

# A Recommendation Based Efficient Secure discover-predict-deliver Content Sharing Scheme

Asiya Zareen<sup>1</sup>, Dr. Mohammed Abdul Waheed<sup>2</sup>, Shridevi Soma<sup>3</sup>

<sup>1</sup>*P.G.Student, Department of Computer Science & Engineering,  
VTU RO PG Centre, Gulbarga, Karnataka, India.*

<sup>2</sup>*Associate Professor, Department of Computer Science & Engineering,  
VTU RO PG Centre, Gulbarga, Karnataka, India.*

<sup>3</sup>*Associate Professor, Department of Computer Science & Engineering,  
PDACE, Gulbarga, Karnataka, India.*

**Abstract**— The increase in the number of smart phone users has lead to the increase in the peer-to-peer adhoc content sharing. Traditional data delivery schemes are not suitable for such networks due to intermittent connectivity between smart phones. Thus new content sharing mechanisms should be proposed. To share the contents in such a scenario, researchers have proposed store-carry-forward as an efficient content sharing scheme where a node stores a message and carries it for certain duration until a communication opportunity arises and then delivers it to the destination. Previous works in this field focused on whether two nodes would encounter each other and the place and time of encounter and did not consider the activities of malicious peers. This paper proposes discover-predict-deliver as efficient content sharing scheme that enable peers to share contents and also presents distributed algorithms that enable a peer to reason about trustworthiness of other peers based on past interactions and recommendations. Interactions and recommendations among peers are evaluated based on importance, recentness and peer satisfaction parameters.

**Keywords**— peer-to-peer, store-carry-forward, discover-predict-deliver, interaction, recommendation.

## I. INTRODUCTION

The number of smart phone users has rapidly increased in the past few years [1]. Users can create various types of contents easily using user friendly interfaces available in the smart phones. However content sharing among smart phones is tedious as it requires several actions such as uploading to centralized servers, searching and downloading contents. One easy way is to rely on adhoc method of peer-to-peer content sharing. Unfortunately with the current adhoc routing protocols, contents are not delivered if a network partition exists between the peers when contents are shared. Therefore Delay Tolerant Network (DTN) routing protocols achieve better performance than traditional adhoc routing protocols. These protocols do not require a centralised server. Hence the contents are stored on smart phones itself.

This paper focuses mainly on efficiency of content discovery and its delivery to the destination. Here we propose recommendation based discover-predict-deliver (DPD) as efficient content sharing scheme for smart phone based DTN's. DPD assumes that smart phones can connect when they are in close proximity that is where the smart phone users stay for a longer duration. Previous studies

have shown that Smartphone users stay indoors for a longer duration where GPS cannot be accessed [15]. It employs a hidden Markov model and Viterbi algorithm [16] to predict the future location of individuals. Therefore the contents are shared where the connectivity between smart phones is possible.

In the previous work, the open nature of peer-to-peer systems exposes them to malicious activity. Building trust relationships among peers can mitigate attacks of malicious peers. A distributed algorithm is proposed in this work that enables a peer to reason about trustworthiness of other peers based on past interactions and recommendations.

## II. RELATED WORK

A delay tolerant network (DTN) is a sparse mobile network where a contemporary source-destination path may not exist between a pair of nodes and messages are forwarded in a store-carry-forward routing paradigm [6].

Vahdat et al [2] proposed Epidemic routing as basic DTN routing protocol in which a node forward a message to every encountered node that does not have a copy of the message. The solution demonstrates the best performance terms of delivery rate and latency but wastes a lot of bandwidth.

An alternative solution was resource based [3], [4], where systems employ "data mules" as message ferries that directly deliver message to the destination. Next, opportunity-based routing protocols use history of encounters to deliver a message to the destination [5], [6], [7]. Prediction based schemes [8], [9], [10], [11], [12] use sophisticated utility functions to determine whether to forward a message to the node. Most of the existing prediction based schemes are based on whether two nodes would encounter each other in the future and not on the place and time of encounter. Yuan et al [12] accurately predicted encounter opportunities by using time of encounters.

Pitkanen et al [14] proposed state-of-the-art content sharing scheme in DTN's. They mainly focused on limiting search query propagation and proposed many query processing methods.

Chang et al [13] proposed a method for searching for a node or an object in a large network and limiting search query propagation. Here a class of controlled flooding search strategies where query/search packets are broadcast

and propagated in the network until a preset TTL (time-to-live) value carried in the packet expires. The goal of our work is to explore to the content sharing problem in Smartphone based DTN's.

