cooperative learning algorithms enhances collaborative caching within UAV networks. This involves multiple UAVs or nodes sharing insights to refine caching decisions network-wide, crucial for coordinated operations such as surveillance or content delivery. Federated Learning (FL) [45], [46], [47] emerges as a key technique in this context that promotes distributed learning and decision-making to continuously refine caching strategies based on collective data, thereby improves content availability and user experience.

These adaptive learning mechanisms allow UAVs to adjust caching policies on-the-fly that responds adeptly to varying user demands and network conditions. By combining the exploration capabilities of MAB [48], [49] with the distributed intelligence of FL [50], [51], [52], UAV networks achieve greater resource efficiency and content delivery performance. This blend of proactive and cooperative learning techniques in caching equips UAVs with the capability to continuously evolve their caching strategies that enhances the efficiency of cache space utilization and the user experience by minimizing content access delays. This continuous improvement cycle ensures UAVs can effectively anticipate and meet changing content needs which enhances the overall efficiency and responsiveness of the network.

1.5 Learning Caching Policies and Enhancing UAV-aided Content Dissemination in Disaster-Affected Areas

In UAV-enhanced communication systems, Anchor UAVs (A-UAVs) with advanced communication technology like satellite links work alongside Micro-Ferrying UAVs (MF-UAVs) to bridge gaps in disaster-struck areas that ensures critical data reaches isolated communities. This network relies on content caching, especially within A-UAVs by employing Smart Cache Duplication to optimize their storage by prioritizing essential data. This strategy faces challenges such as accurately predicting demand and adapting to the varying nature of emergencies, which can alter priority information swiftly.

Addressing the variability in information needs across different communities, where urgency and type of information required can vary significantly, introduces complexity. Quality of Service (QoS) like Tolerable Access Delay (TAD), adds another layer of complexity, with user expectations for prompt content delivery influences caching decisions. Traditional methods struggle in such scenarios due to difficulty in forecasting content popularity and delay tolerances.

To navigate these challenges, this thesis integrates the *Top-k* Multi-Armed Bandit (MAB) algorithm, a technique that permits UAVs to refine caching strategies based on observed demand. This adaptive method moves beyond static pre-loaded caches to a dynamic model that aligns with real-time needs which accommodates the diverse and evolving content requirements and TAD expectations. By adopting a multi-dimensional reward mechanism, the *Top-k* MAB approach enables UAVs to continuously update their caching policies that ensures optimal content availability and meets varying QoS demands. This innovative strategy demonstrates a tailored, responsive approach to content delivery in UAV-supported networks, significantly enhancing the relevance and timeliness of information provided to affected communities.

While the *Top-k* MAB approach offers significant improvements, certain limitations remain, particularly in large-scale scenarios with sparse user activity. Relying on MF-UAVs to relay content availability information across distant regions slows down the learning process, especially when information must traverse multiple communities. Additionally, content that is less frequently requested often suffers from unreliable popularity estimates which reduces the effectiveness of caching decisions. These issues are further complicated as the content pool grows which makes it harder to maintain stable and timely learning. To address these challenges, this thesis introduces a Federated Multi-Armed Bandit (FedMAB) framework, where A-UAVs collaboratively refine their caching strategies by sharing learned models rather than raw data. This cooperation accelerates learning, enhances the stability of caching decisions, and ensures that even low-demand content receives fair consideration that ultimately improves the overall responsiveness and equity of information delivery in critical situations.

1.6 Strategic Joint Deployment of UAVs for Cache Space Utilization

In UAV content dissemination networks, operating micro-ferrying UAVs (MF-UAVs) independently without collaboration can lead to inefficiencies, notably through content duplication across UAVs. This redundancy hampers cache utilization, with UAVs possibly carrying identical data and leaving other crucial information not cached. This approach not only wastes cache space but also fails to address diverse user needs effectively.

Such redundancy also undermines the efficiency of Anchor UAVs (A-UAVs), crucial for their larger cache capacities and broader communication capabilities. When MF-UAVs carry duplicated content, A-UAVs' potential to distribute a varied range of information, especially critical in emergencies, is not fully leveraged.

Implementing a coordinated UAV deployment with a unified caching strategy can rectify these issues. By aligning caching decisions among MF-UAVs, the network optimizes cache space that ensures a broader variety of content is distributed efficiently. This collective approach eliminates unnecessary duplications which allows for strategic content allocation based on user demand, therefore significantly improves information availability and network performance in crisis scenarios.

1.7 Trajectory-aware Collaborative Caching using Swarm of Micro-Ferrying UAVs

Deploying UAVs strategically for content caching is essential for enhancing the functionality of UAV-supported networks. The collaboration between Anchor UAVs (A-UAVs) and Micro-Ferrying UAVs (MF-UAVs) is vital, with A-UAVs acting as central storage hubs for in-demand content, and MF-UAVs distributing this content. The application of the *Top-k* Multi-Armed Bandit Learning technique at A-UAVs facilitates dynamic learning which enables UAVs to adjust their caching strategies in real-time to meet user demand. This method considers the diverse urgency and importance of content requests, along with the content distribution patterns between UAVs, to optimize content availability.

Nonetheless, these strategies face challenges, like the risk of content duplication among MF-UAVs, which can waste valuable cache space. Moreover, without tailoring caching policies to Quality of Experience (QoE) requirements, UAVs may non-selectively cache identical content by ignoring the specific needs of their target areas.

An evolved approach, *Top-k* Multi-Armed Bandit Learning with Selective Caching, addresses these issues by discerningly caching content, taking into account what is already stored across the network to prevent redundancy. This selective strategy ensures a varied and efficient cache usage that aligns more closely with user community demands. Continuous adaptation and learning from environmental interactions allow for the prioritization of highly relevant content, therefore enhancing network performance. This ensures that content is available where and when needed, thereby sustaining a high quality of service and maximizing the UAV network's impact.

However, introducing federated learning presents a new challenge. While FedMAB improves adaptability through collaborative model updates, its aggregation process can inadvertently reduce the effectiveness of selective caching. This creates a trade-off between global coordination and localized efficiency, where selective strategies risk being overshadowed by uniform model consensus. This thesis deliberates on this critical tension and proposes a latency-aware coordination mechanism that preserves the benefits of selective caching without compromising the collaborative strengths of federated learning. By aligning the learning dynamics with the operational constraints, the proposed framework ensures balanced, efficient, and context-sensitive content dissemination across the UAV network. This thesis also attempts to convert this reactive caching policy to a proactive one, by incorporating crowd-counting algorithms. This method uses advances computer vision techniques to adeptly achieve population density which can compensate for any plausible weak estimate as a result of request sampling discrepancies.

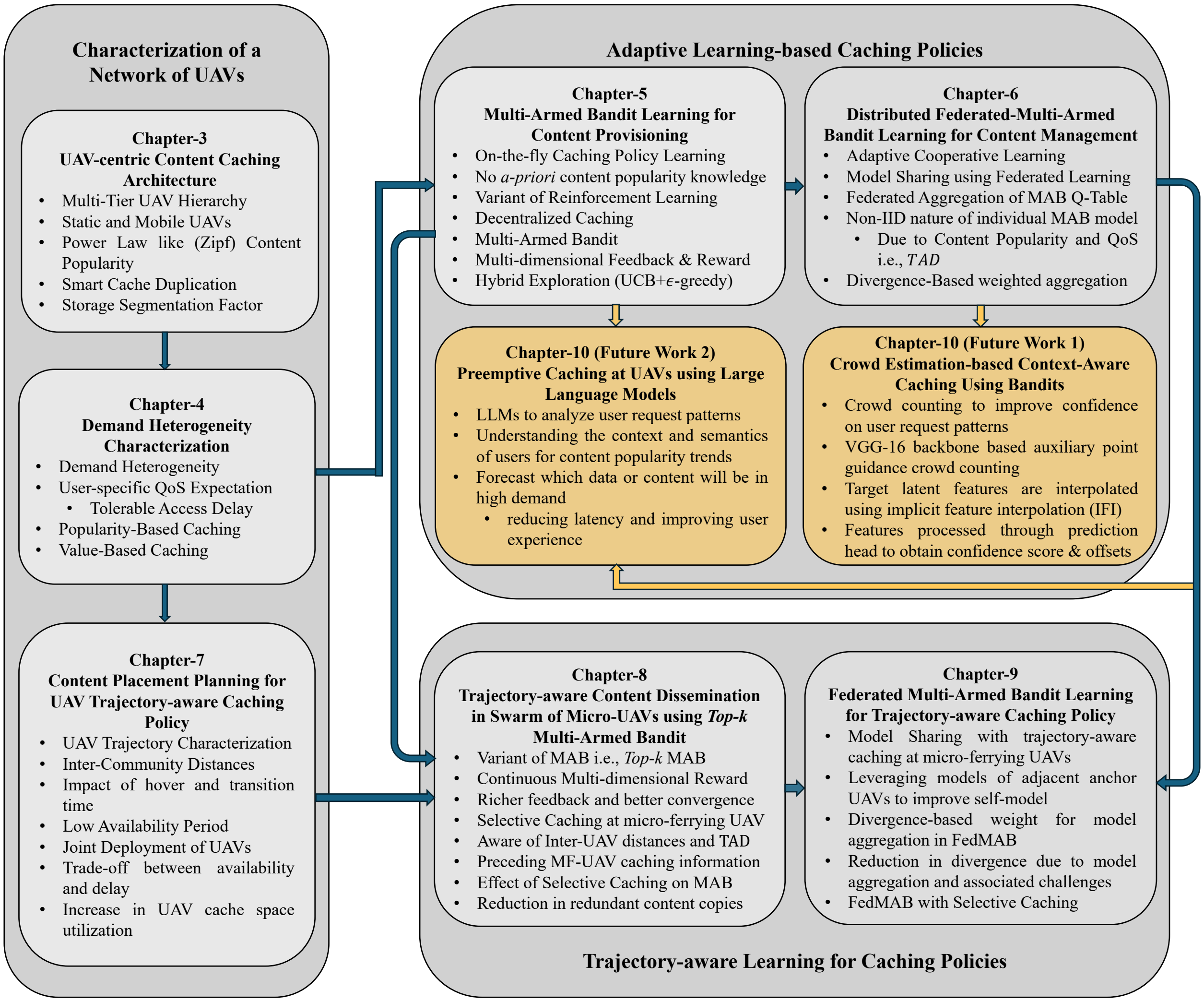

Characterization of a Network of UAVs
Chapter-3
UAV-centric Content Caching Architecture
• Multi-Tier UAV Hierarchy
• Static and Mobile UAVs
• Power Law like (Zipf) Content Popularity
• Smart Cache Duplication
• Storage Segmentation Factor
Chapter-4
Demand Heterogeneity Characterization
• Demand Heterogeneity
• User-specific QoS Expectation
• Tolerable Access Delay
• Popularity-Based Caching
• Value-Based Caching
Chapter-7
Content Placement Planning for UAV Trajectory-aware Caching Policy
• UAV Trajectory Characterization
• Inter-Community Distances
• Impact of hover and transition time
• Low Availability Period
• Joint Deployment of UAVs
• Trade-off between availability and delay
• Increase in UAV cache space utilization
Adaptive Learning-based Caching Policies
Chapter-5
Multi-Armed Bandit Learning for Content Provisioning
• On-the-fly Caching Policy Learning
• No a-priori content popularity knowledge
• Variant of Reinforcement Learning
• Decentralized Caching
• Multi-Armed Bandit
• Multi-dimensional Feedback & Reward
• Hybrid Exploration (UCB+ϵ-greedy)
Chapter-6
Distributed Federated-Multi-Armed Bandit Learning for Content Management
• Adaptive Cooperative Learning
• Model Sharing using Federated Learning
• Federated Aggregation of MAB Q-Table
• Non-IID nature of individual MAB model
• Due to Content Popularity and QoS i.e., TAD
• Divergence-Based weighted aggregation
Chapter-10 (Future Work 2)
Preemptive Caching at UAVs using Large Language Models
• LLMs to analyze user request patterns
• Understanding the context and semantics of users for content popularity trends
• Forecast which data or content will be in high demand
• reducing latency and improving user experience
Chapter-10 (Future Work 1)
Crowd Estimation-based Context-Aware Caching Using Bandits
• Crowd counting to improve confidence on user request patterns
• VGG-16 backbone based auxiliary point guidance crowd counting
• Target latent features are interpolated using implicit feature interpolation (IFI)
• Features processed through prediction head to obtain confidence score & offsets
Chapter-8
Trajectory-aware Content Dissemination in Swarm of Micro-UAVs using Top-k Multi-Armed Bandit
• Variant of MAB i.e., Top-k MAB
• Continuous Multi-dimensional Reward
• Richer feedback and better convergence
• Selective Caching at micro-ferrying UAV
• Aware of Inter-UAV distances and TAD
• Preceding MF-UAV caching information
• Effect of Selective Caching on MAB
• Reduction in redundant content copies
Chapter-9
Federated Multi-Armed Bandit Learning for Trajectory-aware Caching Policy
• Model Sharing with trajectory-aware caching at micro-ferrying UAVs
• Leveraging models of adjacent anchor UAVs to improve self-model
• Divergence-based weight for model aggregation in FedMAB
• Reduction in divergence due to model aggregation and associated challenges
• FedMAB with Selective Caching
Trajectory-aware Learning for Caching Policies

Figure 1.2. Thesis Organization

(The grey blocks are the works that this thesis has achieved, and the orange blocks are the future directions)

1.8 Dissertation Objectives

The objectives of the dissertation are multifaceted, that enhances UAV-based content provisioning systems through advanced algorithmic frameworks:

1- The dissertation characterizes a multi-tiered UAV-aided framework specifically tailored for managing content dissemination in disaster affected area. This framework is anticipated to adeptly cope with the variability in demand for content which ensures that all users receive timely and relevant information. An essential element of this is to enhance the caching decision of UAVs by leveraging knowledge of adjacent communities' UAVs with varying user demands that ensures an uninterrupted flow of information.

2- It achieves the development of a Federated Multi-Armed Bandit Learning-based framework to enhance content delivery via on-the-fly learning of caching policies for UAV-based systems. The intention here is twofold: firstly, to ensure the highest levels of content availability, thus ensuring users have access to the most pertinent information when needed; and secondly, to minimize the amount of downloaded contents that optimizes the efficiency of the network.

3- A core component of this research is the creation and thorough examination of a joint deployment strategy for UAV networks. The strategy aspires to amplify the availability of content across the network, drastically diminish periods when content is unavailable, and amplify the diversity of content within the UAV-aided caching system.

4- Furthermore, the dissertation explores the construction of an adaptive learning-based framework with an integrated *Selective Caching* method. This novel approach is sensitive to UAV trajectories and QoS parameters that focuses on elevating content availability, limiting access delays, reducing redundancy in content replication, and ensures that the cached content sequence mirrors the benchmark for optimal caching sequences.

5- Eventually, this thesis also attempts to improve this reactive learning-based caching policy to a proactive caching policy by incorporating crowd-counting algorithms. This addition aids the Federated Multi-Armed Bandit based caching policy by compensating for any weak reward estimates as a result of low request sampling problem.

The culmination of these objectives is aimed at delivering a robust, responsive, and efficient content delivery service via UAVs, even in the most challenging environments. Through these sophisticated learning mechanisms, UAV networks can dynamically adapt to the immediate needs of different user communities, effectively managing cache space to maintain high service quality even under the constraints and uncertainties inherent in post-disaster environments.

1.9 Scope of the Dissertation

The main goal of this thesis proposal is to propose a Federated Multi-Armed Bandit Learning-based content dissemination system for cache enabled Unmanned Aerial Vehicles to ensure content provisioning in communication infrastructure-less scenario. The proposed methods deliver via service provisioning performance maximization in terms of content availability, content access delay, content low availability period and cached content sequence similarity. The organization of the Thesis proposal is as follows:

This thesis presents a comprehensive exploration of UAV-centric content caching architectures aimed at enhancing communication in challenging environments. Chapter 2 lays the foundational groundwork by reviewing relevant literature which highlights key developments and existing strategies in UAV-assisted communication and content caching. In Chapter 3, we delve into the design of a UAV-centric Content Caching Architecture for Communication-challenged Environments that establishes the core principles behind deploying UAVs to facilitate efficient data delivery where traditional communication infrastructure is lacking or damaged. Chapter 4 expands on this by discussing how to handle demand heterogeneity in UAV-aided Content Caching which addresses the complexities of diverse content demands across different user groups in such environments. Chapter 5 introduces a novel approach that employs Multi-Armed Bandit Learning for Content Provisioning in a Network of UAVs which focuses on optimizing content delivery by learning user preferences and demand patterns over time. Chapter 6 emphasizes on the model sharing strategy by incorporating the concept of Federated Learning on Bandits, which strengthens the caching policy’s confidence and fairness of the bandit-based models across the disaster scenario. In Chapter 7, we explore Content Placement Planning for UAV Trajectory-aware Caching Policy in Infrastructure-less Wireless Networks, aiming to enhance the efficiency of content delivery by optimizing UAV flight paths based on content caching needs and network topology. Chapter 8 further advances this discussion by integrating *Top-k* Multi-Armed Bandit Learning for Content Dissemination in Swarms of Micro-UAVs with a trajectory-aware *selective caching* algorithm. This fine-tunes the content delivery process by identifying the groups of micro-UAVs traversing in close proximity and prioritizing the contents most in demand. Chapter 9 discussed the hurdles faced while incorporating Federated Multi-Armed Bandit to learn the caching policy. It also uncovers the algorithmic additions to tackle the issues of model sharing in the presence trajectory-aware *selective caching* algorithm. Finally, Chapter 10 discusses one of the immediate future extensions of this thesis where it shows the impact of crowd-counting applied on an image dataset that contains real disaster affected images. It explains the effect of precise crowd estimation on the learning-based caching policy. Additionally, it reflects on the research journey, summarizes achievements, and outlines potential future directions. This section highlights the scalability of the proposed architectures and algorithms, their adaptability to emerging hurdles, and their potential impact on future UAV-assisted communication systems. Through this organized structure, the thesis systematically addresses the challenges of UAV-based content caching and delivery, offers innovative solutions and paves the way for future advancements in the field.

Chapter 2: Related Work

This chapter builds upon the foundational motivations outlined in Chapter 1 by offering a structured review of existing literature that informs the development of UAV-assisted content dissemination strategies. It examines five interrelated areas that form the core of this thesis; post-disaster content provisioning frameworks, the role of UAVs and micro-UAVs in disaster response, UAV-based content provisioning through proactive caching, learning-based caching systems for dynamic environments, and recent developments in federated learning and bandit-based reinforcement learning for adaptive, decentralized decision-making.

2.1 Post-Disaster Content Provisioning

In the wake of disasters, whether natural or human-made, the efficient provisioning of content is crucial for effective response and recovery efforts. This entails not only the dissemination of vital information to affected populations and first responders but also the coordination among various agencies and stakeholders involved in the disaster management [53], [54] process. The existing literature on content provisioning frameworks in post-disaster scenarios highlights a variety of approaches, each with its unique set of challenges, solutions, and limitations. These include Mobile Ad-Hoc Networks (MANETs), Delay-Tolerant Networks (DTNs), social media platforms, satellite communications, Content Distribution Networks (CDNs), blockchain technology, and the Internet of Things (IoT).

MANETs and DTNs are often highlighted for their ability to provide flexible and resilient connectivity in the absence of traditional communication infrastructure. For example, [55], [56] discusses the design and application of DTNs in challenging communication environments which includes disaster-impacted areas. On the other hand, social Media Platforms have been increasingly recognized for their role in disaster communication. The authors of [57] examine how social media is utilized for emergency management that emphasizes its capacity for rapid information dissemination and public engagement.

Despite their potential, MANETs and DTNs based frameworks face various limitations and can suffer from connectivity and scalability issues. Similarly, social media platforms may struggle with misinformation and information overload. To tackle these discrepancies, IoT Applications in disaster management are gaining attention for their ability to provide real-time data and enhance situational awareness. The work in [58] discusses the integration of IoT technologies in emergency response systems which highlights their potential to improve monitoring, content provisioning and coordination. This motivates the use of UAVs as a viable solution for content dissemination as an extension to the IoT-assisted response systems.

2.2 Use of UAVs/Micro-UAVs for disaster management services

UAVs, especially micro-UAVs, have emerged as crucial tools in managing and mitigating the aftermath of disasters [59]. Their applications span across various critical tasks which includes aerial surveillance, terrestrial imaging, precision agriculture, and infrastructure inspection in areas struck by calamities [60], [61]. These aerial vehicles excel in gathering real-time data and offers a bird's-eye view of disaster-struck regions, which is instrumental for effective and timely decision-making [23], [24]. The agility and small size of micro-UAVs make them particularly suitable for navigating through constrained spaces that enables assessments in areas that are otherwise inaccessible to traditional disaster response machinery [62].

The utility of UAVs in such scenarios is multifaceted. Primarily, they are deployed for their ability to quickly and efficiently survey large and hard-to-reach areas that provides critical information on the extent of damage, identifying stranded victims, and assessing the needs of affected communities [63]. Their application in precision agriculture, for instance, through yield estimation and crop monitoring, underscores their capability in managing resources and assessing environmental conditions, which can be adapted for disaster assessment and recovery efforts [9].

Several studies have focused on optimizing the capabilities of UAVs for enhanced service provision in disaster management. Works such as [64], [65] propose methods to optimize the flight paths of UAVs and schedule communication tasks. These strategies aim to extend service coverage through optimized UAV hovering times and employing multi-hop relaying between multiple UAVs which includes device-to-device (D2D) routing. Such approaches are pivotal for ensuring continuous and effective communication in disaster-affected areas, especially when traditional communication infrastructure is compromised.

The incorporation of energy-aware strategies which includes the use of multi-armed bandit algorithms [66], [67], focuses on selecting user hotspots for efficient data transmission while minimizing UAV energy consumption. Additionally, employing multiple UAVs at different altitudes [68] or through dynamic leader selection in a master-slave architecture [62] optimizes both the coverage area and energy usage that ensures longer operational periods in critical situations.

While many studies have emphasized enhancing communication range [69] and optimizing flight paths, there’s a noted gap in addressing content placement and caching strategies specific to disaster management scenarios. However, the adaptation of solid-state drives (SSDs) for increased caching capacity [70] suggests a direction towards integrating more sophisticated data handling capabilities in UAVs that enhances their utility in disseminating vital information and services in disaster-struck regions.

Despite these advancements, simply increasing storage space does not directly solve the content availability challenge. This is because an expansion in storage capacity results in higher energy consumption for downloading and updating content, which requires long-range communication equipment (refer Section 1.2.2). As the storage space enlarges, the communication energy expense also scales, which inadvertently reduces the UAV’s flight time by consuming energy that could otherwise support flight operations. In contrast, adopting inter-UAV content exchange methods that utilize low-range communication equipment can significantly save on communication energy which results in conservation of more power for prolonged flight. This approach not only addresses the efficient management of content but also enhances the UAV’s operational time which makes it a more viable solution for disaster management scenarios where endurance and efficient content delivery are critical. This multifaceted utility underscores UAVs’ significance in not only bridging connectivity gaps but also in ensuring timely, efficient, and secure access to vital content and services in diverse operational contexts.

2.3 UAV-based Content Provisioning via Proactive Caching

In the evolving landscape of Unmanned Aerial Vehicle (UAV) technology, the strategic placement and caching of content have emerged as pivotal components in enhancing the efficacy of UAV-based communication networks, particularly within the context of Internet of Things (IoT) networks and disaster management scenarios. This thesis delves into the various methodologies proposed in the literature for optimizing these aspects and sheds light on their potential to revolutionize the way information is disseminated in critical situations.

The advent of Named Data Networking (NDN) architecture in IoT networks represents a significant leap forward in content distribution. Studies such as those referenced in [71] illustrate how UAVs can harness this architecture to gather data directly from the field and deliver it efficiently to the intended recipients, and therefore circumvents the need for retransmission and enhances overall network performance. This approach not only simplifies the data delivery process but also significantly reduces the burden on the network infrastructure.

Further exploring the realm of UAV-enabled communication, research in [72] introduces innovative strategies where UAVs proactively transmit content to a select group of ground nodes (GNs). These GNs are algorithmically chosen to cooperatively cache the necessary content that ensures a broad and efficient distribution network that maximizes accessibility for end users. The employment of a probabilistic cache placement technique, as discussed in [73], aims to further refine this process by enhancing cache hit probabilities that leverages a homogeneous Poisson Point Process for the strategic placement of wireless nodes.

Addressing the challenges faced by small-cell base stations (SBSs) under the strain of high data traffic, several studies [74], [75] propose the use of UAVs as a relief mechanism. By caching enhanced layer information, UAVs can efficiently manage high-definition video streaming requests, with the base layer information handled by the SBSs. This dual-layer approach not only alleviates pressure on the SBSs but also incorporates measures for interference management and security against potential eavesdropping which showcases the multifaceted benefits of UAV integration into existing networks.

The optimization of cache placement in areas with high data traffic is another critical area of research. In [76], the authors utilize greedy algorithms and the Lagrange dual method to strategically determine the content to be cached on UAVs that takes into consideration the dynamic nature of user movement across different coverage areas. This emphasis on adapting to user heterogeneities marks a significant advancement in customizing content availability that directly addresses the limitations of temporally static user movement models.

Despite these innovative approaches, there remains a gap in effectively maximizing cache capacity within UAV-aided content dissemination networks, especially in scenarios characterized by demand heterogeneity. This gap signifies an area ripe for further exploration, as highlighted by the research efforts aimed at traffic offloading methods and learning-based caching strategies. Studies [72], [73], [74], [75] reveal that by considering factors such as content popularity and size, the caching capacity of UAVs can be significantly enhanced. This is particularly evident in UAV-enabled small-cell networks, where data traffic is offloaded from SBSs to UAVs which allows for the proactive caching of popular content and direct delivery to users as needed.

However, these mechanisms, while promising for scenarios of partial infrastructure destruction, face limitations in fully-functional alternatives where communication infrastructures are completely obliterated. Additionally, the reliance on temporally static models of global content popularity [73] in most existing mechanisms fails to capture the real-world complexity and variability of content demands, particularly in disaster scenarios.

This thesis advances the conversation around UAV-based content provisioning systems that focuses on proactive caching [73] as a cornerstone of efficient and resilient communication networks. By critically examining the current state of research and identifying areas for further investigation, this work contributes to the development of more sophisticated, adaptive, and robust UAV-based content distribution frameworks that are capable of meeting the nuanced demands of modern communication challenges. Through proactive caching and strategic content placement, UAVs hold the promise of transforming the landscape of information dissemination, particularly in scenarios where traditional communication infrastructures are compromised or entirely absent.

2.4 Learning-based Caching for UAV-based content dissemination system

The integration of Unmanned Aerial Vehicles (UAVs) into content dissemination networks represents a transformative approach to addressing the dynamic needs of modern communication systems. This particularly important in scenarios characterized by challenging environments and demand heterogeneity. This thesis explores the novel application of learning-based caching strategies and joint optimization of caching and trajectory decision techniques [77] that leverage the agility and flexibility of UAVs to optimize content delivery in various contexts.

Recent studies have proposed innovative methods that combine caching decisions with UAV trajectory planning to minimize content delivery delays and enhance user satisfaction. For instance, research indicated in [78] introduces a system where online decisions are influenced by inputs processed through a Convolutional Neural Network (CNN). This is done in tandem with subsequent caching and trajectory optimizations performed offline using a Clustering-Based Two Layered (CBTL) algorithm. This dual-phase approach meticulously balances immediate decision-making with strategic planning that ensures a cohesive content distribution strategy.

Further advancements in this field have been demonstrated by [79], where a deep Q-learning based framework is employed to jointly optimize UAV trajectory and radio resource allocation. This method specifically addresses the complexities of large networks characterized by an extensive range of state-action pairs that underscores the potential of deep reinforcement learning techniques in navigating the intricacies of UAV-based content dissemination.

However, existing models often overlook the critical factor of content popularity heterogeneity, which varies significantly with the geographic location of users. This oversight limits the applicability of such models in real-world scenarios where user demand and content preferences can shift drastically across different regions. To bridge this gap, our work introduces a nuanced approach that incorporates the variability of content popularity into the learning-based caching and decision-making process.

In the realm of UAV trajectory control, [80] has proposed mechanisms that dynamically adjust UAV missions based on real-time observations which includes the decision to continue service delivery along a planned trajectory or to return to a charging station. This adaptability is crucial in maximizing operational efficiency and ensuring uninterrupted service provision. Similarly, [81] delve into the mathematical formulation of joint optimization problems that aims to find the most energy-efficient trajectories for UAVs while managing radio resources and caching replacements.

Despite these technological strides, previous studies have not sufficiently addressed the specific challenges posed by disaster geographies, such as demand heterogeneity and the physical impacts on UAV flight decisions. Our research fills this critical void by meticulously analyzing how disaster-induced variations in geography and user demand affect caching policies and UAV trajectory planning.

Furthermore, the methodologies explored in [76], [78], [79], [80], [81] primarily utilize long-term estimation techniques, which may not adequately respond to the rapid changes in network conditions and user demand. This thesis argues for the development of more responsive and adaptable learning-based caching mechanisms that can swiftly adjust to evolving environmental and network dynamics. Additionally, there is a notable absence of efforts aimed at maximizing cache space utilization and reducing reliance on costly server downloads through direct UAV-to-user content delivery.

To address these shortcomings, our work develops a comprehensive framework that not only considers the heterogeneity of content popularity but also incorporates adaptive learning methods to optimize UAV caching decisions and flight trajectories in real-time. By leveraging advanced machine learning algorithms which includes reinforcement learning and its variant like Multi-Armed Bandits, we establish a robust benchmark for evaluating the effectiveness of UAV-based content dissemination strategies.

This framework significantly enhances the efficiency of content delivery networks, particularly in disaster recovery operations and other critical scenarios where traditional communication infrastructures are compromised. Through the judicious application of learning-based caching strategies, our approach improves the UAV content dissemination landscape by offering a more agile, responsive, and user-centric model that can dynamically adjust to the unique demands of diverse geographic and operational contexts.

2.5 Federated, Bandit-based and Reinforcement learning for adaptive UAV caching

To address the limitations of static or centralized approaches, recent literature has explored learning-based methods that enable UAVs to adaptively make caching decisions in dynamic environments. Reinforcement learning (RL), particularly through deep Q-networks and actor-critic methods, has been employed to jointly optimize content delivery and trajectory planning [82], [83], [84], [85], [86], [87], [88], [89], [90]. These methods offer adaptability but often rely on centralized infrastructure or long convergence periods which makes them less suitable for disaster scenarios characterized by network volatility and limited infrastructure.

In parallel, Multi-Armed Bandit (MAB) algorithms have been studied for online caching decisions under uncertainty. MAB-based methods treat content selection as a reward-driven exploration-exploitation trade-off. While effective in adapting to changing demand, early implementations typically assumed globally uniform popularity or lacked inter-agent coordination [91], [92], [93], [94], [95], [96], [97], [98], [99]. More recent work has attempted to combine

MABs with contextual information, yet challenges remain in achieving scalable coordination and responsiveness across distributed UAV agents.

Furthermore, Federated learning has gained traction as a way to address scalability and privacy constraints in decentralized environments [100], [101], [102], [103], [104]. In the federated paradigm, each processing node independently learns a local caching policy and periodically contributes to a shared global model through model aggregation, rather than raw data exchange. This is particularly beneficial in disaster zones, where bandwidth and power constraints make centralized updates infeasible.

The integration of federated learning with MABs, referred to as Federated Multi-Armed Bandit (FedMAB) learning, offers a hybrid approach that combines the local adaptivity of MABs with the scalability and privacy-preserving characteristics of federated learning. Unlike methods relying on long-term global demand estimation, FedMAB supports geo-temporal heterogeneity by learning and sharing local caching decisions across UAVs. The literature demonstrates the effectiveness of federated learning approaches in mobile and distributed networks (e.g., FL-based edge classification systems) [105], [106], [107], [108], [109], [110], [111], but their application to content dissemination under full infrastructure failure is still emerging. Also, federated aggregation has been achieved in graph-type learning paradigms such as DNN [112], [113], [114], [115], [116], which is both intuitive and achievable. Amalgamation of federated learning with tabular methods like MAB and RL has fundamental limitations, since the such aggregations are not straight-forward and can have detrimental effect on learning capabilities and contextual loss.

Our thesis builds directly upon these developments by implementing a FedMAB framework tailored for disaster-affected regions. It incorporates demand heterogeneity, varying quality-of-service constraints, and inter-UAV collaboration without requiring centralized coordination. By leveraging both the theoretical advantages and empirical performance of MABs and federated updates, our approach provides a foundation for scalable, resilient, and context-sensitive UAV caching strategies.

2.6 Summary

Existing learning-based UAV-aided content dissemination systems face several challenges that includes a lack of adaptability to rapid changes and demand heterogeneity, inefficient utilization of UAVs' caching capabilities, and insufficient focus on real-time adjustments for optimized content delivery. Our methods designed in this thesis address these drawbacks by incorporating advanced machine learning algorithms that account for demand variability across different regions, optimizing caching strategies according to UAV trajectory in real-time, and ensuring efficient use of UAVs' cache spaces for content delivery. By focusing on adaptability, efficiency, and responsiveness, our approaches enhance the effectiveness of UAV-aided content dissemination systems in meeting diverse operational demands.

In the next chapter, a UAV-aided content dissemination framework is characterized that can tackle content needs from the stranded users from disjoint communities in a disaster affected region. The framework is designed in a scenario where communication infrastructure is completely obliterated due to unforeseen catastrophic events.

Chapter 3: UAV Centric Content Caching for Communication Challenged Scenarios

This chapter introduces a specialized UAV-based caching framework aimed at enhancing content delivery in disaster-affected areas where traditional communication infrastructure is absent. Utilizing both static anchor UAVs for direct content access and mobile ferrying UAVs for broader content distribution, this system focuses on optimizing content availability through strategic caching and content duplication methods tailored to the constraints of UAV storage capacity. The framework's effectiveness is demonstrated through analytical and simulation-based evaluations, highlighting its capacity to adapt to various disaster scenarios, UAV trajectories and operational constraints. Here, the thesis details the framework's architecture, its innovative caching strategies, and the significant role of UAV trajectories in maximizing content accessibility for isolated user communities in crisis situations.

3.1 Motivation

Using UAVs for content provisioning without communication infrastructure faces specific challenges like power limitation, which affects flight duration and operational range, limiting the UAV's ability to deliver content over long distances or for extended periods. Let's outline a basic model for UAV power expenditure using the following parameters.

a) $P_{download}$: Power consumption of the communication module when actively downloading content.

b) P_{tx}: The power used for transmitting content to users, influenced by factors like distance, data rate, and the efficiency of the communication protocol.

c) P_{rx}: The power consumed by the UAV's communication system for receiving content request, depending on receiver sensitivity and signal processing requirements.

d) P_c: This is the power consumed by the UAV's communication system's circuitry, which includes the transmitter circuitry, receiver circuitry, and any signal processing components.

e) P_{idle}: Power consumption of communication module when on but not actively downloading.

f) P_{flight}: Power consumption for keeping the UAV in the air (motors, avionics, etc.).

g) $T_{download}$: Time spent downloading content.

h) T_{tx}: Time taken to transmit content to user.

i) T_{rx}: Time taken to receive content request from user.

j) T_{total}: Total flight time available (average flight time considered is 30 minutes).

k) $E_{battery}$: Total energy available from the UAV's battery. The UAV's battery charge (or energy) can be measured in watt-hours (Wh) or milliamp-hours (mAh).

Based on the above parameters, the depleted energy, remaining energy and remaining on-time can be mathematically approximated as follows:

$$E_{depleted} = P_{download}.T_{download} + (P_{tx} + P_c).T_{tx} + (P_{rx} + P_c).T_{rx} \quad (3.1)$$

$$E_{remaining} = E_{battery} - E_{depleted} \quad (3.2)$$

$$T_{remaining} = \frac{E_{remaining}}{P_{flight} + P_{idle}} = \frac{E_{battery} - E_{depleted}}{P_{flight} + P_{idle}} \quad (3.3)$$

The above expressions show the remaining on-time contingent upon the content download and lateral communication load with the users. To be noted that the transmission power P_{tx} is significantly higher than the reception power P_{rx} owing to the *Friis transmission equation* [117], [118] for free space. The transmission power in wireless communication systems is the power that the transmitter needs to emit to ensure the signal reaches a receiver with sufficient strength (P_r) to be decoded correctly. This relationship can be illustrated through the *Friis transmission equation* in a simplified form, assuming free space and line-of-sight communication:

$$P_r = P_{tx}.G_{tx}.G_r.\left(\frac{\lambda}{4\pi d}\right)^2 \tag{3.4}$$

The equation above defines P_{tx} in relation to P_r, given known values of G_{tx}, G_r, λ and d, which are the gain of the transmitter and receiver antenna, transmitted signal's wavelength and distance between the transmitter and receiver. This shows that increased power consumption with higher transmission power necessitates a careful balance between communication range and operational longevity of the UAVs [30]. Enhancing one can significantly impact the other, potentially reducing the effectiveness of content distribution efforts.

To maintain unhindered content provisioning, the aim should be to maximize the on-time of the UAVs. However, there are inherent energy expenses associated with UAV like the ones cited above. One straight-forward approach is to minimize the communication energy expenditure which includes the content download, transmission and reception expenses. Such approach will depend heavily on the average data consumption rate of individual users. Service providers have reported an average monthly data consumption of 25-30 gigabytes per user [30], [31], [32]. With an average population density of 10^5 users/sq-mile [33], [34], the data requirement per day can reach up to 80-100 terabytes. Attempts to handle such requirements can be made by installing Non-Volatile Memory Express solid-state drive (NVMe SSD) memory cards [35], [36] in UAVs that can store contents with data size of aforesaid magnitude. Nevertheless, the communication energy expenditure can still deplete the battery of the UAVs while storing and replacing contents in these memory devices. To exacerbate the situation, with increase in the data storage capacities of such memory cards, the communication energy expenditure scales which leads to even faster depletion of UAV battery.

This limitation necessitates the contents to be intelligently spread across UAVs. Also, the inability of the total storage capacity offered by the UAV network to store all contents required currently and, in the future, requires efficient content management strategies.

3.2 Design Objective

The objective of this chapter is to design and validate a comprehensive UAV-enabled content dissemination framework optimized for environments lacking fixed communication infrastructure. This involves the development of a detailed architectural model that leverages UAVs for content delivery. Furthermore, it includes formulation of optimal content placement and caching strategies tailored to specific UAV trajectories, and exploration of how these trajectories influence caching efficiency. Additionally, the chapter aims to construct analytical models capable of estimating content availability within this framework, supported by the execution of extensive simulation experiments. These simulations are intended to rigorously test and evaluate the effectiveness of the proposed strategies across a spectrum of network conditions and operational scenarios, thereby ensuring the framework's applicability.

3.3 System Model

3.3.1 UAV Hierarchy

The content distribution system is organized in two layers, namely, the anchor UAVs (i.e., A-UAVs) and the ferrying UAVs (i.e., F-UAVs). As shown in Figure 3.1, each partitioned community of users is served by an A-UAV using a lateral wireless link such as WiFi. A-UAVs can also download content form the internet via an expensive vertical link such as satellite-based internet. One monolithic system design approach is to let the A-UAVs download all needed content, as requested by their local users, via the vertical links. In this approach, with no inter-A-UAV data transfer, the following shortcomings will be encountered. First, there will be duplications of downloads via the expensive vertical links by different A-UAVs due to overlaps in requests for popular contents. This will incur high download costs. Second, storage constraints will cap the number of contents that can be downloaded and stored in an A-UAV, thus limiting the content availability. Finally, due to limited infrastructure availability, some of the communities of users are rendered isolated from content access without a dedicated A-UAVs assigned to them.

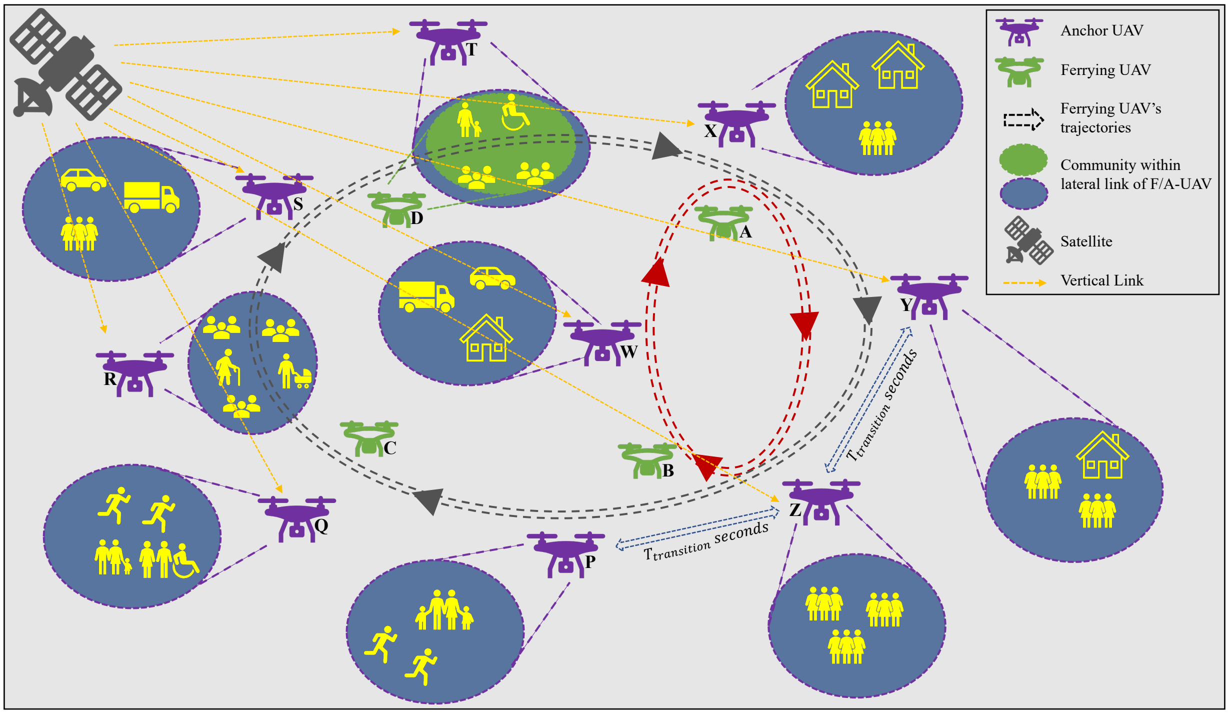


Figure 3.1. Coordinated UAV system for content caching and distribution in environments without communication

To address these problems, a set of ferrying UAVs (i.e., F-UAVs) are introduced. Unlike A-UAVs, the F-UAVs do not possess vertical links, but they do have lateral links such as WiFi, using which they can communicate with the A-UAVs. The role of these UAVs is to cache and transfer content around the A-UAVs such that the users in a community are able to access content that was downloaded by A-UAVs serving other communities via F-UAVs.

3.3.2 Content Request and Provisioning Model

Content requests are generated by the community users and sent to the local A-UAV or a visiting F-UAV, in that order.

Content Popularity and Requests: Studies have shown that content request pattern often follows a Zipf distribution in which a requested content's popularity is a geometric multiple of the next popular content in a larger pool [119]. Popularity of content $'i'$ is given as

$$p_\alpha(i) = \frac{\left(\frac{1}{i}\right)^\alpha}{\sum_{c \in C}\left(\frac{1}{c}\right)^\alpha} \tag{3.5}$$

The parameter C represents the total number of contents in the pool, and the Zipf parameter α determines the skewness of the distribution. Poisson request generation is the most prevalent way to capture real-time user requests.

Tolerable Access Delay and Content Provisioning: For each generated request, a Tolerable Access Delay (TAD) is specified. TAD is a quality-of-service parameter that indicates the duration a requesting user waits before the content is provisioned via download. After receiving a request from one of its community users, the relevant A-UAV first searches its local storage for the content. If not found, it waits for a potential future delivery of the content by one of the traveling F-UAVs. If no F-UAV with that content arrives within the specified TAD, the A-UAV downloads it via the vertical link.

3.4 Caching Policies

The caching related design questions to be addressed are: a) which content to be downloaded in the A-UAVs via the vertical links so that they can serve their own community directly, and the remote communities via the traveling F-UAVs; b) which content to be transferred from the A-UAVs to the F-UAVs via the lateral links, and cached within the F-UAVs subsequently; and finally, c) what inter-community trajectories should be followed by the F-UAVs.

This chapter addresses these questions in that it assumes pre-assigned globally known content popularities and static content pre-placements before user request are generated. In terms of F-UAV trajectories, different pre-programmed trajectories are characterized along with different static content placement strategies. After understating and characterizing such static policies, the goal will be to develop runtime and dynamic mechanisms for all these design components and report it in a future publication.

3.4.1 Caching at Anchor UAVs (A-UAVs)

A naïve strategy for the A-UAVs would be to cache the most popular contents (i.e., following the globally known Zipf distribution) to fill out their individual storage space of C_A contents. This naïve fully duplicated (FD) [120], [121] mechanism has the shortcoming in that it limits the number of accessible contents for all user communities to C_A, the A-UAV cache size. This limitation can be addressed by storing a certain number of unique (exclusive) contents in all the A-UAVs and share those contents across the communities via the traveling F-UAVs. This *Smart Cache Duplication* (SCD) mechanism can effectively increase the access to the number of contents for all users across the entire system, thus improving the overall availability within a given TAD.

Let the size of the duplicate segment of A-UAV cache be $\lambda. C_A$ and that of the unique segment be $(1-\lambda). C_A$where λ is a *duplication factor* that decides the level of content duplication in A-UAVs. This results into $N_A.(1-\lambda). C_A$ unique contents stored across all N_A number of A-UAVs in the system, and these can be shared across all user communities via the mobile F-UAVs. These unique contents have popularities after the top $\lambda. C_A$ popular duplicated contents in all the

A-UAVs. For symmetry, all $N_A.(1-\lambda).C_A$ unique contents are uniformly randomly distributed across N_A number of A-UAVs. The total number of contents in system: $C_{sys} = \lambda.C_A + N_A.(1-\lambda).C_A$.

It should be noted that with λ set to one, the SCD system reduces to the fully duplicated (FD) strategy. With higher λ values, the users have better access to more number of highly popular contents, but to fewer of them with low popularity, that are stored across the system-wide A-UAVs and can be accessed via the mobile F-UAVs. A lower λ creates an opposite effect. The goal is to be able to choose a λ, that strikes the right balance between those effects and maximizes the overall availability.

3.4.2 Caching at Ferrying UAVs (F-UAVs)

The purpose of the F-UAVs is to ferry around $N_A.(1-\lambda).C_A$ unique contents stored in all N_A A-UAVs. In the presence of limited per-F-UAV caching space, C_F, its caching policy can be determined based on its trajectories, the value of λ, and the Zipf parameter defining the content popularity.

Consider a situation in which an F-UAV k is approaching towards the A-UAV i. Let U_i be the set of all unique contents in the entire system except the ones stored in A-UAV i. To maximize content availability for the users in A-UAV *i's* community, the F-UAV should carry as many low popularity contents from set U_i as its cache space permits. To enable such access, F-UAV k should carry C_F top popular contents from the set U_i while approaching A-UAV i. The size of the set U_i can be expressed as $|U_i| = (N_A - 1).(1-\lambda).C_A$. In scenarios when $C_F \leq |U_i|$, the F-UAV should carry the C_F top popular contents as outlined above. Otherwise, the F-UAV will carry all $|U_i|$ unique contents, leaving part of the F-UAV cache (i.e., $C_F - |U_i|$) empty. This causes underutilization of F-UAV cache space due to large λ values, leading to heavy in-A-UAV duplications, thus storing few unique contents.

3.4.3 Trajectory of Ferring UAVs

An F-UAV's trajectory is represented by the *sequence* of visited A-UAVs, and the *hovering duration* at each A-UAV. Trajectory sequence can be categorized as *partitioned* or *global* cycles. With a partitioned trajectory cycle, an F-UAV go around a specific part of the system containing a fixed subset of all the A-UAVs like F-UAVs *A* and *B* follow a partitioned cycle of A-UAVs *X, Y, Z* and *W* in Figure 3.1. With a global cycle, an F-UAV moves around all the A-UAVs in the system like F-UAVs *C* and *D* in Figure 3.1. Intuitively, if the contents cached in the unique segments of A-UAVs have very low popularity then the global sequence cycle would be beneficial. Conversely, when some of the A-UAVs maintain unique contents with comparatively very high popularity, then using partitioned cycle may be rewarding. These will be evaluated in the experiment in Section 3.6.

The cycle time of an F-UAV trajectory is $T_{cycle} = N_A^C \times (T_{Hover} + T_{Transit})$, where N_A^C is the number of A-UAVs in the cycle (partitioned or global), T_{Hover} is the hover duration at each A-UAV, and $T_{Transit}$ is the transit time between two consecutive A-UAVs in a sequence. $T_{Transit}$ depends on the F-UAV flying speed, inter-A-UAV distance, wind speed/directions, and other environmental factors. T_{Hover} should be set to a value which is determined by the data transfer rate and the amount of data needs to be exchanged between F-UAV to/from A-UAV. It should be noted that A-UAVs don't follow a trajectory since they are stationed at their respective communities for uninterrupted content dissemination.

3.5 Content Dissemination Performance

3.5.1 Content Availability

Availability is defined as the probability of finding a requested content within the local A-UAV or a future visiting F-UAV within a TAD. Consider a situation in which a single F-UAV cycles in a round-robin manner through all the A-UAVs with hovering and transit respectively. For a content requested from a community, the F-UAV may or may not be accessible within the specified TAD. This probability is as follows:

$$P_{FA} = \begin{cases} \frac{N_F \times (T_{Hover} + TAD)}{N_A \times (T_{Hover} + T_{Transit})} & for\ TAD < ((\frac{N_A}{N_F} - 1)T_{Hover} + \frac{N_A}{N_F} T_{Transit}) \\ 1 & for\ TAD \geq ((\frac{N_A}{N_F} - 1)T_{Hover} + \frac{N_A}{N_F} T_{Transit}) \end{cases} \tag{3.6}$$

If the TAD is larger than a specific duration, then the F-UAV's accessibility to the requesting community is guaranteed. Otherwise, it follows the first expression in Eqn. 3.6. Note that the physical accessibility to the F-UAV does not guarantee the access to the requested content since the F-UAV can store only a limited number (i.e., C_F) of unique contents. Let P_F be the probability that the requested content can be found within the F-UAV following a caching strategy as stated in Section 3.4. It can be expressed as:

$$P_F = \sum_{i=C_A+1}^{C_A+1+C_{EFF}} p_\alpha(i) \tag{3.7}$$

where, $p_\alpha(i)$ is the Zipf distributed popularity as defined in Section 3.3. The effective cache size of the F-UAV is given as: $C_{EFF} = min\{C_F, (N_A - 1) \times (1 - \lambda) \times C_A\}$. Effective cache size is less than C_F when F-UAVs cache is partly empty i.e., underutilized (see Section 3.4). Now, the probability that requested content can be found within a A-UAV that is local to the request generating community can be expressed as:

$$P_A = \sum_{i=1}^{\lambda \times C_A + (1-\lambda) \times C_A} p_\alpha(i) \tag{3.8}$$

Combining those three probabilities above, the overall availability can be stated as:

$$P_{Avail} = P_A + P_{FA} \times P_F \tag{3.9}$$

To summarize, local contents from A-UAVs (i.e., both duplicate and unique) and unique contents from future visiting F-UAVs contribute towards the overall availability P_{Avail} within a specified TAD. Note that all unavailable contents within the specified TAD will have to be downloaded by the A-UAVs using their expensive vertical links such as the satellite Internet. Therefore, availability indirectly indicates the content download cost in the system.

3.5.2 Low Availability Period

Consider the scenario in Figure 3.2 with two A-UAVs and one F-UAV. The users in a community have access to the content in the F-UAV for a duration of $TAD + T_{Hover}$. Time taken for the F-UAV to come back to the same community before the users in the community will have access to its content again is: $2.T_{Transit} + T_{Hover} - TAD$. This is the period during which the content availability for the users will only be from the local A-UAV, and that is without access to the F-UAV.

This duration is referred to as the *low availability period,* which can be generally expressed as:

$$LAP = \frac{N_A T_{Transit} + (N_A - 1) T_{Hover} - TAD}{N_F} \tag{3.10}$$

where N_A and N_F are the number of A-UAVs and F-UAVs in the system. With higher transit and hovering times and N_A, while the *low availability period* goes up, the overall availability, as derived in Eqns. 1 through 4, goes down.

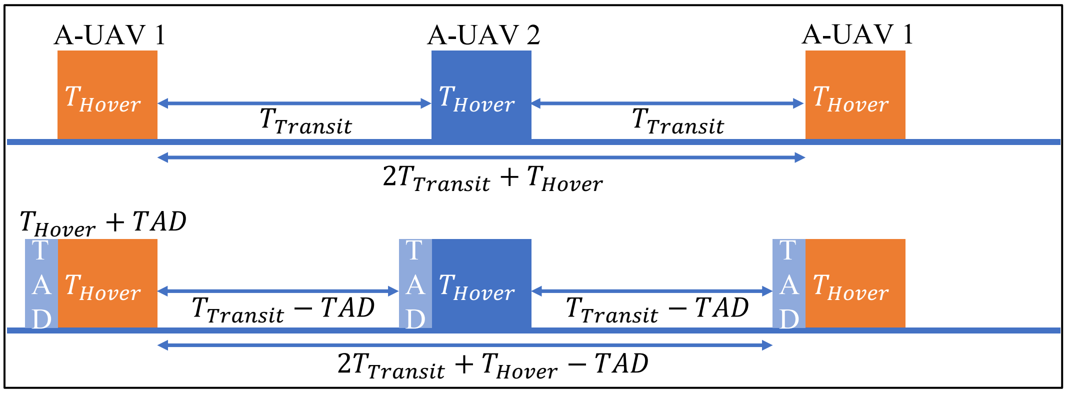


Figure 3.2. (Top) Scenario with $TAD = 0$; (Bottom) With non-zero TAD

3.5.3 Content Access Delay

Any request that is served by a local A-UAV experience zero access delay. There is no access delay if the request for content from F-UAV is generated when the F-UAV is hovering in the community. Therefore, the only scenario with a non-zero access delay would be the one in which the requested content is available at an F-UAV, and it is currently not visiting the requesting community. The probability of that scenario P_{CAD} can be expressed as:

$$P_{CAD} = \begin{cases} \frac{N_F \times TAD}{N_A \times (T_{Hover} + T_{Transit})} & for\ TAD < \frac{T_{cycle}}{N_F} - T_{Hover} \\ \frac{N_F \times \left(\frac{T_{cycle}}{N_F} - T_{Hover}\right)}{N_A \times (T_{Hover} + T_{Transit})} & for\ TAD \geq \frac{T_{cycle}}{N_F} - T_{Hover} \end{cases} \tag{3.11}$$

Note that the access delay is upper bounded by the specified TAD. As per the second expression in Eqn. 3.11, if the TAD is larger than the time it takes for the F-UAV to reach the request generating community, then content is delayed by the time taken by the F-UAV to reach. Conversely, for lower $TADs$, the content is delayed just by the TAD duration.

The average delay incurred in those two cases are:

$$Delay_{av} = \begin{cases} \frac{TAD}{2} & for\ TAD < \frac{T_{cycle}}{N_F} - T_{Hover} \\ \frac{\frac{T_{cycle}}{N_F} - X}{2} & for\ TAD \geq \frac{T_{cycle}}{N_F} - T_{Hover} \end{cases} \tag{3.12}$$

These averages are based on the maximum and the minimum possible delays. Combining P_{CAD} and $Delay_{av}$, the access delay (AD) can be expressed as:

$$AD = P_{CAD} \times \sum_{i=\forall C_F} p_\alpha(i) \times Delay_{av} \tag{3.13}$$

3.6 Experimental Results and Analysis

Experiments were carried out using simulations, for implementing the request generation, UAV caching, and F-UAV movement strategies presented in Sections 3.4. Default experimental parameters are $N_C = 1000$, $N_A = 20$, $N_F = 10, C_A = C_F = 50, \mu = 1, T_{Hover} = 20\ secs, T_{Transit} = 10\ secs, TAD = 20\ secs$ and $\alpha = 1.001$.

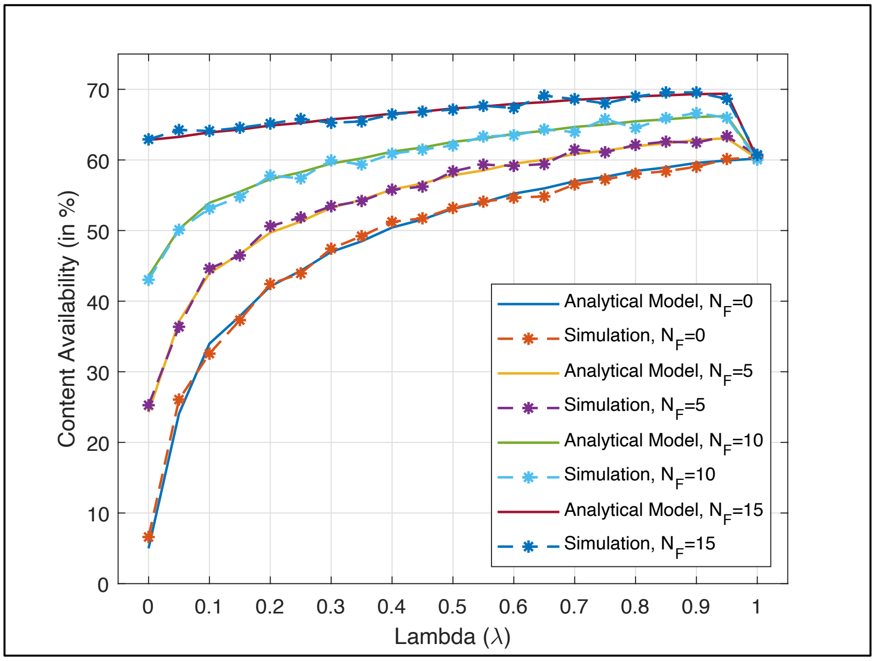


Figure 3.3. Content Availability with changing λ for different N_F

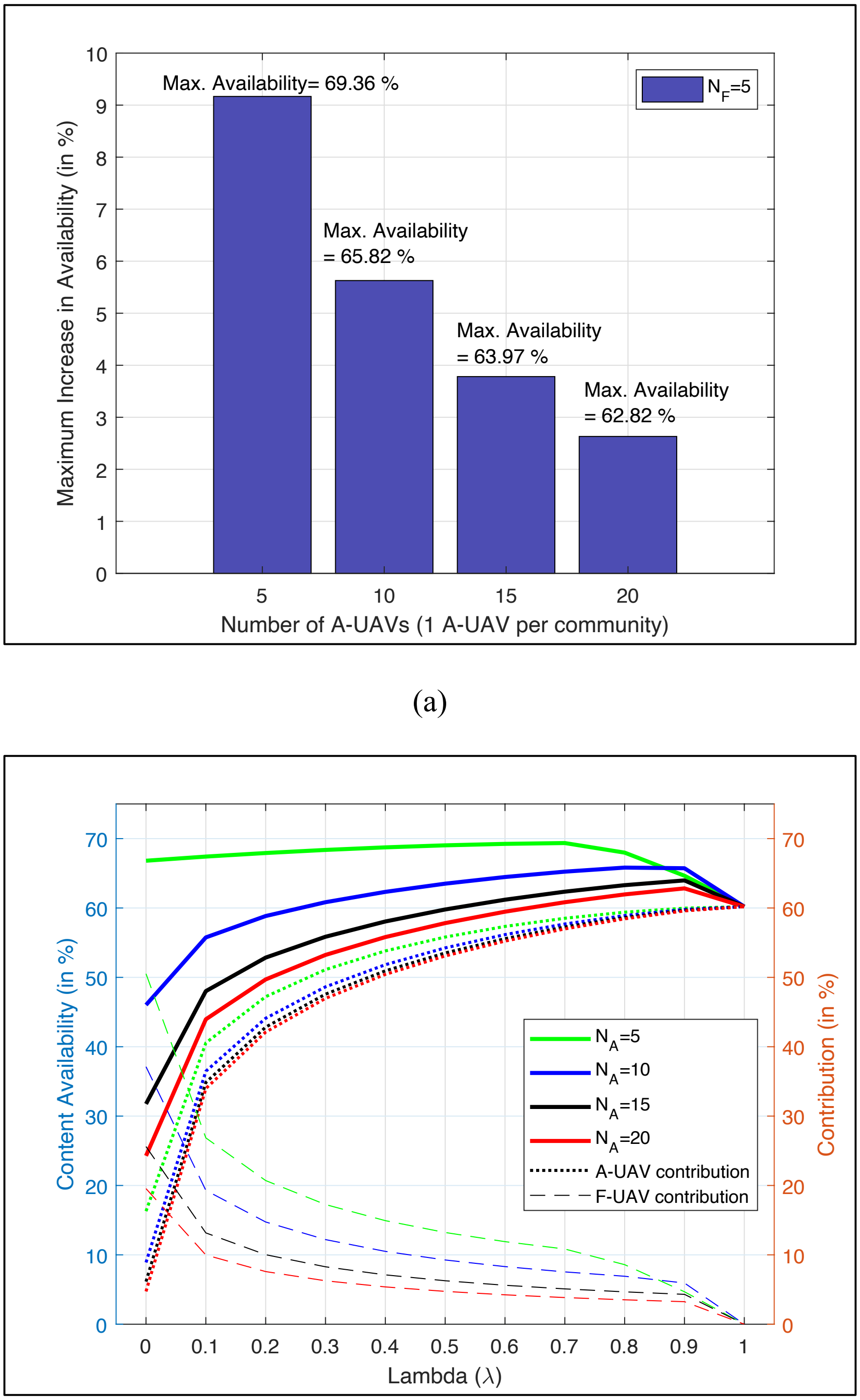
3.6.1 Impacts of F-UAVs on Content Availability

Figure 3.3 depicts the benefits of the ferrying UAVs in terms of improving content availability as defined in Section 3.4. The figures show availability computed analytically and from simulation experiments (i.e., average computed from the success of 10^5 requests for each availability point), both of which are validated through their excellent agreements.

Content availability is evaluated for varying λ values, representing the split between cached duplicated and unique objects within the A-UAVs, as described in Section 3.4. The following observations can be made from Figure 3.3. First, increasing F-UAVs can improve availability by ferrying contents that are not otherwise available to a community in its local A-UAV's cache. Second, the percentage increase in availability is more drastic for lower values of λ for which more unique contents are cached in the A-UAVs. Since the F-UAVs ferry around those unique contents across different communities, the dependance of availability on cached contents in the F-UAVs is more pronounced for smaller λ. Third, there is an optimum duplication factor λ, for which the content availability is the maximum for a given number of A-UAVs, F-UAVs, and default system parameters. Beyond the optimal operating point, availability reduces due to cache underutilization in F-UAVs, as shown in Section 3.4.

3.6.2 Impacts of the Number of User Communities

Figure 3.4(a) shows the impacts of the number of deployed A-UAVs (i.e., number of communities) on availability, while keeping the number of F-UAVs constant. These results are computed analytically from the equations provided in Section 3.5. The numbers show percentage increase in availability compared to the no-F-UAV case. The figure shows that the benefits of data ferrying consistently go down with increasing number of A-UAVs. The main reason for this is in the reduction in probability P_{FA} (i.e., in Eqn. 3.6) of physical access to the F-UAVs due to the increase in their overall cycle times. This can be mitigated using more F-UAVs and is shown later.

