

國立交通大學

資訊管理研究所

碩士論文

在行動隨意網路中利用群集與情境資訊的

服務發現

Service Discovery using Cluster and Context Information in

Mobile Ad Hoc Networks

研究生：潘律甫

指導教授：羅濟群教授

中華民國 102 年 6 月

在行動隨意網路中利用群集與情境資訊的服務發現
Service Discovery using Cluster and Context Information in
Mobile Ad Hoc Networks

研究生：潘律甫

Student：Lu-Fu Pan

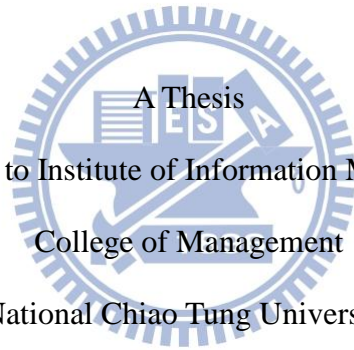
指導教授：羅濟群

Advisor：Chi-Chun Lo

國立交通大學

資訊管理研究所

碩士論文



A Thesis
Submitted to Institute of Information Management
College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in Information Management

June 2013

Hsinchu, Taiwan, the Republic of China

中華民國 102 年 6 月

在行動隨意網路中利用群集與情境資訊的服務發現

研究生：潘律甫

指導教授：羅濟群 老師

國立交通大學資訊管理研究所

摘要

隨著動態網路與行動隨意網路(MANET)的發展，服務發現(Service Discovery)機制在行動隨意網路中可進行資源和服務的共享，因此，如何在行動節點的電力和儲存能力有限的情況下，快速發現正確可用服務的服務發現機制足以影響整個行動隨意網路系統的效能。GSD (Group-based Service Discovery Protocol)為隨意網路中典型的服務發現協定，但是有太多的冗餘封包傳送造成大量的網路傳輸量(Network Overhead)，為了改善這個情況，本論文提出了一個基於群集與情境資訊的隨意網路發現機制：Context Node enhanced Group-based Service Discovery Protocol (CNGSD)，利用群集的形式搭配群組資訊來達到需求封包轉送、增加服務發現的效率與減少冗餘封包的傳送。此外透過情境資訊能夠讓使用者取得更適合的服務，本論文利用 NS-3 進行實驗模擬，模擬結果顯示了 CNGSD 與 GSD 的比較，CNGSD 具有較好的服務發現效能，平均少了 71.2%網路傳輸量；回應時間也比較快，平均加速了 44.5%。

關鍵字：GSD、隨意網路、服務發現、情境資訊

Service Discovery using Cluster and Context Information in Mobile Ad Hoc Networks

Student: Lu-Fu Pan

Advisor: Dr. Chi-Chun Lo

Institute of Information Management
National Chiao Tung University

Abstract

With growing of dynamic networks and mobile ad hoc network (MANET), service discovery is prerequisite part in pervasive environment. It can share the resources and services of MANET. In the case of limited power and storage capacity of the mobile node, an effective mechanism for discovering service affects the performance of the system. GSD (Group-based Service Discovery Protocol) is a typical service discovery protocol in MANET, but it creates enormous network overhead due to redundant packet transmission. In this paper, a cluster-based and context information service discovery approach: Context Node enhanced Group-based Service Discovery Protocol (CNGSD) is proposed. In form of clustering with group information leads to intelligence forwarding the request packets, increase performance of service discovery, and reduce redundant disseminative packets. Besides, clients can get better service matching by using context information. In this paper, we use GSD as benchmark and compare with CNGSD in NS-3. Simulation results show that our approach has better performance than GSD with an average 71.2% less network overhead. The response time also accelerated with an average of 44.5%.

Keyword: GSD, MANET, Service Discovery Protocol, Context Information,

致謝

能夠順利完成研究所碩士的學業，首要感謝我的家人，他們對我的支持與鼓勵，讓我沒有後顧之憂，可以把重心放在學業上，不論遇到什麼挫折與困難，隨時都能夠有一個依靠，讓我順利完成學業。

接著我要感謝我的指導教授羅濟群老師，對我的指導與幫助，讓我不管是在學術領域上還是做人處事的道理上都有更深一層的進步，思維能夠更細膩更周全，此外，我更要感謝實驗室博班的學姐張栩嘉，帶領著我完成論文，不時給我的叮嚀與建議，讓我在研究的路上不會迷失方向，我還要感謝其他的博班學長、學姊：斯寅、志華，在各方面給我的協助與經驗的分享，真的受益良多。

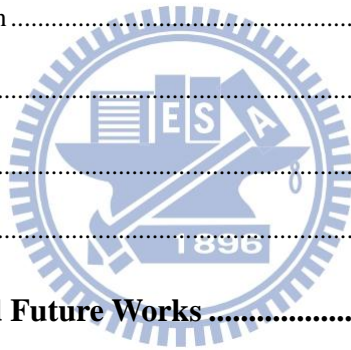
再來要感謝的是泊寰學長，給我機會能夠在學長的公司工讀，讓我能夠有機會不斷的汲取各種技術並且茁壯成長，同時了解現在業界的生態發展，使我能夠具備更多的技能與競爭力。

然後我還要感謝實驗室的同學—建宏、楷名、健彰、家宏、宗育、果、珈媛、岱蓉、美君，他們在學業上與生活上帶給我的點點滴滴與歡樂。當然也少不了學弟妹們帶給我各式各樣的歡樂氣氛，讓我在苦悶的研究生涯中依舊能保持愉快的心情。

Contents

摘要.....	i
Abstract.....	ii
致謝.....	iii
Contents	iv
Chapter 1 Introduction.....	1
1.1 Research Background and Motivation	1
1.2 Research Objective	2
1.3 Research Approach.....	2
1.4 Organization.....	3
Chapter 2 Literature Reviews.....	4
2.1 Mobile Ad Hoc Networks (MANET)	4
2.1.1 Service Discovery Protocol in MANET.....	4
2.2 Overview of GSD	9
2.2.1 Service Advertisements and Peer-to-Peer Caching	9
2.2.2 Service Discovery Routing	11
2.3 Context Information in Service Discovery Protocols	12
2.4 Clustering for MANET	14
2.5 Network Simulator 3 (NS-3).....	15
2.6 Summary of Related Works	15
Chapter 3 A Service Discovery Protocol using Cluster and Context Information in Mobile Ad Hoc Network.....	17
3.1 Design Issues	17
3.2 Notation	18
3.3 Context Node enhanced Group-based Service Discovery Protocol (CNGSD).....	19
3.3.1 Network Formation Phase.....	20

3.3.2 Service Advertisement Phase	22
3.3.3 Service Discovery Process Phase	25
3.3.4 Service Maintain Phase	29
3.3.5 Message Type in CNGSD	31
3.4 Discussion.....	32
Chapter 4 Simulation and Result Analyses.....	33
4.1 Simulation.....	33
4.1.1 Simulation Scenario	33
4.1.2 Simulation Environment	34
4.1.3 Simulation Setup.....	35
4.1.4 Simulation Assumption	35
4.1.5 Simulation Metrics.....	36
4.2 Results and Analyses	36
4.3 Discussion.....	40
Chapter 5 Conclusions and Future Works	41
5.1 Conclusions	41
5.2 Future Works.....	42
REFERENCES.....	43



List of Figures

FIGURE 2.1 EXAMPLE OF SERVICE ADVERTISEMENT MESSAGE IN GSD.....	10
FIGURE 2.2 EXAMPLE OF SERVICE DISCOVERY IN GSD.....	12
FIGURE 3.1 STATE DIAGRAM OF CNGSD	20
FIGURE 3.2 THE PROCEDURE DIAGRAM OF THE “NETWORK FORMATION PHASE”.....	21
FIGURE 3.3 EXAMPLE OF THE “NETWORK FORMATION PHASE” IN CNGSD	21
FIGURE 3.4 THE PROCEDURE DIAGRAM OF “SERVICE ADVERTISEMENT PHASE”.....	23
FIGURE 3.5 THE MESSAGE FLOW OF “SERVICE ADVERTISEMENT PHASE”	23
FIGURE 3.6 EXAMPLE OF “SERVICE ADVERTISEMENT PHASE” IN CNGSD	24
FIGURE 3.7 THE PROCEDURE DIAGRAM OF “SERVICE DISCOVERY PROCESS PHASE”.....	26
FIGURE 3.8 THE MESSAGE FLOW OF “SERVICE DISCOVERY PROCESS PHASE”	27
FIGURE 3.9 EXAMPLE OF SERVICE REQUEST IN CNGSD’S “SERVICE DISCOVERY PROCESS PHASE”	28
FIGURE 3.10 EXAMPLE OF SERVICE REPLY IN CNGSD’S “SERVICE DISCOVERY PROCESS PHASE”	29
FIGURE 3.11 THE PROCEDURE DIAGRAM OF “SERVICE MAINTAIN PHASE”	30
FIGURE 3.12 MESSAGE TYPE IN CNGSD.....	32
FIGURE 4.1 RESPONDS TIME OF SERVICE DISCOVERY PROCESS	37
FIGURE 4.2 PACKETS LOST RATIO	38
FIGURE 4.3 TOTAL NUMBER OF HELLO MESSAGE TRANSMITTED.....	39
FIGURE 4.4 DISTANCE OF NODES IMPACT THE PERFORMANCE.....	40

List of Tables

TABLE 2.1 COMPARISON OF DIRECTORY-BASED AND DIRECTORY-LESS SDPs	8
TABLE 2.2 SYMBOLS USED IN DEMONSTRATING FIGURE	10
TABLE 2.3 SUMMARY OF RELATED WORKS	16
TABLE 3.1 NOTATION DEFINITION	18
TABLE 3.2 SYMBOLS OF EXAMPLES	19
TABLE 4.1 SIMULATION ENVIRONMENT	34
TABLE 4.2 SIMULATION PARAMETERS	35



Chapter 1 Introduction

1.1 Research Background and Motivation

Mobile Ad Hoc Network (MANET) is a kind of dynamic networks, and it could provide various services which include hardware and software without any infrastructure[1]. The clients can connect the services wirelessly. On the other hand, data services like files and music can be transmit from one node to another node. For operating these services, a device must be locate in the network and invoke the required services.

Therefore, Service Discovery Protocol (SDP) plays an important role in the MANET, and lots of articles have been proposed [1-3]. Through SDP make services of MANET to be found and imposed. The normal manner for service discovery is the client who in the network gets the required service and directly communicates with the service. However, the better way for the service discovery needs minimal client implication and gets the most suitable one.


The context information can improve the service matching of the result which be found through SDPs. Although context aware mechanisms are always implemented in directory-based architecture such as [4-6], directory-less solutions do not pay attention to this issue. Combining context information with directory-less architecture, the approach will possess both better service matching and mobility in the MANET.

There are several service discovery protocols have been proposed in MANET and some of them inspired us in this paper, for instance, GSD, Jini, SLP, and so on. We will briefly describe in the following sections.

1.2 Research Objective

A directory-less architecture which should not depend on specific node is proper in the pervasive computing environment. Besides, we need an efficient approach to discover services in the network. A efficient service discovery approach can create little network overhead when the network forward the request message and reply message between services and clients in addition to service advertisement message. Moreover, it can provide a better result for clients with context information which contains dynamic properties depends on the type of services. If combining context information with directory-less architecture, the approach will possess both better service matching and mobility in the MANET.

