MONASH University

Content Sharing using Smart Mobile Devices in Irregular Meeting Places

Shahriar Kaisar

BSc(*Hons*), MSc

A thesis submitted for the degree of *Doctor of Philosophy* at Monash University in 2017

Faculty of Information Technology

Main Supervisor:	A/Prof Joarder Kamruzzaman
Associate Supervisor:	Dr Gour Karmakar
Associate Supervisor:	A/Prof Iqbal Gondal
Honorary Supervisor:	Professor Susan McKemmish

Copyright Notice

Abstract

Decentralized content sharing (DCS) is emerging as a suitable platform for smart mobile device users to generate and share contents without the use of any fixed infrastructure, and thereby does not add to the Internet traffic which, in its current state, is approaching bottleneck in its capacity. Existing DCS approaches consider spatio-temporal regularity in human movement patterns and pre-existing social relationships for the content sharing schemes to work. However, such predictable movement patterns and social relationship information are not available in irregular meeting places such as tourist spots where people visit only for a short period of time, demonstrate spontaneous movement and usually meet strangers. No works exist in the literature that deal with content sharing in such types of environment. Facilitating decentralized content sharing in irregular meeting places will enhance the tourism experience of visitors as many such places lack Internet coverage. In addition, the shared contents will also be useful for promoting events and spreading rapid information during natural disasters.

The main focus of this thesis is to develop a decentralized content sharing technique, specifically targeting the above-mentioned irregular meeting places that will allow visitors to share contents among themselves without any Internet connection. In this regard, this thesis presents a number of novel strategies to facilitate decentralized content sharing in irregular meeting places. A unique group formation mechanism is presented which is based on users' interest scores and stay probability in each point-of-interest (POI) within the meeting areas, as well as on the availability and delivery probabilities of contents in the group. A new administrator selection policy is formulated by taking into account a node's probability of stay in the POI, connectivity with other nodes, available computing and energy resources to serve the group, and its trustworthiness as perceived by other nodes. Delivering contents within a tolerable delay is significantly difficult in DCS due to intermittent network connectivity. To address this issue, a novel utility based message forwarding technique is devised to select a forwarder node based on co-location stay time, connectivity and available resources. An innovative non-monetary incentive mechanism is proposed that awards nodes for sharing and forwarding contents. In addition, a lightweight trust management technique is designed to identify misbehaving nodes. In DCS, content availability and delivery service can be further improved by replicating contents. In this regard, a novel dynamic content distribution scheme is developed which employs economic modelling of demand and supply to proactively replicate contents in strategic positions nearby the potential requesters. Finally, incorporation of all these components forms a suitable platform for employing DCS in irregular meeting places. Extensive simulation has been performed using a popular tourist spot in Victoria, Australia to assess the performance of the developed methods. Simulation results confirm that the developed methods attain high hit and delivery success rate and low latency; comparable to those proposed for work-place type scenarios with routine movement pattern and pre-existing relationships.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: Shair ______________________

Print Name: Shahriar Kaisar

Date: 19 December, 2017

Acknowledgements

First and foremost, I would like to express my earnest gratitude to Almighty Allah, the most merciful, for showering endless blessings upon me and giving me the physical, mental and intellectual strengths for successfully undertaking this research.

I am forever grateful to my parents who have always provided me with endless support, motivation and encouragement to follow my dreams. Many thanks to my siblings for their unconditional love.

I want to express my sincerest indebtedness and heartfelt gratitude to my main supervisor A/Prof. Joarder Kamruzzman and associate supervisor Dr. Gour Karmakar for their invaluable contribution, continuous support, constant guidance, insightful criticism, intuitive suggestions and endless patience. They have always motivated me to cross my limit and inspired me to look at the positive side, even when things went wrong. This research would not have been successfully completed without their relentless involvement. I am very thankful to A/Prof. Iqbal Gondal for his valuable advice and support. I am grateful to Prof. Sue McKemmish, A/Prof. Shyh Wei Teng and Dr. Dengsheng Zhang for their helpful comments and commendable support.

I am forever grateful to Monash University for providing me with financial support in the form of a scholarship to complete this work. I would like to thank Federation University Australia for their support in various forms. I am thankful to all the staff members at the Gippsland campus, especially Ms. Louise Hrynyszyn and Ms. Linda Butler, for their academic support. I convey my thanks to staff at Monash Cloud Cluster (MCC), Monash Graduate Education (MGE) and the Faculty of IT.

I am very fortunate and thankful to have some amazing people as my colleagues and friends: Rajkumar Das, Ali Naqi, Fasahat Siddiqui, Ahammed Sherief, Ahsan Chowdhury, Fabian Hasna, Shakif Azam, Mamun Abdullah, Abdullahi Chowdhury, Joarder Kamal, Priyabrata Karmakar, Mahboob E Khoda, Manzurul Islam, Rakib Hassan, Abdullah Baky and Ashrifa Afroz.

Finally, special thanks to my wife Rehana Parveen for her infinite love, unconditional support, continuous encouragement and relentless patience. I really appreciate your effort in sharing this journey with me and sacrificing the most along the way.

Dedicated to my parents, and to my wife who sacrificed the most for this research.

Abbreviations

AP	Access Point
CAOR	Community-aware Opportunistic Routing
CAR	Context-aware Adaptive Routing
CD	Content Distribution
CDN	Content Delivery Network
CS	Content Sharing
DCS	Decentralized Content Sharing
DSIP	Decentralized Sharing in Irregular meeting Places
DTN	Delay Tolerant Network
E-DSIP	Enhanced Decentralized Sharing in Irregular meeting Places
IM	Instant Messenger
MANET	Mobile Ad-hoc Network
MSN	Mobile Social Network
NCL	Network Central Location
OMSN	Opportunistic Mobile Social Network
OppNet	Opportunistic Network
OSN	Online Social Network
Pdf	Probability density function
PDI	Passive Document Index
PER	Predict-and-Relay
POI	Point-of-interest
QoS	Quality of Service
ROI	Region of Interest
SCF	Store-Carry-and-Forward
SS	Social Selfishness
TA	Trusted Authority
TFT	Tit-for-Tat

Abbreviations (continued)

TTL	Time-to-Live
TTP	Trusted Third Party
UBF	Utility Based message Forwarding
VB	Virtual Bank

Notations

κ	A candidate node for administrator selection
P^{κ}	Stay probability of κ
C^{κ}	Connectivity of κ
Θ	Time window for administrator selection cycle
$T^{\kappa}_{c,f}$	Total Stay time of κ at POI 'c'
$T^{\kappa}_{c,s}$	Time already spent at 'c' by κ
μ_s	Mean stay time
σ_s	Standard deviation of stay time
$erf(.)$	Error function
$\mathbb{H}_n$	Set of n-hop neighbours
$\mathbb{G}$	Set of members of group G
E^{κ}_f	Energy factor of κ
E^{κ}_{min}	Minimum energy required by κ
E^{κ}_r	Remaining energy of κ
E^{κ}_T	Full energy of κ
E^{κ}_u	Energy consumption of κ for regular use
E_{ad}	Energy required to serve as an administrator
α	A content
E_{α}	Energy required to receive and reply a request for content α
λ_r	Request arrival rate
Λ^{κ}	Device configuration of κ
Λ_{min}	Minimum device configuration required
ϑ_{max}	Maximum processing resource available in the market
ϑ^{κ}	Total processing resource at κ
ϑ^{κ}_u	Processing resource currently in use at κ

Notations (continued)

ι_{max}	Maximum memory resource available in the market
ι^{κ}	Total memory resource at κ
ι_u^{κ}	Memory resource currently in use at κ
G	A content sharing group
k	A node
$\mathfrak{g}_G^k$	Gain factor for joining group G calculated by k
ω_G^k	Interest fulfilment probability of k for G
Π_G^k	Content availability probability of k for G
ψ_G^k	Content delivery probability of k for G
$\overline{d}_G^k$	Average delay experienced by group members of G
d_{max}^k	Maximum tolerable delay of k
h_G^k	Hop-distance of k to the administrator of G
n_{max}	Maximum allowable hop-distance to the administrator
i	A content category
Υ_i	Interest score for i
Υ_i^p	Interest score for i from personal profile
Υ_i^r	Interest score for i from online recommendation
Υ_i^e	Interest score for i from location-centric experience
Υ_n^k	Interest score of k for n-th category
$\Upsilon_{i,G}$	Maximum interest score for i in group G
$\gamma_{i,w}^u$	Number of recent posts about i within the time window w by recommender i
ε_i^u	Reliability of recommender u about i
φ_{rec}	Weight for the recent posts
χ_i^u	Length of exposure of recommender u for i
σ_i	Standard deviation for i

Notations (continued)

$\mathbf{U^k}$	Interest matrix of k
I_n^k	n-th category of interest for k
$\mathbf{U_G}$	Interest matrix of group G
$\Gamma_{i,G}^k$	Category-wise Difference between interest of node k and group G for i
$\overline{P_G}$	Average stay probability of members of group G
$\overline{P_{i,G}}$	Average stay probability of members of group G interested in i
$\mathbb{G}_i$	Set of users interested in i in group G
P_i^v	Stay probability of v interested in i
N^k	Total number of categories k is interested in
d_G	Delay observed in group G by administrator
d^v	Delay observed by v
ζ_G^k	Content list factor perceived by k for G
F_G^k	Neighbor factor of k for G
P_G^k	Stay probability of members of G calculated by k
$U_\alpha^{k,d}$	Utility value of k for carrying α to requester d
B_a^k	Buffer space available at k
B_α	Buffer space required to store α
$L^{k,d}$	Co-location stay probability of k and d
TTL_α	The lifetime of the request for content α
Ω^k	Incentive score of node k
Φ^k	Trust factor of k
D_α	Demand for content α
$D_{i,\alpha}$	Demand for α from users interested in category i
$D_{i',\alpha}$	Demand for α from users currently not interested in category i
$\mathbb{G}_{i,\alpha'}$	Set of group members who are interested in i and do not have α
x	A demanding node

Notations (continued)

$\mathscr{I}_\alpha^x$	Interest factor of node x
R_θ^x	Content request generation probability of x within θ
Δ_α^x	Personal preference of x for α
Ψ_α	Popularity of α
ϖ	Scale of the influencing event related to α
β	Event shape factor
φ^x	Affinity of x towards a new interest category
S_α	Supply for content α
$\mathbb{G}_{i,\alpha}$	Set of members of group G who have α
y	A supplier node
C_α^y	Coverage of node y for α
$\mathbb{H}_1^{x,y}$	Set of potential requester within one-hop distance of y
z	A potential distributor node
δ_α^z	Utility value of z for holding a replicated copy of α
ϕ_α^z	Demand coverage of z for α
τ^z	Media access non-contention factor of z
η^u	Probability of node u being busy
η_d^u	Probability of u being busy in delivering content
η_g^u	Probability of u being busy in receiving content
η_r^u	Probability of u being busy in relaying content
ζ^u	List of contents owned by node u
ρ	Multi-hop delivery factor

Publications

Journal

- S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, Decentralized content sharing among tourists in visiting hotspots, Elsevier Journal of Network and Computer Applications, Volume 79, Pages 25-40 , **[CORE: A, Impact factor: 3.5]**.

Conference

- S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, Content sharing among visitors with irregular movement patterns in visiting hotspots, in IEEE 14th International Symposium on Network Computing and Applications (NCA), Sept 28-30, 2015, Cambridge, MA, USA, Pages 230-234, [CORE: A].

- S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, Content exchange among mobile tourists using users interest and place-centric activities, in IEEE 10th International Conference on Information, Communications and Signal Processing (ICICS), December 2-4, 2015, Singapore, Pages 1-5, [CORE: B]. **Received best paper award**.

- S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, Carry me if you can: A utility based forwarding scheme for content sharing in tourist destinations, in IEEE 22nd Asia-Pacific Conference on Communication(APCC), August 25-27, 2016, Indonesia, Pages 261-267, [CORE: B].

- S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, Dynamic content distribution for decentralized sharing in tourist spots using demand and supply, 13th International Wireless Communications and Mobile Computing Conference (IWCMC), June 26-30, 2017, Spain, Pages 2121-2126, [CORE: B].

Publications (continued)

In Progress

- S. Kaisar, J. Kamruzzaman and G. Karmakar, A Dynamic Content Distribution Scheme for Decentralized Sharing in Tourist Hotspots, IEEE Transaction on Mobile Computing, **[CORE: A*, Impact factor: 3.82]**.
- S. Kaisar, J. Kamruzzaman and G. Karmakar, Decentralized Content Sharing: A Survey, ACM Computing Surveys, **[CORE: A*, Impact factor: 6.74]**.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *The only thing that will redeem mankind is cooperation.* —-Bertrand Russell

1.1 Background Information

Mass penetration and worldwide adoption of smart mobile devices (e.g., smartphones and tablets) are the most prominent features observed in recent times. The number of smartphone users across the globe is expected to surpass 2.32 billion by 2017 and over one-third of the world's population by 2018 [1]. During this same timeframe, the number of tablet users is expected to reach 1.23 billion [2]. The driving forces behind such immense popularity include, but are not limited to, the ease and the convenience of use, rich configuration, elegant look, seamless network connectivity and the availability of a range of useful applications. The growth of smart mobile devices has also prompted research in a number of directions, such as (i) content sharing where users generate and share contents, (ii) activity and gesture recognition where the embedded sensors in smartphones are used to identify the current context of a user, (iii) location based services where the locations of the devices are used to provide particular services, (iv) crowdsensing where devices collaboratively sense the environment, (v) indoor localization where the location of a user is accurately detected in an indoor environment, and (vi) identifying usage characteristics. Adoption of smart mobile devices has also changed the way people communicate and use their phones.

A closer look at smartphone usage reveals some interesting statistics, which show that the amount of time users spend on smartphones varies across different application categories. Social networking, entertainment and games are the most popular applications and account for 39%, 15% and 12% of usage time, respectively [3]. In addition, News (4%) and search (6%) also contribute to a significant amount of usage time [3]. A recent report indicates that more than eight of every ten Internet users are expected to use their smart devices for web access in 2017 and 80% of social media time is spent on a mobile device [4]. The same report also suggests that 1.23 billion Facebook users and 82% of Twitter users are accessing these platforms from their mobile devices. Both of these platforms, as well as other social networking platforms, are primarily being used to stay connected and to share contents, such as images, videos, news items or any other information. Statistics about popular video sharing platforms, such as YouTube, also indicate that 70% of its total traffic is generated from mobile devices [5]. The above statistics demonstrate that the evolution of smart mobile devices has significantly impacted the way people access and share contents, and engagement in such activities dominates smartphone usage across all demographics. Therefore, it is of utmost importance to study the way people share contents and consequently improve the content delivery service for better sharing experiences. All these have motivated us to further investigate content sharing using smart mobile devices in this thesis.

Content Sharing (CS) applications enable smart device users to generate and share various types of contents, such as text, music, videos, pictures and news items, with anyone in the world. Such engagements are motivated by the ease of sharing life events and expressing opinions within seconds. The shared contents help in fulfilling information needs, providing instant recommendations, promoting businesses and circulating educational materials. In addition, content sharing is useful during natural disasters or on the battlefield. Figure 1.1 [6] shows a schematic diagram of the content sharing process where smart device users obtain different contents from the server through mobile Internet connection or from other nearby users through ad-hoc networks. Because of technological evolution and ease of use, content sharing approaches have also evolved. A few approaches are available in the existing literature for content sharing and these are discussed in the following section.

1.2 Approaches for Content Sharing

Content sharing approaches using smart mobile devices affect several things, such as Internet traffic, content delivery success rate, and delay in receiving content. Con-

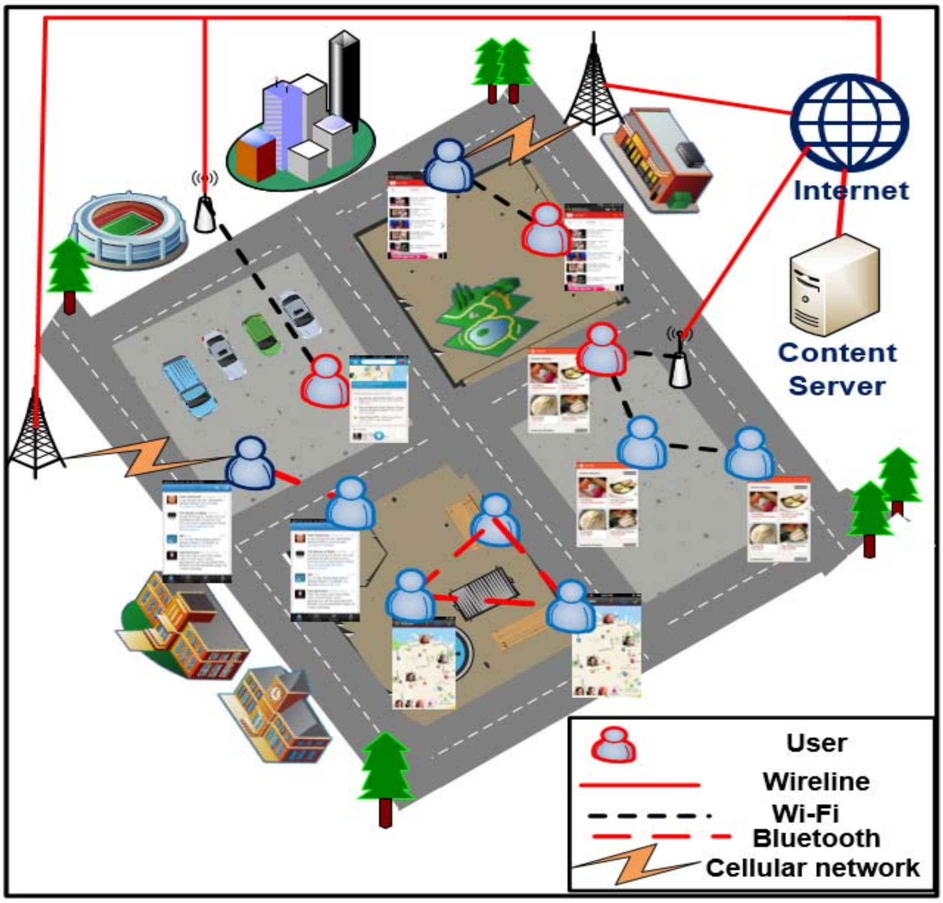

Figure 1.1: Content sharing using smart mobile devices adapted from [6]

sidering the effects of these things, the sharing process can use three types of approach: (i) centralized, (ii) decentralized and (iii) hybrid approach, which are discussed below. This is followed by a discussion on why it is important to study decentralized content sharing further, which is the focus of this thesis.

1.2.1 Centralized Content Sharing

The centralized content sharing approach is an extension of the traditional web based architecture. In this case, users generally use a mobile application or mobile web browser from their smart devices to access a centralized server to share or obtain contents. Applications for accessing the most popular social networking or content sharing websites, such as Facebook[1], Twitter[2] and YouTube[3], are readily available from the app stores [7–9]. Information related to a particular user, including user profile and interests, group affiliations, uploaded contents, and preferences, are uploaded to a centralized server. Figure 1.2 shows a schematic representation of the centralized architecture where devices are connected to a server, namely 'Content Server', through Wi-Fi or a cellular network to obtain content. In this case, user *A* uploads a content to the server, which is later requested by user *B*. The content server stores a list of contents and their associated metadata. When the server receives any request, it matches the request against stored metadata to identify matching content. Afterwards, the server delivers the matching content to the requester. The entire communication process uses an Internet connection.

In this approach, the availability of a central server makes it easier to deal with issues, such as group management, content delivery, and privacy. This type of archi-

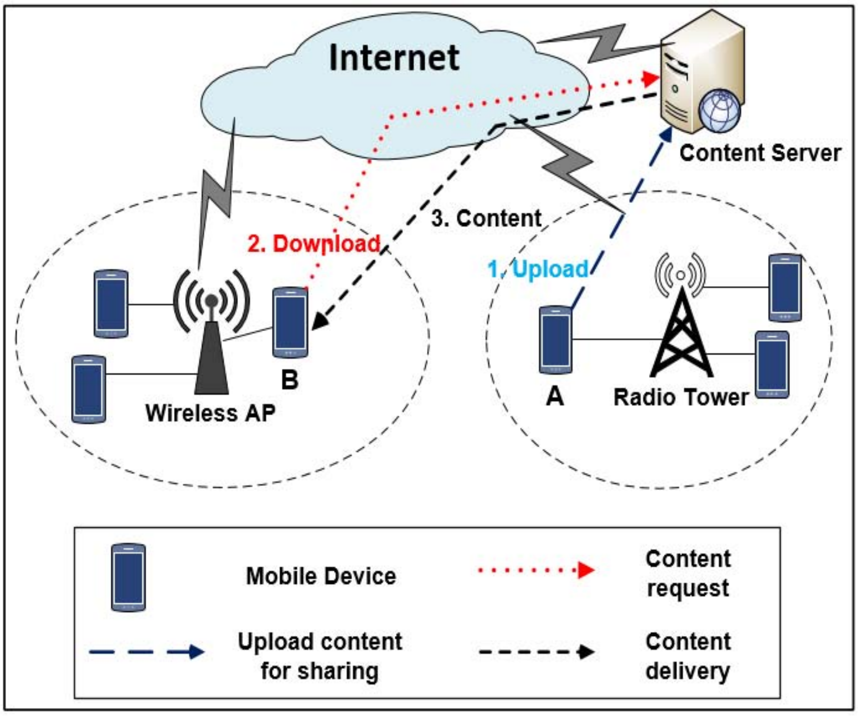

Figure 1.2: Centralized content sharing approach

[1] http://www.facebook.com/
[2] http://www.twitter.com/
[3] http://www.youtube.com/

tecture provides a bridge between the social community and the physical location of a user. However, this approach causes a bottleneck or crashes the server when the number of users is very high [10]. This affects the quality and delivery of contents. In addition, it creates single point failure, as everything is stored in a centralized fashion. The traditional centralized content sharing web-sites also store a huge amount of personal information to provide a more user-centric service. However, from the users' perspective, they might feel uncomfortable to share their personal data with a service provider. Ownership of the content is also an issue for such architecture. This is because everything is uploaded to the server and users may feel that their personal control over the content has decreased. Moreover, they have to obey the rules and regulations imposed by the service provider. Since this approach requires Internet through a service provider, it involves a certain amount of cost and the sharing process only works in situations where an Internet service is available. To address these important issues of a centralized approach, such as scalability, serviceability and cost, decentralized approaches have been evolved.

1.2.2 Decentralized Content Sharing

The amount of Internet traffic produced through centralized sharing approaches is significantly high and they are also subject to a single point failure. In addition, they do not utilize the advanced networking capabilities of smart mobile devices, such as Wi-Fi and Bluetooth communication, which make it possible to communicate with other devices within a certain proximity. Note that the Wi-Fi technology available in smart mobile devices can work in two different modes, namely infrastructure and ad-hoc mode. The infrastructure mode is used for connecting to the Internet through a Wi-Fi Access Point (AP), while the ad-hoc mode is used for communicating with nearby devices and does not require an Internet connection. To take advantage of such Bluetooth and Wi-Fi communication technologies, a decentralized sharing approach is developed.

The Decentralized Content Sharing (DCS) approach is the most promising networking solution because this is more scalable and fault tolerant. However, employing this approach is very challenging as there is no centralized authority to provide contents or store user information. In this approach, mobile devices perform content sharing among themselves using Bluetooth or Wi-Fi (in ad-hoc or peer-to-peer mode) communication without the requirement of an Internet connection. Therefore, this does not incur any extra data transfer or communication costs.

Figure 1.3 shows a schematic representation of the decentralized content sharing approach where users within proximity have formed a content sharing group $G1$. User C is the administrator of this group and responsible for maintaining the content list of all group members. User A generates a request and forwards it towards administrator C. After receiving the request, C determines that user D has a matching content and forwards the request to D. Whenever user D receives this request, it delivers the matching content through its neighbors. For the sake of simplicity, this figure shows a dedicated path from the requester (A) to the content holder (D). However, in real life, such end-to-end connected paths are mostly unavailable because of node (i.e., user) mobility. This type of network is also called an Opportunistic Mobile Social Network (OMSN) and is considered a paradigm of the Delay Tolerant Network (DTN), which basically allows a longer delivery period for any content by holding the content for a certain amount of time if there is no connection between a content holder and a requester. Since an Internet connection is not needed for this approach, it does not introduce any

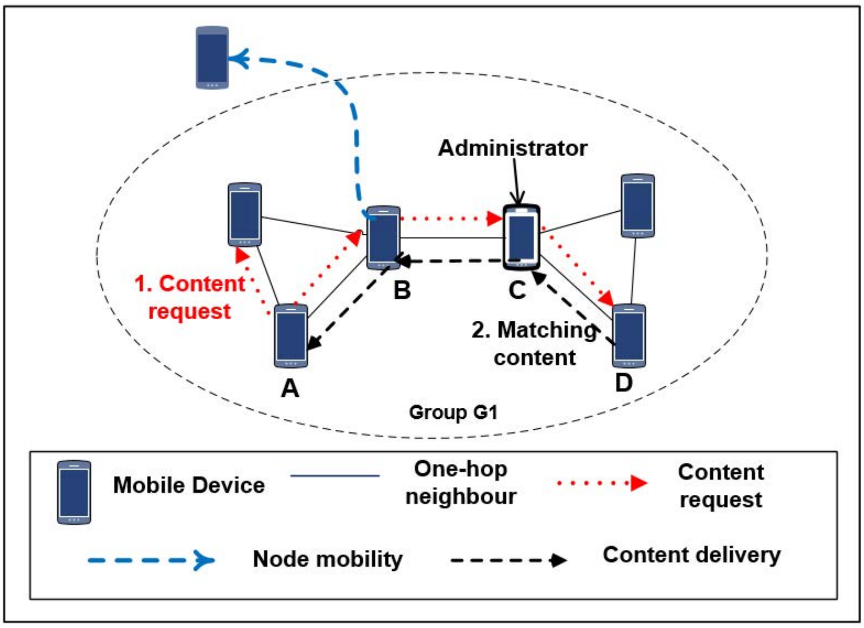

Figure 1.3: Decentralized content sharing approach

network traffic. However, the main problems of a decentralized approach are unreliability and less availability, because of node mobility. To address these problems and have the benefits of the scalability and robustness of a decentralized approach, a hybrid approach has been introduced.

1.2.3 Hybrid Content Sharing

The hybrid content sharing approach combines the benefits of both centralized and decentralized sharing techniques to deliver contents. In this approach, a centralized server tries to ensure that nodes within the communication range of Bluetooth or Wi-Fi can communicate locally rather than always requesting a server [11–13]. This reduces the server's load. In addition, this approach also increases the availability and reliability of the system as the server can always deliver contents in the absence of any suitable co-located user. Figure 1.4 shows a schematic representation of the hybrid content sharing approach. In this instance, both user A and B request the same content. The content server determines that they are co-located and within the communication distance of each other and hence divides the content into two parts and sends 'Part 1' to user A and 'Part 2' to user B. Afterwards, these two users mutually exchange these parts and merge them to obtain the whole content. In this manner, hybrid approach reduces network traffic. However, both centralized and hybrid approaches require an Internet connection and introduce some major problems, which are highlighted below.

Why study decentralized content sharing?

The centralized and hybrid content sharing approaches provide the most reliable and effective services since they use wired or wireless Internet/Intranet connections and can deliver contents more reliably. However, the problems with such approaches are:

(i) *Constant requirement of Internet connection:* Mobile devices can connect to the Internet using a cellular network or Wi-Fi APs. However, poor cellular coverage exists in many parts of the world and Wi-Fi APs are mostly available in indoor environments. For example, the Australian Government published a report in 2013 which shows that only 25% of landmass has mobile coverage and many of these areas suffer from poor

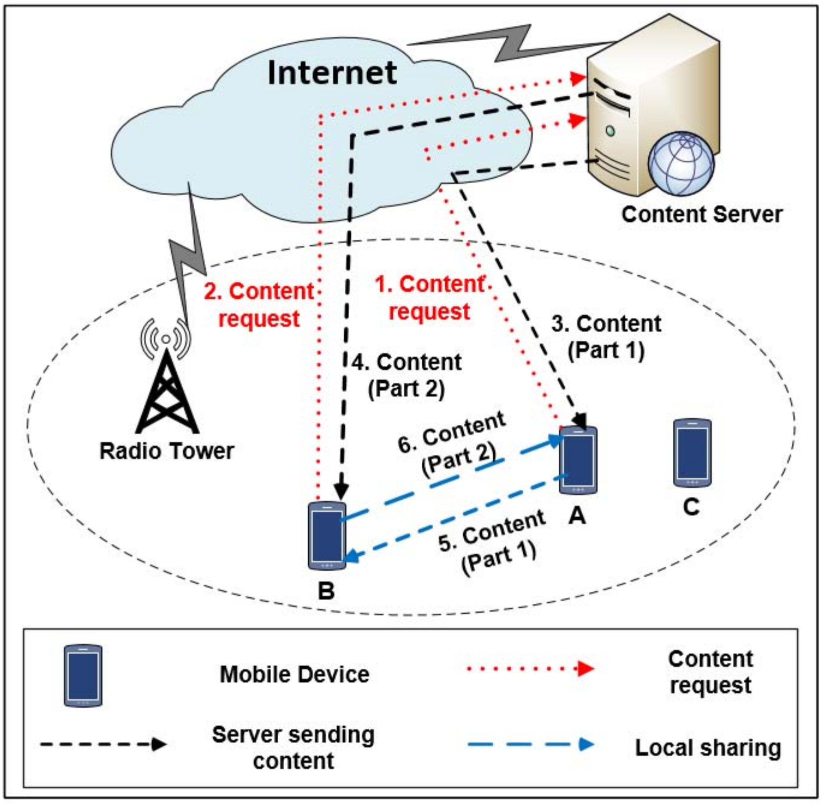

Figure 1.4: Hybrid content sharing approach

connectivity [14]. Although the situation has improved through installation of additional infrastructure, in 2016, information about black spots with inadequate mobile coverage was published by collecting data from the community members [15]. Figure 1.5 [15] shows the locations with inadequate mobile coverage across Australia, indicating that poor cellular coverage is still a major concern in most of the areas. A similar situation exists in many parts of the world. Therefore, Internet connection through a cellular network is not feasible in those places and consequently centralized and hybrid content sharing approaches are unavailable.

(ii) *Increment of network traffic:* A recent report suggests that Internet traffic has already surpassed 1.2 zettabytes per year, or 96 exabytes (nearly one billion gigabytes) per month in 2016 [16]. The same report also indicates that by 2021, network traffic generated from smartphones (33%) will exceed the same from PCs (25%) and Content Delivery Networks (CDN) will contribute 71% of the total IP traffic for delivering contents. Current network infrastructure is already approaching a bottleneck and will be

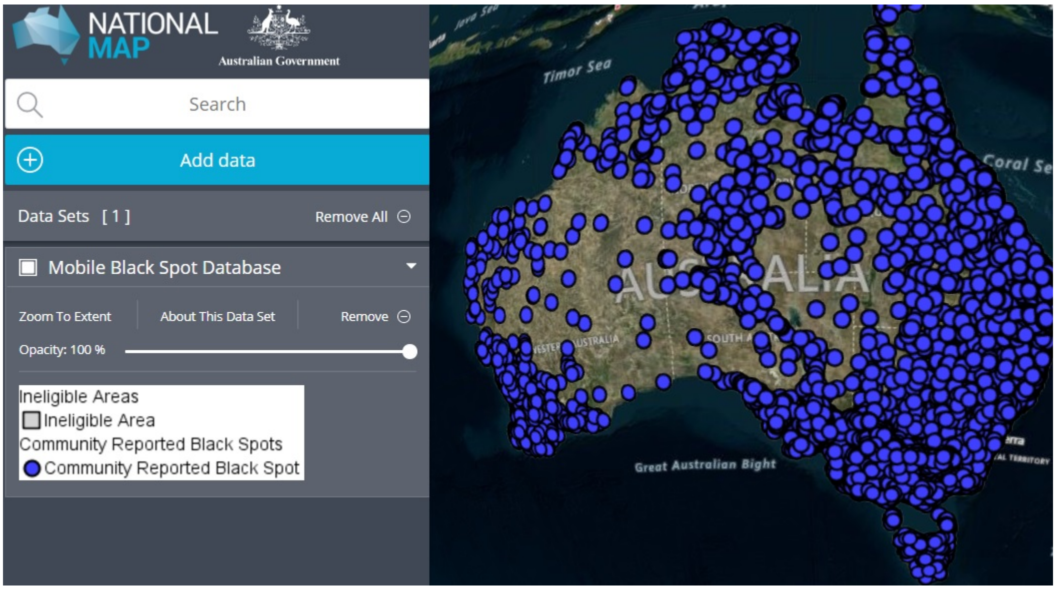

Figure 1.5: Reported locations (blue dots) with inadequate mobile coverage in 2016 in Australia [15]

insufficient to cope with such a huge volume of traffic [17].

(iii) *High cost of Internet connection if accessed through cellular network:* Telstra is the biggest telecommunication service provider in Australia. Their website indicates that checking social networking websites uses 5-10 MB each time, each YouTube user consumes 15-30 MB daily, uploading a photo consumes around 1.5 MB and finally watching a HD TV series of around 22 minutes utilizes around 750 MB of data [18]. Telstra charges $49 for 3 GB of data which can be consumed by downloading just a single movie.

(iv) *Natural disaster:* During a natural disaster, such as earthquake, flood and tsunami, power outage occurs, which results in Internet unavailability in affected areas. Consequently, the centralized and hybrid content sharing approaches become ineffectual for spreading emergency information.

In contrast, decentralized content sharing provides a suitable and free-of-cost platform to enable smart mobile device users to share contents even in places without any Internet connection. Therefore, in this thesis, we aim to investigate the decentralized content sharing approach to advance it further, so that it can provide more reliable services for irregular meeting places (e.g., tourist spots or camping sites) where nodes are highly and unpredictably movable.

1.3 Motivation and Problem Statement

Decentralized content sharing reduces the amount of network traffic and can play a major role in cellular traffic offloading [19]. This will be particularly beneficial in urban scenarios where the network is overloaded, or in challenged networking scenarios where the infrastructure is inadequate to provide mobile coverage, or during a natural disaster because of the destruction of available networking facilities. In addition, the decentralized content sharing approaches can be incorporated into a number of applications, such as proximity marketing, proximal social networking, opportunistic sensing where devices can jointly sense the environment in an opportunistic manner, opportunistic mobile computing where co-located devices can perform computation intensive tasks collaboratively, social gaming where co-located users can participate in different games, and mobile cloud computing where devices can offer a temporary storage solution for others. Furthermore, research in this area is also expected to provide promising solutions for other areas, such Device-to-Device (D2D) communication, where the devices are expected to communicate with each other to perform collaborative tasks, and the smart city initiative. However, there exist a number of practical and critical issues, such as handling node movement patterns and coping with dynamic content demand. The research problems with reference to decentralized content sharing in irregular meeting places are highlighted in the following discussion.

Spontaneous movement and unfamiliar users: Facilitating decentralized content sharing is very challenging because of the dynamic nature of human movement patterns. The existing literature employs frequent encounters [20–22], temporal and spatial regularity of movement patterns [23–25], and social relationships among nodes [26–28] to handle different aspects of the sharing process. These approaches are appropriate for *work-place* type scenarios where people follow a routine schedule to visit home and work places, such as a university campus or office, and mostly meet family members, friends or colleagues. However, there are occasions when people visit places that are out of their regular routine and meet people with whom they do not have any social ties. We use 'irregular meeting place' to denote such situations in this thesis, for example, someone visiting a tourist spot or camping site, or meeting at a festival. *There has been no focus in the literature on facilitating content sharing in such scenarios*.

In irregular meeting places, people usually meet strangers and stay there for a short duration of time (e.g., a couple of hours to a few days). This short duration makes it almost impossible to extract meaningful relationship information or frequent encounters among nodes. In addition, adding further dynamic elements, such as the spontaneous movements of tourists, makes it even more challenging to predict movement. However, employing appropriate DCS approaches, yet to be developed, in irregular meeting places can enhance the visiting experience, as many such places lack Internet connectivity (e.g., the Grand Canyon National Park in the USA and the Sahara desert in Africa [29]). Internet traffic also rises sharply during influencing external events (e.g., major tournaments and festivals), and network infrastructure suffers to deliver satisfactory service [30, 31]. In that respect, DCS can aid in successful content delivery and reduce network load. The shared contents will also be helpful for entertainment or rapid information propagation during emergency situations. Content sharing in such irregular meeting places is highly challenging, mainly because of (i) unknown node distribution and contact patterns, (ii) spontaneous movements of visitors, and (iii) the lack of familiarity among visitors to allow the extraction of meaningful social relationship information or identify their mutual interests.

In DCS, nodes form groups (i.e., communities) based on some criteria such as mutual interest [32, 33] proximity [34, 35] or social relationship [36, 37]. Group members help each other by carrying requests or providing relevant contents. Each group has an administrator to perform tasks related to group management and directory service [33]. Finding appropriate metrics for group formation is very challenging in irregular meeting places as users from different backgrounds with varying interests gather together, and most of them are strangers. The existing literature employs social centrality metrics for administrator selection using frequent encounters among users [33, 38], which is also absent in irregular meeting places.

The formation of a group and selection of an administrator helps in forwarding a request to an appropriate content holder. However, delivering the content within a delay bounded time is still difficult since both the requester and the content holder might move (i.e., change location) during the transfer and nodes have very little or no idea about future encountering nodes to take a learned decision. To address this, a Store-Carry-and-Forward (SCF) technique [39] is employed where a node carries a message (or content) until it finds a suitable forwarder node or the lifetime of the message expires if the destination is unreachable. However, employing such SCF techniques in a tourist spot is very challenging because of spontaneous movements and unfamiliarity among nodes, which does not allow information from past encounters or movement patterns to be used. The limited energy available at nodes further complicates the task.

Selfish and misbehaving nodes: In DCS, nodes are assumed to be cooperative and willing to carry contents for other nodes. In a collaborative content sharing approach, such cooperation is desirable to provide a satisfactory service. Nonetheless, in reality, nodes may not contribute without a proper incentive, even though their resources allow participation [40–42]. In a tourist spot type scenario, this behavior would be more prominent, as nodes are mostly strangers and unlikely to cooperate without proper benefits. Therefore, an appropriate incentive scheme is needed to encourage node participation. Again, any node can demonstrate misbehavior to gain an unfair advantage. A malicious node can spread malware or a virus in the group. It is significantly important to identify such misbehaving nodes to reduce risk. However, detecting such misbehaving nodes is extremely challenging as most of the nodes have no idea about the previous record of other nodes. Spreading a blacklist is also not feasible as the duration of the visits is usually short. Therefore, implementation of a light-weight yet effective trust management scheme is quite demanding but necessary.

Dynamic demand and supply: The performance of DCS approaches can be improved by increasing content availability and reducing delivery latency through content replication [43–45]. The basic idea is to estimate future demand and proactively push matching contents near the prospective requesters. The demand for contents rises sharply in a tourist spot for higher numbers of visitors. For example, during school holidays, more teenagers are expected to visit a tourist spot and they consequently generate more requests as they are the main driving force for online social network or sharing contents. Again, the demand for contents also rises because of some influencing external event [46]. For example, the occurrence of a global sports event (e.g., Olympic games) is likely to generate more interest among people about receiving contents related to that event. The delivery success rate and latency suffer if the demand for content increases and the available content holder(s) leaves the area or has insufficient resources for delivery. Providing a satisfactory performance under such a high and dynamic demand with limited resources is a complex task and requires development of highly adaptive and efficient strategies considering dynamic demand and supply.

To summarize, the existing DCS approaches are effective for sharing contents only in regular life settings with predictable encounters and movement patterns. However, employing them in irregular meeting places will require altogether new methods because of the unique nature and limitations within which DCS has to operate in those scenarios. The scope of this thesis is to investigate the development of a decentralized content sharing technique suitable for irregular meeting places that can effectively address the above-mentioned research problems.

1.4 Research Objectives

To address the major research issues associated with decentralized content sharing in irregular meeting places presented in the previous section, the research objectives of this thesis are:

1. To investigate the development of a framework to facilitate on-demand content sharing among smart mobile device users in irregular meeting places using the information that is readily available or can be gathered on the fly.

2. To construct a robust and secured model to minimize resource consumption, reduce the impact of misbehaving nodes and encourage participation of nodes who are reliable and cooperative in providing services.

3. To estimate future and dynamic content demand and supply accurately, and then to devise a dynamic distribution method considering those approximated content demand and supply and medium access contention to handle a higher number of requests with improved delivery service.

1.5 Thesis Contributions

The previous section highlighted the primary research objectives of this thesis. To achieve these objectives, a number of original contributions are made in this dissertation. Figure 1.6 presents a schematic diagram to show the flow of the research conducted and the novel contributions achieved that are highlighted in different blocks. Block 1 represents novel strategies proposed for interest extraction, group management and administrator selection to develop a framework to facilitate decentralized sharing in tourist spot type scenarios to achieve Objective 1. This basic model is further improved by adding a new utility based forwarding method (Block 2), and an innovative incentive and trust management scheme (Block 3) to accomplish Objective 2. A final improvement of this model is achieved by adding a dynamic content distribution strategy which uses a unique dynamic demand and supply calculation method and a novel distribution scheme to fulfill Objective 3 (Block 4). Finally, the combination of these modules represents a complete decentralized content sharing scheme in irregular meeting places. The contributions of this thesis are briefly summarized below:

1. To create the basic framework for facilitating DCS in irregular meeting places, a new group formation and administrator selection method is devised. The proposed group formation method ensures that not only does a user have common interests with other group members, but also that the group has sufficient resources to offer an acceptable content delivery service. In this regard, factors like interest fulfillment probability, content availability and delivery probabilities are also taken into consideration. A joint optimization problem is formulated considering the above factors that allows a node to join the group that maximizes its benefit. In addition, the proposed administrator selection policy ensures that the administrators are capable of providing service for a longer amount of time and have enough resources to do so. Performance of this method has been analyzed in a popular tourist spot in Victoria, Australia through extensive simulation in network simulator NS3, predominantly using performance metrics, such as delivery success rate and latency. Simulation results show that our proposed approach successfully delivered 59.13% of the requested contents, in most cases within 3 seconds to 11 minutes. The primary idea of the basic framework was published in [47] and a further extended and improved version was published in [48]. (**Block 1 - objective 1**, Chapter 3)

2. The above-mentioned basic framework employs the Spray-and-Wait [49] mes-

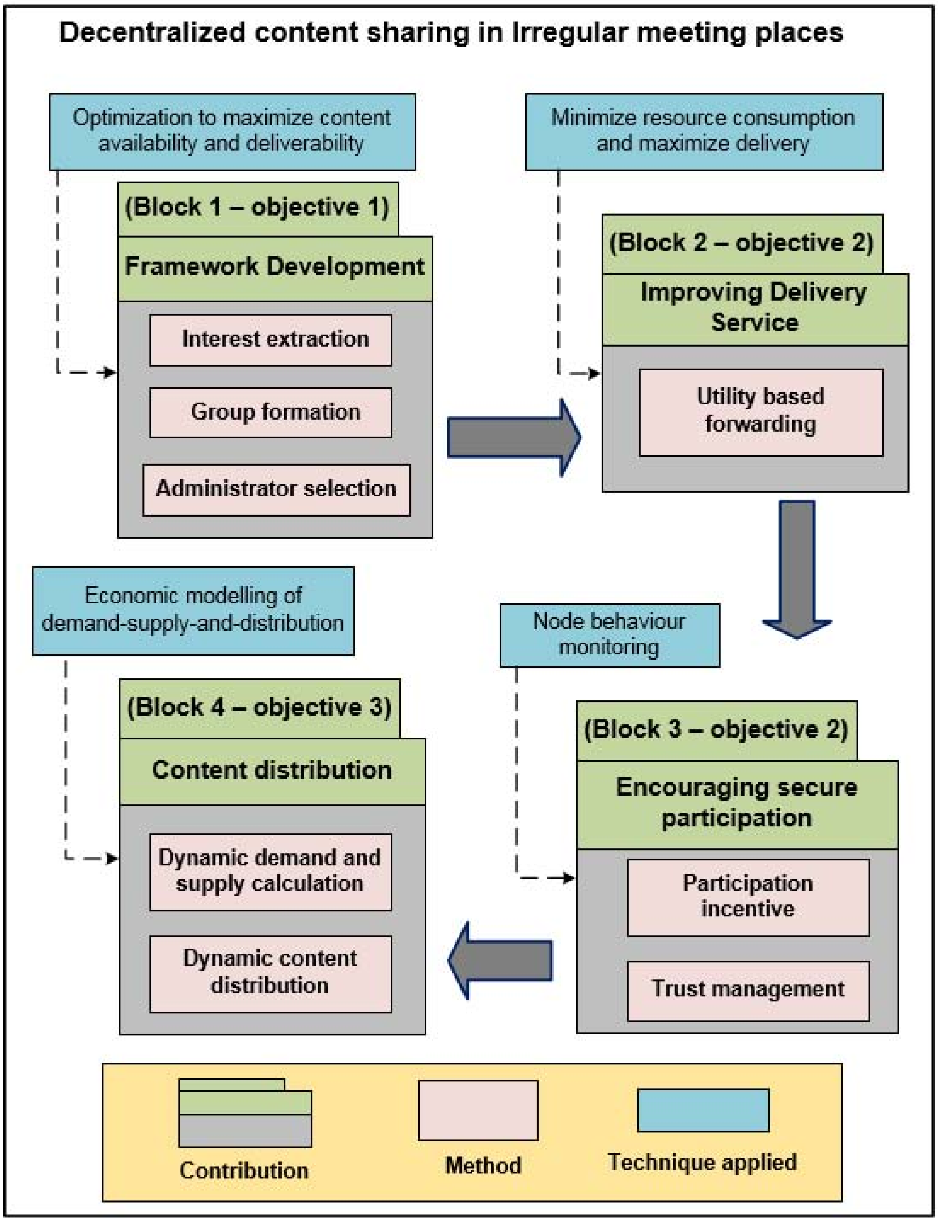

Figure 1.6: A schematic diagram presenting the novel contributions of the thesis

sage forwarding method which blindly forwards a specific number of copies of a message in the network, and thereby consumes unnecessary resources. To alleviate this issue, a novel utility based message forwarding technique is constructed which only uses a single copy of a message for content delivery to minimize resource consumption within the network. In this case, a maximization problem is formulated to identify the node that attains the highest utility value for carrying a content considering its opportunistic contact probability with the destination as well as connectivity within the network. Adoption of this forwarding method increased the content delivery success rate to 66.25%. This message forwarding technique is published in [50]. (**Block 2 - objective 2**, Chapter 4)

3. A unique incentive and trust management scheme is introduced to encourage node participation and identify misbehaving nodes. The incentive scheme provides non-monetary benefits in the form of priority processing to increase participation in the sharing process. To achieve this, a node's behavior is monitored where the administrator of a group tracks the content forwarding and sharing behavior of the group members to assign a reputation score, and nodes with higher reputation scores receive priority for obtaining service from others. Simulation results demonstrate that nodes with higher reputation scores (0.9-1.0) were able to achieve 25% higher delivery success rates compared to nodes with lower scores (0.0-0.3). To identify misbehaving nodes, the administrator assigns a trust score based on the claim made by a node about its participation in the delivery process. Incorporation of these schemes along with the previous modules increase the attractiveness and practical applicability of DCS in tourist attraction type scenarios and this is published in [51]. (**Block 3 - objective 3**, Chapter 4)

4. To improve content availability and delivery service, a novel dynamic content distribution scheme is formulated using the economic modeling of demand, supply, and distribution. This approach proactively replicates a content when its demand surpasses its available supply. In this regard, a new dynamic demand and supply calculation method is developed that identifies contents that are more likely to be requested and need to be replicated. The novelty of the distribution scheme lies in the selection of the distribution locations, which uses a joint optimization of demand coverage and medium access contention factor to identify nodes that are best suited to hold a replica. Incorporation of this scheme produces a 70.45% content delivery success rate, which is nearly comparable to those proposed for work-place type scenarios with regular movement patterns or existing social relationships [33, 52–54]. The basic idea of this distribution scheme is published in [55] and the complete work is currently in progress to be submitted as a journal

paper in IEEE Transaction on Mobile Computing. (**Block 4 - objective 4**, Chapter 5)

Finally, the integration of the above-mentioned modules presents an original work for facilitating decentralized content sharing in irregular meeting places.

1.6 Organization of the Thesis

This chapter has presented some background information and motivation of doing this research. A brief description of the thesis objectives and the thesis contributions is also provided. The rest of this thesis is organized as follows:

Chapter 2: Decentralized Content Sharing: Current Approaches and Major Issues

This chapter provides an in-depth review of the existing literature focusing on different aspects of decentralized content sharing approaches. Since decentralized content sharing is the focus of this thesis, techniques related to this approach are explored in this chapter. The major components and their related issues for employing DCS approaches are analyzed in this chapter to identify potential research gaps and major research challenges.

Chapter 3: Decentralized Content Sharing in Irregular Meeting Places

A basic framework for employing decentralized content sharing in tourist attractions is presented in this chapter. The framework addresses group formation and the administrator selection process for handling the sharing process. The discussion also includes a description of the simulation environment for measuring the performance of the proposed model. Finally, simulation results are presented to highlight the impact of the proposed model.

Chapter 4: Enhancing Content Delivery and Node Participation in DSIP

Chapter 4 presents some techniques for providing an efficient content delivery service. In order to enhance the performance of the proposed model in the previous chapter, a utility based message forwarding technique is presented which can provide a better content delivery service. In addition, an incentive mechanism is discussed that encourages node participation and at the same time helps in identifying misbehaving nodes. The chapter ends with an evaluation of the proposed model.

Chapter 5: A Content Distribution Scheme in E-DSIP Based on Dynamic Demand and Supply

In this chapter, a dynamic content distribution scheme is presented. Several aspects of the distribution scheme, such as capturing content demand, estimating content supply and identifying strategic positions for distributing the replicated content, are discussed in detail. The chapter ends with a comparison with existing approaches, as well as with the models mentioned in the previous chapters.

Chapter 6: Conclusion and Future Work

Finally, Chapter 6 concludes the thesis by highlighting the key findings and the contributions of this work. The limitations of this work and some scope for future engagement are also discussed.

Chapter 2

Decentralized Content Sharing: Current Approaches and Major Issues

As highlighted in the previous chapter, adoption and mass penetration of smart mobile devices have significantly impacted the way people socialize and share contents. An imminent shift from the traditional centralized and hybrid content sharing approaches is needed to reduce the reliance on Internet availability and limit the amount of network traffic. In order to facilitate this, recently a number of decentralized content sharing approaches have been proposed. Some of these approaches consider physical proximity while others use social profiles to enhance the sharing experience of the users. In addition to presenting existing decentralized approaches, this chapter investigates the major components of decentralized content sharing and their relevant issues. After reviewing the literature associated with decentralized approaches, a number of major research challenges have been identified that have set the direction of research for this dissertation. These are also presented in this chapter.

2.1 Decentralized Content Sharing Approaches

As highlighted in Chapter 1, the centralized and hybrid content sharing approaches require an Internet connection to be available for delivering contents. In the case of a decentralized approach, nodes (i.e., mobile devices) share contents among themselves through peer-to-peer connections without requiring an Internet connection. Since the focus of this thesis is to provide a content sharing solution for irregular meeting places where an Internet connection is mostly unavailable, decentralized approaches are the appropriate ones. Therefore, the discussion in this chapter is limited to these approaches.