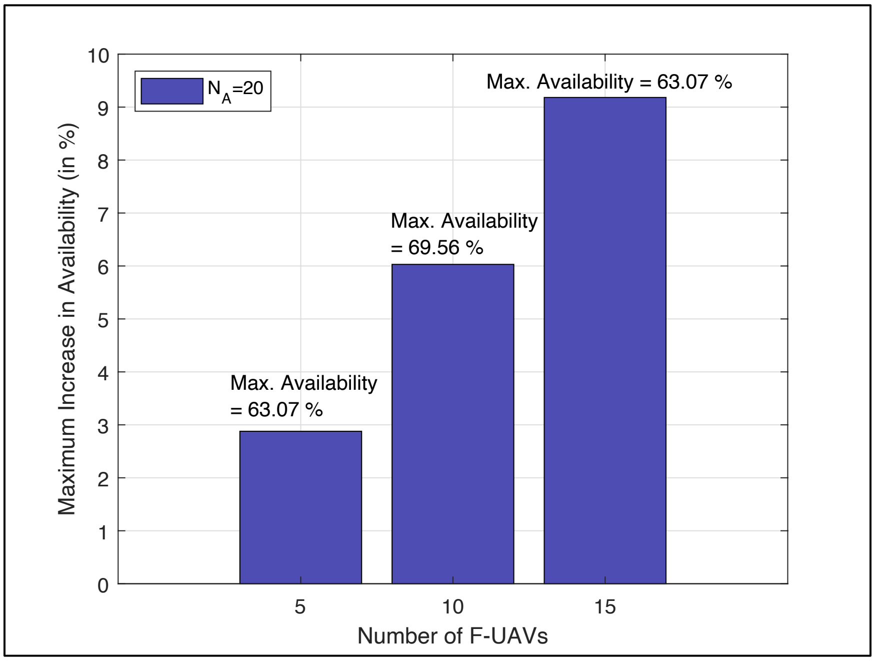


(a)

(b)

Figure 3.4. (a) Maximum increase in Availability with A-UAVs, (b) Contribution % from A- and F-UAVs, (c) Maximum increase in Availability with F-UAVs

Figure 3.4 (cont'd)



(c)

A content can be provisioned to a user either by its local A-UAV or by a visiting F-UAV. Hence, availability has an A-UAV component and an F-UAV component. These two are shown separately in Fig 3.4(b). As expected, as the amount of duplicated cached contents in the A-UAVs go up (i.e., with larger λ), the contribution from the A-UAVs go up accordingly. A-UAVs' contribution, however, is lesser for larger number of communities since the unique contents are uniformly randomly distributed across more A-UAVs as explained in Section 3.4. The contributions of the F-UAVs reduce because of the fall in P_{FA}, as stated for Figure 3.4(a).

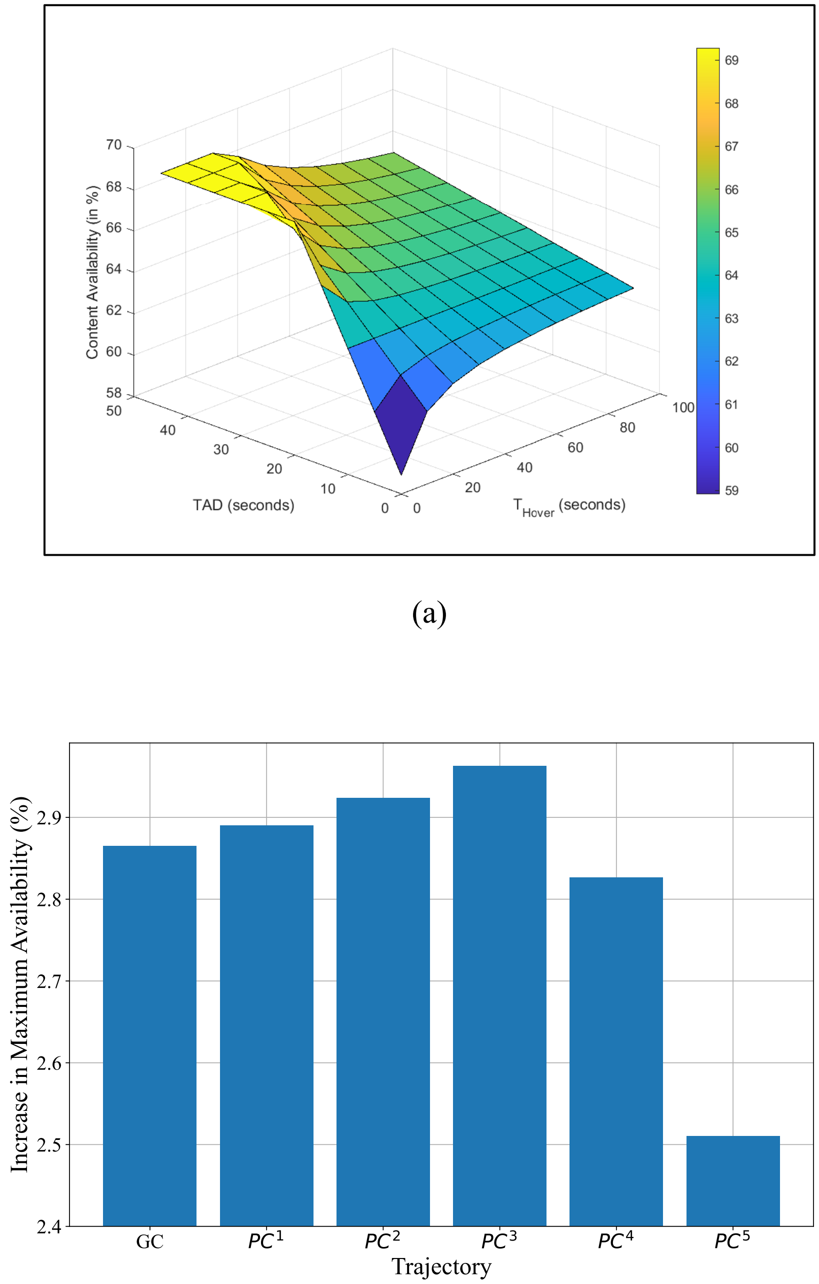
3.6.3 Impacts of Deploying Multiple F-UAVs

Deploying more F-UAVs increase the probability of physical access to the F-UAVs (i.e., P_{FA}), thus improving availability over the corresponding no-F-UAV scenarios which is shown in Figure 3.4(c). The results are with 20 A-UAVs, computed analytically and from simulation. The improvement in P_{FA} with increasing number of F-UAVs can be derived from Eqn. 3.6 as $\Delta P_{FA} =$

$\frac{(N_F^{new} - N_F^{old}) \times (TAD + T_{Hover})}{N_A \times (T_{Hover} + T_{Transit})}$. Here N_F^{new} and N_F^{old} are the number of deployed F-UAVs after and before additional deployments. ΔP_{FA} shows the rise in accessibility of F-UAVs to communities, which in turn, improves overall availability as given in Eqns. 1-4.

3.6.4 Effects of Hover Time and Tolerable Access Delay

Content availability is impacted by both the F-UAV hover time and the user-specified TAD in an interdependent manner. Those dependencies are shown in Figure 3.5(a) with $N_A = 10$, $N_F = 5$, and $\lambda = 0.8$. The figure shows non-monotonic behavior of availability with varying hovering time and TAD. One notable observation is that for low $TADs$, availability increases with increase in hover time T_{Hover} and otherwise for high $TADs$. This can be explained as follows. First, for $TAD < T_{Transit}$, when an F-UAV travels from community *i* to next community j, the F-UAV does not contribute to availability at community i or j for $T_{Transit} - TAD$ duration (see Figure 3.2). In this case, it is advantageous for the F-UAV to hover over a community. Second, for $TAD > T_{Transit}$, increase in hovering time reduces the possibility of the condition $(TAD - T_{Hover}) > T_{Transit}$ to be true. In other words, the possibility of exhausting the given TAD before reaching next community increases. So, it is beneficial to hover less, which increases the accessibility of F-UAVs at future communities in the cycle before TAD expires. Finally, for $TAD = T_{Transit}$, an F-UAV adds to availability within TAD irrespective of its hovering decision.

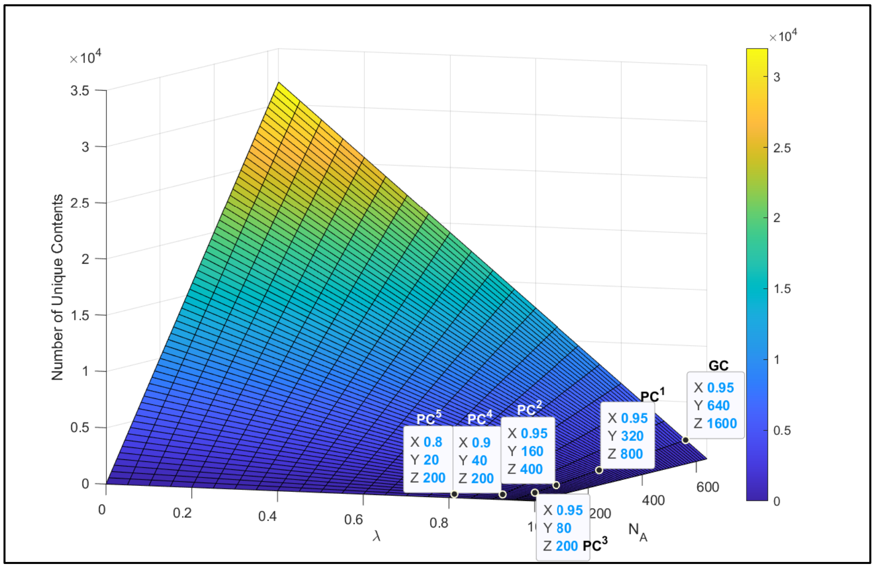


(a)

(b)

Figure 3.5. (a) Availability for variable T_{hover} and TAD, (b) Increase in availability for different trajectories, (c) Unique contents for varying N_A and λ

Figure 3.5 (cont'd)



(c)

3.6.5 Effect of F-UAV Trajectory on Content Availability

Trajectory of an F-UAV has an impact on what content it carries and its contribution to the overall content availability based on A-UAVs in its trajectory cycle. Figure 3.5(b) depicts those impacts for a system with 640 A-UAVs, 128 F-UAVs, $C_F = 200$, $\alpha = 0.8$ and all other default parameters calculated analytically. Increase in availability is reported as a percentage difference between the baseline no-F-UAV case and maximum content availability (i.e., at the optimal duplication factor λ) for specific F-UAV trajectories.

The global cycle (GC) trajectory refers to when an F-UAV visits all A-UAVs in a cycle. The content volume (i.e., $C_A.(1-\lambda).N_A$) to fill the F-UAVs in this trajectory scenario is quite high due to the large number of A-UAVs in the cycle, which is shown in Figure 3.5(c). The next trajectory that was experimented with is partitioned cycle-1 (PC^1). In this, the A- and F-UAVs are divided into two sets, which are 2 sets of 320 A-UAVs and 2 sets of 64 F-UAVs. F-UAVs from the first set cycle around the first set of the A-UAVs, and the same applies to the second sets of A-

UAVs and F-UAVs. Functionally, this scenario is equivalent to scaled down GC system with half as many A-UAVs and F-UAVs used for the GC results. In this scenario, the content availability is slightly larger than the GC case as can be seen in Figure 3.5(b). The reasons are as follows. First, due to less cycle duration probability P_{FA} increases (i.e., in Eqn. 3.6). Second, is the sufficiency of unique contents in the system due to adequate count of A-UAVs in the cycle (Figure 3.5(c)). Third, the optimal duplication factor λ is same for both i.e., 0.95. Thus, any increase in P_{FA} will increase content availability at optimal λ. The second and third partitioned cycles (PC^2 and PC^3) are functionally identical to PC^1 except that in these cases, both F-UAVs and A-UAVs are divided into 4 and 8 equal sets, respectively. Due to enough A-UAVs in the cycles to fill respective F-UAVs, content availability increases. Dividing the A-UAVs and F-UAVs further into 16 and 32 equal sets (i.e., in PC^4, PC^5) leads to reduction in availability due to the fewer A-UAVs in each F-UAV cycle (Figure 3.5(b)). In such cases, the cache space in the F-UAVs go underutilized at the optimal λ value. To ensure adequate filling up of F-UAV, a sub-optimal λ is chosen which reduces duplication. This can be seen in Figure 3.5(c) where λ, reduces from 0.95 to 0.90 for PC^4 and 0.80 for PC^5. This indicate that for a given number of A- and F-UAVs, there exists an optimal partitioning at which the overall content availability can be maximized.

3.6.6 Effects of Content Duplication on Access Delay

Figure 3.6 shows consistent reduction in average access delay with increasing A-UAV duplication factor λ which is computed from the analytical equations given in Section 3.5.

This reduction is explained as follows. With higher λ, less popular contents are cached in F-UAVs. As contents with low popularity are less likely to be requested according to Zipf distribution (see Section 3.3), the average access delay also goes down accordingly. Substantial reduction in access delay due to underutilization of F-UAV's cache, explained in Section 3.4, can be seen in Figure 3.6 for values above $\lambda = 0.95$. It can also be seen that with increase in number of F-UAVs, the average content access delay increases. As, delay is only due to contents that are cached in F-UAVs, more F-UAVs increase the quantity P_{CAD} which adds to access delay.

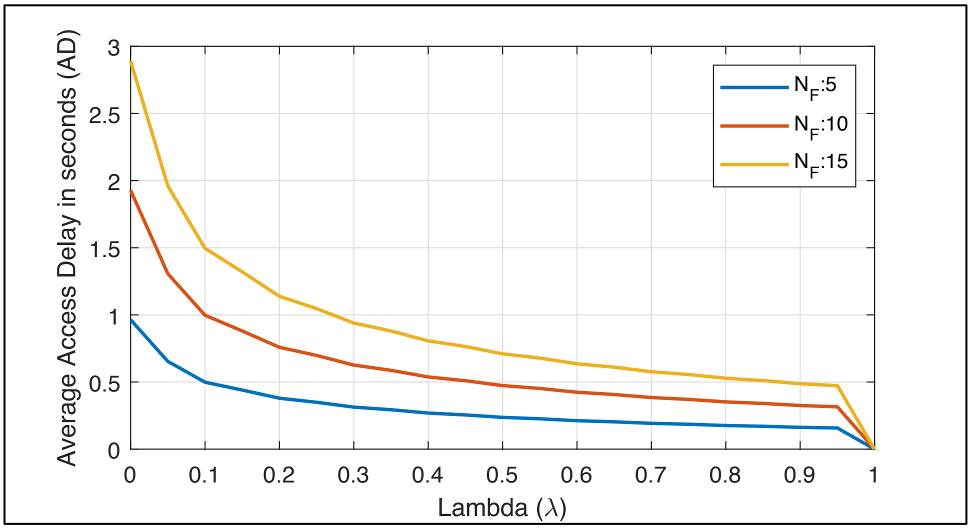


Figure 3.6. Increase in delay with increasing F-UAVs for varying λ

An F-UAV's hover time impacts its overall cycle duration, that affects the duration for which the content availability from that F-UAV to the users remains low. During such Low Availability Periods (LAP), as explained in Eqn. 3.10 in Section 3.5, only the locally cached contents from A-UAV's remain available. LAP reduces when more F-UAVs are deployed. This underlying effect is visible in Figure 3.3 where adding F-UAVs reduces LAP and boosts availability.

3.7 Summary and Conclusion

This chapter investigates caching policies in UAV networks for content dissemination in communication challenged systems. Cache-enabled UAVs serve communities of users in a disaster/war-stricken area by caching popular contents in order to reduce downloading needs using satellites and other expensive vertical links. A framework is adopted in which two types of UAVs, namely anchor UAVs and ferrying UAVs, are deployed. Through analytical modeling and simulation experiments, the chapter establishes an optimal content duplication strategy in which certain number of popular objects are duplicated in all anchor UAVs and certain number of non-duplicated/unique contents are carried in both types of UAVs. It was shown that content availability in such a system can be maximized by appropriately dimensioning the content duplication factor. The system was functionally validated, and performance evaluated for a different scenario including various ferrying UAV trajectories. The next chapter will extend this concept to a heterogenous demand scenario where requests can belong to different popularity distributions. Additionally, to emulate a more realistic scenario the generated requests can be accompanied with user-specific tolerable access delays. Furthermore, dynamic nuances of ferrying UAVs' trajectories are considered to enhance the collective content provisioning capability of the UAV-aided network for all the aforementioned design components.

Chapter 4: Using QoS-aware Caching for Handling Demand Heterogeneity in UAV-based Content Provisioning

4.1 Motivation

In disaster or conflict-affected areas, the collapse of communication infrastructures poses a significant barrier to timely information dissemination and recovery efforts. The deployment of Unmanned Aerial Vehicles (UAVs) as a plausible solution to form ad hoc networks has gained importance, given their ability to navigate and operate in areas without stable infrastructure. Despite this, existing UAV-based communication models are largely inept in environments with total infrastructure failure, especially when faced with the challenge of demand heterogeneity. Such heterogeneity is manifested through varying content popularity, urgency, and Quality of Service (QoS) expectations such as tolerable access delays. This requires a nuanced content caching approach beyond the capabilities of current systems, which rely on static, long-term request pattern estimations. There exists a need for an agile and adaptive UAV-aided content dissemination framework capable of addressing these multifaceted challenges directly, ensuring that critical information reaches all user communities efficiently and reliably.

4.2 Design Objective

The research in this chapter aims to conceptualize and develop a UAV-aided content caching system tailored for communication-challenged environments. This system is envisioned to efficiently accommodate the heterogeneous demands of isolated user communities, optimizing for content availability without excessive reliance on costly vertical connectivity. By leveraging the developed algorithmic approaches for content caching, the framework seeks to ensure high-availability content access across diverse user communities. A pivotal goal is to articulate the interdependencies between user demand patterns, users' urgencies and caching mechanisms, with a focus towards identifying optimal operational configurations that maximize content dissemination efficiency. Through rigorous simulation experiments and analytical modeling, the proposed system will be validated and evaluated, underscoring its potential as a resilient communication solution in the aftermath of disasters.

4.3 System Model

4.3.1 UAV Hierarchy

The content distribution system is organized in two layers, namely, the anchor UAVs (i.e., A-UAVs) and the ferrying UAVs (i.e., F-UAVs). As shown in Figure 3.1, each partitioned community of users is served by a A-UAV using a lateral wireless link such as Wi-Fi. A-UAVs can download content via an expensive vertical link such as satellite-based internet.

One monolithic system design approach is to let the A-UAVs download all needed content, as requested by their local users. In this approach, with no inter-A-UAV data transfer, the following shortcomings will be encountered. First, duplications of downloads will incur high download cost via the expensive A-UAV vertical links due to overlaps in requests from different communities. Second, storage constraints will cap the number of contents that can be downloaded and stored in each A-UAV, thus limiting the content availability. To address these, ferrying UAVs (i.e., F-UAVs) are introduced. Unlike A-UAVs, the mobile F-UAVs do not possess vertical links, but they have lateral links such as Wi-Fi, using which they can communicate with the A-UAVs and the users. These UAVs share the contents downloaded by A-UAVs serving other communities.

After receiving a request from one of its community users, an A-UAV first searches its local storage for the content. If not found, it waits for a potential future delivery of the content by one of the traveling F-UAVs. If no F-UAV with that content arrives within the specified *TAD*, the A-UAV downloads it.

4.3.2 Content Request and Provisioning Model

Content Popularity: Studies have shown [119], [122] that content request patterns follow a Zipf distribution in which a requested content's popularity is a geometric multiple of the next popular content in a larger pool [119]. Popularity of contents is given as:

$$p_\alpha(i) = \frac{\left(\frac{1}{i}\right)^\alpha}{\sum_{k\in C}\left(\frac{1}{k}\right)^\alpha} \tag{4.1}$$

The parameter C represents total number of contents in the pool, and Zipf parameter α determines skewness of the distribution. Popularity sequence of contents at different communities may vary, which introduces popularity heterogeneity, which is the focus of this chapter.

Content Requests: Poisson distributed request generation is a prevalent way to capture user requests in practical networks.

Tolerable Access Delay and Content Provisioning: For each generated request, a Tolerable Access Delay (*TAD*) [123], [124] is specified. *TAD* is a Quality-of-Service parameter that indicates the duration that a user is ready to wait before a requested content can be accessed. Here, if a content is not available from the UAVs within the specified *TAD*, it will be downloaded from a central server using the expensive vertical links of A-UAVs.

4.4 Caching Policies to Handle Heterogeneity

This chapter focuses on following caching related design questions: a) which content to be downloaded and cached in A-UAVs so that they can serve their own community directly, and the remote communities via traveling F-UAVs; b) which contents to be cached when the popularity and *TAD* of contents vary at different communities; c) which content to be transferred from the A-UAVs to the F-UAVs; and, d) what inter-community trajectories to be followed by the F-UAVs.

This chapter addresses these questions with pre-assigned and globally known heterogeneous content popularities, and content pre-placements at A-UAVs. Different pre-programmed F-UAV trajectories are characterized with a multitude of content placement strategies. After understanding such scenarios in this chapter, runtime and dynamic mechanisms has been developed and reported in future chapters.

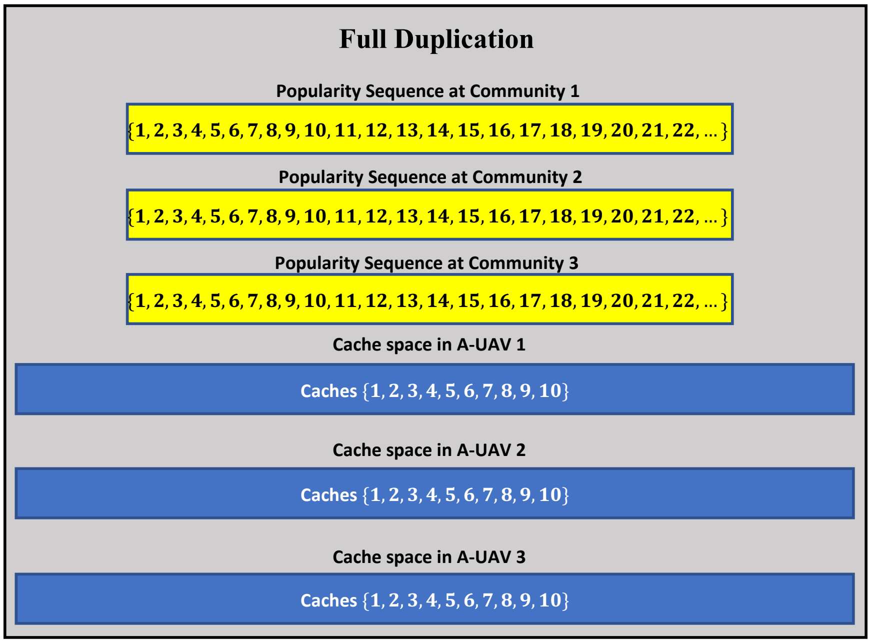


Figure 4.1. Example of FD at 3 A-UAVs with 10 cached contents in the system

4.4.1 Caching at Anchor UAVs (A-UAVs)

A naïve fully duplicated (FD) [119] mechanism limits the number of accessible contents for all user communities to C_A, the A-UAV cache size, due to the duplication of requested contents form the corresponding user communities (see Figure 4.1). This limitation can be addressed by storing a certain number of exclusive contents in all the A-UAVs and share those contents across the communities via the traveling F-UAVs. This *Smart Exclusive Caching* (SEC) mechanism can effectively increase the access of contents for all users across the entire system, thus improving the overall availability within a given *TAD*.

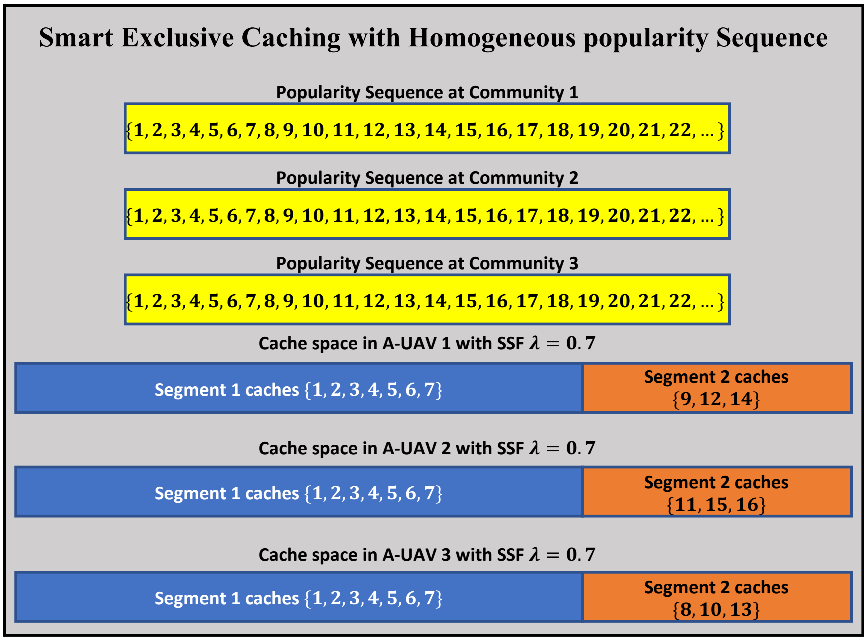

Smart Exclusive Caching with Homogeneous popularity Sequence
Popularity Sequence at Community 1
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, … }
Popularity Sequence at Community 2
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, … }
Popularity Sequence at Community 3
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, … }
Cache space in A-UAV 1 with SSF λ = 0.7
Segment 1 caches {1, 2, 3, 4, 5, 6, 7}
Segment 2 caches {9, 12, 14}
Cache space in A-UAV 2 with SSF λ = 0.7
Segment 1 caches {1, 2, 3, 4, 5, 6, 7}
Segment 2 caches {11, 15, 16}
Cache space in A-UAV 3 with SSF λ = 0.7
Segment 1 caches {1, 2, 3, 4, 5, 6, 7}
Segment 2 caches {8, 10, 13}

Figure 4.2. Content Caching Policy in 3 A-UAVs with Cache Size $C_A = 10$ for Homogeneous Content Popularity Sequence

Suppose we consider a disaster/war-stricken area with *homogeneous content popularity sequence* across all the user- communities and an A-UAV is assigned to each community for content provisioning. According to the SEC mechanism, cache space of an A-UAV has two segments i.e., *Segment-1* and *Segment-2*. Let the size of Segment-1 of A-UAV cache be $C_{S1} = \lambda . C_A$ and that of Segment-2 be $C_{S2} = (1 - \lambda). C_A$, where λ is a *storage segmentation factor* (SSF) that decides the size of Segment-1 of A-UAVs. Top $\lambda . C_A$ popular contents are cached at Segment-1 of A-UAVs. These contents are same across all A-UAVs whereas contents from Segment-2 are different. This results into $C_{S2}^{total} = N_A . (1 - \lambda). C_A$ number of total Segment-2 contents stored across all N_A number of A-UAVs, and these can be shared across all user communities via the mobile F-UAVs. These contents have popularities after the top $\lambda . C_A$ popular Segment-1 contents in all the A-UAVs. For symmetry, all $N_A . (1 - \lambda). C_A$ Segment-2 contents are uniformly randomly distributed across N_A number of A-UAVs. Total number of contents in this content dissemination system is as follows:

$$C_{sys} = \lambda . C_A + N_A . (1 - \lambda) . C_A \tag{4.2}$$

Figure 4.2 shows an example of this caching policy with 3 A-UAVs and storage segmentation factor $\lambda = 0.7$. Contents in Segment-1 are same across all three A-UAVs while those in Segment-2 part of the A-UAV storage are different. Total contents across all A-UAVs are $\{1 - 16\}$ according to Eqn. 4.2.

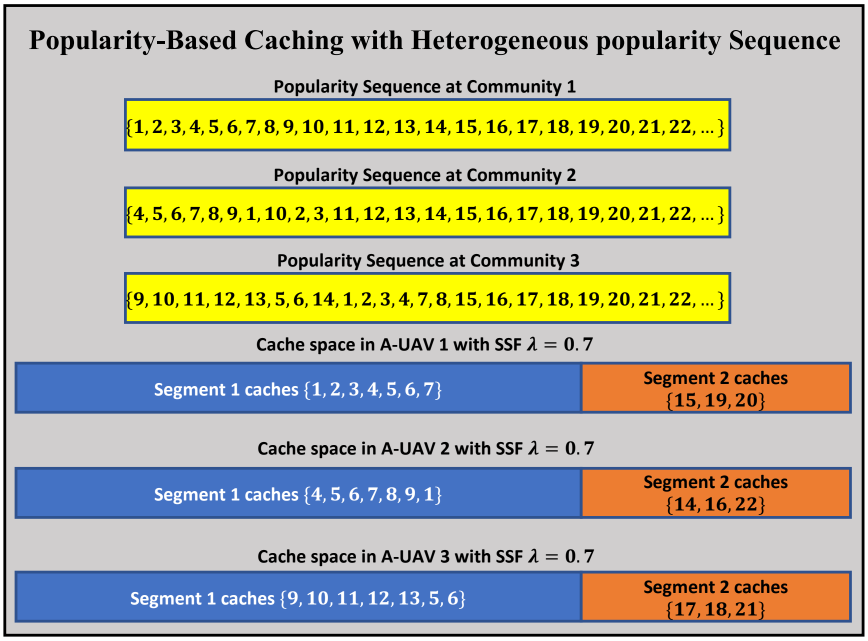

Popularity-Based Caching with Heterogeneous popularity Sequence
Popularity Sequence at Community 1
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, ...}
Popularity Sequence at Community 2
{4, 5, 6, 7, 8, 9, 1, 10, 2, 3, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, ...}
Popularity Sequence at Community 3
{9, 10, 11, 12, 13, 5, 6, 14, 1, 2, 3, 4, 7, 8, 15, 16, 17, 18, 19, 20, 21, 22, ...}
Cache space in A-UAV 1 with SSF λ = 0.7
Segment 1 caches {1, 2, 3, 4, 5, 6, 7}
Segment 2 caches {15, 19, 20}
Cache space in A-UAV 2 with SSF λ = 0.7
Segment 1 caches {4, 5, 6, 7, 8, 9, 1}
Segment 2 caches {14, 16, 22}
Cache space in A-UAV 3 with SSF λ = 0.7
Segment 1 caches {9, 10, 11, 12, 13, 5, 6}
Segment 2 caches {17, 18, 21}

Figure 4.3. Content Caching Policy in 3 A-UAVs with Cache Size $C_A = 10$ for Heterogeneous Content Popularity Sequence

4.4.2 Caching to Cater to Heterogeneous Popularity Sequences

Caching policy described so far assumes that the contents have the same popularity sequence for the requests across all communities. In practice, requests can be heterogeneous in that the popularity sequence of requested contents from different communities can be different. For example, in case of a fire breakout, information about fire trucks and medical care are the most popular contents for the areas in the vicinity of fire. But, for areas which is in the path of fire spread needs logistical support for relocation. In such heterogeneous popularity cases, previous caching policy may not be the best fit since most popular contents may not be the same for all communities.

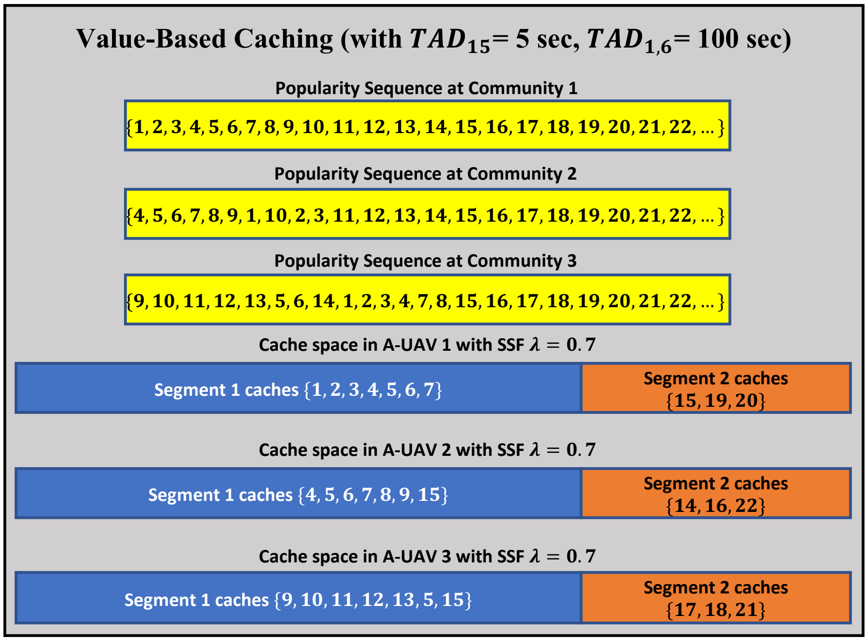
This limitation can be addressed by caching a community's most popular contents in the Segment-1 of its local A-UAV. Figure 4.3 shows a scenario where there are three A-UAVs at their respective communities with different content popularity sequence. These A-UAVs have cached most popular contents according to their communities' content popularity sequence. It can be observed that contents $\{5, 6\}$ are cached in all the A-UAVs, contents $\{1, 2, 3\}$ are cached in A-UAV 1 and contents $\{4,\ 7\}$ are cached in A-UAVs 1 and 2. Contents $\{1, 2, 3,4,7\}$ are called *exclusive* contents of Segment-1 that are cached in one or some of the A-UAVs, but not in all of them, whereas contents $\{5, 6\}$ are called *non-exclusive* contents of Segment-1 that are cached at all A-UAVs. Therefore, unlike SEC, the number of contents in Segment-1 across all A-UAVs may be more than $\lambda . C_A$ i.e., $C_{S1}^{total} = C_{NE} + C_E^{total} \geq \lambda . C_A$. Like SEC, contents in Segment-2 do not repeat across A-UAVs. If $C_{NE}\ and\ C_E^{total}$ are the number of non-exclusive and total exclusive contents in Segment 1, then total number of contents in the system:

$$C_{sys} = C_{NE} + C_E^{total} + N_A . (1 - \lambda) . C_A \geq \lambda . C_A + N_A . (1 - \lambda) . C_A \tag{4.3}$$

Validation of Eqn. 4.3 can be seen in Figs. 4.2 and 4.3. The stored contents across all A-UAVs is $\{1 - 16\}$ with SEC (i.e., Figure 4.2). In the heterogeneous case (i.e., Figure 4.3), the caches contents are $\{1 - 22\}$. The objective here is to choose a λ, that strikes the right balance between the two segments and maximizes the overall content availability in the heterogeneous case.

One limitation of popularity-based caching is related to the tolerable access delay (TAD). The popularity-based caching does not consider content specific TAD. This shortcoming leads to reduced availability for contents with both low TAD and popularity. This is explained using the following example.

Let us consider a content $'x'$ which has popularity higher than $\lambda.C_A$ and another content $'y'$ with popularity lower than $\lambda.C_A$. According to popularity-based caching, content $'x'$ is cached in Segment-1 of an A-UAV, and content $'y'$ is cached in Segment-2 of one of the A-UAVs. Therefore, content $'y'$ is ferried by F-UAVs across all communities. Let the inter-community distances be such that an F-UAV 'j' reaches a community within 30 $seconds$ of departure of the previous F-UAV 'j-1'. If the TAD associated with $'x'$ is 100 $seconds$ and $'y'$ is 5 $seconds$, a request for $'y'$ will rarely be served by the A-UAV/F-UAV content dissemination network. This will lead to $'y'$ being downloaded. This issue is addressed by a value-based caching strategy proposed in the next subsection.



Value-Based Caching (with TAD_{15}= 5 sec, $TAD_{1,6}$= 100 sec)
Popularity Sequence at Community 1
$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, \dots\}$
Popularity Sequence at Community 2
$\{4, 5, 6, 7, 8, 9, 1, 10, 2, 3, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, \dots\}$
Popularity Sequence at Community 3
$\{9, 10, 11, 12, 13, 5, 6, 14, 1, 2, 3, 4, 7, 8, 15, 16, 17, 18, 19, 20, 21, 22, \dots\}$
Cache space in A-UAV 1 with SSF $\lambda = 0.7$
Segment 1 caches $\{1, 2, 3, 4, 5, 6, 7\}$
Segment 2 caches $\{15, 19, 20\}$
Cache space in A-UAV 2 with SSF $\lambda = 0.7$
Segment 1 caches $\{4, 5, 6, 7, 8, 9, 15\}$
Segment 2 caches $\{14, 16, 22\}$
Cache space in A-UAV 3 with SSF $\lambda = 0.7$
Segment 1 caches $\{9, 10, 11, 12, 13, 5, 15\}$
Segment 2 caches $\{17, 18, 21\}$

Figure 4.4. Example of VBC with low TAD content '15' at Segment-1 of A-UAV

4.4.3 Value-based Caching to Handle Heterogeneous TAD

All the policies discussed so far makes caching decision of a content based on its popularity at the community where it is requested. However, the promptness with which a content needs to be provisioned may not always be positively correlated with its popularity. For example, request for logistical support information can be more popular than the information about first responders in a post-disaster situation. However, the TAD for first responder information is expected to be shorter. To prioritize caching of such contents in Segment-1 of A-UAVs, this chapter devices a value-based caching policy where the value of a requested content is calculated from its popularity and its associated TAD. Value of a content $'i'$ can be expressed as:

$$V(i) = \kappa\upsilon \times \frac{p_\alpha(i)}{TAD(i)} = \kappa \times \frac{TAD_{min}}{p_\alpha(1)} \times \frac{p_\alpha(i)}{TAD(i)} \qquad (4.4)$$

Here, $p_\alpha(i)$ is the popularity of the content as per Zipf Distribution, $TAD(i)$ is the tolerable access delay associated with the content request, $\kappa \in [0,1]$ is a scalar weight which increases with decrease in popularity and $'\upsilon'$ is a *normalization constant*. For a given Zipf (popularity) parameter α, the normalization constant is calculated from the minimum possible TAD (TAD_{min}) and the maximum possible popularity, which is $p_\alpha(1)$. The quantity $V(i)$ is bounded between $[0, 1]$ and it increases with increase in $p_\alpha(i)$ and decrease in $TAD(i)$. This value-based caching policy increases the likelihood of contents requested with low TAD to be cached *in Segment-1* of the A-UAVs, thus making them more readily available (see Figure 4.4).

4.4.4 Caching at Ferrying UAVs (F-UAVs)

The purpose of the F-UAVs is to ferry around $C_E^{total} + N_A.(1-\lambda).C_A$ number of contents stored across N_A number of A-UAVs (see Eqn. 4.3). Due to the limitation of per-F-UAV caching space (i.e., C_F), its caching policy should be determined based on its trajectories, the value of λ, the Zipf popularity, and the $TADs$ associated with the contents to be cached.

F-UAV caching policy is explained in the pseudocode below.

Algorithm 4.1. F-UAV Caching Algorithm with Value-based policy at A-UAV

```
Input: Total A-UAVs in its trajectory, $TAD$, next A-UAV $'i'$, present A-UAV $'i-1'$
Output: $C_F$ contents for F-UAV $'j'$
Initialize $C_A$ contents in each A-UAV based on value of contents
while True:
    if F-UAV leaving for next A-UAV $'i'$ then do
        for $k = 0\ to\ length$(A-UAV $'i'$ cache $C_A^i$) do
            Check if $k$ in $C_F$ cache space of F-UAV $'j'$
            if true then do
                Replace $'k'$ with highest value content from $C_A^{i-1}$
                 not cached in F-UAV $'j'$ and A-UAV $'i'$
                end if
            end for
        end if
        Update next A-UAV $'i'$, present A-UAV $'i-1'$
    end while
```

Consider a situation in which an F-UAV *'j'* is approaching towards the A-UAV *'i'*. Let U_i be the set of all exclusive contents in Segment-1 of all A-UAVs and all contents from Segment-2 of all A-UAVs in the entire system except the ones stored in A-UAV *'i'*. To maximize content availability for the users in A-UAV *i's* community, the F-UAV should carry C_F top valued contents (refer Eqn. 4.4) from the set U_i while approaching A-UAV *i*. The size of the set U_i can be expressed as $|U_i| = C_E^{total} + (N_A - 1).(1-\lambda).C_A$. In scenarios when $C_F \leq |U_i|$, the F-UAV should carry the C_F top popular contents as outlined above. Otherwise, the F-UAV should carry all $|U_i|$ contents, leaving part of the F-UAV cache (i.e., $C_F - |U_i|$) empty. This causes underutilization of F-UAV cache space.

The next section discusses the deployment and trajectory planning methods for F-UAVs employed in this chapter to boost content availability for the requesting users.

4.5 Deployment and Trajectory Planning of UAVs

Ferrying UAVs travel across communities to share contents among the A-UAVs. In this setting, the trajectory of F-UAVs can greatly impact content availability to the requesting users.

4.5.1 Trajectory Sequence and Cycle

An F-UAV's trajectory cycle is represented by the *sequence* of A-UAVs that it visits. The cycle time of an F-UAV trajectory is $T_{cycle} = N_A^C \times (T_{hover} + T_{transition})$, *where* N_A^C is the number of A-UAVs in the F-UAV's sequence, T_{hover} is the hover duration at each A-UAV, and $T_{transition}$ is the transition time between two consecutive A-UAVs in the F-UAV's sequence. $T_{transition}$ depends on the F-UAV flying speed, intercommunity distance, wind speed/directions, and other environmental factors. T_{hover} is the minimum duration necessary for completing data exchanged between UAVs.

4.6 Content Dissemination Performance

4.6.1 Content Availability

Content availability is defined as the probability of finding a requested content from the UAV-aided caching paradigm within the specified Tolerable Access Delay (TAD). F-UAV's accessibility within a given TAD while transitioning in round-robin manner across A-UAVs in its trajectory is expressed as:

$$P_{FA} = \begin{cases} \frac{N_F \times (T_{Hover} + TAD)}{N_A \times (T_{Hover} + T_{transition})} & for\, TAD < \left(\frac{N_A}{N_F} - 1\right) T_{hover} + \frac{N_A}{N_F} T_{transition} \\ 1 & for\, TAD \geq \left(\frac{N_A}{N_F} - 1\right) T_{hover} + \frac{N_A}{N_F} T_{transition} \end{cases} \tag{4.5}$$

It can be seen from the condition given in Eqn. 4.5 that when the interval between two visits by an F-UAV to an A-UAV is less than TAD, the contents cached in F-UAV are always accessible. However, when TAD is less than the said interval, the contents in F-UAVs are partially accessible. Note that the physical accessibility to F-UAVs does not guarantee the access to a requested content since the F-UAVs can store only a limited number (i.e., C_F) of contents. Let P_F be the probability that the requested content can be found within a F-UAV. It can be expressed as:

$$P_F = \frac{\sum_{i=C_A+1}^{C_A+1+C_{EFF}} V(i)}{\sum_{i=1}^{C} V(i)} = \frac{\sum_{i=C_A+1}^{C_A+1+C_{EFF}} \kappa \times \frac{p_\alpha(i)}{TAD(i)} \times \frac{TAD_{min}}{p_\alpha(1)}}{\sum_{i=1}^{C} \kappa \times \frac{p_\alpha(i)}{TAD(i)} \times \frac{TAD_{min}}{p_\alpha(1)}} \tag{4.6}$$

where, $V(i)$ is value of the content (Eqn. 4.4); C_A, C_E and C are cache size, exclusive contents in Segment 1 of each A-UAV and total contents respectively. Effective cache size of F-UAV is:

$$C_{EFF} = min\{C_F, (N_A - 1).(C_E + (1 - \lambda).C_A)\} \tag{4.7}$$

Now, P_A, probability of finding a requested content in the local A-UAV of the request generating community, is expressed as:

$$P_A = \frac{\sum_{i=1}^{\lambda \times C_A + (1-\lambda) \times C_A} V(i)}{\sum_{i=1}^{C} V(i)} = \frac{\sum_{i=1}^{\lambda \times C_A + (1-\lambda) \times C_A} \kappa \times \frac{p_\alpha(i)}{TAD(i)} \times \frac{TAD_{min}}{p_\alpha(1)}}{\sum_{i=1}^{C} \kappa \times \frac{p_\alpha(i)}{TAD(i)} \times \frac{TAD_{min}}{p_\alpha(1)}} \tag{4.8}$$

Combining Eqns. 4.5, 4.6 and 4.8, the overall availability is:

$$P_{Avail} = P_A + P_{FA} \times P_F \tag{4.9}$$

To summarize, local contents from A-UAVs and contents from future visiting F-UAVs contribute towards the overall availability P_{Avail} within the specified $TADs$.

Note that all unavailable contents within specified $TADs$ will be downloaded by the A-UAVs using their expensive vertical links such as a satellite Internet. Thus, availability indirectly indicates the content download cost in the system.

4.6.2 Low Availability Period

Consider the scenario in Figure 4.5 with two A-UAVs and one F-UAV. The users in a community can have access to the content in the F-UAV for a duration of $TAD + T_{hover}$. Time taken for the F-UAV to come back to the same community before the users in the community will have access to its content again is: $2.T_{transition} + T_{hover} - TAD$. During this period content availability will be restricted from the local A-UAV.

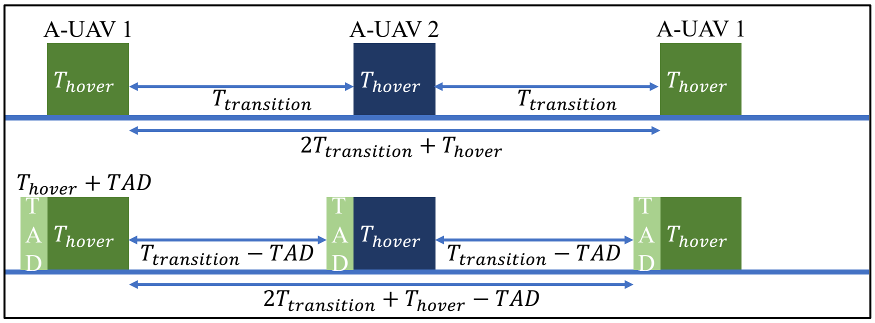


Figure 4.5. (Top) Scenario with $TAD = 0$; (Bottom) With non-zero TAD

This duration is referred to as the *low availability period,* which can be generally expressed as:

$$LAP = \frac{N_A T_{transition} + (N_A - 1) T_{hover} - TAD}{N_F} \tag{4.10}$$

where N_A and N_F are the number of A-UAVs and F-UAVs in the system. Low availability period is an effect of long cycle duration, and do not necessarily indicate the actual content availability in the system.

4.6.3 Content Access Delay

Any request that is served by a local A-UAV or found within a visiting F-UAV experience zero access delay. The only scenario with a non-zero access delay would be the one in which the requested content is available at an F-UAV, and it is currently not visiting the requesting community.

The probability of that scenario P_{CAD} can be expressed as:

$$P_{CAD} = \begin{cases} \frac{N_F \times TAD}{N_A \times (T_{Hover} + T_{Transit})} & for\ TAD < \frac{T_{cycle}}{N_F} - T_{Hover} \\ \frac{N_F \times \left(\frac{T_{cycle}}{N_F} - T_{Hover}\right)}{N_A \times (T_{Hover} + T_{Transit})} & for\ TAD \geq \frac{T_{cycle}}{N_F} - T_{Hover} \end{cases} \tag{4.11}$$

As per the second expression in Eqn. 4.11, if the TAD is larger than the time it takes for the F-UAV to reach the request-generating community, then content is delayed by the time taken by the F-UAV to reach the requesting community. Conversely, for lower $TADs$, the content is delayed just by the TAD duration. Average delay incurred in those two cases are:

$$Delay_{av} = \begin{cases} \frac{TAD}{2} & for\ TAD < \frac{T_{cycle}}{N_F} - T_{Hover} \\ \frac{\frac{T_{cycle}}{N_F} - X}{2} & for\ TAD \geq \frac{T_{cycle}}{N_F} - T_{Hover} \end{cases} \tag{4.12}$$

Using P_{CAD} and $Delay_{av}$, access delay is calculated as follows:

$$AD = P_{CAD} \times \frac{\sum_{i=\forall C_F} \kappa \times \frac{p_\alpha(i)}{TAD(i)} \times \frac{TAD_{min}}{p_\alpha(1)}}{\sum_{i=1}^{C} \kappa \times \frac{p_\alpha(i)}{TAD(i)} \times \frac{TAD_{min}}{p_\alpha(1)}} \times Delay_{av} \tag{4.13}$$

Here, access delay is calculated for all content specific *TADs*.

4.7 Experimental Setup

For experimentation, specific modules were developed for implementing request generation, UAV caching, and F-UAV movement strategies presented in Sections 4.3, 4.4 and 4.5. Unless mentioned otherwise, default experimental parameters in Table 4.1 are used for all experimentations. Each of the data points in the presented results represent an average computed from simulation experiments with 10^6 content requests.

Table 4.1. Default Values for Model Parameters

#	**Variables**	**Default Value**
1	Total number of contents, C	2000
2	Number of A-UAVs, N_A	20
3	Number of F-UAVs, N_F	10
4	Cache space in A-UAV, C_A	100
5	Cache space in F-UAV, C_F	100
6	Poisson request rate parameter, μ	1
7	Hover time of F-UAV, T_{Hover}	20 seconds
8	Transition time of F-UAV, $T_{Transition}$	10 seconds
9	Tolerable Access Delay, TAD	240 seconds
10	Zipf parameter (Popularity), α	0.7
11	Ferrying UAV Trajectory	Round-robin

4.7.1 Heterogeneity in Content Popularity Sequence

To emulate real-life heterogeneity, a swap-based mechanism is used. Two parameters, namely, a swap probability μ and a swap difference δ are used to create different popularity sequences from a given sequence. Swap probability μ is the probability with which the popularities of two contents (e.g., 'x' and 'y') within the original popularity sequence are swapped. The swap difference δ is used to determine which content (e.g., 'y') to swap for content 'x'. The difference between the original sequence and the new sequence using the above method is determined using Smith-Waterman Distance [125]. To capture heterogeneity in content popularity sequence different communities are programmed with different popularity sequences obtained using the method stated above.

4.7.2 Heterogeneity in Tolerable Access Delay (TAD)

This chapter uses a binary request TADs (i.e., low and high-TAD) to incorporate heterogeneity in tolerable access delay. Experiments are conducted by broadly classifying the contents into 5 popularity classes, such that Class-1 contains contents with highest popularity, and Class 5 contents are the least popular. At any given time, γ % of requests for contents from only one class have low TAD. Remaining requests for contents from the said class and all other classes have high TAD.

4.8 Experimental Results and Analysis

4.8.1 Impacts of Value-Based Caching

The overall increase in content availability using value-based content caching and joint-deployment of ferrying UAVs is shown in Figure 4.6. The performance improvement is compared against popularity-based caching policy at A-UAVs and round-robin trajectories of F-UAVs. Content availability is evaluated for varying cache size of the UAVs.

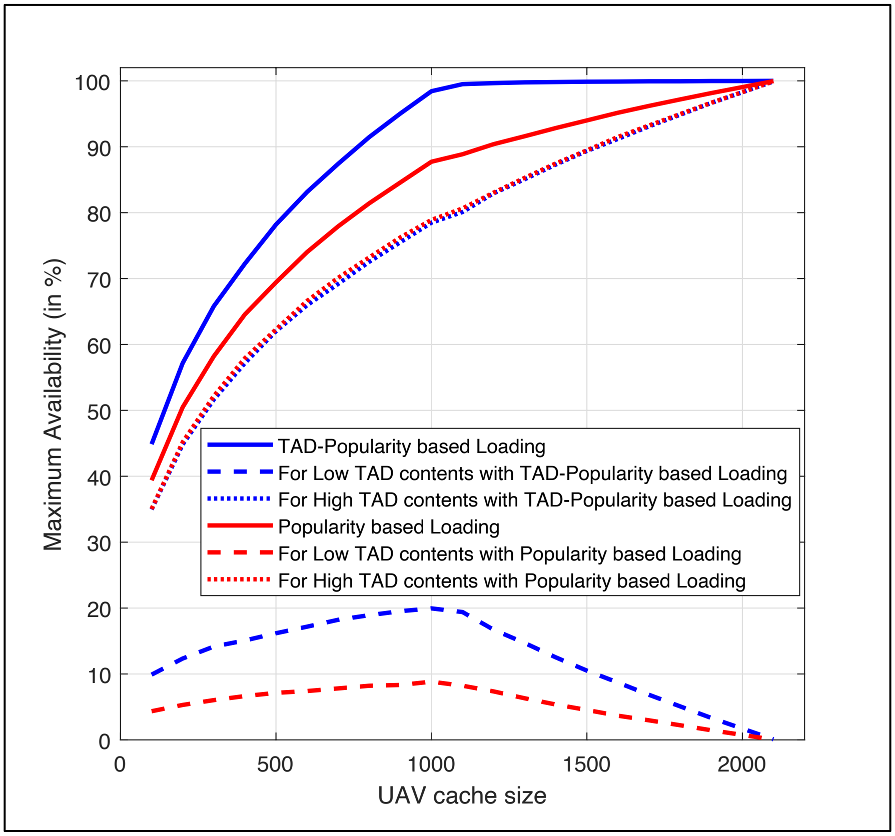


Figure 4.6. Improvement in maximum availability of contents by loading UAVs using value (TAD+Popularity) of contents

It can be seen from Figure 4.6 that a maximum increase in availability of approximately 12% can be achieved by using value-based caching policy at the A-UAVs while F-UAVs are following their respective round-robin trajectory. The benefits of value-based caching in scenarios with multi-dimensional demand heterogeneities are attributed to various factors including heterogeneity in popularity sequence, TAD associated with the content requests, popularity of low TAD contents, and value of a content. The effects of these factors are depicted individually in the following sub-sections.

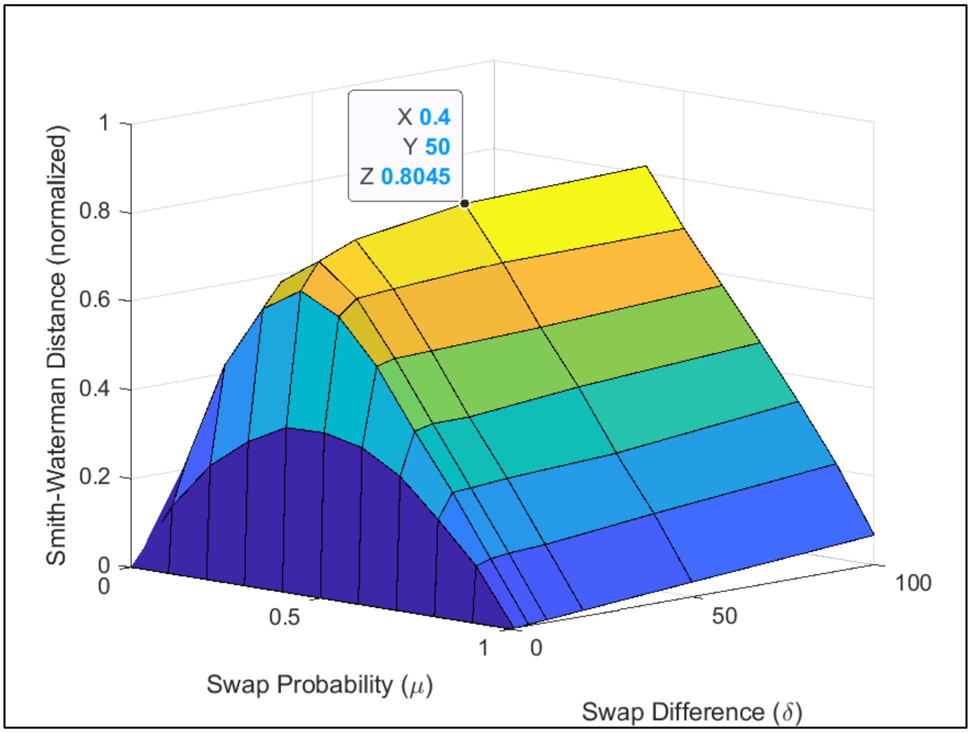


Figure 4.7. Difference between two sequences with varying μ and δ

4.8.2 Effects of Heterogeneity in Content Popularity Sequence

Different popularity sequences are generated using the parameters swap probability μ and swap difference δ.

Figure 4.7 shows the normalized Smith-Waterman distance between the sequences. Maximum difference between two sequences is recorded at $\mu = 0.5$. The difference does not vary substantially with Swap Difference δ for a particular value of μ. However, it shows an increasing trend with increase in δ in the beginning, and then a reduction.

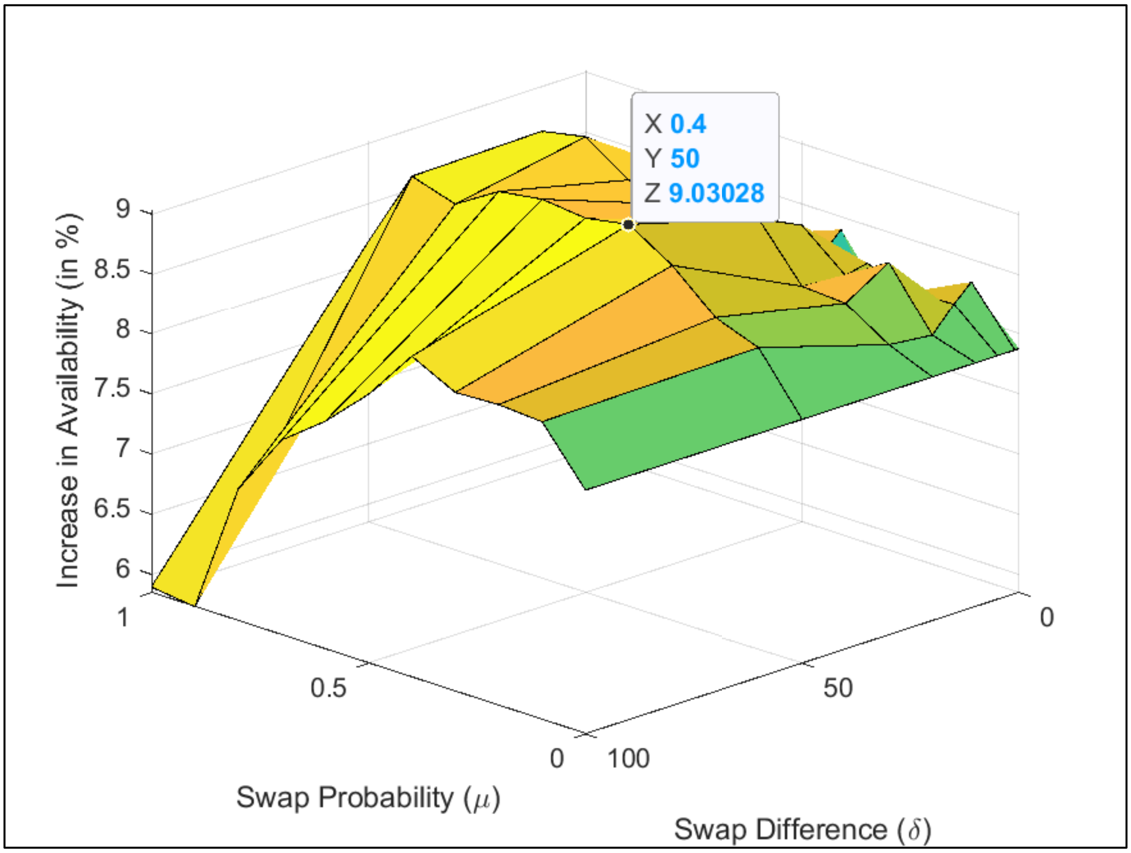
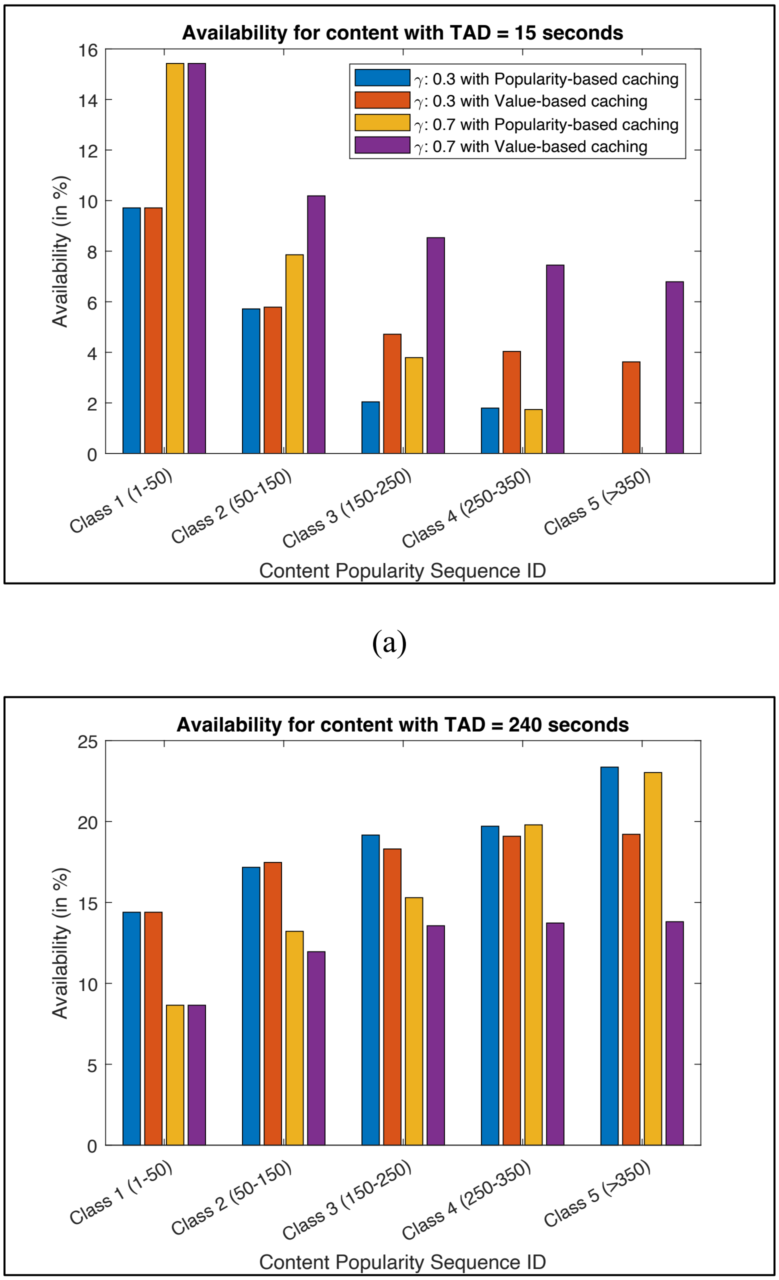


Figure 4.8. Increase in availability with respect to scenario without F-UAVs by varying μ and δ

Figure 4.8 shows the increase in availability while employing popularity-based caching for heterogeneous content sequence at communities. This is compared with a scenario without F-UAVs. The increase in availability is shown for $\alpha = 0.9$ with varying swap probability and swap difference.

The observations are as follows. First, the increase in availability for $\mu = 0$ corresponds to the cases where the popularity sequences are the same in all communities (i.e., the homogeneous case). For such cases, the increase in content availability due to contents ferried by F-UAVs is approximately 7.5%. Second, the maximum increase in availability of about 9% is recorded for

$\mu = 0.4$ and $\delta = 50$. This improvement is due to incorporating popularity-based caching at A-UAVs to tackle heterogeneity in popularity sequence. This demonstrate that benefits of popularity-based caching are higher for scenarios where content sequence is more heterogeneous.