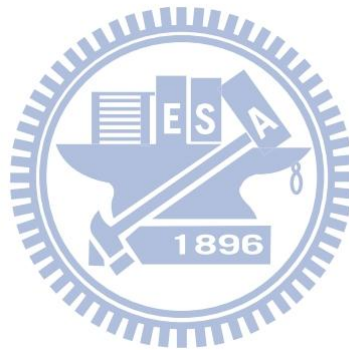
1.3 Research Approach



In this paper we propose an efficient service discovery approach which is based on group-based service discovery protocol (GSD)[7, 8] with context information: Context Node enhanced Group-based Service Discovery Protocol (CNGSD). In this approach, the network will divided into several clusters, and each cluster will elect a clusterhead, called *Context Node* which is mainly responsible for most of the messages dissemination of service discovery. Additionally, CNGSD has Service Weight Table (SWT) to find out the services according to clients' preferences. CNGSD is a two-tier and directory-less architecture. By using clusterheads deal with more dissemination can create less overhead than GSD, and imposing SWT can facilitate clients find the most suitable service in MANET.

1.4 Organization

The remaining part of this paper is structured as follows. Chapter 2 contains the literature reviews including the typical directory-based and directory-less SDPs. Second, the overview of GSD will be described. Third, we briefly introduce context information in SDPs. Fourth, clustering algorithm for MANET will be presented. Last, a summary and comparison of these related work is given. Chapter 3, we will describe the proposed approach CNGSD. In Chapter 4, we present the simulation result and analysis of GSD and GNGSD. Chapter 5 concludes this paper and proposes some future works.



Chapter 2 Literature Reviews

In this chapter, we first briefly introduce Mobile Ad Hoc Network and the related Service Discovery Protocol (SDPs). In the end of this section, a comparison table of SDPs is given. Second, an overview of GSD is presented. Third, we present the context information in service discovery protocol. Fourth, clustering algorithm for MANET will be introduced. Last, a summary of these related works is given.

2.1 Mobile Ad Hoc Networks (MANET)

MANET [1-3] consists of a set of mobile nodes, as a typical dynamic network without infrastructure in pervasive environments. Each mobile node in a MANET is free to move independently, and links with neighbor nodes frequently for maintaining network topology. Clients in the network can connect the services wirelessly. In the past, MANET applies to military operation, disaster rescue, and environment monitoring because of the resource limitation in nodes. However, with the growth of nodes capacity, mobile nodes are able to share the resources and data services recently. Thus, how to find the required services is an important issue in the MANET. We will introduce the major service discovery protocol in the following section.

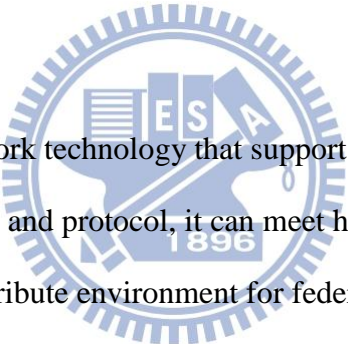
2.1.1 Service Discovery Protocol in MANET

Generally, SDPs classified into two categories: directory-based and directory-less[9]. In directory-based SDPs, there is a directory node in the networks, and request packets from clients have to forward to the directory node first. Then the node will search the matching service in the network which always implicated in wire network. In the beginning, services in directory-based architecture should register with directory node when they join the network. For example, Salutation, Jini, UPnP, and SLP are all directory-based architecture.

Salutation[2, 3]

Salutation is one of the major cooperation architecture to deal with service discovery and exploitation a wide set of application and equipment in mobile ad hoc network. In salutation architecture, there are two core components: salutation manager and transport manager. Salutation manager is a directory in the network and it is defined as a service broker: “A service provider registers its capability with a Salutation Manager. When a client asks its local Salutation Manager for a service search, the search is performed by coordination among Salutation Managers. Then, the client can use the returned service. ”. Transport Manager s provide reliable communication channel, regardless of what the underlying network transports are.

Jini[2, 3, 10]



Jini is middleware network technology that support a set of application programming interface (API) and protocol, it can meet home network requirement. Jini technology provides distribute environment for federating clients, devices, and services through Java Remote Method Invocation (RMI). Each device publishes its interface that other devices in the network can communicate with it and access the specific service. All the service providers register their service announce with lookup service first. In this process, service providers upload the serialized Java object, named service proxy. The lookup service is the repository of services offered by the network. When clients request for specific services, they sent service requests to lookup service and lookup service will return the service proxy object to clients if there are matching service. Once the client gets the proxy, it can interact with the network service directly through the proxy and establish a communication channel.

UPnP[2, 3, 5]

Universal Plug and Play is a peer to peer architecture in mobile ad hoc network; it can connect intelligent applications, wireless devices, and PCs in the network. A device can join the network dynamically and get an IP address which is automatic configured. Besides, it convey its capabilities on request, learns the presence and capabilities of other ones in the network. UPnP leverages TCP/IP, UDP, HTTP, and XML, to enable control and data transmit among network devices in the home or indoors seamlessly. UPnP uses Simple Service Discovery Protocol (SSDP) in service discovery process. Through SSDP, devices announce their presence and discovery others or services. SSDP uses HTTP over multicast and unicast UDP. UPnP consists of three major parts: devices, services, and control point. Control point in UPnP provides capability of service discovery and control devices by receiving the description of devices or services. After retrieving the service description, control point sends control messages which are expressed in XML by Simple Object Access Protocol (SOAP) to the target device and waits for service reply.

SLP[2-4]

Service Location Protocol is a decentralized, lightweight, scale and extensible service discovery protocol designed for TCP/IP networks to enterprise environment. there are three agents in SLP: User Agents(UA), Service Agents(SA), and Directory Agents(DA). UA perform service discovery on behalf of clients software by the return URL of service, DA as a centralize service repository store information about services announcement in the network, and SA advertise the description and attribute on behalf of services and should refresh the registration with DA periodically. SLP provides a flexible and scalable solution for supporting hosts with access to information about the existence, location, and configure network services by scope mechanism.

However, the above service discovery protocols face some issues about single point of failure, directory selection and maintenance. In contrast, directory-less SDPs use broadcast to send request packets to the neighbor nodes in the network and try to find whether matching services or not. For instance, GSD, Konark, and PDP are directory-less architecture.

Konark[1, 3]

Konark is a middleware for discovery and delivery services in multi-hop ad hoc networks. It support push and pull models by using cache in every devices in the network. Moreover, Konark exploits XML to describe its service descriptions and uses multicast to announce and discovery services. It allows services delivery by lightweight HTTP server on all devices which host services.

PDP[1, 3]

PDP is a distributed service discovery protocol in ad hoc network and merges features of push and pull models. PDP considers the limitation of every device in the network, for instance, device power and processing capabilities. All messages in PDP are broadcasted; all devices cooperate through their replies and the information in their caches. Because PDP takes into account the needs of different application, it allows reducing the power consumption.

GSD[7, 8]

GSD is a typical service discovery protocol in ad hoc network and we will introduce the detail of GSD in section 2.2.

The directory-less service discovery protocols also encounters some issues, including frequency of service advertisement, scheduling and priority, and broadcast storm. The comparison between directory-based and directory-less architecture as shown in Table 2.1.

Table 2.1 Comparison of Directory-based and Directory-less SDPs

	Directory-based	Directory-less
Directory Node	Yes	No
Role	Service provider (Server) Service requestor (Client) Service directory	Service provider (Server) Service requestor (Client)
Suitable environment	Wire network	MANET / Wireless
Service provide	Registration	Broadcast
Service request	Through directory nodes	Broadcast
Issue	1. Single point of failure. 2. Directory selection and maintenance.	1. Frequency of service advertisement. 2. Scheduling and priority. 3. Broadcast storm.
Techniques	Salutation / Jini / SLP / UPnP	GSD / Konark / PDP

2.2 Overview of GSD

Group-based service discovery protocol(GSD) is characterized in 1) advertising of services are bounded in the vicinal nodes 2) peer-to-peer caching of service advertisements dynamically 3) group-based selective forwarding of service discovery packets. With these characteristic, GSD provides an efficient mechanism in service discovery, increased the bandwidth usage and has mobility.

2.2.1 Service Advertisements and Peer-to-Peer Caching

Every service provider advertises itself service advertisement periodically to vicinal nodes. Rather than traditional solution which broadcast service advertisement to the entire node in the network, GSD limits the spreading range of service advertisement through the field *ADV_DIAMETR*. The field can determines the count of traveling hop. Moreover, the service advertisement includes the *Service-Description* which describes the providing service and *Service-Group* which represents the corresponding service group of the service.

When receiving the service advertisement, each node will store the information in its *Service Cache* and maintain *Service Information Cache* that is called peer-to-peer caching demonstrates the service group of neighbor nodes belong to. As the result of service advertisement, a node can know not only the providing service but also groups about its neighbors.

Figure 2.1 shows an example about service advertisement in GSD with $ADV_DIAMETR = 1$. The symbols of figure are summary in Table 2.2.

In Figure 2.1, when receiving service advertisement message, each node will structure its *Service Information Cache*. If the network has any change for example, new node into the network, service advertisements need to update. Before forwarding

the packet to next node, the number of hops will be decreased by 1. Therefore, the limit of service advertisement message is 1, the remaining hop is 0.

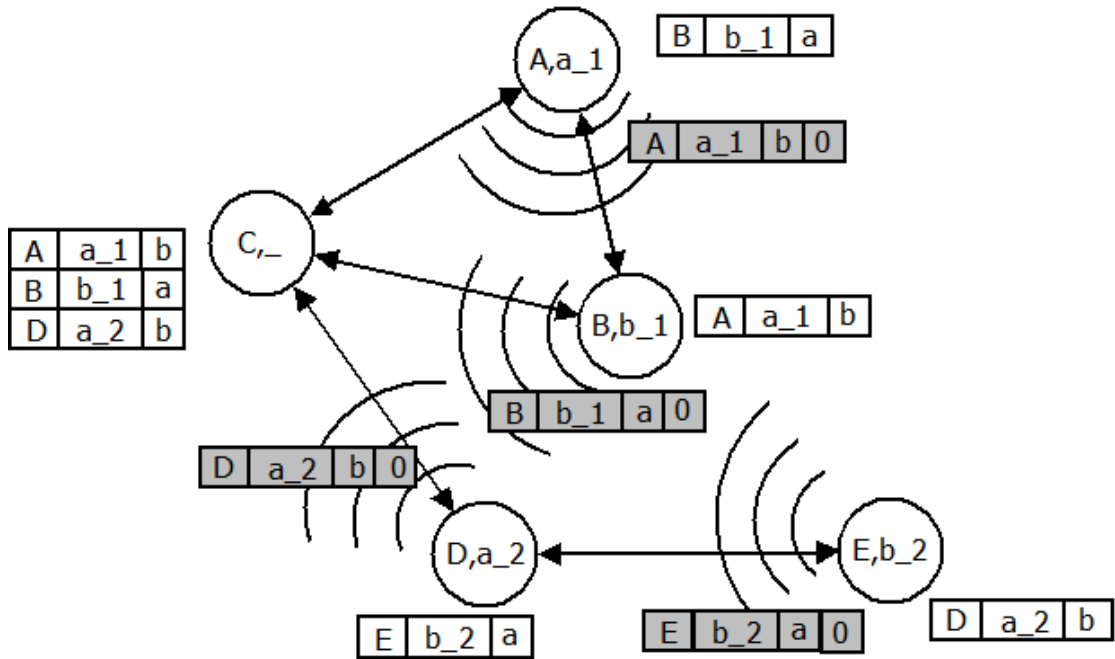


Figure 2.1 Example of service advertisement message in GSD

Table 2.2 Symbols used in demonstrating figure

Symbol	Indication
Circle	Node in the MANET
Arcs	Direction of packet transmission
Double-headed arrow	Both end of arrow are vicinal nodes
Text in the circle	ID of the node and its providing service
White table adjoining circle	Service Information Cache of the node (not all fields are shown)
Grey table over the arcs	Contents of packet transmission (not all fields are shown)

2.2.2 Service Discovery Routing

After advertising services, the network is ready for receiving service request.

