國立政治大學資訊科學系

Department of Computer Science

National Chengchi University

碩士論文

Master’s Thesis

用於混合式耐延遲網路之適地性服務資料搜尋方法

Location-based Content Search Approach in Hybrid Delay Tolerant Networks

研　究　生：李欣諦

指導教授：蔡子傑

中華民國一零一年七月

July 2012

用於混合式耐延遲網路之適地性服務資料搜尋方法

Location-based Content Search Approach in Hybrid Delay Tolerant Networks

研 究 生：李欣諦　　Student：Hsin-Ti Lee

指導教授：蔡子傑　　Advisor：Tzu-Chieh Tsai

國立政治大學
資訊科學系
碩士論文

A Thesis
submitted to Department of Computer Science
National Chengchi University
in partial fulfillment of the Requirements
for the degree of
Master
in
Computer Science

中華民國一零一年七月

July 2012

用於混合式耐延遲網路之適地性服務資料搜尋方法

摘要

在耐延遲網路上，離線的使用者，可以透過節點的相遇，以點對點之特定訊息繞送方法，將資訊傳遞至目的地。如此解決了使用者暫時無法上網時欲傳遞資訊之困難。因此，在本研究中，當使用者在某一地區，欲查詢該地區相關之資訊，但又一時無法連上網際網路時，則可透過耐延遲網路之特性，尋求其它同樣使用本服務之使用者幫忙以達到查詢之目的。

本論文提出一適地性服務之資料搜尋方法，以三層式區域概念，及混合式節點型態，並透過資料訊息複製、查詢訊息複製、資料回覆及資料同步等四項策略來達成使用者查詢之目的。特別在訊息傳遞方面，提出一訊息佇列選擇演算法，賦予優先權概念於每一訊息中，使得較為重要之訊息得以優先傳送，藉此提高查詢之成功率及減少查詢之延遲時間。最後，我們將本論文方法與其它查詢方法比較評估效能，其模擬結果顯示我們提出的方法有較優的查詢效率與延遲。

關鍵字：耐延遲網路、適地性服務、內容、查詢、路由協定

Location-based Content Search Approach in Hybrid Delay Tolerant Networks

Abstract

In Delay Tolerant Networks (DTNs), the offline users can, through the encountering nodes, use the specific peer-to-peer message routing approach to deliver messages to the destination. Thus, it solves the problem that users have the demands to deliver messages while they are temporarily not able to connect to Internet. Therefore, by the characteristics of DTNs, people who are not online can still query some location based information, with the help of users using the same service in the nearby area.

In this thesis, we proposed a Location-based content search approach. Based on the concept of three-tier area and hybrid node types, we presented four strategies to solve the query problem. They are Data Replication, Query Replication, Data Reply and Data synchronization strategies. Especially in message transferring, we proposed a Message Queue Selection algorithm. We set the priority concept to every message such that the most important one could be sent first. In this way, it can increase the query success ratio and reduce the query delay time. Finally, we evaluated our approach, and compared with other routing schemes. The simulation results showed that our proposed approach had better query efficiency and shorter delay.

Keywords: Delay Tolerant Networks, Location-based, Content, Query, Routing protocol

致謝辭

研究所這兩年，時間不長也不短，但學習到的事物卻相當的有份量。過去是資訊管理系的我，鼓足勇氣，一腳踏入資訊科學領域，這過程相當感謝許多人的指導以及提攜，時常聽人說，讀研究所是為了提升自己獨立解決問題的能力，而在政大資科所的這兩年訓練想必也有所成長。

能夠完成這份論文，最要感謝指導教授蔡子傑老師的細心教導，讓我能順利的從一開始的發想到確定方向，進而一步一步的完成碩士論文，在每一次的討論，老師都能精闢點出問題所在，並給予受用的回饋，讓我在研究過程中有所指引而不迷失方向，使得能如此順利的完成碩士論文，由衷的感謝蔡子傑老師的提攜及指導。

而接著我要感謝幫助我能快速的瞭解實驗室研究及相關生活問題解答的學長姐，文卿、勇麟、偉敦、凱禎、彥嵩、賀翔，其中要感謝即時雨勇麟學長，在我剛進入政大，還在為生活費苦惱時，即時的提供了英文系工讀的機會，讓我能在研究所這兩年間不需煩惱生活費問題，而同時也感謝我的同學們，英明、昶瑞、冠傑、郁翔在修課及研究上能互相討論及幫忙，另外也謝謝學弟們，煜泓、宇軒、建淳在校慶軟體開發時的幫忙，才能順利的完成老師的各項需求，此外，我要感謝我的好朋友舜然，在辛苦上班結束後還能播空陪我一起玩遊戲舒解壓力，在我煩悶的研究生活中，讓我能在撰寫論文時繃緊的神經能有所放鬆。最後祝福各位在未來不論是工作或求學期間都能夠事事順心、健康快樂。

最後我要感謝我的家人，在我逆境時能給予最大的支持及鼓勵，讓我能一次又一次的突破困境，一路讀到了碩士，進而取得學位，沒有你們，我的求學之路不會如此順遂，謝謝你們。

TABLE OF CONTENT

LIST OF FIGURE

LIST OF TABLE

CHAPTER 1

Introduction

1.1 Background

Since the Android phone, iPhone, iPad appeared, smartphone and tablet are explosive growth [24]. In the past, user used to take the laptop to go anywhere. Now, almost everyone has smartphone. And after Apple Inc. announced iPad, it set off a buying spree. This phenomenon subverts the traditional user behavior, so someone [25] called the "Post-PC era" coming. Therefore, mobile correlation technique becomes the hot issues in these years.

The mobile network technology is approaching maturity, now. Such as 3G(UMTS, CDMA2000), 3.5G(HSDPA) or 3.9G(WiMax). And they already have commercial products. So people can use their mobile device playing Facebook or searching some information about their location. Although the usability of the mobile network is well, it may not be used in some scenarios. (1) No network coverage: People in some position they can't receive the network signal, or (2) Too many users: if there are too many use the same service in the same area, it will cause serious network congestion, even the network will be interrupt, or (3) Infrastructure broken: if the base station of the cellular network or the Wi-Fi access point doesn't work, users can not connect to the Internet, or even (4) the user never use the mobile network, etc. Both of these scenarios are parts of the disconnect reasons. Despite the user can't connect to the Internet. If user still wants to send message to someone or search some local information, there is a technique which could achieve user's requirement, Delay Tolerant Networks.

Delay Tolerant Networks is an intermittently connective network [1], all nodes are lacking of continue connectivity [2]. This is used in some place, just like Battlefield [7], Outer space [8], disaster or emergency environment [9] [10], etc. Node-to-Node transfer message is that main feature. However, node sends message to the destination by encounter nodes, it will spend much time on delivery. But, if you want to send the message out, in DTNs, it is possible to deliver message in spite of the high transmission time. And we will discuss the related research in Section 2.

1.2 Motivation

As we in an area, we want to search some information about that area, but, we cannot connect to Internet. So what can we do? In DTNs, there are two ways to do. First, ask the encounter node for the information. If it doesn't have, we can ask it for help to spread the searching message. This action could increase the searching success probability.

We can see Figure 1. At the time t1, Node A wants to search information in this area, but it can't connect to the Internet. Thus, Node A asks its encounter node B for information. But Node B doesn't have the information which Node A required. Then Node B helps Node A for searching information. At the time t2, Node B encounters Node C, and Node C can connect to the Internet. So Node B asks Node C for the information, Node C can search in the remote server, and then at time t3, Node C replies the information which Node B required. Finally, at the time t4, Node A receives the searching information from Node B.

This is a simple scenario we discuss above. In the reality network, it isn't easy to search information. We have to overcome some issues, and we will discuss it in the next section.

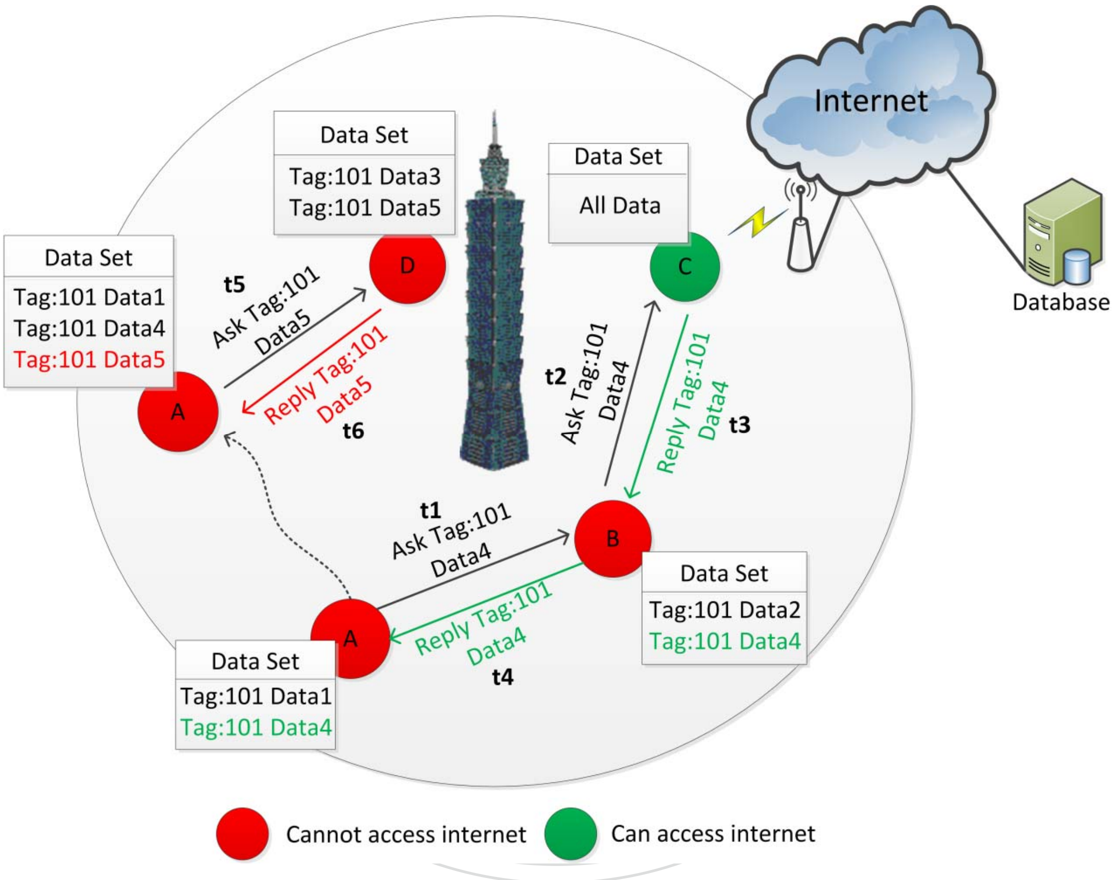


Figure 1: Problem scenario

1.3 Objective

Our objective is user in an area could get the information in a shortest time. In order to achieve this purpose, we assume all nodes in the network are collaborative and some nodes can connect to the Internet. The collaborative means all nodes are willing to store other messages into their storage and help other nodes to delivery messages. And we don’t consider the incentive issue. Thus, there are some challenge issues we have to discuss, and we divided three parts of these, then described below:

1.3.1 Replication issues

(1) Data Replication: Something unique data usually stored in a place or in a node, if we want to get this data, it is difficult to query it. Due to we don’t know where it is, we have to node-to-node ask for the data. It would spend much time to search data, and it would have high probability of searching failure. So, if the data could spread around the network, it would increase the searching success probability. But, if the data is unrestricted spread, it will cause the higher overhead, like Epidemic. So we have to investigate how to replicate the data, and how to decide the data transfer. We will describe in Section 3.1.

(2) Query Replication: Like the Data Replication issue. We want to search some data, and we have to send query message to get the data. If we have just one query copy, it is difficult to get. So, if we can ask the other node for help, it can increase the query success probability. But, we cannot unlimited spread the query messages and we have to investigate a suitable rule to replicate it. We will describe in Section 3.2.

1.3.2 Data Reply issue

When the query message looked up the data we want, the replier has to reply the data back to the querier. But, in the network, every node will move, we can’t return the data from the original path. So we have to find the other way to transfer the data. Therefore, we will investigate how the replier response the data to the querier, and this is the most important issue in the thesis. Then, the detail rule we will describe in Section 3.3.

1.3.3 Data Synchronize issue

This is an independent issue in the thesis. Every node can create new data. If node can't connect to the remote database server, it just can share the data by the encounter node. If we would like to share this data with every node, we have to sync this data to the remote server. Thus, every node which can connect to the Internet in the network can query this data. Therefore, if we can't connect to the Internet, we have to ask the node who can connect to for help to sync the data. The detail will be described in Section 3.4.

1.4 Organization

The rest of this thesis is organized as follows. Chapter 2 introduces related works in common DTN routing protocol and the Query-based DTN routing protocol. In Chapter 3, we present our searching rule, location-based content search approach. In Chapter 4, we present the simulation results. The final, Chapter 5, we concludes this work and discuss the future work of this thesis.

CHAPTER 2

Related Work

In such a variety of DTN routing researches, we focus on query-based routing schemes. And we divided the routing protocol into two parts, Common DTN routing protocol and Query-based DTN routing protocol, the former, we will introduce some common routing approaches, like Epidemic, Spray and Wait etc., and the latter, we will discuss the approaches which are related to our work. Then, the routing protocol will be discussed below.

2.1 Common DTN routing protocol

We can divide into two kinds of routing protocol: Flooding-based routing and Forwarding-based routing.

2.1.1 Flooding-based routing schemes

Flooding-based means node will have a number of the message copies [15], and it will send this copies to all the encounter nodes. We introduce some approaches below. (1) The basic approach, Epidemic [3], whatever nodes encountered, it will send all the messages it have to encounter nodes. It doesn’t have any selection rule, just sending until the connection dropped. This approach could get high performance about message delivery success ratio and delivery latency, because a message will be replicated to many nodes, it has high opportunity to send to the destination. But, it will make high overhead, because there are too many redundant messages, and if nodes don't have enough buffer size, it will overflow quickly. In order to overcome this defect, (2) Spyropoulos et al. proposed Spray and Wait approach [4], they limit the messages replication to L copies, and there are two phase: Spray phase and Wait phase. In Spray phase, when source node encounters another Node B, it will give half copies to Node B to spread these copies. And the other nodes do the same rule, until nodes just have one copy, then they turn to wait phase. In wait phase, all the nodes which have the message copy couldn't send the copy to another node only if they encounter the destination node. This approach will greatly reduce the transmission overhead, but, it just uses the limitation copies, so the message delivery ratio will be not good as Epidemic. In order to improve this defect, (3) the same authors propose an enhance scheme [5]. They change the wait phase [4] where are routed using direct transmission, they let the waiting nodes have the opportunity to send the message out. It records every encounter timers, and the timer related a utility value, then use this value to decide whether delivery or not. This rule will keep the low overhead, and enhance the message delivery ratio. There are some approaches [6][12][16] use the history information to predict the delivery probability and decide the transmission. (4) PRoPHET is the representative of this kind of routing approach. Every node has to maintain a delivery predictability table, and every encounter they calculate new prediction value. When nodes encountered, they compare their prediction value, then delivery messages to node have highest prediction value. It gets higher delivery rate than Epidemic, but, because of it spends more time to calculate the prediction value and maintain the predictability table, its delivery latency doesn't well enough.

