Federation University ResearchOnline

<https://researchonline.federation.edu.au>

Copyright Notice

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Kaisar, S., Kamruzzaman, J., & Karmakar, G. (2019). A dynamic content distribution scheme for decentralized sharing in tourist hotspots. *Journal of Network and Computer Applications*, 129, 9–24.

Which has been published in final form at:
<https://doi.org/10.1016/j.jnca.2018.12.011>

See this record in Federation ResearchOnline at:
<https://researchonline.federation.edu.au/vital/access/manager/Index>

A Dynamic Content Distribution Scheme for Decentralized Sharing in Tourist Hotspots

Shahriar kaisar¹, Joarder Kamruzzaman², and Gour Karmakar^{2,1}

Abstract

Decentralized content sharing (DCS) is emerging as a suitable platform for smart mobile device users to generate and share contents seamlessly without the requirement of a centralized server. This feature is particularly important for places that lack Internet coverage such as tourist attractions where users can form an ad-hoc network and communicate opportunistically to share contents. Existing DCS approaches when applied for such type of places suffer from low delivery success rate and high latency. Although a handful of recent approaches have specifically targeted improvement of content delivery service in tourist spot like scenario, these and other DCS approaches do not focus on contents' demand and supply which vary considerably due to visitor in-and-out flow and occurrence of influencing events. This is further compounded by the lack of any content distribution (replication) scheme. The content delivery service will be improved if contents can be proactively distributed in strategic positions based on dynamic demand and supply and medium access contention. In this paper, we propose a dynamic content distribution scheme (DCDS) considering these practical issues for sharing contents in tourist attractions. Simulation results show that the proposed approach significantly improves (7~32%) delivery performance.

I. INTRODUCTION

Content sharing using smart mobile devices is becoming highly popular as the devices provide a convenient way to generate and share contents (e.g., music, videos, news items) on the fly at any place, at any time. The shared contents help in fulfilling information needs, providing instant recommendations, promoting businesses and circulating education materials. In addition, content sharing is useful during natural disaster or in battlefield [1]. The sharing process can use three types of architecture- (i) centralized, (ii) decentralized and (iii) hybrid architecture. Existing sharing techniques mostly use either a centralized or hybrid architecture, which needs constant Internet connection for locating and delivering contents. However, a decentralized content sharing (DCS) technique is significantly important for a number of reasons. DCS is the only architecture capable of providing content sharing service without the requirement of Internet coverage, truly facilitating sharing content anywhere, anytime. This feature is highly important as places like

1. Faculty of Information Technology, Monash University, Australia.

2. School of Science, Engineering and Information Technology, Federation University Australia, Victoria, Australia.

E-mail: shahriar.kaisar@monash.edu, joarder.kamruzzaman@federation.edu.au and gour.karmakar@monash.edu

camping sites, remote tourist spots are unlikely to have Internet connectivity and many parts of the world still have limited or unreliable connectivity [2]. Since DCS does not generate IP traffic but still supports content sharing, it can also play a major role to reduce Internet bottleneck as recent report suggests IP traffic from smartphone will exceed that of PC by 2021 and Content Delivery Networks will contribute to 71% of the total IP traffic [3]. Another attractive feature is that DCS provides a free-of-cost platform by utilizing peer-to-peer communication among devices, while other architectures may turn out be costly if cellular data is the only means of connectivity at the place of sharing as data plans are still costly in many parts of the world (e.g., Australia where a A\$49 mobile plan allows only 2GB data [4]). Most importantly, in many circumstances (e.g., natural disaster), the centralized and hybrid approaches become ineffectual for spreading emergency information, leaving DCS as the only viable option.

In DCS, users within the close vicinity form an ad-hoc network and share contents using peer-to-peer communication through Wi-Fi/Bluetooth connection. Employing such DCS approaches in irregular meeting places such as a tourist spot or camping site can enhance the visiting experience of tourists as many places lack Internet connectivity (e.g., Grand Canyon national park in the USA and Sahara Desert in Africa [5]). In addition, DCS can also aid in successful content delivery and reduce network load during influencing external events (e.g., major tournaments, festivals, disaster) when Internet traffic rises sharply and the network infrastructure suffers to deliver satisfactory service [6], [7]. Although a handful of approaches employed DCS in tourist spot type scenarios [8]–[10], they did not consider dynamic content distribution, which is important to provide acceptable content delivery service to match varying content demand with available supply in a dynamic network. They also did not estimate demand and supply for contents, which varies considerably in a tourist spot. The demand for contents rises sharply for higher number of tourist visits during holidays or occurrence of influencing external events. The content delivery success rate and content access time will suffer if the demand for content surpasses the available supply. Therefore, adoption of a dynamic content distribution scheme considering dynamic content demand and supply is essential for the practical implementation of DCS in tourist spot type scenario.

Demand-supply-and-distribution is an extensively studied topic in economics and widely exercised for product distribution in business and retail organizations. The basic idea is to increase supply in response to an expected high demand. In addition, the delivery of products at the right time to the right people depends on the selection of appropriate distribution locations [11]. The guaranteed delivery requires the communication infrastructure (e.g., roads and highways) to be available from the distribution center to the demanding customers who are reachable within minimum transportation cost [12]. In case of DCS in tourist like spot, if we can estimate future content demand and accordingly distribute contents proactively in strategically positioned nodes so that any possible request from a nearby node is better served considering

the optimization of demand-supply and distribution based on economic theory along with transmission channel condition, it is highly likely that both delivery and access time will be improved. However, serving high demand for contents with limited resources is a multi-dimensional problem and a content distribution policy needs to address: (i) future prediction of demand, (ii) determination of available supply and (iii) the distribution of contents considering dynamic demand-supply and contention among nodes for wireless medium access.

A few approaches in existing literature have employed demand aware content distribution [13]–[15]. However, they were proposed for centralized or hybrid architecture-based content sharing and assumed that nodes have identical demand. This is impractical since in a tourist spot like scenario, users of different background and interest come together and generate varying demand for different contents. They considered neither dynamic content supply which is essential as tourists enter or leave a particular point-of-interest (POI) after performing activities there, nor contentions among participating nodes in content distribution, both of which are important for improving delivery success and latency. There also exist some approaches for handling content replication and caching issues in wireless ad-hoc networks. In replication based approaches [16]–[18], predictable encounters or movement patterns of participants were used, while caching based techniques [19], [20] mostly employed the principle of locality. None of them uses dynamic demand and supply nor distribute contents in appropriate locations considering contention among nodes to reduce the impact of collisions.

In this paper, we propose an economic theory based dynamic content distribution scheme (DCDS) considering the dynamic demand-and-supply and contention among nodes for decentralized content sharing in tourist spot like visiting places. We consider that nodes are divided into multiple local groups and one of the group members is selected as the local administrator. The administrator captures the dynamic demand of the group members for different contents using users' interest and personal preference, content popularity, and the impact of influencing events on content demand. Available supply for contents is captured using the network coverage, stay probability of node(s) holding the contents and their available energy. Using this information, an administrator determines whether any additional distribution is needed to meet the future demand. To bring the benefits of economic theory-based distribution system, our proposed distribution scheme uses an optimization model for determining appropriate nodes for proactive distributions, considering their ability to meet future demand and contention in the network due to the placement of the distributed content. Adoption of such dynamic content distribution policy yields higher delivery success rate and lower latency. The novelty of the proposed approach lies in the use of dynamic content distribution scheme considering both dynamic demand and matching supply for contents. Similar to economic modeling of demand-supply-and-distribution, the proposed method considers practical and important parameters, which

improve content delivery service and reduce content access delay. In summary, the following contributions are made in this paper:

- dynamically captures the demand and supply for a particular content using the information that are readily available and can be gathered on-the-fly.
- proposes a dynamic content distribution scheme that increases supply to meet demand through identification of strategic locations that are best suited to hold the distributed contents, considering demand coverage and contention among nodes.
- extensive simulation confirm that the proposed approach brings significant improvement in delivery success and latency compared to previous methods.

The rest of the paper is arranged as follows: Section II provides a brief description of related works. Section III describes the network model used in this study while Section IV presents the proposed dynamic content distribution scheme for DCS. Section V evaluates the performance of the proposed scheme and finally, Section VI presents some concluding remarks.

II. RELATED WORKS

In decentralized content sharing, nodes are organized into groups based on their mutual interest and one of the group members acts as an administrator [9], [21]. Upon generating a request, the requester forwards it to immediate neighbours. If the neighbours fail to deliver any matching content, the request is forwarded to its group administrator who directs it to an appropriate content holder. Since a requester first checks with neighbouring nodes for content availability, the main aim of a content distribution scheme is to proactively place a content near potential requesters so that delivery success rate and latency can be improved. Factors to consider in this regard are dynamic demand assessment, available resource management, node mobility and potential impact of replica placement.

Existing approaches for content replication/caching can be broadly categorized into: (i) community-based and (ii) community independent approach. The former considers that nodes are part of a community and replication is done to help community members, while the latter suggests that nodes independently decide whether to replicate a content or not considering personal and neighbours' benefit. Note that, multi-copy forwarding strategies for mobile ad-hoc networks [22] also replicate a content to ensure better content delivery. However, they assume that the destination for a requested content is already known and hence the focus is to deliver the content only, which is significantly different in our case where contents need to be proactively placed into strategic positions before the requests are made. Therefore, these strategies are not usable here.