When a node needs services, it will forward the request message to selected nodes in its *ADV_DIAMETR-hop* neighbor set. Each selected node receives request packet and transmits further, unless the count of traveling hop exceeds the limit or finds matching service. Figure 2.2 shows an example of service discovery in GSD, and symbols are the same as Figure 2.1.

Once node *C* requires service “*b_2*” belonging to service group “*b*”, and there is no matching service in its *Service Information Cache*. But node *C* learns that both node *A* and *D* have service belonging to group “*b*” in their vicinity. Therefore, node *A* and *D* will receive the request packet respectively as selected nodes. Take the request packet $\langle C, b_2, b, A, 1 \rangle$ sent to node *A* for example: 1) the source address is node *C*; 2) the requested service is “*b_2*”; 3) the requested service belongs to group “*b*”; 4) the packet is sent to node *A*; 5) the packet remaining number of hops is “*1*”. When receiving request packet, node *A* will forward the packet further. Once node *B* gets the packet, it will respond a reply packet instead of transmitting it further.

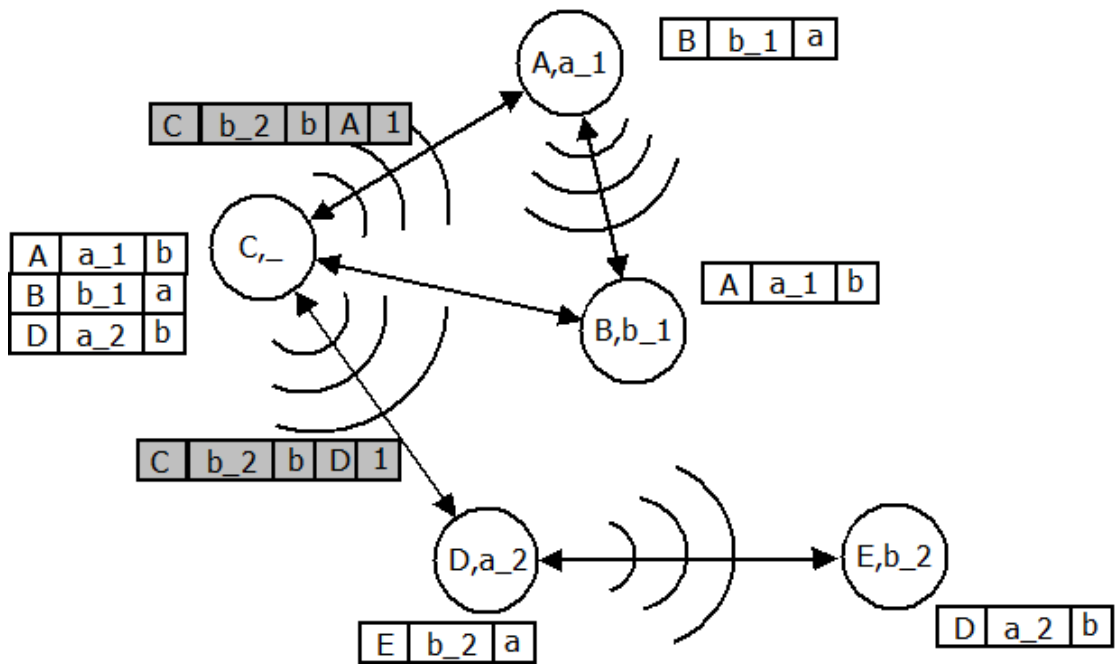


Figure 2.2 Example of service discovery in GSD

2.3 Context Information in Service Discovery Protocols

The context information refers to the surrounding environment and reflects current state of the service. Context is a situation of an entity (person, place or object) that is relevant to the interaction between a user and an application[11]. It can improve the service matching of the result which be found through SDPs.

Context aware mechanisms are always implemented in directory-based architecture such as[4-6].In [6],the model provides a context aware architecture and focus on dynamic service attribute. This model is based on directory-based service discovery protocol, SLP. The service attributes are divided into *static* which are assigned when the service is initially deployed like the printer is color, and *dynamic* which change as time goes by after the service has been set, for instance, the numbers of prints jobs in the queue. Besides, this model provides “*Table of Weights Service (TOW)*” for computing the ranking of matching services and the ranking is based on the score which is computed through the weight vector. The weight vector can be sent

by the client or use the default weight vector. It allows client to decide the priority of service attributes and gets the most suitable service which will obtain the highest score. But the model is suitable for fix network likes home network not for pervasive environments.

However, combining context information with directory-less architecture, the solution will possess both service quality and mobility in the MANET such as [12, 13]. In [12], the solution has used the GSD's model to categorize services into groups. In this model, the network is organized into a two-tier hierarchy and the resource-rich nodes, called *Ultrapeers (UPs)*; resource-constrained nodes, called *Leaves (LFs)*. *UPs* are responsible for most of the communication in the network. Moreover, *Ups* in the domain will select the most resource-rich *Up* as domain *Service Description Keeper (SDK)*, which store all the service advertisement information in its domain. In the beginning, *LF* in various domains selects an *UP* and sends service advertisements to the serving *UP*. After receiving service advertisements, *Ups* aggregate the information and transmit to the *SDK*. If the service request is from *LFs*, the packet will send to the serving *UP* to find the matching service or selectively forward the packet further. The model is context aware and suitable in pervasive network.

2.4 Clustering for MANET

There are several clustering algorithm in the MANET like [14-17]. But in this paper, we do not pay attention on clustering algorithm, but we assume the clusterhead is elected based on the criteria[17] which determine by certain nodes parameters, like: ideal degree, transmission power, mobility and battery power. A good clustering algorithm affects the performance of MANET, and provides good stability.

WCA- a weighted clustering algorithm for mobile ad hoc network takes nodes degree, distances, mobility, and consumed battery power into account. It defines the number of nodes that a clusterhead can handle is δ .

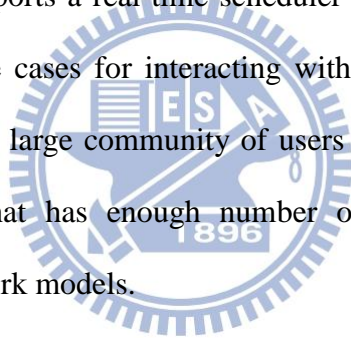
- 1) The degree of each node v , $d_v = |N(v)| = \sum_{v' \in v, v' \neq v} \{dist(v, v') < tx_{range}\}$.
- 2) The degree-difference for each node v , $\Delta v = |d_v - \delta|$.
- 3) Compute the sum of distances $D_v = \sum_{v' \in N(v)} \{dist(v, v')\}$.
- 4) Compute the average of speed for each node till current time T. Define the mobility $M_v, M_v = \frac{1}{T} \sum_{t=1}^T \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$.
- 5) Compute the cumulative time, P_v , during the node v acts as a clusterhead, implies how much battery power has been consumed.
- 6) Use $W_v = w_1 \Delta v + w_2 D_v + w_3 M_v + w_4 P_v$ to calculate the weight of each node and choose the node with the smallest W_v as the clusterhead.

Clusterheads are responsible for maintain the topology of the network, and involve in most of traffic. Therefore, clusterheads usually served by resource-rich nodes, such as processing and transmission power. Other nodes register with the nearest clusterhead and become members of specific cluster. Finally, the network is divided to several clusters, and clusterheads play the critical role in the network.

2.5 Network Simulator 3 (NS-3)

NS-3[18] is a discrete-event network simulator targeted primarily for research and educational use. The goal of the ns-3 project is to develop a better, open simulation environment for networking research. It should be aligned with the simulation needs of modern networking research and should encourage community contribution, peer review, and validation of the software.

In addition, it supports research on both IP and non-IP based network including models for Wi-Fi, WiMax, LTE for network 1 and 2 layers. Besides, it supports variety of static and dynamic routing protocols, for instance, OLSR and AODV for IP-based applications. NS-3 also supports a real-time scheduler which facilitates a number of "simulation-in-the-loop" use cases for interacting with real systems. NS-3 wants to spread this workload over a large community of users and developers for creating a good network simulator that has enough number of high-quality validated and develops more related network models.



2.6 Summary of Related Works

The summary of related works is shown in Table 2.2. GSD[7, 8] and S.S. Yau and G.G. Pohare's[12] model are the nearest solution in pervasive environment. GSD is a directory-less solution and based on grouping for services in the network, but it doesn't have context information.

S.S. Yau and G.G. Pohare's model depends on *Ultrapereers (UPs)* and it is context aware. However, it does not mention how to organize the network and elect the *Ultrapereers (UPs)*, *Service Description Keeper (SDK)*. Network in the model divided into several domains and each domain has more than one *UP*. These *UP* will elect the most resource-rich as *SDK*. Thus, the model needs to operate more elections when the

network is formed. Moreover, because the *SDK* stores all the service information in the serving domain, network will modify lots of service advertisement information whenever services update.

S. Cuddy, M. Katchabaw and H. Lutfuyya's[6] model focus on dynamic attribute and use weight vector to ranking the matching services. Make it easier to deal with context information but it directory-based not suitable in dynamic environments.

Table 2.3 Summary of related works

	Network	Property	Architecture	Context Information	Clustering
GSD	Pervasive	Static	Directory-less	No	No
Pohare	Pervasive	Static	Distributed-Directory	Yes	Yes
Lutfiyya	Fix	Static, Dynamic	Directory-based	Yes	No
CNGSD	Pervasive	Static, Dynamic	Directory-less	Yes	Yes

Chapter 3 A Service Discovery Protocol using Cluster and Context Information in Mobile Ad Hoc Network

In this chapter, we present a cluster-based and context information service discovery protocol in MANET. First, we define the basic assumption of the paper. The detail of CNGSD will be introduced following.

3.1 Design Issues

Our approach is based on group-based service discovery protocol (GSD)[7, 8], but the network is divided into several clusters, and each cluster elects the clusterhead by the algorithm[17] firstly. All nodes in the network have enough memories for saving the service cache and service advertisement message. In addition, nodes have an omni-directional antenna and support context data collection component are necessary.

GSD is a typical service discovery protocol in MANET, but it has enormous network overhead due to redundant packet transmission, as a result, we want to propose an effective approach which service responds time is short, packet delivery ratio is stable and overhead is lower than GSD.

3.2 Notation

The notation definition is shown in Table 3.1, and Table 3.2 explains the symbols of examples which are presented in the following sections.

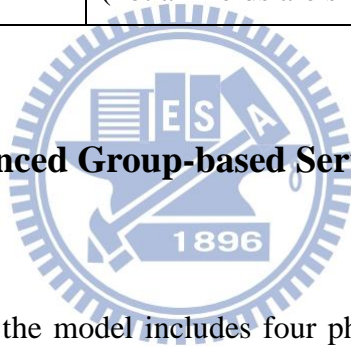
Table 3.1 Notation definition

Notation	Description
CN	Context node
SD	Service description
SGI	Service group information
SWT	Service weight table
VCP_{ijk}	The monitored value that the context data collection component[6] return for property VCP_k when node j requests the context information for service i .
w_{ijk}	The weight that node j assigns to service i for the property k , $w_{ijk} = (w_{ij1}, w_{ij2}, \dots, w_{ijn})$ which $w_{ij1} + w_{ij2} + \dots + w_{ijn} = 1$

Table 3.2 Symbols of examples

Symbol	Indication
Circle	Node in the MANET
Arcs	Direction of packet transmission
Double-headed arrow	Both end of arrow are vicinal nodes
Text in the circle	ID of the node and its providing service
White table adjoining circle	Service Information Cache of the node (not all fields are shown)
Grey table over the arcs	Contents of packet transmission (not all fields are shown)

3.3 Context Node enhanced Group-based Service Discovery Protocol (CNGSD)



A general overview of the model includes four phases. Network formation and service advertisement, service discovery process, and service maintain. Each of these parts will be explained individually. The state diagram of CNGSD is shown in Figure 3.1.

