On A Novel Adaptive UAV-Mounted Cloudlet-Aided Recommendation System for LBSNs

Fengxiao Tang, Student Member, IEEE, Zubair Md. Fadlullah, Senior Member, IEEE, Bomin Mao, Student Member, IEEE, Nei Kato, Fellow, IEEE Fumie Ono, Member, IEEE, and Ryu Miura, Member, IEEE.

Abstract—Location Based Social Networks (LBSNs) have recently emerged as a hot research area. However, the high mobility of LBSN users and the need to quickly provide access points in their interest zones present a unique research challenge. In order to address this challenge, in this paper, we consider the Unmanned Aerial Vehicles (UAVs) to be a viable candidate to promptly form a wireless, meshed offloading backbone to support the LBSN data sensing and relevant data computations in the LBSN cloud. In the considered network, UAV-mounted cloudlets are assumed to carry out adaptive recommendation in a distributed manner so as to reduce computing and traffic load. Furthermore, the computational complexity and communication overhead of our proposed adaptive recommendation are analyzed. The effectiveness of the proposed recommendation system in the considered LBSN is evaluated through computer-based simulations. Simulation results demonstrate that our proposal achieves much improved performance compared to conventional methods in terms of accuracy, throughput, and delay.

Index Terms—Location Based Social Network (LBSN), recommendation system, Unmanned Aerial Vehicle (UAV), edge computing, cloudlet.

1 INTRODUCTION

R

ecently, due to the proliferation of next-generation, high-speed mobile communication networks, Location Based Social Networks (LBSNs) emerged as a popular platform to tightly integrate human users with physical locations and cyber elements. [1], [2] For instance, LBSN platforms include Foursquare, Twitter, Gowalla, Facebook Places, Street, WeChat, and so forth, are currently dominating the mainstream social networks. A unique demand of a typical LBSN consists of the need for the cyber elements (e.g., LBSN service cloud) to appropriately understand the users interests/preferences, location property [3], inter-user relationship [4], inter-location dependencies, and user-location correlations. For example, the location based recommendation system can suggest suitable locations, friends, and activities to the users depending on their respective interests, locations, and activity history data which are stored in the cloud servers. However, traditionally, the legacy applications of recommendation systems typically employ centralized computing, support only passive recommendation (i.e., only after a user query, the system responds with a relevant recommendation), and usually consider limited user mobility (e.g., long term user movement over days, weeks, or months). Furthermore, the traditional LBSN recommendation systems do not take into account the real-time performance of the recommendations. Our focus in this paper, as shown in Fig. 1, is to address these limitations of the conventional LBSN recommendation systems. We consider a highly dynamic LBSN environment where we aim to integrate both passive and active recommendation systems in a seamless manner to meet the real-time requirements of highly dynamic and frequent LBSN user mobility in addition to long term user movement patterns. As demonstrated in our earlier work [5], Unmanned Aerial Vehicles (UAVs) can be utilized to construct highly dynamic networks such as LBSNs. In this paper, we consider multiple UAVs as a good candidate to quickly form a meshed offloading backbone for sensing and dissemination of the LBSN data as depicted in Fig. 2. Besides, mobile edge computing has emerged as a key enabler for next 5G networks [6] to move computation and communication load towards the edge of the network to harness computational capabilities that are currently untapped in edge nodes. [7] The UAVs can, thus, facilitate flexible deployment of edge nodes for LBSN data delivery as well as help reduce deployment costs. [8] In our proposed UAV based LBSN, a new cloudlet technology is employed at the UAVs to reduce traffic load and computational complexity at the LBSN cloud servers. A UAV-mounted cloudlet is a trusted, resource-rich computer or cluster of computers that may be connected to the UAV and available for use by nearby mobile devices [9]. With high computing ability, large storage, and flexible connectivity, the cloudlet can perform a part of the computations of the recommendation systems that would be otherwise carried out by the fixed LBSN service cloud depicted in Fig. 2. Based on the UAV-mounted cloudlet-aided LBSN, we propose a novel, adaptive recommendation system. The proposed recommendation system, first, deploys cloudlets in a balanced way to share the computing load with the central controller to accelerate the recommendation process.
Our proposed cloudlets based adaptive and fast recommendation algorithm consists of both passive and novel active recommendations.

As shown in the Fig. 2, the passive recommendation is similar to the traditional recommendation method whereby the service cloud only responds to a user query by providing appropriate recommendation to cater to the user’s demands and/or preferences. In addition to the passive recommendation, our proposal incorporates the active recommendation, which consists in an automatic push service facilitated by the cloudlet even without a query from the user. In case of the active recommendation, the nearby cloudlet handles almost all the computing tasks and communicates with the user so as to reduce the computational cost and communication overhead of service cloud while satisfying the real-time recommendation demands of the user. Besides, to better exploit the computation and communication ability of cloudlets, and save both deployment and operation cost, a resource balanced cloudlet deployment algorithm is proposed to initialize the UAV-mounted cloudlet-aided LBSN recommendation system. Based on such deployed cloudlets, the joint area recommendation exchange mechanism is considered in our proposed adaptive recommendation algorithm. Furthermore, we researched the real-time performance of recommendation in the UAV-mounted cloudlet-aided LBSN scenario and analytically evaluate its computational performance.

The remainder of this paper is organized as follows. Sec. 2 surveys related works on cloudlet-aided LBSN and the recommendation problem. Sec. 3 presents the considered LBSN and cloudlet system model. Then, this section further describes the formulation of our problem. Sec. 4.1 discusses how to deploy the cloudlet mounted UAVs. Our adaptive recommendation system is proposed in Sec. 4.2. We analyzed the real-time performance and computational complexity in Sec. 5. The performance of our proposal is evaluated through extensive computer-based simulations in Sec. 6. Finally, Sec. 7 concludes the paper.

