Decentralized Content Sharing Among Tourists in Visiting Hotspots

Shahriar kaisar[1], Joarder Kamruzzaman[2,1], Gour Karmakar[2], and Iqbal Gondal[2,1]

Abstract

Content sharing with smart mobile devices using decentralized approach enables users to share contents without the use of any fixed infrastructure, and thereby offers a free-of-cost platform that does not add to Internet traffic which, in its current state, is approaching bottleneck in its capacity. Most of the existing decentralized approaches in the literature consider spatio-temporal regularity in human movement patterns and pre-existing social relationship for the sharing scheme to work. However, such predictable movement patterns and social relationship information are not available in places like tourist spots where people visit only for a short period of time and usually meet strangers. No works exist in literature that deals with content sharing in such environment. In this work, we propose a content sharing approach for such environments. The group formation mechanism is based on users' interest score and stay probability in the individual region of interest (ROI) as well as on the availability and delivery probabilities of contents in the group. The administrator of each group is selected by taking into account its probability of stay in the ROI, connectivity with other nodes, its trustworthiness and computing and energy resources to serve the group. We have also adopted an incentive mechanism as encouragement that awards nodes for sharing and forwarding contents. We have used network simulator NS3 to perform extensive simulation on a popular tourist spot in Australia which facilitates a number of activities. The proposed approach shows promising results in sharing contents among tourists, measured in terms of content hit, delivery success rate and latency.

1 Introduction

Mass adoption of smart mobile devices is one of the most prominent features observed in the recent years. The number of smartphone users worldwide is expected to surpass 2 billion by 2016 and over one-third of the world's population by 2018 [1]. The key features behind smartphones' popularity include attractive look, easy-to-use interface, advanced configuration, seamless network connectivity and a range of useful applications. Content sharing applications enable smart device users to generate and share various types of contents such as music, videos, pictures and news items with anyone in the world. Such engagements are motivated by the ease to share events of life and express opinions within seconds. People

- *1. Faculty of Information Technology, Monash University, Australia. E-mail: shahriar.kaisar@monash.edu*
- *2. School of Engineering and Technology, Federation University Australia, Victoria, Australia. E-mail:{joarder.kamruzzaman, gour.karmakar, Iqbal.gondal}@federation.edu.au*

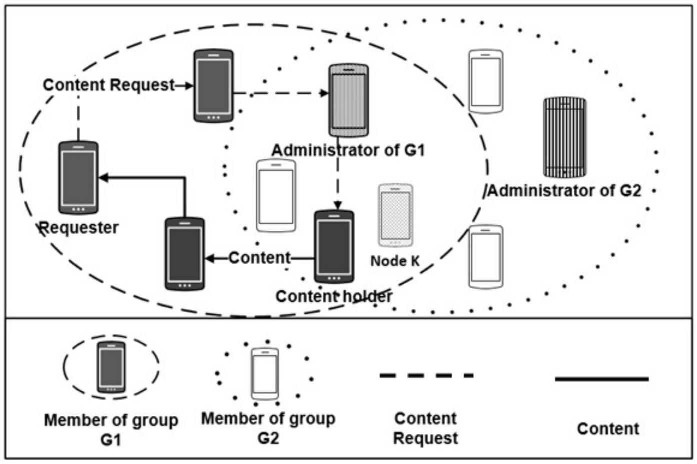


Fig. 1: Group-based decentralized content sharing

are using smartphones and content sharing applications on a regular basis to obtain information and interact with other people, and as a medium of entertainment.

In literature, content sharing is proposed using three different architectures, namely, (i) decentralized, (ii) centralized and (iii) hybrid architecture. In case of a decentralized architecture [2], nodes form an ad-hoc network using either Bluetooth or Wi-Fi technology and share contents through peer-to-peer connections without requiring Internet connectivity. In a centralized architecture [3], contents are stored at a central server which delivers them to nodes (mobile devices) on requests using Internet connection through cellular or Wi-Fi network. The hybrid architecture [4] also uses central servers to store contents, but divides a single content into multiple portions if multiple users request the same content and each portion is disseminated to one of the requesters through the Internet. Later the requesters share the portions among themselves using Bluetooth or Wi-Fi and finally merge them to get the full content. The centralized and hybrid approaches provide effective solutions by delivering contents in a timely manner as they utilize Internet connections. One problem associated with these approaches is the constant requirement of Internet connection which is not available everywhere. For example, there are approximately 6000 locations with inadequate mobile coverage across metropolitan and regional areas of Australia [5] and similar situation exists in many parts of the world. Another problem is the Internet traffic which is growing alarmingly and approaching a bottleneck. Reports suggest that bandwidth demand is increasing faster than our capacity to deliver it and current infrastructure is insufficient to meet future demand [6]. Internet connection via cellular network is also costly. Therefore, decentralized approach is regarded as a suitable alternative to deliver contents.

Decentralized approach is scalable and fault tolerant but most challenging as there is no central server to store and deliver contents or user information. In this approach, mobile nodes usually form groups (i.e., communities) based on certain criteria such as content interest and proximity, and share contents using Bluetooth or Wi-Fi. Figure 1 shows the schematic diagram of groups in decentralized approach, where users have created groups *G1* and *G2* that are physically overlapped as users with similar interest may be physically apart. Each group has an administrator to perform tasks related to group management and directory service. For example, an administrator can store the list of contents of the group members and direct content requests to the content holder [2]. Some approaches do not use any grouping such as the works proposed in [7]. However, content sharing without grouping makes it difficult to locate a particular content since no one knows who has which content. On the other hand, sharing content list with everyone (i.e., not limiting within the defined group) produces more traffic. Different group formation methods have been proposed in the literature [2], [8], [9], [10], [11], [12]. Works in [8], [9] consider that similar minded people tend to meet around the same location and form groups around those locations. Social relationship based groups are considered in [10], [11], [12] and it is argued that individuals tend to meet similar set of people on a regular basis and should form a group with them [10]. SPOON [2] considers that people with similar interest meet more often and should form groups based on the interest similarity and meeting frequency. Similarity in regular mobility pattern [13], meeting frequency [8] and social relationship [14] has also been exploited for message forwarding.

The above mentioned decentralized approaches take advantage of the temporal and spatial regularity in human movement patterns along with their social relationships. Some approaches collect social relationship information from existing social network while others build it over time based on meeting time and frequency. However, there are occasions where people go to some places which are out of their regular life and meet people with whom they do not have any existing social relationship. For example, someone visiting a tourist spot or camping site, or meeting in a festival. There has been very little focus for facilitating content sharing in such scenarios. Content sharing in such irregular meeting places is much more challenging, because in regular meeting places people usually maintain a regular spatio-temporal pattern for movement which is absent here. Group formation and management becomes more difficult due to the unavailability of such information, and added dynamic features, e.g., irregular movement, changing node distribution and request patterns further complicate content sharing task. Many of those places may lack Internet connectivity altogether and hence content sharing in such places is beneficial as contents can be used for information sharing and entertainment purposes. The developed approaches can also be utilized in other scenarios for rapid information distribution during natural disaster, collaborative sensing, opportunistic computing and proximity marketing. Our preliminary works [15], [16] propose the fundamental concepts for group management and administrator selection for content sharing in tourist attraction. However, they lack proper consideration of some practical aspects (e.g., content availability and delivery probabilities, participation incentive) for group formation and require further improvement.

In this paper, we propose interest matching and content deliverability based grouping for content sharing among tourists in visiting hotspots, more generally among people in irregular meeting places. We consider that users are interested about performing different activities while visiting a tourist spot and have different levels of interest for participating in those activities. An interest score represents the level of interest of a user to perform activities or consume contents (e.g., music, videos, news) in a tourist spot. We obtain such interest score from a user's personal profile and location centric experiences, and online recommendations from friends. Nodes form groups considering interest fulfilment probability, content availability and delivery probability, and expected delay in delivery. Such information is obtained from group administrators and one-hop neighbours. The group administrators are selected based on their stay probability in the current point-of-interest (POI) and connectivity with other nodes along with their trustworthiness, remaining energy and device configuration.

The objective of our work is to provide a solution for content sharing in irregular meeting places which is not well studied in the literature. In that regard our work makes the following contributions:

- We propose techniques for content sharing in places where predictable regular movement pattern and social relationship information are not available or difficult to establish since the stay duration is usually not long enough. The techniques include group formation and its administrator selection process, and an utility based message forwarding technique.
- To make the scheme practically acceptable by users, it includes a mechanism that offers incentive for participation in sharing contents as well as a trust model to reduce security risks.
- We use network simulator NS3 to evaluate our scheme and perform extensive simulation considering a popular tourist spot in Victoria, Australia that run many tourist activities. Simulation results suggest that our approach attains acceptable hit and success rate and low delivery latency. The proposed administrator selection policy also achieves higher average lifetime for administrators.

The rest of the paper is arranged as follows: Section 2 provides a brief description of related works. Section 3 presents different aspects of the proposed approach and Section 4 evaluates the performance of the proposed approach. Finally, Section 5 presents some concluding remarks.

2 Related Work

For the reasons illustrated in the previous section, the focus of this work is the decentralized content sharing architecture, and hence the discussion in this section is limited to this approach. Decentralized content sharing can be classified into publish-subscribe and on-demand schemes. In the publish-subscribe schemes [11], [17] the publisher nodes (content generators) publish contents such as movies, music and news items on some channels and the other nodes (subscribers) subscribe to them. The publisher nodes periodically publish some contents and disseminate those over the network. Whenever a subscriber node meets a publisher, it first copies contents relating to personal interest and then copies contents that might be of interest to future neighbours. Such schemes perform well when users are looking for contents from particular channels. However, they do not address on-demand content delivery, where users are looking for a particular content instead of contents designated to a particular channel. Visiting hotspots, which is the focus of our work here, necessitates on-demand content sharing.

Grouping is the most important aspects of decentralized content sharing since it provides a better way of managing the user preferences and helps providing content on-demand. Some of the proposed methods have considered the temporal and spatial regularity to detect groups while others have focused on social properties for group construction.

Flocks [18] is a dynamic group creation mechanism which employs user profile and their location. For example, players within a certain area can form a group to communicate messages. MobilisGroups [9] is another group formation strategy which considers location as well as meeting time. Johari *et al.* [8] have further utilized meeting location and frequency to form a group around a base station considering it as a POI. Contact probability based approaches have been suggested in [19], [20] where the authors have argued that people with similar interests form a group and usually meet at the same place. These works assume that all group members are physically close and can be reached using multi-hop communication which may not always hold true for a large outdoor area where connectivity may be intermittent.

SPOON [2] considers an interest based group where it is argued that people with similar interests meet more often and the list of interests is obtained from the stored files in mobile devices. It also uses meeting frequency as a measure for group creation and degree centrality for administrator selection. However, such regular meeting frequency and degree centrality measures are not available in irregular meeting places.

Another approach for group formation emphasizes on the social-relationship, arguing that people who meet more often have stronger social ties, and have more common characteristics regardless of the place of meeting. MOPS [10] is one such grouping method, however, group initialization and maintenance remain a problem for this approach. To handle initialization and maintenance a broker based grouping method has been studied in [21]. Here, the authors have considered broker nodes to initiate the grouping process and used meeting frequency and common interests for group creation. Similar social relationship-based groups are also used in [11] which employs a publish-subscribe type data dissemination and forms group with respect to social relationship such as family and friends. Administrator selection based on social relationship is also investigated in [22].

In summary, grouping methods found in the literature mostly assume existence of spatio-temporal regularity in human movement patterns or pre-existing social relationships and exploit that information. However, such information are unavailable in irregular meeting places where networks are formed for a short amount of time since users enter a particular POI, perform some activities, and may not come back to the same POI during the same trip. This does not allow enough learning period to capture relationships, if any exists at all. Group formation method in such a dynamic environment needs to collect and utilize information that are readily available and utilize those on the fly to provide satisfactory performance.

Another issue that affects content sharing success is the message forwarding techniques which can be broadly classified into two categories: mobility pattern based and social relationship based techniques. The first approach considers that nodes usually visit a particular set of locations in a predictable sequence on a regular basis which can be utilized for successful message forwarding.

MobySpace [23] and meetings and visits [24] uses mobility patterns of the users as well as the places they visit for packet forwarding. Predict-and-relay [25] considers that people's movement between landmarks is semi-deterministic provided that enough historical information about movement is available. However, none of these works consider the temporal properties of movements. Recently, Talipov *et al.* [13] have utilized both temporal and spatial regularity of movement pattern while forwarding requests. They proposed adding future time and location of the requester along with the content requests which can be used by the recipient to take forwarding decisions. The problem associated with mobility pattern based approaches is that they require a predictable regular movement pattern which is almost unlikely to be available in places such as a tourist spot.

Social relationship based message forwarding is inspired by relationships among participating nodes extracted from online social networks or built over time based on the time they spent together. Daly *et al.* first proposed SimBet [27] that uses betweenness centrality and then SimbetTS [14] that additionally considers tie strength with the destination for taking forwarding decisions. These techniques do not consider group membership for forwarding where nodes are only interested in forwarding messages for other members of the same group. Bubble rap [12] is another approach where messages are forwarded among the group members based on their popularity which is measured in terms of frequency of meeting with other group members.