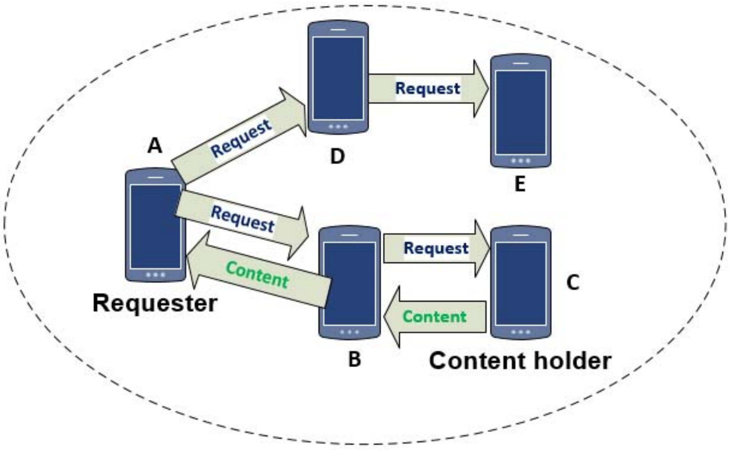

Figure 2.1: Decentralized content sharing without any grouping

The Decentralized content sharing approach is the most promising and cost effective networking solution as it is scalable and fault tolerant. However, employing this approach is also very challenging because there is no centralized authority to provide contents or store user information. In this approach, mobile devices perform content sharing among themselves using Bluetooth or Wi-Fi communication without the requirement of an Internet connection. Figure 2.1 shows a schematic representation of the decentralized content sharing approach. It shows that user **A** generates a content request and forwards the request to its neighbors (**B** and **D**). In turn, the neighbors also forward the request until it reaches a node that holds a matching content. When a content holder (i.e., node having a matching content for that request, user **C** in Fig. 2.1) receives the request, it delivers the content back to the requester. However, blindly forwarding requests to everyone unnecessarily consumes more resources. For example, forwarding a request for sports related content to a user who is not interested in sports at all, will not result in a successful delivery but still consume resources.

In order to address the above issue, a group-based decentralized content sharing approach is used where nodes are divided into groups based on their mutual interest, and content requests are forwarded to group members only. Figure 2.2 shows such a group-based decentralized content sharing approach. The figure shows that users are divided into group ***G1*** and ***G2*** based on their mutual interest and hence the members are overlapped physically as some of them have common interests. Each group (i.e., community) has a local administrator (e.g., Admin of ***G1*** and ***G2***) who performs tasks

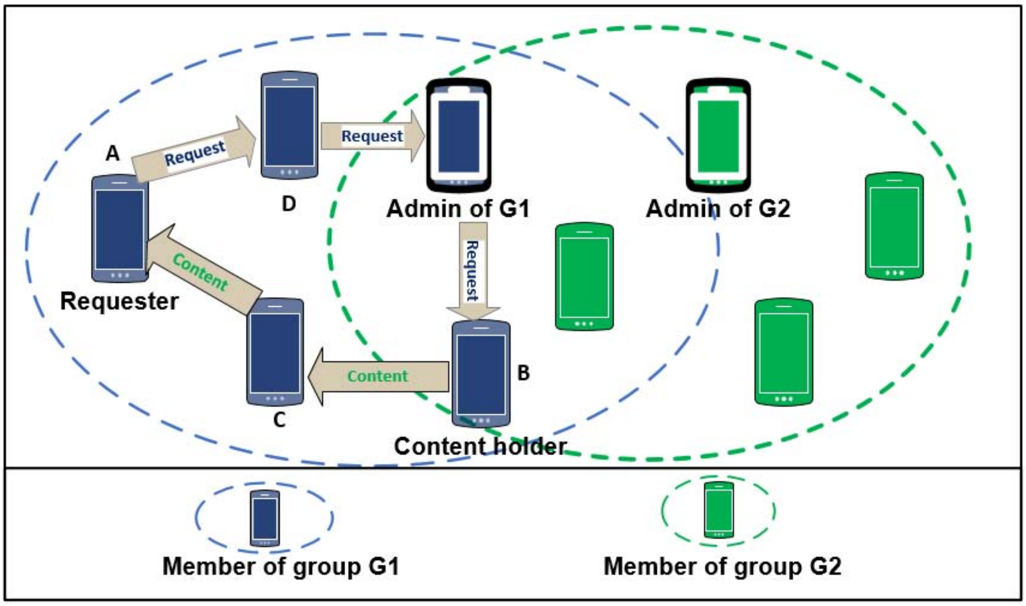

Figure 2.2: Group based decentralized content sharing

related to group management and provides a directory service. For example, the administrator can store the list of contents of the group members and direct the content request to an appropriate content holder. The figure shows that user **A** sends a request for accessing a content, which is forwarded to the content holder **B** via the administrator (Administrator of ***G1***). After receiving the request, the content holder delivers the content via other users, if the requester is outside of its communication range.

Media sharing in urban transport proposed by McNamara *et al.* [56], is one of the earlier works proposed for decentralized content sharing. It suggested that there exists regularity in the movement patterns of people commuting via the train services. For example, people who are going to work in the morning and/or returning from work in the evening tend to meet a common set of people while traveling and they are termed as *familiar strangers*. The authors proposed a content sharing protocol using a co-location prediction method that uses the history of earlier encounters to identify the source for copying a content. However, they only used single-hop communication, and assumed that often people would be within the communication range of each other to create a history of encounter, which might not be true when trains leave every 5-10 minutes at a busy station and have many compartments. This work also does not use any concept of grouping. In contrast, work in [36] proposed a group-

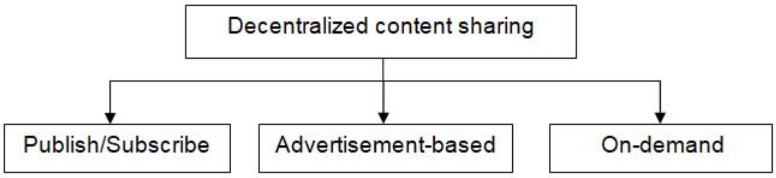

Figure 2.3: Data dissemination techniques in decentralized content sharing

based decentralized sharing approach where nodes are divided into groups based on long-term neighboring relationships. The history of previous encounters is used to determine such relationships between nodes.

Content sharing over a delay tolerant networks has also received considerable attention from researchers recently [25, 33]. The difference between the traditional DTN and content sharing over DTN is that the traditional DTN assumes that the source has a message to forward and it knows the identity of the destination, which is useful in situations such as wildlife monitoring or sending sensor data to a sink. In contrast, content sharing through DTN is a two-step procedure where the content must be located first before it can be delivered, which is very challenging due to the unavailability of a centralized server. Furthermore, both the requester and the content holder might change their positions in the meantime.

Content sharing using a decentralized approach employs a number of data dissemination procedures (Figure 2.3), which can be categorized as: publish/subscribe, advertisement based and on-demand approaches. These approaches are discussed below.

2.1.1 Publish-Subscribe

Publish-subscribe based decentralized content sharing approaches consider that mobile devices can play the role of both publisher and subscriber, and are capable of using multi-hop communication for delivering and receiving content. The publisher devices publish contents, such as movies, music, and news, on some channels and the subscribers subscribe to them. In this case, the interest of a subscriber is considered from a coarse level. For example, a user might subscribe to a channel publishing movie or music related content. The publisher nodes periodically publish some

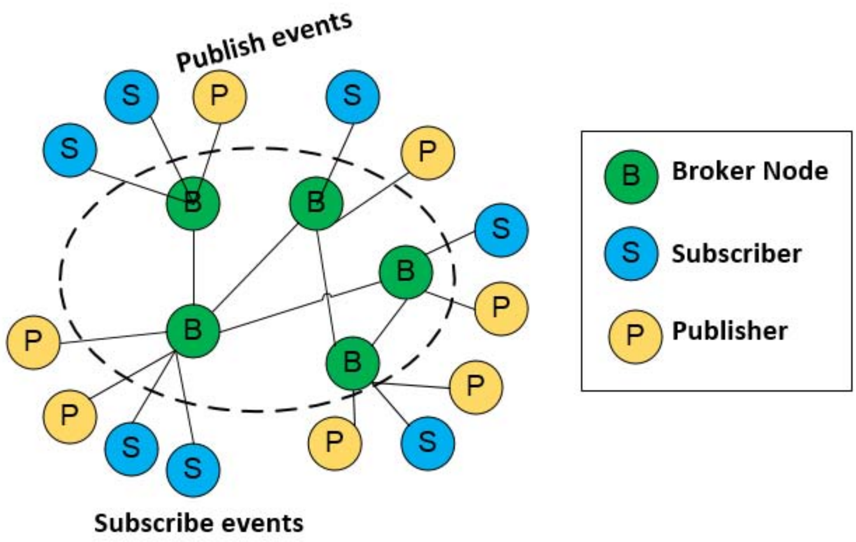

Figure 2.4: A distributed publish/subscribe system adapted from [57]

contents in their channels and try to disseminate those contents over the network. These approaches generally use the idea of having cooperative nodes in the network where everyone obtains content reflecting personal interest as well as contents that their neighbors are interested in. Whenever a subscriber node meets any publisher, it first copies contents reflecting personal interest and then copies the contents that might be of interest to its future neighbors, if meeting time and device memory permit. Such publish/subscribe based data dissemination has been addressed in [37, 57–59].

A distributed publish/subscribe system proposed in [57] is one of the earlier works to use such a data dissemination technique. Figure 2.4 [57] shows a schematic representation of the proposed approach. It shows that participating nodes can be divided into publisher, subscriber and broker nodes. Note that a single node can also play multiple roles for different contents. The publisher nodes publish different contents about different events and the subscriber nodes can mention their interests towards those events. Both the publisher and the subscriber inform about their published content and event of interest, respectively, to the broker nodes. After receiving information from the publishers and subscribers, the broker nodes try to match the published contents against the interests of the subscriber. If any match is found, the broker nodes transfer the contents to the subscriber. The authors considered that the participating nodes are part of different communities (i.e., groups) and the node that has the strongest connection (in terms of frequent encounters) with other members of the community is selected as a broker.

In contrast, Boldrini *et al.* [37] use a broker-less publish-subscribe system assuming that a single node can play the role of a publisher and subscriber for different contents and they can exchange contents whenever they meet each other. For example, if a publisher who publishes music-related contents meets a subscriber interested in music, the publisher can directly deliver the content without the presence of any broker. Zhou *et al.* [59] proposed a similar approach, where at the beginning, each node broadcasts the list of channels it is interested in, which is collected and stored by other nodes to get an overall view of the available channels in the network. Afterwards, encountering nodes exchange contents according to their own interest and also store some contents for their expected future neighbors.

Although publish/subscribe schemes are successful in delivering contents that match the interest at a coarse level, they are unable to meet user interest at a finer level, and hence might deliver unnecessary contents. For example, a user might be interested in 'rock' music; however, he might not be interested to listen to any song by a particular singer. In addition, publish/subscribe dissemination schemes do not address on-demand content delivery, where users are looking for a particular content (e.g., trailer of a particular movie) instead of content designated to a particular channel (e.g., a movie channel).

2.1.2 Advertisement Based

Advertisement-based content sharing approaches consider that nodes will generate some contents and later will try to disseminate those contents over the network. This type of dissemination does not require explicit user subscription. The generated contents can be some sort of advertisement or notifications. Such advertisements are helpful for proximity marketing where some nodes may advertise some promotional offers in stores/restaurants or warn about some natural disaster. Advertisement based decentralized content sharing approaches are employed in [32, 35, 60].

Lubke *et al.* [35] suggested that a user can create and advertise a group, based on his personal interest or some event, which is visible to other users within the proximity.

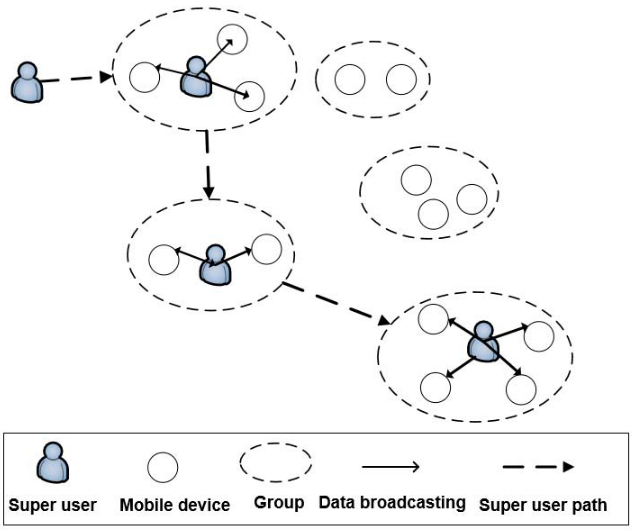

Figure 2.5: Geo-community based broadcast scheme adapted from [60]

Later, those having matching interest can join the group. After the event is completed, the group members can share their memories (e.g., pictures or videos) related to the event. Another interesting work on an advertisement based content sharing approach is proposed in [60]. Figure 2.5 [60] shows the schematic representation of this approach. This work addressed the problem of broadcasting some information to the members of different communities (i.e., groups) who are physically separated from each other. It assumed that a super user can visit all these communities to maximize the spread of information or minimize the amount of delay. In this regard, regularity in the movement patterns of the participants was used to model user mobility as a semi-markov process, and then a greedy adaptive routing algorithm was employed to determine the visiting sequence of such communities.

Although the advertisement based content sharing approaches are more appropriate when rapid propagation of some information is needed (e.g., during natural disasters) or for advertisement purposes, they are not suitable for obtaining on-demand content or distributing content according to user interest, as techniques based on advertisements cannot provide such facilities. In this case, a more sophisticated approach is required that considers the interest of the user before delivering any content.

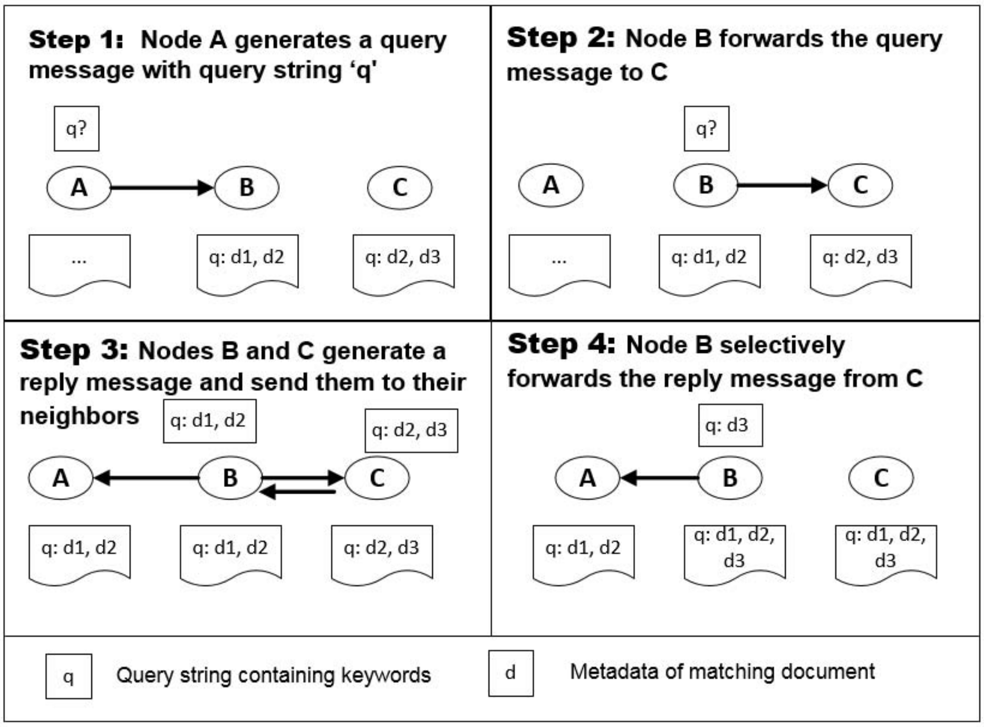

Figure 2.6: Passive document index (PDI) adapted from [61]

2.1.3 On-Demand

On-demand decentralized content sharing is a query-based data dissemination technique where a user generates a request for a particular content specifying its requirement. For example, a request can be for "latest news on the US election 2017". Keywords are extracted from such a request to determine related content. In some cases, the requester can also specify keywords for obtaining matching content rather than submitting a search query. The requests are forwarded to the neighbors of the requesting node. If any of the neighbors hold a matching content, they reply with that content. Otherwise, the request is again forwarded by the neighbors. In this way, nodes keep forwarding a request until it reaches a content holder who tries to deliver the content to the requester. In the absence of any administrator (i.e., decentralized approaches without any group and administrator), it is very challenging initially to forward a request to an appropriate content holder. Another problem is content delivery, even if the content holder receives the request, because everyone in such an environment is dynamically moving.

On-demand content sharing has been employed in several works [25, 33, 38, 56, 61]. Lindemann *et al.* [61] were among the first to propose On-demand content sharing. They proposed that mobile devices can collaboratively create a document indexing service called the passive document index (PDI) for file sharing applications. A mobile device generates a query and forwards it to its one-hop neighbors. Upon receiving the query, the neighbor checks local files and replies with only the identifier (i.e., meta-data of file) of the matching files. The neighbors also forward this query to their other neighbors. If any other nodes have a matching file, it also replies with the identifier, which is then used by everyone to update their PDI. In this manner, nodes generate a directory for contents held by other nodes. Figure 2.6 [61] shows a schematic diagram of this approach.

Although the above work provides a generic solution to obtain lists of files (i.e., contents) currently held by other nodes, it does not address the problem of identifying the content holder to be selected for obtaining the actual content. In this regard, McNamara *et al.* [56] suggested that the content holder should be selected based on the co-location stay probability. In this case, the content holder who is expected to stay within the reach of the requester for a longer amount of time is selected as the content source. This approach assumes that nodes meet frequently for similar amounts of time, which might not be the case for all types of environment. None of the above approaches uses any grouping technique or an administrator. In contrast, works in [33, 38] used a group based on-demand content sharing approach. Gao *et al.* [38] suggested that nodes should keep a copy of the content in multiple central nodes (i.e., administrators) which are easily accessible by other nodes and content can be obtained from these central nodes upon request. In comparison, a group administrator is used in [33] for maintaining a list of contents currently owned by the group members. Content requests are hence always forwarded to the administrator who directs them to appropriate content holders. A more detailed description of the group based decentralized content sharing approaches is provided in Section 2.2.1.

To summarize, on-demand decentralized content sharing considers that content can be delivered upon request. It is assumed that a requester will have sufficient idea about the content it wants, and will mention that adequately while generating queries. A set of keywords can be extracted from such a query or, alternatively, a requester

Data Dissemination Technique	Pros	Cons
Publish-subscribe	• Does not require user input for each content request • Allows users to mention their interest from a coarse level	• Consumes unnecessary resources by delivering additional contents • Users needs to know the list of available channels before subscription
Advertisement	• Useful for broadcasting information among users • Helpful in promoting events or business	• Not useful for providing on-demand contents • Consumes additional resources to deliver unnecessary contents
On-demand	• Delivers more user-centric contents to match fine-grained user interest • Only delivers contents when requested by users • Provides contents within short time if communication path is available	• Requires user input for each content request • Needs extraction of user profile and other information to provide efficient content delivery service

Table 2.1: Different data dissemination techniques for DCS

needs to specify some properties of the content (e.g., content creator, name of the content, tag) which is then compared against the stored contents to identify matching ones. Although the on-demand content sharing approach requires user input to obtain a content, it matches the interest of a user at a fine level and delivers contents that are more useful to the user. It also does not disseminate unnecessary content.

2.1.4 Why the On-demand Approach is Selected?

The previous section highlighted different data dissemination techniques for decentralized content sharing. All the techniques are useful in different scenarios to deliver contents and have their potential advantages and disadvantages, which are highlighted in Table 2.1. However, the focus of this thesis is to provide a content sharing solution in irregular meeting places where users will be more interested in receiving particular contents about the facilities or activities available in a tourist spot and are more likely to consume contents that match their interest on a fine level. In this case, delivering unnecessary contents will make them less interested in using the application. For example, in the case of a publish-subscribe approach, contents related to a particular channel are delivered. In this case, it might meet the generic interest of users (e.g., contents related to fishing or boating), however, they might only be interested to obtain contents related to the place they are currently visiting (e.g., contents related to fishing and boating facilities available at location '*l*'). The same is true for advertisement based techniques where other users proactively push contents without explicitly requesting them. To this end, the on-demand content sharing is the most appropriate solution to deliver user-oriented contents upon request that can fulfill the needs of the users more effectively. In this approach, users explicitly request contents they are interested to consume and only the matching contents are delivered in response. Such fine-level interest matching is important to enhance the sharing experience of users as well as minimize resource consumption by avoiding unnecessary sharing of contents. The network under consideration already suffers from intermittent connectivity and limited resources, hence using the available resource and communication opportunities in a better way to meet the content demand of users is very important. Therefore, in this thesis, the focus is on providing an on-demand decentralized content sharing service to users. The following section will focus on different issues associated with the major components of decentralized content sharing and the way they are addressed in the literature.

2.2 Major components and their issues in decentralized content sharing

The major components of a decentralized content sharing approach include group formation, message forwarding, content replication, participation incentives and trust management. The group formation part addresses how groups are formed among the participating devices while the message forwarding part determines the way the contents are delivered to appropriate requester. The incentive mechanism addresses how the users are provided with encouragement to actively participate in the content sharing process while trust management addresses how the misbehaving nodes are handled. The content replication policy addresses when a content needs to be proactively replicated, and how many copies need to be replicated and where (i.e., in which

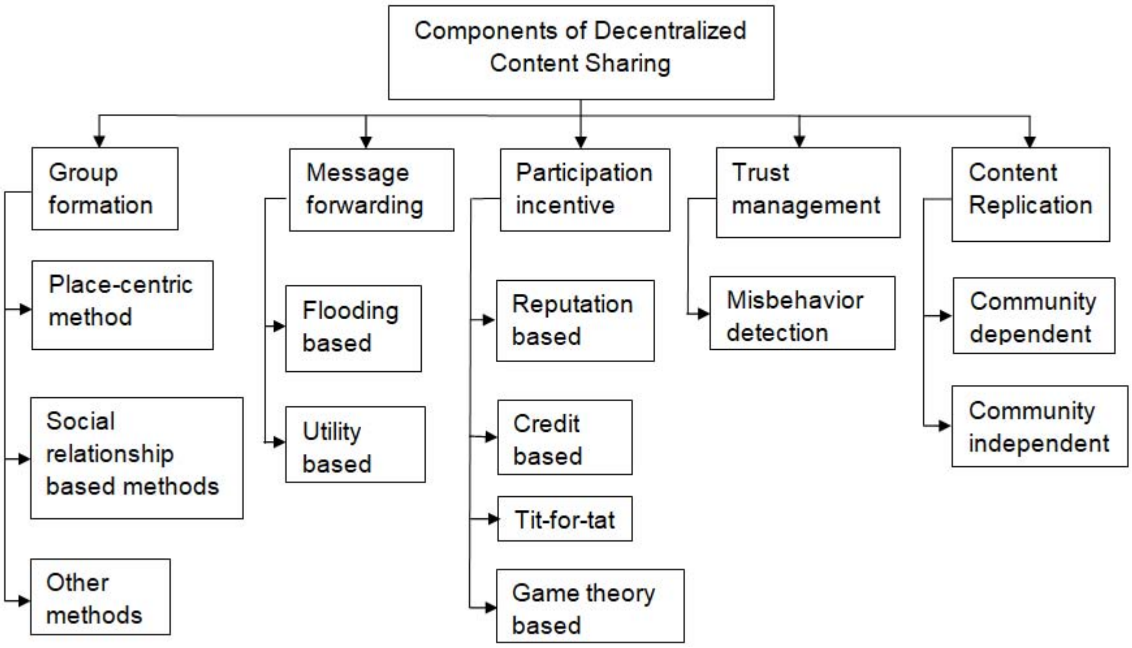

Figure 2.7: Major components of decentralized content sharing

nodes) those copies should be placed. The existing literature has focused on different aspects of these major components and proposed some interesting solutions. Figure 2.7 depicts a schematic representation of the major components and issues related to decentralized content sharing approaches. These issues are discussed in detail in the following sections.

2.2.1 Group Formation

Group formation (i.e., community construction) is one of the most important aspects of decentralized content sharing as it provides a better way to manage user information and helps provide on-demand content. The group members have common interests and can serve each other. The centralized and hybrid approaches enjoy the benefits of having a centralized server, which helps to manage groups easily, however, it is relatively complicated in a decentralized approach.

Some of the content sharing approaches do not use any grouping, such as the work proposed in [25, 56]. However, content sharing without any grouping makes it difficult to find a particular content since no one knows who has which content. In this case, nodes keep forwarding a request until it reaches a node with the matching content. This approach consumes lots of unnecessary resources by forwarding requests to nodes who are not interested in that type of content and hence are unlikely to have any relevant content. It is possible to maintain a distributed content list for all the participating nodes; however, sharing the content list with everyone (i.e., not limiting within the defined group) produces more traffic. Therefore, several group formation methods have been proposed in the literature [32, 34, 35, 62].

Some of the proposed methods have considered the temporal and spatial regularity of movement patterns to form groups while others have focused on social relationships for group formation. A few works have also focused on both movement patterns and social relationships together, along with mutual interest for group formation. The group formation methods can be broadly categorized into three categories: (i) place-centric method, (ii) social relationship based method and (iii) other methods. Techniques discussed in the first category emphasize nodes visiting common locations more frequently, while the second category focuses on a group of nodes meeting more frequently, regardless of the location. Details of these group formation methods are discussed in the following sections.

2.2.1.1 Place-Centric Methods

Place-centric groups are created around specific locations considering that nodes who meet regularly around a particular location are more likely to have some common interest and should form a group to share contents. The argument behind such approaches is that people show a spatial and temporal regularity in their daily life and are expected to meet a common set of people with similar interests at the same place on a regular or a semi-regular basis. For example, graduate students meeting at campus. It is considered that people spend different amounts of time in different places that can be represented by a matrix ($LocT$) presented in Eq. (2.1), where l_n represents a location and t_n represents time spent at that location. Groups are formed at locations where people spend more time than a threshold.

$$LocT = ((l_1, t_1), (l_2, t_2), ..., (l_n, t_n)) . \tag{2.1}$$

Urbiflocks [34] is a distributed framework proposed for group creation and maintenance where a user is responsible for initiating a group. During the group creation phase, users indicate the purpose or interest of the group. Afterwards, other users are automatically added to this newly created group based on their profiles (e.g., hobbies) and locations. For example, a group created for badminton players would automatically add other users who are nearby, interested in badminton and/or friends of the group initiator. Once the group is created, the maintenance of the group is automatically handled by the system. MobilisGroups [35] further extended this work by introducing a time restriction along with the location for group formation. For example, they considered a scenario where many participants are interested in a particular event (e.g., BarCamp). An organizer of the event created a group for the event, which was only visible at the venue of the event four weeks before the event. Participants were only allowed to join the group if they were physically present at the location during the event and shared messages and contents.

Johari *et al.* [63] have also considered creating a group around a base station. The base station was considered as the Point-Of-Interest (POI) and a single group was formed around it. Whenever the number of visits by a user surpassed a pre-defined threshold, it joined the group available there. The underlying assumption of this work was that users visiting a particular place frequently have a better chance of meeting other users with the same interest in that place. Costa *et al.* [64] have also considered grouping nodes that are physically close to each other in a particular location. Similar approaches are employed in [65–67] where the authors have argued that people with similar interests form a group and usually meet in the same place. However, the assumption of users being physically close might not be true for a large outdoor area where people with similar interests might be far away from each other. In addition, people with different interests may meet frequently in the same place, while people with the same interest may also meet at different places. For example, students frequently meeting at university campus might have different interest for consuming contents. Some students might be interested in news related contents, while others might be interested in music related contents. Again, a group of friends who might share common interests can meet at different locations, such as shopping centers, playgrounds or university. In this case, their relationship should get priority over the place they are meeting. This type of variation is not well addressed in the above-mentioned works.

Most of the above-mentioned works did not use the idea of using a coordinator or administrator for the group (i.e., community). The work proposed in [65] assumes the existence of a super user that will move around for delivering data to other nodes. Although such super users might be available for disseminating some advertisements (e.g., for promotional events or natural disaster), they are not suitable for on-demand content delivery.

In summary, this type of approach requires repeated visits to certain locations for users to form content sharing groups. Although frequent visits to a particular location show a level of mutual interest (e.g., colleagues at offices), other factors, such as online friendship or interaction, are likely to indicate a more accurate reflection of content sharing characteristics of users and hence need to be considered for group formation. Such social relationship based group formation methods are discussed in the next section.

2.2.1.2 Social Relationship Based Methods

This type of group formation method emphasizes social relationships among nodes, regardless of the place of meeting. In this case, frequently encountered nodes form a group considering that they have a better chance of helping each other for sharing content as a higher meeting frequency indicates high probability of meeting again in the near future, as well as having a common interest. For example, a group of friends might meet frequently at the university campus, playground or dormitory and are more likely to have common interests among them for sharing contents.

MOPS [36] is one of the social relationship based group formation methods where nodes with higher closeness form a group. In this approach, each node (u) recorded the encounter time and duration of meeting with another node (v) in the network and calculated a closeness metric ($C_{uv} \in [0, 1]$) to represent the time-space relationship with that node. A higher value of the closeness metric indicates greater probability of meeting in the future. A training period was used to calculate the value of such closeness metrics. The closeness between node u and v was measured using the following equation as,

$$C_{uv} = exp(-\frac{AVG(D_{u,v})^2}{2\sigma^2}). \tag{2.2}$$

Here, $AVG(D_{uv})$ represents the average inter-contact period between nodes u and v, and σ shows a scaling parameter for the separation period. The average inter-contact period ($AVG(D_{uv})$) was calculated using the ratio of the total number of times nodes were separated and the number of total separations. Smaller values of $AVG(D_{uv})$ indicate that nodes were separated for a shorter time and hence had stronger ties. When the closeness value was more than a pre-defined threshold, nodes were considered as local neighbors and formed a group. The idea of such local neighbors with direct communication was further extended with the introduction of virtual links that represented an indirect connection. A virtual link between two participating nodes suggests that one of them is reachable from the other node via neighbors, if direct communication is unavailable. The virtual link between nodes u and v was calculated using the following equation.

$$C_{uv} = max_{p \in P}\left\{\prod_{<x,y>\in p} C_{xy}\right\}. \tag{2.3}$$

In the above equation, P represents the set of all paths between u and v that are less than or equal to $k - hops$ and $< x, y >$ represents an edge in the path. The path closeness for a path p is represented by $\prod_{<x,y>\in P} C_{xy}$ which is the product of all edges along the path. Similar to local neighbors, the nodes connected with virtual links formed a group when their closeness metric was greater than the threshold. A similar social relationship based group formation was used in [37, 57, 68] where participating nodes were divided into groups, such as family, friends and familiar neighbors. Group formation based on frequent opportunistic contacts and common interest was explored in [32]. In this work, broker nodes were first selected based on their popularity among other nodes. The popularity value of a node was calculated considering the number of other nodes it encountered during previous time windows. Afterwards, the broker nodes performed the responsibility of adding new group members based on the interest of that node.

One of the recent studies called SPOON [33] has also argued that people with similar interests tend to meet each other more often and constructed a group based on interest similarity and meeting frequency. It used the stored content lists in a participant's device to calculate the interest of that user. Whenever two nodes $N1$ and $N2$ met each other, they exchanged information about their current affiliation with groups and their interests. Two distinct cases were considered to handle the group formation task: (i) none of them are part of any existing group and (ii) at least one of them is a part of a group. In the first case, if their interest similarity and meeting frequency were greater than a pre-defined threshold, they formed a new group together. Otherwise, if one of them was already a part of an existing group, the other joined that group if the similarity values exceeded the threshold. Interest similarity value was calculated using the cosine similarity metric as,

$$\mathrm{sim}(\mathbf{v1},\mathbf{v2}) = \frac{\sum_{q=1}^{Q} w_{1_q} * w_{2_q}}{\sqrt{\sum_{q=1}^{Q} w_{1_q}^2}\sqrt{\sum_{q=1}^{Q} w_{2_q}^2}}. \tag{2.4}$$

In the above equation $\mathbf{v1}$ and $\mathbf{v2}$ represent two interest vectors from two users, Q represents the total number of common keywords among them, w_{1_q} and w_{2_q} show the weight of the q-th common keyword in $\mathbf{v1}$ and $\mathbf{v2}$, respectively. The contact frequency was calculated using the number of times the nodes met each other. SPOON also used the concept of a central node (an administrator) for each group to handle group management tasks, such as storing the list of contents of other group members and directing content requests towards appropriate content holders. The administrator nodes were selected based on their centrality value as,

$$D(P_i) = \sum_{v=1, v\neq u}^{G_m} w_{uv}, \tag{2.5}$$

where, w_{uv} is the weight between two members of a group and G_m is the total number of members in that group. To select a member with a high amount of connection with other members of the group, the value of w_{uv} was set to 1 whenever the contact frequency between two nodes was greater than a threshold.

Similar central nodes were used in other studies for information dissemination, routing and caching purposes [26, 38]. Common social metrics used for central node selection in social relationship based groups include degree centrality, closeness centrality and betweenness centrality. These metrics can be calculated using the following equations.

$$\text{Degree centrality}, C_D(u_i) = \sum_{k=1}^{K} a(u_i, u_k)\,. \tag{2.6}$$

The above equation is used for degree centrality, which refers to the number of direct links a node has with other nodes. Here, u_i and u_k represent two nodes, and $a(u_i, u_k) = 1$ denotes that node u_i has a direct link to node u_k, and 0 means no link exists.

The closeness centrality can be calculated as,

$$\text{Closeness centrality}, C_c(u_i) = \frac{K}{\sum_{k=1}^{K} d(u_i, u_k))}\,, \tag{2.7}$$

where, K is the total number of nodes and $d(u_i, u_k)$ represents the geodesic distance between node u_i and u_k. The geodesic distance refers to the shortest path connecting two nodes.

The betweenness centrality refers to the node that lies on the shortest geodesic path that connects other nodes in the network which is calculated as,

$$\text{Betweenness centrality}, C_B(u_i) = \sum_{k=1}^{K} \sum_{j=1}^{k-1} \frac{g_{u_j u_k}(u_i)}{g_{u_j u_k}}\,, \tag{2.8}$$

where, $g_{u_j u_k}(u_i)$ represents the number of geodesic paths between node u_j and u_k that includes node u_i, and $g_{u_j u_k}$ represents the total number of geodesic paths between u_j and u_k.

In summary, social relationship information obtained from existing social networks or collected over time provides a good platform for creating and managing group membership as tightly knit members can be easily identified from such information. However, groups based on social relationships fail to take into account the importance of locations, which facilitate content sharing among co-located peers. Also, most works assumed availability of a social network graph or built one over time, which is not always possible due to limited contact. Another problem is that they fail to capture the change of relation dynamics with respect to place and change of interest with respect to place. For example, the relationship with people over time might change. The static group formation method fails to incorporate that change. Again, people might be interested to form different kinds of relationship with the same people based on the context. This necessitates a dynamic group formation method. For example, at the work-place, people might want to share work related content with colleagues, but they might be interested to share different types of content with the same colleagues outside the office.

2.2.1.3 Other Methods

There are also a few other noteworthy approaches apart from those mentioned in the previous sections. Thilakarathna *et al.* [69] have mentioned group formation based on individual content. They have considered a centralized architecture where the server maintains three matrices, namely, user-user social relationship matrix, content-content similarity matrix and user-content similarity matrix. From these three matrices, the server tries to predict the user who will be the more likely consumer of a particular content. Afterwards, the server creates a group for each of the contents. However, such schemes are not directly applicable for decentralized sharing and also maintaining a single group for each content is computationally expensive when the number of contents is very high.

Yu *et al.* [70] have proposed an approach for group formation using a hybrid content sharing approach. Nodes keep track of their mobility profile, that includes both location information and meeting frequency, with other nodes and send it to a server where an influence graph is calculated. Finally, nodes are clustered into communities based on the similarity of their mobility patterns. This scheme is also not directly applicable to DCS as it needs a server.

Another interesting group formation method has been proposed by Tian *et al.* [71] where the authors have suggested formation of short-term communities, which they call spontaneous community, based on user profiles. The profile of a user consists of attributes such as demographics, current context (location, velocity), interest, and list of permanent friends. The profile of user u_i is represented using a profile vector $\mathbf{p_{im}}(t) = p_{i1}(t), p_{i2}(t), ...p_{i|\mathcal{F}|})$, where $p_{im}(t)$ represents the value of feature m at time t and $\mathcal{F}$ is the set of all available features. Similarly, each community $Com^k(t)$ is also represented by a community profile vector $\mathbf{cp_m^k}(t)$. Whenever a newcomer u_j enters a new location, one of the existing users u_i provides it with the list of available communities and their profile vectors. The new comer identifies its benefits of joining a particular community. The benefit is calculated using the dissimilarity score of personal profile and community profile as,

$$SC^k(u_j, Com^k(t)) = \sum_{m=1}^{|\mathcal{F}|} W_m^k(t) d_m(\mathbf{p_{jm}}(t), \mathbf{cp_m^k}(t)). \tag{2.9}$$

Here, $\mathbf{cp_m^k}(t)$ is the received community profile of the k-th community and $W_m^k(t)$ represents the weight of feature m in this community. $\mathbf{p_{jm}}(t)$ represents the profile vector of user u_j. If the dissimilarity score is smaller than a certain community threshold then the user joins the community. This approach required members to be present within the communication range of each other to form a community. It also suffers from initialization issues for starting a group. The system could learn over time and identify important features for different communities based on user behavior, but during the early stages it requires user inputs for joining communities.

Table 2.2 summarizes some key features of the group formation methods. The existing literature mostly assumed spatio-temporal regularity in human movement patterns and exploited that information for creating groups. However, there are some scenarios in irregular meeting places, such as tourist spots or camping sites where such regular movement patterns or pre-existing social relationships do not exist. In such scenarios, 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. Therefore, group formation in such dynamic environments is more complicated. Another important aspect that is mostly overlooked in existing works is the probability of finding sufficient contents in a group and successfully obtaining them upon request. Obtaining contents related to one's own interest is the main objective of decentralized content sharing. Therefore, a node needs to check that contents related to its personal interest are available in a group and those contents can be delivered upon request by the group members, which has not been addressed in the literature. Since interest is one of the most important factors for forming a content sharing group, the following section highlights existing interest extraction techniques.

Method name	**Group formation metric**	**Requirement**	**Joining criteria**	**Group initiation**
Urbiflocks [34]	friendship or physical proximity	×	matching hobbies and distance within pre-defined limit	manually by user
MobilisGroups [35]	mutual interest, location and time restriction	×	presence within certain proximity at a particular time	manually by user
CACBR [63]	common point interest	frequent visit to a POI	contact strength greater than a threshold	base station initiates
Costa *et al.* [64]	physical proximity		presence at a particular location	base station initiates
Geo-community [65]	common hobbies, social functions, and occupations	frequent visit to particular locations	sojourn time greater than a threshold	not mentioned
CAOR [72]	common interest	frequent visit to particular locations	number of visits greater than a threshold	not mentioned
MOPS [36]	neighboring relationship	frequent encounters	closeness metric greater than a threshold	node initiates
Yoneki *et al.* [57]	proximity or common interest	frequent encounters or common neighbors	contact duration or number of common neighbors greater than a threshold	node initiates
Socker [32]	common social activity	frequent encounters for broker selection	user can manually define	broker initiates
SPOON [33]	common interest frequent meeting	frequent encounters	interest similarity and contact frequency greater than a threshold	node initiates

Table 2.2: Group formation methods in decentralized content sharing schemes

2.2.1.4 Interest Extraction

In decentralized content sharing, nodes exchange their interest information with their neighbors upon meeting and form or join a group based on mutual interest. Since interest is one of the basic criteria for joining a group, this section discusses some of the techniques proposed in the existing literature for determining the interests of a particular user.

Social networking platforms such as Facebook and Twitter provide an efficient way of tracking user interest (e.g., music, movies or news items). In this case, interest is extracted from a user's previous posts, comments and uploaded contents. For example, a user might be frequently posting about sports updates which represents a high level of interest in sports related content. Another user might be interested in rock music and usually uploads relevant contents. Such interest extraction mechanisms are very popular for centralized content sharing approaches where the central server can track the activity of a user, determines interest for different content categories and dynamically updates the level of interest (i.e., interest score). Such activity based interest extraction is proposed in [73, 74]. For a decentralized content sharing approach, since there is no centralized server, the content sharing application installed in a user's device can determine the interest of the user based on the past activity of that user. Work in [75–77] proposed such activity based interest extraction from the history of content access patterns of a user. It is also possible to obtain interest directly from the user through input, which is explored in publish-subscribe based content dissemination where the basic idea is that users will specify their own interests and accordingly will subscribe to channels publishing relevant contents [59, 78]. One of the recent approaches has also used stored content lists in a user's device to extract interest information [33].

Most of the above-mentioned approaches fail to take into account the dynamic nature of the interest of a user, which is expected to change based on the context (e.g., location, surrounding neighbors). To address this issue, Tian *et al.* [71] proposed an interest extraction technique where users showed different levels of interest in different communities. A community-aware profile was maintained by a user's device to reflect the user's interest in different content categories inside a community. Such community-aware profiles were updated based on the reaction of the user in response to a published content as,

$$p_{im}^{k}(t+1) = \begin{cases} (1-\delta_s)p_{im}^{k}(t) + a_m\delta_s, \text{ If } a \text{ is interesting to } u_i \\ (1-\delta_s)p_{im}^{k}(t) - a_m\delta_s, \ otherwise\,. \end{cases} \tag{2.10}$$

In the above equation, $p_{im}^{k}(t)$ represents the interest of user i for interest feature m inside community k during time t, a_m is a content with feature m and δ_s is a weighting factor. Although the proposed approach gives an indication that the interest of a user might change based on the context, it only relies on the publication of relevant content to capture this. An improvement considering the feature or facilities available in a particular area can be more accurate in capturing the dynamic change of user interest.

2.2.2 Message Forwarding

Message forwarding plays a major role in successful content delivery. It is very challenging in DCS as there is no fixed end-to-end path from source to destination due to node mobility. Therefore, mostly a store-carry-and-forward method [39] is used to handle this, where nodes keep carrying a message until they meet another forwarder node (i.e., relay node) or the destination node. The aim of message forwarding techniques is twofold: the first part focuses on forwarding a content request to an appropriate content holder, while the second part deals with delivering the matching content back to the requester. The concept is somewhat similar to the techniques proposed for a delay tolerant network [79] and an Oportunistic Network (OppNet) [80]. The main difference is the first part of the decentralized content sharing approaches that deals with forwarding the request to an appropriate content holder, which is not addressed in DTN and OppNet. In a group-oriented decentralized content sharing approach with the presence of an administrator, the first part becomes manageable, since the administrator can maintain the content list of all group members and hence can forward a request to an appropriate content holder. However, it is still challenging due to intermittent connectivity among nodes. A decentralized content sharing approach without any grouping mechanism suffers to handle the first part as the content requests must be blindly forwarded until they reach nodes with the matching contents. Since the size of the content request is smaller (generally $\leq 1-2$ KB), making multiple copies of a request and sending them over different paths does not consume much network resources.

The problem is more prominent for the second part of the message forwarding, which handles content delivery from the content holder to the requester as the size of the content is significantly larger than the request. Although making multiple copies increases the probability of successful content delivery, at the same time it also consumes more network resources (e.g., bandwidth, energy). The forwarding approaches for DTN and OppNet consider that the destination is already known and focus on the successful delivery of content/message from source to destination which is the same as the delivery of content from a content holder to the requester. Therefore, this section also highlights the forwarding methods for DTN and OppNet, which are extensively studied in the existing literature.

The existing literature proposed for message forwarding techniques can be classified into two categories: (i) flooding based approach and (ii) utility based approach. Flooding based approaches blindly create multiple copies of a message without considering resource availability and forward them until some certain criteria are reached (e.g., maximum hop traveled or maximum distance traveled). On the other hand, in a utility based approach, a message is forwarded to the next node (i.e., relay or forwarder node) based on some context or suitability of that node. Both of these message forwarding techniques are discussed below.

2.2.2.1 Flooding Based

One basic and representative message forwarding scheme is called epidemic routing [81], where a node carrying a message forwards it to every other node it meets who does not have a copy. Although epidemic routing yields most successful deliveries with lower latency, it requires a high amount of resources and creates unnecessary copies in the network which, in turn, increase network congestion. To reduce the overhead of epidemic routing, a number of strategies have been proposed in the literature.

The problem of making redundant copies of the same message using the epidemic routing scheme is termed the 'broadcast storm problem' in [82] and a number of strategies are proposed to handle such a problem. The proposed strategies include forwarding based on a probabilistic metric, a counter, distance and location. In a probabilistic metric based scheme, nodes only rebroadcast a message when a randomly generated number is greater than some pre-defined threshold while the counter based scheme uses a counter and compares it with the threshold value to determine if a message should be forwarded. When a node receives the message for the first time, the counter value is initialized with 1. During subsequent reception of the same message from other nodes, the counter value is incremented by 1. A node only forwards a message when this counter value is less than some predefined threshold. In the case of a distance based scheme, a node only rebroadcasts if the distance from the source node is greater than some distance threshold. In contrast, a location based scheme considers the location of the transmitting host and the additional coverage it can provide before rebroadcasting a message. However, selections of appropriate threshold values are not discussed in this work.

Chang *et al.* [83] used a Time-to-Live (TTL) value to control the quantity of redundant messages. Each message was initialized with a TTL value, which was decremented after every successful transfer to subsequent nodes. The forwarding process ended when the value became *zero*. In contrast, Pitkanen *et al.* [84] considered the number of hops the message has already traveled to control the message propagation. Similar hop-limited flooding is also investigated in [85] to determine its efficiency in varying network conditions. Recently, Lu *et al.* [86] proposed a method where the broadcast transmission range is controlled considering energy consumption and delivery predictability to gain better energy efficiency.

Spray-and-Wait [49] is another notable work which spreads a specific number of copies (R) of a message in the network. In the spray-and-Wait method, whenever a node meets another node with $m > 1$ copies available for forwarding, it forwards $\lfloor \frac{m}{2} \rfloor$ copies and keeps $\lceil \frac{m}{2} \rceil$ copies for itself. If any node has only a single copy available, it holds on to that copy until it meets the destination. Figure 2.8 [49] shows the working procedure of the Spray-and-Wait forwarding approach. Here, node A has 7 copies of a message available at time t_1. When it meets node B and E at time t_2 and t_4, respectively, it transfers 3 copies to them. Similarly, node B transfers 1 copy each to nodes C and D during interval t_4. In this manner, a source node distributes copies of a message in the network upon meeting other nodes. However, determining the appropriate value of R (i.e., no. of copies) in the initial stage in a network with unknown parameters (i.e., total number of nodes) is difficult since the estimation requires periodic contacts and a significant learning period. A variation of this approach is investigated in [87], where a node with a single copy of the message forwards the copy to a better relay node if it

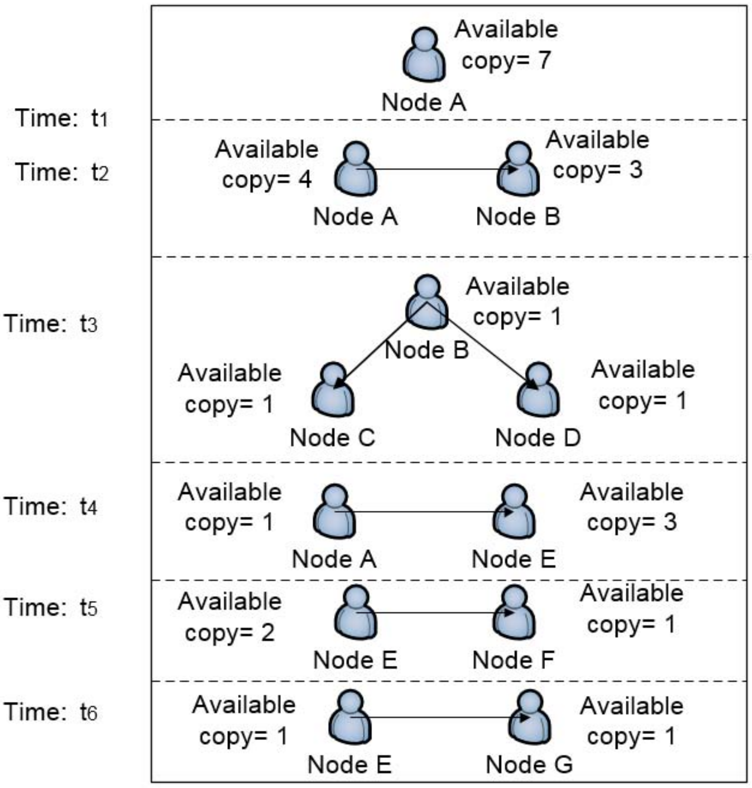

Figure 2.8: Spray-and-Wait message forwarding protocol adapted from [49]

does not meet the destination. Similar to the work proposed in [49], Grossglauser *et al.* [88] explored a two-hop flooding approach, where the source node distributes m copies of a message to relay nodes who hold on to them until they meet the destination. The delivery from source to destination happens via two-hop communication. Niu *et al.* [89] proposed a forwarding method where each node is permitted to spread k number of copies of the same message. In this case, any node with a message to forward can introduce additional k copies in the network rather than the source node.

One major concern of these approaches is the selection of the appropriate value for initial spread under dynamic network conditions with varying numbers of nodes and their movements. Recently, Wu *et al.* addressed this in [90] and proposed a copy limited epidemic message forwarding approach where all the nodes are divided into multiple communities and the source node determines the optimal number of copies required in different communities to achieve a certain delivery success rate within tolerable delay. They formulated a classic combinatorial optimization problem using the above constraints and solved it using a constrained non-linear optimization solver.

Li *et al.* [91] proposed another interesting work to control the spread of information. They used the concept of 'Turf', which was considered as a logical location of the receiver in the temporal and spatial domain. Receivers who were located within the same 'Turf' as the sender (i.e., co-located for a longer time or remained within close distance) were allowed to receive all the information. Otherwise, no information was passed.

To summarize, flooding based message forwarding approaches mostly depend on spreading a specific (or unlimited) number of copies of a message in the network. Since, more messages are spread in the network, it is likely that at least one of them reaches the destination within a short time and hence the achieved delivery success rate is usually high in these methods. However, the amount of required resources for these approaches increases substantially along with the number of copies, which results in significant energy drain. Determining the appropriate number of copies to spread in the network is also challenging without the presence of any global observer and in a dynamic network. In addition, these approaches do not consider the suitability of a relay node for successful delivery and hence result in making redundant copies of a message and needlessly drain resources. To address this issue, a utility based forwarding approach is adopted by researchers considering the capability of a node for delivering a message before forwarding a copy of the message. These approaches are discussed in the following section.

2.2.2.2 Utility Based

In this approach, whenever two nodes meet with each other, they first forward the messages for which the other node is the destination. Afterwards, they exchange a summary of the messages they are currently carrying and their respective utility (e.g., delivery probability or expected delay) for successful delivery. For any of the messages, if the other node has a higher utility, it is selected as a relay node and the message is forwarded. The utility value is measured in terms of different aspects. Some of the approaches utilize frequent encounters as the metric of calculation and a node that meets the destination more often is selected as the new relay node. Another type of

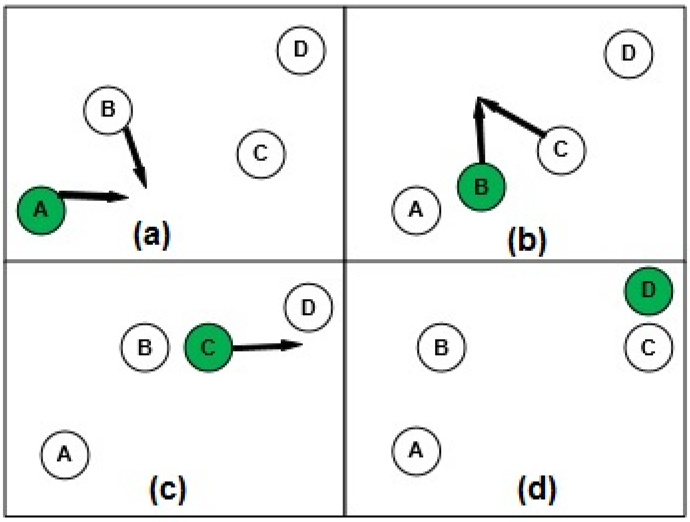

Figure 2.9: Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) adapted from [21]

approach employs social attributes such as common interest with the destination or popularity of a node as the metric for taking forwarding decisions. Physical contexts, such as mobility pattern and the location of the nodes, were also analyses to take forwarding decisions. Different utility based approaches are discussed in the following sections.