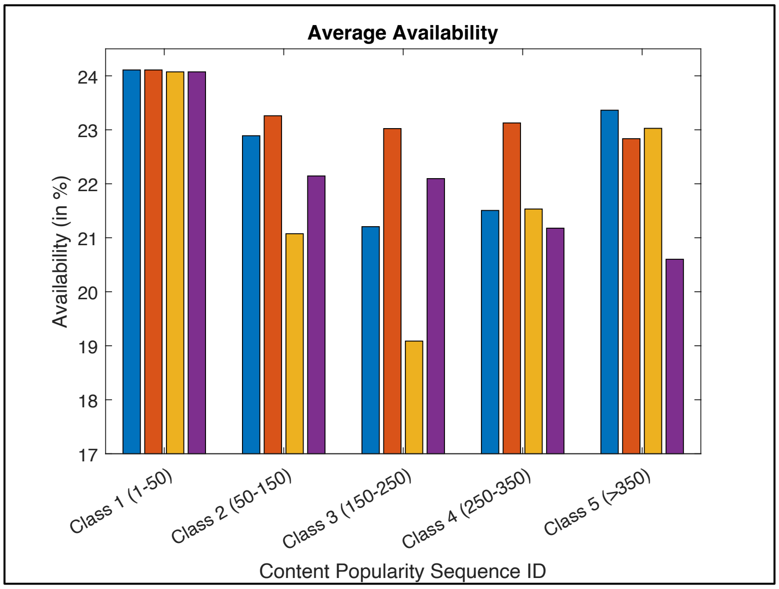


(a)

(b)

Figure 4.9. (a) Availability of Low *TAD* contents, (b) Availability of High TAD contents, (c) Average Availability

Figure 4.9 (cont'd)



(c)

4.8.3 Impact of Value of Contents with Different *TAD*

To observe the benefits of value-based caching, experiments are conducted with popularity parameter $\alpha = 0.4$, low $TAD = 15\ seconds$ and high TAD = 240 $seconds$. The analysis is done for five content popularity classes. The parameter γ is varied between 0.3 and 0.7 to evaluate the effect of low and high probability of low TAD contents within a given class. Rest of the parameters are according to Table 4.1. Performance comparison between value-based and popularity-based caching policy is shown in Figure 4.9 separately for low and high TAD content requests, along with the average availability.

The observations are as follows. First, availability of low TAD contents is more while employing value-based caching as opposed to popularity-based caching for all popularity classes and γ. The increase in popularity escalates for lower popularity classes such that Class 5 observes maximum increase in availability of low TAD contents by approximately 7% (Figure 4.9a). This is because value-based caching favors the storage of low TAD contents by increasing their computed value (see Section 4.4.3). However, if a content is highly popular, its value does not improve by a large margin. This can be seen in Figure 4.9a where availability of low TAD contents doesn't increase if they belong to Class 1. Second, availability of high TAD contents reduce for all popularity classes and across all values of γ, while employing value-based caching. This is due to the replacement of high TAD contents by low TAD (high value) contents at the A-UAVs. Figure 4.9b shows this adverse effect where the maximum reduction in availability of high TAD contents can be observed at Class 5. Third, average availability, while employing value-based caching policy, is best for middle range of content popularities. This can be seen in Figure 4.9c, where increase in average availability is maximum (i.e., approximately 4%) when low TAD content requests are from Class 3, beyond which it tapers off. The physical meaning of this phenomenon is that if a very low popular content has low TAD, it is not beneficial to cache it in A-UAVs. This is because a very low popularity content is less likely to be requested. Finally, for higher γ, the effect of value-based caching is comparatively severe since more contents from a class have low TAD. These effects manifest differently when the cache space of UAVs is varied.

Effects of scaling caching capacity of UAVs while employing value-based caching are discussed next.

4.8.4 Impacts of UAV cache Size on Value-Based Caching

To explore the extent of value-based caching toward increasing availability, cache space is varied. Parameters are set as follows; High $TAD = 240$ seconds, Low $TAD = 5$ seconds, $\gamma = 0.95$, and remaining parameters according to Table 4.1.

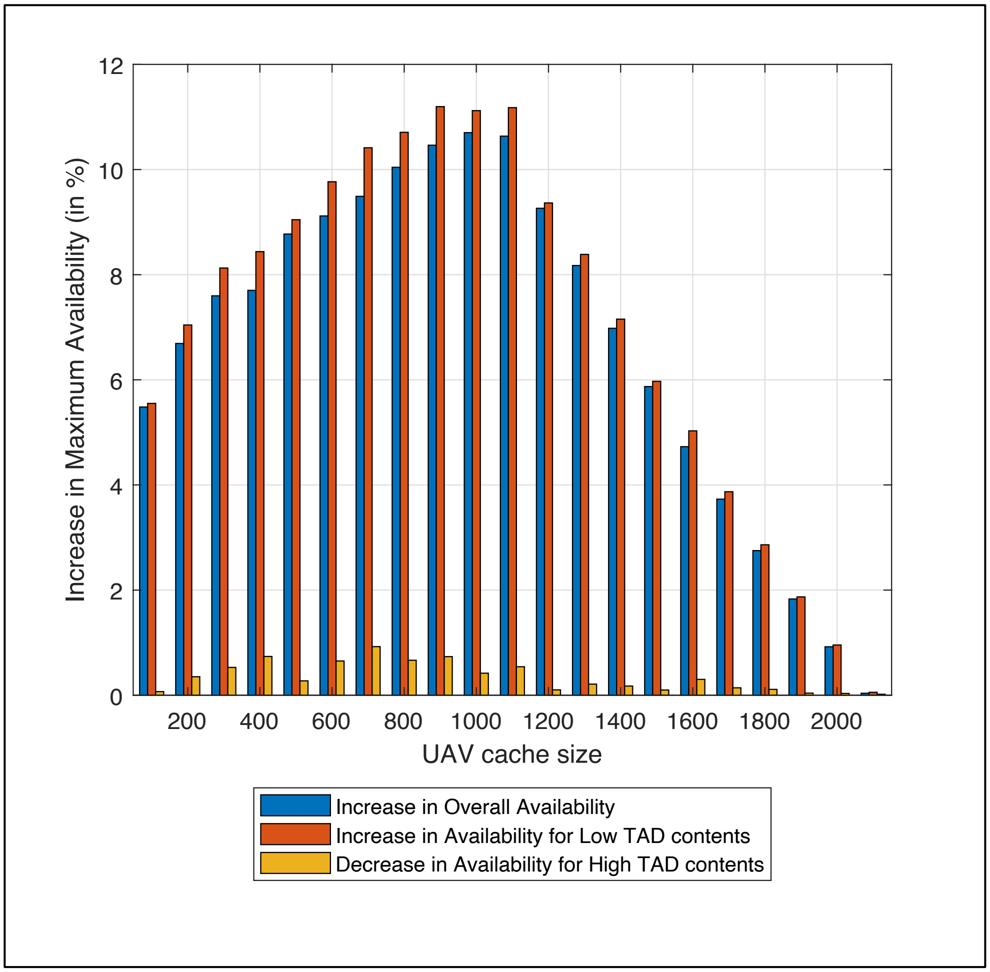


Figure 4.10. Increase in Availability with value-based caching with respect to popularity-based caching for increasing cache space

Figure 4.10 compares the impact of increasing cache size on content availability while employing value-based and popularity-based caching individually. The observations from the figures are as follows. First, for a given total number of contents viz, 2000, maximum increase in availability with value-based caching is recorded for UAV cache sizes in the range 900-1100. This increase in availability is approximately 12%. Beyond the cache size of 1100, availability of low TAD content reduces due to very low popular contents being cached at the A-UAVs. Second, the increase in overall availability is attributed to the increase in availability for low TAD contents.

Third, availability for high TAD contents reduce marginally. Due to increased value of low TAD contents, high TAD contents are replaced at A-UAVs and their availability reduces.

Beyond the extents of value-based caching, content availability, can be also achieved by exploiting F-UAV trajectories as discussed below.

4.9 Summary and Conclusion

This chapter designs a UAV-aided content dissemination system to enable content availability for users in the absence of communication infrastructure in disaster scenarios. Two types of UAVs are used, namely, anchor UAVs and ferrying UAVs. Anchor UAVs provide contents to users in their respective communities at all times while ferrying UAVs provide contents intermittently by sharing those cached in the anchor UAVs. Popularity based caching policy has been introduced which takes the heterogeneity in content popularity sequence into consideration to cache content is the anchor UAVs. Value-based caching policy has been explored where a content is cached in a A-UAV when it is likely to be requested and its associated tolerable access delay is low, which signifies urgency of requirement. The developed caching policies, deal with demand heterogeneity by associating value to a content based on its popularity and tolerable access delay. Together the popularity-based and value-based caching policy improve content availability by approximately 12%. The next chapter on this topic will include incorporating adaptive algorithms to learn the caching policy and ferrying UAV trajectories on-the-fly in time-varying disaster regions.

Chapter 5: Multi-Armed Bandit Learning for Content Provisioning in Network of UAVs

In disaster-hit regions, the obliteration of communication infrastructure leaves communities isolated, deprived of crucial information for survival and relief. This chapter introduces an innovative solution where a UAV-based content dissemination network is designed to operate autonomously of traditional communication systems. Addressing the inherent challenges such as limited UAV energy, storage and flight capabilities necessitates a sophisticated approach to content management, making UAVs a vital link in disseminating essential information.

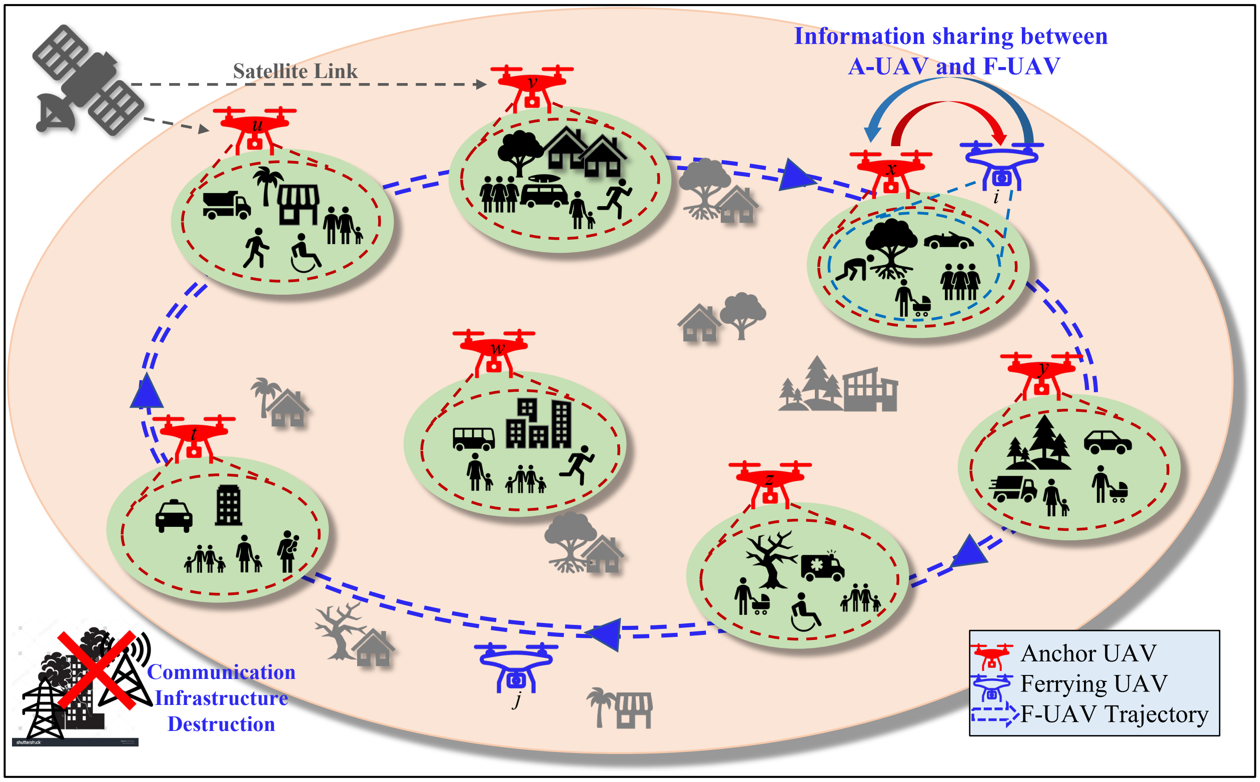


Figure 5.1. Coordinated UAV system for content caching and distribution in environments without communication infrastructure

5.1 Motivation

This chapter is shaped by the dire necessity for a resilient, UAV-assisted content distribution system capable of functioning in the absence of conventional communication networks. The challenge is amplified by the diverse and urgent information needs of isolated communities, coupled with UAV operational limitations. Traditional content distribution strategies often neglect the nuanced demand and spatial-temporal request heterogeneity. Hence, the research in this chapter designs an adaptive, decentralized caching strategy, employing a *Top-k* Multi-Armed Bandit Learning model, to ensure the prioritized delivery of critical content to affected populations via on-the-fly learning of caching policies.

5.2 Design Objectives

The primary goal of the research conducted in this chapter is to develop a UAV-aided content caching and dissemination framework that can dynamically adapt to the unique demands of disaster-stricken communities. By employing a *Top-k* Multi-Armed Bandit Learning model, the system aims to optimize content caching decisions in real-time that takes into account the geographical and temporal variations in content popularity as well as the heterogeneous demands of the users. The objectives are multi-fold:

a. This chapter designs a decentralized learning mechanism that enables UAVs to make informed caching decisions on-the-fly, therefore, maximizing the relevance and accessibility of content to stranded users.

b. The designed method incorporates a multi-dimensional reward structure within the learning model that accounts for both local and global content popularity trends that facilitates an optimal caching strategy that improves overall content dissemination.

c. This chapter will explore the interactions between the dynamically learned caching policies, Quality of Service (QoS) expectations (specifically, tolerable access delay), and user demand patterns, aiming to fine-tune the learning model for enhanced performance.

d. The designed framework has been rigorously tested and validate through simulation experiments and analytical modeling which ensures its effectiveness in a range of disaster scenarios, UAV configurations, and content popularity distributions.

This research endeavors to bridge the gap in current UAV-based communication solutions by introducing an agile, adaptive caching system that responds to the immediate needs of disaster-affected populations. This can potentially transform the landscape of emergency communication and information dissemination in the face of infrastructure collapse.

5.3 Caching based on Content Pre-loading at Anchor UAVs

This section discusses caching policies based on content pre-loading at A-UAVs that assumes pre-assigned, static, and globally known content popularities. After understanding the limitations of these caching policies, this chapter proposes a runtime, dynamic and adaptive *Top-k* Multi-armed Bandit based caching mechanism, which is explained in a later section.

5.3.1 Pre-loading Policies at Anchor UAVs (A-UAVs)

The *Fully Duplicated* (*FD*) mechanism [91] is a naive approach that allows A-UAVs to download content from vertical links upon request by local users. However, the FD mechanism has limitations such as content duplication, high vertical link download costs, and suboptimal utilization of UAV cache space. *Smart Exclusive Caching* (*SEC*) [91]

overcomes the limitations of the FD mechanism by storing a set number of unique contents in all A-UAVs and sharing them among communities via F-UAVs. Assuming globally known *homogeneous content popularity* across all user communities, the SEC mechanism divides the cache into two segments. Segment-1 contains the top $\lambda . C_A$ popular contents cached in all A-UAVs, while Segment-2 contains unique contents $(1-\lambda). C_A$, where λ is the *Storage Segmentation Factor*. Total contents in the system as per SEC is given as:

$$C_{sys} = \lambda . C_A + N_A . (1-\lambda). C_A \tag{5.1}$$

Popularity-Based Caching (*PBC*) [93] is employed when different communities have different content preferences. PBC divides the cache space of a A-UAV into two segments, considering the *heterogeneous popularity* sequence of the local community. Segment-1 caches the most popular contents, which can be exclusive to a A-UAV (C_E) or non-exclusive i.e., may be cached across multiple A-UAVs (C_{NE}), while Segment-2 is the same as SEC. To be noted that total unique contents in the system can be denoted as C_E^{total}, which leads to total replicated contents across the system to be represented as follows:

$$C_{replicated} = C_{NE} + N_A . (1-\lambda). C_A \tag{5.2}$$

Therefore, by modifying Eqn. 5.1, total number of contents in the system can be expressed as:

$$C_{sys} = C_{NE} + C_E^{total} + N_A . (1-\lambda). C_A \geq \lambda . C_A + N_A . (1-\lambda). C_A \tag{5.3}$$

Value-Based Caching (*VBC*) [93] further enhances the caching policy by storing top-valued contents in Segment-1 of A-UAV, where *value* of contents comprises of their popularity and tolerable access delay. Value of a content 'i' be calculated as:

$$V(i) = \kappa \upsilon \times \frac{p_\alpha(i)}{TAD(i)} = \kappa \times \frac{TAD_{min}}{p_\alpha(1)} \times \frac{p_\alpha(i)}{TAD(i)} \tag{5.4}$$

In this equation, $p_\alpha(i)$ represents the content's popularity as per the Zipf distribution, $TAD(i)$ is the content's tolerable access delay, κ is a scalar weight that increases as popularity decreases, and υ is a normalization constant. The normalization constant is calculated for a given Zipf (popularity) parameter α using the minimum possible TAD (TAD_{min}) and the maximum possible popularity, which is $p_\alpha(1)$. The value of $V(i)$ is bounded between [0,1] and increases as $p_\alpha(i)$ increases and

$TAD(i)$ decreases and can present a holistic quantifiable measure for caching decision.

The caching policy for F-UAVs remains the same for all the discussed and forthcoming caching policies for A-UAVs [91], [93], [126]. An F-UAV ferries content from already visited A-UAVs to future visiting A-UAVs in its trajectory. The caching policy of an A-UAV determines the utility of an F-UAV where every A-UAV should maintain sufficient contents in its cache space to optimize the F-UAV cache utilization.

5.3.2 Limitations of Cache Pre-loading at A-UAVs

The caching policies discussed in this section rely on pre-loading content into A-UAVs, which has certain limitations. This approach assumes a priori knowledge of the popularity distribution of all the content in the system, which can hinder practical feasibility during deployment. Local popularity estimation of requested content within individual A-UAVs can partially alleviate this issue, but it cannot adjust the crucial storage segmentation factor (λ) (see section 5.3.1) for maximizing availability across the entire system of A-UAVs and their communities. Collaborative global popularity estimation can be introduced, but it fails to capture demand heterogeneity across different A-UAV communities.

The limitations listed above can be addressed by employing a *Top-k* Multi-armed Bandit (*Top-k* MAB) learning-based caching mechanism at the A-UAVs, which is explained in the following section. This paradigm is able to leverage the expected reward maximization attribute of MAB and intelligence sharing nature of proposed multi-dimensional reward structure for caching decision at the A-UAVs.

5.4 Decentralized Caching with Multi-Armed Bandit

Once a A-UAV is deployed into a community, its subsequent action is to decide which contents to download (via its vertical link) and cache such that content availability to the requesting users can be maximized. This goal is achieved by employing a *Top-k Multi-Armed Bandit* learning agent in the A-UAV.

5.4.1 *Top-k* Multi-Armed Bandit Learning

Multi-Armed Bandit is a classic problem in reinforcement learning [127] and decision-making. At each round t, an agent chooses an arm A_t out of N arms, denoted by $A_1, A_2, \ldots, A_N$, and observes a reward R_t. Each arm i has an unknown reward distribution with mean μ_i and variance σ_i^2. The agent's goal is to maximize the total expected reward R_T over T rounds, where T is the total number of rounds (time horizon):

$$R_T = max \sum_{t=1}^{T} E[R_t] \tag{5.5}$$

This chapter uses a variant of MAB called *Top-k Multi-Armed Bandit* [127], [128]. Here, the agent has to choose k arms out of a larger set of N arms, as opposed to choosing one arm in classical MAB, and receives a reward for each arm in the chosen set. The goal of the agent is to maximize the total cumulative reward R_T obtained over a finite time horizon T:

$$R_T = max \sum_{t=1}^{T} \sum_{i=1}^{k} E[R_{i,t}] \tag{5.6}$$

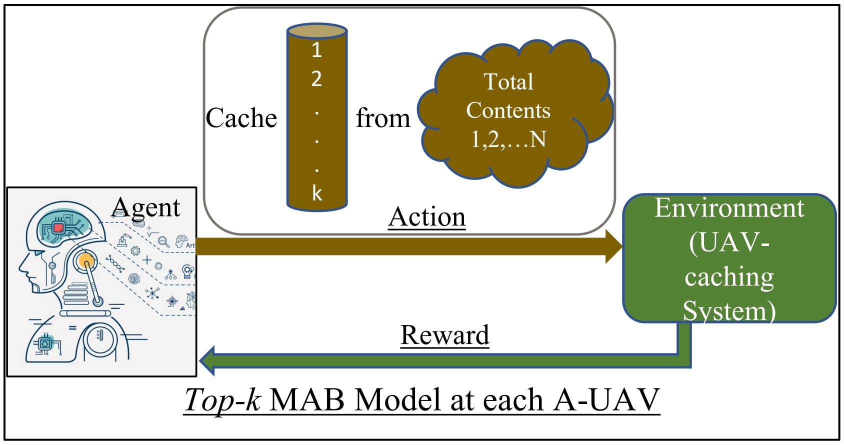

Cache
1
2
.
.
.
k
from
Total Contents 1,2,…N
Agent
Action
Environment (UAV-caching System)
Reward
Top-k MAB Model at each A-UAV

Figure 5.2. *Top-k* Multi-Armed Bandit Learning for Caching Policy at A-UAVs

5.4.2 Decentralized Caching using *Top-k* Multi-Armed Bandit

In the scenario of UAV-caching, there is a *Top-k* MAB agent in each A-UAV. Here, choosing each content for caching corresponds to choosing an arm. The 'k' of *Top-k* MAB agent corresponds to the caching capacity of A-UAV, i.e., $k = C_A$. The agent's aim is to select 'C_A' contents out of a larger set of 'N' contents to be cached in an A-UAV such that content availability to the users can be maximized. Here, the UAV-aided content dissemination system is the learning environment where the A-UAVs interact through their actions of choosing specific sets of contents to be cached. The feedback from the environment for the taken actions are in the form of rewards/penalties. Actions are rewarded when cached contents are requested by the users and are served to the users within the given tolerable access delay or penalized otherwise. The top C_A contents that accumulate most reward from the corresponding community and other communities are chosen to be cached at a A-UAV. It should be noted that the *Top-k* MAB agents in the A-UAVs are provided with no *a priori* information about the content popularity at the corresponding user communities.

A learning decision epoch for each *Top-k* MAB agent is set according to the F-UAVs accessibility at the corresponding community (i.e., an F-UAV's visiting frequency). This is because the F-UAVs carries the content availability information from the communities in its trajectory that is leveraged for learning at the A-UAVs' *Top-k* MAB agents using appropriately designed multi-dimensional rewards. The agent learns to cache contents via the multi-dimensional reward structure which has three parts: namely *local*, *ferrying,* and *global* reward. The first corresponds to the increase in availability at an A-UAV's corresponding community i.e., *increase in local availability* (δ_l). The second is related to the contents that are cached in an A-UAV, and are responsible for increase in availability at other communities i.e., *ferried content availability*

(δ_f). A global reward is received when cached contents add to increase in average availability across all communities. This is called *increase in global availability* (δ_g). The three types of rewards are given below:

$$R_i^L = \begin{cases} 1, & for\ \delta_l > 0 \\ -1, & for\ \delta_l < 0 \end{cases} \tag{5.7}$$

$$R_i^F = \begin{cases} 1, & for\ \delta_f > 0 \\ -1, & for\ \delta_f < 0 \end{cases} \tag{5.8}$$

$$R_i^G = \begin{cases} 1, & for\ \delta_g > 0 \\ -1, & for\ \delta_g < 0 \end{cases} \tag{5.9}$$

In the above equations, R_i^L, R_i^F, and R_i^G are rewards according to increase in availability for content 'i' cached in an A-UAV.

Learning is achieved using a tabular method where a Q-table is maintained for all contents in the system. The value corresponding to each content is called a Q-value or action-value [127]. The agent updates the Q-value for a content at every learning epoch according to the multi-dimensional rewards in Eqns. 5.7-5.9 from the interaction with the environment (UAV-aided content dissemination system) and learns the best actions (contents cached). The recursive expression which explains Q-value update for a content 'i' is given as follows:

$$Q(i) \leftarrow Q(i) + \alpha\big(r(i) - Q(i)\big) \tag{5.10}$$

Here, $Q(i)$ represents the Q-value of a content 'i'; $r(i)$ is the reward received by caching content 'i'; α is a hyper-parameter which controls the learning rate. The Q-values for all contents are initialized with zero to ensure no *a priori* information for a *Top-k* MAB agent. Also, it ensures equal importance to all contents for caching decisions. An epsilon-greedy (ϵ-greedy) exploration strategy is implemented. Such exploration strategy guarantees that every content gets to be cached in an A-UAV. As learning progresses, exploration decays and best contents with highest Q-values are exploited with the aim of maximizing accumulated reward which improves the caching policy and thus increases content availability.

The proposed algorithm enables *Top-k* MAB agents in A-UAVs to learn the caching policy, and the contents cached at A-UAVs emulate the cache pre-loading segmentation behavior described in Section 5.3.1. However, the caching policy and corresponding content availability may fluctuate due to less request for less popular content, leading to weak or unstable reward estimates. This results in Q-values that are highly sensitive to requests for less popular content and less sensitive to requests for popular content. Therefore, changes in Q-values of less popular content may lead to intermittent variations in caching, particularly in Segment-2 (refer Section 5.3.1). Also, there can be $\binom{N}{k}$ combination of contents to be sampled by the *Top-k* MAB agent for caching. Due to this the reward estimation for each content occurs after large intervals, which leads to a weak estimate of reward distribution as N increases. These oscillations can be controlled by empirically selecting ϵ and its decay rate. To reduce the dependence of caching policy on the choice of ϵ, Upper Confidence Bound (UCB) strategy is used [127], [128]. The *Top-k* MAB agent maintains an upper confidence bound on the expected reward of each content, and selects the set of C_A contents with highest UCB at each epoch.

$$U_t(i) = Q_t(i) + \sqrt{\frac{\alpha_u \log(t)}{N_t(i)}} \tag{5.11}$$

Here, $U_t(i)$ is the UCB of content 'i' at epoch 't'; $Q_t(i)$ is the updated Q-value at epoch 't'; α_u is a hyperparameter that controls the degree of exploration; $N_t(i)$ is the number of time content 'i' has been requested till epoch 't'. The first term represents the reward estimate, and the second term depicts the uncertainty in reward estimate. UCB selects the content that has high potential for high reward but hasn't been requested frequently. The promotes exploration without externally inducing an exploration parameter such as ϵ. For this chapter, ϵ-greedy exploration strategy is applied according to the UCB values, as shown in Step 7-16 in Algorithm 5.1.

The following pseudo code explains the caching policy at a A-UAV with a *Top-k* MAB agent.

Algorithm 5.1. Caching policy at a A-UAV with *Top-k* MAB Learning

```
1.  Initialization:
        a. N: Total contents in the system
        b. $C_A$: Caching capacity of an A-UAV
        c. $Q$: Array of size $C_A$ initialized with 0's (Q-table).
        d. $\epsilon$: Exploration rate
        e. $\alpha$: Learning rate for Q-table update.
        f. $\alpha_u$: Degree of exploration (if UCB used)
2.  Load A-UAV's cache with $C_A$ randomly chosen contents.
3.  while True:
      \\ Check for learning epoch
4.    if F-UAV is visiting A-UAV then do
5.      for $i = 0$ $to$ $length$(A-UAV cache size $C_A$) do
6.        Get reward $r(i)$  \\ according to Eqns. 5.7-5.9
7.        Update $Q(i)$      \\ $Q(i) \leftarrow Q(i) + \alpha[r(i) - Q(i)]$
                             \\ $Q(i) \leftarrow U(i)$ if UCB employed
8.      end for
9.      $value = \boldsymbol{copy}(Q)$ \\ make a copy of Q-table
        \\ Reload contents (Select arms)
10.     for $i = 0$ $to$ $length$(A-UAV cache size $C_A$) do
```

Algorithm 5.1. (cont'd)

```
            **Generate** random number '$x$'
            **if** $x < \epsilon$ **then do**
                **Load** 1 randomly chosen content to A-UAV
            **else**
                $c_{max} = \boldsymbol{argmax}(value)$
                **Load** $c_{max}$ to A-UAV
                **Set** $value[c_{max}] = -inf$
            **end if**
        **end for**
    **end if**
    **Check** for $\epsilon$ decay condition.
    **if** true **then do**
        **Update** $\epsilon$
    **end if**
**end while**
```

This *Top-k* MAB agent at a A-UAV learns a near optimal caching policy within a finite time horizon and approaches the best caching policy asymptotically. The cached contents can boost content availability at their respective communities as well as at other distant communities via F-UAVs.

Table 5.1 Default Values for Model Parameters

#	Variables	Default Value
1	Total number of contents, C	1000
2	Number of A-UAVs, N_A	12
3	Number of F-UAVs, N_F	3
4	Cache space in A-UAV, C_A	100
5	Cache space in F-UAV, C_F	100
6	Poisson request rate parameter, μ	1 request/sec
7	Hover time of F-UAV, T_{Hover}	600 seconds
8	Transition time of F-UAV,	300 seconds
9	Zipf parameter (Popularity), α	0.7
10	Ferrying UAV Trajectory	Round-robin

5.5 Experiments and Results

Experiments are performed to analyze the performance of the proposed *Top-k* MAB learning-based caching mechanism with a discrete event simulator. The simulator accomplishes content request generation while maintaining an intra-event interval according to exponential distribution and following a Zipf popularity distribution [126]. To perform the cache pre-loading, the mathematical expressions are included in the simulator. To capture heterogeneity in content popularity sequence at different communities, contents are swapped with pre-decided probability [93] and the difference between the sequences are determined using Smith-Waterman Distance [125]. The experimental parameters for the proposed *Top-k* MAB learning based caching and cache pre-loading policies are listed in Table 5.1. The performance evaluation of the proposed mechanism is accomplished via the following metrics.

5.5.1 Performance Metrics

Content Availability (P_{avail}): It is defined as the ratio between cache hits and generated requests within a time interval. Cache hits are the content provided to the users from the contents cached in the UAV-aided caching system (without download). Therefore, content availability indirectly indicates the content download cost of a systems as well.

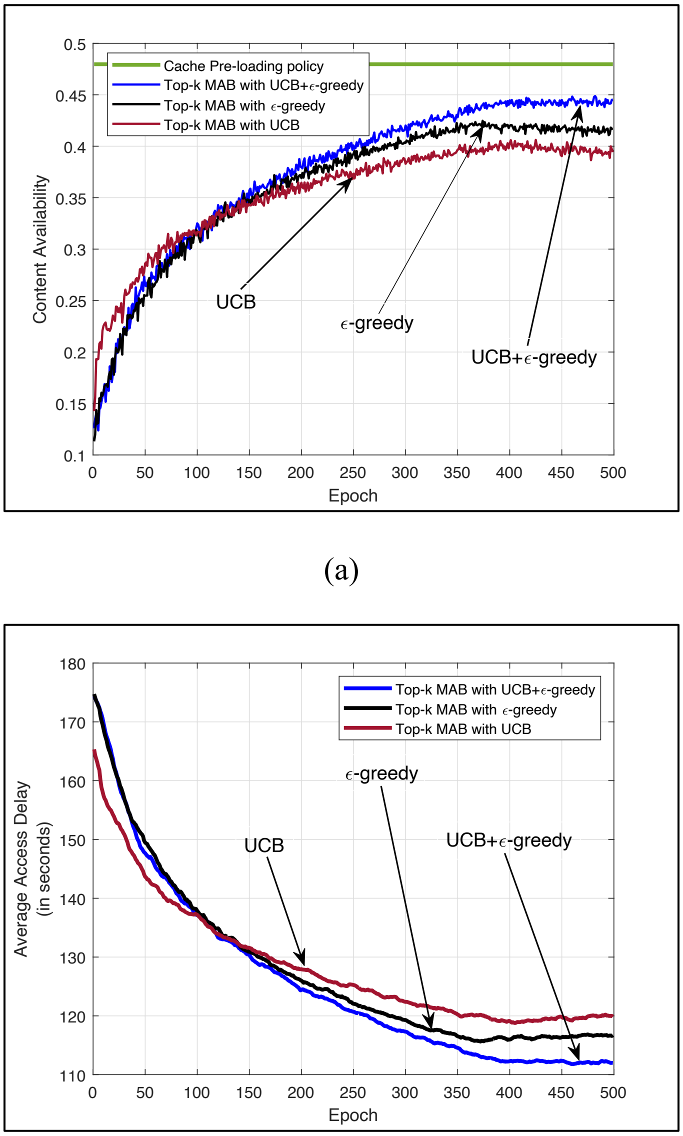
Jaro-Winkler Similarity (JWS): It is a similarity measure that is used to compute the similarity between two sequences [129]. It is computed by calculating the number of matches, number of transpositions requires within the matches and the similarity in prefix of both sequences. JWS is used to compute the similarity between the content sequence from the learnt caching policy and content sequence according to cache pre-loading.

Access Delay (AD): Performance of *Top-k* MAB model is also evaluated based on the access delay which is the end-to-end delay between the generation of content request and its provisioning form the cached contents in the UAVs. This chapter reports the epoch-wise average access delay to show the improvement in caching policy as learning progresses.

5.5.2 Effect of Exploration Strategies on Learnt Caching Policy

In order to understand the viability of the proposed *Top-k* MAB learning-based caching policy in scenarios with demand heterogeneity, two type of content popularity sequence are used. Every consecutive community has a different popularity sequence. For ϵ-greedy strategy, initial exploration is $\epsilon = 1$ with decay rate of 0.0025 per epoch. The degree of exploration in UCB is set to $\alpha_u = 2$. Figure 5.3a shows the convergence behavior of the learnt caching policy with a comparison of exploration strategies employed in the *Top-k* MAB model.

The convergence behavior is shown in term of content availability from the learnt caching policy. The observations from Figure 5.3(a) are as follows. First, the figure shows that by employing *Top-k* MAB agent at every A-UAV, a near optimal caching policy can be learnt. The algorithm is able to leverages the multi-dimensional reward structure, as explained in Eqns. 5.7-5.9, to achieve content availability close to the benchmark performance (see section 5.3.1). Second, when the agent uses UCB exploration strategy, the content availability settles at a sub-optimal value. However, during the initial learning epochs the content availability increases promptly due to high upper confidence value of all contents, which avoids exploitation. This is due to low sampling of requests. As learning progresses, the sparse request for unpopular contents keeps the upper confidence value high which maintains consistent exploratory behavior.



(a)

(b)

Figure 5.3. Comparison between exploration strategies in *Top-k* MAB and pre-loading using (a) Content Availability; (b) Access Delay

An algorithmically induced ϵ value in ϵ-greedy strategy avoids this continuous uncertainty behavior due to ϵ decay. This can be seen from the content availability with ϵ-greedy exploration strategy which is better than the performance with UCB. Finally, to maintain the initial surge in content availability and to limit the unbounded exploratory behavior, ϵ-greedy exploration is applied on the UCB values of the content. It can be seen that such hybrid exploration strategy helps to boost the content availability closer to the benchmark performance by 5%. Similarly, Figure 5.3(b) shows the convergence behavior of the *Top-k* MAB learning-based caching agent in terms of access delay. This is computed for a TAD of 300 seconds and it is observed that as learning progresses, the access delay for requested contents reduce while the content availability increases simultaneously. This manifests the improvement in learnt caching policy over the learning epochs. The best reduction in access delay is observed when ϵ-greedy exploration is applied on the UCB values of the content.

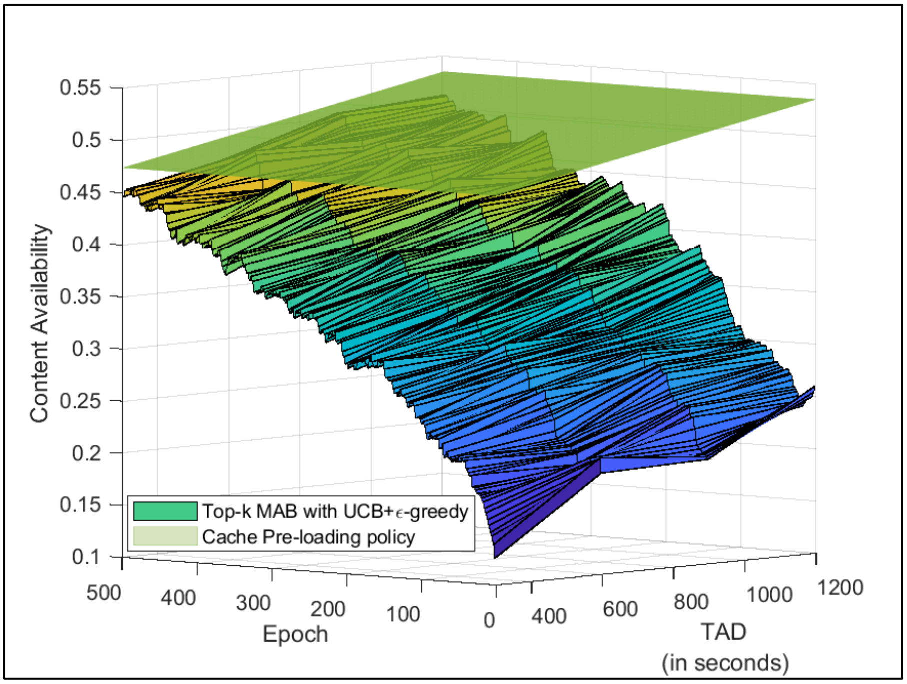
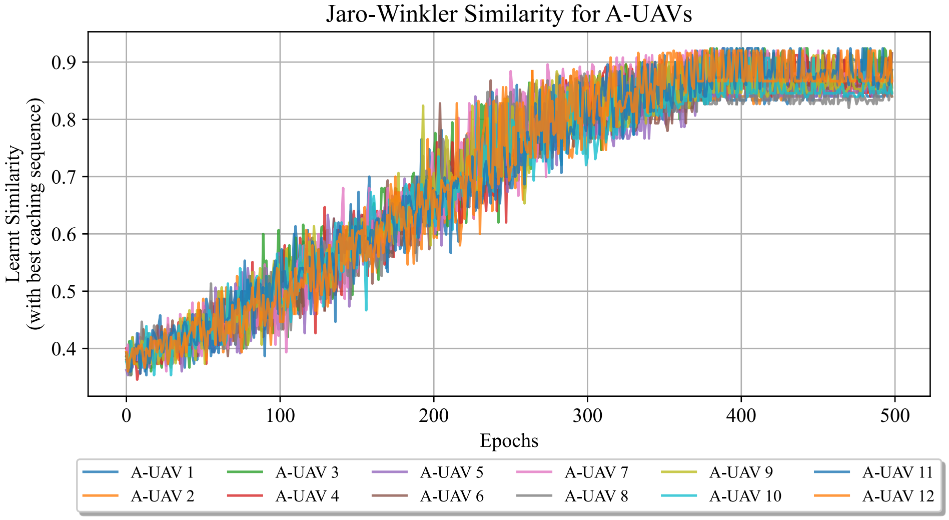


Figure 5.4. Change in learnt caching policy of A-UAV with *TAD*

5.5.3 Impact of Tolerable Access Delay on Learning Performance

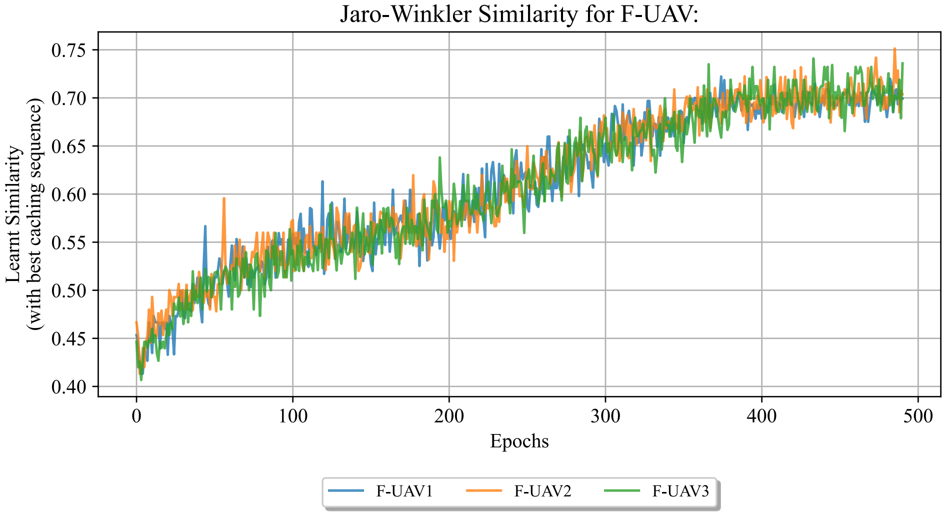
To show the learning capability of the proposed *Top-k* MAB model, experiments are conducted with varying TADs ranging from 300 to 1200 seconds. The content availability according to the learnt caching policy with varying TAD is shown in Figure 5.4. The figure demonstrates the behavior of the proposed caching mechanism with respect to the benchmark performance, computed from the cache pre-loading policy discussed in Section 5.3.1. Following observations can be made from Figure 5.4. First, the learnt caching policy achieves performance closer to the benchmark for all values of TAD. Second, the best possible performance (i.e., the benchmark) changes with change in TAD. The *Top-k* MAB agents in the A-UAVs adapts to the user defined TAD. It can be observed in Figure 5.4 that the learning performance varies along with TAD. In other words, the role of multi-dimensional reward structure of the MAB agent becomes more evident with higher TAD. Especially, the information related to the global availability i.e., δ_f and δ_g (refer Section 5.4.2), are derived from large count of content requests. This improves the estimated reward at A-UAVs thus impacting their caching decision.



(a)

Figure 5.5. Jaro-Winkler similarity for (a) A-UAVs and (b) F-UAVs

Figure 5.5 (cont'd)



(b)

5.5.4 Cache Similarity of Learnt Sequence with Best Sequence

The effect of learning on the cached content sequence is demonstrated in Figure 5.5. Figure 5.5(a) plots Jaro-Winkler Similarity (JWS) of cached content sequences for all 12 A-UAVs. The key observation are as follows. First, the JWS between the best caching sequence from cache pre-loading policy (see Section 5.3.1) and the cached content sequences learnt by the *Top-k* MAB agents at A-UAVs converge near 0.9, with a certain variance. Physically, this represents higher degree of similarity post convergence, where 1 indicates complete similarity and 0 implies no similarity. Second, the cached contents improve over epochs as learning progresses. Lower JWS values at initial epochs signifies that A-UAVs have no *a priori* content popularity information, local or global. As the MAB agents learn, over epochs of generated content requests, the cached contents in A-UAVs become more similar to the best caching sequence. Third, JWS is an indirect representation of the storage segmentation factor (λ), which is used to decide the segment sizes according to cache pre-loading policies. A higher JWS implies that, along with learning the caching policy, the *Top-k* MAB agents learn to emulate the said segmentation behavior. Finally, the partial dissimilarity of the cached content sequence can be ascribed to the uncertainty associated with the Q-values of contents with low popularity. Also, this leads to an oscillatory convergence of JWS for A-UAVs. This behavior manifests in the JWS for F-UAVs as well, which is shown in Figure 5.5(b). Since, F-UAVs ferry contents that are requested less frequently, the low popularity of such contents leads to a comparatively sluggish improvement of its JWS as compared to JWS improvement of A-UAVs.

5.6 Summary and Conclusion

In this chapter, UAV-aided content dissemination system is designed which can learn the caching policy on-the-fly without *a priori* content popularity information. Two types of UAVs are introduced to revive content provisioning in a disaster/war-stricken scenario viz. anchor and ferrying UAVs. Cache-enabled anchor UAVs are stationed at each stranded community of users for uninterrupted content provisioning. Ferrying UAVs act as content transfer agents across anchor UAVs. The evolution of pre-loading-based caching policies are discussed which requires *a priori* information about content popularity. A decentralized *Top-k* Multi-Armed Bandit Learning-based caching policy is proposed to ameliorate the limitation of cache pre-loading. It learns the caching policy on-the-fly with the help of a multi-dimensional reward structure with encapsulates local and global availability information. The forthcoming chapters on this research will include the characterization of shared intelligence across UAVs, and UAV trajectories and deployment strategies which can build the foundation for developing distributed learning-model sharing approaches to improve content provisioning.

Chapter 6: Distributed Federated-Multi-Armed Bandit Learning for Content Management in Connected UAVs

In the aftermath of disasters such as earthquakes, floods, or armed conflicts, survivors are often forced to relocate into regions with severely compromised or entirely destroyed communication infrastructure. In these situations, access to critical information, ranging from emergency services and rescue updates to weather conditions and medical logistics, can determine the success of relief efforts. The UAV-aided caching system introduced in this chapter builds directly on the learning mechanisms developed in Chapter 5 by extending them into a federated and distributed framework that is better suited to such fragmented and high-variability environments.

As described earlier, this chapter presents a two-tiered content dissemination architecture. Communities of stranded users are each served by Anchor UAVs (A-UAVs), which maintain vertical connectivity to centralized content repositories. A set of Ferrying UAVs (F-UAVs), which operate without vertical links, travel between A-UAVs and propagate cached content throughout the network. The key advancement lies in the introduction of Federated Multi-Armed Bandit (FedMAB) Learning, a decentralized learning mechanism that enables UAVs to optimize their caching policies on-the-fly based on local user demands while periodically aggregating their learning to ensure system-wide coordination.

6.1 Motivation

The caching policies developed in Chapter 5 addressed on-the-fly learning within individual UAVs, but they remain limited in scope when faced with distributed user communities exhibiting strong geo-temporal variations in content demand. In such disconnected environments, local demand at one community may be vastly different from another, often driven by the immediate vicinity to the disaster, accessibility to relief, or evolving user needs. Moreover, content requests are not uniform in urgency. Some may require immediate access (e.g., evacuation routes), while others can tolerate delay (e.g., food distribution updates).

This chapter is motivated by the need to enhance learning agility, model generalization, and coordination across UAVs deployed in such multi-community environments. While a single UAV may adapt to its local context, the opportunity to share learning models across UAVs unlocks faster convergence, improved robustness, and reduced reliance on repeated exploration. Federated Multi-Armed Bandit (FedMAB) Learning makes this possible by allowing each UAV to independently learn caching decisions while periodically aggregating Q-values or model updates. This ensures each UAV not only reflects its local reality but benefits from insights gathered elsewhere.

Unlike prior works that assume globally known popularity or rely on slow-to-adapt function approximators, this chapter introduces a collaborative and distributed learning framework that prioritizes responsiveness and scalability. In doing so, it aligns learning-based caching strategies with the practical realities of post-disaster operations, limited backhaul, varying QoS needs, and non-uniform content value across communities.

6.2 Design Objective

The primary objective of this chapter is to present a UAV-aided content caching and dissemination framework that can learn optimal caching policies on-the-fly using Federated Multi-Armed Bandit (FedMAB) Learning. The system is designed to operate effectively in infrastructure-deficient disaster scenarios and adapt to geo-temporal variations in content demand.

a. This chapter proposes a UAV-based caching framework that allows each UAV to autonomously learn its content caching policy in real time by analyzing locally observed content request patterns.

b. It introduces Multi-Armed Bandit learning algorithms that jointly consider local observations and shared insights from other UAVs to improve caching decisions that reflect both local and global content popularity.

c. It presents a federated model aggregation technique that enables UAVs to periodically exchange their learned Q-tables, thereby enhancing the overall caching efficiency without exchanging raw data.

d. It investigates the relationship between the learned caching strategies and Quality of Service (QoS) expectations by incorporating Tolerable Access Delay (TAD) as a key constraint in content relevance and urgency.

e. It explores the trade-offs between content demand variability and the responsiveness of learned caching policies that offers insights into parameter tuning for optimal policy convergence.

f. It validates the effectiveness of the proposed caching model through simulation-based experiments and analytical evaluations across diverse disaster configurations and content demand patterns.

Through these objectives, the chapter seeks to demonstrate a scalable, adaptive, and decentralized caching strategy that aligns with the operational realities of disconnected, post-disaster environments. It further aims to deliver a robust, distributed solution that enhances the agility, resilience, and efficiency of UAV-assisted content dissemination under extreme conditions where infrastructure-based communication is no longer viable.

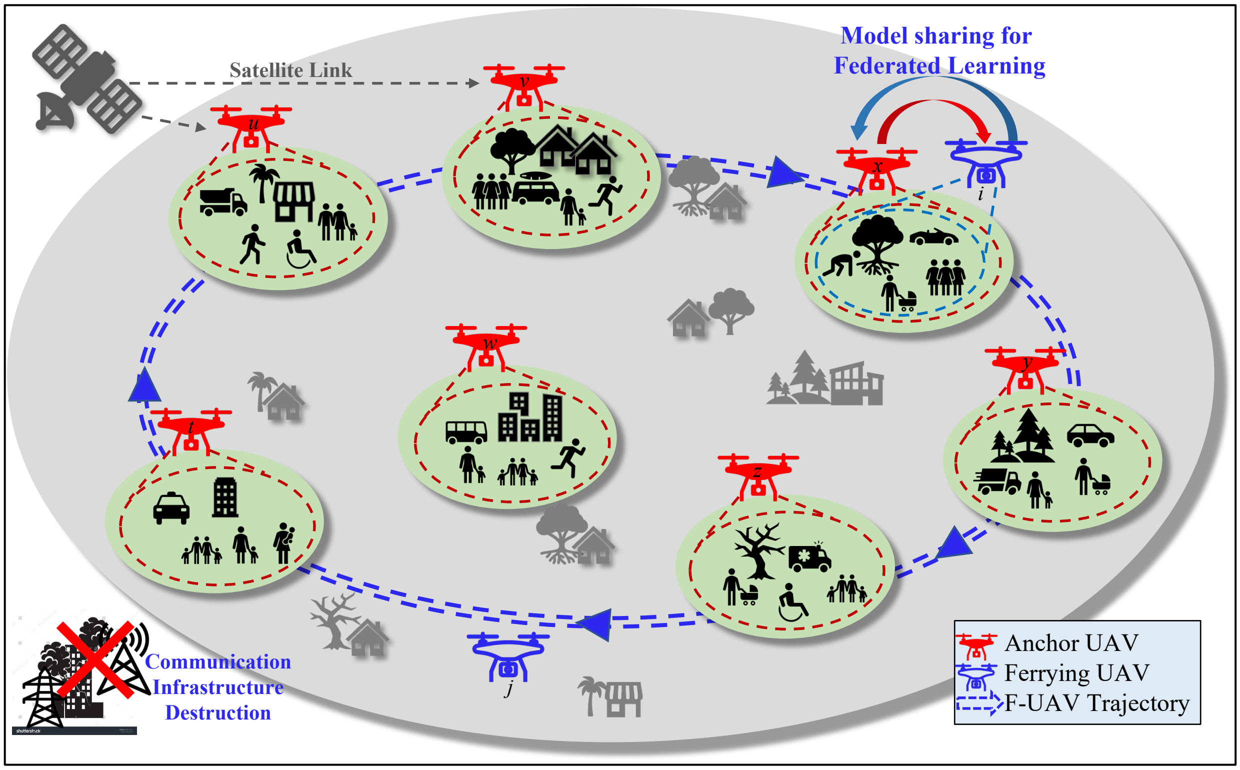


Figure 6.1. Coordinated UAV system for content caching and distribution in environments without communication infrastructure

6.3 System Model

6.3.1 UAV Hierarchy

As shown in Figure 6.1, a two-tiered UAV-assisted content dissemination system is deployed. Each community is served by a dedicated A-UAV that uses a lateral wireless connection (i.e., WiFi etc.) to communicate with users in that community. To be noted that that the role of A-UAVs can be served by ground vehicles with similar mobility restrictions and communication equipment for both vertical and lateral links. The system in Figure 6.1 introduces a set of ferrying UAVs (F-UAVs), which are mobile and only have lateral communication links such as Wi-Fi. The lateral links are used for transferring content between the A-UAVs and the users in the community that the F-UAV is currently visiting at. Unlike the A-UAVs, the F-UAVs do not possess vertical links. The F-UAVs act as content transfer agents across different user communities by selectively transferring content across the A-UAVs. The ferrying UAVs also provide a means for isolated communities to access content, making the system more resilient and accessible for all users.

When a user in a community requests a content, the serving local A-UAV first checks its local storage. If the content is not found, the A-UAV waits for a potential delivery by a passing F-UAV. This allows the content to be cached and transferred around the A-UAVs, thus enabling users in different communities to access content that was downloaded by other A-UAVs. If no F-UAV arrives within the specified tolerable access delay (TAD), only then the A-UAV downloads the content via its expensive vertical link. This way, the proposed two-tiered UAV-assisted content dissemination system is able to mitigate the limitations of the FD approach.

6.3.2 Content Demand and Provisioning Model

The generated content requests from the users in a community follow different popularity distributions and quality of services as outlined below.

Content Popularity: Research has shown that the pattern of content requests from a population often follows a Zipf distribution [91], [119], [126], where the popularity of a content is proportional to the inverse of its rank and is a geometric multiple of the next popular content. Popularity of content 'i' is given as:

$$p_\alpha(i) = \frac{\left(\frac{1}{i}\right)^\alpha}{\sum_{k\in C}\left(\frac{1}{k}\right)^\alpha} \tag{6.1}$$

The Zipf parameter, α, determines the distribution's skewness, while the total number of contents in the pool is represented by the parameter C. It should be noted that while the request for a specific content from a user follows Zipf distribution, the inter-request time from a user follows the popular exponential distribution.

Tolerable Access Delay: For each requested content, the user specifies a Tolerable Access Delay (TAD) [123], [124], which serves as a quality-of-service parameter and represents the amount of time the user is willing to wait before the content is provisioned for download.

Content Provisioning: Upon receiving a request from one of its community users, the relevant A-UAV first searches its local storage for the content. If the content is not found, the A-UAV waits for a potential future delivery by a traveling F-UAV. If no F-UAV arrives with the requested content within the specified TAD, the A-UAV then proceeds to download it through its vertical link, which is usually expensive. In other words, the system attempts to provision the requested content without incurring the cost of downloading from the centralized server by waiting for the user-specified in order to access it from potentially passing F-UAVs.

6.4 Limitations of Cache Pre-loading at A-UAVs

All the caching policies described in this section relies on content pre-loading in the A-UAVs. Such preloading leads to the following limitations. Such preloading majorly assumes prior knowledge of the underlying popularity distributions of the entire content population in the system. This assumption can seriously impede practical feasibility from a deployment standpoint. The impacts of the assumption can be partially mitigated by estimating local popularity of the contents requested within individual A-UAV's communities. Such estimates, however, would fail to adjust the storage segmentation factor (λ), which is crucial for maximizing availability across the entire system of A-UAVs and users in their communities. Although global content popularities can be estimated by introducing collaboration among the local popularity estimation modules, such collaboration would fail to capture demand heterogeneity across the communities of different A-UAVs.

The limitations listed above can be addressed by employing a Federated Multi-armed Bandit (f-MAB) learning-based caching mechanism at the A-UAVs. This paradigm is able to leverage the expected reward maximization attribute of MAB and intelligence sharing nature of Federated Learning for caching decision at the A-UAVs. The f-MAB learning based caching policy is presented in the following section.

6.5 Federated Multi-Armed Bandit Learning for Caching

Once a A-UAV is deployed into a community, its subsequent action is to decide which contents to download (via its vertical link) and cache such that content availability to the requesting users can be maximized. This goal is achieved by employing a Top-k Multi-Armed Bandit learning agent in the A-UAV.

6.5.1 *Top-k* Multi-Armed Bandit Learning

Multi-Armed Bandit is a classic problem in reinforcement learning [130] and decision-making, where an agent is faced with a set of actions or “arms” to choose from, each associated with an unknown reward distribution. The objective of the agent is to maximize the total expected reward over a sequence of trials or rounds [127]. Formally, let there be N arms, denoted by $A_1, A_2, \ldots, A_N$. Each arm i has an unknown reward distribution with mean μ_i and variance σ_i^2. At each round t, the agent chooses an arm A_t and observes a reward R_t drawn independently from the reward distribution of the chosen arm. The agent's goal is to maximize the total expected reward R_T over T rounds, where T is the total number of rounds (time horizon):

$$R_T = max \sum_{t=1}^{T} E[R_t] \tag{6.2}$$

This thesis uses a variant of MAB called *Top-k Multi-Armed Bandit* [128]. Here, the agent has to choose k arms out of a larger set of N arms, as opposed to choosing one arm in classical MAB, and receives a reward for each arm in the chosen set. The goal of the agent is to maximize the total cumulative reward R_T obtained over a finite time horizon T:

$$R_T = max \sum_{t=1}^{T} \sum_{i=1}^{k} E[R_{i,t}] \tag{6.3}$$

6.5.2 Decentralized Caching using *Top-k* Multi-Armed Bandit

In the scenario of UAV-caching, there is a *Top-k* MAB agent in each A-UAV. Here, choosing each content for caching corresponds to choosing an arm. The 'k' of *Top-k* MAB agent corresponds to the caching capacity of A-UAV; i.e., $k = C_A$. The agent's aim is to select 'C_A' contents out of a larger set of 'N' contents to be cached in an A-UAV such that content availability to the users can be maximized. Here, the UAV-aided content dissemination system is the learning environment where the A-UAVs interact through their actions of choosing specific sets of contents to be cached, as shown in Figure 6.2. The feedback from the environment for the taken actions are in the form of rewards/penalties. Actions are rewarded when cached contents are requested by the users and are served to the users within the given tolerable access delay. Otherwise, the actions are penalized. The top C_A contents that accumulate most reward from the corresponding community and other communities are chosen to be cached at an A-UAV. It should be noted that the *Top-k MAB* agents in the A-UAVs are provided with no *a priori* information about the content popularity at the corresponding user communities.

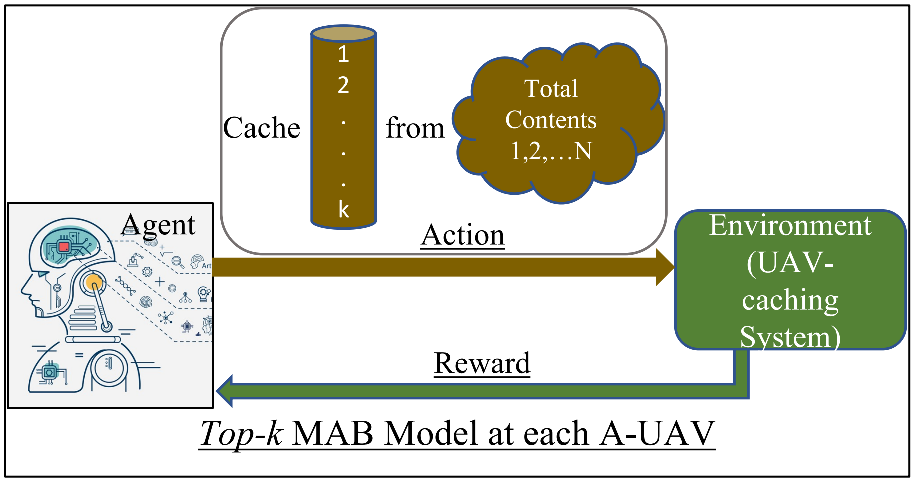

Cache
1
2
.
.
.
k
from
Total Contents 1,2,…N
Agent
Action
Environment (UAV-caching System)
Reward
Top-k MAB Model at each A-UAV

Figure 6.2. *Top-k* Multi-Armed Bandit Learning for Caching Policy at A-UAVs

A learning decision epoch for each *Top-k* MAB agent is set according to the F-UAVs accessibility at the corresponding community (i.e., an F-UAV's visiting frequency). This is because the F-UAVs carries the content availability information from the communities in its trajectory. Such content availability information is leveraged for learning at the A-UAVs' *Top-k* MAB agents using appropriately designed rewards. The agent learns to cache contents via a multi-dimensional numerical reward structure which has three parts: namely *local*, *ferrying,* and *global* reward. The first corresponds to the increase in availability at an A-UAV's corresponding community i.e., *increase in local availability* (δ_l). The second is related to the contents that are cached in an A-UAV, and are responsible for increase in availability at other communities i.e., *ferried content availability* (δ_f). A global reward is received when cached contents add to increase in average availability across all communities. This is called *increase in global availability* (δ_g). The three types of rewards are given below:

$$R_i^L = \begin{cases} 1, & for\ \delta_l > 0 \\ -1, & for\ \delta_l < 0 \end{cases} \tag{6.4}$$

$$R_i^F = \begin{cases} 1, & for\ \delta_f > 0 \\ -1, & for\ \delta_f < 0 \end{cases} \tag{6.5}$$

$$R_i^G = \begin{cases} 1, & for\ \delta_g > 0 \\ -1, & for\ \delta_g < 0 \end{cases} \tag{6.6}$$

In the above equations, R_i^L, R_i^F, and R_i^G are rewards according to increase in availability for content 'i' cached in an A-UAV.

Using the aforementioned *Top-k* MAB model, a A-UAV agent learns the caching policy that can serve the user requests which increases content availability across the communities. Learning is achieved using a tabular method where a Q-table is maintained for each action i.e., each content to be cached in an A-UAV. The value corresponding to each content is called a Q-value or action-value [127], [130]. The agent updates the Q-value for a content at every learning epoch according to the rewards in Eqns. 7-9 from the interaction with the environment (UAV-aided content dissemination system) and learns the best actions (contents cached). The expression which explains Q-value update for a content 'i' is given as follows:

$$Q(i) \leftarrow Q(i) + \alpha[r(i) - Q(i)] \tag{6.7}$$

Here, $Q(i)$ represents the Q-value of a content 'i'; $r(i)$ is the reward received by caching content 'i'; α is a hyper-parameter which controls the learning rate. The Q-values for all contents are initialized with zero to ensure no *a priori* information for a *Top-k* MAB agent. Also, it ensures equal importance to all contents for caching decisions. An epsilon-greedy (ϵ-greedy) exploration strategy is implemented. Such exploration strategy guarantees that every content gets to be cached in an A-UAV. As learning progresses, exploration decays and best contents with highest Q-values are exploited with the aim of maximizing accumulated reward which increases content availability.

Based on Algorithm 6.1, which captures the concept discussed thus far, the *Top-k* MAB agents in the A-UAVs learn the caching policy. After learning converges, the contents cached at A-UAVs emulate the cache pre-loading segmentation behavior. However, the caching policy and the corresponding average content availability remains oscillatory due to the low request rates for the less popular contents. Due to low sampling (request generation) of less popular contents, their reward estimates are weak (or unstable). This means that the Q-value of highly popular contents are less sensitive to content requests, whereas the Q-values for less popular contents are very sensitive to requests. In other words, when a request for a popular content is generated and served to the requesting user, the updated Q-value of that content doesn't change drastically. However, when an unpopular content is requested and served its Q-value changes abruptly. A sudden increase or decrease in Q-value of less popular contents may result in its addition to or removal from the cache of a A-UAV. This leads to intermittent variations in caching of some contents, which corresponds mostly to cached contents in Segment-2, as mentioned in cache pre-loading policies. Such oscillation depends on and can be controlled by the choice of ϵ and its decay rate. The following pseudo code explains the caching policy at a A-UAV with a *Top-k* MAB agent.

