

AUTONOMOUS PRIORITY BASED ROUTING FOR ONLINE SOCIAL  
NETWORKS

A dissertation submitted  
to Kent State University in partial  
fulfillment of the requirements for the  
degree of Doctor of Philosophy

by

Salem Othman

August 2018

Dissertation written by

Salem Othman

B.S., University of Omar Mukhtar, Libya, 2001

M.S., Attahadi University, Libya, 2006

Ph.D., Kent State University, 2018

Approved by

\_\_\_\_\_, Chair, Doctoral Dissertation Committee  
Javed I. Khan

\_\_\_\_\_, Members, Doctoral Dissertation Committee  
Feodor Dragan

\_\_\_\_\_  
Hassan Peyravi

\_\_\_\_\_  
Brian Castellani

\_\_\_\_\_  
Xinyue Ye

Accepted by

\_\_\_\_\_, Chair, Department of Computer Science  
Javed I. Khan

\_\_\_\_\_, Dean, College of Arts and Sciences  
James L. Blank

## TABLE OF CONTENTS

<b>LIST OF FIGURES .....</b>	<b>IX</b>
<b>LIST OF TABLES .....</b>	<b>XII</b>
<b>DEDICATION.....</b>	<b>XIII</b>
<b>ACKNOWLEDGEMENTS .....</b>	<b>XIV</b>
<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 Motivation .....	1
1.2 Applications: .....	3
1.3 Social Routing Process .....	4
1.4 The Stratified Privacy Model .....	6
1.5 Problem Description.....	9
1.6 Solution Approach.....	10
1.7 Contributions.....	11
1.8 Assumptions .....	13
1.9 Complexities and Difficulties of the Study .....	13
1.10 Dissertation Organization.....	15
<b>CHAPTER 2 BACKGROUND AND RELATED WORKS ON ROUTING AND PRIVACY ISSUES IN OSN.....</b>	<b>16</b>
2.1 The Human Dynamics Models.....	16
2.2 Social Routing .....	18
2.3 Quantifying the importance of individuals in OSNs .....	23
2.4 Privacy Metrics: .....	24

<b>CHAPTER 3 SOCIAL ONLINE ROUTING (SOR) PROTOCOL OVERVIEW ...</b>	<b>26</b>
3.1 What is SOR protocol?.....	26
3.2 SOR Basics.....	26
3.3 Peer-to-peer Social Consumer/Provider Model .....	27
3.4 SOR Messages.....	28
3.4.1 I-need Message Structure .....	28
3.4.2 I-have Message Structure .....	34
3.4.3 I-thank Message Structure.....	36
3.4.4 I-like/dislike Message Structure.....	37
3.4.5 I-ack Message Structure .....	38
3.5 SOR Tables .....	38
3.5.1 Messages Table (MeT).....	38
3.5.2 Forwarding Table (FoT).....	39
3.5.3 Routing Table (RoT) .....	40
3.5.4 Self-Interest Table (SiT).....	40
3.5.5 Peer-like/dislike Table (P2T) .....	41
3.6 Policies .....	42
3.6.1 Propagation Policy .....	42
3.6.2 Self-Interest Policy .....	42
3.6.3 Peer-Interest Policy .....	43
3.6.4 Policy-based I-need Message Checkup Process.....	43
3.6.5 I-like/dislike Module .....	44

3.7	Attribute-based Languages.....	45
3.7.1	Link-Attribute-based Propagation Language (LAP):.....	46
3.7.2	Node-Attribute-based Propagation Language (NAP): .....	47
3.8	Forwarding and Routing.....	48
3.9	Stratified Privacy.....	48
<b>CHAPTER 4 REACHABILITY AND EFFICIENCY OF SOR .....</b>		<b>50</b>
4.1	Forwarding .....	50
4.1.1	I-need Module .....	51
4.1.2	I-have Module .....	53
4.1.3	I-thank Module.....	54
4.1.4	I-ack Module .....	56
4.2	Human Queue Model .....	61
4.3	Social-based Routing.....	62
4.3.1	Topology aware Shortest-Path-Based Routing Algorithm (CSP).....	67
4.3.2	Social-Priority-Based Routing Algorithm (SPB <sub>S</sub> ) .....	67
4.3.3	Queue aware Social-Priority-Based Routing Algorithm (SPB <sub>D</sub> ).....	68
4.4	Experiment Validity .....	69
4.4.1	Choice of dataset .....	69
4.4.2	Choice of algorithms .....	70
4.5	Experiments on the Efficiency of the Routing Protocol .....	71
4.6	Conclusion.....	77
<b>CHAPTER 5 STUDY ON THE PRIVACY PRESERVATION ABILITY OF SOR</b>		<b>78</b>

5.1	Privacy Background and Terminology .....	78
5.2	Privacy in Social Networks .....	81
5.3	SOR Functionality and Privacy Issues .....	82
5.4	Stratified Privacy Mechanism in SOR .....	83
5.5	Peer Privacy Request:.....	86
5.6	Attacker Model.....	87
5.6.1	Definitions and Assumptions .....	87
5.6.2	Background Knowledge .....	90
5.6.3	Goal of the Attack .....	92
5.7	Theoretical Attack analysis using Proxima.....	93
5.7.1	Identity Anonymity of Service Consumer .....	93
5.7.2	Proxima Matrices .....	96
5.7.3	Proxima Distributions .....	100
5.7.4	Proxima Degree of Anonymity .....	103
5.7.5	Privacy Analysis and Evaluation.....	106
5.8	Validity of Privacy Requirements .....	119
	<b>CHAPTER 6 SOCIAL PRIORITY.....</b>	<b>125</b>
6.1	Related Work.....	126
6.2	Social Characteristics .....	126
6.2.1	Centralities and Gender.....	127
6.2.2	Social Characteristics' Availability and Accessibility .....	129
6.3	Preliminaries.....	130

6.3.1	The Used Notations .....	130
6.3.2	Social Priority .....	131
6.3.3	Problem Definition .....	132
6.3.4	Complexity of the Problem .....	132
6.4	General Social Priority Framework .....	133
6.4.1	Social Metrics .....	133
6.5	Social Priority Computation .....	136
6.5.1	Constructing Social Priority Matrix (SPA) .....	136
6.5.2	Social Priority Matrix decomposition .....	137
6.6	Analysis of social priority in some sample real networks .....	139
6.6.1	Dataset Descriptions .....	139
6.6.2	Analysis .....	140
6.7	Discussion .....	142
6.8	Conclusion .....	144
	<b>CHAPTER 7 ONLINE SOCIAL NETWORK SIMULATOR .....</b>	<b>145</b>
7.1	OMNeT++ .....	145
7.2	Simulator Architecture .....	145
7.3	Online User Interface .....	149
	<b>CHAPTER 8 CONCLUSIONS AND FUTURE WORK .....</b>	<b>153</b>
8.1	Contribution .....	153
8.2	Limitations and Future Work .....	156
8.2.1	Incentivization .....	156

8.2.2	Misbehaving .....	156
8.2.3	Privacy of advertisement .....	156
8.2.4	Security .....	157
8.2.5	Social Priority .....	157
8.2.6	Reachability .....	158
8.2.7	Routing Loops Prevention .....	158
8.2.8	Application .....	159



## LIST OF FIGURES

Figure 1: Peer-to-peer Social Consumer/Provider Model .....	28
Figure 2: I-need Message Structure .....	28
Figure 3: I-have Message Structure .....	35
Figure 4: Time Dependencies between CT, RDe, SST, and SET Fields .....	35
Figure 5: I-thank Message Structure .....	36
Figure 6: I-like/dislike Message Structure .....	37
Figure 7: I-ack Message Structure .....	38
Figure 8: Policy-based I-need Message Checkup Process .....	44
Figure 9: Flow Diagram of I-like/dislike Module .....	45
Figure 10: I-like/dislike Module .....	45
Figure 11: Flow Diagram of I-need Module .....	52
Figure 12: Flow Diagram of I-have Module .....	54
Figure 13: Flow Diagram of I-thank Module .....	55
Figure 14: Flow Diagram of I-ack Module (Send-with-Hope) .....	56
Figure 15: Flow Diagram of I-ack Module (Send-with-Knowledge) .....	57
Figure 16: Flow Diagram of I-ack Module (Extend-with-Knowledge) .....	57
Figure 17: Windows and forwarders types of Extend-with-Knowledge .....	61
Figure 18: Node Anatomy .....	64
Figure 19: End-To-End Routing Delay .....	74
Figure 20: Network Throughput .....	75

Figure 21: Hop Count .....	75
Figure 22: Steps for getting Node and Group Global IDs .....	84
Figure 23: Using Defensive Mechanisms by Alice .....	87
Figure 24: Illustration of Victim, First-Forwarder, Last-Forwarder, Attacker Nodes and the Attack Path.....	88
Figure 25: Elena’s Circles of Friends .....	89
Figure 26: Bob’s Background Knowledge (Local and Null Identity Graphs).....	91
Figure 27: Goal of Bob as an attacker .....	93
Figure 28: Computation of Proxima Matrix .....	97
Figure 29: Small Social Network (SSN).....	97
Figure 30: (A) Fixed Proxima Matrix of SSN. (B) Scatter Plot of the Fixed Proxima Matrix of SSN.....	98
Figure 31: (A) Integrated Proxima Matrix of SSN. (B) Scatter plot of the Integrated Proxima Matrix of SSN.....	99
Figure 32: Fixed Proxima distribution of SSN .....	102
Figure 33: Integrated Proxima distribution of SSN .....	103
Figure 34: Degree of Anonymity vs Distance of Fixed and Integrated Proxima .....	104
Figure 35: Fixed Proxima Degree of Anonymity of Offline Social Networks.....	110
Figure 36: Integrated Proxima Degree of Anonymity of Offline Social Networks.....	111
Figure 37: Fixed Proxima Degree of Anonymity of Online Social Networks .....	116
Figure 38: Integrated Proxima Degree of Anonymity of Online Social Networks .....	117
Figure 39: Social Characteristic Vectors and In-Social Priorities .....	131

Figure 40: Steps of Social Priority Computation .....	136
Figure 41: Histograms of Social Priorities of DS-1 .....	141
Figure 42: Histograms of Social Priorities of DS-2.....	141
Figure 43: Histograms of Social Priorities of DS-3.....	142
Figure 44: Histograms of Social Priorities of DS-4.....	142
Figure 45: Receptionist Parameters .....	146
Figure 46: Generator Parameters .....	146
Figure 47: Router Parameters .....	147
Figure 48: Forwarder Parameters.....	147
Figure 49: Executer Parameters .....	147
Figure 50: The internal interconnection of node’s components .....	148
Figure 51: Small Social Network.....	149
Figure 52: Online Social Network Simulator architecture .....	150
Figure 53: Online User Interface (connecting to a server) .....	151
Figure 54: Online User Interface (parameters configuration and result) .....	152

## LIST OF TABLES

Table 1: Queue Parameters .....	65
Table 2: Statistical information of Google+ datasets.....	73
Table 3: Privacy Fields of the I-need Message.....	86
Table 4: Global Maximum and Minimum of Degree of Anonymities .....	106
Table 5: Statistical information of the offline social networks.....	108
Table 6: Fixed Proxima Distributions of Offline Social Networks .....	109
Table 7: Integrated Proxima Distributions of Offline Social Networks .....	109
Table 8: Global Maximum and Minimum of Fixed Proxima Degree of Anonymity.....	112
Table 9: Global Maximum and Minimum of Integrated Proxima Degree of Anonymity .....	113
Table 10: Statistical information of the Online Social Networks .....	114
Table 11: Fixed Proxima Distributions of Online Social Networks .....	114
Table 12: Integrated Proxima Distributions of Online Social Networks .....	115
Table 13: Global Maximum and Minimum of Fixed Proxima Degree of Anonymity...	118
Table 14: Global Maximum and Minimum of Integrated Proxima Degree of Anonymity .....	118

## **DEDICATION**

This dissertation is dedicated to my always encouraging, ever faithful parents, my brilliant and outrageously loving and supportive wife, Fatema, our sweet, and lovely beautiful kids, Ahmed, Alla, and Aya.

Thank you for your love, support, and encouragement.

## **ACKNOWLEDGEMENTS**

First, I would like to express sincere appreciation to my advisor, Dr. Javed I. Khan, who has constantly supported me with his insight and patience. Without the excellent research environment provided by Dr. Khan, I could have not completed my doctoral degree. In addition, I am thankful to my doctoral committee members, Dr. Hassan Peyravi, Dr. Feodor F. Dragan, Dr. Brian Castellani, and Dr. Xinyue Ye, for agreeing to be on the board, reviewing my works, and their suggestions, criticism and support. I am also thankful to my school division, the Computer Science Department, for allowing me to conduct my research and providing any assistance requested.