Contact Pattern Based Forwarding Approach

This type of approach considers the frequency or recency of meeting with other nodes in the network and predicts the probability of successfully delivering a message to its destination. Fresher encounter search (FRESH) [20] is an early encounter based forwarding approach where every node keeps track of the encounter time with other nodes and a node with the most recent encounter with the destination is selected as the relay node. Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [21] is one of the prominent works that considers the probability of meeting other nodes based on the frequency of previous encounters. In this case, each node maintains a metric called *delivery predictability* denoted as $P_{(a,b)} \in [0,1]$ suggesting the probability of node a successfully delivering a message to node b. The value of $P_{(a,b)}$ is updated whenever node a meets node b as,

$$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) * P_{init} . \tag{2.11}$$

Here, $P_{init} \in [0,1]$ is an initialization constant. If a pair of nodes does not meet for a long time, the value of $P_{(a,b)}$ decreases using the following equation.

$$P_{(a,b)} = P_{(a,b)_{old}} * \gamma^k . \tag{2.12}$$

Here, γ is an *ageing constant* and k is the number of time units elapsed after the last meeting. The authors also considered *transitive delivery predictability*. In this case, if node b regularly meets node c and node a regularly meets node b, the delivery predictability from node a to node c can be calculated as,

$$P_{(a,c)} = P_{(a,c)_{old}} + (1 - P_{(a,c)_{old}}) * P_{(a,b)} * P_{(b,c)} * \beta , \tag{2.13}$$

where, $\beta \in [0,1]$ is a scaling constant that determines the weight of the transitive property. The forwarding mechanism in this approach considers that a higher meeting frequency between two nodes indicates that they are more likely to meet again. Therefore, this approach selects a relay node that has higher delivery predictability for the destination. Figure 2.9 [21] depicts the working procedure of this approach. In this figure, node A is the source and node D is the destination. When, node A meets node B, it forwards the message to B as its delivery predictability is higher. Similarly, B forwards the message to C upon encounter and finally, C delivers the message to D. Although PROPHET considers delivery predictability, it does not account for available buffer space or remaining energy of the relay node to take forwarding decisions. To address this issue, recently a number of approaches have been proposed [92–94] that use available buffer space and remaining energy along with delivery predictability to select a relay node.

A similar encounter based forwarding approach is also adopted in [22, 95–97]. Burgess *et al.* [22] prioritize packets for transmission as well as deletion. Each node maintains and updates the probability of encountering other nodes (f^{u_i}) using incremental averaging (i.e., $\sum_{u_i} f^{u_i} = 1$). The cost of delivering a packet is calculated considering the sum of probabilities of encountering all the nodes in the path as,

$$cost(u_i, u_j, ..., u_d) = \sum_{u_k = u_i}^{u_d - 1} \left[1 - \left(f^{u_k}_{u_k+1} \right) \right] . \tag{2.14}$$

Here, $f^{u_k}_{u_k+1}$ shows the path cost from u_k to $u_k + 1$ and u_d is the destination node. Finally, the path with the lowest cost is selected and packets are sorted using this cost value. During a transfer opportunity, higher ranked packets are transferred first. Packets with higher hop-count are also deleted first in this approach when the buffer is full. Spyropoulos *et al.* [95] proposed a single copy based forwarding approach where each node maintains a utility value for delivering a content to every other node in the network. This utility value decreases according to the last encounter time. Nodes only forward a message whenever they meet another node with higher utility. Localcom [96] considers that nodes are part of different communities, which is detected using their frequent encounter history, and identifies gateway nodes that meet members from multiple communities more often. In this method, intra-community message forwarding employs higher similarity and short hop-distance among community members, while the inter-community communication uses the gateway nodes to send a message to the community of the destination node and later perform intra-community message forwarding to deliver the message. The performance of this scheme was evaluated through test driven simulation where two datasets were used, namely (i) Haggle [98] and (ii) reality-mining [99] dataset. In the former dataset, the movement of 41 students in a conference environment was collected while the latter consists of the movement pattern of 97 students and staff in MIT. To analyze the performance of the forwarding method, 1000 packets were used where the source and destination were randomly selected. The performance of Localcom was compared against epidemic routing [81], PROPHET [21] and BUBBLE RAP [100]. A successful delivery ratio of 30-40% across all approaches was reported for the Reality mining dataset while the Haggle trace produced 80-85% successful delivery. Localcom achieved a higher successful delivery ratio than PROPHET and BUBBLE RAP while epidemic routing produced the best result in all cases in terms of delivery ratio. Inherently, the epidemic routing also produced the highest number of message exchanges than other approaches and hence consumed the highest amount of resources.

Context-aware Adaptive Routing (CAR) [101] is another forwarding protocol that uses a Kalman-filter based prediction technique to determine a node's utility for carrying a message. In CAR, utility value depends on future co-location and the change in degree of connectivity. If two nodes are co-located during a current time interval, the possibility of their co-location in future time intervals increases. Similarly, a node with a more dynamic neighbor-set (i.e., more unique nodes) is assigned a higher utility value. Finally, the best relay node is selected using these utility values and the Kalman-filter based prediction technique. The performance of CAR was compared against epidemic routing [81] where 200 nodes generated 1000 messages for random destinations and the simulation was run for 8 hours. The reported delivery success ratio for epidemic routing and CAR was 62.7% and 49.9%, respectively.

Although the above approaches consider contact frequency, they neglect the fact that the contact opportunity might be insufficient to complete a message transfer. To address this issue, [102, 103] proposed a contact duration aware forwarding approach. Kim *et al.* [102] suggested that since the contact duration might be insufficient to transfer the whole message, a *forwarding failure* occurs when part of the message cannot be delivered due to time constraint. Therefore, nodes can divide a message into several fragments and independently forward each fragment to the destination to utilize all contact opportunity and avoid *forwarding failure*. A more simplified forwarding approach is proposed in [103] where a relay node for intra-community forwarding is selected based on both frequency and duration of encounter with the destination node. If a node belongs to the same community as the destination node, and has higher contact frequency with longer duration than the current message carrier node, it is selected as the relay node. In case of inter-community forwarding, the global centrality of a node is used, which is calculated using the encounter frequency and duration of a node with all other nodes in the network, regardless of their community membership.

Gao *et al.* [104] exploited transient social contact patterns to undertake forwarding decisions. They considered that the contact patterns might vary during different times of the day. For example, classmates might encounter more frequently during day time while that might not be the case during night time. Therefore, instead of considering the cumulative contact distribution, they focused on transient contact distribution to identify nodes that are more likely to meet within a certain time period.

Community-aware Opportunistic Routing (CAOR) [72] employed the idea of home community and the presence of a static virtual throwbox in each community. A home community is assigned to each node based on its visiting frequency to particular locations. It is also argued that nodes are expected to meet at their home community

Protocol name	**Metric for utility calculation**	**Community**	**Energy of relay**	**Buffer space of relay**	**No of Message copies**
FRESH [20]	recent contact time	×	×	×	multiple
PROPHET [21]	contact probability	×	×	×	multiple
Ea-PROPHET [92]	contact probability, energy consumption and available buffer space	×	✓	✓	multiple
E-PROPHET [93]	contact probability and energy consumption	×	✓	×	multiple
RRPHETI [94]	contact probability energy and buffer	×	✓	✓	multiple
MaxProp [22]	contact probability	×	×	×	multiple
Seek-and-Focus [95]	recent contact time	×	×	×	single
Localcom [96]	encounter history similarity	encounter based	×	×	multiple
User-centric [97]	contact patterns and interest	×	×	✓	multiple
CAR [101]	co-location and connectivity	×	×	×	single
Kim *et al.* [102]	contact duration	×	×	×	multiple
Gondaliya *et al.* [103]	contact frequency duration	k-clique distributed	×	×	multiple
Transient [104]	transient contact pattern	frequent contact based	×	×	multiple
CAOR [72]	visiting frequency	visiting pattern based	×	×	single
SGBR [105]	meeting frequency	encounter based	×	×	multiple

Table 2.3: Contact pattern based message forwarding approaches

more often than at other places. The message forwarding between a pair of nodes is then converted into forwarding between two communities and the relay nodes are selected based on the expected minimum delivery delay using a reverse Dijkstra algorithm. Another interesting work proposed by Abdelkader *et al.* [105] provided a mathematical formulation for optimal forwarding strategy assuming the presence of a global observer in the network. The global observer collected encounter patterns among all the nodes and accordingly selected the best forwarding path considering limited available resources (e.g., buffer space, contact opportunity). A heuristic based forwarding protocol was also proposed in [105] using the average contact rate among participating nodes. Their simulation results indicated that their proposed forwarding protocol SGBR (60-90%) outperformed epidemic routing [81] (35-70%), PROPHET [21] (45-85%), Spray-and-Wait [106] (45-85%) and MaxProp [22] (60-90%) in terms of successful delivery.

The above approaches achieve higher delivery success rate and lower latency when nodes frequently meet each other on a regular basis, which provides an opportunity for predicting future contact and message delivery options. Table 2.3 shows some key features of the contact pattern based approaches. However, they are ineffectual for irregular meeting places (e.g., tourist spot or camping site) where such regular encounter is unavailable or insufficient to obtain meaningful information. In addition, these approaches do not consider any social attribute or characteristic of a user (e.g., popularity in the society or tie strength) for taking forwarding decisions. Inspired by such social attributes of the participants, researches have investigated a number of forwarding approaches, which are discussed in the following section.

Social Attribute Based Forwarding Approach

This category of forwarding approaches analyzes the underlying social alliance among participating nodes and leverages it for improving the delivery service. The social alliance is usually captured using a popularity or similarity metric. In the case of a popularity metric, the basic idea is to forward the message to the most popular node who is more likely to meet the destination. In contrast, the similarity metric based approaches identify a relay node that has more commonality with the destination node. Daly *et al.* [26] were among the first to explore the social relationships among nodes to undertake forwarding decisions. They employed social similarity and centrality to calculate a node's utility for forwarding a message. Social similarity between a pair of nodes was calculated using the number of common neighbors among them, while the betweenness centrality is measured with the help of an ego network depicting the number of connections a node has with other nodes. The utility value obtained from social similarity of a node u for delivering a message to destination v in comparison to node k is calculated as,

$$SimUtil_u(v) = \frac{Sim_u(v)}{Sim_u(v) + Sim_k(v)}\,. \tag{2.15}$$

Similarly, the betweenness centrality is also used to calculate a utility value as,

$$BetUtil_u = \frac{Bet_u}{Bet_u + Bet_k}\,. \tag{2.16}$$

Finally, using these two values, the utility value of node u in comparison to node k is calculated as,

$$SimBetUtil_u(v) = \varphi_{sim}\, SimUtil_u(v) + \varphi_{bet}\, BetUtil_u\,. \tag{2.17}$$

Here, φ_{sim} and φ_{bet} are tunable parameters to assign weight to the components of the utility calculation. Finally, a node with a higher utility value was selected as the relay node for carrying the message. This idea was further extended in [107] where another metric called *tie strength* was introduced for utility calculation which measured the intensity of a relationship calculated in terms of frequency, recency and duration of encounter. PeopleRank [27] is another approach that is inspired by the PageRank [108] algorithm used by Google to rank web pages. PeopleRank assigns a higher rank to the most popular and highly connected nodes in the network. Messages are then forwarded from a lower rank node to a higher rank node assuming that a higher ranked node has a better chance of meeting other nodes.

In real life, people are usually part of different groups (i.e., communities) and meet the group members more often. For example, students at a university campus or co-workers in an office. Such group related information can be utilized for message forwarding. LABEL [28] was one of the first works to realize this and proposed that each node has a 'label' associated with it that expresses its affiliation. The relay nodes were selected based on having similar affiliation (i.e., labels) as the destination node. This idea was further extended in BUBBLE RAP [100] which is a prominent social attribute based message forwarding protocol. This approach considered that nodes can be associated with multiple groups and have different rankings (i.e., popularity) in different

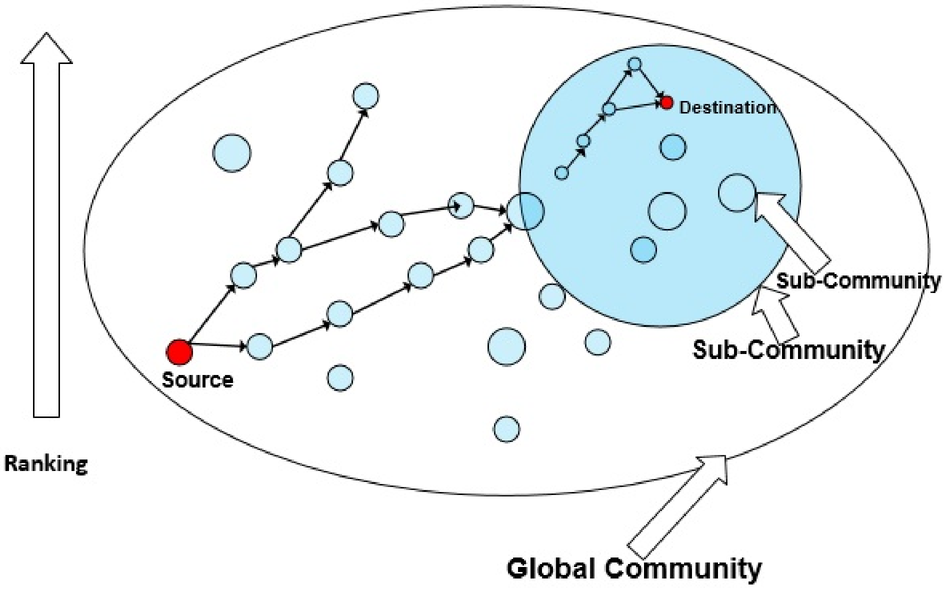

Figure 2.10: Bubblerap forwarding adapted from [100]

groups. Each node is assigned a global ranking and a local ranking (i.e., ranking inside the community). The ranking of a node was calculated using a degree and betweenness centrality. For message forwarding, initially a node with higher global ranking was selected until the message reached a member of the same group as the destination node. Afterwards, local ranking in that particular group was used to propagate the message among group members hoping that a more popular member in the group is more likely to meet the destination. The method is illustrated in Figure 2.10 [100]. Here, the source node continues to send a message via nodes with higher global ranking. Finally, when the message is received inside the sub-community of the destination node, ranking inside that community is used for delivering the message.

A similar social attribute based approach was proposed by Xia *et al.* [52]. They were inspired by an artificial bee colony algorithm, where bees are able to identify a nectar source with maximum density, and proposed a forwarding approach where nodes are able to determine density in a group (i.e., degree centrality or number of unique nodes seen) and keep track of the tie strength (i.e., amount of time spent with group members). For inter-community message forwarding density values were used while intra-community message forwarding utilized social tie values. Although these types of social popularity based approaches are likely to yield successful delivery, they also might result in congestion near the popular node whose energy will drain due to communication from other nodes. To evaluate the performance of this approach, a simulation was performed using an area of $4.5 \times 3.4\ km^2$ and 40 nodes where the size of the contents was varied within 500-1024 KB. Simulation results indicated that this approach achieved 55-75% delivery success rate while outperforming epidemic routing [81] (35-65%) and PROPHET [21] (35-55%).

Social similarity or commonality is another popular social attribute widely used in the existing literature for message forwarding purposes. SocialGreedy [109] is one such approach which assumed that social profiles of users are readily available and can be used for taking message forwarding decisions. They calculated the social distance between two users using attributes such as nationality, language, affiliation, city, country and interest/hobbies collected from their social profile. Finally, the relay node was selected that had shorter distance (i.e., more similarity) with the destination. However, the selection of appropriate social features (i.e., attributes) was not investigated. To address this issue, Wu *et al.* in [110] proposed that important social features can be extracted using the Shannon entropy formula as,

$$E(F_i) = -\textstyle\sum_{k=1}^{n} P(x_k) log_2 P(x_k),\ (i = 1, 2, ..., m'). \tag{2.18}$$

Here, $E(F_i)$ shows entropy of the i-th feature F_i, $P(.)$ depicts the probability of mass function of F_i and $\langle x_1, x_2, ..., x_n \rangle$ are the possible values of F_i. They also proposed two forwarding approaches, namely (i) node-disjoint routing and (ii) delegation-based routing. In node-disjoint forwarding, feature difference with the destination node was resolved in a step-by-step manner (i.e., one feature at a time) to forward the message. In the delegation-based routing method, feature closeness with the destination was checked to select an appropriate relay node. However, this type of approach is difficult to implement in a decentralized environment as there is no fixed global entity to collect feature information from everyone and then identify important features for further processing. In addition to proposing a social attribute based group formation method (Section 2.2.1.2), SPOON [33] also proposed a social attribute based forwarding approach. It used interest similarity and meeting frequency among participants for taking forwarding decisions, arguing that people with similar interests tend to meet each other more frequently than others. Whenever a node has a message to forward, it calculates the fitness function of the neighbors using the following equation:

$$\text{Fitness score, } \mathbf{F} = \varphi_{own}\, sim(v_{dest}, \tilde{v_u}) + (1 - \varphi_{own})\, sim(v_{dest}, v_{H_u}) \tag{2.19}$$

Here, $\tilde{v_u}$ represents u's own interest vector while v_{H_u} represents the interest of other nodes seen by node u in the past. $sim(v_{dest}, \tilde{v_u})$ measures node u's interest similarity with the destination and $sim(v_{dest}, v_{H_u})$ shows the interest similarity of node u's expected future neighbors with the destination. Whenever a node meets a neighbor with a higher fitness score (**F**), it forwards the message to that node. However, it can be argued that people with similar interests might form a community but they might have different activity levels inside the community. To address this, Li *et al.* proposed a Local Activity and Social Similarity (LASS) based forwarding technique [66]. They suggested that the activity level of a node varies in different groups. Therefore, both social similarity and local activity within a group should be used to calculate forwarding utility.

Social aware networking (SANE) [111] is another recent work that considered interest as a social attribute of the users and interest similarity among them as a metric for selecting the forwarder node. The interest profile of a user was represented using an m-dimensional vector where each component indicated a user's level of interest in a topic and m represented the total number of interests (i.e., topic). A cosine similarity metric was used to identify interest similarity among two encountered nodes. Finally, the node with higher interest similarity with the destination was selected as the forwarder node considering that the interest of a node also dictates its movement in the network.

Another interesting work was recently proposed by Yuan *et al.* in [53]. They used both betweenness centrality and social similarity along with a personality value to calculate a forwarding utility, and finally used this value to select a relay node. Unlike previous works, [53] calculated the above social attributes from GPS traces considering that people visit a particular set of hotspots more regularly. The top-k hotspots were mutually identified by collaborative exchange of visiting records among users. Nodes who visited popular hotspots more often attained higher centrality values while

Protocol name	Metric for utility calculation	Community	Energy of relay	Buffer space of relay	No of Message copies
Simbet [26]	betweenness centrality and similarity	×	×	×	single
SimbetTS [107]	betweenness centrality, similarity and tie strength	×	×	×	multiple and single
PeopleRank [27]	popularity	×	×	×	single
LABEL [28]	similar affiliations	×	×	×	multiple
BUBBLE RAP [100]	global and local popularity	k-clique distributed	×	×	single
Beeinfo [52]	density and tie-strength	interest based	×	×	multiple
SocialGreedy [109]	social similarity tie-strength	×	×	×	single
Hypercube [110]	social feature and social closeness	×	×	×	multiple
SPOON [33]	interest similarity	interest similarity and meeting frequency	×	×	multiple
LASS [66]	social similarity and activity	geographic proximity	×	×	multiple
SANE [111]	interest similarity	×	×	×	multiple
Hotnet [53]	social similarity, betweenness centrality personality	×	×	×	single

Table 2.4: Social attribute based message forwarding approaches

nodes with similar visiting patterns achieved higher similarity values. In addition, the individual's habits for visiting these hotspots were taken into account to calculate a personality value. Finally, a utility value called a 'Hotnet' metric was calculated considering all the above components. The Hotnet metric of node u for destination v is calculated as,

$$Hotnet_{u,v} = per_u * gra_{u,v}. \tag{2.20}$$

Here, per_u shows u's personality score and is calculated using Shannon entropy.

$gra_{u,v}$ shows the gravitation between node u and v and calculated as,

$$gra_{u,v} = \mathbf{G} \frac{B_c^u B_c^u}{Sim(u,v)^2}, \tag{2.21}$$

where $\mathbf{G}$ represents the gravitational constant, B_c^u shows the betweenness centrality of u and $Sim(u,v)$ depicts the similarity between u and v. The performance of Hotnet was assessed using two real world datasets and compared against Simbet [26] and PeopleRank [27]. Simulation results indicated that Hotnet achieved a higher success rate (40-80%) than Simbet (38-60%) and PeopleRank (22-62%).

Some key features of the proposed social attribute forwarding approaches are highlighted in Table 2.4. Social attribute based forwarding approaches require extraction of social metrics from existing social networks or building them over time from regular social contexts. Therefore, they are more appropriate in scenarios where such relationships already exist through available social networks or can be easy to understand in *work-place* type scenarios among classmates or colleagues. However, these approaches usually do not consider the current physical context of participants (e.g., instantaneous movement or current location) which might provide faster delivery options. To address this issue, a group of forwarding protocols are proposed, which are discussed below.

Physical Context Based Forwarding Approach

This group of forwarding protocols analyzes the current physical context, such as movement direction, location, mobility pattern and current time, to assign a utility value for forwarding a message and accordingly selects a relay node. MV routing [112] is one of the earliest works in this category that used frequency of visits to particular locations to assign a utility value. Static destinations are considered in this work and nodes maintain a probability metric for visiting those destinations which is calculated by counting the number of past visits within a particular time period. For example, node k maintains a probability metric $P_0^k(R_i)$ depicting its probability of delivering a message in region R_i which is calculated as, $t_{R_i}^k/t$, that is the ratio of the number of rounds k visited region R_i to the total number of rounds. If k visited region R_i higher number of times, it is likely that it will visit that place again. However, assumptions of such static destinations are unrealistic in DCS. To address this issue, Leguay *et al.* proposed another mobility pattern based forwarding approach called MobySpace [23].

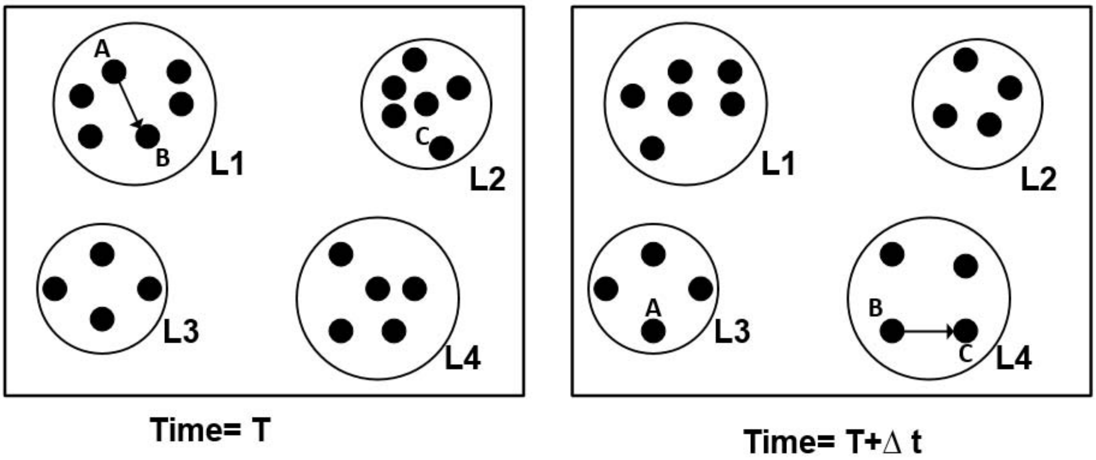

Figure 2.11: Predict-and-Relay (PER) forwarding adapted from [24]

In this approach, participating nodes maintain the history of their past visits considering the frequency and duration of visits to particular places. They identified that it follows a power-law distribution which essentially indicates that there are only few locations that people visit more frequently. In this approach, each node maintains a K-dimensional metric called MobySpace to record the list of their top K visited locations and the probability of visiting them. The similarity between the mobility patterns of two nodes is calculated using the Euclidean distance. Finally, a node with higher similarity value with the destination is selected as the relay node.

Although the similarity between mobility patterns indicates that the nodes are likely to meet again, it does not guarantee that these nodes will meet soon to enable faster delivery. Therefore, Huang *et al.* [113] proposed a technique where the distance and the moving direction of a node are taken into account. In this method, a node that has a smaller distance with the destination and is moving towards it is selected as the relay node. A message dropping policy is also proposed where messages that have traveled more hops are dropped first, if the buffer becomes full. A similar distance based forwarding approach is also proposed in [114, 115]. However, these approaches mostly consider static destinations or the location of the destination is known *a priori*, which is usually not the case for DCS.

Predict-and-Relay (PER) [24] is a prominent message forwarding technique that uses predictable movement patterns to identify a relay node. This approach considers that nodes follow a semi-deterministic trajectory by visiting a particular set of locations called landmarks on a regular basis and a time homogeneous semi-markov model was employed to illustrate the transition of nodes from one landmark to another. The transition probability between landmarks as well as the sojourn time in a particular landmark were calculated using historical data. Whenever nodes encountered each other, such information was exchanged so that nodes were aware of the movement patterns of other nodes in the network. Finally, a message forwarding utility was calculated considering the contact probability between nodes and message delivery delay. The approach is demonstrated in Figure 2.11 [24]. In this figure, node A is the source and node C is the destination. Based on the forwarding utility, node A forwards the message to node B at time T in landmark $L1$, who ultimately delivers the message to node C in landmark $L4$ during time $T+\Delta t$. Note that nodes have changed position due to movement from time T to $T+\Delta t$.

A similar movement pattern based forwarding approach was also proposed in [116] where nodes shared their own mobility pattern as well as the mobility pattern of their neighbors (reachable within 2-hop) when they encountered other nodes. A confidence value for movement history indicating the recency of the information was also used. The above approaches require maintenance of mobility history of all the encountered nodes which introduces message overhead and consumes significant network resources. To solve this, Talipov *et al.* [25] proposed an interesting approach where each node only maintains its own mobility history and does not share this with its neighbors. While generating a content request, a node calculates its own mobility information that includes future locations at different time intervals and adds this information with the request. This mobility information is used by a node to calculate message delivery probability (i.e., utility). For example, if the mobility information of destination node v and another node u is represented by $M_v = l_{v,t}, l_{v,t+\delta}, \ldots, l_{v,t+k\delta}$ and $M_u = l_{u,t}, l_{u,t+\delta}, \ldots, l_{u,t+k\delta}$, respectively, then the utility value of node u for delivering a message to node v can be calculated as,

$$
\begin{aligned}
Util_u(v) &= \sum_{m=t}^{t+k\delta} U_m, \\
U_m &= \begin{cases} \frac{t}{m}, \text{ if} |l_{u,m} - l_{v,m}| \leq R_{u,v} \\ 0, \text{ otherwise}. \end{cases}
\end{aligned}
\tag{2.22}
$$

Here, $R_{u,v}$ represents the radio communication range of u and v within which they can detect the presence of each other and successfully communicate. The performance of DPD was assessed using trace driven simulation with 100 nodes for 72 hours where the movement of nodes was dictated using the working day movement model [117] and 10 nodes randomly generated content request in every 30 minutes. DPD achieved a delivery success rate of 80-85% in different scenarios.

Dsearching [118] is another recent work along a similar line that uses sharing of mobility information to enhance message forwarding services. In this work, the entire area is initially divided into sub-areas assuming that a particular sub-area can only contain at most one popular spot that is frequently visited by all nodes (e.g., library or dormitory in a university campus). Since mobile devices have built-in GPS, they can easily identify their locations and determine the sub-area they are currently in. Nodes maintain and update their visiting records to generate a mobility history. A set of nodes called 'host' nodes are selected for each sub-area that are more likely to stay longer in that region. Unlike [24], in this work mobility information is shared with only the host nodes in a sub-area. Mobility information of a destination node collected from such host nodes are used for reaching them.

Although sharing mobility patterns with other nodes provides a faster way to deliver contents, it also raises privacy concerns. To overcome these issues, Lu *et al.* [119] proposed that nodes should only record their own mobility pattern without sharing it with other nodes. They also considered that mobility patterns vary across different time slots due to variation of movement during weekdays and weekends. Using this information, each node calculates its own utility value for forwarding a message to a particular destination. In this case, only the identity of the destination is used to calculate the utility value and a node with higher utility value is selected as the relay node. ALERT [120] is another protocol that provides anonymity protection by hiding node identity and routes from outsiders. It provides identity protection by using a collision resistant SHA-1 hash function to hash a node's MAC address and current time to generate its identity rather than simply using its MAC address. This way an outsider is unable to determine whether a particular MAC address holder is present in the network. For message forwarding, it dynamically selects an intermediate node in every step of the forwarding. To achieve this, ALERT performs hierarchical zone partition

Protocol name	**Metric for utility calculation**	**Community**	**Energy of relay**	**Buffer space of relay**	**No of Message copies**
MV [112]	visiting frequency	×	×	×	multiple
Mobyspace [23]	similar visiting pattern	×	×	×	single
DAER [113]	distance and moving direction	×	✓	×	multiple
POR [114]	distance and message size	×	×	×	multiple
MPAD [115]	speed, current moving direction and distance	×	✓	×	multiple
PER [24]	landmark transition and sojourn time probability	×	×	×	single
GeoDTN [116]	visiting pattern and confidence score	×	×	×	multiple
DPD [25]	movement pattern similarity	×	×	×	multiple
LOOP [119]	mobility record including location, time slot and day type	×	×	×	multiple

Table 2.5: Physical context based message forwarding approaches

that basically continuously partitions the whole area into alternative horizontal and vertical partitions until the destination node is not on a separate zone. After that, a temporary destination is selected which is a node in the same zone as the destination and a message is forwarded to that node.

Table 2.5 represents some key features of the physical context based approaches. These approaches require the existence of regular movement patterns of participating nodes, where they visit some set of locations on a regular basis and spend similar amounts of time in them. In our regular life, such routine movements are easy to realize as people usually visit their home, office or university campus following some predictable schedule. However, learning such predictable patterns requires a learning period and a regular or semi-regular schedule to be followed by the participants.

In summary, flooding based message forwarding techniques blindly spread multiple copies of a message in the network and hence are likely to consume unnecessary resources. In contrast, utility based message forwarding techniques require frequent encounters on a regular basis, social relationships among participants built over time or scheduled movement patterns to calculate the utility value and select an appropriate relay node. Although these approaches are appropriate for *work-place* type scenarios, they are ineffectual for irregular meeting places, such as tourist spots or camping sites, as nodes are expected to demonstrate spontaneous random movements and will meet strangers with whom they might not have any social ties. Most of the existing approaches do not consider the remaining energy or buffer space of the relay node before selection, which is an important indicator of the node being capable of successfully carrying and delivering the content. In addition, the above approaches assume that nodes are cooperative and help each other for carrying and delivering contents. However, in real life, selfishness is common among participants and some sort of incentive is needed to encourage them. To address this, a number of incentive schemes are proposed in the literature that are discussed in the following section.

2.2.3 Participation Incentive

The message forwarding approaches discussed in the previous section rely on the altruistic behavior of participating nodes suggesting that they work in a collaborative manner for carrying and delivering contents for other nodes. However, such altruistic behavior might not be found in reality as users are more likely to act in a selfish manner without some form of incentive. The reasons behind such selfish behavior may be due to the nature of the use of the device, saving battery life, or the privacy concerns of the users about taking part in the sharing process. To handle such selfish behaviors, different incentive schemes have been proposed in the literature to induce users to participate at different phases of the content sharing process. The proposed incentive schemes can be broadly categorized into: (i) reputation based, (ii) credit based, (iii) tit-for-tat and (iv) game-theory based schemes, which are discussed in the following sections.

2.2.3.1 Reputation Based

This type of scheme monitors the behavior of participating nodes, identifies well-behaved and misbehaving nodes, and accordingly calculates a reputation score. Whenever any node forwards a packet for another node, the reputation score increases and a higher reputation score ensures priority for obtaining a service. In contrast, whenever a node misbehaves, its reputation score decreases and when the score falls below a certain threshold, that node is usually excluded from the network.

He *et al.* [121] were among the first to employ a reputation based incentive mechanism. They used neighbor monitoring in which a node can overhear the transmission of a forwarder node and identify whether it has forwarded a message or not. The reputation score of a node is calculated using the ratio of the number of messages actually forwarded by that node to the number of total messages sent to it for forwarding. Reputation scores are also shared in the neighborhood to employ indirect observation. Finally, messages generated from misbehaving nodes with lower reputation scores are not forwarded by their neighbors to punish them. Although this scheme provides a basic framework for employing reputation based incentive, it fails to address the fact that in this kind of network nodes are likely to move away and become unable to overhear the transmission to identify successful forwarding. Consequently, some nodes will not be unduly punished even if they provide the service. Uddin *et al.* [122] addressed this issue and proposed that the destination node can send a delivery report for successful delivery that includes a list of all forwarder nodes. Such reports are flooded in the network to help everyone to assign a ranking (i.e., reputation score) for the forwarder nodes. A further improvement is proposed in [123], where an intermediate forwarder node sends a positive feedback after receiving a message. Nodes also exchange their list of encountered nodes in the past. Finally, this list along with the feedback message is used to identify nodes that had an encounter opportunity but did not forward. A similar scheme is presented by Bigwood and Henderson [124], where nodes exchange their list of encountered nodes. Upon meeting the destination, this allows a sender node to identify a misbehaving forwarder node that did not forward a message. To assess the performance of their approach, the authors performed a simulation with a content size of 1 MB and the request lifetime was varied from 2 hours to 10 days.

Across different scenarios, this method achieved delivery success rates within 80-85%.

The above schemes suggest that nodes should monitor their neighbors and maintain a reputation score for them, which requires excessive message passing among neighboring nodes. The amount of message overhead also increases to maintain a consistent score across all nodes. To this end, MobiID [125] suggested a more user-centric approach where nodes maintain their own forwarding evidence and demonstrate it when required. In this case, whenever a node forwards a message for another node, it receives a reputation certificate which it later uses as a proof of delivery. This allows a node to keep track of its own reputation score. Along a similar line, Mei and Stefa [126] suggested that a forwarder node should forward a message to at least two other nodes in the network and keep the proof of delivery. Upon meeting a sender node, a forwarder node should show the proof of delivery or of still having the message in the memory in case it did not have any opportunity to forward. If a forwarder node fails to provide any proof, the sender node reports the issue to a central authority who punishes the misbehaving node by excluding it from the network. However, maintenance of such central authority is an issue with this approach.

Another interesting work is presented in [127], where nodes mostly rely on their individual experience to determine the selfishness of a node rather than reports received from their neighbors. In this case, nodes only consider whether other nodes have forwarded their messages rather than their neighbors' messages. It is argued that a node may behave differently with different nodes due to its underlying social relationship (e.g., more helpful towards a friend than a stranger), and hence personal experience should be given more weight than indirect observation reported by other nodes. This work also considered that a forwarder node might decide not to forward a message due to low battery and thereby should not be punished for this. However, this method requires extraction of social relationships from online social network information or frequent encounters among nodes to identify the same, which is unavailable in our scenario. In addition, the list of past forwards must be exchanged, which is likely to create message overhead in the network. Furthermore, a node might cheat to gain unfair benefits by providing false claims of successful message forwarding, which is not addressed in this work.

Overall, a reputation based incentive scheme requires direct and indirect observation from participating nodes to be shared in the network to maintain a reputation score. However, without the presence of a central authority, such schemes will result in inconsistent reputation scores for the same node as not everyone receives the same report due to intermittent connectivity. Furthermore, an aggressive punishment scheme will discourage nodes to participate at all in the sharing process. Therefore, a credit based scheme is proposed in the literature where a central server provides an appropriate incentive to the participants for their service, which is presented in the next section.

2.2.3.2 Credit Based

A credit based incentive scheme employs the notion of virtual currency or credit for rendering the message forwarding service. Forwarder nodes earn some credits for participating in the successful delivery of a message, which are paid by the sender node. A node can also utilize its earned credit to obtain some services in the future. A generic assumption in this approach is that a Trusted Third Party (TTP) is available for credit management. Another issue is different types of cheating attack by selfish nodes to gain undue advantage [128].

SMART [40] is a multi-layer credit-based incentive scheme that uses an offline security manager for key distribution and a Virtual Bank (VB) for credit management. It is based on the idea of a layered coin approach where the sender node generates a base layer and forwards the message. The intermediate forwarder nodes generate an endorsed layer using the previous layered coin. Each node periodically submits its collected layered coins to the VB to obtain credit that verifies the submitted coins, calculates credit for the forwarder nodes and accordingly charges the sender. The amount of credit is calculated using the size of the message. In this scheme, credit is only given for successful delivery. Lu *et al.* [41] further extended this and suggested that nodes should also be given some incentive for unsuccessful delivery in the form of reputation, where the reputation score of a node increases if it participates in an attempted but unsuccessful delivery. It is also shown that nodes with a higher reputation are more likely to get help from other nodes. The amount of reward for forwarding a message is calculated based on the distance it is carried by a node.

A similar VB based credit management scheme is employed by Chen *et al.* [129]. They proposed two incentive schemes, namely earliest path singular rewarding (EPSR) and earliest path cumulative rewarding (EPCR). The former gives rewards to the forwarding nodes that participate in the earliest delivery path (i.e., delivery within minimum time), while the latter gives reward to nodes for their participation in any delivery path. The reward is given considering the time a forwarder carries a message before forwarding. In contrast, [130] considers that the reward should be inversely proportional to the total hop count in the delivery path. A multi-receiver scenario is considered in this work where multiple nodes are interested in receiving a single message. Nodes try to select a forwarder who is more likely to meet more receivers for benefit maximization. The performance of this approach was also evaluated using MIT reality mining and Haggle dataset where the average size of the content was set to 250 Kb. This approach achieved a delivery success rate of 60-85% across different scenarios.

Recently, a community based incentive scheme was proposed in [131] where nodes are divided into communities based on their social relationships. Two types of credit, namely social and non-social credit, are employed in this work. Social credit is obtained for relaying the data of the community members while the other is gained for carrying non-members' data. Similarly, a node can use its earned social credit for intra-community forwarding and the non-social one for inter-community message forwarding.

In summary, the credit based incentive schemes employ a virtual bank or a central server for credit management. Such a virtual bank or central authority is difficult to establish for decentralized sharing as the considered scenario lacks Internet connectivity to maintain uninterrupted connection. Therefore, reciprocity based incentive schemes are proposed, which do not require a central authority to provide a management service and consider that nodes show reciprocal behavior and only help those nodes who help them in return. Such reciprocity based incentive schemes are discussed in the following section.

2.2.3.3 Tit-for-Tat

A Tit-for-Tat (TFT) scheme considers that the amount of service provided to a node should be equal to the service received from that node. In the case of message forwarding, every node forwards as much traffic as the other node has forwarded for it. This is a popular technique in wired networks but difficult to employ in a decentralized setting as there is no central authority to monitor node behavior to provide administrative control. Another issue is the bootstrapping and cold start problem as the scheme needs to handle the situation when two nodes meet for the first time.

Shevade *et al.* [132] proposed a pair-wise TFT scheme for rewarding well-behaved nodes. Instead of keeping track of the misbehaving nodes, this scheme keeps track of the good behaviors of a node that helped in forwarding in the past. In this case, if node u carries some packets for node v, it expects that v will also carry the same amount and otherwise stops cooperation. To address bootstrapping issues, this work considers generosity, where a node initially gets some free service at the beginning. However, both nodes might not be the most appropriate forwarder for each other, and hence are likely to have unequal service to offer which makes the scheme ineffective. To address this, Krifa *et al.* [54] proposed a trading based TFT scheme for buying, storing and trading contents where nodes also carry contents for other nodes they are expected to meet in future. A node uses the history of past request patterns and the list of encountered nodes to identify the contents it needs to carry to maximize the utility value during future encounters. A similar trading based technique is also proposed in [59], where encountering nodes initially obtain contents reflecting personal interest and use the remaining time to obtain contents reflecting the interest of the neighbors they are expected to meet in future. The utility value for carrying a particular content is calculated considering the number of interested nodes expected to be met before the TTL expires.

Li *et al.* [133] were the first to address Social Selfishness (SS) suggesting that nodes are more willing to provide service to other nodes with whom they have stronger social ties. In this scheme, nodes consider their social tie with the previous hop from whom they receive a message or with the sender of a message to determine their willingness to further carry it. The idea of such social selfishness is also employed in [134–136].

Work in [134] employs social selfishness along with similarity between users' profiles to determine whether to carry a message for another node. In contrast, Yu *et al.* [135] argued that resources available at a node also dictate the decision of carrying a message as node u might have stronger ties with another node v, but resource scarcity (e.g., low battery or storage space) at u will make it not carry any message. Another interesting work proposed by Liu *et al.* [137] suggested that a node should consider its social relationship with a community rather than individual members to decide whether to carry a message. The idea is that if a node u helps any members of a community by carrying its messages, all the members of that community will also carry its messages in return. The performance of this approach was analyzed using trace driven simulation where the message size was varied within 500-1024 KB and the simulation results indicated that a delivery success rate of 80-85% was achieved in different scenarios.

A service priority based incentive scheme is proposed in [138] where a secure management authority (SMA) is considered for the allocation of such priority. Nodes can send their service claim to SMA after providing a service to another node. The SMA assigns new priority for the provided service and broadcasts this value to update everyone. The role of SMA is similar to the VB used in a credit-based scheme and hence difficult to establish in a decentralized sharing.

Although the proposed TFT schemes ensure fairness as nodes only get the amount of service they provide to others, they also require repeated encounters, which are unlikely to be available in irregular meeting places. In addition, calculating content utility in a distributed manner is also difficult since the encounter information available at a node may only give a partial view and lead to inaccurate calculation.

2.2.3.4 Game Theory Based

This category of incentive scheme employs game theory to handle selfish nodes, considering them as rational players who try to maximize their own profit. Nodes try to select a strategy for gaining personal benefit and hence the proposed mechanism tries to reach Nash equilibrium where no node can unilaterally change its strategy to gain unfair benefits. Most of the game theory based schemes adopt a credit based method while a few use reputation based incentives and Tit-for-tat.

Li *et al.* [139] were among the first to employ a game theoretic model to deal with node selfishness. They considered a principal-agent model where both the sender and the receiver of a message were termed as principals who were interested in exchanging a message through the intermediate relay nodes termed as agents. At the beginning, the sender collects the cost of sending a message from the relay nodes and selects the path with the lowest cost. In this scenario, the authors explored a hidden information game where the cost of carrying a message is only known to agents, and a hidden action game where the principal is unable to determine whether the agent intentionally dropped a message. To address these issues, they proposed a strategy-proof payment scheme that made the agents truthfully declare their cost and reached a sub-game perfect equilibrium suggesting that the relay nodes had to deliver their promised quality of service to get paid. However, their assumption of static nodes makes this approach ineffective for a dynamic network with node mobility.

Mobicent [42] is another credit based game theoretic model that addresses the path selection from a source to destination for downloading a content. It uses TTP for verification and payment, and considers that helper nodes help mobile clients to download their content and get paid for their service. Initially, a mobile client contacts TTP to determine the cost of download. After receiving information about available paths, a client decides whether to minimize transmission cost or delay using a path auction game. Finally, the mobile client receives the content in an encrypted format and can only decode it by contacting the TTP who provides the key after the payment.

In contrast, [140] and [141] employ a reputation based and TFT based scheme, respectively using a game theoretic model. Work in [140] proposes a user-centric scheme where nodes can store their own reputation and display it as a proof when needed. It uses a Bayesian game approach to design reasonable cost and reward to ensure fairness. Buttyán *et al.* [141] devised a message exchange scheme among encountering nodes as a non-cooperative game and showed that the overall message delivery rate can be improved if nodes follow Nash equilibrium. In this case, the exchange occurs in a message-by-message manner where node u transfers a message to node v and waits until it gets another message in exchange from v before transferring the next one.

Ning *et al.* [142] studied the data dissemination issues in DTN and proposed a credit based incentive scheme using a two-person cooperative game. They employed direct and indirect interest and considered that nodes are willing to pay for contents related to their direct interest. Since multiple nodes might be interested in the same content, this scheme calculates the expected reward and utility for carrying a content using past encounter history to determine the number of nodes expected to be met who have interest (both direct and indirect) in this content. When two nodes meet with each other, they try to maximize their utility using Nash equilibrium. They further extended this idea in [143] and proposed a technique to encourage node participation for ad dissemination in MSN. This method assumed that an ad provider can issue a virtual check for propagating advertisements which are carried by participating nodes for getting virtual credits. When an intended receiver receives such an ad for the first time, it signs the check, which is cashed by the relay node to earn virtual credits from the ad provider. Such virtual credits can be later used by a node for disseminating its own ad. Similar to [142], this scheme also used a two-player cooperative game for nodes to exchange contents and virtual checks upon encounter. In this regard, nodes consider Ad Category Contact Likelihood (ADCL) and Check Reward Contact Likelihood (CRCL). The former calculates the probability of node u meeting the nodes who would be interested in a particular ad, while the latter calculates the probability of meeting the ad provider. Nodes are only allowed to trade contents in exchange of contents and virtual checks for the same. Therefore, nodes try to exchange those contents and checks that will maximize their benefits in terms of their respective ADCL and CRCL values. Finally, the exchange process is solved using a Nash bargaining solution.

The idea of social selfishness was recently studied in [144, 145]. Xia *et al.* investigated a signaling game approach to study the impact of uncertain cooperation among cooperative and socially selfish nodes. They utilized Bayesian Nash equilibrium to analyze the initial interaction between encountering nodes and a perfect Bayesian equilibrium to determine their response strategies during subsequent communication. In contrast, [145] applied a bargaining game among nodes to buy the forwarding service of each other using virtual currency. This method also incorporated the idea of reputation and Tit-for-tat as the buyer with higher reputation gets some discount for buying a service and nodes were forced to follow a ‘give one to get one’ policy.

Table 2.6 highlights some of the key features of the existing incentive schemes.

Method name	**Reward type**	**Punishment strategy**	**Calculation metric**	**Technique used**
SORI [121]	×	service disruption	Percentage of forwarded message	neighbor monitoring and indirect observation
RELICS [122]	message priority	×	number of forwarded message	delivery report flooding
RADON [123]	×	×	number of forwarded message	direct observation and indirect monitoring
MobiID [125]	×	blacklisting	number of forwarded message	sharing reputation information
SMART [40]	virtual credit	×	TTL and earliest delivery path	profit sharing
Chen *et al.* [129]	virtual credit	×	contribution time earliest delivery path	TTP based monitoring
MuRIS [130]	virtual money	×	hop-count	path information sharing
MobiTrade [54]	×	×	size of content	content trading
SIS [138]	service priority	×	number of bundles forwarded	SMA based monitoring
SID [142]	virtual currency	×	message matching user's interest	two-person cooperative game

Table 2.6: Incentive schemes to encourage participation in DCS

Overall, the incentive schemes available in the existing literature require an authority to monitor node behavior in reputation based schemes while the credit based schemes mostly depend on the availability of a trusted third party server. In contrast, a TFT scheme requires frequent encounters to ensure fairness. Therefore, employing a proper incentive scheme in decentralized sharing remains a challenging issue.

Some of the above-mentioned schemes already employ a misbehavior detection method to protect against malicious users to ensure that they do not get any unfair benefits. Additionally, there are some approaches that specifically address trust management issues in content sharing and they are discussed in the next section.

2.2.4 Trust Management

In decentralized content sharing, misbehaving nodes are those who try to gain unfair advantage from others without actually providing any service in return. In this regard, we use the term 'trust' to indicate the reliability of a node for providing a service or making a claim for providing a service. For example, a trustworthy node is expected to forward a packet when its energy and contact opportunity permits. Again, a trustworthy node will only make an honest claim for the services it has provided. In contrast, a misbehaving node will try to cheat the system by intentionally dropping messages, although it has the opportunity and capability to forward. Additionally, misbehaving nodes can also make false claims for some services in which they did not participate at all. The purpose of a trust management scheme is to identify such trustworthy and misbehaving nodes as the misbehaving nodes, in addition to making false claims, may forge some other parameters to become an administrator of a group and start spreading malicious contents. Since there is no centralized authority in DCS, trust management emphasizes detecting misbehaving nodes collaboratively. Existing misbehavior detection schemes are explored below.

2.2.4.1 Misbehaviour Detection

Misbehavior detection is mainly performed through a node monitoring or watchdog approach. In this case, nodes usually observe the behavior of their neighboring nodes to determine their trust level and assign a trust score. A node with a lower trust score is considered as a misbehaving or selfish node and punished by its neighbors. Punishment is usually provided in the form of service disruption or blacklisting.

SReD [146] is an early work for misbehavior detection where each node uses its own experience and information from its neighbors to determine the nature of the surrounding nodes. A node calculates its neighbor's trust score using a local trust value that the node itself calculates, and a local reputation value that is reported by other neighbors. The local trust value is assigned according to the forwarding behavior and cryptographic operation capability. It is assumed that a node can observe its neighbors' forwarding behavior in promiscuous mode and determine the fraction of the messages forwarded by that neighbor. The cryptographic operation capability is assessed by checking whether a received cipher text from a neighbor can be correctly decrypted using a symmetric key. Information about a neighbor's trust score is also collected from other neighbors and considered as local reputation value. Finally, the local trust and reputation value is used by a node to calculate its neighbors' overall trust score. However, in decentralized sharing, the forwarding operation might be delayed because of intermittent connectivity and it is not always feasible to monitor the forwarding behavior of a neighboring node. Therefore, Li *et al.* [147] proposed a positive feedback message based approach to calculate trust. After forwarding a message, each node sends a feedback message to the previous hop that uses this to monitor forwarding behavior. This approach employs both direct and indirect observation to calculate belief, disbelief and uncertainty about the forwarding behavior of a node. Belief indicates node u's belief about the good forwarding behavior of node v and is calculated as,

$$Be_{u,v} = \varphi_{dir} Be^{dir}_{u,v} + \varphi_{ind} Be^{ind}_{u,v}\,, \tag{2.23}$$

where, $Be^{dir}_{u,v}$ indicates u's belief about the good forwarding behavior of v obtained from direct observation while $Be^{ind}_{u,v}$ represents the same obtained from other neighbors. φ_{dir} and φ_{ind} show the weight of direct and indirect observation, respectively. Similarly, disbelief indicates u's disbelief about the good forwarding behavior of v and calculated as,

$$Db_{u,v} = \varphi_{dir} Db^{dir}_{u,v} + \varphi_{ind} Db^{ind}_{u,v}\,, \tag{2.24}$$

where, $Db^{dir}_{u,v}$ and $Db^{ind}_{u,v}$ show u's disbelief about the good forwarding behavior of v from direct and indirect observation, respectively. Since positive feedback messages can be lost due to long delays or network partition, an uncertainty metric is also used to indicate that nodes are unsure about the behavior of another node, which is calculated as,

$$Un_{u,v} = 1 - Be_{u,v} - Db_{u,v}\,. \tag{2.25}$$

Finally, these three metrics are used by node u to calculate v's trust score as,

$$Trust_{u,v} = Be_{u,v} + \sigma_{ra} Un_{u,v}\,, \tag{2.26}$$

where, σ_{ra} is a weighting factor that relies on the principle of insufficient reasoning and, by default, is initialized as 0.5.

A similar behavior monitoring based approach is also investigated in [148–150]. Salehi *et al.* [148] used the idea of a positive feedback message as in [147], and further proposed that nodes can periodically broadcast their trust value, which will essentially help other nodes to update their trust score. Ahmed *et al.* [149] suggested that nodes can calculate the expected number of messages from their neighbors in a particular period and accordingly classify them as cooperative and selfish nodes based on their actual contribution. In [150], the authors proposed that while forwarding a message, nodes can add their IDs and sign it with their individual keys and the destination node can check against this signed list to verify if everyone actually participated in the forwarding process. The destination node also sends an acknowledgement message, including the list of forwarder nodes, which is used by everyone to update their trust values. In contrast, [151] suggests that nodes can keep their own record of forwarding behavior and exchange it with other nodes upon encounter. From this record, a node can decide about the selfishness of other nodes. Additionally, this approach also considers that nodes might be unable to forward a message because of low resources, which should not be considered as selfishness. However, a malicious node might cheat the system by generating fake records.

As alluded to before, misbehaving nodes can also make false claims or forge forwarding evidence to indicate their participation. To address such cheating behavior, [152] assumed the existence of a Trusted Authority (TA) who can verify the forwarding evidence. It considers that the TA can periodically arrive in the network and probabilistically check a subset of forwarding evidence to determine if the forwarders are generating fake evidence or making false claims. Similar TA based misbehavior detection is also proposed in [153] where the Sybil attackers (i.e., nodes that change their identities to gain unfair advantage) are identified. However, such a trusted authority is difficult to establish and maintain in a decentralized setting without an Internet connection.