2 Related works

A cross-domain recommendation model for processing and computing information in CPS was proposed by Ma et al. in [10]. While the cross-domain recommendation model is used to improve the recommendation ratio based on rating patterns, it does not consider any location specific detail, which is critical for LBSNs. Recently, advances in wireless communication technologies [11], [12], [13] and location-acquisition have contributed to the emergence of LBSN services like Foursquare, Twinkle, and GeoLife [14]. By employing such LBSN services, the users are able to easily share their respective geo-spatial locations as well as location-related contents with the outside world through online platforms. The survey in [15] distinguished LBSNs from conventional social networks and discussed their unique properties, challenges, and opportunities such as location context awareness, location recommendations, activity recommendations, community discovery, graph heterogeneity, rate of growth, and so forth. While the survey provided a comprehensive taxonomy of LBSNs from several perspectives, it did not take into account mobility models within LBSNs. On the other hand, the work in [16] considered human movement and mobility patterns subject to geographic and social constraints, and presented a structured model to predict the locations and dynamics of future users movement in the social networks. Chorley et al. [17] investigated the personality and use of LBSNs. The work also discussed how massive user participation has resulted in a widespread proliferation of LBSNs, particularly in the urban areas throughout the developed world. Furthermore, the emerging applications of LBSNs for urban planning, traffic forecasting, advertising, and recommendations based on the collaborative social knowledge learning are presented in [18], [19]. Based on Collaborative Filtering (CF), the work in [20] proposed a cloud-based recommendation system by employing bi-clustering and fusion. However, the work mainly aimed to solve the cold-start problem associated with CF mechanisms. Moreover, it can be understood from the work that CF may not be particularly suitable for LBSNs due to its implicit assumption that all user/item ratings contribute the same to recommendation. In addition to temporal and spatial models to predict mobility, their work proposed a regularity-based mining of the LBSNs consisting of Markov-based predictors. How LBSN data can become big data sources, particularly for travel demand and activity modeling, was discussed in [21], [22], [23]. However, the common shortcoming in the aforementioned works is that the wireless network technologies facilitating the LBSNs were not taken into consideration. The edge computing is recently used in networks to share the load with the central cloud [6], [7]. As one implementation of edge computing, the cloudlet can be treated as a computational cluster of nodes in the edge of the network for provisioning resources to nearby users [24]. Our earlier work in [25] researched the service delay optimization problem in edge computing with two cloudlets. On the other hand, one of our earlier works in [26] demonstrated how emerging wireless networks aided by UAVs can become useful in enabling communications in ultra-dense environments in urban locations. To the best of our knowledge, however, no previous work has investigated the importance of a UAV-aided backbone for supporting LBSNs. The research works in [27], [28] proposed content retrieval/computation systems at the edge over Device-to-Device (D2D) communication links to reduce cellular traffic volume in mobile networks. Furthermore, mobile edge computing via a UAV-mounted cloudlet is researched in [8], which shows that the UAV-mounted cloudlet can have high communication and computation ability in the edge of the network. Our UAV-mounted cloudlet-aided network can be considered similar in spirit to the mentioned traffic reduction work as it also aims to facilitate the content retrieval and computation at the UAVs near the edge, i.e., close to the LBSN users. While the recommendation system for UAV based LBSN is yet to be seen in the existing literature, the closest research work in a similar area adopting vehicles can be found in [29]. Based on the location datasets and contexts collected in a number of large cities in the USA and China, a data driven approach was designed to recommend both regular taxis and carpooling vehicles to the users. However, the work considered a fixed dispatching center cloud server for carrying out the recommendation tasks.
Legacy recommendation system using passive query is not adequate for high real-time requirement of LBSN users. Existing recommendation systems consider limited user mobility, e.g., weekly/monthly user movement. Traditional recommendation systems based on content, link analysis, and collaborative filtering do not consider real-time performance of the recommendation.

Our focus:
- Investigate how UAV-mounted cloudlets can be leveraged in a highly dynamic LBSN environment.
- Integrate both passive and active recommendation systems in a seamless manner so as to meet the high real-time requirement of users.
- Consider highly dynamic and frequent LBSN user mobility in addition to a longer term user movement.
- Consider real-time recommendation in UAV-aided cloud-mounted LBSN scenario and analytically evaluate its computational performance.

3 **System Model and Problem Formulation**

In a traditional social network, all data aggregation and computation are performed only at the fixed cloud servers. This results in both network congestion and significantly high computational burden on the cloud servers. Fig. 2 demonstrates the transmission and computing bottleneck that limits the LBSN service quality. The figure also shows a new cloudlet-based system model to solve this problem which is illustrated in the figure. In other words, the biggest difference between our proposed network and the traditional communication network is the deployment of flexible UAVs with high performance computing cloudlets. With the deployed cloudlets and edge computing technique, both computing and traffic load can be shared from central cloud to the edge of network. For brevity, consider the system to be a two-layer model, i.e., the social users-UAV layer and the UAVs-cloudlet-Cloud Server layer. Instead of imposing the data aggregation and computation solely on the fixed cloud server as in the case of a conventional social network, the mobile cloudlets in our considered system permits the storage and processing of some data in their localized control areas. The cloudlet based system model is described below.

The social users sense data from physical systems, embed their local information, and transmit to the UAVs. These are referred to as the LBSN data. The UAVs deliver the LBSN data over the UAV-based meshed backbone to the fixed cloud or mobile cloudlet with an adaptive algorithm, which will be later proposed in Sec. 4. Both the fixed cloud and the mobile cloudlets, upon processing and computing the data using their respective computational resources, send the relevant result/information to the LBSN users using the aforementioned network.

In LBSN, the recommendation system is a critical requirement of social users. However, conventional recom-
mendation systems in LBSN mostly focus on the accuracy of recommended items based on content, link analysis, and collaborative filtering, and do not consider the real-time performance. Besides, they only consider limited user mobility, e.g., weekly/monthly user movement. This is why we need to focus on the important requirement of real-time performance and high mobility of users. Therefore, depending on the presented cloudlet based network model, the new recommendation system model is proposed and described next.

Unlike other LBSN works, which assume that the network elements can be simply borrowed from the existing literature, we argue that the cloudlet based backbone network and corresponding recommendation system have some unique characteristics and challenges. Therefore, we aim to focus on the network aspects in the remainder of this section by describing our considered network and recommendation models followed by a formal problem formulation.