ContentPlace [18] suggests that a node may belong to several communities (e.g., working or family community) and takes replication decisions considering the expected visiting sequence to such communities

(e.g., most frequently visited, most likely next, and current). In contrast, [23] takes replication decisions considering frequent encounters, while past request pattern of the encountered nodes is considered in [24]. Although these approaches are suitable for work-place type scenarios where nodes frequently meet other group members (e.g., in office or university campus), they are inapplicable in tourist spots where nodes mostly meet strangers, form spontaneous communities and are unlikely to encounter the same set of nodes frequently.

PodNet [25] is a community independent replication scheme which suggests that nodes should divide the buffer space into private and public spaces. The former is used for replicating contents of personal interest, while the latter is utilized for replicating contents that might be of interest to future encountering nodes. Upon contact, nodes first exchange contents reflecting personal interest and then the remaining time is used for replicating contents in the public buffer space. Due to buffer space constraint, decisions like which content to replicate and how to replace contents in the public space become important. To address this issue, Ma *et al.* [26] proposed a scheme where encountering nodes determine which content to be replicated based on the history of encounter and content popularity. Works in [16], [17] consider meeting frequency and regularity in movement patterns to perform content replications. Recently, Wu *et al.* [27] used data transmission capability and service efficiency calculated in terms of successful delivery ratio over a given period, along with encounter duration and frequency among nodes to handle cache management. However, these approaches are ineffectual for tourist spots type scenarios because of infrequent encounters and irregular movement patterns.

Gao *et al.* [28] proposed a scheme for mobile ad-hoc network where contents are replicated in multiple central locations called network central locations (NCLs), which are selected based on a probabilistic metric that evaluates data transmission delay in delay tolerant networks. Whenever a node generates some content, it replicates the content in its nearest NCL. For content access, requests are forwarded to NCL, who then delivers the content. Although this scheme ensures easy access to NCLs for uploading and downloading contents, it also indicates that the NCLs will be congested when multiple nodes simultaneously make requests. Similar to this approach, Ma *et al.* [29] proposed a learning automata-based caching policy for delay-tolerant networks, where a set of caching nodes (CNS) is selected through a voting phase. In this case, each node assigns some weights to other nodes to show their preferred caching location based on contact frequency. Nodes with higher weights are selected at the end of the voting phase to form the CNS to hold cached contents for other nodes. However, this approach does not address selection of appropriate contents to be cached at CNS.

Cooperative caching schemes increase content availability by caching the most popular contents collaboratively [19], [20], [30], [31]. Following the working principle of a cache (principle of locality), popular

contents are identified based on the frequency or recency of past request patterns. Radenkovic *et al.* [30] also used social metrics (e.g., betweenness, centrality and tie strength), remaining energy and available buffer space to determine appropriate nodes to hold the cached contents. On the other hand, cache replacement policies were proposed by Naz *et al.* [31] where each node selects a suitable replacement node before moving away from a region. However, absence of consideration of dynamic demand-supply in these approaches is likely to create abundant copies of popular contents while neglecting contents which may become ultimately popular or might be of interest to some nodes or of interest due to some events.

To address this, a few distribution schemes utilize content demand. For example, Shakkottai and Johari [13] and Alasaad *et al.* [14] calculate the evolution of demand for contents using the word-of-mouth propagation. In contrast, Meng *et al.* [15] calculates the demand for a content using node accessibility and probability of receiving a request for that content. They have also proposed a population ecology-based replication policy for unstructured peer-to-peer network. Recently, Ioannidis *et al.* [32] have also used observed request frequency to calculate demand and a greedy eviction policy with path replication to determine optimal content placement. However, the above approaches are not applicable in our case as they rely on Internet connection for content delivery, assume nodes to be always accessible if they are online and consider the presence of a static storage (i.e., server) to permanently hold contents. They also do not consider node mobility which is an important aspect of our considered scenario. In this regard, Reich and Chaintreau [33] considered a dynamic network and calculated content demand from popularity distribution and performed replication based on the added utility that a replica can provide in terms of delivery time. Although content popularity is an important measure for determining its demand, other factors such as personal preference, user interests and impact of recent events are also important indicators in determining a content's demand which were not addressed in [33]. Similarly, no consideration was given to content supply or contention among the participating nodes to reduce collision while placing the replicated contents.

Recently, caching techniques for device-to-device (D2D) content sharing have received significant attention [34]–[38]. The basic idea is to leverage the information from physical domain and social domain to allow co-located users to exchange contents without using the existing social network servers or base stations through Internet connection. This way D2D communication reduces network traffic. However, these approaches use predictable movement patterns [34], frequent encounter and exiting social network relationship [35], [39] to determine appropriate caching locations, which is ineffectual in our considered scenario where nodes demonstrate spontaneous movement and mostly meet strangers. In contrast, Sun *et al.* [40] used user preference and content popularity to determine the probability of a user requesting that content and accordingly used a collaborative caching policy to place that content. A similar approach is proposed in [36], where contents are proactively cached based on their popularity and current caching

distributions. Yi *et al.* [37] also used content popularity along with mobility and social relationship of the carrier node to manage caching decisions. In contrast, Xu *et al.* [38] used a time threshold based reactive caching strategy to determine whether to cache a content in an intermediate node while forwarding it back to the requester. However, the above approaches did not consider impact of event on content access, which is essential as the demand for relevant content rises during influencing external events [7]. In addition, they also did not consider medium access contention, which is important for enhancing content delivery service.

While the above works deal with content distribution in general within the context of DCS, a few articles have also addressed decentralized content sharing specially in tourist places [8]–[10]. However, they mostly focused on interest calculation, group management and content delivery. They neither addressed content distribution nor dynamic demand and supply for contents, which is essential to improve content delivery service under high demand or during influencing external events. Note that our preliminary work [41], related to the work presented in this paper, proposed an intuitive content distribution method considering dynamic demand and supply for contents. However, it lacked proper consideration of: (i) demand expected from currently *uninterested* but potentially would be interested users, (ii) probability of receiving request from the demanding nodes, and (iii) contention among nodes to reduce the impact of collision during the distribution process. These factors are particularly important as (i) and (ii) are needed for accurate demand calculation, while (iii) is essential in guaranteeing successful delivery, minimizing latency and keeping network congestion at a minimal level.

III. NETWORK MODEL

We consider decentralized content sharing in a tourist spot where the whole area is divided into several POIs. Each POI hosts activities available within the region and can be identified using any map application and/or utilizing online information provided by tourist portals or visitor information center. For example, a tourist spot can have POIs such as ‘Fishing’, ‘Camping’, ‘Food and shopping’, and ‘Bush Walking’ that represent activities available in those regions. After arriving at a tourist spot, users enter one of these POIs based on their interest, perform relevant activities and then either move on to other POIs or leave the spot.

Note that, our formulation does not depend on any particular mobility model for movement inside or across POIs. However, to assess the performance of the proposed approach, in the simulation random-waypoint model [42] is used (Section V). Upon entering a POI, a node calculates its own (i.e., user’s) level of interest from different sources (e.g., personal profile, online recommendation and past experience) and joins content sharing groups based on mutual interest with other nodes. The initial groups are formed on ad-hoc basis based on mutual interest of nearby nodes and the membership dynamically changes as nodes move in and out of the region.

One of the group members acts as the group administrator and perform group and content request management tasks in our scheme. The administrator nodes store relevant meta-data from other group members, which includes: list of group members along with their interest score, personal preference for content consumption, stay probability and expected stay time in the POI, available resources in terms of buffer space and remaining energy, content request probability and the administrator will also have a summary of the contents currently held by group members including content name, size, popularity or trending score, file vector with list of associated keywords and their weight, information about the event if the content is related to one. To store the above information an administrator can essentially maintain two list: one with all the information related to group members and the other with information related to the list of contents currently held by all group members. Storing and maintaining these two lists will only consume few KB of data. For example, storing a content list summary for 50 nodes with 50 contents each, consume about 100 KB of data. When a node joins a content sharing group, it will also pass its own meta-data and content information to the administrator who will update relevant group and content information lists. Group members also send updated information if there is any change (e.g., new content added). Note that there might be multiple local groups within a POI, each managed by its local administrator. This way the architecture remains decentralized without the presence of any global administrator. The administrator nodes are selected based on their probability of staying within the region, connectivity with other nodes and available resources in terms of computation capability and energy level. They also receive higher priority from other group members while obtaining requested contents as an incentive. A backup administrator is also used who takes over if the current administrator fails due to unforeseen reasons. For a collaborative content sharing scheme to work properly, some incentive mechanism and protection against malicious nodes are needed. The content distribution scheme proposed in this study is built on the DCS network model in [9], which in details describes administrator selection procedure, group management and content request handling, an incentive scheme to encourage node participation and a trust model to reduce security risks. Interested readers are directed to that study.

Figure 1 depicts DCS in a tourist spot with several POIs. Here, POI 1 depicts two overlapping content sharing groups along with the group members and their administrators. In group G1, a requester generates a request which is forwarded to the content holder via the administrator. The content holder delivers the matching content back to the requester. Although an end-to-end path for communication is shown in the figure for better understanding, such static paths are mostly unavailable in DCS due to node mobility. Node movement shown in the figure indicates that some of them will move to another POI, while others might leave the area.

Built on the network model presented above, this paper focuses on content distribution considering

Fig. 1: Decentralized content sharing in tourist attraction

dynamic demand and supply for contents, which is discussed in the following section.