I would like to thank my lovely wife, Fatema Nafa, who supported my kids and me and allowed us to share the time with her during her study. I would also like to thank my kids, Ahmed, Alla, and Aya for being patient when I was busy, and I did not have time for them. I would like to thank my friend, Mohamed Alhadji (may God bless his soul in peace), for supporting me in my Master's Degree in Sirte, Libya, and also here in the USA. I also would like to thank Marcy Curtiss, Jan Kotila, and Cheryl Cunnagin for being supportive and carrying about my life and education. I also would like to thank all my brothers and sisters for their encouragements and support. Last but not least, I would like to thank my friends and especially Basher Egnies, Farhat Embarak, Ali Rashed, Ali

Emgatif, Ali Albakush, Ahmad Fansha, and Abdulsalam Omar Alfrgany for their warm feelings and encouragement.

Salem Othman

August 2018, Kent, Ohio, USA

## CHAPTER 1

### **Introduction**

This first chapter introduces the motivation for the work, possible applications of social routing protocol; a framework for privacy options; the social routing process; the formal description of the problems that need to be solved, and the solution approaches; as well as the contributions, the assumptions, and difficulties encountered during this study; and the document layout.

#### **1.1 Motivation**

Complex Online Social Networks (OSNs) such as Facebook, LinkedIn, and Google+ are built based on meaningful social relations and are used for sharing comments, photos, and videos. However, users sometimes face situations that require interaction with people not directly connected to them (for example, looking for somebody to write a recommendation for a job at a particular company, searching for tutors, or looking for babysitters). In some cases, people may know exactly who can help, but cannot ask them directly because of privacy concerns, incentivization, and authority levels. In other cases, people may not even know who could potentially help them out. Despite the fact that OSNs platforms are designed to intuitively assist users, they lack the ability to quickly and directly connect their users to the types of people or experts they may seek.



In current OSNs, individuals can only interact with their adjacent neighbors. Any interaction beyond that with a desired person or persons requires a special routing algorithm for finding optimum paths to them. However, a standalone routing algorithm is not enough, and a protocol is needed to 1) propagate the needed services in the network, 2) share selective information for the routing algorithm to function, and 3) find the optimum paths to the desired people. Finding the optimum path is twofold: check if there is a path between two individuals in OSN (reachability), and check if it is the most efficient path (efficiency), which this study refers to as end-to-end routing delay. To ensure reachability, some information elements such as identity information and connectivity information must be shared. Other information elements like status information and priority information must be shared to ensure efficiency. However, sharing these four elements of information is a big privacy concern, and any routing protocol for OSNs must first give individuals an option to hide or disclose any of the information elements, and second, satisfy these privacy options.

Different individuals may have different privacy options. Therefore, providing the same level of privacy protection to each individual user may not provide adequate privacy protection, and in addition may cause people and platform owners to not use the routing protocol. Thus, different levels of protection should be defined for every individual user and incorporated into the routing protocol. This allows users to set how much information can be shared with others. Therefore, the main purpose of this dissertation is to design a protocol that can ensure reachability and efficiency while satisfying privacy options at the same time.

However, design and implementation of protocol for OSNs is not an easy task and requires privacy, security, and performance goals to be achieved. The protocol must be designed to be 1) *reliable*: where messages reach their intended destinations with high probability; 2) *efficient*: where messages reach their intended destinations with reasonable end-to-end latency and network overhead; 3) *scalable*: where the protocol is able to scale to a large number of participants; 4) *secure*: where messages are protected against malicious nodes, selfish nodes, and non-cooperative nodes; 5) *privacy-preserving*: where the protocol requires small amounts of information to be distributed in order to function; and 6) *distributed*: where messages are forwarded node-by-node and there is no central point of control. This dissertation focuses specifically on *efficiency* and *privacy*, leaving other requirements for future studies.

## **1.2 Applications:**

A routing protocol for OSNs can be implemented either as an application in current online social networks platforms (e.g., Facebook and Google Plus), or as a pure Peer-to-Peer system. Using such a protocol, individuals in OSNs can ask for service and stop receiving unwanted services. Such a protocol is able to support a large number of very specific online service applications including: 1) an online recommendation system: to ask people to write a statement describing the ability or expertise of an individual; 2) an online endorsement system: to ask people to endorse specific skills or expertise of an individual; 3) an online charity donations system: to ask an individual or an organization to donate a gift; and 4) an online advertisements system: to tell people about a product or service, and others. These applications are waiting for such a protocol to emerge. In these

ways, individuals will be able to reach desired people beyond adjacent neighbors and ask them for endorsements, donations, and other services without needing to add them to their circles.

### **1.3 Social Routing Process**

The social psychologist Stanley Milgram [Milgram, 1967; Travers & Milgram, 1969] studied message routing in real-world social networks. It refers to the problem of reaching a target node in a network through a short chain of intermediate nodes, where each node is only provided with local information. Milgram's studies involved a number of people randomly chosen from Nebraska and Kansas who were asked to route a letter (document) to the target (a lawyer in Boston). Milgram provided some personal information about the target, but placed the restriction that the participants could not forward the document directly to the target; rather, each participant could only advance the document by forwarding it to a single friend that he or she knew on a first-name basis, with the goal of reaching the target as quickly as possible [Barbella, Kachergis, Liben-Nowell, Sallstrom, & Sowell, 2007; Epstein, Goodrich, Löffler, Strash, & Trott, 2013; Lattanzi, Panconesi, & Sivakumar, 2011; Liben-Nowell, Novak, Kumar, Raghavan, & Tomkins, 2005]. Milgram found that participants were able to efficiently route messages by using only local information and simple social information about targets, such as ethnicity, occupation, name, and location.

Although no social routing protocol currently exists, a number of researchers have investigated various social routing processes that exist in natural society to understand the mystery of how social routing works [Banerjee & Basu, 2008; Liben-Nowell et al., 2005;

Michlmayr, Pany, & Kappel, 2007; Schurgot, Comaniciu, & Jaffres-Runser, 2011; Suthaputchakun & Sun, 2011].

For example, Liben [Liben-Nowell et al., 2005] and Adamic [Adamic & Adar, 2005] showed that routing strategies that utilize the geographic distance among people in real-world social networks did not create a very effective routing solution. In their studies, they also found routing strategies to be more effective when the data in participant routing was complete and the structures of participants were well defined. On the other hand, Jia and his colleagues [S. Jia, St Juste, & Figueiredo, 2013] explored the impact of leveraging two social dimensions for routing in a social graph. They were able to show that the routing performance improved with two social dimensions: geographic location and personal interest. They mentioned that no definite decentralized algorithm based on local information and individual considerations has yet been found for efficient social routing in most naturally occurring social networks, especially those created by online social networking sites such as Facebook and Google+.

Similar work has been done to identify what factors play a role in routing messages between people. Dodds and his colleagues conducted an experimental study by asking people to forward a message through acquaintances to one of eighteen target persons from thirteen countries they were unfamiliar with [Dodds, Muhamad, & Watts, 2003]. They found that to be successful, social routing does not require highly connected hubs. In contrast, professional ties with intermediate to weak strength are enough for successful social searches.

Milgram and a large amount of work inspired by his experiment have investigated the issue of reachability in a social network, yet these are far from creating a fully usable social routing protocol. Furthermore, a limited number of social features have been used to guide routing and privacy options, but privacy requirements have not been taken into consideration. Because of these limitations, none of these algorithms were really used in current OSNs. This study attempts to bridge the gap by designing a routing protocol that allow information to be shared to make social routing possible in OSNs, while taking privacy into consideration. Some research questions in this context are:

1. Is it possible to construct a protocol that will satisfy reachability, efficiency and privacy in order to apply social routing in current OSNs platforms such as Facebook and Google+?
2. Is it possible to quantify the social priority between two individuals in OSNs by using social characteristics?
3. Is it possible to provide defensive mechanisms that individuals can use to protect their Identity information?
4. How can the degree of anonymity of individuals in OSNs be quantified?

#### **1.4 The Stratified Privacy Model**

Any routing protocol must exchange a set of *information elements*. Social routing requires differentiated sharing of each of these elements. The sharing modes of these information elements depend on the semantics of the information, as well as the *role* of the individuals in the society. A wide variety of information elements and roles can exist in an OSN. However, the four minimum pieces of information elements necessary are

Identity information, Connectivity information, Status information, and Priority information. Identity information and Connectivity information are needed to ensure reachability, while Status information and Priority information are needed to ensure efficiency. The intrinsic difference between Status information and Priority information is that Status information is globally disclosable, while Priority information is selectivity disclosable. At this time, several definitions are necessary in order to better understand the relationships between the four information elements, privacy options, and individuals, and their roles in an OSN:

- **Identity information:** it is a unique identifier of each node in the OSN. It should be disclosed up to varied degrees in the OSN. In a regular network, every node should identify itself; on the contrary, in an OSN some nodes share their identities with other nodes, while others do not. Although a large set of variations may exist, in this model only four options are considered: Real Identity, Globally unique Pseudo Identity, Locally unique Pseudo Identity, and Null Identity (the four identities will be explained in detail in chapter five). Using these identities:
  1. If individual in OSN sends a message and chose to be hidden from others it must be hidden.
  2. If individuals in OSN sends a message and chose to be anonymized from others it must be anonymized.
- **Connectivity information:** it defines who is connected to whom. In OSN this type of information is not freely exchangeable. There are two possible options for sharing Connectivity information: 1) a node that allows its adjacent neighbor to

disclose that they are connected, and 2) a node that does not allow its adjacent neighbor to disclose that they are connected.

- **Status information:** it tells how busy an individual node happens to be. Like a conventional network, a certain aspect of a social node behaves like a queue. In this study, for example, a social node is modeled as a queue. A situation may occur where nodes are busy and cannot receive more load because their queues may be overloaded, or they may manually or automatically utilize their requests which can decrease or increase arriving messages rates. Queue parameters of the M/M/1 queue in this study are arriving rate ( $\lambda$ ), service rate ( $\mu$ ) and queue size (L). This information can be globally shared in the network. However, individuals can decide to hide or disclose their status information. The queue model is explained in detail in Chapter Four.
- **Priority information:** it determines priority levels between individuals in OSNs. Unlike a conventional network, a social node uses other considerations which also impact performance. When messages are received, not every individual gets its service with the same priority. For example, family members may have a higher level of importance than others, and different friends have different priorities assigned to them based on their social centralities. Normally, not all individuals in OSN are willing to share this kind of information. Therefore, priority information can be either disclosed or hidden. In the latter case, neighbors can estimate it by using social characteristics. Chapter Six presents a model for estimating priority between two individuals.

- **Privacy Options:** Of the one hundred and twenty-eight possible privacy options, four are associated with Identity information; two with Connectivity information, eight with Status information (two options for each of the three parameters), and two with Priority information. Individuals can choose from these privacy options. However, social routing can be achieved with different end-to-end routing delays using the disclosed information elements in the case of most open choices where all information is shared. There are also the most restrictive choices (*restrictive privacy*) where one chooses null identity, does not allow for the propagation of connectivity, and shares no queue or priority information. A good protocol still needs to ensure reachability in the first case, and near optimum performance in the second case. There are also cases in between these privacy choices (e.g. priority information is shared), where performance should degrade gracefully.