### III. CONTENT SHARING

In this section we analyse the problem of content sharing in delay tolerant networks and describe alternative solutions. As stated in the introduction, we focus on mobile opportunistic networking scenarios where the nodes communicate using the DTN bundle protocol [18]. Some devices in the network store content which they are willing to share with others. All nodes are willing to cooperate and supply a limited amount of their local system resources (bandwidth, storage, and processing power) to assist other nodes. Our goal is to allow users to issue queries for content that is stored on other nodes anywhere in the network and assess the chances of such a node to obtain the sought information. To facilitate searching, we assume that nodes are able to perform searches on their local storage and find the relevant results for a given query.

The content sharing process is categorized into two phases: content discovery phase and the content delivery phase. In the content discovery phase, the user inputs in a content sharing application requests for content. The application first searches the content in its own database and if not found, the application generates a query that is forwarded based on the user's request. When the content is found, the content delivery phase is initiated, and the content is forwarded to the query originator.

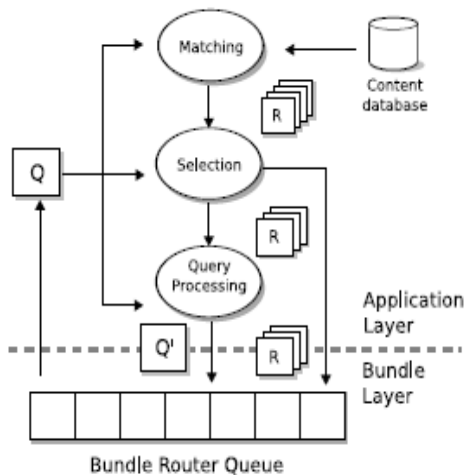


Fig. 1 Processing of incoming query.

#### A. Content Discovery

In content discovery, most systems focus on how to formulate queries, which depends on assumptions about the format of the content to be discovered. A general protocol should support many different types of queries and content, but we abstract from the actual matching process in order to focus on discovering content over the network.

The simplest strategy to discover and deliver contents is Epidemic routing [2]. However, due to resource limitations, Epidemic routing is often wasteful, so we need to consider methods that limit the system resources spent on both

content discovery and delivery. Ideally, a query should only be forwarded to neighbours that hold matching contents or those are on the path to other nodes having matching content [13]. Different nodes should return no overlapping responses to the requester. As global knowledge or active coordination is not an option in our scenario, each node can only make independent forwarding decisions. These forwarding decisions should achieve a good trade off between discovery efficiency and required resources. Similar limitations apply to content delivery. A number of methods proposed by Pitkanen et al. [14] can be used for limiting the spreading of queries. Further, we examine two alternatives for limiting the spreading of queries: a split query lifetime limit and a query distance limit. We use a controlled replication-based [9] routing scheme that outperforms a single-copy scheme. A single-copy scheme turns both query lifetime and distance limits into random walk, and the scheme is not effective when content-carrier nodes (i.e., destinations) are not known. By contrast, the controlled replication-based scheme distributes a fixed number of message copies and avoids the excessive spread of messages. The difference between our work and that of Pitkanen et al. is that our work splits the query lifetime for discovering and delivering contents. Also, proposed work uses spatial limitation for query distribution. The details of the spatiotemporal limitation for query distribution are given below.

When a node needs some contents, it generates query information  $Q$  that contains a node identifier ID (e.g., IMEI number), the creation time of the query  $t_{qc}$ , the query lifetime  $T$ , the replication size, and the  $q$  node's mobility information

$$M_q = \{l_{q,t_{qc}}, l_{q,t_{qc}+\delta}, \dots, l_{q,t_{qc}+k\delta} \mid k = \lceil T/\delta \rceil\}.$$

When a query-carrying node encounters another node, the carrying node performs the following forwarding decision to spread the query:

$$forward(Q) = \begin{cases} \text{true} & \text{if } (t - t_{qc} < (1 - \alpha) \cdot T) \\ \text{false} & \text{else} \end{cases} \quad (1)$$

where  $\alpha \in [0,1]$  is the ratio of the discovery period to the delivery period, which indicates what percentage of the query lifetime is used for discovery and what percentage is used for delivery. We will analyse the appropriate value for in the evaluation section. When the query-carrying node finds that forwarding the query will not influence sharing performance due to the expiration of the query lifetime, the node terminates spreading the query. In case the above forwarding decision function is true, the node performs the next forwarding decision:

$$forward(Q) = \begin{cases} \text{true} & \text{if } (|l_{f,t} - l_{q,t}| < H) \\ \text{false} & \text{else} \end{cases} \quad (2)$$

where  $l_{f,t}$  is the location of the forwarding node  $f$  at time  $t$ ,  $l_{q,t}$  is the predicted location of the query originator node  $q$  at time  $t$ , and  $H$  is the threshold distance for query spreading (e.g., 20 miles), which is set by the user. The importance of the above decision function lies on not spreading the query in unnecessary regions. For example, a

node may carry a query travelling long distance although the content can be found in the node’s residing area. The content delivery cost will be very high, or the content may not be delivered due to the query lifetime expiration. Therefore, limiting the query spread to a certain region is reasonable.