3.1 Considered Network Model

Consider the UAV-aided LBSN as a three-dimensional topology. Let graph $G = (V + U, E)$ represent the network, where $V$ denotes the set of big vertices in the network including mobile UAVs $V_u$, fixed service cloud $V_f$, and mobile cloudlet $V_m$. And $U$ denotes the set of $R$ users, where $U = \{u_1, u_2, \ldots, u_R\}$, and all users are randomly located in the network. Let $N$ denote the number of existing UAVs in the system that are represented by the set, $V_u = \{v_{u1}, v_{u2}, \ldots, v_{uN}\}$. Each UAV in the considered network is represented by its own features, i.e., latitude, longitude, and height. $H$ cloudlets in the set $V_m$ are represented as $V_m = \{v_{m1}, v_{m2}, \ldots, v_{mH}\}$. For simplicity, we consider only one fixed service cloud in the network that is denoted by $v_f$.

$E$ represents the edges set in the graph $G$. Furthermore, the edge $e \in E$ in the graph means the link between two vertices. The weight, $w(e)$, represents the connection ability of the link $e$. This weight depends on many factors such as transmission distance, transmission power, interference, bandwidth, and so on.

As described above, the set $V$ consists of $V_u$, $V_f$ and $V_m$. The fixed service cloud is initially assumed to be in a fixed geographic location. Because of the limited resources (including computing resource, traffic resource, and communication delay) of the fixed service cloud, a number of mobile cloudlets are mounted on the UAVs to offer recommendation service to the LBSN users. In order to balance the resource of all users, the strategy of cloudlets deployment should be globally considered. By taking into account the real environment, the deployment locations of the cloudlet are restricted by many factors such as connection distance, cost, safety, interference, and so on. Hence, the final candidate locations should be finite in some limited areas. Therefore, how to effectively select the most suitable locations consists in an important research problem.

3.2 Considered Recommendation Model

The location based recommendation system can be categorized into four classes based on objectives such as locations, activities, users, and social media [15]. In this paper, considering the high dynamic of users, we mainly focus on the locations and activities recommendation for the LBSN users that can be also easily expanded to the users and social media recommendation.

In location recommendation, we consider the user preferences to be mainly affected by two factors, i.e., user interest and local preference. Like the model defined in the research work in [30], we define the interest of a user as $\rho_u$, and the local preference of the location $l$ as $p_l$.

Assume that the notion $\Sigma$ represents the set of event or venue (spatial item) in the considered LBSN. This set contains a series of events or venues $\sigma$ in a certain area $l_l$. Because of the different subjects of user interest, we consider the characteristics of the user side and location side, respectively.

On the user side, we categories all the data of different users. Let the data of user $u$ be represented as $D_u$, and the history activities of user $u$ at venue $\sigma$ in $l_l$ be denoted as a tuple $(u, \sigma, l_l)$. Every activity in $\sigma$ contains many content words of $u$. The content set is expressed as $c_{\sigma}$. The set of contents in different users is represented as $C$. The whole profile data $D_{u\sigma}$ of user $u$ in $l_l$ is represented as a tuple $D_{u\sigma} = (u, \sigma, l_l, c_{\sigma})$.

On the other hand, in case of the location side, suppose that the data with subject of location $l$ is represented as $D_l$, the connection between topic $\omega$ and venue set $\Sigma$ in $D_l$ is denoted by a topic model $\phi_{\omega}$. Additionally, consider the connection between the topic $\omega$ and content set $C$ as another topic model, $\phi'_{\omega}$. Those two models represent two types of relationship of topics with other social elements. In a practical system, both the models should be used in order to obtain the real local preference.

With the representation of both user and location sides, the likelihood of user $u$ in location $l$ can be simply expressed as a combination of user interest $\rho_u$ and local preference $p_l$ that is referred to as an LCA-LDA model in [30]. Furthermore, with the high mobility of users, the movement and distance $d_e$ between the user and event are also considered in our model as follows.

$$P(\sigma|\rho_u, \rho_l, \phi, \phi') = \lambda_u P(\sigma|\rho_u, \phi, \phi') + (1 - \lambda_u - \lambda_f)P(\sigma|\rho_l, \phi, \phi') + \lambda_f(1/(1 + e^{-1/d_e})),$$

where $P(\sigma|\rho_u, \phi, \phi')$ and $P(\sigma|\rho_l, \phi, \phi')$ denote the probability that the spatial item $\sigma$ is generated according to user interest and local preference, respectively. Furthermore, $\lambda_u$ and $\lambda_f$ are the mixing wights of the three factors, respectively.

Next, we consider the connection of topic to preference. In this vein, we may reformulate eq. 1 as follows.

$$\sum_{c \in C_{\sigma}} P(\sigma, c|\rho_u, \rho_l, \phi, \phi') = \lambda_u \sum_{\omega} P(\sigma, c|\omega, \phi, \phi') P(\omega|\rho_u) + (1 - \lambda_u - \lambda_f) \sum_{\omega} P(\sigma, c|\omega, \phi, \phi') P(\omega|\rho_l) + \lambda_f(1/(1 + e^{-1/d_e})),$$

where we have divided the formulation into location and user sides in order to conveniently process this recommen-
As described in the network model, every mobile could utilize limited cloudlets in the network for sharing load with the central cloud. However, with unbalanced load and the complexity of the infrastructure of large scale UAV-aided LBSN, how to deploy the resource assignment and cloudlet deployment problem as the shortest path problem in the considered graph. Because the weight between two vertices represents the resource assigned between them that is constructed by many elements such as bandwidth, amount of data need to be transferred, risk of intermediate node drop-off, and so forth. Our final objective of this shortest path problem can be expressed below,

\[
\min \sum_{i=1}^{H} \sum_{j=1}^{N+H+1} w(v_{mi}, v_{aj}).
\]

With the above objective function, we propose a UAV-mounted cloudlet deployment algorithm to find the suitable locations of the cloudlets in Sec. 4.1.

After the suitable locations of the cloudlets are determined, each UAV/user within the control area of the cloudlet is assigned to it. The recommendation requirements of the users can be simply serviced in the cloudlet instead of sending all the recommendation inquiries to the single fixed service cloud and waiting for a significantly long time in the service queue.

### 3.3.2 Recommendation Mechanism Problem

As we defined earlier, the passive recommendation can be easily employed in the conventional cloud server. But with the proposed active recommendation model, because of the high mobility and irregularity of users, the realtime recommendation is critical that requires low latency traffic. This means that the recommendation message needs fast transmission (i.e., with small delay). But with the conventional cloud-based network, all computations and transmissions are integrated to the central cloud, and lead to heavy load and congestion. Thus, the deployment of cloudlets is crucial to share the load of central cloud.