Algorithm 6.1. Caching policy at a A-UAV with *Top-k* MAB Learning

```
1. Initialization:
        a. N: Total contents in the system
        b. $C_A$: Caching capacity of an A-UAV
        c. $Q$: Array of size $C_A$ initialized with 0's (Q-table).
        d. $\epsilon$: Exploration rate
        e. $\alpha$: Learning rate for Q-table update.
2. Load A-UAV's cache with $C_A$ randomly chosen contents.
3. while True:
4.     if F-UAV is visiting A-UAV then do
       \\ Check if F-UAV visits A-UAV i.e., for learning epoch
5.         for $i = 0$ $to$ $length$(A-UAV cache size $C_A$) do
```

Algorithm 6.1. (cont'd)

```
        \\ Loop through every content in A-UAV
            **Get** reward $r(i)$  \\ According to Eqns. 7-9
            **Update** $Q(i)$      \\ $Q(i) \leftarrow Q(i) + \alpha[r(i) - Q(i)]$
                                   \\ $Q(i) \leftarrow U(i)$ if UCB employed
        **end for**
        $value \ = \ \boldsymbol{copy}(Q)$ \\ make a copy of Q-table
        **for** $i = 0 \ to \ length($A-UAV cache size $C_A)$ **do**
                \\ Reload contents (Select arms)
            **Generate** random number '$x$'
         \\ For $\epsilon$-Greedy action selection strategy
            **if** $x < \epsilon$ **then do**
                **Load** 1 randomly chosen content to A-UAV
            **else**
                $c_{max} = \boldsymbol{argmax}(value)$
                **Load** $c_{max}$ to A-UAV
            **Set** $value[c_{loaded}] = -inf$
         \\ To avoid redundant reloading of same content
            **end if**
        **end for**
    **end if**
    **Check** for $\epsilon$ decay condition.
    **if** true **then do**
```

Algorithm 6.1. (cont'd)

```
23.         Update $\epsilon$
              \\ $\epsilon = \epsilon \times decay_\epsilon$ where, $decay_\epsilon = 0.99$
24.     end if
25. end while
```

To reduce the dependence of caching policy on the choice of ϵ, Upper Confidence Bound (UCB) strategy is used [127]. The *Top-k* MAB agent maintains an upper confidence bound on the expected reward of each content, and selects the set of C_A contents with highest UCB at each epoch.

$$U_t(i) = Q_t(i) + \sqrt{\frac{\alpha_u \log(t)}{N_t(i)}} \tag{6.8}$$

Here, $U_t(i)$ is the UCB of content 'i' at epoch 't'; $Q_t(i)$ is the updated Q-value at epoch 't'; α_u is a hyperparameter that controls the degree of exploration; $N_t(i)$ is the number of time content 'i' has been requested till epoch 't'. The first term represents the reward estimate, and the second term depicts the uncertainty in reward estimate. UCB selects the content that has high potential for high reward but hasn't been requested frequently. This promotes exploration without externally inducing an exploration parameter such as ϵ. For this chapter, ϵ-greedy exploration strategy is applied according to the UCB values, as shown in Step 7-16 in Algorithm 6.1.

This *Top-k MAB* agent at a A-UAV learns a near optimal caching policy within a finite time horizon and approaches the best caching policy asymptotically. However, the learning method for caching encounters the following limitations. First, since the global content availability information is ferried by F-UAVs, a large disaster area with multiple communities makes the learning sluggish. Second, communities with a smaller number of users will results in reduced content requests received by the local A-UAV. In such scenarios the requests for less popular contents are even less due to heavily skewed content popularity at communities that follows a Zipf distribution. This leads to weaker estimate of reward distribution and sensitive (unstable) Q-values for less popular contents, thus making the caching policy very sensitive to unpopular content requests. Finally, there can be $\binom{N}{k}$ combination of contents to be sampled by the *Top-k MAB* agent for caching. Due to this the reward estimation for each content occurs after large intervals, which leads to a weak estimate of reward distribution as N increases.

These limitations can be alleviated by using a Federated Multi-Armed Bandit Learning for caching policy that aggregates *Top-k MAB* models from all the A-UAVs. The mechanism is explained below.

6.5.3 A Brief on Federated Aggregation

Federated learning (FL) [131], [132], [133] is a distributed machine learning technique that allows multiple devices/servers to collaboratively train a model without actually sharing their local data with a central server/arbitrator. In this approach, data remains on individual devices/servers, and only model updates are exchanged between them.

Some of the popular federated learning techniques include federated aggregation (also called weighted aggregation) [131], [132], secure aggregation [133], and differential privacy [132] aggregation. In federated aggregation and weighted aggregation, each device's model update is multiplied by a weight which reflects the importance or contribution of a device towards the final model. These models are then combined by taking a weighted average, which is called an aggregated model. For secure aggregation, the model parameters of a device are encrypted before sending them to another device for aggregation. In differential privacy, random noise is added to model updates before sending them to another device, in order to preserve privacy of individual devices. Such noisy update of models is done using Laplace, Gaussian, or exponential mechanisms

[132]. Overall, federated learning enables improved global model training, decentralized training, privacy, security, and scalability.

For the application in this thesis, improved global training and scalability attributes of FL is focused on. Federated learning is achieved using the federated or weighted aggregation technique to combine model updates that reflects a collective knowledge of all devices in the system. Federated aggregation process typically involves three main steps:

Initialization: Let us assume, 'm' devices in the system and each device 'i' is initialized with a model represented by a vector of parameters denoted by 'w_i'.

Local training: Each device/server 'i' trains the model locally using its own local data and updates 'w_i'.

Model aggregation: The aggregation is done at a central server which is chosen *a priori* based on the idea of a central arbitrator or a group leader. This central server contains the weight 'W_i' of each device based on its importance or intended contribution towards the aggregated model. One common approach to assign weight to a device is by using the number of data samples 'n_i' available at each device 'i'. This is shown below:

$$W_i = \frac{n_i}{\sum_{\forall i} n_i} \tag{6.9}$$

Another approach is to use some measure of a device's performance, like accuracy 'acc_i' of the device's local model, to assign weight to a device.

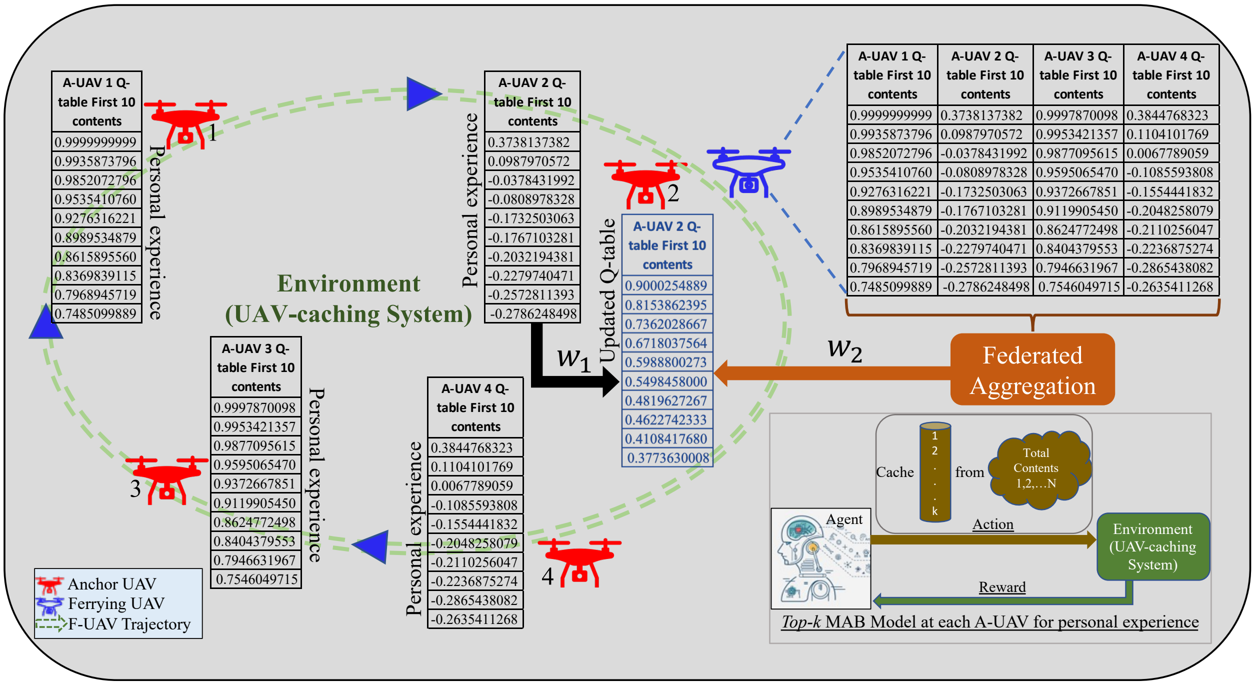
$$W_i = \frac{acc_i}{\sum_{\forall i} acc_i} \tag{6.10}$$

In general, the choice of weights depends on the specific application and the aim of the federated learning system. The locally trained models are sent to the central server and aggregated to create a new, improved model via weighted aggregation, as shown in the expression below:

$$w_{aggregated} = \frac{\sum_{i=1}^{m} W_i . w_i}{\sum_{i=1}^{m} W_i} \tag{6.11}$$

where '$w_{aggregated}$' is the aggregated model which improves the desired performance by giving more weight to better performing individual models and less weight to adversely performing models.

One advantage of federated learning is that it reduces the need to transfer large amounts of data to a central server/arbitrator for processing. This can be particularly useful in situations where the data is large, and network connectivity is limited or expensive. Another advantage is that Federated learning also enables real-time updates of models, making it possible to quickly adapt to changes/anomalies in the data or environment. The application in this chapter leverages these advantages of Federated Learning.



A-UAV 1 Q-table First 10 contents
0.9999999999
0.9935873796
0.9852072796
0.9535410760
0.9276316221
0.8989534879
0.8615895560
0.8369839115
0.7968945719
0.7485099889

A-UAV 2 Q-table First 10 contents
0.3738137382
0.0987970572
-0.0378431992
-0.0808978328
-0.1732503063
-0.1767103281
-0.2032194381
-0.2279740471
-0.2572811393
-0.2786248498

A-UAV 3 Q-table First 10 contents
0.9997870098
0.9953421357
0.9877095615
0.9595065470
0.9372667851
0.9119905450
0.8624772498
0.8404379553
0.7946631967
0.7546049715

A-UAV 4 Q-table First 10 contents
0.3844768323
0.1104101769
0.0067789059
-0.1085593808
-0.1554441832
-0.2048258079
-0.2110256047
-0.2236875274
-0.2865438082
-0.2635411268

A-UAV 2 Q-table First 10 contents
0.9000254889
0.8153862395
0.7362028667
0.6718037564
0.5988800273
0.5498458000
0.4819627267
0.4622742333
0.4108417680
0.3773630008

A-UAV 1 Q-table First 10 contents	A-UAV 2 Q-table First 10 contents	A-UAV 3 Q-table First 10 contents	A-UAV 4 Q-table First 10 contents
0.9999999999	0.3738137382	0.9997870098	0.3844768323
0.9935873796	0.0987970572	0.9953421357	0.1104101769
0.9852072796	-0.0378431992	0.9877095615	0.0067789059
0.9535410760	-0.0808978328	0.9595065470	-0.1085593808
0.9276316221	-0.1732503063	0.9372667851	-0.1554441832
0.8989534879	-0.1767103281	0.9119905450	-0.2048258079
0.8615895560	-0.2032194381	0.8624772498	-0.2110256047
0.8369839115	-0.2279740471	0.8404379553	-0.2236875274
0.7968945719	-0.2572811393	0.7946631967	-0.2865438082
0.7485099889	-0.2786248498	0.7546049715	-0.2635411268

Figure 6.3. Federated Multi-Armed Bandit Learning for Caching Policy at Anchor UAVs

6.5.4 Distributed Caching with Federated Multi-Armed Bandit

For UAV-caching scenario, each device in the classical definition of Federated Learning is analogous to a A-UAV. The local data, based on which each UAV's *Top-k* MAB model is updated, corresponds to the information about the cached contents, cache hits, and content availability. To be noted that, cache hits comprise of the number of times a content cached in a A-UAV is requested and served within the given tolerable access delay (TAD). The model, in this case, refers to the Q-table of the *Top-k* MAB agent of A-UAV. The role of central server for model aggregation is played by the mobile F-UAVs. F-UAVs are chosen because of their ability to access model parameters (i.e., Q-tables) of all the A-UAVs in their respective trajectories. The aggregated model at an F-UAV is sent to the A-UAV in its vicinity using the lateral link with an objective of improving the *Top-k* MAB model developed at the A-UAV. The aim of this updated *Top-k* MAB model is to help a A-UAV to decide its respective contents to be cached based on the top 'C_A' Q-values. Since this updated caching method possesses the model sharing attribute of Federated Learning and expected reward maximization attribute of MAB, this is called as Federated Multi-Armed Bandit Model (f-MAB).

According to the first step in standard learning paradigm of Federated Learning, the Q-table is initialized at each A-UAV. The learning epoch is set based on the F-UAV visiting frequency. Similar to the Q-value update procedure from *Top-k* MAB, Q-values of individual *Top-k* MAB agents of the A-UAVs update at each learning epoch. The Q-tables are updated following the recursive equation for reward estimate of MAB, as shown in Eqn. 6.7. These Q-values captures the respective communities' content request pattern and the respective A-UAVs' caching decisions. This is called *personal experience* of *Top-k* MAB model of a A-UAV (refer to Figure 6.3). This process is synonymous with the second step i.e., local training stage in a standard

Federated Learning paradigm. After gaining personal experience for an epoch duration, the model at a A-UAV can improve using models of its adjacent A-UAVs via the F-UAVs. The model aggregation operation in f-MAB also occurs according to the frequency of F-UAV visits near a A-UAV in its trajectory. To be noted that quality of the aggregated model depends on the freshness of information (i.e., Q-tables) ferried by an F-UAV. Therefore, with increase in the number of F-UAVs in UAV-aided caching system the quality of aggregated model improves.

Model aggregation in case of regression/classification based model such as neural networks involve aggregation of model parameters such as coefficients/weights [131]. For f-MAB, model aggregation involves aggregation of Q-values. As explained in the previous sub-section, every device has a weight associated with it which is related to its importance during aggregation. Similarly, in the case of UAV-caching using f-MAB, the weight for each A-UAV's model, when the F-UAV is visiting A-UAV 'x', is given by the following expression.

$$W_{x,y} = \frac{1-KL(P_x||P_y)}{\sum_{l=1}^{m}(1-KL(P_x||P_l))} \tag{6.12}$$

Here, '$W_{x,y}$' represents the weight associated with A-UAV y's model when the F-UAV is at A-UAV x. The parameter 'm' is the total number of A-UAVs in the system. P_x and P_y are content popularity distributions estimated at the A-UAVs x and y, respectively. $KL(P_x||P_y)$ is the Kullback-Leibler divergence or relative entropy [134] which measures the difference between the distributions. Using these weights from Eqn. 6.12, the aggregated model is computed using the following expression.

$$Q_x^{agg}(i) = \sum_{y=1}^{m} W_{x,y}.Q_y(i) = \frac{\sum_{y=1}^{m}\left(1-KL\left(P_x||P_y\right)\right).Q_y(i)}{\sum_{l=1}^{m}(1-KL(P_x||P_l))} \tag{6.13}$$

In Eqn. 6.13, $Q_x^{agg}(i)$ refers to the aggregated Q-value of content 'i' at A-UAV 'x'. The physical significance of using KL divergence to compute weight or importance of a A-UAV's model is as follows. The term “$1 - KL\left(P_x || P_y\right)$” represents how similar the content popularity distributions are at communities near A-UAV ‘x’ and ‘y’. Therefore, if estimated content popularity distribution of A-UAV ‘y’ is similar to that of A-UAV ‘x’ the weight associated with A-UAV y’s model is more and vice versa.

After the aggregation of Q-tables at the visiting F-UAV, the aggregated Q-table ‘Q^{agg}’ replaces the existing Q-table at a A-UAV. This is shown in Figure 6.1 where F-UAV i is visiting A-UAV x, therefore it has access to the Q-table of A-UAV x and model aggregation is done at F-UAV i. Whereas F-UAV j is transiting and has not reached the next A-UAV in its trajectory, hence no model aggregation takes place at F-UAV j.

These aggregated Q-tables improve the estimated reward associated with each content. The number of requests generated at a community during a learning epoch duration may not be sufficient for a better reward estimate at its respective A-UAV. Due to this, in the absence of model aggregation, the A-UAVs have weaker estimate of the reward distribution. Through Q-table aggregation, the estimated rewards are improved without physically getting content requests information across all the A-UAVs.

However, Q-table aggregation loses the context of local popularity when there exists demand heterogeneity across different communities. This problem is analogous to personalization-generalization problem in Federated Learning [131], [132]. A A-UAV’s Q-table updated with personal experiences corresponds to personalized (local) model and the aggregated Q-table represents the generalized (global) model. This chapter uses weighted averaging technique to preserve the local popularity context along with the improvement in reward estimate. This is given as:

$$Q_x^{upd}(i) = w_1 . Q_x(i) + w_2 Q_x^{agg}(i) \tag{6.14}$$

This is equivalent to

$$Q_x^{upd}(i) = w_1.\, Q_x(i) + \frac{e^{-\beta_d t}}{\beta_s}.\, \big(1 - Q_x(i)\big).\, Q_x^{agg}(i) \tag{6.15}$$

In equations 6.14 and 6.15, 't' is the epoch number. The weights w_1 and w_2 decide the contribution of the local and global (aggregated) models towards the updated model 'Q_x^{upd}'. From equations 6.14 and 6.15, w_2 can be expressed as follows:

$$w_2 = \frac{e^{-\beta_d t}}{\beta_s}.\, \big(1 - Q_x(i)\big) \tag{6.16}$$

Here, β_d and β_s represents the weight decay factor and scaling factor. The parameters ensures that the contribution of global (aggregated) model reduces as learning progresses. This idea is backed by the assumption that, as learning progresses, with the help of model aggregation the individual local models will be reflect the true value of contents. After this the requirement for model aggregation may not be necessary to improve reward estimate. Also, the expression in Eqn. 6.16 has the term "$\big(1 - Q_x(i)\big)$", which is a representation of *regret* [127]. This ensures that the updated f-MAB model can be better than the existing *Top-k* MAB model while maintaining the Q-value within the true value of a content.

For the simulation experiments, w_1 is set empirically to 0.99. The motivation behind choosing such a high value for w_1 is that the local content popularity at a community does not vary with time. To be noted that the choice of the weight w_1 in Eqn. 6.14 and 6.15 is essential, especially in scenarios with time-varying content popularity. In such scenarios, the content popularity estimates such as P_x, P_y (along with their Q-values) change with time. To determine the weight w_1 in such cases, the following expression can be used:

$$w_1 = 1 - \frac{JSD\big(P_x^t, P_x^{t'}\big)}{ln\,2} \Rightarrow w_1 = 1 - \frac{\left[\frac{1}{2}\Big(KL(P_x^t||M) + KL\big(P_x^{t'}||M\big)\Big)\right]}{ln\,2} \tag{6.17}$$

Here, $JSD\left(P_x^t, P_x^{t'}\right)$ is the Jensen-Shannon Divergence [134] which shows dissimilarity between P_x^t and $P_x^{t'}$. P_x^t and $P_x^{t'}$ are content popularity distributions at time t and t'. M is the average distribution calculated as $M = \left(P_x^t + P_x^{t'}\right)/2$. $KL(P_x||M)$ is the Kullback-Leibler divergence or relative entropy [134] which measures the difference between an individual distribution (P_x^t or $P_x^{t'}$) and the average distribution M. The weight w_1 will be high if the content popularity distribution doesn't experience substantial change within time t to t' and vice versa. This weight w_1 determines the importance of the local model in case of time-varying local popularity and the dependance of updated Q-table Q_x^{upd} on it.

Eqn. 6.15 can be rewritten by substituting w_1 and Q_x^{agg} as:

$$Q_x^{upd}(i) = \left(1 - \frac{\left[KL(P_x^t||M) + KL\left(P_x^{t'}||M\right)\right]}{2.ln\, 2}\right). Q_x(i) + \frac{e^{-\beta_d t}}{\beta_s}.\left(1 - Q_x(i)\right).\frac{\sum_{y=1}^{m}\left(1 - KL\left(P_x||P_y\right)\right).Q_y(i)}{\sum_{l=1}^{m}(1 - KL(P_x||P_l))} \tag{6.18}$$

The hyperparameters β_d and β_s can be explored empirically to ensure that caching policy at a A-UAV doesn't lose the local context of content popularity, especially when there exists heterogeneity in content popularity across communities.

The personalization-generalization problem handling in f-MAB based caching is shown in Figure 6.3 where model aggregation is done at F-UAV and the model at A-UAV '2' is updated using a weighted average of personal and aggregated model. Algorithmically, the implementation of f-MAB based caching policy at A-UAV is similar to *Top-k* MAB based caching except for line 7 in Algorithm 6.1. The Q-table update uses Eqn. 6.18 along with Eqn. 6.7 to incorporate Federated Learning along with the concepts of *Top-k* Multi-Armed Bandit. The new Q-table is used for caching decision, where top C_A contents with highest Q values are cached at an A-UAV. Such contents can boost content availability at their respective communities as well as at other distant communities via F-UAVs.

Table 6.1. Default Values for Model Parameters

#	**Variables**	**Default Value**
1	Total number of contents, C	1000
2	Number of A-UAVs, N_A	12
3	Number of F-UAVs, N_F	3
4	Cache space in A-UAV, C_A	200
5	Cache space in F-UAV, C_F	200
6	Poisson request rate parameter, μ	0.5 request/sec
7	Hover time of F-UAV, T_{Hover}	600 seconds
8	Transition time of F-UAV, $T_{Transition}$	300 seconds
9	Zipf parameter (Popularity), α	0.4
10	Ferrying UAV Trajectory	Round-robin

6.6 Experiments and Results

A discrete event simulator was used for experimentally evaluating the performance of the proposed f-MAB and *Top-k* MAB learning-based caching mechanisms. Content requests are generated using an exponential distribution for the inter-request intervals, and a Zipf distribution for the content popularity control (refer to Eqn. 6.1). Cache pre-loading is done using the mathematical expressions, to evaluate the best achievable benchmark performance when the content popularities across all communities are known *a priori*. Unless specified otherwise, the parameter values form Table 6.1 are used as defaults.

The following performance metrics are evaluated.

Content Availability (P_{avail}): It is defined as the ratio between cache hits and generated requests within a time interval. Cache hits are the content provided to the users from the caches in the UAV-aided caching system. Meaning, when a content is downloaded from the cloud because it was not available in the caches of both types of UAVs. In other words, content availability indirectly indicates the reduction of from-cloud download cost by deploying smart caching.

Cache Distribution Optimality (*CDO*): This determines the optimality of the learnt caching policy in terms of the caching sequence. Jaro-Winkler Similarity (JWS) [93] is used to represent *CDO*, by computing the similarity between the content sequence from the learnt caching policy and content sequence according to cache pre-loading. It is computed by calculating the number of matches, number of transpositions required within the matches and the similarity in prefix of both sequences. It is a normalized similarity measure where 1 represents optimal caching and 0 means non-optimal caching.

Access Delay (AD): Access delay is defined as the total latency between when a content request is generated and it is delivered to the user from the cache of any of the UAVs. *AD* is reported over time to demonstrate how it improves as the caching policy learning progresses.

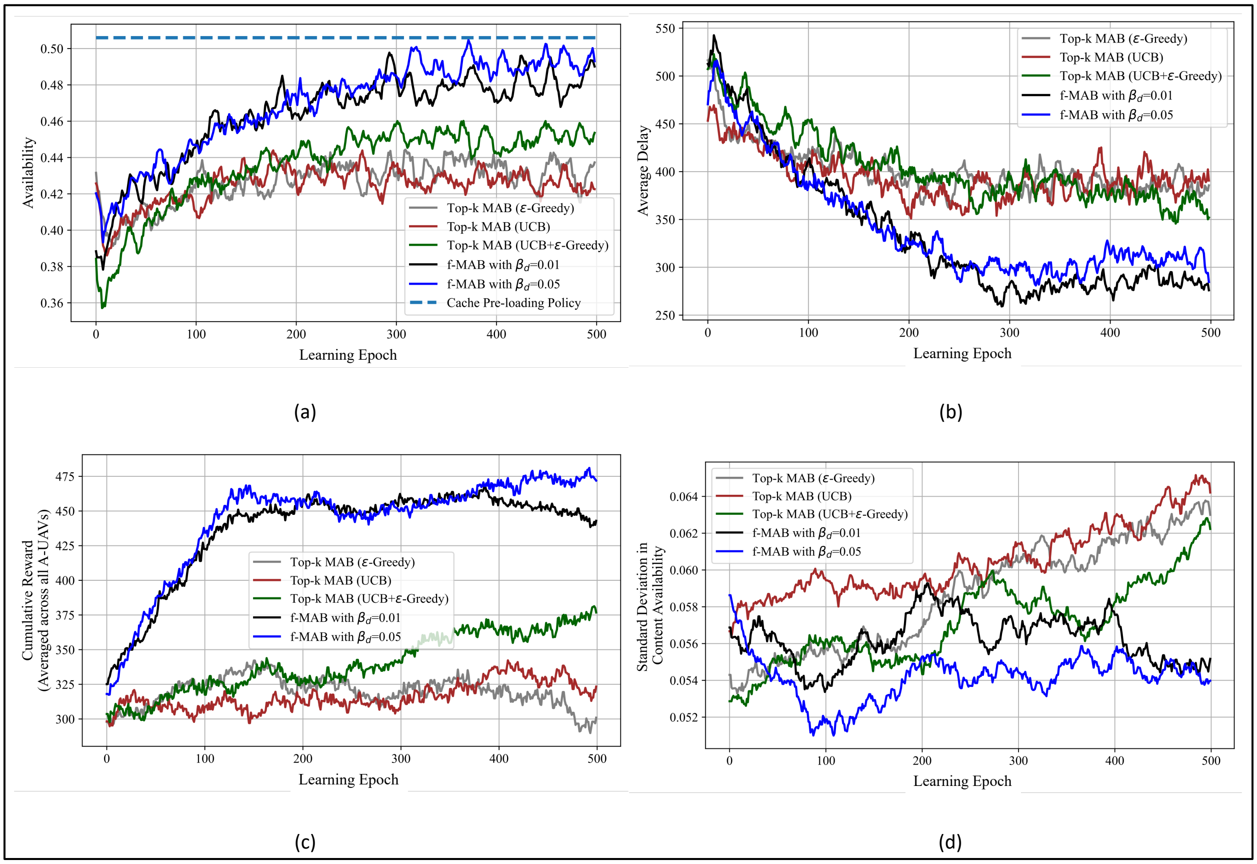
6.6.1 Effect of Caching Mechanisms and Exploration Strategies

In order to understand the viability of the proposed caching policies in scenarios with demand heterogeneity, a unique content popularity sequence is used for each community. In order to capture heterogeneity in content popularity sequence at different communities, contents are swapped with pre-decided probability [93] and the difference between the sequences are determined and maintained using Smith-Waterman Distance (SWD) [93]. This is a normalized distance measure where SWD value of 1 means that the content popularity sequences are completely different and an SWD of 0 means no difference in content popularity sequences. Additionally, two different request generation rates, 0.5 and 0.01 requests/ second, are used across the communities for capturing demand heterogeneity. To implement f-MAB, the weight decay factor is set to $\beta_d = 0.01, 0.05$ and scaling factor of $\beta_s = 2$ is chosen empirically. Two values of β_d are used to demonstrate the effect of personalization-generalization problem in Federated Multi-Armed Bandit Learning, which is explained later. For the ϵ-greedy strategy, initial exploration is set as $\epsilon = 1$, which is made to decay at the rate of 0.0025 per learning epoch. The degree of exploration in Upper Confidence Bound (UCB) exploration strategy is set to $\alpha_u = 2$.

Figure 6.4 shows the convergence behavior of the learnt caching policies with a comparison of f-MAB and different exploration strategies employed in the *Top-k* MAB model. The graph in Figure 6.4a is shown in learning dynamics in terms of the improvements in content availability.

The observations from Figure 6.4a are as follows. First, the figure shows that by employing f-MAB agent at every A-UAV, a near-optimal caching policy can be learnt. The algorithm is able to leverage the intelligence sharing attribute of federated learning in order to achieve content availability that is close to the benchmark performance. The model sharing approach in federated learning reduces the inherent dependance of A-UAVs' MAB models on their respective content requests only. By including the aggregated model for Q-table updates (see Eqn. 6.18), the Q-values at each A-UAV captures the requests generated across all communities. Such Q-values represent improved reward estimates, which leads to better learning towards a more effective caching policy.

The said improvement in reward can be seen in Figure 6.4c, where f-MAB ensure consistent higher rewards created by the learning of an improved caching policy.



(a)

(b)

(c)

(d)

Figure 6.4. Comparison between f-MAB, different exploration strategies in *Top-k* MAB and Cache Pre-Loading in terms of (a) Content Availability; (b) Access Delay; (c) Cumulative reward; (d) Epoch-wise Standard Deviation in Content Availability of A-UAVs

The second observation is that with an increase in weight decay factor β_d, the content availability increases. As discussed previously, the weight decay factor helps in balancing the personalization-generalization problem in Federated Multi-Armed Bandit Learning. Physically, this means that a very slow decay of aggregated model's weight (refer to Eqn. 6.16) may increase generalization, leading to a replicated caching behavior across all A-UAVs. The effect of over generalization can also be observed in Figure 6.4c, where, as learning progresses, the line corresponding to $\beta_d = 0.01$

accumulates less rewards as compared to the one with $\beta_d = 0.05$. A more detailed analysis of weight decay factor and its effect on content availability is provided in Figure 6.5.

The third observation is regarding the performance comparisons between the f-MAB and the *Top-k* MAB approaches with various exploration strategies. It is shown that the multi-dimensional reward structure of the *Top-k* MAB models at the A-UAVs help generating caching policies that show performance improvement during the initial learning epochs. These were also highlighted as through Eqns. 7-9. As the learning progresses, the performance improvement tapers off after a point of learning. This effect is due to the insufficiency of content requests at individual A-UAVs which leads to weak estimated Q-values.

Finally, when the agent uses the standalone UCB exploration strategy, the content availability settles at a sub-optimal value. However, during the initial learning epochs, the content availability increases promptly due to high upper confidence value of all contents, which avoids exploitation. This can be seen in Eqn. 6.8, where $N_t(i)$ represents the number of requests generated for content 'i' during a learning epoch 't'. During the initial learning epochs, requests generated for all contents are less, which keeps their upper confidence values high. Physically, this means that due to initial high confidence on all contents, the model fails to prioritize a subset of contents to cache, thus leading to exploratory behavior. As the learning progresses, the sparse requests for unpopular contents keep the upper confidence value high which maintains consistent exploratory behavior. An algorithmically induced ϵ value in ϵ-greedy strategy avoids this consistent exploratory behavior due to ϵ decay. However, ϵ is a predetermined exploration parameter which is controlled by its decay rate (refer Algorithm 6.1). A faster decay can limit the exploration capability of the proposed algorithm, thus forcing it to converge to a suboptimal learnt caching policy.

Therefore, to maintain the initial surge in content availability and to limit the unbounded exploratory behavior, ϵ-greedy exploration is applied on the upper confidence bound values of the content. Such hybrid exploration strategy helps to boost the content availability beyond their respective non-hybrid performance. Specifically, such hybrid exploration strategy applied in conjunction with f-MAB approach is able to achieve a performance improvement of 7% compared to the *Top-k* MAB with any standalone exploration strategy.

Figure 6.4b shows the convergence behavior of f-MAB and *Top-k* MAB in terms of access delay. This is computed for a TAD of 1800 seconds and it is observed that as learning progresses, the access delay for requested contents reduces while the content availability increases. The best reduction in access delay is observed when f-MAB is applied in tandem with the described UCB/ϵ-greedy hybrid exploration.

Another way of representing the learning convergence behavior is the standard deviation (SD) in epoch-wise content availability across all A-UAVs. This characterizes the fairness in learnt caching behavior across all A-UAVs, as learning progresses. The physical significance of observing the standard deviation of availability is as follows. A content 'i' with low popularity cached at A-UAV 'x' can assist to increase availability at A-UAV 'y' via F-UAV. If popularity of 'i' is high at the A-UAV 'y', it serves more user requests and commensurately achieves high ferrying reward R_i^F and global reward R_i^G at the A-UAV 'x' (refer to Eqns. 8 and 9). If ferrying reward R_i^F and global reward R_i^G have high values, Q-value of 'i' increases at A-UAV 'x', which leads to caching of a low popularity content. This violates the estimation criteria for non-IID samples where a content is cached at a A-UAV depending on its estimate from another A-UAV with different content popularity preferences. This phenomenon contributes to the standard deviation in content availability.

The comparison in Figure 6.4d reveals that the f-MAB strategy decreases the standard deviation in availability, indicating synchronized learning of caching policies among all A-UAVs. In contrast, the *Top-k* MAB strategy exhibits an increased standard deviation, implying unfair learning behavior among A-UAVs due to non-identically independently distributed (non-IID) content request patterns. The proposed f-MAB approach minimizes the absolute dependence on the reward structure and incorporates the concept of local and global popularity through personalized and aggregated models. This leads to fairly simultaneous content availability improvement across communities, as depicted in Figure 6.4d. The residual standard deviation after convergence is due to the inherent demand heterogeneity and sparse intra-community request generation. Figure 6.4d demonstrates the superior performance of f-MAB over *Top-k* MAB for learnt caching decision-making.

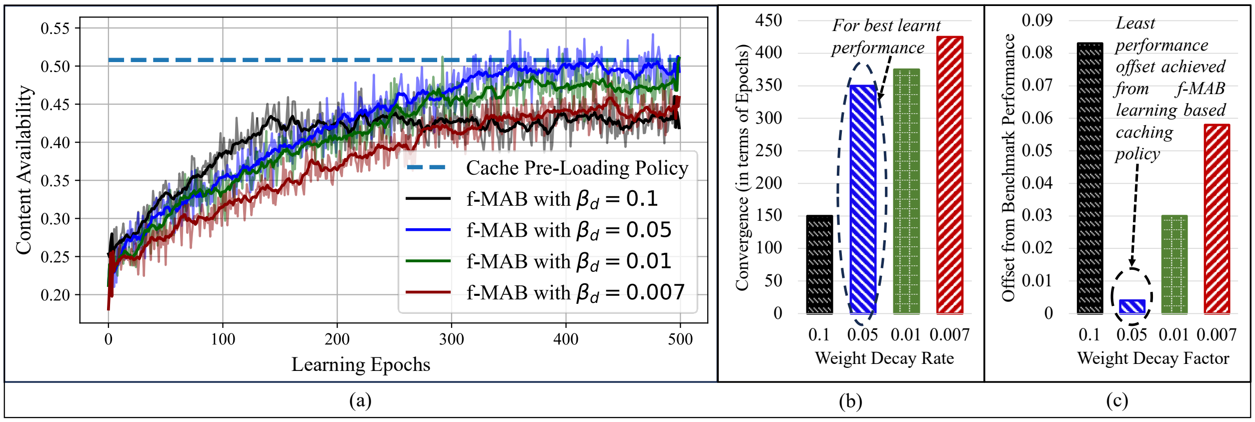
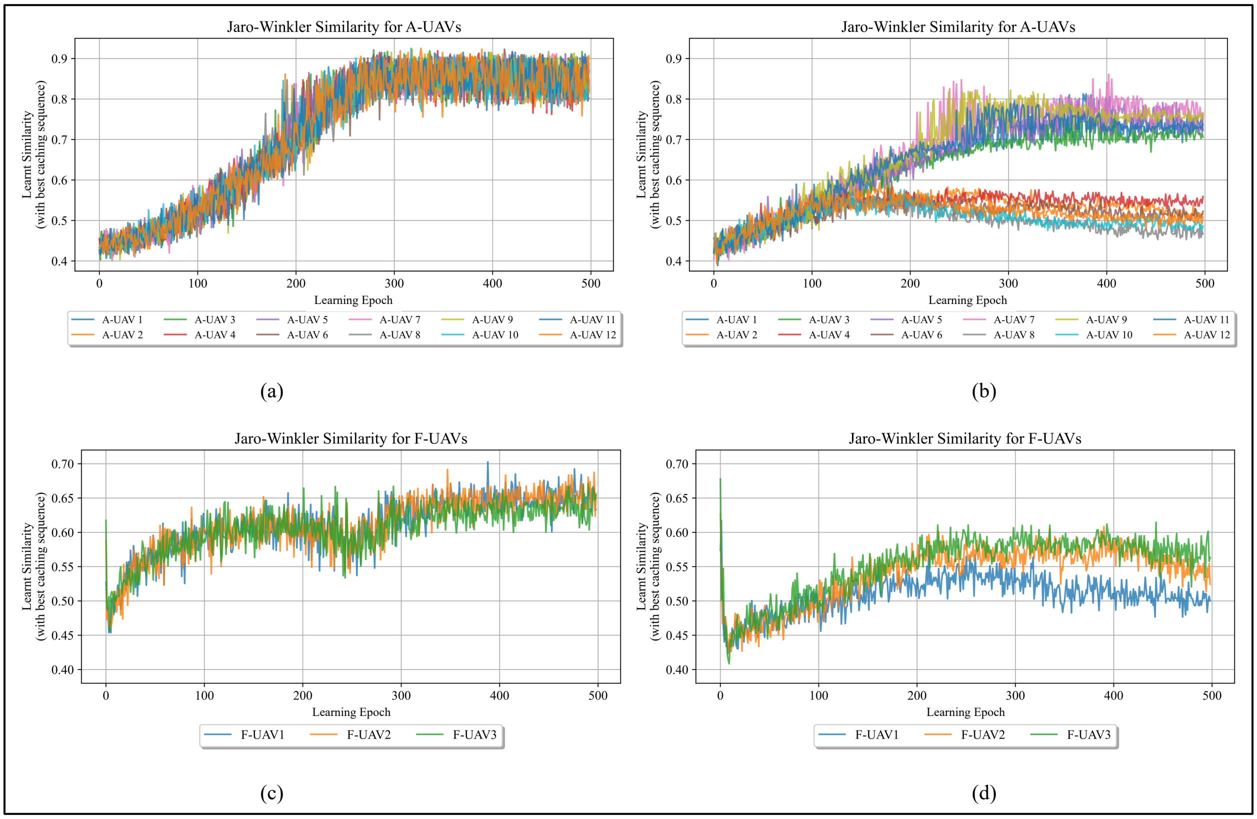


Figure 6.5. (a) Effect of weight decay factor β_d on content availability; (b) Convergence with β_d; (c) Offset from benchmark performance

Note that in f-MAB, the aggregated model (see Eqn. 6.15) and its contribution towards the update of the Q-values is controlled by a weight decay factor β_d (see Eqn. 6.16). The effects of β_d on content availability is shown in Figure 6.5. The observations are as follows. First, the best content availability is achieved with $\beta_d = 0.05$. It can also be observed in Figure 6.5c that the achieved content availability offset from the benchmark performance is minimum, as shown. Second, a higher weight decay factor ensures an initial surge in performance, but it tapers as the learning progresses. Figure 6.5c shows that with such β_d, the learning performance settles down at a suboptimal value. Finally, lower values of β_d make the learning sluggish, which can be observed in Figure 6.5a and 6.5b. Also, the suboptimality of the achieved contentavailability can be seen in Figure 6.5c. Therefore, weight associated with aggregated model Q_x^{agg} must be computed with careful and empirical selection of β_d.



(a)

(b)

(c)

(d)

Figure 6.6. (a) *JWS* at A-UAVs with f-MAB; (b) *JWS* at A-UAVs with *Top-k* MAB; (c) *JWS* at F-UAVs with f-MAB; (d) *JWS* at F-UAVs with *Top-k* MAB

6.6.2 Quality of Learnt Cache Sequence

This section reports the quality of algorithmically learnt cache sequences in terms of their similarities with the theoretically best possible cache sequence. To be noted that the best possible caching sequence can be derived from cache pre-loading policies. The quality of a learnt caching policy is reported in terms of cache distribution optimality (CDO) which can be calculated from Jaro-Winkler Similarity (JWS) [93].

Cache distribution optimality can be inferred as an indirect representation of the storage segmentation factor (λ), which is used to decide the segment sizes according to cache pre-loading policies. A higher JWS implies that, along with learning the caching policy, the MAB agents learn to emulate the said segmentation behavior. The cached contents become close to optimum as the learning progresses. Lower JWS values at initial epochs signifies that the A-UAVs have no *a priori* content popularity information, neither local nor global. As the MAB agents learn in time with generated content requests, the cached contents in the A-UAVs become more similar to the best caching sequence. Thus, indirectly, it learns to emulate cache segmentation along with the increase in cache distribution optimality. The partial dissimilarity of the cached content sequence can be ascribed to the uncertainty associated with the Q-values of contents. Also, this leads to an oscillatory convergence of JWS for A-UAVs (refer Figure 6.6a and 6.6b). This behavior manifests in the JWS for F-UAVs as well, due to its dependance on caching decisions at A-UAVs.

Figs. 6.6a and 6.6b plot Jaro-Winkler Similarity of cached content sequences for all 12 A-UAVs while employing f-MAB and *Top-k* MAB, respectively. The key observations are as follows. First, the JWS between the best caching sequence with cache pre-loading policy and the learnt caching sequences with f-MAB agents converge near 0.9, although with a certain variance. Physically, this represents high cache distribution optimality, where 1 indicates complete similarity and 0 indicates no similarity. Second, the learnt caching sequence with *Top-k* MAB agents show initial increase in learnt similarity. However, it tapers off as learning progresses. That is due to the subpar Q-values of content, which can be seen from the weak reward estimates in Figure 6.4c. Third, it can be seen that learnt caching sequences with *Top-k* MAB has high variance. The reason is twofold: global rewards' precedence over local penalties and the agent's unawareness about global popularity. Intermittently, cached contents accumulate huge rewards due to global rewards which supersedes local penalties. This leads to bad caching decisions locally, thus resulting in reduced content availability. Also, an agent's unawareness about global popularity fails to limit the offset due to bad caching decision. Finally, the adeptness of learnt caching sequence at the A-UAVs affects the learnt caching sequence at the F-UAVs. Figure 6.6c and 6.6d shows the JWS of cached content at 3 F-UAVs with f-MAB and *Top-k* MAB, respectively. Since, F-UAVs ferry contents that are requested less frequently, the low popularity of such contents leads to a comparatively sluggish improvement of its JWS as compared to JWS improvement for the A-UAVs. Due to bad caching decisions with *Top-k* MAB at A-UAVs, the caching decision at F-UAVs is affected more as compared to f-MAB, which is shown in Figure 6.6c and 6.6d. It shows 10-15% less JWS for *Top-k* MAB as compared to f-MAB learning-based policy, which indicates lower cache distribution optimality.

6.6.3 Impacts of Tolerable Access Delay

To gain insights about the learning capabilities of the proposed f-MAB and *Top-k* MAB models, experiments are conducted with varying tolerable access delays (TAD) ranging from 1200 to 2400 seconds. The content availability according to the learnt caching policies with varying TAD is shown in Figs. 6.7a and 6.7b. The figures demonstrate the behavior of the proposed caching mechanisms viz. f-MAB and *Top-k* MAB with respect to the benchmark performance, which is computed using the cache pre-loading policy. The two figures 6.7a and 6.7b are different ways to emphasize the learning behavior for varying TAD scenarios.

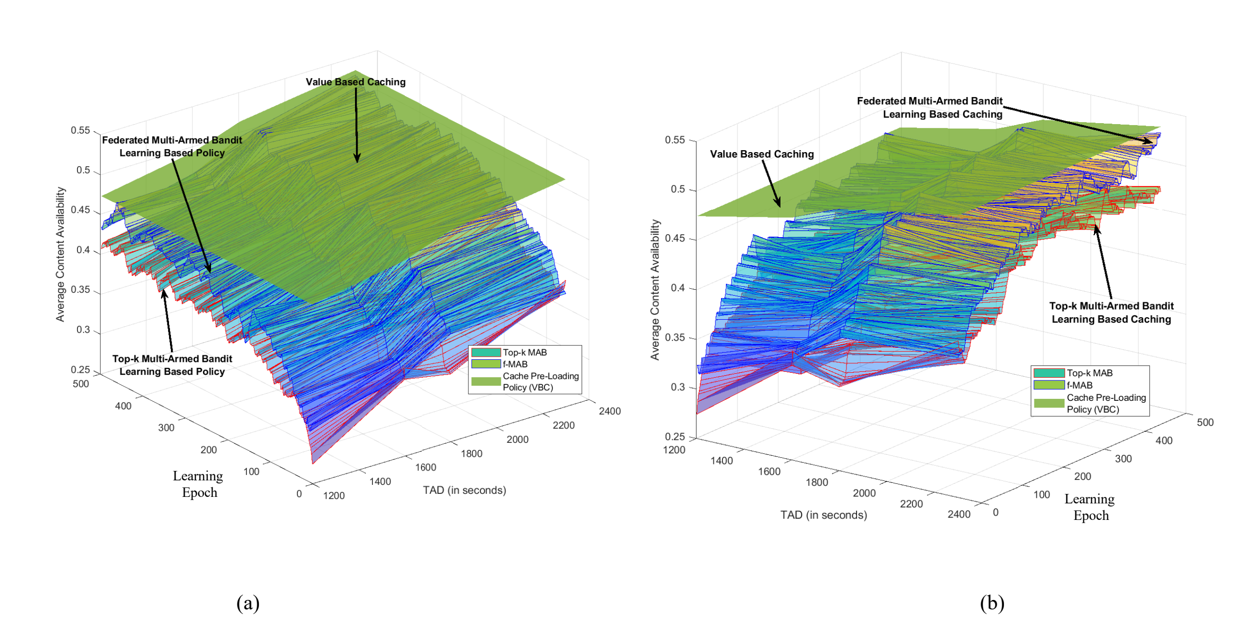
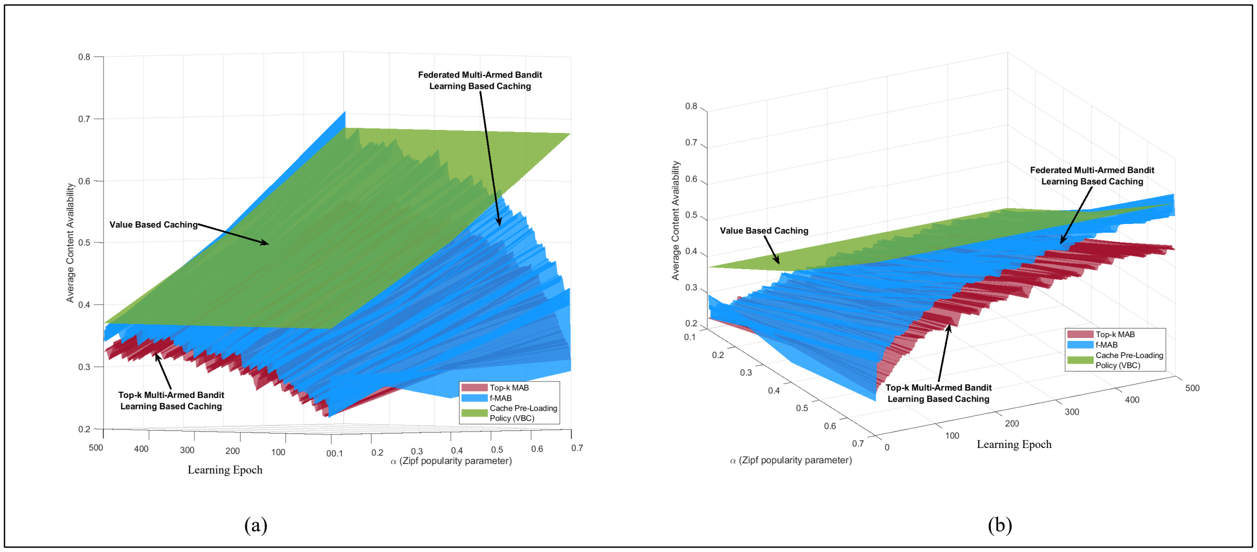


Figure 6.7. (a-b) Two different ways to show content availability performance with different *TADs*

Following observations can be made from Figure 6.7. First, the learnt caching policy with f-MAB learning based caching mechanism achieves performance closer to the benchmark for all values of TAD. Second, Figs. 6.7a-b show that the best possible performance (i.e., the benchmark) changes with change in TAD. Third, the f-MAB and *Top-k* MAB agents in the A-UAVs adapt to the user defined TAD via dynamic learning. In other words, the role of multi-dimensional reward structure of MAB and model sharing approach of federated learning becomes more evident as content TAD increases. Especially, the information related to the global availability i.e., δ_f and δ_g, are derived from large count of content requests. This improves the estimated reward at the A-UAVs, thus impacting their caching decision. Since, f-MAB model leverages the personal experience of individual A-UAVs, with enhanced performance of *Top-k* MAB, f-MAB's performance improves commensurately.



(a)

(b)

Figure 6.8. (a-b) Two different ways to show content availability performance with different Zipf Popularity Skewness

6.6.4 Impacts of Content Popularity Skewness

Content popularity skewness, represented by the Zipf parameter α, can change the importance of all contents such that with increase in α the most popular content becomes more popular and the popularity of less popular content falls. Figs. 6.8a and 6.8b show two different ways to show the proposed learning-based mechanisms' ability to cope with different Zipf popularity skewness α. Both f-MAB and *Top-k* MAB policies adjust to the modification in α. Due to the increase in popularity of highly requested contents with increase in α, Q-values of popular contents develop comparatively faster than that with lower α. This behavior favors the learning progression of both f-MAB and *Top-k* MAB agents. Similar to the observation till now, f-MAB's performance, in terms of content availability, is better than that of *Top-k* MAB. However, this improvement comes with added pre-convergence computational complexity, which is shown in Figure 6.9. The computational load for both of the proposed caching methods are calculated for 1800 requests per epoch. The computational load for *Top-k* MAB is calculated for the recursive Q-value update equation [128], which is constant. On the other hand, for f-MAB computation scales with number of contents, due to the weight calculation using KL divergence (refer to Eqns. 15-21). Note that the additional computation with f-MAB tapers off post-convergence due to the improved Q-values of contents. Physically, this implies that the f-MAB caching agent has learnt to balance the local content requirements of the respective communities along with the global need of the disaster effected regions.

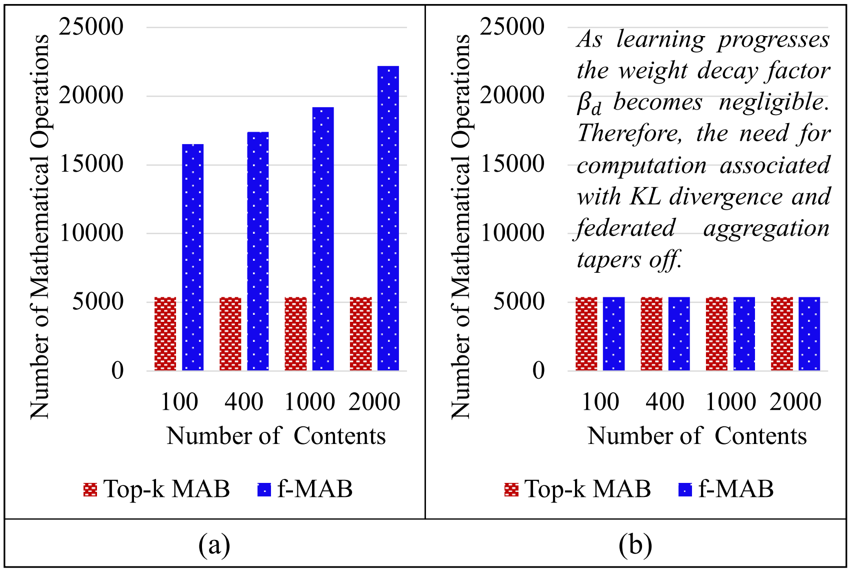


Figure 6.9. Computation complexity (a) before convergence, and (b) after convergence

It should be noted that the aforementioned experiments have been conducted and explained for a heterogeneous scenario to show the generalization capabilities of the proposed caching mechanisms.

A homogeneous demand scenario is a special case of the generalized heterogeneous case. With homogeneity in both content popularity and *TAD*, the benchmark performance is computed using Smart Exclusive Caching (*SEC*), whereas for homogenous *TAD* with community-specific content popularity, Popularity Based Caching (*PBC*) decides the benchmark performance. It should also be noted that both *SEC* and *PBC* are special cases of Value Based Caching (*VBC*). The proposed f-MAB and *Top-k* MAB models are still applicable in the aforementioned scenarios for on-the-fly learning of caching policies.

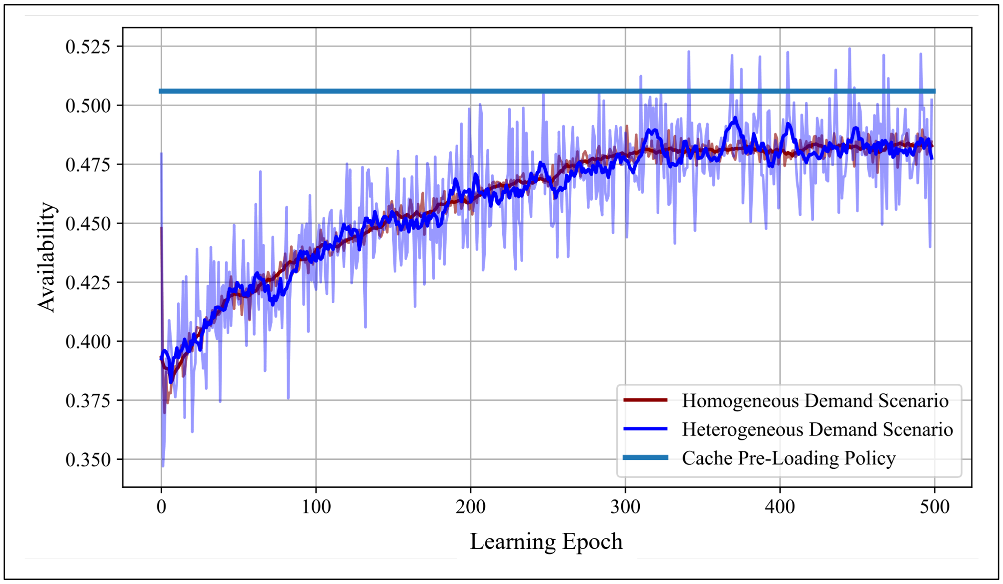


Figure 6.10. Federated Multi-Armed Bandit Learning based Caching Performance comparison in Heterogeneous and Homogeneous Scenarios

Figure 6.10 shows that applicability of f-MAB to learn the caching policy in heterogeneous as well as homogeneous demand scenarios. The performance improvements in both scenarios are comparable.

6.7 Summary and Conclusion

In this chapter, a UAV-aided content dissemination system is proposed which can learn the caching policy on-the-fly without *a priori* content popularity information. Two types of UAVs are introduced to support content provisioning in a disaster/war-stricken scenario viz. anchor and ferrying UAVs. Cache-enabled anchor UAVs are stationed at each stranded community of users for uninterrupted content provisioning. Ferrying UAVs act as content transfer agents across the anchor UAVs. The evolution of pre-loading-based caching policies, which requires *a priori* information about content popularity, are discussed. A decentralized *Top-k* Multi-Armed Bandit

Learning-based caching policy is proposed to ameliorate the limitation of cache pre-loading. It learns the caching policy on-the-fly by maximizing estimated reward for the increase in local and global content availability. To improve Q-value estimates, a distributed Federated-Multi-Armed Bandit Learning-based caching policy is proposed. This method combines the Q-values of all anchor UAVs to produce a better estimate of top popular content at a community. Future work on this research includes algorithmically coping with time-varying content popularity and adaptive trajectory planning in the presence of operational unreliabilities of the UAV. The next chapter includes the characterization of UAV trajectories and deployment strategies which can build the foundation for developing trajectory-aware learning-model sharing techniques to enhance content dissemination.

Chapter 7: Benchmarking UAV Trajectory-Aware Caching Policies in Infrastructure-Less Networks

7.1 Motivation

The pursuit of this research is driven by the critical need for reliable communication in environments where disasters or conflicts have compromised or completely destroyed traditional communication infrastructure. The urgency to develop solutions that can quickly and efficiently bridge these communication gaps is paramount. Unmanned Aerial Vehicles (UAVs) present a promising avenue for addressing this challenge due to their flexibility and rapid deployment capabilities. However, the effective use of UAVs in such scenarios requires a deep understanding of their operational dynamics. Specifically, how the planning of their flight paths, including hovering and transitioning behaviors, impacts the availability of essential content and the costs associated with its delivery. This chapter aims to fill this gap by exploring the intricate dynamics of UAV trajectory planning in communication-challenged scenarios.

7.2 Design Objective

The primary goal of this work is to enhance the accessibility of critical information in areas where standard communication systems are no longer viable. To achieve this, the chapter introduces a novel Joint Deployment of Ferrying UAVs (JDFU) algorithm designed to optimize content availability across various scenarios. This algorithm represents a significant advancement over traditional content caching and UAV deployment strategies by dynamically adjusting to the specific needs and constraints of disaster-affected environments. A key part of this objective is to understand the trade-offs between the UAVs' operational parameters and the tolerable delays in accessing requested content. By doing so, the research seeks to identify an operational sweet spot that ensures maximum content availability. Additionally, the development of simulation experiments and analytical models is crucial for validating the effectiveness of the proposed trajectory planning and deployment strategies. These will also be used in subsequent chapters as performance benchmarks for learning models that can develop trajectory-aware caching policies. These tools are not only instrumental in assessing the performance of the JDFU algorithm but also in fine-tuning the overall approach to UAV-aided content dissemination in challenging conditions.

7.3 System Model

7.3.1 UAV Hierarchy

The two-tier UAV-aided content dissemination system is shown in Figure 3.1, where each partitioned community of users is served by a A-UAV using a lateral wireless link such as Wi-Fi. With a naïve fully duplicated (FD) approach where A-UAVs download all contents requested by the users [120], with no inter-A-UAV data transfer, the following shortcomings will be encountered. First, there will be duplications of downloads via the expensive vertical links by different A-UAVs due to the overlaps in requests from different communities for popular contents. This will incur high download costs. Second, storage constraints will cap the number of contents that can be downloaded and stored in each A-UAV, thus limiting the content availability. Finally, due to limited infrastructure availability, some of the communities of users can be rendered isolated from content access without dedicated A-UAVs assigned to them.

To address these, a set of ferrying UAVs (i.e., F-UAVs) are introduced. Unlike A-UAVs, the mobile F-UAVs do not possess vertical links, but they do have lateral links such as Wi-Fi, using which they can communicate with the A-UAVs and the users. The role of these UAVs is to cache and transfer content around the A-UAVs such that the users in a community are able to access content that was downloaded by A-UAVs serving other communities.

After receiving a request from one of its community users, a A-UAV first searches its local storage for the content. If not found, it waits for a potential future delivery of the content by one of the traveling F-UAVs. If no F-UAV with that content arrives within the specified tolerable access delay (TAD), the A-UAV downloads it via the vertical link.

To address the question about trajectory planning in terms of F-UAV trajectories, different pre-programmed trajectories are characterized along with below mentioned static content placement strategies.

7.3.2 Caching Policies

Caching at Anchor UAVs (A-UAVs): As mentioned before that the FD mechanism has the shortcoming in that it limits the number of accessible contents for all user communities to C_A, the A-UAV cache size. This limitation can be addressed by storing a part of the A-UAV's cache with same contents viz. duplicate contents and the remaining cache space with unique contents. The unique contents in all the A-UAVs are shared across the communities via the traveling F-UAVs. This Smart Cache Duplication (SCD) mechanism can effectively increase the access to the number of contents for all the users across the entire system, thus improving the overall availability within a given TAD.

Let the size of the duplicate segment of A-UAV cache be $(\lambda. C_A)$ and that of the unique segment be $((1-\lambda). C_A)$where λ is a duplication factor that decides the level of content duplication in A-UAVs. This results into $N_A. (1-\lambda). C_A$ unique contents stored across all N_A number of A-UAVs in the system, and these can be shared across all user communities via the mobile F-UAVs. These unique contents have popularities after the top $(\lambda. C_A)$ popular duplicated contents in all the A-UAVs. For symmetry, all $N_A. (1-\lambda). C_A$ unique contents are uniformly randomly distributed across N_A number of A-UAVs. The total number of contents in the system: $C_{sys} = \lambda. C_A + N_A. (1 - \lambda). C_A$.

Caching at Ferrying UAVs (F-UAVs): The purpose of the F-UAVs is to ferry around $N_A. (1 - \lambda). C_A$ unique contents stored in all N_A A-UAVs. In the presence of limited per-F-UAV caching space, C_F, its caching policy can be determined based on its trajectories, the value of λ, and the Zipf parameter defining the content popularity.

Consider a situation in which an F-UAV k is approaching towards the A-UAV i. Let U_i be the set of all unique contents in the entire system except the ones stored in A-UAV i. To maximize content availability for the users in A-UAV i's community, the F-UAV should carry as many low popularity contents from set U_i as its cache space permits. To enable such access, F-UAV k should carry C_F top popular contents from the set U_i while approaching A-UAV i. The size of the set U_i can be expressed as $|U_i| = (N_A - 1). (1 - \lambda). C_A$. In scenarios when $C_F \leq |U_i|$, the F-UAV should carry the C_F top popular contents as outlined above. Otherwise, the F-UAV will carry all $|U_i|$ unique contents, leaving part of the F-UAV cache (i.e., $C_F - |U_i|$) empty. This causes underutilization of F-UAV cache space due to large λ values, leading to heavy in-A-UAV duplications, thus storing few unique contents.

7.4 Content Request and Provisioning Model

Content requests are generated by the users in communities and sent to their respective local A-UAVs or to a visiting F-UAV, in that order or preference.

Content Popularity: Studies have shown [119] that content request patterns often follow a Zipf distribution in which a requested content's popularity is a geometric multiple of the next popular content. Popularity of contents is given as $p_\alpha(i) = (1/i)^\alpha / \sum_{k \in C} (1/k)^\alpha$. The parameter C

represents the total number of contents in the pool, and the Zipf parameter α determines the skewness of the distribution.

Content Requests: Poisson distributed request generation is the most prevalent way to capture user requests in practical network scenarios.

Tolerable Access Delay and Content Provisioning: For each generated request, a Tolerable Access Delay (*TAD*) [123], [124] is specified. *TAD* is a Quality-of-Service parameter that indicates the duration that a user is ready to wait before a requested content can be accessed. From the network's perspective, if the content is not available from the A-UAV or visiting F-UAVs within the specified *TAD*, it will have to be downloaded from a central server using the expensive vertical link on the A-UAVs. Therefore, to reduce that downloading cost, the contents cached in F-UAVs need to be more readily available within the A-UAV/F-UAV network.

7.5 Trajectory-aware Content Placement Planning for Ferring UAVs

7.5.1 Trajectory Sequence and Cycle

An F-UAV's trajectory is represented by the *sequence* of visited A-UAVs, and *hovering duration* at each A-UAV. Figure 3.1 shows F-AUVs *A* and *B* follow a *partitioned cycle* of A-UAV sequence *X, Y, Z* and *W*, whereas F-UAVs *C* and *D* follow a *global cycle*. Choice of sequence depends on the popularity of contents cached in A-UAVs.

The cycle time of an F-UAV trajectory is $T_{cycle} = N_A^C \times (T_{hover} + T_{transition}), where\ N_A^C$ is the number of A-UAVs in the F-UAV's sequence, T_{hover} is the hover duration at each A-UAV, and $T_{transition}$ is the transition time between two consecutive A-UAVs in the F-UAV's sequence. $T_{transition}$ depends on the F-UAV flying speed, intercommunity distance, wind speed/directions, and other environmental factors. T_{hover} should not be less than a minimum duration that is required for successful data transfer between UAVs and users. Minimum hover time is determined by (a) the data transfer rate; (b) the amount of data needs to be exchanged between F-UAV to/from A-UAV; (c) multipath fading; (d) shadowing; fading due to height of UAVs etc.

7.5.2 Joint Deployment of Ferrying UAVs (JDFU) Algorithm

The caching mechanism discussed so far are based on round-robin trajectories using which an F-UAV sequentially visits all the A-UAVs with equal hover durations at each A-UAV. In the presence of multiple F-UAVs, since the same trajectory is used by all the F-UAVs with equal spacing, the time gap between two consecutive visits of an F-UAV to a community (i.e., a A-UAV) is T_{cycle}/N_F. Let us consider a scenario where an F-UAV $'i'$ leaves an A-UAV $'j'$ and the next F-UAV $'i+1'$ reaches A-UAV $'j'$ after T_{cycle}/N_F duration. If a requested content from the F-UAVs has a Tolerable Access Delay $TAD > T_{cycle}/N_F$, then the content is served to the user with a minimum of $TAD - T_{cycle}/N_F$ extra time before exhausting the TAD. If this extra time is more than the time gap between two consecutive F-UAV visits (T_{cycle}/N_F), then this extra time can be leveraged by deploying multiple F-UAVs in groups flying together as explained below.