IV. DYNAMIC CONTENT DISTRIBUTION

Though the content distribution problem in DCS can be compared to the demand-supply-and-distribution method for product distribution, one significant difference between them is that the volume of demand for a product usually depends on its price and income of buyers, while demand for a content depends on its popularity, interest of participating users and external event occurrence during the visiting time. This dictates modeling of demand differently in a collaborative content sharing. For content distribution, demand is considered as the number of expected content requests, supply represents content availability, and distribution refers to the placements of replicated contents. When expected demand surpasses available supply, content distribution is performed to increase supply and availability, and thereby meet the future demand. In our proposed approach, the group administrator maintains and updates group content list (i.e., list of contents for all group members) and periodically performs the dynamic content distribution (DCD) using this list. The interval between consecutive DCD cycles is denoted by θ . A lower value of θ indicates that the administrator consumes more resource for handling the distribution procedure, while a higher value suggests that the distribution scheme will fail to capture sudden change (e.g., more requests during an event). At the beginning of each distribution cycle, the administrator calculates demand and supply for each content in the group content list, determines which contents require additional distributions and identifies the group members who are best positioned for carrying the replicated contents. Afterwards, it instructs the

identified group member(s) to copy the content from one of the suppliers (i.e., content holders). Different stages of our proposed dynamic distribution scheme (demand, supply and distribution) are presented in the following sections.

A. Capturing Dynamic Content Demand

Content request is motivated by a user's interest in a particular content category. Content category refers to contents related to activities available in a tourist attraction (e.g., contents related to accommodation, food and shopping facilities, fishing, camping, bush walking) or generic media types (e.g., music, movies or news items). In addition, events may also motivate users to request relevant contents. For example, Bob is usually not interested in fishing but while visiting a tourist spot, he might learn about a fishing competition at that place and becomes interested to receive contents related to that event. Similarly, a major event (e.g., bushfire) is likely to prompt everyone to receive updated news. Note that such impact of influencing events on content demand is not considered in current literature, although it has been reported that content request increases due to an influencing event [43]. To capture both personal interest and impact of an influencing external event on content demand, the administrator calculates the demand for content α belonging to category i as,

$$D_{\alpha} = D_{\alpha}^i + D_{\alpha}^{i'} \quad (1)$$

Here, D_{α}^i represents the demand generated by users who are already interested in category i to which content α belongs and $D_{\alpha}^{i'}$ shows the expected demand by members who are currently not interested in i but are expected to become interested in α if it is related to an influencing external event. If α is not related to any event, $D_{\alpha}^{i'} = 0$ which nullifies its impact in Eq. (1). Demand for α from users interested in category i is calculated as,

$$D_{\alpha}^i = \sum_{x=1}^{|\mathbb{G}_{\alpha'}^i|} I_{\alpha}^x P^x R^x \quad (2)$$

Here, $\mathbb{G}_{\alpha'}^i$ represents the set of group members who are interested in category i , but *do not* have content α . I_{α}^x represents interest factor of node $x(\in \mathbb{G}_{\alpha'}^i)$ for α , P^x represents stay probability of node x within the group during θ duration, and R^x represents the probability of generating any content request within that duration. Since users stay in a group for a limited time, demand expected within that period is needed to be considered. Here, the interest factor indicates that a user x is interested in α , however, it is not known when x will generate a request and whether it will be staying in the current group during the current θ interval. Therefore, we calculate the demand as a combination of interest factor, stay probability and

request probability. For a single node D_α^i ranges within $[0\sim 1]$ suggesting its probability of requesting α . The calculation of each component is discussed below.

The interest factor reflects the probability of a user being interested in a particular content (e.g., a lady gaga song) given that it is already interested in that content category (e.g., music). Some users are influenced by trending topics for consuming a content, while others stick to their personal preference. The interest factor of user x for content α is calculated using both criteria as,

$$I_\alpha^x = \Upsilon^{x,i} \xi_\alpha^x + (1 - \xi_\alpha^x) \Phi_\alpha, \quad (3)$$

where, $\Upsilon^{x,i}$ shows the interest *score* of user x for content category i , and ξ_α^x and Φ_α depict the personal preference and the trending scores for α , respectively. $\Upsilon^{x,i}$ represents the generic level of interest of a user for a particular category of content. Details of such interest score calculation in a tourist spot can be found in [9]. However, to reflect personal preference and the popularity of content α , ξ_α^x and Φ_α have been included in this paper as the new aspects of interest calculation. Each group member sends its own $\Upsilon^{x,i}$ to the administrator while joining the group.

Similar to the work presented in [21], we consider that the content sharing app can automatically collect the personal preference of a user for content access using document clustering technique [44]. In this case, a file vector [21] is generated for each file from the meta-data associated with previously requested files. The file vector for content α having n number of keywords is depicted as, $v_\alpha = (k_{1\alpha}, w_{1\alpha}; k_{2\alpha}, w_{2\alpha}; \dots; k_{n\alpha}, w_{n\alpha})$, where $k_{n\alpha}$ represents n -th keyword and $w_{n\alpha}$ represents its weight in describing the file containing α . The raw weight of each keyword is extracted using text retrieval techniques [45] as, $w_{k\alpha} = 1 + \log(M_{k\alpha})$, where $M_{k\alpha}$ is the frequency of keyword k in the file. The normalized weight for each keyword associated with a file is calculated as, $\overline{w_{k\alpha}} = w_{k\alpha} / \sum_{q=1}^{q=n} w_{q\alpha}$. Finally, the weight of keyword k across all files is calculated as,

$$w_k = \frac{1}{F_\alpha} \sum_{\alpha=1}^{F_\alpha} \overline{w_{k\alpha}} \quad (4)$$

Here, F_α indicates the total number of files. Each user calculates its own set of top- n preferred keywords and their associated weights, stores them in its preference vector ($v^x = (k_1^x, w_1^x; k_2^x, w_2^x; \dots; k_n^x, w_n^x)$) and sends it to the administrator while joining a group. Similarly, the content holders also send file vectors v_α while sending their content list summary to the administrator. Note that a higher value of n introduces more traffic while a lower value might result in potential mismatch between a user's preference with a content. The administrator determines the expected personal preference score of a user x for a content α using cosine

similarity as,

$$\xi_{\alpha}^{x*} = \begin{cases} \frac{\sum_{k=1}^K w_k^x * \overline{w_{k\alpha}}}{\sqrt{\sum_{k=1}^K (w_k^x)^2} * \sqrt{\sum_{k=1}^K (\overline{w_{k\alpha}})^2}}, & \text{if } K > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Here, K represents the total number of matching keywords k between a user's preference vector v^x and a file vector v_{α} for α , while w_k^x and $\overline{w_{k\alpha}}$ show their weight in respective vectors.

Note that, it is possible to find a complete or near mismatch ($K \rightarrow 0$) between v^x and v_{α} . It will effectively nullify the impact of a user's interest in Eq. (3), which is undesirable as the mismatch might be a result of inaccurate tagging or insufficient information. Additionally, a user might still be interested in a particular content due to its generic interest in that category, and hence the personal preference score is calculated as,

$$\xi_{\alpha}^x = \max \{ \xi_{\alpha}^{x*}, \delta^{x,i} \}. \quad (6)$$

The value of $\delta^{x,i}$ depends on the distribution of a user's interest across all L categories and its relative interest in category i . It is calculated as below.

$$\delta^{x,i} = \frac{\Upsilon^{x,i}}{\sum_{l=1}^L \Upsilon^{x,l}}. \quad (7)$$

Content demand by users can also be motivated by trending topics. Trending topics refer to the issues which have received significant attention from the people at recent times. Now-a-days most of the social networking and content sharing platforms (e.g., Facebook, YouTube and Twitter) provide a list of trending topics. A list of such trending topics according to different location along with a trending score is also available online (e.g., <https://www.google.com/trends/> provides a list of trending topics and their scores between 0 and 100). The content sharing app can use such trending list to assign a trending score within $[0 \sim 1]$ to each downloaded content. Note that updated trending information is not a requirement for sharing purpose, rather just aids in more accurate calculation; hence, the app will only update such information when Internet connection is available. Such trending score Φ_{α} for content α is forwarded by the content holder to the administrator who use it to calculate interest factor using Eq. (3).

The stay probability P^x of a node x indicates its probability of staying within the current POI. A higher stay probability suggests that the node is more likely to stay during the current θ . The stay probability is calculated considering the probability distribution function (pdf) of stay time of a node in different POIs in a tourist spot. Details of P^x calculation can be found in [9].

The request probability R^x refers to the probability of a node generating a content request within the current calculation window θ . A node can calculate its own request generation rate λ_r^x and informs the

administrator. The request generation follows a Poisson process [18] and hence the administrator calculates the probability of node x generating one or more requests within θ as,

$$\begin{aligned} R^x &= P(N_\alpha^x \geq 1), \text{ [} N_\alpha^x \text{ is the number of requests by } x \text{]} \\ &= 1 - P(N_\alpha^x = 0) \\ &= 1 - e^{-\lambda_r^x \theta}. \end{aligned} \tag{8}$$

Finally, the administrator uses the value of interest factor I_α^x , stay probability P^x and request probability R^x to determine the demand for α from *interested* group members. Estimation of the demand from *uninterested* group members are discussed below.

As alluded before, it is expected that the occurrence of an influencing event will impact the uninterested users and some of them will eventually be interested in consuming related contents. Existing literature considers information diffusion among uninterested users and model the demand generated by them using the Bass diffusion model [13], [14]. However, they do not consider the presence of any influencing event to motivate users for demanding contents whereas, to attract tourists, many different types of events occur in a tourist spot. Again, they suggest that the ultimate number of interested users only depends on the initial set of interested users. We also argue that in a tourist spot, the scale of an external event determines what fraction of the uninterested users will become interested in consuming a content. For example, a small-scale event (e.g., a magic show) will have less impact than a mega event (e.g., a popular carnival). Furthermore, access to event related contents are expected to increase sharply immediately before and after the event as well as during the event. This has not been considered in literature. Based on these characteristics, we define the demand generated from uninterested group members as,

$$D_\alpha^{i'} = \Omega \Phi_\alpha e^{-\beta t^2} \sum_{x=1}^{|\mathbb{G}^{i'}|} \varphi^x. \tag{9}$$

Here, Ω shows the scale of the influencing event related to α , Φ_α is the trending score of α , β shows the shape factor and t is a time factor (explained later). $\mathbb{G}^{i'}$ is the set of group members who are currently not interested in content category i and φ^x is the affinity of node x towards a new interest category.