2.1.2 Forwarding-based routing schemes

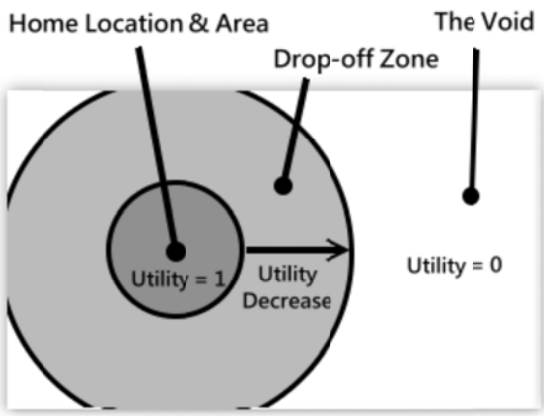
Forwarding-based routing protocols in the network, a message just have one copy [15], and then use different approaches to forward the copy to destination. We introduce some kinds of these schemes, (1) Location-based Routing [17]: it uses GPS to locate nodes position, through different distance algorithm to predict the probability of the message forward to the destination. When nodes encountered, they will compare which one has the highest probability to get to the destination position, then carry it to forward. (2) Gradient Routing: this approach give every node a weighted value, and it represents whether this node suitable for sending message to receiver or not. Different approaches have different way to calculate weighted value. ZebraNet [19] uses history of nodes tracking to get the weighted value, then determine message deliver. In [18], they use social network concept to classify their network node like bubbles, and use the ranking value to decide the messages to.

2.2 Query-based DTN routing protocol

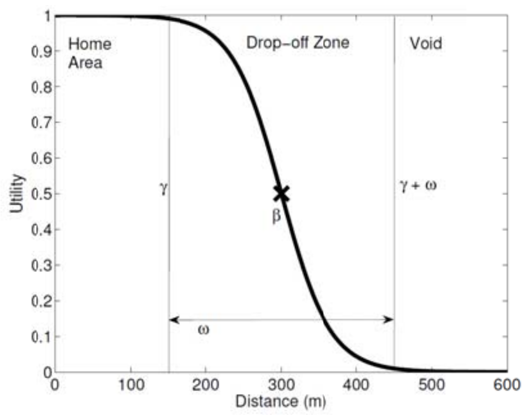
In the past research, many papers [3-6, 15-20] focus on how to send data from source node to destination node, and it doesn't matter the message response problem. It is the one way transmission protocol. If a node sends a request to destination node, it wouldn't know the message being delivered or not. Therefore, one way transmission is not enough, we have to consider how to response the message that we can look up some information. Several works focus on query in DTNs [11-14, 22], in these papers, not only nodes send request to the message owner, but the owners have to send the response message back. The challenge is all nodes are moving and we don't know where they going, so asker is difficult to get the answer. We introduce four works about this kind of request and response pattern research. Then, it describes below:

2.2.1 Locus: A Location-based Data Overlay for Disruption-tolerant Networks [11]

This research proposed a query mechanism, and use three-tier location area concept (see in Figure 2.a). Let users could look up the relevant information according to its position. Every data object will be given a utility value, and this value will vary with the distance between the creation position of data object and the position of the node which carry this data. Figure 2.b shows the changing of the utility value. If data object near its creation location, the utility of that data object will be high. When nodes encountered, they will change their data objects. Before changing objects, node sorts their data objects by utility value. And the data objects with the higher utility value will be sent first. It is because these data objects were created near here. And this rule would let the data objects be centered on its home area, then, users could query data quickly. But it just use the distance between node and home area, if a node will go out this area, but it is near the home area, encounter node still send data objects to it, then, the transmission is waste. And Locus doesn't propose an information reply approach, it just uses the multi-copy routing [4][15] to reply data objects.



a. Data object utility region



b. Utility Function

Figure 2: Locus approach [11]

2.2.2 Performance Evaluations of Data-centric Information Retrieval

Schemes for DTNs [12]

This research proposed a complete approach of data search steps, it includes Data copy, Query copy and Data reply. The main contribute of the research is they use Friendliness Metric (FM, it is the average number of nodes it encounters during an observation window) to decide the data delivery. In Data copy, they propose K-copy replicate scheme, like the spray phrase of Spray and Wait routing. When Node A encountered Node B, Node A will compare the FM value with Node B, If Node B's value greater than Node A's, it indicates Node B will encounter more nodes, and Node A gives a data copy of the K copies to Node B. Then, Node A follows this rule to send copies until it has only one copy. This is the data spreading rule. In Query copy, they propose two schemes, WSS (W-copy selective query spraying) and LNS (L-hop neighborhood query spraying). In WSS, every query have w number copies, when nodes encountered, they compare the FM value first, which one have the highest value, the other node gave half query copies to it, and then keep spreading until w = 1.

In LNS, it is single-copy scheme, they use L hops to limit to query spread. The scheme doesn't compare the FM value. When Node A encountered Node B, Node A asks Node B for query data first, if not, Node A sends the only one query to Node B, then the L hops decreased once. Until L hops is 0, it doesn't spread anymore. In Reply Data, this approach [12] just use the PRoPHET [6] routing scheme to reply data to Querier. This research assumes the node moving in a fixd moving pattern, so nodes could get the accurate FM value. If nodes move in an unpredictable path, the FM value might be inaccurate.

2.2.3 Cooperative File Sharing in Hybrid Deelay Tolerant Networks [13]

This paper introduces a Hybrid DTN routing scheme in file sharing. In DTNs, some nodes can use Wi-Fi, 3G, etc. connects to the Internet, and some nodes can't access Internet. Every data will be divided into many fragments. And every fragment has a metadata to descript. Then these metadata will be spread out the network. When user wants to query a data, it has to find all the metadata of the data to download all the fragments. Like BitTorrent [23] shares rule. But, if one fragment misses, node couldn't return to the original data.

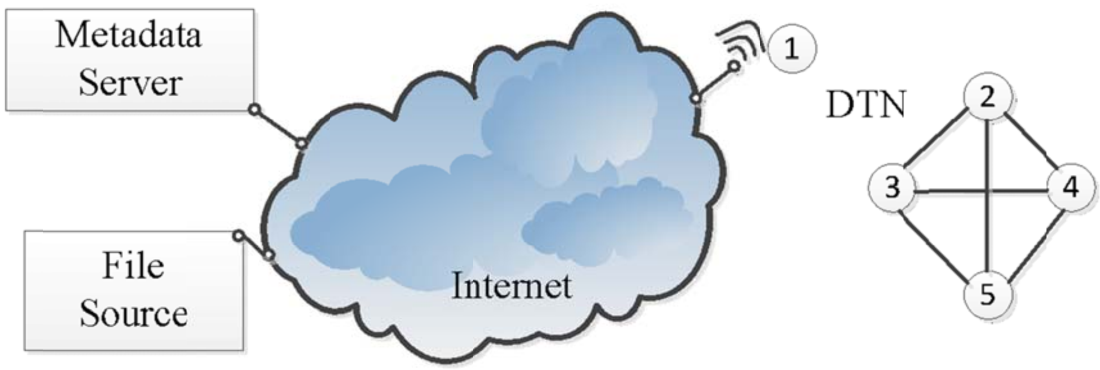


Figure 3: Hybrid DTN example [13]

2.2.4 Searching for Content in Mobile DTNs [14]

This research focuses on query how to forward and how many queries are replicated. They propose an estimation rule to avoid the query replicates too much. When Node A encountered Node B, Node A will estimate the query had been replicated numbers first. The estimate value is calculated by query hop count and network nodes, then, Node A could get a Utility value (u_{est}), if it greater than a threshold T, Node A could send the query to Node B. If not, Node A couldn't. Therefore, if this approach can estimate the query replication number accurately, it will greatly decrease transmission overhead and look up data efficiently. But, they just use common multi-copy routing schemes to reply data. And the simulation result, they don't have compare with other approaches, so we can't know how this research performed.

CHAPTER 3

Location-Based Content Search Approach

We proposed Location-Based Content Search Approach (LCS) in hybrid DTN to solve the query problem in case of disruptive Internet connection. In our research scenario, all nodes in the network are divided into two types, (1) Online-node and (2) Offline-node. Online-node can connect to the internet to access the remote server, and Offline-node cannot connect to that. All nodes have own database and it includes four tables (see Figure 4), (a) Data Message Table、(b) Encounter Nodes Table、(c) Query Message Table and (d) Reply Message Table. Then we describe below:

(a) Data_table: This table is in charge of the data message storing which is created from node self or received from the other nodes. DataId is the data message unique id, UID is the node id who created this data message, Time is the message creation time or replication time, Data is this message's content, Sync is whether this message synchronized to the remote server or not. The detail will be described in Section 3.1.

(b) ENs_table: This table is responsible for recording encounter node. UID is the node id of the encounter node, Time is the encounter time, Longitude and Latitude is the encounter location.

(c) Query_table: This table is in charge of the query message storing, if node is

Online-node, all query messages are received from other Offline-node. QryId is the query message unique id, Time is this message creation time or replication time, QryMsg is the content, isQueried is whether this message queried data or not. The detail will be described in Section 3.2.

(d) Reply_table: this table is in charge of the reply message storing which is created from the node that received query message and had match data. RepId is the unique id of this message, Time is this message creation time or replication time, Data is the reply data content, DesUID is the unique id of the destination node. The detail will be described in Section 3.3.

Data_table

Column name	Attribute
DataId	INT
UID	INT
Time	DATETIME
Data	STRING
Sync	BOOLEAN

ENs_table

Column name	Attribute
UID	INT
Time	DATETIME
Longitude	DOUBLE
Latitude	DOUBLE

Query_table

Column name	Attribute
QryId	INT
Time	DATETIME
QryMsg	STRING
isQueried	BOOLEAN

Reply_table

Column name	Attribute
RepId	INT
Time	DATETIME
Data	STRING
DesUID	INT

Figure 4: Database Tables

In this thesis, our goal is wish the Offline-node could get the information they need by the other nodes as soon as possible. In order to reach this objective, we proposed four strategies to solve this problem, (1) Data replication strategic, (2) Query replication strategic, (3) Data reply strategic, (4) Data synchronization and update.

In this thesis, we assume the area range is known, and we divided the area into three parts, Inside Area, Border Area and Outside Area, and it shown as Figure 5.

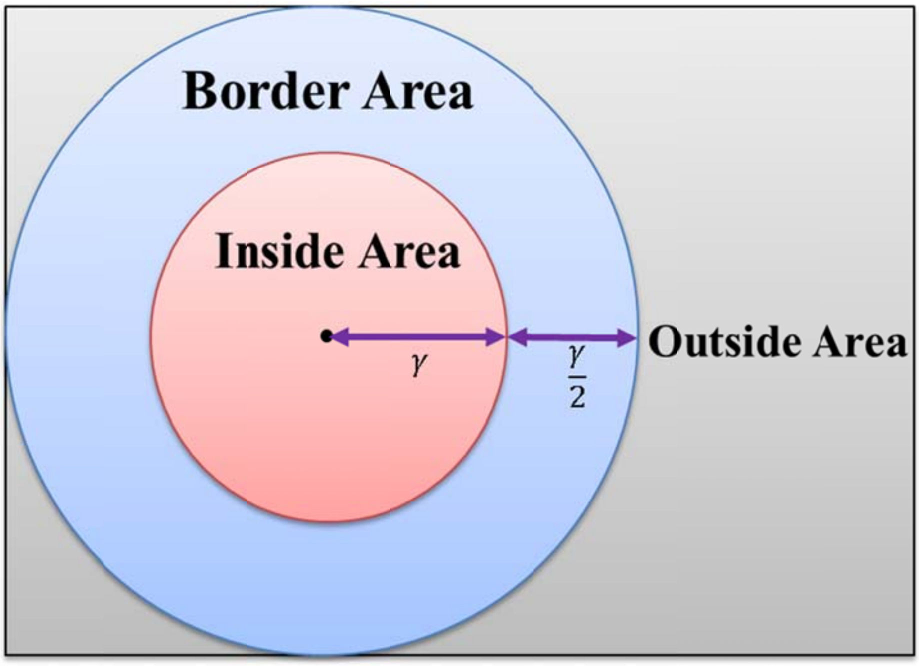


Figure 5: Three parts area figure

Inside Area is within a radius of γ meters from area central, outside the Inside Area $\frac{\gamma}{2}$ meters is Border Area and the other area is Outside Area. This intention is let nodes in different area use different ways to select messages and deliver messages. In order to identify the node position, we refer to the Vincenty's formula[26](Formula 1, detail see the [26]), this formula considered the earth is ellipsoid, and use Longitude and Latitude two points to calculate the shortest distance s, and the deviation can be accurate to 0.5 millimeter. We use the node position point and the Area Central Point (CP) to calculate the distance d, if $d \leq r$, it indicates the node is in the Inside Area, if

$d > r \ and \ d \leq \left(r + \frac{r}{2}\right)$, it indicates the node is in the Border Area, if $d > \left(r + \frac{r}{2}\right)$, it means the node is in the Outside Area.

$$s = bA(\sigma - \Delta\sigma) \quad (1)[26]$$

When two nodes encounter, they will change their metadata first (shown as Table 1), it includes node's position, node's type (Online-node or Offline node), node past moved path point, node encounter nodes, node collected Data message index list, Query message index list and Reply message index list.