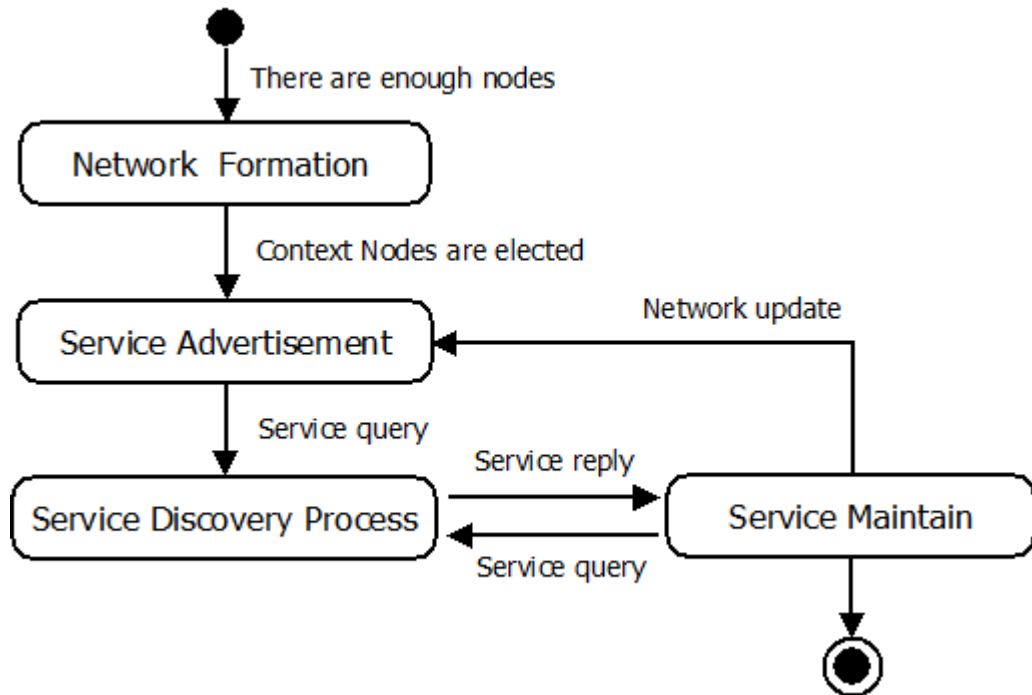


Figure 3.1 State Diagram of CNGSD

3.3.1 Network Formation Phase

In the beginning, the network is divided into clusters; each cluster is formed containing only one *CN* and several nodes. The “Network Formation Phase” can be summarized as follows:

- 1) Each node broadcasts 1-hop hello messages to discover neighbor nodes and stores the latest hellos which sent by neighbor nodes in its cache.
- 2) Use “Weighted Clustering Algorithm”[17] to election clusterhead as *CN*.

The procedure diagram of the “Network Formation Phase” is shown in Figure 3.2, and the example of CNGSD in this phase is shown in Figure 3.3. The symbols of the example are shown in Table 3.2.

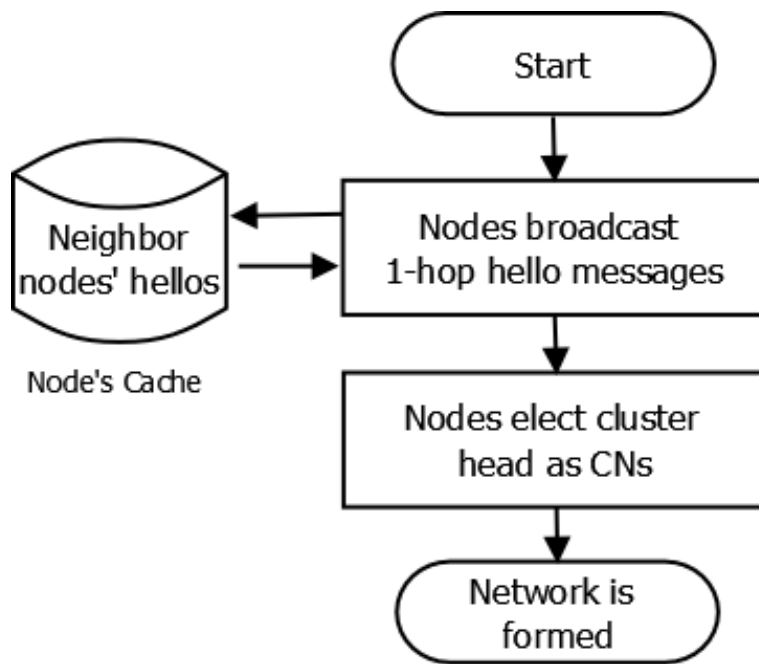


Figure 3.2 The procedure diagram of the “Network Formation Phase”

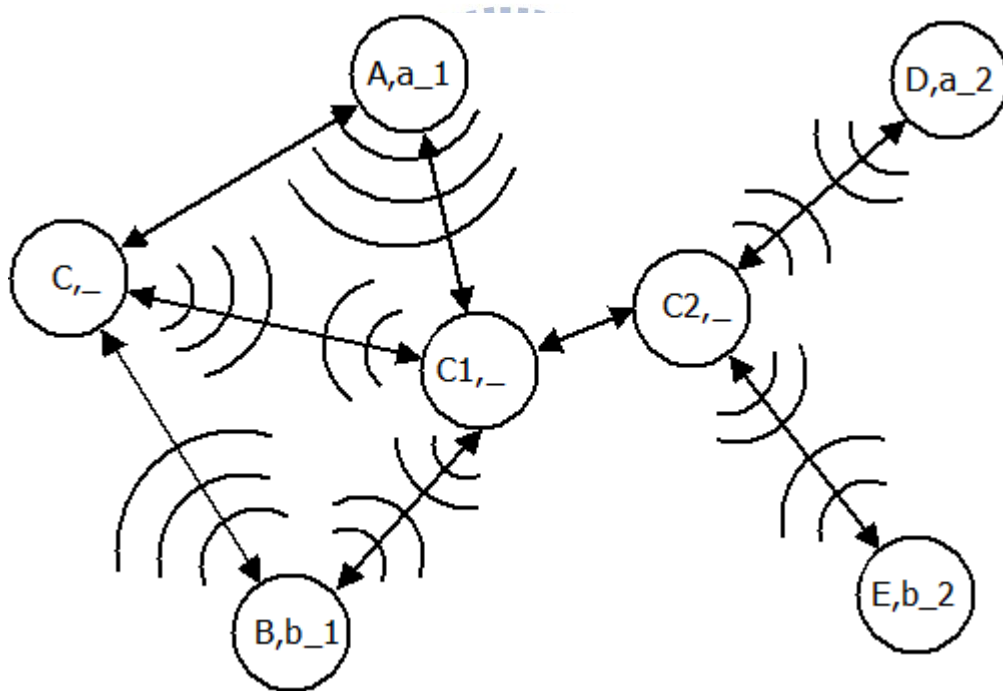


Figure 3.3 Example of the “Network Formation Phase” in CNGSD

3.3.2 Service Advertisement Phase

After “Network Formation Phase”, each node has its serving clusterhead, called *CN*. For the network be able to access, service advertisement is necessary. The “Service Advertisement Phase” is summarized as follow:

- 1) Each node connects to only one *CN* and sends *Service Advertisement Message* in a single-hop count to serving *CN*.
 - 1.1) *Service Advertisement Message* contains *SD* and *SD* is divided into two properties:
 - (a) Static Property: Description of the service which is assigned when the service is first deployed.
 - (b) Context Property: Context Information related to the service. The value changes after the service has been deployed.
- 2) The *CN* integrates all the *SD* in the cluster as *Service Cache Message* which stores all the *SD* in its cluster.
- 3) The *CN* exchanges *SGI* with neighbor *CN* which in the 1-hop range.
- 4) Network Services are ready for access.

The connections between any pair of directly connected nodes are maintained using the 1-hop periodic hello messages. Figure 3.4 presents the procedure diagram of “Service Advertisement Phase”, and Figure 3.5 shows the message flow of this phase.

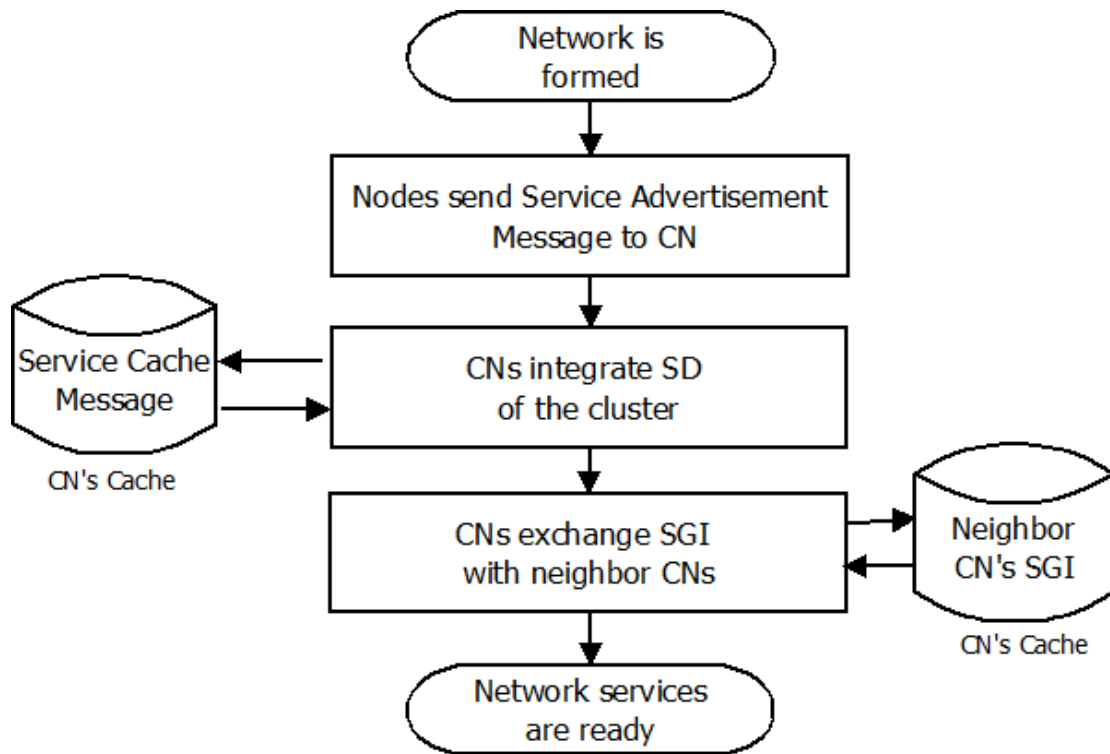


Figure 3.4 The procedure diagram of “Service Advertisement Phase”