## 1.5 Problem Description

Given a directed graph  $G = (V, E)$  that represents an online social network, i.e., each vertex  $u$  in  $V$  represents an individual and an edge  $(u, v)$  in  $E$  represents some relation between  $v$  and  $u$ . Each node  $u \in V$  has *information elements*: Identity information  $e_i$ , Connectivity information  $e_c$ , Status information  $e_s$ , and Priority information  $e_p$ . Each node  $u \in V$  can disclose or hide any of the information elements (as described in the Stratified Privacy Model). Each node  $u \in V$  is associated with 1) a *social characteristic vector*  $x^u \in \mathbb{R}^k$ , where each element of the vector  $x^u(j) \in D_j$ . The notation  $x^u(j)$  denotes the value of  $j^{\text{th}}$  social characteristic associated with node  $u$ . 2) a *social characteristic matrix*  $A^u \in \mathbb{R}^{dk}$ , where  $d=|L_{\text{nei}}(u)|$  is the number of  $u$ 's adjacent neighbors and  $k$  is the



number of social characteristics associated with nodes in the set  $L_{\text{nei}}(u)$ .  $A_{(i),(c)}^u \in \mathbb{R}^{1 \times k}$  denotes the  $i$ -th row of matrix  $A^u$  which corresponds to a vector  $x^i$  of an adjacent neighbor  $i$ .  $A_{(i),(j)}^u \in \mathbb{R}^{d \times 1}$  denotes its  $j$ -th column which corresponds to the values of social characteristic  $C_j$  for node  $u$ 's adjacent neighbors.  $A_{(i,j)}^u$  refers to the  $i^{\text{th}}$  adjacent neighbor's social characteristic value of  $C_j$ . Each node  $u \in V$  has queue: a discipline  $B_u(t)$  (Priority), and queue parameters  $\lambda_u^f$  (the number of messages arriving at  $u$ 's queue per unit time),  $\mu_u^f$  (the number of messages departing the  $u$ 's queue per unit time), and  $L_u^f(t)$  (the number of messages in  $u$ 's queue at time  $t$ .) The goal is to achieve the following:

- For each node  $u$  use  $x^u$  and  $A^u$  to find the social priorities between node  $u$  and each  $v \in S(u)$ , where  $S(u)$  is  $u$ 's adjacent neighbors.
- Given source and target individuals  $u, v \in V$ , find the optimum path  $P = \langle u, u_0, \dots, v \rangle$  using the available information of  $(e_i, e_c, e_s, e_p)$ .
- For a node  $u \in V$  find how likely it can be identified as the sender of message  $m$  given that  $u$  chooses to be anonymized or hidden and the adversary at distance  $d$  from  $u$  in  $G$  and receives the message  $m$ .

## 1.6 Solution Approach

This study focuses on two main topics. Firstly, the work looks at designing a protocol for social routing in OSNs. It is an important concern to provide messages to propagate services and share selective individual information. Secondly, the study aims to provide methods that satisfy privacy options of Identity information. The study's objectives are detailed as the following:

- Provide messages structures for carrying information, and table structures for maintaining an entry for each message.
- Provide Attribute-based languages for message propagation.
- Provide algorithms for social routing.
- Provide defensive mechanisms that individuals can use to protect their Identity information.
- Guarantee reachability, even in the case of having maximum privacy.
- Quantify the degree of anonymity of individuals in OSNs.
- Quantify the social priority between two individuals in OSNs using the social characteristics.
- Design and implement an online simulator to validate the performance of social routing algorithms.

## **1.7 Contributions**

The main goal of this dissertation is to design a protocol for OSNs that allows individuals to send and receive services to/from others who are not directly connected to them. The contributions of this study are the following:

- A Social Online Routing (SOR) protocol for supporting social routing on OSNs has been designed. It satisfies privacy options and minimizes end-to-end routing delays corresponding to the information elements exchanged under the Stratified Privacy Model.

- The level of anonymity achieved by Pseudo and Null identities is analyzed. The study takes into consideration attacks that attempt to identify the sender of the messages. For this, the Proxima Matrix and Proxima distributions are introduced to analyze the degree of anonymity of individuals.
- Because humans execute their tasks based on a perceived priority, it is necessary to present and utilize a computational framework that can analyze how people give social priority to each other where five social metrics have been proposed to estimate social priority. However, some possible extensions to this framework are: 1) combining the requester (who is asking) with the task content (the type of task), 2) including indirect social priority, which is given to people the user may just hear about (a friend of a friend) but with whom no direct connection is given to, and 3) including dynamic social priority, which is not fixed but changes over time.
- A simulator is designed and implemented in order to evaluate the study's proposed protocol. Using real datasets from Google Plus, the simulator is used to evaluate end-to-end routing delays corresponding to the information elements exchanged under the Stratified Privacy Model. This simulator can also be used by other scientists using different models to evaluate their work. A user needs to 1) connect to a server (Amazon Web Services, Microsoft Azure, etc.) to run the simulator; 2) create a new

folder or connect to an existing one to save his/her datasets and results; 3) upload a social graph or generate a random graph; 4) configuration some parameters (e.g. the number of messages to be generated by each node, the routing algorithm, the queue type and so on.); 5) run the simulator and get the result (Total\_Delay, Total\_Delay\_statistics\_ByNode, Network\_TotalDelay, etc).

Each of these contributions will be discussed in detail in the subsequent chapters.

## 1.8 Assumptions

The study and its programs and purposes involve a few assumptions that need to be addressed. They are as follows:

1. In cases where there are hidden values of the Status information and the Priority information, system wide default values can be used.
2. The connectivity relationship is known to both end points.
3. Nodes will not violate the local contract with their adjacent neighbors.
4. Each individual node is a priority queue.

## 1.9 Complexities and Difficulties of the Study

- **Complexity of social network anonymization:** the nearest work to this study involved a perturbation-based scheme (e.g. k-Anonymity problem) which is a hard problem [Verykios et al., 2004]. The perturbation-based scheme is currently used by organizations such as government agencies and hospitals to anonymize networks by adding noise (injecting random

nodes and edges) to achieve privacy before releasing them to a third-party for different purposes (e.g., analysis). However, the anonymization mechanism should be much more intricate because it needs to work in real-time. The two main differences between this study's proposed mechanism (Identities) and the perturbation-based scheme are that 1) the mechanism for this study is done locally by each node based on local preferences, whereas the perturbation-based scheme is done globally by an algorithm matching each node with others in the graph; and 2) while the perturbation-based scheme is applied offline and with full knowledge of nodes in the graph, the proposed mechanism in this study is performed online with partial knowledge about the network. According to the research done for this study, this is the first time an online real-time perturbation-based scheme has been introduced to the literature.

- **Complexity of quantifying the degree of anonymity:** the anonymity metric is used to determine the degree of anonymity a system provides against a specific anonymity attack. However, measuring degrees of anonymity is not a trivial task and, according to literature [Wagner & Eckhoff, 2015], 1) there is no consensus metric that should be used to quantify anonymity, 2) The availability of data or appropriate assumptions determine whether a metric can be used in a specific scenario. For these reasons, Proxima Matrix and Proxima distributions are introduced in order to quantify the degree of anonymity of service for the consumer.

- **Complexity of quantifying social priority:** Estimating an exact social priority between two individuals in OSNs is not easy and cannot even be easily estimated in offline social networks. However, at least in OSNs there is some interaction, communication and collaboration datasets between individuals that implicitly reflect a lot of information about the relationship between them. Such datasets can be used to approximate values of social priority.

### **1.10 Dissertation Organization**

The structure of the dissertation is as follows: Chapter 2 discusses the previous work relevant to routing and privacy in OSNs. Chapter 3 presents an overview of the Social Online Routing (SOR) Protocol architecture. Chapter 4 introduces the forwarding process and its modules (I-need, I-have, I-thank, and I-ack), along with the routing process and its algorithms (Topology aware, Social Priority aware, and Queue aware) and the experiment results of end-to-end routing delays. Chapter 5 presents the ability of SOR to meet the privacy requirements. Chapter 6 presents Social Priority. Chapter 7 presents the general architecture of the study's Online Social Network Simulator, introducing the parameters, rules and methods of internal flow of messages inside each SOR node. Chapter 8 presents the conclusion of the work developed in this dissertation.

## CHAPTER 2

### **Background and Related Works on Routing and Privacy Issues in OSN**

This chapter provides a concise and comprehensive overview of the research disciplines that underpin the study. The design of SOR stands on the shoulders of four previous studies (Human Dynamics, Social Routing, Quantifying the importance of individuals, and Privacy metrics). *Human Dynamics* aims to understand how human modeled in other studies due to human is a central node in our study and modeled as priority queue. *Social Routing* seeks to understand the mystery of how social routing work as well as investigate reachability issues in social networks. *Quantifying the importance of individuals* tries to estimate the priority between two individuals in OSN, presenting several efforts of inferring the influence between a pair of nodes. *Privacy metrics* shows that in order to provide a framework to estimate the anonymity provided by SOR, a set of studies for understanding and measuring privacy of end-users in OSNs was presented.

#### **2.1 The Human Dynamics Models**

Human Dynamics models are used broadly to understand human activity patterns. In that direction, a number of models have been proposed, such as the priority-queueing-based model [Barabasi, 2005], the adaptive-interests based model [Han, Zhou, & Wang, 2008], the memory effects-based model [Karsai, Kaski, Barabási, & Kertész, 2012], and the aging model [Blanchard & Hongler, 2007]. Barabási's model has been selected in particular for this study because of its simplicity and generality, as well as being the basis

for all other models. Barabási assumed that each individual has a priority list with  $L$  tasks, each task being assigned a priority value  $x_i \in [0, 1]$ , where  $i=1, \dots, L$ , chosen from  $\alpha(x)$  distribution. These priority values derive from human decision-making (whenever an individual is presented with multiple tasks and chooses among them based on some perceived priority parameters). His model was designed to predict the time interval between two consecutive actions by the same individuals. Actually, Barabási theoretically discussed three queueing protocols: First-In-First-Out, Random Choice, and the Deterministic protocol. A wide variety of research studies related to priority queueing have been conducted since then [Dezsö et al., 2006; Iribarren & Moro, 2009; Malmgren, Stouffer, Motter, & Amaral, 2008; Joao Gama Oliveira & Barabási, 2005; Vazquez, 2005; Vázquez et al., 2006; Vazquez, Racz, Lukacs, & Barabasi, 2007]. Furthermore, Schwartz [Schwartz, 1978] explained that theoretically social priority controls the decision to perform one task before or after another, and in reality, a social network is a set of different kinds of queues. Additionally, Larson [LARSON, 1987] investigated the individual's attitudes toward queues and the factors which may influence them (such as social injustice, which is defined as violation of first in, first out).

Recently, researchers at Google [Aberdeen, Pacovsky, & Slater, 2010] studied the information overload in Gmail and introduced the Priority Inbox, in which a tool attempts to alleviate such overload by learning a per-user statistical model of importance and ranking mail by how likely the user is to act on that mail. In 2013, another researcher from IBM and his colleague [Mukherjee & Garg, 2013] investigated the severity of the problem when software professionals got involved with multiple tasks in projects and



were inundated by too many notifications from the work-item tool. They proposed a TWINY, a machine learning-based approach to prioritize notifications. The study by Lubarski and his colleague [Pawel Lubarski & Morzy, 2012] utilized the priority queue to measure the importance of users in online social networks based on email communication patterns.

It is clear from the literature that humans have infinite daily tasks. Humans store these tasks in a queue-like structure. Each human has his or her own methods for ordering and selecting tasks. Each task has a certain importance which determines the priority that the human will give to the task. The priority determines the task position in the human queue. Based on the priority, task processing could be either serial or parallel. Humans either insert tasks into the queue in random order or based on priority. In the first case, the task is removed from the queue based on priority (low or high). In the second case, the queue is ordered, and the first task is removed in the priority queue.

## **2.2 Social Routing**

One of the earliest studies on social routing was done in the 1960s by social psychologist Stanley Milgram [Milgram, 1967; Travers & Milgram, 1969]. Milgram's work involved forwarding letters to a selected target (stockholder) who lived in Sharon, Massachusetts. The goal of sending letters was to examine people's ability to find routes to a destination within the social network of the American population. His experiment revealed that there was something special in the structure of natural and man-made complex systems, where a letter can be routed efficiently between any pair of nodes without a global view of the network. The big lesson from this study was not only that the

distance between people is six steps, but also that people have the ability to find the shortest paths using only local information about their social ties. Since that time, a number of network models [Kleinberg, 2006; J. M. Kleinberg, 2000; Watts, Dodds, & Newman, 2002], real-world experiments [Dodds et al., 2003; Kossinets & Watts, 2006], and simulations on network data [Adamic & Adar, 2005; Liben-Nowell et al., 2005; Schnettler, 2009] have been published to comprehend the “small-world” phenomenon in social networks.

Kleinberg [Kleinberg, 2006] surveyed the basic models of small-world networks and decentralized search algorithms. Kleinberg's work [J. M. Kleinberg, 2000] on navigation in a small-world highlighted the fact that it was easier to find short chains between points in some networks than others. Watts and his colleagues [Watts et al., 2002] presented a model that offered an explanation of social network searchability in terms of recognizable personal identities: sets of characteristics measured along a number of social dimensions.