Recently, Waluyo *et al.* [154] proposed an interesting approach where each node maintains its own personalized trust score of another node. They argued that different nodes might be interested about different Quality of Service (QoS) metrics. For example, node u_i might be interested about transmission delay while node u_j is more interested about the quality of received content and hence they will assign different scores for the same forwarding action of node u_k. They considered four QoS, namely (i) transmission delay (T_d) (ii) accuracy of reported resource (A_r), (iii) quality (Q_r) and (iv) maliciousness of received content (M_r) to determine a node's trust. Since each node has its own QoS requirement, trust value of node v is calculated by other nodes as,

$$Ts_v = \frac{\sum_{j=1}^{J} [T_d A_r Q_r M_r / 4]}{J}, \tag{2.27}$$

where, J shows the number of services provided by node v as perceived by other nodes. However, this approach requires a node to contact all of its neighbors before assessing another node's trust score before each transaction, which will introduce significant message overhead in the network.

In summary, the above-mentioned approaches use behavior monitoring to identify misbehaving nodes where nodes share their experience with other nodes by flooding the network. Frequent exchange of such messages will essentially increase message overhead and create network congestion, while infrequent exchanges might lead to delayed detection. In addition, an attacker node might propagate false recommendations to gain an unfair advantage.

2.2.5 Content Replication and Cooperative Caching

Content replication and cooperative caching strategies are adopted to enhance the content delivery service by increasing content availability and reducing delivery latency. The basic idea is to identify potential requests and proactively push matching contents near the prospective requesters. Existing strategies can be broadly categorized into: (i) community independent strategies and (ii) community-based strategies. Community independent replication strategies do not consider the existence of community, rather nodes take decisions whether to replicate a content considering personal and neighbors' benefits. In contrast, community-based strategies consider the existence of communities among participating nodes and propose a replication policy for helping community members to receive requested contents. Both of these are discussed in the next section.

2.2.5.1 Community Independent Strategies

The concept of community membership is absent in this category. Nodes are considered as individual entities and they replicate contents based on their observed request patterns or that of their neighbors. Replication is performed in order to reduce overall content access delay and increase content availability.

PodNet [155] is one of the first community independent replication strategies that suggest that nodes should divide the available buffer space into private and public space. The private buffer space is used for storing contents of personal interest, while the public buffer space is utilized for replicating contents that might be of interest to future encountering nodes. Upon contact, nodes first exchange contents reflecting their personal interest and then use the remaining time for replicating contents in the public buffer space. To determine the contents to be replicated in the public buffer space, four solicitation strategies are proposed, namely (i) most solicited, (ii) least solicited, (iii) uniform and (iv) inverse proportional. The first strategy replicates the most popular contents while the second one selects the least popular ones to increase diversity. The popularity of contents is calculated considering the past request patterns observed by a node. In uniform solicitation, nodes randomly select a content using uniform distribution while the fourth strategy suggests that the solicitation probability should be inversely proportional to content popularity. Ma *et al.* [156] further extended this work by considering meeting probability with the potential requesters along with their personal preference. They considered that movement patterns of participating nodes are non-random in nature and nodes that spend longer time together are expected to meet more often, which is then used to calculate a meeting probability. In addition, nodes broadcast their request summaries and personal preferences towards a particular content. Finally, the replication is performed jointly considering the probability of meeting a potential requester, content popularity (calculated from the request summary) and personal preference. An extended version of this work is presented in [157] where a pair-wise solicitation strategy is considered, suggesting that encountering nodes can jointly determine the optimal strategy for replicating a content.

A few other works have also used content popularity to identify the contents to be replicated [43, 44, 158–160]. Ioannidis *et al.* [158] proposed a voting mechanism based policy where each user independently determines the utility value of caching a content and reports this to its encountered node. The utility value is calculated using the number of pending requests observed for a content and the reports received from other nodes. Upon meeting an access point, a node replicates the contents with higher utility. Similar to this approach, Chen *et al.* [43] proposed that encountering nodes should replicate contents based on their priority value calculated in terms of the number of requests received and the size of the content. They also considered meeting frequency among nodes and available resources (buffer space) for replication.

Moualla *et al.* [159] proposed another interesting strategy where a node proactively pushes contents to its neighbours' cache. The potential contents are identified using their popularity, which is calculated using the number of copies available in the network. A higher number indicates higher popularity. The potential candidates for pushing the replicated content are identified using social proximity where a user with the same contents is considered more similar and given higher preference. In contrast, [44] suggested that the access points can proactively push popular contents in a user's device to reduce delivery latency. However, their simplified assumption of nodes with identical mobility, cache capacity and request pattern is unrealistic. Wang *et al.* [160] further improved this and proposed a mobility aware content replication strategy where a node replicates a content from a nearby edge-network device (e.g., AP) considering the gain achieved by carrying it. The gain is calculated in terms of social popularity of a content and mobility pattern of the node. This method considers that contents have varying popularity in different regions. Therefore, the social popularity is calculated using this variation and the influence of the user who generated or shared this content. In addition, this method also uses past visiting patterns to determine the probability of visiting a particular region. However, they deployed a coordinate server for gain calculation, which is difficult to establish in decentralized sharing without an Internet connection.

Gao *et al.* [38] proposed a scheme for a Mobile Ad-hoc Network (MANET) where contents are replicated in multiple central locations called Network Central Lcations (NCL), which are selected based on a probabilistic metric that evaluates data transmission delay in delay tolerant networks. In this case, the central location only reflects easy accessibility by other nodes and does not indicate a centralized replication method; rather such multiple locations are selected for placing the replicated contents. The capability of node u for being a central location is calculated as,

$$C_u = \frac{1}{|\mathcal{V}|} \sum_{v=1}^{|\mathcal{V}|} p_{u,v}(t)\,, \tag{2.28}$$

where, $p_{u,v}(t)$ shows the average probability of a successful transfer from u to v within time t and $\mathcal{V}$ is the set of all nodes. Whenever a node generates some content, it replicates the content in the nearest NCL. If the buffer space at the NCL is full, then another node closer to the NCL in terms of hop-distance is selected as the temporary cache node. For content access, requests are forwarded to NCL, who delivers the content to the requester. Although this scheme ensures that the requester can easily access NCL for uploading and downloading contents, it also indicates that the NCL will be congested when multiple nodes simultaneously request contents.

Recently, Zhao *et al.* [161] proposed a contact duration aware data replication strategy suggesting that the encounter duration among nodes might not be long enough to replicate the whole content. Therefore, they proposed a packet-level replication strategy where a content can be divided into multiple packets and a node can selectively replicate a sub-set of such packets. The benefits of replicating a content are identified using its popularity, current availability and the probability of delivering it to other nodes. The popularity is calculated in terms of the number of pending requests observed for a content with respect to the total number of pending requests. Nodes share their available content list for calculating current availability and finally historical contact rates among nodes is used to determine delivery probability.

In summary, most of the proposed strategies use observed content request patterns or content popularity as the main criteria for replication. Although content popularity is an important indicator, other issues, such as a user's personal interest or preference for accessing contents, need proper consideration for accurate request estimation. In addition, the proposed approaches do not consider community membership among nodes, which is important in decentralized settings as nodes are more likely to form communities based on their mutual interest and eventually will only help fellow community members by replicating contents for them. Such community based replication strategies are explored in the next section.

2.2.5.2 Community Based Strategies

Community based replication strategies assume that nodes can be divided into different communities and the replication decisions should be taken considering the expected request pattern of the community members.

ContentPlace [37] is a prominent community based content replication strategy that suggests that a node may belong to several communities (e.g., working or family community). Each node visits such communities frequently, which can be used to determine the probability of visiting a particular community. The utility value of a content varies in different communities and should be calculated considering the visiting sequence of a node. In addition, the utility value of a content also depends on its access probability, availability and size. A node uses its personal interest to calculate local access probability. A binary value is used to show the access probability as 0 or 1. Whenever two nodes encounter, they first exchange their content list and local access probability values. Afterwards, they update the utility value of all the contents in their combined content list. If any node identifies that the other has some contents with higher utility than its currently owned contents, it replicates those contents. Considering the visiting sequence of a node, this method investigated five different strategies, namely (i) most frequently visited, (ii) most likely next, (iii) future, (iv) present, (v) uniform social. The first strategy gives more weight to contents that are popular in the frequently visited community while the second one considers the contents likely to be consumed in the community the node is expected to visit next. Similarly, the third one ignores the contribution of the present community in the utility calculation while the fourth strategy only considers content consumption in the current community. Finally, the last strategy assigns equal weight for all communities. However, identifying such a visiting sequence requires repetitive visits to those communities on a regular basis with a predictable pattern, which is difficult to establish in irregular meeting places, which are the focus of this thesis.

Zhuo *et al.* [45] proposed a community based replication policy which suggests that the contents should be replicated at nodes with higher centrality. The centrality value of a node was calculated using its inter-meeting time with other nodes of the same community. A node with higher encounter rates was assigned a higher centrality value. They proposed that when a node with lower centrality meets a node with higher centrality, it should replicate contents in that node. However, they did not suggest which contents should be replicated or how many replicas should be created. Along a similar line, Cabaniss and Madira [162] suggested that a node's ranking in a group depends on its encounter frequency with other nodes and their request patterns. If a node u meets other nodes frequently who generate a greater number of requests, u should be assigned a higher rank. Nodes with a higher rank are considered as the social repository or throwbox and carry contents for all group members. When a repository node meets another node with a higher rank, it transfers the whole repository (i.e., all contents).

GCC [163] is another community based strategy that maintains a single replica of a content considering the recency of the request. Whenever a node u receives a content for another node v as a response to v's previous request, node u checks whether any other community members have already replicated it and keeps this copy otherwise.

Overall, the community based replication strategies fail to capture content demand as they use partial information collected from encountering nodes and do not consider the user's interest and preference for accessing contents. They also do not consider the dynamic content demand, which changes over time. A few strategies in the existing literature consider such varying content demand and are discussed below.

2.2.5.3 Economic Modelling for Content Distribution

A few distribution strategies utilize content demand to determine a potential distribution strategy. For example, [164] and [165] calculate the evolution of demand for a content using the word-of-mouth propagation. Figure 2.12 shows the evolution of demand for a file presented in [165]. In this case, the demand generated for a content depends on the initial set of interested users in this content and the time elapsed since the content was generated. Both methods assume that each user generates identical demand, which is unrealistic as content access patterns vary across different users as their level of interest is different. In addition, these approaches assume that each user is equally likely to become interested even if they are initially uninterested in this category of content. This is again impractical as users are likely to demonstrate different

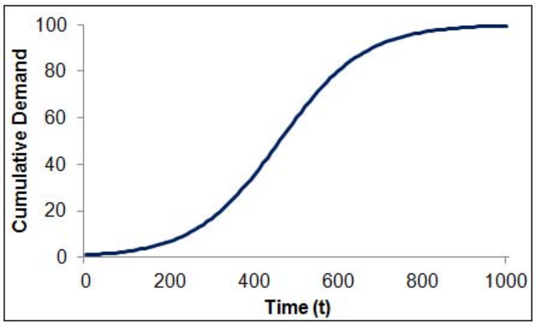

Figure 2.12: The evolution of demand for a file adapted from [165]

levels of affinity towards contents they are currently uninterested. Some of the users may become easily interested in a new topic while others might stick to their personal interest for accessing contents. In contrast, Meng *et al.* [166] calculated the demand for a content using node accessibility and the probability of receiving a request for that content. They have also proposed a population ecology based replication policy for an Internet based unstructured peer-to-peer network.

The above strategies are not applicable in decentralized content sharing as they rely on an Internet connection for content delivery, assume nodes to be always accessible if they are online and consider the presence of a static storage (i.e., server) to permanently hold contents. In contrast, Reich and Chaintreau [167] considered the decentralized setting, calculated content demand based on their popularity and performed replication based on the added utility a replica can provide in terms of delivery time. However, their assumption of identical demand and storage space across all nodes is impractical as alluded to before. Similarly, no consideration was given to content supply, which changes dynamically in irregular meeting places as nodes might leave the area after performing their activities.

2.3 Major Challenges to Employ Decentralized Content Sharing in Irregular Meeting Places

This chapter has highlighted the state-of-the-art decentralized content sharing techniques. They employ a routine movement pattern or frequent encounter among participating nodes to handle major issues in DCS. However, the focus of this dissertation is irregular meeting places where such visiting patterns are absent. Therefore, employing DCS in such an environment poses a number of significant research challenges, which are outlined below:

(i) The dynamic nature of human movement already imposes enough challenges for implementing decentralized content sharing approaches where a constant connection is unavailable. Adding a further dynamic element (e.g., irregular movement patterns) because of the sheer nature of the places or gatherings (e.g., tourist spot) makes it more challenging to form and manage groups to facilitate content sharing.

(ii) Delivering a requested content within a delay bounded time is difficult since this is a two-step process of locating the content and then delivering it to the requester. In this case, both the requester and the content holder might move during the transfer and nodes have very little or no idea about future encountering nodes to take a learned decision. Therefore, meeting acceptable delivery times in irregular meeting places (i.e., tourist attraction) is far more challenging.

(iii) Nodes might demonstrate selfish behavior, which is difficult to handle without any global observer or central authority. Again, a more drastic approach might discourage the nodes to participate in content sharing at all. The well-behaved nodes also require some sort of encouragement to continue sharing or participation. Therefore, a proper incentive mechanism is needed for sharing and relaying contents.

(iv) The presence of malicious nodes is also another important issue. Since nodes meet for a short time in a tourist attraction, it becomes very difficult to identify and populate the list of malicious nodes during that period. A simplistic yet effective solution is needed to make DCS acceptable to users.

(v) As highlighted in Section 2.2.5.3, the existing literature considers content demand to be identical across all users and uninterested users are equally likely to generate the same amount of demand. This is inappropriate in irregular meeting places as users of different backgrounds and interest levels come together and

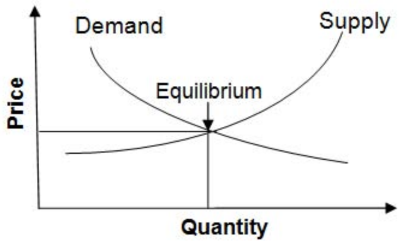

Figure 2.13: Demand and supply curve

will consequently generate varying content demand. In addition, existing methods also consider that content holder(s) are always accessible and hence calculate a static content supply. This is also impractical for DCS in the tourist spot type scenario as current content holder(s) might leave the POI, which will effectively reduce content supply. Again, new content holder(s) might enter the area and increase content supply. Therefore, a dynamic content demand and supply calculation is needed in irregular meeting places. In economic modeling, demand and supply are very much related, as presented in Figure 2.13. It shows that at the equilibrium point, the amount of supply is equal to the amount of demand. Therefore, to meet the dynamic demand, we need to increase the supply. To solve these major research issues associated with decentralized content sharing applicable to irregular meeting places, conceptual models to capture the dynamic demand and their corresponding supplies are yet to be developed.

(vi) Content replication enhances the content delivery service by increasing delivery success rates and decreasing latency. However, content replication decisions, such as which content to replicate, how many replicas to be created and where to place them considering both demands and contention among participating nodes, have not been addressed yet. This is difficult since there is no global knowledge about the popularity of contents or their access patterns. In addition, excessive replication and poor selection of replica placement will create congestion in the network.

2.4 Conclusion

This chapter has investigated the decentralized content sharing approaches and their data dissemination techniques. In addition, major issues including group formation, message forwarding, incentive and trust management, and content replication in DCS are explored to analyze the current state of the research. Based on the analysis, a number of research challenges have been identified for implementing decentralized content sharing in irregular meeting places. Based on these, a number of research objectives have been identified (see Section 1.4). The next chapter will introduce a basic framework for this purpose and the subsequent chapters will advocate further improvement according to these research objectives.

Chapter 3

Decentralized Content Sharing in Irregular Meeting Places

The literature review presented in Chapter 2 highlighted that the existing decentralized content sharing techniques depend on frequent encounters among users and their social relationships to address group formation and administrator selection issues and are only suitable for work-place type scenarios. However, because of the inherent dynamic nature of irregular meeting places, including the spontaneous movement of tourists and the probability of meeting strangers, existing schemes developed for work-place type scenarios are not effective here. This chapter addresses these issues and introduces a basic framework to facilitate *Decentralized content Sharing in Irregular meeting Places (DSIP)*, which includes unique methods for group formation, administrator selection and content delivery (**Block 1 - objective 1** in Fig 1.6). In DSIP, nodes (i.e., users) join a group considering that their interest can be fulfilled and requested contents will be delivered within tolerable delay, while the administrators are selected based on their ability to effectively serve group members for a longer amount of time. Extensive simulation was performed using a popular tourist spot in Victoria, Australia to assess the performance of the DSIP approach. Simulation results indicate that DSIP achieves acceptable delivery success rates and latency, and shows better performance, compared with relevant existing approaches when applied to such tourist spots.

3.1 Basic Framework for DCS in Irregular Meeting Places

We consider decentralized content sharing in irregular meeting places where different activities are available. The areas designated for activities are called Points-of-Interest (POIs) and can be detected using any map application and/or online information provided by the visitor information center. 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. Users are interested in performing different activities while visiting this tourist spot and have different levels of interest for participating in those activities. Figure 3.1 depicts DCS in a tourist spot with several POIs. Here, POI 1 depicts two overlapping content sharing groups along with the group members and their administrators. In group G1, a requester generates a request which is forwarded to the content holder via the administrator. The content holder delivers the matching content back to the requester. Although an end-to-end path for communication is shown in the figure for better understanding, such static paths are mostly unavailable in DCS because of node mobility. The node movement shown in the figure indicates that some of them will move to another POI, while others might leave the area.

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.

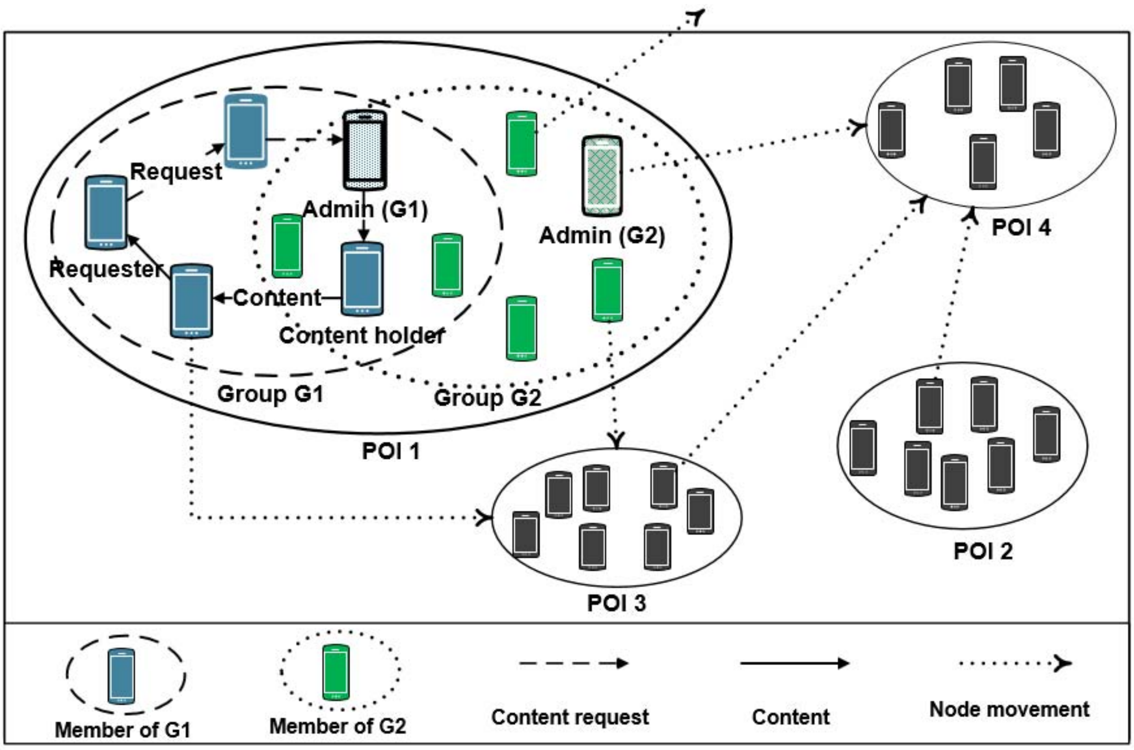

Figure 3.1: Schematic diagram for decentralized content sharing in irregular meeting places

If it receives a response from existing administrators before the timer expires, it joins one of the groups based on the criteria described later. In the 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, and declares itself as one of the candidates for the administrator of the group to be formed. The administrator is finally selected through a selection process and starts the 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 both cases, 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 aspect of content sharing, which is discussed in Section 3.1.2.

3.1.1 Administrator Selection

In addition to participating in content sharing, an administrator creates and manages a group content list and directs content requests towards the content holder(s). Existing approaches in the literature employ social metrics to select administrators [33, 38] which are not available in a tourist spot for the reasons stated in earlier sections. We employ administrator selection policies utilizing the information readily available in tourist spots. We consider the stay probability of a node in a POI, its connectivity value and available resources 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. 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 the literature [168, 169] suggests that 2-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 because of low battery or some other unforeseen reasons, the backup administrator takes the responsibility for the group and declares itself as the new administrator. The administrator periodically (every 30 minutes) 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:

$$
\begin{aligned}
&select \quad \kappa \\
&maximize \quad P^{\kappa} C^{\kappa} \\
&s.t. \quad E_f^{\kappa} > E_{min}^{\kappa} \quad and \quad \Lambda^{\kappa} > \Lambda_{min}\,.
\end{aligned} \tag{3.1}
$$

Here, P^{κ}, C^{κ}, E_f^{κ} and Λ^{κ} represent the stay probability, connectivity value, energy factor and available computational resources, respectively, of node κ. The minimum amount of battery life and computational resources required to provide an acceptable service as administrator are denoted by E_{min}^{κ} and Λ_{min}, respectively. Calculation of these components is as follows.

The 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 [170] 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 ‘c’ for $T_{c,f}^{\kappa}$ $(= T_{c,s}^{\kappa} + \Theta)$ time considering it has already spent $T_{c,s}^{\kappa}$ time in it. Taking stay time of node κ at POI 'c' as T_c^κ, the stay probability is calculated as,

$$
\begin{aligned}
P^\kappa\left(T_{c,s}^\kappa \le T_c^\kappa \le T_{c,f}^\kappa\right) &= \int_{T_{c,s}^\kappa}^{T_{c,f}^\kappa} \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(lnt-\mu_s)^2}{2\sigma_s^2}} dt \\
&= \frac{1}{2}\left(erf\left[\frac{ln(T_{c,s}^\kappa+\Theta)-\mu_s}{\sqrt{2}\sigma_s}\right] - erf\left[\frac{lnT_{c,s}^\kappa-\mu_s}{\sqrt{2}\sigma_s}\right]\right).
\end{aligned} \tag{3.2}
$$

Here, μ_s is the mean and σ_s is the standard deviation of the stay time, and $erf(.)$ represents error function. μ_s and σ_s can be calculated from personalized historical data stored by the content sharing app or online resources, such as the tourism department [171] or statistics bureau [172], where statistics about average 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., $\mu_s/2$) 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 neighbors. Each node periodically sends its neighbor list with the *hello* message which is used by the neighboring nodes to maintain a neighborhood table. From the neighborhood table node κ calculates its connectivity value as,

$$
C^\kappa = log(1+|\mathbb{H}_1|+\frac{|\mathbb{H}_2|}{2}+\frac{|\mathbb{H}_3|}{3}+\ldots+\frac{|\mathbb{H}_n|}{n})/log(H_T). \tag{3.3}
$$

Here, $\mathbb{H}_n$ depicts the set of n-hop neighbors and H_T is used for normalization purposes. At the beginning, when no group or administrator exists, the initiator (node that initiates the administrator selection process) will calculate and broadcast the value of H_T as, $H_T = 2V$, where V is the number of nodes participating in the administrator selection process. Afterwards, the administrator will update the value of H_T and broadcast it to the group members, if the number of current group members ($|\mathbb{G}|$) exceeds H_T. The administrator updates the value as, $H_T = |\mathbb{G}| + 2V$. A higher value of n introduces more traffic in the network as nodes have to broadcast a list of their (n-*1*)-hop neighbors to maintain the neighborhood table. As a trade-off, we use 2-hop connectivity (n=2) as in [168, 169]. In the literature, different values for the *hello* interval (10 sec∼10 min) are proposed [25, 56]. A short interval introduces more traffic whereas a long interval broadcasts outdated information. As a trade-off, we use 5-min as the *hello* interval in our approach. The logarithm function in the numerator and the denominator of Eq. (3.3) ensures diminishing marginal utility and normalized connectivity value, respectively. A higher connectivity value ensures that the selected node will have more neighbors to provide services to the group members.

Node κ calculates its energy factor as,

$$E_f^{\kappa} = E_r^{\kappa} / E_T^{\kappa}, \tag{3.4}$$

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

$$E_{min}^{\kappa} = \left(E_u^{\kappa} + E_{ad}\right) / E_T^{\kappa}. \tag{3.5}$$

Here, E_u^{κ} represents the energy consumption for regular use (e.g., call, sms, apps) and the battery drain for other reasons, and E_{ad} 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} for regular usage. Since request arrival in the content sharing process follows a Poisson distribution [37], the energy required to serve the group members is calculated as,

$$E_{ad} = E_{\alpha} \lambda_r \Theta. \tag{3.6}$$

Here, E_{α} shows the energy required to receive and reply to a request for content α, and λ_r 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 E_{α} using the energy model presented in [173]. Nodes calculate the request arrival rate from past experience of content sharing, and update it over time.

The resource availability metric is calculated as,

$$\Lambda^{\kappa} = (\vartheta^{\kappa} - \vartheta_u^{\kappa})(\iota^{\kappa} - \iota_u^{\kappa})/\vartheta_{max}\iota_{max}\,, \tag{3.7}$$

where, 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 are used for normalization purposes. The minimum resource required Λ_{min} is calculated by taking the median value of the device configurations currently available in the market.

3.1.2 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 3.1.1, starts the administrator selection process in collaboration with surrounding nodes which are not members of any group. After the administrator is selected, it starts the group formation procedure. In the second case, there is an 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 neighbors 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 a response from a single or multiple groups. For example, in Fig. 3.2 a new incoming node k receives information from the surrounding groups ***G1*** and ***G2***.

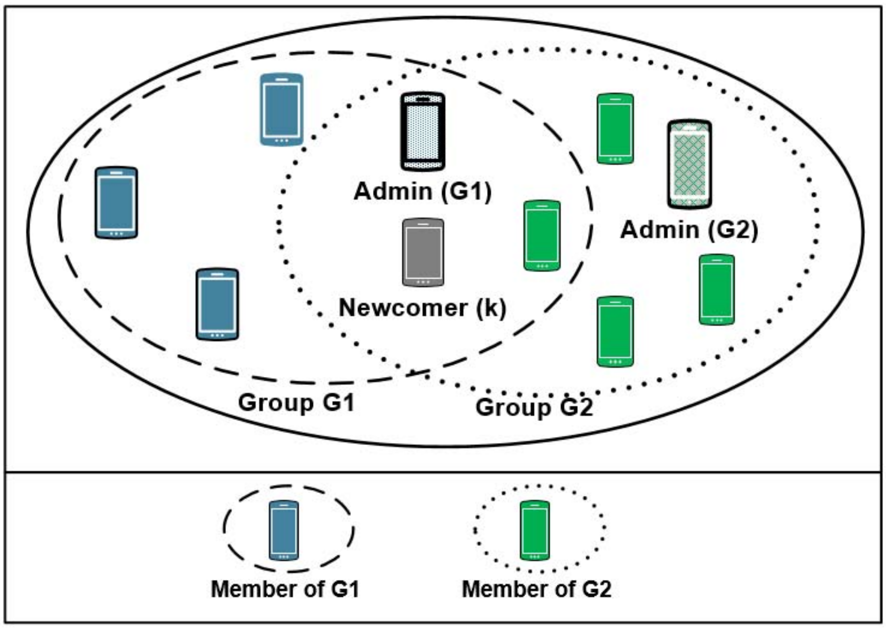

Figure 3.2: Decentralized content sharing with overlapping groups

If the node receives a response from a single group, 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 matching 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 the administrator. We consider the interest fulfillment 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,

$$\begin{aligned} &select \quad G \\ &maximize \quad \mathfrak{g}_G^k = \omega_G^k \Pi_G^k \psi_G^k \\ &s.t. \quad \overline{d}_G^k \leq d_{max}^k \quad and \quad h_G^k \leq n_{max}. \end{aligned} \tag{3.8}$$

Here, $\mathfrak{g}_G^k$ 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_{max}^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_{max} depicts the maximum allowable hop-distance to the administrator. To assign the value of n_{max}, 2-hop connectivity is considered because of the reasons mentioned in Section 3.1.1. In Eq. (3.8), 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. 3.2, node k calculates the gain factor for groups ***G1*** and ***G2***, and sends a joining request to the group administrator that yields the highest gain factor. If none of the groups are able to provide the expected service (i.e., delay within tolerable limit), the newcomer 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 an updated summary if there is any change (i.e., new content is added).

3.1.2.1 Interest Fulfillment Probability

Interest fulfillment 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, such as music, videos or news items. We consider both activity and content related interest while calculating interest fulfillment 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 the user's interest score from their personal profile, online recommendations, and location-centric experiences. We use multiple sources for interest score calculation as this ensures relevancy in calculation.

Personal Profile: Content sharing applications facilitate a 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 generalized personal interest score regardless of the nature or facilities available in any tourist spot. The interest score for category i from such a personal profile is calculated as,

$$\Upsilon_i^p = Y_i / Y_{max}. \tag{3.9}$$

Here, Y_i represents the level of interest for category i and Y_{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 go boating while visiting a particular spot, which is very famous for boating because of the nature of the waves and weather. This type of characteristic 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 in some of the recommended activities or content types when (s)he receives such suggestions. For example, if close friends of a user express opinions 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 analyzing them automatically using a sentiment-based text analysis technique [174] to derive positive or negative recommendations from each post, however, development or appropriate selection of such a technique is out of the scope of this thesis. 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,

$$\Upsilon_i^r = \varphi_{rec} \frac{\sum_{u \in \mathbb{V}_i} \gamma_{i,w}^u \varepsilon_i^u}{\sum_i^N \sum_{u \in \mathbb{V}_i} \gamma_{i,w}^u \varepsilon_i^u} + (1 - \varphi_{rec}) \frac{\sum_{u \in \mathbb{V}_i'} \gamma_{i,w'}^u \varepsilon_i^u}{\sum_i^N \sum_{u \in \mathbb{V}_i'} \gamma_{i,w'}^u \varepsilon_i^j}. \tag{3.10}$$

Here, $\gamma_{i,w}^u$ is the number of recent posts within the time window w $(t_1 \sim t_2)$ about category i by recommender u (online friend/family/tourist website), $\gamma_{i,w'}^u$ depicts the same within the time window w' $(t_{-o}{\sim}t_1$, where $-o$ refers to the past o time units), ε_i^u represents the reliability of recommender u 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 φ_{rec} represents the weight for the recent posts. A higher φ_{rec} suggests more emphasis on the recent posts. The value of φ_{rec} is calculated using the weight calculation of exponential moving average as, $\varphi_{rec} = 2/(W+1)$, 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 a 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 that are consistent across many people. We calculate the reliability of a recommender [175] u about category i during time window w as,

$$\varepsilon_{i,w}^u = \frac{\chi_i^u}{\max_{\forall u}(\chi_i^u)}\left(1 - |\overline{\xi_{i,w}} - \xi_{i,w}^u|\right). \tag{3.11}$$

Here, χ_i^u expresses the length of exposure of recommender u 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 a greater number of posts made by a recommender indicates more exposure. $\overline{\xi_{i,w}}$ and $\xi_{i,w}^u$ represent the median of all the recommendations and the recommendation score assigned by u about category i within time window w. To incorporate an earlier reliability score of a recommender, the reliability ε_i^u of recommender u can be calculated using the exponential moving average of $\varepsilon_{i,w}^u$ as,

$$\varepsilon_i^u = \varphi_r \varepsilon_{i,w}^u + (1-\varphi_r)\varepsilon_{i,w'}^u, \tag{3.12}$$

where, φ_r 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 l. 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,

$$\Upsilon_i^e = Z_i / \sum_{j=1}^{N_l} Z_j \,. \tag{3.13}$$

Here, Z_i depicts the number of times i occurred during past visits, and N_l shows the total number of categories consumed at location l.

For the initial visit to a place we cannot calculate (3.13) 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 the tourism research department [171] which can be collected automatically by the content sharing app. The interest score from location-centric experience for the initial visit is calculated as,

$$\Upsilon_i^e = \varphi_l \Psi_{i,l} + (1 - \varphi_l) \sum_{l'=1}^{M} \Im_{i,l'} / M_v \,. \tag{3.14}$$

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

Having obtained three types of interest scores for category i ($\Upsilon_i^p, \Upsilon_i^r, \Upsilon_i^e$), appropriate weight on each is needed to calculate the final score. The interest score obtained from the personal profile is provided by the user and is a better representation to reflect the actual interests of the user. This suggests that we should give more emphasis to scores obtained from personal profiles. However, sometimes interest scores obtained from other sources influence the behavior of the user too. For example, Alice might be very interested in boating, but while visiting a particular tourist spot l, she never goes boating as it is unsafe for her to do so there because of big waves. In this case, the location-centric experience will reflect her accurate interest score. Another user, Bob, who is usually reluctant to go fishing, may be convinced to try fishing in spot l on receiving highly positive recommendations for fishing there. In this case, online recommendations have a 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 the personal profile as it reflects a more accurate level of interest. On the other hand, a high variation suggests that online recommendations and location-centric experiences may imply a different level of interest than the personal profile because of the nature of the place or others' suggestions, as discussed earlier, and therefore the 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,

$$\Upsilon_i = \Upsilon_i^p(1-\sigma_i) + \sigma_i(\Upsilon_i^r + \Upsilon_i^e)/2\,. \tag{3.15}$$

While visiting a tourist spot, the content sharing app calculates interest scores for different activities and content types using the method discussed above. Such interest scores are used to generate the user interest matrix $\mathbf{U^k}$ of user k as,

$$\mathbf{U^k} = \{(I_1^k\ \Upsilon_1^k), (I_2^k\ \Upsilon_2^k), \ldots, (I_n^k\ \Upsilon_n^k)\}\,, \tag{3.16}$$

where, I_n^k indicates n-th category of interest and Υ_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,

$$\Upsilon_{i,G} = \max_{v \in \mathbb{G}} (\Upsilon_i^v) \,. \tag{3.17}$$

Here, $\mathbb{G}$ is the set of all members in group G. Whenever a new node joins a group, the administrator updates $\Upsilon_{i,G}$. 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, $\mathbf{U_G} = \{(I_{1,G}\,\Upsilon_{1,G}), (I_{2,G}\,\Upsilon_{2,G}), ..., (I_{n,G}\,\Upsilon_{n,G})\}$. The interest fulfillment 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,

$$\Gamma_{i,G}^k = \begin{cases} 0, & \text{if} \quad \Upsilon_{i,G} \geq \Upsilon_i^k \, or, \Upsilon_i^k = 0 \\ \Upsilon_i^k, & \text{if} \quad \Upsilon_{i,G} = 0 \quad and \quad \Upsilon_i^k \neq 0 \\ \Upsilon_i^k - \Upsilon_{i,G}, & \text{otherwise}\,. \end{cases} \tag{3.18}$$

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

$$\omega_G^k = 1 - \frac{1}{N^k} \sum_{i=1}^{N^k} \Gamma_{i,G}^k \,, \tag{3.19}$$

where, ω_G^k represents the interest fulfillment probability for group G calculated by node k and N^k represents its total number of categories of interest. A higher interest fulfillment value indicates that the group has a greater probability of fulfilling the user's interest.

3.1.2.2 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. The literature states that a user's perception of content availability depends on the reception of content within the required time [176]. 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 3.1.1 and sends it to the administrator. The administrator uses this information to update the category-wise average stay probability of the group members which is depicted as,

$$\overline{P_{i,G}} = \frac{1}{|\mathbb{G}_i|} \sum_{v \in \mathbb{G}_i} P_i^v \,. \tag{3.20}$$

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^v is the stay probability of node $v \in \mathbb{G}_i$. 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,

$$\Pi_G^k = \frac{1}{N^k} \sum_{i=1}^{N^k} \left(\Upsilon_i^k \overline{P_{i,G}} \right) , \tag{3.21}$$

where, Υ_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 the content sharing app. 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,

$$d_G = \varphi_d d_{cur} + (1 - \varphi_d) d_{prev} \,. \tag{3.22}$$

Here, d_G represents the maximum delay experienced by the administrator, d_{cur} shows the maximum delay observed during the current period, d_{prev} represents the maximum delay in the past and φ_d is the weight used for exponential moving average calculation (φ_d is calculated using a 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 newcomer node k, the administrator as well as its one-hop neighbors, reply with their observed maximum delay. The delay experience of one-hop neighbors is used as the newcomer node has a high probability of experiencing similar delay for message delivery. Afterwards, node k determines the average delay for group G as,

$$\overline{d}_G^k = \frac{\sum_{v=1}^{|\mathbb{H}_{1,G}^k|} d^v + d_G}{1 + |\mathbb{H}_{1,G}^k|}, \tag{3.23}$$

where, $\mathbb{G}_{1,G}^k$ is the subset of $\mathbb{G}$ who are also one-hop neighbors of node k, d^v is the maximum delay reported by node $v \in \mathbb{H}_{1,G}^k$, and d_G is the delay reported by the administrator of G.

3.1.2.3 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 the 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 neighborhood and a content list factor (discussed below) in the calculation of content delivery probability. After receiving an inquiry from a newcomer node k, the administrator sends the average stay probability of group members ($\overline{P_G}$) and the group content list factor ($\zeta_{f,G}$) along with the other information stated previously. Node k calculates the stay probability of the members of group G for content delivery as,

$$P_G^k = \frac{\sum_{v=1}^{|\mathbb{H}_{1,G}^k|} P^v + \overline{P_G}}{1 + |\mathbb{H}_{1,G}^k|}. \tag{3.24}$$

Here, P^v represents the stay probability of node $v \in \mathbb{H}^k_{1,G}$, and $\overline{P_G}$ is the average stay probability reported by the administrator of G (averaged over all i as per (3.20)). 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 a sufficient number of relay nodes is accounted as the neighbor factor as, $F^k_G = |\mathbb{H}^k_{1,G}|/|\mathbb{H}^k_1|$, where $|\mathbb{H}^k_1|$ represents the total number of one-hop neighbors 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 a certain limit. We consider that 2-copies of a content is sufficient to ensure successful delivery as evidenced in the works presented in [177]. 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, $\zeta^k_G = \zeta_{f,G}/\zeta_{f,max}$ where, $\zeta_{f,G}$ represents the content list factor reported by the administrator of G, and $\zeta_{f,max}$ represents maximum content list factor reported by any of the administrators. Finally, node k calculates the content delivery probability for group G as,

$$\psi^k_G = P^k_G F^k_G \zeta^k_G . \tag{3.25}$$

Using Eq. (3.19), (3.21), (3.23) and (3.25) in Eq. (3.8), the inquirer node k selects the group among the advertised ones that yields it the highest gain factor and then sends a join request to the administrator of that group. The group formation process is outlined in Algorithm 1.

3.1.3 Other Grouping Issues

3.1.3.1 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. (3.1)) to better qualify as the administrator of that

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. (3.8) for all groups in $\mathbb{g}_G^k = \left\{\mathbb{g}_G^k | 1 \leq g \leq G_T\right\}$, where $G_T =$ the number of groups from whom node k receives info;
  if $\mathbb{g}^k = null$
    k calculates self-assessment value using Eq. (3.1) and stores in a set $\mathbb{A}^v = \{\mathbb{A}^v | 1 \leq v \leq A_T\}$, where A_T = the number of nodes including itself from whom it receives broadcast info;
    if $arg(max(\mathbb{A}^v)) = k$
      Start a new group;
    end if
  else
    Find group G with maximum Gain i.e., $G = arg(max(\mathbb{g}_G^k))$;
    Send join request to the administrator of G;
  end if
end procedure

group. In such a 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 updates about this change because of delays in broadcasting and node mobility, effectively creating a situation where multiple nodes are treated as an 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 3.1.1. When the new administrator is selected, the current administrator passes the group information to the newly selected administrator.

3.1.3.2 Changing Group Membership

Some nodes might also be separated from other group members because of their mobility. We consider a node's connectivity to the administrator as the criterion for changing group membership. Nodes check *hello* messages from the neighbors in the neighborhood table to see if the administrator is reachable within n-hop distance, as discussed in Section 3.1.1. If the administrator is not found from the neighborhood 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 [33] and looks for other existing groups and joins a new group based on the group selection criteria.

3.1.3.3 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 *G1* where none of its one-hop neighbors are from group *G1*. In this case, for message forwarding it requires the service of members from other groups and joins another group *G2* for message forwarding purposes and indicates to *G2*'s administrator about its willingness to carry messages for group *G2* in lieu of the services it will get. Upon receiving such a request, group *G2* administrator accepts it and requests its members to carry messages for node k who, in turn, carries messages for group *G2*. In this case, messages from the members of the content sharing group get priority for forwarding by node k.

3.1.4 Request Generation and Content Forwarding and Delivery

Nodes generate content requests based on their interest from which keywords are extracted using the text analysis technique discussed in [33]. Such keywords can be matched against the metadata associated with a particular content using the content matching methods discussed in [178, 179] to locate appropriate content. The requester first sends the request to its one-hop neighbors, who match the request with their stored content list to determine if they have the matching content or not. If none of the neighbors 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)). Afterwards, the content holder(s) delivers the content to the requester.

The requests and contents are forwarded using a forwarding protocol. In this work, we have used the Spray-and-Wait [49] protocol for message and content forwarding as it spreads only a limited number of messages in the network to avoid congestion. The protocol sprays a specific number of copies (R) of a message in the network. Whenever a node meets another node with $r > 1$ ($r \leq R$) copies available for spraying, it forwards $\lfloor \frac{r}{2} \rfloor$ copies to the second node and keeps $\lceil \frac{r}{2} \rceil$ copies for itself. If any node has only a single copy available for forwarding, it holds on to that copy until it meets the destination. As content requests are smaller in size we use a higher number of copies ($R = 7$) for forwarding the request. However, content itself is much larger in size and more copies creates more congestion in the network. We use $R = 2$ for content delivery, as discussed in Section 3.1.2.3 that two copies are sufficient for successful content delivery.

3.2 Simulation Environment and Results

3.2.1 Simulation Environment

To the best of our knowledge, there is no publicly available dataset that demonstrates movement of tourists in a tourist spot. Therefore, we generated synthetic mobility data to explore the content sharing process in a tourist spot using Lakes Entrance, which is one of the prominent tourist spots in Victoria, Australia, as the location . 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 area using the negative binomial distribution, which are consistent with the characteristics of tourists in tourist spots [180]. For generating individual node mobility, activity related interest scores and stay time were used. The average stay time in a POI was taken considering the tourism related information provided in [171, 172], and the stay time of a particular node in a particular POI was determined by using ±20% variance. 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 [181] 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 fish 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∼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. Figure 3.3 shows a map of Lakes Entrance where the POIs are marked with different colors representing different activities. To simulate content sharing, we used the NS3 network simulator. NS3 took the mobility file generated by a java program as input to direct nodes' movement during simulation. Nodes were initialized with random battery power where 100 units represented the full capacity. Periodic battery drain (0.07 unit/minute) and drain for data transfers (8×10^{-8} unit/byte) were also used. Battery level was increased gradually for a random amount of time whenever nodes were in the charging state.

We considered that contents are of the following types: Text (40%, 512 KB), Photos (20%, 1MB), Audio (20%, 5MB [56]) and Video (20%, 6.5MB [182]), where the first number within the parenthesis represents the percentage of that type of content and the second number represents mean content size. Abhari *et al.* [183] suggest that the file size in a popular content sharing network (i.e., YouTube) follows Gamma distribution, hence the size of individual contents was determined using that distribution with shape parameter (=2.0), as in [69]. We considered nine types of content, of which six

Figure 3.3: Map of Lakes Entrance with highlighted POIs

are related to activity based interest, such as 'Accommodation', 'Bowling', 'Camping', 'Fishing', 'Food and Shopping' and 'Walking', and three are related to content related to interest, such as 'Movies', 'Music' and 'News'. We considered 30 contents for each category and a total of 270 contents in the network. Nodes were initially assigned with 20∼30 contents across different categories. Nodes generated content requests based on their interest following a Zipf distribution [69] and the request generation time followed an exponential distribution as in [38]. In the simulation, we have used IEEE 802.11g in ad-hoc mode at 54 Mbps data rate operating at Channel#1 with 20 MHz width and the application installed in each mobile device communicates in this channel. NS3 already provisions MAC layer and physical layer modules for IEEE 802.11 standards. For modeling the wireless channel in NS3, we employed the constant speed propagation delay model and range propagation loss model. Some simulation parameters are listed in Table 3.1.

3.2.2 Simulation Results

We used three metrics, namely, (i) hit rate, (ii) delivery success rate and (iii) average delivery latency to assess the performance of the proposed DSIP approach. Hit rate represents the percentage of the requests for which contents are successfully located by

Table 3.1: Simulation parameter

Simulation parameters	Value
Simulation area	10 Km X 5 Km
Number of nodes	150
Duration	3 days
Avg. node arrival	10-30 node/hr
Content request rate	2-6 request/hr
Request life time	6 hr
Range of IEEE 802.11g communication	150 m

the requester and delivery success rate depicts the percentage of successful deliveries for the same. The average delivery latency shows the average time it takes to obtain a requested content. These metrics are selected because hit rate indicates the impact of selecting appropriate administrator nodes who are able to successfully direct content requests towards appropriate content holders, while success rate relates to the extent of communication coverage and delivery service available in a group, and finally latency denotes a requesting node's reachability from the content holder via other group members. Additionally, we also investigated average administrator lifetime, administrator expiration rate (for low battery) and control message overhead. Average administrator lifetime depicts how long an administrator serves the group members on average. Administrator expiration rate shows the percentage of administrators who died because of low battery.

None of the existing approaches in the literature are devised for decentralized content sharing in tourist spots. In the absence of any comparable approach, we compared our approach with one of the recent ones called SPOON [33], 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 periods cannot be accommodated in tourist spot type scenarios, we considered all nodes to have a similar degree centrality for SPOON. To make a fair comparison between DSIP and SPOON, we employed the Spray-and-Wait [49] message forwarding protocol for request propagation and content delivery. Since content requests are smaller in size than actual content, we used a higher number of copies (7 copies) for forwarding content requests and a lower number of copies (2 copies) for actual content delivery. The t-test comparing our scheme and SPOON in terms of hit rate, success rate and delivery latency at various arrival and request rates yielded p-values, $p \leq 2.48 \times 10^{-6}$, $p \leq 4.16 \times 10^{-6}$ and $p \leq 9.42 \times 10^{-6}$, respectively at 99% confidence level, validating their performance differences as being statistically significant. Each simulation was run 20 times and the results presented in this section are the average of 20 runs.

Figs. 3.4 and 3.5 show the improvements achieved with respect to hit and success rate when the proposed DSIP approach is employed. For example, hit and success rates achieved by DSIP are 63.48% and 59.13%, respectively, compared to 45.43% and 42.56% in SPOON for an arrival rate of 25 nodes/hr. Improvements in our method with respect to SPOON for different arrival rates range from 15-19% for hit rate and 13-17% for delivery success rate. Such improvements are achieved because of the proposed group formation and administrator selection mechanisms. The group formation mechanism in the DSIP approach considers interest fulfillment 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

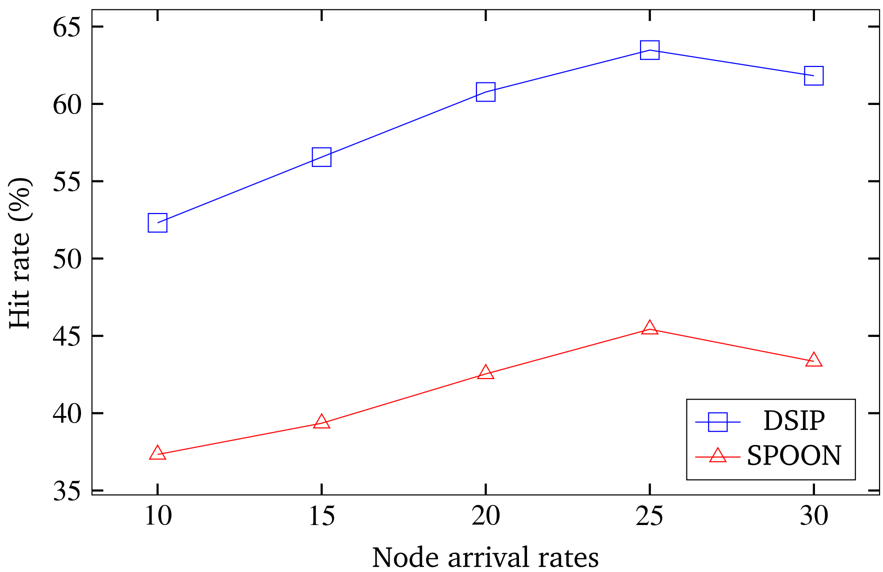

Figure 3.4: Hit rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr

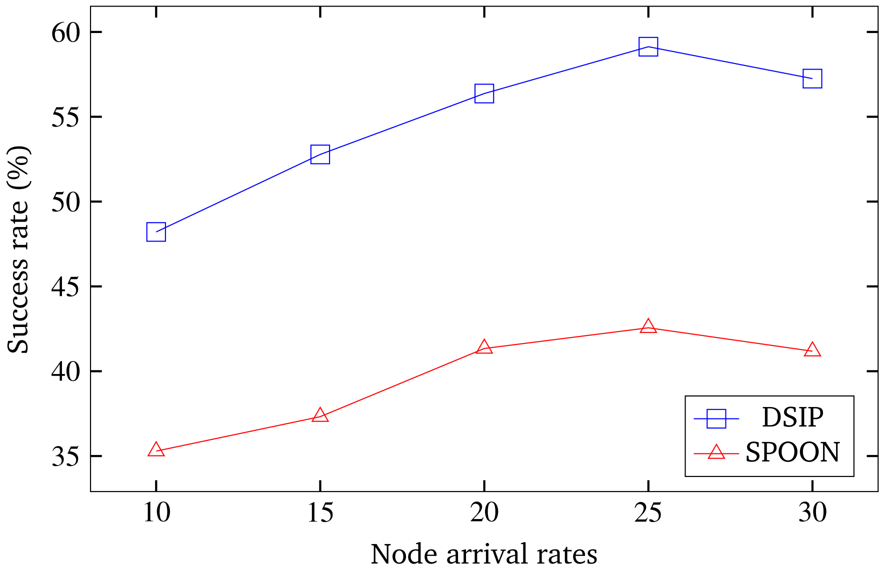

Figure 3.5: Success rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr

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. However, when the arrival rate exceeds a certain value (i.e., 25 nodes/hr for the given settings), both rates start to decrease slightly because more nodes generate more requests, which causes congestion. Existing approaches proposed for work-place type scenarios, where regular movement patterns and meeting frequency are established, reported hit and success rates within the range of 75-80% [33, 54], as highlighted in Chapter 2. In that respect, our method attains acceptable performance in tourist spots where movement patterns are irregular and social ties are mostly non-existent.