Table 1: Metadata format

Category	Value
Position	Longitude, Latitude
TYPE	{0,1}
Path	$\{P_{t-1}, P_{t-2}\}$
Encounter	$\{UID_1, UID_2, \ldots.., UID_n\}$
Data list	$\{DataId_1\_UID_1\_Time_1\_Sync_1, \ldots\ldots, DataId_n\_UID_n\_Time_n\_Sync_n\}$
Query list	$\{QryId_1\_Time_1\_isQueried_1, \ldots., QryId_n\_Time_n\_isQueried_n\}$
Reply list	$\{RepId_1\_Time_1, RepId_2\_Time_2, \ldots.., RepId_n\_Time_n\}$

When two nodes encountered, we call ourself as Node A, encounter node as Node B, the Location-based Content Search approach (LCS) workflow is shown as Figure 6, and the transmission protocol is shown as Figure 7. First, Node A and Node B always detect connection, when encountering, Node A will send a metadata request to Node B, after receiving the request, Node B returned its metadata to Node A, when Node A received the metadata from Node B, the first step is check and filer the identical messages from Node A's database, then keep the message which Node B doesn't have, and rebuilt that metadata. The next step is Message Selection, it selects the messages which can be sent to Node B. There are three strategies step we proposed to select the messages, (1) Data replication strategic, Node A selected the Data messages according to Node A's position. The detail description is in Section 3.1. (2) Data Reply strategic, get the Query message from metadata, if Node A's database have the queried data message, Node A will generate a Reply message. The detail description is in Section 3.2. (3) Query replication strategic, this strategic must be handled after Data Reply strategic complete, and the queries which queried data shouldn't send out anymore. The detail description is in Section 3.3. After completing three message selection steps, Node A has chosen the messages which will be sent to Node B. The next step, Node A will check the Data messages which doesn't synchronize to the remote server. If Node A is an Online-node, it will sync this Data messages to the server, if not, Node A doesn't do anything, and the detail will be described in Section 3.4. Before sending messages to Node B, Node A has to schedule to messages which will be sent first. Because of the contact time of two nodes are short, it couldn't send all the messages to the other nodes. Therefore, how send the messages to the encounter node efficiency, it is shown in Section 3.5. Then, Node A sends the selection message queue to Node B. As Node B receiving, it will check the Data messages whether it sync to server or not. If Node B is an Online-node, then sync the message to server, if not, skip this action, and then finish this contact.

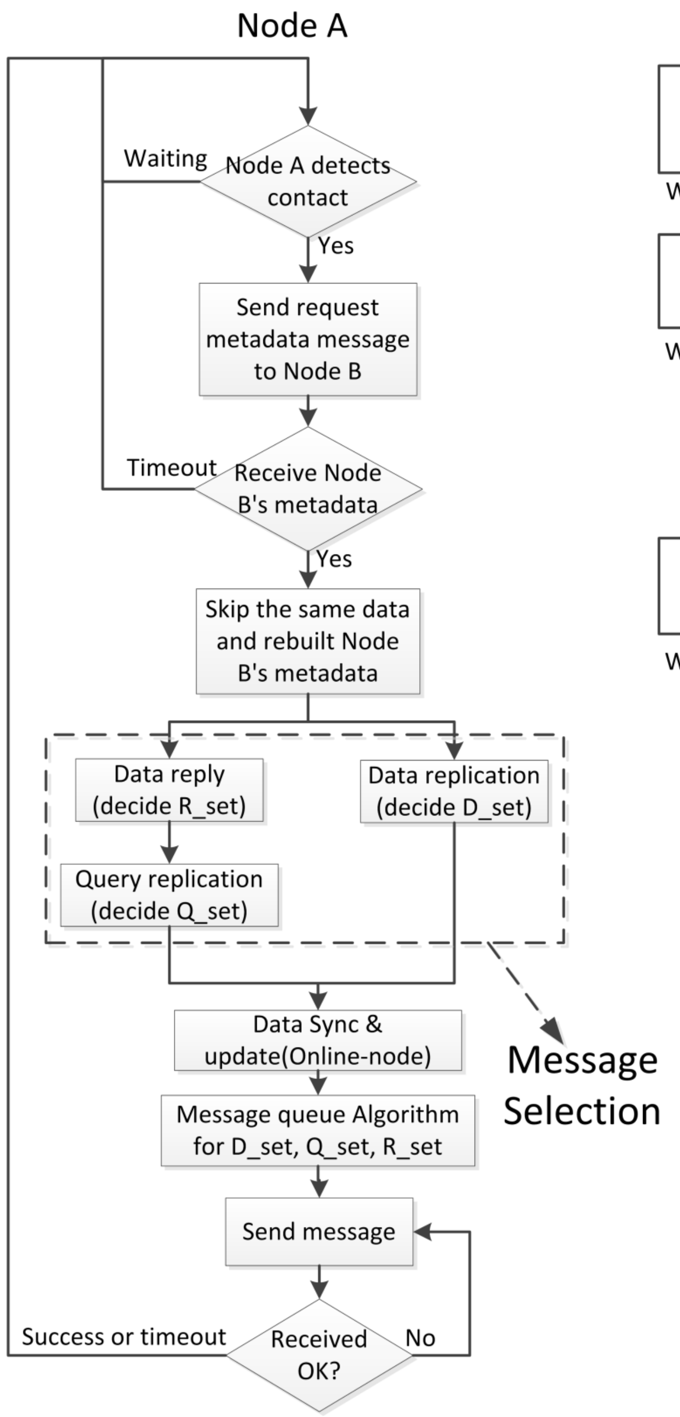


Figure 6: LCS workflow

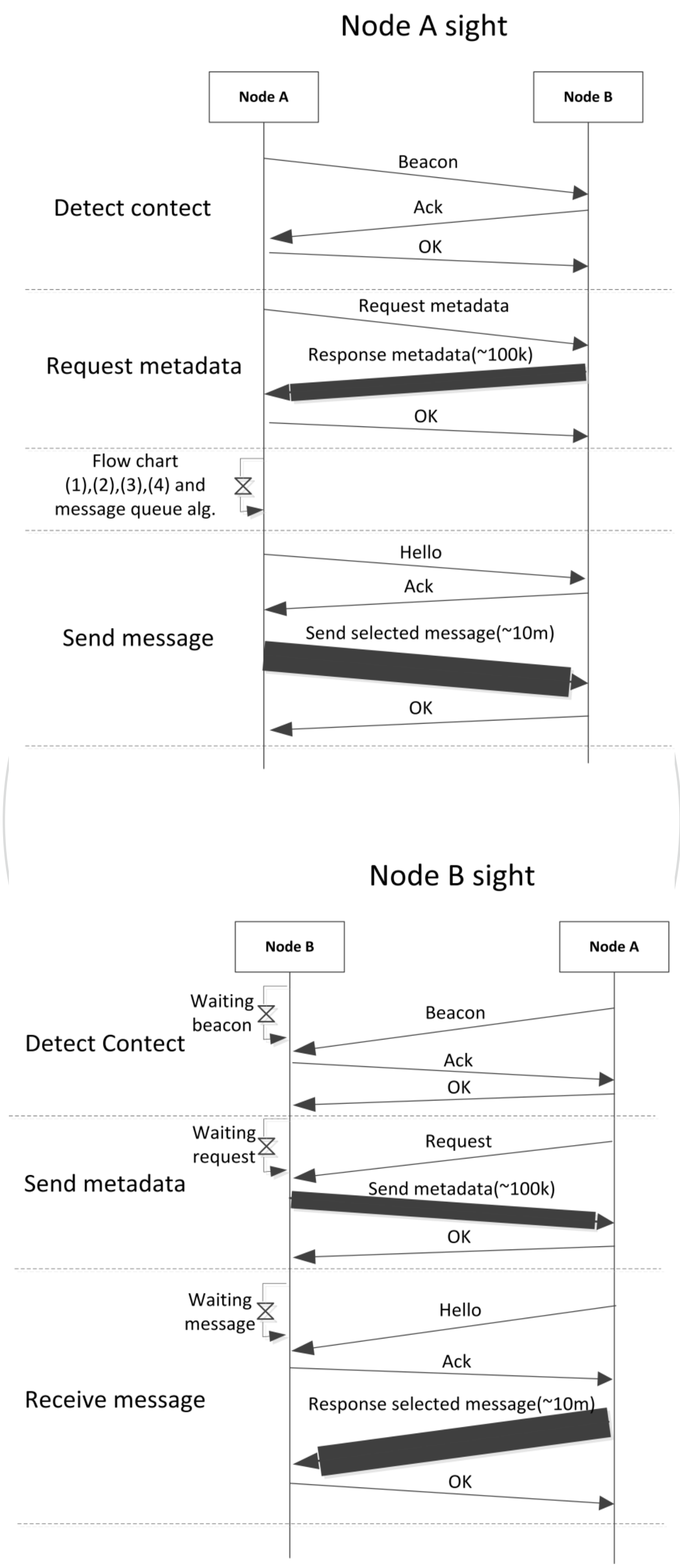


Figure 7: LCS transmission protocol

3.1 Data replication strategic

When two nodes encountered in the map, the processing step can be seen in Figure 8, and the selection rule is shown in Formula 2, (1) If the distance d between Node A and Area Central Point (CP) is less than or equal to radius γ, it indicates Node A is in the Inside Area. When Node A encountered Node B, it will add all the Data messages into the Data dataset *D_set* to prepare to send to the Node B. The main purpose is let the Inside Area fulfill with related Data messages. If any node wondered to query in this area, it can get the Data messages quickly and has high query success rate. But the encounter time of the two nodes and the transmission are limited, two nodes have to change itself metadata before transport. Then, Node A skips the identical Data messages, and adds the others into *D_set*; (2) If the distance d between Node A and CP is greater than radius γ and less than $\gamma + \frac{\gamma}{2}$, it indicates Node A is in the Border Area. According to Node B's past path, if $\Delta N_b.d < 0$, it indicates Node B will likely enter to the Inside Area, then, Node A add the Data messages into the *D_set*; (3) if the distance d between Node A and CP is greater than $\gamma + \frac{\gamma}{2}$, it indicates Node A is in the Outside Area, then, Node A shouldn't do anything.

We use $\Delta N_b.d$ to predict whether Node B's will enter to the Inside Area or not. In Formula 3, $N_b.d_{t-1}$ is the distance between Node B position P_{t-1} and CP at time t-1, and $N_b.d_{t-2}$ is the distance between Node B position P_{t-2} and CP at time t-2. Then, subtract $N_b.d_{t-1}$ from $N_b.d_{t-2}$, if the result is less than 0, it means Node B is likely to enter to the Inside Area, if the result is granter then 0, it means Node B is likely to move away from the Inside Area.

$$add(data, D\_set) = \begin{cases} True\ , & if\ N_a.d \leq r \\ True\ , & if\ N_a.d > r\ \ and\ \ N_a.d \leq \left(r + \frac{r}{2}\right) and\ \Delta N_b.d < 0 \\ False, & otherwise \end{cases} \quad (2)$$

$$\Delta N_b.d = (N_b.d_{t-1} - N_b.d_{t-2}) \quad (3)$$

In order to avoid data overflow, we set all the Data messages a storing time TTS (Time-to-Store) (Formula 4). TTS means the time of any message be created or replicated then store in some node. When TTS expired, the messages will drop anymore. Therefore, it can reduce the node loading.

$$drop(data) = \begin{cases} True\ , & if\ Time\_to\_store\ expire \\ False, & otherwise \end{cases} \quad (4)$$

Otherwise, if Node B is an Online-node, Node A doesn't need to send Data messages to Node B. Because the Online-node can connect to the Internet to get the messages by itself, it doesn't require wasting the resource to send messages.

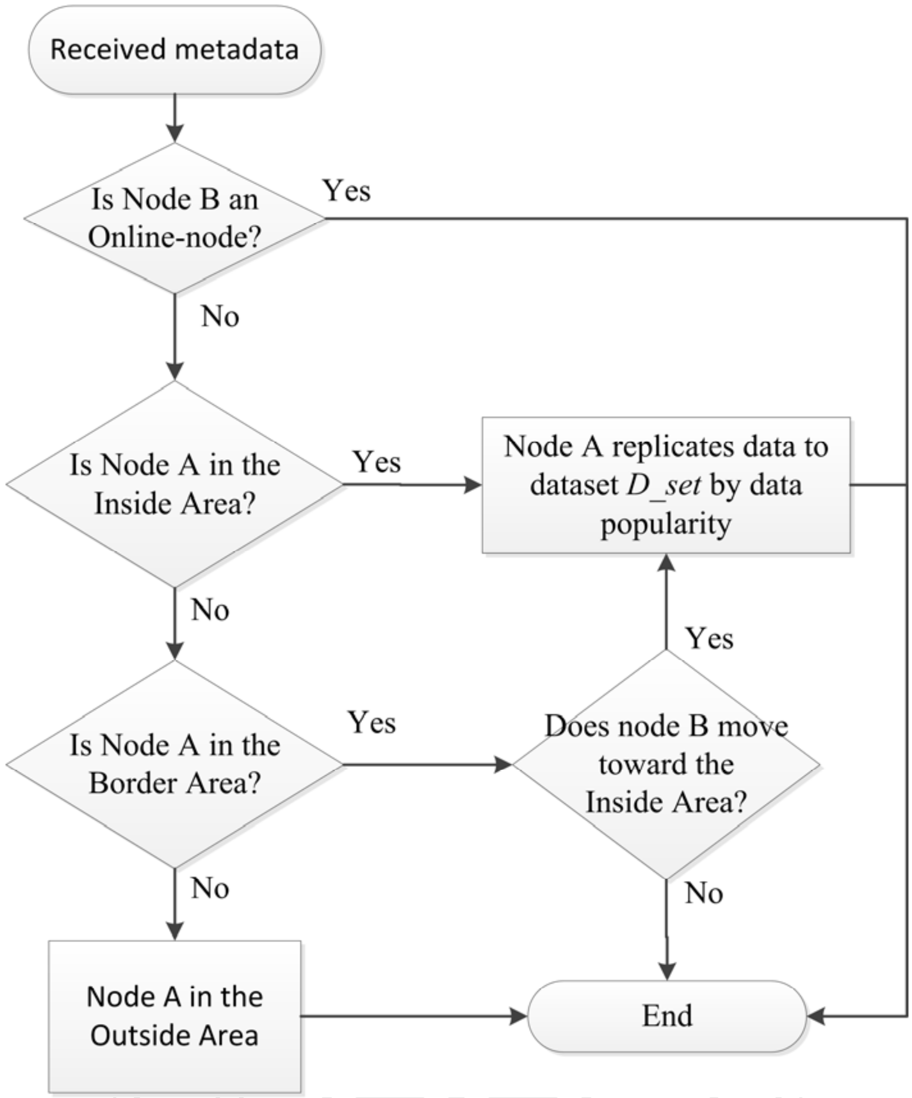


Figure 8: Data replication strategic flow chart

3.2 Query replication strategic

This strategic has to do after the Data reply strategic, and the goal of this strategic is let user queried data in a shortest time. This is a three-tier Query strategic, from Node A's local database to Node B's local database then remote server. Query replication detail rule can see in Formula 5 and Figure 9 flow chart. When Node A receiving query messages, it has to check the database whether the query messages have the match data. If has, it doesn't need to do Query replication, if not, there are some rules to do. (1) Node A is an Online-node. It does nothing. (2) Node A is an

Offline-node. When Node A in the Inside Area, and then adds all the related Query messages to the Query dataset *Q_set*; When Node A in the Border Area, then Node A has to predict Node B's direction ($\Delta N_b.d$, the detail can see in Section 3.1, and also is shown in Formula 3), if $\Delta N_b.d$ less than 0, it indicates Node B is likely to enter to the Inside Area, then, add the related Query messages to the *Q_set*. If not, then do nothing. The main purpose is Data replication strategic will centralize the related Data messages in the Inside Area. Therefore, we send the queries to the Inside Area, and there is more opportunity to query success. It has two reasons. One, There are many related messages in the Inside Area. The other, if any node leave from Inside Area, we expect it could carry much more related messages; When Node A in the Outside Area, it shouldn't do anything.