To determine the scale of an influencing event, we need to first match the content with a particular event and then determine the scale of that event. A number of recent works have focused on identifying events from social media and associating related contents with them [46], [47]. Becker *et al.* [46] suggests that using the features associated with a content (annotations like title, tags) and automatically generated information (e.g., content creation time), it is possible to map a content to an event. They further extended this work in [47] and suggested that collecting event features (e.g., title, time and location) from event

Fig. 2: Impact of shape factor on event scale

aggregation platforms (e.g., Last.fm, EventBrite and Facebook) enables to effectively match events with contents. We consider that the content sharing app can use these event aggregation platforms or online resources (<http://www.australia.com/en/events.html> and <http://eventful.com/> provide a list of events across Australia and USA, respectively) to collect and store information about future events and use the methods proposed in [47] to map a content to an event. The impact of events on tourism industry and tourist places has been extensively studied by researchers from the perspective of tourism industry and the economic and social impact of events on tourist spots [48], [49]. Getz [48] has ranked events into four categories based on the descending order of tourist demand and attractiveness as, (i) mega-events, (ii) periodic hallmark events, (iii) regional events and (iv) local events. Jago [49] investigated a number of events across Australia (42 in total), classified them into the above mentioned four categories and assessed their attractiveness. We use these studies as a baseline to classify an event and accordingly assign an event scale within [0~1]. The calculation of trending score Φ_α is similar to the one discussed before.

The shape factor determines the shape of the curve representing the impact of an event with time, especially immediately before or after and during that event. Some event related contents become popular on the day of the event and vanishes just after the event, while others may stay popular for a longer time. The content sharing app can use historical information to determine the value of the shape factor β within [0~1] by observing content access patterns of the user during similar events in the past. Figure 2 shows the impact of shape factor β at different time. A larger value suggests that the impact is usually short lived. Literature suggests that event related contents are usually generated and consumed from one week before

the event up to one week after the event [47]. Therefore, by default we use $\beta = 0.1$.

The time factor determines the relationship between the current time t_c and event start time t_s . Using the information gathered on event occurrence time from different event aggregation platform or online resources, the time factor is calculated as,

$$t = \begin{cases} 0, & \text{if } t_c = t_s \\ t_s - t_c, & \text{if } t_c < t_s \\ t_c - t_s, & \text{otherwise.} \end{cases} \quad (10)$$

Users' interest in different categories is reflected by their interest score. In addition, a user may become interested in a new category based on many situations. Such attribute is called the affinity of a node towards a new interest category. From historical data, a node calculates its own affinity score φ^x using the ratio of the number of requests generated for contents in uninterested categories with respect to the total number of requests. Nodes send their φ^x to the group administrator. Finally, the administrator calculates the demand $D_\alpha^{i'}$ from uninterested users using Ω , Φ_α , β , t and φ^x as per (9).

B. Content Supply

After calculating demand for a content α as per (1), the administrator calculates its available supply. In DCS, existing content holder(s) may leave the current POI or have insufficient resource to deliver α . Therefore, in contrast to existing approaches' [13]–[15] assumption on guaranteed availability of content holders, we calculate dynamic supply using the number of content holder(s), their coverage, available resources and stay probability. A higher number of content holders increase the available supply of a content. Similarly, the coverage and stay probability of the content holder(s) also impact the perceived content availability. Coverage measures whether the content holder(s) are within the reach of the potential requesters, while stay probability predicts whether they are likely to remain in the group. Remaining energy is also used to assess whether a content holder has enough resource needed for delivery. Finally, the supply for content α is calculated as,

$$S_\alpha = \sum_{y=1}^{|\mathbb{G}_\alpha^i|} C_\alpha^y P^y \Psi^y. \quad (11)$$

Here, $\mathbb{G}_\alpha^i$ represents the set of members of group G who have content α , C_α^y shows the coverage of node $y \in \mathbb{G}_\alpha^i$ for content α , P^y shows the stay probability of node y , and Ψ^y is the energy factor for node y .

The coverage value of node y for α is calculated as,

$$C_\alpha^y = h_{x,y}^1, \quad (12)$$

where, $h_{x,y}^1$ represents the number of potential requesters x who are within 1-hop distance of supplier y . Nodes maintain a neighbourhood table to store the list of their n -hop neighbours and the administrator of a group is reachable within n -hop distance. Similar to [50], we use 2-hop connectivity (i.e., $n = 2$) as a higher value of n introduces more network traffic.

Energy factor of node y is calculated based on its remaining energy and energy required to deliver α as,

$$\Psi^y = \begin{cases} 1, & \text{if } E_a^y \geq E_r^y \\ \frac{E_a^y}{E_r^y}, & \text{otherwise.} \end{cases} \quad (13)$$

Here, E_r^y shows the required energy for regular activities and delivering α , while E_a^y shows node y 's available energy. The available energy can be calculated using a ratio of remaining energy and full energy while the required energy can be calculated considering energy drain due to regular activities (e.g., SMS, call and apps) and required energy to deliver α considering its size. Details of such energy calculation can be found in [9]. Note that the users can set a minimum energy level beyond which the content sharing app will stop participating in the sharing process. By default, this value is set to 20%, which can be changed by individual user.

Following economic modelling, to successfully meet increasing demand, we need to increase the supply and identify appropriate nodes that will be able to successfully deliver the requested contents, if a replicated one is placed there. This is described in terms of content distribution in the following section.

C. Content Distribution

The administrator checks the amount of expected demand D_α and available supply S_α to determine if any additional distribution is required and distributes content α when the following condition is true,

$$D_\alpha - S_\alpha > 0. \quad (14)$$

The purpose of the distribution mechanism is to proactively place replicated copies of the content in strategic positions so that they become easily accessible to potential requesters. The administrator considers the set of group members who does not have α ($\mathbb{G}_{\alpha'}$) as the potential locations for placing replicated copies and determines the utility (discussed later) of each of those nodes. The administrator first selects the node that maximizes the utility for placing a replicated copy and remove it from $\mathbb{G}_{\alpha'}$. Afterwards, the administrator recalculates the value of demand and supply, and checks (14) to determine if any additional copy is required.

The utility value of a node reflects its ability to meet its own and one-hop neighbours' demand as well as its available resources to deliver the content when requested. The selection policy is formulated as,

demand-meeting perspective both nodes 1 and 2 are equally appropriate candidate. Now, if α is placed at node 1 or 2 and a request is generated then node 1 and 2 will keep the shared medium (i.e., channel) busy for delivering α . During this transmission time, neighbour of node 1 and 2 will be unable to communicate using the channel. Therefore, we select a node which will cause least contention for its neighbours for delivering the content. In this case, the candidate supplier node z will require t_α time to deliver α and will create no contention if none of its neighbour intend to transmit at that time. Therefore, we calculate the probability of having no contention for node z as,

$$\tau^z = 1 - \frac{\sum_{u=1}^{h_{u,z}^1} \eta_u}{\mathbb{G}}. \quad (17)$$

Here, η_u shows the probability that node u will be busy and will contend during transfer t_α by node z and $h_{u,z}^1$ is the number of 1-hop neighbours of z . To calculate the probability of node u being busy during t_α , we need to find its content exchange (delivering or getting requested content) probability and relay probability. We represent the probability of node u being busy in delivering, getting or relaying a content as,

$$\eta_u = \min \{1, (\eta_u^d + \eta_u^g + \eta_u^r)\}, \quad (18)$$

where, η_u^d and η_u^g are the probability of node u delivering and getting a content, respectively, and η_u^r is the probability of u relaying a content. Node u delivers a content if it receives a request and holds a matching content. Without loss of generality, we consider that the request arrival at node u follows a Poisson process, and determine the probability of u delivering a requested content for its 1-hop neighbour v within time t_α as,

$$\begin{aligned} \eta_u^{d,h^1} &= \sum_{v=1}^{h_{v,u}^1} P(N_\alpha^v \geq 1) \cdot P(u \text{ delivering the content to } v) \\ &= \sum_{v=1}^{h_{v,u}^1} (1 - e^{-\lambda_r^v t_\alpha}) \cdot \frac{|\zeta^u - \zeta^v|}{|\zeta^G - \zeta^v|} \\ &\quad \times \frac{|\zeta^u - \zeta^v|}{|\zeta^u - \zeta^v| + \sum_{\substack{j=1 \\ j \neq u}}^{h_{j,v}^1} |\{\zeta^u - \zeta^v\} \cap \zeta^j|}. \end{aligned} \quad (19)$$

Here, ζ^u represents the list of contents owned by node u . Similar to Eq. (8), the first part of (19) determines the probability of node v generating a request. The second part of the equation calculates the probability of u having a matching content as a ratio of contents currently possessed by u but not v and

the contents expected to be requested by v (i.e., contents currently possessed by other group members but not v). Finally, the third part considers the probability that other 1-hop neighbours of v might possess the same content and deliver it upon request.