**B. Content Delivery**

When query matching content is discovered, the content carrying node should send only a subset of the results. This requirement is necessary to limit the amount of resources used both locally and globally for transmitting and storing the responses, and to remove potential duplicates. The query originator assigns a limit for both the number of replications and the volume of content that should be generated. When nodes need to forward a query message, the limitations included in the query message are used to make a forwarding decision. If the volume of the content exceeds the response limit, the node needs to select which ones to forward. For example, when AP hotspots are huge in number, the set of AP hotspots with the most reliable Internet speed is chosen and forwarded to the query originator.

In this paper, we propose a general solution for content delivery, and we examine how to deliver contents rather than which content to deliver. Nodes broadcast a one-hop beacon message to indicate their existence. This beacon message contains the list of queries and the list of content headers the node is carrying. Both queries and content headers include the destination node’s (i.e., query generator node) mobility information. Based on the mobility information in the content headers, the nodes decide to exchange actual contents. Allowing all nodes to carry the same content is wasteful, so DPD uses the multiple-copy case Spray and Focus [10] with a special utility function to limit the number of content replicas. Spray and Focus routing has two phases: the spray phase and the focus phase. In the spray phase, for every message originating at a source node, L message copies are initially spread, forwarded by the source and possibly other nodes receiving a copy, to L distinct “relays.” In the focus phase, utility-based routing is used, that is, node i forwards a copy of the message with destination d to another node j, if  $U_j(d) > U_i(d)$ , and for all  $m : U_d(d) > U_m(d)$ . As the efficiency of utility-based routing is decided by the utility function, we concentrate on designing a promising utility function.

Given mobility information  $M_d = \{l_{d,t}, l_{d,t+\delta}, \dots, l_{d,t+k\delta}\}$  of the destination node d and mobility information  $IM_i = \{l_{i,t}, l_{i,t+\delta}, \dots, l_{i,t+k\delta}\}$  of the intermediate node i, the utility function of the intermediate node  $U_i(d)$  is computed as follows:

$$U_i(d) = \sum_{m=t}^{t+k\delta} U_m, U_m = \begin{cases} \frac{t}{m} & \text{if } |l_{i,m} - l_{d,m}| \leq R \\ 0 & \text{else,} \end{cases} \quad (3)$$

where t is the current time,  $k = \lceil T/d \rceil$  is the number of time instances remaining until the query/content lifetime expires, and T is the remaining query lifetime. Here,  $U_i(d) \in [0,1]$  indicates the path similarity between nodes i and d with a duration factor. Thus, nodes that meet the destination node (i.e., come into radio range R) earlier and stay for a longer

duration will have a higher utility function. If a node never comes in the destination node’s radio range, the path similarity will be 0. Then, the utility function is computed again as follows:

$$U_i(d) = -\frac{|l_{i,w} - l_{d,w}|}{R} \cdot \frac{w}{t}, \quad (4)$$

where  $w = \arg \min_m \{|l_{i,m} - l_{d,m}|\}, m = \{t, t + \delta, \dots, t + k\delta\}$ .

Here,  $U_i(d) \in (-\infty, 0)$  indicates how much progress in time and distance the content would make in the case it was carried by an intermediate node i. Thus, nodes which come earlier and closer to the destination node would have a higher utility function.

**IV. IMPLEMENTATION**

With the growing number of smart phone users, peer-to-peer adhoc content sharing is expected to occur more often. Thus, new content sharing mechanisms should be developed as traditional data delivery schemes are not efficient for content sharing due to the sporadic connectivity between smart phones. Currently data delivery is achieved through store-carry-forward protocols, in which a node stores a message and carries it until a forwarding opportunity arises through an encounter with other nodes.

The traditional content sharing scheme mainly had the following disadvantages:

- 1) Did not focus on the geographic search query propagation limit.
- 2) Did not address the problem of indoor content sharing.
- 3) The open nature of peer-to-peer systems exposes them to malicious activity.

In this paper we propose, recommendation based discover-predict-deliver as an efficient content sharing scheme for smart phone-based DTNs. DPD assumes that the communications between smart phones arise in a set of locations where Smartphone carriers stay for a long duration. It employs distributed algorithms to recommend other peers about the malicious activity of a peer.