In addition, if the query represents it had been queried ($query.isQueried = true$), then we don't have to add this query into the *Q_set*.

$$add(query, Q\_set) = \begin{cases} True, & if\ N_a.d \le r \\ True, & if\ N_a.d > r\ and\ N_a.d \le \left(r + \frac{r}{2}\right)\ and\ \Delta N_b.d < 0 \\ False, & if\ Database(query) = true \\ False, & if\ query.isQueried = true \\ False, & otherwise \end{cases} \quad (5)$$

However, if we don't have any limitation of replicate query message, it will keep replicating in spite of it has queried. Therefore, we need a mechanism to stop it. Every Query messages set a TTS, if the TTS expired (Formula 6), and then drop it. This way will inhibit query replication. There are some defects in the way. If TTS is too short, it would query data hardly. If TTS is too long, it would still create many query copies. So we add one more rule. If query message find the match data, this query message's attribute *isQueried* will be change to TRUE. And next time, nodes encountered, they change their metadata, then nodes will know what query message had been queried, and it will skip it to add to Q_set.

$$drop(query) = \begin{cases} True, & if\ Time\_to\_store\ expire \\ False, & otherwise \end{cases} \tag{6}$$

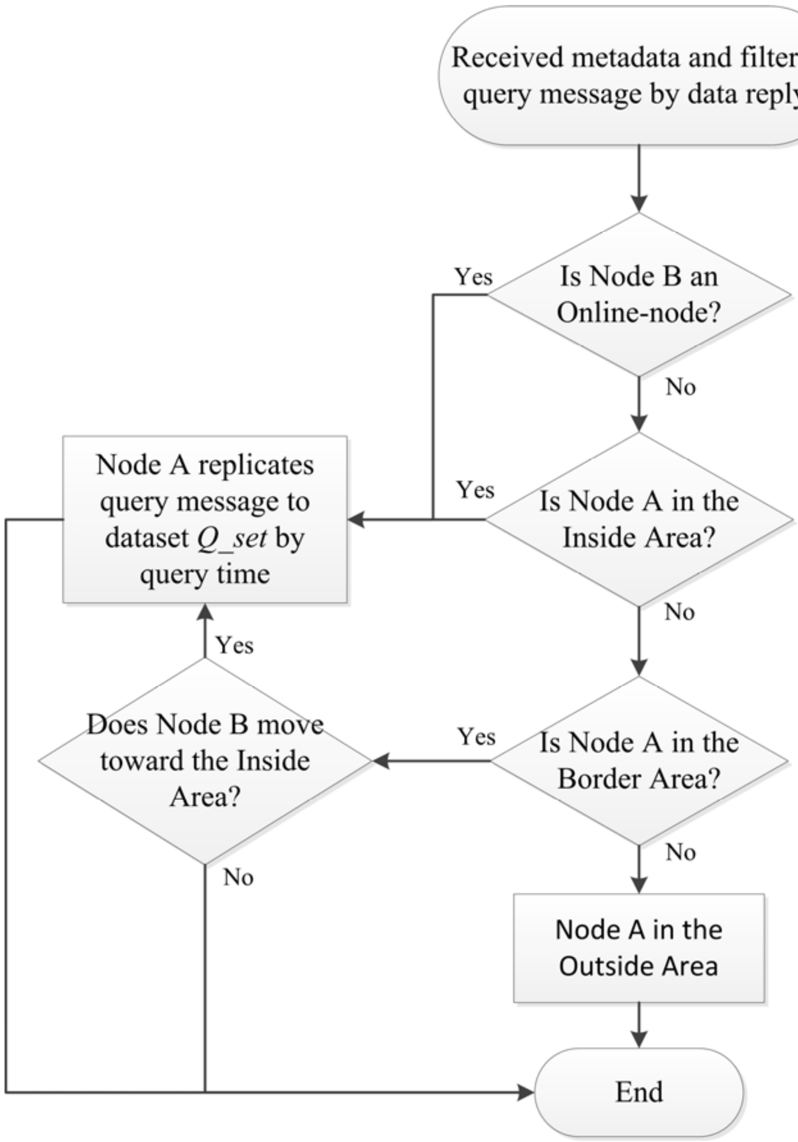


Figure 9: Query replication strategic flow chart

3.3 Data Reply strategic

As shown in Figure 11. Node A received Node B's metadata, it will check all query messages of Node B's metadata. If Node A has the match data, it will generate a Reply data *R_data* of the Query message and change the Query message attribute *isQueried* to TRUE, then we can call Node A as Replier.

It is difficult to send the Reply data to the Querier. Because all nodes in the network are keeping moving, we can't use the original packet delivery path to send back to the Querier. Therefore, we have to use the other rule to send Reply data.

Replier (Node A) can know the Querier's location from Query message, and send the Reply data to that location. But, as time goes by, Querier may not still stay there. So, when Replier (Node A) encounters other node (Node B), it will check the Encounter Table of Node B whether it had encountered Querier or not. If it had, we will check the encounter time whose close the recent time. If Node B's encounter time close, we will update (Formula 7) the Querier position of Replier's (Node A) metadata.

$$update(meta.querier_{loc}) = \begin{cases} True\,, & if\ (Querier\ \in N_{b_{ENs\_table}})\ and\ (N_b.querier_{time} > N_a.querier_{time}) \\ False, & otherwise \end{cases} \quad (7)$$

Then we have to distinguish which Node will move toward to the Querier, see the Formula 8, if Replier (Node A) won't move toward to the Querier but Node B will do, it would add the *R_data* to the Reply dataset *R_set*. Otherwise, Replier (Node A) will move toward to the Querier but Node B won't, it just keeps the *R_data* and does nothing.

$$add(R\_data, R\_set)$$

$$= \begin{cases} True\,, & if\ N_a.to(querier) = false\ \ and\ N_b.to(querier) = true \\ False, & if\ N_a.to(querier) = true\ \ and\ N_b.to(querier) = false \\ True\,, & if\ two\ nodes\ toward\ to\ querier\ \&\ (D_a > D_b) \\ False, & if\ two\ nodes\ toward\ to\ querier\ \&\ (D_a > D_b) \\ Fasle, & if\ two\ nodes\ don't\ toward\ to\ querier \end{cases} \tag{8}$$

If Replier (Node A) and Node B are moving the same direction and forward to the Querier, we have to calculate whose moving path will close to the Querier. In Figure 10, we can use the position P_{now}, the past position P_{t-1} to determine this node's moving direction, then we add the Querier position P_q and use the basic Triangle Area formula (Formula 9) to calculate the area value $\mathbb{A}$, as we know the distance between P_{now} and P_{t-1}, we can get the close Querier distance D. If D_a greater than D_b, it indicates Node B will move close to the Querier, then we add the *R_data* to the *R_set* and waiting for transmission. Otherwise, If D_a less than D_b, it indicates Replier (Node A) will move close to it, then we just keep the *R_data* and moving.

$$\mathbb{A} = \frac{1}{2}\begin{vmatrix} x_1 & x_2 & x_3 & x_1 \\ y_1 & y_2 & y_3 & y_1 \end{vmatrix} \tag{9}$$

However, we could not replicate messages unlimited, so we have to make some rule to inhibit it. All the Reply data will set a TTS, when the TTS expired or *R_data* had forwarded to the querier , this message *R_data* will drop anymore (Formula 10).

$$drop(R\_data) = \begin{cases} True\,, & if\ Time\_to\_store\ expire \\ True\,, & if\ R\_data\ had\ forwarded\ to\ the\ querier \\ False, & otherwise \end{cases} \tag{10}$$

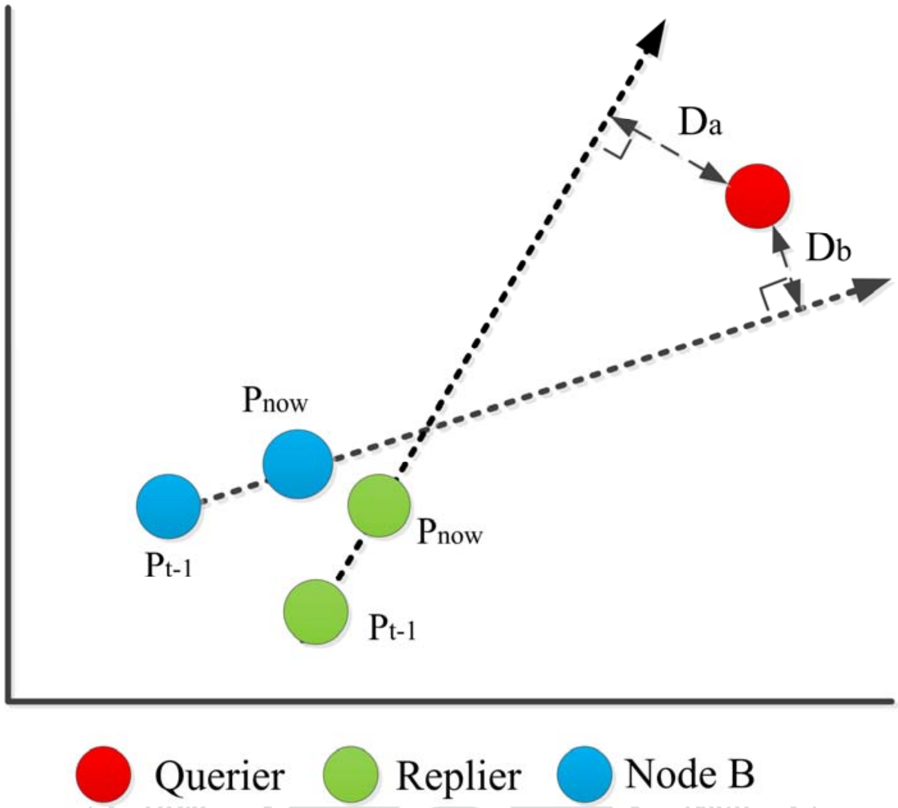


Figure 10: Moving direction and the relative distance

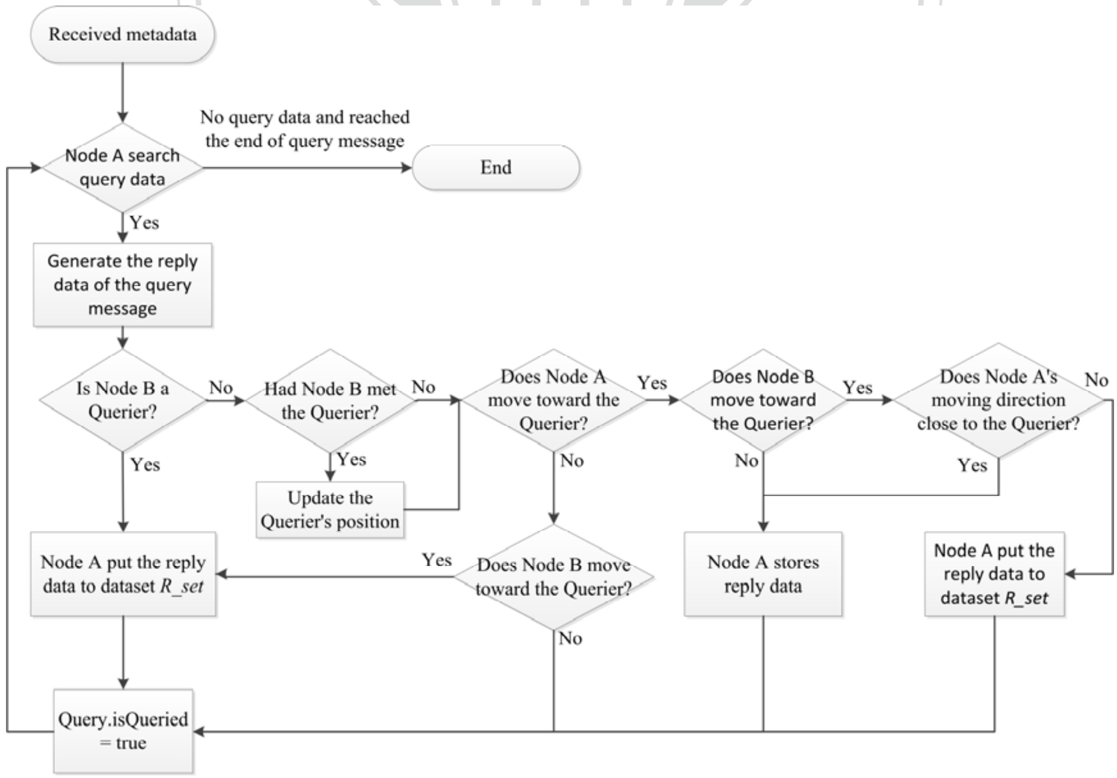


Figure 11: Data reply strategic flow chart

3.4 Data synchronization and update strategic

Whatever Online-node or Offline-node, it would create data messages. Beside keeping the data in local database, it has to upload this data messages to the remote server to share this information. Figure 12 shows this strategic. When node received messages, the first step is distinguishing the type of node, if it is Online-node, it should upload all the data messages which doesn’t sync to the remote server directly. Then change the Sync attribute of this data message to TRUE. Otherwise, if it is Offline-node, it shouldn’t do anything.