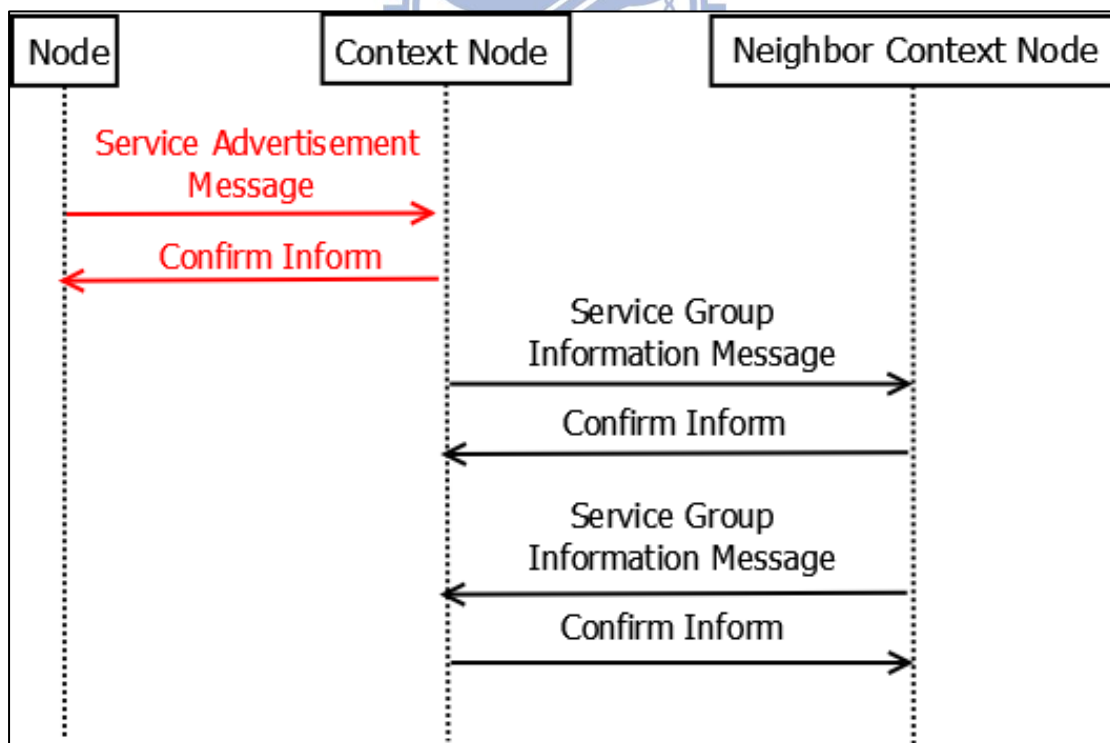


Figure 3.5 The message flow of “Service Advertisement Phase”

In Figure 3.6, there is an example of “Service Advertisement Phase” in CNGSD. Take *Context Node1* cluster for instance, node *A* and *B* sent their *Service Advertisement Message* to *Context Node1*. The packet $\langle A(C1), a_1 (distance, queue), 0 \rangle$ presents: 1) the source address is node *A*, and belongs to cluster “*C1*”; 2) the static property of service description is “*a_1*” and dynamic properties are “*distance*” and “*queue*”; 3) the packet remaining number of hops is “0”. The *SGL* packet $\langle C1, “a, b” \rangle$ shows the source cluster is “*C1*”, and there are services belongs to group “*a*” and “*b*”. The packet will send to neighbor *CNs*.

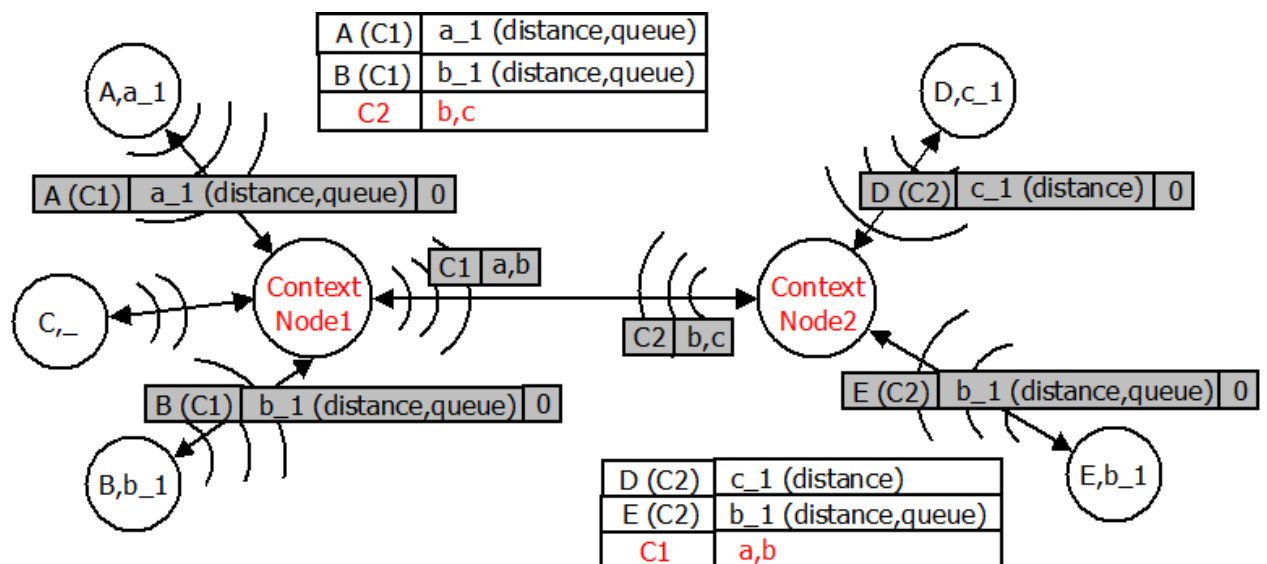


Figure 3.6 Example of “Service Advertisement Phase” in CNGSD

3.3.3 Service Discovery Process Phase

Once the network is formed and the services are advertised to their serving *CNs*, the network is ready for receiving service request from nodes in any cluster. The “Service Discovery Process Phase” can be summarized as follow:

- 1) Nodes in any cluster can send *Service Request Message* to its serving *CN*.
- 2) The *CN* first checks its *Service Cache Message* to find matching services in the local cluster.
- 3) If *SGI* has the required service group, the *CN* will also selectively forward the *Service Request Message* to the target *CN*.
 - 3.1) If no *SD* match at the *CN* and no required service group in the *SGI*, the *Service Request Message* will not transmitted further.
- 4) If matching services are found, corresponding nodes will collect the *Context Properties* of the service imposing context data collection component and return *Service Reply Message* to requesting *CN*.
- 5) When receiving the *Service Reply Message*, the requesting *CN* uses the *SWT* to rank the score of matching service.
 - 5.1) The context properties of matching services will be ordered, and be renumbered a value from high to low which represents the most suitable to inferior.
 - 5.2) Requesting node can send its weight vector for the required service to its serving *CN* or use the default weight vector.

6) The CN determines the most suitable service which has the highest score and returns to requesting node.

6.1) Score is computed by $\sum VCP_{ijk} * w_{ijk}$.

6.2) VCP_{ijk} : The monitored value that the context data collection component[6] return for property VCP_k when node j requests the context information for service i .

6.3) w_{ijk} : The weight that node j assigns to service i for the property k ,
 $w_{ijk} = (w_{ij1}, w_{ij2}, \dots, w_{ijn})$ which $w_{ij1} + w_{ij2} + \dots + w_{ijn} = 1$

The procedure diagram of “Service Discovery Process Phase” is shown in Figure 3.7, and the message flow is shown as Figure 3.8.

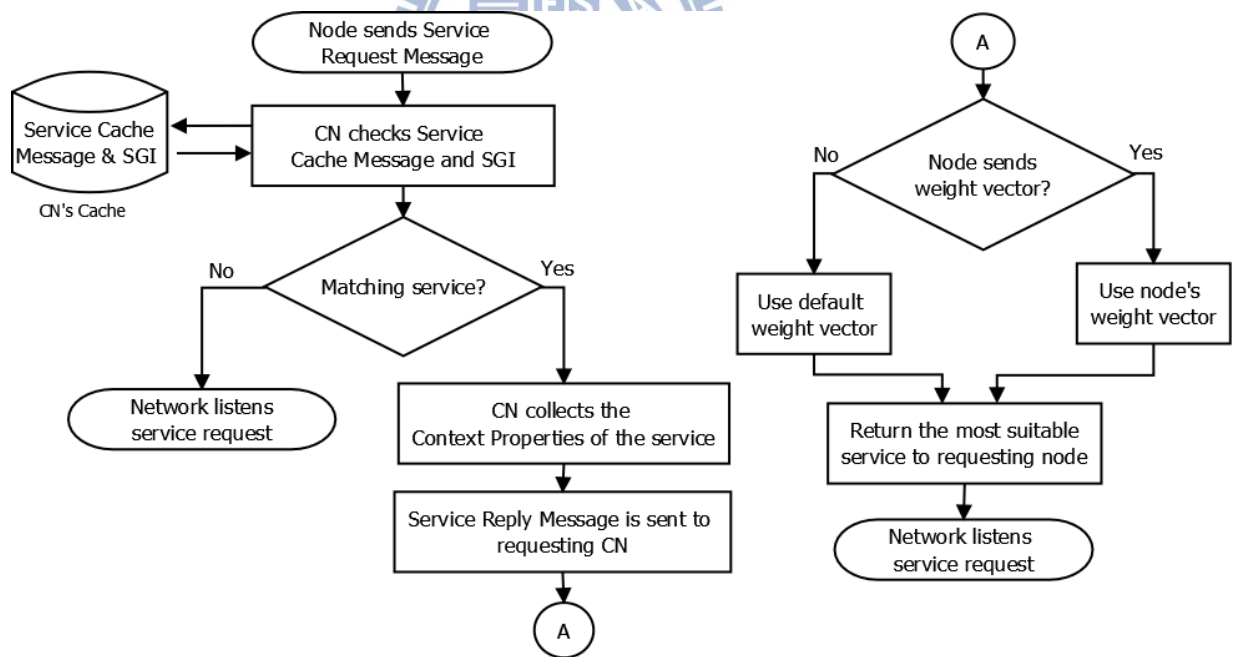


Figure 3.7 The procedure diagram of “Service Discovery Process Phase”

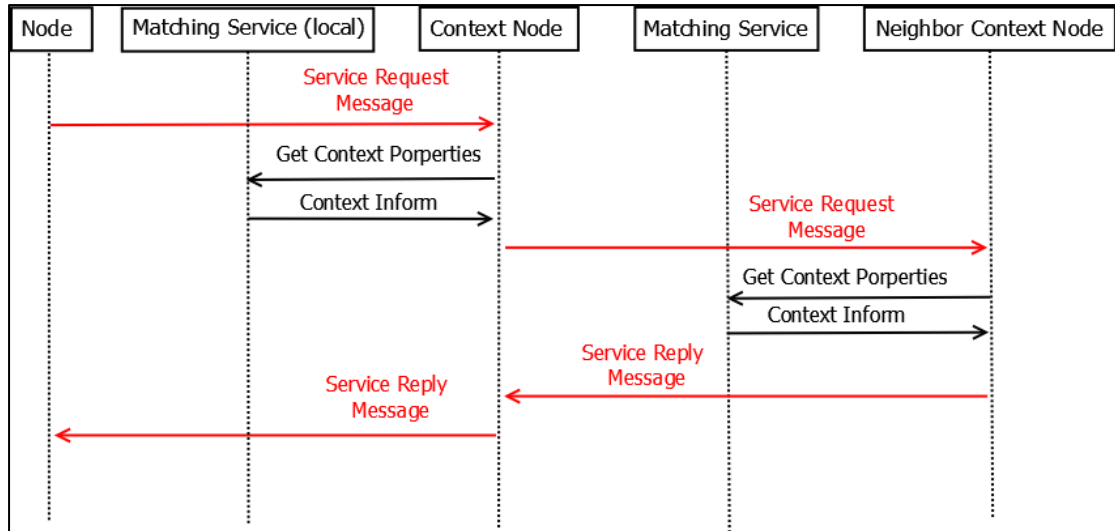


Figure 3.8 The message flow of “Service Discovery Process Phase”

Figure 3.9 shows the example of service request in CNGSD’s “Service Discovery Process Phase”. Node *C* creates a *Service Request Message* to *Context Node1* for getting service “*b₁*”. When receiving the packet, *Context Node1* first checks its *Service Cache Message* to find matching service. Although *Context Node1* finds service “*b₁*” in its cluster, *Context Node1* still forward the packet to neighbor clusterhead, *Context Node2* because of the *SGI*. It shows that there is also matching service group in the cluster “*C2*”. The packet $\langle C(C1), b_1 (distance, queue), b, C1, 1 \rangle$ means: 1) the source address is node *C*, and belongs to cluster “*C1*”; 2) the static property of service description is “*b₁*” and dynamic properties are “*distance*” and “*queue*”; 3) the requested service belongs to group “*b*”; 4) the destination is *Context Node1*; 5) the packet remaining number of hops is “*1*”.