Figures 3.6 and 3.7 show hit and success rates with varying content request rate. Both hit and success rates increase with increasing request rate. The reason for this is that with an increasing request rate, the probability of generating similar requests by nodes in the same neighborhood becomes higher, and hence a requester can obtain contents from one of the neighbors who has already requested and obtained those contents. Both rates start to slightly decrease when the request rate exceeds a certain limit (5 requests/hr in this setting), because more requests generate more congestion in the network. DSIP attains 60.33% hit rate and 54.11% delivery success rates even when

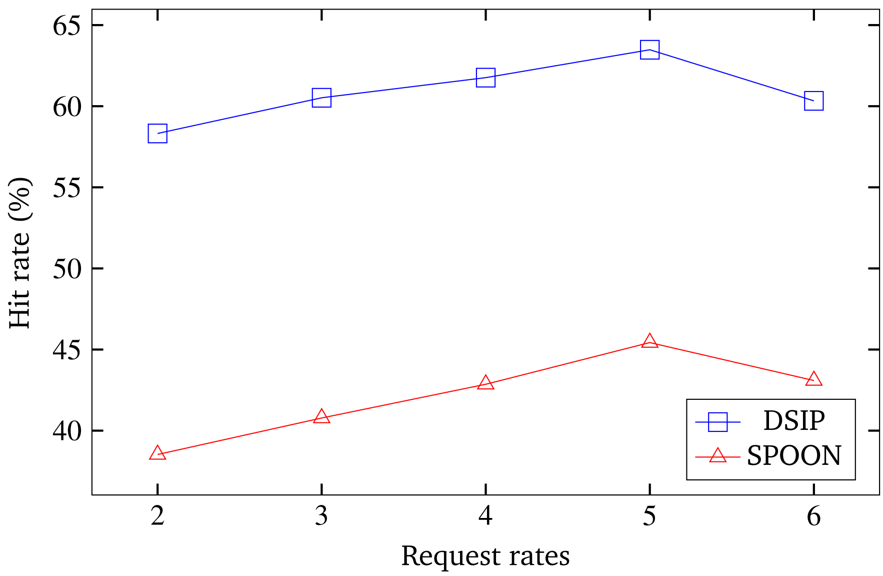

Figure 3.6: Hit rate for different mean request rates (number of requests/hr) keeping the mean node arrival rate fixed at 25 nodes/hr

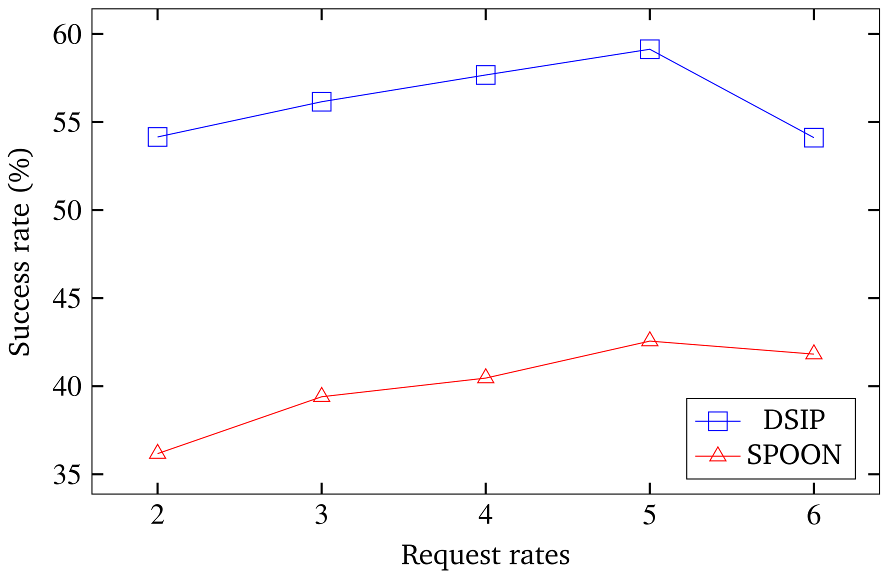

Figure 3.7: Success rate for different mean request rates (number of requests/hr) keeping the mean node arrival rate fixed at 25 nodes/hr

the request rate is very high (6 requests/hr). Our method achieves 12-20% higher rates than SPOON, suggesting that it is more suitable in situations where the request rate is high because of the presence of more active users generating a higher number of requests in the network. However, the difference between the relevant hit rate and success rates for our proposed approach (DSIP) ranges from 4-6.5% suggesting that there is a scope for improvement and adoption of a more appropriate message forwarding technique may result in much higher delivery success rate and consequently will improve the overall performance of the DSIP approach. The possibility of employing an effective message forwarding technique with the aim to improve performance is explored in the next chapter.

The average delivery latency is presented in Fig. 3.8 where latency values for successful deliveries are sorted in ascending order and averaged over different intervals as indicated in the *x*-axis of the figure for convenience of presentation. Note that only for this given simulation, we considered 10,000 identical content requests across both the approaches for fair comparison. Both DSIP and SPOON approaches complete up to

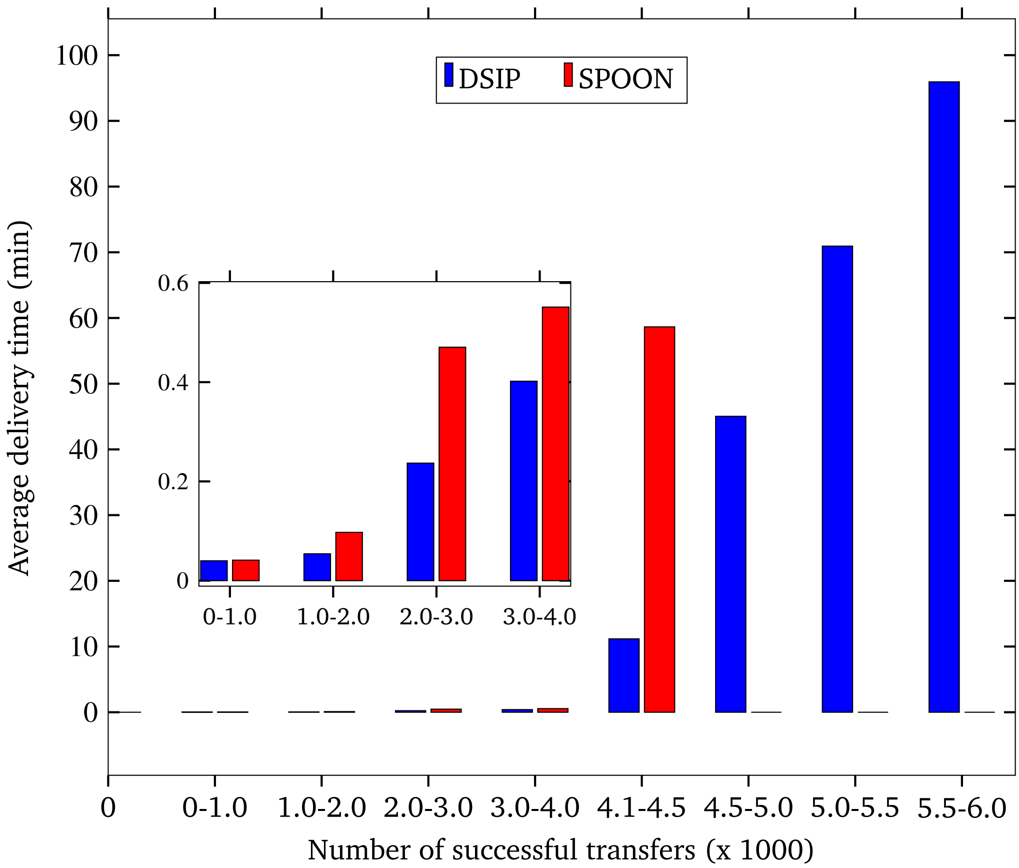

Figure 3.8: Average delivery latency for successful deliveries for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

4×10^3 successful transfers within a short time (0.40 mins for DSIP vs. 0.55 mins for SPOON). Afterwards (up to 4.8×10^3), SPOON shows significantly higher delay than DSIP (SPOON 38.65 mins vs. DSIP 1.16 mins). Results also show that SPOON could only deliver up to 4.8×10^3 successful transfers for the given settings whereas DSIP could deliver 6.0×10^3 requests with a relatively longer delay ($\geq$ 45 mins on average) after 5.0×10^3 requests. In some cases, the content holders are located multiple hops away from the requester, and this requires longer time for request propagation and delivery of contents. This suggests that DSIP 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 times (e.g., few hours to a couple of days) and delivery is more important than delay.

Figures 3.9 and 3.10 show content list growth, which is the average increment of the percentage of contents after sharing at the end of simulation for varying node arrival and request rates. Growth increases with an increasing node arrival or content request rate for both DSIP and SPOON. Results show that the content list growth rate is much higher in DSIP as more contents were successfully received. One observation from the figures is that after a certain arrival rate (25 nodes/hr in Fig. 3.9) and request

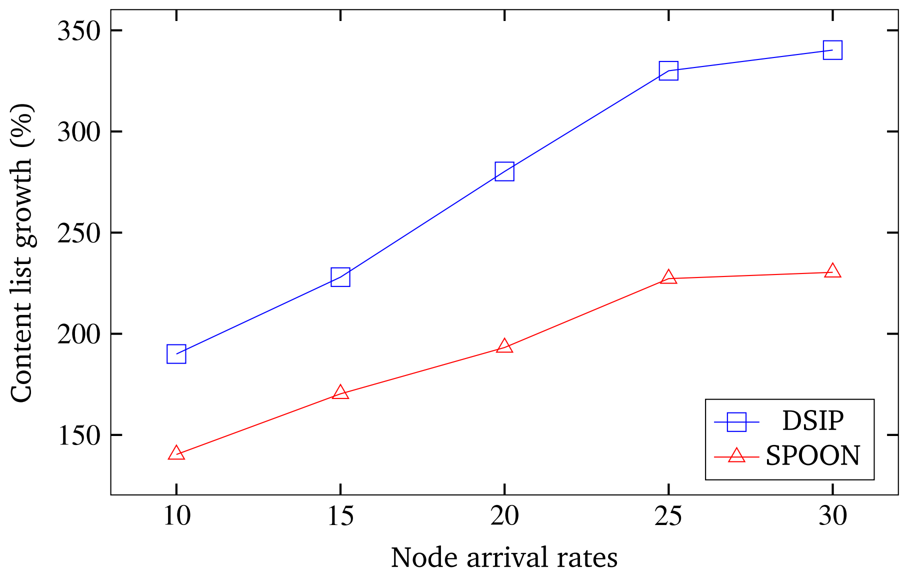

Figure 3.9: Content list growth for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

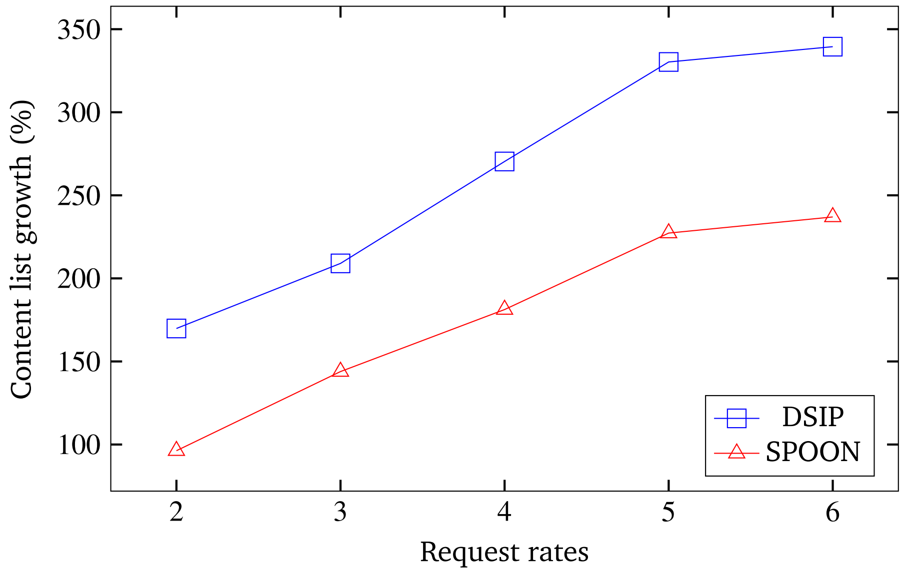

Figure 3.10: Content list growth for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

rate (5 requests/hr in Fig. 3.9) the increment is not substantial, suggesting that their upper bounds have been reached. This is because a higher arrival or request rate results in more requests that cause higher congestion in the network and thereby limits the number of successfully delivered contents.

Control messages were used for various aspects of group formation, group updates, administrator selection and content request propagation. Figures 3.11 and 3.12 show the percentage of message overhead (total control message size/total successfully delivered content size) for DSIP and SPOON approaches across different node arrival and request rates. The figures show that our proposed DSIP approach generated slightly more control message overhead than SPOON (2.3% vs. 0.92% for an arrival rate of 25 nodes/ hr with a request rate of 5 requests/hr). We used a neighborhood 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 the much higher hit and success rates achieved by our proposed approach, slightly higher control message overhead is a trade-off that brings benefits for content sharing in our given scenario. Moreover, as the node arrival rate or request rate start to increase, the size of successfully delivered contents also increases and the proposed method shows lower overhead. Figure 3.11 shows that after

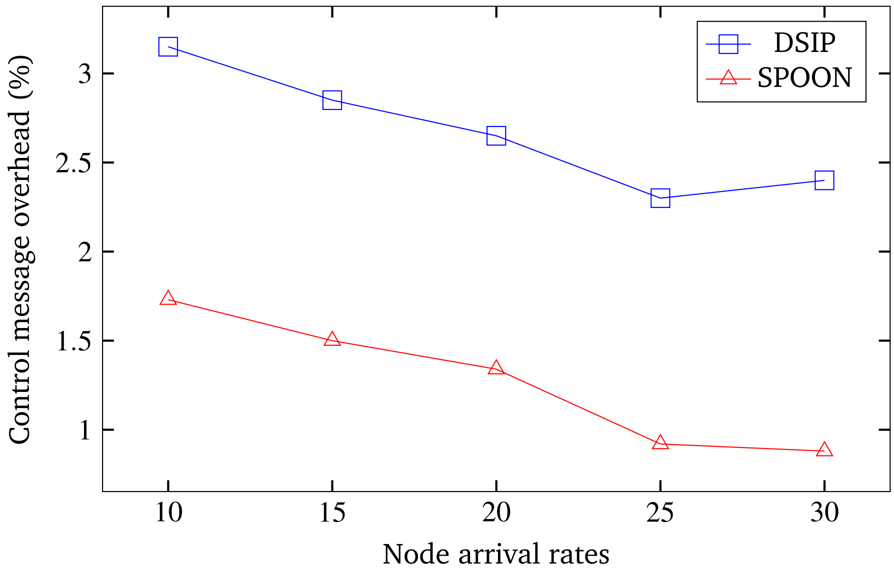

Figure 3.11: Control message overhead for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

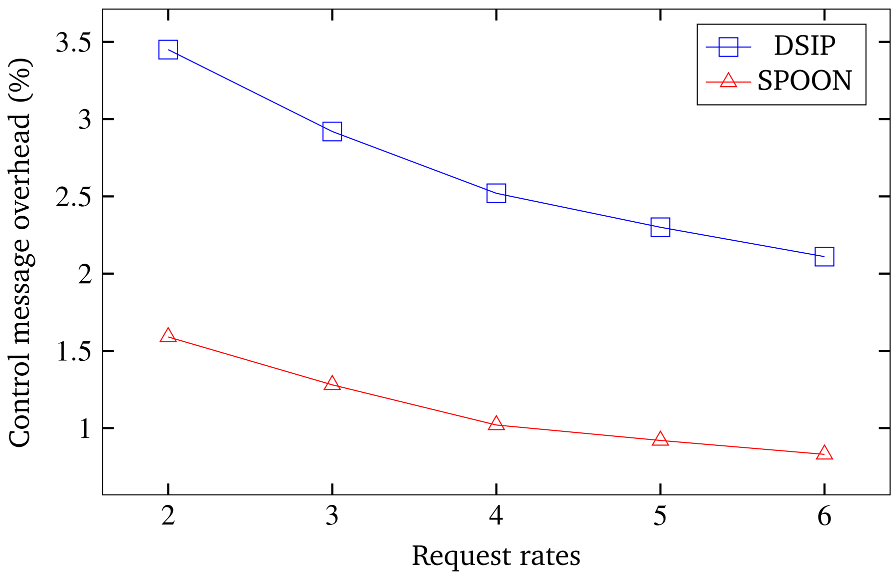

Figure 3.12: Control message overhead for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

a certain arrival rate (25 nodes/hr in current setting), message overhead sightly increases as the size of the neighborhood table increases with an increasing node arrival rate.

3.2.2.1 Impact of Administrator Selection Policy

The proposed DSIP approach incorporates an administrator selection policy which uses stay probability and connectivity of a node along with its available resources to select an administrator. The impact of such an administrator selection policy is exhibited in Figs. 3.13 and 3.14. Both figures show that the average lifetime decreases with an increasing node arrival or content request rate. The reason for this is that with higher arrival or request rate more requests are generated in a group, which in turn requires the administrator to consume more energy to serve those requests and hence a shorter average lifetime is achieved. The proposed DSIP approach results in much higher average lifetime for an administrator than SPOON (130 mins vs. 40 mins at an arrival rate of 25 nodes/hr with a request rate of 5 requests/hr) across all settings. Better results are obtained for DSIP as stay probability is used as one of the criteria for selecting administrators.

Figures 3.15 and 3.16 show the expiry rate of the administrators for low battery. Similar to Figs. 3.13 and 3.14, these figures also demonstrate that a higher node arrival rate or content request rate result in a higher expiry rate because of low battery for administrators as they need to use more resources to serve a higher number of requests,

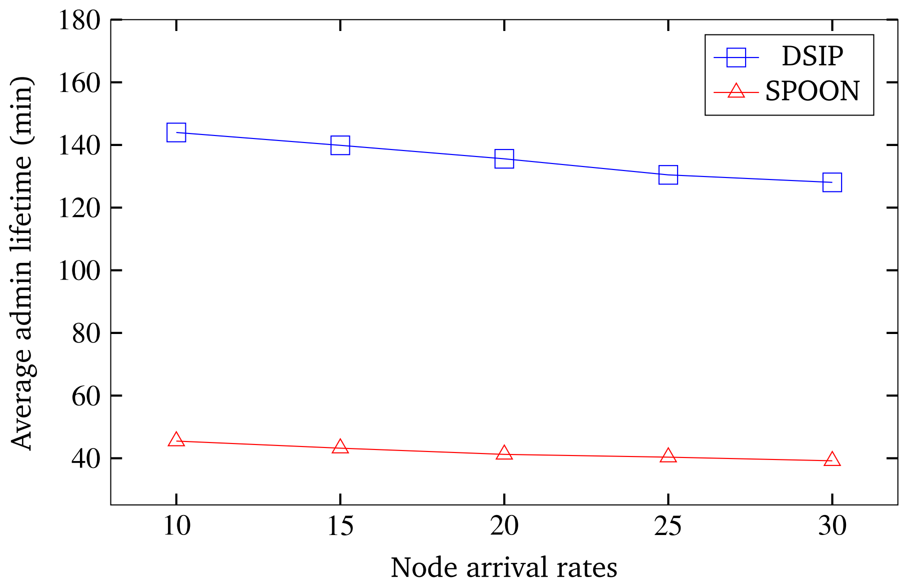

Figure 3.13: Average lifetime of administrator nodes for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

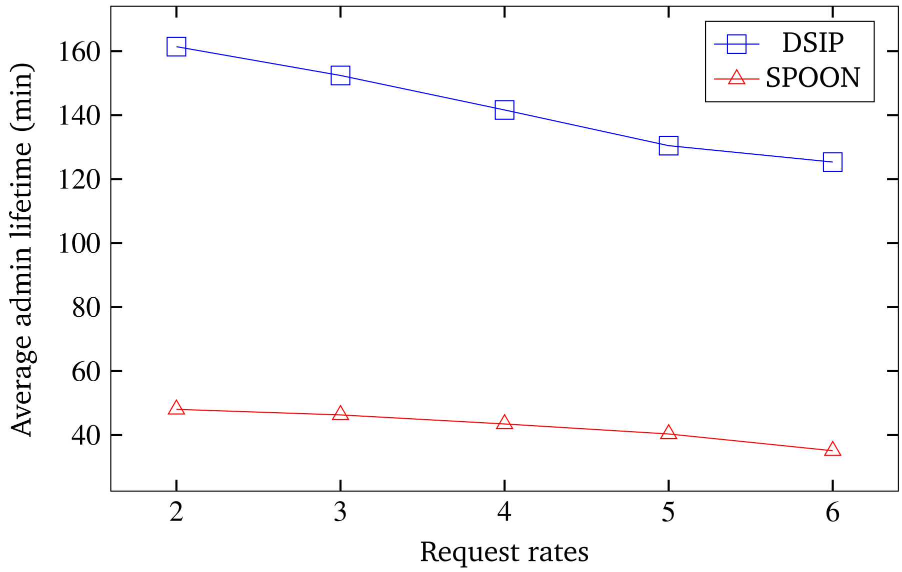

Figure 3.14: Average lifetime of administrator nodes for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

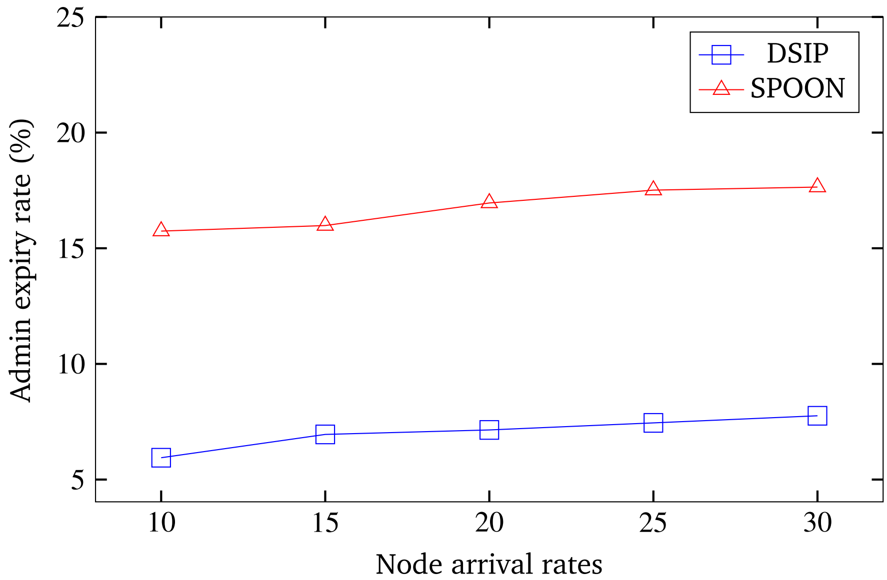

Figure 3.15: Administrator expiry rate because of low battery for different mean node arrival rates keeping the mean content request rate fixed at 5 requests/hr

as alluded before. The figures also show that the rate is much higher for SPOON since it does not consider remaining battery lifetime for administrator selection.

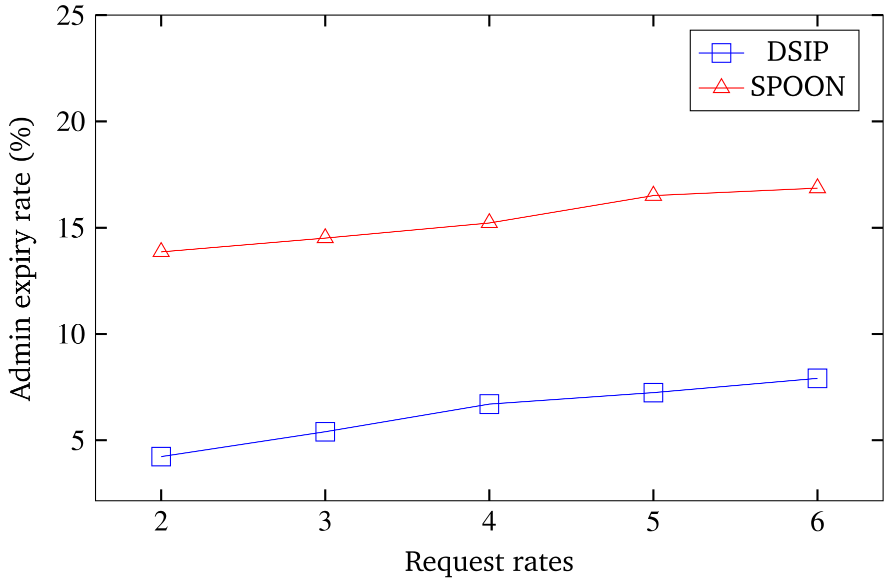

Figure 3.16: Administrator expiry rate because of low battery for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

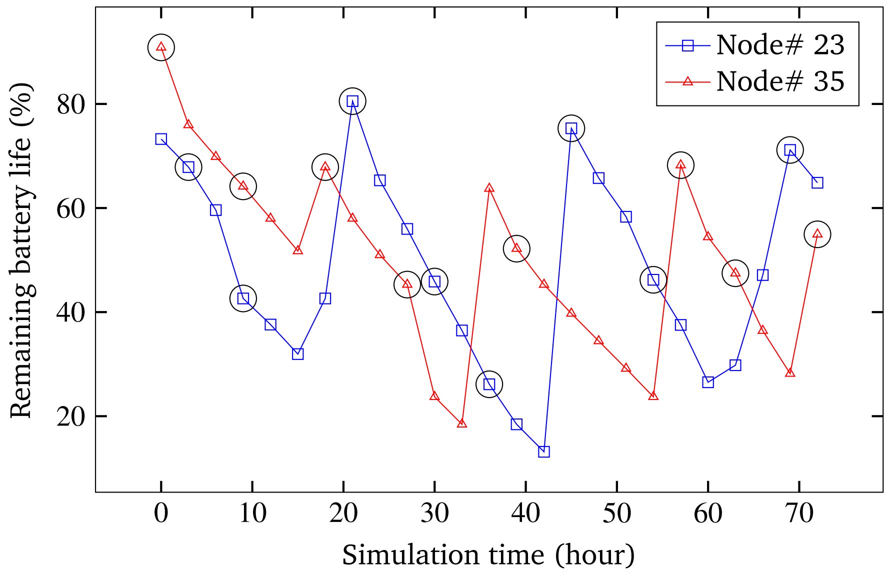

Figure 3.17: Remaining battery life for administrators. The steep energy rise shows charging of the devices. The circle shows the point in simulation when nodes were selected as administrator

Figure 3.17 shows the remaining battery lifetime for two of the nodes (Node# 23 and Node# 35) which were selected as administrators most frequently (8 times) in various POIs among all nodes. The figure suggests that although the nodes were selected as administrators many times, they never ran out of battery, because of the consideration of energy issues (Section 3.1.1) in the selection process that selects a node as administrator only when it is suited.

The administrator handover rate in different POIs is presented in Figure 3.18. The figure shows that the rate is much higher for SPOON compared to DSIP as the former does not consider the stay probability for selecting administrators. The figure also suggests that 'Camping' and 'Accommodation' need a higher administrator handover rate than the other POIs. The reason for this is that these POIs have a larger area and hence during simulation more groups existed in those areas which resulted in a higher number of administrator handovers.

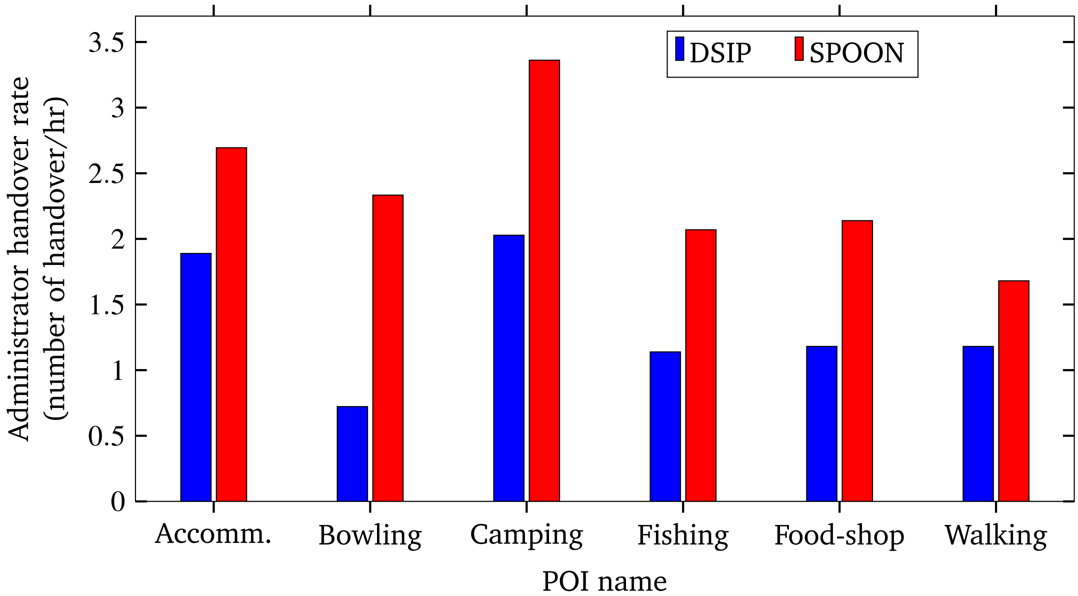

Figure 3.18: Administrator handover rate in different POIs for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

3.2.2.2 Performance in Different POIs

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. 3.19 suggests that hit and success rates are high in 'Accommodation',

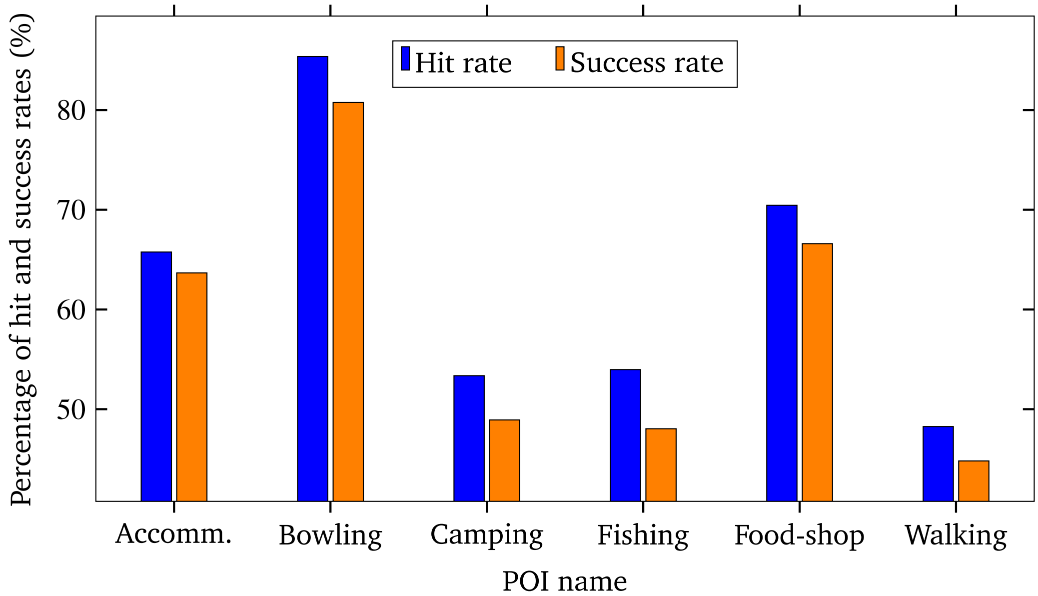

Figure 3.19: Hit and success rates in different POIs for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

'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 the 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 with each other's facilitating message and content delivery. In the case of 'Food and Shopping', we considered a 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 a 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 stay there for a random time ranging from 0.5-2 hrs) suggesting infrequent contact of nodes that yield lower hit and success rates. In comparison, 'Walking' suffers because of the 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.

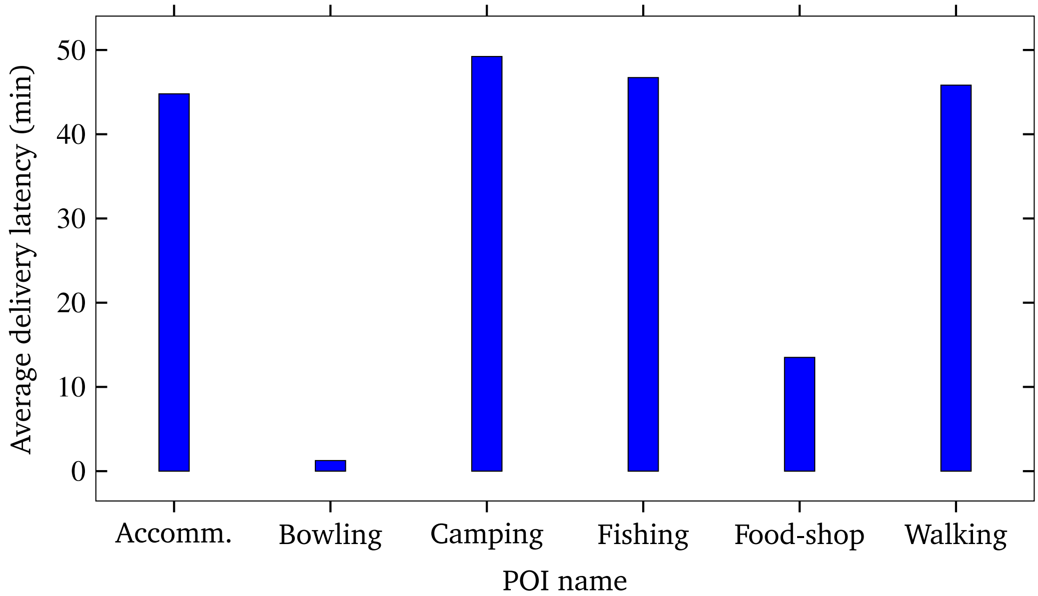

Figure 3.20: Average delivery latency in different POIs for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

Fig. 3.20 shows the average delivery latency in different POIs. The figure suggests that 'Bowling' attains the lowest latency as nodes remain within the communication range more often, as stated before. On the other hand, in 'Accommodation', 'Camping' and 'Fishing', the more static nature of the nodes in these areas leads to infrequent contacts, causing longer delays. In contrast, the higher delivery observed in the 'Walking' POI is due to highly dynamic movement of nodes. Note that for 75% of the delivery, the latency is much lower, similar to the first part of Fig. 3.8 (up to 4.5×10^3 successful delivery), as explained in the related discussion on Pg. 111. The purpose of this figure is to show the variation of latency for different POIs.

3.3 Conclusion

This chapter has presented a framework to enable smart mobile device users to share contents in irregular meeting places such as tourist spots. The proposed DSIP approach uses 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 fulfillment probability, content availability and delivery probabilities are also taken into consideration for a node to join a group. An optimization problem is formulated which allows a node to join a group by maximizing its benefit in terms of those factors. The administrator of a group is selected through a competitive process on the basis of the candidate node's probability of stay within the group, its connectivity with neighboring nodes, and its capability to serve the group during its stay. For performance evaluation, the characteristics and environment of a tourist spot were applied to a recent content sharing method called SPOON, which was developed for sharing contents in work-place type scenarios, and was used to compare with our proposed DSIP approach. A popular tourist spot in Victoria, Australia was used as the test location for the simulation. Results obtained from the simulation suggest that the DSIP approach outperforms SPOON with respect to hit rate, success rate, delivery latency and longer administrator lifetime, and hence is suitable for sharing contents in the tourist spot type scenario.

The performance of DSIP can be further improved through incorporation of an intelligent and more appropriate message forwarding protocol. DSIP uses the Spray-and-Wait message forwarding protocol which blindly spreads a specific number of copies of a message, as alluded in Section 3.2.2. In this regard, the performance can be improved by selecting an appropriate forwarder node based on its capability of successfully delivering a content. Additionally, DSIP lacks an incentive scheme to encourage participation and identify misbehaving nodes, which are important for the practical adoption of the approach. To address these issues, the next chapter will introduce mechanisms to improve the message forwarding protocol, encourage participation and identify misbehaving nodes.

Chapter 4

Enhancing Content Delivery and Participation in DSIP

The previous chapter has presented a method to facilitate decentralized content sharing in irregular meeting places (DSIP). Although the proposed DSIP method achieves improved and acceptable content delivery service considering the characteristics of irregular meeting places such as spontaneous movements and unfamiliar users, its performance can be further improved. To this end, an appropriate message forwarding scheme is needed which will ensure that the requester and the content holder/forwarder will meet in the near future to allow successful delivery. In addition, the DSIP method assumes that nodes are cooperative and help each other by carrying contents. However, in real life, there are some selfish nodes who try to obtain contents from others, but do not deliver or forward content for others. In this regard, an appropriate incentive mechanism is needed to encourage participation. Furthermore, some nodes might misbehave by making false claims through incentive mechanisms to gain unfair benefits. A trust management scheme is also needed to identify such misbehaving nodes. To address these issues, this chapter presents an *Enhanced DSIP (E-DSIP)* method which includes a utility based message forwarding technique to improve content delivery service (**Block 2 - objective 2** in Fig 1.6), and an incentive and trust management scheme to increase participation and minimize risks (**Block 3 - objective 2** in Fig 1.6). Extensive simulation has been performed to assess the performance of E-DSIP and the obtained results suggest that E-DSIP outperforms DSIP and SPOON in terms of delivery success rate and latency.

4.1 Utility Based Forwarding Technique

Similar to the DSIP method presented in Chapter 3, we consider content sharing in irregular meeting places where the whole area is divided into several POIs. The E-DSIP method uses the group formation and administrator selection techniques presented in the previous chapter and focuses on efficient message forwarding, more specifically content delivery, since the size of the control messages is much smaller than the actual content, as highlighted in Section 3.2.2. We consider that while generating a content request, a requester specifies its next activity and the lifetime of the request after which it is not interested in obtaining the content. If a requester does not mention its next activity, we use historical information from past visits to this location or interest scores for each activity if no previous history is available, to predict the next activity.

Figure 4.1 shows a schematic representation of the message forwarding problem. It shows that node Q_C in POI 1 is a content carrier node (i.e., content holder or a forwarder) with content α_1 and needs to deliver the content to requester node Q_R which is not within its communication range. Now, with the single copy message forwarding technique, node Q_C needs to decide whether to keep the message (i.e., requested content α_1) or to forward it to one of the neighbors within its communication range (i.e., Q_X, Q_Y or Q_Z). If node Q_C randomly selects one of the nodes (e.g., Q_Y or Q_Z), that node might leave the current POI before delivering the content. Similarly, carrier node V_c (in POI 2 at the right-hand side of the figure) needs to deliver content α_2 to requester V_R; however, in this case, the requester itself might move to another POI before obtaining the content. In general, the message forwarding problem can be considered as finding the best node to carry the message for successful delivery whenever a new message is received or a new node is encountered. Note that existing approaches in the literature for the work-place type scenario based on routine movement patterns or social relationships are not effective here, as discussed before. To address this issue, this section presents a utility based forwarding technique for increasing successful delivery. In addition, E-DSIP also includes an incentive scheme to encourage node participation, which is discussed in Section 4.2.1, and a trust management scheme to identify misbehaving nodes presented in Section 4.2.2.

When a carrier node (e.g., Q_C and V_C in Fig. 4.1) has a content to deliver, it can either carry the content by itself for future delivery or forward the content to one of its neighbors. 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

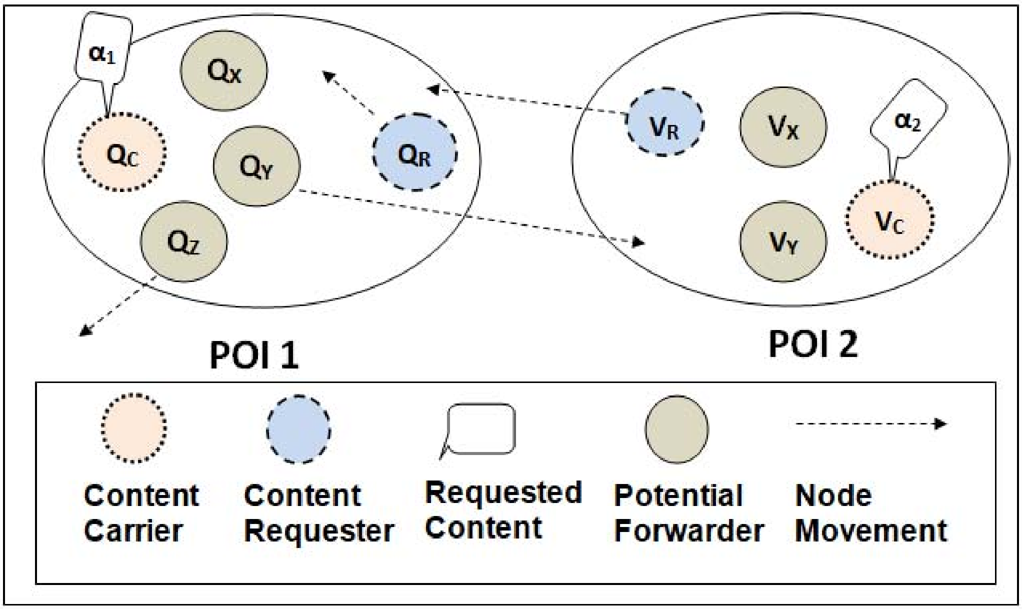

Figure 4.1: Message forwarding in DCS

destination node (i.e., requester) for a longer time enhances the carrier's chance of having more opportunistic contact. Similarly, a carrier node with more neighbors has a better probability of delivering the content rather than a stand-alone node. It also needs to have sufficient resources (e.g., buffer space, energy) to carry the content until delivery. The carrier node uses the above-mentioned criteria to calculate a utility value. It also broadcasts a summary of the request to its immediate neighbors, who calculate their own utility value for carrying the content, and sends it back to the content carrier node. Afterwards, the content carrier node considers itself and all of its one-hop neighbors as the potential forwarders and selects the node with the highest utility value as the forwarder. The selected forwarder node then carries the content until it meets the destination or another node with higher utility value, or the lifetime of the request. Thus the selection process is formulated as a maximization problem as,

$$\begin{aligned} select \quad & k \\ maximize \quad & U_{\alpha}^{k,d} \\ s.t. \quad & E_f^k > E_{\alpha} \quad and \quad B_a^k > B_{\alpha}. \end{aligned} \tag{4.1}$$

Here, $U_{\alpha}^{k,d}$ represents node k's utility to forward content α to destination node (i.e., content requester) d. E_f^k and E_{α} represent the energy factor of node k and energy required to forward α, respectively. B_a^k and B_{α} represent the buffer space available to node k and the minimum space required to store α, respectively.

A node's ability to successfully deliver a content depends on its expected co-location probability with the destination node. Here, co-location refers to staying in the same POI, but not necessarily within each other's communication range. If both the forwarder and the destination nodes are co-located for a longer time, the probability of contact opportunity and successful delivery increases. Moreover, the forwarder node needs sufficient neighbors to ensure that it can deliver the content, which is calculated using its connectivity value. Considering all these factors, the utility value is calculated as,

$$U_{\alpha}^{k,d} = \begin{cases} 1, & \text{if} \quad h^{k,d} = 1 \\ L^{k,d} C^k, & \text{otherwise}. \end{cases} \tag{4.2}$$

Here, $h^{k,d}$ shows the hop-distance between node k and d. $L^{k,d}$ represents the co-location stay probability of the forwarder (k) and the destination node (d), and C^k is the connectivity value of node k. The co-location stay probability is calculated as,

$$L^{k,d} = \varphi_c L_{cur}^{k,d} + (1 - \varphi_c) L_{next}^{k,d} . \tag{4.3}$$

In Eq. (4.3), $L_{cur}^{k,d}$ shows the co-location stay probability of node k and d in the *current POI*, and $L_{next}^{k,d}$ depicts the same in the *next POI*. $L_{next}^{k,d}$ is considered as the requester might soon finish its current activity and move into the next POI, before obtaining the content. In that case, the content can still be delivered by a forwarder who is likely to move to the same POI, if it meets the requester there before the lifetime of the request expires. φ_c is used as a weighting factor. The co-location stay probability of node k and node d at current POI 'c' is determined as,

$$L_{cur}^{k,d} = \begin{cases} 1, & \text{if} \quad T_{c,r}^k \geq T_{c,r}^d \\ \frac{T_{c,r}^k}{T_{c,r}^d}, & \text{otherwise}. \end{cases} \tag{4.4}$$

Here, $T^k_{c,r}$ and $T^d_{c,r}$ show the remaining stay time of nodes k 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 prospective forwarder node also calculates its own expected remaining time and the co-location stay probability. Remaining stay time of a node is calculated using the probability density function (pdf) of stay time of nodes in a POI, and time already spent in that POI. Similar to the stay probability calculation in Eq. (3.2) in Chapter 3, we consider that stay time follows a Log-normal distribution and the expected stay time of user k at POI c can be calculated as,

$$T^k_c = e^{\mu_c + \frac{1}{2}\sigma^2_c}\,. \tag{4.5}$$

In the above equation μ_c and σ_c represent the mean and standard deviation of stay time at POI c which can be obtained if sufficient information about a tourist's stay time in that POI is available. Otherwise, a generalized average stay time can be obtained from statistics provided by the tourism research department [171]. Finally, remaining stay time at a POI c can be calculated as,

$$T^k_{c,r} = T^k_c - T^k_{c,s}\,, \tag{4.6}$$

where $T^k_{c,s}$ shows the time already spent by node k in 'c'. The value of $L^{k,d}_{next}$ can be calculated similarly using Eq. (4.4)-(4.6). Both requester and forwarder nodes calculate their own values and the requester attaches these values with the content request.

The weighting factor φ_c dictates which co-location stay probability (i.e., in the current POI or the next POI) should be given more priority and is calculated as,

$$\varphi_c = \begin{cases} 1, & \text{if} \quad T^k_{c,r} \geq TTL_\alpha \\ \frac{T^k_{c,r}}{TTL_\alpha}, & \text{otherwise}\,, \end{cases} \tag{4.7}$$

where TTL_α represents the expiry time of the content request α. If the destination node stays longer in the current POI than the request's life time, the co-location stay probability in the current POI is given the highest weight, otherwise nodes k and d have the possibility to meet in the next POI and the weight is determined by the ratio of remaining stay time and request lifetime.

The connectivity value and energy related information are calculated using the method discussed in Section 3.1.1 of Chapter 3. Users specify the amount of storage they are willing to allocate for the content sharing app and also for carrying other users' requests. By default, any suitable value can be used by the content sharing app. Node k checks the available storage space and the size of content α to decide whether to carry the content or not.

Finally, node k calculates its utility value using Eq. (4.2) and reports it back to the content carrier node. Upon receiving utility values from all of its one-hop neighbors, the carrier node checks who has the highest utility using Eq. (4.1). If the carrier node itself has the highest utility, it continues to carry the content, otherwise it forwards the content to the node offering highest utility. In case of a tie, the carrier node first checks the available energy and then the buffer space to take the forwarding decision. If all the values are the same, ties are broken by selecting a random node. In this manner, successive carrier nodes carry the content until it is successfully delivered or the request lifetime expires. Algorithm 2 outlines the message forwarding process, which is initiated when a carrier node Q_c receives a content or meets a new node.

Algorithm 2 Utility based message forwarding

procedure FORWARD MESSAGE
    //Q_C is the current carrier, Q_R is the destination and α is the content.
    if $h^{Q_C,Q_R} = 1$
        Deliver content α to Q_R;
    else
        Carrier node Q_C broadcast a summary of content α;
        Q_C calculates utility value $U_\alpha^{Q_C,Q_R}$ for delivering α using (4.2);
        Each potential forwarder node Q_F calculates $U_\alpha^{Q_F,Q_R}$ using (4.2) and send to Q_c;
        Q_c selects node Q_F with maximum utility i.e., $Q_F = arg(max(U_\alpha^{Q_F,Q_R}))$ using (4.1);
        if $U_\alpha^{Q_F,Q_R} > U_\alpha^{Q_C,Q_R}$
            Forward the content to Q_F;
        end if
    end if
end procedure

When Q_c receives a content α, it broadcasts a summary of α to obtain the utility values from its neighbors. Q_c also calculates its own utility value and compares it with the values received from its neighbors. Finally, Q_c selects the node with the highest utility value as the forwarder node, and accordingly forwards the content, if the selected node is a neighbor.

4.2 Scheme for Encouraging Participation

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. This is particularly important in irregular meeting places as nodes are mostly unfamiliar with each other and unlikely to cooperate without any proper incentive mechanism. Again, misbehaving nodes may try to obtain unfair benefits by submitting false claims. To address these issues, this section presents an incentive scheme to encourage participation and a trust management scheme to minimize security risks.

4.2.1 Participation Incentive

As highlighted in Chapter 2, works in the existing literature provide incentives in the form of monetary reward (e.g., virtual currency) [40, 142] or non-monetary (e.g., exchange of contents between nodes, if those contents are perceived as useful) [54, 59]. However, monetary reward is not directly applicable here because of the lack of any central server to log services provided by nodes and a coordinator for processing claims and payments/rewards, while the non-monetary rewards proposed in the literature need 'global knowledge' about content popularity and sufficient contact history to be known to each node. In our given scenario where people meet mostly strangers but still want to share contents through a collaborative participatory scheme, providing incentives by prioritizing service-providing nodes' requests so that they experience better delivery success rate is most appropriate. Therefore, 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 a 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, the 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. 4.2 with the role of the nodes as labelled. Here, node V_A represents the administrator of the group while V_C and V_R represent the content holder and requester, respectively. V_{F1} and V_{F2} represent two forwarder nodes. Although nodes are mobile and usually there is no end-to-end path from source to destination, a static communication path is presented in Fig. 4.2 for the sake of simplicity and better representation. In our scheme, a requester (V_R) sends a content request to the administrator which also includes information about its previous delivery report and pending service claim (Step 1). The delivery report includes information about

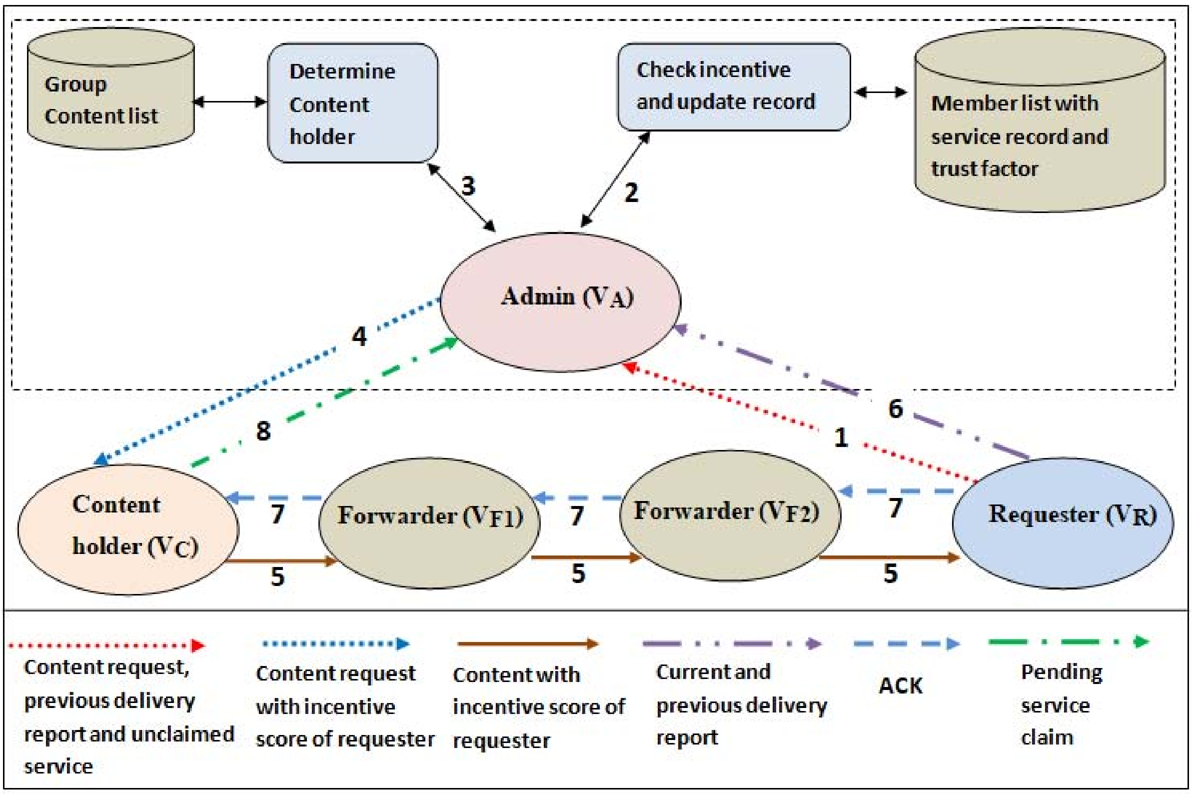

Figure 4.2: Participation incentive in decentralized content sharing

the delivery of a content previously requested by V_R. On the other hand, the service claim includes a report about deliveries in which V_R itself participated for delivering contents to other nodes. When the administrator receives the request, it checks the current incentive score of V_R and updates its trust factor (discussed in Section 4.2.2) based on any claim submitted by V_R (Step 2). The administrator forwards the request to the appropriate content holder V_C (Step 4) who delivers the content to the requester through forwarder V_{F1} and V_{F2} (Step 5). After receiving the content, V_R sends a delivery report to the administrator V_A (Step 6), notifying the receipt of the content from the holder (V_C) through the forwarders (V_{F1}, V_{F2}) as the delivery path is recorded. The report by V_R shows the request ID, list of participatory nodes (V_{F1}, V_{F2} and V_C in Fig. 4.2) and the size of the content delivered, and this report forms the basis of the incentive score calculation and update. In the case where a delivery report is lost on the way to the administrator, the current report also contains three previous reports with their respective request IDs so that any pending request can be processed. The administrator calculates the incentive score of each participatory node k for delivering a content α after receiving a delivery report as,

$$\Omega^k(t) = \frac{\varsigma_\alpha}{\varsigma_{max}}\left(1 - \frac{d_\alpha}{TTL_\alpha}\right). \tag{4.8}$$

Here, ς_α is the size of content α and ς_{max} is the maximum size of any content in the group content list. d_α represents the delay for obtaining content α and TTL_α shows the request lifetime for the same. We consider that nodes should be given higher incentive for carrying a larger content as well as delivering it within a shorter time. Therefore, a combination of content size and delay is used in Eq. (4.8). Finally, the administrator updates the incentive score of node k as,

$$\Omega^k = \varphi_i \Omega^k(t) + (1 - \varphi_i)\Omega^k(t-1), \tag{4.9}$$

where φ_i represents a weighting factor for exponential moving average calculation and $\Omega^k(t-1)$ is the previous incentive score of node k. The value of φ_i is calculated using the weight calculation method for exponential moving average discussed in Section 3.1.2.1. Each node is assigned an initial incentive score of 0.1 after joining a group, which can be increased by providing service to others.