This approach also makes the most popular node more congested. The social relationship based message forwarding techniques are also not applicable in an environment like tourist spot where people stay for a short duration, which is usually not long enough to build meaningful relationship.

Several content sharing approaches have also considered incentive scheme to encourage participation by nodes. Incentive schemes proposed in literature provide both monetary and non-monetary benefits. Practical Incentive (PI) [26] is a credit-based incentive mechanism which uses a third party server for credit management and provides monetary incentive for carrying messages. Similarly, a third-part server based incentive mechanism has been proposed in [27], [28]. However, the problem with such credit based scheme is the unavailability of such static third party server in environments under consideration. On the other hand, scheme such as MobiTrade [29] suggests use of non-monetary incentive where content itself is used as currency and nodes can obtain content if they have something to offer in exchange. It might be difficult for a node to decide which content to store when the storage space or contact opportunity is limited. ConSub [17] extends [29] by addressing this by using a utility function reflecting the benefits of storing a particular content based on the future prediction of content exchange with neighbours. However, these schemes require 'global knowledge' about content popularity and sufficient contact history is to be known to each node, both of which is difficult to establish in a tourist spot.

Trustworthiness and security is another important aspect of decentralized content sharing as nodes might forge some parameters to gain unfair advantage or spread malicious contents. Work in [30] considers monitoring node behaviour to assign reputation score to identify trustworthy nodes in opportunistic network. The reputation score is obtained from direct experience and recommendation from neighbours. Similar behaviour monitoring based reputation calculation is also proposed in [31], [32]. However, none of these works use such reputation score for administrator selection to filter out misbehaving nodes. Secured message exchange through key based encryption is proposed in [33], [34]. Jia *et al.* [33] have proposed a two-channel based public-key distribution scheme where nodes within proximity use an authenticated secure channel (i.e., channel which does not allow an adversary to modify message) to exchange verification information, and an insecure broadband wireless channel to exchange public key information. Recent work in [34] proposes composite trust-based public key management where nodes mutually certify each other's key based on their trust value. However, these approaches are not applicable in a tourist spot which lacks an infrastructure for key management and distribution, and certifying each other is not feasible as nodes are mostly unfamiliar.

Another stream of research concentrates on studying the tourist movement patterns, though communication among tourists or sharing contents has not been addressed at all. Works in [35], [36] assess the popularity of different regions in a tourist spot from a macroscopic level. They mainly focus on identifying key attractions, their level of attractiveness and the sequence in which they were visited. However, they did not consider individual movement pattern of tourists which is an important ingredient for capturing microscopic node behaviour and gives no indication about the types of contents they may be interested to share. Again their models are too specific to a particular site and time of the year, lacking generality, and building such movement pattern for each tourist site is costly and time consuming; hence not suitable in our case. Xia *et al.* [37] have proposed a semi-markov process to capture the spatial and temporal properties of human movement patterns at a tourist spot that can be used to capture attractive POIs in a tourist spot. Their model do not consider personal preferences and popularity of activities in the calculation of attractiveness or stay time. They also do not suggest any model to capture individual's movement pattern inside the POIs. Though the above works studied tourist behaviour, their focus is not content sharing at all.

3 Proposed Approach

The proposed approach considers the availability of different activities in a tourist spot. The areas designated for activities are called POIs and can be detected using any map application and/or online information provided by the visitor information centre. For example, a tourist spot can have different regions such as 'Bowling', 'Fishing', 'Camping', 'Accommodation', 'Food and Shopping' and 'Bush walking' which represent activities performed in those areas. While visiting a tourist spot when a user enters a POI, the content sharing application installed in a user's device automatically looks for existing groups and administrators by broadcasting a query message and setting up a timer to obtain this information. If it receives response from existing administrators before the timer expires, it joins one of the groups based on the criteria described in Section 3.2. In case of no response within the expiry time, it assumes that no group or administrator currently exists. In that case, the node communicates with other nodes within its range which do not belong to any group with the aim of selecting an administrator for the group to be formed.

In scenarios like visiting hotspots, nodes are most likely unfamiliar to one another and hence at the initiation of a new group formation process, information about trustworthiness of a node to others is unavailable. As detailed in the following section, to avoid excessive message passing, our administrator selection process is based on declaring self-assessment value that reflects a candidate node's suitability to be an administrator. By trustworthiness we mean how much a node can be trusted with the veracity of its declared self-assessment value so that the node would not be malicious in manipulating the selection process to become an administrator and then act maliciously, such as spread malicious contents or forge self-assessment value to always be selected as the administrator. Our scheme employs a trust model (Section 3.5.2) to assess trustworthiness of nodes once a group is established by observing how well behaved they are and uses this measure in filleting out untrustworthy nodes from subsequent administrator selection rounds.

Since trustworthiness information is not available for the very first time a group is to be formed, the initial administrator selection process includes an additional voting phase to reduce the probability of a node with forged self-assessment value being selected as an administrator. The voting process is as follows. Once multiple nodes are within one another's radio range, they broadcast *hello* messages to build neighbourhood tables. These tables are then shared to build a complete list of all nodes willing to form a new group which constitute the voter list. Since initially a small number of nodes are expected, the message overhead is low. Each node then votes for one candidate in the list randomly except itself and broadcasts its vote, and the candidate with the maximum vote is selected (or in case of a tie the one with lower node id) as the initiator. This selected node (i.e., initiator) then collects self-assessment value of other nodes for administrator selection and assigns a random trust factor to the participating nodes for filtering. This process minimizes the risk of a node taking over as administrator by forgery in two ways: (i) the voting process randomly selects an initiator and the voting result is known to all nodes, preventing a rouge node to declare itself as an initiator and (ii) random trust factor assignment at the second stage may assign a low value to a rouge node leaving it out of the candidate list for administrator selection.

In addition, as in [2], [13], we consider that the content sharing application will execute in a secured environment in memory to protect against spread of malicious contents or programs by other nodes.

Moreover, during the installation process of the content sharing application, the user of a mobile device will require to go through a registration process detailing information that identifies the device and owner. The content sharing app will use the IMEI (International Mobile Equipment Identity) number of the handset as the ID, which is a global identifier assigned by the handset manufacturer to uniquely identify each handset. During the installation time, the application can also check for any reported previous malicious activity by the user from publicly available security databases (e.g., www.lost.amta.org.au/IMEI can check the IMEI number to detect whether it is a lost or stolen device). If the content sharing app is commercially distributed, stricter checking such as verification through credit card can be employed. These checking, with identifiable details, will act as deterrent for users to do any malicious activity as the devices/users can be traced. Most of the applications now-a-days notify users during installation about the type of data that the application will collect in order to provide the functionalities it has been designed for (e.g., GPS location, access to WiFi router, camera, contacts and calendar). The users have the option to accept or reject before consenting to installation and the installation process only proceeds when a user agrees. Similarly our content sharing application will seek consent of users during installation. Additionally, a user must consent to the condition that she/he will not disclose any data that own device receives about others for the sharing application to work.

The administrator is finally selected through a selection process and starts group formation process. For existing groups, a new administrator needs to be selected every time the current administration session expires or handover becomes unavoidable. In this case, the current administrator collects self-assessment value from other participating nodes and assesses their trustworthiness to filter out the misbehaving nodes. Administrator selection plays an important role in the content sharing process as administrators are responsible for group management tasks and the selection process is presented below. Group formation or maintenance is another important aspects of content sharing which is discussed in Section 3.2.

3.1 Administrator Selection

In addition to participating in content sharing, an administrator creates and manages group content list and directs content requests towards the content holder(s). Existing approaches in the literature employ social metrics to select administrators [2], [22] which are not available in a tourist spot for reasons stated in earlier sections. We employ administrator selection policies utilizing the information readily available in tourist spots. We consider stay probability of a node in a POI, its connectivity value, available resources and its trust factor as the criteria for administrator selection. Stay probability ensures the availability of an administrator for a specific period of time for serving the group members, connectivity value takes account of its communication coverage inside the group, available resources ensure that the selected node is able to provide satisfactory services, and trust factor indicates whether the candidate is a well behaved node or not. Nodes within n-hop distance are considered as a contender for being an administrator. A smaller value of *n* ensures close connectivity among group members but increases the number of groups inside the POIs. We use 2-hop connectivity *(n=2)* as a trade-off as literature [38], [39] suggests that two-hop connectivity is sufficient to achieve maximum network throughput and minimize additional costs. Nodes that are more than 2-hop away are not considered for administrator selection. The node with the highest self-assessment value is selected as the new administrator and the node with the second highest self-assessment value is selected as the backup administrator. If the administrator fails due to low battery or some other unforeseen reasons, the backup administrator takes the responsibility of the group and declares itself as the new administrator. The administrator periodically sends group related information to the backup administrator, if there is any change (e.g., new member and updated content lists). The administrator selection process is formulated as:

$$\begin{aligned} & select \quad \kappa \\ & maximizing \quad P^{\kappa}\, C^{\kappa} \\ & s.t. \quad E^{\kappa} > E^{\kappa}_{m} \quad and \quad \Lambda^{\kappa} > \Lambda_{m} \quad and \quad \Phi^{k} > \Phi_{med} \end{aligned} \tag{1}$$

Here, P^{κ}, C^{κ}, E^{κ} and Λ^{κ} represent the stay probability, connectivity value, remaining battery life and available computational resource, respectively of node κ. The minimum amount of battery life and computational resource required to provide an acceptable service as administrator is denoted by E^{κ}_{m} and Λ_{m}, respectively. Φ^{k} and Φ_{med} depicts the trust factor of node k and the median trust factor, respectively. Calculation of these components is as follows.

Stay probability of a node represents its probability of staying at the current POI. A higher stay probability indicates that the node is expected to stay for a longer time at the current POI and therefore less administrator handover is needed. We use the probability density function (pdf) of stay time in POIs to calculate the stay probability. The content sharing app can store activities performed by a user in different POIs and their duration to build a history. If sufficient information is available about activities of a user and their duration then it can be used to calculate a personalized pdf to calculate stay probability more accurately. Existing works [37] suggest that the stay time in different POIs in a tourist spot follows different distributions such as Log-normal, Gamma or Weibull. Without loss of generality we use Log-normal distribution in our work for stay time calculation. We use a time window Θ for administrator selection cycle and determine the probability that node κ stays at the current POI for $T^{\kappa}_{f} = T^{\kappa}_{v} + \Theta$ time considering it has already spent T^{κ}_{v} time in it. Taking stay time of node κ as T^{κ}, the stay probability is calculated as,

$$P^{\kappa}\left(T_v^{\kappa} \leq T^{\kappa} \leq T_f^{\kappa}\right) = \int_{T_v^{\kappa}}^{T_f^{\kappa}} \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(ln\,t-\mu)^2}{2\sigma^2}} dt$$
$$= \frac{1}{2}\left(erf\left[\frac{ln(T_v^{\kappa}+\Theta)-\mu}{\sqrt{2}\sigma}\right] - erf\left[\frac{lnT_v^{\kappa}-\mu}{\sqrt{2}\sigma}\right]\right) \tag{2}$$

Here, μ is the mean and σ is the standard deviation of the stay time, and $erf(.)$ represents error function. μ and σ can be calculated from personalized historical data stored by the content sharing app or online resources such as tourism department [40] or statistics bureau [41] where statistics about time spent by people in different POIs for performing different activities is provided. A higher value of Θ ensures that the administrator stays for a longer session which in turns exhausts its resources; in contrast, a lower value introduces more traffic for frequent administrator handover. Therefore, as a trade-off we use half of the average time spent (i.e., $\mu/2$) in respective POIs for the value of Θ.

The connectivity value ensures that the administrator has enough coverage to communicate with the group members. It is calculated considering connectivity with the immediate and multi-hop neighbours. Each node periodically sends its neighbour list with the *hello* message which is used by the neighbouring nodes to maintain a neighbourhood table. From the neighbourhood table node κ calculates its connectivity value as,

$$C^{\kappa} = log(1 + h^1 + \frac{h^2}{2} + \frac{h^3}{3} + ... + \frac{h^n}{n})/log(H) \tag{3}$$

Here, h^n depicts the number of *n*-hop neighbours and H is used for normalization purpose only. At the beginning, when no group or administrator exists, the initiator will calculate and broadcast the value of H as $H = 2V$, where V is the number of nodes participated in the initial voting. Afterwards, the administrator will update the value of $H(= |\mathbb{G}| + 2V$, where $\mathbb{G}$ depicts a set of all members of group G) and broadcast it to the group members, if the number of current group members ($|\mathbb{G}|$) exceeds H. When a new node joins a group, the administrator sends the updated value of H to that node. A higher value of *n* introduces more traffic in the network as nodes have to broadcast list of their (*n-1*)-hop neighbours to maintain the neighbourhood table. As a trade-off we use 2-hop connectivity (*n*=2) as in [38], [39]. In literature, different values for *hello* interval (10 sec$\sim$10 min) are proposed [13], [42]. A short interval introduces more traffic whereas a long interval broadcasts outdated information. As a trade-off, we use 5-min as *hello* interval in our approach. The logarithm function in the numerator and the denominator of Eq. (3) ensure diminishing marginal utility and normalized connectivity value, respectively. A higher connectivity value ensures that the selected node will have more neighbours to provide services to the group members.