Adamic and Adar [Adamic & Adar, 2005] simulated an experiment on a network of emails and a student networking website to address the question of how participants in a small world experiment are able to find shortest paths in a social network using only local information about their immediate contacts. More recently, researchers performed geographic routing simulations on a Live Journal social graph [Liben-Nowell et al., 2005], and observed that the probability of friendship between two users is inversely proportionate to their geographic proximity. However, Milgram and these studies

investigated the issue of reachability in a social network and yet these are far from a fully usable social routing protocol.

Despite such studies of routing techniques in transportation networks and in wired as well as wireless communication networks, no protocol has existed until recently for social routing, which can be used in online social networks, such as Facebook, Google+, and LinkedIn.

On the other hand, there is extensive literature on search and routing techniques. In general, interval routing [Gavoille, 2000, 2001], routing labeling schemes [Thorup & Zwick, 2001], greedy routing [Giordano & Stojmenovic, 2004], geographic routing [Giordano & Stojmenovic, 2004], compass routing [Giordano & Stojmenovic, 2004], etc. are routing techniques mainly proposed for wireless networks and/or transportation networks. Other routing techniques are designed to work with P2P networks [Banerjee & Basu, 2008; Castro, Druschel, Ganesh, Rowstron, & Wallach, 2002; Fujii, Ren, Hori, & Sakurai, 2009; Linnolahti, 2004; Xu, Min, & Hu, 2003]. This study in particular focuses on routing and forwarding techniques in OSNs from social aspects where social concepts and theories such as communities, context, and social information are used to guide the forwarding processes. Social aspects are used because people manage all social sites, and because forwarding processes such as *share* in Facebook and *retweet* in twitter are affected by human social characteristics. Moreover, node characteristics in current online social networks are available with some degree of accessibility. All that is needed is a method (e.g. incentivization techniques) that encourages nodes to exchange those characteristics in a way that protect nodes' privacy.

**Routing Based on Context and Social Information:** Social context information (like nodes' interests, friends, locations, trusts, preferences, priority, and so on) is also important information to forward messages toward the destination. OSNs such as Facebook, LinkedIn, and Google+ (being the prime examples) produce an unprecedented amount of social context information because on these networks people specify their relationships, update their statuses and share content with others [Kabir, Han, Yu, & Colman, 2012]. The social context information can be used in creating social routing applications [Anderson, Kourtellis, Finnis, & Iamnitchi, 2010]. Moreover, PeopleRank [Mtibaa, May, Diot, & Ammar, 2010], which is similar to the PageRank idea, gives higher weight to nodes if they are socially connected to other important nodes of the network. The context-aware framework, HiBop [Boldrini, Conti, & Passarella, 2008], can learn and represent through context information the users' behavior and their social relations, and using this knowledge to drive the forwarding process. Ramana et al. [Ramana, Chari, & Kasiviswanth, 2010] gave an overview about trust and current research in trust-based routing. SemAnt [Michlmayr et al., 2007] introduced a distributed content-based routing algorithm that used taxonomies to enhance search performance in peer-to-peer networks. For a detailed survey, one can refer to [L. Liu & Jing, 2012; Schurgot et al., 2011]. However, none of these studies has attempted to use a unified framework to combine all such characteristics to improve the social routing and make it more realistic and useful for societies.

**Priority Based-Routing Techniques:** Several studies in wired and wireless networks exist on using priority as an effective direction of research to forward messages

toward destinations. Farzad et al. [Farzad, Olver, & Vetta, 2008] considered a priority-based selfish routing model where agents may have different priorities on a link. An agent with a higher priority on a link can traverse it with a smaller delay or cost than one with a lower priority. In [Rajkumar & Sharma, 2008] a new priority-based routing scheme to handle biased call request patterns efficiently was proposed. Two factors determined the primary function: geographical context and network usage patterns. Wan [Wan, 2012] developed a simple priority-based dynamic assignment algorithm for multipath routing. The idea of the algorithm was to dynamically assign a packet to a suitable path based on the priority of a video packet. Meng and his colleagues [Meng et al., 2007] proposed a new routing algorithm, Priority-Based Routing (PBR), to balance the energy consumption of the sensor nodes in multi-sink sensor networks. Suthaputchakun et al. [Suthaputchakun & Sun, 2011] presented a Priority based Routing Protocol (PRP) in Vehicular Ad hoc Network (VANET), according to message types.

A common theme in the literature mentioned above is the notion of enriching edges with extra information such as trust, influence, homophile, priority etc. Our work is complementary and uses Social Priority (SP) to determine the position of messages in the queue.

**The Important Knowledge for Routing:** Some research has been done in developing knowledge-based routing algorithms. For instance, route selection with imperfect knowledge was introduced and simulated in [Feuz & Allan, 2012]. In [Feuz & Allan, 2013], group formation and knowledge sharing was introduced to study their impact on route selection. While their work focuses on how pedestrians select routes

using various preference criteria, this study views the impact of the imperfect knowledge of social priorities on routing in OSNs using SOR, the study's proposed protocol.

**Social-Based Forwarding and Routing Strategies:** Several efficient algorithms have been proposed for transferring messages between sources and destinations based on the social characteristics in different networks (e.g. ad hoc network, delay tolerant network, opportunistic network, and social networking services). Türkes and his colleagues [Türkes, Scholten, & Havinga, 2013] presented a social unicast routing scheme called RoRo-LT, which is based on self-assessment of people's daily routines for forwarding. To improve the forwarding efficiency of mobile networks, Hui and Sastry [Hui & Sastry, 2009] suggested that routes can be computed by using virtual world information and communities. Shen and his colleagues [S. Li et al., 2014] presented Centaur, which is an application-level user-assisted message dissemination solution for OSNs. The FSF (Friend list-based Social Forwarding), an opportunistic routing scheme designed to exploit social network information and pre-existing online social network information, was presented in [Socievole, De Rango, & Marano, 2013]. However, the work of this study intersects and combines features of these models where social characteristics are used to guide forwarding and routing algorithms.

### 2.3 Quantifying the importance of individuals in OSNs

Several efforts have been made to infer edge weights in social graphs. Hangal et al. [Hangal, MacLean, Lam, & Heer, 2010] presented a method for estimating the edge influence between a pair of nodes in DBLP (a computer science bibliography network) and Twitter datasets. In their study, the influence between nodes A and B is the

proportion of B's investments in A. In the DBLP dataset, for example, they use the number of papers that two co-authors share as a measure of investment. In Twitter, influence weights over the edges were assigned as the number of times user B retweeted user A, divided by the total of B's retweets. Influence is the proportion of interactions between one node and another node, to all of its interactions. In reality, investment alone is not a good way to predict the influence between pairs of nodes. It might be that two co-authors do not publish a paper together, but they may still have an influence on one another. In general, the studies of estimating edge weights can be either 1) Binary edge weight [Leskovec & Horvitz, 2008; Tyler, Wilkinson, & Huberman, 2005]; 2) signed edge weight [Leskovec, Huttenlocher, & Kleinberg, 2010b]; or 3) relationship tie strength, and weighted edges [Gilbert & Karahalios, 2009; Xiang, Neville, & Rogati, 2010]. However, the study focuses on the importance of the sender by using a set of social characteristics such as Gender, Degree centrality, Closeness centrality, Betweenness centrality, and Eigenvector centrality to estimate the priority between two individuals in OSN. This is done because each individual's social characteristic has a social impact (power) and expresses how important the individual is to her direct neighbors and all others in the OSN.

#### **2.4 Privacy Metrics:**

most of the recent privacy research focuses on understanding and measuring privacy of end-users in OSNs. Wagner et. Al. [Wagner & Eckhoff, 2015] classified privacy metrics using eight categories (Uncertainty, Information Gain/loss, Similarity/Diversity, Indistinguishability, Adversary's Success Probability, Error, Time,

and Accuracy/Precision). The uncertainty metric measured the certainty of an adversary. There are many uncertainty metrics built on Anonymity Set Size and Entropy. The Anonymity Set Size, used as an intermediate step to compute the degree of anonymity, counts the number of individuals that could potentially be a targeted individual [Kesdogan, Egner, & Büschkes, 1998; Reiter & Rubin, 1998].

Entropy [Serjantov & Danezis, 2002] measured the uncertainty associated with predicting the value of a random variable  $X$ . Despite the fact that there are some Entropy-based metrics (e.g. Asymmetric Entropy [Ayday, Raisaro, Hubaux, & Rougemont, 2013], Renyi Entropy [Clauß & Schiffner, 2006], Normalized Entropy [Diaz, Seys, Claessens, & Preneel, 2002], Conditional Entropy [Diaz, Troncoso, & Danezis, 2007], Cross Entropy [Merugu & Ghosh, 2003], and Cumulative Entropy [Freudiger, Raya, Félegyházi, Papadimitratos, & Hubaux, 2007], etc.), a set of papers [Hamel, Grégoire, & Goldberg, 2011; Shokri, Theodorakopoulos, Le Boudec, & Hubaux, 2011; Syverson, 2009] argue against the use of entropy as a privacy metric for the following reasons: 1) It is influenced by outliers in data; 2) It is easy to get the same entropy values from different probability distributions; 3) It is not easy to generate an accurate probability distribution of the members of an anonymity set; and 4) The absolute value of entropy cannot be interpreted and does not convey much meaning.

Because of these criticisms, this study uses an Anonymity Set-Size-based measure and focuses on measuring the degree of anonymity of individuals in OSNs using an Anonymity Set Size-based metric.



## CHAPTER 3

### Social Online Routing (SOR) Protocol Overview

This section will explain the standard SOR architecture as described in [Othman & Khan, 2015] and introduces the core features and concepts in the protocol.

#### 3.1 What is SOR protocol?

It is a decentralized-distributed service-providing protocol. It is a protocol through which Online Social Networks' nodes can exchange information to support Social Routing based upon who needs to know what.

#### 3.2 SOR Basics

There are four distinct parts of the SOR protocol:

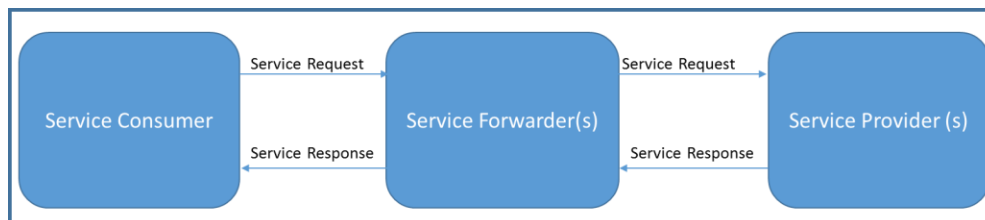
- Service (SR): a utility, commodity, accommodation, or activity that is required or demanded by the public and provided by organization or individual(s) such as LinkedIn's request for endorsements, and Facebook's request to join in a Cause, a need for a babysitting, etc. Let  $SR = \{sr_1, \dots, sr_m\}$  be a set of services supported by the OSN.
- Service Provider (SeP): an individual who owns/has the service and generates the I-have message to inform consumers in the network of the service that it can provide.
- Service Consumer (SeC): an individual who needs the service and generates the I-need message to get the service from the SeP.

- Service Forwarder (SeF): an individual who participates in forwarding the I-need message from consumers to providers, the I-have message from providers to consumers, and the I-thank Messages in both directions.

In this study, the Service Consumer and Consumer, the Service Forwarder and Forwarder, and the Service Provider and Provider are used interchangeably.

### **3.3 Peer-to-peer Social Consumer/Provider Model**

Sets of models were introduced in the literature for different purposes, such as Producer/Consumer [Paykin & Zdancewic, 2015], Publish/Subscribe [Huang & Garcia-Molina, 2004], Client-Server [Bertocco, Ferraris, Offelli, & Parvis, 1998], and Advertiser/Audience [Haishan Liu, Pardoe, & Liu]. Individuals in OSNs tend to not accept messages (e.g., ads) from other individuals or platforms, defining them as spam. Because of this, the Social peer-to-peer Consumer/Provider model is introduced as a model for OSNs as shown in Figure 1. The consumer first expresses his or her need for a service by sending an I-need message; the producer(s) will then respond by sending an I-have message. This is in contrast to an Advertiser/Audience model, where advertisers send their advertisements directly to an audience, and Publish/Subscribe model where publishers and subscribers do not have information about one another, preferring to communicate with each other through a broker. The communication in this study's model is based on social ties (e.g. friends) associated with social factors (e.g. trust and social priority).