Most of the computing resources and traffic loads can be shared by the cloudlet, which can efficiently reduce the transmission delay of LBSN contents to the LBSN cloud. Thus, the real-time performance of recommendation can be significantly improved, which permits the active recommendation to be carried out to meet the real-time requirements of users. In section 5, we further analyze the performance improvement of recommendations with cloudlets based on queuing theory.

Even though the cloudlet-aided is able to facilitate the active real-time recommendation service to the LBSN users, it may still confront several problems, which are described below.

- Switching Between Passive and Active Recommendation Models: We define new recommendation models of active and passive, but While we have defined passive and active recommendation models,
how they can be effectively combined in the entire recommendation mechanism may pose a challenge. In other words, how to switch between the two recommendation models is a significant research challenge. In order to address this issue, time or user moved distance may be used as the switching factor. However, setting the appropriate time interval or user moved distance threshold becomes critical to accurately switch between the recommendation models.

- Cloudlet Data Computing: The cloudlet is a powerful server to share a portion of the tasks, which would be otherwise carried out at the fixed cloud. However, the cloudlet still is limited by its storage and computational limits. In other words, all the user data cannot be stored in the cloudlet. Therefore, the decision on how much data should be stored in the cloudlet, and which portion of the data should be processed for recommendation consists in a research problem.

- Cloudlet Handover: When the cloudlet is deployed on the UAV in the considered LBSN, the control area of the cloudlet is determined. However, due to the irregular mobility of an LBSN user, the user may go through one area to another area, and this may trigger the need to perform cloudlet handover. In other words, the recommendation work should be transferred from the original cloudlet to the new one in a fast and seamless manner. With the original cloudlet having different data and performing various processing tasks, it may not be trivial for the cloudlet to simply transfer the relevant user information on the fly. Therefore, the effective cloudlet handover consists in a research challenge also.

In order to address these identified problems, an adaptive recommendation mechanism is proposed in the following section.

4 Cloudlet-aided Adaptive Recommendation Mechanism

With the problems discussed in Sec. 3, a novel cloudlet-aided adaptive recommendation mechanism is proposed.

4.1 Cloudlet Deployment Algorithm

When the cloudlets are deployed in the UAV-aided network, the network traffic can be optimized. Consider the communication overhead and complexity of control, the size of the cloudlet control area cannot be so large which is mainly decided by the number of UAVs \( N_q \) in each area. Hence, in each control area, there are one cloudlet and \( N_q \) UAVs. The minimal weight \( w_h \) of the control area can be calculated by eq. 4.

The cloudlet deployment problem is already simplified as the shortest path problem in Sec. 3. Before delving into the deployment process, some assumptions are given as inputs of the deployment algorithm. In particular, the adjacency graph and matrix are given to each UAV and cloudlet.

In the beginning of the cloudlet deployment algorithm, an iteration structure \( B \) of partitions combinations set of sets is given to store all the possible control area combinations set \( p \). Thus, the output of our deployment algorithm is to provide an optimal area combination set \( Z \in B \), which has the same structure of \( p \). The iteration structure \( B \) is constructed with a vector, which can easily pop its element set \( p \) and assess whether it is empty. Therefore, our algorithm is to traverse the entire vector to calculate the minimal weight \( w_h \) of \( p \in B \) and set the optimal one to \( Z \) until \( B \) is empty. The steps of the cloudlet deployment are shown in Alg. 1.

Algorithm 1 Cloudlet Area Deployment Algorithm

Input: The adjacency matrix \( A \) of \( G \), shortest path matrix \( D \) of adjacency matrix \( A \). Iteration set of area combinations \( B \).

Output: Optimal area combination set \( Z \).

1: \( w_h = \infty \)
2: while Area iteration set \( B \) \(!=\) \( \emptyset \) do
3: Pop the first area combination set \( P \) of \( B \)
4: Calculate the minimal weight of set \( p \) with eq. 4 as \( \min (p) \)
5: if \( w_h > \min (p) \) then
6: \( w_h = \min (p) \)
7: \( Z = p \)
8: \( Z \) if
9: Update the combination information to all UAVs and cloudlets.
10: end while

It is worth noting that each control area includes one cloudlet and several UAVs. The size of the control area is decided by the transmission range of all those UAVs. In order to provide full coverage to all the users in the control area, there must be some overlapping areas between different cloudlets. We refer to such overlapping areas as the joint areas, which provide the ability of cloudlet handover (i.e., to transit the service from one cloudlet to another). The service transition mechanism is described next.

4.2 Adaptive Recommendation Algorithm

The proposed recommendation algorithm is constructed by passive and active recommendation models. As input, there is a trigger used to switch between the two models. When a user presents her inquiry, the trigger switches to the passive model. Otherwise, the trigger keeps in active model. We discuss the adaptive recommendation algorithm in the following two parts.

4.2.1 Passive Model Algorithm

When a user \( u \) checks in and asks a query, the trigger switches to the passive recommendation model. The user information of \( u \) is recorded as input, which includes user location and query location that are not necessarily the same. For example, the user in location \( A \) can ask for the recommendation of location \( B \). The output is the recommendation list of the query location.

With the recommendation model described in Sec. 3, the data in both location and user sides are recorded in the LBSN service cloud by calculating eq. 2. The result is a list \( L \) of all ranked recommendation items. After the passive recommendation is completed, the user gives feedback \( f_L \), the cloud updates the data \( D_u \) of user \( u \) depending on the feedback. The latest updated data \( D_u \) are retained in the cloud. Also, the cloud receives the information regarding which cloudlet the user belongs to. After the query and answer process are finished, it checks whether \( D_u \) exists in this cloudlet. If not, the cloud sends the updated data \( D_u \) to the cloudlet. The process is shown in the steps of Alg. 2.
Algorithm 2 Passive Recommendation Algorithm

Input: Query \( q_v \) of user \( u \), user data \( D_u \).
Output: Recommendation list \( L \), updated user data \( D_u' \).