Here we introduce an F-UAV trajectory mechanism, termed as *Joint Deployment of Ferrying UAVs* (JDFU), to leverage this extra time. In this mechanism, multiple F-UAVs fly together while following the same trajectory at the same time. With N_F number of F-UAVs in a system, they can be deployed in groups of different sizes while employing JDFU. If they are deployed with N_F^G number of F-UAVs per group, then there will be N_F/N_F^G such groups viz. ($N_F^G \times N_F/N_F^G$) F-UAVs. This is called the *JDFU configuration*. It should be noted that the groups of N_F^G F-UAVs still visit the A-UAVs (at their corresponding communities) in a round-robin manner.

It has two major advantages. First, in case of high TAD requests, the extra time of $TAD - T_{cycle}/N_F$ is reduced where the F-UAV is not adding to the content availability. It is explained as follows. T_{cycle}/N_F is the time it takes an F-UAV to reach the next community in its sequence after the previous F-UAV leaves the community. If $TAD > T_{cycle}/N_F$, then the F-UAV reaches the community $TAD - T_{cycle}/N_F$ before it exhausts the TAD. Employing JDFU can leverage this duration to improve availability of contents at other communities. Second, employing JDFU increases the effective caching capacity of F-UAVs as compared to F-UAVs deployed without JDFU. This is explained as follows. If F-UAVs follow their respective trajectories (without JDFU), then they carry only the C_F most popular contents out of $(N_A - 1).(1 - \lambda).C_A$ from the cache of A-UAVs in their trajectories (refer Section 7.3.2). By employing JDFU, the F-UAVs can carry $N_F^G.C_F$ contents out of $(N_A - 1).(1 - \lambda).C_A$ cached in A-UAVs as opposed to C_F contents. Here N_F^G is the number of F-UAVs traversing in a group. Therefore, the effective caching capacity of the F-UAVs increases form C_F to $N_F^G.C_F$ which can significantly enhance the content availability.

However, this increase in effective cache size comes with the following downsides. First, the time interval during which the content availability depends only on the A-UAVs increases. The explanation is as follows. While employing JDFU, the equal spacing between group of F-UAVs depends on the number of groups. This means that the time taken by a group of F-UAVs $'k'$ to reach an A-UAV $'j'$ after the previous group of F-UAVs $'k-1'$ leaves the community will increase with increase in N_F^G. This is the time during which content availability depends only on A-UAVs. Second, to completely fill the increased effective cache of the F-UAVs, the *duplication factor* λ should be lowered. A lower λ ensures enough contents in Segment-2 of the A-UAVs to avoid underutilization of F-UAV cache (refer Section 7.3.2). While lower λ helps in increasing the effective cache utilization of the group of F-UAVs, it reduces the number of most popular contents in Segment-1 of the A-UAVs. However, if the TAD is sufficiently high, popular contents cached in F-UAVs due to low λ can be accessed before exhausting the TAD. Therefore, the reduction in λ will have minimum or no effect on content availability for high TAD.

The pseudo code to calculate N_F^G algorithmically and determine JDFU configuration is as follows.

Algorithm 7.1. JDFU Algorithm

```
Input: Total UAVs $N_A$, $N_F$, $TAD$ and $T_{transition}$
Output: JDFU configuration
Initialize $T_{hover}$, F-UAV trajectory to round-robin, $N_F^G = 1$
while True:
    compute $T_{cycle}$
    if $TAD > \frac{T_{cycle} \times N_F^G}{N_F} - T_{hover}$ then do
        Increment $N_F^G$
        while True:
            Decrement $\lambda$
            if $(N_A - 1).(1 - \lambda).C_A \geq N_F^G.C_F$ then do
                Cache $N_F^G.C_F$ in $N_F^G$ F-UAVs
                break
            end if
        end while
    else if $T_{hover} > T_{hover}^{min}$ then do
        Decrement $T_{hover}$
    end if
    compute JDFU configuration from $N_F^G$
```

Algorithm 7.1. (cont'd)

```
19. end while
```

The increase in content availability by employing JDFU largely depends on the TAD of the requested contents. When a content cached in F-UAVs is requested with very high TAD, it allows the F-UAVs to reach the request generating community with a maximum delay of TAD. This shows that the benefit of employing JDFU is directly proportional to the specified TAD in the content requests. Intercommunity distances also contribute to the increase in availability while employing JDFU algorithm in which closely located communities can be reached by F-UAV groups before the specified TAD. This phenomenon is elaborated later along with supporting experimental results.

7.6 Content Dissemination Performance and Experimental Results

Content availability is used as a metric to evaluate the performance of the proposed algorithm. It is defined as the probability of finding a requested content from the UAV-aided caching paradigm within the specified TAD. In the case of an F-UAV transitioning in round-robin manner across the A-UAVs in its trajectory, the F-UAV's accessibility within a given TAD is expressed as follows.

$$P_{FA} = \begin{cases} \dfrac{N_F \times (T_{Hover} + TAD)}{N_F^G . N_A \times (T_{Hover} + T_{transition})} & for\ TAD < \left(\dfrac{N_F^G . N_A}{N_F} - 1\right) T_{hover} + \dfrac{N_F^G . N_A}{N_F} T_{transition} \\ 1 & for\ TAD \geq \left(\dfrac{N_F^G . N_A}{N_F} - 1\right) T_{hover} + \dfrac{N_F^G . N_A}{N_F} T_{transition} \end{cases} \tag{7.1}$$

The relative difference between the TAD and the time taken by an F-UAV or a group of F-UAVs to reach an A-UAV after the previous group has left decides the accessibility of F-UAVs. Note that the physical accessibility to the F-UAV does not guarantee the access to the requested content since the F-UAV or the group of F-UAVs can store only a limited number (i.e., $N_F^G . C_F$) of unique contents. Let P_F be the probability that the requested content can be found within the F-UAV or group of F-UAVs. It can be expressed as:

$$P_F = \sum_{i=C_A+1}^{C_A+1+C_{EFF}} p_\alpha(i) \tag{7.2}$$

where, $p_\alpha(i)$ is the Zipf distributed popularity as defined in Section 7.4. The effective cache size of the F-UAV is given as: $C_{EFF} = min\{N_F^G \times C_F, (N_A - 1) \times (1 - \lambda) \times C_A\}$.

Now, let P_A be the probability that the requested content can be found within the A-UAV that is local to the community from which the content request was generated. This is expressed as:

$$P_A = \sum_{i=1}^{\lambda \times C_A + (1-\lambda) \times C_A} p_\alpha(i) \tag{7.3}$$

Combining those three probabilities above, the overall availability can be stated as:

$$P_{Avail} = P_A + P_{FA} \times P_F \tag{7.4}$$

To summarize, local contents from A-UAVs (i.e., both duplicate and unique) and unique contents from future visiting F-UAVs contribute towards the overall availability P_{Avail} within a specified TAD. Note that all unavailable contents within the specified *TAD* will have to be downloaded by the A-UAVs using their expensive vertical links such as the satellite Internet. Therefore, availability indirectly indicates the content download cost in the system.

Before exploring the impact of employing JDFU algorithm on content availability, it is important to understand the effects for hover and transition time with respect to tolerable access delay, which is discussed next. For experimentation, specific modules were added to implement request generation, UAV caching, and F-UAV movement strategies. For all experiments, $N_c = 2000$, $C_A = C_F = 100$, Poisson request rate $\mu = 1$ requests/second and Zipf parameter $\alpha = 0.7$.

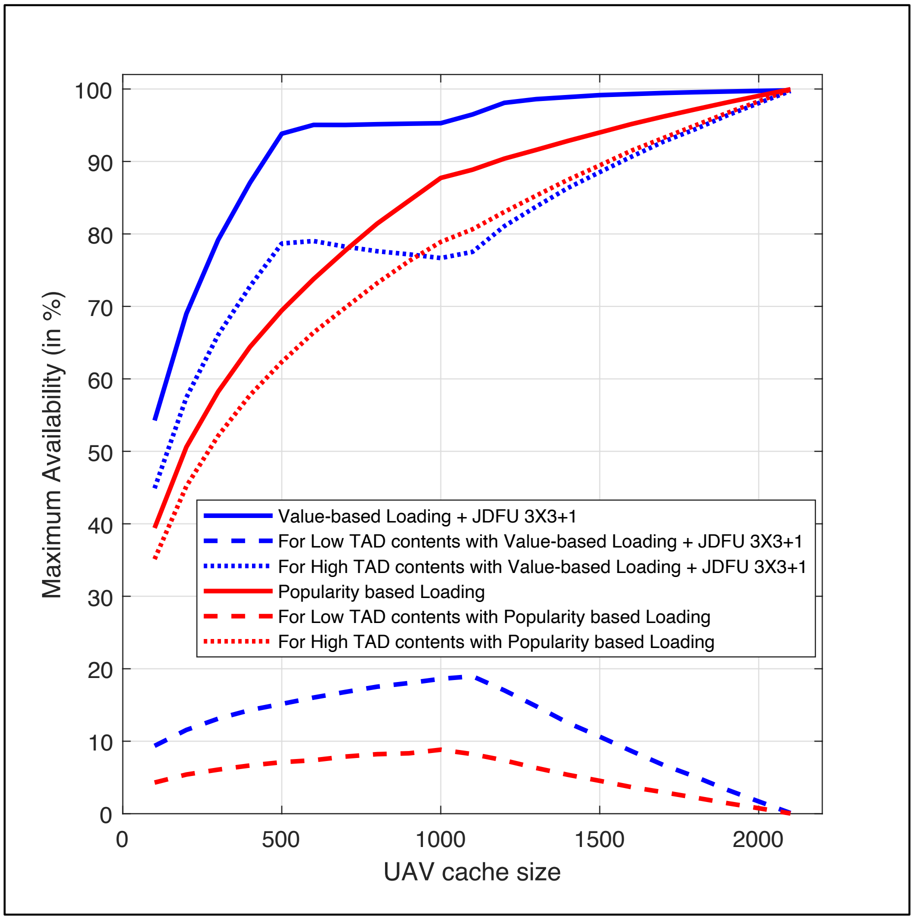


Figure 7.1. Improvement in maximum availability of contents by loading UAVs using value of contents and deploying F-UAVs in groups

7.6.1 Impacts of Value-Based Caching and JDFU

The overall increase in content availability using value-based content caching and joint-deployment of ferrying UAVs is shown in Figure 7.1. The performance improvement is compared against popularity-based caching policy at A-UAVs and round-robin trajectories of F-UAVs (without JDFU). Content availability is evaluated for varying cache size of the UAVs.

It can be seen from Figure 7.1 that a maximum increase in availability of approximately 25% can be achieved by using value-based caching policy at the A-UAVs, and JDFU for the F-UAVs. The benefits of value-based caching along with JDFU in scenarios with multi-dimensional demand heterogeneities are attributed to various factors including heterogeneity in popularity sequence, TAD associated with the content requests, popularity of low TAD contents, value of a content, and configuration of JDFU. The effects of these factors are depicted individually in the following sub-sections.

7.6.2 Effects of Hover Time and Tolerable Access Delay

F-UAV hover time and TAD have interdependent impact on the content availability. This is shown in Figure 7.2 with $N_A = 20$, $N_F = 10$ and $T_{transition} = 10\ seconds.$

The surface plot shown in Figure 7.2 is nonmonotonic with respect to content availability when hover time and tolerable access delay are varied. The most noticeable observation is the dichotomous behavior of availability with increase in hover time for low and high $TAD.$ For low TAD, availability increases with increase in hover time, whereas for high TAD, availability decreases with longer hover time. The explanation for such behavior is as follows. First, for $TAD < T_{transition}$, while traversing from present community i to next community j, an F-UAV doesn't contribute to availability for $T_{transition} - TAD$ duration (refer Figure 7.2). This means that some of the requests generated for contents cached in F-UAVs will be downloaded because of partial inaccessibility of F-UAV. Hence, hovering over a community is more beneficial for content availability even though it may be an unfair increase in average availability (availability increases only for the community where the F-UAV is hovering).

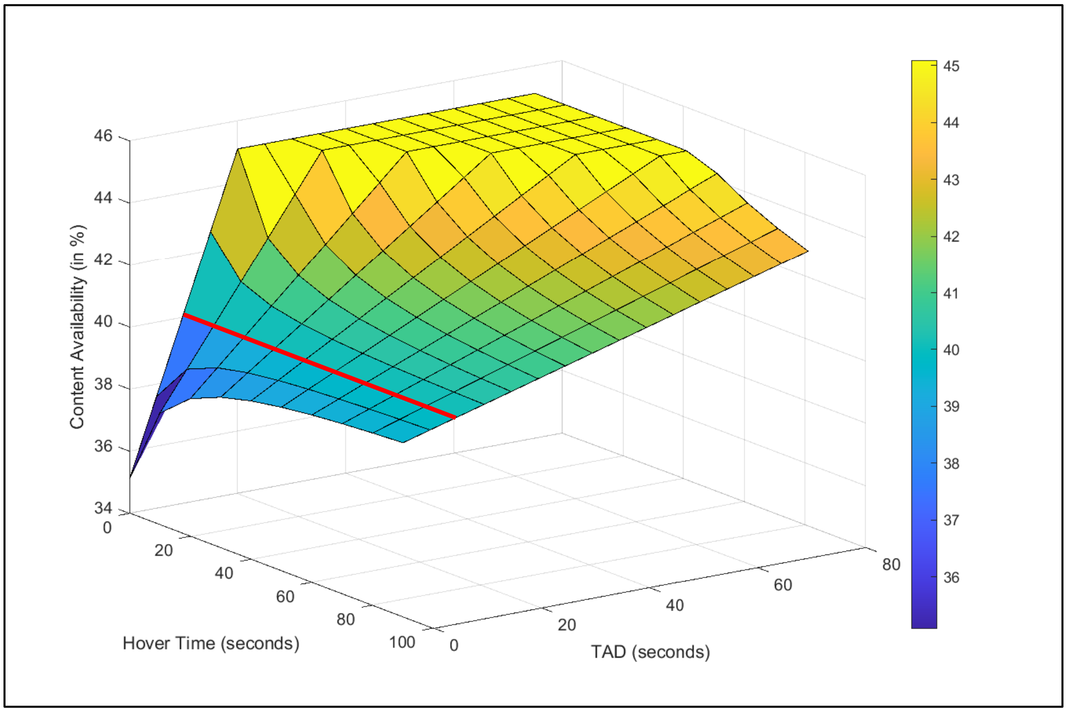


Figure 7.2. Content availability for different TAD with varying hover time

The region to the left of the red line in Figure 7.2 shows this effect. Second, for $TAD > T_{transition}$, increase in hovering time reduces the possibility of the condition $(TAD - T_{hover}) > T_{transition}$ to be true. In other words, the possibility of exhausting the given TAD before reaching next community increases. So, it is beneficial to hover less, which increases the accessibility of F-UAVs at future communities in the cycle before TAD expires. This behavior is capture in Figure 7.2 in the region right to the red line. Finally, for $TAD = T_{transition}$, an F-UAV can add to availability within TAD, at all times, irrespective of its hovering decision. If F-UAV decides to hover at the present community i, it caters to all the requests generated at i whereas transiting to next community j ensures accessibility of F-UAV at j since it reaches j within the TAD. The red line in Figure 7.2 shows this behavior. It should also be noted that this contradicting behavior is for a fixed transition time $T_{transition}$. With the same number of UAVs, effect of transition time on availability for varying tolerable access delay is discussed next.

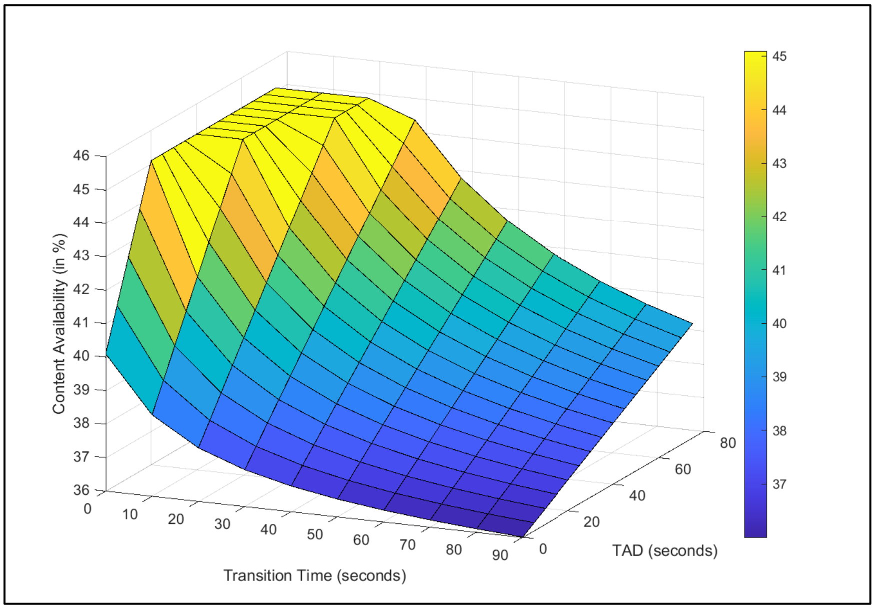


Figure 7.3. Content availability for different TAD with varying transition time

7.6.3 Effects of Transition Time and Tolerable Access Delay

The effect of $T_{transition}$ and TAD on content availability is shown in Figure 7.3 for $T_{hover} = 20\ seconds.$

There are two major observations. First, content availability reduces with increasing transition time. High transition time reduces accessibility of F-UAVs at future visiting communities, which leads to reduction in availability (refer Eqn. 7.1). Second, with increase in $TAD,$ content availability increases. This statement is intuitively supported since more TAD entails more time allowed for an F-UAV to reach the request generating user community. These observations can also be verified from the Eqn. 7.1 and Eqn. 7.4, where P_{FA} is directly proportional to TAD and inversely proportional to $T_{transition}$. Therefore, the maximum content availability occurs at least transition time $T_{transition}$ and highest user-specified TAD.

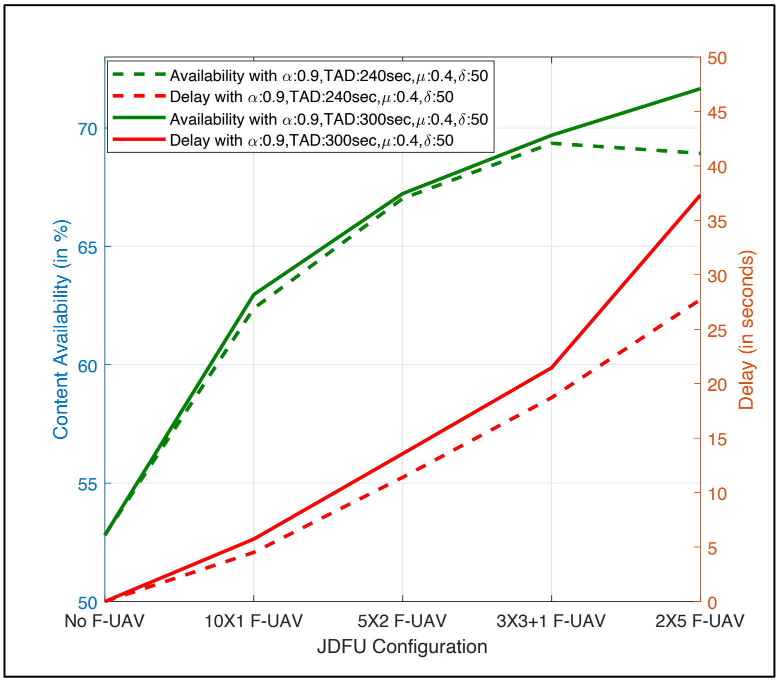


Figure 7.4. Availability and delay with JDFU for different TAD

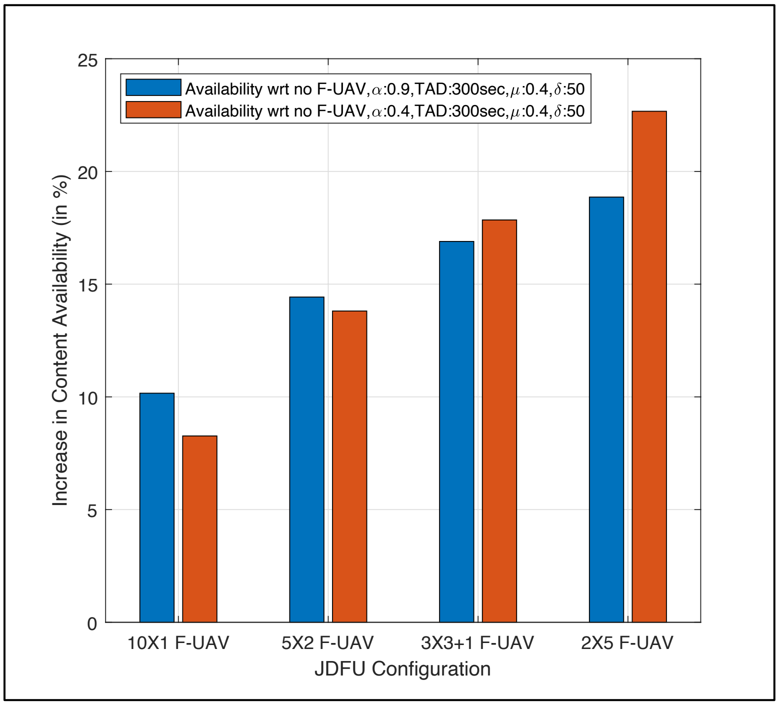


Figure 7.5. Increase in availability with JDFU for different α

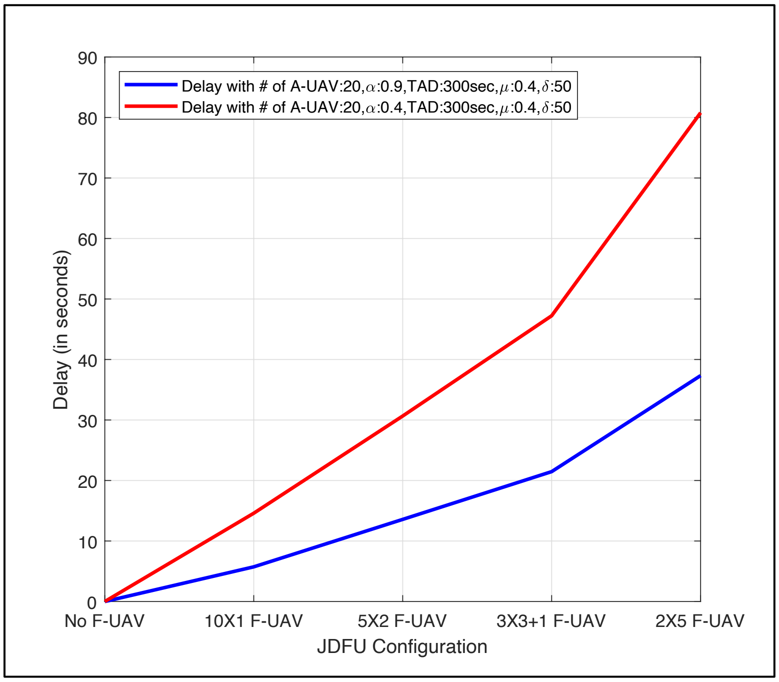


Figure 7.6. Delay with JDFU for different α

7.6.4 Effect of Joint Deployment of Ferrying UAVs (JDFU)

To show the benefits of JDFU, the first set of experiments are conducted with popularity parameter $\alpha = 0.9$, $TAD = 240\ and\ 300\ seconds$, swap probability $\mu = 0.4$, and swap difference $\delta = 50$. Rest of the parameters are as per Table 7.1.

Figure 7.4 shows the increase in availability with increase in F-UAV group size for the JDFU configuration (Section 7.5.2). The following inferences can be derived from the figure. First, with increase in group size, the availability increases. However, for $TAD = 240\ seconds$, the increase in availability is restricted for 2×5 configuration of JDFU with 5 F-UAVs in each group. This is because groups of F-UAVs do not reach the next community in their trajectory before the TAD expires. Second, with increase in TAD to 300 $seconds$, the benefit of JDFU is retained for 2×5 configuration of F-UAVs since the group of F-UAVs reach the next community in their trajectory before the TAD expires. Third, content access delay increases with increase in F-UAV group size for JDFU. Finally, with increase in TAD, delay increases proportionally.

Figure 7.5 and 7.6 show the effects of different popularity distribution on the increase in availability and delay while employing JDFU. Important observation from Figure 7.5 shows a comparison between the increase in availability for $\alpha = 0.9$ and 0.4 with varying configurations of JDFU. First, for smaller F-UAV group size in JDFU configuration, higher α ensures more increase in availability of contents. This is because for higher α, popular contents are more likely to be requested. Second, for larger F-UAV group size in JDFU configuration, lower α produces more availability. This is because less popular contents cached in F-UAVs are more likely to be requested, which is not in the case of high α. Figure 7.6 shows that, delay increases with low α due to the content requests being more distributed across all contents cached in A-UAVs and F-UAVs. This is an attribute of the Zipf distribution (refer Section 7.4) and JDFU configurations. When the F-UAV group sizes increase, the contents served by F-UAVs increase as well. Due to increase in time a group takes to reach a community, delay increases with the group size. Next, the benefits of JDFU are explored by increasing the caching capacity of UAVs.

7.6.5 Impacts of UAV cache Size on JDFU

This experiment discusses the effects of JDFU on availability, when caching capacity of UAVs is increased. For best results, value-based caching policy is followed at the A-UAVs. Parameters are set as follows; High $TAD = 240\,seconds$, Low $TAD = 5\,seconds$, $\gamma = 0.95$, JDFU configuration is $3 \times 3 + 1$ with 3 F-UAVs in each group and remaining parameters are according to default value in Table 7.1.

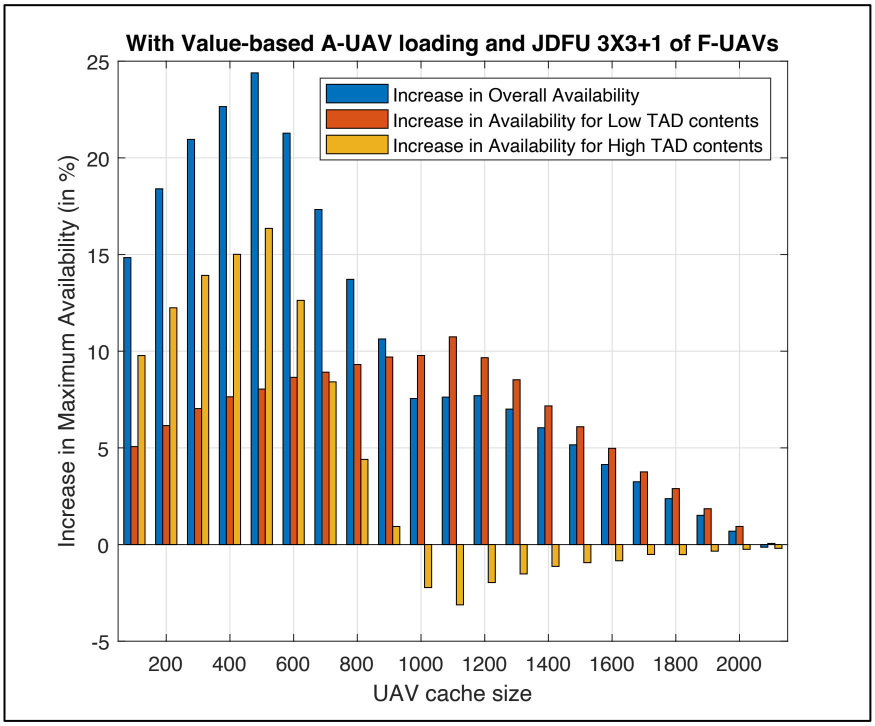


Figure 7.7. Increase in availability with value-based caching and JDFU as compared to popularity-based caching

Fig 7.7 shows the combined effects of JDFU and value-based caching policy towards increase in availability with respect to the popularity-based caching policy. The observations are as follow. First, increase in availability is maintained for all cache sizes until A-UAV's cache size is equal to total number of contents in the system. It can be observed in Figure 7.7 that beyond cache size of 2000, the benefits of JDFU ceases to exist since an A-UAV can cache all 2000 contents irrespective of any caching policy. Second, as opposed to the value-based caching policy favoring the availability of low TAD contents, with JDFU along with value-based caching, both low and high TAD contents have high availability. Third, availability of contents increases with increase in cache size till a certain cache size and tapers off beyond it. This can be explained as follow. For cache size of 500, 1 A-UAV and 3 F-UAVs store all contents in the content pool viz. 2000. Beyond this, any increase in cache size entails underutilization of the F-UAV cache space while employing

JDFU. To compensate for the underutilization of F-UAV cache space, the *storage segmentation factor* λ is reduced (refer Section 7.3), which reduces availability. Finally, high$-TAD$ contents do not contribute to the increase in availability beyond cache size of 1000. The reasons are twofold. One is the excessive reduction of λ to compensate for the space underutilization of F-UAVs, and the other is high popularity contents being replaced by very low popularity contents with low TAD, due to their increased value by employing value-based caching policy (see Eqn. 4.4).

7.6.6 Hovering Required to Maximize Content Availability with JDFU Algorithm

To explore the benefits and limits of JDFU algorithm, 10 F-UAVs are deployed in groups of $N_F^G = 2, 3 \; and \; 5$. For this experiment, $N_A = 20$ and $T_{transition} = 10 \; seconds$. Figure 7.8-7.11 shows the impact of JDFU for varying T_{hover} and TAD.

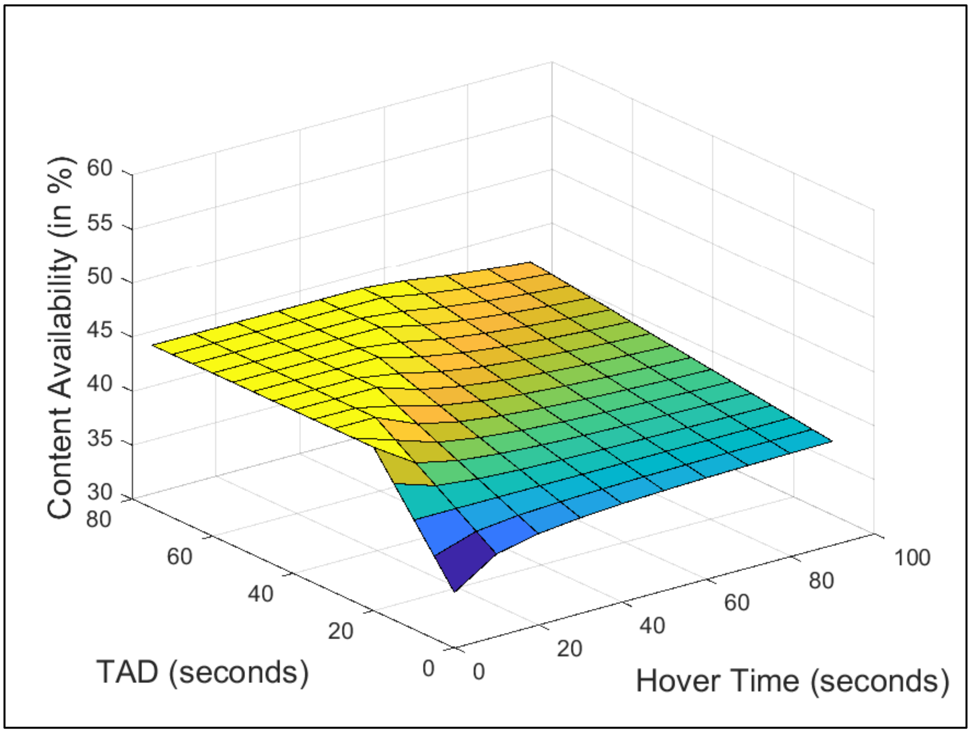


Figure 7.8. Content Availbility Without JDFU

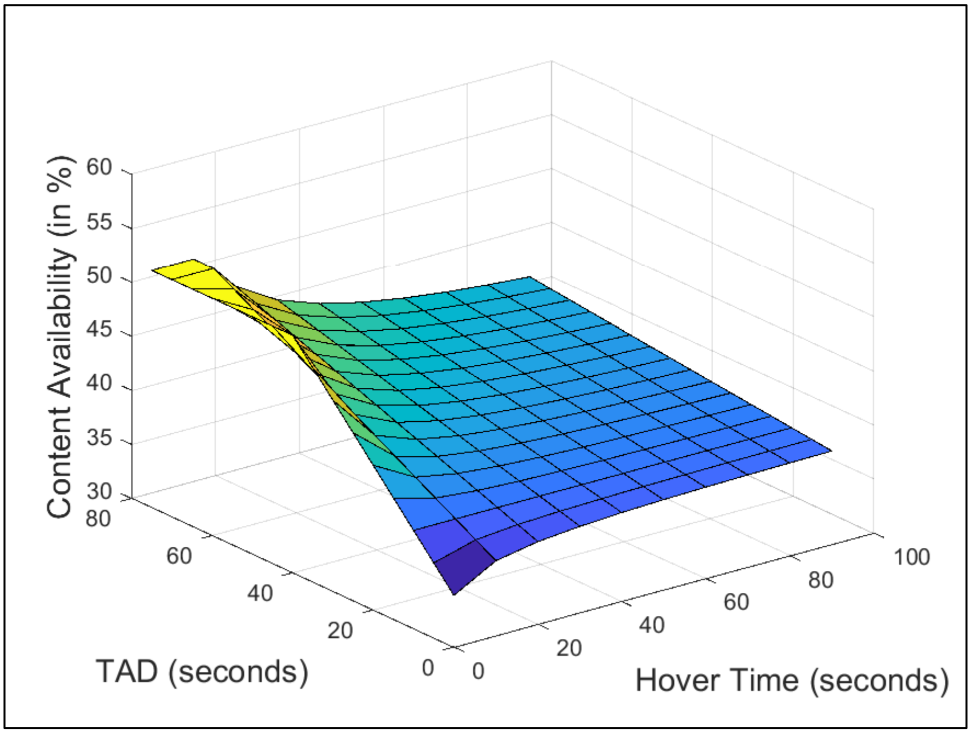


Figure 7.9. Content Availbility with JDFU configuration 5 × 2

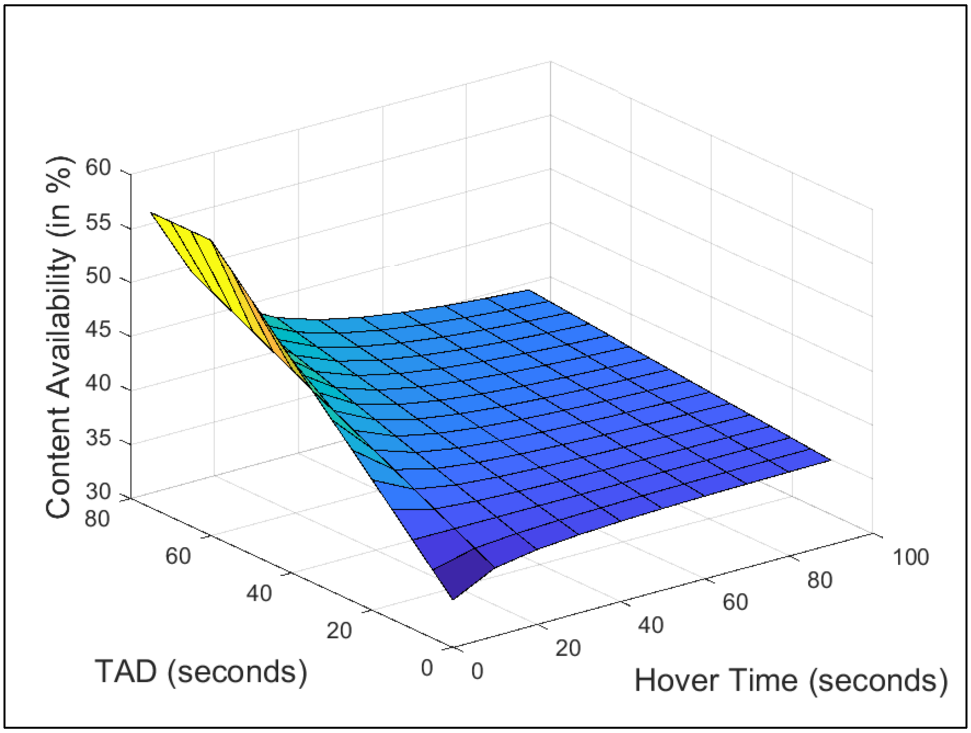


Figure 7.10. Content Availbility with JDFU configuration 3 × 3 + 1

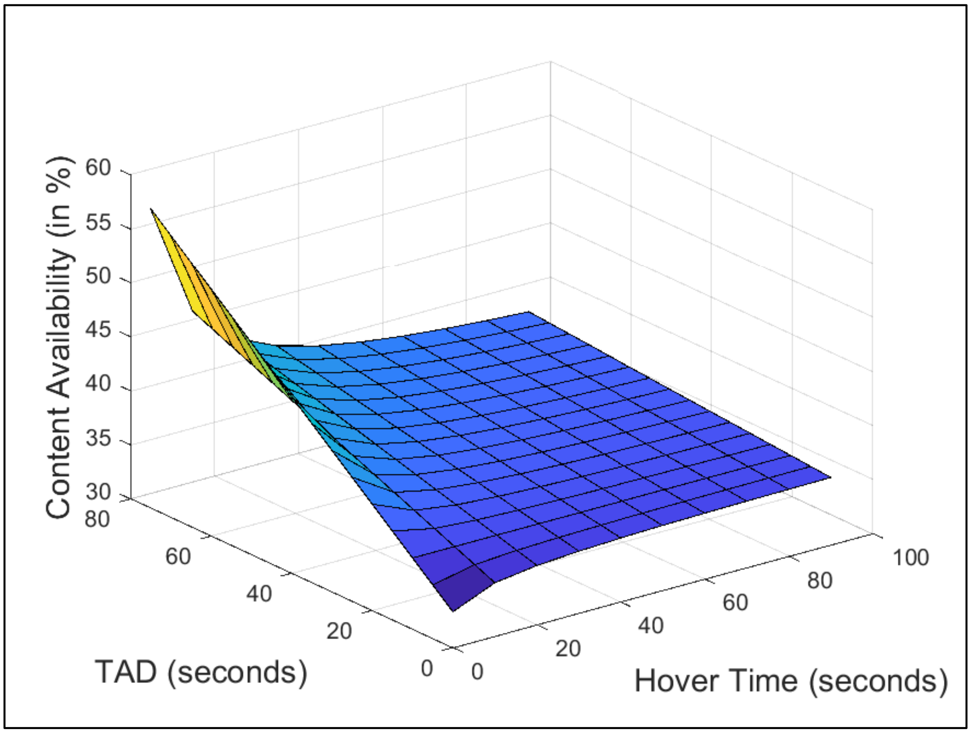


Figure 7.11. Content Availbility with JDFU configuration 2 × 5

The observations are as follows. First, maximum content availability attained using JDFU algorithm is with the configuration 2×5. Here, F-UAVs are deployed in groups of 5. Since there are a total of 10 F-UAVs, there are 2 such groups. To fill the cache space of 5 F-UAVs, 500 unique contents are required. With 20 A-UAVs in the system, λ (*duplication factor*) is set to 0.75 so that the unique content in the system is $C_A.(1-\lambda).N_A = 100.(1-0.75).20 = 500$. Therefore, F-UAVs ferry 500 contents as opposed to 100 contents without JDFU deployment, which increases content availability. Second, next two JDFU configurations, viz. $3 \times 3 + 1$ and 5×2, are functionally similar to 2×5 except that the groups are of 3 F-UAVs and 2 F-UAVs respectively. F-UAVs ferry 300 and 200 contents for their respective configurations, which is less than 500 contents with 2×5 configuration. This explains the reason for less availability with $3 \times 3 + 1$ and 5×2 JDFU configurations as compared to 2×5 configuration. The lowest maximum availability is attained for no JDFU configuration. Third, for high hover time T_{hover}, JDFU algorithm fails to provide more availability. This is due to the reduction in P_{FA} which reduces the content availability (see Eqn. 7.1-7.4).

7.6.7 JDFU with Different Inter-community Distances

For the experiments so far, transition time, which represents the intercommunity distance, is kept fixed at $T_{transition} = 10\ seconds$. Figure 7.12 discusses the impact of JDFU with the context of inter-community distances. It considers three scenarios, namely, communities located nearby, moderately apart, and far apart, and their effects on availability while applying the aforementioned proposed mechanisms.

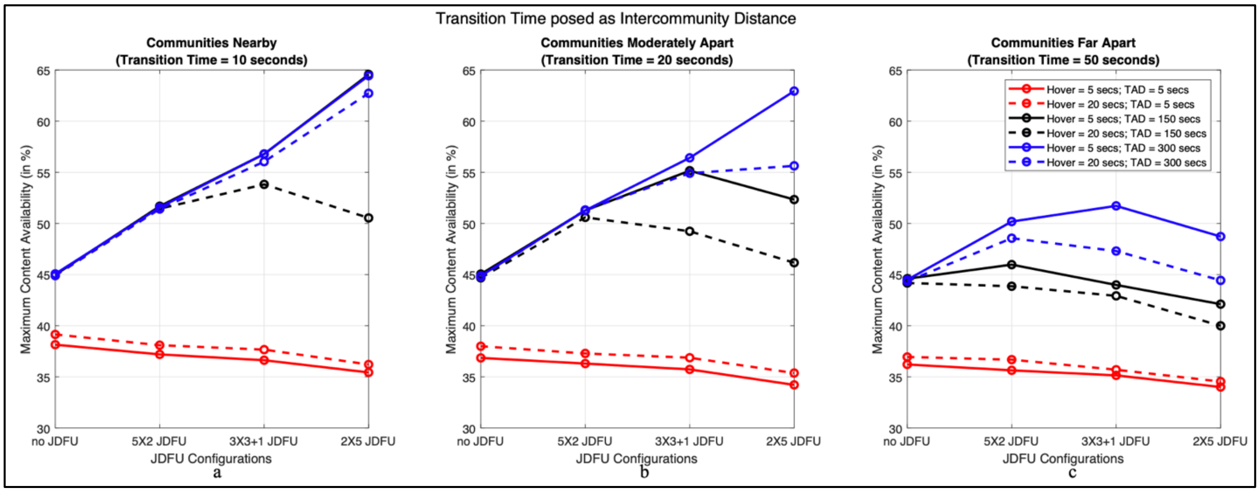


Figure 7.12. Benefits of JDFU for different intercommunity distances

In Figure 7.12, red, black, and blue lines represent low, moderate and high TAD values, respectively. Solid and dashed lines are used to depict less and more hovering durations respectively. The key observations from Figure 7.12 are as follows. First, employing JDFU boosts content availability for all combinations of inter-community distances and $TADs$ except for very low TAD values. This can be seen in across Figure 7.12a-c. Second, the benefits of JDFU diminishes with increase in inter-community distances. This can be observed in Figure 7.12b and 7.12c where group size of 3 F-UAVs adds to availability more for moderate inter-community separation as compared to far apart communities. Third, with increase in TAD, the benefit of JDFU is substantial. This can be seen in Figure 7.12c where group size of 3 F-UAVs produce more content availability with $TAD = 300$ seconds as compared to $TAD = 150$ seconds. Finally, more hovering is beneficial for all inter-community distance scenarios except for very low TAD values [92], [135] (Figure 7.12a-c).

All of these observations are attributed to accessibility of F-UAVs (refer Eqn. 7.1), which shows that increase in intercommunity distances decrease probability of accessibility (P_{FA}) whereas increase in TAD increases P_{FA}. Increase in Low availability period (LAP) can also be used as a measure to describe the reduction in accessibility of F-UAVs (Figure 4.5).

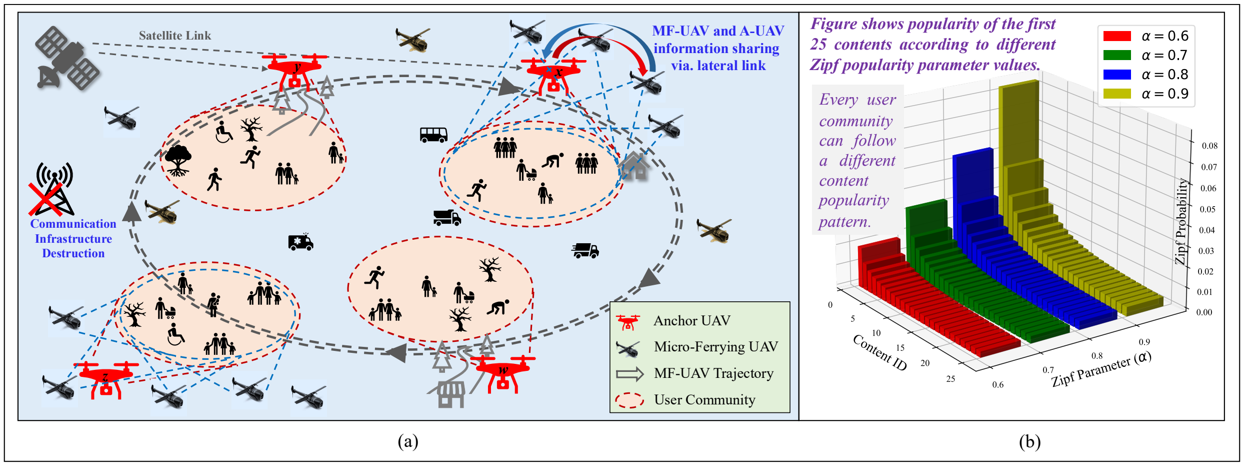
It should be noted that JDFU benefits the availability of requested contents irrespective of the caching policy employed at the UAVs. However, performances can be enhanced if the caching policy is well formulated like value-based caching. Although the experiments are conducted for round-robin trajectory, JDFU can be used to improved content availability while using other trajectories as well. Therefore, joint deployment of ferrying UAVs and value-based caching policy are generalized algorithmic solutions for the caching decision problems in communication-challenged environments.

7.7 Summary and Conclusion

The chapter explores trajectory characterization and planning in a UAV-aided networks for content dissemination in infrastructure-less systems. Cache-enabled UAVs serve communities of users in a disaster/war-stricken area by caching popular contents in order to reduce content downloading using satellites and other expensive vertical links. A framework is adopted in which two types of UAVs, namely anchor UAVs and ferrying UAVs, are deployed. Through analytical modeling and simulation experiments, the chapter establishes a trajectory design paradigm which considers the user-specified tolerable access delay and the nature of the disaster/war-stricken region. It is shown that content availability can be maximized by appropriately choosing the hover time of ferrying UAVs at each community. It also introduces a novel *Joint Deployment of Ferrying UAVs* (JDFU) algorithm which can leverage user-specified tolerable access delay associated with requested content and intercommunity distances to improve content availability by deploying ferrying UAVs in groups. The system has been functionally validated, and performance is evaluated for different scenarios including stochastic content request generation and various ferrying UAV trajectories. The next chapter on this topic will include incorporating runtime, dynamic and adaptive mechanisms to learn trajectory-aware caching policies befitting ferrying UAV trajectories. Such learning-driven caching policies can be developed on-the-fly for all those design components so that content popularities, optimal caching, and the best UAV trajectories can be learnt online in time-varying disaster regions.

Chapter 8: *Top-k* Multi-Armed Bandit Learning for Trajectory-Aware Caching in Swarms of Micro-UAVs

Continuing from the trajectory considerations discussed in the previous chapter, this chapter advances our understanding of how Micro-Unmanned Aerial Vehicles (Micro-UAVs) can be effectively utilized for content dissemination in environments devoid of standard communication infrastructures due to disasters or conflicts. We now focus on enhancing the adaptability of these UAV systems through trajectory-aware, adaptive caching strategies. These strategies are designed to dynamically respond to changing conditions and demands in disaster-stricken areas, leveraging the mobility and flexibility of Micro-UAVs.



Satellite Link
MF-UAV and A-UAV information sharing via. lateral link
Communication Infrastructure Destruction
Anchor UAV
Micro-Ferrying UAV
MF-UAV Trajectory
User Community
(a)
Figure shows popularity of the first 25 contents according to different Zipf popularity parameter values.
Every user community can follow a different content popularity pattern.
α = 0.6
α = 0.7
α = 0.8
α = 0.9
Zipf Probability
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
Content ID
0
5
10
15
20
25
Zipf Parameter (α)
0.6
0.7
0.8
0.9
(b)

Figure 8.1. (a) Coordinated UAV system for content caching and distribution in environments without communication infrastructure; (b) Zipf Popularity Distribution

8.1 Motivation

Effective communication during disasters is crucial for efficient relief operations and timely dissemination of vital information. Micro-UAVs present a promising solution to the disruption of traditional communication networks, capable of navigating and servicing isolated or inaccessible areas. However, the potential of these UAVs is not fully realized without addressing the challenges posed by their limited storage capacities and the dynamic nature of disaster environments. There is a compelling need for a content management system that not only understands the geographic and temporal aspects of content demand but also integrates the flight trajectories of the UAVs. Such a system would ensure that content delivery is both strategic and context-aware, maximizing the impact and utility of the UAVs deployed in these critical scenarios.

8.2 Design Objective

The primary objective of this chapter is to design a decentralized, trajectory-aware, adaptive content management system utilizing Micro-UAVs that optimizes content delivery to disaster-affected populations. The following design goals will guide the development of this system:

a) This chapter develops a trajectory-aware adaptive caching policy that not only responds to changes in content popularity and user demand but also incorporates UAV flight paths and operational constraints. This trajectory-aware approach ensures that caching decisions enhance the overall efficiency and content dissemination via cache-enabled UAVs.

b) Utilizing a *Top-k* Multi-Armed Bandit (MAB) learning approach, the system adapts to real-time changes in content popularity and user demand. This learning is informed by shared data across Micro-UAVs, optimizing content availability on each UAV.

c) Furthermore, this chapter implements a Selective Caching Algorithm to effectively manage the trade-off between Micro-UAV storage and their accessibility via minimization of content redundancy. By ensuring that only essential content is stored and disseminated, this mechanism reduces the storage burden and improves the responsiveness of the UAVs to critical needs. It focuses on the joint geographical deployment of Micro-UAVs to manage this trade-off, ensuring that UAVs are deployed in a manner that maximizes content reach while considering their regional accessibility.

d) It analyzes how adaptive caching decisions influenced by the learning algorithms affect quality of service, particularly in terms of the Tolerable Access Delay (TAD), which measures the urgency of different types of information and community expectations.

e) The proposed mechanism enables the system to modify caching decisions in real-time, based on immediate feedback from the environment and user interactions. Such real-time adaptability accommodates sudden changes in content demand and UAV operational conditions.

Through these objectives, this chapter aims to further develop the capabilities of Micro-UAVs in delivering critical information under challenging circumstances, ensuring that they operate not only as carriers of content but as smart, adaptive components of a larger disaster response strategy. This trajectory-aware caching model is intended to be robust yet flexible, capable of adapting to both the physical and informational landscapes of emergency scenarios.

8.3 System Model

8.3.1 UAV Hierarchy

As shown in Figure 8.1, a two-tiered UAV-assisted content dissemination system is deployed. Each community is served by a dedicated A-UAV that uses a lateral wireless connection (i.e., WiFi etc.) to communicate with users in that community. The system introduces a set of low-power-budget Micro-UAVs for the role of ferrying (MF-UAVs). These are unlike A-UAVs which operate with a much larger power budgets. MF-UAVs are mobile and possesses only lateral communication links such as Wi-Fi. Unlike the A-UAVs, the MF-UAVs do not possess expensive vertical communication interfaces such as satellite links etc. Effectively, the MF-UAVs act as content transfer agents across different user communities by selectively transferring content across the A-UAVs through their lateral links.

8.3.2 Content Demand and Provisioning Model

The content popularity distribution, quality of services and content provisioning are outlined below.

Content Popularity: Research has shown that user content request patterns often follow a Zipf distribution [91], [92], where the popularity of a content is proportional to the inverse of its rank, and is a geometric multiple of the next popular content. Popularity of content 'i' is given as:

$$p_{\alpha}(i) = \left(\frac{1}{i}\right)^{\alpha} \Big/ \sum_{k \in N} \left(\frac{1}{k}\right)^{\alpha} \tag{8.1}$$

The Zipf parameter, α, determines the distribution's skewness, while the total number of contents in the pool is represented by the parameter N. The inter-request time from a user follows the popular exponential distribution [91].

Tolerable Access Delay: For each requested content, the user specifies a Tolerable Access Delay (TAD) [70], which serves as a quality-of-service parameter and represents the amount of time the requesting user can wait before the content is downloaded.

Content Provisioning: Upon receiving a request from one of its community users, the relevant A-UAV first searches its local storage for the content. If the content is not found, the A-UAV waits for a potential future delivery by a traveling MF-UAV. If no MF-UAV arrives with the requested content within the specified *TAD*, the A-UAV then proceeds to download it through its vertical link. Since vertical links such as satellite links are expensive, smart caching strategies that can make the content accessible from the UAVs can be effective in reducing content provisioning costs.

8.4 Caching based on Content Pre-loading at A-UAVs

This section discusses caching policies based on content pre-loading at A-UAVs that assumes pre-assigned, static, and globally known content popularities. After understanding the limitations of these caching policies, the chapter designs a runtime, dynamic, and adaptive *Top-k* Multi-armed Bandit based caching mechanism, which is explained in a Section 8.5.

8.4.1 Pre-loading Policies at Anchor UAVs (A-UAVs)

The *Fully Duplicated* (*FD*) mechanism [91] is a naive approach that allows A-UAVs to download content from vertical links upon request by local users. FD has major limitations including content duplication, high vertical link download costs, and underutilization of UAV cache space. This means that with a cache size of C_A contents per UAV, the total caching capacity of the system is limited to C_A. *Smart Exclusive Caching* (*SEC*) [91], [92] overcomes those limitations of FD by storing a set number of unique contents in all A-UAVs and sharing them among communities via traveling MF-UAVs. Assuming globally known *homogeneous content popularity* across all user communities, the *SEC* mechanism divides the cache into two segments of size C_{S1} and C_{S2}. Segment-1 contains the top $C_{S1} = \lambda. C_A$ popular contents cached in all A-UAVs, while Segment-2 contains unique contents $C_{S2} = (1 - \lambda). C_A$, where λ is a *Storage Segmentation Factor*. This results into $C_{S2}^{total} = N_A. (1 - \lambda). C_A$ number of total Segment-2 contents stored across all N_A number of A-UAVs, and these can be shared across all user communities via the mobile MF-UAVs. This factor needs to be adjusted and fine-tuned based on various network, content, and demand conditions. Total number of contents in the system as per *SEC* is given as:

$$C_{sys} = \lambda. C_A + N_A. (1 - \lambda). C_A \tag{8.2}$$

Popularity-Based Caching (*PBC*) [93] is employed when different communities have different content preferences. Considering the *heterogeneous popularity* sequence of a community, the *PBC* approach, like SEC, divides the cache space of the local A-UAV into two segments of size C_{S1} and

C_{S2}. Segment-1 caches the most popular contents, which can be exclusive to a A-UAV (C_E) or non-exclusive i.e., may be cached across multiple A-UAVs (C_{NE}), such that, $C_{S1} = C_E + C_{NE}$. To be noted that according to the exclusivity of contents in C_{S1}, the total number of exclusive contents across all A-UAVs is termed as C_E^{total}. Segment-2 is the same as that in *SEC*. Therefore, by modifying Eqn. 8.2, the total number of contents in the system can be expressed as:

$$C_{sys} = C_{NE} + C_E^{total} + N_A.(1-\lambda).C_A \Rightarrow C_{sys} \geq \lambda.C_A + N_A.(1-\lambda).C_A \tag{8.3}$$

Value-Based Caching (*VBC*) [93] further enhances the caching policy by storing top-valued contents in Segment-1 of the A-UAVs, where *value* of contents comprises of their popularity and tolerable access delay. Value of a content 'i' is calculated as:

$$V(i) = \kappa v^* \times \frac{p_\alpha(i)}{TAD(i)} \Rightarrow V(i) = \kappa \times \frac{TAD_{min}}{p_\alpha(1)} \times \frac{p_\alpha(i)}{TAD(i)} \tag{8.4}$$

In this equation, $p_\alpha(i)$ represents the content's popularity as per the Zipf distribution, $TAD(i)$ is the content's tolerable access delay, κ is a scalar weight that increases as popularity decreases, and v^* is a normalization constant. The normalization constant is calculated for a given Zipf (popularity) parameter α using the minimum possible TAD (TAD_{min}) and the maximum possible popularity, which is $p_\alpha(1)$, i.e., $v^* = TAD_{min}/p_\alpha(1)$. The value of $V(i)$ is bounded between [0,1], and it increases as $p_\alpha(i)$ increases and $TAD(i)$ decreases. The content's value presents a holistic quantifiable measure for caching decision.

The caching policy for micro-ferrying UAVs remains the same for all the above-discussed caching policies for A-UAVs, which will be discussed in the forthcoming Section 8.5. An MF-UAV ferries content across the A-UAVs it visits along its trajectory. The caching policy of A-UAVs determines the utility of MF-UAVs where every A-UAV should maintain sufficient contents in its cache space to maximize the MF-UAV cache space utilization.

8.4.2 Limitations of Cache Pre-loading at A-UAVs

The caching policies discussed in this section rely on pre-loading content into A-UAVs, which has certain limitations. These approaches assume *a priori* knowledge of the popularity distribution of all the content in the system, which can hinder practical feasibility during deployment. Local popularity estimation of requested content within individual A-UAVs can partially alleviate this issue, but it cannot adjust the crucial storage segmentation factor (λ) (see Section 8.4.1) for maximizing availability across the entire system of A-UAVs and their communities. Collaborative global popularity estimation can be introduced, but it fails to capture locally meaningful demand heterogeneity across different communities.

8.5 Decentralized Caching with Multi-Armed Bandit

This section presents a plausible solution for the aforementioned shortcomings by using *Top-k Multi-Armed Bandit* learning for caching decisions at the A-UAVs. This facilitates faster learning and is adaptive to heterogeneous user demand patterns through information sharing via micro-UAVs. Based on the forthcoming mechanism, the caching policy for micro-ferrying UAVs is also modified to leverage their ubiquity, which is discussed later.

8.5.1 *Top-k* Multi-Armed Bandit Learning

Multi-Armed Bandit is a classic problem in reinforcement learning [130] and decision-making. At each round t, an agent chooses an arm A_t out of N arms, denoted by $A_1, A_2, \ldots, A_N$, and observes a reward R_t. Each arm i has an unknown reward distribution with mean μ_i and variance σ_i^2. The agent's goal is to maximize the total expected reward R_T over T rounds, where

T is the total number of rounds (time horizon):

$$R_T = max \sum_{t=1}^{T} E[R_t] \tag{8.5}$$

This thesis uses a variant of MAB called *Top-k Multi-Armed Bandit* [128]. Here, the agent has to choose k arms simultaneously out of a larger set of N arms, and it receives a reward for each arm in the chosen set. This is in contrast to choosing only one arm in classical MAB approaches. The goal of the agent is to maximize the total cumulative reward R_T obtained over a finite time horizon T:

$$R_T = max \sum_{t=1}^{T} \sum_{i=1}^{k} E[R_{t,i}] \tag{8.6}$$

8.5.2 Caching at A-UAV using *Top-k* Multi-Armed Bandit

In the scenario of UAV-caching, there is a *Top-k* MAB agent in each A-UAV. Here, choosing each content for caching corresponds to choosing an arm. The 'k' of *Top-k* MAB agent corresponds to the caching capacity of A-UAV, i.e., $k = C_A$. The agent's aim is to select 'C_A' contents out of the total pool of 'N' contents to be cached in an A-UAV such that the content availability to the users can be maximized. Here, the UAV-aided content dissemination system is the learning environment where the A-UAVs interact through their actions of choosing specific sets of contents to be cached. The feedback from the environment for the taken actions are in the form of rewards/penalties. Micro-ferrying UAVs play a crucial role in transferring information across the UAV-aided system, which helps in the computation of appropriate rewards/penalties, as shown in Figure 8.2. Actions are rewarded when cached contents are requested by the users and are served to the users within the given tolerable access delay or penalized otherwise. The top C_A contents that accumulate most reward from the corresponding community and other communities are chosen to be cached at a A-UAV. It should be noted that the *Top-k* MAB agents in the A-UAVs are provided with no *a priori* information about the content popularity at the corresponding user communities.

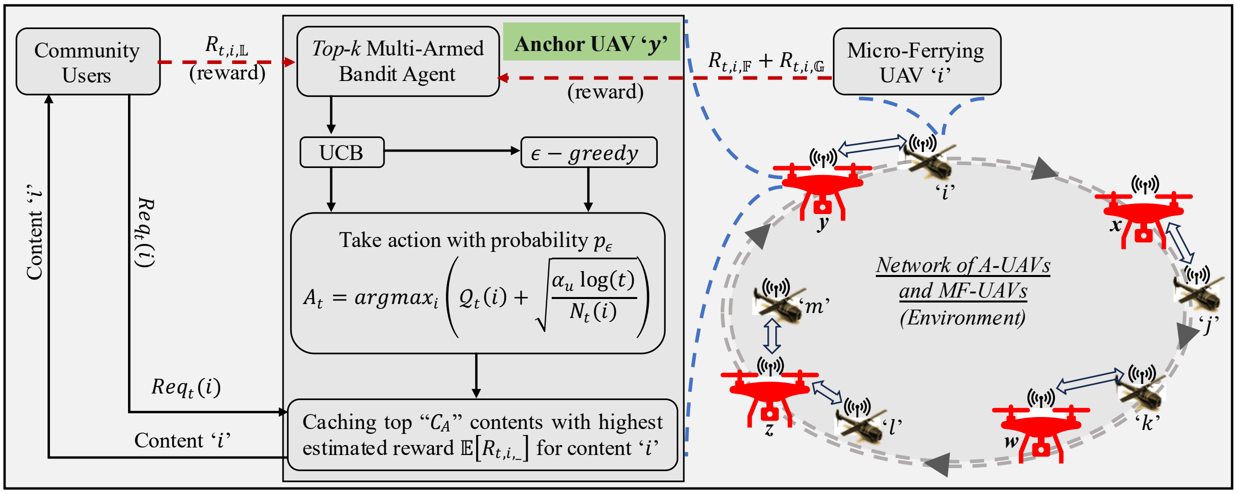

Community Users
$R_{t,i,\mathbb{L}}$
(reward)
Top-k Multi-Armed Bandit Agent
Anchor UAV 'y'
$R_{t,i,\mathbb{F}} + R_{t,i,\mathbb{G}}$
Micro-Ferrying UAV 'i'
(reward)
UCB
$\epsilon - greedy$
Take action with probability p_ϵ
$A_t = argmax_i \left(Q_t(i) + \sqrt{\frac{\alpha_u \log(t)}{N_t(i)}} \right)$
Content 'i'
$Req_t(i)$
$Req_t(i)$
Content 'i'
Caching top "C_A" contents with highest estimated reward $\mathbb{E}[R_{t,i,\_}]$ for content 'i'
Network of A-UAVs and MF-UAVs (Environment)
'i'
y
x
'm'
'j'
z
'l'
w
'k'

Figure 8.2. *Top-k* Multi-Armed Bandit Learning for Caching Policy at A-UAVs

A good choice for learning decision epoch in each *Top-k* MAB agent is according to the MF-UAVs accessibility at the corresponding community (i.e., an MF-UAV's visiting frequency). This is because the MF-UAVs carry the content availability information from the communities in its trajectory. Such information is leveraged for learning at the A-UAVs' *Top-k* MAB agents using appropriately designed multi-dimensional rewards. The agent learns to cache contents via the multi-dimensional reward structure which has three parts, namely, *local*, *ferrying,* and *global* rewards. Let $\mathbb{L}$, $\mathbb{F}$ and $\mathbb{G}$ denote the sets of locally requested contents, contents requested at other communities, and contents requested across all communities, respectively. These contents can be served to the users directly by a A-UAV or indirectly via the visiting MF-UAVs. If a cached content is served to a user within the given *TAD* and an increase in content availability is observed, the content is rewarded. The type of reward is determined by the set to which the cached content belongs. The expressions for three types of rewards are given as follows:

$$R_{i,\mathbb{L}} = \mathbb{I}_1(i \in \mathbb{L}, \delta_{\mathbb{L}} \geq 0) + \mathbb{I}_{-1}(i \notin \mathbb{L}, \delta_{\mathbb{L}} < 0) \tag{8.7}$$

$$R_{i,\mathbb{F}} = \frac{1}{N_A - 1} \sum_{j=1, j \neq \mathbb{X}}^{N_A} \mathbb{I}_1(i \in \mathbb{F}, \delta_{\mathbb{F}} \geq 0) + \frac{1}{N_A - 1} \sum_{j=1, j \neq \mathbb{X}}^{N_A} \mathbb{I}_{-1}(i \notin \mathbb{F}, \delta_{\mathbb{F}} < 0) \tag{8.8}$$

$$R_{i,\mathbb{G}} = \frac{1}{N_A} \sum_{j=1}^{N_A} \mathbb{I}_1(i \in \mathbb{G}, \delta_{\mathbb{G}} \geq 0) + \frac{1}{N_A} \sum_{j=1}^{N_A} \mathbb{I}_{-1}(i \notin \mathbb{G}, \delta_{\mathbb{G}} < 0) \tag{8.9}$$

$$where, \mathbb{I}_1(A) = \begin{cases} 1, & if\ A\ is\ true \\ 0, & otherwise \end{cases}$$

The above equations compute the reward according to increase in availability due to content 'i' cached at A-UAV '$\mathbb{X}$'. Here, $R_{i,\mathbb{L}}$, $R_{i,\mathbb{F}}$, and $R_{i,\mathbb{G}}$ are local, ferrying, and global rewards respectively. The terms $\delta_{\mathbb{L}}$, $\delta_{\mathbb{F}}$ and $\delta_{\mathbb{G}}$ correspond to the increase in local availability, ferried content availability and global availability respectively. Each type of reward is contingent upon the condition in the indicator function $\mathbb{I}_{1/-1}(i)$. The first terms in Eqns. 8.7, 8.8 and 8.9 represent the reward accumulated by caching content 'i' at A-UAV '$\mathbb{X}$', whereas the second term is the penalty associated with adverse condition. To be noted that $R_{i,\mathbb{F}}$, and $R_{i,\mathbb{G}}$ are higher if the content 'i' is requested and served at more communities.