Similarly, the probability of u delivering a request to another group member q who is not its 1-hop neighbour (i.e., $q \notin \mathbb{H}_u^1$) can be defined as,

$$\begin{aligned} \eta_u^{d,h^m} &= \sum_{q=1}^{h_{q,u}^m} (1 - e^{-\lambda_r^q t_\alpha}) \cdot \rho \cdot \frac{|\zeta^u - \zeta^q|}{|\zeta^G - \zeta^q|} \\ &\times \frac{|\zeta^u - \zeta^q|}{|\zeta^u - \zeta^q| + \sum_{j=1}^{h_{j,q}^m} |\{\zeta^u - \zeta^q\} \cap \zeta^j|}, \\ &[\text{for } m > 1, q \notin \mathbb{H}_u^1, \text{ and } j \notin \mathbb{H}_q^1], \end{aligned} \quad (20)$$

where, ρ represents multi-hop delivery factor which measures the fraction of content requests that are served using multi-hop communication for this group (i.e., $\rho = \rho^m / \rho^t$, where ρ^m and ρ^t shows the number of multi-hop and total deliveries, respectively). Whenever a requester receives a content, it notifies the administrator about the delivery who updates the value of ρ . Finally, η_u^d is calculated by adding η_u^{d,h^1} and η_u^{d,h^m} using (19) and (20).

The probability of receiving a content within the time t_α can be calculated using the similar method discussed above. In this case, u will be a requester and its 1-hop and multi-hop neighbours are content holders. Therefore, the probability of u receiving a content from its one-hop neighbour can be depicted as,

$$\begin{aligned} \eta_u^{g,h^1} &= (1 - e^{-\lambda_r^u t_\alpha}) \cdot \sum_{v=1}^{h_{v,u}^1} \frac{|\zeta^v - \zeta^u|}{|\zeta^G - \zeta^u|} \\ &\times \frac{|\zeta^v - \zeta^u|}{|\zeta^v - \zeta^u| + \sum_{\substack{j=1 \\ j \neq v}}^{h_{j,u}^1} |\{\zeta^v - \zeta^u\} \cap \zeta^j|}. \end{aligned} \quad (21)$$

Similar to Eq. (20), the probability of node u receiving a content from its multi-hop neighbour is depicted as,

$$\begin{aligned} \eta_u^{g,h^m} &= (1 - e^{-\lambda_r^u t_\alpha}) \rho \sum_{q=1}^{h_{q,u}^m} \frac{|\zeta^q - \zeta^u|}{|\zeta^G - \zeta^u|} \\ &\times \frac{|\zeta^q - \zeta^u|}{|\zeta^q - \zeta^u| + \sum_{j=1}^{h_{j,u}^m} |\{\zeta^q - \zeta^u\} \cap \zeta^j|}, \\ &[\text{for } m > 1, q \notin \mathbb{H}_u^1, \text{ and } j \notin \mathbb{H}_u^1]. \end{aligned} \quad (22)$$

Finally, the value of η_u^g can be achieved by adding η_u^{g,h^1} and η_u^{g,h^m} using Eq. (21) and (22). To determine the probability of node u being used as relay during t_α , we need to find the portion of multi-hop deliveries in the group and the probability of u being used in them. The probability of u being selected as relay is,

$$\eta_u^r = (1 - e^{-\lambda_r^G t_\alpha}) \cdot \rho \cdot \frac{\psi^u}{\sum_{j=1}^{h_{v,u}^1} \psi^v}. \quad (23)$$

Here, λ_r^G represents the request generation rate of all group members (i.e., $\lambda_r^G = \sum_{j=1}^G \lambda_r^j$) and ψ^u represents u 's relaying capability. Note that, the relaying capability depends on the forwarding protocol being used. For a random selection, each node will have the same value. In this work, we use the same method proposed in [10] for determining ψ^u . The administrator uses η_u^d , η_u^g and η_u^r in Eq (18) to determine η_u , which is then used in Eq. (17) to calculate the non-contention factor τ^z .

Once the appropriate node z is selected by Eq. (15), the administrator recalculates demand and supply for content α using demand generated from z as *zero*, and the additional supply provided by z using (11). The administrator checks Eq. (14) to determine if additional copies are required and continues the above procedure until (14) is false. After all appropriate nodes are selected, the administrator instructs the supplier nodes to copy α to the respective newly selected content holders. Algorithm 1 outlines the dynamic content distribution method executed by the group administrator. The computational complexity of Algorithm 1 is presented below.

D. Computation Complexity

Let $G1$ be a content sharing group and N_α , N_d , N_s , N_z and N_g be the number of contents, number of demanding, supplier, candidate distributor and total nodes, respectively, in $G1$. Here, $(N_d, N_s, N_z) \leq N_g$. The aggregated computational cost of calculating D_α and S_α in steps 3 and 4 of Algorithm 1 is, $C_{3,4} = O(N_d) + O(N_s)$. Since $(N_d + N_s) \leq N_g$, $C_{3,4} \simeq O(N_g)$. The computational cost of step 5 is, $C_5 = O(1)$. The computational complexity for step 7 to calculate candidate distributor node z is, $C_7 = O(N_z) + O(N_\alpha \times N_g)$, as $O(N_\alpha \times N_g)$ is required to calculate the probability of having no contention τ_z using (17). C_7 can be presented as, $O(N_g) + O(N_\alpha \times N_g)$ as $N_z < N_g$. Again, step 8 has the computational complexity of $C_8 = O(1)$, while $C_{9,10}$ can be written as $O(N_g)$ similar to $C_{3,4}$. Since $|\mathbb{G}_{\alpha'}^i| < N_g$ and τ_z in (17) is calculated once and before the start of Algorithm 1, considering the loop in step 6, the overall computational complexity of steps 6 to 11 can be written as, $C_{6,11} = O(N_g(N_g + 1 + N_g)) = O(2N_g^2 + N_g) \simeq O(N_g^2)$. The computational cost of step 12 is, $C_{12} = O(1)$. Therefore, the complexity for steps 3 to 12 can be calculated as, $C_{3,12} = O(N_g + 1 + N_g^2 + 1) \simeq O(N_g^2)$. Finally, considering the loop in step 2 and the computational complexity for (17), the overall complexity of the algorithm becomes $O(N_\alpha \times N_g^2) + O(N_\alpha \times N_g) \simeq O(N_\alpha \times N_g^2)$.

```

1: procedure DISTRIBUTE CONTENT
2:   for all  $\alpha$  in group content list do
3:     Calculate  $D_\alpha$  using (1);
4:     Calculate  $S_\alpha$  using (11);
5:     Initialize  $\mathbb{N} = \emptyset$ , where  $\mathbb{N}$  is the set of
       nodes selected for distributing  $\alpha$ 
6:     while  $D_\alpha - S_\alpha > 0$  and  $|\mathbb{N}| < |\mathbb{G}_{\alpha'}^i|$  do
7:       Select node  $z$  using (15);
8:       Add  $z$  to  $\mathbb{N}$ , i.e.,  $\mathbb{N} = \{\mathbb{N}\} \cup \{z\}$ ;
9:       Recalculate  $D_\alpha$  considering demand
       from node  $z$  as zero;
10:      Recalculate  $S_\alpha$  considering
        $z$  as a supplier;
11:     end while
12:     Send message to supplier  $y$  to copy  $\alpha$ 
       in all  $z \in \mathbb{N}$  ;
13:   end for
14: end procedure

```

Algorithm 1: Dynamic content distribution

The content sharing group is formed based on the co-located neighbours who have mutual interest in sharing contents. Therefore, it is expected that the number of group members (N_g) will be in the order of tens. On the other hand, the number of contents that nodes are interested to share is expected to be in the order of hundreds. Considering a content sharing group with 10 members and around 50 contents to share, the computational cost is around 5000. Smart mobile devices are currently equipped with fast processor and memory [1], and hence it is expected that these devices now-a-days have enough capability to perform this calculation in real-time to distribute contents dynamically.

V. PERFORMANCE EVALUATION

There is no publicly available dataset that demonstrates the movement of tourists in a tourist spot. Therefore, a synthetic mobility data was used to evaluate the performance of the proposed system. Lakes Entrance, which is a popular tourist spot in Victoria, Australia, was selected as the location. Activities

Fig. 4: Map of Lakes entrance with marked POIs

available within the respective POIs ('Accommodation', 'Bowling', 'Camping', 'Fishing', 'Food and Shopping' and 'Bush-Walking') in that spot were identified using information from Australian tourist portals. Figure 4 shows a map of Lakes Entrance where the POIs are marked with different colours. Tourist arrival into the simulation area and stay duration followed Poisson distribution and negative binomial distribution, respectively as reported in [51]. Usually tourism research organizations (e.g., [52]) collect and provide data on tourists' participation in different activities and in our simulation stay time inside a particular POI was assigned using the average time of that activity reported in [52], [53] with $\pm 20\%$ variation. Movement inside a POI followed the well-established random-waypoint (RWP) mobility model [42] with random pauses and at a speed of 0~5 km/hour being governed by the nature of the activity within that POI. For example, inside the 'Fishing' area, a user randomly selected a location and stayed at that location for the whole session or moved to another random location after some random time, while in 'Bush-walking' a user started from a particular spot and continued along the trail with some random pauses in-between. In contrast, in 'Food and shopping' POI the movement of nodes followed RWP mobility model with relatively low speed and long pause to represent the movement of tourists around shops and inside restaurants.