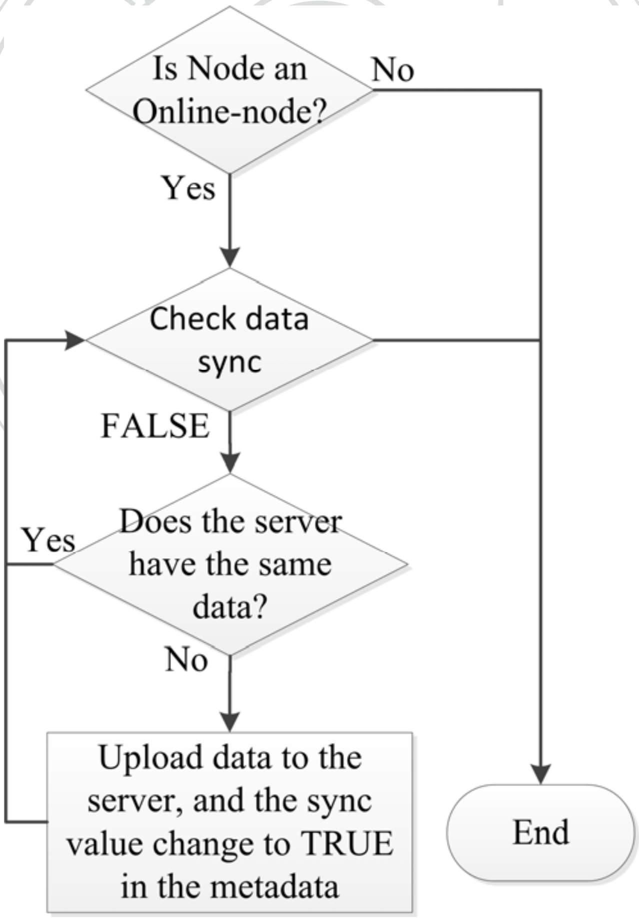


Figure 12: Data sync and update strategic flow chart

3.5 Message queue selection algorithm

When all the messages are chosen (D_set, Q_set and R_set), Node A has to send these messages to Node B. But the messages cannot be parallel transferred. Therefore, we have to decide which one will be sent first. In the thesis, shown in Figure 13, we proposed a simple message queue selection algorithm, within the weight value *d*, *q* and *r*; and progressive value *S* to calculate and get the transmission Message queue *M*.

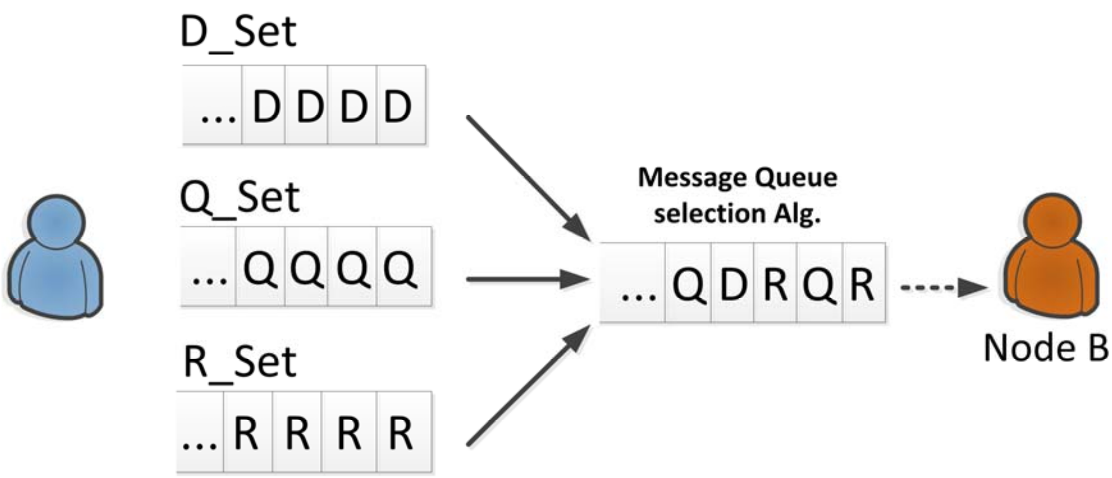


Figure 13: Message Queue Selection

We can get the dataset D_Set, Q_Set and R_Set from all the replication strategies we describe above. Then we give the dataset a different weight value *d*, *q*, *r* and $d + q + r = 1$. The first step, we find the dataset of highest weight value *MaxSet*, then, the *MaxSet* dataset pop a message and enqueue to *M*. This will be the first transmission message, then, we reset the weight value of *MaxSet* dataset, and add the Progressive value *S* to the other weight. Repeat the steps until all messages enqueue to M, and then we can get a complete Message queue M. Finally, Node A according to M sends all the messages to Node B.

The beginning results of this selection queue are similar to Weight Fair Queue

(WFQ), the highest weight of messages will be sent more. After several times, the message will be selected by Round Robin (RR), because of the weight value and progressive value.

Message queue selection algorithm

Input: Data dataset *D_set*, Query dataset *Q_set*, Reply dataset *R_set*, Data weight *d*, Query weight *q*, Reply weight *r*, Progressive value *S*

Output: Message queue *M*

```
While((size(D_set) + size(Q_set) + size(R_set)) != 0)
MaxSet = Max(d,q,r).getSet()
M.enqueue(popMsg(MaxSet))
Switch(MaxSet)
Case D_set: d = d.original, q = q + S, r = r + S
Case Q_set: d = d + S, q = q.original, r = r + S
Case R_set: d = d + S, q = q + S, r = r.original
End switch
If size(MaxSet) = 0 then MaxSet.weight = 0
End while
Return M
```

CHAPTER 4

Simulation Results

In this chapter, we discuss the simulation results, and we compare with three protocols. LCS (we proposed), Locus [11] and Epidemic [3]. We make some variation in Epidemic, because of Epidemic doesn't matter the query-reply situation. We set the area concept and let nodes having query scenario and reply action in Epidemic.

Then, in the results, we will show some validation parameters. And we explain how the equation be calculated below.

(1) Query-Reply Success Ratio: The equation is shown below. The number of success reply message means all the reply messages success delivery from Replier to Querier, and the same data we won't be repeatable recorded. And the number of query message means how many numbers the query message be generated, so the replicate query messages we won't record.

(2) Query Delivery Ratio: The equation is shown below. The number of query delivered means the all query messages which delivered to the data source node and the number of query message described above.

(3) Overhead: The messages which are redundant transferred are overhead. Success reply message means the first message reply to the Querier and the same messages replied afterword do not include. Query delivered means the first message delivered to the Replier, and as above, the same messages don't include.

(4) Latency: We calculate from the time of send the first query message by the Querier to the time of receive the first reply message by the Replier. And the results we get the medium number to analyze.

- Query-Reply Success Ratio = $\frac{number\ of\ success\ reply\ message}{number\ of\ query\ message}$
- Query Delivery Ratio = $\frac{number\ of\ query\ delivered}{number\ of\ query\ message}$
- Overhead = $\frac{\#relay\ message\ -\ \#success\ reply\ message\ -\ \#query\ delivered}{\#success\ reply\ message\ +\ \#query\ delivered}$
- Latency = $\text{Time (Received first reply message)} - \text{Time (sent first query message)}$

4.1 Simulation Setup

We use ONE(Opportunistic Network Environment simulator) [27](shown as Figure 15) and the map of Taipei101 surrounding area (Figure 14) to validate our approach. All nodes are pedestrian, and we set a random destination point to them. When they reached the destination, then we give them the other destination. And all nodes move to the position by shortest path algorithm.

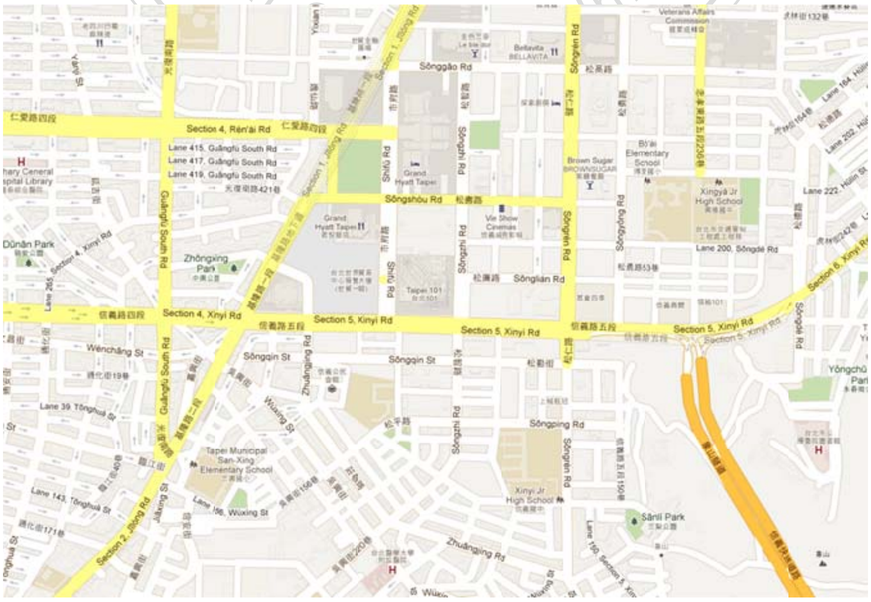


Figure 14: Map of Taipei101 surrounding area

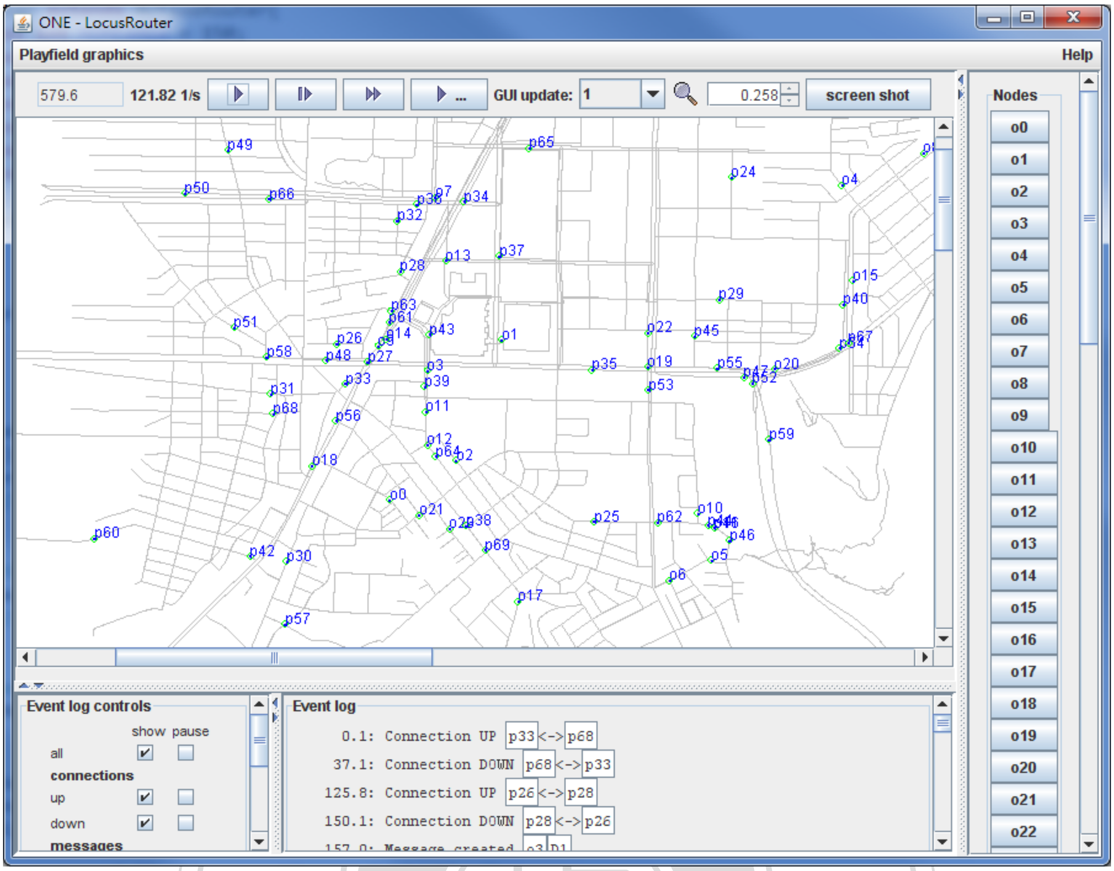


Figure 15: ONE simulator

4.2 Simulation Settings

The simulation setting is shown as Table 2. The map area is 3000m * 2500m, and the simulation time is 43200 seconds, it is equivalent to 12 hours. In order to initialize data sources, we have to set a warm up time 1000 seconds. The node data transmission rate is 2Mbps, and the transmission rage is 10m. All nodes divided to types, 35% is Online-node and 65% is Offline-node. The message size of Data message and Reply message are 500KB~1MB, and the Query message is 50KB~100KB. All the messages have a creation interval, the Data message is 120~180 seconds, and the Query message is 200~400 seconds. The node buffer size is 500MB. All nodes are pedestrian, so the moving speed is 1.8km~5.4km per hour. And the messages' TTS is 18000 seconds, it is equivalent to 5 hours.

Table 2: Simulation Settings

Area	**3000m x 2500m**
Simulation Times	43200 Sec
Warm Up Time	1000 Sec
Data Rate	2Mbps
Radio Range	10m
Online-node : Offline-node	35% : 65%
Message Size	
Data/Reply message	500K~1MB
Query message	50K~100K
Interval of message creation	
Data message	120-180 Sec
Query message (offline-node only)	200-400 Sec
Buffer size	300MB
Velocity	1.8~5.4 km/h
Time-To-Store(TTS)	18000 sec

4.3 Simulation Results

Before comparing to other approaches, we analyze some parameters to our approach. First we will discuss the effect of the percentage of node type and the radius of Inside Area. Then, we will compare with other approaches to analyze the performance. And we will discuss below.

4.3.1 The percentage of node type

We simulate four node type scenarios. All the Online-node can't query data, because we assume they can query from the Internet directly. They just create or reply data. We can see Figure 16 Query-Reply success ratio. If there are no Online-nodes, it get low success ratio. Because all nodes are Offline-node, they can't know all the data messages.