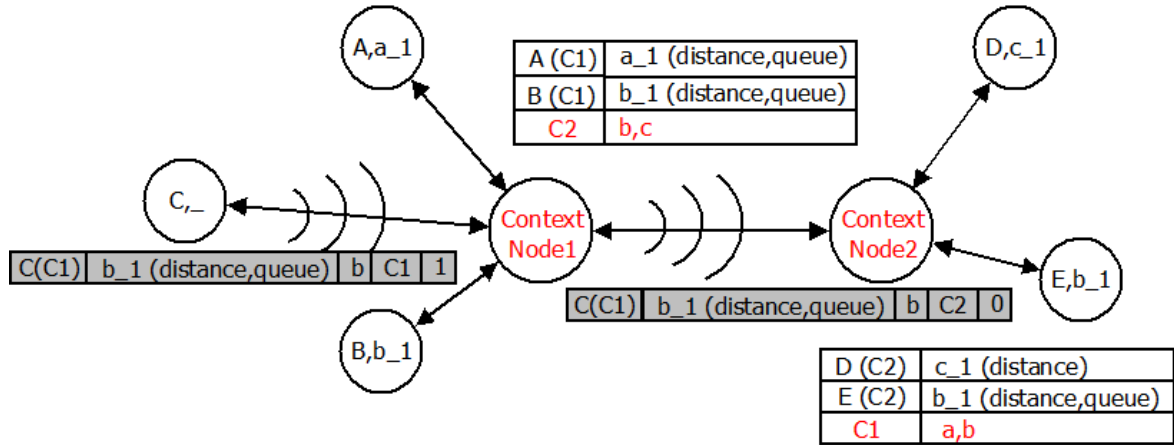
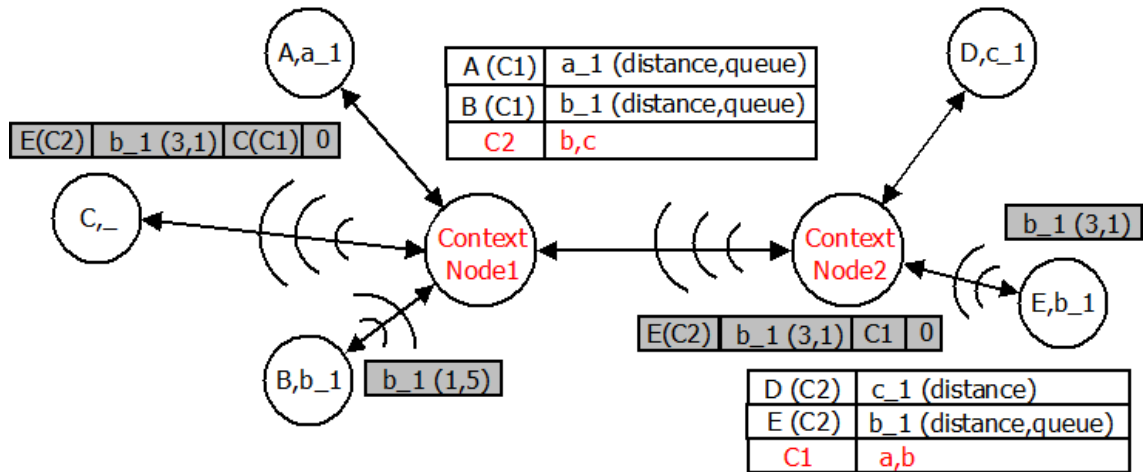


Figure 3.9 Example of service request in CNGSD’s “Service Discovery Process Phase”

In Figure 3.10, when *CNs* find the matching service, the corresponding node will use the collect the *Context Properties* of the service imposing context data collection component[6] and return *Service Reply Message* to requesting *Context Node1*. Node *B* and *E* responds the value of *Context Properties* individually. For example, $\langle b_1 (1, 5) \rangle$ represents $VCP_{distance} = 1$ and $VCP_{queue} = 5$ in node *B*. When receiving the *Service Reply Message*, *Context Node1* orders the value of *Context Properties*, and renumbers the value from high to low which represents the most suitable to inferior. The value of “*distance*” and “*queue*” in node *B* are renumbered as “2” and “1”. Next, *Context Node1* uses the *SWT* to rank the score of matching services. Take the weight vector $(w_{b_1,C,distance}, w_{b_1,C,queue}) = (0.4, 0.6)$, for instance, “*queue*” has higher priority in weight vector. In the end, node *E* obtains score “1.6”, and will be return to node *C*.



Context Properties are renumbered B: b_1 (2,1) E:b_1 (1,2)

Score B: $0.4*2+0.6*1 = 1.4$

Score E: $0.4*1+0.6*2 = 1.6$ **V**

Figure 3.10 Example of service reply in CNGSD's "Service Discovery Process Phase"

3.3.4 Service Maintain Phase

Whether requesting node gets the required service, the network will into the "Service Maintain Phase" after service discovery process. In this phase, the network continuous listens to service request unless there are any changes in the network. The following conditions will cause network update:

- (i). Node leaves.
- (ii). New Node joins.
- (iii). *Service Cache Message* expired.
- (iv). Node adjusts static property of *SD*.
- (v). De-registration. (Life Time in the Service Advertisement Message is set to zero)

The “Service Maintain Phase” can be summarized as follow:

- 1) Only the node which occurs changes will transmits the *Service Advertisement Message* to its serving *CN* for renewing the *Service Cache Message*.
- 2) The *CN* will renew its *Service Cache Message* and *SGI*.
- 3) The *CN* exchanges the updated *SGI* with neighbor *CNs* which in the 1-hop range.

The procedure diagram of “Service Maintain Phase” is shown in Figure 3.11, and the message flow is the same as “Service Advertisement Phase” has been shown in Figure 3.5.

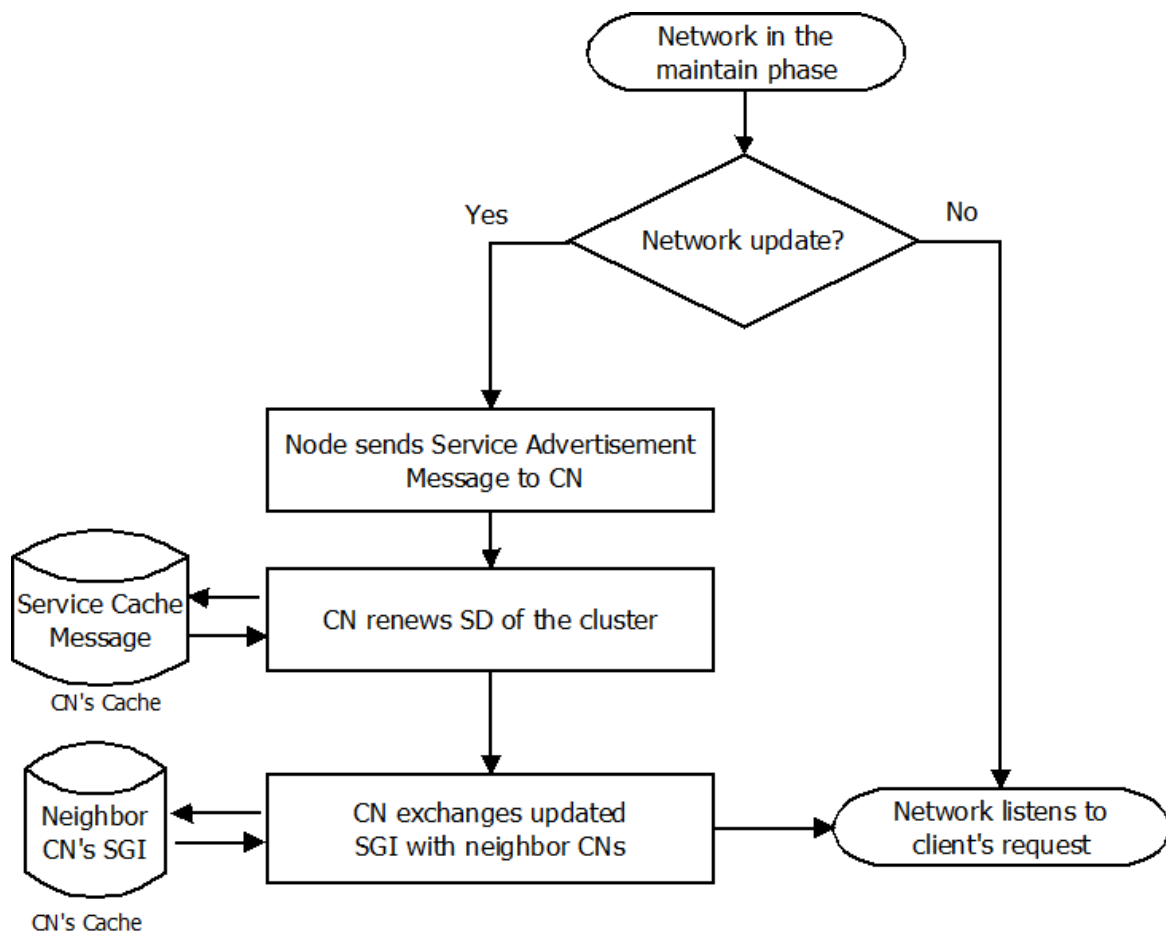
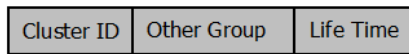
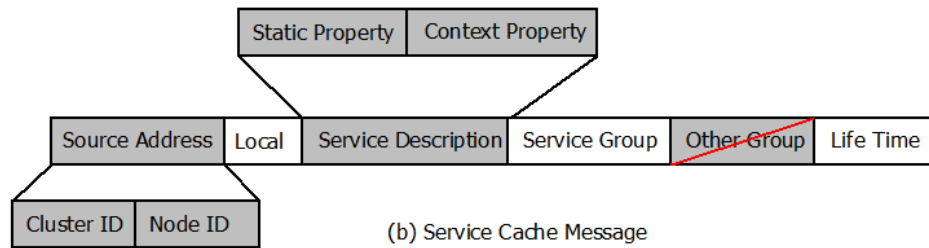
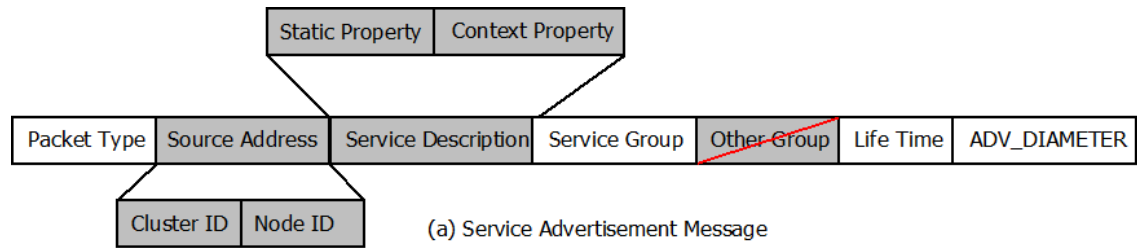


Figure 3.11 The procedure diagram of “Service Maintain Phase”

3.3.5 Message Type in CNGSD

In Figure 3.12 presents the detail of message type in CNGSD, the gray fields represent the modification in GSD filed and the white fields are identical with GSD.