The reward based on incentive score is implemented as follows. On receiving a content request from V_R, the administrator checks its current incentive score from the updated incentive list of all the 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 (V_C), the administrator also includes the requester's incentive score. V_C adds this new request to its pending request queue and processes all requests in order of decreasing scores whenever it meets a forwarder V_{F1}. Similarly, forwarders also check their pending request queue and forward 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 V_R. Processing requests on the basis of incentive exhibits benefits for active nodes in terms of higher delivery success rates and lower delays, as demonstrated using the simulation results in Section 4.3.

4.2.2 Trust Management

Trust management is another important aspect of implementing decentralized content sharing in irregular meeting places as nodes are mostly strangers and unaware about the trustworthiness of other users. The DSIP method presented in the previous chapter uses a self-assessment value to select an administrator where a node calculates its own value and broadcasts it among its neighbors. A question then arises whether the value put forward by each node can be trusted by others. Without any proper trust management scheme, it becomes impossible to determine whether a node is trustworthy. 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. Therefore, it is very important to identify nodes with ill-intention and filter them out while selecting an administrator. In this regard, we propose a trust management scheme where, once a group is established, the administrator maintains the trust factor of the nodes to assess their trustworthiness by observing how well behaved they are.

The trust factor is used to identify misbehaving nodes and prevent them from becoming the administrator of a group. The trust factor measures a node's inclination to claim forged value, i.e., a node who tends to make false claims for incentive is likely to forge the 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 comparing it with the delivery report the administrator receives from the requester. The administrator maintains and updates trust factors 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, as discussed in the previous section, but the trust factor is updated only after a service claim is lodged and verified. Refer to Fig. 4.2, when V_R sends a delivery report to V_A, it also sends an acknowledgement along the delivery path to nodes V_C, V_{F1} and V_{F2} (Step 7 indicated by dotted line) and the nodes claim service (Step 8) 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,

$$\Phi^k(f) = \begin{cases} \frac{1}{2}\left(\Phi^k(r-1) + \nu_h^k/\nu_T^k\right), \text{if honest claim} \\ \Phi_{med}\left(1 - \nu_f^k/\nu_T^k\right), \text{otherwise}. \end{cases} \tag{4.10}$$

Here, $\Phi^k(f)$ shows the trust factor of k after its f-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 is calculated by the administrator. The above formulation limits the trust factor between [0∼1], gradually increases the trust factor for well-behaving nodes towards the high value, and punishes misbehaving nodes by lowering their value below Φ_{med}, ensuring their exclusion from administrator selection in the succeeding round.

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 re-sends the report with its next content request and next delivery report; each report with corresponding request ID. If the administrator

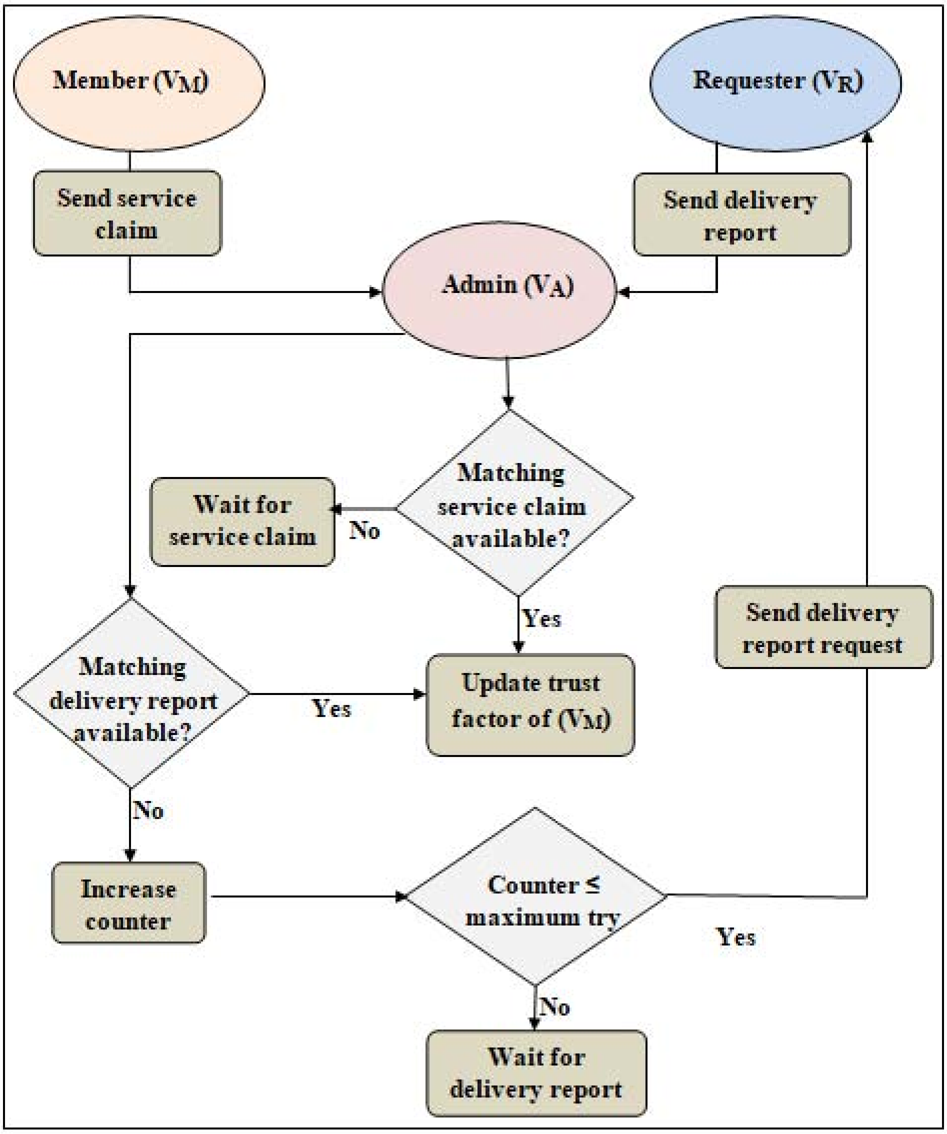

Figure 4.3: Trust factor update process

has not received any delivery report for a service claim, it sends a query message to which the latter 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. Figure 4.3 shows a flowchart for the trust factor update process followed by the administrator.

Minimizing Security Risk: We consider that the content sharing application will execute in a secured environment in memory to protect against the spread of malicious contents or programs by other nodes as in [25, 33]. In this regard, a received content can be first scanned in a virtual environment for it to expose any malicious action and stored in local memory only if found safe. Moreover, during the installation process of the content sharing application, the user of a mobile device will be required 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. Using IMEI has the advantage that a node can relate requests and other information coming from an IMEI only, not being able to link directly with the user associated with the IMEI, and this provides better protection against privacy concerns. 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 checks with identifiable details will act as a deterrent for users to perform any malicious activity as the devices/users can be traced.

Most 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, the content sharing application will seek the consent of users during installation. Additionally, the users must consent to the condition that they will not disclose any data that their own device receives about others for the sharing application to work.

In irregular meeting places, nodes are mostly strangers and perhaps meeting for the first time. Therefore, the trustworthiness information is not available for the very first time a group is to be formed. To address this, the initial administrator selection process is revised in the E-DSIP method to include 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 neighborhood tables. These tables are then shared to build a complete list of all nodes willing to form a new group, which constitutes the voter list. Since initially a small number of nodes is expected to be present, 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). This selected node then collects self-assessment values 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 malicious node from declaring itself as an initiator and (ii) random trust factor assignment at the second stage may assign a low value to a malicious node leaving it out of the candidate list for administrator selection.

During the subsequent administrator selection process, the current administrator of a group uses the trust factors of the group members to filter out misbehaving nodes and exclude them from the pool of candidates for selecting an administrator. In this regard, Eq. (3.1) in Chapter 3 is modified as,

$$\begin{aligned} &\textit{select} \quad \kappa \\ &\textit{maximize} \quad P^{\kappa} C^{\kappa} \\ &\textit{s.t.} \quad E_f^{\kappa} > E_{min}^{\kappa} \quad \textit{and} \quad \Lambda^{\kappa} > \Lambda_{min} \quad \textit{and} \quad \Phi^{\kappa} > \Phi_{med}\,, \end{aligned} \tag{4.11}$$

where the trust factor Φ^{κ} is used as a hurdle for the administrator selection process.

4.3 Simulation Environment and Results

We used the same simulation environment and dataset described in Section 3.2.1 to assess the performance of the E-DSIP method. Similar to Chapter 3, we use three metrics for performance evaluation, namely (i) hit rate, (ii) delivery success rate and (iii) average delivery latency. Additionally, we also investigated the control message overhead, energy of forwarder nodes and the impact of the proposed incentive and trust management schemes. The proposed E-DSIP approach is compared with the DSIP approach presented in Chapter 3. Both DSIP and E-DSIP use the same group formation and administrator selection techniques. Therefore, a comparison with DSIP shows the improvement achieved for the proposed utility based forwarding scheme. In addition, E-DSIP is also compared with SPOON [33]. Note that for fair comparison, the Spray-and-Wait [49] message forwarding was used for both DSIP and SPOON in the previous chapter. However, the original message forwarding approach in SPOON uses an interest oriented routing technique which is used in this chapter for SPOON. The t-test comparing E-DSIP with other methods in terms of hit rate, success rate and delivery latency at various arrival and request rates yielded p-values, $p \leq 4.03 \times 10^{-5}$, $p \leq 8.75 \times 10^{-6}$ and $p \leq 1.06 \times 10^{-5}$, respectively at 99% confidence level, validating their performance differences as being statistically significant. Each simulation was run 20 times and the results presented in this section are the average of 20 runs.

The effect of different node arrival rates is presented in Figs. 4.4 and 4.5. Both figures show that hit and success rates increase with a higher node arrival rate as more nodes are available to participate in the sharing process. The E-DSIP approach achieves the highest rate in all the cases with a hit rate of 68.57% and success rate of 66.25%, compared to 63.48% hit rate and 59.13% success rate achieved by the DSIP approach for an arrival rate of 25 nodes/hr. In comparison, SPOON achieves much lower hit (49.86%) and success rates (48.46%) for the same arrival rate (i.e., 25 nodes/hr). The improvement of the E-DSIP over the DSIP approach is because of the utility based message forwarding method which selects a forwarder node based on co-location stay time, connectivity value and available resources. A higher co-location stay time increased the number of opportunistic contacts between the forwarder and the requester nodes as they remained in the same POI for a longer time. In addition, the higher connec-

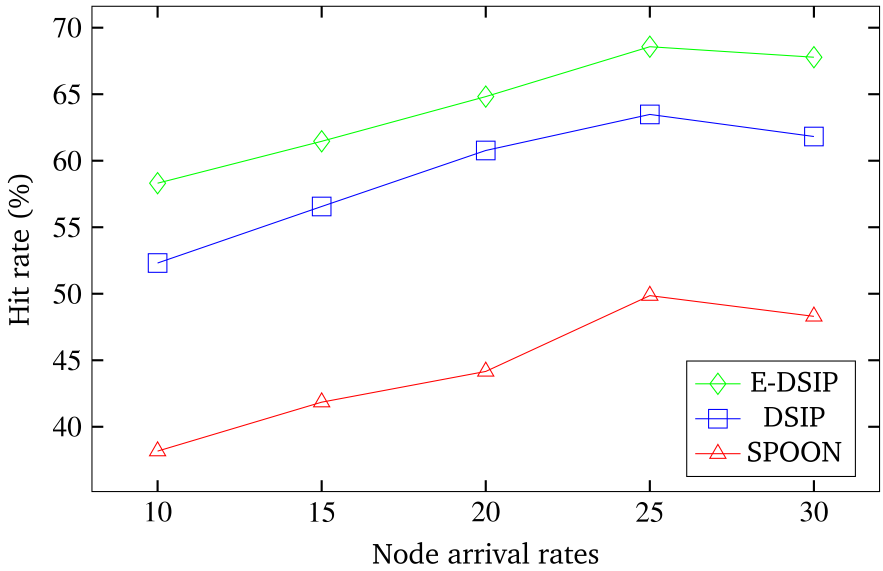

Figure 4.4: Hit rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr

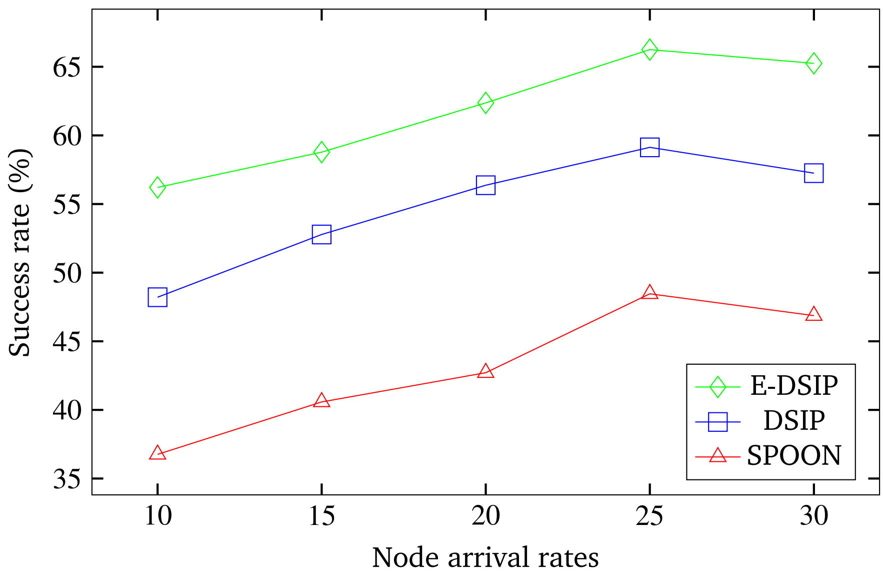

Figure 4.5: Success rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr

tivity value ensured that the selected forwarder node had more neighbors to facilitate successful delivery, and finally available resources ensured that the forwarder was capable of carrying the content for a longer time. Another observation is that because of the utility based message forwarding policy, the difference between the hit and success rate for E-DSIP is also lower than that of DSIP, as more often the requesters were able to receive their contents successfully when the requested contents were located.

We varied content request rate to see its impact on hit and delivery success rates and the results are presented in Figs. 4.6 and 4.7. Both figures show that hit and success rates initially start to increase with an increasing request rate. This is because, at a higher request rate, nodes from the same neighborhood generate more requests for the same content which, in turn, enable them to successfully receive that content from a neighbor who has already requested and received that content. However, after a certain request rate (5 requests/hr in this setting) both rates start to decrease as the network becomes congested with a higher request rate. The E-DSIP approach achieves a 66.27% hit rate and 63.78% success rate even when the request rate is very high (6 requests/hr). In contrast, other approaches achieve comparatively low hit (DSIP 60.33% and SPOON 46.84%) and success rates (DSIP 54.11% and SPOON 45.42%) for the same request rate (6 requests/hr). The improvement of the E-DSIP method compared to DSIP and SPOON ranges from 5-20%, suggesting that it is capable of handling a higher number of requests for DCS in irregular meeting places. Note that E-DSIP uses only a single copy of a content for successful delivery while DSIP uses 2-

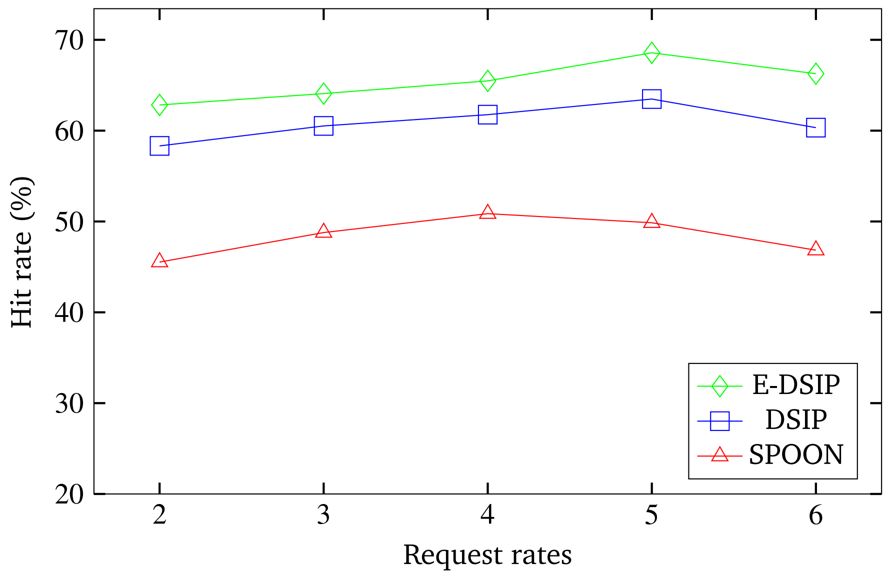

Figure 4.6: Hit rate for different mean request rates (number of requests/hr) keeping the mean node arrival rate fixed at 25 nodes/hr

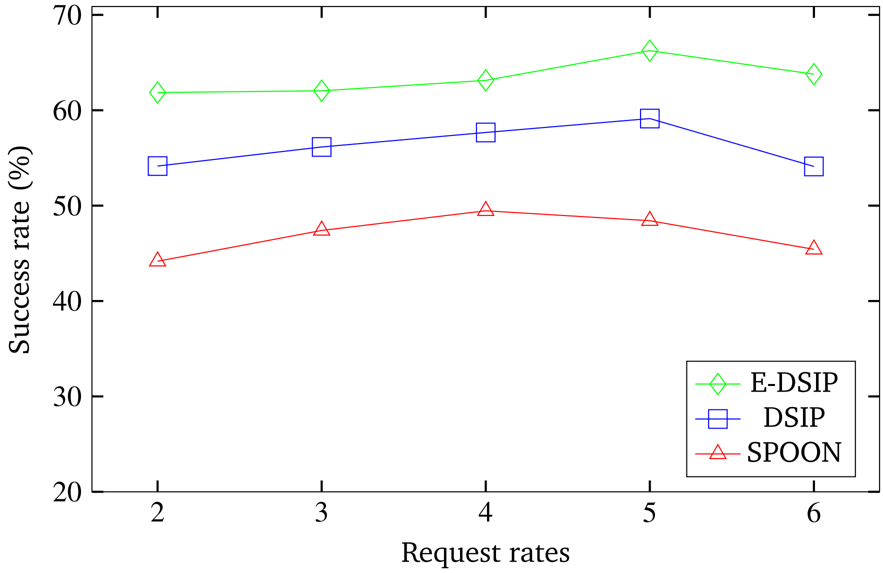

Figure 4.7: Success rate for different mean request rates (number of requests/hr) keeping the mean node arrival rate fixed at 25 nodes/hr

copies for the same. In this regard, E-DSIP is able to minimize resource consumption along with improving delivery performance.

Fig. 4.8 represents the average latency for successfully receiving contents. The latency values for successful deliveries are sorted in ascending order and averaged over various intervals for convenience of presentation. The latency is comparatively low up to 4×10^3 successful transfers across all approaches (e.g., E-DSIP 0.35 mins, DSIP 0.40 mins and SPOON 0.51 mins). Afterwards from 4×10^3 to 4.8×10^3 requests, the E-DSIP approach shows significantly low delays (4.38 mins) compared to other approaches (DSIP 11.16 mins and SPOON 52.32 mins). The reason for getting a comparatively low delay for E-DSIP is that it selects a forwarder node with higher connectivity, and hence has a better opportunity of delivering the content faster through more neighbors. The figure also suggests that the E-DSIP method is capable of delivering 6.6×10^3 requests successfully while DSIP and SPOON can handle 6×10^3 and 4.8×10^3 requests, respectively. Since the E-DSIP method uses available resources as one of the criteria for selecting a forwarder node, it can intelligently select nodes who are capable of carrying a content for a longer amount of time for successful delivery without running out of resources. In this manner, the E-DSIP method achieves a higher number of successful deliveries within a shorter time than other approaches.

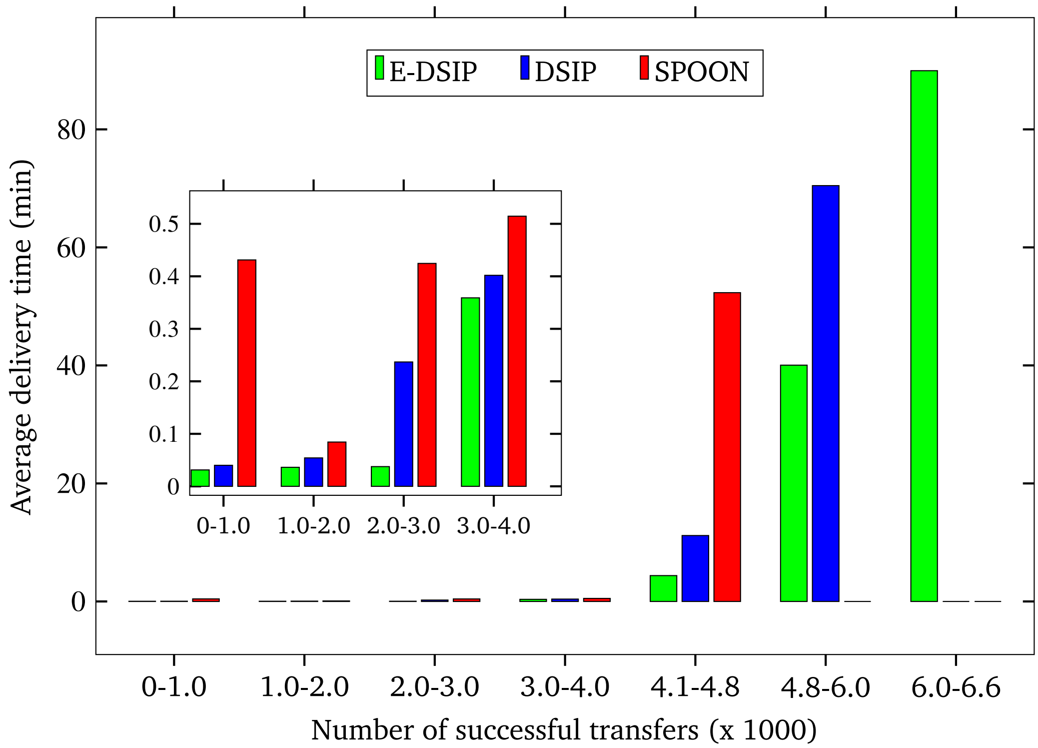

Figure 4.8: Average delivery latency for successful deliveries for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

To handle different aspects of the E-DSIP method, such as group formation, message forwarding, incentive and trust managements, we used control messages to maintain communication among nodes. The percentage of control message overhead across various node arrival and content request rates is presented in Figs. 4.9 and 4.10. The control message overhead is calculated as the ratio of the total size of control messages to the total size of successfully delivered contents. Both the figures show that the control message overhead decreases along with an increasing node arrival or content request rate. The number of successfully delivered contents increases with an increasing node arrival or content request rate, which consequently increases the total size of successfully delivered contents and results in lower message overhead. The E-DSIP method generated lower message overhead than DSIP, but a slightly higher overhead than SPOON across all arrival and content request rates. The E-DSIP method used a neighborhood table to calculate connectivity of nodes and used it for message forwarding and the group formation process. In addition, E-DSIP also used delivery reports and service claims to handle incentive and trust management issues. All of these

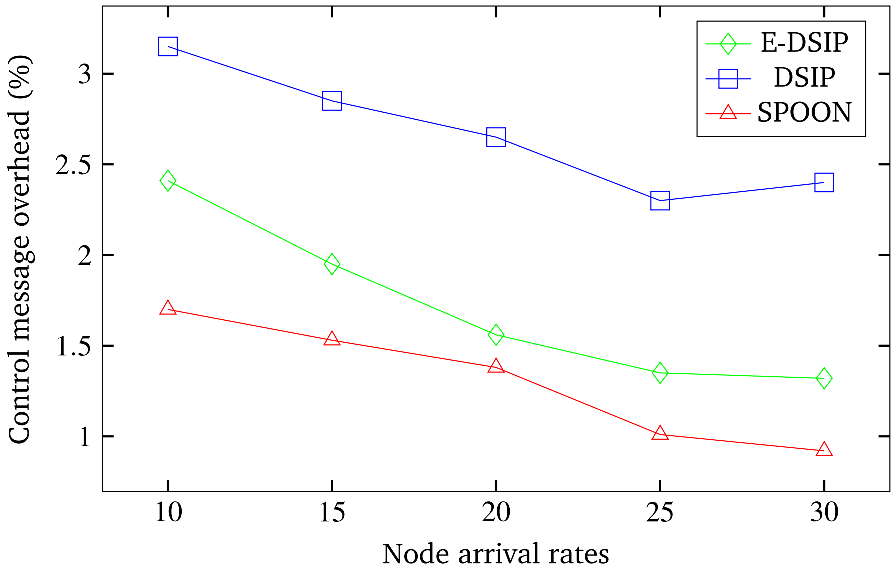

Figure 4.9: Control message overhead for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

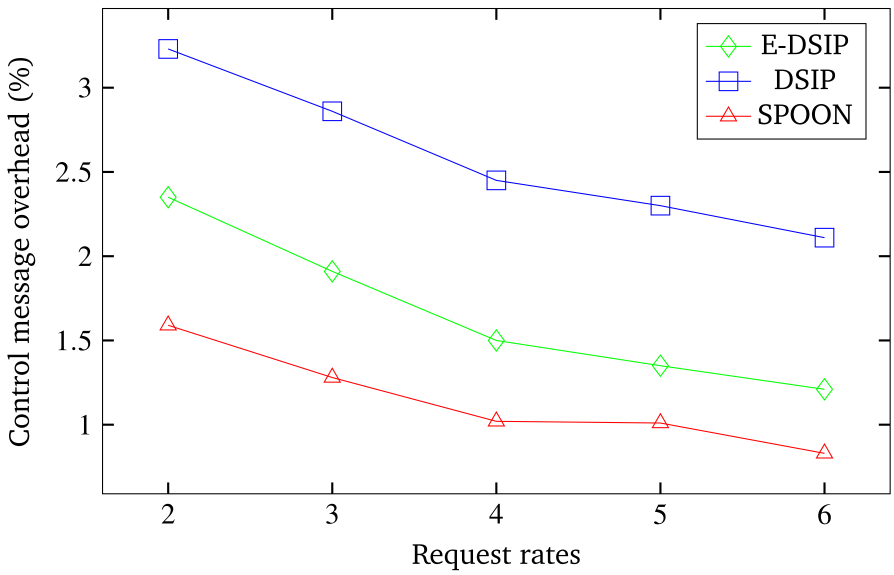

Figure 4.10: Control message overhead for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

contributed to a slightly higher message overhead for E-DSIP compared to SPOON. Since the delivery success rate of E-DSIP is much higher than DSIP, the percentage of message overhead is lower for E-DSIP, as both of them use similar control messages for group formation and administrator selection.

The proposed E-DSIP approach uses the co-location stay probability in the current POI as well as in the next POI for selecting a forwarder node. The impact of such a selection policy is presented in Fig. 4.11. Here, the 'Current POI' indicates that the requester was able to successfully receive a content in the same POI where it generated the request, while the 'Next POI' indicates that after generating the request, the requester moved to the next POI and received the content. The figure shows that the E-DSIP approach achieves highest successful deliveries in the current POI (E-DSIP 61.31% vs. others 57.73% max) as well as in the next POI (E-DSIP 4.82% vs. others 1.40% max). The reason for obtaining a higher rate is that the node offering a higher co-location stay time with the destination is selected as the forwarder node which, in turn, ensures that the forwarder node will stay in the same POI as the destination node, even if one of them moves to the next POI before successful delivery.

The proposed E-DSIP method uses the available energy at a node for selecting an appropriate forwarder for carrying a content. The impact of such an energy aware forwarder selection policy is presented in Fig. 4.12, which shows the remaining energy of two nodes (Node# 3 and Node# 5) during simulation. Although Node# 3 and Node# 5 were selected as forwarders for carrying six and five contents, respectively,

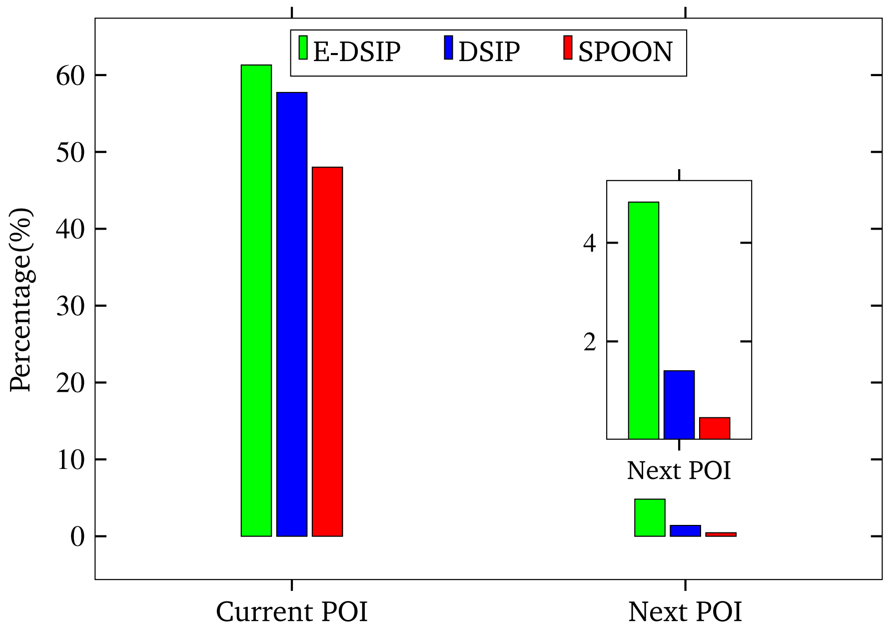

Figure 4.11: Successful content delivery and co-location of nodes for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

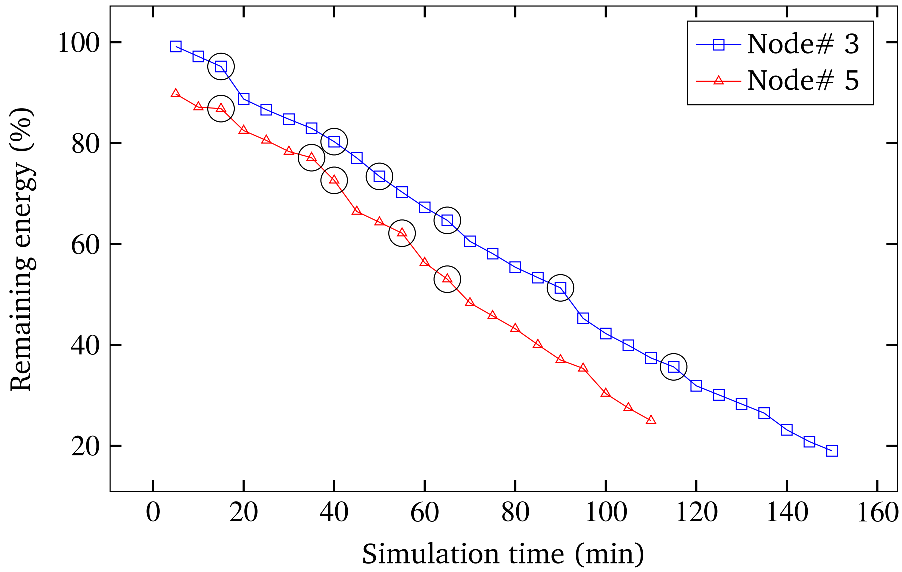

Figure 4.12: Remaining energy of forwarder nodes before the next charging cycle (a circle represents the time in simulation when the nodes were selected as forwarders and carried a content for delivery)

they did not run out of battery before their next charging cycles. The reason for this is that the selection process assessed the remaining energy of a node and only copied a content, when the node had sufficient energy to carry it further.

4.3.1 Impact of Incentive Mechanism

The E-DSIP approach incorporates an incentive mechanism to encourage node participation, as discussed in Section 4.2.1. Figure 4.13 shows the impact of the proposed incentive scheme. 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 a higher incentive score experiences better service for its requests in return for the service it provides to the group. For example, nodes with high (0.9∼1.0) incentive scores experienced greater hit and success rates than those experienced by nodes with low (0.0∼ 0.3) incentive scores (78.21% and 77.34% vs 57.27% and 52.23%). The reason for this is that nodes with higher incentive scores received higher priority from the administrator and forwarders for receiving contents. The figure also shows that the difference between hit and delivery success rates is higher for nodes with low incentive

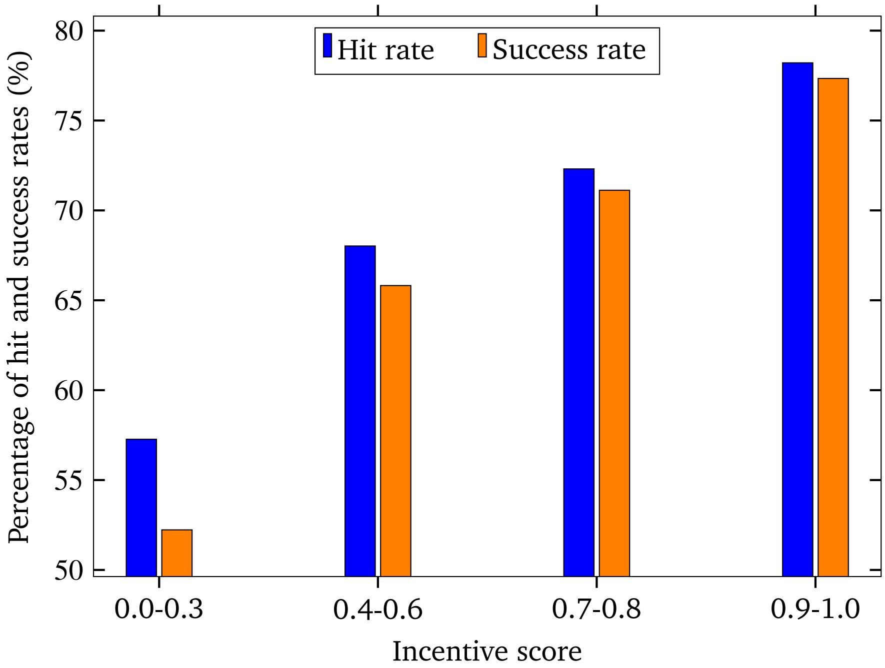

Figure 4.13: Impact of incentive score on hit and success rates for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

scores, suggesting that they were unable to receive the contents in some cases because of low priority treatment of contents by the forwarding nodes, though the requested contents were located. This feature suggests that a node can only gain a better sharing experience by providing service to others and this works as an encouragement to participate in the sharing process.

Figure 4.14 shows the average delivery latency experienced by nodes with different incentive scores for successful deliveries. Similar to Fig. 4.13, we observed the latency experienced by nodes in the 'Food and Shopping' POI. The figure shows that nodes with high (0.9∼1.0) incentive scores received their contents within a short time compared to nodes with lower (0.0∼ 0.3) incentive scores (on average 8.34 mins vs 28.25 mins). Since nodes having low incentive scores receive lower priority from the administrator and forwarder nodes, it takes a significantly higher amount of time to receive the requested contents. In this manner, nodes are encouraged to increase their participation by carrying contents for others and in return they will receive faster delivery for their requested contents.

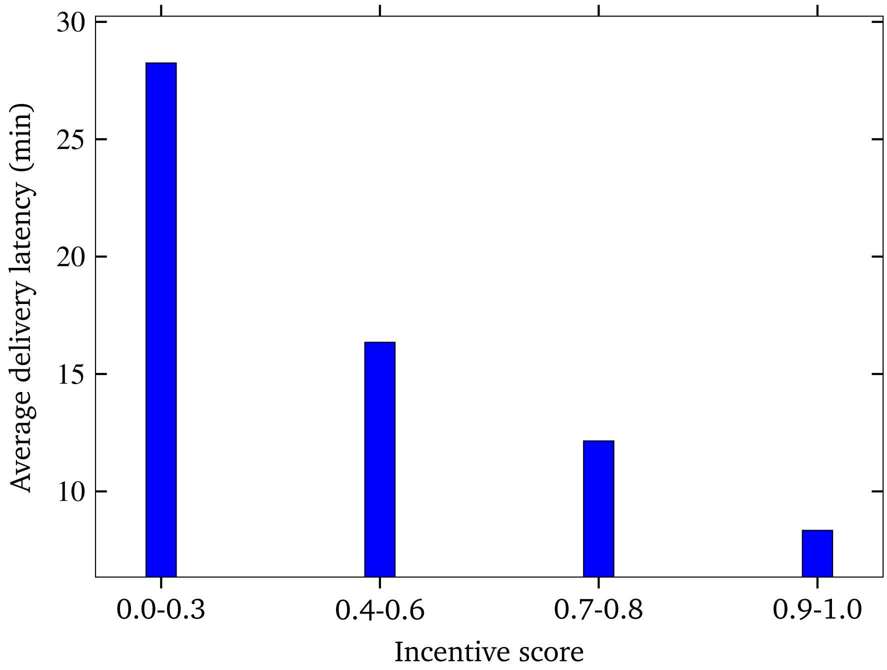

Figure 4.14: Impact of incentive score on average delivery latency for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

4.3.2 Impact of Trust Factor

As described in Section 4.2.2, we used a trust factor to identify misbehaving nodes and exclude them from the pool of candidates for the administrator selection process. Figure 4.15 shows the way the trust factor of two nodes (Node# 52 and Node# 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 its trust factor increased because of making honest claims. Therefore, this node always passed the hurdle to be a candidate for the administrator selection process (Eq. (4.11)). 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 was prevented from being a candidate for the administrator selection process. In this manner, the proposed E-DSIP approach minimizes risks.

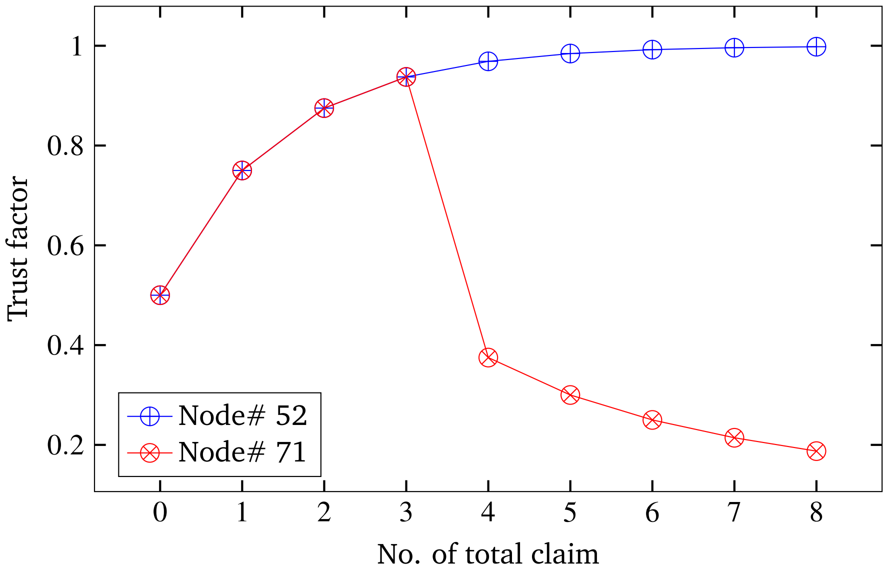

Figure 4.15: Change of trust factor in 'Food and Shopping' POI. Here, 0-th claim indicates initial scenario where every node has same trust factor

4.4 Conclusion

This chapter has presented the E-DSIP approach by incorporating a utility based forwarding technique, an incentive mechanism and a trust management scheme. Message forwarding plays a major role in successful content delivery in decentralized content sharing. In this regard, a utility based message forwarding technique is presented in this chapter which selects a forwarder node based on the node's co-location stay probability with the destination, connectivity with neighbors, and available energy and buffer spaces. Simulation results demonstrate that the incorporation of this utility based forwarding technique in the E-DSIP approach has improved the content delivery service in terms of hit and success rates, and delivery latency, compared to the DSIP approach presented in the previous chapter. In addition, the proposed forwarding technique also alleviates the requirement of having a routine movement pattern or pre-existing social relationship, which is primarily used in the existing literature and mostly non-existent in the environments under consideration (i.e., irregular meeting places).

To handle selfishness of nodes and encourage participation in the sharing process, an incentive scheme is presented which gives rewards in the form of priority processing. Simulation results demonstrate that nodes with higher incentive scores achieved better content sharing experiences in terms of delivery success rates and latency. To achieve a higher incentive score, nodes needed to participate and carry contents for other nodes. This has certainly worked as an encouragement to get more involved in the sharing process to obtain better service. A light-weight trust management scheme is also presented which identifies misbehaving nodes trying to make false claims about their participation and excludes them from the pool of candidates for becoming an administrator. Adoption of such a policy minimizes the security risks.

The E-DSIP approach produces significant improvement over the DSIP approach, however, it will face difficulty in providing a satisfactory content delivery service if the demand for contents rises sharply. In visiting hotspots, demand for contents usually increases rapidly because of a high number of visitors. Again, the supply of contents can also change because of users leaving the group or having insufficient resources. In this context, a content distribution scheme is needed to provide a satisfactory content delivery service, which is discussed in the next chapter.

Chapter 5

A Content Distribution Scheme in E-DSIP Based on Dynamic Demand and Supply

The preceding chapter has presented an enhanced decentralized content sharing method for irregular meeting places (E-DSIP) that has improved content delivery service and encouraged node participation. However, it does not focus on content demand and supply that vary considerably because of visitor in-and-out flow changes and the occurrence of influencing events in tourist spot type scenarios. This is further compounded by the lack of any content distribution (replication) scheme considering dynamic demand and supply. The content delivery success rate and content access time suffer if the demand for contents surpasses their available supply. In this regard, to improve the content delivery service, contents are to be proactively distributed in strategic positions. To address these issues, this chapter presents a content distribution scheme that calculates dynamic demand and supply for contents, determines which contents need additional distribution and accordingly proactively distributes contents. We call our previously proposed E-DSIP (in Chapter 4) method employing this content distribution scheme as *DSIP-CD*. The content distribution scheme uses a joint optimization model of dynamic demand and supply and nodes' contention to identify appropriate nodes for placing distributed contents for improving content delivery service and time (**Block 4 - objective 3** Fig 1.6). Extensive simulation has been performed to validate the performance of the DSIP-CD approach and simulation results demonstrate that the adoption of such a content distribution scheme yields higher delivery success rates and lower latency that are comparable to those proposed for work-place type scenarios.

5.1 Embedding Content Distribution Scheme in E-DSIP

Both DSIP-CD and E-DSIP uses the same group formation and administrator selection policies presented in Chapter 3, and message forwarding, incentive and trust management techniques articulated in Chapter 4. In addition, the DSIP-CD approach focuses on employing a content distribution scheme using the economic model of dynamic demand, supply and distribution.

Demand-supply-and-distribution is an extensively studied topic in economics and widely exercised for product distribution in business and retail organizations. The basic idea is to increase supply in response to an expected high demand. In addition, the delivery of products at the right time to the right people depends on the selection of appropriate distribution locations [184]. The guaranteed delivery requires the communication infrastructure (e.g., roads and highways) to be available from the distribution center to the demanding customers who are reachable within minimum transportation cost [185]. In our case, if we can estimate future content demand and accordingly distribute contents proactively in strategically positioned nodes so that any possible request from a nearby node is better served considering the joint optimization of demand-supply and distribution based on economic theory along with transmission channel condition, it is highly likely that both delivery and access time will be improved. However, serving high demand for contents with limited resources is a multi-dimensional problem and a content distribution policy needs to address: (i) future prediction of demand, (ii) determination of available supply and (iii) the distribution of contents considering dynamic demand-supply and contention among nodes for wireless medium access.

Though the content distribution problem can be compared to the demand, supply and distribution method for product distribution, one significant difference between them is that the volume of demand for a product usually depends on its price and the income of the buyers, while demand for a content depends on its popularity, interest of participating users and external event occurrence during the visiting time. This dictates modeling of demand differently in collaborative content sharing. For content distribution, demand is considered as the number of expected content requests, supply represents content availability, and distribution refers to the placements of replicated contents. When expected demand surpasses available supply, content distribution is performed to increase supply and availability, and thereby meet the future demand. In DSIP-CD, the group administrator maintains and updates the group content list (i.e., list of contents for all group members) similar to DSIP presented in Chapter 3. In addition, the administrator periodically performs the content distribution (CD) using this list in DSIP-CD. The interval between consecutive CD cycles is denoted by θ. A lower value of θ indicates that the administrator consumes more resources for handling the distribution procedure, while a higher value suggests that the distribution scheme will fail to capture sudden change (e.g., more requests during an event). At the beginning of each distribution cycle, the administrator calculates demand and supply for each content in the group content list, determines which contents require additional distribution and identifies the group members who are best positioned for carrying the replicated contents. Afterwards, it instructs the identified group member(s) to copy the content from one of the suppliers. Different stages of our proposed distribution scheme (demand, supply and distribution) are presented in the following sections.

5.1.1 Capturing Dynamic Content Demand

Content request is motivated by a user's interest in a particular content category. Content category refers to contents related to activities available in a tourist attraction (e.g., contents related to accommodation, food and shopping facilities, fishing, camping, bush walking) or generic media types (e.g., music, movies or news items). In addition, events may also motivate users to request relevant contents. For example, Bob is usually not interested in fishing but while visiting a tourist spot he might learn about a fishing competition at that place, and becomes interested in receiving contents related to that event. Similarly, a major event (e.g., bush-fire) is likely to prompt everyone to receive updated news. Note that such impact of influencing events on content demand is not considered in the current literature, although it has been reported that content request increases because of an influencing event [46]. To capture both personal interest and the impact of an influencing external event on content demand, the administrator calculates the demand for content α belonging to category i as,

$$D_\alpha = D_{i,\alpha} + D_{i',\alpha}. \tag{5.1}$$

Here, $D_{i,\alpha}$ represents the demand generated by users who are already interested in category i to which content α belongs and $D_{i',\alpha}$ shows the expected demand by members who are currently not interested in i but are expected to become interested in α if it is related to an influencing external event. If α is not related to any event, $D_{i',\alpha} = 0$ which nullifies its impact in Eq. (5.1). Demand for α from users interested in category i is calculated as,

$$D_{i,\alpha} = \sum_{x=1}^{|\mathbb{G}_{i,\alpha'}|} \mathscr{I}_{\alpha}^{x} P_{\theta}^{x} R_{\theta}^{x}. \tag{5.2}$$

Here, $\mathbb{G}_{i,\alpha'}$ represents the set of group members who are interested in category i, but *do not* have content α. $\mathscr{I}_{\alpha}^{x}$ represents interest factor of node $x(\in \mathbb{G}_{i,\alpha'})$ for α, P_{θ}^{x} represents the stay probability of node x within the group during θ duration, and R_{θ}^{x} represents the probability of generating any content request within θ. Since users stay in a group for a limited time, demand expected within that period must be considered. Here, the interest factor indicates that a user x is interested in α, however we do not know when this user will generate a request and whether it will be staying in the current group during the current θ interval. Therefore, we calculate the demand as a combination of interest factor, stay probability and request probability. For a single node $D_{i,\alpha}$ ranges within $[0 \sim 1]$ suggesting its probability of requesting α. The calculation of each component is discussed below.

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

$$\mathscr{I}_{\alpha}^{x} = \Upsilon_{i}^{x} \Delta_{\alpha}^{x} + (1 - \Delta_{\alpha}^{x}) \Psi_{\alpha}, \tag{5.3}$$

where, Υ_{i}^{x} shows the interest *score* of user x for content category i, and Δ_{α}^{x} and Ψ_{α} depicts the personal preference and the trending scores (i.e., popularity) for α, respectively.

The interest *score* for a content category Υ_{i}^{x} represents the generic level of interest of a user for a particular category of content whose calculation is presented in Eq. (3.15) in Chapter 3. However, to reflect personal preference and the popularity of content α, Δ^x_α and Ψ_α have been included in this chapter to make it more practical. The group members send their Υ^x_i to the administrator while joining the group, as mentioned in Chapter 3.

Similar to the work presented in [33], we consider that the content sharing app can automatically collect the personal preferences of a user for content access using a document clustering technique [186]. In this case, a file vector [33] is generated for each file from the metadata associated with previously requested files. The file vector for content α having n number of keywords is depicted as, $\mathbf{v}_\alpha = (q_{1\alpha}, w_{1\alpha}; q_{2\alpha}, w_{2\alpha}; \ldots; q_{n\alpha}, w_{n\alpha})$, where $q_{n\alpha}$ represents the n-th keyword and $w_{n\alpha}$ represents its weight in describing the file containing α. The raw weight of each keyword is extracted using text retrieval techniques [187] as, $w_{q\alpha} = 1 + log(f_{q\alpha})$, where $f_{q\alpha}$ is the frequency of keyword q in the file. The normalized weight for each keyword associated with a file is calculated as, $\overline{w_{q\alpha}} = w_{q\alpha} / \sum_{p=1}^{p=n} w_{p\alpha}$. Finally, the weight of keyword q across all files is calculated as,

$$w_q = \frac{1}{\mathcal{F}_\alpha} \sum_{\alpha=1}^{\mathcal{F}_\alpha} \overline{w_{q\alpha}}. \tag{5.4}$$

Here, $\mathcal{F}_\alpha$ indicates the total number of files. Each user calculates its own set of top-n preferred keywords and their associated weight, stores them in its preference vector $(\mathbf{v}^x = (q^x_1, w^x_1; q^x_2, w^x_2; \ldots; q^x_n, w^x_n)$ and sends it to the administrator while joining a group. Similarly, the content holders also send file vectors $\mathbf{v}_\alpha$ while sending their content list summary to the administrator. Note that a higher value of n introduces more traffic while a lower value might result in potential mismatch between a user's preference with a content. The administrator determines the expected personal preference score of a user x for a content α using cosine similarity as,

$$\Delta^{x^*}_\alpha = \begin{cases} \frac{\sum_{q=1}^{Q} w^x_q * \overline{w_{q\alpha}}}{\sqrt{\sum_{q=1}^{Q} (w^x_q)^2} * \sqrt{\sum_{q=1}^{Q} (\overline{w_{q\alpha}})^2}}, \text{if} Q > 0 \\ 0, \text{ otherwise.} \end{cases} \tag{5.5}$$

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

Note that it is possible to find a complete or near mismatch ($Q \rightarrow 0$) between $\mathbf{v}^x$ and $\mathbf{v}_\alpha$. It will effectively nullify the impact of a user's interest in Eq. (5.3), which is undesirable as the mismatch might be a result of inaccurate tagging or insufficient information. Additionally, a user might still be interested in a particular content because of its generic interest in that category, and hence the personal preference score is calculated as,

$$\Delta_\alpha^x = max \left\{ \Delta_\alpha^{x^*}, \pi_i^x \right\}. \tag{5.6}$$

The value of π_i^x depends on the distribution of a user's interest across all N^x categories and its relative interest in category i. It is calculated as below.

$$\pi_i^x = \frac{\Upsilon_i^x}{\sum_{j=1}^{N^x} \Upsilon_j^x}. \tag{5.7}$$

Content demand by users can also be motivated by trending topics. Trending topics refer to the issues that have received significant attention from people in recent times. Now-a-days most of the social networking and content sharing platforms (e.g., Facebook, YouTube and Twitter) provide a list of trending topics. A list of such trending topics according to different locations along with a trending score is also available online (e.g., https://www.google.com/trends/ provides a list of trending topics and their scores between 0 and 100). The content sharing app can use such a trending list to assign a trending score within [0∼1] to each downloaded content. Note that updated information is not a requirement for sharing purposes, rather just aids in more accurate calculation; hence, the app will only access such information when an Internet connection is available. Such a trending score Ψ_α for content α is forwarded by the content holder to the administrator who uses it to calculate the interest factor using Eq. (5.3).