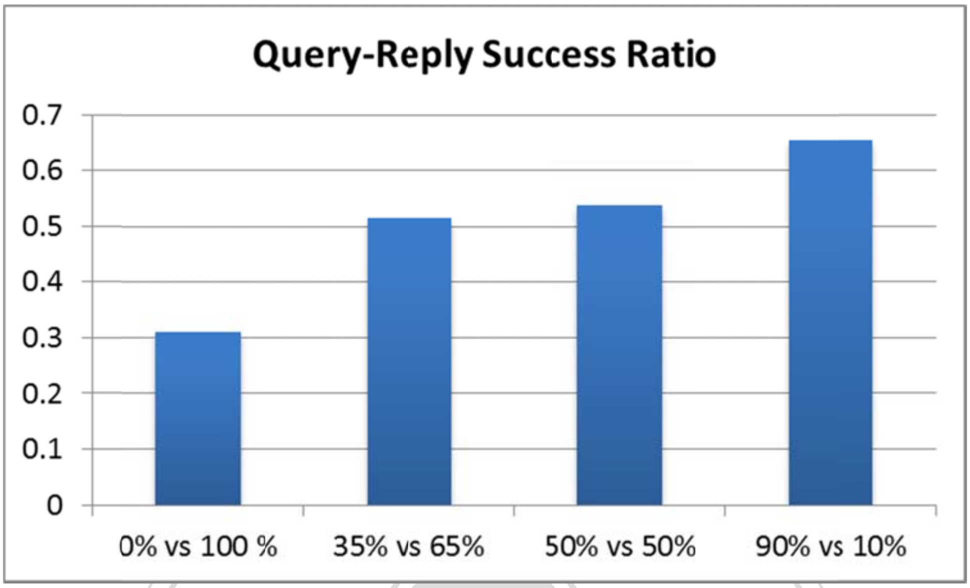


Figure 16: Query-Reply Success Ratio (Node Type)

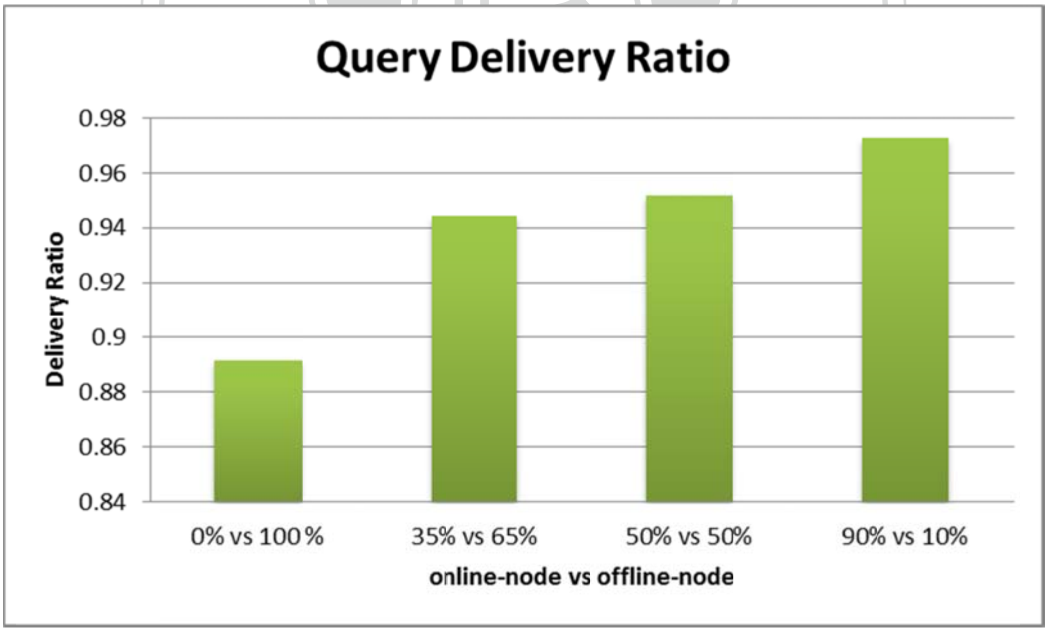


Figure 17: Query Delivery Ratio (Node Type)

Thus, they cannot get data easily. They have to spend more time (see Figure 19) and more replicate (see Figure 18) to find the data. And we can see Figure 17 to validate our opinion, in the first case (0% vs. 100%), it get the worst query delivery ratio. However, if the network full of Online-node, it means they don't need any more replication (low overhead), because it is easily to encounter Online-node, and they can directly ask them for data. So, they get the data quickly (low latency). Therefore, the fourth case (90% vs. 10%) is the best results of all the charts.

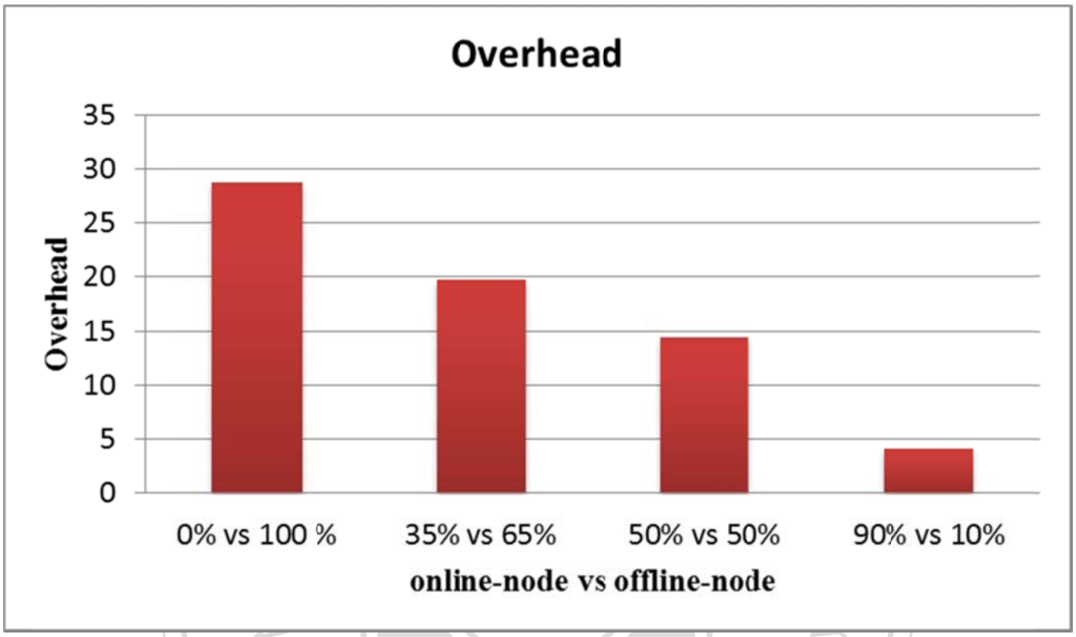


Figure 18: Overhead (Node Type)

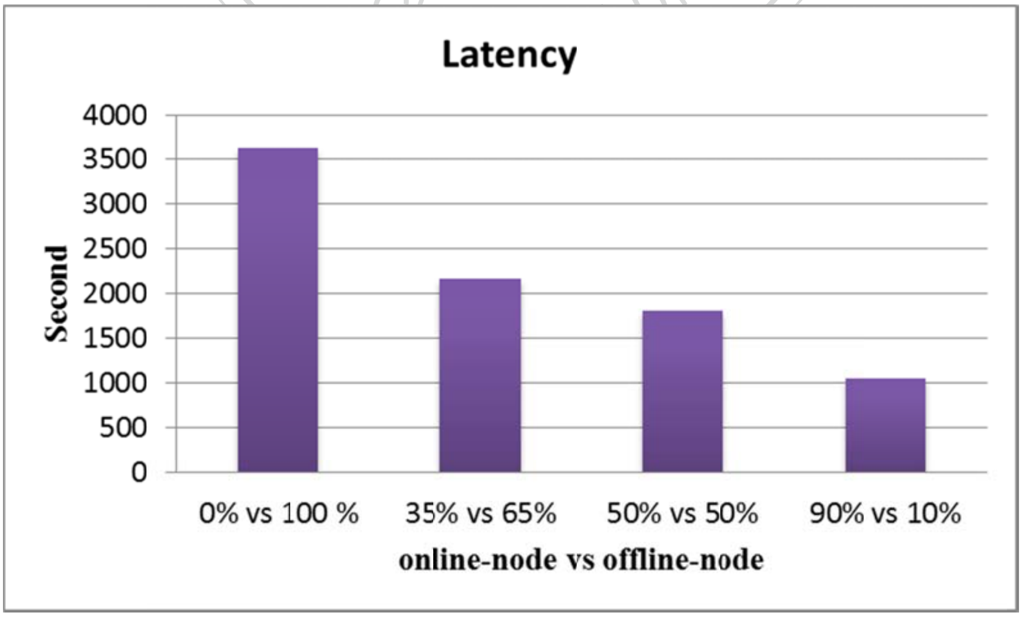


Figure 19: Latency (Node Type)

4.3.2 Radius of Inside Area

We show the results of radius changing. In Figure 20, it is the Query-Reply success ratio, and we can see the small radius performed a bad result. Because our approach doesn't matter the outside area, if the radius too small, there are a few nodes in the Inside Area, and then the related data doesn't be centralized in the area. And the big radius performed badly, too. Because the large Inside Area will allow more message replication, and the data will be more complicated and be spread around this big area, then, it is more difficult to look up data. So, the suitable radius is between 300 and 500.

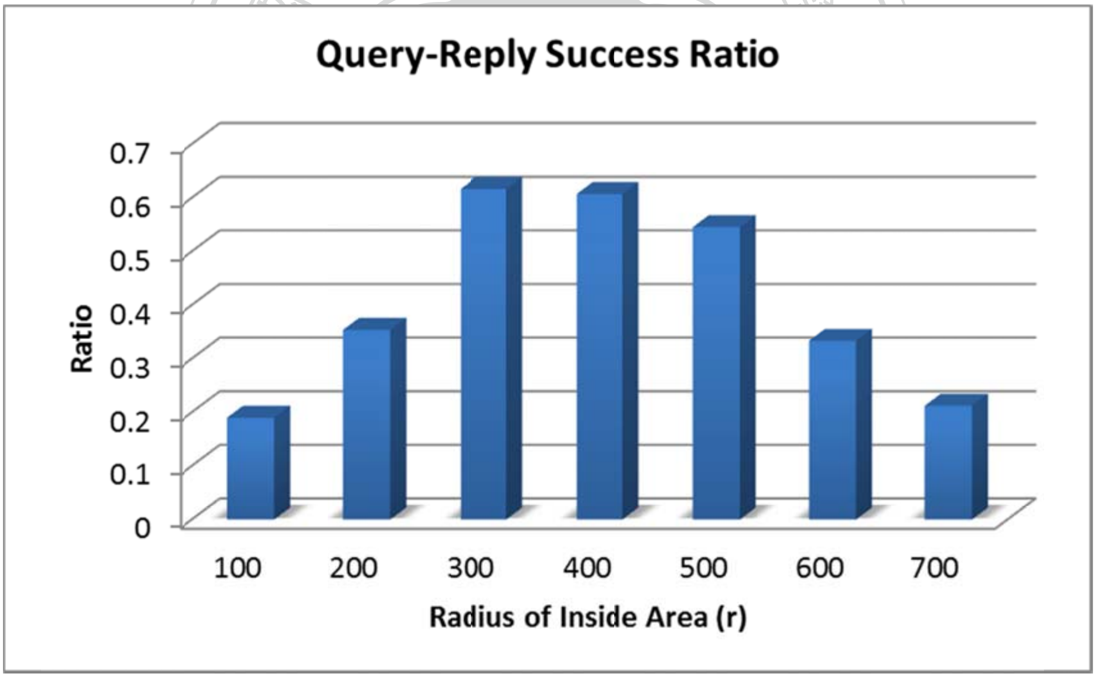


Figure 20: Query-Reply Success Ratio (Radius of Inside Area)

As we discuss above, there are a few nodes in the small radius of inside area, so, it doesn't have much replication activity, if any node would like to query in this area, it will spend much time to search the data source. Then, in Figure 21, it performed lower overhead in the small radius. On the other hand, in the big radius, it got higher overhead. And in Figure 22, nodes spent much time to query in the small radius, and in the big radius, nodes would look up quickly. Otherwise, the selected results base on the environment settings. If there are different transmission ranges, buffer size, node speed, etc., we might get the different suitable radius. The radius 500m is the suitable size in our scenario.

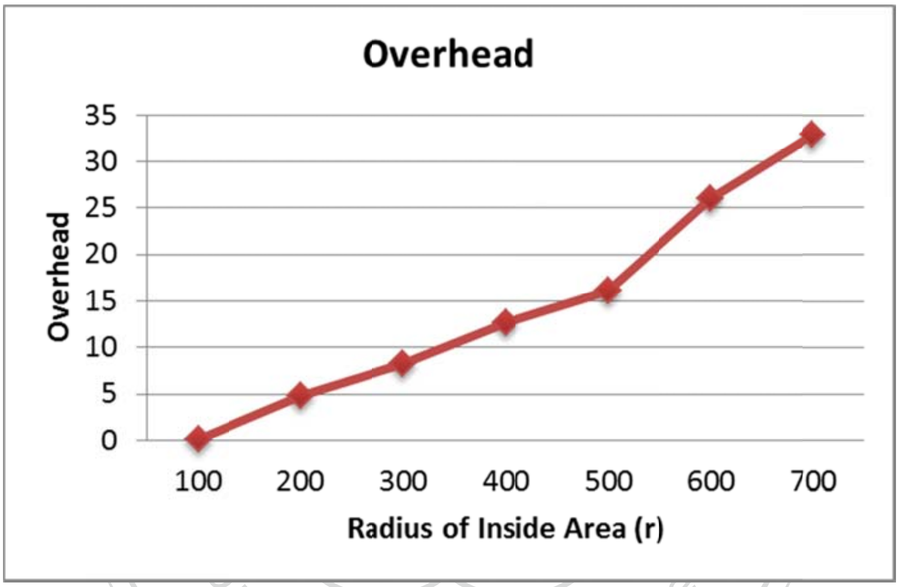


Figure 21: Overhead (Radius of Inside Area)

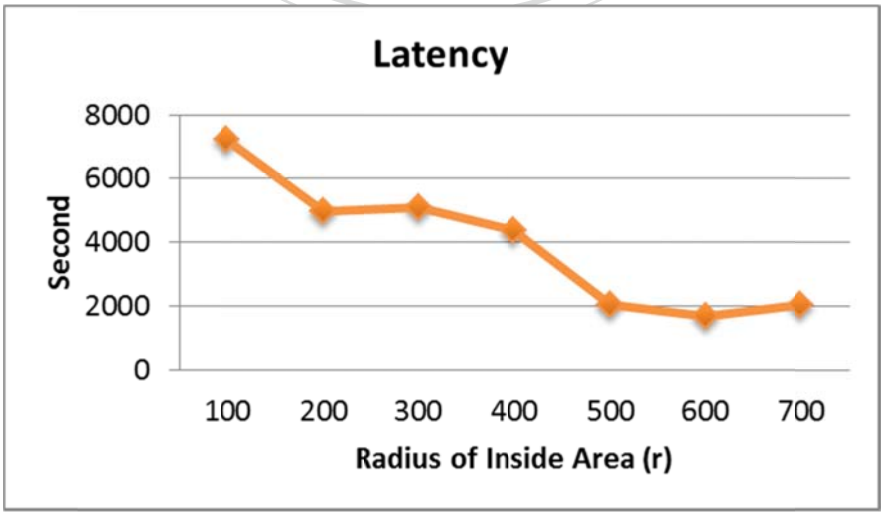


Figure 22: Latency (Radius of Inside Area)

4.3.3 Node Density

In the section, we analyze the results with other approaches, Locus and Epidemic routing schemes. And we want to see what the importance part of our approach is, so we make two changing in our approach. One is we don't matter the Data replication strategic and we only use the Epidemic routing to spread the data messages, and the other is we don't care the Query replication strategic, just use Epidemic to disseminate the query messages.