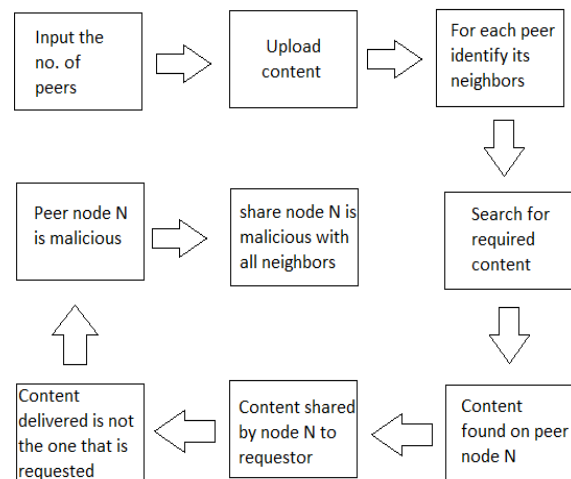


Fig. 3 Block diagram of the proposed Architecture

The proposed system has the following advantages:

- 1) We develop an efficient and robust content sharing scheme that correctly delivers contents to the destination.

- 2) We evaluate the proposed scheme by implementing using java platform based on smart phone nodes.
- 3) We enable a peer to reason about the trustworthiness of other peers based on past interactions and recommendations.

The modules in the proposed system are as follows:

- 1) Dynamic Neighbour Discovery
- 2) Movement Tracking
- 3) Mobility Learning
- 4) Discovering and Learning Meaningful Places
- 5) Mobility Prediction
- 6) Trustworthiness of Peer

1) *Dynamic Neighbour Discovery*: A Neighbour discovery is an important task for routing protocols. Especially in delay-tolerant networking, efficient neighbour discovery significantly improves the performance of the routing protocols. However, most protocols validated with simulations do not address this issue as these protocols assume that nodes always perceive neighbours with frequent hello messages. In real implementations, frequent hello messages are not acceptable due to high energy consumption. In our implementation, we have found that the content sharing performance can be improved with a simple dynamic neighbour discovery. In dynamic neighbour discovery, each peer node can discover its neighbours by adding their neighbours in the database. This is achieved by adding peer name, port number and system name for each neighbour. Thus neighbours are easily discovered.

2) *Movement Tracking*: In Life Map [19], the Activity Manager monitors the acceleration vector of a three-axis accelerometer and detects the motion of the user. The motion detector function of the Activity Manager is basically a classifier M that has two outputs: moving or stationary. When the user is walking, running, or moving in a vehicle, the motion is classified as moving, whereas when the user stays at a certain location, the motion is classified as stationary.

3) *Mobility Learning*: In daily life, people typically visit a number of places, but not all of these are meaningful for learning people’s mobility. Indeed, DPD requires the discovery of locations where content sharing can be performed. Content sharing is successfully performed in places where Smartphone users stay long enough, as perceiving the existence of other nodes and message exchanging requires several minutes depending on the size of the message, the bandwidth, and the network interface. Hence, we are basically interested in discovering places where the user stays longer than certain duration (i.e., meaningful places) and the context in user movement (i.e., paths). Currently available location technologies focus on providing geographical information. This information is insufficient to discover meaningful places because the physical location is not exactly generated at the same place despite the fact that a user generally has a similar life pattern every day. In addition, this information cannot distinguish a place that has a similar geocode but different floors. In modern society, places are normally located in multiple floor buildings. Thus, the logical information of

meaningful places has more benefit to the proposed scheme as content sharing is conducted in indoor environments.

4) *Discovering and Learning Meaningful Places*: Currently available location technologies focus on providing geographical information. This information is insufficient to discover meaningful places because the physical location is not exactly generated at the same place despite the fact that a user generally has a similar life pattern every day. In addition, this information cannot distinguish a place that has a similar geocode but different floors. In modern society, places are normally located in multiple floor buildings. Thus, the logical information of meaningful places has more benefit to the proposed scheme as content sharing is conducted in indoor environments.

5) *Mobility Prediction*: As DPD uses location information to estimate if a node approaches the destination of the content or diverges from the destination, the prediction of nodes’ mobility information is essential.

6) *Trustworthiness of peers*: Here we present distributed algorithms that enable a peer to reason about the trustworthiness of other peers based on past interactions and recommendations. Peers can create their own trust network in their proximity by using local information. Two contexts of trust, service, and recommendation are defined to measure trustworthiness in providing services and giving recommendations. Interactions and recommendations are evaluated based on importance, recentness, and peer satisfaction parameters.

The operations when receiving a recommendation and having an interaction are as follows:

- i. Service Trust Metric
- ii. Reputation Metric
- iii. Recommendation Trust Metric
- iv. Selecting Service Providers

Figure 4 shows the operations when receiving a recommendation and having an interaction.

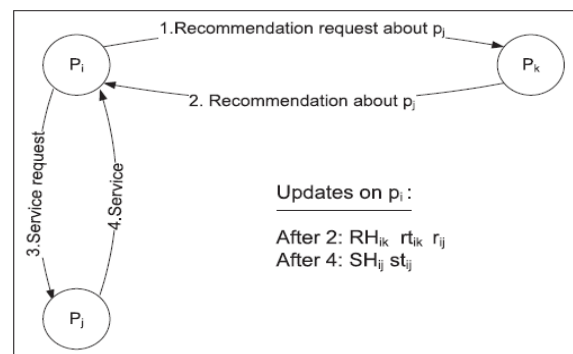


Fig.4 Operations when receiving a recommendation and having an interaction.