The stay probability P_θ^x of a node x indicates its probability of staying within the current POI. A higher stay probability suggests that the node is more likely to stay during current θ. The stay probability can be calculated using Eq. (3.2) in Chapter 3.

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

Poisson process [37] and hence the administrator calculates the probability of node x generating one or more requests within θ as,

$$\begin{aligned} R_\theta^x =& P(\aleph_\alpha^x >= 1), \; [\aleph_\alpha^x \text{is the number of request by} x] \\ =& 1 - P(\aleph_\alpha^x = 0) \\ =& 1 - e^{-\lambda_r^x \theta}. \end{aligned} \tag{5.8}$$

Finally, the administrator uses the value of interest factor $\mathscr{I}_\alpha^x$, stay probability P_θ^x and request probability R_θ^x to determine the demand for α from *interested* group members. Estimation of the demand from *uninterested* group members is discussed below.

As alluded before, it is expected that the occurrence of an influencing event will impact the uninterested users and some of them will eventually be interested in consuming related contents. The existing literature considers information diffusion among uninterested users and models the demand generated by them using the Bass diffusion model [164, 165]. However, they do not consider the presence of any influencing event to motivate users for demanding contents whereas, to attract tourists, many different types of events occur in a tourist spot. Again, they suggest that the ultimate number of interested users only depends on the initial set of interested users. We also argue that in a tourist spot, the scale of an external event determines what fraction of the uninterested users will become interested in consuming a content. For example, a small-scale event (e.g., a magic show) will have less impact than a mega-event (e.g., a popular carnival). Furthermore, access to event-related contents is expected to increase sharply immediately before and after the event, as well as during the event. This has not been considered in the literature. Based on these characteristics, we define the demand generated from uninterested group members as,

$$D_{i',\alpha} = \varpi \Psi_\alpha \, e^{-\beta t^2} \sum_{x=1}^{|\mathbb{G}_{i'}|} \varphi^x. \tag{5.9}$$

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

To determine the scale of an influencing event, we need to first match the content with a particular event and then determine the scale of that event. A number of recent works have focused on identifying events from social media and associating related contents with them [188, 189]. Becker *et al.* [188] suggests that using the features associated with a content (annotations like title, tags) and automatically generated information (e.g., content creation time), it is possible to map a content to an event. They further extended this work in [189] and suggested that collecting event features (e.g., title, time and location) from event aggregation platforms (e.g., Last.fm, EventBrite and Facebook) enables effective matching of events with contents. We consider that the content sharing app can use these event aggregation platforms or online resources (http://www.australia.com/en/events.html and http://eventful.com/ provide a list of events across Australia and USA, respectively) to collect and store information about future events and use the methods proposed in [189] to map a content to an event. The impact of events on the tourism industry and tourist places has been extensively studied by researchers from the perspective of the tourism industry and the economic and social impact of events on tourist spots [190, 191]. Getz [190] has ranked events into four categories based on the descending order of tourist demand and attractiveness as, (i) mega-events, (ii) periodic hallmark events, (iii) regional events and (iv) local events. Jago [191] investigated a number of events across Australia (42 in total), classified them into the above-mentioned four categories and assessed their attractiveness. We use these studies as a baseline to classify an event and accordingly assign an event scale (ϖ) within $[0{\sim}1]$. The calculation of trending score Ψ_α is similar to the one discussed before.

The shape factor determines the shape of the curve representing the impact of an event with time, especially immediately before or after and during that event. Some event-related contents become popular on the day of the event and vanish just after the event, while others may stay popular for a longer time. The content sharing app can use historical information to determine the value of the shape factor β within $[0{\sim}1]$ by observing the content access patterns of the user during similar events in the past. The literature suggests that event-related contents are usually generated and consumed from one week before the event up to one week after the event [189]. Therefore, by default, we use $\beta = 0.1$. Figure 5.1 shows the impact of shape factor at different times.

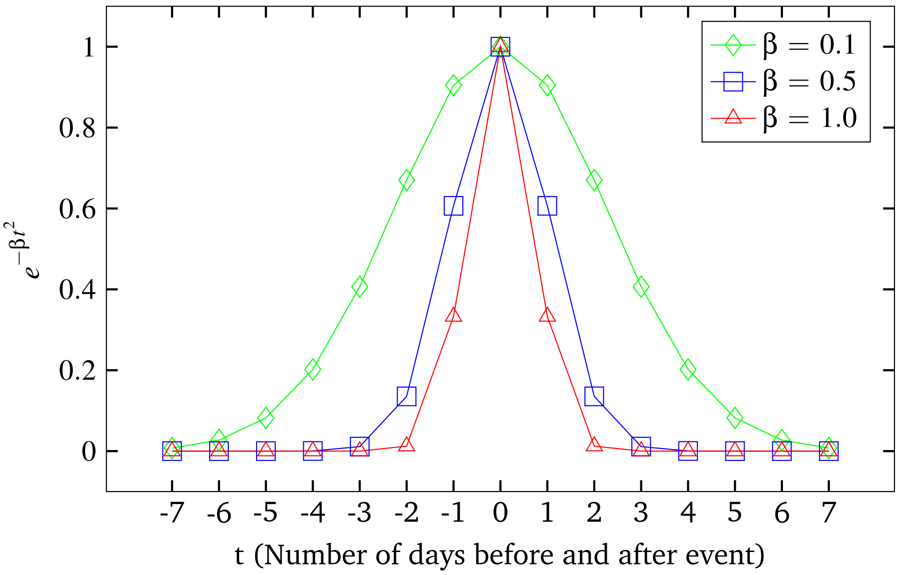

Figure 5.1: Impact of shape factor on event scale

A larger value suggests that the impact is usually short lived.

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

$$t = \begin{cases} 0, & \text{if } t_c = t_s \\ t_s - t_c, & \text{if } t_c < t_s \\ t_c - t_s, & \text{otherwise.} \end{cases} \tag{5.10}$$

Users' interest in different categories is reflected by their interest score. In addition, a user may become interested in a new category based on many situations. Such an attribute is called the affinity of a node towards a new interest category. From historical data, a node calculates its own affinity score φ^x using the ratio of the number of requests generated for contents in uninterested categories with respect to the total number of requests. Nodes send their φ^x to the administrator. Finally, the administrator calculates the demand from uninterested users using ϖ, Ψ_α, β, t and φ^x as per (5.9).

5.1.2 Content Supply

After calculating demand for a content α as per (5.1), the administrator calculates its available supply. In DCS, existing content holder(s) may leave the current POI or have insufficient resources to deliver α. Therefore, in contrast to the existing approaches' [164–166] assumption on guaranteed availability of content holders, we calculate dynamic supply using the number of content holder(s), their coverage, available resources and stay probability. A higher number of content holders increases the available supply of a content. Similarly, the coverage and stay probability of the content holder(s) also impact the perceived content availability. Coverage measures whether the content holder(s) is within the reach of the potential requesters, while stay probability predicts whether they are likely to remain in the group. Remaining energy is also used to assess whether a content holder has sufficient resources necessary for delivery. Finally, the supply for content α is calculated as,

$$S_\alpha = \sum_{y=1}^{|\mathbb{G}_{i,\alpha}|} C_\alpha^y P^y E_f^y. \tag{5.11}$$

Here, $\mathbb{G}_{i,\alpha}$ represents the set of members of group G who have content α, C_α^y shows the coverage of node $y \in \mathbb{G}_{i,\alpha}$ for content α, P^y shows the stay probability of node y, and E_f^y is the energy factor for node y. Eq. (5.11) is formulated considering the combination of C_α^y, P^y and Ψ^y as all have impact on the availability of α.

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

$$C_\alpha^y = |\mathbb{H}_1^{x,y}|, \tag{5.12}$$

where, $\mathbb{H}_1^{x,y}$ represents the set of potential requester x who are within one-hop distance of supplier y. The energy factor of node y is calculated based on its available energy and energy required to deliver α as,

$$E_f^y = \begin{cases} 1, & \text{if} \quad E_a^y \geq E_r^y \\ \frac{E_a^y}{E_r^y}, & \text{otherwise}, \end{cases} \tag{5.13}$$

Where, E_r^y shows the required energy for regular activities and delivering α, while E_a^y

shows node y's available energy, whose calculation is provided in Eq. (3.4)-(3.5) in Chapter 3.

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

5.1.3 Content Distribution

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

$$D_\alpha - S_\alpha > 0. \tag{5.14}$$

The purpose of the distribution mechanism is to proactively place replicated copies of the content in strategic positions so that they become easily accessible to potential requesters. The administrator considers the set of group members who do not have α ($\mathbb{G}_{\alpha'}$) as the potential locations for placing replicated copies and determines the utility (discussed later) of each of those nodes. The administrator first selects the node that maximizes the utility for placing a replicated copy and removes it from $\mathbb{G}_{\alpha'}$. Afterwards, the administrator recalculates the value of demand and supply, and checks (5.14) to determine if any additional copy is required.

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

$$\begin{aligned} &select \quad z \\ &maximize \quad \delta_\alpha^z = \phi_\alpha^z \tau^z \\ &s.t \quad E_a^z \geq E_r^z \quad and \quad B_a^z \geq B_r^z. \end{aligned} \tag{5.15}$$

In Eq. (5.15), δ_α^z shows the utility of node z for holding a replicated copy of content α, ϕ_α^z is z's demand coverage for α, τ^z shows the media access non-contention factor of node z. E_a^z and B_a^z depict node z's available energy and buffer space, respectively, while E_r^z and B_r^z indicate required energy and buffer space. We consider the non-contention factor in calculating the utility value of a node to keep the contention generated by the newly replicated content to a minimal level. Finally, ϕ_α^z ensures that z's demand coverage is also equally important to τ^z. For this reason, δ_α^z is defined as their combination.

Demand coverage indicates the amount of demand a particular node is expected to meet if a replicated content is placed in that node. The demand coverage of node z is calculated as,

$$\phi_\alpha^z = D_\alpha^z + \sum_{x=1}^{|\mathbb{H}_1^{x,z}|} D_\alpha^x. \tag{5.16}$$

Here, D_α^z is the demand generated by node z while D_α^x shows the demand generated by node x, who is located within 1-hop distance of node z.

The non-contention factor takes account of the shared communication medium (i.e., channel) among neighbors and the number of nodes that will be impacted if a replicated content is placed at a node which is expected to deliver the content when requested. Consider the scenario in Figure 5.2, where nodes that are expected to generate demand for content α are denoted by X_1, X_2 and X_3. Note that our demand calculation method considers that even an uninterested node may generate content demand, but for the sake of this example, we consider that only these three nodes will generate the demand. Y_1 and Y_2 represent two supplier nodes while nodes $X_1 \sim X_3$ and $Z_1 \sim Z_4$ are the candidates for placing the replicated content. Node Z_2 has five one-hop neighbors (X_1, X_2, Y_2, Z_3, Z_4) and node Z_1 has three one-hop neighbors (X_1, X_2, Y_1). Here, nodes X_1 and X_2 can only meet their own demand as none of their one-hop neighbors are generating any demand, while both nodes Z_1 and Z_2 are capable of meeting the demands of both X_1 and X_2. From the demand-meeting perspective, both nodes Z_1 and Z_2 are equally appropriate candidates. Now, if the replicated content is placed at node Z_1 or Z_2 and a request is made, then they will keep the shared medium (i.e., channel) busy while delivering the content. During delivery transmission, their neighbors will be unable to communicate. Therefore, we select a node that will cause least contention for its neighbors for delivering the content. In this case, the candidate

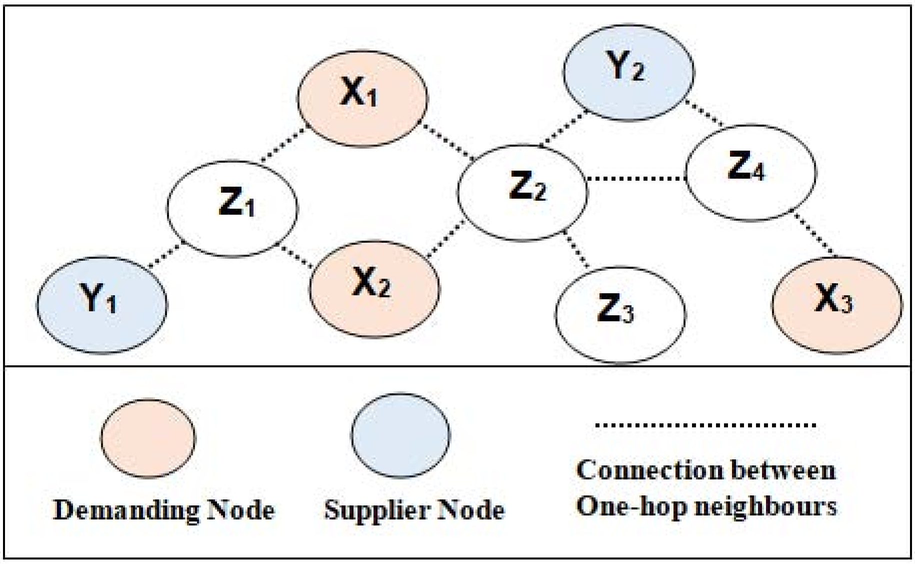

Figure 5.2: Selecting location (node) for replica placement

supplier node z will require t_α time to deliver α and will create no contention if none of its neighbors intend to transmit at that time. Hence, we calculate the probability of having no contention for node z as,

$$\tau^z = 1 - \frac{\sum_{u=1}^{|\mathbb{H}_1^z|} \eta^u}{\mathbb{G}}. \tag{5.17}$$

Here, η^u shows the probability that node u will be busy and will contend during transfer t_α by node z and $\mathbb{H}_1^z$ is the set of one-hop neighbors of z. To calculate the probability of node u being busy during t_α, we need to find its content exchange (delivering or getting requested content) probability and relay probability. We represent the probability of node u being busy in delivering, getting or relaying a content as,

$$\eta^u = min\left\{1, (\eta_d^u + \eta_g^u + \eta_r^u)\right\}, \tag{5.18}$$

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

$$\begin{aligned}\eta^u_{d,h_1} &= \sum_{v=1}^{|\mathbb{H}^u_1|} P(\aleph^v_\alpha >= 1).P\,(u\,\text{delivering the content to}\,v)\\ &= \sum_{v=1}^{|\mathbb{H}^u_1|}(1-e^{-\lambda^v_r t_\alpha}).\frac{|\zeta^u-\zeta^v|}{|\zeta_G-\zeta^v|}\\ &\quad\times\frac{|\zeta^u-\zeta^v|}{|\zeta^u-\zeta^v|+\sum\limits_{\substack{k=1\\k\neq u}}^{|\mathbb{H}^v_1|}|\{\zeta^u-\zeta^v\}\cap\zeta^k|}.\end{aligned} \tag{5.19}$$

Here, ζ^u represents the list of contents owned by node u. Similar to Eq. (5.8), the first part of (5.19) determines the probability of node v generating a request. The second part of the equation calculates the probability of u having a matching content as a ratio of contents currently possessed by u but not v and the contents expected to be requested by v (i.e., contents currently possessed by other group members but not v). Finally, the third part considers the probability that other one-hop neighbors of v might possess the same content and deliver it upon request.

Similarly, the probability of u delivering a request to another group member r who is not its one-hop neighbor (i.e., $r \notin \mathbb{H}^u_1$) can be defined as,

$$\begin{aligned}\eta^u_{d,h_m} &= \sum_{r=1}^{|\mathbb{H}^u_m|}(1-e^{-\lambda^r_r t_\alpha}).\rho.\frac{|\zeta^u-\zeta^r|}{|\zeta_G-\zeta^r|}\\ &\quad\times\frac{|\zeta^u-\zeta^r|}{|\zeta^u-\zeta^r|+\sum\limits_{k=1}^{|\mathbb{H}^r_m|}|\{\zeta^u-\zeta^r\}\cap\zeta^k|},\\ &\quad[\text{for}\,m>1,\, r\notin\mathbb{H}^u_1,\,\text{and}\,k\notin\mathbb{H}^r_1],\end{aligned} \tag{5.20}$$

where, ρ represents the multi-hop delivery factor which measures the fraction of content requests that are served using multi-hop communication for this group (i.e., $\rho = \rho^m/\rho^t$, where ρ^m and ρ^t show the number of multi-hop and total deliveries, respectively). Whenever a requester receives a content, it notifies the administrator about the delivery who updates the value of ρ. Finally, η^u_d is calculated by adding η^u_{d,h_1} and η^u_{d,h_m} using (5.19) and (5.20).

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

$$\eta^u_{g,h_1} = (1 - e^{-\lambda^u_r t_\alpha}) . \sum_{v=1}^{|\mathbb{H}^u_1|} \frac{|\zeta^v - \zeta^u|}{|\zeta_G - \zeta^u|} \times \frac{|\zeta^v - \zeta^u|}{|\zeta^v - \zeta^u| + \sum_{\substack{k=1 \\ k \neq v}}^{|\mathbb{H}^u_1|} |\{\zeta^v - \zeta^u\} \cap \zeta^k|}. \tag{5.21}$$

Similar to Eq. (5.20), the probability of node u receiving a content from its multi-hop neighbor is depicted as,

$$\eta^u_{g,h_m} = (1 - e^{-\lambda^u_r t_\alpha})\, \rho \sum_{r=1}^{|\mathbb{H}^u_m|} \frac{|\zeta^r - \zeta^u|}{|\zeta_G - \zeta^u|} \times \frac{|\zeta^r - \zeta^u|}{|\zeta^r - \zeta^u| + \sum_{k=1}^{|\mathbb{H}^u_m|} |\{\zeta^r - \zeta^u\} \cap \zeta^k|}, \quad [\text{for}\, m > 1,\, r \notin \mathbb{H}^1_u,\, \text{and}\, k \notin \mathbb{H}^1_u]. \tag{5.22}$$

Finally, the value of η^u_g can be achieved by adding η^u_{g,h_1} and η^u_{g,h_m} using Eqs. (5.21) and (5.22). To determine the probability of node u being used as relay during t_α, we need to find the portion of multi-hop deliveries in the group and the probability of u being used in them. The probability of u being selected as relay is,

$$\eta^r_u = (1 - e^{-\lambda_{r,G} t_\alpha}) . \rho . \frac{U^u}{\sum_{v=1}^{|\mathbb{H}^u_1|} U^v}. \tag{5.23}$$

Here, $\lambda_{r,G}$ represents the request generation rate of all group members (i.e., $\lambda_{r,G} = \sum_{k=1}^{\mathbb{G}} \lambda^k_r$) and U^u represents u's relaying capability. Note that the relaying capability depends on the forwarding protocol being used. For a random selection, each node will have the same value. Here, we use the method in Eq. (4.1) in Chapter 4 for determining U^u. The administrator uses η^u_d, η^u_g and η^u_r in Eq (5.18) to determine η^u, which is then used in Eq. (5.17) to calculate the non-contention factor τ^z.

Once the appropriate node z is selected by Eq. (5.15), the administrator recalculates demand and supply for content α using demand generated from z as *zero*, and the additional supply provided by z using (5.11). The administrator checks Eq. (5.14) to determine if additional copies are required and continues the above procedure until (5.14) is false. After all appropriate nodes are selected, the administrator instructs the supplier nodes to copy α to the respective newly selected content holders. Algorithm 3 outlines the content distribution method executed by the group administrator.

Algorithm 3 Content distribution

procedure DISTRIBUTE CONTENT
  for all α in group content list **do**
    Calculate D_α using (5.1);
    Calculate S_α using (5.11);
    Initialize $\mathbb{N} = \emptyset$, where $\mathbb{N}$ is the set of nodes selected for distributing α
    while $D_\alpha - S_\alpha > 0$ and $|\mathbb{N}| < |\mathbb{G}_{i,\alpha'}|$ **do**
      Select node z using (5.15);
      Add z to $\mathbb{N}$, i.e., $\mathbb{N} = \{\mathbb{N}\} \cup \{z\}$;
      Recalculate D_α considering demand from node z as *zero*;
      Recalculate S_α considering z as a supplier;
    end while
    Send message to supplier y to copy α in all $z \in \mathbb{N}$;
  end for
end procedure

5.2 Simulation Environment and Results

5.2.1 Simulation Environment

To assess the performance of the proposed DSIP-CD scheme, extensive simulation has been performed. We use the same simulation environment as highlighted in Section 3.2.1 of Chapter 3 except the set of contents shared in the network. Since event-related content access was analyzed in DSIP-CD, we extended the set of contents to be shared in the network to add event-related contents. In this dataset, we considered 320 contents across nine categories in the network. Fifty of them were related to five events that were scheduled at different times (two were scheduled before the start of the simulation, two during the simulation while one after the simulation). The size and type (e.g., text, image, audio or video) of individual contents were assigned using the method discussed in Section 3.2.1. Initially, nodes were assigned with a random number of contents (mean size 27.52), considering their interest. Nodes generated content requests based on their interest following a Zipf distribution and the request generation time followed an exponential distribution, as in Chapter 3. Note that nodes were also influenced by external events randomly and generated requests for associated contents. The simulation was run for three days with 150 nodes.

5.2.2 Simulation Results

Similar to Chapters 3 and 4, the performance of the proposed DSIP-CD approach is analyzed using three metrics, namely, (i) hit rate, (ii) delivery success rate and (iii) average delivery latency. Additionally, we have also investigated control overhead, administrator lifetime, energy consumption by nodes, evolution of demand and supply for a content and the distribution of replicated contents to meet dynamic demand in DSIP-CD.

The proposed DSIP-CD approach is compared against the DSIP (Chapter 3) and E-DSIP (Chapter 4) approaches presented in the previous chapters. Although DSIP-CD and E-DSIP use the same underlying group formation, administrator selection and message forwarding technique, the latter does not employ any content distribution scheme; therefore, a comparison between them will reveal the benefit of the proposed content distribution scheme. The interval between consecutive distribution cycles was set to θ=30 mins in our simulation. DSIP-CD is also compared with NCL [38], which uses a set of central locations in a network that are selected based on their accessibility by other nodes for holding the replicated contents, as discussed in Section 2.2.5.1. Note that the term central location only reflects easy accessibility by other nodes and does not indicate a centralized replication method; rather multiple such locations are selected for placing the replicated contents. We also compared DSIP-CD against SPOON [33] which is a notable work in decentralized content sharing for *work-place* type scenarios. Although SPOON does not use any replication or caching method, it employs interest similarity and meeting frequency based community construction and interest similarity based forwarding strategy to improve delivery service, and a comparison with SPOON shows how our approach performs against a notable work which addresses a different aspect of decentralized sharing. The characteristics of a tourist spot were applied to SPOON for fair comparison. The t-test comparing DSIP-CD with other methods in terms of hit rate, success rate and delivery latency at various arrival and request rates yielded p-values, $p \leq 6.33 \times 10^{-6}$, $p \leq 5.89 \times 10^{-6}$ and $p \leq 7.25 \times 10^{-6}$ at 99% confidence level, respectively, and suggests that their performance differences are statistically significant. Each simulation was run 20 times and the averaged results are presented below.

5.2.2.1 Impact on Overall Performance

The arrival rate of the nodes was varied from 10-30 nodes/hr to understand its impact on hit and delivery success rates, and the outcomes are presented in Figs. 5.3 and 5.4. Both rates initially start increasing along with an increasing node arrival rate across all approaches since a higher node arrival rate implies that more nodes were available inside the area for sharing contents. However, when the arrival rate exceeds a certain limit (i.e. 25 nodes/hr in our current setting), both rates decrease slightly across all approaches, because more nodes subsequently generate more requests, causing congestion. DSIP-CD always achieves the highest hit and delivery success rates, achieving a hit rate of 73.31% and delivery success rate of 70.45% for an arrival rate of 25 nodes/hr. In comparison, E-DSIP achieves a 66.76% hit rate and 63.95% success rate, DSIP achieves a 60.09% hit rate and 55.31% success rate, while SPOON and NCL achieve much lower hit (SPOON 46.12% and NCL 48.39%) and success rates (SPOON 44.32% and NCL 37.33%) for the same arrival rate. The proposed DSIP-CD approach proactively distributes contents based on the estimated future demand which essentially increases availability of contents near the positions from where future requests are expected. When requests are ultimately generated for those contents, they can be served from those positions (i.e., by nodes with replicated content), and hence the DSIP-CD approach achieves a 7%∼34% improvement for hit and success rates over other approaches. One interesting observation from Figs. 5.3 and 5.4 is the perfor-

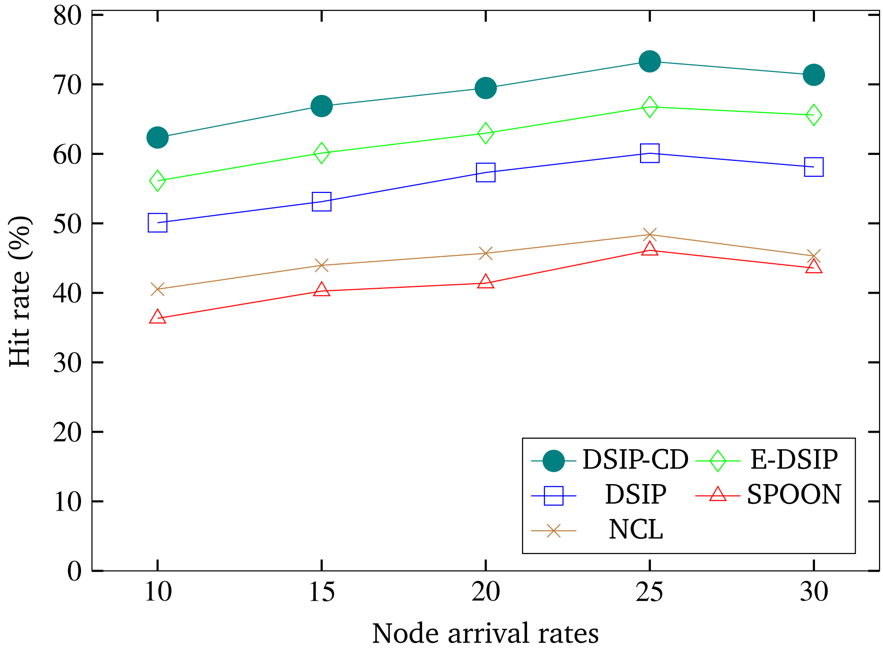

Figure 5.3: Hit rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr

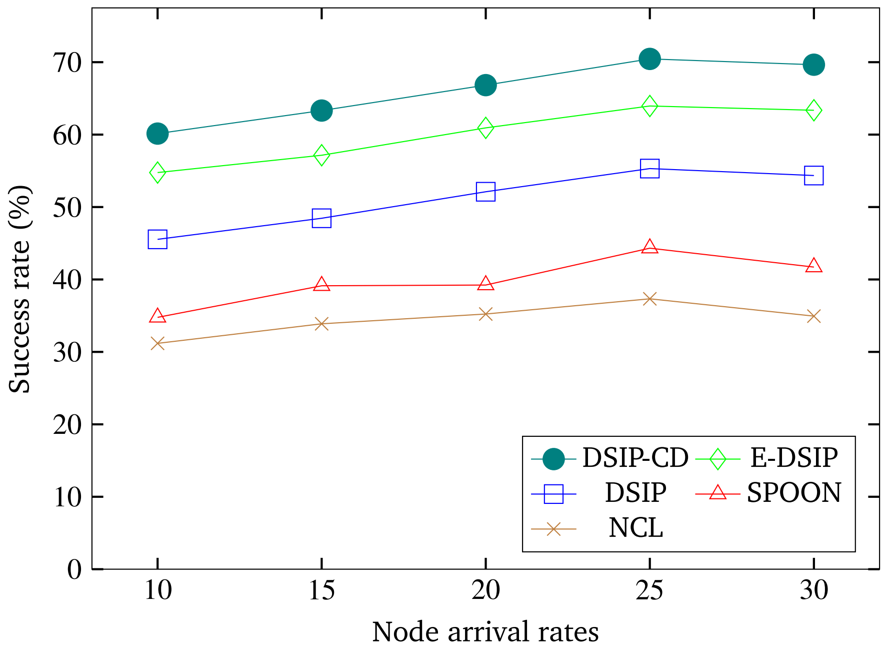

Figure 5.4: Success rate for different mean node arrival rates (number of arriving nodes/hr) keeping the mean per node content request rate fixed at 5 requests/hr

mance of the NCL and SPOON approaches. Although NCL achieves a similar hit rate to SPOON, its delivery success rate is much lower. The reason for this is that in NCL contents are replicated in easily accessible locations (i.e., central locations) which essentially help in locating the contents (hence higher hit rate), but at the same time, it also makes the central locations more congested and unable to deliver, as many nodes simultaneously try to communicate with them for uploading and downloading contents.

The impact of varying content request rates is presented in Figs. 5.5 and 5.6. The figures show that both hit and success rates start to increase at the beginning with an increasing request rate, however, after exceeding a particular request rate (5 requests/hr in our current setting) both metrics start to decline slightly. Initially, with an increasing request rate, nodes located within the same neighborhood generate more requests for the same content, which has already been requested and obtained by a nearby node. However, both rates start declining as more requests generate more congestion in the network. The proposed DSIP-CD approach achieves a 71.43% hit rate and 68.94% success rate at the request rate of 6 requests/hr. In contrast, other approaches achieve

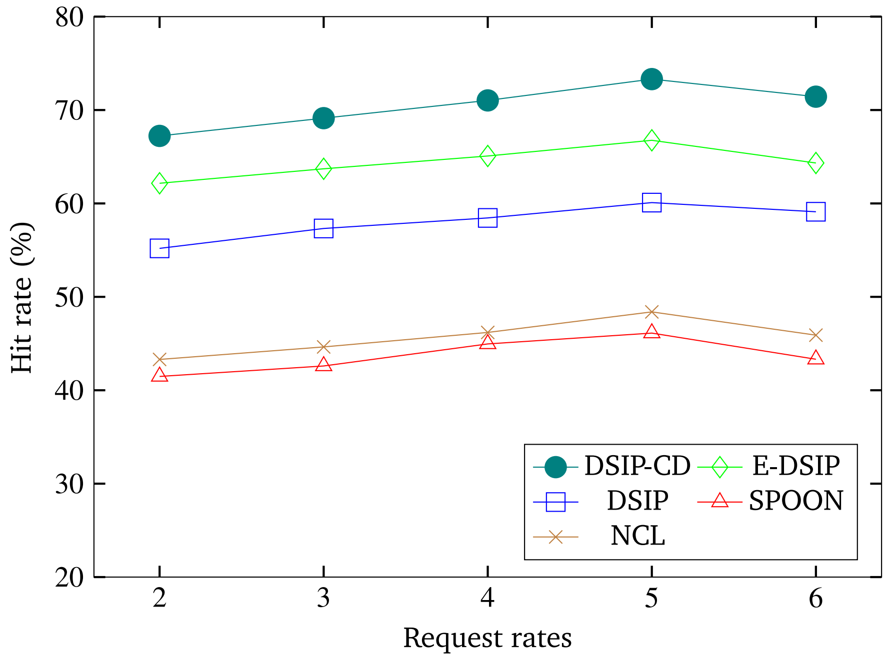

Figure 5.5: Hit rate for different mean request rates (number of requests/hr) keeping the mean node arrival rate fixed at 25 nodes/hr

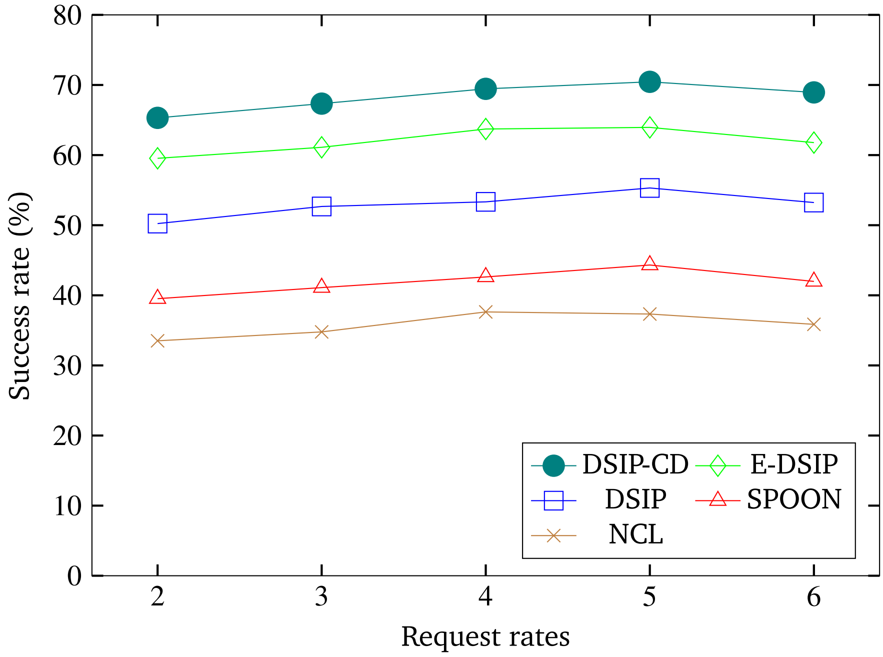

Figure 5.6: Success rate for different mean request rates (number of requests/hr) keeping the mean node arrival rate fixed at 25 nodes/hr

much lower hit (43%∼64%) and success rates (35%∼62%) for the same request rate. In the literature, the reported hit and success rates achieved by decentralized content sharing in *work-place* type scenarios range from 75%∼80% [33, 52–54]. From that perspective, the DSIP-CD approach obtains comparable results in tourist spots, where movement patterns are unpredictable and social ties are mostly unavailable.

The average latency for successful content delivery is shown in Fig. 5.7. For the sake of fair comparison among approaches, only for this particular experiment, we considered 10,000 *identical* requests (i.e., same node requested same contents) across all approaches. In Fig. 5.7, the latency values for successful content deliveries are sorted in ascending order and averaged over the particular number of deliveries for convenience of presentation. All the approaches can complete up to 3000 successful deliveries within a short time (e.g., 0.01 mins (DSIP-CD), 0.03 mins (E-DSIP), 0.03 mins (DSIP), 0.03 mins (SPOON) and 0.02 mins (NCL)). Since NCL only uses central locations for content delivery, it experiences higher delay after 3×10^3 requests and can only deliver up to 3.8×10^3 requests for the given settings with an average delay of 45.17 mins beyond 3×10^3 requests. SPOON can deliver up to 4.5×10^3 requests while incurring higher delays (61.6 mins) beyond 3.8×10^3 requests. The DSIP approach can deliver up to 5.5×10^3 requests while experiencing comparatively high delay after 4.5×10^3 requests (DSIP-CD 6.23 mins, E-DSIP 20.32 mins and DSIP 57.32 mins). In contrast, the DSIP-CD approach can deliver up to 6.3×10^3 requests with significantly lower delay (DSIP-CD 11.92 mins vs. E-DSIP 55.38 mins). The reason for such lower delay for DSIP-CD is the content distribution policy which proactively replicates the contents near the potential requesters, which essentially enables requesters to obtain requested contents within a short time. For the given settings, E-DSIP can only deliver up to 6.3×10^3 requests, while DSIP-CD can successfully deliver up to 7×10^3 requests. Note that beyond 6.3×10^3 requests, the delivery latency of DSIP-CD grows (40.48 mins) as sometimes content holders might be multiple hops away from requesters and forwarding requests as well as obtaining matching contents require a longer time, hence a longer delay is observed. Although the NCL approach uses multiple locations for placing the replicated contents, it does not consider the contention

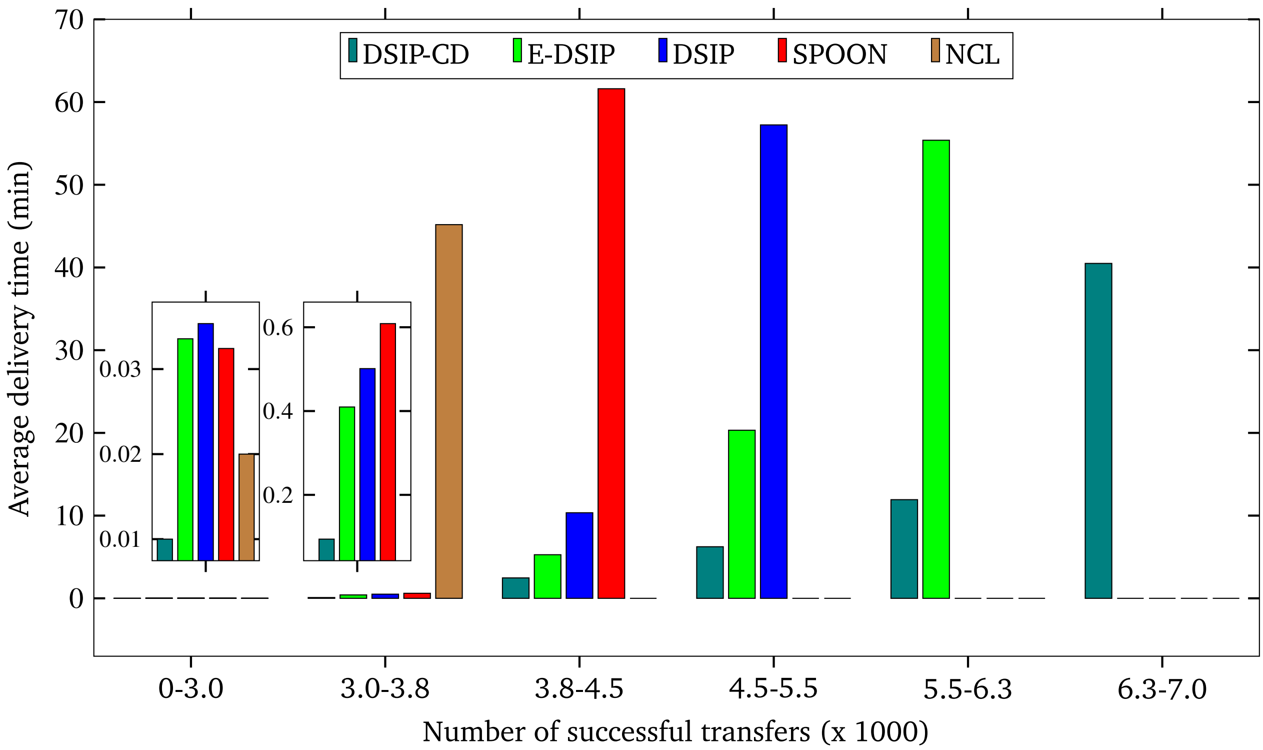

Figure 5.7: Average delivery latency for successful deliveries for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

factor for placing the replicated contents and hence suffers because of congestion in those central locations which results in longer delays and lower delivery success rates. Although the delay is longer for a small portion of contents (i.e., 10%), the DSIP-CD approach is capable of delivering a higher number of contents (7%∼32% higher) even if the delay is longer for some of them, where other approaches fail altogether.

Administrators perform a number of group management and content sharing tasks, as highlighted in Section 3.1.1. In addition, the administrator also handles content distribution tasks in DSIP-CD that drain their energy. Therefore, administrator lifetime is an important issue since frequent administrator handover is not desirable. Figures 5.8 and 5.9 show the average administrator lifetime across all approaches for varying node arrival and content request rates. Both figures show that DSIP-CD achieves a much higher administrator lifetime than NCL and SPOON, while achieving a slightly lower lifetime than E-DSIP and DSIP. Since DSIP-CD, E-DSIP and DSIP use the same administrator selection technique based on the stay probability of nodes, they result in a higher administrator lifetime. However, in DSIP-CD, an administrator periodically calculates demand, supply and distribution, and sometimes replicates a content to meet future demand, and hence achieves a slightly lower lifetime than E-DSIP and DSIP. In

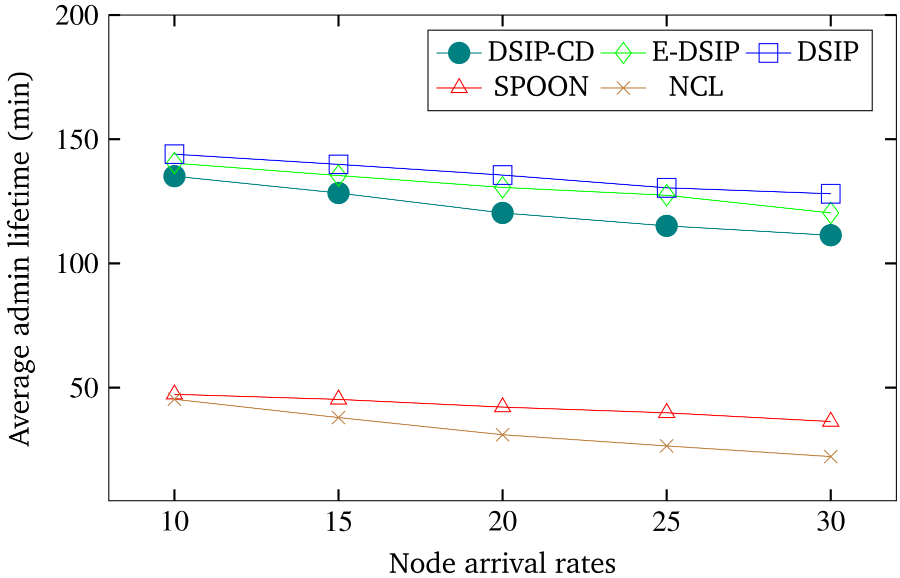

Figure 5.8: Average lifetime of administrator nodes for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

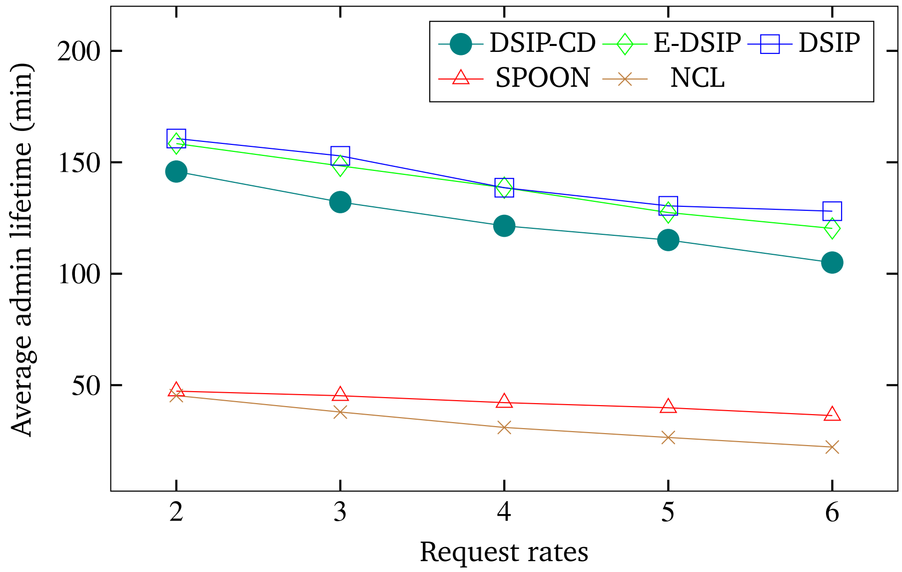

Figure 5.9: Average lifetime of administrator nodes for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

contrast, in NCL, stay probability was not considered at all and administrators were mostly busy in receiving/delivering contents from other nodes, which in turn resulted in such lower lifetime. SPOON also does not consider the stay probability of a node for selecting an administrator and hence achieves such a low administrator lifetime.

Figures 5.10 and 5.11 show the percentage of message overhead which was calculated as the ratio of the size of control messages to the size of the successfully delivered contents. DSIP-CD generated lower message overhead than E-DSIP and DSIP, and a slightly higher overhead than SPOON and NCL. DSIP-CD uses some group-related information (e.g., group content list and stay probability of members), a neighborhood table and some content and event specific information (e.g., content popularity and event scale) for content distribution. Therefore, the message overhead is slightly higher than SPOON and NCL at a lower arrival (10 nodes/hr in Fig. 5.10) and request rate (2 requests/hr in Fig. 5.11), the differences being $< 1\%$ in both cases. However, as the arrival and request rate increase, the number of successfully delivered contents increases, which essentially starts reducing the overall message overhead, and the proposed approach produces nearly similar overhead as produced by other approaches.

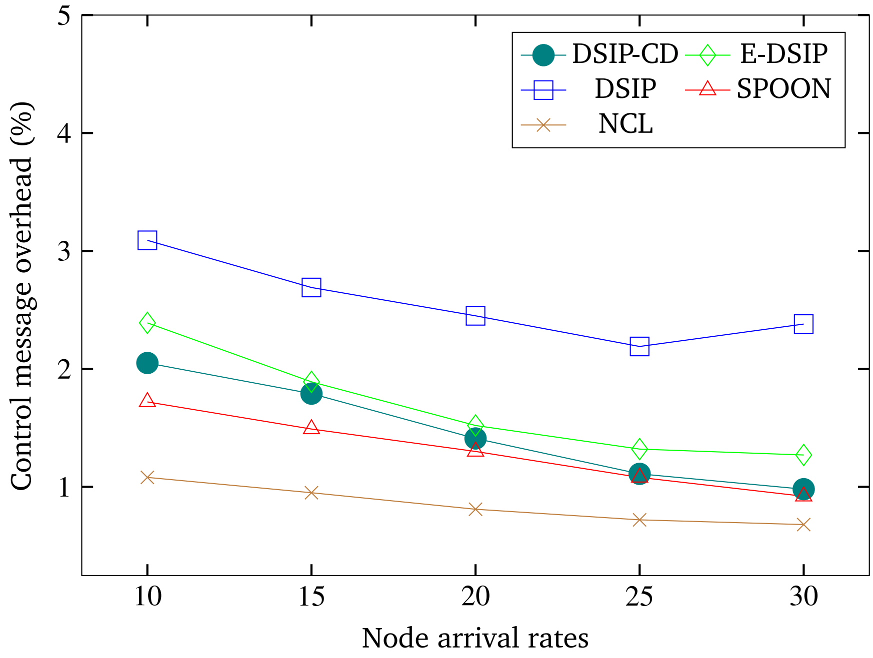

Figure 5.10: Control message overhead for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

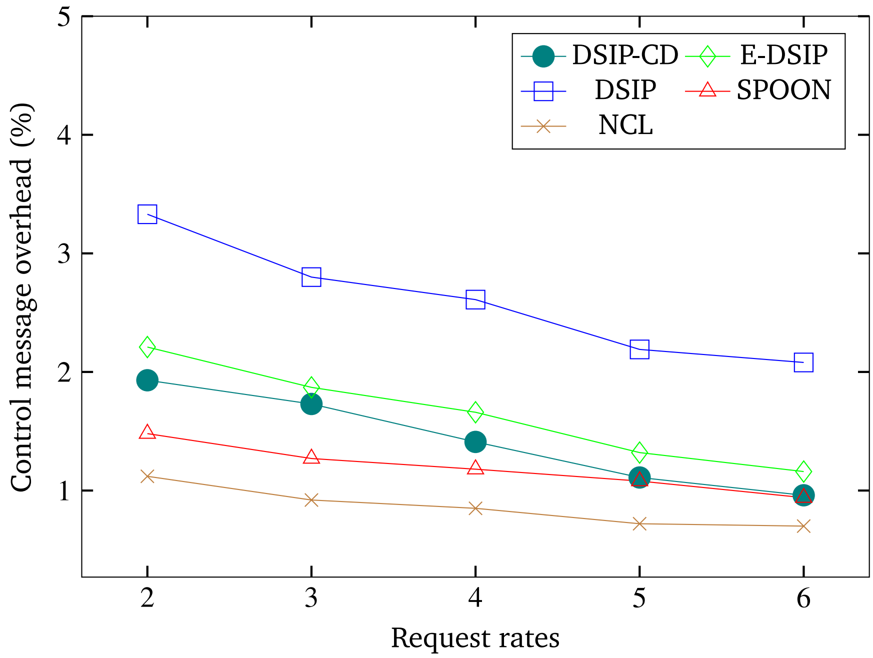

Figure 5.11: Control message overhead for different mean content request rates keeping the mean node arrival rate fixed at 25 nodes/hr

5.2.2.2 Impact on Event-Related Contents

The DSIP-CD approach considers the impact of influencing events on content request while calculating future demand. The impact of such policies on hit and success rates for different node arrival rates is presented in Figs. 5.12 and 5.13. Both figures suggest that DSIP-CD achieved the highest percentage of hit and delivery success rates for event based content requests across all cases. The improvement achieved by DSIP-CD ranges from 12%∼40%. The reason for this lies in its proactive distribution of content taking event induced demand into account. In this regard, these figures justify the consideration of the impact of an event on content access for capturing the demand.