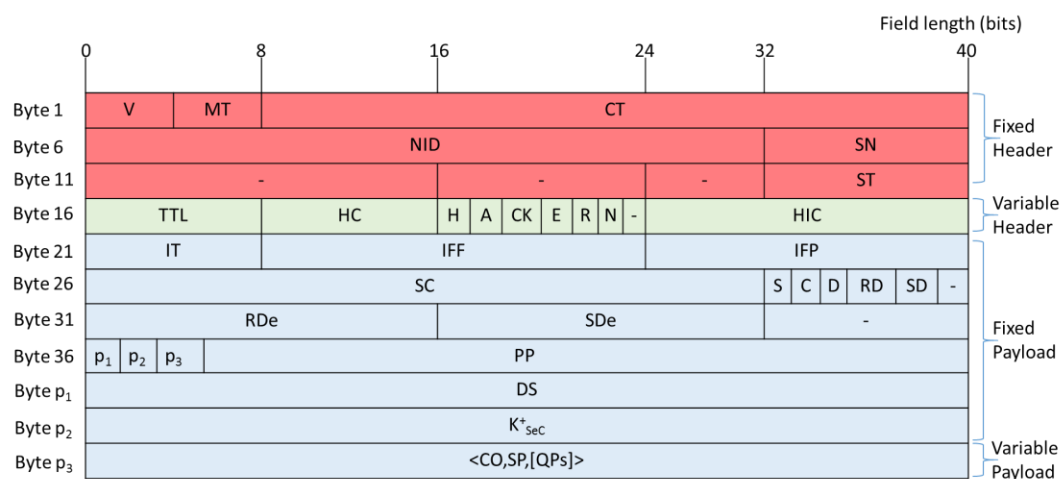
**Figure 1: Peer-to-peer Social Consumer/Provider Model**

### 3.4 SOR Messages

SOR uses five different messages: 1) I-need Message (InM) for carrying service information; 2) I-have Message (IhM) for carrying service provider information; 3) I-thank Message (ItM) for carrying service response (accepting or rejection) information; 4) I-like/dislike message (IdM) for carrying service interest policies; and 5) I-Ack Message (IaM) for carrying acknowledgment from particular forwarders to consumers.

#### 3.4.1 I-need Message Structure

The I-need message format is detailed in (Figure 2).



**Figure 2: I-need Message Structure**

The fixed and variable headers are present in every SOR message. The consumer assigns the fields in fixed headers and forwarders are not allowed to change them. The fixed header has the following fields (discussed in more detail later):

Version (V): Here, 4 bits identifies the version of SOR protocol and the version is currently assigned as “one” (1).

Message Type (MT): Here, 4 bits identifies the type of the message (i.e. 0, 1, 2, 3, 4 refer to I-need, I-have, I-thank, I-like/dislike, and I-ack messages respectively).

Creation Time (CT): Here, 4 bytes is an unsigned number containing the number of seconds that have elapsed since midnight 12:00 AM, 1 January 1970 00:00:00 UTC. This field is used to 1) break the tie in case two messages’ IDs are similar; and 2) reduce the size of bytes by allowing deadline fields to be an unsigned integer, two bytes.

I-need-ID (NID): Here, 4 bytes is a randomly generated value that uniquely identifies the I-need message. Sequence Number (SN): here, 1 byte is an incrementing counter which is started from a random number or from “one” and identifies a copy of the I-need message. The consumer can send different copies of the I-need message with same or different sequence number(s) to different neighbors, but no more than 256 copies.

Service Type (ST): Here, 1 byte tells the type of the service. The Product and Service Codes (PSC) [Brock, 2001] that describe products and services can be used as an ST. However, only four types are currently defined — babysitter (0), endorsement (1), recommendation (2), and tutoring (3).

Variable header fields reside between the fixed header and the fixed payload and, based on the purpose, their values can be changed at each hop.

Time-To-Live (TTL): Here, it is 1 byte that specifies how far the message is allowed to go on the OSN, in terms of node hops. Each node decreases the value of the TTL field by one prior to transmission. If the TTL field drops to zero, it is discarded.

Hop Count (HC): Here, 1 byte is the difference between the initial TTL (at the consumer) and the final TTL value (at the current node); however, the current node does not know the initial TTL. Because of this, it is associated with the message header. It can be used to specify from how far the message came.

Has-Hidden (H): Here, 1 bit informs the forwarder/provider if the Pathway ID Sequence (CO) field contains hidden IDs or not. Providers perform routing computations using default values instead of the actual Social Priority (SP) and Queue Parameters (QPs) values, and in turn, the hidden nodes must use these default values.

Has-Anonymous (A): Here, 1 bit tells the receiver node that the CO field has some anonymization IDs. Neither of the two fields tell how many of the IDs are hidden or anonymized. However, the receiver node can compare the value of the HC field with the content of CO field to estimate how many hidden/anonymized IDs are in the path.

Acknowledge (CK): Here, 2 bits tells some nodes in the OSN to send an acknowledgment back to the consumer. Its value could be zero (CK=0), which means no acknowledgement is needed to be sent back and named Send-with-Hope, and has No-Network-Overhead. It also could be one (CK=1), meaning that the last node that is going to drop the I-need message (because of the TTL=0 or an expiration of any other deadline field) must send an acknowledgment back to the consumer; named Send-with-Knowledge, it has Less-Network-Overhead. It could also be two (CK=2), which means

the last node which is going to drop the I-need message must do two things: 1) Send acknowledgment back to the consumer and then 2) forward the I-need message to the next stage with a new TTL=3, CK=1, and E=1, named Extend-with-Knowledge, More-Network-Overhead. The extender must keep the QPs and SPs in its table and clear them from the I-need message.

The extend mechanism will be explained in subsection 4.1.4. For Extended (E): Here, it is 1 bit. If the extended bit is set, the last node which decrements the TTL to be zero is allowed to assign a new value to the TTL (the default is 3) and retransmit the I-need message in the OSN, sending an acknowledgment message back to the consumer. The Extended field refers to the I-need message, which has been forwarded more than the usual three steps. In other words, intermediate nodes are responsible for this message, in addition to the originator. Although this is not normal, it is based on the application and network structure where there are networks with a long diameter (the scheme is described in CHAPTER 4).

Peer Privacy Request (R): Here, 1 bit is used by nodes to tell the next hop either to anonymize (R=0) or hide (R=1) it.

Peer Anonymize Name (N): Here, 2 bits are used by a node to tell the next hop that its ID is either 0 (Null: use null identity in the CO field, but SP and QP might contain values); 1 (Local: use Real Identity such as IP address); 2 (Global: use Globally unique Pseudo Identity); or 3 (Pseudo: use the Locally unique Pseudo Identity). The scheme is described in CHAPTER 5.

Header Integrity Checksum (HIC): This is a 16-bit checksum, where the checksum is computed over the header to ensure the integrity of the I-need message after its transmission from consumer to provider.

The fixed payload fields reside between the variable header and the variable payload. The consumer assigns values for the fields, and intermediate nodes are not allowed to change them.

Incentive Type (IT): Here, 1 byte indicates the incentive type (such as micropayment, point, etc). The incentives of forwarders and providers (IFF: 2 bytes, IFP: 2 bytes) are values that encourage intermediate nodes to forward the I-need message and to encourage the providers to provide a service by sending the I-have message. Some applications may require incentives.

Service Code (SC): It is 4 bytes and is an ID used to describe the service. The United Nations Standard Products and Services Code [Schulten et al., 2001] could be used, as it is an open, global, multi-sector standard for efficient, accurate, classification of products and services and can be used as a SC value.

Service Scope (S): Here, 1 bit defines the service the consumer expects to receive from the provider; it can be private or public.

Service Action (C): Here, 1 bit tells the providers how the consumer expects to receive the service (e.g. Online, Offline).

Service Delivery (D): Here, 1 bit tells the provider how urgent/ normal the service is to the consumer.

Response Deadline Type (RD): Here, 2 bits refers to the type of value in the RDe which might be minutes (RD=0), hours (RD=1), or days (RD=2).

Service Deadline Type (SD): Here, 2 bits refers to the type of value in the SDe which might be minutes (SD=0), hours (SD=1), or days (SD=2).

Response Deadline (RDe): Here, 2 bytes tells the provider who has received the I-need message that it is required to send in order to notify the consumer by the associated deadline. Service Deadline (SDe): Here, 2 bytes indicates a time required by the consumer for performing a certain action.

Both deadlines play a central role in protocol performance (i.e., forwarding and providing processes) and in protocol scalability (i.e., size of tables used).  $p_1$ ,  $p_2$ ,  $p_3$ : variable length field with 1-4 bytes refers to the beginning of PP, DS, and  $K^+_{\text{sec}}$  respectively.

Propagation Policy (PP): This starts at byte 36 and ends at byte  $p_1-1$  where  $p_1-36$  = length of PP; it is assigned by the consumer and guides forwarders to select the next set of forwarders (two languages are used for the propagation policy as described in section 3.7).

Digital Signature (DS): This starts at byte  $p_1$  and ends at byte  $p_2-1$  where  $p_2-p_1$  = size of DS. Common sizes of DSs are 128 or 256 bytes; they are used to recognize if the propagation rules have been tampered with.

Service Consumer's public key ( $K^+_{\text{sec}}$ ): This starts at byte  $p_2$  and ends at byte  $p_3-1$  where  $p_3-p_2$  = the size of  $K^+_{\text{sec}}$ . A 2048-bit modulus can theoretically fit over exactly 256 bytes (since  $256*8 = 2048$ ), but more bytes are needed to encode other values; it is a key



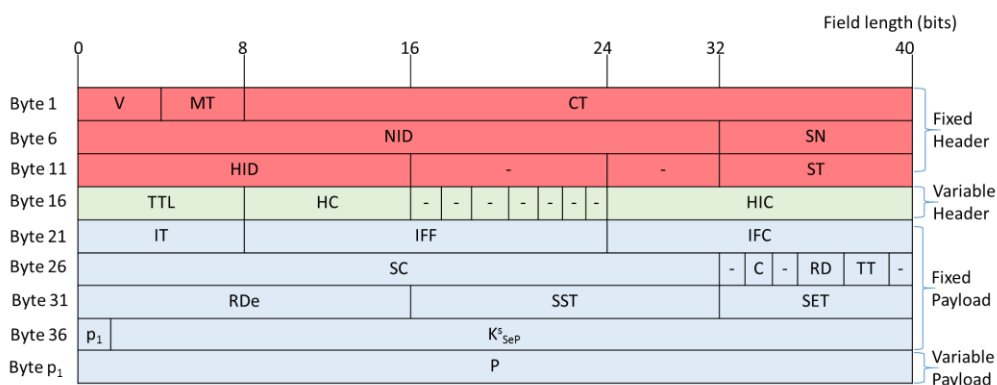
disseminated with the I-need message for 1) I-have message confidentiality where providers encrypt their I-have messages and only the consumer can decrypt them; 2) consumer authentication where consumer and providers confirm the identities of each other; and 3) I-need Integrity where forwarders and providers ensure that the I-need message was altered without detection.

The variable payload fields reside after the fixed payload. The values of their fields are assigned by the intermediate nodes (forwarders). A Pathway ID Sequence (CO) is a path (chain of nodes) that starts from the service consumer and ends at the service provider. Each intermediate node adds the local, global, pseudo, or null identity of its previous adjacent neighbor (the scheme is described in CHAPTER 5). A Pathway Social Parameter Sequence (SP) is a set of estimated Out-Social Priorities. Each node can add the SP value (Social Priority is described in CHAPTER 6). A Pathway Queue Parameters Sequence (QP) is tuple and consists of an arriving message rate ( $\lambda$ ), a utilization rate ( $\mu$ ) and a queue size (L) at a particular time (Cognitive Human Queue is described in 4.2). <CO, SP, [QPs]>: This begins at byte  $p_3$  and ends at the last byte of the message; it is a tuple of the three values of CO, SP and QPs.

The encryption of the content of the I-need message reveals the sender identity. Thus, the content of the I-need message is not allowed to be encrypted (it will be proved in chapter five).

### **3.4.2 I-have Message Structure**

The fields of the I-have message are described in (Figure 3).

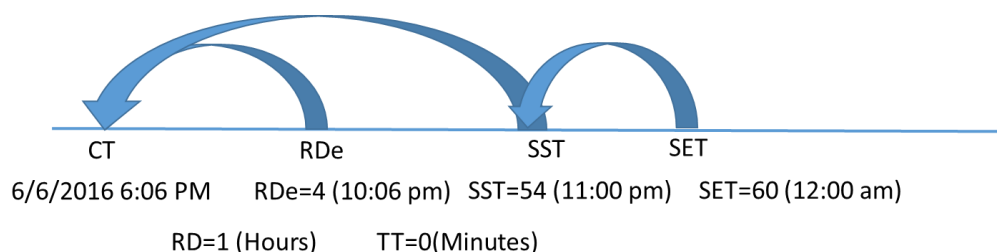


**Figure 3: I-have Message Structure**

I-have-ID (HID): It is 2 bytes, and is a randomly generated value that uniquely identifies the I-have message.