- i. *Service Trust Metric*:

When evaluating an acquaintance’s trustworthiness in the service context, a peer first calculates competence and integrity belief values using the information in its service



history. Competence belief represents how well an acquaintance satisfied the needs of past interactions .Let friend request denote the competence belief of  $p_i$  about  $p_j$  in the service context. Average behaviour in the past interactions is a measure of the competence belief. A peer can be competent but may present erratic behaviour. Consistency is as important as competence. Level of confidence in predictability of future interactions is called integrity belief. Let  $I_{bij}$  denote the integrity belief of  $p_i$  about  $p_j$  in the service context. Deviation from average behaviour ( $cb_{ij}$ ) is a measure of the integrity belief.

ii. Reputation Metric

The reputation metric measures a stranger’s trust worthiness based on recommendations. In the following two sections, we assume that  $p_j$  is a stranger to  $p_i$  and  $p_k$  is an acquaintance of  $p_i$ . If  $p_i$  wants to calculate  $r_{ij}$  value, it starts a reputation query to collect recommendations from its trustworthy acquaintances and requests their recommendations. Let  $\eta_{max}$  denote the maximum number of recommendations that can be collected in a reputation query and  $|S_j|$  denote the size of a set  $S$ . In the algorithm,  $p_i$  sets a high threshold for recommendation trust values and requests recommendations from highly trusted acquaintances first. Then, it decreases the threshold and repeats the same operations.

iii. Recommendation Trust Metric:

Facebook has an incredible audience, 950 million strong and counting. This audience is immensely attractive to Brands and Marketers around the world. We've seen explosive growth in brand pages, types of advertising and other fun ways to monetize this audience. Don't invent new metrics, use online versions of Reach and GRPs to measure success. The value of Facebook in "spreading word of mouth," "getting your brand in front of friends of fans," and "engaging fans with five to seven posts a week on your fan page."They closed with the Facebook Insights tool (which is quite nice). This blog post is about the above recommendations, and their merit.

Assume that  $p_i$  wants to get a particular service.  $p_j$  is a stranger to  $p_i$  and a probable service provider. To learn  $p_j$ 's reputation,  $p_i$  requests recommendations from its acquaintances. Assume that  $p_k$  sends back a recommendation to  $p_i$ . After collecting all recommendations,  $p_i$  calculates  $r_{ij}$ . Then,  $p_i$  evaluates  $p_k$ 's recommendation, stores results in  $RH_{ik}$ , and updates  $rt_{ik}$ . Assuming  $p_j$  is trustworthy enough,  $p_i$  gets the service from  $p_j$ . Then,  $p_i$  evaluates this interaction and stores the results in  $SH_{ij}$ , and updates  $st_{ij}$ .

iv. Selecting Service Providers:

When  $p_i$  searches for a particular service, it gets a list of service providers. Considering a facebook application, either post share the links to other peer .Connecting the all people with recommendation of multiple peers, checking integrity is a problem since any part of the file downloaded from an uploader might be inauthentic. Service provider selection is done based on service trust metric, service history size, competence belief, and integrity belief values.

When  $p_i$  wants to download a file, it selects an uploader with the highest service trust value.

TABLE I

```

Algorithm 1. GETRECOMMENDATIONS( $p_j$ )
1:  $\mu_{rt} \leftarrow \frac{1}{|A_i|} \sum_{p_k \in A_i} rt_{ik}$ 
2:  $\sigma_{rt} \leftarrow \frac{1}{|A_i|} \sqrt{\sum_{p_k \in A_i} (rt_{ik} - \mu_{rt})^2}$ 
3:  $th_{high} \leftarrow 1$ 
4:  $th_{low} \leftarrow \mu_{rt} + \sigma_{rt}$ 
5:  $rset \leftarrow \emptyset$ 
6: while  $\mu_{rt} - \sigma_{rt} \leq th_{low}$  and  $|rset| < \eta_{max}$  do
7:   for all  $p_k \in A_i$  do
8:     if  $th_{low} \leq rt_{ik} \leq th_{high}$  then
9:        $rec \leftarrow \text{RequestRecommendation}(p_k, p_j)$ 
10:       $rset \leftarrow rset \cup \{rec\}$ 
11:     end if
12:   end for
13:    $th_{high} \leftarrow th_{low}$ 
14:    $th_{low} \leftarrow th_{low} - \sigma_{rt}/2$ 
15: end while
16: return  $rset$ 
    
```