In Figure 23, we can see our approach and Locus have a similar performance in query-reply success ratio, because both of we are using the region concept to centralize the messages. But, as the number of nodes increase, Locus will perform well than ours. Because our region concept base on local area, if there are many nodes in this area, it will have many messages (data, query, reply) in every node. Although nodes have rich data source, the transmission rate is fixed, and nodes intermittent connected, they can't send all the messages. So, some messages might be ignore, and the performance isn't well. And we want to see the importance part of our approach, we modify it in two types. (1) We use epidemic routing to replace the data replication strategic, and (2) we use epidemic routing to replace the query replication. The result shows the (1) type got worse performance. Because the data messages couldn't be centralized to the inside area, the messages would spread to whole network. It is difficult to query unique data message in the network.

Otherwise, in order to compare fairly, we present a scenario with only Offline-node in our approach. Although the success ratio is worse than Locus, but the overhead and latency are better than Locus. And the result will discuss below.

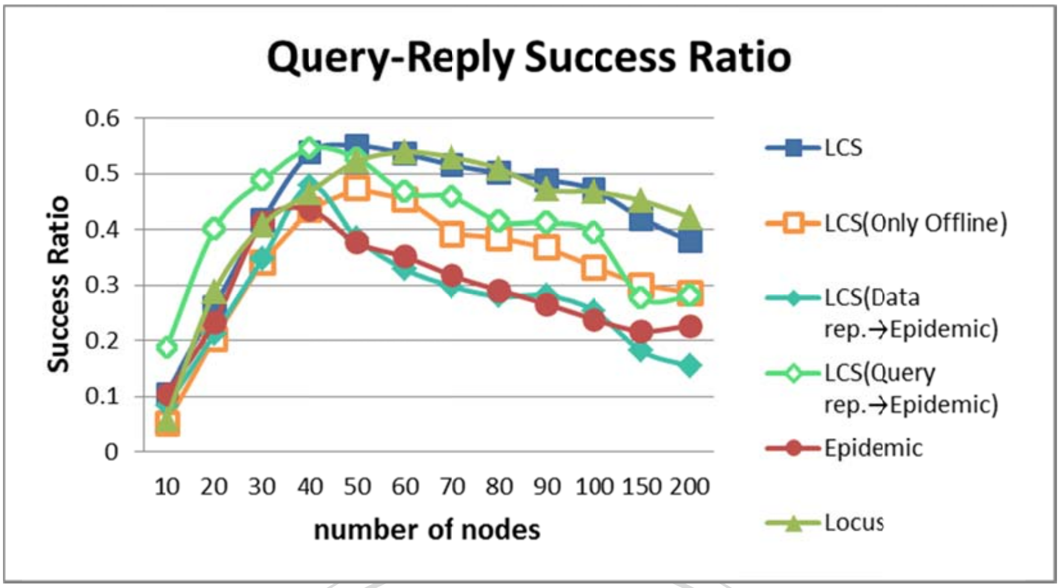


Figure 23: Query-Reply Success Ratio (Node Density)

And we can see the Figure 24 and Table 3, in our approach, there are 71% of the nodes success searching data in Inside Area. It means nodes in the Inside Area will have high probability to query success.

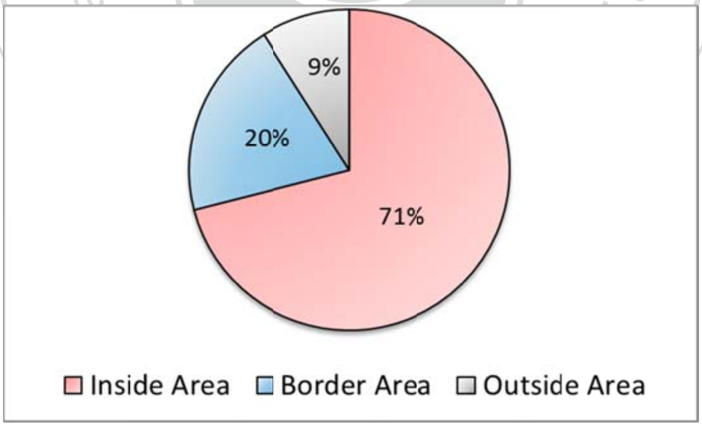


Figure 24: Node Distribution of Success Searching

Table 3: Detail number of Node Distribution in Success Searching

Node Density	30	40	50	60	70	80	90	100	150	200
Inside Area	15	15	22	23	25	23	23	18	17	13
Border Area	3	7	7	6	6	4	8	6	5	4
Outside Area	0	5	3	1	3	0	3	2	3	5

Figure 25 shows the query delivery ratio, we define the query delivery as Querier sent the query message, and it delivery to the Replier who has the data source. We can see the figure, all the approaches have high delivery ratio, because data message doesn't just one copy spread to the network, and the query message does, either. In all the query messages, we just need one query message to find the data source, then, it is success delivery. Therefore, it is easy to find the Replier, but how to reply this data to the Querier is the most difficult.

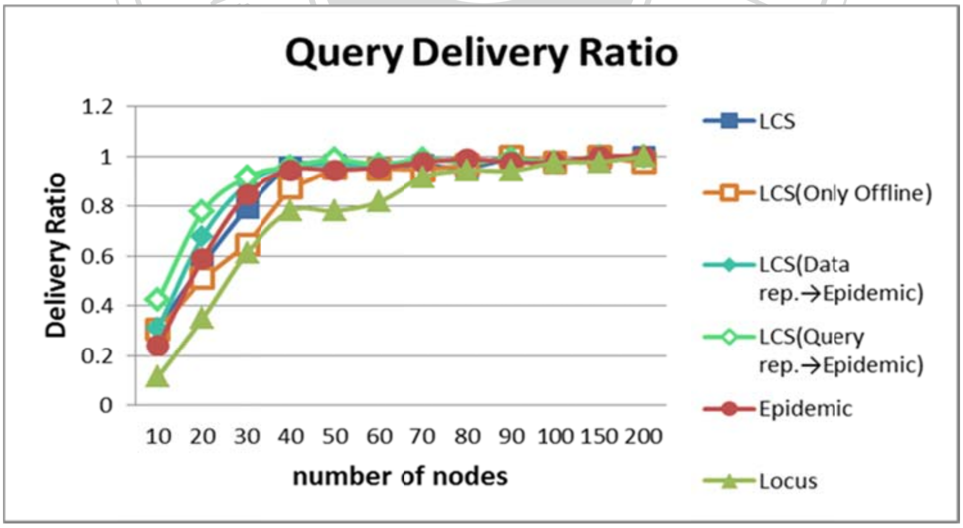


Figure 25: Query Delivery Ratio (Node Density)

Figure 26 is overhead, and we can see the result of Locus, although the query success ratio of Locus is higher than LCS, but its overhead is larger than LCS.

Because Locus centralizes the same data in the same area, if Querier closed to this area, it can use a few query replicate to look up data quickly. But, if Querier far away from this area, it has to replicate more message in delivery. So the overhead will be higher. And LCS is a wide range of the local area, it spreads all the messages in the area, thus, LCS has higher opportunity to spend lower cost to get the data.

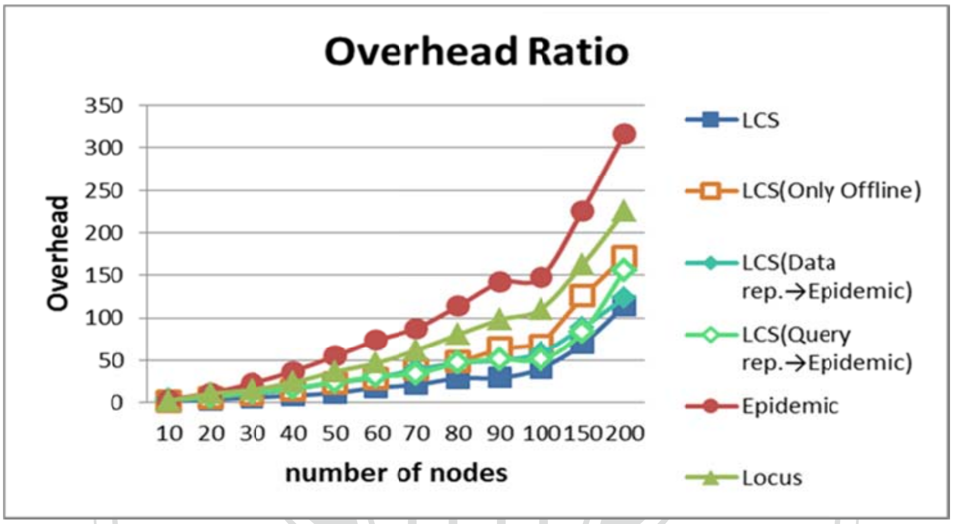


Figure 26: Overhead (Node Density)

Figure 27 is Query Latency. We can see the result of our approach LCS is much lower than Locus and Epidemic. Because all the data will be spread in the area, if node in the Inside Area or Border Area, it will quickly get the data. Although Locus is used the area concept, too. They will centralize the same data in the area, if nodes want to query that data, but they are not near the area, it will spend much time to query data.

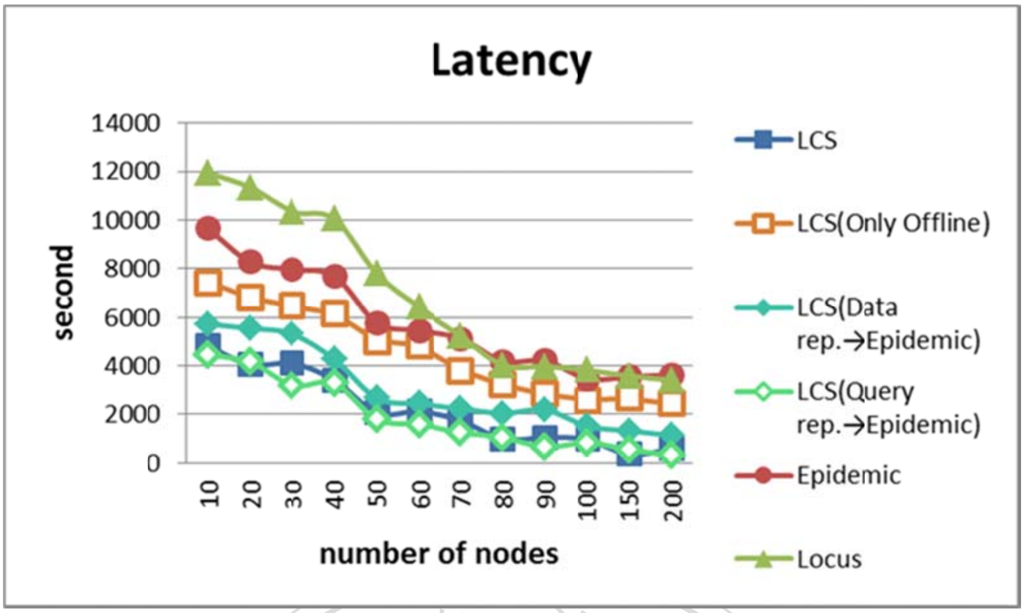


Figure 27: Latency (Node Density)

4.3.4 Buffer Size

Figure 28 shows the query-reply success ratio. We can see the result of our approach LCS and Locus. As the buffer size increased, nodes could store more messages, and it could increase the query success probability. Our approach and Locus have similar rule to replicate messages, so the success ratio are similar. But LCS has the advantage of overhead and latency.

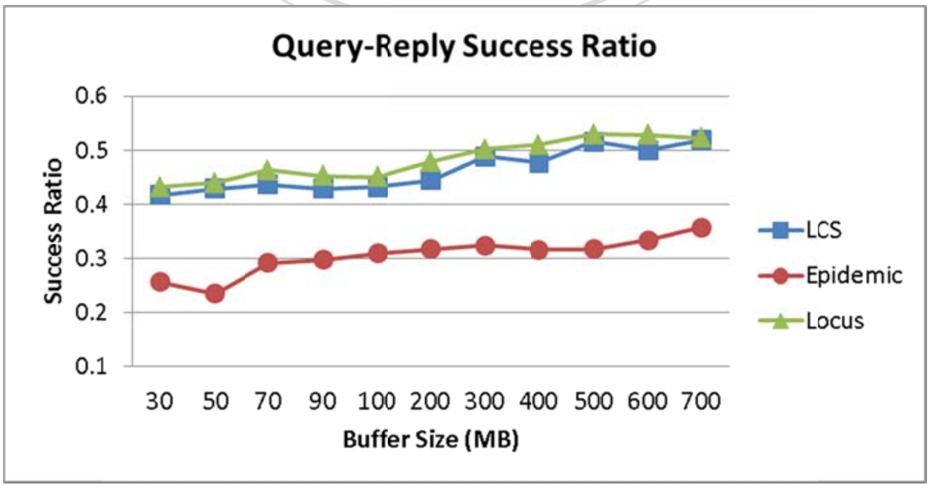


Figure 28: Query-Reply Success Ratio (Buffer Size)

Figure 29 is the overhead. As the buffer size increase, all the overhead of the approaches increase slowly. Because node has more space to store messages, when nodes encountered, they can send more messages. But the transmission rate and the encounter time are limited, it can't send more as buffer size increased.

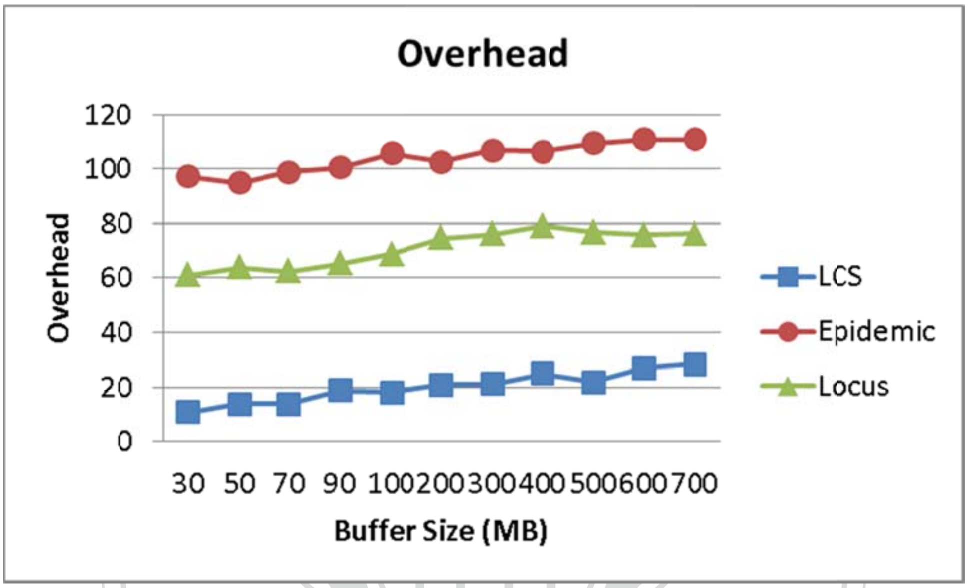


Figure 29: Overhead (Buffer Size)

Figure 30 is the query latency. As the buffer size increase, all the approaches decrease the latency results. Because nodes have more space to store messages, it is possible to forward the reply message to the Replier. Our approach LCS is the best one in all the results. We use the local region concept to determine the message forwarding, and the big storage could store more important messages, like reply message, then, it could enhance the delay time of query-reply activity.