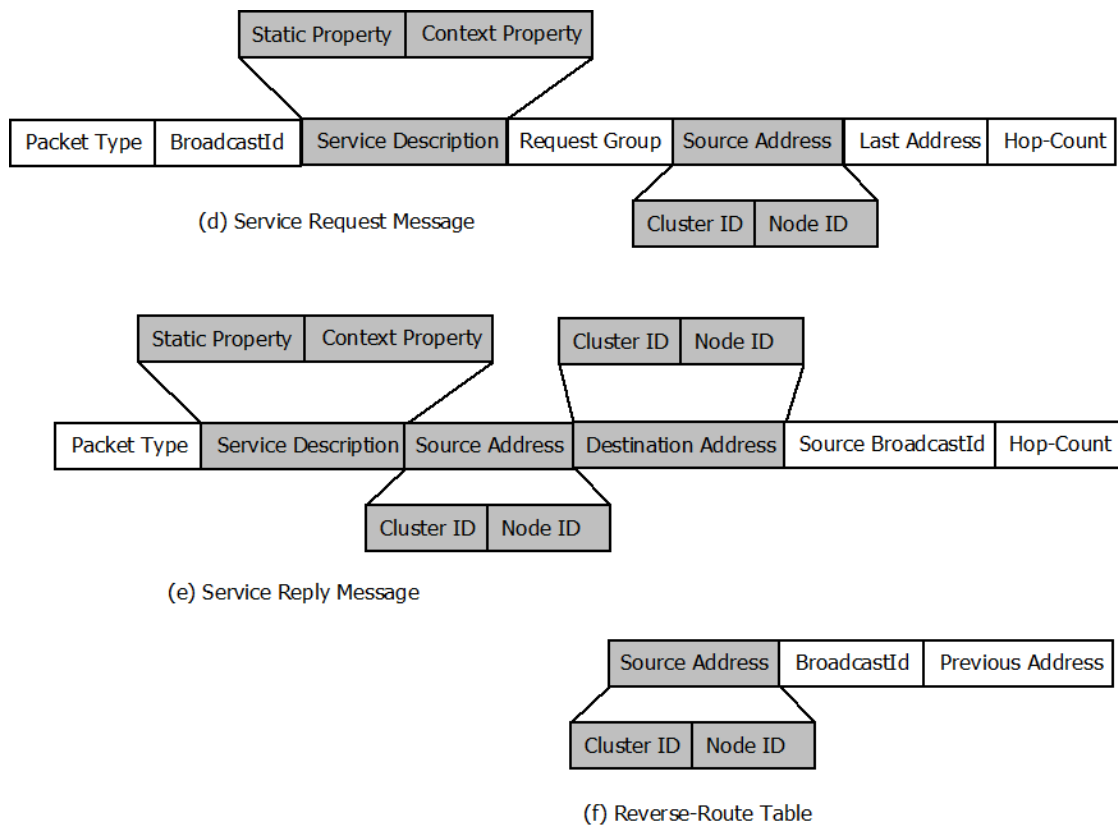


Figure 3.12 Message type in CNGSD

3.4 Discussion

In this chapter, we introduce the four phase in the CNGSD and explain every phase separately. There are “Network formation”, “Service Advertisement”, “Service Discovery Process”, and “Service Maintain”. We also provide the procedure diagram and message flow for reference. Last, we provide the message types which are used in CNGSD four phases.

Chapter 4 Simulation and Result Analyses

This chapter presents the simulation of CNGSD. First, simulation environment includes simulation design, assumption and performance metrics in section4.1. Next, we evaluate the performance of CNGSD and compare with GSD in section4.2. Finally, there is a discussion in the end of this chapter.

4.1 Simulation

In this section, we introduce the simulation environment and make some assumption in our experiment. Next, we have several metrics for evaluating the GSD and CNGSD.

4.1.1 Simulation Scenario

Today a career fair hold in open site, there are service information counters and company vendors which are classified in several service group including engineer, computer sciences, commerce.....etc. Besides, these vendors are created by ad-hoc. Assume that there are many job seeker carrying smart phones on site. In order to quickly find the interesting company vender and finish the resume or search the service information counter, the worker in every vendor have laptop or pad to provide data services. There is a job seeker named James who entered the site near service information counter and he wants to search some computer sciences vendors. By using his smart phone and input the service description, service requirement and service group, James can find the desire company and get service reply.

4.1.2 Simulation Environment

We implemented our approach and GSD in the NS-3 simulator which is a discrete-event network simulator and open environment for network education and research. In addition, it supports research on both IP and non-IP based network including models for Wi-Fi, WiMax, LTE for network 1 and 2 layers. Besides, it supports variety of static and dynamic routing protocols, for instance, OLSR and AODV for IP-based applications[18]. For the sake of above feature in NS-3, we select it as our implementation environment for simulating the related experiments. Table 4.1 displays the simulation environment.

Table 4.1 Simulation environment

Simulation Environment	
Hardware Environment	<ul style="list-style-type: none">• Intel Core i5 CPU M430 @ 2.27GHz 2.27GHz• 2 GB RAM
Software Environment	<ul style="list-style-type: none">• Linux (Ubuntu)• NS-3 (Network Simulator)
Program Language	<ul style="list-style-type: none">• C++ / Python

4.1.3 Simulation Setup

The simulations were performed in NS-3, and used the wireless model with dynamic routing protocol AODV. Simulation set the nodes from 10 up to 50 and distributed in a random radius of disc from 30m to 800m. Simulation time up to 1200 seconds for simulating the stable situation, the frequency of hello messages is 1-second for all nodes. For GSD, the advertisements message diameter is 1-hop. Thus, the broadcast range is limited in 1-hop. Table 4.2 shows the common simulator parameters in the experiments.

Table 4.2 Simulation parameters

Simulation Parameter	
Network Area	Random radius in a disc 30m up to 800m
Number of nodes	10 up to 50
Mac Layer	802.11
Routing Layer	AODV
Simulation Time	100 seconds to 1200 seconds
Hellos-Update-Rate	1 seconds

4.1.4 Simulation Assumption

We divide the nodes into several clusters and every node has its service group initially. These nodes are distributed randomly, thus in every cluster has several service group. Because NS-3 doesn't support the clustering models, we need to setup the cluster manually and the cluster nodes are determined. The simulation focus on the service discovery process phase and the target service exists in the network.

4.1.5 Simulation Metrics

There are three metrics for evaluating the performance of CNGSD and GSD. The descriptions are as follows:

1) Service Responds Time (SRT): First received RREP (Route Reply) time - First transmitted RREQ (Route Request) time.

2) Packet Lost Ratio (PLR):
$$\frac{\text{Total numbers of lost packets}}{\text{Total numbers of transmitted packets}}$$

3) Total number of hello message transmitted (THMT): Total number of hellos transmitted by all nodes in the network for maintaining the topology.

4.2 Results and Analyses

In this section, we evaluate the performance of CNGSD and GSD by using the previous metrics: service responds time (SRT), packet lost ratio (PLR), and total hello message transmitted (THMT).

SRT

Figure 4.1 represents the responds time of service discovery process. When increasing the number of nodes, the responds time of both GSD and CNGSD don't have rapid growth. Because of the intelligence routing, both of them uses the service group to forward the service request. However, CNGSD exploits *CN* to reduce the transmitted packets and has shorter responds time in the simulation. The average responds time of GSD is 46.06 seconds and CNGSD is 25.56 seconds.

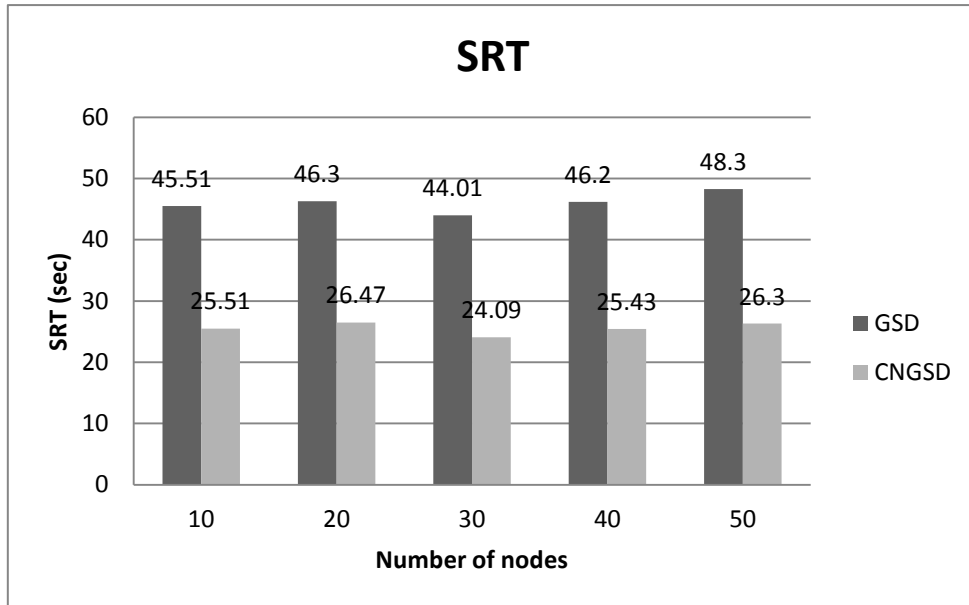


Figure 4.1 Responds time of service discovery process

PLR

Figure 4.3 displays the packet lost ratio; the less lost the better efficiency in service discovery process. Through the chart we can realize that both GSD and CNGSD are stable service discovery approach. As simulation time increase, the packet lost ratio has no fluctuations. And the result shows that the average packets lost ratio of GSD is 15.83% and CNGSD is 11.5%.

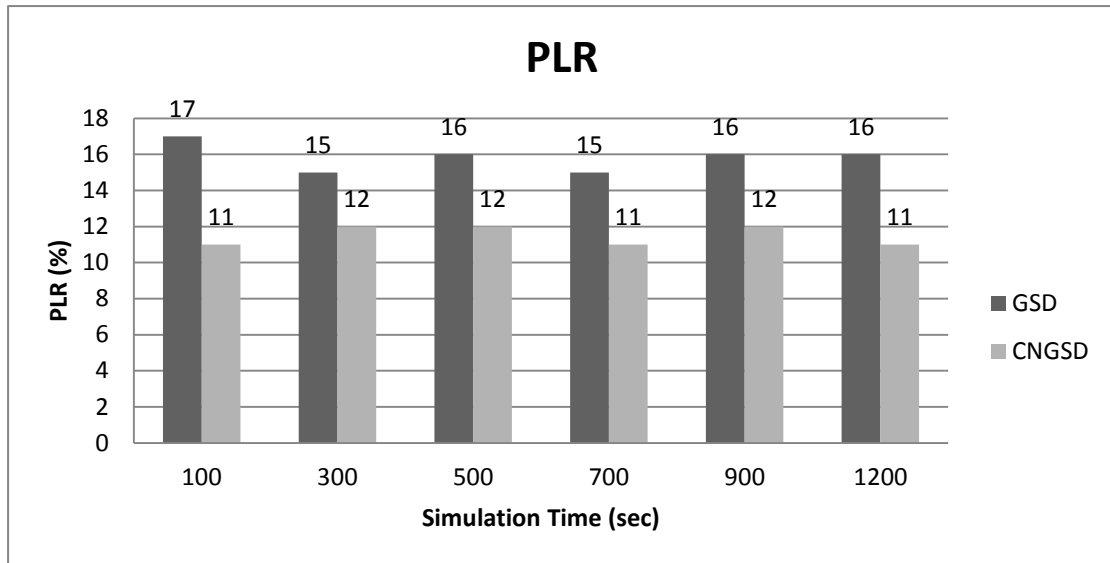


Figure 4.2 Packets lost ratio

THMT

Figure 4.4 shows the total number of hellos transmitted by all nodes in the network for maintaining the topology. It represents the network overhead of GSD and GNGSD. As time goes by, the transmitted packets will increase. But CNGSD greatly reduce the rate of increase. Due to the service advertisements are sent to only *CNs* rather than sent to all neighbor nodes. Moreover, *CNs* are responsible for delivery most of service request and reduce redundant packets transmission as well. The average packets transmitted of GSD is 8261 and CNGSD is 2372.