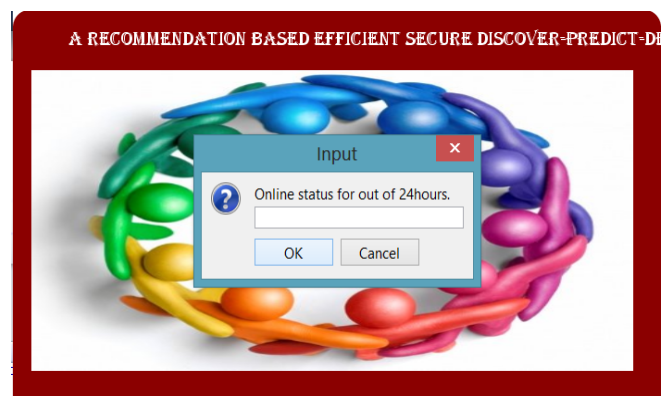
Fig. 5 Algorithm for getting recommendations.

V. RESULTS

After implementing the proposed system on java platform using NetBeans IDE, the results obtained are as follows:



The execution begins by running the Main.fx file. We create peer nodes by clicking the PEER button. We can create any number of peers . The Exit button exits all the peers.



The online status out of 24 hours for each peer is entered which indicates the online time for each peer.



Each node can add its neighbours by entering the peer name, node number and system name of the neighbouring peers.

Each peer has a unique peer name and port number that identifies each peer. There are many operations that each peer can perform such as:

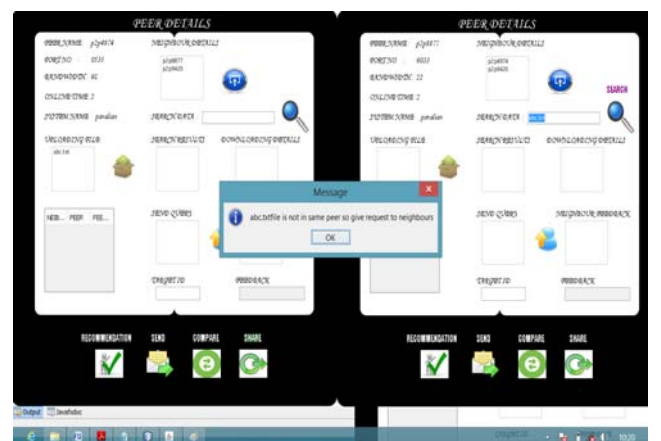
- Add Neighbour, which adds the neighbour nodes of each peer.
- Upload File, which allows peer nodes to upload files so that they are visible to other neighbouring nodes.
- Search Data, which allows nodes to search for files in neighbouring peers.
- SEND, which allows a node to send a file to its neighbour.
- COMPARE, which allows a node to compare a requested file to the one that is downloaded.
- RECOMMEND, which allows a node to recommend other nodes about a malicious node
- SHARE, which allows to shares the malicious activity of one peer information to all the neighbouring nodes.



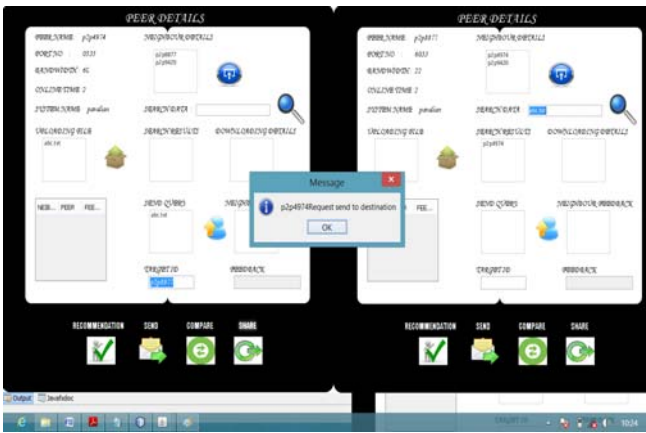
Every node uploads the files that are available in its database so that other nodes are able to request such files.