1. User \( u \) checks in, and presents the query \( q_v \).
2. Cloud \( v_f \) gets the id \( v_{mu} \) of cloudlet which user \( u \) belongs to.
3. Cloud \( v_f \) calculates the recommendation result as list \( L \) with eq. 2.
4. Cloud \( v_f \) sends recommendation list \( L \) to user \( u \).
5. If \( u \) gives feedback \( f_L \) of \( L \), then
6. Update user data as \( D_u' = D_u + f_L \).
7. end if
8. If \( D_u = \emptyset \) in cloudlet \( v_{mu} \) then
9. Cloud \( v_f \) sends \( D_u' \) to \( v_{mu} \).
10. end if

4.2.2 Active Model Algorithm

When the passive recommendation model is not triggered, the active model is switched on. Different from the conventional recommendation, the active recommendation model is a totally new mechanism of recommendation system, which is triggered by a timer instant of the user’s query.

A fixed interval \( t \) in the timer is given to trigger the active recommendation algorithm. The location of user \( u \) is obtained to evaluate whether its distance from the location in the last interval is larger than a threshold \( d \). If the distance exceeds the threshold, eq. 2 is executed to obtain the result list \( L \) and to calculate whether any item in \( L \) is close to \( u \) within the threshold \( d/u \). If this condition is satisfied, then the result is pushed to the user. Furthermore, any feedback from the user is also used to update the user data \( D_u \). Otherwise, this interval is ignored.

On the other hand, the location of the user is recorded in the cloudlet with each time interval \( t_f \) when the user is moving to a joint area (i.e., when the user may move to another cloudlet control area). In such a case, the data transfer algorithm is processed between those cloudlets.

Regardless of the number of control areas in the joint area, \( D_u \) of user \( u \) in the old area should transfer to all other cloudlets. If the data of \( u \) already exists in another cloudlet, it is updated to the new \( D_u' \). During the time the user stays in the joint area, the old cloudlet continues to perform the recommendation service to \( u \) until the user entirely moves to another control area.

By considering the memory size of a cloudlet to be much smaller than that of the fixed LBSN cloud, the cloudlet only stores all locations data \( D_l \) within its own control area. The \( D_l \) includes all users’ topics about the venues and items within location \( l \) and always keep the last updated data from the cloud with time interval \( T \). With the same time interval \( T \), the user’s new data \( D_u' \) is received from the cloud for updating \( D_u \). When the user gives feedback of any recommendation, \( v_{mu} \) and \( v_f \) both update their \( D_u \) to \( D_u' \).

In the following section, we analyze the real-time performance and computational time complexity of the proposed UAV-mounted cloudlet-aided adaptive recommendation algorithm in the considered LBSN.

5 PERFORMANCE ANALYSIS

5.1 Real-time Performance Analysis

In order to analyze the real-time performance of cloudlets based recommendation and conventional cloud based recommendation, we divide the whole recommendation process into two parts, i.e., computing and communication parts. Then, we consider the total real-time delay combined with the process of the two parts. We define the computation time of recommendation as \( T_{comp} \). Consider that the communication process is a \( h \)-hop transmission process, the communication time is defined as the total \( h \)-hop end-to-end packet delay, and each hop service time (i.e., transmission delay) is denoted as \( T_{serv} \). For simplicity, consider all nodes in the network have even traffic load, the total communication time (i.e., end-to-end packet delay) can be approximately calculated as \( T_{com} = h \cdot T_{serv} \).

To further refine this process, we consider the computation time \( T_{comp} \) to be constructed with computing process queue time \( t_{cpro} \) and computing time \( t_{comp} \). The computing time mainly depends on the time complexity of the recommendation algorithm and hardware conditions, and the service time series can be treated as a general distribution. Besides, as mentioned above, the communication time is calculated with multi-hop transmission, and for each hop transmission, we divide it into processing queue time \( t_{qpro} \) and transmission time \( t_{trans} \). The one hop service time can be calculated as \( T_{serv} = t_{qpro} + t_{trans} \). In order to simply analyze the whole process, we use two queuing processes to model the entire whole process as shown in Fig. 3.

We divide the arrival recommendation tasks of the computing queue into two parts, one comes from the passive user inquiry process and the other is generated by the periodically active recommendation process. Assume that the passive inquiry recommendation task from users is
a Poisson process, and the service time of each cloudlet computing service is a general distribution. Then, the whole process can be treated as an M/G/1 queue. With the arrival of periodically active recommendation tasks, the active computing process should be a D/G/1 queue, according to the formula of Khinchin-Pollaczek, the average packet queue number of both $N_{D/G/1}$ and $N_{M/G/1}$ can be obtained as follows:

$$N_{D/G/1} = N_{M/G/1} = \rho + \frac{\rho^2}{1 - \rho} \cdot \frac{1 + c_b^2}{2}.$$ (5)

Here $\rho$ is the utilization of the queue and $c_b$ is the coefficient of variance (CoV) of the service time.

By Little’s Law, given the arrival rate $\lambda$, the average packet waiting time $t_{pro}$, which is the packet delay of the computing queue, can be derived as

$$t_{pro} = \frac{N}{\lambda} = \frac{2\rho - \rho^2(1 - c_b^2)}{2\lambda(1 - \rho)}.$$ (6)

Here, the arrival rate $\lambda$ is combined with both active and passive query arrival tasks. And for single service queue, $\rho$ is equal to the product of $\lambda$ and the service time of the queue, which is equal to $t_{comp}$. Then, the final average computation time $T_{comp}$ is calculated as follow:

$$T_{comp} = t_{pro} + t_{comp} = \frac{4\rho - \rho^2(3 - c_b^2)}{2\lambda(1 - \rho)}. \quad \text{ (7)}$$

Following the computation of the recommendation algorithm, the recommendation packets are generated from each recommendation task. The departure process of those generated packets from the computing queue can be treated as a general independent, interarrival time distribution. Then, the packets go into the transmission queue for transmission, and this process can be treated as a G/G/1 queue

Besides the generated recommendation packets, there are also some relayed packets from other cloudlets. Those packets come from the transmission of other nodes using IEEE 802.11, which can be approximated as a Poisson process [31]. Then, the transmission queue is constructed by generated packets and relayed packets. Thus, the transmission process can be viewed as a confluence of data arrival process.