Learning is achieved using a tabular method where a Q-table is maintained for all contents in the A-UAVs. The value corresponding to each content is called a Q-value or action-value [136]. The agent updates the Q-value for a content at every learning epoch according to the multi-dimensional rewards in Eqns. 8.7-8.9 from the interaction with the environment (UAV-aided content dissemination system) and learns the best actions (contents cached). The recursive expression which explains Q-value update for a content 'i' at A-UAV '$\mathbb{X}$' is given as follows:

$$\mathcal{Q}_{t+1}(i) = (1 - \alpha)\mathcal{Q}_t(i) + \alpha \left(R_{t,i,\mathbb{L}} + \mathbb{I}_1(\delta) \left(R_{t,i,\mathbb{F}} + R_{t,i,\mathbb{G}} \right) \right) \tag{8.10}$$

Here, $\mathcal{Q}_t(i)$ represents the Q-value of a content 'i' at t^{th} epoch; $R_{t,i,\_}$ is the respective reward received by caching content 'i'; δ represents the condition for the indicator function $\mathbb{I}_1(\mu)$ which is 1 if micro-ferrying UAVs are present in the communication range of A-UAV '$\mathbb{X}$' or 0 otherwise; α is a hyper-parameter which controls the learning rate. The Q-values for all contents are initialized with zero to ensure no *a priori* information for a *Top-k* MAB agent. Also, it ensures equal importance to all contents for caching decisions. As learning progresses, Q-values improve and best contents with highest Q-values are cached with the aim of maximizing accumulated reward which improves the caching policy and thus increases content availability.

Note that there can be very large number, i.e., $\binom{N}{k}$, of combinations of contents to be sampled by the *Top-k* MAB agent for caching. Consequently, the reward estimation for each individual content combination occurs infrequently, only after large intervals. This can lead to a weak estimates of reward distribution, as the global content population size N increases. This issue is handled by empirically selecting ϵ and its decay rate in the ϵ-greedy action selection policy [137]. To reduce the dependence of a caching policy on the choice of ϵ, an Upper Confidence Bound (UCB) strategy is used [137]. The *Top-k* MAB agent maintains an upper confidence bound on the expected reward of each content, and selects the set of C_A contents with the highest UCB at each epoch.

$$\mathcal{U}_t(i) = \mathcal{Q}_t(i) + \sqrt{\frac{\alpha_u \log(t)}{N_t(i)}} \tag{8.11}$$

Here, $\mathcal{U}_t(i)$ is the UCB of content 'i' at epoch 't'; $\mathcal{Q}_t(i)$ is the updated Q-value at epoch 't'; α_u is a hyperparameter that controls the degree of exploration; $N_t(i)$ is the number of time content 'i' has been requested till epoch 't'. The first term represents the reward estimate, and the second term depicts the uncertainty in reward estimate. UCB selects the content that has high potential for high reward but hasn't been requested frequently. This promotes exploration without externally inducing an exploration parameter such as ϵ. For this chapter, $\mathcal{U}_t(i)$ is used in place of $Q_t(i)$ to cache content 'i', as shown in Step 7-14 in Algorithm 8.1.

The following pseudo code explains the caching policy at a micro-ferrying UAV with a *Top-k* MAB agent.

Algorithm 8.1 Caching policy at a A-UAV with *Top-k* MAB Learning

```
1.  Initialization:
        a. N: Total contents in the system
        b. $C_A$: Caching capacity of an A-UAV
        c. $\mathcal{U}$: Size $|C_A|$ initialized with 0's (Q-table with UCB)
        d. $\alpha$: Learning rate for Q-table update
        e. $\alpha_u$: Degree of exploration (in UCB)
2.  Load A-UAV's cache with $C_A$ randomly chosen contents.
3.  while True:
4.     Check for learning epoch at A-UAV i.e., at $t^{th}$ epoch
5.     if True then do
6.        for $i = 0$ $to$ $length$(A-UAV cache size $C_A$) do
7.           Get reward $R_{t,i,\_}$ \\ according to Eqns. 8.7-8.9
8.           Update $\mathcal{U}(i)$      \\ from Eqns. 8.10 and 8.11
9.        end for
10.       $value = \boldsymbol{copy}(\mathcal{U})$ \\ make a copy of UCB values
          \\ Reload contents (Select arms)
11.       for $i = 0$ $to$ $length$(A-UAV cache size $C_A$) do
```

Algorithm 8.1. (cont'd)

```
            $c_{max} = \boldsymbol{argmax}(value)$
            **Load** $c_{max}$ to A-UAV
            **Set** $value[c_{max}] = -\infty$
        **end for**
    **end if**
**end while**
```

8.5.3 Proof of convergence

Within a finite time horizon, the *Top-k* MAB agent at a A-UAV converges to a caching policy which approaches the benchmark caching policy asymptotically. The proof of convergence lies in the intrinsic regret minimizing characteristics of MAB [138], which is shown below.

$$C_A = \{i | i \in N, 1 \le i \le k\} = \underset{k}{\operatorname{argmin}}\big(Regret(T)\big)$$

$$= \underset{k}{\operatorname{argmin}}\left(\sum_{t=1}^{T}\left(\max_{k} \sum_{i=1}^{k} R_{t,i^*} - \sum_{i=1}^{k} R_{t,i}\right)\right) \quad (8.12)$$

where, T is the total number of epochs (time horizon); k is the number of contents cached at each epoch; i^* represents the optimal caching action; i is the caching action selected by the *Top-k* MAB agent at t^{th} epoch. Eqn. 8.12 shows the difference between the reward obtained by the algorithm and the reward obtained by caching with benchmark policy. Post-convergence, the instantaneous regret should be minimum, which is experimentally proven in this chapter. Ideally for a perfectly designed reward structure the regret should asymptotically vanishes, i.e., $\lim_{T \to \infty} \frac{Regret(T)}{T} = 0$ [129].

The convergence of estimated rewards (Q-values) to the true values (expected reward) in a MAB setup, including *Top-k* MAB scenarios, can be analyzed using the Law of Large Numbers (*LLN*) [140] and concepts of stochastic approximation. For simplicity, this work initially considers the proof for a single arm and then extend the idea to all 'k' arms in the *Top-k* selection. According to weak law of large numbers [141], the estimated value of a content 'i' will be at a minute offset 'ϵ_i' from its true value, which is shown in the following expression:

$$\left|\lim_{T\to\infty} Q_{t+1}(i)\right| - \mu_i^* < \epsilon_i \Rightarrow \left|\lim_{T\to\infty}\frac{1}{n}\sum_{t=1}^{n}\left[R_{t,i,\mathbb{L}} + \mathbb{I}_1(\delta)\left(R_{t,i,\mathbb{F}} + R_{t,i,\mathbb{G}}\right)\right]\right| - \mu_i^* < \epsilon_i \quad (8.13)$$

Here, a single content/arm 'i' has a true value of μ_i^* , and $Q_{t+1}(i)$ represent the estimated reward (Q-value) of content 'i' after it has been selected 'n' times. The reward is taken from the second term (weighted reward) of Eqn. 8.10. For convergence, the weight 'α' is chosen empirically in such a way that it satisfies the *Robbins-Monro stochastic approximation condition* [139] for non-constant 'α', namely, $\sum_n \alpha_n(i) = \infty$ and $\sum_n \alpha_n(i)^2 < \infty$. To be noted that the weight 'α' is manifestation of '$1/n$' in Eqn. 8.13. Now, extending the concept to all top 'k' contents, Eqn. 8.13 can be modified using Eqn. 8.6:

$$\left|\lim_{T\to\infty}\frac{1}{n}\sum_{t=1}^{n}\left[\sum_{i=1}^{k} Q_{t+1}(i)\right]\right| - \sum_{i=1}^{k}\mu_i^* < \sum_{i=1}^{k}\epsilon_i$$

$$\Rightarrow \left|\lim_{T\to\infty}\frac{1}{n}\sum_{t=1}^{n}\left[\sum_{i=1}^{k}\left[R_{t,i,\mathbb{L}} + \mathbb{I}_1(\delta)\left(R_{t,i,\mathbb{F}} + R_{t,i,\mathbb{G}}\right)\right]\right]\right| - \sum_{i=1}^{k}\mu_i^* < \sum_{i=1}^{k}\epsilon_i \quad (8.14)$$

The convergence proof for each of the top 'k' contents individually follow the same logic as for the single content, provided each content is sampled infinitely often. Each content, including the top 'k' contents, must be selected infinitely often as the number of total selections $T \to \infty$. This requirement is met in practice by exploration strategies (like ϵ-greedy/UCB) that ensure all arms are explored sufficiently over time.

With an assumption on the success of the *Top-k* MAB based caching policy, let's say that the ideal sequence of contents are cached at A-UAVs, which is $C_A = \{i^* | i^* \in N, 1 \leq i^* \leq k\}$. For this caching decision, $\sum_{i=1}^{k} \epsilon_i = 0$, according to the expression given in Eqn. 8.14. Therefore, the instantaneous regret post-convergence can be derived from Eqn. 8.12 and 8.14, as follows:

$$\max_{k} \sum_{i=1}^{k} \left[R_{t,i^*,\mathbb{L}} + \mathbb{I}_1(\delta)\left(R_{t,i^*,\mathbb{F}} + R_{t,i^*,\mathbb{G}} \right) \right] - \sum_{i=1}^{k} \left[R_{t,i,\mathbb{L}} + \mathbb{I}_1(\delta)\left(R_{t,i,\mathbb{F}} + R_{t,i,\mathbb{G}} \right) \right] \approx 0 \quad (8.15)$$

The evidence of convergence, supporting the above expression is shown in Figure 8.7, where near-optimal contents cached at A-UAVs leads to $\sum_{i=1}^{k} \epsilon_i \approx 0$. According to the learnt caching policy, the cached contents can boost content availability at their respective communities as well as at other distant communities via MF-UAVs.

8.5.4 Selective Caching at Micro-Ferrying UAVs (MF-UAVs)

Ideally, the purpose of the MF-UAVs is to ferry around a subset of $C_E^{total} + N_A.(1-\lambda).C_A$ number of contents stored across N_A number of A-UAVs (see Section 8.4). Due to the limitation of per-MF-UAV caching space (i.e., C_{MF}), its caching policy should be determined based on its trajectories, learnt caching policy at A-UAVs, content request patterns, and the $TADs$ associated with the contents to be cached.

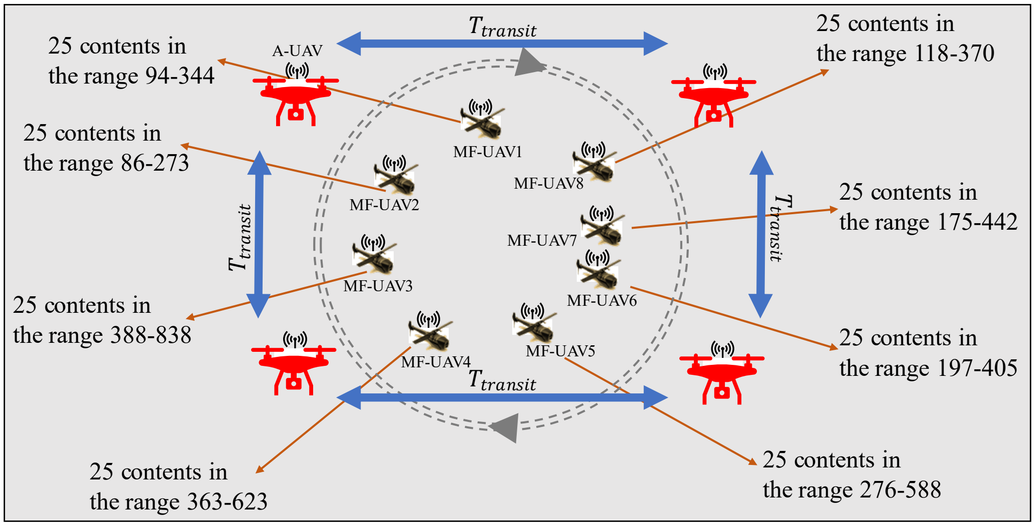

25 contents in the range 94-344
A-UAV
$T_{transit}$
25 contents in the range 118-370
MF-UAV1
25 contents in the range 86-273
MF-UAV8
MF-UAV2
$T_{transit}$
$T_{transit}$
25 contents in the range 175-442
MF-UAV7
MF-UAV3
MF-UAV6
25 contents in the range 388-838
MF-UAV5
MF-UAV4
25 contents in the range 197-405
$T_{transit}$
25 contents in the range 276-588
25 contents in the range 363-623

Figure 8.3. Algorithmic selection of cached contents at MF-UAVs in conjunction with *Top-k* Multi-Armed Bandit learning at A-UAV

MF-UAV caching policy is explained in the pseudocode below.

Algorithm 8.2 MF-UAV Caching Algorithm with *Top-k* MAB learning-based caching policy at A-UAVs

```
Input: Total A-UAVs in its trajectory, $TAD$, next A-UAV '$x$', present A-UAV '$x-1$'
Output: $C_{MF}$ contents for MF-UAV '$y$'
Caching at A-UAVs using Top-k MAB policy (Algorithm 8.1)
while True:
    if MF-UAV leaving for next A-UAV '$x$' then do
        // Contents that are not in the future visiting A-UAV
        Update ferrying content knowledge
        // Function call from the present A-UAV '$x-1$'
        Call content-wise_TAD ( )
        // Present A-UAV sends MF-UAV visiting frequency
        Call MF-UAV_visiting_frequency ( )
        // Check what content the last MF-UAV ferried
        Call Check_previous_MF-UAV_roster ( )
            Return roster contents with respective TADs
        // Compute request interval for last MF-UAV roster
            Calculate least popular content's request interval
            Check if request time is less than its TAD and
          MF-UAV visiting duration
            if True then do
                Cache same roster
```

Algorithm 8.2. (cont'd)

```
            else
                Cache next best roster
            end if
            Check if other MF-UAVs flying with MF-UAV '$y$'
            for $l$ = 0 to $length$(MF-UAVs flying together) do
                for $k = 0\ to\ length$(A-UAV '$x$' cache $C_A^x$) do
                    Check if $k$ in $C_{MF}$ cache space of MF-UAV '$y$'
                    if True then do
                        Replace '$k$' with highest value content
                    from $C_A^{x-1}$ not cached in MF-UAV '$y$'
                    and A-UAV '$x$'
                    end if
                end for
                Cache next best roster
            end for
        end if
        Update next A-UAV '$x$', present A-UAV '$x - 1$'
    end while
```

The role of MF-UAVs is to ferry contents from the previously visited A-UAVs to the future visiting A-UAV such that the future visiting A-UAV gets the benefit of contents cached at other A-UAVs. In Algorithm 8.2, this process is described in detail. Figure 8.3 shows the impact of this collaborative algorithm.

Consider a situation in which an MF-UAV ‘y’ is ready to leave the A-UAV ‘$x - 1$’. Before caching contents, it needs the following information from A-UAV ‘$x - 1$’; 1) What are the contents eligible for ferrying; 2) What is the MF-UAVs visiting frequency; 3) What roster of ferrying content did the last MF-UAV ferry, where roster is the grouping of contents based on their popularity or value; 4) Are the next roster contents likely to be requested within the given TAD; and 5) Are MF-UAVs flying in close proximity with each other. Based on these information MF-UAV ‘y’ selectively caches contents while maintaining diversity in the contents cached by other MF-UAVs in its proximity. This means, if MF-UAVs are flying while maintaining proximity with each other or in groups, they ferry contents from consecutive rosters. To be noted that the size of a roster is same as an MF-UAV’s cache size. Therefore, if MF-UAVs are flying in groups of N_{MF}^{G} (group size), then the number of contents cached by the group is $N_{MF}^{G} \times C_{MF}$. Such selective caching policy at MF-UAVs ensures content availability maximization by avoiding redundant cache duplication.

8.6 Experimental Results and Content Dissemination Performance

Simulation experiments are performed to analyze the performance of the proposed *Top-k* MAB learning-based caching mechanism and selective caching at the micro-ferrying UAVs. An event-driven simulator accomplishes content request generation while maintaining an intra-event interval according to exponential distribution and following a Zipf popularity distribution (refer to Eqn. 8.1). To capture heterogeneity in content popularity sequence at different communities, contents are swapped with pre-decided probability [142] and the difference between the sequences are determined using Smith-Waterman Distance [125]. Default experimental parameters for the proposed *Top-k* MAB learning based caching and cache pre-loading policies are listed in Table

8.1.

Table 8.1. Default Values for Model Parameters

#	**Variables**	**Default Value**
1	Total number of contents, C	2000
2	Number of A-UAVs, N_A	4
3	Number of MF-UAVs, N_{MF}	8
4	A-UAV's Cache space (as number of contents), C_A	200
5	MF-UAV's Cache space (as number of contents), C_{MF}	25
6	Poisson request rate parameter, μ (in request/sec)	1
7	Hover rate of MF-UAV, $R_{Hover} = T_{Hover}/T_{Trajectory}$	$1/6$
8	Transit rate of MF-UAV, $R_{Transit} = T_{Transit}/T_{Trajectory}$	$1/12$
9	Zipf parameter (Popularity), α	0.4
10	Micro Ferrying UAV Trajectory	Round-robin

The performance evaluation of the proposed mechanism is accomplished via the following metrics.

Content Availability (P_{avail}): Defined as the ratio between cache hits and generated requests for a given tolerable access delay. Cache hits are the content provided to the users from the contents cached in the UAV-aided caching system (without download). Therefore, content availability indirectly indicates the content download cost of a systems as well.

Cache Distribution Optimality (*CDO*): This determines the optimality of the learnt caching policy in terms of the caching sequence. Jaro-Winkler Similarity (*JWS*) [143] is used to represent *CDO*, by computing the similarity between the content sequence from the learnt caching policy and content sequence according to cache pre-loading. It is computed by calculating the number of matches, number of transpositions required within the matches and the similarity in prefix of both sequences. It is a normalized similarity measure where 1 represents optimal caching and 0 means non-optimal caching.

Access Delay (*AD*): Performance of *Top-K* MAB model and selective caching policy for micro-ferrying UAVs is also evaluated based on the access delay which is the end-to-end delay between the generation of content request and its provisioning from the cached contents in the UAVs. This chapter reports the epoch-wise average access delay to show the improvement in caching policy as learning progresses.

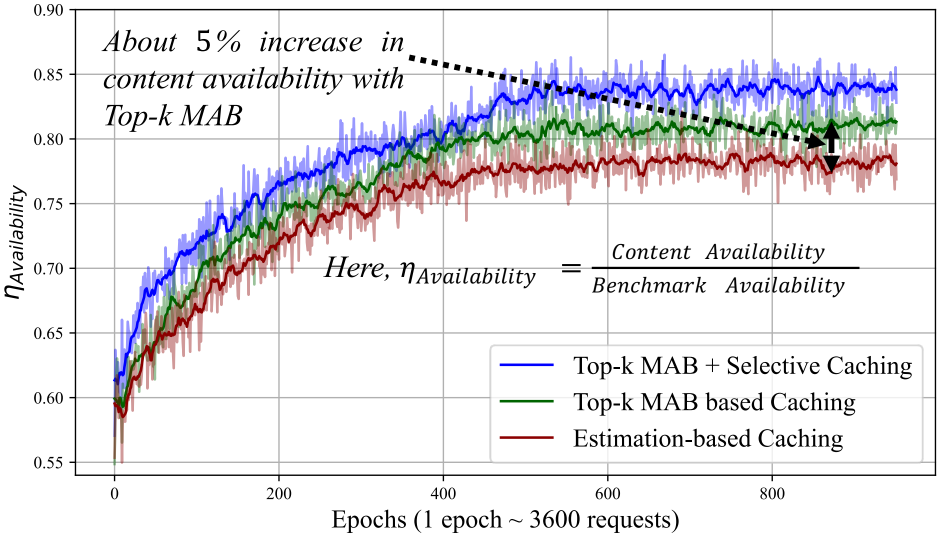


Figure 8.4. Increase in Content Availability with Top-k MAB and Selective Caching Policy

8.6.1 Effect of Exploration Strategies on Learnt Caching Policy

In order to understand the viability of the proposed *Top-k* MAB learning-based caching policy in scenarios with demand heterogeneity, two type of content popularity sequence are used. This is achieved with adjacent communities having different popularity sequences. For UCB exploration strategy, the degree of exploration is set to $\alpha_u = 2$. Also, to show the effectiveness of selective caching at micro-ferrying UAVs (MF-UAVs), *TAD* Ratio R_{TAD} for contents $\{51 - 75\}$ are kept lower than the default R_{TAD} i.e., $1/8$. To be noted that *TAD*s are represented as a ratio with respect to trajectory time ($T_{Trajectory}$) to ensure generalizability of the proposed algorithms. Figure 8.4 and 8.5 shows the convergence behavior of the learnt caching policy with *Top-k* MAB model at the A-UAVs, and selective caching at the MF-UAVs. The convergence behavior is shown in terms of content availability from the learnt caching policy.

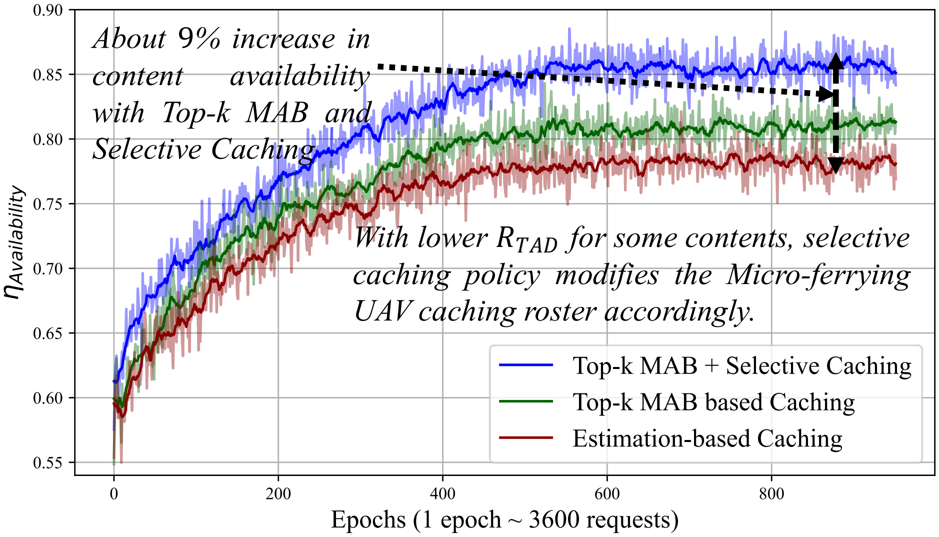


Figure 8.5. Responsiveness of Selective caching to user demand i.e., *TAD*

The observations from Figure 8.4 and 8.5 are as follows. First, the figure shows that by employing *Top-k* MAB agent at every A-UAV and selective caching at MF-UAVs, a caching policy can be learnt which can provide content dissemination performance closer to the benchmark performance [142]. The algorithm is able to leverage the multi-dimensional reward structure, as explained in Eqns. 8.7-8.9, to learn the caching policy on-the-fly (see Section 8.5.2). Second, the selective caching policy at micro-ferrying UAVs leverages the shared information between themselves and with the A-UAVs to boost the content availability closer to the benchmark performance by approximately 9% (see Figure 8.5). It utilizes the currently visiting A-UAV's caching information and the preceding MF-UAV's caching decision to algorithmically select its own contents for caching, which is also shown in Figure 8.3. Such selective caching will reduce the redundancy of multiple copies of the same content available through multiple sources at the same time. Difference in the effectiveness of selective caching can be observed in Figure 8.4 and 8.5, where caching decisions at MF-UAVs differ due to the difference in R_{TAD} in both scenarios. Third, when the agent uses UCB exploration strategy, during the initial learning epochs the content availability increases promptly due to high upper confidence value of all contents, which avoids excessive exploitation. This is due to low sampling of requests. As learning progresses, the sparse request for unpopular contents keeps the upper confidence value high which maintains consistent exploratory behavior. Figure 8.4 and 8.5 shows that such exploration strategy alone helps to boost the content availability closer to the benchmark performance by approximately 5% more than popular estimation-based methods [78], [79], [80], [81].

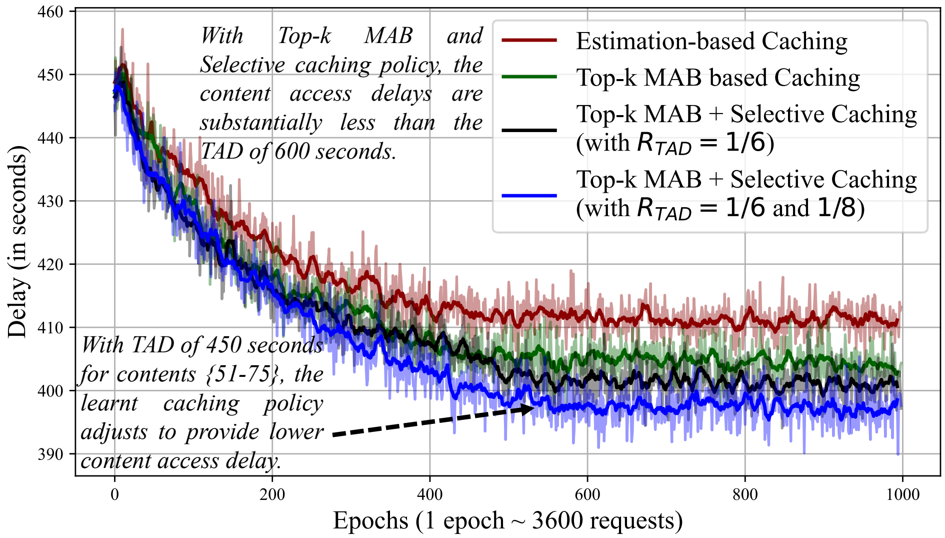


Figure 8.6. Delay with *Top-k* MAB and Selective Caching Policy

Similarly, Figure 8.6 shows the convergence behavior of the *Top-k* MAB learning-based caching agent at the A-UAVs and selective caching at micro-ferrying UAVs in terms of access delay. It is observed that as learning progresses, the access delay for requested contents reduces while the content availability increases. This shows the improvement in learnt caching policy over the learning epochs and its effect on content access delay. The best reduction in access delay is observed when Upper Confidence Bound (UCB) exploration is used at the *Top-k* MAB agent of A-UAVs and selective caching is applied at micro-ferrying UAVs.

8.6.2 Cache Similarity of Learnt Sequence with Best Sequence

The effects of learning on the cached content sequence are demonstrated in Figure 8.7. It plots Cache Distribution Optimality (*CDO*) of the cached content sequences for all the A-UAVs in terms of Jaro-Winkler Similarity (*JWS*).

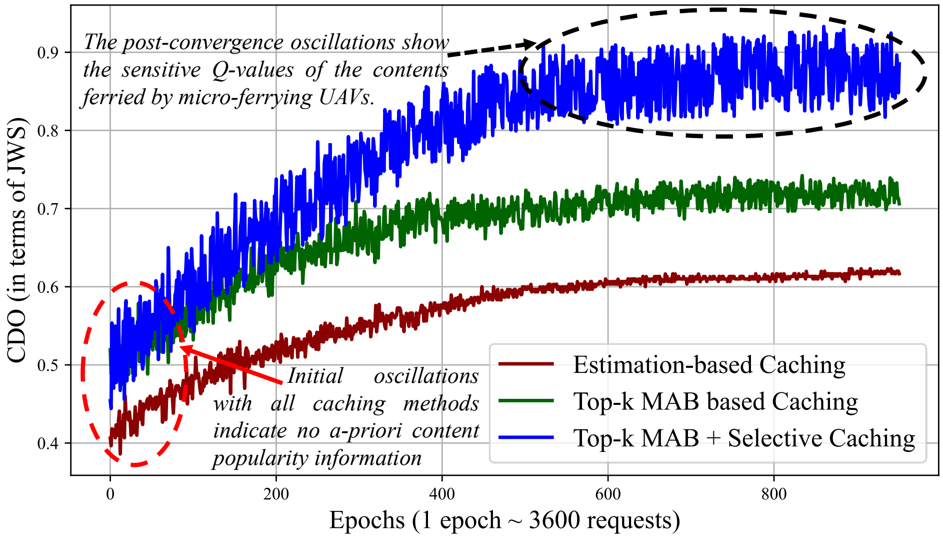


Figure 8.7. Learnt cached content sequence's similarity with benchmark sequence

The key observation are as follows. First, the average CDO between the benchmark caching sequence from cache pre-loading policy (see Section 8.4) and the cached content sequences learnt by the *Top-k* MAB agents at A-UAVs converge near 0.9, although with a certain variance. Physically, this represents higher degree of similarity after convergence, where 1 indicates complete similarity and 0 implies no similarity. Second, the cached contents improve over epochs as learning progresses. Lower CDO values after the initial epochs signify that the A-UAVs have no *a priori* local or global content popularity information. As the MAB agents learn, over epochs of generated content requests, the cached contents in the A-UAVs become more similar to the best caching sequence. Third, CDO is an indirect representation of the storage segmentation factor (λ), which is used to decide the segment sizes according to cache pre-loading policies [93]. A higher CDO implies that, along with learning, the caching policy, the *Top-k* MAB agents learn to emulate the said segmentation behavior. Finally, the partial dissimilarity of the cached content sequence can be ascribed to the uncertainty (or regret) associated with the Q-values of contents with low popularity. Also, this leads to an oscillatory convergence of CDO for the A-UAVs.

The impacts of selective caching at micro-ferrying UAVs can be distinctly seen in Fig 8.7. Selective caching at the MF-UAVs along with *Top-k* MAB caching agent at A-UAVs leads to a CDO of nearly 0.9. Note that this depends on effective caching capacity of the MF-UAVs, which is dictated by the TADs associated with content requests and the MF-UAVs visiting frequency at A-UAVs (refer Algorithm 8.2). The dependance of contents' Q-values on such information also adds to the post-convergence oscillation. To be noted that for the computation of CDO, the benchmark caching sequence is derived by considering the same effective caching capacity as the selective caching algorithm at the micro-ferrying UAVs.

8.6.3 Leveraging the Micro-Ferrying UAVs for Better Effective Caching Capacity

To elaborate on the ability of selective caching at micro-ferrying UAVs to exploit effective caching capacity, experiments are conducted with different *TAD* Ratios R_{TAD}. The comparison of performance is done with a scenario where there is one relatively larger ferrying UAV (F-UAV). Such F-UAVs can have sophisticated communication equipment as payload including a larger caching capacity (≥ total caching capacity of all MF-UAVs). The content availability according to the learnt caching policy with 24 MF-UAVs is shown in Figure 8.8. The remaining parameters are set according to the default values provided in Table 8.1.

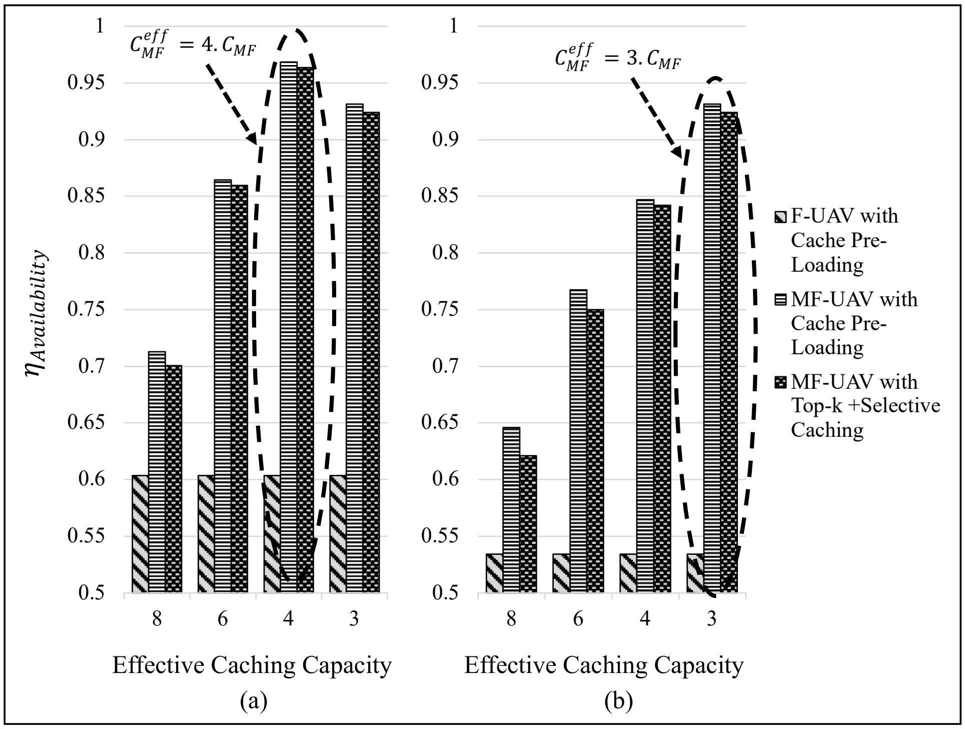


Figure 8.8. (a) Best learnt C_{MF}^{eff} for $R_{TAD} = 1/6$, (b) for $R_{TAD} = 1/8$

Following observations can be made from Figure 8.8 a. First, for a given R_{TAD} =1/6, the best content availability achieved is with effective caching capacity of $4.C_{MF}$ i.e., four times the caching capacity of an MF-UAV. Physically, this means that the 4 MF-UAVs fly very close to each other. Within the fleet of such closely flying MF-UAVs none of the pending content requests, for the ones cached at the MF-UAVs, expire by exceeding their respective *TADs*. Second, content availability increases with increase in effective caching capacity up to a certain point beyond which it decreases with further increase in effective caching capacity. This is due to two opposing effects: a) low availability period [91] for a content increases with increase in effective caching capacity which eventually decreases content availability, and b) with increase in effective caching capacity content availability increases due to more types of contents cached at MF-UAVs. Therefore, selective caching at the MF-UAVs handles the trade-off between these opposing behaviors by choosing a caching policy that increases the effective caching capacity without increasing the low availability period of contents cached at MF-UAVs.

Note that the previous explanation is valid for a particular R_{TAD}. The best learnt effective caching capacity differs when the $TADs$ associated with the content requests change. This is demonstrated in Figure 8.8 b where due to a decrease in R_{TAD} from 1/6 to 1/8, the best learnt effective caching capacity decreases. Therefore, it can be said that the learning capability of the $Top\text{-}k$ MAB agents at A-UAVs have an indirect dependence on the effective caching capacity of the MF-UAVs.

This also emphasizes the motivation behind employing micro-UAVs in the role of ferrying contents. With a given cost budget for UAVs in a content dissemination system, micro-UAVs provide flexibility in caching policies such that their effective caching capacity can be altered to fit to the users' needs. This facility cannot be leveraged with relatively larger and pricier UAVs, especially under equipment cost constraints.

8.7 Summary and Conclusion

In this chapter, a micro-UAV aided content dissemination system is proposed which can learn caching policies on-the-fly without *a priori* content popularity information. Two types of UAVs are introduced for content provisioning in a disaster/war-stricken scenario viz. anchor UAVs and micro-ferrying UAVs. Cache-enabled anchor UAVs are stationed at each stranded community of users for uninterrupted content provisioning. Micro-ferrying UAVs act as content transfer agents across the anchor UAVs. A decentralized $Top\text{-}k$ Multi-Armed Bandit Learning-based caching policy is proposed to ameliorate the limitation of existing caching methods. It learns the caching policy on-the-fly by maximizing the estimated multi-dimensional reward for the increase in local and global content availability. It is shown that a $Top\text{-}k$ MAB learning based caching policy achieves a content availability of ≈82% of maximum achievable content availability. To improve the Q-value estimates, Selective Caching Algorithm is introduced at micro-ferrying UAVs. This method combines the shared information between anchor UAVs and micro-ferrying UAVs to reduce redundant copies of contents and to produce a better estimate of top popular content at a community. Selective caching at micro-ferrying UAVs along with *Top-k* MAB learning-based caching policy at anchor UAVs boosts the content availability to ≈87% of maximum achievable content availability. With the proposed caching policies, a scaled-up micro-UAV aided network is shown to attain a content availability of nearly 95% of maximum achievable content availability. Future work on this research includes algorithmically coping with time-varying content popularity and adaptive trajectory planning in the presence of operational unreliabilities of the UAV. Furthermore, the next experiments will focus on model sharing approaches like Federated Learning in the presence of selective caching at Micro-Ferry UAVs.

Chapter 9: Federated Multi-Armed Bandit Learning for Trajectory-aware Caching Policy in Content Dissemination System using Swarm of UAVs

In the aftermath of large-scale disasters such as earthquakes, floods, and armed conflicts, survivors are often left in isolated regions without functional communication infrastructure. Traditional content dissemination mechanisms become ineffective, creating an urgent need for adaptive and resilient alternatives. Building upon the trajectory-aware caching framework developed in Chapter 8, this chapter introduces a federated, learning-driven solution that further enhances content availability in fragmented environments.

Specifically, this chapter presents a Federated Multi-Armed Bandit (FedMAB) learning approach where UAVs collaborate by sharing learned models rather than raw user data. Through this strategy, UAVs jointly optimize their caching decisions while preserving their nuanced local content caching perspective and minimizing overgeneralization of the shared models. The architecture builds upon a two-tier structure of anchor UAVs (A-UAVs) and micro-ferrying UAVs (MF-UAVs) that incorporates selective caching strategies and federated model aggregation to dynamically adapt to varying user demands, diverse content priorities, and tolerable access delays.

9.1 Motivation

Although decentralized learning through Multi-Armed Bandit algorithms enhances content caching decisions at individual UAVs, isolated learning can result in slow convergence and weak reward estimation, especially under heterogeneous and dynamic content demand. Furthermore, trajectory-aware caching strategies, while effective, remain vulnerable to operational uncertainties and shifting user preferences.

This chapter is motivated by the need to accelerate learning convergence, to enhance caching robustness across a geographically distributed UAV swarm, and to ensure coordinated decision-making without relying on centralized control. Federated Multi-Armed Bandit Learning addresses these challenges by allowing UAVs to share their learned models that enables rapid adaptation, scalable decision-making, and resilient operation in disaster-affected regions.

9.2 Design Objective

The primary objective of this chapter is to create a distributed and federated learning framework that enables UAVs to dynamically learn and optimize trajectory-aware caching policies in environments where conventional communication infrastructure is unavailable.

a) First, this chapter designs a Federated Multi-Armed Bandit (FedMAB) based caching framework that enables UAVs to collaboratively refine their caching decisions while maintaining the privacy of user demand information.

b) Second, it introduces a multi-dimensional reward structure that captures local content demand, ferrying-based dissemination patterns, and global content popularity to guide effective and adaptive caching strategies.

c) Third, the chapter presents a divergence-based weighted aggregation method to ensure that UAVs experiencing similar content request patterns contribute more significantly during federated model updates, thereby improving the alignment between local and global caching priorities.

d) Fourth, it designs a Selective Caching Algorithm for micro-ferrying UAVs (MF-UAVs), which strategically minimizes content redundancy across the swarm while maximizing overall content accessibility for isolated communities.

e) Fifth, the chapter develops a controlled latency mechanism for federated model updates that balances learning responsiveness with caching stability, ensuring that UAVs can adapt efficiently while maintaining high system performance.

f) Finally, it validates the designed FedMAB framework through extensive simulation experiments and analytical modeling, demonstrating its effectiveness in enhancing content availability, reducing access delay, improving cache optimality, and enabling adaptability under changing user preferences.

Through these objectives, this chapter aims to establish a robust, scalable, and resilient UAV-aided content dissemination system that responds intelligently to real-world challenges encountered during disaster recovery operations.

9.3 System Model

9.3.1 UAV Hierarchy

As shown in Figure 9.1, a two-tiered UAV-assisted content dissemination system is deployed. Each community is served by a dedicated A-UAV, which operate with much larger power budgets compared to Micro-UAVs described next. The A-UAVs use lateral wireless connections (i.e., WiFi etc.) to communicate with users in that community. A-UAVs can download content via an expensive vertical link such as satellite-based internet. The system introduces a set of low-power-budget Micro-UAVs [63] for the role of ferrying (MF-UAVs). MF-UAVs are mobile and possesses only lateral communication links such as Wi-Fi. Unlike the A-UAVs, the MF-UAVs do not possess expensive vertical communication interfaces such as satellite links etc. Effectively, the MF-UAVs act as content transfer agents across different user communities by selectively caching and transferring content across the A-UAVs through their lateral links.

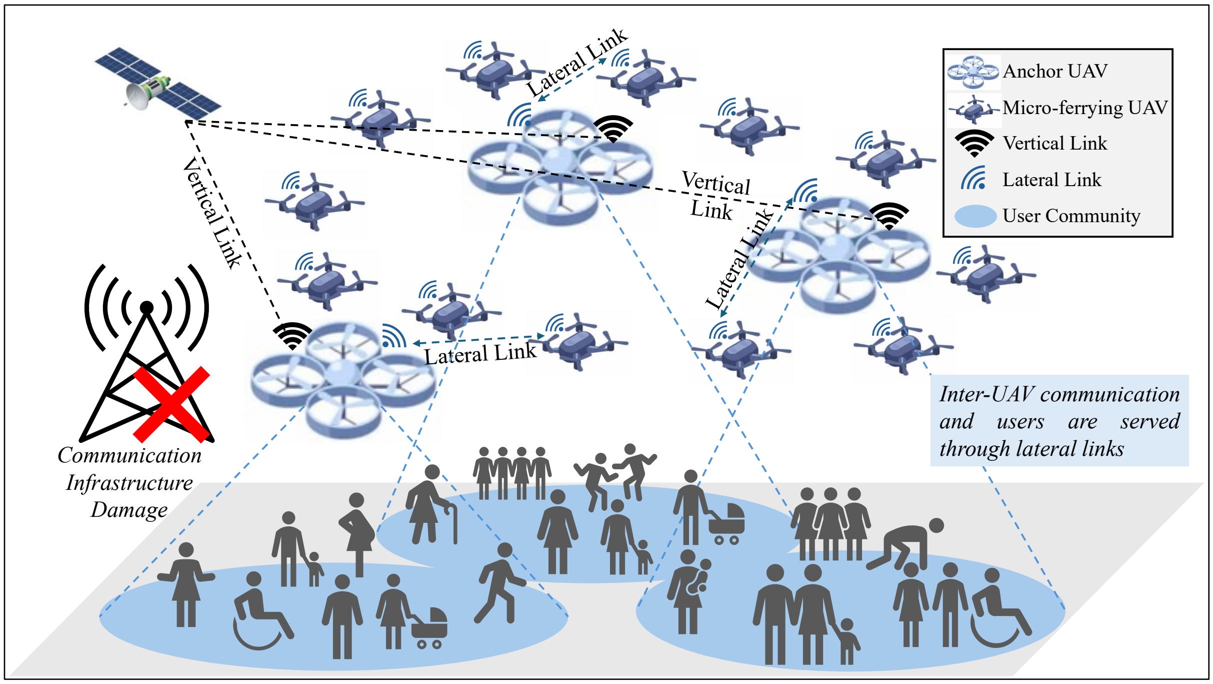


Figure 9.1. Coordinated UAV system for content dissemination in environments without communication infrastructure

Discussion: The concept of hierarchical UAV structuring and content clustering is aligned with the principles of efficient content dissemination. While the designed framework does not explicitly impose a higher-layer structure for clustering communities, aspects of hierarchical coordination already emerge through the two-tier UAV system. The Anchor UAVs (A-UAVs) inherently serve as local caching coordinators for their respective communities, while Micro-Ferrying UAVs (MF-UAVs) transport content across different regions, effectively creating a layered distribution system without rigid structuring.

Additionally, content placement decisions in FedMAB, to be discussed later, naturally result in implicit clustering, as the learning model prioritizes frequently requested content within specific regions which ensures that communities receive relevant cached data without requiring manual segmentation. This data-driven approach allows the system to dynamically adapt to evolving content access patterns rather than relying on predefined clusters.

The model also integrates an implicit indexing mechanism through its Q-value system, also to be discussed in the forthcoming sections, where content importance is dynamically ranked based on request patterns. This ranking ensures that MF-UAVs retrieve and distribute the most relevant content without needing a predefined indexing structure.

9.3.2 Content Demand and Provisioning Model

The content popularity distribution, quality of services and content provisioning are outlined below.

<u>Content Popularity</u>: Research has shown that user content request patterns often follow a power law distribution such as the Zipf distribution [91]. In Zipf distribution, the popularity of a content is proportional to the inverse of its rank, and is a geometric multiple of the next popular content. Popularity of content 'i' is given as:

$$\mathcal{P}_\alpha(i) = \left(\frac{1}{i}\right)^\alpha \Big/ \sum_{k\in C} \left(\frac{1}{k}\right)^\alpha \tag{9.1}$$

The Zipf parameter α determines the distribution's skewness, while the total number of contents in the pool is represented by the parameter C. The inter-request time from a user follows the popular exponential distribution [91].

Note that Zipf distribution is widely recognized as an appropriate model for content popularity, with empirical validation across multiple domains, including publications, online video platforms, social media, and recommendation systems such as Netflix, Instagram and more. This distribution effectively captures the heavy-tailed nature of content requests, where a small fraction of items accounts for the majority of demand.

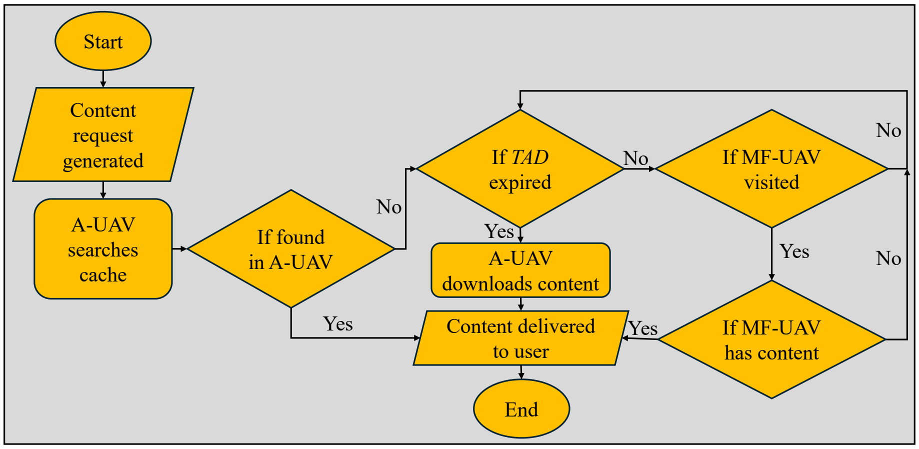


Figure 9.2. Content Delivery Process

Tolerable Access Delay (*TAD*): For each generated request, a *TAD* [70] is specified. *TAD* is a Quality-of-Service parameter that indicates the duration that a user is ready to tolerate before its requested content must be provisioned. Operationally, if a content is not available from the UAVs within the specified *TAD*, it must be downloaded from a central server using the expensive vertical links of A-UAVs. To be noted that *TAD* is request specific in which it is different for different contents depending on the requesting user's urgency.

Content Provisioning: Upon receiving a request from one of its community users, the locally deployed A-UAV first searches its local storage for the content. If the content is not found, the A-UAV waits for a potential future delivery by a traveling MF-UAV. If no MF-UAV arrives with the requested content within the specified *TAD*, the A-UAV then proceeds to download it through its vertical link. Since vertical links such as satellite links are expensive, smart caching strategies that can make the content accessible from the UAVs can be effective in reducing the overall content provisioning costs.

9.4 Content Caching Problem Formulation

The caching problem focuses on selecting which contents should be stored at UAVs to maximize content availability while considering storage constraints, access latency, and varying user demand across communities. For a given number of Anchor and Micro-ferrying UAVs, the caching problem at the UAVs can be defined as follows.

$$\underset{\forall n \in \mathcal{N}}{maximize} \left[\frac{1}{\mathcal{N}} \left[\sum_{n=1}^{\mathcal{N}} \mathbb{P}_n^{avail} \right] \right] \tag{9.2}$$

$$such\ that \sum_{i=1}^{N_A} |C_A| + \sum_{j=1}^{N_{MF}} |C_{MF}| < |C| \tag{9.3}$$

$$and\ \mathcal{T}_{\hbar}^{serve} - \mathcal{T}_{\hbar}^{req} \leq TAD_{\hbar}, \hbar \in \mathcal{H}_n, \forall\, \mathcal{H}_n = \{1,2,3,\cdots\} \tag{9.4}$$

where, $\mathbb{P}_n^{avail} = \frac{\mathcal{H}_n}{\mathcal{R}_n}$, $\mathcal{H}_n$ is the number of contents provisioned at community 'n' by the UAV system (both A-UAVs and MF-UAVs), $\mathcal{R}_n$ is the total number of requests made by users at community 'n', $\mathcal{N}$ is the number of commmunities, C is the total contents in pool, C_A is the cache of each A-UAV, C_{MF} is the cache of MF-UAVs, N_A is the number of A-UAVs, N_{MF} is the number of MF-UAVs, $\mathcal{T}_{\hbar}^{req}$ is the time at which a content '$\hbar$' is requested by a user, $\mathcal{T}_{\hbar}^{serve}$ is the time when content '$\hbar$' is served to the user by the UAV system, and $TAD_{\hbar}$ is the tolerable access delay associated with content '$\hbar$'.

The caching problem focuses on maximizing the overall content availability, as shown in Eqn. 9.2. This objective is constrained by maintaining the cumulative caching capacities of the UAVs below the total number of contents in the content pool, which is captured in Eqn. 9.3. An additional constraint is imposed by the tolerable access delay associated with a content served to the user by the UAV-aided system (refer to Eqn. 9.4).

9.5 Benchmark Caching Policy with A-Priori Demand Knowledge

This section focuses on the following caching related design questions: a) which content to be downloaded and cached in the A-UAVs so that they can serve their own community directly, and the remote communities via the traveling MF-UAVs; b) which contents to be cached when the popularity and TAD of contents vary at different communities; c) which content to be transferred from the A-UAVs to the MF-UAVs; and, d) what is the benchmark caching policy with heterogeneous content popularity at each user community and heterogeneity in request-specific TAD.

These questions are addressed by formulating a benchmark caching policy with *a priori* known heterogeneous content popularities. This benchmark caching policy also considers and modifies the caching policy to cater to the request specific TADs. After understanding the benchmark, runtime and dynamic mechanisms will be developed in a next section.

9.5.1 Caching at Anchor UAVs (A-UAVs)

For simplicity, let us consider a disaster/war-stricken area with homogeneous content popularity across all the user communities. An A-UAV is assigned to each community for content provisioning. The number of A-UAVs in the system is denoted by N_A. In such a scenario, the effective caching capacity of A-UAVs can be maximized by storing a certain number of unique contents in all the A-UAVs, and share those contents across the communities via the traveling MF-UAVs. To maximize the effective caching capabilities of all N_A A-UAVs, the cache space of each A-UAV is divided into two segments [91], namely, *Segment-1* and *Segment-2*. Let the sizes of Segment-1 and Segment-2 of the A-UAV cache be $|C_{S1}|$ and $|C_{S2}|$ respectively. They can be expressed as follows:

$$|C_{S1}| = \lambda \times |C_A| \tag{9.5}$$

$$|C_{S2}| = (1 - \lambda) \times |C_A| \tag{9.6}$$

where λ is a *storage segmentation factor* (SSF) that decides the split between the segments within a A-UAV [91]. The top $\lambda. |C_A|$ popular contents are cached in Segment-1. These contents are same across all A-UAVs whereas contents stored in Segment-2 are different. This results into the number of total Segment-2 contents stored across all N_A A-UAVs to be:

$$\left|C_{S2}^{total}\right| = N_A \times (1 - \lambda) \times |C_A| \tag{9.7}$$

These contents are shared across all user communities via the mobile MF-UAVs. These contents have popularities after the top $\lambda.|C_A|$ popular Segment-1 contents in all the A-UAVs. For symmetry, all $N_A \times (1 - \lambda) \times |C_A|$ Segment-2 contents are uniformly randomly distributed across N_A number of A-UAVs. Hence, for a given Zipf parameter α which determines the distribution's skewness, the total number of contents in the system is as follows:

$$\left|C_{sys}^{\alpha}\right| = \lambda \times |C_A| + N_A \times (1 - \lambda) \times |C_A| \Rightarrow \left|C_{sys}^{\alpha}\right| = \big(\lambda + N_A \times (1 - \lambda)\big) \times |C_A| \tag{9.8}$$

Now consider a heterogeneous demand scenario in which every community has a different demand pattern, and each content is requested with a fixed pre-decided TAD. The above caching policy is modified as follows to address such a situation. Some contents from Segment-1, termed as *exclusive* contents, are cached in one or some of the A-UAVs, but not in all of them [93]. Whereas the remaining contents from Segment-1, termed as *non-exclusive* contents, are cached at all the A-UAVs [91]. Therefore, unlike the homogeneous popularity scenario, the number of contents in Segment-1 across all A-UAVs may be more than $\lambda \times |C_A|$ due to the different A-UAV specific exclusive contents. This shown below:

$$\left|C_{S1}^{total}\right| = |C_{NE}| + \left|C_{E}^{total}\right| \geq \lambda \times |C_A| \tag{9.9}$$

Similar to the caching policy in a homogeneous popularity scenario, contents in Segment-2 do not repeat across the A-UAVs. If $C_{NE}\ and\ C_{E}^{total}$ are the non-exclusive and total exclusive contents in Segment 1, then total number of contents in the system can be modified from Eqn. 9.8, and can be expressed as follows:

$$\left|C_{sys}^{\alpha}\right| = |C_{NE}| + \left|C_{E}^{total}\right| + N_A \times (1 - \lambda) \times |C_A| \Rightarrow \left|C_{sys}^{\alpha}\right| \geq \big(\lambda + N_A \times (1 - \lambda)\big) \times |C_A| \tag{9.10}$$

The classification of content as exclusive or non-exclusive is determined by predefined access constraints that specify whether a piece of content is intended for a single community or multiple communities. Exclusive content is assigned to a specific segment and is cached at designated A-UAVs serving that community. Non-exclusive content is intended for broader dissemination and is made available across multiple A-UAVs to maximize accessibility. In the benchmark models, these classifications dictate where content is stored which ensures exclusive content remains within its intended segment while non-exclusive content is widely distributed. However, in FedMAB, that is to be discussed in the forthcoming section, caching decisions are not constrained by predefined exclusivity labels. Instead, the learning model determines caching locations dynamically based on observed request patterns. Content initially classified as exclusive or non-exclusive may be placed in different locations if the bandit-driven learning process identifies a more efficient caching strategy. This ensures that caching adapts to real-world demand rather than being restricted by static classifications.

To be noted that the above stated caching policies take the contents' popularity into consideration while making the caching decisions. However, the promptness with which a content needs to be provisioned, i.e., the TAD, may not always be positively correlated with its popularity. Therefore, unlike cache space optimization done till now, the caching policy needs modification from a perspective that considers a content's importance. Hence, unlike the cache space optimization undertaken thus far, the caching policy requires modification from a standpoint that considers the significance of content.

Now consider a demand heterogeneous scenario where every community has a different demand pattern, and each content is requested with its own specific TAD [91]. If a content is requested with less TAD, this implies that the user is not willing to wait for a visiting MF-UAV to deliver the content. Therefore, caching such time-critical contents at the A-UAVs becomes imperative. To prioritize caching of such contents in Segment-1 of A-UAVs, this chapter devices a value-based caching policy where the value of a requested content '$\mathscr{h}$' is calculated from its popularity and its TAD, and is as follows:

$$\mathcal{V}(\mathscr{h}) = \kappa \times \frac{TAD_{min}}{p_{\alpha}(1)} \times \frac{\mathcal{P}_{\alpha}(\mathscr{h})}{TAD_{\mathscr{h}}} \Rightarrow \mathcal{V}(\mathscr{h}) = \kappa \upsilon \times \frac{\mathcal{P}_{\alpha}(\mathscr{h})}{TAD_{\mathscr{h}}} \tag{9.11}$$

Here, $\mathcal{P}_{\alpha}(\mathscr{h})$ is the popularity of the content as per Zipf Distribution, $TAD_{\mathscr{h}}$ is the tolerable access delay associated with the content request, $\kappa \in [0,1]$ is a scalar weight which increases with decrease in popularity and υ is a *normalization constant*. For a given Zipf (popularity) parameter $\alpha,$ the normalization constant is calculated from the minimum possible TAD (TAD_{min}) and the maximum possible popularity, which is $\mathcal{P}_{\alpha}(1)$. The quantity $\mathcal{V}(\mathscr{h})$ is bounded between $[0, 1]$, it increases with increase in $\mathcal{P}_{\alpha}(\mathscr{h})$, and it decreases with $TAD_{\mathscr{h}}$. This value-based caching policy increases the likelihood of contents requested with low TAD to be cached *in Segment-1* of the A-UAVs, thus making them more readily available. To be noted that the cache space maximization method developed in Eqns. 9.5-9.10 still applies to this scenario. Here, the contents to be cached are chosen based on their values instead of their popularity, which is shown below:

$$\left|C_{sys}^{\mathcal{V}}\right| = |C_{NE}| + \left|C_{E}^{total}\right| + N_A \times (1-\lambda) \times |C_A| \tag{9.12}$$

9.5.2 Caching at Micro-Ferrying UAVs (MF-UAVs)

The purpose of the MF-UAVs is to ferry $\left|C_{E}^{total}\right| + N_A \times (1-\lambda) \times |C_A|$ number of contents stored across N_A number of A-UAVs (see Eqn. 9.10). Due to the limitations of per-MF-UAV caching space [63] (i.e., $|C_{MF}|$), their caching policy should be determined based on the trajectories, the value of λ, the Zipf popularity, and the $TADs$ associated with the contents to be cached [126].

Consider a situation in which an MF-UAV *'j'* is approaching towards the A-UAV *'i'*. Let U_i be the set of all exclusive contents in Segment-1 of all A-UAVs and all contents from Segment-2 of all A-UAVs in the entire system except the ones stored in A-UAV *'i'*. To maximize content availability for the users in A-UAV i*'s* community, the MF-UAV should carry $|C_{MF}|$ top valued contents (refer to Eqn. 9.12) from the set U_i while approaching A-UAV i.

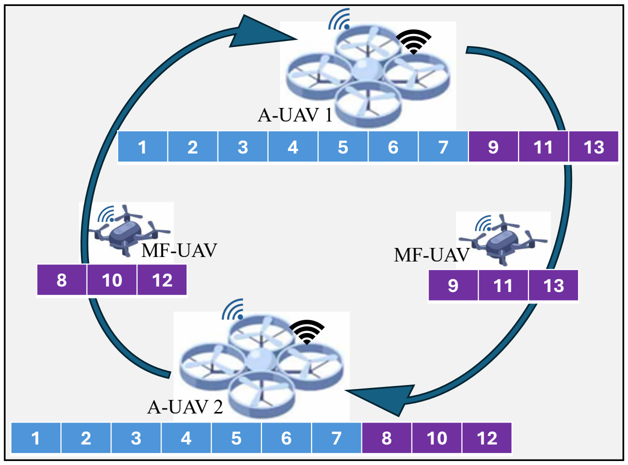


Figure 9.3. Caching Policy at MF-UAVs

The size of the set U_i can be expressed as $|U_i| = \left|C_E^{total}\right| + (N_A - 1).(1 - \lambda).|C_A|$. In scenarios when $|C_{MF}| \le |U_i|$, the MF-UAV should carry the $|C_{MF}|$ top popular contents as outlined above. Otherwise, the MF-UAV should carry all $|U_i|$ contents, leaving part of the MF-UAV cache (i.e., $|C_F| - |U_i|$) empty. This implies that an apt choice of caching policy at A-UAVs affect the utilization of MF-UAV's cache.

MF-UAV caching policy is explained in the pseudocode below.

Algorithm 9.1. MF-UAV Caching Algorithm with Value-based policy executed at the A-UAVs

```
1. Input: Total A-UAVs in its trajectory, $TAD$, next A-UAV '$i$', present A-UAV '$i - 1$'
2. Output: $C_{MF}$ contents for MF-UAV '$j$'
3. Initialize $C_A$ contents in each A-UAV based on value of contents
```

Algorithm 9.1. (cont'd)

```
while True:
    if MF-UAV leaving for next A-UAV '$i$' then do
        for $k = 0\ to\ length$(A-UAV '$i$' cache $C_A^i$) do
            Check if $k$ in $C_{MF}$ cache of MF-UAV '$j$'
            if true then do
                Replace '$k$' with highest value content from $C_A^{i-1}$ not cached in MF-UAV '$j$' & A-UAV '$i$'
            end if
        end for
    end if
    Update next A-UAV '$i$', present A-UAV '$i-1$'
end while
```

9.5.3 Theoretical Performance Upper-Bound

In this section, a theoretical performance upper-bound is computed when the A-UAVs and MF-UAVs follow the benchmark caching policy as described in Section 9.5.1. Let us consider a UAV-caching system where there are N_A number of A-UAVs, and N_{MF} number of MF-UAVs. The number of MF-UAVs traveling in a group is denoted by N_{MF}^G. MF-UAVs traverse the complete disaster region in $\mathcal{T}_{Cycle}$ seconds. The hover ratio is $\mathcal{R}_{Hov}$, which is the ratio of the time an MF-UAV stays at a community before leaving for the next to $\mathcal{T}_{Cycle}$. The transition ratio is $\mathcal{R}_{Trans}$, which is the ratio between the time on MF-UAV takes to travel from one community to the next and $\mathcal{T}_{Cycle}$. For simplicity, the inter-community distances are kept the same. The content request pattern is heterogeneous across communities with popularity parameter of α. Every request '$\hbar$' is accompanied by its respective $TAD_{\hbar}$.