Node κ calculates its remaining battery life as,

$$E^{\kappa} = E_r^{\kappa}/E_f^{\kappa} \tag{4}$$

where, E_r^{κ} and E_f^{κ} represent the remaining energy of κ and its full energy. The minimum energy E_m^{κ} required by node κ to serve as an administrator is calculated as,

$$E_m^\kappa = \left(E_u^\kappa + E_a\right)/E_f^\kappa \tag{5}$$

Here, E_u represents the energy consumption due to regular use (e.g., call, sms, apps) and the battery drain for other reasons, and E_a shows the energy required to serve as an administrator during the time window Θ. We assume the content sharing or other app can periodically record the energy usage status, this information is used to estimate the energy drain E_u^κ due to regular usage. Since request arrival in content sharing process follows a Poisson distribution [11], the energy required to serve the group members is calculated as

$$E_A = \epsilon\,\lambda\Theta \tag{6}$$

Here, ϵ shows the energy required to receive and reply a request and λ represents the request arrival rate. Since the size of the content request and reply (i.e., content description metadata and list of holder(s)) packets are fixed, packet size is used to calculate the value of ϵ using the energy model presented in [43]. Nodes calculate request arrival rate from past experience of content sharing and update it over time.

The resource availability metric is calculated as,

$$\Lambda^\kappa = (\vartheta^\kappa - \vartheta_u^\kappa)(\iota^\kappa - \iota_u^\kappa)/\vartheta_{max}\iota_{max} \tag{7}$$

Here, ϑ^κ and ι^κ denote the available resources at node κ for processor and RAM, respectively. ϑ_u^κ and ι_u^κ are the currently used respective resources by the operating system and other running applications. ϑ_{max} and ι_{max} represent the maximum configuration available in the market and used for normalization purpose. The minimum resource required Λ_m is calculated by taking the median value of the device configurations currently available in the market. Calculation of trust factor (Φ^k) is discussed in Section 3.5.2 in details and Φ_{med} denotes the median trust factor.

3.2 Group Formation

Whenever an incoming node enters a POI, two cases may arise. In the first case, it does not find any existing group/administrator and hence, following the method in Section 3.1, starts administrator selection process in collaboration with surrounding nodes which are not member of any group. After the administrator is selected, it starts group formation procedure. In the second case, there is existing group(s) and the newcomer node broadcasts a query message to obtain information about that group(s). The query message includes information such as its own interest categories (activity e.g., 'bowling' or content type e.g., 'movie') and interest scores along with its expected stay time. Whenever the administrator of the group receives this query message, it replies with group information, namely the maximum interest score and the number of contents in each category, average stay probability of all group members, category-wise average stay probability (calculated on members in specific category), and its own delay experience for receiving messages. Note that, the administrator only sends out group information if the observed delay in the group is within the tolerable delay of the current group members suggesting that a new member may be accommodated into the group. Other non-administrator nodes who are one-hop neighbours of the inquirer also reply with a message that contains their group id, stay probability at the current POI and their own delay experience for receiving messages.

A node may get response from a single or multiple groups. For example, in Fig. 1 a new incoming node *k* receives information from the surrounding groups G1 and G2. If the node receives response from a single group, then it decides to join that group. If information from multiple groups is received then the node uses group joining criteria to decide which group to join. From the content sharing perspective, it is important to consider before joining a group that the group has contents of the user's interest which will be available during the user's stay in the group and delivered within its tolerable delay limit through group members and administrator. We consider interest fulfilment probability (ω_G^k), content availability probability (π_G^k) and content delivery probability (ψ_G^k) of group *G* for node *k* as the joining criteria. Using these criteria, an objective function for joining a group is defined as,

$$\begin{aligned} &select \quad G \\ &maximizing \quad \Upsilon_G^k = \omega_G^k \, \pi_G^k \, \psi_G^k \\ &s.t. \quad \overline{D}_G^k \leq D^k \quad and \quad h_G^k \leq n \end{aligned} \tag{8}$$

Here, Υ_G^k represents the gain factor for joining group *G* for node *k*. $\overline{D}_G^k$ shows the average delay experienced by group members of G for receiving messages, and D^k shows the maximum tolerable delay of node *k*. h_G^k is the hop-distance of node *k* to the administrator of group *G* and *n* depicts the maximum allowable hop-distance to the administrator as discussed in Section 3.1. In Eq. (8), we use multiplication to ensure that all components are given equal weight. If any of the components is unavailable we consider its value as the average (0.5) for the range [0∼1]. In the example in Fig. 1, node *k* calculates the gain factor for groups *G1* and *G2*, and sends a joining request to the group administrator that yields highest gain factor. If none of the groups are able to provide expected service (i.e., delay within tolerable limit), the new comer node k can start a new group with the help of other nodes. Upon receiving the joining request the administrator accepts and sends an acknowledgement back. The new member then sends a summary of its own content list to the administrator who updates the group content list accordingly. The group members also send updated summary if there is any change (i.e., new content is added).

3.2.1 Interest fulfilment probability

Interest fulfilment probability computes whether a user will be able to obtain contents of interest if it joins a particular group. Content access in a tourist spot is motivated by the activities available in that place, for example, contents related to fishing or camping. Content access can also be generic regardless of the activities available like music, videos or news. We consider both activity and content related interest while calculating interest fulfilment probability. The level of interest in a category (we use category to denote an activity or content type) is represented using an interest score. We assume that the content sharing program installed in a user's device can automatically calculate user's interest score from personal profile, online recommendation, and location-centric experiences. We use multiple sources for interest score calculation as it ensures relevancy in calculation.

Personal profile Content sharing application facilitates the user to maintain a personal profile which stores and updates any category and the level of interest associated with that category. A list of such activities (e.g., boating, fishing, sight-seeing) and contents (e.g., music, movies, news) with their assigned levels of interest (from 0 to 10) can be created manually by the user. Such a list is used to calculate a generalised personal interest score regardless of the nature or facilities available in any tourist spot. Interest score for category *i* from such personal profile is calculated as,

$$\beta_i = L_i / L_{max} \tag{9}$$

Here, L_i represents the level of interest for category *i* and L_{max} represents the maximum assigned level (i.e., 10 in this case). The interest score for a particular category might change based on certain facilities available in a particular tourist spot. For example, a tourist might have a low interest score for boating but may be interested to do boating while visiting a particular spot which is very famous for boating due to the nature of the waves and weather. This type of characteristics is captured later in the location-centric experience calculation.

Online Recommendations In real life, when users are not familiar with a tourist spot they seek online recommendations. A user might become interested about some of the recommended activities or content types when (s)he receives such suggestions. For example, if close friends of a user express opinion about enjoying camping on a site, this may convince the user to do so. Online Social networks (OSNs) and Instant Messengers (IMs) such as Facebook, Twitter, WhatsApp and Viber provide instant feeds and responses from interactive friends/followers and recommenders. Content sharing applications can obtain such recommendations by collecting posts about a category and analyze them automatically using a sentiment-based text analysis technique [44] to derive positive or negative recommendations from each post, however, development or appropriate selection of such technique is out of the scope of this paper. A score can be extracted from such interactions/recommendations by counting the number of positive posts and comments about a category. We use a recent time window w to prioritize more recent posts or comments. A user's interest score for category *i* from online recommendations is calculated as,

$$\gamma_i = \Delta \frac{\sum_{j \in \mathbb{V}_i} \alpha^j_{i,w} \Psi^j_i}{\sum_i^N \sum_{j \in \mathbb{V}_i} \alpha^j_{i,w} \Psi^j_i} + (1 - \Delta) \frac{\sum_{j \in \mathbb{V}'_i} \alpha^j_{i,w'} \Psi^j_i}{\sum_i^N \sum_{j \in \mathbb{V}'_i} \alpha^j_{i,w'} \Psi^j_i} \tag{10}$$

Here, $\alpha^j_{i,w}$ is the number of recent posts within the time window w ($t_1{\sim}t_2$) about category i by recommender j (online friend/family/tourist website), $\alpha^j_{i,w'}$ depicts the same within the time window w' ($t_{-x}{\sim}t_1$, where $-x$ refers to the past x time units), Ψ^j_i represents the reliability of recommender j about category i, $\mathbb{V}_i$ is the set of all recommenders for category i within w, N is the total number of categories under consideration, and Δ represents the weight for the recent posts. A higher Δ suggests more emphasis on the recent posts. The value of Δ is calculated using the weight calculation of exponential moving average as, $\Delta = 2/(W+1)$, where, W $(=w + w')$ is the total size of the window.

Since there is no face-to-face communication for online recommendations and the recommenders are most likely unfamiliar to the user, the reliability of recommendations is an issue. We integrate reliability measure of the recommenders in the proposed approach. We consider that the content sharing app receives different levels of recommendation from different recommenders. From these recommendation values we find the majority supported recommendations (i.e., the median of all recommendations) and any variation from it represents a deviation of reliability. Such consideration is consistent with real life scenarios where a user is more likely to trust recommendations which are consistent across many people. We calculate the reliability of a recommender [45] j about category i during time window w as,

$$\Psi^j_{i,w} = \frac{\eta^j_i}{\max_{\forall j}\left(\eta^j_i\right)}\left(1 - |\overline{\chi_{i,w}} - \chi^j_{i,w}|\right) \tag{11}$$

Here, η^j_i expresses the the length of exposure of recommender j for category i which is calculated using the number of online posts a particular recommender made for i within time window W. We consider that more number of posts made by a recommender indicates more exposure. $\overline{\chi_{i,w}}$ and $\chi^j_{i,w}$ represent the median of all the recommendations and the recommendation score assigned by j about category i within time window w. To incorporate earlier reliability score of a recommender, the reliability Ψ^j_i of recommender j can be calculated using the exponential moving average of $\Psi^j_{i,w}$ as,

$$\Psi^j_i = \varphi\Psi^j_{i,w} + (1-\varphi)\Psi^j_{i,w'} \tag{12}$$

Here, φ is the weight for exponential moving average calculation and can be calculated using the methods stated earlier.

Location-centric experience Different tourist locations provide different types of attractions and facilities to visitors. Activity and content consumption in a tourist spot are certainly influenced by the nature of a tourist spot. For example, a particular user Alice might only be interested in fishing whenever she visits location Q. We consider such characteristics as location-centric experiences which suggest that, while visiting the same place a user is more likely to perform similar activities and consume similar contents. Experiences from visits (i.e., activities performed or content consumed) at a particular tourist spot are saved in a user's device for future interest score calculation. The interest score for category *i* at a particular tourist location is calculated as,

$$\delta_i = \rho_i / \sum\nolimits_{y=1}^{N_Q} \rho_y \tag{13}$$

Here, ρ_i depicts the number of times *i* occurred during past visits, and N_Q shows the total number of categories consumed at location *Q*.

For the initial visit to a place we cannot calculate (13) directly. In that case, we calculate content related interest as the number of times a particular type of content has been accessed over all content access. For activity related interest, we use the popularity of an activity in that particular spot and the participation in the same activity by the user in other places. Such popularity information is available from tourism research department [40] which can be collected automatically by the content sharing app. The interest score from location-centric experience for the initial visit is calculated as,

$$\delta_i = \theta\phi_i + (1-\theta)\sum_{z=1}^{M} \Im_{i,z}/M \tag{14}$$

Here, ϕ_i represents the popularity of *i* at the current tourist location which indicates the fraction of visitors performing *i*. *M* represents the total number of visits by that user to similar places where *i* was available, $\Im_{i,z}$ represents whether *i* occurred during a visit to *z*, and takes binary value (0 or 1). *θ* represents the weighting factor and calculated as, *θ*= 2/(*M*+1). If the number of visits to places where *i* was available is small, then *θ* is high and the popularity of *i* at current spot gets emphasized.