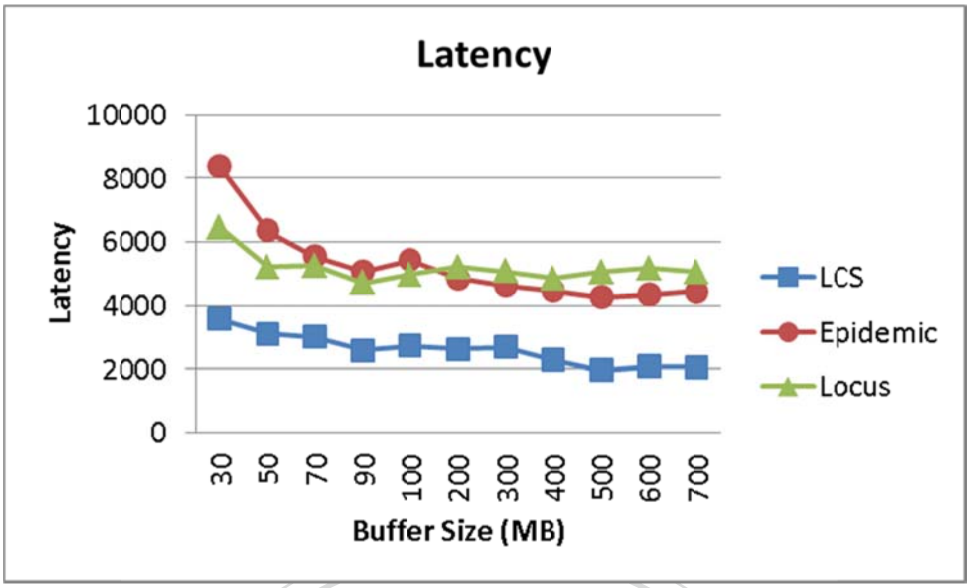


Figure 30: Latency (Buffer Size)

4.3.4 Time to Store

This section we want to see the influence of messages live time changing. Figure 31 shows the query-reply success ratio results, we can see the low TTS parameters, every approach display the bed performance, because messages just have short time to live, they don't have much time to delivery, so the success ratio doesn't well. In the low TTS, Locus is worse than LCS, it is because they will try their best to centralize the data in its area, but the live time of data isn't enough, it drops quickly, so nodes are difficult to query data.

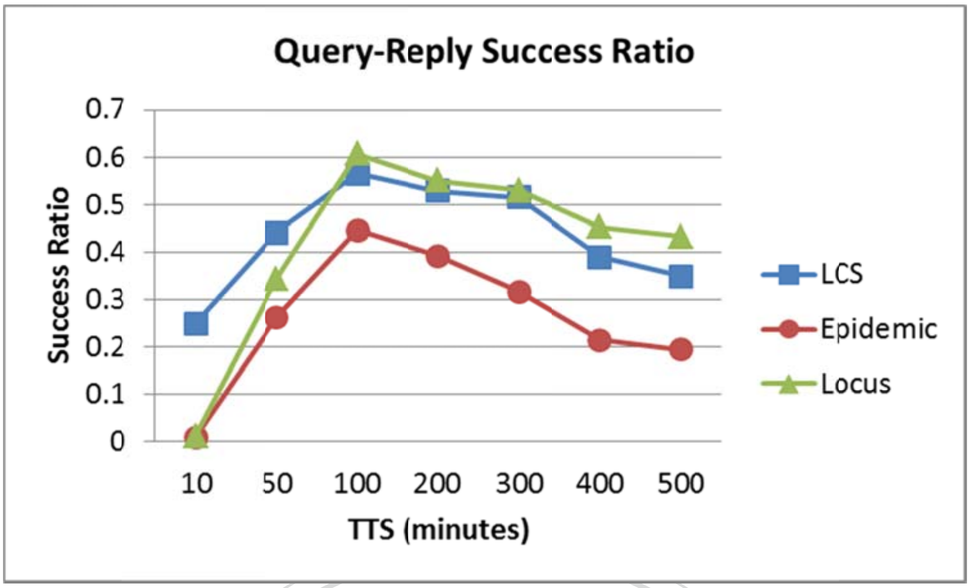


Figure 31: Query-Reply Success Ratio (TTS)

Figure 32 shows the overhead results, as the TTS growth up, the overhead of all the approaches will increase. Because messages have a long time to stay in the network, if reply messages success delivery to the Querier, the other message which doesn't send to the Querier still exist in the network. Then, these messages will increase the overhead.

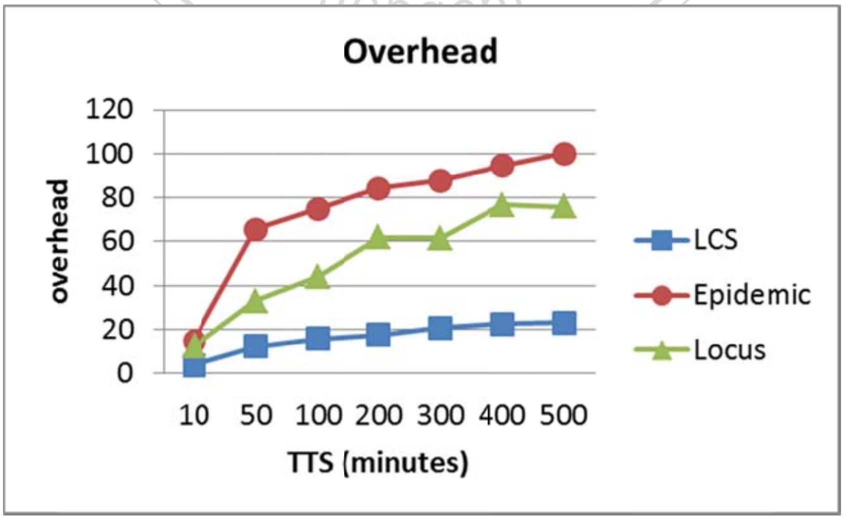


Figure 32: Overhead (TTS)

Figure 33 shows the latency results. As the TTS growth, the latency of all the approaches will increase. Because there are many messages in every node, and in the sending queue, the old message will be sent first. So the new one is difficult to be sent, then, it could spend much time to forward.

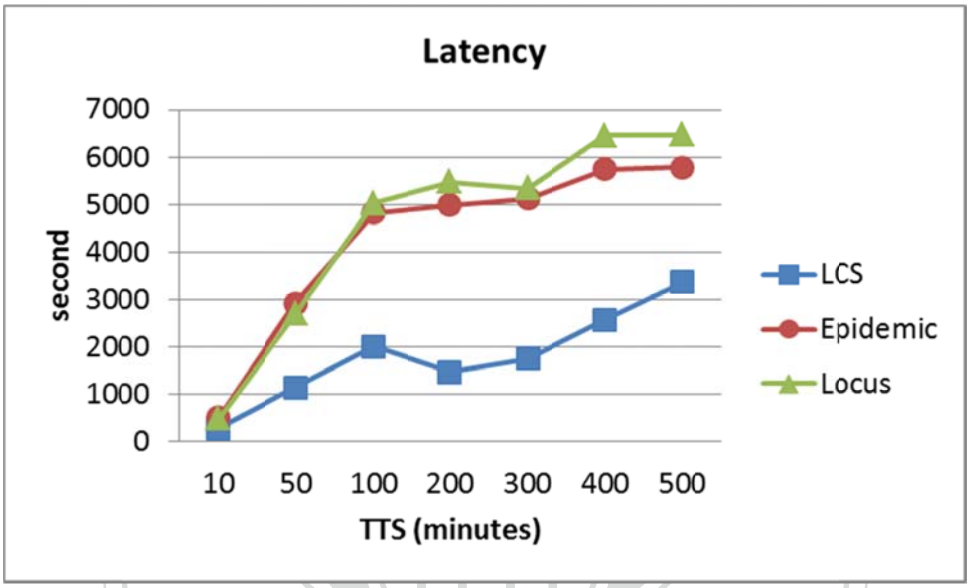


Figure 33: Latency (TTS)

CHAPTER 5

Conclusions and Future Work

In this thesis, we proposed a location-based content search approach. We use four strategies to achieve our objective. It is Data replication strategic, Query replication strategic, Data reply strategic and Data synchronize and update. Then we proposed a three-tier area concept, and every strategic in different area will have different rule to replicate messages. We divided all nodes into two parts, Online-node and Offline-node. If Offline-node wants to search some information, besides the other Offline-node, it can ask Online-node for searching. And in the sending message period, we proposed a message queue selection algorithm to decide which message will be sent first. Finally, we evaluate the results with Epidemic and Locus. And we use some parameters to verify our approach. Although LCS is worse than Locus in Query-Reply success ratio, in Overhead and Latency, LCS is much better than Locus.

In the future, we will consider the adaptive weighted value of messages. According to the message forwarding times or the estimation numbers of unique messages in some area. The weighted value of messages could be revised, and the nodes can spread the most important message out. Finally, we will implement this work into Plastory [28] system which is a mobile storytelling platform we developed before.

REFERENCE

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall and H. Weiss, "Delay-Tolerant Networking Architecture", RFC4838, April 2007, [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[2] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", In Proc. SIGCOMM, August 2003.

[3] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep. CS-2000-06, Duke University, July 2000.

[4] T. Spyropoulos, K. Psounis, C.S. Raqhavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks", EE, USC, USA, In Proc. SIGCOMM, August 2005.

[5] T. Spyropoulos, K. Psounis, C.S. Raqhavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility", USC, EECS, USA, In Proc. PERCOM, March 2007.

[6] A.Lindgren, A.Doria, and O. Schel'en, "Probabilistic Routing in Intermittently Connected Networks", LUT, Sweden, In Proc. SIGMOBILE, vol. 7-3, July 2003.

[7] D.K. Cho, C.W. Chang, M.H. Tsai, M. Gerla, "Networked medical monitoring in the battlefield", CS, UCLA, USA, In Proc. MILCOM, November 2008.

[8] C. Caini, P. Cornice, R. Firrincieli, D. Lacamera, S. Tamagnini, "A DTN Approach to Satellite Communications", DEIS/ ARCES, University of Bologna, Italy, In Proc. IEEE Journal on Selected Areas in Communications, vol. 26-5, June 2008.

[9] Y. Sasaki, Y. Shibata, “Distributed Disaster Information System in DTN Based Mobile Communication Environment”, Japan, In Proc. BWCCA, November 2010.

[10] P. Jinag, J. Bigham, E. Bodanese, “Adaptive Service Provisioning for Emergency Communications with DTN”, EECS, Queen Mary University of London, UK, In Proc. WCNC, March 2011.

[11] Nathanael Thompson, Riccardo Crepaldi and Robin Kravets, “Locus: A Location-based Data Overlay for Disruption-tolerant Networks”, CS, University of Illinois Urbana-Champaign, USA, In Proc. CHANTS, September 2010.

[12] P. Yang, M.C. Chuah, “Performance evaluations of data-centric information retrieval schemes for DTNs”, CSE, Lehigh University, USA, In Proc. MILCOM, November 2008.

[13] Cong Liu, Jie Wu, Xin Guan and Li Chen, “Cooperative File Sharing in Hybrid Delay Tolerant Networks”, In Proc. ICDCSW, June 2011.

[14] Mikko Pitk¨anen, Teemu K¨arkk¨ainen, Janico Greifenberg, and J¨org Ott, “Searching for Content in Mobile DTNs”, Helsinki University of Technology, Finland, In Proc. PERCOM, March 2009.

[15] S.C. Nelson, M. Bakht, R. Kravets, “Encounter–Based Routing in DTNs”, CS, University of Illinois at Urbana-Champaign, USA, In Proc. INFOCOM, April 2009.

[16] E.C.R de Oliveria, C.V.N. de Albuquerque, “NECTAR: A DTN Routing Protocol Based on Neighborhood Contact History”, Universidade Federal Fluminense, Brasil, In Proc. SAC, March 2009

[17] J. LeBrun, C.N. Chuah, D. Ghosal, M. Zhang, “Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks”, UC Davis, USA, In Proc. VTC, May 2005.

[18] P. Hui, J. Crowcroft, E. Yoneki, “BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks”, University of Cambridge, UK, In Proc. MobiHoc, May

2008.

[19] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, D. Rubenstein, “Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet”, Princeton University, USA, In Proc. ASPLOS-X, October 2002.

[20] G. Sollazzo, M. Musolesi, C. Mascolo, “TACO-DTN: a time-aware content-based dissemination system for delay tolerant networks”, CS, University College London, UK, In Proc. MobiOpp, June 2007.

[21] S. Carrilho, G. Valadon, H. Esaki, “Hikari: DTN message distribute system”, The University of Tokyo, Japan, IPSJ SIG Technical Report, January 2009.

[22] C.E. Palazzi, A. Bujari, “A Delay/Disruption Tolerant Solution for Mobile-to-Mobile File Sharing”, University of Padova, Italy, Wireless Day, In Proc. IFIP, October 2010.

[23] B. Cohen, “Incentives Build Robustness in BitTorrent”, May 2003

[24] “InMobi Mobile Market 2011 Review: Explosive Growth in Mobile, Tablet Impressions Increase 771%*”, InMobi, January 2012.

Available: http://www.inmobi.com/press-releases/

[25] “Steve Jobs Declares Post-PC Era”, InformationWeek, June 2010.

Available: http://www.informationweek.com/news/software/productivity_apps/225300193

[26] T. Vincenty, “Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations”, Survey Review, Vol.23, 1975.

[27] Ari Keränen, Jörg Ott, Teemu Kärkkäinen, “The ONE Simulator for DTN Protocol Evaluation”, In Proc. SimuTools, March 2009.

[28] Sicai Lin, Tzu-Chieh Tsai, Shindi Lee, Sheng-Chih Chen, "Location-Based Mobile Collaborative Digital Narrative Platform", In Proc. The Third International Conference on Creative Content Technologies (CONTENT 2011), Sep 2011.