We can obtain the average transmission time $t_{pro}$ as $(t_{pro} + t_{tran})$, and the average packet waiting time for the transmission queue can be calculated by using the Kingman’s formula. Then, the average service time is expressed as follow:

$$T_{serv} = \frac{\lambda \rho^2(c_a^2 + c_b^2)}{2(1 - \rho)}.$$ (8)

Here, $c_a$ denotes the CoV for arrivals. And the whole end-to-end transmission delay can be calculated as:

$$T_{comu} = \sum_{i=1}^{h} \frac{h_{i} \rho^2(c_a^2 + c_b^2)}{2(1 - \rho_i)}.$$ (9)

From Eqs. 7 and 9, we can find that the main factors to the transmission and computation delay are $\lambda$, $\rho$, and $h_i$, where $\rho$ is decided by $\lambda$, and queue service time $t_{tran}$ and $t_{comp}$. If we want to reduce the final delay of recommendation process and improve the real-time performance, from those equations, it seems that we can decrease packets arrival rate $\lambda$, or the computing and transmission time, or the transmission hop number $h$. Inspired by this conclusion, we can consider the following three methods to meet our purpose.

By using edge computing through cloudlets, each user can communicate with computation center with less hops. We can also reduce the number of service users and decrease the packets arrival rate $\lambda$ via dividing the whole service area from one central cloud to many edge cloudlets. Moreover, the computing time of each cloudlet is also decreased through splitting the whole data set into different areas. From our proposed cloudlets-aided recommendation, almost all the main factors of recommendation delay are decreased. Hence, our proposal can lead to much better real-time performance.

5.2 Computational Complexity Analysis

In our proposed Adaptive UAV-Mounted Cloudlet-Aided recommendation algorithm, there are two components, i.e., cloudlet deployment and adaptive recommendation parts. And the time complexity of these two parts is calculated as follows.

The first part is shown in Alg.1. According to Eq.4, the time complexity of Step 4 can be calculated as $O(|H| \cdot |N + H + 1|)$. Because $N + H + 1$ is equal to total number of vertives, $|V|$, in network and $H < |V|$, the upper bound of Step 4 is $O(|V|^2)$. Then, the time complexity of Alg.1 is $O(|V| \cdot |B|)$ where $|B|$ denotes the number of combination iteration set and can be calculated as $|B| = A(N,H) = N! / H!$. As $N! / H! < N^N < |V|^{|V|}$, the upper bound of the time complexity of Alg.1 is $O(|V|^{|V|} \cdot |V|^2) = O(|V|(|V|^2 + 2))$. From complexity of Alg. 1, the time complexity is a little high. However, this deployment is executed only once at the initial part, and will not affect the performance of the adaptive recommendation.

For Alg. 3, we calculate the time complexity of Step 14. From the algorithm described on [30], the time complexity of interest and preference probability on both user side and location side mainly depends on the data set of $D_u$ and $D_l$ and also be affected by content set $C$. By considering the topic set of all topics is $Z$, the time complexity on both sides can be simply calculated as $O(|D_u| \cdot |C| \cdot |Z| + |D_l| \cdot |C| \cdot |Z|)$. Consider all the $D_l$, $D_u$, $C$ and $Z$ are part of the total data set $D$, and $|D_u|, |D_l|, |C|, |Z| < |D|$. Then, the upper bound can be denoted as $O(|D|^4)$. Moreover, in Alg.3, the two loops are mainly decided by active recommendation time interval $t_i$ and data base update time interval $T$. Thus, we can simply calculate the total time complexity as $O(|D|^{4t_i/T})$. 

Fig. 3. The queuing process of recommendation at a given node.
From the complexity of cloudlet-aided adaptive recommendation algorithm, we can see that, the main factors of complexity are the size of data set and two kinds of time interval. This means the time complexity of both using cloudlets and central cloud is at the same level. Because of the splitting of data set into many cloudlets, the size of data set can be decreased in the central cloud. Thus, from the real-time performance and time complexity analysis, the cloudlets-aided adaptive recommendation outperforms the conventional cloud based one. Furthermore, in the following section, we further compare the performance of our proposal with conventional algorithms based on simulations.

6 EXPERIMENTAL EVALUATION

In this section, we present the performance evaluation of our proposal based on computer simulations. In particular, we compare the performance of our proposed adaptive recommendation system with two conventional recommendation systems. The first conventional recommendation system is assumed without active recommendation. The second conventional recommendation system uses adaptive recommendation, however, without employing cloudlets.

6.1 Comparison of Proposal With Conventional Recommendation Systems

For sake of simplicity in our conducted simulations, we assume two user mobility models, namely by walking and driving vehicles, respectively. In the different models, the users have different moving strategies and moving speeds. They check in and present queries for recommendation in a random fashion. We define the recommendation to be valid when the recommendation result pushed to the user contains the nearest recommendation item to be within the threshold distance $d_v$. Otherwise, this recommendation is referred to as an invalid one. Based on the valid number of recommendations, we further define the recommendation precision. Then, We compare the recommendation precision between random queries and responses based recommendation system and our proposed adaptive recommendation algorithm.

Our considered simulation scenario is described as follows. A network with a fixed cloud, several cloudlets, and multiple users is constructed with the cloudlet deployment algorithm described in Sec. 4.1. The entire network covers a diameter of 2 km of land, with 256 users, 16 UAVs, and 4 cloudlets. We set the fixed cloud, cloudlets, and UAVs to have the same transmission range. When the users follow the walking mobility model, the moving speed is 5 km/h (1.389m/s). On the other hand, in case of the user driving a vehicle, the moving speed is set to 60 km/h (16.667m/s). Additionally, in this network, we have 10 out of 30 venues to recommend to users each time. When a user expresses a query and receives the relevant recommendation list, if the nearest venue $\sigma$ in the recommendation list is within a threshold distance $d_v$ from the user $u_i$ (i.e., $d(u_i, \sigma) < d_v$), we consider this recommendation to be valid. After time duration $t$, the percentage of valid user number out of the total user number is defined as a delay bounded precision $P_{delay}$. Consider we can simply define this delay bounded precision as:

$$f(i) = \begin{cases} 1 & d(u_i, \sigma) < d_v, \\ 0 & d(u_i, \sigma) \geq d_v, \end{cases}$$

$$P_{delay} = \frac{\sum_{i=1}^{R} f(i)}{R}.$$  (11)

For ease of description, we refer to the aforementioned delay bounded precision of recommendation, simply as the recommendation precision throughout the remainder of the paper. In case of the passive recommendation, the user checks in and sends a query to the recommendation system whenever deemed required. Here, we set this passive query as a Poisson stochastic process with random frequency within the maximum value, $f_i$. $f_i$ is set to 1/1000s, which means that the user will check and ask a passive recommendation randomly but at least once within 1000s. When the recommendation is received, regardless of the active or passive one, the user has a probability $p_t$ to change her randomly moving strategy to approach the nearby venues indicated in the recommendation result list. Unlike the conventional passive recommendation, the proposed adaptive recommendation system includes both passive and active models. In other words, in our proposal, the users have more chances to change their moving strategies. The active recommendation trigger interval $t_i$ in the active recommendation model is set to 1s. We simulate the user moving and recommendation process in the above configured scenario using C++, and the simulation results are discussed below.