The impact of different content request rates on hit and delivery success rates for event based content requests is presented in Figs. 5.14 and 5.15. Both figures show that DSIP-CD produced the highest percentage of hit and delivery success rates across all cases. DSIP-CD achieved a 75.33% hit rate and 73.15% success rate for a request rate of 5 requests/hr for the given setting for event based content requests. In contrast, the hit and success rates achieved by other approaches range from 45-62% and 36-

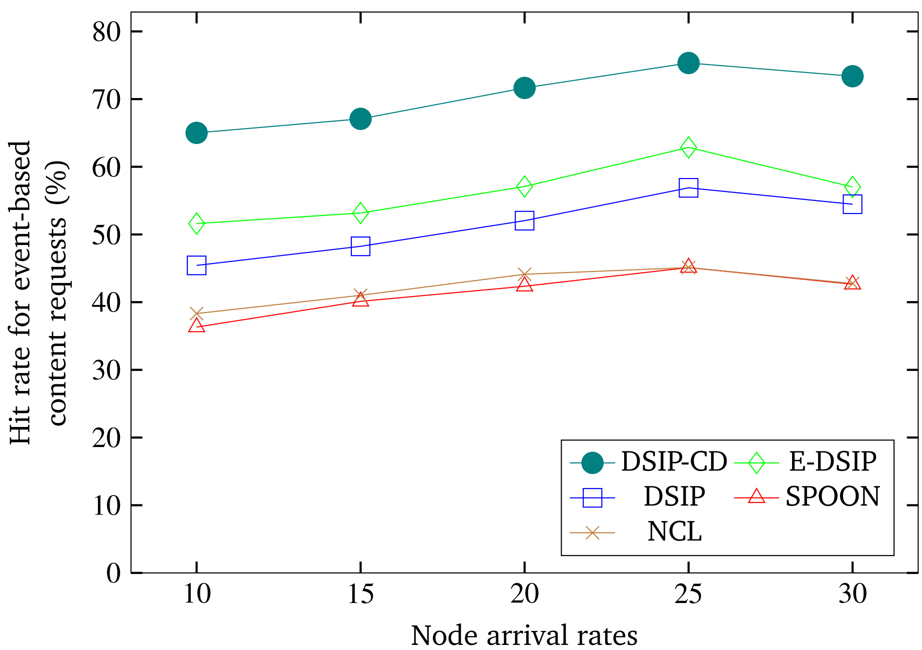

Figure 5.12: Hit rate for event based content requests for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

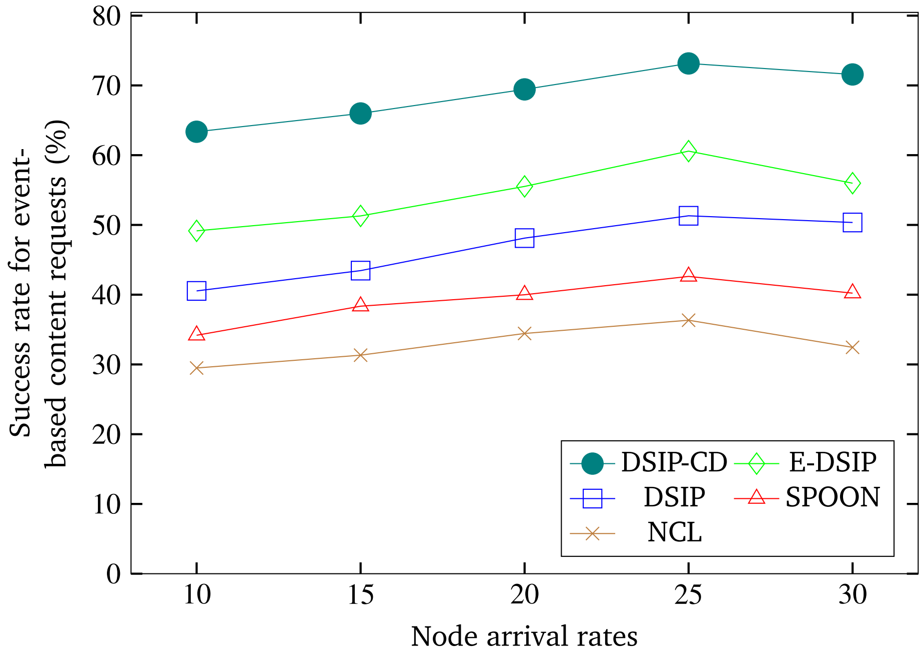

Figure 5.13: Success rate for event based content requests for different mean node arrival rates keeping the mean per node content request rate fixed at 5 requests/hr

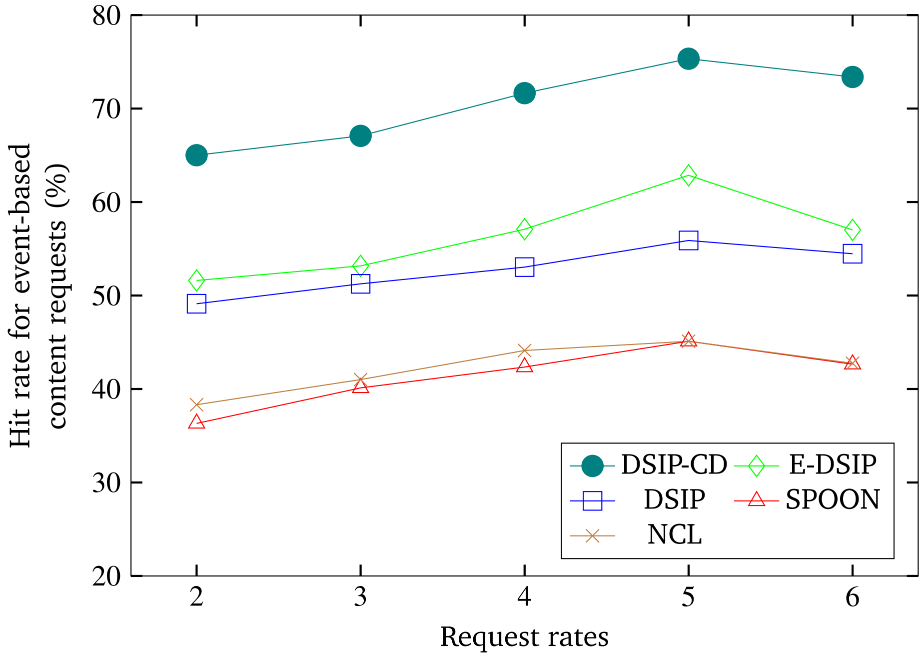

Figure 5.14: Hit rate for event based content requests for different mean request rates keeping the mean node arrival rate fixed at 25 nodes/hr

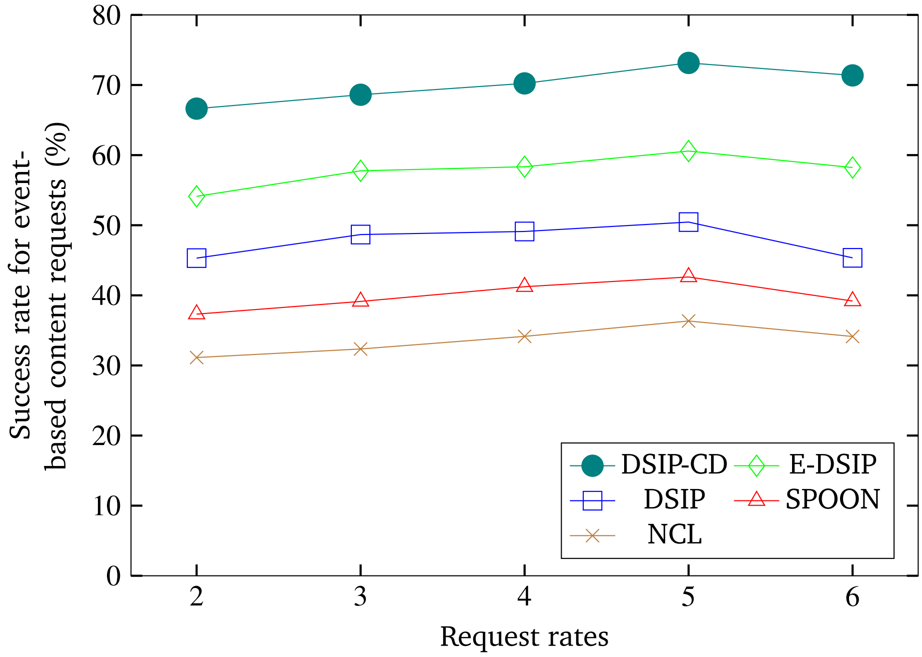

Figure 5.15: Success rate for event based content requests for different mean request rates keeping mean node arrival rate fixed at 25 nodes/hr

60%, respectively, for the same request rate. The superiority of DSIP-CD over other approaches is achieved because of the adoption of the content distribution scheme which considers the impact of influencing external events on content access for calculating demand and accordingly proactively distributes contents.

5.2.2.3 Impact of Distribution

As discussed before, the administrators periodically calculate dynamic demand and supply for a content to determine if additional distribution is needed. Figure 5.16 shows varying demand and supply calculated for a particular content by an administrator inside the 'Food and shopping' POI. The figure shows that at the beginning (during cycle 1), supply for this content was lower than the expected demand, hence proactive distribution (two copies) was performed to meet future demand. During cycles 2 and 3, the expected demand was lower than the available supply and no distribution was required. Afterwards, the amount of expected demand surpassed the available supply in cycles 4 and 5 as new nodes joined the group and were expected to generate demand while few existing suppliers left the group, which reduced available supply in

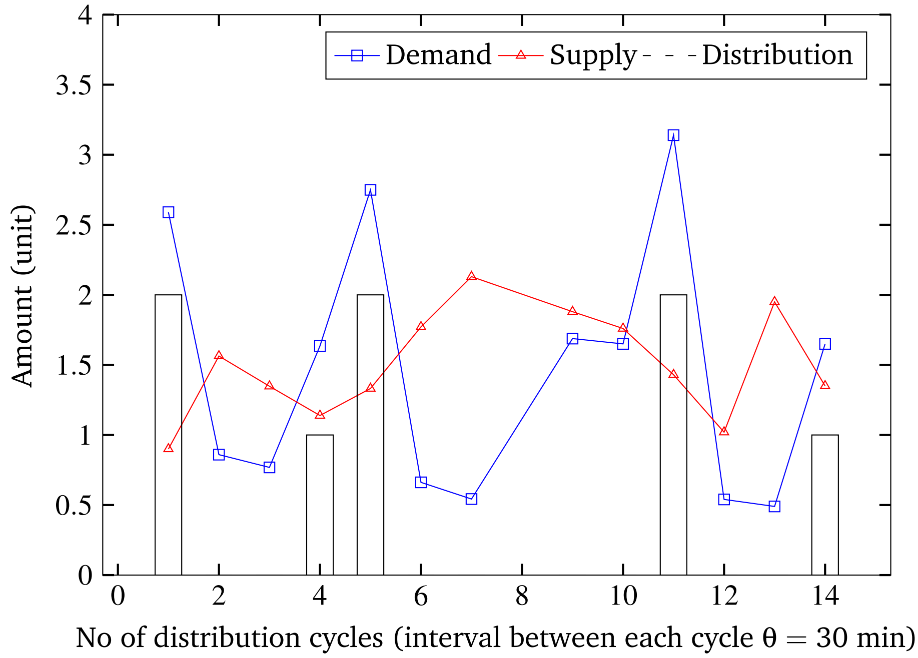

Figure 5.16: Demand and supply for a content in a group inside the 'Food and shopping' POI for the mean node arrival rate of 25 nodes/hr and the mean request rate of 5 requests/hr

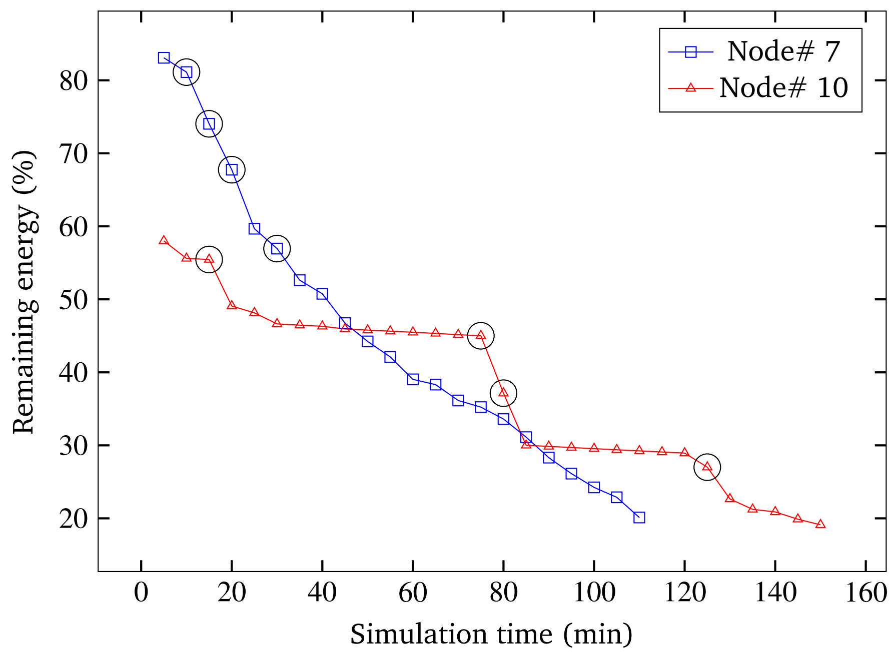

Figure 5.17: Remaining energy of two nodes before their next charging cycles (a circle represents an occurrence of replication at that node)

cycle 4. Therefore, distribution was done during those cycles. As more content holders joined the group, the amount of available supply increased during cycles 6 and 7 and the demand remained lower than the supply up to cycle 10. Finally, expected demand surpassed available supply during cycles 11 and 14, and additional distribution was needed during those cycles.

In DSIP-CD, the remaining energy of a node was assessed for the selection of a node for content distribution purposes. Figure 5.17 demonstrates the impact of such a selection process by showing that although Node# 7 and Node# 10 were selected four times for distribution purposes, they did not run out of battery before their next charging cycles.

Figure 5.18 shows a snapshot of the locations of demanding, supplier and distributor nodes for a popular content in simulation. Since it was a popular content, more demanding nodes (marked by red squares) were present in the network. The number of distributor nodes (marked by blue plus signs) was also higher to meet expected high demand. The figure also shows that the distributor nodes were positioned near the demanding nodes. In some cases, a few distributor nodes were enough to meet the

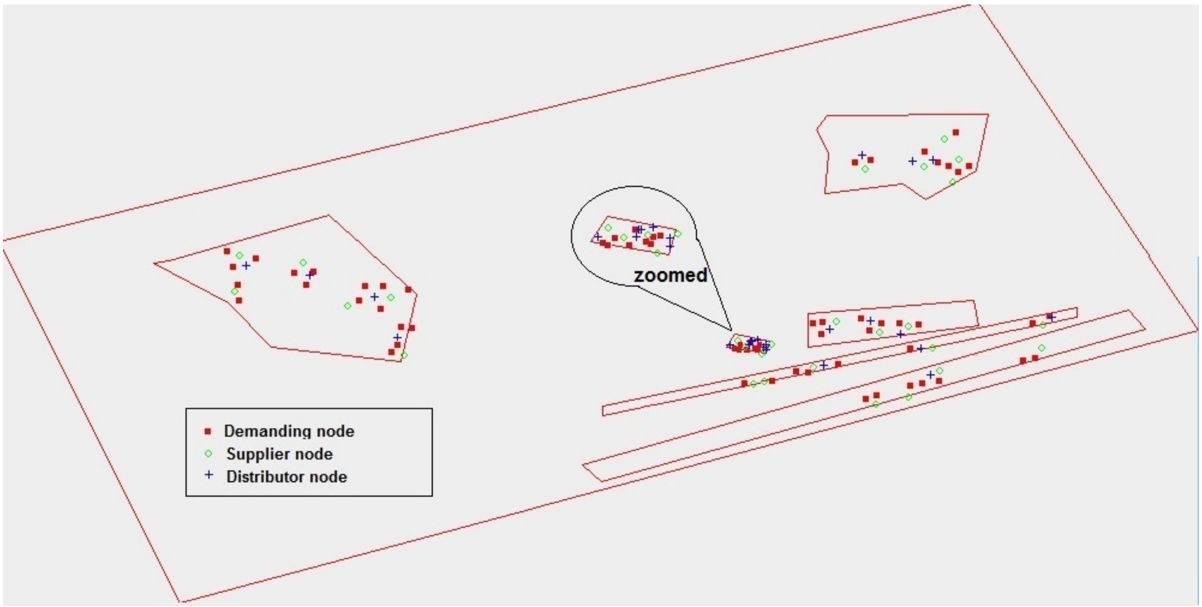

Figure 5.18: A snapshot of the positions of the demanding, supplier and distributor nodes for a popular content in different POIs. A supplier node is one which already holds the content and a distributor is one where a replicated copy is placed

expected demand ('Camping' POI on the left side of Fig. 5.18), while others required more distributor nodes ('Bowling' POI in the middle of Fig. 5.18). Nodes generated varying demand and supply for a content in different POIs and hence the number of distributor nodes also varied in different regions.

5.3 Conclusion

This chapter has presented a content distribution scheme for decentralized content sharing in irregular meeting places employing the concept of economic theory of supply and demand in a marketplace. Whenever the expected demand surpasses the available supply, the proposed DSIP-CD approach proactively replicates content at strategic locations to improve delivery service. DSIP-CD estimates future demand considering user interest, content popularity and the impact of influencing events on content demand. Available supply for contents is calculated considering the content holder's coverage, stay probability and available energy. Finally, the locations for placing additional copies are selected based on the joint optimization of demand coverage and medium access contention. We have performed extensive simulation to assess the performance of DSIP-CD. Results obtained from simulation indicate that DSIP-CD is capable of handling a significantly higher number of requests while attaining better delivery performance than competing DCS methods applied in tourist hotspots.

We presented the DSIP method in Chapter 3 to facilitate decentralized content sharing in irregular meeting places. To improve content delivery service and encourage node participation in the DSIP method, the E-DSIP method was presented in Chapter 4. Finally, this chapter has presented the DSIP-CD method which uses a content distribution scheme to further improve hit rate, delivery success rate and latency. However, further ways can be investigated to improve the the proposed DSIP-CD approach, which are presented in the next chapter.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

Decentralized content sharing approaches are gaining popularity as they provide a convenient way to generate and share contents on the fly in any place, at any time. In this approach, users within the close vicinity form an ad-hoc network and share contents using peer-to-peer communication through Wi-Fi/Bluetooth connection. It is a free-of-cost infrastructure-less platform which reduces reliance on Internet connection and thereby, avoid adding to Internet traffic. The shared contents are useful for spreading information, staying connected, marketing products, delivering entertainment and keeping up to date. However, implementing DCS is very challenging, mainly because of the lack of persistent connections among participating nodes and a mechanism to handle dynamic content demand and supply in an efficient and robust way. The problems are more prominent in irregular meeting places, such as tourist spots, camping sites or gatherings during festivals, where the short stay duration does not allow enough learning period, and people mostly meet strangers.

To address these research challenges and advance the DCS approach, in this thesis, we have made a number of significant contributions that are presented as follows:

- *A basic framework for DCS*: A novel strategy was presented to facilitate decentralized content sharing in irregular meeting places. A unique interest extraction method was developed which incorporates information from sources, such as personal profiles, recommendations from other users, and facilities available in a tourist spot. This interest information is further used during the group formation process which considers mutual interest as well as the probability of a group to provide relevant contents within tolerable delay. To achieve this, a joint optimization problem was formulated for a node to select the most appropriate group to join. In addition, an intelligent administrator selection policy was devised which selects appropriate administrators through a competitive process who are able to provide service for a longer amount of time, resulting in higher administrator lifetime and lower handover rate. Extensive simulation was performed using a popular tourist spot in Victoria, Australia to assess the performance of the proposed approach. Simulation results confirm that the developed model achieves 59.13% delivery success rate, which is 12-20% higher compared to existing relevant approaches for DCS when applied to the tourist spot type scenario. The latency attained for successful delivery is mostly between 3 seconds to 11 minutes, which is acceptable in a delay tolerant network like this. However, in this approach, the Spray-and-Wait [49] technique is used as a message forwarding protocol, which uses redundant copies of a message. This creates a pathway for the introduction of a utility based message forwarding technique to enhance content delivery services further.

- *Utility based message forwarding*: A new utility based message forwarding technique was constructed which uses information that is readily available or can be collected on-the-fly and suitable for tourist spot like scenarios. This technique maximizes the probability of successful delivery by considering opportunistic encounter probability and connectivity in a group while selecting a forwarder node. Additionally, resource consumption is minimized by using only a single copy of a message for delivery. Simulation results show the improvement achieved with respect to delivery success rates range from 6-8% because of the adoption of this forwarding method. Up to now, our framework lacked a mechanism to encourage node participation and promote honest nodes that are also crucial for efficient and effective content delivery success and timely delivery.

- *Incentive and trust management scheme*: To encourage node participation, a novel incentive scheme was introduced which employs node behavior monitoring and provides non-monetary incentive in the form of priority processing. The incentive score of a node increases if it provides service to others and in return a node with a higher incentive score obtains higher priority for obtaining service from others. Furthermore, misbehaving nodes are identified through a lightweight trust management method which identifies nodes that try to commit forgery by submitting false delivery claims. Such misbehaving nodes are excluded from the candidate pool of potential administrators to reduce the overall security risk of the framework. Simulation results demonstrate that nodes with higher incentive scores (0.9-1.0) obtain 25% higher content delivery success rate compared to nodes with lower incentive scores (0.0-0.3), and hence this works as a motivation for users to participate in the sharing process. The demand and supply of content is dynamic in nature. To improve delivery success and time, it is of paramount importance to embed a content distribution mechanism on the bedrock of the proposed framework considering dynamic demand and supply and medium access contention.

- *Content distribution by approximating dynamic demand and supply*: An innovative content distribution strategy was formulated employing the economic modeling of dynamic demand and supply, and medium access contention. A content is only replicated whenever its estimated demand surpasses its available supply. New strategies for demand and supply calculation were also adopted which dynamically calculate both metrics considering user interest, content popularity, the impact of influencing external events on content access and the capability of a content holder for delivering a requested content. The distribution policy intelligently selects the location for placing a replica considering the joint optimization of demand coverage and medium access contention. We have performed extensive simulation to evaluate the performance of the proposed distribution scheme and results confirm that the developed model is capable of handling a higher number of requests while attaining high delivery success rates and low latency, compared with existing relevant approaches for DCS, when applied in our given scenario. Our approach successfully delivered 70.45% of the requested contents, which is nearly comparable to DCS approaches proposed for work-place type scenarios with regular movement patterns and existing social relationships [33, 52–54].

Finally, incorporation of all the above modules provides a suitable platform for decentralized content sharing among visitors in irregular meeting places. Since this is the first attempt at employing decentralized content sharing in irregular meeting places, we believe that it has made a significant contribution in the field of decentralized sharing and will encourage the relevant research community to investigate further in this particular scenario. The research conducted in this thesis will be beneficial for the community to avail content sharing service in visiting hotspots which lack internet connection or have poor coverage. In addition, methods developed in this thesis can also be incorporated to other research areas, such as Internet-of-things, device-to-device communication, health monitoring applications, emergency networks, vehicular ad-hoc networks and opportunistic mobile sensing and computing.

6.2 Future Works

This thesis has presented a complete and novel framework for facilitating decentralized content sharing in irregular meeting places. However, since this work is the first of its kind and similar to other existing approaches, the proposed framework can be further extended in a number of directions that are presented as follows:

- *Group mobility:* The group formation and message forwarding strategies presented in this thesis primarily focus on the individual user for identifying their interests and movement patterns. However, sometimes a large group of tourists (i.e., a herd) such as families, co-workers or friends can also visit together and may engage in group activities. Although an individual might not be interested in a particular activity, because they are traveling with others, the user might end up performing those activities. Such herding behavior can be further utilized in interest calculation. In this case, some new issues need to be investigated that can exploit the user's affinity towards such herding activity, the probability of engaging in such activity during the current visit and the overall interest score of all group members for a particular activity. It will be interesting to further analyze how these newly adopted issues will impact overall interest score calculations. In addition, this group mobility can be incorporated in message forwarding techniques where delivering a message to any member of a herd will eventually increase the probability of successful delivery. In this regard, selection of the forwarder node needs to additionally check whether the destination and the potential forwarder are from the same herd or if the forwarder has a higher probability of meeting anyone from the same herd as the destination node.

- *Ambassador nodes:* In some cases, relevant contents might not be available among the group members and hence they may need to be fetched from other content sharing groups. In this regard, similar to the work presented in [33], ambassador nodes can be used who can carry the request from group $G1$ to the administrator of group $G2$. In response, the administrator of $G2$ can send the matching content to any member of group $G1$ through ambassador for $G1$. However, selection of appropriate ambassador node in irregular meeting places is difficult, as unlike [33], nodes usually do not travel between groups frequently to allow a learning period. Again, administrators are also unaware about the availability of contents in other groups. To address this, administrators can share the group content list with other administrators which will also increase message overhead. In addition, the ambassador nodes can be selected considering their stay probability within the group, their movement direction and the probability of moving towards a particular group. However, if multiple groups have matching contents, determining the appropriate group and their corresponding ambassador to obtain the content within its lifetime is a very challenging task, and requires further research.

- *Multi-channel communication:* NS3 simulation in this thesis employs single channel operation where all the groups use the same channel to communicate. Implementing non-overlapping channels for nearby groups is expected to increase the performance of our proposed scheme (i.e., channels allocated in adjacent groups will be different). However, channel assignment in IEEE 802.11 is an NP-hard problem and very complex. One way to address this would be through group administrators who can mutually communicate to assign a channel for the group members. In this case, an administrator can calculate the utility of a channel use in a group in terms of demand considering the number of group members, their request frequency and the size of requested contents. Based on this, the group with the highest demand can be allocated a better channel in terms of channel quality and available bandwidth. Finally, it requires further investigation to ensure that nearby groups operate at non-overlapping channels to avoid mutual interference considering node mobility.

- *Punishment strategies:* The incentive scheme presented in this thesis provides rewards for successful content delivery to keep the mechanism lightweight, reduce message overhead and discourage cheating behavior. This feature is efficient for encouraging the good behavior of nodes so that they are more willing to participate in the sharing process. Although misbehaving nodes are also identified through the trust management scheme and prevented from becoming administrators, no direct punishment strategy has been adopted. In this case, a stricter punishment strategy can be incorporated where the misbehaving nodes with a lower trust score will be given some punishment in the form of service disruption or complete exclusion. However, it is difficult to determine the appropriate strategy as a stricter policy might discourage participation at all. In this case, the administrator can exploit the median and standard deviation of the trust score of current group members to punish a node with a lower trust score or propagate the list of such misbehaving nodes to other groups. The initial punishment for nodes can be given in the form of service disruption, where misbehaving nodes will be required to provide service to other nodes to increase their trust score before obtaining any service from other nodes. If a misbehaving node continues to make false claim, stricter policy such as exclusion from the system can be adopted. The administrator can broadcast IMEI information of such a node to inform other groups about this. Overall, further investigation is required to implement which of the above techniques is suitable for the proposed DCS framework.

- *Privacy and security implications:* One concern of decentralized content sharing is the security implication, though in this thesis like other studies, we assume devices to be securely configured. In addition, a lightweight behavior monitoring based trust management method is also presented in this thesis. This method is useful in identifying nodes with ill intention and stops them from becoming an administrator and spreading malicious contents. However, it is still possible for a node to behave well and then spread malicious contents in response to a content request. One way to handle this is that the received content can be sandboxed first and checked for any malicious action. Contents are stored in local memory only if they are found as safe. Access to the content sharing application and its users also needs to be restricted within the shared directory to provide better security. Note, for confidentiality, a Key based encryption technique is ineffectual in a tourist spot like scenario as there is no central authority to manage and distribute digital certificates and keys. In addition, the unfamiliarity among nodes also makes the mutual authentication process ineffective. In this regard, the administrator can distribute authentication keys only to highly trusted group members to facilitate secure communication among trustworthy nodes. However, identifying a lightweight and suitable encryption and authentication technique would be very challenging and needs further research.

- *Buffer replacement:* The proposed dynamic content distribution scheme selects appropriate node for placing a replicated content considering that it has enough available buffer space to hold the content. However, in some cases it might be more beneficial to select a node that is better positioned to hold the content, but has limited buffer space to do so. In this case, a buffer replacement policy is required to identify contents currently held by a node which can be replaced with the new one. One way to achieve this is to identify contents with low demand and replace them with the new ones. However, this may create a conflicting situation where the node itself might be interested in consuming a content α while other group members may generate very low demand for α. In this regard, an optimization technique is needed for buffer replacement that will maximize benefits for individual nodes, as well as for the whole group. The administrator will calculate the benefit of replacing a particular content in the buffer only if it is substantially beneficial to all the group members. Altogether, identifying an appropriate buffer replacement policy needs further investigation.

References

[1] “Smartphone users worldwide,” https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, accessed 26 July 2017.

[2] “Tablet users worldwide,” https://www.statista.com/statistics/377977/tablet-users-worldwide-forecast/, accessed 26 July 2017.

[3] “The mobile story,” https://www.iabaustralia.com.au/uploads/uploads/2015-11/1448287200_7a87cd54d624da241efa7087c362d82e.pdf, accessed 26 July 2017.

[4] “Mobile marketing statistics,” https://www.wiredseo.com/mobile-marketing-statistics-2017/, accessed 26 July 2017.

[5] “Youtube statistics,” http://expandedramblings.com/index.php/youtube-statistics/#.U4WETpSSzUc, accessed 26 July 2017.

[6] X. Liang, K. Zhang, X. Shen, and X. Lin, “Security and privacy in mobile social networks: challenges and solutions,” *IEEE Wireless Communications*, vol. 21, no. 1, pp. 33–41, 2014.

[7] “Google play,” https://play.google.com/store/apps?hl=en, accessed 26 July 2017.

[8] “Apple app store,” https://www.apple.com/au/ios/app-store/, accessed 26 July 2017.

[9] “Microsoft app store,” https://www.microsoft.com/en-AU/store/apps/windows?rtc=1, accessed 26 July 2017.

[10] N. Vastardis and K. Yang, “Mobile social networks: Architectures, social properties, and key research challenges,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1355–1371, 2013.

[11] L. Militano, A. Iera, and F. Scarcello, "A fair cooperative content-sharing service," *Elsevier Computer Networks*, vol. 57, no. 9, pp. 1955 – 1973.

[12] M. Taghizadeh, K. Micinski, S. Biswas, C. Ofria, and E. Torng, "Distributed cooperative caching in social wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1037–1053, 2013.

[13] P. Jassal, K. Yadav, A. Kumar, V. Naik, V. Narwal, and A. Singh, "Unity: Collaborative downloading content using co-located socially connected peers," in *IEEE PerCom 2013*, pp. 66–71.

[14] "Mobile blackspot program discussion paper," https://www.communications.gov.au/have-your-say/mobile-black-spot-programme-discussion-paper, accessed 26 July 2017.

[15] "Mobile blackspot program," https://www.communications.gov.au/what-we-do/phone/mobile-services-and-coverage/mobile-black-spot-program, accessed 26 July 2017.

[16] "Cisco vni forecast," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, accessed 26 July 2017.

[17] "Bandwidth explosion," http://arstechnica.com/business/2012/05/bandwidth-explosion-as-internet-use-soars-can-bottlenecks-be-averted/, accessed 26 July 2017.

[18] "Increase in mobile data usage," https://crowdsupport.telstra.com.au/t5/Data-Usage/Increase-in-Mobile-Data-usage-with-Smartphones-and-4G-handsets/td-p/127496, accessed 26 July 2017.

[19] S. Trifunovic, S. T. Kouyoumdjieva, B. Distl, L. Pajevic, G. Karlsson, and B. Plattner, "A decade of research in opportunistic networks: Challenges, relevance, and future directions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 168–173, 2017.

[20] H. D. M. G. M. Vetterli, "Efficient route discovery in mobile ad hoc networks using encounter ages," in *ACM MobiHoc 2003*, pp. 1–3.

[21] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *Springer Service assurance with partial and intermittent resources*, pp. 239–254, 2004.

[22] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine *et al.*, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *IEEE INFOCOM 2006*, pp. 1–11.

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

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

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

[26] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *ACM MobiHoc 2007*, pp. 32–40.

[27] A. Mtibaa, M. May, C. Diot, and M. Ammar, "Peoplerank: Social opportunistic forwarding," in *IEEE INFOCOM 2010*, pp. 1–5.

[28] P. Hui and J. Crowcroft, "How small labels create big improvements," in *IEEE PerCom 2007*, pp. 65–70.

[29] "The world's prettiest places without internet access," http://www.huffingtonpost.com.au/entry/pretty-places-without-internet-access_n_3961920, accessed 29 July 2017.

[30] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "A first look at cellular network performance during crowded events," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 17–28, 2013.

[31] J. Erman and K. K. Ramakrishnan, "Understanding the super-sized traffic of the super bowl," in *ACM IMC 2013*, pp. 353–360.

[32] 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,” *Springer Wireless Personal Communications*, vol. 78, no. 1, pp. 755–783, 2014.

[33] K. Chen, S. Member, H. Shen, S. Member, and H. Zhang, “Leveraging social networks for p2p content-based file sharing in disconnected manets,” *IEEE Transaction on Mobile Computing*, vol. 13, no. 2, pp. 235–249, 2014.

[34] 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,” in *ACM SAC 2011*, pp. 425–432.

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

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

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

[38] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, “Cooperative caching for efficient data access in disruption tolerant networks,” *IEEE Transaction on Mobile Computing*, vol. 13, no. 3, pp. 611–625, 2014.

[39] T. Small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in *ACM SIGCOMM 2005*, pp. 260–267.

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

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

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

[43] K. Chen and H. Shen, "Maximizing p2p file access availability in mobile ad hoc networks though replication for efficient file sharing," *IEEE Transaction on Computers*, vol. 64, no. 4, pp. 1029–1042, 2015.

[44] T. Wang, P. Hui, S. Kulkarni, and P. Cuff, "Cooperative caching based on file popularity ranking in delay tolerant networks," *ACM ExtremeCom 2012*, pp. 11–16.

[45] X. Zhuo, Q. Li, G. Cao, Y. Dai, B. Szymanski, and T. La Porta, "Social-based cooperative caching in dtns: A contact duration aware approach," in *IEEE MASS 2011*, pp. 92–101.

[46] J. Liu, P. Dolan, and E. R. Pedersen, "Personalized news recommendation based on click behavior," in *IEEE IUI 2010*, pp. 31–40.

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

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

[49] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *ACM SIGCOMM 2005*, pp. 252–259.

[50] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Carry me if you can: A utility based forwarding scheme for content sharing in tourist destinations," in *IEEE APCC 2016*, pp. 261–267.

[51] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Decentralized content sharing among tourists in visiting hotspots," *Elsevier Journal of Network and Computer Applications*, vol. 79, pp. 25 – 40, 2017.

[52] F. Xia, L. Liu, J. Li, A. M. Ahmed, L. T. Yang, and J. Ma, "Beeinfo: Interest-based forwarding using artificial bee colony for socially aware networking," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1188–1200, 2015.

[53] P. Yuan, H. Ma, and H. Fu, "Hotspot-entropy based data forwarding in opportunistic social networks," *Elsevier Pervasive and Mobile Computing*, vol. 16, pp. 136–154, 2015.

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

[55] S. Kaisar, J. Kamruzzaman, G. Karmakar, and I. Gondal, "Dynamic content distribution for decentralized sharing in tourist spots using demand and supply," in *IEEE IWCMC 2017*, pp. 2121–2126.

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

[57] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/-subscribe communication in delay tolerant networks," in *ACM MSWiM 2007*, pp. 225–234.

[58] H. Jafarpour, S. Mehrotra, and N. Venkatasubramanian, "A fast and robust content-based publish/subscribe architecture," in *IEEE NCA 2008*, pp. 52–59.

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

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

[61] C. Lindemann and O. P. Waldhorst, "A distributed search service for peer-to-peer file sharing in mobile applications," in *IEE Peer-to-Peer Computing 2002*, pp. 73–80.

[62] D. Zhang, Z. Wang, B. Guo, X. Zhou, and V. Raychoudhury, "A dynamic community creation mechanism in opportunistic mobile social networks," in *IEEE SOCIALCOM 2011*, pp. 509–514.

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

[64] C. A. da Costa, D. Zaupa, J. L. V. Barbosa, R. da Rosa Righi, J. de Camargo, N. Daroit, and A. C. Yamin, "A spontaneous social network based on mobile devices," *Springer Social Network Analysis and Mining*, vol. 4, no. 1, p. 158, 2014.

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

[66] 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 Transaction on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 734–743, 2014.

[67] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1682–1695, 2014.

[68] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *ACM MobiArch 2007*, pp. 1–8.

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

[70] N. Yu and Q. Han, "Context-aware communities and their impact on information influence in mobile social networks," in *IEEE PERCOM 2012*, pp. 131–136.

[71] Z. Tian and N. Samaan, "A novel approach for modeling and managing spon-

taneous social communities over manets," *Springer Social Network Analysis and Mining*, vol. 3, no. 3, pp. 685–700, 2013.

[72] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1682–1695, 2014.

[73] D. Li, Q. Lv, X. Xie, L. Shang, H. Xia, T. Lu, and N. Gu, "Interest-based real-time content recommendation in online social communities," *Elsevier Knowledge-Based Systems*, vol. 28, pp. 1–12, 2012.

[74] D. Palsetia, M. M. A. Patwary, K. Zhang, K. Lee, C. Moran, Y. Xie, D. Honbo, A. Agrawal, W.-k. Liao, and A. Choudhary, "User-interest based community extraction in social networks," in *ACM SIGKDD*, 2012, pp. 1–5.

[75] H. Wirtz, J. Rüth, T. Zimmermann, and K. Wehrle, "Interest-based cloud-facilitated opportunistic networking," in *ACM MobiCom 2013*, pp. 63–68.

[76] K. Thilakarathna, A. C. Viana, A. Seneviratne, and H. Petander, "Mobile social networking through friend-to-friend opportunistic content dissemination," in *ACM MobiHoc 2013*, pp. 263–266.

[77] A. Iamnitchi, M. Ripeanu, E. Santos-Neto, and I. Foster, "The small world of file sharing," *IEEE Transactions on Parallel and Distributed systems*, vol. 22, no. 7, pp. 1120–1134, 2011.

[78] R.-I. Ciobanu, R.-C. Marin, C. Dobre, and V. Cristea, "Interest-awareness in data dissemination for opportunistic networks," *Elsevier Ad Hoc Networks*, vol. 25, pp. 330–345, 2015.

[79] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM 2003*, pp. 27–34.

[80] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[81] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-200006, Duke University*, 2000.

[82] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Springer Wireless networks*, vol. 8, no. 2/3, pp. 153–167, 2002.

[83] N. B. Chang and M. Liu, "Controlled flooding search in a large network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 436–449, 2007.

[84] M. Pitkanen, T. Karkkainen, J. Greifenberg, and J. Ott, "Searching for content in mobile dtns," in *IEEE PerCom 2009*, pp. 1–10.

[85] M. Vojnovi and A. Proutiere, "Hop limited flooding over dynamic networks," in *IEEE INFOCOM 2011*, pp. 685–693.

[86] Y. Lu, W. Wang, L. Chen, Z. Zhang, and A. Huang, "Distance-based energy-efficient opportunistic broadcast forwarding in mobile delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5512–5524, 2016.

[87] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *IEEE PerCom 2007*, pp. 79–85.

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

[89] J. Niu, D. Wang, and M. Atiquzzaman, "Copy limited flooding over opportunistic networks," *Elsevier Journal of Network and Computer Applications*, vol. 58, pp. 94–107, 2015.

[90] J. Wu, Y. Zhu, L. Liu, B. Yu, and J. Pan, "Energy-efficient routing in multi-community dtn with social selfishness considerations," in *IEEE GLOBECOM 2016*, pp. 1–7.

[91] X. Li, J. Teng, B. Zhang, A. C. Champion, and D. Xuan, "Turfcast: A service for controlling information dissemination in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 250–262, 2014.

[92] B. B. Bista and D. B. Rawat, "Ea-prophet: An energy aware prophet-based routing protocol for delay tolerant networks," in *IEEE AINA 2017*, pp. 670–677.

[93] B. B. Bista and D. B. Rawat, "Enhancement of prophet routing in delay tolerant networks from an energy prospective," in *IEEE TENCON 2016*, pp. 1579–1582.

[94] H. Jiaxin, X. Chunxiu, and W. Yuewei, "Resource-efficient routing protocol based on historical encounter time interval in dtn," in *IEEE ICCC 2016*, pp. 2026–2031.

[95] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, 2008.

[96] F. Li and J. Wu, "Localcom: a community-based epidemic forwarding scheme in disruption-tolerant networks," in *IEEE SECON 2009*, pp. 1–9.

[97] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *IEEE INFOCOM 2011*, pp. 3119–3127.

[98] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *ACM SIGCOMM 2005*, pp. 244–251.

[99] N. Eagle and A. S. Pentland, "Reality mining: sensing complex social systems," *Springer Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.

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

[101] M. Musolesi and C. Mascolo, "Car: context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2009.

[102] S.-H. Kim, Y. Jeong, and S.-J. Han, "Use of contact duration for message forwarding in intermittently connected mobile networks," *Elsevier Computer Networks*, vol. 64, pp. 38–54, 2014.

[103] N. N. Gondaliya, D. Kathiriya, and M. Shah, "Contact frequency and contact duration based relay selection approach inside the local community in social delay tolerant network," in *Springer SIST 2016*, pp. 609–617.

[104] W. Gao, G. Cao, T. La Porta, and J. Han, "On exploiting transient social contact patterns for data forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 151–165, 2013.

[105] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "Sgbr: A routing protocol for delay tolerant networks using social grouping," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2472–2481, 2013.

[106] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.

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

[108] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Elsevier Computer networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[109] K. Jahanbakhsh, G. C. Shoja, and V. King, "Social-greedy: a socially-based greedy routing algorithm for delay tolerant networks," in *ACM MobiOpp 2010*, pp. 159–162.

[110] J. Wu and Y. Wang, "Hypercube-based multipath social feature routing in human contact networks," *IEEE Transactions on Computers*, vol. 63, no. 2, pp. 383–396, 2014.

[111] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless routingin pocket switched networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 252–261, 2015.

[112] O. Brock, B. Burns, and B. Levine, "Mv routing and capacity building in disruption tolerant networks," in *IEEE INFOCOM 2005*, pp. 398–408.

[113] H.-Y. Huang, P.-E. Luo, M. Li, D. Li, X. Li, W. Shu, and M.-Y. Wu, "Performance evaluation of suvnet with real-time traffic data," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3381–3396, 2007.

[114] X. Li, W. Shu, M. Li, H. Huang, and M.-Y. Wu, "Dtn routing in vehicular sensor networks," in *IEEE GLOBECOM 2008*, pp. 1–5.

[115] J. Zhu, J. Cao, M. Liu, Y. Zheng, H. Gong, and G. Chen, "A mobility prediction-based adaptive data gathering protocol for delay tolerant mobile sensor network," in *IEEE GLOBECOM 2008*, pp. 1–5.

[116] A. B. Link, D. Schmitz, and K. Wehrle, "Geodtn: Geographic routing in disruption tolerant networks," in *IEEE GLOBECOM 2011*, pp. 1–5.

[117] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *ACM SIGMOBILE 2008*, pp. 33–40.

[118] K. h. Chen , H. Shen, and L. Yan, "Dsearching: Using floating mobility information for distributed node searching in dtns," *IEEE Transactions on Mobile Computing*, vol. 15, no. 1, pp. 121–136, 2016.

[119] S. Lu, Y. Liu, Y. Liu, and M. Kumar, "Loop: A location based routing scheme for opportunistic networks," in *IEEE MASS 2012*, pp. 118–126.

[120] H. Shen and L. Zhao, "Alert: An anonymous location-based efficient routing protocol in manets," *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1079–1093, 2013.

[121] Q. He, D. Wu, and P. Khosla, "A secure incentive architecture for ad hoc networks," *Wiley Wireless Communications and Mobile Computing*, vol. 6, no. 3, pp. 333–346, 2006.

[122] M. Y. S. Uddin, B. Godfrey, and T. Abdelzaher, "Relics: In-network realization of incentives to combat selfishness in dtns," in *IEEE ICNP 2010*, pp. 203–212.

[123] N. Li and S. K. Das, "Radon: reputation-assisted data forwarding in opportunistic networks," in *ACM MobiOpp 2010*, pp. 8–14.

[124] G. Bigwood and T. Henderson, "Ironman: Using social networks to add incentives and reputation to opportunistic networks," in *IEEE SOCIALCOM 2011*, pp. 65–72.

[125] L. Wei, H. Zhu, Z. Cao, and X. S. Shen, "Mobiid: A user-centric and social-aware reputation based incentive scheme for delay/disruption tolerant networks," in *Springer ADHOC-NOW 2011*, pp. 177–190.

[126] A. Mei and J. Stefa, "Give2get: Forwarding in social mobile wireless networks of selfish individuals," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, pp. 569–582, 2012.

[127] R.-I. Ciobanu, C. Dobre, M. Dascalu, S. Trausan-Matu, and V. Cristea, "Collaborative selfish node detection with an incentive mechanism for opportunistic networks," in *IEEE IM 2013*, pp. 1161–1166.

[128] F. Xia, L. Liu, J. Li, J. Ma, and A. V. Vasilakos, "Socially aware networking: A survey," *IEEE Systems Journal*, vol. 9, no. 3, pp. 904–921, 2015.

[129] H. Chen, W. Lou, Z. Wang, and Q. Wang, "A secure credit-based incentive mechanism for message forwarding in noncooperative dtns," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6377–6388, 2016.

[130] Y. Wang, M.-C. Chuah, and Y. Chen, "Incentive based data sharing in delay tolerant mobile networks," *IEEE wireless communications*, vol. 13, no. 1, pp. 370–381, 2014.

[131] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, and W. Zhang, "Cais: a copy adjustable incentive scheme in community-based socially aware networking," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3406–3419, 2017.

[132] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang, "Incentive-aware routing in dtns," in *IEEE ICNP 2008*, pp. 238–247.

[133] Q. Li, W. Gao, S. Zhu, and G. Cao, "A routing protocol for socially selfish delay tolerant networks," *Elsevier Ad Hoc Networks*, vol. 10, no. 8, pp. 1619–1632, 2012.

[134] K. Wei, M. Dong, K. Ota, and K. Xu, "Camf: Context-aware message forwarding in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2178–2187, 2015.

[135] L. Yu and P. Liu, "A service-based selfish routing for mobile social networks," *SAGE International Journal of Distributed Sensor Networks*, vol. 11, no. 6, pp. 1–8, 2015.

[136] R.-I. Springer Ciobanu, R.-C. Marin, C. Dobre, and V. Cristea, "Onside-self: A selfish node detection and incentive mechanism for opportunistic dissemination," in *Internet of Things (IoT) in 5G Mobile Technologies*, 2016, pp. 317–332.

[137] L. Liu, Q. Yang, X. Kong, H. B. Liaqat, A. M. Ahmed, N. Deonauth, and F. Xia, "Com-bis: a community-based barter incentive scheme in socially aware networking," *SAGE International Journal of Distributed Sensor Networks*, vol. 11, no. 8, pp. 1–14, 2015.

[138] Y. Xie and Y. Zhang, "A secure, service priority-based incentive scheme for delay tolerant networks," *Wiley Security and Communication Networks*, vol. 9, no. 1, pp. 5–18, 2016.

[139] X.-Y. Li, Y. Wu, P. Xu, G. Chen, and M. Li, "Hidden information and actions in multi-hop wireless ad hoc networks," in *ACM MobiHoc 2008*, pp. 283–292.

[140] L. Wei, Z. Cao, and H. Zhu, "Mobigame: A user-centric reputation based incentive protocol for delay/disruption tolerant networks," in *IEEE GLOBECOM 2011*, pp. 1–5.

[141] L. Buttyán, L. Dóra, M. Félegyházi, and I. Vajda, "Barter trade improves message delivery in opportunistic networks," *ElsevierAd Hoc Networks*, vol. 8, no. 1, pp. 1–14, 2010.

[142] T. Ning, Z. Yang, X. Xie, and H. Wu, "Incentive-aware data dissemination in delay-tolerant mobile networks," in *IEEE SECON 2011*, pp. 539–547.

[143] T. Ning , Z. Yang, H. Wu, and Z. Han, "Self-interest-driven incentives for ad dissemination in autonomous mobile social networks," in *IEEE INFOCOM 2013*, pp. 2310–2318.

[144] F. Xia, B. Jedari, L. T. Yang, J. Ma, and R. Huang, "A signaling game for uncertain data delivery in selfish mobile social networks," *IEEE Transactions on Computational Social Systems*, vol. 3, no. 2, pp. 100–112, 2016.

[145] B. Jedari, L. Liu, T. Qiu, A. Rahim, and F. Xia, "A game-theoretic incentive scheme for social-aware routing in selfish mobile social networks," *Elsevier Future Generation Computer Systems*, vol. 70, pp. 178–190, 2017.

[146] Z. Xu, Y. Jin, W. Shu, X. Liu, and J. Luo, "Sred: A secure reputation-based dynamic window scheme for disruption-tolerant networks," in *IEEE MILCOM 2009*, pp. 1–7.

[147] N. Li and S. K. Das, "A trust-based framework for data forwarding in opportunistic networks," *Elsevier Ad Hoc Networks*, vol. 11, no. 4, pp. 1497–1509, 2013.

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

[149] A. M. Ahmed, X. Kong, L. Liu, F. Xia, S. Abolfazli, Z. Sanaei, and A. Tolba, "Bodmas: Bio-inspired selfishness detection and mitigation in data management for ad-hoc social networks," *Elsevier Ad Hoc Networks*, vol. 55, pp. 119–131, 2017.

[150] 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.

[151] R.-I. Ciobanu, C. Dobre, M. Dascălu, Ş. Trăuşan-Matu, and V. Cristea, "Sense: A collaborative selfish node detection and incentive mechanism for opportunistic

networks," *Elsevier Journal of Network and Computer Applications*, vol. 41, pp. 240–249, 2014.

[152] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.

[153] K. Zhang, X. Liang, R. Lu, K. Yang, and X. S. Shen, "Exploiting mobile social behaviors for sybil detection," in *IEEE INFOCOM 2015*, pp. 271–279.

[154] A. B. Waluyo, D. Taniar, W. Rahayu, and B. Srinivasan, "Trustworthy data delivery in mobile p2p network," *Elsevier Journal of Computer and System Sciences*, vol. 86, pp. 33–48, 2017.

[155] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 1, pp. 65–67, 2008.

[156] Y. Ma, R. Kibria, and A. Jamalipour, "Cache-based content delivery in opportunistic mobile adhoc networks," in *IEEE GLOBECOM 2008*, pp. 1–5.

[157] Y. Ma and A. Jamalipour, "A cooperative cache-based content delivery framework for intermittently connected mobile ad hoc networks," *IEEE Wireless Communications*, vol. 9, no. 1, pp. 366–373, 2010.

[158] S. Ioannidis, L. Massoulie, and A. Chaintreau, "Distributed caching over heterogeneous mobile networks," in *ACM SIGMETRICS 2010*, vol. 38, no. 1, pp. 311–322.

[159] G. Moualla, P. A. Frangoudis, Y. Hadjadj-Aoul, and S. Ait-Chellouche, "A bloom-filter-based socially aware scheme for content replication in mobile ad hoc networks," in *IEEE CCNC 2016*, pp. 359–365.

[160] Z. Wang, L. Sun, M. Zhang, H. Pang, E. Tian, and W. Zhu, "Propagation- and mobility-aware d2d social content replication," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1107–1120, 2017.

[161] J. Zhao, X. Zhuo, Q. Li, W. Gao, and G. Cao, "Contact duration aware data replication in dtns with licensed and unlicensed spectrum," *IEEE Transactions on Mobile Computing*, vol. 15, no. 4, pp. 803–816, 2016.

[162] R. Cabaniss and S. Madria, "Content distribution in delay-tolerant networks using social context," in *IEEE WMNC 2014*, pp. 1–8.

[163] I.-W. Ting and Y.-K. Chang, "Improved group-based cooperative caching scheme for mobile ad hoc networks," *Elsevier Parallel and Distributed Computing*, vol. 73, no. 5, pp. 595–607, 2013.

[164] S. Shakkottai and R. Johari, "Demand-aware content distribution on the internet," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 476–489, 2010.

[165] A. Alasaad, S. Gopalakrishnan, and V. C. Leung, "A hybrid approach for cost-effective media streaming based on prediction of demand in community networks," *SpringerTelecommunication Systems*, vol. 59, no. 3, pp. 329–343, 2015.

[166] X. Meng, Y. Wang, and Y. Gong, "Perspective of space and time based replica population organizing strategy in unstructured peer-to-peer networks," *Elsevier Journal of Network and Computer Applications*, vol. 49, pp. 1–14, 2015.

[167] J. Reich and A. Chaintreau, "The age of impatience: optimal replication schemes for opportunistic networks," in *ACM CoNEXT 2009*, pp. 85–96.

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

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

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

[171] "Tourism research australia," http://www.tra.gov.au/research/regional.html, accessed 20 March 2017.

[172] "Australian bureau of statistics," http://www.abs.gov.au/AUSSTATS/abs@.nsf/ViewContent?readform&view=ProductsbyTopic&Action=Expand&Num=3.5.15, accessed 20 March 2017.

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

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

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

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

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

[178] Y. Lai, Z. Chen, W. Wu, and T. Ma, "Multiple-resolution content sharing in mobile opportunistic networks," *Wiley Wireless Communications and Mobile Computing*, vol. 15, no. 16, pp. 1991–2003, 2015.

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

[180] M. Kruger and M. Saayman, "The determinants of visitor length of stay at the kruger national park," *AOSIS koedoe*, vol. 56, no. 2, pp. 1–11, 2014.

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

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

[183] A. Abhari and M. Soraya, "Workload generation for youtube," *Springer Multimedia Tools and Applications*, vol. 46, no. 1, pp. 91–118, 2010.

[184] B. Balcik and B. M. Beamon, "Facility location in humanitarian relief," *International Journal of Logistics Research and Applications*, vol. 11, no. 2, pp. 101–121, 2008.

[185] P. Murali, F. Ordóñez, and M. M. Dessouky, "Facility location under demand uncertainty: Response to a large-scale bio-terror attack," *Elsevier Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 78–87, 2012.

[186] H. Schütze and C. Silverstein, "Projections for efficient document clustering," in *ACM SIGIR 1997*, vol. 31, no. SI, pp. 74–81.

[187] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Taylor & Francis Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[188] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *ACM WSDM 2010*, pp. 291–300.

[189] H. Becker, D. Iter, M. Naaman, and L. Gravano, "Identifying content for planned events across social media sites," in *ACM WSDM 2012*, pp. 533–542.

[190] D. Getz, "Event tourism: Definition, evolution, and research," *Elsevier Tourism management*, vol. 29, no. 3, pp. 403–428, 2008.

[191] L. K. Jago, "Special events and tourism behaviour: a conceptualisation and an empirical analysis from a values perspective," *Ph.D. dissertation, Department of Hospitality, Tourism and Marketing, Victoria University*, 1997.