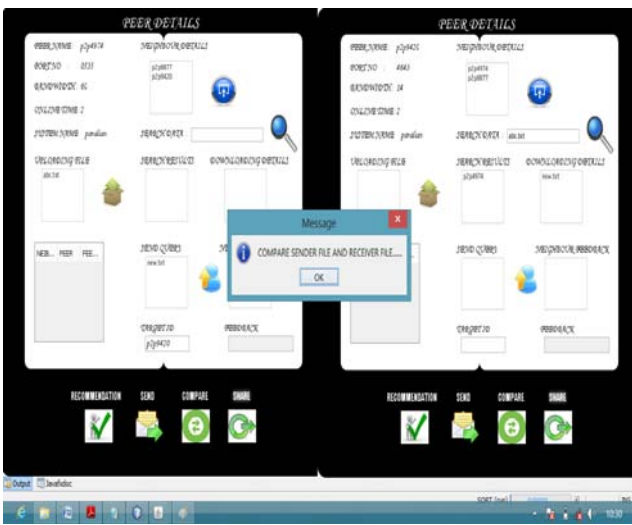
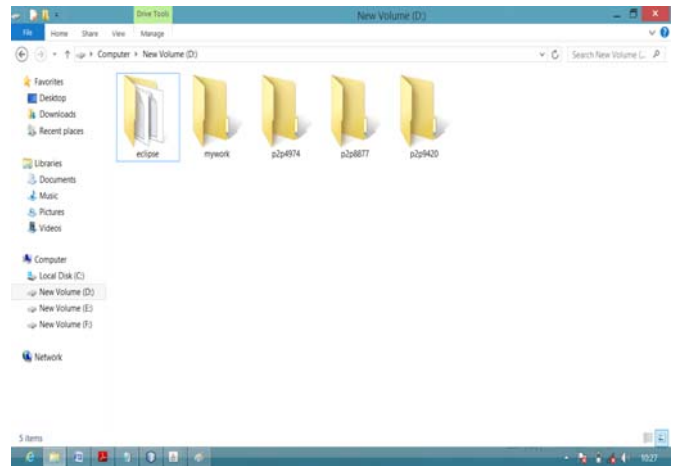
We can create any number of peer nodes each with a unique peer name and port number that helps them to communicate with each other.



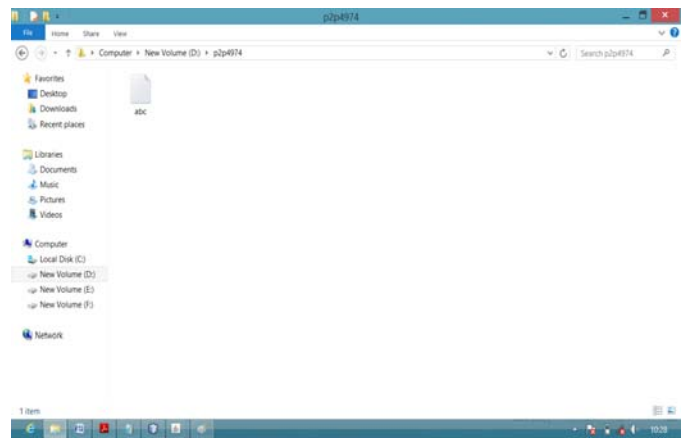
A peer searches for a file by entering the name of the file. The file if not available in its database, the request is sent to all neighbours.



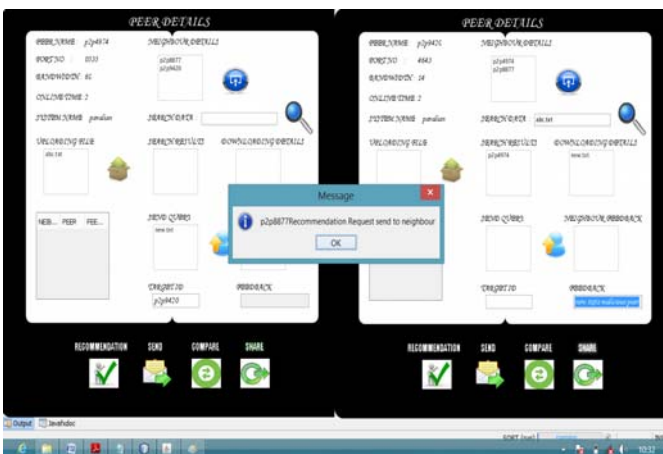
The file that is requested is found in neighbouring peer and it is sent to the requestor.



The requested file and downloaded file is compared. If they are not same, then the sender is a malicious peer. This information is shared with all neighbours using by recommendation.



We have evaluated the proposed protocol and observe that contents are efficiently discovered and shared with the neighboring peers. Also malicious peers are correctly identified.



Finally, we can see in D:\ there are 3 folders with peer names. These are the workspaces for each peer. When the contents are shared, they can be easily seen in their respective folders.

Also we have compared the proposed recommendation based content sharing scheme with Epidemic and Spray and Wait protocols. The results are compared based on the sharing efficiency which is the percentage of discovered and delivered contents in the total generated contents. Figure below shows the comparison graphs.