Fig. 4 demonstrates the comparison between the passive and active recommendation systems. For ease of notation, we refer to the conventional passive recommendation as “pa-rec” while the proposed adaptive recommendation is referred to as “ad-rec”. Fig. 4(a) demonstrates the situation of recommendation precision in both “pa-rec” and “ad-rec” with the change in the process time $t$ when the users adopts to follow the walking mobility strategy. The result indicates that the growth of the valid number of recommendations by employing “ad-rec” is much faster than that by “pa-rec”. Furthermore, after half an hour, the precision of recommendation using “ad-rec” approaches the maximum while that incurred using “pa-rec” increases rather slowly. Fig. 4(b) shows a similar trend of recommendation precision when the users adopt the driving vehicle mobility strategy. In Figs. 4(c) and 4(d), we grab a set of data in a moment when the simulation runs for a duration of 1000s, and show how the recommendation precision changes with the variation of the threshold distance $d_v$. From those figures, we can notice that regardless how the value of $d_v$ is changed, the proposed “ad-rec” always outperforms the traditional “pa-rec” method in terms of the recommendation accuracy.

6.2 Comparison of the Adaptive Recommendation Using Cloudlets and Without Using Cloudlets

Until now, the simulation results have indicated that the proposed adaptive recommendation can significantly improve the recommendation precision. Furthermore, the higher the active recommendation rate (i.e., the smaller the recommendation interval $t_i$), the higher the precision of recommendations. However, this also means that the LBSN cloud server should carry out the recommendation
computations more frequently which increases its computational burden while significantly increasing the network traffic load. In order to alleviate this situation, we proposed the cloudlet-based recommendation mechanism to efficiently balance the computing and traffic loads into each cloudlet, and reduce the overall traffic congestion in the entire network. Now, we compare our proposed cloudlet-based adaptive recommendation algorithm (referred to as “cloudlet-rec” for brevity) with the conventional cloud-based recommendation method dubbed “cloud-rec”.

For this comparison, in our configured experimental environment described earlier, we set the packet size of both the query and response information to 1024bit. All users, UAVs, and cloud connections are assumed to be over wireless links. The bandwidth of the link between the user and UAV is set to 1Mbps. On the other hand, the bandwidth within the UAVs and cloudlets is set to 10Mbps. The topology of the network is considered to be same as the earlier delineated topology. Because the traffic load is quite similar in case of both walking and vehicular mobility strategies of the users, we only show the result of the walking mobility model. Also, we ignore the packet loss by assuming unlimited buffer at each node. The active recommendation trigger interval \( t_i \) is set to 1s.

For simplifying the simulation, We obtain the recommendation with fixed time \( t = 1000\)s, \( t_i = 1s, p_t = 0.1, f_t = 1/1000s \).

The throughput of the network is considered to be same as the earlier delineated topology. Because the traffic load is quite similar in case of both walking and vehicular mobility strategies of the users, we only show the result of the walking mobility model. Also, we ignore the packet loss by assuming unlimited buffer at each node. The active recommendation trigger interval \( t_i \) is set to 1s.

For simplifying the simulation, We obtain the recommendation computing time by taking a part of a real dataset [32].

Fig. 4. Precision comparison of proposed adaptive recommendation (ad-rec) with conventional passive recommendation (pa-rec) in walking and driving moving strategies.

Fig. 5. Network performance comparison of the packet delay and throughput with cloud-based and cloudlet-based algorithms.
comprising 50000 events, and directly running the recommendation algorithm (discussed in Sec. 3.2) for the dataset. The computations are conducted on a workstation with an Intel core i7 3.60 GHz processor and 16GB Random Access Memory (RAM), and the average values are used as result. The computing time of the recommendation process with the whole 50000 events inside the dataset is found to be 125ms. Then, we divide the dataset into 16 parts based on different areas such that the computing time of each part is 5ms in average. As shown in Sec. 4.2.2, each cloudlet only contains one part of the database as its own area. Then, we apply the adaptive recommendation in both cloud-based and cloudlet-based algorithms. The results of packet delay and throughput are shown in Fig. 6.

Next, Fig. 5(a) demonstrates the packet delay with cloud-based and cloudlet-based recommendation algorithms with fixed $t_i = 1s$, $p_i = 0.1$, $f_t = 1/1000s$. We can notice that with time going on, the delays of both types of recommendation increase because the congestion in network has already occurred. However, the speeds at which the delays increase in the two algorithms are different. The proposed cloudlet-based recommendation always outperforms the cloud-based one in the event of a congestion. Moreover, when a network congestion occurs, the throughput always approaches the upper bound (i.e., the bottleneck) and will not increase further with time. Additionally, Fig. 5(b) shows the obvious advantage of cloudlet-based algorithm in terms of throughput.

Furthermore, with the delay and throughput analysis, we can notice that, the high delay of recommendations may cause recommendation precision to decrease. For example, when the cloud receives the location information of user in time $t$, and performs the location based recommendation with delay $T_{comp}+T_{comu}$ to the user, the decision arrives to the user at $t'$. The user is, however, already in a new location in time $t'$, and the recommendation based on the old location may lead the user to a wrong place. And the recommendation precision decreases with the increase in the delay $(T_{comp}+T_{comu})$. In Figs. 6(a) and 6(b), we compare the recommendation precision with different numbers of users between cloud-based and cloudlet-based algorithms in case of the walking and driving scenarios, respectively. Here, the parameters of the network are set as $t = 1000s$, $t_i = 1s$, $p_i = 0.1$, $d_v = 300m$, $f_t = 1/200s$.

**Fig. 6.** Recommendation precision comparison with different numbers of users between cloud-based and cloudlet-based algorithms in case of the walking and driving scenarios, respectively.