Having obtained three types of interest scores for category *i* ($\beta_i, \gamma_i, \delta_i$), appropriate weight on each is needed to calculate the final score. The interest score obtained from personal profile is provided by the user and is a better representation to reflect the actual interest of the user. This suggests that we should give more emphasis to scores obtained from personal profiles. However, sometimes interest score obtained from other sources influences the behaviour of the user too. For example, Alice might be very interested in boating, but while visiting a particular tourist spot M_Q, she never does boating as it is unsafe for her to do boating there due to big waves. In this case the location-centric experience will reflect her accurate interest score. Another user Bob who is usually reluctant to go for fishing, may be convinced to try fishing in spot M_Q on receiving highly positive recommendations for fishing there. In this case online recommendations have huge impact on his interest score. We use the standard deviation σ_i of interest scores for a category *i* which is the variation of scores across different sources and places to determine which source should be given more weight. When this variation is low more weight is given to personal profile as it reflects more accurate level of interest. On the other hand, a high variation suggests that online recommendation and location-centric experiences may imply different level of interest than the personal profile due to the nature of the place or others' suggestions as discussed earlier, and therefore, personal profile is given lower weight. The value of σ_i ranges from 0$\sim$1 as all the interest score values are within 0$\sim$1. Finally, we calculate the overall interest score as,

$$S_i = \beta_i(1-\sigma_i) + \sigma_i(\gamma_i + \delta_i)/2 \tag{15}$$

While visiting a tourist spot the content sharing app calculate interest score for different activities and content types using the method discussed above. Such interest scores are used to generate the user interest matrix U^k of user *k* as,

$$U^k = \{(I_1^k\ S_1^k), (I_2^k\ S_2^k), ..., (I_n^k\ S_n^k)\} \tag{16}$$

Here, I_n^k indicates *n*-th category of interest and S_n^k indicates the score for that category. The administrator of the group also keeps a list of the maximum interest score for each category available in the current group as,

$$S_{i,G} = \max_{l \in \mathbb{G}} \left(S_i^l\right) \tag{17}$$

Here, $\mathbb{G}$ is the set of all members in group *G*. Whenever a new node joins a group the administrator updates $S_{i,G}$. Upon receiving an inquiry from a newcomer *k* in the POI, the administrator finds the matching categories from the group interest list and sends their maximum group interest score as, $U_G = \{(I_{1,G}\ S_{1,G}), (I_{2,G}\ S_{2,G}), ..., (I_{n,G}\ S_{n,G})\}$. The interest fulfilment probability checks whether the advertised group has any member with at least the same level of interest and hence the capability of serving future requests from *k*. We calculate the category-wise difference in interest score for *G* and *k* as,

$$\tau_{i,G}^k = \begin{cases} 0, & \text{if} \quad S_{i,G} \geq S_i^k\, or,\, S_i^k = 0 \\ S_i^k, & \text{if} \quad S_{i,G} = 0\, and\, S_i^k \neq 0 \\ S_i^k - S_{i,G}, & \text{otherwise} \end{cases} \tag{18}$$

Finally, the interest fulfilment probability ω_G^k of the advertised group *G* w.r.t. *k* is calculated as,

$$\omega_G^k = 1 - \frac{1}{N^k}\sum_{i=1}^{N^k} \tau_{i,G}^k \tag{19}$$

Here, ω_G^k represents the interest fulfilment probability for group *G* calculated by node *k* and N^k represents its total number of categories of interest. A higher interest fulfilment value indicates the group has more probability of fulfilling the user's interest.

3.2.2 Content Availability Probability

Content availability probability indicates the probability that the content of interest will be available in a group during a node's stay in the group and it can be delivered within an acceptable amount of delay. Literature states that a user's perception of content availability depends on the reception of content within the required time [46]. We use the category-wise average stay probability of the group members and the average delay experience of a group to measure the content availability probability for that group. Category-wise average stay probability for a category *i* represents the average stay probability of the group members who are interested in *i*.

Upon joining a group a new member calculates its stay probability using the methods presented in 3.1 and sends it to the administrator. The administrator use this information to update category-wise average stay probability of the group members which is depicted as,

$$\overline{P_{i,G}} = \frac{1}{|\mathbb{G}_i|} \sum_{l \in \mathbb{G}_i} P_i^l \tag{20}$$

Here, $\overline{P_{i,G}}$ represents the average stay probability of group *G* for category *i*, $\mathbb{G}_i$ is the subset of $\mathbb{G}$ who are interested in *i*, and P_i^l is the stay probability of node $l \in \mathbb{G}_i$. As alluded before, the administrator sends $\overline{P_{i,G}}$ to the inquirer node *k* which then calculates the content availability for group *G* as,

$$\Pi_G^k = \frac{1}{N^k} \sum_{i=1}^{N^k} \left(S_i^k \, \overline{P_{i,G}} \right) \tag{21}$$

Here, S_i^k shows the interest score of node *k* for category *i* and N^k is the total number of categories *k* is interested in.

Content sharing being a delay tolerant application, users will be interested in the ultimate delivery of the requested content while setting a maximum tolerable delay, which can be set by the user using content sharing application. We use the average of the maximum tolerable limits if different requirements are used by a single user for obtaining different contents. The administrator updates its own maximum delay information from the messages received from other group members using an exponential moving average as,

$$d_G = \Gamma d^c + (1 - \Gamma) d^p \tag{22}$$

Here, d_G represents the maximum delay experienced by the administrator, d^c shows the maximum delay observed during current period, d^p represents the maximum delay in the past and Γ is the weight used for exponential moving average calculation (Γ is calculated using similar method stated before). We use the interval between two consecutive *hello* messages (5 min) as the size of a single time period for updating delay. The administrator node checks the timestamps of the received message to update maximum delay. Non-administrator group members also maintain such maximum delay statistics.

On receiving group inquiry messages from a new node *k*, the administrators as well as its one-hop neighbours reply with their observed maximum delay. The delay experience of one-hop neighbours is used as the newcomer node has high probability of experiencing similar delay for message delivery. Afterwards, node *k* determines the average delay for group *G* as,

$$\overline{D}_G^k = \frac{\sum_{l=1}^{|\mathbb{G}^{k,1}|} D^l + d_G}{1 + |\mathbb{G}^{k,1}|} \tag{23}$$

Here, $\mathbb{G}^{k,1}$ is the subset of $\mathbb{G}$ who are also one-hop neighbours of node *k*, D^l is the maximum delay reported by node $l \in \mathbb{G}^{k,1}$, and d_G is the delay reported by the administrator of *G*.

3.2.3 Content Delivery Probability

We further calculate content delivery probability to assess a group's capability to deliver contents to requesters once they join a group. Successful content delivery relies on the presence of sufficient relay nodes and the number of available copies of the requested content. Unlike content availability probability, in this case we utilize average stay probability of all group members regardless of their interest as all of them can take part in relaying the content. We also use neighbourhood and content list factor (discussed below) in the calculation of content delivery probability. After receiving inquiry from a newcomer node *k*, the administrator sends the average stay probability of group members ($\overline{P_G}$) and group content list factor (ζ_G) along with the other information stated previously. Node *k* calculates the stay probability of the members of group *G* for content delivery as,

$$P_G^k = \frac{\sum_{l=1}^{|\mathbb{G}^{k,1}|} P^l + \overline{P_G}}{1 + |\mathbb{G}^{k,1}|} \tag{24}$$

Here, P^l represents the stay probability of node $l \in \mathbb{G}^{k,1}$, and $\overline{P_G}$ is the average stay probability reported by administrator of *G* (averaged over all *i* as per (20)). A higher value of stay probability indicates that the members are more likely to stay longer and have a better chance of delivering the content. The presence of sufficient number of relay nodes is accounted as the neighbour factor as, $F_G^k = |\mathbb{G}^{k,1}|/h^{k,1}$, where $h^{k,1}$ represents total number of one-hop neighbours around node *k* irrespective of group membership.

We consider that a group with more contents with multiple copies has a better probability of successfully delivering any request. Although, multiple copies of a single content with different members enhance the probabilities of successful delivery, it does not add any extra benefit when the number of copies exceeds certain limit. We consider that 2-copies of a content is sufficient to ensure successful delivery as evidenced in the works presented in [47]. A group content list factor is the sum of the weights for *k*'s interested categories where a weight of 0.5 is used if a single copy of content is available in the group, and a weight of 1 is used if 2 or more copies are available. Node *k* also calculates the normalized content list factor for group *G* as, $\zeta_G^k = \zeta_G/\zeta_{max}$ where, ζ_G represents the content list factor reported by the administrator of *G*, and ζ_{max} represents maximum content list factor reported by any of the administrators. Finally, node *k* calculates the content delivery probability for group *G* as,

$$\psi_G^k = P_G^k \, F_G^k \, \zeta_G^k \tag{25}$$

Using Eq. (19), (21), (23) and (25) in Eq. (8), the inquirer node *k* selects the group amongst the advertised ones that yields it highest gain factor and then sends a join request to the administrator of that group. The group formation process is outlined in Algorithm 1.

3.3 Other Grouping Issues

3.3.1 Arrival of Node with Higher Self-assessment Value

There might be cases where a newly joined group member may have a higher self-assessment value ($P^\kappa C^\kappa$ in Eq. (1)) to better qualify as the administrator of that group. In such case, the new node may be elected as administrator through a handover process. Such handover policy is more likely to create instability among group members as not all members will receive timely update about this change due to delay in broadcasting and node mobility; effectively creating a situation where multiple nodes are treated as administrator by different members. It will also create excessive handover and will require extra control messages to manage everything, introducing overhead in the network. To avoid instability and reduce message overhead we do not change the administrator until the current administrative session is over. When the current administrator is close to the end of the session, it starts the administrator selection procedure as discussed in Section 3.1. When the new administrator is selected, the current administrator passes the group information to the newly selected administrator.

3.3.2 Changing Group Membership

Due to mobility some nodes might be separated from other group members. We consider a node's connectivity to the administrator as the criteria for changing group membership. Nodes check *hello* messages from the neighbours in the neighbourhood table to see if the administrator is reachable within *n*-hop distance as discussed in Section 3.1. If the administrator is not found from the neighbourhood table within the *hello* message interval then the node starts a timer. If the administrator is not found within three consecutive *hello* intervals then the node assumes a disconnection with the group as in [2] and looks for other existing groups and joins a new group based on the group selection criteria.

3.3.3 Relaying data for another group

We consider *n*-hop connectivity with the administrator while joining a group. Our group joining criteria does not require any member to be within its one-hop distance. It might happen that node *k* joins a group *G* where none of its one-hop neighbours are from group *G*. In this case, for message forwarding it requires

Algorithm 1: Group Formation Procedure

```
procedure GROUP FORMATION
    Node $k$ broadcasts message to obtain available group and administrator info;
    It calculates and stores gain factor using Eq. (8) for all groups in $\Upsilon^k_G = \{\Upsilon^k_G | 1 \leq G \leq q\}$,
    where $q$ = the number of groups from whom node $k$ receives info;
    if $\Upsilon^k = null$
        $k$ starts the voting process by broadcasting a voter list $\mathbb{V}$;
        $k$ broadcasts its own vote and stores vote received from others in a set $V^\ell = \{V^\ell | 1 \leq \ell \leq x\}$,
        where $x$ = the number of votes received by each node;
        if $arg(max(V^\ell)) = k$
            Initiate administrator selection;
            $k$ calculates self-assessment value using Eq. (1), assigns random trust factor and stores
            assessment values in a set $A^\ell = \{A^\ell | 1 \leq \ell \leq y\}$, where $y$ = the number of nodes including
            itself from whom it receives broadcast info and has higher trust factor than median;
            if $arg(max(A^\ell)) = k$
                Start a new group;
            else
                $k$ selects and notifies $f = arg(max(A^\ell))$;
                $f$ starts a new group;
            end if
        end if
    else
        Find group $G$ with maximum Gain i.e., $G = arg(max(\Upsilon^k_G))$;
        Send join request to the administrator of $G$;
    end if
end procedure
```

the service of members from other group and joins another group *B* for message forwarding purposes and indicates to B's administrator about its willingness to carry messages for group *B* in lieu of the services it will get. Upon receiving such request, group *B* administrator accepts it and requests its members to carry messages for node *k* who, in turns, carries messages for group *B*. In this case messages from the members of the content sharing group get priority for forwarding by node *k*.

3.4 Request Generation and Content Forwarding and Delivery

Nodes generate content request based on their interest from which keywords are extracted using text analysis technique discussed in [2]. Such keywords can be matched against the meta-data associated with a particular content using content matching methods discussed in [48], [49] to locate appropriate content. The requester first sends the request to its one-hop neighbours, who match the request with their stored content list to determine if they have the matching content or not. If none of the neighbours reply, the requester forwards the request towards the administrator. When the administrator receives a request from a group member, it checks the group content list (collected from group members) to determine who has the matching content. If the content is found, the administrator forwards the request to the content holder(s) and notifies the requester about its availability along with a summary (i.e., content size and id of content holder(s)). In this case, a group might have multiple content holders for a single content as well as the administrator itself might have a copy. The administrator initially checks its neighbourhood table to determine if any of the content holders are located within its one-hop distance and notifies the node from whom the latest *hello* message is received about the request. If no suitable content holders are found, the administrator itself delivers the content. However, this is the least preferred option as other nodes might experience longer delays if the administrator itself is busy in delivering the content. On receiving a request, the holder delivers the matching content to the requester. The following describes the content forwarding mechanism.