For sharing purpose, we considered 320 contents across eight categories in the network out of which 50 contents were related to five events which were scheduled at different times (two were scheduled before the start of the simulation, two during the simulation while one after the simulation). Contents were of the following media types: text (40%, 512 KB), photo (20%, 1MB), audio (20%, 5MB [54]) and video (20%, 6.5MB [55]), where the first number within the parenthesis shows the percentage of that type of content in the pool and the second number shows mean content size. Abhari *et al.* [56] suggests that the file size in a popular content sharing network (i.e., YouTube) follows Gamma distribution, hence the size of individual contents was determined using that distribution with shape parameter ($=2.0$) as in [57]. Initially, nodes were assigned with random number of contents (mean size 27.52) considering their interest as in [54].

Nodes generated content requests based on their interest following a Zipf distribution [57] and the request generation time followed an exponential distribution as in [28]. We used NS3 as the simulation platform where nodes were configured with IEEE 802.11g standard in ad-hoc mode at 54 Mbps data rate and 150m as their communication range. NS3 provides built-in module for MAC layer and physical layer implementation of IEEE 802.11 standards. A contention based DCF scheme with RTS/CTS and 7-retry was used in the MAC layer in case of a failed transmission. For modelling wireless channel, constant speed propagation delay model and range propagation loss model was used. At the beginning of the simulation, random battery power was assigned to each node where 100 units represented the full capacity. We considered periodic battery drain (0.07 unit/minute) and drain due to data transfers (8×10^{-8} unit/byte). The simulation was run for 3 days with 150 nodes.

A. Simulation results

The performance of the proposed DCDS approach is analyzed using three metrics, namely, (i) hit rate, (ii) delivery success rate and (iii) average delivery latency. Hit rate represents the percentage of the requests for which contents are successfully located by the requester and delivery success rate depicts the percentage of successful deliveries for the same. The average delivery latency shows the average time it takes to obtain a requested content. These metrics are selected because hit rate indicates the impact of content distribution, while success rate relates to the extent of communication coverage and stay probability of content holders, and finally latency denotes a requesting node's reachability from the content holder. Additionally, we also investigated control overhead, administrator lifetime, energy consumption by nodes, evolution of demand and supply for a content and the distribution of replicated contents to meet dynamic demand in DCDS.

The proposed DCDS approach is compared against the method presented in [9] (termed as DCST afterwards). Although DCDS and DCST use the same underlying group formation, administrator selection and message forwarding technique, the latter does not employ any content distribution scheme; therefore, a comparison between them will reveal the benefit of the proposed content distribution scheme. The interval between consecutive distribution cycle was set to $\theta=30$ mins in our simulation. DCDS is also compared with NCL [28] which uses a set of central locations in a network that are selected based on their accessibility by other nodes for holding the replicated contents as discussed in Section II. Note that the term central location only reflects easy accessibility by other nodes and does not indicate a centralized replication method; rather multiple such locations are selected for placing the replicated contents. The DCDS approach is also compared with SCR [17]. Since SCR is more suitable for *work-place* type scenarios with predictable regular movement pattern and existing social relationship information, the characteristics of a tourist spot were applied to SCR for fair comparison. SCR uses regional mobility prediction to determine the most

Fig. 5: (a) Hit rate and (b) success rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr;

likely region a node is expected to visit and its social influence among the users for spreading a content. In this approach, a coordinate server calculates the gain of placing a set of replicated contents in a node, and only selects those contents that maximizes the gain. In absence of a permanent coordinate server in our considered scenario, we randomly selected an administrator for this task among nearby nodes. The *t*-test comparing DCDS with other methods in terms of hit rate, success rate and delivery latency at various arrival and request rates yielded *p*-values, $p \leq 6.33 \times 10^{-6}$, $p \leq 5.89 \times 10^{-6}$ and $p \leq 7.25 \times 10^{-6}$ at 99% confidence level, respectively and suggests their performance differences are statistically significant. Each simulation was run 20 times and the averaged results are presented below.

The arrival rate of the nodes was varied from 10-30 nodes/hr to understand its impact on hit and delivery success rates, and the outcomes are presented in Fig. 5. Both rates initially start increasing along with an increasing node arrival rate across all approaches since a higher node arrival rate implies that more nodes were available inside the area for sharing contents. However, when the arrival rate exceeds certain limit (i.e. 25 nodes/hr in our current setting) both rates decrease slightly across all approaches, because more nodes subsequently generate more requests causing congestion. The proposed approach always achieves the highest hit and delivery success rates, achieving a hit rate of 73.31% and delivery success rate of 70.45% for an arrival rate of 25 nodes/hr. In comparison, DCST approach achieves 66.76% hit rate and 63.95% success rate, while SCR (50.06% and 48.43%) and NCL (48.39% and 37.33%) achieve much lower hit and success rates for the same arrival rate. The proposed DCDS approach proactively distributes contents based

Fig. 6: (a) Hit rate and (b) success rate for different mean request rates (number of requests/hr) keeping mean node arrival rate fixed at 25 nodes/hr

on the estimated future demand which essentially increases availability of contents near the positions from where future requests are expected. When requests are ultimately generated for those contents, they can be served from those positions (i.e., by nodes with replicated content), and hence DCDS approach achieves 7%~34% improvement for hit and success rates over other approaches. One interesting observation from Fig. 5 is the performance of NCL and SCR approaches. Although NCL achieves similar hit rate as SCR, its delivery success rate is much lower. The reason being, in NCL contents are replicated in easily accessible locations (i.e., central locations) which essentially help in locating the contents (hence higher hit rate), but at the same time it also makes the central locations more congested and unable to deliver as many nodes simultaneously try to communicate with them for uploading and downloading contents.

The impact of content request rate is presented in Fig. 6. The figures show that both hit and success rates start to increase at the beginning with an increasing request rate, however, after exceeding a particular request rate (5 requests/hr in our current setting) both metrics start to decline slightly. Initially, with an increasing request rate, nodes located within the same neighbourhood generate more requests for the same content which has already been requested and obtained by a nearby node. However, both rates start declining as more requests generate more congestion in the network. The proposed DCDS approach achieves 71.43% hit rate and 68.94% success rate at the request rate of 6 req/hr. In contrast, other approaches achieve much lower hit (43%~64%) and success rate (35%~62%) for the same request rate. In literature, the reported hit and success rates achieved by decentralized content sharing in *work-place* type scenarios ranges from

Fig. 7: Average delivery latency for successful deliveries for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

75%~80% [21], [58]. From that perspective, the DCDS approach obtains acceptable results in tourist spots, where movement patterns are unpredictable and social ties are mostly unavailable.

The average latency for successful content delivery is shown in Fig. 7. Only for this particular experiment, the total number of requests generated for DCDS was limited to 10000. To be fair with other methods the same set of 10000 requests were made while experimenting with DCST, SCR and NCL. In Fig. 7, the latency values for successful content deliveries are sorted in ascending order and averaged over particular number of deliveries for convenience of presentation. All the approaches can complete up to 3000 successful deliveries within a short time (e.g., 0.01 mins (DCDS), 0.03 mins (DCST), 0.03 mins (SCR) and 0.02 mins (NCL)). Since NCL only uses central locations for content delivery, it experiences higher delay after 3×10^3 requests and can only deliver up to 3.8×10^3 requests for the given settings with an average delay of 45.17 mins beyond 3×10^3 requests. SCR can deliver up to 4.8×10^3 requests while incurring higher delays (60.47 mins) beyond 3.8×10^3 requests. In contrast, our DCDS approach can deliver up to 6.3×10^3 requests with significantly lower delay (DCDS 11.92 mins vs. DCST 55.38 mins). The reason for such lower delay for DCDS is the content distribution policy which proactively replicates the contents near the potential requesters, which essentially enable requesters to obtain requested contents within a short time. For the given settings, DCST can deliver up to 6.3×10^3 requests, while DCDS can deliver up to 7×10^3 requests. However, beyond 6.3×10^3 requests its delivery latency grows (40.48 mins) as sometimes content holders might be multiple hops away from requesters and forwarding requests as well as obtaining

Fig. 8: Delivery success rate for event-based content requests across different mean (a) node arrival rate and (b) content request rate

matching contents require longer time, hence a longer delay is observed. Note that though NCL approach uses multiple locations for placing the replicated contents, it does not consider contention factor for placing the replicated contents and hence suffers due to congestion in those central locations which results in longer delay and lower delivery success rate. Although the delay is longer for a small portion of contents (i.e., 10%), DCDS approach is capable of delivering higher number of contents (7%~32% higher) even if the delay is longer for some of them, where other approaches fail altogether.

The DCDS approach considers the impact of influencing events on content request while calculating future demand. Impact of such policies is presented in Fig. 8 (a) and (b). The figures suggest that DCDS achieves the highest percentage of successful deliveries for event-based content requests across all the cases for different node arrival and average request rates. The increment achieved by DCDS ranges from 10%~39%.

Administrators perform a number of group management and content sharing and distribution tasks that drain their energy. Therefore, administrator lifetime is an important issue since frequent administrator handover is not desirable. Figure 9 (a) shows that DCDS achieves much higher administrator lifetime than NCL and SCR, while achieving slightly lower lifetime than DCST. A similar trend is also found in Fig. 9 (b), which suggests that the DCST and DCDS achieved lower administrator expiry rate compared to SCR and NCL. Since stay probability and remaining energy are used as the administrator selection criteria in both DCDS and DCST, they result in higher administrator lifetime and lower administrator expiry rate due

Fig. 9: (a) Average lifetime of administrator nodes and (b) Administrator expiry rate for different request rates keeping mean node arrival rate fixed at 25 nodes/hr

to low battery. However, in DCDS, an administrator periodically calculates demand, supply and distribution, and sometimes replicates a content to meet future demand, and hence achieve slightly lower lifetime and higher administrator expiry rate than DCST. In contrast, in NCL stay probability and remaining energy was not considered at all and administrators were mostly busy in receiving/delivering contents from other nodes which in turn resulted in such lower lifetime.