The performance upper-bound has three important parts, namely, the probability $\mathbb{P}_A$ that the content is found in an A-UAV 'i', the probability of a content being found in MF-UAV $\mathbb{P}_{MF}$, and the probability that an MF-UAV is accessible near a A-UAV before content requests expire $\mathbb{P}_{Access}$. The accessibility probability $\mathbb{P}_{Access}$ is computed according to a condition $\mathbb{T}_{cond}$ which is given below:

$$\mathbb{T}_{cond} = \left(\frac{N_{MF}^{G} \times N_A}{N_{MF}} - 1\right) \times \mathcal{R}_{Hov} \times \mathcal{T}_{Cycle} + \left(\frac{N_{MF}^{G} \times N_A}{N_{MF}}\right) \times \mathcal{R}_{Trans} \times \mathcal{T}_{Cycle}$$

$$\Rightarrow \left(\left(\frac{N_{MF}^{G} \times N_A}{N_{MF}} - 1\right) \times \mathcal{R}_{Hov} + \left(\frac{N_{MF}^{G} \times N_A}{N_{MF}}\right) \times \mathcal{R}_{Trans}\right) \times \mathcal{T}_{Cycle} \qquad (9.13)$$

Eqn. 9.13 computes the time an MF-UAV takes to revisit a location. To ensure that the formulation remains applicable across different UAV configurations, it is important to clarify how the grouping of MF-UAVs and the accessibility factors influence the caching process. The first term in parentheses in Eqn. 9.13 does not become zero, because N_{MF}^{G} represents a dynamically determined grouping of MF-UAVs based on their flight dynamics and proximity. Since the grouping varies depending on how MF-UAVs ferry content and reduce redundant transmissions, the ratio $\frac{N_{MF}^{G} \cdot N_A}{N_{MF}}$ does not equal one which ensures the first term remains nonzero. Consequently, the first term does not become negative because all involved parameters are strictly positive, and by definition, $N_{MF}^{G} \leq N_{MF}$. The subtraction of one ensures that the formulation correctly accounts for accessibility relative to the number of A-UAVs in the MF-UAVs' trajectory cycle.

Additionally, $\mathcal{R}_{Hov}$ is formulated as a probability weight rather than a strict ratio to $\mathcal{T}_{Cycle}$ that ensures adaptability in determining the weighted contribution of hovering time. By treating $\mathcal{R}_{Hov}$

as a probability weight, the framework allows for dynamic adjustments based on MF-UAV group behavior which prevents an over-simplified proportionality that does not account for variations in flight patterns and spatial arrangements. This ensures that the influence of hovering time is contextually adjusted rather than statically imposed, preserving the generality of the formulation. Depending on the condition being satisfied, $\mathbb{P}_{access}$ is computed using the following piece-wise expression:

$$\mathbb{P}_{access} = \begin{cases} \dfrac{N_{MF} \times \left(\mathcal{R}_{Hov}\mathcal{T}_{Cycle} + \overline{TAD}\right)}{N_{MF}^{G} \times N_A \times \left(\mathcal{R}_{Hov}\mathcal{T}_{Cycle} + \mathcal{R}_{Trans}\mathcal{T}_{Cycle}\right)}, & for\, \overline{TAD} < \mathbb{T}_{cond} \\ 1, & for\, \overline{TAD} \geq \mathbb{T}_{cond} \end{cases}$$

$$= \begin{cases} \dfrac{N_{MF} \times \left(\mathcal{R}_{Hov}\mathcal{T}_{Cycle} + \overline{TAD}\right)}{N_{MF}^{G} \times N_A \times \left((\mathcal{R}_{Hov} + \mathcal{R}_{Trans})\mathcal{T}_{Cycle}\right)}, & for\, \overline{TAD} < \mathbb{T}_{cond} \\ 1, & for\, \overline{TAD} \geq \mathbb{T}_{cond} \end{cases} \tag{9.14}$$

Here, $\overline{TAD}$ is the mean TAD, which is used for generalization. The second part of the piece-wise expression in Eqn. 9.14 shows that for a very large TAD, the contents in an MF-UAV are always accessible. However, for $\overline{TAD}$ less than the $\mathbb{T}_{cond}$, the contents in MF-UAVs are partially accessible. Note that the physical accessibility to MF-UAVs does not guarantee the access to a requested content since the MF-UAVs can store only a limited number of contents. The probability $\mathbb{P}_{MF}$ that a content can be found in a MF-UAV is given below:

$$\mathbb{P}_{MF} = \left[\sum_{\hbar \in \{|C_E^{total}| + N_A.(1-\lambda).|C_A|\}}^{\hbar \notin \{|C_{NE}^{i}| + |C_E^{i}|\}} \mathcal{V}(\hbar)\right] \Big/ \left[\sum_{\forall C} \mathcal{V}(\hbar)\right]$$

$$\Rightarrow \mathbb{P}_{MF} = \frac{\sum_{\hbar \in \{|C_E^{total}| + N_A.(1-\lambda).|C_A|\}}^{\hbar \notin \{|C_{NE}^{i}| + |C_E^{i}|\}} \kappa \times \dfrac{TAD_{min}}{p_\alpha(1)} \times \dfrac{\mathcal{P}_\alpha(\hbar)}{TAD_\hbar}}{\sum_{\hbar=1}^{C} \kappa \times \dfrac{TAD_{min}}{p_\alpha(1)} \times \dfrac{\mathcal{P}_\alpha(\hbar)}{TAD_\hbar}} \tag{9.15}$$

The above expression considers the value of the contents from Eqn. 9.11. Now, $\mathbb{P}_A$, the probability of finding a requested content in the local A-UAV of the request generating community, is expressed as:

$$\mathbb{P}_A = \left[\sum_{\forall\ |C_{NE}|+|C_E|} \mathcal{V}(\hbar)\right] \Big/ \left[\sum_{\forall C} \mathcal{V}(\hbar)\right]$$

$$\Rightarrow \mathbb{P}_A = \left[\sum_{\hbar \in |C_{NE}|+|C_E|} \kappa\upsilon \times \frac{\mathcal{P}_\alpha(\hbar)}{TAD_\hbar}\right] \Big/ \left[\sum_{\hbar=1}^{C} \kappa\upsilon \times \frac{\mathcal{P}_\alpha(\hbar)}{TAD_\hbar}\right] \qquad (9.16)$$

Combining Eqns. 9.14, 9.15 and 9.16, the average content availability at a community 'n' can be expressed as:

$$\mathbb{P}_n^{avail} = \mathbb{P}_A + \mathbb{P}_{access} \times \mathbb{P}_{MF} \qquad (9.17)$$

Eqn. 9.17 shows that the contents from A-UAV 'i' and contents from future visiting MF-UAVs contribute towards the average availability $\mathbb{P}_n^{avail}$ at community 'n' within the specified $TADs$. Note that all unavailable contents within specified $TADs$ will be downloaded by the A-UAVs using their expensive vertical links such as a Satellite Internet link. Thus, availability indirectly indicates the content download cost in the system.

The aim of the learning-based caching policy, discussed in the next section, is to achieve the above-mentioned benchmark performance in terms of content availability. The proposed learning is achieved in a distributed manner in which all UAVs learn the caching policy without *a priori* demand information and without explicit sharing of user request data.

9.6 Federated Multi-Armed Bandit Learning for Content Caching

9.6.1 Caching Policy using *Top-k* Multi-Armed Bandit

Upon deployment in a community, a A-UAV's primary task is to optimize content availability for users by determining which contents to download and cache through its vertical link. One of the ways to approach this objective involves the utilization of a *Top-k* Multi-Armed Bandit (*Top-k* MAB) learning agent within the A-UAV.

The *Top-k* MAB learning, a variant of the classical Multi-Armed Bandit problem in reinforcement learning, is employed to maximize the cumulative reward $\mathbb{R}(T)$ over a finite time horizon T [128]. In contrast to the traditional MAB, this variant involves choosing k arms simultaneously from a set of M arms and receiving individual rewards for each arm selected.

$$\mathbb{R}_T = \max\left[\sum_{t=1}^{T}\left(\sum_{i=1}^{k}\mathbb{E}[\mathbb{R}_t(i)]\right)\right] \tag{9.18}$$

Each A-UAV is assumed to be equipped with a *Top-k* MAB agent. Here, the selection of content for caching corresponds to choosing an arm, with 'k' in '*Top-k*' representing the caching capacity (C_A) of the A-UAV. The agent's objective is to choose 'C_A' contents from a larger set of 'C' contents in order to maximize content availability for users.

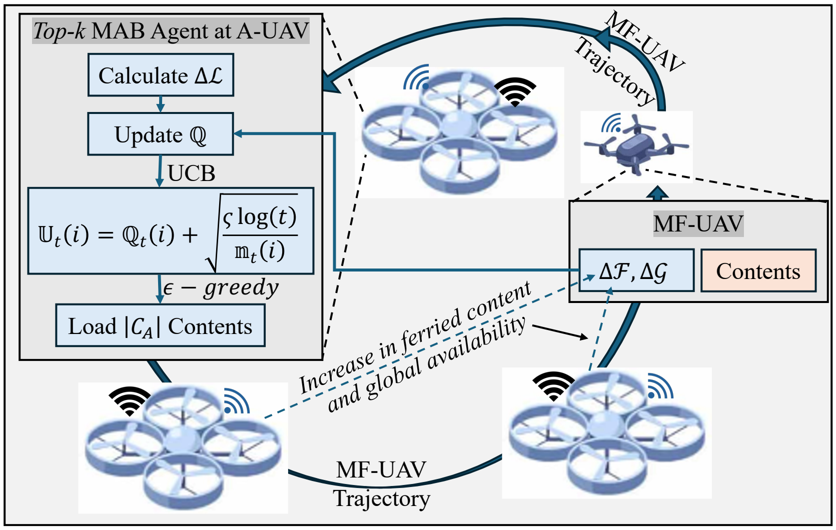

Top-k MAB Agent at A-UAV
Calculate $\Delta\mathcal{L}$
Update $\mathbb{Q}$
UCB
$\mathbb{U}_t(i) = \mathbb{Q}_t(i) + \sqrt{\frac{\varsigma \log(t)}{\mathbb{m}_t(i)}}$
$\epsilon - greedy$
Load $|C_A|$ Contents
MF-UAV Trajectory
MF-UAV
$\Delta\mathcal{F}, \Delta\mathcal{G}$
Contents
Increase in ferried content and global availability
MF-UAV Trajectory

Figure 9.4. *Top-k* Multi-Armed Bandit Learning for Caching Policy at A-UAVs

In the UAV-aided content dissemination environment, A-UAVs interact by selecting specific content sets (i.e., MAB actions) for caching. The feedback from the environment for the taken actions are in the form of rewards/penalties. Micro-ferrying UAVs play a vital role in transferring information across the system, contributing to the computation of rewards and penalties. Actions are rewarded when cached contents are requested and served within the tolerable access delay. Otherwise, they are penalized.

The learning epoch for each *Top-k* MAB agent is strategically chosen based on the MF-UAVs' accessibility at the corresponding community. Therefore, epoch duration is influenced by the visiting frequency of MF-UAVs. MF-UAVs carry the content availability information of the already visited A-UAVs in its trajectory. The *Top-k* MAB agents leverage such information and learn to cache contents through a multi-dimensional reward structure, encompassing the *local*, *ferrying*, and *global* rewards. These rewards are contingent upon the availability of the sets of locally served contents ($\mathcal{L}$), contents served at other communities via ferrying ($\mathcal{F}$), and overall contents served across all communities ($\mathcal{G}$). These contents can be served to the users directly by a A-UAV or indirectly via the visiting MF-UAVs. If a cached content is served to a user within the given *TAD*, and an increase in content availability is observed, the caching decision for the content is rewarded. The type of reward is determined by the set to which the cached content belongs to. The expressions for three types of rewards are given as follows:

$$\mathbb{R}(i,\mathcal{L}) = [(i \in \mathcal{L}) \land (\Delta_{\mathcal{L}} \geq 0)] - [(i \notin \mathcal{L}) \land (\Delta_{\mathcal{L}} < 0)] \tag{9.19}$$

$$\mathbb{R}(i,\mathcal{F}) = \frac{1}{N_A - 1} \sum_{j=1, j \neq \mathbb{X}}^{N_A} [(i \in \mathcal{F}) \land (\Delta_{\mathcal{F}} \geq 0)] - \frac{1}{N_A - 1} \sum_{j=1, j \neq \mathbb{X}}^{N_A} [(i \notin \mathcal{F}) \land (\Delta_{\mathcal{F}} < 0)]$$

$$\Rightarrow \mathbb{R}(i,\mathcal{F}) = \frac{1}{N_A - 1} \sum_{j=1, j \neq \mathbb{X}}^{N_A} [(i \in \mathcal{F}) \land (\Delta_{\mathcal{F}} \geq 0)]$$

$$+ \frac{1}{N_A - 1} \sum_{j=1, j \neq \mathbb{X}}^{N_A} \Big[[\neg[(i \notin \mathcal{F}) \land (\Delta_{\mathcal{F}} < 0)]] - 1 \Big] \tag{9.20}$$

$$\mathbb{R}(i,\mathcal{G}) = \frac{1}{N_A}\sum_{j=1}^{N_A}\left[(i \in \mathcal{G})\wedge\left(\Delta_{\mathcal{G}} \geq 0\right)\right] - \frac{1}{N_A}\sum_{j=1}^{N_A}\left[(i \notin \mathcal{G})\wedge\left(\Delta_{\mathcal{G}} < 0\right)\right]$$

$$\Rightarrow \mathbb{R}(i,\mathcal{G}) = \frac{1}{N_A}\sum_{j=1}^{N_A}\left[(i \in \mathcal{G})\wedge\left(\Delta_{\mathcal{G}} \geq 0\right)\right] + \frac{1}{N_A}\sum_{j=1}^{N_A}\Big[\big[\neg\left[(i \notin \mathcal{G})\wedge\left(\Delta_{\mathcal{G}} < 0\right)\right]\big] - 1\Big] \quad (9.21)$$

$$where, [A] = \begin{cases} 1, & if\ A\ is\ true \\ 0, & otherwise \end{cases}$$

The above equations are used for computing the reward received by a *Top-k* MAB agent at A-UAV ‘$\mathbb{X}$’. Caching content ‘i’ at A-UAV ‘$\mathbb{X}$’ is rewarded if it leads to an increase in availability. Here, $\mathbb{R}(i,\mathcal{L})$, $\mathbb{R}(i,\mathcal{F})$, and $\mathbb{R}(i,\mathcal{G})$ are local, ferrying and global rewards, respectively. The terms $\Delta_{\mathcal{L}}$, $\Delta_{\mathcal{F}}$ and $\Delta_{\mathcal{G}}$ correspond to the increase in local availability, ferried content availability, and global availability, respectively. Each type of reward is contingent upon satisfying the condition ‘$f(i)$’ in the *Iverson bracket* “$[f(i)]$“. The first terms in Eqns. 9.7, 9.8 and 9.9 represent the reward accumulated by caching content ‘i’ cached at A-UAV ‘$\mathbb{X}$’, whereas the second term is the penalty associated with adverse condition. To be noted that $\mathbb{R}(i,\mathcal{F})$, and $\mathbb{R}(i,\mathcal{G})$ are higher if the content ‘i’ is requested and served at more number of communities.

Learning employs a tabular approach where a Q-table is maintained for all contents in A-UAVs. Each content corresponds to a Q-value or action-value [130] in the Q-table. The Q-value indicates the importance of a content depending on its popularity and frequency of request. Additionally, it indirectly captures the geographical relevance of the content which is related to where the content has been requested in the disaster region. The *Top-k* MAB agent updates the Q-value for a content at each learning epoch based on the multi-dimensional rewards (Eqns. 9.19-9.21). These rewards are derived from the interactions of a A-UAV’s agent with the UAV-aided content dissemination system, shaping its understanding of optimal actions (contents to cache). The recursive Q-value update expression for content ‘i’ at A-UAV “$\mathbb{X}$“ is given as follows:

$$\mathbb{Q}_{t+1}(i) = (1-\alpha_L)\mathbb{Q}_t(i) + \alpha\left\{\mathbb{R}_t(i,\mathcal{L}) + \begin{pmatrix}[(x,\mathscr{y},z)_{A-UAV'\mathbb{X}'} = (x,\mathscr{y},z)_{MF-UAV}] \\ \times\left(\mathbb{R}_t(i,\mathcal{F}) + \mathbb{R}_t(i,\mathcal{G})\right)\end{pmatrix}\right\} \tag{9.22}$$

In this context, $\mathbb{Q}_t(i)$ denotes the Q-value associated with content ‘i’ at the ‘t^{th}’ epoch. $\mathbb{R}_t(i,\_)$ signifies the corresponding reward gained from caching content ‘i’. The term “$[(x,\mathscr{y},z)_{A-UAV'\mathbb{X}'} = (x,\mathscr{y},z)_{MF-UAV}]$“ defines the condition inside the *Iverson bracket*, taking the value 1 if micro-ferrying UAVs are within the communication range of A-UAV “$\mathbb{X}$“ and 0 otherwise. The hyper-parameter “α_L“ governs the learning rate.

Initially, all the Q-values for contents start at zero, ensuring no prior information for the *Top-k* MAB agent and assigning equal importance to all contents for caching decisions. As the learning process advances, Q-values evolve, and the best contents, characterized by the highest Q-values, are cached. This approach aims to maximize the cumulative reward, subsequently enhancing the caching policy and, in turn, improving content availability.

The *Top-k* MAB agent faces a challenge due to the numerous content combinations $\binom{C}{k}$ it must sample for caching to get the best possible estimated values for all contents. An infrequent sampling results in weak reward distribution estimates, especially as the global content population (C) increases. To address this, ϵ and its decay rate are empirically chosen in the *ϵ-Greedy* action selection policy [139]. To reduce policy dependence on ϵ, an Upper Confidence Bound (UCB) strategy [128] is employed.

$$\mathbb{U}_t(i) = \mathbb{Q}_t(i) + \sqrt{\frac{\varsigma \log(t)}{\mathbb{m}_t(i)}} \tag{9.23}$$

The UCB, denoted as $\mathbb{U}_t(i)$, is calculated using the updated Q-value $\mathbb{Q}_t(i)$, controlling hyperparameter ‘ς’, and the number of times content ‘i’ has been requested $\mathbb{m}_t(i)$. This strategy aids in content selection by favoring items with high reward potential and infrequent requests, thus promoting exploration without introducing an external exploration parameter ϵ.

Algorithm 9.2. Caching policy at a A-UAV with *Top-k* MAB Learning

```
**Initialization:**
    a. $C$: Total contents in the system
    b. $C_A$: Caching capacity of an A-UAV
    c. $\mathbb{U}$: Size $|C_A|$ initialized with 0's (Q-table with UCB)
    d. $\alpha$: Learning rate for Q-table update
    e. $\varsigma$: Degree of exploration (in UCB)
**Load** A-UAV's cache with $C_A$ randomly chosen contents.
**while** True:
    **Check** for learning epoch at A-UAV i.e., at $t^{th}$ epoch
    **if** True **then do**
        **for** $i = 0\ to\ length$(A-UAV cache size $C_A$) **do**
            **Get** reward $\mathbb{R}_t(i,\_)$  \\ according to Eqns. 9.19-9.21
            **Update** $\mathbb{U}_t(i)$          \\ from Eqn. 9.23
        **end for**
        $value = \boldsymbol{copy}(\mathbb{U})$       \\ make a copy of UCB values
        **for** $i = 0\ to\ length$(A-UAV cache size $C_A$) **do**       \\ Reload contents (Select arms)
            $c_{max} = \boldsymbol{argmax}(value)$
            **Load** $c_{max}$ to A-UAV
            **Set** $value[c_{max}] = -\infty$
        **end for**
```

Algorithm 9.2. (cont'd)

```
    end if
end while
```

The segment-based caching benchmarks in this study are used solely for theoretical comparisons and do not reflect how the *Top-k* MAB framework operates. These benchmarks represent deterministic edge cases for establishing performance bounds, whereas the learning-based *Top-k* MAB model is fundamentally more capable in handling fairness dynamically. Caching decisions are determined through a multi-dimensional reward structure that accounts for individual content popularity, inter-community content influence, and the global impact of caching choices. Unlike benchmarks that impose static constraints, the combinatorial nature of *Top-k* MAB inherently ensures fairness by dynamically adjusting caching priorities based on real-time content request patterns.

The Q-values computed in the combinatorial bandit model quantify content importance rather than predefined allocation constraints. Caching ability depends solely on UAV storage capacity that ensures fairness emerges as a function of storage limitations rather than arbitrary segment size constraints.

Furthermore, *Top-k* MAB's caching efficiency is entirely dependent on request generation patterns (i.e., sampling), unlike deterministic benchmarks that rely on predefined segment structures. Larger communities inherently generate higher request volumes which leads them to receive more cached content organically through bandit learning. This eliminates the need for manually imposed fairness constraints. Similar effects can be expected while content requests manifest urgency or emergency, for which the learning-based model adapts such that it can determine the importance of contents.

Complexity Analysis: To evaluate the computational feasibility of the *Top-k* Multi-Armed Bandit algorithm, we analyzed the complexity of its key components, including initialization, reward calculation, action selection, and the Q-value updates. These components collectively determine the scalability of the algorithm for large-scale UAV-assisted content dissemination systems.

The algorithm begins with an initialization phase where each A-UAV creates a Q-table to track the rewards associated with all N contents. This initialization step, which is performed only once, has a time complexity of $O(N)$. Given the one-time nature of this step, its computational burden does not affect the scalability of the system in a major way. During each learning epoch, the algorithm computes rewards from k arms selected from the total N contents. This reward calculation, which occurs across T epochs, involves evaluating the multi-dimensional reward structure. As $k \ll N$, the complexity of this step is $O(T.k)$, making it computationally light enough for real-time operations. The most computationally intensive step is the action selection process, where the Upper Confidence Bound (UCB) method ranks all N contents to identify the top k arms to cache. Sorting the contents at each epoch results in a complexity of $O(N \log N)$, which, over T epochs, leads to a total complexity of $O(T.N \log N)$. While this step dominates the overall computational complexity, it is till bounded by logarithmic growth, thus keeping it scalable for large content pools. Finally, the Q-values of the selected arms are updated at each epoch based in the observed rewards. This update process has a complexity of $O(T.k)$, which remains manageable due to the small value of k relative to N.

Combining these components, the overall time complexity of the *Top-k* MAB algorithm is $O(N) + O(T.k) + O(T.N \log N) + O(T.k)$. Simplifying this for $k \ll N$, the dominant term is $O(T.N \log N)$. Therefore, the overall time complexity is $O(T.N \log N)$, with the action selection step contributing the most significant computational cost. This complexity demonstrates the algorithm's suitability for large-scale systems, as its logarithmic scaling ensures efficiency even with large number of contents. Furthermore, the *selective caching* mechanism employed by MF-UAVs, which will be explained in a forthcoming subsection, with a linear complexity of $O\left(N_{Trajectory}.(C_{MF}+N_A)\right)$, complements the learning-based approach by ensuring effective content distribution without introducing significant computational overhead. Note that here C_{MF} is the cache size of MF-UAVs, N_A is the number of A-UAVs, and $N_{Trajectory}$ is the number of trajectory changes.

Thus, the overall complexity of the *Top-k* MAB algorithm is dominated by $O(T.N\log N)$, which makes it scalable for large content pools. The linear complexity of the MF-UAV caching mechanism further supports efficient operation in distributed settings. The designed framework is computationally efficient, scalable and is well-suited for real-time UAV-assisted content dissemination in disaster-stricken environments.

There are a few limitations of the above learning based caching method. First, relying on MF-UAVs to ferry global content availability information makes learning slow. Especially so in large disaster areas with multiple communities. Second, communities with fewer users result in fewer content requests for the corresponding local A-UAVs. The problem is particularly compounded for the less popular contents for which the popularity reduces drastically following Zipf distribution (see Section 9.3.2). This lack of requests creates a learning challenge, leading to less accurate reward distribution estimates [140]. This results in unstable Q-values for less popular content. Finally, the *Top-k* MAB agent has to sample $\binom{C}{k}$ content combinations. This results in infrequent reward estimations with increasing C (i.e., the total number of contents in pool), thus weakening the estimate of the reward distribution [140]. This leads to sensitive and unstable Q-values of contents.

These challenges can be mitigated by employing a Federated Multi-Armed Bandit (FedMAB) Learning approach. Such an approach involves integrating the *Top-k* MAB models from all A-UAVs. The mechanism is presented below.

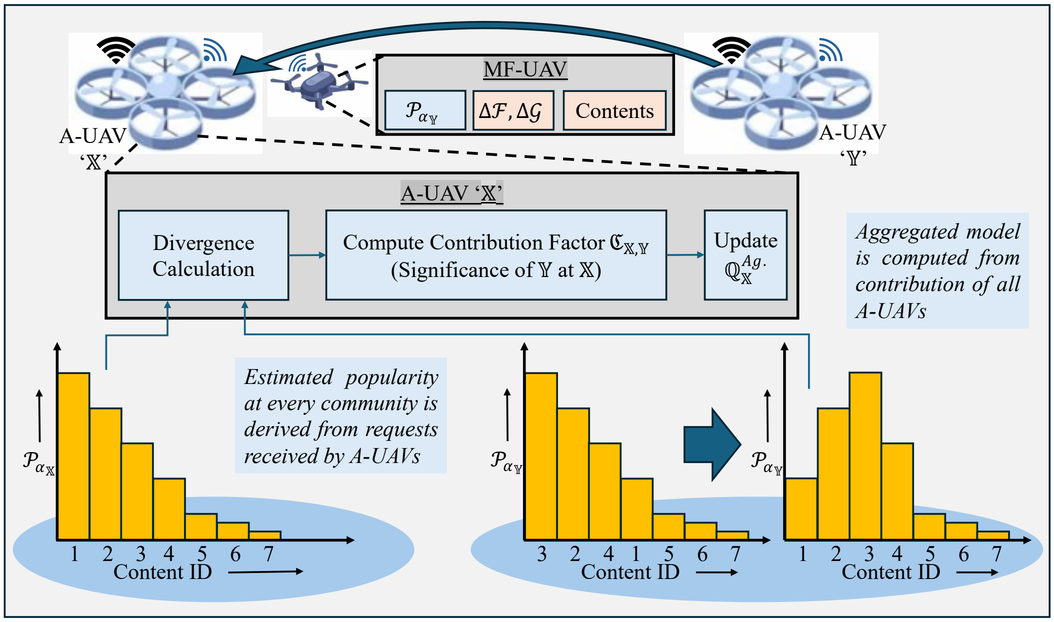
9.6.2 Distributed Caching with Federated Multi-Armed Bandit

This mechanism applies the principles of Federated Learning [63], [132], [133], [144] to the UAV-caching scenario. Each A-UAV serves as a client [144] in Federated Learning, with its local model representing information about cached contents, cache hits [72], and content availability. Note that cache hits indicate how often a content cached in an A-UAV is requested and served within the user-species TAD, as defined in Section 9.3.2. The Q-table of the *Top-k* MAB agent serves as the model for each A-UAV.

MF-UAVs acts as model aggregators [144], chosen for their ability to access Q-tables of A-UAVs in their respective trajectories. They aggregate the acquired Q-tables aiming to improve the *Top-k* MAB model at each A-UAV. They receive the Q-tables from all A-UAVs in their trajectories, and send the aggregated model back to the A-UAVs. This aggregated model helps the A-UAVs to decide as to which content to cache based on the top '$|C_A|$' Q-values.

As per the standard Federated Learning paradigm, Q-tables are initialized at the A-UAVs, and learning epochs are set based on the MF-UAVs' visiting frequencies. The learning epoch's dependance on MF-UAVs' visit is important to capture the rewards $\mathbb{R}(\mathcal{F})$ and $\mathbb{R}(\mathcal{G})$. Note that these rewards are associated to the ferried contents and their impact on global availability (refer Eqns. 9.20 and 9.21). The Q-values of individual *Top-k* MAB agents are updated at each epoch, thus capturing the latest content request patterns and A-UAV caching decisions. This is termed as "*personal experience*" which is akin to the local training stage in Federated Learning. After gaining personal experience, an A-UAV's model is improved by exchanging information with its adjacent

A-UAVs through the traveling MF-UAVs. To be noted that the quality of the aggregated model depends on the freshness of information ferried by the MF-UAVs. To be noted that cost constraints can reduce the deployment of high-cost UAVs, leading to less frequent information updates. Therefore, leveraging more number of affordable MF-UAVs within the UAV-assisted caching system ensures more consistent information collection, crucial for maintaining model accuracy under budgetary limitations.


MF-UAV
$\mathcal{P}_{\alpha_{\mathbb{Y}}}$
$\Delta\mathcal{F}, \Delta\mathcal{G}$
Contents
A-UAV 'X'
A-UAV 'Y'
A-UAV 'X'
Divergence Calculation
Compute Contribution Factor $\mathfrak{C}_{\mathbb{X},\mathbb{Y}}$ (Significance of $\mathbb{Y}$ at $\mathbb{X}$)
Update $\mathbb{Q}_{\mathbb{X}}^{Ag.}$
$\mathcal{P}_{\alpha_{\mathbb{X}}}$
1 2 3 4 5 6 7
Content ID
$\mathcal{P}_{\alpha_{\mathbb{Y}}}$
3 2 4 1 5 6 7
Content ID
$\mathcal{P}_{\alpha_{\mathbb{Y}}}$
1 2 3 4 5 6 7
Content ID

Aggregated model is computed from contribution of all A-UAVs

Estimated popularity at every community is derived from requests received by A-UAVs

Figure 9.5. Contribution factor for Federated Multi-Armed Bandit implementation at A-UAVs

Unlike weight matrix aggregation in neural networks in regression/classification models [132], Federated Multi-Armed Bandit involves the aggregation of Q-values. Each A-UAV's contribution during aggregation is determined based on its importance. This is synonymous to weights associated with devices in classical Federated Learning algorithm. Such contributions can be defined as *contribution factor* which is crucial during the aggregation process. It is calculated based on the similar between the estimated popularity distributions of contributing A-UAVs. The mechanism is shown using the following expressions:

$$\mathfrak{C}_{\mathbb{X},\mathbb{Y}} = [\rho - \mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{Y}})] \Bigg/ \left[\sum_{\mathbb{l}=1}^{N_A} \left(\rho - \mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{l}})\right)\right] \qquad (9.24a)$$

$$\text{where}, \rho = \max\left(\sum_{\mathbb{l}=1}^{N_A} \mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{l}})\right) \qquad (9.24b)$$

Using the Eqn. 9.25, the aggregated model can be shown as:

$$\mathbb{Q}_{\mathbb{X}}^{Ag.}(i) = \sum_{\mathbb{y}=1}^{N_A} \left(\mathfrak{C}_{\mathbb{X},\mathbb{y}} \times \mathbb{Q}_{\mathbb{y}}(i)\right)$$

$$= \left[\sum_{\mathbb{y}=1}^{N_A} \left\{\left(\rho - \mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{y}})\right) \times \mathbb{Q}_{\mathbb{y}}(i)\right\}\right] \Bigg/ \left[\sum_{\mathbb{l}=1}^{N_A} \left(\rho - \mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{l}})\right)\right] \qquad (9.25)$$

In Eqn. 9.24, the contribution factor $\mathfrak{C}_{\mathbb{X},\mathbb{Y}}$ denotes the significance of A-UAV $\mathbb{Y}$’s model when the MF-UAV is at A-UAV ‘$\mathbb{X}$’, where N_A is the total number of A-UAVs. $\mathcal{P}_{\mathbb{X}}$ and $\mathcal{P}_{\mathbb{Y}}$ are content popularity distributions estimated at A-UAVs ‘$\mathbb{X}$’ and ‘$\mathbb{Y}$’, respectively. The KL divergence [134], denoted as ‘$\mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{Y}})$’, quantifies the distinction between these distributions. Thus, the term “$\rho - \mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{Y}})$“ represents how similar the content popularity distributions are near A-UAVs ‘$\mathbb{X}$’ and ‘$\mathbb{Y}$’. To be noted that the term ‘ρ’ is included in the expressions due to the unbounded nature of ‘$\mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}||\mathcal{P}_{\mathbb{Y}})$’.

Using the contribution factor from Eqn. 9.24, the aggregated model is determined through Eqn. 9.25. In this equation, $\mathbb{Q}_{\mathbb{X}}^{Ag.}(i)$ signifies the aggregated Q-value of content ‘i’ at A-UAV ‘$\mathbb{X}$’. A higher importance is assigned to A-UAV “$\mathbb{y}$’s” model if its estimated content popularity distribution is more similar to that of A-UAV ‘$\mathbb{X}$’, and vice versa.

Aggregating the Q-tables enhances the estimated reward associated with each content. In a learning epoch, the generated requests at a community might be insufficient for an accurate reward estimate at its A-UAV. Without model aggregation, A-UAVs end up with a weaker estimate of the reward distribution. Q-table aggregation, as proposed above, enhances the estimated rewards without requiring content request information from all the A-UAVs.

However, Q-table aggregation overlooks local popularity nuances when demand varies among the communities. This issue mirrors the personalization-generalization problem in Federated Learning [144], [145]. A-UAV's Q-table, updated with personal experiences using Eqns. 9.22 and 9.23, is akin to a personalized (local) model, while the aggregated Q-table in Eqn. 9.25 signifies a generalized (global) model. This chapter employs weighted averaging to retain local popularity context while improving reward estimation. This can be expressed as:

$$\mathbb{Q}_{\mathbb{X}}^{Upd.}(i) = \omega_1 \mathbb{Q}_{\mathbb{X}}(i) + \omega_2 \mathbb{Q}_{\mathbb{X}}^{Ag.}(i) \tag{9.26}$$

In the given context, the weights ω_1 and ω_2 are critical in determining the influence of both local and global (aggregated) models in updating the model $\mathbb{Q}_{\mathbb{X}}^{Upd.}$. For the experiments in this chapter, ω_1 is empirically set to 0.99, indicating a strong preference for local content popularity, which is assumed to be relatively stable over time. However, a correct choice of ω_1 is pivotal, especially in scenarios where the content popularity is dynamic. For such cases, the adaptive selection of ω_1 is governed by:

$$\omega_1 = 1 - \left[\mathfrak{D}_{JS}\left(\mathcal{P}_{\mathbb{X}}^{t}, \mathcal{P}_{\mathbb{X}}^{t'}\right)/ln\,2\right];\ \omega_1 \colon \mathfrak{D}_{JS}\left(\mathcal{P}_{\mathbb{X}}^{t}, \mathcal{P}_{\mathbb{X}}^{t'}\right) = \frac{1}{2}\left(\mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}^{t}||\mathcal{M}) + \mathfrak{D}_{KL}\left(\mathcal{P}_{\mathbb{X}}^{t'}||\mathcal{M}\right)\right) \tag{9.27}$$

Here, $\mathfrak{D}_{JS}\left(\mathcal{P}_{\mathbb{X}}^{t}, \mathcal{P}_{\mathbb{X}}^{t'}\right)$ is the Jensen-Shannon Divergence [134], indicating the dissimilarity between content popularity distributions at times t and t'. $\mathcal{M}$ is the mean distribution calculated as $\mathcal{M} = \left(\mathcal{P}_{\mathbb{X}}^{t} + \mathcal{P}_{\mathbb{X}}^{t'}\right)/2$ and $\mathfrak{D}_{KL}(\mathcal{P}_{\mathbb{X}}^{-}||\mathcal{M})$ is the Kullback-Leibler divergence. A high ω_1 value implies minimal change in content popularity over time.

The weight associated with the aggregated model ω_2 in Eqn. 9.26 can be expressed as:

$$\omega_2 = e^{-\beta_d t} \times \left(1 - \mathbb{Q}_{\mathbb{X}}(i)\right)/\beta_s \tag{9.28}$$

Here, β_d and β_s represent the weight decay factor and scaling factor, respectively. The parameters ensure that the contribution of global (aggregated) model reduces as learning progresses. The formulation of Eqn. 9.28 reflects the decreasing relevance of the global model as learning advances, with an embedded regret component “$1 - \mathbb{Q}_{\mathbb{X}}(i)$“. This idea is backed by the assumption that as learning progresses, the local models will reflect the true value of contents. Hence, the expressions for weights ω_1 and ω_2 from Eqns. 9.27 and 9.28 can be replaced in Eqn. 9.26:

$$\mathbb{Q}_{\mathbb{X}}^{Upd.}(i) = \left[1 - \left\{\mathfrak{D}_{JS}\left(\mathcal{P}_{\mathbb{X}}^{t}, \mathcal{P}_{\mathbb{X}}^{t'}\right)/ln\,2\right\}\right] \times \mathbb{Q}_{\mathbb{X}}(i) + \left[\frac{e^{-\beta_d t}}{\beta_s} \times \left(1 - \mathbb{Q}_{\mathbb{X}}(i)\right)\right] \times \mathbb{Q}_{\mathbb{X}}^{Ag.}(i) \qquad (9.29)$$

The hyper-parameters ‘β_d’ and ’β_s’ should be empirically optimized to maintain local relevance in content popularity, particularly in heterogeneous environments.

This framework integrates Federated Learning with *Top-k* Multi-Armed Bandit principles for high-performance caching in autonomous UAVs (A-UAVs). It leverages model aggregation at MF-UAVs and updates A-UAV models with a balance of personalized and aggregated data. The updated Q-table is used for caching decisions, thus prioritizing contents with the highest Q-values for enhanced content availability across all user communities.

The dependency on MF-UAVs for model aggregation does not introduce significant bottlenecks, as multiple MF-UAVs operate in parallel to facilitate decentralized updates across different A-UAVs. This naturally distributes the computational load by preventing a single UAV from becoming an aggregation bottleneck. Additionally, the contribution-based weighting mechanism (Eqn. 9.24) ensures that only relevant updates are merged which reduces unnecessary model exchanges and further optimizes aggregation efficiency.

For even greater scalability, an alternative approach could involve hierarchical aggregation, where A-UAVs first perform localized model updates before forwarding refined models to MF-

UAVs for global merging. This multi-level aggregation structure could reduce communication overhead in networks with a larger number of UAVs that can further enhance system scalability. While the current approach remains effective under the evaluated network conditions, future work can explore hierarchical aggregation strategies to optimize performance in extremely large deployments.

The designed FedMAB framework ensures privacy in federated learning-based caching by limiting shared information to content popularity distribution parameters and Q-values, which serve as reward distribution estimates. Unlike traditional federated learning approaches that involve direct transmission of graph-based models which represents user data or raw request logs, FedMAB does not share individual user content requests, behavioral patterns, or identifiable data. The aggregation process exchanges only learned model parameters relevant to content caching decisions which makes the framework inherently privacy-preserving. Since Q-values represent estimated rewards from caching actions rather than explicit user information, the system effectively optimizes caching policies without exposing sensitive data. Furthermore, these Q-values undergo aggregation at MF-UAVs, where updates are computed based on local learning experiences, without requiring access to underlying user-generated request data.

This decentralized learning mechanism ensures that UAVs collaboratively refine caching strategies while preserving privacy. Since no direct content request logs or personal information are transmitted, this structure ensures secure model sharing which minimizes the risk of data leakage while maintaining collaborative learning efficiency.

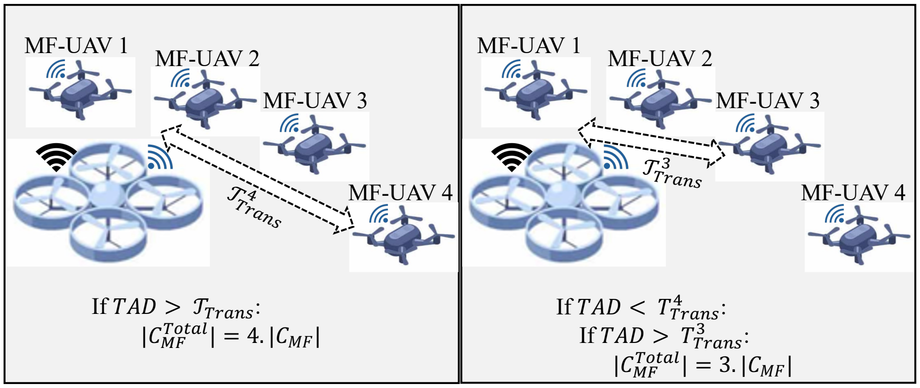


Figure 9.6. Increase in collective caching capacity of MF-UAVs through Selective Caching

9.6.3 Selective Caching at Micro-Ferrying UAVs (MF-UAVs)

The role of MF-UAVs is to ferry contents from the previously visited A-UAVs to the future visiting A-UAVs such that the future visiting A-UAVs get the benefit of contents cached at other A-UAVs. Ideally, the purpose of the MF-UAVs is to ferry around a subset of $C_E^{total} + N_A.(1-\lambda).C_A$ number of contents stored across N_A number of A-UAVs (see Section 9.5.1). However, such implementation leads to replication of all ferried contents, resulting in underutilized cache space at the MF-UAVs. Due to the limitation of per-MF-UAV caching space (i.e., C_{MF}), their caching policy should be jointly determined based on their trajectories, learnt caching policy at the A-UAVs, content request patterns, and the tolerable access delays ($TADs$) associated with the contents to be cached. A "*Selective Caching*" mechanism as the MF-UAV caching policy is explained in the pseudocode below.

Algorithm 9.3. Selective Caching Algorithm MF-UAV with FedMAB caching at A-UAV

```
Input: Total A-UAVs in its trajectory, $TAD$, next A-UAV '$\mathbb{X}$', present A-UAV '$\mathbb{X} - 1$'
Output: $C_{MF}$ contents for MF-UAV '$\mathcal{Y}$'
Caching at A-UAVs using FedMAB policy // Eqns. 9.19-9.29
while True:
    if MF-UAV leaving for next A-UAV '$\mathbb{X}$' then do
```

Algorithm 9.3. (cont'd)

```
    // Contents that are not in the future visiting A-UAV
    **Update** ferrying content knowledge
    // Function call from the present A-UAV '$\mathbb{X} - 1$'
    **Call** content-wise_TAD ( )
    // Present A-UAV sends MF-UAV visiting frequency
    **Call** MF-UAV_visiting_frequency ( )
    // Check what content the last MF-UAV ferried
    **Call** Check_previous_MF-UAV_roster ( ) **Return** roster contents with respective TADs
    // Compute request interval for last MF-UAV roster
        **Calculate** least popular content's request interval
        **Check** if request time is less than its TAD and MF-UAV visiting duration
        **if** True **then do**
            Cache same roster
        **else**
            Cache next best roster
        **end if**
        **Check** if other MF-UAVs flying with MF-UAV '$\mathcal{Y}$'
        **for** $l$ = 0 to $length$(MF-UAVs flying together) **do**
            **for** $k = 0\ to\ length$(A-UAV '$\mathbb{X}$' cache $C_A^{\mathbb{X}}$) **do**
                **Check** if $k$ in $C_{MF}$ cache space of MF-UAV '$\mathcal{Y}$'
                **if** True **then do**
```

Algorithm 9.3. (cont'd)

```
                        Replace '$k$' with highest value content from $C_A^{\mathbb{X}-1}$
                        not cached in MF-UAV '$\mathcal{Y}$' and A-UAV '$\mathbb{X}$'
                    **end if**
                **end for**
                Cache next best roster
            **end for**
        **end if**
        **Update** next A-UAV '$\mathbb{X}$', present A-UAV '$\mathbb{X}-1$'
    **end while**
```

In Algorithm 9.3, the process of selective caching is described in detail. Consider a situation in which an MF-UAV '$\mathcal{Y}$' is ready to leave the A-UAV '$\mathbb{X}-1$'. Before caching contents, it needs the following information from A-UAV '$\mathbb{X}-1$'; 1) what are the contents eligible for ferrying to A-UAV '$\mathbb{X}$'; 2) what is the MF-UAVs visiting frequency; 3) what roster of ferrying content did the last MF-UAV ferry, where roster is the grouping of contents based on their popularity or value; 4) are the next roster contents likely to be requested within the given TAD; and 5) are MF-UAVs flying in groups. Based on these information, MF-UAV '$\mathcal{Y}$' selectively caches contents which helps in maintaining diversity in the contents cached by all MF-UAVs in its vicinity. This means, if MF-UAVs are flying in groups or traversing in close proximity from each other, they ferry contents from consecutive rosters. To be noted that the size of a roster is same as an MF-UAV's cache size. Therefore, if subsets of MF-UAVs are considered collectively as a group of N_{MF}^{G} (group size), then the number of contents cached by the group is $N_{MF}^{G} \times C_{MF}$. Such selective caching policy at MF-UAVs ensures content availability maximization by avoiding redundant content replication.

9.6.4 Enhancing Federated Learning with Controlled Latency

The use of A-UAVs equipped with federated multi-armed bandit (FedMAB) learning algorithms offers a promising avenue for adaptive learning and decision-making based on user demands and network conditions. However, the model aggregation nature of FedMAB, while enhancing content delivery services, can inadvertently diminish the benefits of selective caching strategies. Especially so when such a strategy is crucial for managing a UAV-network's storage resources effectively.

To address this, a nuanced latency approach that integrates Federated Multi-armed Bandit learning at A-UAVs with selective caching at MF-UAVs is proposed. This approach maintains the integrity and benefits of both federated learning at A-UAVs and selective caching at MF-UAVs by introducing controlled latency into the A-UAVs' learning cycles.

Mechanism Details: The modified FedMAB learning algorithm with latency introduces a deliberate delay in the divergence-based weighted computation updates of A-UAVs. In simpler terms it adds a delay between the *Top-k* MAB update and the model aggregation at A-UAVs. This delay is managed through a *latency_counter*, which tracks the number of learning epochs elapsed since the last federated learning update. Only when this counter exceeds a predefined threshold, T_L, does the A-UAV proceed with its learning and cache update process via federated learning (refer to Eqns. 9.24-9.29). This controlled latency allows MF-UAVs more time for data analysis and informed decision-making regarding selective caching.

During the latency period, A-UAVs continue to collect data, learn via *Top-k* MAB agents, and perform their regular operational functions. However, they postpone the federated learning cycle's execution, allowing MF-UAVs to assess and analyze the cached content across various A-UAVs.

MF-UAVs can then identify which contents are likely to be in higher demand and ensure their availability by ferrying them between A-UAVs. This synchronization of learning with the mobility patterns of MF-UAVs enables more strategic and informed decisions regarding content caching and distribution.

Algorithm 9.4. Federated Multi-Armed Bandit Learning with Strategic Latency for A-UAVs

```
1. Input:
      a. $C$: Total contents in the system.
      b. $C_A$: Caching capacity of an A-UAV.
      c. $T_L$: Latency Threshold.
2. Initialization:
      a. Set latency_counter to 0 for each A-UAV
      b. Initialize each A-UAV's cache with randomly selected $C_A$ contents
      c. Set Q-values for all content to 0          // These values help track content demand.
      d. Define learning rate ($\alpha$) and exploration parameter ($\varsigma$)
3. Main Loop:
4.    While the system is running:
5.          Check if it's time (current epoch) for a learning update
              // This could be determined by MF-UAV flight time
6.          Calculate reward $\mathbb{R}_t(i,\_)$ for content $i$ in A-UAV
7.          Update the Q-value for all cached contents using MAB
              // Based on calculated reward and the learning rate ($\alpha$)
8.          If latency_counter >= $T_L$ then:
9.                Compute Divergence-based Weights
```

Algorithm 9.4. (cont'd)

```
10.             **Update** Q-values using Eqns. 9.24-9.29
11.             **Reset** *latency_counter* to 0
                  // Indicating an update has been completed.
12.         **Else If** *latency_counter* <= $T_L$ **then**:
13.             **Increment** *latency_counter* by 1
                  // This delays the Federated learning update cycle
14.         **Copy** the Q-values to a temporary list for manipulation
15.         **For** each slot in the A-UAV's cache:
16.             **Select** not cached content with the highest Q-value
17.             **Update** the cache to include this content
                  // Replace the least demanded content if necessary.
18.             **Update** the selected content's Q-value to $-\infty$
                  // In the temporary list to avoid reselection
19.     **Repeat** steps 4-18 for an adaptive system
```

This latency-based approach enhances content availability across the network and optimizes the use of network resources, ensuring a balance between learning efficacy and caching efficiency. By integrating the dynamic learning capabilities of A-UAVs with the selective caching strategies of MF-UAVs, the system becomes more resilient, efficient, and user-centric.

9.7 Experimental Results and Content Dissemination Performance

Simulation experiments were conducted to evaluate the performance of the designed FedMAB learning-based caching mechanism and selective caching at micro-ferrying UAVs. An event-driven simulator was used to generate content requests, maintaining intervals between events according to an exponential distribution and following a Zipf popularity distribution (see Eqn. 9.1). To account for variations in content popularity across different communities, contents were swapped with a predetermined probability [93], and differences between sequences were maintained using the Smith-Waterman Distance [125]. The default system parameters for the FedMAB-based caching and cache pre-loading policies are provided in Table 9.1.

Table 9.1: Default Values for Model Parameters

#	Variables	Default Value
1	Total number of contents, C	2000
2	Number of A-UAVs, N_A	4
3	Number of MF-UAVs, N_{MF}	8
4	A-UAV's Cache space (content count), C_A	200
5	MF-UAV's Cache space, C_{MF}	25
6	Poisson request rate parameter, μ (request/sec)	1
7	Hover rate of MF-UAV, $R_{Hover} = T_{Hover}/T_{Trajectory}$	$1/6$
8	Transit rate of MF-UAV, $R_{Transit} = T_{Transit}/T_{Trajectory}$	$1/12$
9	Zipf parameter (Popularity), α	0.4
10	Micro Ferrying UAV Trajectory	Round-robin

In the simulation, the impact of lateral link range on content dissemination has been implemented. An MF-UAV begins serving content upon entering the WiFi transmission range of a community, even before reaching its boundaries. The duration during which the MF-UAV starts transmitting content, denoted as Δt_{comm}, is influenced by its transit speed. If Δt_{comm} is significantly shorter than the Poisson-distributed content request generation time (T_{req}), the adjusted hover time remains approximately the same ($\hat{T}_{Hover} \approx T_{Hover}$). Conversely, if Δt_{comm} is comparable to or exceeds T_{req}, the adjusted hover time increases to $\hat{T}_{Hover} \approx T_{Hover} + \Delta t_{comm}$, while the transit time decreases to $\hat{T}_{Transit} \approx T_{Transit} - \Delta t_{comm}$.

The performance of the designed mechanism was evaluated using the following metrics:

Content Availability (P_{avail}): This is the ratio of cache hits to generated requests within a tolerable access delay. Cache hits refer to content provided to users from the UAV-cached content without needing a download. Content availability indirectly reflects the content download cost of the system.

Cache Distribution Optimality (CDO): This metric assesses the optimality of the learned caching policy in terms of the caching sequence. The Jaro-Winkler Similarity (JWS) [143] measures CDO by comparing the similarity between the content sequence from the learned caching policy and the cache pre-loading sequence. It considers the number of matches, required transpositions, and prefix similarity of both sequences. A normalized similarity measure, where 1 indicates optimal caching and 0 indicates non-optimal caching, is used.

Access Delay (AD): Performance of FedMAB model and selective caching policy for micro-ferrying UAVs is also evaluated based on the access delay which is the end-to-end delay between the generation of content request and its provisioning from the cached contents in the UAVs. This chapter reports the epoch-wise average access delay to show the improvement in caching policy as learning progresses.

9.7.1 Effect of Controlled Latency Induced Federated Learning on Content Availability

To understand the applicability of the designed FedMAB-based caching policy along with selective caching, experiments were conducted with different durations of controlled latency. This is achieved with caching policies learnt through models that update with different levels of latency. Each MAB model uses a hybrid exploration strategy including both UCB and ϵ-greedy, where the degree of exploration is set to $\alpha_u = 2$. Also, to show the effectiveness of selective caching at micro-ferrying UAVs (MF-UAVs), *TAD* Ratio R_{TAD} for contents $\{51 - 75\}$ are kept lower than the default R_{TAD} i.e., $1/8$. To be noted that *TAD*s are represented as a ratio with respect to trajectory time ($T_{Trajectory}$) to ensure generalizability of the designed algorithms. Figure 9.7 shows the convergence behavior of the learnt caching policy with FedMAB model at the A-UAVs, and selective caching at the MF-UAVs. The comparison emphasizes on the effects of controlled latency.

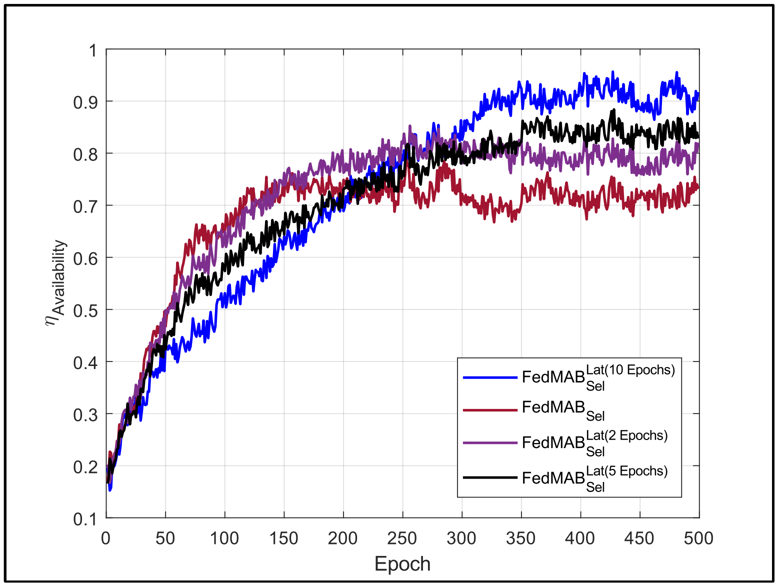


Figure 9.7. Increase in content availability by controlling the learning latency in Federated Learning aided caching policy

The convergence behavior is shown in terms of relative content availability, which is the ratio between content availability achieved using the designed method and the deterministic baseline method from Eqns. 9.13-9.17. The key outcomes are given below. First, the best content availability achieved is with the maximum induced latency while implementing FedMAB to learn caching policy. This parameter controls the application of divergence-based weight computation, eventually the aggregation of the *Top-k* MAB (refer Eqns. 9.24-9.29 and Algorithm 9.4). Second, the promptness in learning behavior is more apparent in the models with least latency or no latency. However, the converged learning performance is subpar, and it can be seen via the attained content availability. Third, the learning progression is inversely proportional to the controlled latency for aggregation, whereas the learning performance is directly proportional to it. For least controlled latency, the individual model's epoch-wise reward estimate $\mathbb{E}[\mathbb{R}] \neq \mathbb{r}^*$ is weak due to limited content requests experienced within an epoch's duration. Here, $\mathbb{R}$ is the reward received during t^{th} epoch and $\mathbb{r}^*$ is the true reward. Also, due to the mobility of the MF-UAVs, the accessibility of ferry and global content availability information can't be guaranteed, leading to a weak and sensitive estimated reward. On the contrary, with a high controlled latency, the individual model's reward is substantially stable i.e., $\mathbb{E}[\mathbb{R}] \approx \mathbb{r}^*$. Additionally, due to the induced latency for model aggregation, the content availability information from adjacent communities can be accessed via MF-UAVs with high likelihood. This leads to a better overall reward estimate, therefore improving content caching policy. However, the explicit introduction of latency to the learning algorithm makes the model update process sluggish, which can be seen in Figure 9.7.

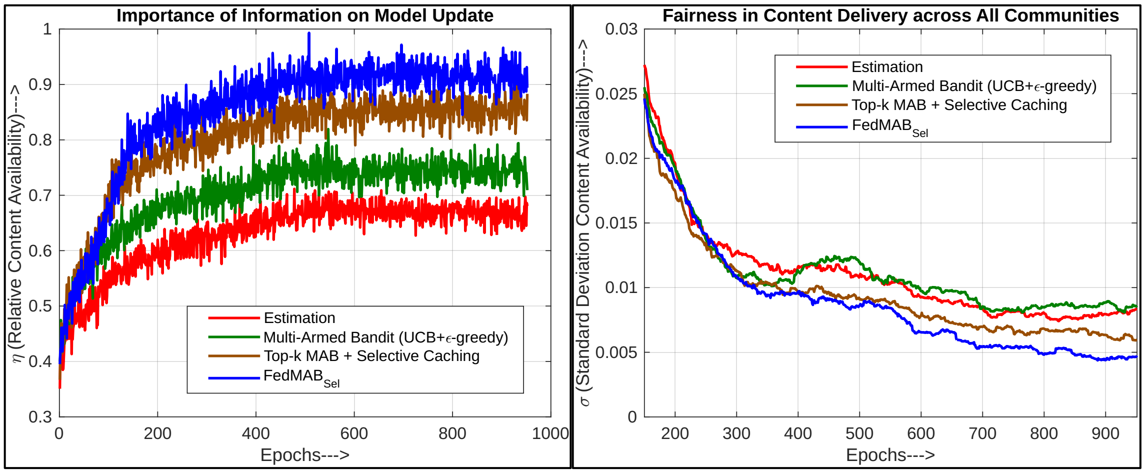


Figure 9.8. (Left) Evolution of learning-based caching policy with information sharing; (Right) Uniformity of performance at all A-UAVs

9.7.2 Evolution of Learning based Caching Policies and their Impacts

The evolution of the learning-based caching policies designed in this chapter and their comparison are shown in terms of relative content availability. The observations from Figure 9.8 are as follows. First, the figure shows that by employing FedMAB model along with selective caching, a caching policy can be learnt which can provide content dissemination performance closer to the benchmark performance [93]. The benchmark performance, using Value-based Caching, is calculated with the aid of *apriori* information on content popularity and takes into consideration the heterogeneity in user demand (see Eqns. 9.9-9.12). The designed FedMAB algorithm is able to leverage the multi-dimensional reward structure and divergence-based weighted aggregation to account for heterogeneity [146], [147], as explained in Eqns. 9.19-9.29, to learn the caching policy on-the-fly (see Section 9.6.1 and 9.6.2). Second, the selective caching policy at micro-ferrying UAVs leverages the shared information between themselves and with the A-UAVs to boost the content availability closer to the benchmark performance by approximately 20% (see Figure 9.8). It utilizes the currently visiting A-UAV's caching information and the preceding MF-UAV's caching decision to algorithmically select its own contents for caching, which is also shown in Figure 9.6. Such selective caching will reduce the redundancy of multiple copies of the same content available through multiple sources at the same time. Third, the difference in the efficacy and limitations of selective caching can be observed in Figure 9.7 and 9.8, where caching decisions at MF-UAVs differ due to the model aggregation in both scenarios. The effectiveness of controlled latency can be seen here in Figure 9.8, where the benefits of divergence-based weighted aggregation is preserved along with leveraging the pros of selective caching. Fourth, when the agent uses UCB exploration strategy, during the initial learning epochs the content availability increases promptly due to high upper confidence value of all contents, which avoids excessive exploitation. This is due to low sampling of requests. As learning progresses, the sparse request for unpopular contents keeps the upper confidence value high which maintains consistent exploratory behavior. Figure 9.8 shows that such exploration strategy alone helps to boost the content availability closer to the benchmark performance by approximately 10% more than popular estimation-based methods [78], [79], [80], [81]. Finally, the standard deviation across the performances of all A-UAVs is recorded, which shows the progression of the learning-based caching policy. Note that $\text{FedMAB}_{\text{Sel}}$ shows lowest standard deviation, which shows highest level of fairness in the performance. Here, the σ is computed as average of 150 learning epochs. Also, contrary to the performance behavior of the MAB algorithm with hybrid action selection strategy, it shows more nonuniform increase in performance with respect to estimation-based methods. This can be attributed to intermittent accessibility of MF-UAVs, therefore limiting information access. This behavior is not seen in both learning variants with selective caching as the caching information is spanned across multiple MF-UAVs.

Discussion: $\text{FedMAB}_{\text{Sel}}$ allows maximum evolution of the caching policy such that increased TAD is leveraged to achieve highest content availability. The performances of *Top-k* MAB with Selective caching and multi-dimensional reward structure follows in that order. These observations can be used to deepen the understanding of the components of $\text{FedMAB}_{\text{Sel}}$. With high TAD, the ferrying and global reward i.e., $\mathbb{R}(i, \mathcal{F})$, and $\mathbb{R}(i, \mathcal{G})$ respectively, brings the estimated reward $\mathbb{R}_T$ closer to the true mean, as it allows more time for the MF-UAVs to transit before the request expires. Furthermore, the effectiveness of the selective caching algorithm boosts with high TAD, since it allows more MF-UAVs to collaborate allowing them to avoid caching copies of same contents amongst themselves. The last component of $\text{FedMAB}_{\text{Sel}}$, that is the divergence-based weighted updates of the models allows each A-UAV to have explicit knowledge of the content popularity at adjacent communities, therefore avoiding content replication at A-UAVs. The consolidation of these components of $\mathrm{FedMAB_{Sel}}$ results in increased content availability with the primary objective of this work. Note that a high value of TAD also allows for unconstrained application of controlled latency, which proves to be beneficial in boosting content availability as shown in Figure 9.7 and 9.8.

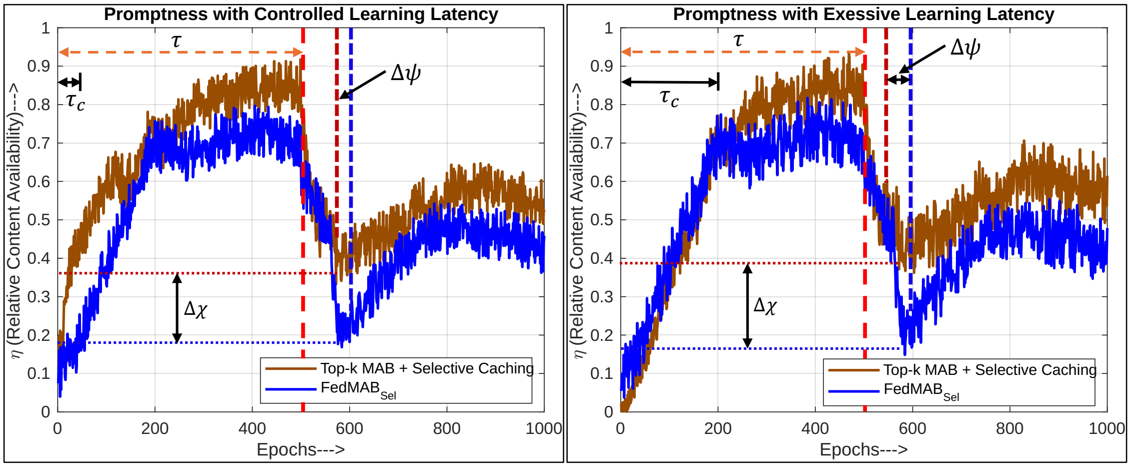


Figure 9.9. Balance between reactiveness and performance of $FedMAB_{Sel}$ caching policy in case of time-varying user preferences

9.7.3 Adaptability with Changing User Preferences

The adaptability of the developed learning-based caching mechanisms is further emphasized in Figure 9.9. It showcases that the ability of $\mathrm{FedMAB_{Sel}}$ approach to learn the caching policy in a setting where the user preference changes over time. It goes on to highlight the reactive nature of the $\mathrm{FedMAB_{Sel}}$, where content availability increases more promptly compared to the standalone *Top-k* MAB implementation or any of its predecessors. Note that dynamic user preference patterns are simulated using Smith-Waterman Distance-based sequence swapping [125] and changing the Zipf parameter (refer Eqn. 9.1). Moreover, the comparison between the reactiveness of the learning-based caching polices are depicted in terms of 3 different measures, namely reactiveness time (ψ), lowest performance point (χ) and crossover ratio (ζ). Reactiveness time (ψ) captures the time taken for the system to start improving its performance after the demand scenario change. χ represents the lowest point in performance after the demand scenario change, just before the system begins to recover. Crossover ratio (ζ) represents the ratio of the time before which one algorithm's performance surpasses another (i.e., $\tau - \tau_c$), relative to the time constant (τ which is the duration of the fixed demand scenario). Here, τ_c refers to the time when an algorithm's performance surpasses another. Therefore, crossover ratio ζ can be expressed as $\zeta = \frac{\tau - \tau_c}{\tau}$. For interpretability, the case where performance of an algorithm doesn't surpass its predecessor, $\tau_c = \tau$. This indicates that there is no relative improvement in performance within the time constant τ. The performance seen with $\text{FedMAB}_{\text{Sel}}$ exhibit relatively lower values for ψ and higher values for χ with any level of controlled latency in $\text{FedMAB}_{\text{Sel}}$. This indicates the promptness of the developed caching method as compared to its predecessors. Crossover ratio ζ, on the other hand, shows a more nuanced observation. For controlled latency of 2 epochs, ζ is high but it reduces for latency of 10 epochs, although with improved relative performance. This shows that a high controlled latency in divergence-based weighted updates for $\text{FedMAB}_{\text{Sel}}$ can improve performance significantly, but it comes with a cost of the model's reactiveness. Therefore, a realistic assumption on the dynamic nature of the content demand pattern suggests that for user preferences with high time constant τ, the reactiveness of the $\text{FedMAB}_{\text{Sel}}$ is relatively high as compared to the learning-based caching mechanisms discussed above.