When a content carrier node (i.e., content holder or carrier) meets a neighbour, it can either continue carrying the content by itself for future delivery or forward the content. Successful content delivery relies on the probability of nodes (i.e., both carrier and requester) staying in the same POI and the carrier node having a higher connectivity and enough available resources. Staying in the same POI as the requester node increases the probability of having more opportunistic contact. A carrier node uses these criteria to calculate a utility value. Afterwards, it considers itself and all its one-hop neighbours as the potential forwarder and selects the node with the highest utility value as the forwarder. The selected forwarder node then carries the content, and if it meets another node with higher utility value then forwards the content to that node. The forwarder selection process can be formulated as a maximization problem as,

$$\begin{aligned} &select \quad f \\ &maximizing \quad O_a^f \\ &s.t. \quad E^f > E_a \quad and \quad B^f > B_a \end{aligned} \tag{26}$$

Here, O_a^f represents node f's utility to forward content a. E^f and E_a represent energy available to node f and minimum energy required to forward a, respectively, while B^f and B_a represent the buffer space available to node f and the minimum space required to store a.

A node's capability to successfully deliver a content depends on its expected co-location probability with the destination node and its connectivity value. Therefore, the utility value of node f to deliver content a is calculated as,

$$O_a^f = \begin{cases} 1, & \text{if } h^{f,d} = 1 \\ R^{f,d} C^f, & \text{otherwise} \end{cases} \tag{27}$$

Here, $h^{f,d}$ shows the hop-distance between node f and requester node d. $R_a^{f,d}$ represents the co-location stay probability of nodes f and d, and C^f is the connectivity value of node f. The co-location stay probability is calculated as,

$$R^{f,d} = \varpi R_c^{f,d} + (1 - \varpi) R_n^{f,d} \tag{28}$$

In Eq.(28), $R_c^{f,d}$ refers to the co-location stay probability of nodes f and d in the current POI 'C', and $R_n^{f,d}$ shows the same in the next POI. $R_n^{f,d}$ is considered as the requester might finish its current activity and moves into the next POI before obtaining the content. In that case, the content can still be delivered by a forwarder moving to the same POI, if it meets the requester before the lifetime of the request expires. ϖ is used as a weighting factor. The co-location stay probability of nodes f and d at current POI 'C' is determined as,

$$R_c^{f,d} = \begin{cases} 1, & \text{if } T_c^f \geq T_c^d \\ \frac{T_c^f}{T_c^d}, & \text{otherwise} \end{cases} \tag{29}$$

Here, T_c^f and T_c^d show the remaining stay time of nodes f and d in the current POI, respectively. The requester calculates its own expected remaining time in the POI and sends it along with the request. Similarly, the forwarder node also calculates its own expected remaining time and calculates the co-location stay probability. Similar to Section 3.1, using the probability density function (pdf) of stay time, the expected stay time of user f at POI 'C' can be calculated as,

$$T^f = e^{\mu + \frac{1}{2}\sigma^2} \tag{30}$$

In the above equation μ and σ are as defined in Eq. (2) w.r.t. POI 'C'. Finally, remaining stay time at the current POI 'C' can be calculated as, $T_c^f = T^f - T_0^f$, where, T_0^f is the time already spent by node f in 'C'. The weighting factor ϖ in Eq. (28) dictates which co-location stay probability (i.e., in the current POI or the next) should be given more priority and calculated as,

$$\varpi = \begin{cases} 1, & \text{if } T_d^c \geq Q_a \\ \frac{T_d^c}{Q_a}, & \text{otherwise} \end{cases} \tag{31}$$

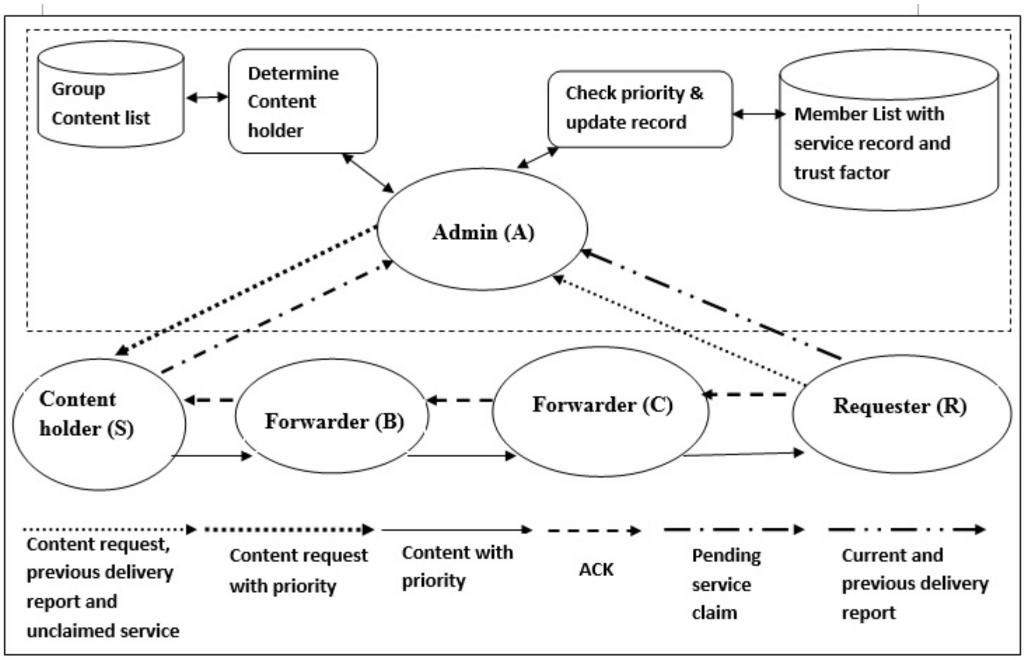

Group Content list
Determine Content holder
Check priority & update record
Member List with service record and trust factor
Admin (A)
Content holder (S)
Forwarder (B)
Forwarder (C)
Requester (R)
Content request, previous delivery report and unclaimed service
Content request with priority
Content with priority
ACK
Pending service claim
Current and previous delivery report

Fig. 2: Participation incentive in decentralized content sharing

Here, Q_a represents the lifetime of the content request a. If the requester node stays longer in the current POI, co-location at current POI is given higher priority and vice-versa.

The connectivity value and energy related information are calculated using the method discussed in Section 3.1. Users specify the amount of storage they are willing to allocate for the content sharing app and also for carrying other users' request. By default any suitable value can be used by the content sharing app. Node f checks the available storage space and the size of content a to decide whether to carry the content or not.

Finally, node f calculates its utility value using Eq. (27) and reports it back to the content carrier node. Upon receiving utility value from a neighbour, the carrier node decides whether to forward the content or not. This way successive carrier nodes carry the content until it is successfully delivered or the request lifetime expires.

3.5 Participation incentive and Node Trust Factor calculation

3.5.1 Incentive score

In a collaborative participatory scheme like content sharing, nodes are considered cooperative, where they help each other by providing and forwarding contents. However, in real life, selfish nodes exist, where nodes may not fully or at all cooperate without any incentive even though their resources permit participation. To encourage participation, we employ an incentive mechanism that rewards participation of a node through priority processing by the administrator, content holder and forwarders, which is expected to yield higher delivery success rate for its requests. The priority is assigned by an incentive score, calculated by the administrator. In order to keep the message passing and calculation overhead to a minimum, incentive score calculation for a node is based on its previous participation in successful end-to-end content delivery.

Consider the scenario depicted in Fig. 2 with the role of the nodes as labelled. In our scheme, a requester (R) sends a delivery report to the administrator (A) notifying the receipt of the content from the holder (S) through the forwarders (B, C) as the delivery path is recorded. The report by R shows the request ID, list of participatory nodes (S, B and C in the example in Fig. 2) and the size of the content delivered, and this report forms the basis of incentive score calculation and update. In case a delivery report is lost on the way to the administrator, the current report also contains three previous reports with their respective request ID so that any pending request can be processed.

The administrator updates the incentive score of each participatory node k for delivering a content after receiving a delivery report as,

$$\Omega^k(t) = \varrho \frac{\varsigma_a}{\varsigma_{max}} \left(1 - \frac{d_a}{Q_a}\right) + (1 - \varrho)\Omega^k(t-1) \tag{32}$$

Here, ς_a is the size of content a and ς_{max} is the maximum size of any content in the group content list. d_a represents delay for obtaining content a and Q_a shows the request lifetime for the same. ϱ represents a weighting factor for exponential moving average calculation and $\Omega^k(t-1)$ is the previous incentive score of node k. Each node is assigned an initial incentive score of 0.1 after joining a group, which can be increased by providing service to others. In return, nodes with a higher incentive score get higher priority for receiving content.

The reward based on incentive score is implemented as follows. On receiving a content request from R, the administrator checks its current incentive score from the updated incentive list of all group members. The administrator processes all its pending requests by sorting them in descending order of incentive scores, so that the request with the highest score is served first. While forwarding the request to the content holder, the administrator also includes the requester's incentive score. S adds this new request to its pending request queue and processes all requests in order of deceasing scores whenever it meets a forwarder B. Similarly, forwarders also checks their pending request queue and forwards content to the next node as per the incentive score of pending requests. Each node also updates the delivery path to record the list of participating nodes in the delivery to R. Processing requests on the basis of incentive exhibits benefit for active nodes in term of higher delivery as demonstrated in simulation results.

3.5.2 Trust factor calculation

As alluded before in Section 3.1, trust factor is used to identify misbehaving nodes and prevent them from becoming the administrator of the group. The trust factor measures a node's inclination to claim forged value, i.e., a node that tends to make false claim for incentive is likely to forge self-assessment value to inflate its claim to become administrator. This is implemented by requiring each node to claim its service to the administrator for rendering or forwarding data and the administrator comparing the claim with the delivery report it receives from the requester. The administrator maintains and updates trust factor for all group members. Each node is initially assigned with a trust factor of 0.5 (assigned after the group is formed), and it is updated when a service claim is received from a node. Note that the incentive is updated and awarded as soon as the delivery report is received by the administrator, but the trust factor is updated only after a service claim is lodged and verified. Refer to Fig. 2, when R sends a delivery report to A, it also sends an acknowledgement along the delivery path to nodes C, B and S (indicated by dotted line) and the nodes claim service only after receiving this message. The trust factor value increases if a node makes an honest service claim and decreases if a false claim is made. The following cases may happen here as discussed below.

Case 1: In this case, the administrator successfully receives the delivery report before the service claim. It matches the claim against the report to check for false (claim against a request ID which was not delivered or not part of forwarding) or over-claim (e.g., size of the content forwarded). The administrator updates the trust factor of node k as,

$$\Phi^k(z) = \begin{cases} \frac{1}{2}\left(\Phi^k(z-1) + \nu_h^k/\nu_T^k\right), \text{ if honest claim} \\ \Phi_{med}\left(1 - \nu_f^k/\nu_T^k\right), \text{ otherwise} \end{cases} \tag{33}$$

Here, $\Phi^k(z)$ shows the trust factor of node k after its z-th claim. ν_t^k, ν_f^k, and ν_T^k represent the number of honest, false and total claims made by node k, respectively. Φ_{med} is the median trust factor of the group and calculated by the administrator. The above formulation limits trust factor between [0$\sim$1], gradually increases trust factor for well-behaving nodes towards high value and punishes misbehaving nodes by lowering their value below Φ_{med}, ensuring their exclusion from administrator selection in the succeeding round as per Eq. (1). Note that trust factor is used as hurdle for administrator selection.

Case 2: If the administrator receives the service claim before receiving the delivery report, it waits for the report before updating the incentive score and trust factor. This may happen due to loss of delivery report because of mobility and lost connection. To handle this, the requester node resends the report with its next content request and next delivery report; each report with corresponding request ID. If the administrator has not received any delivery report for a service claim, it sends a query message to which the latter (i.e., requester) responds. The requester node responds to this query with its latest delivery report. If the administrator still does not receive the corresponding delivery report, it stores the service claim and does not process it. Note that in our scheme a node is not punished till its claim is proved false.

4 Performance Evaluation

4.1 Simulation setup

Since there is no publicly available dataset that demonstrates movement of tourists in a tourist spot, we generated synthetic mobility data to explore the content sharing process in a tourist spot using Lakes Entrance as a test location which is one of the prominent tourist spots in Victoria, Australia. We marked the POIs for available activities {'Accommodation', 'Bowling', 'Camping', 'Fishing', 'Food and Shopping', 'Walking'} using a map application after collecting information from tourist portals and local maps that suggested various activities available in Lakes entrance along with their designated areas. For generating the synthetic mobility data, nodes entered the simulation area using the Poisson distribution and stayed in the whole tourist area using the negative binomial distribution, which are consistent with the characteristics of tourists in tourist spots [50]. For generating individual node mobility, activity related interest scores and stay time were used. Using the average stay time of tourists in respective activities provided in the tourism related information [40], [41] as a basis, the stay time of a particular node in a particular POI in our simulation was assigned using ±20% variation. After finishing a particular activity, nodes moved to the next activity based on their interest score choosing a random speed. Different variations of the Random-waypoint mobility model [51] were used to generate movement inside a particular POI as the nature of the activity determines the way tourists move in that particular POI. For example, for 'Fishing' a node would randomly choose a particular spot inside the 'Fishing' region and stay at that spot for a random amount of time before moving to another. Such characteristics resemble fishing in real life where people start fishing at a particular spot and then either stay at that spot for the whole duration or change if fishes are not available there. Walking is a more dynamic activity where people start from some point and then continue along the track to reach another point. Sometimes random pauses are found during the transition from one point to another. Within a particular POI, a node's movement speed was limited between 0$\sim$5 km/hour. These properties were captured while generating synthetic mobility data in a java program. Arrival of all nodes and their movement in the way stated above were stored in a mobility file which was then taken as input to NS3 network simulator to dictate a node's movement during simulation. The left hand side of Fig. 3 shows a map of Lakes Entrance where the POIs are marked with different colours representing different activities and the right hand side shows a snapshot of the mobility of a single node inside the 'Food and Shopping' POI generated using the Random-Waypoint mobility model. To simulate content sharing we used the NS3 network simulator. Nodes were initialized with random battery power where 100 units represented the full capacity. Periodic battery drain (0.07 unit/minute) and drain due to data transfers (8×10^{-8} unit/byte) were used as in [15]. Battery level was increased gradually for a random amount of time whenever nodes were in the charging state. Backup administrators are updated every 30 mins or at handover times, whichever occurs earlier.