Figure 10 (a) shows the percentage of message overhead which was calculated as the ratio of the size of control messages to the size of the successfully delivered contents. DCDS generated lower message overhead than DCST and a slightly higher overhead than SCR and NCL. DCDS uses some group related information (e.g., group content list and stay probability of members), a neighbourhood table and some content and event specific information (e.g., content popularity and event scale) for proactive content distribution. Therefore, the message overhead is slightly higher than SCR and NCL at a lower request rate (2 requests/hr), the difference being $< 1\%$. However, as the request rate increases, the number of successfully delivered contents increases which essentially starts reducing the overall message overhead, and the proposed approach produces nearly similar overhead as produced by other approaches.

The impact of varying the interval between successive distribution cycle (θ) is presented in Fig. 10 (b). The figure shows that initially hit and success rates slightly increase with the increasing value of θ , but both start to decrease after reaching a plateau. The reason is that with a lower value of θ , the administrator performs content distribution with partial information about the group and contents, as it takes some time

Fig. 10: (a) Control message overhead for different request rates keeping mean node arrival rate fixed at 25 nodes/hr (b) Hit and success rates for different value of θ for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

to receive updates from all group members. On the other hand, a higher value of θ indicates that the administrator reacted late in responding to dynamic demand and supply.

As discussed before, the administrators periodically calculate dynamic demand and supply for a content to determine if additional distribution is needed. Figure 11 shows varying demand and supply calculated for a particular content by an administrator inside the ‘Food and shopping’ POI. The figure shows that at the beginning (during cycle 1), supply for this content was lower than the expected demand, hence proactive distribution (2 copies) was performed to meet future demand. During cycle 2 and 3, the expected demand was lower than the available supply and no distribution was required. Afterwards, the amount of expected demand surpassed the available supply in cycle 4 and 5 as new nodes joined the group and were expected to generate demand while few existing suppliers left the group which reduced available supply in cycle 4. Therefore, distribution was done during those cycles. As more content holders joined the group, the amount of available supply increased during cycle 6 and 7 and the demand remained lower than the supply up to cycle 10. Finally, expected demand surpassed available supply during cycle 11 and 14, and additional distribution was needed during those cycles. In DCDS, the remaining energy of a node was assessed for the selection of a node for content distribution purpose. Figure 12 demonstrates the impact of such a selection process by showing that although Node#7 and Node#10 were selected four times for distribution purposes, they did not run out of battery before their next charging cycles. Note that in our scheme, whenever any two

Fig. 11: Demand and supply for a content in a group inside the 'Food and shopping' POI for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

Fig. 12: Remaining energy of two nodes before their next charging cycles (a circle represents an occurrence of replication at that node)

Fig. 13: A snapshot of the positions of the demanding, supplier and distributor nodes for a popular content in different POIs. A supplier node is one which already holds the content and a distributor is one where a replicated copy is placed

nodes have contact opportunity, they first exchange the contents requested by themselves. Afterwards, they exchange contents as a relay node. Finally, the replicated contents are placed at a node, if time permits. The reason both nodes' energy was degrading fast is that both received their own requested contents, delivered some requested contents and copied replicated contents at each contact. However, Node#7 was degrading faster compared to Node#10 as the former exchanged/copied more contents with exchanging nodes at the contacts than the other node.

Figure 13 shows a snapshot of the locations of demanding, supplier and distributor nodes for a popular content in simulation. Since it was a popular content, more demanding nodes (marked by red square) were present in the network. The number of distributor nodes (marked by blue plus sign) was also higher to meet expected high demand. The figure also shows that the distributor nodes were positioned near the demanding nodes. In some cases, a few distributor nodes were enough to meet expected demand ('Camping' POI on the left side of Fig. 13), while others required more distributor nodes ('Bowling' POI in the middle of Fig. 13). Nodes generated varying demand and supply for a content in different POIs and hence the number of distributor nodes also varied in different regions.

The inherent nature of the activities available in different POIs governs the movement of nodes inside that POI. Consequently, the performance of the proposed approach varies for different POI related activities. Figure 14 shows that the hit and delivery success rates were highest inside the 'Bowling' POI while 'Fishing',

Fig. 14: Hit and success rates in different POIs for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

Fig. 15: Average delivery latency in different POIs for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

‘Camping’ and ‘Accommodation’ attained comparatively low hit and success rates. ‘Bowling’ is an indoor activity, and hence the movement of nodes was restricted within the indoor area which produced frequent encounters among nodes and generated such high hit and delivery success rates. In contrast, nodes were more static inside the ‘Accommodation’, ‘Fishing’ and ‘Camping’ regions as the nature of those activities dictates users to remain stationary for a longer period of time, and hence had limited or no contacts with other nodes during that period, resulting in comparatively low hit and success rates. For example, in ‘Accommodation’ area nodes mostly entered to take rest and hence had low mobility while in ‘Fishing’ nodes remained stationary at a particular point for a longer time before moving to another one.

Fig. 15 shows the average delivery latency in different POIs. It shows that in ‘Bowling’ the latency was the lowest as nodes frequently came into contact with each other. The latency is also lower in ‘Food and shopping’ POI as the movement of nodes inside this area had relatively low speed and long pause to represent the movement of tourists around shops and inside restaurants which gave enough opportunity for nodes to encounter and share contents. In contrast, the latency is comparatively high in ‘Accommodation’, ‘Camping’ and ‘Fishing’ regions due to infrequent contacts among nodes. Note that for 90% of the successful content deliveries, the latency is much low and similar to the first part of Fig. 7 (up to 6.3×10^3 successful delivery) and as highlighted in the related discussion of that figure. This figure is to present the variation of latency in different POIs.

The impact of node mobility is presented in Figs. 16 (a) and (b). For better representation, results obtained from ‘Walking’ and ‘Food and shopping’ only are highlighted in these figures. To better understand the impact of movement speed on the sharing process, only for this experiment, we randomly assigned different movement speed (i.e., 0-2 km/hr, 2-4 km/hr and 4-6 km/hr) to participating nodes which was used for movements within the POI. The figures suggest that delivery success rate decreases with an increasing movement speed while the average delivery latency increases for the same. With a higher movement speed, it is more likely that nodes go beyond the communication range of their neighbours, resulting in higher latency and lower success rate.

To determine the impact of frequent movement of tourists from one POI to another, we performed experiments by varying the stay time in various POIs, as in our simulation setup, a shorter stay time in POIs causes higher inter-POI movements. The results are shown in Figs. 17 (a) and (b). For example, average stay time in ‘Walking’ and ‘Food and shopping’ was considered as 1 hour and 2 hours, respectively, as the walking trail in Lakes entrance is 5 km long and takes on average 1 hour to complete [53]. These values were used in all preceding experiments and is marked as standard in this figure. We varied this standard stay time by up to $\pm 50\%$ for this experiment. The results show that a lower stay time (i.e., higher inter-POI movement) caused lower hit and success rates, and higher latency. The reason is that although nodes were

Fig. 16: (a) Delivery success rates and (b) average delivery latency in 'Walking' and 'Food and shopping' POI for different movement speed of nodes

Fig. 17: (a) Hit and success rates and (b) average delivery latency for variation of stay time in different POIs

able to locate the content, in several cases, they were unable to successfully receive them. Since nodes were co-located for a shorter duration, the delivery latency also increased. On the other hand, increasing the average stay time increased the performance of DCDS as nodes had more opportunity to communicate. Overall, even in a highly dynamic spot, the proposed approach achieved acceptable service considering the dynamic node movement and limited communication opportunities.

VI. CONCLUSION

This paper proposes a dynamic content distribution scheme for decentralized content sharing in tourist spots employing the concept of economic theory of supply and demand in a marketplace. Whenever the expected demand surpasses the available supply, the proposed approach proactively replicates content at strategic locations to improve delivery service. We estimated future demand considering user interest, content popularity and the impact of influencing events on content demand. Available supply for contents is calculated considering content holder's coverage, stay probability and available energy. Finally, the locations for placing additional copies are selected based on the optimization of demand coverage and medium access contention. Results obtained from extensive simulation using a popular tourist spot in Australia indicate that, the proposed approach is capable of handling significantly higher number of requests while attaining better delivery performance than competing DCS methods applied in tourist hotspots. Future research will extend the current work by incorporating a buffer management scheme whereby a node with limited buffer but better positioned for replication will use an optimization technique to maximize benefits for individual as well as for the whole group. Conducting a real-life experiment by distributing a content sharing app among tourists in a visiting hotspot also remain a part of our future work.

REFERENCES

- [1] T. Strayer, S. Nelson, A. Caro, J. Khoury, B. Tedesco, O. DeRosa, C. Clark, K. Sadeghi, M. Matthews, J. Kurzer *et al.*, "Content sharing with mobility in an infrastructure-less environment," *Elsevier Computer Networks*, vol. 144, pp. 1–16, 2018.
- [2] "Mobile black spot program," <https://www.communications.gov.au/publications/mobile-black-spot-programme-guidelines-version-11>, accessed 25 November 2018.
- [3] "CISCO forecast," <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>, accessed 25 November 2018.
- [4] "Increase in mobile data usage," <https://crowdsupport.telstra.com.au/t5/Data-Usage/Increase-in-Mobile-Data-usage-with-Smartphones-and-4G-handsets/td-p/127496>, accessed 25 November 2018.
- [5] "Places without Internet," http://www.huffingtonpost.com.au/entry/pretty-places-without-internet-access_n_3961920, accessed 25 November 2018.
- [6] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "A first look at cellular network performance during crowded events," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, 2013, pp. 17–28.
- [7] J. Erman and K. K. Ramakrishnan, "Understanding the super-sized traffic of the super bowl," in *ACM IMC*, 2013, pp. 353–360.