Note that while the Zipf model is used to represent content request generation patterns, the designed Federated Multi-Armed Bandit (FedMAB) framework remains agnostic to the specific distribution governing user requests. The caching decisions are purely data-driven that relies on observed request patterns rather than prior assumptions about the underlying distribution. If content requests were generated according to an alternative distribution, such as Normal, T, or

Beta, the learning process would naturally adjust caching policies to match the observed demand structure. The adaptability of FedMAB ensures that the system remains effective under diverse content popularity dynamics, therefore optimizing caching decisions based on real-world user behavior rather than a predefined statistical model.

Discussion: The choice of *TAD* for the experiments is such that it is less than the hovering and transiting duration together. This is done to emphasize the reactive nature of the algorithm by constraining the allowed duration for a request, before it is served via download. It should be highlighted that keeping the *TAD* too high allows the MF-UAVs to reduce the caching frequency of those contents. On the contrary, for very low *TAD*, the model overestimates the value of those contents leading to them being cached at A-UAVs allowing ready availability.

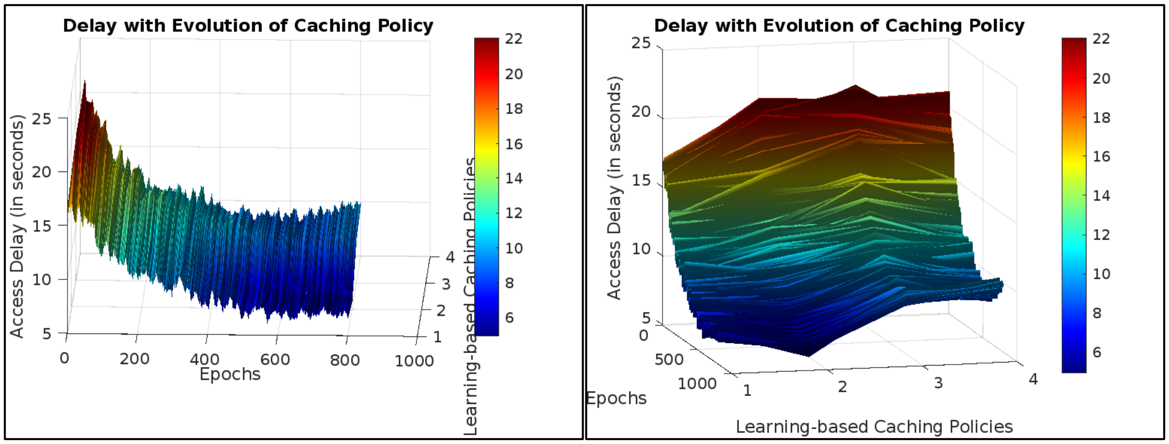


Figure 9.10. Access delay as a determinant for the choice of learning-based caching policy (Two viewing perspective)

9.7.4 The Interplay Between Learning Latency and Content Access Delay

The choice of learning-based caching policy with respect to the access delay has been highlighted in Figure 9.10. This figure emphasizes on the various components, namely multidimensional reward structure, selective caching and divergence-based weighted aggregation, the amalgamation of which leads to the proposed $\text{FedMAB}_{\text{Sel}}$ caching policy. Additionally, it also scrutinizes the behavior of these components under the influence of shared information in varying learning-based caching scenarios. The observations are as follows. First, both Figure 9.8 and 9.10 shows that with increase in shared information content availability and access delay improves irrespective of the caching policy used. However, the efficacy of different versions of the learning-based caching policies varies. Second, Figure 9.10 demonstrates the effect on access delay while applying different versions of the learning-based caching policy as learning progresses. It can be seen that for each learning-based caching policy the delay decreases with increase in epochs, which is intuitive. Along with providing more contents from the hierarchical UAV-aided content dissemination system, the delay decreases since more relevant contents are stored at A-UAVs. Third, with the implementation of multi-dimensional reward structure and the hybrid action selection strategy, access delay decreases, since the relevance of the contents cached in both A-UAVs and MF-UAVs improve. Finally, it can be that as the learning-based caching policy evolves the access delay reduces till a certain point. However, an increase in delay can be seen for $\mathrm{FedMAB}_{\mathrm{Sel}}$ based policy. The reasons are multifaceted. When the model evolves from standalone MAB to *Top-k* MAB with multi-dimensional reward structure, the caching policy of the A-UAVs improve leading to high value content being cached at A-UAVs. This results in better content being ferried via the MF-UAVs without adding significant delay. $\mathrm{FedMAB}_{\mathrm{Sel}}$ along with improving the caching policy for A-UAVs improves the policies for MF-UAVs jointly, which allows more content to be ferried from adjacent communities before exhausting the requests lifetime. This increases the dependance on the hierarchical UAV-aided dissemination system for content provisioning. Therefore, an increase in access delay is observed along with a boost in content availability, which is the primary objective of this work (refer Section 9.4).

Discussion: The designed framework inherently mitigates latency through a combination of controlled learning updates, demand-aware caching policies, and dynamic UAV coordination. As detailed in Algorithm 9.4, the system regulates the timing of federated model updates to balance responsiveness and caching stability which ensures that UAVs do not prematurely overwrite cached content while maintaining adaptability to real-time requests. The incorporation of Tolerable Access Delay (*TAD*) constraints ensures that content with lower latency requirements is prioritized at A-UAVs, while content with more relaxed delay tolerance is efficiently transferred via MF-UAVs.

Additionally, high-demand conditions naturally strengthen the learning process of the Federated Multi-Armed Bandit (FedMAB) framework. Increased accessibility of MF-UAVs in such scenarios exposes the model to a broader range of content requests which allows it to refine caching decisions based on diverse demand patterns. However, local high-demand conditions are not entirely dependent on MF-UAVs; each UAV continues learning independently to track sudden surges in requests. This ensures that the personalized Q-table remains highly reliable, as increased demand inherently leads to higher sampling, which is particularly beneficial for combinatorial bandits. This greater sampling rate allows UAVs to gain stronger confidence in learned content priorities, improve caching efficiency via high confidence *contribution factor* $\mathfrak{C}_{\mathbb{X},\mathbb{Y}}$ and reduce overall latency.

The results in Figure 9.7 and 9.10 empirically validate these latency mitigation mechanisms. By leveraging controlled federated learning updates, demand-aware caching, and increased sampling during high-demand periods, the system effectively maintains low access delays under varying demand conditions that ensures robust performance in real-world UAV-assisted content dissemination scenarios.

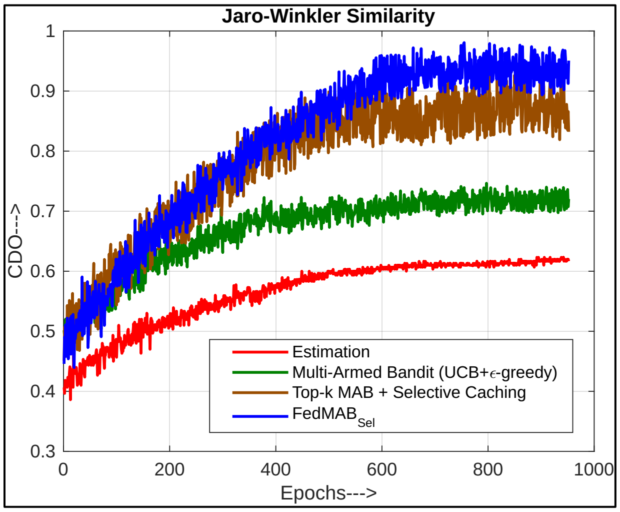


Figure 9.11. Learnt cached content sequence's similarity with benchmark sequence

9.7.5 Cache Similarity of Learnt Sequence with Best Sequence

The effects of learning on the cached content sequence are demonstrated in Figure 9.11. It plots Cache Distribution Optimality (*CDO*) of the cached content sequences for all the A-UAVs in terms of Jaro-Winkler Similarity (*JWS*). The key observation are as follows. First, the average CDO between the benchmark caching sequence from cache pre-loading policy (see Section 9.5) and the cached content sequences learnt by the FedMAB agents at A-UAVs converge near 0.95, with relatively less variance with respect to its *Top-k* MAB predecessor. Physically, this represents higher degree of similarity after convergence, where 1 indicates complete similarity and 0 implies no similarity. Second, the cached contents improve over epochs as learning progresses. Lower CDO values after the initial epochs signify that the A-UAVs have no *a priori* local or global content popularity information. As the MAB agents learn over multiple epochs of generated content requests, the cached contents in the A-UAVs become increasingly similar to the optimal caching sequence, which, in turn, improves the efficacy of FedMAB. Third, CDO is an indirect representation of the storage segmentation factor (λ), which is used to decide the segment sizes according to cache pre-loading policies [91], [142]. A higher CDO implies that, along with learning, the caching policy, the FedMAB agents learn to emulate the said segmentation behavior. Finally, the partial dissimilarity of the cached content sequence can be ascribed to the uncertainty (or regret) associated with the Q-values of contents with low popularity. Also, this leads to an oscillatory convergence of CDO for the A-UAVs.

The impacts of selective caching at micro-ferrying UAVs can be distinctly seen in Fig 9.6. Selective caching at the MF-UAVs along with *Top-k* MAB caching agent at A-UAVs leads to a CDO of nearly 0.9, although with a certain variance. Note that this depends on effective caching capacity of the MF-UAVs, which is dictated by the TADs associated with content requests and the MF-UAVs visiting frequency at A-UAVs (refer Algorithm 9.3). The dependance of contents' Q-values on such information also adds to the post-convergence oscillation. Such oscillatory uncertainties are mitigated by the FedMAB, which enhances the value difference between in-demand and low demand contents, therefore improving the expect reward. To be noted that for the computation of CDO, the benchmark caching sequence is derived by considering the same effective caching capacity as the selective caching algorithm at the micro-ferrying UAVs.

9.8 Conclusion

In this chapter, we design a micro-UAV-assisted content dissemination system that learns caching policies on the fly without prior knowledge of content popularity. Two types of UAVs are introduced for content provisioning in disaster or war-stricken scenarios; anchor UAVs and micro-ferrying UAVs. Cache-enabled anchor UAVs are stationed at each stranded community of users to provide uninterrupted content delivery, while micro-ferrying UAVs act as content transfer agents between the anchor UAVs.

To overcome the limitations of existing caching methods, we introduce a decentralized Federated Multi-Armed Bandit (FedMAB) learning-based caching policy. This method leverages the collective intelligence of all A-UAVs to increase promptness in learning the caching policy, while reducing the redundant copies of the contents across the network. The policy at each A-UAV learns the caching decisions dynamically by maximizing an estimated multi-dimensional reward aimed at increasing both local and global content availability. Our results show that the FedMAB learning-based caching policy achieves approximately 88% of the maximum achievable content availability.

To further improve the Q-value estimates, we implement a Selective Caching Algorithm at the micro-ferrying UAVs. This method leverages shared information between anchor UAVs and micro-ferrying UAVs to further reduce the redundant content copies and provide a better estimate of the most popular content within a community. Combining selective caching at micro-ferrying UAVs with the FedMAB learning-based caching policy at anchor UAVs boosts content availability to approximately 94% of the maximum achievable level. With the designed caching policies, a scaled-up micro-UAV-assisted network is shown to attain content availability close of the maximum achievable content availability.

Discussion and Future Work: Future work includes developing algorithms to handle time-varying content popularity and implementing adaptive trajectory planning to address operational unreliabilities of the UAVs. Additionally, it is necessary to explore methods for preserving the richness of information when converting multi-modal disaster data into smaller-sized formats to enhance effective content caching capacity.

While the developed framework has been validated through simulations, real-world deployment would require addressing practical challenges that includes UAV coordination, wireless interference, and energy constraints. The decentralized structure of the system mitigates coordination complexity by allowing A-UAVs and MF-UAVs to operate autonomously which leverages federated learning to refine caching strategies without requiring continuous central control.

A transition to operational implementation would follow a phased approach. Initial deployment could involve small-scale testbed experiments, where UAVs execute FedMAB-based caching policies in controlled environments. This would allow for real-world validation of caching performance under dynamic network conditions. The next phase would involve field trials in real-world UAV-assisted networks, where interference, energy efficiency, and UAV mobility constraints could be evaluated. Future extensions could also explore hardware integration with UAV control algorithms which can ensure that caching decisions align with real-time flight dynamics.

By adopting a structured deployment roadmap, the designed framework can be progressively refined for real-world applications while maintaining its efficiency in adaptive caching.

Chapter 10: Conclusions and Future Works

This thesis investigates the complex problem of content dissemination in environments that lack communication infrastructure due to natural disasters or conflicts. To overcome this challenge, the research introduces a hierarchical UAV-assisted content dissemination framework designed to provide essential information to isolated communities. The proposed architecture incorporates Anchor UAVs (A-UAVs), which provide stationary caching points with costly satellite-like backhaul connectivity, and Micro-Ferrying UAVs (MF-UAVs) that transfer cached contents across different communities without direct backhaul connections.

A central aspect of this thesis is the use of Multi-Armed Bandit (MAB) and Federated Multi-Armed Bandit (FedMAB) learning methodologies to dynamically and adaptively cache contents. Unlike traditional caching strategies that rely on static or globally known content popularity, the proposed model captures local content demands and temporal variations without requiring centralized coordination. FedMAB enables UAVs to collaboratively enhance their caching policies by sharing learned models rather than user requests directly, thus ensuring a scalable learning framework.

The research further extends the MAB framework by integrating trajectory-aware caching policies. This approach considers UAV mobility patterns and content request dynamics to determine optimal caching strategies. By effectively leveraging trajectories, the system significantly improves content delivery efficiency, reduces redundancy in content storage, and increases overall availability of critical information to users.

Another important contribution is the Selective Caching Algorithm developed specifically for MF-UAVs. This algorithm strategically selects and manages cached contents to maximize their availability while minimizing redundant storage across UAV fleets. Through careful evaluation, this method demonstrated substantial improvements in caching efficiency and system performance compared to traditional pre-loading benchmarks.

Extensive simulations were conducted to validate the effectiveness of the proposed approaches against established benchmark models, including other traditional caching policies. Results indicated that the amalgamation of federated learning, multi-armed bandits and trajectory-aware caching policies significantly outperform traditional approaches that achieves high levels of content availability, reduced access delay, and improved caching stability under varying user demand patterns.

This research delivers a practical and robust solution for UAV-based content dissemination, effectively addressing the critical challenges posed by infrastructure-deprived environments. By incorporating advanced federated learning techniques, bandit algorithms, adaptive caching, and trajectory-awareness, the developed framework ensures resilience, scalability, and responsiveness. The methodologies and algorithms established through this thesis lay the groundwork for future research in adaptive UAV systems which emphasizes a balanced approach between operational efficiency, learning responsiveness, and strategic content management.

10.1 Key Findings and Design Guidelines

Given below are the essential core ideas that can be deduced from the results presented in this thesis.

a) Federated Model Aggregation Enhances Scalability: The integration of Federated Multi-Armed Bandit learning enables collaborative policy refinement across multiple UAVs without relying on centralized coordination. This allows each A-UAV to build locally relevant models while benefiting from global content popularity trends shared through MF-

UAV-based model aggregation. The system achieves rapid convergence of caching policies which makes it suitable for large-scale and dynamic disaster-response networks.

b) Mobility-Aware Caching Improves Efficiency: Considering UAV trajectory patterns and user request dynamics substantially enhances content delivery performance. Caching decisions informed by UAV mobility trajectories reduce redundancy in content storage and improve overall content availability. To optimize system performance, UAV caching algorithms should dynamically adapt based on real-time trajectory data and content demand predictions. By aligning caching decisions with known or predicted UAV flight paths, the system maximizes content exposure to target communities and minimizes missed delivery opportunities.

c) Selective Caching at MF-UAVs Optimizes Network Utility: The Selective Caching Algorithm at MF-UAVs strategically curates content based on urgency, request likelihood, and caching history. This approach improves effective cache utilization across UAV swarms that minimizes overlap in stored content and increases system-wide availability. When deployed at scale, this mechanism ensures maximum diversity and relevance of cached items.

d) Multi-Dimensional Reward Structures Enable Contextual Adaptation: The use of local, ferrying, and global reward components in caching decision-making leads to finely-tuned learning behavior. This structure allows UAVs to dynamically adapt to regional content demand variations and system-wide utility metrics. It supports learning policies that are simultaneously locally responsive and globally efficient.

e) Hierarchical Architecture Increases Robustness: A two-tier UAV network architecture comprising A-UAVs and MF-UAVs ensures operational continuity even under constrained conditions. The hierarchical structure simplifies learning coordination, offloads content ferrying, and enhances fault tolerance. As a result, the system scales robustly with minimal central infrastructure.

f) Dynamic Adaptability Under Changing Conditions: The proposed federated multi-armed bandit and trajectory-aware caching solutions consistently outperform traditional static and centralized methods, particularly in unpredictable and dynamic scenarios. Emphasizing dynamic adaptability allows the system to swiftly adjust caching strategies in response to shifting user demands and environmental changes that maintains high system performance and reliability.

g) Balanced Optimization Across System Constraints: The framework accounts for trade-offs among QoS, computation, communication cost, cache capacity, and UAV budgets. Learning-driven caching policies help balance content relevance against delivery feasibility which ensures that the system operates within resource bounds while maintaining high service quality.

10.2 Future Directions

10.2.1 Crowd Estimation-Based Context-Aware Caching

This direction explores a context-aware caching mechanism that leverages crowd estimation and environmental sensing to guide content placement decisions. Traditional MAB approaches rely solely on observed request frequency, without accounting for the urgency, relevance, or situational context of content demand. In disaster-stricken environments, such limitations can significantly impair content availability when observational data is sparse or missing.

The proposed solution integrates multimodal context extraction using techniques such as image captioning (e.g., CLIP), crowd estimation models like auxiliary point guidance, and structured image analysis to infer user and environmental conditions from onboard UAV camera sensors. These multimodal signals are processed to generate real-time context scores that encapsulate content urgency, relevance, damage severity, and inferred user intent.

A context scoring model is developed to estimate content demand when explicit request data is unavailable. For example, crowd density estimation or disaster damage detection informs content selection priorities when direct user input is missing. These context-driven metrics are then integrated into the Federated Multi-Armed Bandit (FedMAB) framework to dynamically update caching strategies based on environmental insights.

The proposed system includes a pipeline for population and popularity estimation, and temporal decay modeling that balances short-term surges in demand with long-term value retention. This helps UAVs adapt their caching policy in response to changing real-world conditions.

By incorporating multimodal insights into the reward formulation and decision logic, the caching framework aligns delivery priorities with real-world urgency. The use of Federated Learning enables distributed UAVs to share model parameters without central coordination which ensures that learning remains adaptive and robust in communication-challenged environments. This approach opens new avenues for deploying intelligent, context-aware caching in extreme scenarios where traditional demand estimation falls short.

10.2.2 Large Action Models (LAM) for Enhanced Decision Sampling and Strategic Caching

This future direction proposes integrating Large Action Models (LAM) with Federated Multi-Armed Bandit (FedMAB) learning frameworks to enhance the decision-making processes in UAV-assisted content dissemination systems. Traditional MAB methods typically deal with a limited set of discrete actions which constrains the ability to efficiently explore and exploit a vast action space in highly dynamic environments. LAM addresses this limitation by generating extensive and diverse action spaces based on historical data, simulation outcomes, and predictive modeling.

Incorporating LAM with FedMAB can significantly expand the scope and precision of caching policies by enabling UAVs to sample and assess a broad spectrum of potential actions rapidly. The system can dynamically construct and evaluate large corpora of actions for various scenarios, such as sudden shifts in content popularity, unexpected UAV failures, or rapid environmental changes. By combining predictive analytics from machine learning models, such as Transformer-based predictors, with LAM-generated action spaces, UAVs can proactively formulate and test multiple strategic scenarios before deployment.

Furthermore, the integration of federated learning within LAM allows UAV networks to collaboratively refine their large-scale action databases without compromising local autonomy. Each UAV contributes insights based on local context and outcomes that enhances collective decision-making accuracy. This collective intelligence enables the system to rapidly identify optimal actions from a comprehensive and evolving action repository, thus significantly enhancing the robustness and adaptability of content dissemination policies.

This innovative approach to large-scale decision sampling facilitates more accurate, proactive, and resilient caching strategies. Future research will explore algorithmic advancements to optimize action space generation and evaluation, along with scalability improvements to support extensive deployments in real-world, resource-constrained disaster-response scenarios.

10.2.3 Contextual Federated Multi-Armed Bandit Learning for Collective Caching

This direction of the thesis extension seeks to design and implement a novel caching policy framework for a swarm of micro-ferrying unmanned aerial vehicles (UAVs) which utilizes Contextual Federated Multi-Armed Bandit Learning. The primary focus is on maximizing content availability through a collective caching strategy that intelligently adapts to the heterogeneous and time-varying demands of user content requests, which follow a Zipf popularity distribution. By integrating contextual variables derived from the UAVs' operational environment, such as their flight patterns and proximity in formations, this framework aims to enhance the efficiency and responsiveness of content delivery networks in diverse scenarios.

At the core of the suggested system is the development of a caching algorithm tailored to manage and exploit the complexities of a multi-UAV system characterized by a hierarchical structure UAVs. The caching algorithm will be designed to continuously learn and adapt to the operational context of the UAV swarm. By observing the trajectories and grouping patterns of the UAVs, the algorithm will adjust the cached content on each UAV to minimize redundancy and ensure a diverse range of data is available across the network. This approach leverages the inherent characteristics of the UAVs' flight routes and community engagement patterns that makes it possible to tailor content delivery to specific regional demands dynamically.

Furthermore, the proposed caching policy will incorporate a multivariate Contextual Federated Multi-Armed Bandit (CFMAB) learning model that utilizes a complex aggregation of Q-values derived from multiple UAVs operating in concert. This model will allow for the sharing and updating of multivariate model information across the swarm without significant delays, facilitating a responsive and adaptive system capable of handling non-independent and identically distributed (non-*IID*) data scenarios [148]. The use of a joint distribution-based divergence mechanism in the CFMAB model will help to synchronize and optimize the caching decisions across the UAV swarm which will enhance the overall system's efficiency and effectiveness.

An essential component of this framework is its ability to balance the trade-off between the effective caching capacity of the UAV fleet and their accessibility. This balance is critical to maintaining high levels of service quality, especially in terms of meeting the QoS expectations and tolerable access delays specified by users. The caching policy will be continuously refined through the CFMAB learning process, which will study the interactions between the learnt caching policies and the QoS expectations. Such continuous learning will enable adapting the network's behavior to optimize content delivery based on actual user experiences and feedback.

The proposed caching policy framework for the UAV swarm will utilize advanced contextual muti-armed bandits and federated learning techniques to dynamically adapt to changing environmental and user demand conditions. By addressing the challenges of multivariate, non-*IID* data in a real-world application [149], this research will not only enhance the performance of UAV-based content distribution networks but also contribute significant insights and methodologies to the field of distributed machine learning and autonomous vehicle coordination.

10.2.4 Adaptive Trajectory Planning in the Presence of Operational Unreliabilities of the Micro UAVs

The proposed initiative to enhance the effectiveness of unmanned aerial vehicles (UAVs) in content distribution by addressing the issue of operational unreliabilities requires an integrated approach that utilizes both adaptive trajectory route planning and advanced caching techniques. This approach aims to refine the current framework, which primarily focuses on predetermined trajectories for UAVs and Micro-UAVs, by incorporating dynamic decision-making capabilities that respond to fluctuations in operational stability.

The central objective of this enhancement is to implement adaptive trajectory route planning that compensates for the unreliability of Micro UAV operations. This entails the design of a dynamic flight plan that not only adjusts in real-time to the status and availability of UAVs within the network but also ensures that the collective storage and dissemination capabilities of the fleet are not compromised due to individual UAV failures.

At a higher level, the primary goal of this development is to ensure that the network of UAVs can maintain high levels of content availability and reliability even when individual units fail or deviate from their expected operational parameters. This is achieved by dynamically modifying the caching policies and trajectory plans of the remaining UAVs. Such adjustments are crucial not only for optimizing the distribution of content across the affected areas but also for ensuring that the collective capabilities of the UAV swarm are utilized efficiently. Furthermore, the design of the flight plan also plays a critical role. It must be flexible enough to allow for real-time reconfiguration which can enable UAVs to regroup or reposition themselves effectively to cover potentially uncovered areas due to the failure of one or more units. This strategic regrouping is expected to maximize the content availability by leveraging the undiminished parts of the network to compensate for the affected segments.

Through the integration of these methodologies, the framework aims to provide a resilient, scalable, and highly adaptive solution to the challenges posed by the operational unreliability of Micro UAVs in disaster-stricken or isolated regions, thus enhancing the effectiveness of UAV-assisted communication and content dissemination systems.

10.2.5 Integration of Proactive and Predictive Caching with Federated Multi-Armed Bandit Learning

In the proposed framework for integrating proactive and predictive caching methods with Federated Multi-Armed Bandit Learning (FMABL) or Contextual Federated Multi-Armed Bandit (CFMAB) learning for caching in UAV/Micro-UAV networks, we aim to transition from traditional reactive caching strategies to a more predictive model. Traditional methods primarily respond to immediate content requests, gradually improving the caching decisions based on received user feedback and the associated rewards. These systems, while adaptive, typically initiate improvements only after the content requests are made by UAVs or Micro-UAVs. Such reactive approaches can limit the efficiency and responsiveness of the network, especially in dynamic and fast-evolving operational environments.

The principal objective of this new proposal is to implement Transformer Learning algorithms [150], [151] to anticipate changes in content popularity. Transformer Learning models, known for their effectiveness in understanding time-series data and patterns in sequence prediction tasks, will be employed to forecast the fluctuating demands for content [152]. By predicting these changes, the system can proactively adjust caching policies before actual requests occur, potentially enhancing the network's efficiency and user satisfaction.

This proactive approach is augmented by an advanced caching policy developed through MAB/FMABL/CFMAB. The MAB/FMABL/CFMAB framework leverages the decentralized nature of UAV networks to aggregate insights from individual UAV experiences, even in a non-independent and identically distributed (non-*IID*) data environment [145]. This method allows for a dynamic adaptation of caching strategies that are both informed by local conditions and enhanced through a collective learning process. The integration of Transformer Learning predictions with such caching policy constitutes a significant enhancement over existing methods. By forecasting content popularity trends, the Transformer model provides a predictive input that the MAB/FMABL/CFMAB system uses to pre-adjust its strategies. This means that the caching decisions can be refined in anticipation of future demands rather than solely in reaction to past requests. For example, if the Transformer Learning model predicts a surge in demand for specific types of content in a particular area, the FMABL system can proactively direct UAVs to cache this content in anticipation, rather than waiting for the demand to manifest physically.

Moreover, this proactive caching approach will not only improve the timeliness and relevance of the content delivered but also optimizes the network's resources. By reducing the need for rapid, reactive changes in caching strategies, the system can operate more smoothly and efficiently that focuses its computational and communication resources on maintaining optimal service rather than constant adjustment.

In other words, the integration process involves the continuous training of the Transformer model with historical and real-time data to refine its predictive accuracy. Concurrently, the MAB/FMABL/CFMAB framework adjusts its parameters based on both the predictions from the Transformer model and the ongoing feedback from the UAV network. This dual-input system ensures that the caching policy remains robust and adaptable, capable of handling the inherent uncertainties and variability of UAV-assisted content delivery environments. It combines the predictive power of Transformer Learning with the adaptive, decentralized learning capabilities of MAB/FMABL/CFMAB to create a caching system that not only responds to but anticipates user needs and content popularity trends.

BIBLIOGRAPHY

[1] B. Mukherjee, M. F. Habib, and F. Dikbiyik, “Network adaptability from disaster disruptions and cascading failures,” *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 230–238, May 2014, doi: 10.1109/MCOM.2014.6815917.

[2] S. V. Kartalopoulos, “Surviving a disaster [optical communications],” *IEEE Commun. Mag.*, vol. 40, no. 7, pp. 124–126, Jul. 2002, doi: 10.1109/MCOM.2002.1018017.

[3] H. Rong, Z. Wang, H. Jiang, Z. Xiao, and F. Zeng, “Energy-Aware Clustering and Routing in Infrastructure Failure Areas With D2D Communication,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8645–8657, Oct. 2019, doi: 10.1109/JIOT.2019.2922202.

[4] M. Matracia, N. Saeed, M. A. Kishk, and M.-S. Alouini, “Post-Disaster Communications: Enabling Technologies, Architectures, and Open Challenges,” *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1177–1205, 2022, doi: 10.1109/OJCOMS.2022.3192040.

[5] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, “Deep Learning for Radio Resource Allocation With Diverse Quality-of-Service Requirements in 5G,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2309–2324, Apr. 2021, doi: 10.1109/TWC.2020.3041319.

[6] M. El Tanab and W. Hamouda, “Fast-Grant Learning-Based Approach for Machine-Type Communications With NOMA,” in *ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada: IEEE, Jun. 2021, pp. 1–6. doi: 10.1109/ICC42927.2021.9500606.

[7] K.-H. Liu and W. Liao, “Intelligent Offloading for Multi-Access Edge Computing: A New Actor-Critic Approach,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland: IEEE, Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149387.

[8] M. S. M. Gismalla *et al.*, “Survey on Device to Device (D2D) Communication for 5GB/6G Networks: Concept, Applications, Challenges, and Future Directions,” *IEEE Access*, vol. 10, pp. 30792–30821, 2022, doi: 10.1109/ACCESS.2022.3160215.

[9] I. O. Sanusi, K. M. Nasr, and K. Moessner, “Radio Resource Management Approaches for Reliable Device-to-Device (D2D) Communication in Wireless Industrial Applications,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 905–916, Sep. 2021, doi: 10.1109/TCCN.2020.3032679.

[10] D. Shumeye Lakew, U. Sa’ad, N.-N. Dao, W. Na, and S. Cho, “Routing in Flying Ad Hoc Networks: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1071–1120, 2020, doi: 10.1109/COMST.2020.2982452.

[11] M. Maad Hamdi, L. Audah, S. Abduljabbar Rashid, A. Hamid Mohammed, S. Alani, and A. Shamil Mustafa, “A Review of Applications, Characteristics and Challenges in Vehicular Ad Hoc Networks (VANETs),” in *2020 International Congress on Human-Computer*

Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey: IEEE, Jun. 2020, pp. 1–7. doi: 10.1109/HORA49412.2020.9152928.

[12] M. Li, F. R. Yu, P. Si, W. Wu, and Y. Zhang, "Resource Optimization for Delay-Tolerant Data in Blockchain-Enabled IoT With Edge Computing: A Deep Reinforcement Learning Approach," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9399–9412, Oct. 2020, doi: 10.1109/JIOT.2020.3007869.

[13] J. Liang *et al.*, "LTP for Reliable Data Delivery From Space Station to Ground Station in the Presence of Link Disruption," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 38, no. 9, pp. 24–33, Sep. 2023, doi: 10.1109/MAES.2023.3290134.

[14] L. Yang *et al.*, "Resource Consumption of a Hybrid Bundle Retransmission Approach on Deep-Space Communication Channels," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, no. 11, pp. 34–43, Nov. 2021, doi: 10.1109/MAES.2021.3094787.

[15] L. Yang, R. Wang, J. Liang, Y. Zhou, K. Zhao, and X. Liu, "Acknowledgment Mechanisms for Reliable File Transfer Over Highly Asymmetric Deep-Space Channels," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 9, pp. 42–51, Sep. 2022, doi: 10.1109/MAES.2022.3192508.

[16] L. Yang, R. Wang, Y. Zhou, J. Liang, K. Zhao, and S. Burleigh, "An Analytical Framework for Disruption of Licklider Transmission Protocol in Mars Communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5430–5444, May 2022, doi: 10.1109/TVT.2022.3153959.

[17] M. Zhang, M. EI-Hajjar, and S. X. Ng, "Intelligent Caching in UAV-Aided Networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 739–752, Jan. 2022, doi: 10.1109/TVT.2021.3125396.

[18] S. Gu, X. Sun, Z. Yang, T. Huang, W. Xiang, and K. Yu, "Energy-Aware Coded Caching Strategy Design With Resource Optimization for Satellite-UAV-Vehicle-Integrated Networks," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5799–5811, Apr. 2022, doi: 10.1109/JIOT.2021.3065664.

[19] Y. Zhou *et al.*, "Caching and UAV Friendly Jamming for Secure Communications With Active Eavesdropping Attacks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 11251–11256, Oct. 2022, doi: 10.1109/TVT.2022.3186730.

[20] Y. Tian, G. Pan, M. A. Kishk, and M.-S. Alouini, "Stochastic Analysis of Cooperative Satellite-UAV Communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 3570–3586, Jun. 2022, doi: 10.1109/TWC.2021.3121299.

[21] H. Kong, M. Lin, W.-P. Zhu, H. Amindavar, and M.-S. Alouini, "Multiuser Scheduling for Asymmetric FSO/RF Links in Satellite-UAV-Terrestrial Networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 8, pp. 1235–1239, Aug. 2020, doi: 10.1109/LWC.2020.2986750.

[22] J.-H. Lee, J. Park, M. Bennis, and Y.-C. Ko, “Integrating LEO Satellites and Multi-UAV Reinforcement Learning for Hybrid FSO/RF Non-Terrestrial Networks,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3647–3662, Mar. 2023, doi: 10.1109/TVT.2022.3220696.

[23] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, “Air-to-air visual detection of micro-UAVs: An experimental evaluation of deep learning,” *IEEE Robotics and automation letters*, vol. 6, no. 2, pp. 1020–1027, 2021.

[24] A. Mohammadi, Y. Feng, C. Zhang, S. Rawashdeh, and S. Baek, “Vision-based autonomous landing using an MPC-controlled micro UAV on a moving platform,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS*, IEEE, Sep. 2020, pp. 771–780.

[25] K.-B. Kang, J.-H. Choi, B.-L. Cho, J.-S. Lee, and K.-T. Kim, “Analysis of Micro-Doppler Signatures of Small UAVs Based on Doppler Spectrum,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 3252–3267, Oct. 2021, doi: 10.1109/TAES.2021.3074208.

[26] W. Wang *et al.*, “Energy-Constrained UAV-Assisted Secure Communications With Position Optimization and Cooperative Jamming,” *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4476–4489, Jul. 2020, doi: 10.1109/TCOMM.2020.2989462.

[27] Y. Zeng and R. Zhang, “Energy-Efficient UAV Communication With Trajectory Optimization,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017, doi: 10.1109/TWC.2017.2688328.

[28] M. Monwar, O. Semiari, and W. Saad, “Optimized Path Planning for Inspection by Unmanned Aerial Vehicles Swarm with Energy Constraints,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 1–6. doi: 10.1109/GLOCOM.2018.8647342.

[29] J. Gu, H. Wang, G. Ding, Y. Xu, Z. Xue, and H. Zhou, “Energy-Constrained Completion Time Minimization in UAV-Enabled Internet of Things,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5491–5503, Jun. 2020, doi: 10.1109/JIOT.2020.2981092.

[30] “What is the maximum range for a UAV’s remote control?” Accessed: Mar. 26, 2024. [Online]. Available: https://www.linkedin.com/advice/0/what-maximum-range-uavs-remote-control-skills-drones-zdfqc

[31] “Mobile data traffic forecast – Mobility Report.” Accessed: Mar. 26, 2024. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast

[32] “Understanding Average Data Usage Per Month for Home Internet - CompareInternet.com.” Accessed: Mar. 26, 2024. [Online]. Available: https://www.compareinternet.com/blog/average-data-usage-per-month-home-internet/

[33] J. O. Klompmaker *et al.*, “Racial, Ethnic, and Socioeconomic Disparities in Multiple Measures of Blue and Green Spaces in the United States,” *Environ Health Perspect*, vol. 131, no. 1, p. 017007, Jan. 2023, doi: 10.1289/EHP11164.

[34] H. Ritchie, E. Mathieu, and M. Roser, “Which countries are most densely populated?,” *Our World in Data*, Feb. 2024, Accessed: Mar. 26, 2024. [Online]. Available: https://ourworldindata.org/most-densely-populated-countries

[35] Q. Xu *et al.*, “Performance analysis of NVMe SSDs and their implication on real world databases,” in *Proceedings of the 8th ACM International Systems and Storage Conference*, Haifa Israel: ACM, May 2015, pp. 1–11. doi: 10.1145/2757667.2757684.

[36] J. Zhang, F. Meng, L. Qiao, and K. Zhu, “Design and Implementation of Optical Fiber SSD Exploiting FPGA Accelerated NVMe,” *IEEE Access*, vol. 7, pp. 152944–152952, 2019, doi: 10.1109/ACCESS.2019.2947181.

[37] Tanwir, G. Hendrantoro, and A. Affandi, “Early result from adaptive combination of LRU, LFU and FIFO to improve cache server performance in telecommunication network,” in *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Surabaya, Indonesia: IEEE, May 2015, pp. 429–432. doi: 10.1109/ISITIA.2015.7220019.

[38] H. Gomaa, G. G. Messier, C. Williamson, and R. Davies, “Estimating Instantaneous Cache Hit Ratio Using Markov Chain Analysis,” *IEEE/ACM Trans. Networking*, vol. 21, no. 5, pp. 1472–1483, Oct. 2013, doi: 10.1109/TNET.2012.2227338.

[39] S. Maffeis, “Cache management algorithms for flexible filesystems,” *SIGMETRICS Perform. Eval. Rev.*, vol. 21, no. 2, pp. 16–25, Dec. 1993, doi: 10.1145/174215.174219.

[40] X. Wang *et al.*, “Deep Reinforcement Learning: A Survey,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 35, no. 4, pp. 5064–5078, Apr. 2024, doi: 10.1109/TNNLS.2022.3207346.

[41] J. Oh *et al.*, “Discovering Reinforcement Learning Algorithms,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 1060–1070. Accessed: Apr. 26, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/0b96d81f0494fde5428c7aea243c9157-Abstract.html

[42] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, “Multi-Armed-Bandit-Based Spectrum Scheduling Algorithms in Wireless Networks: A Survey,” *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 24–30, Feb. 2020, doi: 10.1109/MWC.001.1900280.

[43] X. Zhou and B. Ji, “On Kernelized Multi-Armed Bandits with Constraints”.

[44] A. Kalvit and A. Zeevi, “A Closer Look at the Worst-case Behavior of Multi-armed Bandit Algorithms,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 8807–8819. Accessed: Apr. 26, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/49ef08ad6e7f26d7f200e1b2b9e6e4ac-Abstract.html

[45] C. Shi and C. Shen, “Federated Multi-Armed Bandits,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, Art. no. 11, May 2021, doi: 10.1609/aaai.v35i11.17156.

[46] D. C. Nguyen *et al.*, “Federated Learning for Smart Healthcare: A Survey,” *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–37, Mar. 2023, doi: 10.1145/3501296.

[47] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, “Federated Learning for Internet of Things: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021, doi: 10.1109/COMST.2021.3075439.

[48] A. K. Bhuyan, H. Dutta, and S. Biswas, “Multi-Armed Bandit Learning for Content Provisioning in Network of UAVs,” in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, Dec. 2023, pp. 1143–1148. doi: 10.1109/GLOBECOM54140.2023.10437463.

[49] A. K. Bhuyan, H. Dutta, and S. Biswas, “Top-k Multi-Armed Bandit Learning for Content Dissemination in Swarms of Micro-UAVs,” Jan. 15, 2025, *arXiv*: arXiv:2404.10845. doi: 10.48550/arXiv.2404.10845.

[50] A. K. Bhuyan, H. Dutta, and S. Biswas, “Towards Federated Multi-Armed Bandit Learning for Content Dissemination Using Swarm of UAVs,” *ACM Trans. Internet Things*, p. 3733841, May 2025, doi: 10.1145/3733841.

[51] A. K. Bhuyan, H. Dutta, and S. Biswas, “Federated Multi-Armed Bandit Learning for Caching in UAV-aided Content Dissemination,” *Ad Hoc Networks*, vol. 151, p. 103306, Dec. 2023, doi: 10.1016/j.adhoc.2023.103306.

[52] A. K. Bhuyan, H. Dutta, and S. Biswas, “Distributed Federated-Multi-Armed Bandit Learning for Content Management in Connected UAVs,” *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 130–136, Dec. 2023, doi: 10.1109/IOTM.001.2300081.

[53] C. Mouradian, N. T. Jahromi, and R. H. Glitho, “NFV and SDN-Based Distributed IoT Gateway for Large-Scale Disaster Management,” *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4119–4131, Oct. 2018, doi: 10.1109/JIOT.2018.2867255.

[54] Y. Liu, F. Zhou, C. Chen, Z. Zhu, T. Shang, and J.-M. Torres-Moreno, “Disaster Protection in Inter-DataCenter Networks Leveraging Cooperative Storage,” *IEEE Trans. Netw. Serv. Manage.*, vol. 18, no. 3, pp. 2598–2611, Sep. 2021, doi: 10.1109/TNSM.2021.3089049.

[55] H. Verma and N. Chauhan, “MANET based emergency communication system for natural disasters,” in *International Conference on Computing, Communication & Automation*, Greater Noida, India: IEEE, May 2015, pp. 480–485. doi: 10.1109/CCAA.2015.7148424.

[56] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe Germany: ACM, Aug. 2003, pp. 27–34. doi: 10.1145/863955.863960.

[57] L. Palen and A. L. Hughes, “Social Media in Disaster Communication,” in *Handbook of Disaster Research*, H. Rodríguez, W. Donner, and J. E. Trainor, Eds., Cham: Springer International Publishing, 2018, pp. 497–518. doi: 10.1007/978-3-319-63254-4_24.

[58] G. Broll, E. Rukzio, M. Paolucci, M. Wagner, A. Schmidt, and H. Hussmann, “Perci: Pervasive Service Interaction with the Internet of Things,” *IEEE Internet Comput.*, vol. 13, no. 6, pp. 74–81, Nov. 2009, doi: 10.1109/MIC.2009.120.

[59] H. Sami, R. Saado, A. E. Saoudi, A. Mourad, H. Otrok, and J. Bentahar, “Opportunistic UAV Deployment for Intelligent On-Demand IoV Service Management,” *IEEE Trans. Netw. Serv. Manage.*, vol. 20, no. 3, pp. 3428–3442, Sep. 2023, doi: 10.1109/TNSM.2023.3242205.

[60] X. Liu *et al.*, “Challenges and opportunities for autonomous micro-UAVs in precision agriculture,” *IEEE Micro*, vol. 42, no. 1, pp. 61–68, 2022.

[61] J. Gago *et al.*, “Nano and micro unmanned aerial vehicles (UAVs): a new grand challenge for precision agriculture?,” *Current protocols in plant biology*, vol. 5, no. 1, p. 20103, 2020.

[62] S. Misra, P. K. Deb, and K. Saini, “Dynamic leader selection in a master-slave architecture-based micro UAV swarm,” in *2021 IEEE Global Communications Conference (GLOBECOM*, IEEE, Dec. 2021, pp. 1–6.

[63] W. Wen, Y. Jia, and W. Xia, “Federated learning in SWIPT-enabled micro-UAV swarm networks: A joint design of scheduling and resource allocation,” in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP*, IEEE, Oct. 2021, pp. 1–5.

[64] N. Zhao, “UAV-assisted emergency networks in disasters,” *IEEE Wireless Communications*, vol. 26, no. 1, pp. 45–51, 2019.

[65] X. Liu *et al.*, “Transceiver Design and Multihop D2D for UAV IoT Coverage in Disasters,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1803–1815, Apr. 2019, doi: 10.1109/JIOT.2018.2877504.

[66] E. M. Mohamed, S. Hashima, and K. Hatano, “Energy aware multiarmed bandit for millimeter wave-based UAV mounted RIS networks,” *IEEE Wireless Communications Letters*, vol. 11, no. 6, pp. 1293–1297, 2022.

[67] A. Amrallah, E. M. Mohamed, G. K. Tran, and K. Sakaguchi, “Optimization of UAV 3D Trajectory in a Post-disaster Area Using Dual Energy-Aware Bandits,” *IEICE Communications Express*, 2023.

[68] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage,” *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, 2016.

[69] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[70] Y. Zeng, "Wireless communications with unmanned aerial vehicles Opportunities and challenges," *Wireless communications with unmanned aerial vehicles Opportunities and challenges IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[71] W. Ejaz, M. A. Azam, S. Saadat, F. Iqbal, and A. Hanan, "Unmanned Aerial Vehicles enabled IoT Platform for Disaster Management," *Energies*, vol. 12, no. 14, Art. no. 14, Jan. 2019, doi: 10.3390/en12142706.

[72] X. Xu, Y. Zeng, Y. L. Guan, and R. Zhang, "Overcoming Endurance Issue: UAV-Enabled Communications With Proactive Caching," *IEEE J. Select. Areas Commun.*, vol. 36, no. 6, pp. 1231–1244, Jun. 2018, doi: 10.1109/JSAC.2018.2844979.

[73] X. Lin, J. Xia, and Z. Wang, "Probabilistic caching placement in UAV-assisted heterogeneous wireless networks," *Physical Communication*, vol. 33, pp. 54–61, Apr. 2019, doi: 10.1016/j.phycom.2019.01.004.

[74] N. Zhao *et al.*, "Caching UAV Assisted Secure Transmission in Hyper-Dense Networks Based on Interference Alignment," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018, doi: 10.1109/TCOMM.2018.2792014.

[75] N. Zhao *et al.*, "Caching Unmanned Aerial Vehicle-Enabled Small-Cell Networks: Employing Energy-Efficient Methods That Store and Retrieve Popular Content," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 71–79, Mar. 2019, doi: 10.1109/MVT.2018.2881228.

[76] T. Zhang, Y. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Cache-Enabling UAV Communications: Network Deployment and Resource Allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7470–7483, Nov. 2020, doi: 10.1109/TWC.2020.3011881.

[77] A. K. Bhuyan and H. Dutta, "Design of a Heuristic IoT-Based Approach as a Solution to a Self-Aware Social Distancing Paradigm," in *Soft Computing Techniques in Connected Healthcare Systems*, CRC Press, 2023.

[78] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV Caching and Trajectory in Aerial-Assisted Vehicular Networks: A Learning-Based Approach," *IEEE J. Select. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, Dec. 2020, doi: 10.1109/JSAC.2020.3005469.

[79] T. Zhang, "Caching placement and resource allocation for cache-enabling UAV NOMA networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12897–12911, 2020.

[80] S. Chai and V. K. N. Lau, "Online Trajectory and Radio Resource Optimization of Cache-Enabled UAV Wireless Networks With Content and Energy Recharging," *IEEE Trans. Signal Process.*, vol. 68, pp. 1286–1299, 2020, doi: 10.1109/TSP.2020.2971457.

[81] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, "UAV-Assisted Content Delivery in Intelligent Transportation Systems-Joint Trajectory Planning and Cache Management," *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 8, pp. 5155–5167, Aug. 2021, doi: 10.1109/TITS.2020.3020220.

[82] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint Optimization of Caching, Computing, and Radio Resources for Fog-Enabled IoT Using Natural Actor–Critic Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019, doi: 10.1109/JIOT.2018.2878435.

[83] X. Wu, X. Li, J. Li, P. C. Ching, V. C. M. Leung, and H. V. Poor, "Caching Transient Content for IoT Sensing: Multi-Agent Soft Actor-Critic," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5886–5901, Sep. 2021, doi: 10.1109/TCOMM.2021.3086535.

[84] S. Araf, A. S. Saha, S. H. Kazi, N. H. Tran, and Md. G. R. Alam, "UAV Assisted Cooperative Caching on Network Edge Using Multi-Agent Actor-Critic Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2322–2337, Feb. 2023, doi: 10.1109/TVT.2022.3209079.

[85] W. Jiang, D. Feng, Y. Sun, G. Feng, Z. Wang, and X.-G. Xia, "Proactive Content Caching Based on Actor–Critic Reinforcement Learning for Mobile Edge Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1239–1252, Jun. 2022, doi: 10.1109/TCCN.2021.3130995.

[86] C. Wang *et al.*, "Heterogeneous Edge Caching Based on Actor-Critic Learning With Attention Mechanism Aiding," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3409–3420, Nov. 2023, doi: 10.1109/TNSE.2023.3260882.

[87] M. Yan, M. Luo, C. A. Chan, A. F. Gygax, C. Li, and C.-L. I, "Energy-Efficient Content Fetching Strategies in Cache-Enabled D2D Networks via an Actor-Critic Reinforcement Learning Structure," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17485–17495, Nov. 2024, doi: 10.1109/TVT.2024.3419012.

[88] X. Gao, Y. Sun, H. Chen, X. Xu, and S. Cui, "Joint Computing, Pushing, and Caching Optimization for Mobile-Edge Computing Networks via Soft Actor–Critic Learning," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 9269–9281, Mar. 2024, doi: 10.1109/JIOT.2023.3323433.

[89] Y. Xiao, H. Yu, Y. Yang, Y. Wang, J. Liu, and N. Ansari, "Adaptive Joint Routing and Caching in Knowledge-Defined Networking: An Actor-Critic Deep Reinforcement Learning Approach," *IEEE Transactions on Mobile Computing*, vol. 24, no. 5, pp. 4118–4135, May 2025, doi: 10.1109/TMC.2024.3521247.

[90] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep Reinforcement Learning-Based Edge Caching in Wireless Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 48–61, Mar. 2020, doi: 10.1109/TCCN.2020.2968326.

[91] A. K. Bhuyan, H. Dutta, and S. Biswas, “Towards a UAV-centric Content Caching Architecture for Communication-challenged Environments,” in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, IEEE, Dec. 2022, pp. 468–473.

[92] A. K. Bhuyan, H. Dutta, and S. Biswas, “UAV Trajectory Planning For Improved Content Availability in Infrastructure-less Wireless Networks,” in *2023 International Conference on Information Networking (ICOIN)*, Bangkok, Thailand: IEEE, Jan. 2023, pp. 376–381. doi: 10.1109/ICOIN56518.2023.10048929.

[93] A. K. Bhuyan, H. Dutta, and S. Biswas, “Handling Demand Heterogeneity in UAV-aided Content Caching in Communication-challenged Environments,” in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Boston, MA, USA: IEEE, Jun. 2023, pp. 107–116. doi: 10.1109/WoWMoM57956.2023.00025.

[94] X. Xu, M. Tao, and C. Shen, “Collaborative Multi-Agent Multi-Armed Bandit Learning for Small-Cell Caching,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2570–2585, Apr. 2020, doi: 10.1109/TWC.2020.2966599.

[95] P. Blasco and D. Gündüz, “Multi-armed bandit optimization of cache content in wireless infostation networks,” in *2014 IEEE International Symposium on Information Theory*, Jun. 2014, pp. 51–55. doi: 10.1109/ISIT.2014.6874793.

[96] X. Xu and M. Tao, “Decentralized Multi-Agent Multi-Armed Bandit Learning With Calibration for Multi-Cell Caching,” *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2457–2472, Apr. 2021, doi: 10.1109/TCOMM.2020.3045050.

[97] G. Tabei, Y. Ito, T. Kimura, and K. Hirata, “Design of Multi-Armed Bandit-Based Routing for in-Network Caching,” *IEEE Access*, vol. 11, pp. 82584–82600, 2023, doi: 10.1109/ACCESS.2023.3301961.

[98] Y. Han, L. Ai, R. Wang, J. Wu, D. Liu, and H. Ren, “Cache Placement Optimization in Mobile Edge Computing Networks With Unaware Environment—An Extended Multi-Armed Bandit Approach,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 8119–8133, Dec. 2021, doi: 10.1109/TWC.2021.3090440.

[99] S. A. Bitaghsir, A. Dadlani, M. Borhani, and A. Khonsari, “Multi-Armed Bandit Learning for Cache Content Placement in Vehicular Social Networks,” *IEEE Communications Letters*, vol. 23, no. 12, pp. 2321–2324, Dec. 2019, doi: 10.1109/LCOMM.2019.2941482.

[100] S. Liu, J. Yu, X. Deng, and S. Wan, “FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022, doi: 10.1109/TITS.2021.3099368.

[101] Z. Wang *et al.*, “Asynchronous Federated Learning Over Wireless Communication Networks,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 6961–6978, Sep. 2022, doi: 10.1109/TWC.2022.3153495.

[102] H. Ye, L. Liang, and G. Y. Li, "Decentralized Federated Learning With Unreliable Communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 487–500, Apr. 2022, doi: 10.1109/JSTSP.2022.3152445.

[103] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated Learning Over Wireless IoT Networks With Optimized Communication and Resources," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16592–16605, Sep. 2022, doi: 10.1109/JIOT.2022.3151193.

[104] H. Chen, M. Xiao, and Z. Pang, "Satellite-Based Computing Networks with Federated Learning," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 78–84, Feb. 2022, doi: 10.1109/MWC.008.00353.

[105] J. Lee, F. Solat, T. Y. Kim, and H. V. Poor, "Federated Learning-Empowered Mobile Network Management for 5G and Beyond Networks: From Access to Core," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 2176–2212, 2024, doi: 10.1109/COMST.2024.3352910.

[106] X. Zhou *et al.*, "Decentralized P2P Federated Learning for Privacy-Preserving and Resilient Mobile Robotic Systems," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 82–89, Apr. 2023, doi: 10.1109/MWC.004.2200381.

[107] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-Effective Federated Learning in Mobile Edge Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3606–3621, Dec. 2021, doi: 10.1109/JSAC.2021.3118436.

[108] R. Yu and P. Li, "Toward Resource-Efficient Federated Learning in Mobile Edge Computing," *IEEE Network*, vol. 35, no. 1, pp. 148–155, Jan. 2021, doi: 10.1109/MNET.011.2000295.

[109] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: an efficient federated learning framework for heterogeneous mobile clients," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, New Orleans Louisiana: ACM, Oct. 2021, pp. 420–437. doi: 10.1145/3447993.3483278.

[110] M. Gecer and B. Garbinato, "Federated Learning for Mobility Applications," *ACM Comput. Surv.*, vol. 56, no. 5, pp. 1–28, May 2024, doi: 10.1145/3637868.

[111] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. S. Quek, and G. Min, "Mobility-Aware Cluster Federated Learning in Hierarchical Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8441–8458, Oct. 2022, doi: 10.1109/TWC.2022.3166386.

[112] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, "Federated Learning With Spiking Neural Networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6183–6194, 2021, doi: 10.1109/TSP.2021.3121632.

[113] C. He *et al.*, “FedGraphNN: A Federated Learning System and Benchmark for Graph Neural Networks,” Sep. 08, 2021, *arXiv*: arXiv:2104.07145. doi: 10.48550/arXiv.2104.07145.

[114] K. Xie *et al.*, “Efficient Federated Learning With Spike Neural Networks for Traffic Sign Recognition,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9980–9992, Sep. 2022, doi: 10.1109/TVT.2022.3178808.

[115] Z. Li, T. Lin, X. Shang, and C. Wu, “Revisiting Weighted Aggregation in Federated Learning with Neural Networks,” in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, Jul. 2023, pp. 19767–19788. Accessed: May 13, 2025. [Online]. Available: https://proceedings.mlr.press/v202/li23s.html

[116] C. Liu, C. Lou, R. Wang, A. Y. Xi, L. Shen, and J. Yan, “Deep Neural Network Fusion via Graph Matching with Applications to Model Ensemble and Federated Learning,” in *Proceedings of the 39th International Conference on Machine Learning*, PMLR, Jun. 2022, pp. 13857–13869. Accessed: May 13, 2025. [Online]. Available: https://proceedings.mlr.press/v162/liu22k.html

[117] Y. H. Cho and W. J. Byun, “Generalized Friis Transmission Equation for Orbital Angular Momentum Radios,” *IEEE Trans. Antennas Propagat.*, vol. 67, no. 4, pp. 2423–2429, Apr. 2019, doi: 10.1109/TAP.2019.2891438.

[118] W. J. Byun and Y. Heui Cho, “Analysis of a 200-GHz OAM Radio Link Using a Generalized Friis Transmission Equation,” in *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, Atlanta, GA, USA: IEEE, Jul. 2019, pp. 1051–1052. doi: 10.1109/APUSNCURSINRSM.2019.8888820.

[119] R. Wang, S. Biswas, S. Das, and J. Rao, “Collaborative Caching for Dynamic Map Dissemination in Vehicular Networks,” in *2019 IEEE ComSoc International Communications Quality and Reliability Workshop (CQR)*, Apr. 2019, pp. 1–6. doi: 10.1109/CQR.2019.8880103.

[120] P. Blasco and D. Gunduz, “Learning-Based Optimization of Cache Content in a Small Cell Base Station,” Feb. 21, 2014, *arXiv*: arXiv:1402.3247. Accessed: Mar. 21, 2024. [Online]. Available: http://arxiv.org/abs/1402.3247

[121] R. Wang, J. Rao, C. Zhou, and S. Biswas, “Connectionless Edge-Cache Servers for Reducing Cellular Bandwidth Usage in Vehicular Networks,” in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bangalore, India: IEEE, Jan. 2021, pp. 516–524. doi: 10.1109/COMSNETS51098.2021.9352746.

[122] B. Lee, “Web Caching and Zipf-like Distributions : Evidence and Implications,” *IEEE INFOCOM, 2009*, 2009, Accessed: Mar. 21, 2024. [Online]. Available: https://cir.nii.ac.jp/crid/1570854176352908672

[123] H. Yao, C. Bai, M. Xiong, D. Zeng, and Z. Fu, “Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing,” *Concurrency and*

Computation: Practice and Experience, vol. 29, no. 16, p. e3975, 2017, doi: 10.1002/cpe.3975.

[124] S. Ali, N. Rajatheva, and W. Saad, "Fast Uplink Grant for Machine Type Communications: Challenges and Opportunities," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 97–103, Mar. 2019, doi: 10.1109/MCOM.2019.1800475.

[125] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, Mar. 1981, doi: 10.1016/0022-2836(81)90087-5.

[126] A. K. Bhuyan, H. Dutta, and S. Biswas, "UAV Trajectory Planning For Improved Content Availability in Infrastructure-less Wireless Networks," in *2023 International Conference on Information Networking (ICOIN*, IEEE, Jan. 2023, pp. 376–381.

[127] A. Slivkins, "Introduction to Multi-Armed Bandits," *MAL*, vol. 12, no. 1–2, pp. 1–286, Nov. 2019, doi: 10.1561/2200000068.

[128] W. Cao, J. Li, Y. Tao, and Z. Li, "On Top-k Selection in Multi-Armed Bandits and Hidden Bipartite Graphs," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015. Accessed: Feb. 29, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/hash/ab233b682ec355648e7891e66c54191b-Abstract.html

[129] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A Comparison of String Metrics for Matching Names and Records".

[130] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. in Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 1998.

[131] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021, doi: 10.1016/j.neucom.2021.07.098.

[132] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.

[133] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing Federated Learning on Non-IID Data with Reinforcement Learning," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada: IEEE, Jul. 2020, pp. 1698–1707. doi: 10.1109/INFOCOM41043.2020.9155494.

[134] A. P. Majtey, P. W. Lamberti, and D. P. Prato, "Jensen-Shannon divergence as a measure of distinguishability between mixed quantum states," *Phys. Rev. A*, vol. 72, no. 5, p. 052310, Nov. 2005, doi: 10.1103/PhysRevA.72.052310.

[135] A. K. Bhuyan, H. Dutta, and S. Biswas, "Towards a UAV-centric Content Caching Architecture for Communication-challenged Environments," in *GLOBECOM 2022 - 2022*

IEEE Global Communications Conference, Rio de Janeiro, Brazil: IEEE, Dec. 2022, pp. 468–473. doi: 10.1109/GLOBECOM48099.2022.10001616.

[136] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.

[137] W. Cao, J. Li, Y. Tao, and Z. Li, "On top-k selection in multi-armed bandits and hidden bipartite graphs," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[138] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[139] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[140] T. C. T. Kotiah, "Chebyshev's inequality and the law of large numbers," *International Journal of Mathematical Education in Science and Technology*, vol. 25, no. 3, pp. 389–398, May 1994, doi: 10.1080/0020739940250310.

[141] T. C. T. Kotiah, "Chebyshev's inequality and the law of large numbers," *International Journal of Mathematical Education in Science and Technology*, vol. 25, no. 3, pp. 389–398, May 1994, doi: 10.1080/0020739940250310.

[142] A. K. Bhuyan, H. Dutta, and S. Biswas, "Handling Demand Heterogeneity in UAV-aided Content Caching in Communication-challenged Environments," in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Boston, MA, USA: IEEE, Jun. 2023, pp. 107–116. doi: 10.1109/WoWMoM57956.2023.00025.

[143] M. Cheatham *et al.*, "On the efficient execution of bounded Jaro-Winkler distances," *Semant. web*, vol. 8, no. 2, pp. 185–196, Jan. 2017, doi: 10.3233/SW-150209.

[144] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," 2018, doi: 10.48550/arXiv.1806.00582.

[145] A. K. Bhuyan, H. Dutta, and S. Biswas, "Unsupervised Speaker Diarization in Distributed IoT Networks Using Federated Learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, no. 2, pp. 1934–1946, Apr. 2025, doi: 10.1109/TETCI.2024.3482855.

[146] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous Federated Learning: State-of-the-art and Research Challenges," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 1–44, Mar. 2024, doi: 10.1145/3625558.

[147] E. T. Martínez Beltrán *et al.*, "Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023, doi: 10.1109/COMST.2023.3315746.

[148] A. K. Bhuyan and J. H. Nirmal, “Comparative study of voice conversion framework with line spectral frequency and Mel-Frequency Cepstral Coefficients as features using artficial neural networks,” in *2015 International Conference on Computers, Communications, and Systems (ICCCS)*, Nov. 2015, pp. 230–235. doi: 10.1109/CCOMS.2015.7562906.

[149] A. K. Bhuyan, H. Dutta, and S. Biswas, “Unsupervised Quasi-Silence based Speech Segmentation for Speaker Diarization,” in *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, May 2022, pp. 170–175. doi: 10.1109/SETIT54465.2022.9875932.

[150] Y. Kim, “Sequence-to-Sequence Learning with Latent Neural Grammars,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 26302–26317. Accessed: Apr. 26, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/dd17e652cd2a08fdb8bf7f68e2ad3814-Abstract.html

[151] D. Cai and W. Lam, “Graph Transformer for Graph-to-Sequence Learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, Art. no. 05, Apr. 2020, doi: 10.1609/aaai.v34i05.6243.

[152] L. Yang, T. L. J. Ng, B. Smyth, and R. Dong, “HTML: Hierarchical Transformer-based Multi-task Learning for Volatility Prediction,” in *Proceedings of The Web Conference 2020*, Taipei Taiwan: ACM, Apr. 2020, pp. 441–451. doi: 10.1145/3366423.3380128.