We considered contents are of following media types: Text (40%, 512 KB), Photos (20%, 1MB), Audio

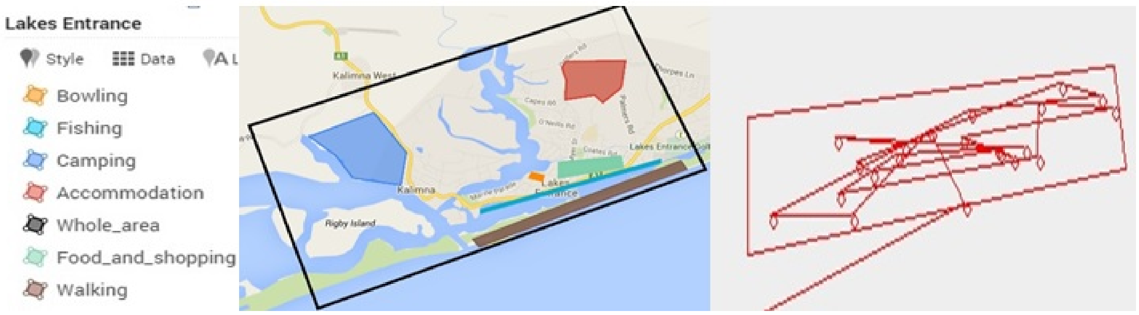


Fig. 3: Map of Lakes Entrance (left) and snapshot of a single node's movement inside the 'Food and Shopping' POI using Random-Waypoint mobility model [51] (right)

TABLE 1: Simulation parameter

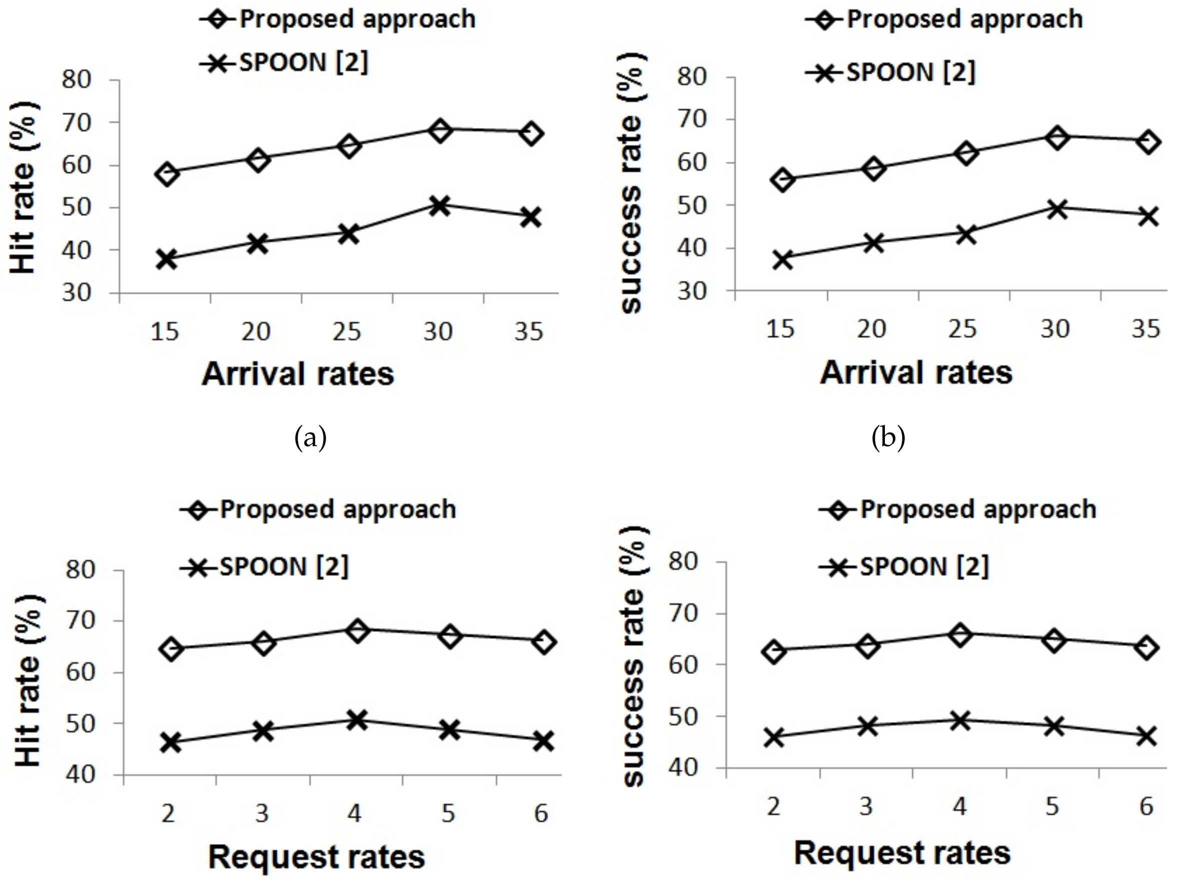
Simulation parameters	Value
Simulation area	10 Km X 5 Km
Number of nodes	150
Duration	3 days
Avg. node arrival	15-35 node/hr
Content request rate	2-6 request/hr
Request life time	6 hr
Range of IEEE 802.11n communication	150 m

(20%, 5MB [42]) and Video (20%, 6.5MB [52]), where the first number within the parenthesis represents the percentage of that type of content and the second number represents mean content size. We have used gamma distribution with shape Y=2.0 as in [3] for allocating the size of individual content. We considered nine categories of content of which six are related to activity based interest, namely 'Accommodation', 'Bowling', 'Camping', 'Fishing', 'Food and Shopping' and 'Walking', and three are related to content media types, namely 'Movies', 'Music' and 'News'. We considered 30 contents for each category totalling 270 contents for sharing in the network. Nodes were assigned with 20∼30 contents across different categories. Nodes generated content requests based on their interest. The request generation time followed exponential distribution as in [3], and individual requests were generated following a Zipf distribution as in [22]. In simulation, we have used IEEE 802.11n in ad hoc mode at 54 Mbps data rate operating at Channel#1 with 20 MHz width and the application installed in each node communicates in this channel for content sharing purpose. NS-3 already provisions MAC layer and physical layer modules for IEEE 802.11 standards. In MAC layer, we used contention based DCF scheme with RTS/CTS and 7-retry in case of a failed transmission. For modelling wireless channel in NS3, we employed constant speed propagation delay model and range propagation loss model. Some simulation parameters are listed in Table 1.

4.2 Simulation results

We use metrics, namely hit rate, delivery success rate, average delivery latency, average administrator lifetime, administrator expiration rate (due to low battery) and control message overhead to measure the performance of the proposed scheme. Hit rate and delivery success rate represent the percentage of the requests for which contents are successfully located and obtained by the requester, respectively. The average delivery latency is the average time it takes to obtain a single content after the request is generated. Average administrator lifetime depicts how long an administrator serves the group members on average. Administrator expiration rate shows the percentage of administrators died due to low battery. None of the existing approaches in literature are devised for content sharing solutions for tourist spots. In the absence of any comparable approach, we compared our approach with one of the recent ones called SPOON (2014) [2], which addresses content sharing in work-place type scenarios. We applied the characteristics and environment of a tourist spot on SPOON to capture its performance in this scenario. SPOON uses the stored content lists for calculating users' interest. It also considers degree centrality which requires a significant learning period for selecting administrators. Since degree centrality measurement from social network data is unavailable and such longer learning period cannot be accommodated in tourist spot like scenarios, we considered all nodes to have similar degree centrality (0.5) for SPOON. SPOON also uses interest oriented routing for message forwarding. The t-test comparing our scheme and SPOON in terms of hit and success rates at various arrival and request rates yielded p-values, $p \leq 2.48 \times 10^{-6}$ and $p \leq 9.42 \times 10^{-6}$ at 99% confidence level, validating their performance differences being statistically significant. Each simulation was run 20 times and the averaged results are presented in this section.

Fig. 4a and 4b show the improvements achieved with respect to hit and success rate when the proposed approach is employed. For example, hit and success rates achieved by the proposed approach are 68.48% and 66.13%, respectively compared to 50.86% and 49.46% in SPOON for an arrival rate of 30 nodes/hr. Improvements in our method over SPOON w.r.t. arrival rate ranges from 34-52% for hit rate and 33-48% for delivery success rate. Such improvements are achieved due to the proposed group formation and administrator selection mechanisms. The group formation mechanism in our approach considers interest fulfilment probability as well as the content availability and delivery probabilities, while the administrator selection ensures their longer availability to direct content requests to the appropriate content holders. Results show that the hit and success rates increase with an increment of node arrival rate. The reason being, higher arrival rate means more nodes are available in the simulation area which, in turn, increases the hit and success rates in both approaches. However, when the arrival rate exceeds certain value (i.e., 30 nodes/hr for the given settings), both rates start to decrease slightly because more nodes generate more requests which cause congestions. Existing approaches proposed for work-place type scenarios, where regular movement pattern and meeting frequency are established, reported hit and success rates within the range of 75-80% [2], [29]. In that respect, our method attains acceptably good performance in tourist



(a) (b)

(c) (d)

Fig. 4: (a) Hit rate and (b) success rate for different node arrival rates keeping average per node content request rate fixed at 4 request/hr; (c) hit rate and (d) success rate for different request rates keeping average node arrival rate fixed at 30 nodes/hr

spots where movement patterns are irregular and social ties are mostly non-existent.

Results presented in Fig. 4c and 4d show hit and success rates with varying content request rate. Both hit and success rates increase with increasing request rate. The reason is that with an increasing request rate, the probability of generating similar requests by nodes in the same neighbourhood becomes higher, and hence a requester can obtain contents from one of the neighbours who has already requested and obtained those contents. Both rates start to slightly decrease when request rate exceeds certain limit (4 request/hr), because more requests generate more congestion in the network. Our method attains acceptable hit (66.27%) and success (63.78%) rates even when the request rate is very high (6 request/hr). Our method's improvement over SPOON ranges from 32-40% suggesting that it is more suitable in situations where the number of visitors are very high during holiday seasons (e.g., Christmas vacation) generating more requests.