Service Time Type (TT): This is 2 bits, and refers to the type of value in the SST and SET which might be minutes (TT=0), hours (TT=1), or days (TT=2).

Service Start Time (SST): this is 2 bytes; and Service End Time (SET): this is 8 bits, Both SST and SET tell the consumer when the provider can provide the service. The consumer must then respond by sending an I-thank message before the RDe deadline and the provider must commit to service between the SST and the SET. Figure 4 demonstrates the dependencies between CT, RDe, SST, and SET.



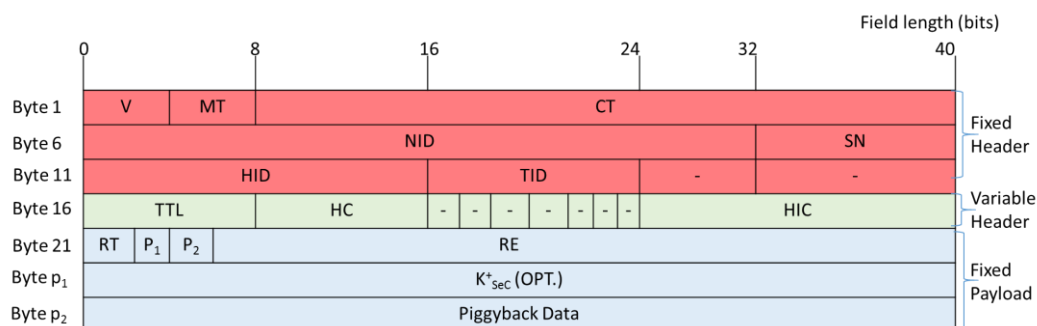
**Figure 4: Time Dependencies between CT, RDe, SST, and SET Fields**

Service Provider's encrypted symmetric key ( $K_{SeP}^s$ ): This starts at byte 36 and ends at byte  $p_1$ ; it is a random key used to encrypt the I-have message. The symmetric key is then encrypted by the consumer's public key, which associates with the I-need message to form a digital envelope.

The path (P): This starts at byte  $p_1$  and is a set of intermediate nodes chosen by the provider to forward the I-have message toward the consumer. The other fields are the same as in an I-need message.

### 3.4.3 I-thank Message Structure

The I-thank message format is discussed in Figure 5.



**Figure 5: I-thank Message Structure**

I-thank-ID (TID): It is 1 byte, and is a randomly generated value that uniquely identifies the I-thank message.

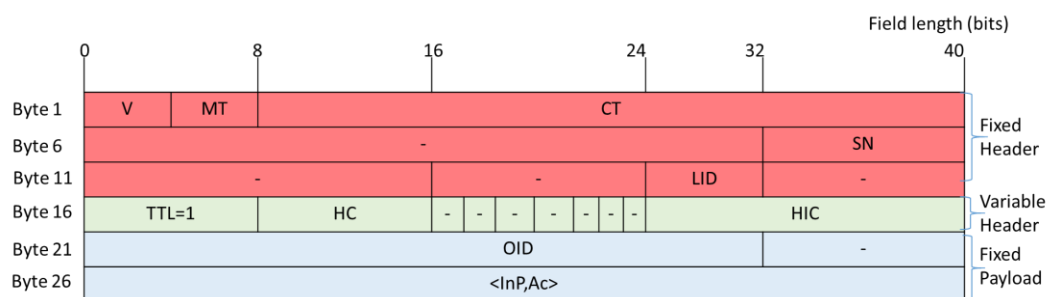
Response Type (RT): It is 1 bit, and is either granted (I-thank-sure: the consumer accepted the I-have offer) or denied (I-thank-but: the consumer rejected the I-have offer and the reason is in the Reason field.).

Reason (RE): This string type starts at byte 21 and ends at byte  $p_1-1$ ; it tells the provider why the I-have message has been rejected. The consumer can send a new public

key,  $K_{\text{Sec}}^+$ , to the provider. The consumer can send data in the I-thank-message (Piggyback) to the provider. The message is encrypted using the symmetric key sent with the I-have message. The other fields are the same as in the I-need and I-have messages.

### 3.4.4 I-like/dislike Message Structure

The contents of the I-like/dislike message are detailed in Figure 6.

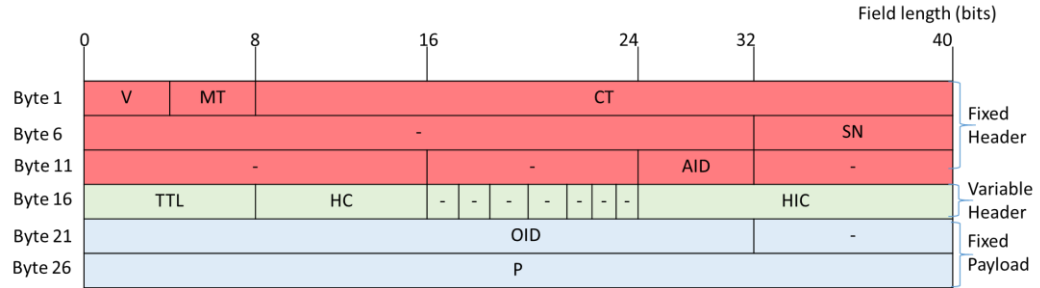


**Figure 6: I-like/dislike Message Structure**

I-like/dislike-ID (LID): This is 1 byte, and is a randomly generated value that uniquely identifies the I-like/dislike message.

The originator-ID (OID): This is 4 bytes, and is a randomly generated value that uniquely identifies the sender of the message. If the TTL value is one, then the I-like/dislike message is allowed to progress by one step (only to an adjacent neighbor). The in-self-interest policy (InP) is a set of rules that reflects the kind of I-need messages that the node is willing to receive/not-receive from their adjacent neighbors. The node asks its adjacent neighbors to help it by not sending some I-need messages based on this policy. The action (Ac) indicates the like or dislike of the InP. The <InP, Ac> tuple starts at byte 26 of the message. The other fields are the same as in the I-need and I-have messages.

### 3.4.5 I-ack Message Structure



**Figure 7: I-ack Message Structure**

The fields of this message (as shown in Figure 7) are similar to the fields of I-like/dislike message, except that the sender of this message has to do routing computation (discussed in CHAPTER 4) and assign a path (P) to the message. The forwarders follow this path to the consumer.

The I-ack-ID (AID): It is 1 byte, and is a randomly generated value that uniquely identifies the I-ack message.

## 3.5 SOR Tables

To transmit the I-need messages from consumer to provider, the I-have message from provider to consumer, and finally the I-thank message in both directions, each SOR node maintains these table-like data structures: *Messages Table*, *Forwarding Table*, and *Routing Table*. *Self-Interest Table* and *Peer-like/dislike Table*. These tables are used to store the policies.

### 3.5.1 Messages Table (MeT)

The MeT maintains an entry for the I-need, I-have, I-thank, and I-ack messages. It is a complete log/record of all of the messages that the node has sent, forwarded,

originated, and/or deleted. Messages can be in multiple states: arrived at time  $t_1$ , Forwarder at time  $t_2$ , and deleted at time  $t_3$ . There is a timestamp for each action and what has been done. Each MeT entry has nine fields– NID, HID, TID, AID, Belong-to, Message-Type, Message-As-Obj., Action-Time, and Status. The NID is 4 bytes long and, in combination with HID, TID, and AID, it uniquely identifies an entry. The Belong-to identifies the originator of the message which is either the current node (0) or other nodes (1). The Message-Type helps to distinguish between messages (0: I-need, 1: I-have, 2: I-thank, 3: I-like/dislike, and 4: I-Ack). The Message-As-Obj is the message object. Whenever a message is created (0), forwarded (1) or deleted (2), a time and the action are recorded in the Action-Time and the Status fields, respectively.

### 3.5.2 Forwarding Table (FoT)

FoT maintains an entry for each I-need message and its I-have, I-thank, and I-ack messages. The FoT table is used by all nodes (consumers, forwarders, or providers) to trace the I-need message and its responses messages (I-have, I-thank, and I-ack). Each FoT entry has fourteen fields – NID, SN, PP, RDe,  $k_{\text{seC}}^+$ , Received-From, Sent-To, Name, SP,  $[\text{QP}_s]$ , HID, RDe<sub>2</sub>, TID, and AID. Whenever the I-need message is created or forwarded, the NID, SN, PP, RDe, and  $k_{\text{seC}}^+$  fields are collected from the I-need message and maintained in an entry in the table. This entry will be deleted if no I-have message is received and the RDe deadline has expired. The I-need message could be received from more than one, neighbor; thus, a new entry is created for each message and the Received-From field assigns the sender's port or ID. The Sent-to field contains all ports/IDs of the next-hop(s). According to the type of anonymization request, the Name field saves the

used identity (Local, Global, Null, or Pseudo). The current node can send different SPs and QPs to different outgoing neighbors so that the fields SP and [QPs] can keep the sent values. The current node can send an I-have message back to the consumer; for that, a new record needs to be added with the value Me in the field Sent-To, or it can forward the I-have message. The field HID keeps the I-have message ID. The consumer must respond by the deadline  $RDe_2$  associated with the message I-have message, or the record will be deleted. The TID and AID fields keep the IDs of the I-thank and I-ack messages.

### **3.5.3 Routing Table (RoT)**

The RoT table is a set of rules, often viewed in table format, that is used to determine where messages traveling over the OSN will be directed. In the current version of SOR, the provider may not send an I-have message without receiving an I-need message. In addition, the forwarder cannot forward the I-have message without an entry in their Forwarding table showing the time the I-need message was received and forwarded. The RoT can be used to accelerate the routing computation in case of static networks.

To carry out the like and dislike functions, each SOR node maintains two tables: The Self-Interest Table, and the Peer-like/dislike Table.

### **3.5.4 Self-Interest Table (SiT)**

The SiT stores a) an In-Self-Interest Policy, a set of rules that reflects the kind of I-need messages that the node is willing to receive or not from its adjacent neighbors; and b) an Out-Self-Interest Policy, a set of rules that reflects the kind of I-need messages the

node is or is not willing to forward to adjacent neighbors. The node filters outgoing I-need messages by using an Out-Self-Interest policy, but relies on its incoming neighbors to filter out incoming I-need messages based on its In-Self-Interest which is sent to adjacent neighbors using an I-like/dislike message. Each SiT entry has six fields: PID, Self-Interest-Policy, Direction, My-Action, Intensity, and List-Of-Neighbors. The Policy identifier (PID) is a randomly generated value that uniquely identifies the table entries. The Self-Interest-Policy is a set of rules that reflects the kind of I-need messages that the node is willing or not willing to send or receive to/from its adjacent neighbors. The Direction is a value which identifies the In-Self-Interest-Policy (0) for incoming adjacent neighbors and Out-Self-Interest-Policy (1) for outgoing adjacent neighbors. The My-Action (i.e. like (0) or dislike (1)) indicates whether the forwarder is going to forward the I-need message or not. The Intensity is the number of I-need messages that are received by the node and that match its Self-Interest-Policy. A system-wide threshold setting by the user or an agent is used to execute an action when the Intensity exceed the threshold. A List-Of-Neighbors is a set of incoming or outgoing neighbors.

### **3.5.5 Peer-like/dislike Table (P2T)**

When a node receives the I-like/dislike message from its adjacent neighbor(s), it maintains the attached In-Self-Interest in P2T in the field Peer-Interest-Policy. To prevent a permanent block, the length of time in which messages of information are stored on the tables is based on the deadline known by all nodes in the protocol. Each P2T entry has four fields: Neighbor-ID, LID, Peer-Interest-Policy, and Action-Requested-By-Peer. The Neighbor-ID is a randomly-generated value that uniquely



identifies the neighbor. The LID is a randomly-generated value that uniquely identifies the I-like/dislike message. The Peer-Interest-Policy is an In-Self-Interest-Policy sent by the peer. The Action-Requested-By-Peer is the My-Action (i.e. like (0) or dislike (1)) of the peer, indicating whether or not the forwarder is going to forward the I-need message.

### 3.6 Policies

To determine which forwarder gets which message (i.e. I-need message), the SOR node uses policies of special sets of rules used for message filtering and forwarding.