7 CONCLUSION

Location Based Social Networks (LBSNs) are becoming increasingly popular and derived numerous applications recently. However, the high mobility of LBSN users and the need to quickly provide access points in their interest zones create a unique challenge. In this paper, we discussed how the UAVs can appear as a viable candidate to quickly construct a meshed offloading backbone to support the LBSN data sensing and relevant data computations in the LBSN cloud. Also, the concept of cloudlet was proposed to efficiently deploy edge-computing at the UAVs in the considered LBSN in order to reduce computing burden at the cloud servers and alleviate the network congestion. Depending on such a UAV-mounted, cloudlet-aided LBSN, a new adaptive recommendation mechanism was proposed to further improve the accuracy of location-based recommendation system. The computational complexity and communication overhead of our proposed adaptive recommendation are analyzed. Computer-based simulation results demonstrated the effectiveness of our proposed recommendation mechanism in the UAV-mounted, cloudlet-aided LBSN in terms of high recommendation precision and throughput as well as low packet delay.
ACKNOWLEDGMENT

This work was supported (in part) by the Japanese Ministry of Internal Affairs and Communications in R&D on Communication Network with Efficient Frequency Utilization for Unmanned Aircraft Systems (UAS) for the duration of 2016 to 2018.

REFERENCES


Feng Xiao Tang (S’16) received the B.E. degree in measurement and control technology and instrument from the Wuhan University of Technology, Wuhan, China, in 2012 and the M.S. degree in software engineering from the Central South University, Changsha, China, in 2015. Currently, he is pursuing the Ph.D. degree in the Graduate School of Information Sciences (GSIS) at Tohoku University, Sendai, Japan. His research interests are unmanned aerial vehicles system, game theory optimization and machine learning algorithm.

Bomin Mao (S’15) received the BSc degree in telecommunications engineering and M.S. degree in software engineering at Xidian University, China, in 2012 and 2015, respectively. Currently, he is pursuing Ph.D. degree at the GSIS, Tohoku University. His research interests are unmanned aerial vehicles system, and the application of unmanned aerial wireless relay networks and the application of unmanned aerial wireless relay networks.

Zubair Md. Faddullah (M’11, SM’13) is serving as an Associate Professor at the GSIS, Tohoku University. His research interests are in the areas of 5G, social networks, smart grid, and network security. He was a recipient of the prestigious Dean’s and President’s awards from Tohoku University in March 2011 for his outstanding research contributions. He was also recipient of the IEEE Communications Society Asia Pacific Board Outstanding Young Researcher Award in 2015 and the NEC Tokin foundation award in 2016.

Bomin Mao

Nei Kato (M’03, A’04, SM’05, F’13) Nei Kato received his Bachelor Degree from Polytechnic University, Japan, in 1986, M.S. and Ph.D. Degrees in information engineering from Tohoku University, in 1988 and 1991 respectively. He joined Computer Center of Tohoku University as an assistant professor in 1991, and was promoted to full professor position with Graduate School of Information Sciences, Tohoku University, in 2003. He became a Strategic Adviser to the President of Tohoku University in 2013 and the Director of Research Organization of Electrical Communication (ROEC), Tohoku University in 2015. He has been engaged in research on computer networking, wireless mobile communications, satellite communications, ad hoc & sensor & mesh networks, smart grid, and pattern recognition. He has published more than 300 papers in peer-reviewed journals and conference proceedings. He currently serves as a Member-at-Large on the Board of Governors of the IEEE Communications Society, the Chair of IEEE Communications Society Sendai Chapter, a Vice Chair of Fellow Committee of IEEE Computer Society (2016), a member of IEEE Computer Society Award Committee (2015-2016) and IEEE Communications Society Award Committee (2015-2017), the Editor-in-Chief of IEEE Network Magazine (2015-7), the Associate Editor-in-Chief of IEEE Internet of Things Journal (2013-2015), an Area Editor of IEEE Transactions on Vehicular Technology (2014-). He has served as the Chair of Satellite and Space Communications Technical Committee (2010-2012) and Ad Hoc & Sensor Networks Technical Committee (2014-2015) of IEEE ComSoc respectively, the Chair of IEICE Satellite Communications Technical Committee (2011-2012). His awards include Minoru Ishida Foundation Research Encouragement Prize (2003), Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee (2005), the FUNAI information Science Award (2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion (2008), the IEICE Network System Research Award (2009), the IEICE Satellite Communications Research Award (2011), the KDDI Foundation Excellent Research Award (2012), IEICE Communications Society Distinguished Service Award (2012), IEICE Communications Society Best Paper Award (2012), Distinguished Contributions to Disaster-resilient Networks R&D Award from Ministry of Internal Affairs and Communications, Japan (2014). Outstanding Service and Leadership Recognition Award 2016 from IEEE Communications Society and Best Paper Awards from IEEE ICC/GLOBECOM/WCNC/VTc. Besides his academic activities, he also serves on the expert committee of Telecommunications Council, Ministry of Internal Affairs and Communications, and as the chairperson of ITU-R SG4 and SG7, Japan. Nei Kato is a Distinguished Lecturer of IEEE Communications Society and Vehicular Technology Society. He is also a fellow of IEICE.

Fumie Ono received her B.E., M.E., and PhD degrees in Electrical Engineering from Ibaraki University, in 1999, 2001, and 2004, respectively. She was with the Faculty of Engineering, Tokyo University of Science, from 2004 to 2006 as a research associate. From 2006 to 2011, she was an assistant professor in the Devision of Electrical and Computer Engineering, Yokohama National University. She is currently with Dependable Wireless Lab, Wireless Network Research Institute, NICT, Yokosuka, Japan.

Ryu Miura (M’93) received the B.E., M.E., and PhD degrees in Electrical Engineering from Yokohama National University, Yokohama, Japan, in 1992, 1994, and 2000, respectively. He joined Radio Research Laboratory, Tokyo, Japan in 1994, which was reorganized to National Institute of Information and Communications Technology (NICT) in 2004. Now he is a director of Dependable Wireless Lab, Wireless Network Research Institute, NICT, Yokosuka, Japan, where he works on disaster-resilient wireless networks and the application of unmanned aerial wireless relay networks. Dr. Miura is also a member of IEICE.

Nei Kato

Fumie Ono

Ryu Miura