The average delivery latency is presented in Fig. 5a where latency values for successful deliveries

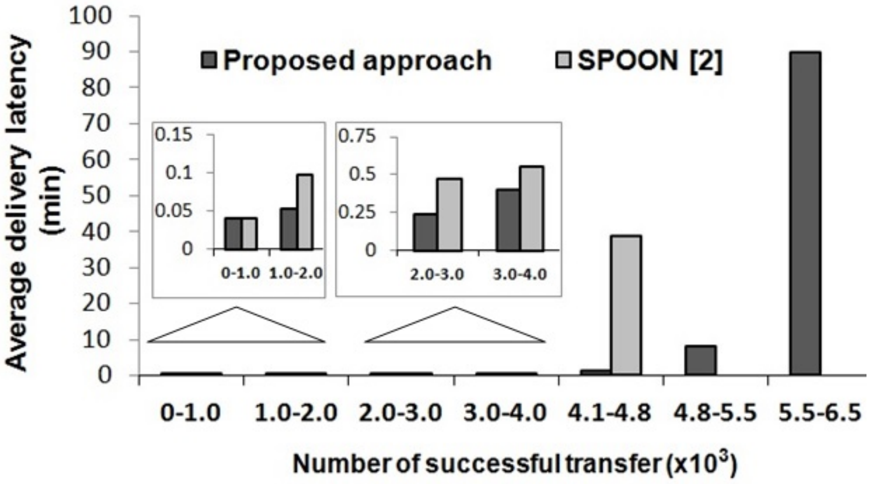
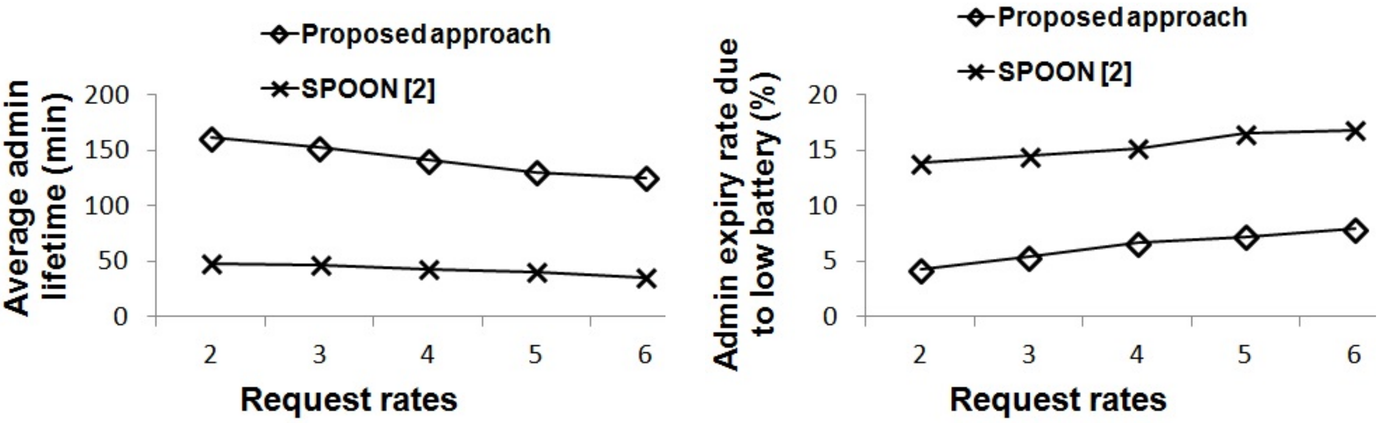


Fig. 5: Average delivery latency

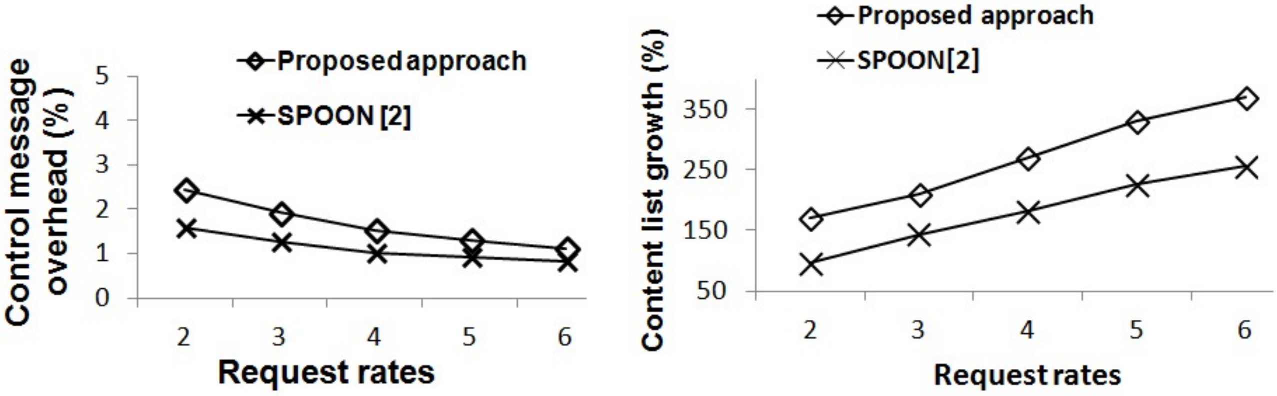


(a)

(b)

Fig. 6: (a) Average lifetime for administrators for different request rates, and (b) administrator expiry rate due to low battery

are sorted in ascending order and averaged over each 1000 deliveries for convenience of presentation. Note that only for this given simulation, we considered 10,000 identical content requests across both the approaches so that a fair comparison can be made. Both our proposed and SPOON approaches complete up to 4×10^3 successful transfers within a short time (0.40 mins for our approach vs. 0.55 mins for SPOON). Afterwards (up to 4.8×10^3), SPOON shows significantly higher delay than our proposed approach (SPOON 38.65 mins vs. proposed 1.16 mins). Results also show that SPOON could only deliver up to 4.8×10^3 successful transfers for the given settings whereas our approach could deliver 6.5×10^3 transfer with a relatively longer delay (89 mins) after 5.5×10^3 requests. In some cases, the content holders are located multiple hops away from the requester, and it requires longer time for request propagation and



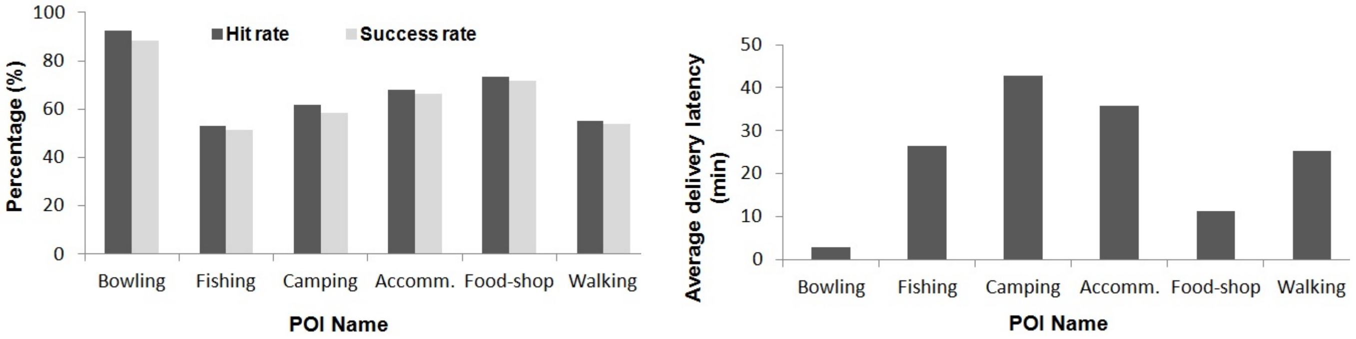
(a) (b)

Fig. 7: (a) Control message overhead, and (b) content list growth

delivery of contents. This suggests that our approach is capable of managing a high number of requests and delivering contents even if the delay is longer for some requests where SPOON fails altogether. Such long delays are acceptable considering the environments under investigation (i.e., tourist spots) where people usually stay for longer time (e.g., few hours to couple of days) and delivery is more important than delay.

As articulated in Section 3.1, our proposed approach selects administrators based on the stay probability, connectivity, device configuration, and remaining battery lifetime. The impact of such administrator selection policy is exhibited in Fig. 6a and 6b. It shows that the proposed approach results in much higher average lifetime for an administrator than SPOON (161 mins vs. 48 mins at a request rate of 2 requests/hr). Better results are obtained due to the fact that stay probability is used as one of the selection criteria in our approach. Figure 6b shows the expiry rate of the administrators due to low battery. The rate is much higher for SPOON since it does not consider remaining battery lifetime for administrator selection.

Control messages were used for various aspects of group formation, group updates, administrator selection and content request propagation. Figure 7a shows the percentage of message overhead (total control message size/total successfully delivered content size) for the proposed and SPOON approaches. The figure shows that our proposed approach generated slightly more control message overhead than SPOON (2.45% vs. 1.59% for request rate of 2 requests/hr). We used a neighbourhood table to calculate connectivity value during administrator selection and various stay probability measures to calculate content availability and delivery probabilities which contributed to increased overhead. Considering much higher hit and success rates achieved by our proposed approach, slightly higher control message overhead is a trade-off that brings benefit for content sharing in the given scenario. Moreover, at higher request rate the aggregated size of all successfully delivered contents also increases and our proposed method shows almost similar overhead as SPOON.



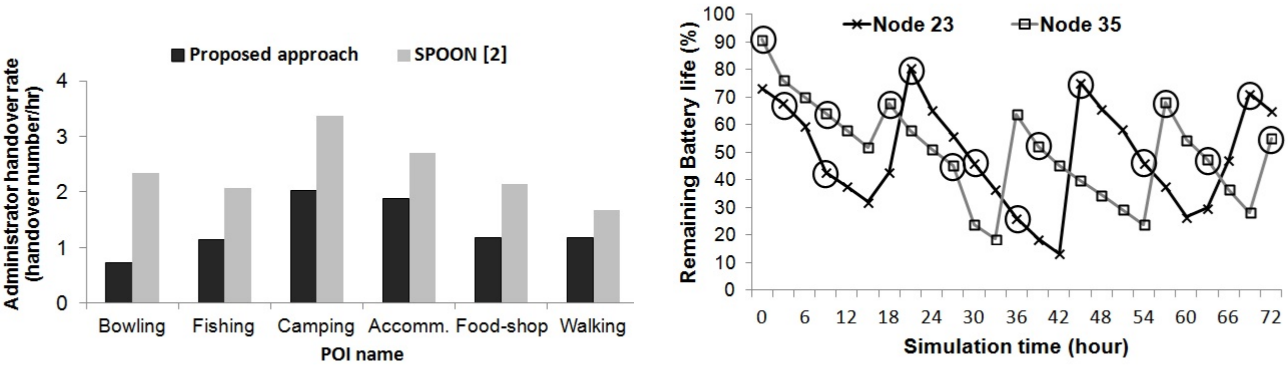
(a) (b)

Fig. 8: (a) Hit rate and Success rate in different POIs, and (b) delivery latency in different POIs

Figure 7b shows the content list growth which is the average increment of the percentage of contents after sharing at the end of simulation. Growth increases with an increasing content request rate for both the proposed and SPOON approaches. Results show that the growth rate is much higher in our approach as more contents are successfully received.

Nodes followed different mobility patterns in different POIs because of the inherent nature of the activities available in those regions. We investigated the performance in different POIs to assess the impact of such different mobility patterns on our proposed approach. Fig. 8a suggests that hit and success rates are high in ‘Accommodation’, ‘Bowling’ and ‘Food and Shopping’ regions. In ‘Accommodation’ nodes have low mobility as they enter the area mostly to sleep at night, and hence are more static, therefore we achieved high hit and success rates. In case of ‘Bowling’, although nodes move from one point to another, the movement is restricted inside an indoor environment of relatively smaller area and nodes come into frequent contact of each other facilitating message and content delivery. In case of ‘Food and Shopping’, we considered comparatively larger area than ‘Bowling’, and nodes moved inside the area following the well-established random-waypoint mobility model with considerably lower speed and longer pause times to demonstrate users’ movement around the shops and inside restaurants. As a result nodes remained within the communication range for longer time and generated higher hit and success rates. ‘Fishing’ and ‘Walking’ yield comparatively lower hit and success rates. In ‘Fishing’ the mobility is very low (nodes pick-up a point and stays there for a random time ranging from 0.5-2 hrs) suggesting infrequent contacts of nodes that yield lower hit and success rates. In comparison, ‘Walking’ suffers due to highly dynamic nature of nodes moving along the walking trail. Since all nodes move with random pause time, it gives them fewer opportunities to communicate and exchange messages.

Fig. 8b shows the average delivery latency in different POIs. The figure suggests that ‘Bowling’ attains lowest latency as nodes remain within the communication range more often as stated before. On the other hand, in ‘Accommodation’ and ‘Camping’ a more static nature of the nodes in this areas leads to infrequent



(a) (b)

Fig. 9: (a) Administrator handover rate in different POIs, and (b) remaining battery life for administrator. The steep energy rise shows charging of the devices. The circle shows the point in simulation when nodes were selected as administrator.

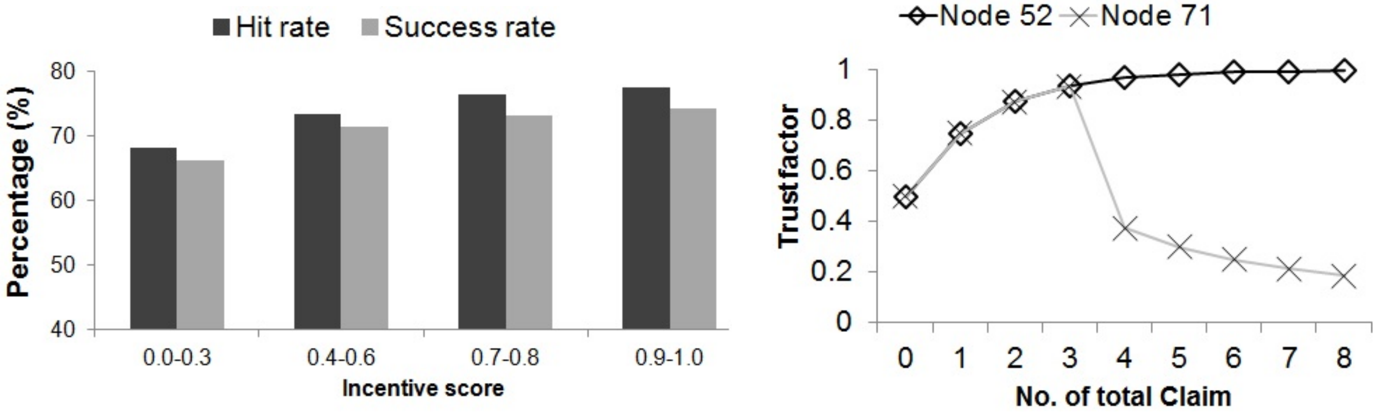
contacts, causing longer delay. Note that for 85% of the delivery, the latency is much low similar to the first part of Fig. 5a (up to 5.5×10^3 successful delivery) and as explained in related discussion in pg. 30. The purpose of this figure is to show the variation of latency for different POIs.