- [8] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Content sharing among visitors with irregular movement patterns in visiting hotspots," in *IEEE NCA 2015*, pp. 230–234.
- [9] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Decentralized content sharing among tourists in visiting hotspots," *Elsevier Journal of Network and Computer Applications*, vol. 79, pp. 25 – 40, 2017.
- [10] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Carry me if you can: A utility based forwarding scheme for content sharing in tourist destinations," in *IEEE APCC 2016*, pp. 261–267.
- [11] B. Balcik and B. M. Beamon, "Facility location in humanitarian relief," *International Journal of Logistics Research and Applications*, vol. 11, no. 2, pp. 101–121, 2008.
- [12] P. Murali, F. Ordóñez, and M. M. Dessouky, "Facility location under demand uncertainty: Response to a large-scale bio-terror attack," *Elsevier Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 78–87, 2012.
- [13] S. Shakkottai and R. Johari, "Demand-aware content distribution on the internet," *IEEE Networking*, vol. 18, no. 2, pp. 476–489, 2010.
- [14] A. Alasaad, S. Gopalakrishnan, and V. C. Leung, "A hybrid approach for cost-effective media streaming based on prediction of demand in community networks," *SpringerTelecommunication Systems*, vol. 59, no. 3, pp. 329–343, 2015.
- [15] X. Meng, Y. Wang, and Y. Gong, "Perspective of space and time based replica population organizing strategy in unstructured peer-to-peer networks," *Elsevier Journal of Network and Computer Applications*, vol. 49, pp. 1–14, 2015.
- [16] K. Chen and H. Shen, "Maximizing p2p file access availability in mobile ad hoc networks through replication for efficient file sharing," *IEEE Computers*, vol. 64, no. 4, pp. 1029–1042, 2015.
- [17] Z. Wang, L. Sun, M. Zhang, H. Pang, E. Tian, and W. Zhu, "Propagation-and mobility-aware d2d social content replication," *IEEE Mobile Computing*, vol. 16, no. 4, pp. 1107–1120, 2017.
- [18] C. Boldrini, M. Conti, and A. Passarella, "Contentplace: social-aware data dissemination in opp. networks," in *ACM MSWiM*, 2008, pp. 203–210.
- [19] T. Le, Y. Lu, and M. Gerla, "Social caching and content retrieval in disruption tolerant networks (dtms)," in *IEEE ICNC*, 2015, pp. 905–910.
- [20] G. Moualla, P. A. Frangoudis, Y. Hadjadj-Aoul, and S. Ait-Chellouche, "A bloom-filter-based socially aware scheme for content replication in mobile ad hoc networks," in *IEEE CCNC*, 2016, pp. 359–365.
- [21] K. Chen, S. Member, H. Shen, S. Member, and H. Zhang, "Leveraging social networks for p2p content-based file sharing in disconnected manets," *IEEE Mobile Computing*, vol. 13, no. 2, pp. 235–249, 2014.
- [22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *ACM SIGCOMM 2005*, pp. 252–259.
- [23] I.-W. Ting and Y.-K. Chang, "Improved group-based cooperative caching scheme for mobile ad hoc networks," *Elsevier Parallel and Distributed Computing*, vol. 73, no. 5, pp. 595–607, 2013.
- [24] R. Cabaniss and S. Madria, "Content distribution in delay-tolerant networks using social context," in *IEEE WMNC*, 2014, pp. 1–8.
- [25] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 1, pp. 65–67, 2008.
- [26] Y. Ma, R. Kibria, and A. Jamalipour, "Cache-based content delivery in opportunistic mobile adhoc networks," in *IEEE GLOBECOM*, 2008, pp. 1–5.
- [27] J. Wu, Z. Chen, and M. Zhao, "Secm: status estimation and cache management algorithm in opportunistic networks," *The Journal of Supercomputing*, pp. 1–19, 2018.
- [28] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, "Cooperative caching for efficient data access in disruption tolerant networks," *IEEE Mobile Computing*, vol. 13, no. 3, pp. 611–625, 2014.

- [29] Z. Ma, H. Wang, K. Shi, and X. Wang, "Learning automata based caching for efficient data access in delay tolerant networks," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–19, 2018.
- [30] M. Radenkovic, V. S. H. Huynh, and P. Manzoni, "Adaptive real-time predictive collaborative content discovery and retrieval in mobile disconnection prone networks," *IEEE Access*, 2018.
- [31] S. Naz, R. N. B. Rais, P. A. Shah, S. Yasmin, A. Qayyum, S. Rho, and Y. Nam, "A dynamic caching strategy for ccn-based manets," *Elsevier Computer Networks*, 2018.
- [32] S. Ioannidis, E. Yeh, E. Yeh, and S. Ioannidis, "Adaptive caching networks with optimality guarantees," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 737–750, 2018.
- [33] J. Reich and A. Chaintreau, "The age of impatience: optimal replication schemes for opportunistic networks," in *ACM CoNEXT*, 2009, pp. 85–96.
- [34] K. Machado, A. Boukerche, E. Cerqueira, and A. A. Loureiro, "A socially-aware in-network caching framework for the next generation of wireless networks," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 38–43, 2017.
- [35] X. Li, X. Wang, P.-J. Wan, Z. Han, and V. C. Leung, "Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design," *IEEE Journal on Selected Areas in Communications*, 2018.
- [36] H. Qu, G. Ren, J. Zhao, Y. Shi, and Z. Tan, "Performance analysis of the content dissemination mechanism with proactive content fetching and full-duplex d2d communication: an evolutionary perspective," *Springer Mobile Networks and Applications*, pp. 1–24, 2018.
- [37] C. Yi, S. Huang, and J. Cai, "An incentive mechanism integrating joint power, channel and link management for social-aware d2d content sharing and proactive caching," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 789–802, 2018.
- [38] C. Xu, M. Wang, X. Chen, L. Zhong, and A. L. Grieco, "Optimal information centric caching in 5g device-to-device communications," *IEEE Transactions on Mobile Computing*, 2018.
- [39] K. Thilakarathna, A. C. Viana, A. Seneviratne, and H. Petander, "Design and analysis of an efficient friend-to-friend content dissemination system," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 702–715, 2017.
- [40] S. Sun, M. Liu, Z. Jiao, X. Pang, and S. Chen, "User-centric content sharing via cache-enabled device-to-device communication," *Elsevier Journal of Network and Computer Applications*, vol. 115, pp. 103–115, 2018.
- [41] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Dynamic content distribution for decentralized sharing in tourist spots using demand and supply," in *IEEE IWCMC 2017*, pp. 2121–2126.
- [42] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transaction on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.
- [43] J. Liu, P. Dolan, and E. R. Pedersen, "Personalized news recommendation based on click behavior," in *IEEE IUI*, 2010, pp. 31–40.
- [44] H. Schütze and C. Silverstein, "Projections for efficient document clustering," in *ACM SIGIR*, vol. 31, no. SI, 1997, pp. 74–81.
- [45] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Taylor & Francis Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.
- [46] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *ACM WSDM*, 2010, pp. 291–300.
- [47] H. Becker, D. Iter, M. Naaman, and L. Gravano, "Identifying content for planned events across social media sites," in *ACM WSDM*, 2012, pp. 533–542.
- [48] D. Getz, "Event tourism: Definition, evolution, and research," *Elsevier Tourism management*, vol. 29, no. 3, pp. 403–428, 2008.
- [49] L. K. Jago, "Special events and tourism behaviour: a conceptualisation and an empirical analysis from a values perspective," *Ph.D. dissertation, Department of Hospitality, Tourism and Marketing, Victoria University*, 1997.

- [50] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *IEEE INFOCOM*, vol. 3, 2013, pp. 1360–1369.
- [51] M. Kruger and M. Saayman, "The determinants of visitor length of stay at the kruger national park," *koedoe*, vol. 56, no. 2, pp. 1–11, 2014.
- [52] "Tourism research Australia," <http://www.tra.gov.au/research/regional.html>, accessed 27 November 2018.
- [53] "Lakes Entrance - Walk to The Entrance," <http://www.melbourneplaygrounds.com.au/melbourneplaygrounds-info.php?id=53961>, accessed 27 November 2018.
- [54] L. McNamara, C. Mascolo, and L. Capra, "Media sharing based on colocation prediction in urban transport," in *ACM MobiCom 2008*, pp. 58–69.
- [55] X. Che, B. Ip, and L. Lin, "A survey of current youtube video characteristics," *IEEE Transaction on MultiMedia*, vol. 22, no. 2, pp. 56–63, 2015.
- [56] A. Abhari and M. Soraya, "Workload generation for youtube," *Springer Multimedia Tools and Applications*, vol. 46, no. 1, pp. 91–118, 2010.
- [57] K. Thilakarathna, H. Petander, J. Mestre, and A. Seneviratne, "Mobitribe: Cost efficient distributed user generated content sharing on smartphones," *IEEE Transaction on Mobile Computing*, vol. 13, no. 9, pp. 1–14, 2013.
- [58] A. Krifa, C. Barakat, and T. Spyropoulos, "Mobitrade: trading content in disruption tolerant networks," *ACM Challenged networks*, pp. 31–36, 2011.