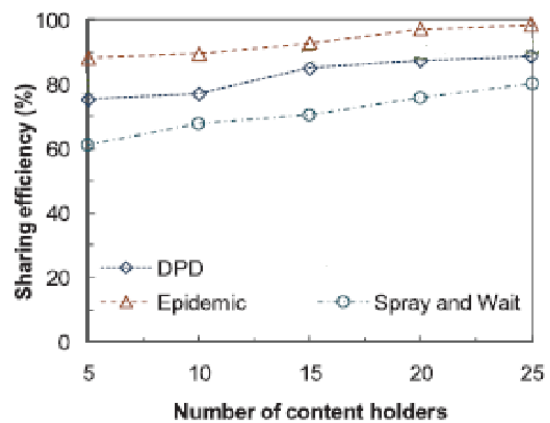


Fig. 5 Proposed system vs. Existing system

## VI. CONCLUSIONS

In this paper we have proposed an efficient content sharing scheme for Smartphone based DTN's. We have proposed discover-predict-deliver as a content sharing mechanism that discovers the content and delivers it to the proper destination. The scheme also provides the mobility information of individuals. We have attempted to utilize the availability and communication technologies of today's Smartphone. We have also compared our proposed scheme with the Epidemic [2] and Spray-and-Wait [8] protocols.

In this paper, we have also proposed recommendation scheme that enables a peer to reason about the trustworthiness of other peers. This helps peers to mitigate attacks of malicious activity. The simulation of the proposed scheme shows that contents can be correctly delivered to the neighboring nodes.

Finally, we can say that our system can still be improved by considering the privacy of the user.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MEST) (No. 2011-0000156, 2011-0015332). A preliminary version of this study has been presented in IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 12, NO. 13, MARCH 2013 [20]. Any view, findings and conclusions expressed in this paper are those of the authors.

## REFERENCES

- [1] T3I Group LLC, <http://www.telecomweb.com>, 2010.
- [2] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," technical report, Dept. of Computer Science, Duke Univ., Sept. 2000.
- [3] A. Balasubramanian, B.N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," Proc. ACM SIGCOMM, pp. 373-384, 2007.
- [4] R.C. Shah, S. Roy, S. Jain, and W. Brunette, "Data Mules: Modeling a Three-Tier Architecture for Sparse Sensor Networks," Elsevier Ad Hoc Networks J., vol. 1, pp. 215-233, Sept. 2003.
- [5] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," SIGMOBILE Mobile Computer Comm. Rev., vol. 7, no. 3, pp. 19-20, 2003.
- [6] C. Liu and J. Wu, "An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks," Proc. ACM MobiHoc, pp. 14, 2009.
- [7] J. Wu, M. Lu, and F. Li, "Utility-Based Opportunistic Routing in Multi-Hop Wireless Networks," Proc. 28th Int'l Conf. Distributed Computing Systems (ICDCS '08), pp. 470-477, 2008.
- [8] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking (WDTN '05), pp. 252-259, 2005.
- [9] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case," IEEE/ACM Trans. Networking, vol. 16, no. 1, pp. 63-76, Feb. 2008.
- [10] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," IEEE/ACM Trans. Networking, vol. 16, pp. 77-90, Feb. 2008.
- [11] I. Cardei, C. Liu, J. Wu, and Q. Yuan, "DTN Routing with Probabilistic Trajectory Prediction," Proc. Third Int'l Conf. Wireless Algorithms, Systems, and Applications (WASA '08.), pp. 40-51, 2008.
- [12] Q. Yuan, I. Cardei, and J. Wu, "Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks," Proc. 10th ACM MobiHoc, pp. 95-104, 2009.
- [13] N.B. Chang and M. Liu, "Controlled Flooding Search in a Large Network," IEEE/ACM Trans. Networking, vol. 15, no. 2, pp. 436-449, Apr. 2007.
- [14] M. Pitkanen, T. Karkkainen, J. Greifenberg, and J. Ott, "Searching for Content in Mobile DTNs," Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PERCOM '09), pp. 1-10, 2009.
- [15] I. Smith, J. Tabert, T. Wild, A. Lamarca, A. Lamarca, Y. Chawathe, Y. Chawathe, S. Consolvo, S. Consolvo, J. Hightower, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild," Proc. Third Int'l Conf. Pervasive Computing (PERCOM '05), pp. 116-133, 2005.
- [16] Z. Ghahramani, "An Introduction to Hidden Markov Models and Bayesian Networks," Hidden Markov Models: Applications in Computer Vision, pp. 9-42, World Scientific, 2002.
- [17] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi, "Understanding Individual Human Mobility Patterns," Nature, vol. 453, pp. 779-782, 2008.
- [18] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Network Architecture," RFC4838, Jan. 2007.
- [19] Y. Chon and H. Cha, "LifeMap: Smartphone-Based Context Provider for Location-Based Service," IEEE Pervasive Computing Magazine, vol. 10, no. 2, pp. 58-67, Apr.-June 2011.
- [20] Elmurod Talipov, Yohan Chon, and Hojung Cha, Member, IEEE- "Content Sharing over Smartphone-Based Delay-Tolerant Networks"- IEEE Transactions On Mobile Computing, Vol. 12, No. 3, March 2013