Figure 9a shows hourly administrator handover rate in different POIs calculated over the simulation duration. It shows that the rate is much higher for the SPOON approach compared to the proposed approach as the former does not consider the stay probability for selecting administrators. 'Camping' and 'Accommodation' need higher administrator handover rate than the other POIs as they have larger area and hence more groups exist.

Figure 9b shows the remaining battery lifetime for two of the nodes (Node 23 and 35) which were selected as administrators for most number of times (8 times) in various POIs among all nodes. The figure suggests that although the nodes were selected as administrators for many times they never ran out of battery, due to the consideration of energy issues (Section 3.1) in the selection process that selects a node as administrator only when it is suited.

Figure 10a shows the impact of the incentive scheme adopted in the proposed approach to encourage participation. For this particular figure, we considered the Food and shopping POI only and observed hit and success rates for the requests made by nodes whose incentive scores fell within the bands shown. The figure shows that a node with higher incentive score experiences better service for its requests in return to the service it provides to the group. For example, nodes with high (0.9∼1.0) incentive score experienced greater hit and success rates than those experienced by nodes with low (0.0∼ 0.3) incentive score (77.51% and 74.34% vs 68.27% and 66.23%). The reason is that nodes with higher incentive score receive higher priority from the administrator and forwarders for locating and delivering of requested contents.

As described in Section 3.5.2, we used a trust factor to identify misbehaving nodes and excluded them



(a) (b)

Fig. 10: (a) Impact of incentive score, and (b) Change of trust factor in 'Food and Shopping' POI. In (b) 0th claim indicates initial scenario where every node has same trust factor (Section 3.5.2)

from the pool of candidates for administrator selection. Figure 10b shows how the trust factor for two nodes (Node 52 and 71) varied within the 'Food and Shopping' POI with respect to their service claims. The figure shows that Node 52 continued to behave well and always passed the hurdle to be a candidate for the administrator (Eq. (1)). On the other hand, Node 71 initially made honest claims and its trust factor increased as long as its claims were true (up to the third claim). Afterwards, it started misbehaving and its trust factor continually decreased to a very low value as it made false claims and thereby were prevented from being a candidate for administrator selection.

Our method also includes a scheme where a node might relay messages of another group in return of similar service it gets from the members of that group. Disabling this feature in simulation revealed its impact showing slight reduction in hit and success rates (2∼3%). For example, at an arrival rate of 30 nodes/hr, the drop in hit and success rates are (1.88%) and (2.64%), respectively. It shows that some benefits are attainable when nodes relay messages for another group. Our future implementation will also allocate non-overlapping channels to adjacent groups and investigate its impact on our results.

5 Conclusion

Decentralized content sharing approaches offer a free-of-cost platform for sharing contents. Most of the decentralized approaches utilize predictable movement patterns of users and their existing social relationships for content sharing solutions, and hence applicable to share contents in workplace-like scenarios alone. However, in places like tourist spots or camping sites people exhibit irregular movement patterns and meet others with whom they most likely do not have any previous social relationships. Our method considers those scenarios and presents an activity and content interest based score calculation to facilitate content sharing. While interest similarity is the primary basis for group formation, factors like interest fulfilment probability, content availability and delivery probabilities are also taken into consideration for a node to join a group. We formulated an optimization problem that allows a node to join the group that maximizes its benefit in terms of those factors. The administrator of the group is selected through a competitive process on the basis of the candidate node's probability of stay within the group, its connectivity with neighbouring nodes, its trustworthiness and capability to serve the group during its stay. For performance evaluation, we applied the characteristics and environment of a tourist spot to a recent content sharing method called SPOON [2] which was developed for sharing contents in work-place type scenarios, and compared the results. We used a popular tourist spot in Victoria, Australia as the test location for the simulation. Results obtained from the simulation suggest that our approach outperforms SPOON with respect to hit rate, success rate, delivery latency and longer administrator lifetime, and hence is suitable for sharing contents in tourist spots. Results also show that our incentive scheme rewards nodes that provide service to other group members by prioritizing their requests that yields better sharing experience for them. Overall, our approach attains acceptable level of performance considering the constraints we have in tourist spot like scenarios. Future research will extend the current work considering separate lists for sharing and consumption by users and having administrator level exchange of interest lists in such case.

REFERENCES

[1] http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694l.

[2] 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.

[3] K. Thilakarathna, H. Petander, J. Mestre, and A. Seneviratne, “Mobitribe: Cost efficient distributed user generated content sharing on smartphones,” *IEEE Mobile Computing*, vol. 13, no. 9, pp. 1–14, 2013.

[4] M. Taghizadeh and S. Biswas, “Community based cooperative content caching in social networks,” *ACM MobiHoc*, pp. 257–262, 2013.

[5] http://www.amta.org.au/articles/6000-locations-nominated-for-Mobile-Black-Spot-Program-database.

[6] http://arstechnica.com/business/2012/05/bandwidth-explosion-as-internet-use-soars-can-bottlenecks-be-averted/.

[7] K. Yadav, V. Naik, and A. Singh, “Mobishare: cloud-enabled opportunistic content sharing among mobile peers,” *IIIT-D Technical Report*, vol. IIITD-TR-2012-009, 2012.

[8] R. Johari, N. Gupta, and S. Aneja, “Cacbr: context aware community based routing for intermittently connected network,” *ACM MSWiM*, pp. 137–140, 2013.

[9] R. Lubke, D. Schuster, and A. Schill, “Mobilisgroups: Location-based group formation in mobile social networks,” *IEEE PerCom*, pp. 502–507, 2011.

[10] F. Li and J. Wu, “Mops: Providing content-based service in disruption-tolerant networks,” *IEEE ICDCS 2009*, pp. 526–533, 2009.

[11] C. Boldrini, M. Conti, and A. Passarella, “Contentplace: social-aware data dissemination in opp. networks,” *ACM MSWiM*, pp. 203–210, 2008.

[12] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay-tolerant networks,” *IEEE Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[13] E. Talipov, Y. Chon, and H. Cha, "Content sharing over smartphone-based delay-tolerant networks," *IEEE Mobile Computing*, vol. 12, no. 3, pp. 581–595, 2013.

[14] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *IEEE Mobile Computing*, vol. 8, no. 5, pp. 606–621, 2009.

[15] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Content sharing among visitors with irregular movement patterns in visiting hotspots," *IEEE NCA*, pp. 230–234, 2015.

[16] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Content exchange among mobile tourists using users' interest and place-centric activities," *IEEE ICICS*, pp. 1–5, 2015.

[17] H. Zhou, J. Chen, J. Fan, Y. Du, and S. Das, "Consub: incentive-based content subscribing in selfish opportunistic mobile networks," *IEEE Selected Areas in Communication*, vol. 31, no. 9, pp. 669–679, 2013.

[18] E. G. Boix, A. L. Carreton, C. Scholliers, T. Van Cutsem, W. De Meuter, and T. D'Hondt, "Flocks: enabling dynamic group interactions in mobile social networking applications," *ACM SAC*, pp. 425–432, 2011.

[19] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun, "Geocommunity-based broadcasting for data dissemination in mobile social networks," *IEEE Parallel and Distributed Systems*, vol. 24, no. 4, pp. 734–743, 2013.

[20] Z. Li, C. Wang, S. Yang, C. Jiang, and X. Li, "Lass: Local-activity and social-similarity based data forwarding in mobile social networks," *IEEE Parallel and Distributed Systems*, vol. 26, no. 1, pp. 734–743, 2014.

[21] Z. Wang, X. Zhou, D. Zhang, Z. Yu, and D. Zhang, "Socker: Enhancing face-to-face social interaction based on community creation in opportunistic mobile social networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 755–783, 2014.

[22] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, "Supporting cooperative caching in disruption tolerant networks," *IEEE ICDCS*, pp. 151–161, 2011.

[23] J. Leguay, T. Friedman, and V. Conan, "Dtn routing in a mobility pattern space," *ACM SIGCOMM on DTN*, pp. 276–283, 2005.

[24] B. Burns, O. Brock, and B. N. Levine, "Mora routing and capacity building in disruption-tolerant networks," *Elsevier Ad hoc networks*, vol. 6, no. 4, pp. 600–620, 2008.

[25] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: An efficient routing in disruption-tolerant networks," *ACM MobiHoc*, pp. 95–104, 2009.

[26] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss, "Pi: A practical incentive protocol for delay tolerant networks," *IEEE Wireless Communication*, vol. 9, no. 4, pp. 1483–1493, 2010.

[27] B. B. Chen and M. C. Chan, "Mobicent: a credit-based incentive system for disruption tolerant network," *IEEE INFOCOM*, pp. 1–9, 2010.

[28] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "Smart: A secure multilayer credit-based incentive scheme for delay-tolerant networks," *IEEE Vehicular Technology*, vol. 58, no. 8, pp. 4628–4639, 2009.

[29] A. Krifa, C. Barakat, and T. Spyropoulos, "Mobitrade: trading content in disruption tolerant networks," *ACM Challenged networks*, pp. 31–36, 2011.

[30] M. Salehi and A. Boukerche, "A comprehensive reputation system to improve the security of opportunistic routing protocols in wireless networks," *IEEE GLOBECOM*, pp. 1–6, 2015.

[31] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *ACM Mobile computing and networking*, pp. 255–265, 2000.

[32] T. Ghosh, N. Pissinou, and K. Makki, "Towards designing a trusted routing solution in mobile ad hoc networks," *Springer Mobile Networks and Applications*, vol. 10, no. 6, pp. 985–995, 2005.

[33] Z. Jia, X. Lin, S.-H. Tan, L. Li, and Y. Yang, "Public key distribution scheme for delay tolerant networks based on two-channel cryptography," *Elsevier Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 905–913, 2012.

[34] J.-H. Cho, R. Chen, and K. S. Chan, "Trust threshold based public key management in mobile ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 44, pp. 58–75, 2016.

[35] B. Mckercher and G. Lau, "Movement patterns of tourists within a destination," *Tourism Geographies*, vol. 10, no. 3, pp. 355–374, 2008.

[36] D. Orellana, A. K. Bregt, A. Ligtenberg, and M. Wachowicz, "Exploring visitor movement patterns in natural recreational areas," *Tourism management*, vol. 33, no. 3, pp. 672–682, 2012.

[37] J. C. Xia, P. Zeephongsekul, and D. Packer, "Spatial and temporal modelling of tourist movements using semi-markov processes," *Tourism Management*, vol. 32, no. 4, pp. 844–851, 2011.

[38] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," *IEEE INFOCOM*, vol. 3, pp. 1360–1369, 2001.

[39] H. A. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman)," *IEEE WoWMoM*, pp. 1–6, 2007.

[40] http://www.tra.gov.au/research/regional.html.

[41] http://www.abs.gov.au/AUSSTATS/abs@.nsf/viewcontent?readform&view=ProductsbyTopic&Action=expandwithheader&Num=1.

[42] L. McNamara, C. Mascolo, and L. Capra, "Media sharing based on colocation prediction in urban transport," *ACM MobiCom*, pp. 58–69, 2008.

[43] E. Harjula, O. Kassinen, and M. Ylianttila, "Energy consumption model for mobile devices in 3g and wlan networks," *IEEE CCNC*, pp. 532–537, 2012.

[44] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," *Mining text data*, pp. 415–463, 2012.

[45] M. R. Hassan, G. Karmakar, and J. Kamruzzaman, "Reputation and user requirement based price modeling for dynamic spectrum access," *IEEE Mobile Computing*, vol. 13, no. 9, pp. 2128–2140, 2014.

[46] N. Christin, A. S. Weigend, and J. Chuang, "Content availability, pollution and poisoning in file sharing peer-to-peer networks," *ACM EC*, pp. 68–77, 2005.

[47] R. Panta, R. Jana, F. Cheng, Y. Chen, and V. Vaishampayan, "Phoenix: Storage using an autonomous mobile infrastructure," *IEEE Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1863–1873, 2013.

[48] Y. Lai, Z. Chen, W. Wu, and T. Ma, "Multipleresolution content sharing in mobile opportunistic networks," *Wireless Communications and Mobile Computing*, 2014.

[49] E. Nordstrm, C. Rohner, and P. Gunningberg, "Haggle: Opportunistic mobile content sharing using search," *Computer Communications*, vol. 48, pp. 121–132, 2014.

[50] 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.

[51] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

[52] X. Che, B. Ip, and L. Lin, "A survey of current youtube video characteristics," *IEEE MultiMedia*, vol. 22, no. 2, pp. 56–63, 2015.