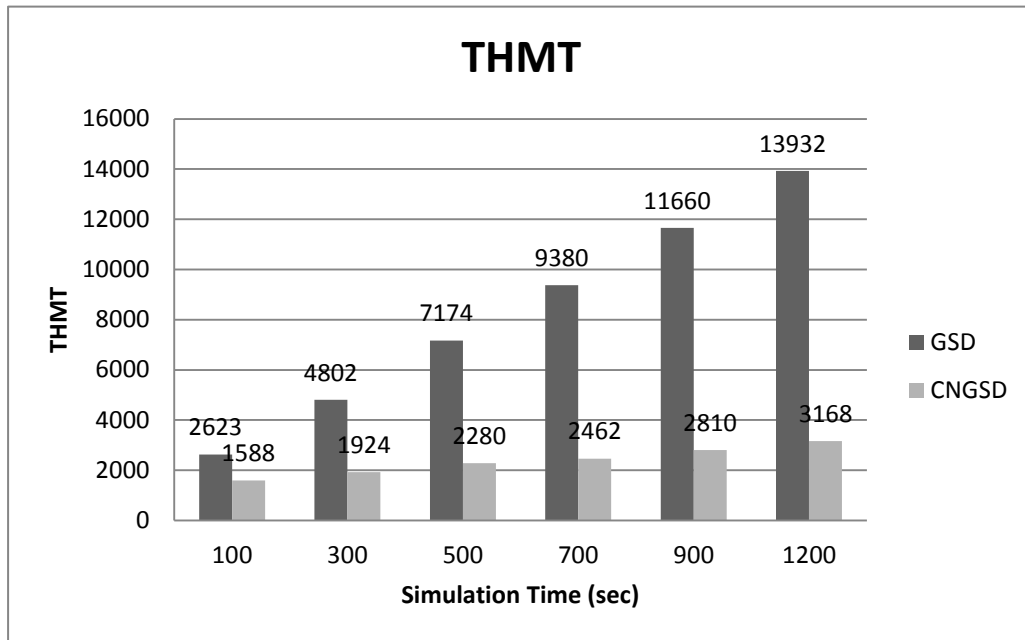


Figure 4.3 Total number of hello message transmitted

PLR and Distance of nodes

Figure 4.5 shows the impact of nodes distance, and we can see that when increasing the distance to 100m the packet delivery ratio begins to decline. Once the distance is 800m, the lost ratio is 47%. Therefore, the previous simulations are all in the distance under 50m for having the better performance. However, in our scenario the radius 50m in a disc is enough.

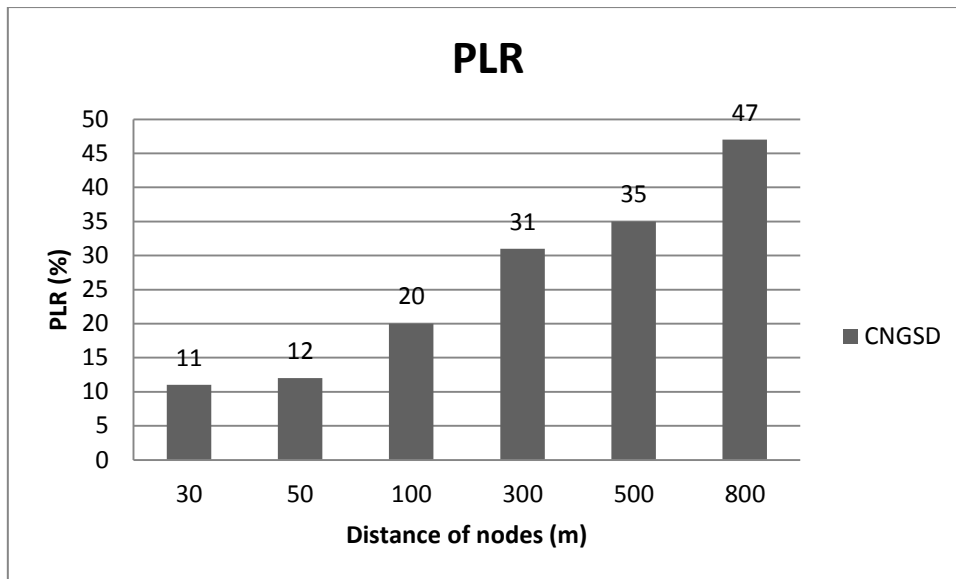


Figure 4.4 Distance of nodes impact the performance

4.3 Discussion

In our approach, the response time is shorter than GSD due to the reduction of redundant packets forwarded. We use *CN* to forward most of the traffic, and store cluster service description messages. As a result, services can be found faster. CNGSD enhances the average by 44.5%

Moreover, the packet loss ratio is lower than GSD. It means CNGSD is stable and efficient. The average packet loss ratio of CNGSD is 11.5% lower than 15.83% of GSD. Besides, the total overhead of our approach is significantly lower than GSD, due to the decrease in data transmission between nodes. CNGSD decreases 71.2% network overhead.

Although the distance will impact the packet loss ratio of CNGSD, the range of major existing wireless transmission technologies does not exceed 100m. Thus, it is not a big problem for us. In conclusion, CNGSD is an efficient service discovery approach in MANET.

Chapter 5 Conclusions and Future Works

In this chapter, we will give the conclusion and recommend future works for this paper. In section 5.1, we give the conclusions for CNGSD and some future works can be done based on this paper is discussed in section 5.2.

5.1 Conclusions

In this paper, we proposed an efficient approach; CNGSD-Context Node enhanced Group-based Service Discovery Protocol, using cluster and group information to intelligence forwarding the service request packets. It is based on GSD and improves the service responds time and the overhead of network by using the clusterhead- Context Node. Nodes in the network send the service description only to the nearest Context Node rather than all the neighbors. In this way, CNGSD can reduce the redundant packets transmitted, also it can reduce the transmission of hellos which for maintaining the network topology.

Context Node is responsible for most of traffic in the network. Service request and service reply can be transmitted through Context Node, greatly reducing the number of node forward. The benefit of using Context Node reflects on the responds time of service discovery process and total hello messages transmitted. The simulation results show that CNGSD enhances the service responds time with an average 44.5% and decreases 71.2% network overhead. It represents that CNGSD has better performance than GSD in service discovery process.

Moreover, the service descriptions of CNGSD have dynamic attributes and uses *Service Weight Table (SWT)* to rank the score of matching services. Clients can find out the most suitable services according to their preferences. It provides the solution for selecting identical services in the network.

5.2 Future Works

In this section, we propose the future works which can put more effort to improve based on this paper. One is the leader election algorithm used in “Network Formation Phase”, the other is context aware mechanism used in the context node. The detail of these issues will be described in the following.

(1) Leader Election Algorithm Analysis in MANET

In this paper, we do not pay attention on clustering algorithm and the clusterhead is elected based on the criteria [17]. We focus on the service discovery process. However, clustering is an important issue in mobile ad-hoc network; a good leader election algorithm can affect the performance. Thus, clustering is a big issue to research and analysis if CNGSD wants to improve in the future.

(2) Evaluate Quality of Service (Qos) Metrics

With the service quality requirement, context aware is a potential direction to develop in the future. In this paper, we just use context information with weight mechanism to determine the better matching service. However, we do not evaluate the Qos metrics, for instances, throughput, errors, jitter and so on. We hope this issue could implement in the future.

REFERENCES

- [1] Mian, A.N., R. Baldoni, and R. Beraldi, *A survey of service discovery protocols in multihop mobile ad hoc networks*. Pervasive Computing, IEEE, 2009. **8**(1): p. 66-74.
- [2] Lee, C. and S. Helal, *Protocols for service discovery in dynamic and mobile networks*. International Journal of Computer Research, 2002. **11**(1): p. 1-12.
- [3] Cho, C. and D. Lee, *Survey of service discovery architectures for mobile ad hoc networks*. term paper, Mobile Computing, CEN, 2005. **5531**.
- [4] Hughes, E., et al. *Service recommendation using SLP*. in *IEEE International Conference on Telecommunications 2001*. 2001.
- [5] Song, H., et al. *UPnP-based sensor network management architecture*. in *Proc. ICMU Conf.* 2005.
- [6] Cuddy, S., M. Katchabaw, and H. Lutfiyya. *Context-aware service selection based on dynamic and static service attributes*. in *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005), IEEE International Conference on*. 2005. IEEE.
- [7] Chakraborty, D., et al. *GSD: A novel group-based service discovery protocol for MANETS*. in *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*. 2002. IEEE.
- [8] Chakraborty, D., et al., *Toward distributed service discovery in pervasive computing environments*. Mobile Computing, IEEE Transactions on, 2006. **5**(2): p. 97-112.
- [9] Ververidis, C.N. and G.C. Polyzos, *Service discovery for mobile ad hoc networks: a survey of issues and techniques*. Communications Surveys & Tutorials, IEEE, 2008. **10**(3): p. 30-45.
- [10] Gupta, R., S. Talwar, and D.P. Agrawal, *Jini home networking: a step toward pervasive computing*. Computer, 2002. **35**(8): p. 34-40.
- [11] Perkins, C., et al., *Service location protocol*. RFC2165, 1997.
- [12] Yau, S. and G. Pohare, *An Efficient Approach to Situation-Aware Service Discovery in Pervasive Service Computing Environments*. Ubiquitous Intelligence and Computing, 2009: p. 68-82.
- [13] Ilka, M., M.c. Niamanesh, and A. Faraahi. *A context-aware and group-based service discovery in mobile ad hoc networks*. in *Systems and Informatics (ICSAI), 2012 International Conference on*. 2012. IEEE.
- [14] Rahman, M.M. and A. Nahar, *Modified bully algorithm using election commission*. arXiv preprint arXiv:1010.1812, 2010.
- [15] Arghavani, A., E. Ahmadi, and A. Haghighat. *Improved bully election*

- algorithm in distributed systems.* in *Information Technology and Multimedia (ICIM), 2011 International Conference on.* 2011. IEEE.
- [16] Park, S.-H., Y. Kim, and J.S. Hwang. *An efficient algorithm for leader-election in synchronous distributed systems.* in *TENCON 99. Proceedings of the IEEE Region 10 Conference.* 1999. IEEE.
- [17] Chatterjee, M., S.K. Das, and D. Turgut, *WCA: A weighted clustering algorithm for mobile ad hoc networks.* *Cluster Computing*, 2002. 5(2): p. 193-204.
- [18] NS-3, <http://www.nsnam.org/overview/what-is-ns-3/>.