#### 3.6.1 Propagation Policy

The propagation policy, a set of edge and node attributes-based-rules complementary to the service fields in the I-need message, guides forwarders to determine next-hop(s). The service fields of the I-need message describe mandatory information such as deadlines and service types, and the propagation policy describes any other details related to nodes, edges, and service. This kind of policy is generated by the consumer and then associated with the I-need message. For example, the policy  $r_2 = \langle [c_1:\text{age}>18, c_2:\text{country}=\text{Libya}, c_3:\text{country}=\text{USA}] :: [((c_2|c_3)\&c_1)] \rangle$ , means that the next forwarder must be Libyan or American and his/her age must be older than 18 years.

#### 3.6.2 Self-Interest Policy

The self-interest policy is a set of message attribute-based-rules which determines the interest of a node in particular kinds of I-need messages. There are two Self-Interest Policies: *In-Self-Interest Policy*, and *Out-Self-Interest Policy*. There are some situations

where a node might tell its incoming adjacent neighbors that an In-Self-Interest Policy is different from the one that it has.

### **3.6.3 Peer-Interest Policy**

The peer-interest policy is a set of In-Self-Interest policies of outgoing neighbors which is sent to the node. There are four policies, as the Peer-Interest Policy and In-self-Interest Policy may sometimes differ, particularly when the node shows a difference from what it had before.

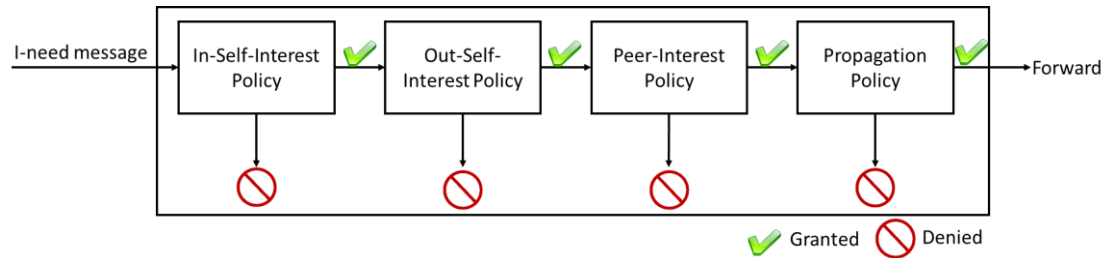
### **3.6.4 Policy-based I-need Message Checkup Process**

The Self-Interest Table (SiT), Peer-like/dislike Table (P2T) and ID-Card (IDC) are all used for forwarding. The IDC entries maintain social features of the node, like a unique identifier UID, Name, etc. The SiT entry records the In-Self-Interest and Out-Self-Interest policies. The P2T contains a set of In-Self-Interest policies (called the Peer Interest Policy) of adjacent neighbors. Figure 8 shows the forwarding process.

1. When the forwarder receives an I-need message, it first checks it against the In-Self-Interest Policy in the SiT, and drops the message with no further processing if there is a match with a dislike action.
2. If there is a match with a like action, then it checks the message against the Out-Self-Interest policy in the SiT, and drops the message with no further processing if there is a match with a dislike action.

3. If there is a match with a like action, then it checks the message against the Peer Interest Policies in the P2T, and drops the message with no further processing if there is a match with dislike action.
4. If there is a match with a like action, then it checks the message propagation policy (PP) in I-need message against the IDC and link attributes, and drops the message with no further processing if there is no match.
5. If there is a match, then it forwards the message to a set of the next chosen forwarders.

The In-Self-Interest Policy overrides all other policies. Furthermore, the Out-Self-Interest Policy overrides the Peer-Interest Policy.

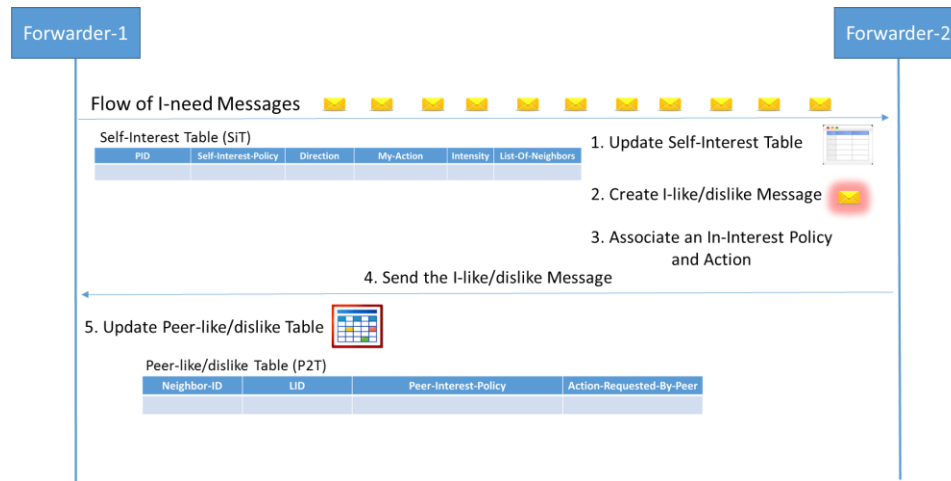


**Figure 8: Policy-based I-need Message Checkup Process**

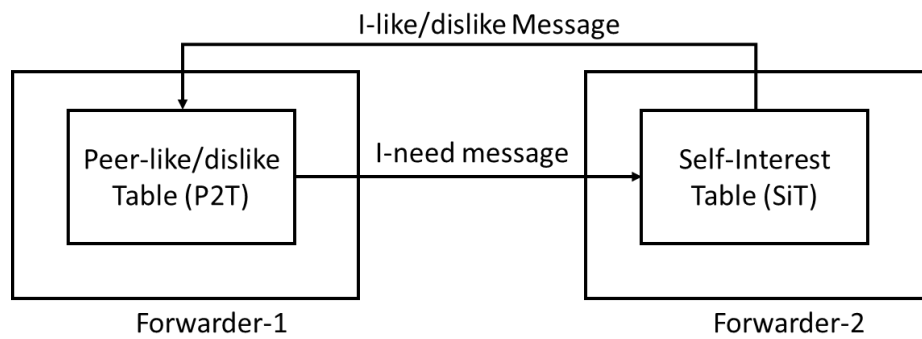
### 3.6.5 I-like/dislike Module

Each node keeps track of the incoming flow of the I-need messages (as shown in Figure 9 and in Figure 10). The node creates I-like/dislike messages when the Intensity value of any entry in the Self-Interest Table (SiT) exceeds a predefined threshold  $\delta$  (which is assigned by a user or an agent). The I-like/dislike message associated with In-Interest-Policy and Action is then forwarded to an adjacent neighbor(s). Once the I-

like/dislike message arrives at the adjacent neighbor (s), the peer-like/dislike Table is updated with the new rule. The node will not receive any undesirable I-need message. In this way, any undesired behavior can be handled appropriately.



**Figure 9: Flow Diagram of I-like/dislike Module**



**Figure 10: I-like/dislike Module**

### 3.7 Attribute-based Languages

This study defines two languages for policies: The Link-Attribute-based Propagation Language and the Node-Attribute-based Propagation Language.

### 3.7.1 Link-Attribute-based Propagation Language (LAP):

The LAP is a set of rules that enables service consumers to control who can see the I-need message. The forwarder sends the I-need message to its adjacent neighbors based on a predefined set of rules that are associated with the I-need message in the field PP. The rules can be used for a single link or for a set of consecutive links (Path).

The LAP syntax for a rule has two parts: The label, and the set of logic binary expressions separated by a special marker (::). The Syntax of a LAP rule is:

$$\text{Label} = \langle \text{expression}_1:: \text{expression}_2:: \dots :: \text{expression}_k \rangle$$

The special symbol (?) is used in this study to substitute any type of used types. Assuming there are sets of relationships types on a link: Friend (F), Colleague (C), and Enemy (E), the (?) could be (F|C|E). For example,

- Friends\_Colleague\_Anyone =  $\langle \text{FFC?} \rangle$  means the first, second, and third edges must be Friend, Friend, and Colleague respectively and the fourth edge can be any one of (F|C|E). However, the path length must be 4.
- Anyone\_Friend =  $\langle ?\text{F} \rangle$  means the first edge can be anyone of (F|C|E), but the second edge must be their friends and the path length must be 2.

The special symbol (\*), coming in a a superscript, is also used here to refer to the repetition of type. The type of relationship can be repeated many times by using integer numbers or as many as possible by using the star symbol. For example,

- Friends\_Colleague\_Anyone =  $\langle \text{F}^2\text{C}^1? \rangle$  equals  $\text{FFC}(\text{F|C|E})$ .
- All\_Friends =  $\langle \text{F}^* \rangle$  equals (FFFFFF...) and means the relationship on all edges must be friend and the path can be any length.

- All= <?\*> equals (F|C|E)\* and means any relationship type and any path length.

### 3.7.2 Node-Attribute-based Propagation Language (NAP):

The NAP is a set of rules that enables service consumers to control who can see the I-need message based on its attributes. Each node can have a set of attributes and each attribute is a <attr: value>, where attr is an attribute-identifier; and value is the attribute-value (for example, <country: USA>, <age: 18>, <city: Kent>). The current node matches the next candidate node's attributes with the attributes-based propagation rules in the I-need message in order to verify if it can receive the I-need message or not.

The NAP syntax for a rule has two parts: The label and two expressions separated by a special marker (::), representing the condition-variables and the condition. The Syntax of a NAP rule is:

$$\text{label} = \langle [\text{condition-variables}] :: [\text{condition}] \rangle$$

The conjunction (&), disjunction (|), and negative (~) logic operators are used in the two expressions. The equality/inequality operators: [=, !=, <, >, >=, <=] are only used with the condition variables. The simple and composed condition variables are defined as follows:

- **The simple condition variables** are in the form [label:Variable?Value] where ? is the equality/inequality operator. For example, [c<sub>1</sub>:age>18, c<sub>2</sub>:country= Libya, c<sub>3</sub>:country=USA].

- **The composed condition variables** are in form [label:label?Label] where? is the logic operator. For example: [c4:age<18, c5:age>10, c6:c4&c5].

The condition can be simple or composed and is in the form (label?Label), where ? is the logic operator. For example, LybUSA18 = <[c1:age>18, c2:country= Libya, c3:country=USA] :: [(c2|c3)&c1]> which means that the next forwarder must be Libyan or American and his/her age must be older than 18 years. NotLybUSAbut18 = <[c1:cge>18, c2:country= Libya, c3:country=USA] :: [(c2~|c3~)&c1] >means that the forwarder must not be Libyan or American, but that his/her age must be older than 18 years. AgeG10L18 = [c4:age<18, c5:age>10, c6:c4&c5] :: [(c6)], which means the forwarder age must be between 10 and 18 years.

### 3.8 Forwarding and Routing

Forwarding is the process responsible for helping consumers find their potential providers in a scalable and efficient way. It consists of four modules: I-need, I-have, I-thank, and I-ack. Routing is the process of finding the best/shortest path from a provider to a consumer. Based on the available knowledge, the routing process has three routing algorithms: Topology aware algorithm, algorithm, and Queue aware algorithm. Both forwarding and routing are discussed in detail in CHAPTER 4.

### 3.9 Stratified Privacy

One of the standard features of SOR protocol is supporting stratified privacy for consumers, forwarders and producers. The forwarders can participate in forwarding

messages that are either anonymized or hidden. Along the return path, the I-have message is encrypted as it flows from the satisfying producer to the consumer. A few schemes for privacy were proposed and analyzed in this study, the study showed that these schemes provide strong privacy for forwarders as well as both the consumer and provider at particular levels. The schemes and their details are discussed in detail in CHAPTER 5.



## CHAPTER 4

### **Reachability and Efficiency of SOR**

This chapter first discusses the forwarding process and its modules (I-need, I-have, I-thank, and I-ack) and introduces the routing process and its algorithms (Topology aware Shortest-Path-Based Routing Algorithm, Social-Priority-Based Routing Algorithm, and Queue-aware Social-Priority-Based Routing Algorithm).

#### **4.1 Forwarding**

The consumer forms a propagation policy which includes the features of a desired service, as well as the characteristics of forwarders and potential providers using specific predefined languages: Link-Attribute-based Propagation Language (LAP) and a Node-Attribute-based Propagation Language (NAP). The consumer puts the propagation policy, in addition to other attributes, into a new I-need message and broadcasts it into the network. The intermediate nodes (Forwarders) use this propagation policy along with other policies (i.e. Self-Interest Policy, Peer Interest Policy) to forward the I-need message toward the potential providers. The forwarders might add some information to the I-need message like Connectivity information, Social Priority (SP), and/or Queue Parameters (QPs). Once the I-need message reaches a provider, the provider replies by sending an I-have message back to the consumer through a path (computed using a few routing algorithms based on the available information). When the consumer receives the I-have message and is willing to get the service from the sender (provider), then an I-thank message is sent to the provider by using the same path taken by the I-have message

to get back to the provider. Once the I-thank message reaches the provider, another protocol can be used by either online (e.g. bitcoin) or offline communication based on the needed services. Both the online and the offline communications are beyond the scope of this dissertation. To further explain necessary concepts of this work, four modules need to be introduced at this point: the I-need Module, the I-have Module, the I-thank Module, and the I-ack Module.

#### **4.1.1 I-need Module**

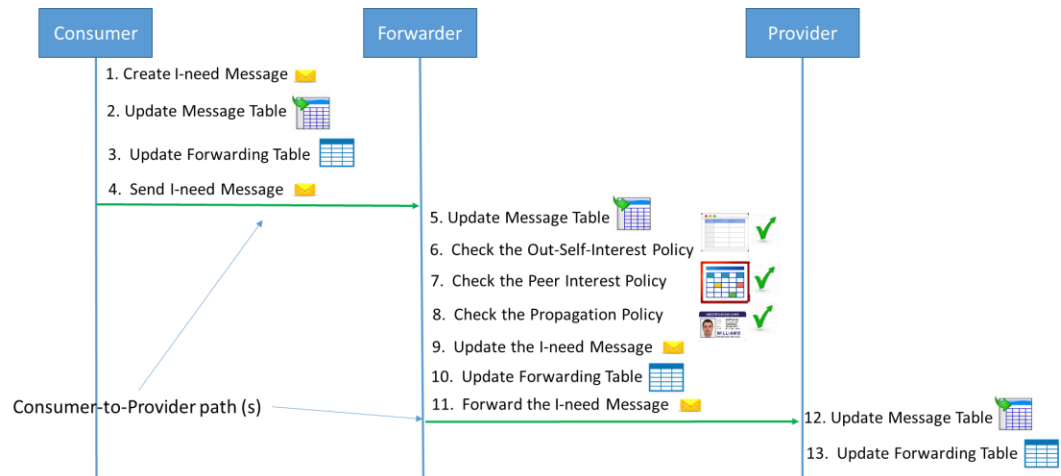
The goal of this module is to transmit the I-need message from its consumer to its potential provider through a set of forwarders. The Message Table (MeT) and the Forwarding Table (FoT) are used by all participants to keep track of the I-need message. Furthermore, other data structures (the node ID-Card [IDC], the Self-Interest Table [SiT], and the Peer-like/dislike Table [P2T]) are used for matching the policies and for checking if the next-hop is eligible to receive the message. The IDC entries maintain social features of the node such as unique identifier UID, Name, etc. The SiT table entry records the In-Self-Interest and Out-Self-Interest policies. The P2T table contains a set of In-Self-Interest policies (known as Peer Interest Policies) of adjacent neighbors.

First, the consumer creates the I-need message and associates it with the propagation policy and other fields, inserts a new entry into the MeT and the FoT tables, chooses the next-hops based on its Out-Self-Interest, and then sends the I-need message to the chosen adjacent neighbors (as depicted in Figure 11 in steps 1-4).

Second, once a forwarder receives the I-need message, it inserts the message into the MeT table. It then checks the message against the forwarder's Out-Self-Interest

policy, and drops the message with no further processing, if there is no match. If there is a match, it checks the message against the Peer-Interest Policies in the P2T table, and drops the message with no further processing, if there is no match. If there is a match, then it checks the message propagation policy (PP) against the IDC and link attributes, and drops the message with no further processing if there is no match. If there is a match, then it updates the I-need message by adding information to the field <CO,SP,[QPs]> and by updating the FoT table. Finally, it forwards the message to a set of next chosen hop(s), as demonstrated in steps 5-11 in Figure 11. The forwarding process is a kind of controlled information flood.

Finally, when the I-need message reaches the provider, it adds a new entry to the MeT and to the FoT tables respectively (steps 12 and 13). It then creates an I-have message and sends it back to the consumer (as described in the next Module).



**Figure 11: Flow Diagram of I-need Module**

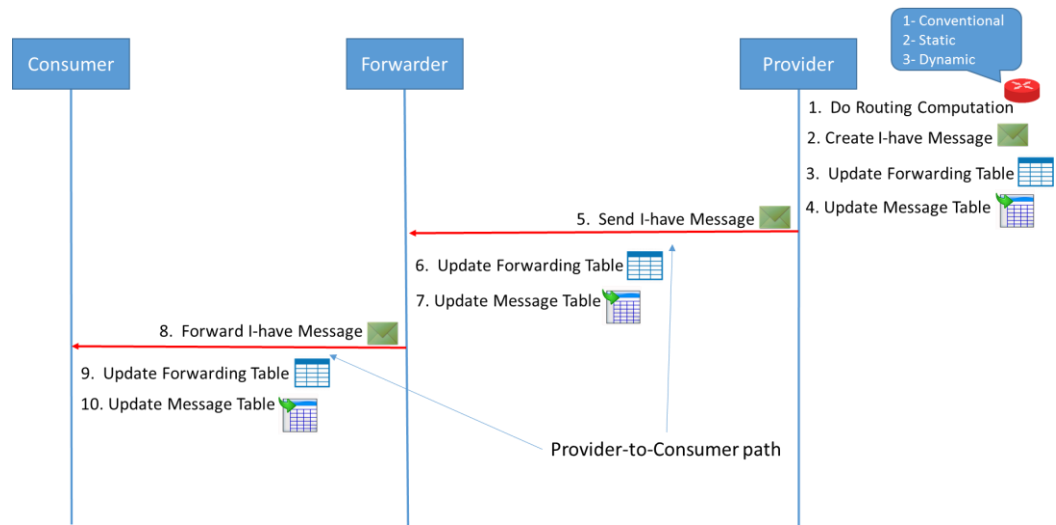
### 4.1.2 I-have Module

The goal of this module is to transmit the I-have message from its provider to the consumer through a set of forwarders. The Message Table (MeT) and the Forwarding Table (FoT) are used by all participants to keep track of the I-have message.

First (steps 1-5 in Figure 12), the provider may receive a set of I-need messages with the same NID, but from different adjacent neighbors. It collects the connectivity, social priorities, and queue parameters from these messages, using them to build a social graph. Based on the available information in the graph, the provider decides which routing algorithm must be used (section 4.2 describes the routing algorithms in detail). The routing algorithm is used to compute the best/shortest path to the consumer. A new I-have message is then created and associated with the computed path. The MeT and FoT tables are updated and the I-have message is forwarded to the first node in the path.

Second (steps 6-8), once the forwarder receives the I-have message, it inserts the message into the MeT table. Each forwarder in the path queues and processes the I-have message in its own basis (e.g., social priority, first-come-first-served, etc.) and then forwards it to next-hop. During the forwarding process, the FoT table is updated. The forwarding process is one-to-one because the path is determined by the providers making forwarders follow the given path.

Finally, (steps 9 and 10), when the I-have message reaches the consumer, the MeT and FoT tables are updated and a new I-thank message is created if the consumer is willing to get the provided service (as described in the next module).



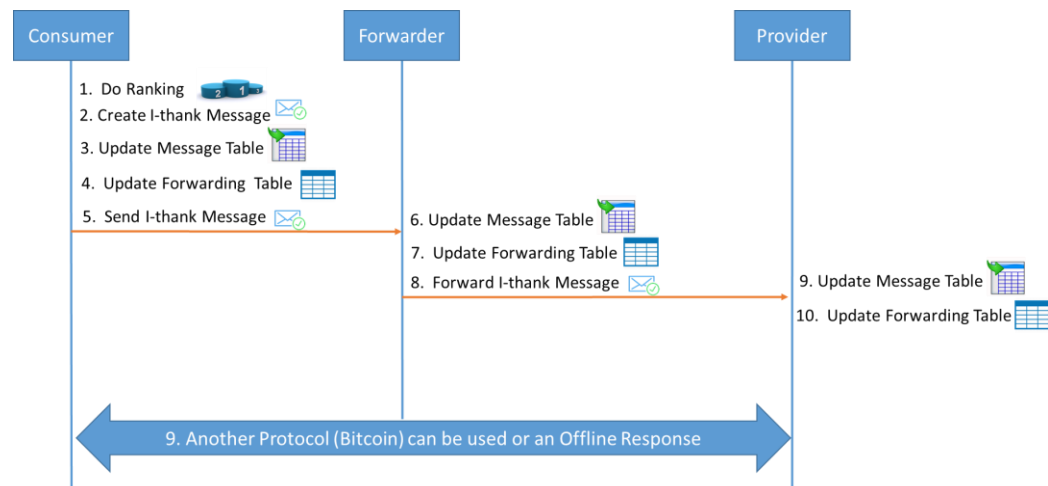
**Figure 12: Flow Diagram of I-have Module**

### 4.1.3 I-thank Module

The goal of this module is to transmit the I-thank message from its consumer to the provider through a set of forwarders. First (steps 1-5 in Figure 13), the consumer may receive a set of I-have messages with different HID and created at different times, but for the same I-need message. The consumer ranks them based on certain criteria (which are not discussed in this dissertation and kept open for application needs) choosing the top  $k$  ( $k=1,2,\dots,n$ ) I-have messages. An  $(k)$  I-thank-sure message is created and sent back to  $k$  providers. For reducing the protocol complexity, the consumer does not need to send  $(n-k)$  I-thank-but to other  $n-k$  providers. Instead, the consumer knows that after the deadline given in the I-have message, the intermediate nodes (forwarders) and the  $n-k$  providers will delete the I-need and I-have related entries. The MeT and FoT tables are then updated, and the I-thank-sure message is sent to the previous I-have message sender.

Second (steps 6-8), once the forwarder receives the I-thank-sure message, it inserts the message into the MeT table. Based on the keys associated with the message (HID and NID), the forwarders look up the FoT table and send the message to the previous sender of the I-have message. The forwarding process is one-to-one based on the information stored in the FoT table.

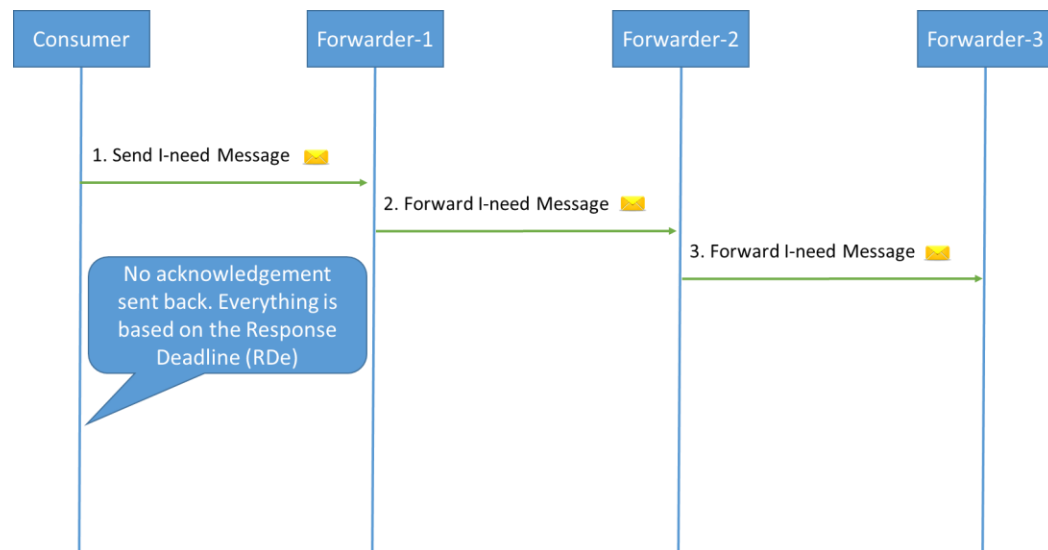
Finally, when the I-thank-sure message reaches the provider, the MeT and FoT tables are updated. Furthermore, a virtual channel can be established between the consumer and provider, where they can exchange information using the Piggyback Data field in the I-thank message (kept open for applications needs). If not, any other protocol (such as Bitcoin) can be used, or offline communication can be established based on the application requirement.



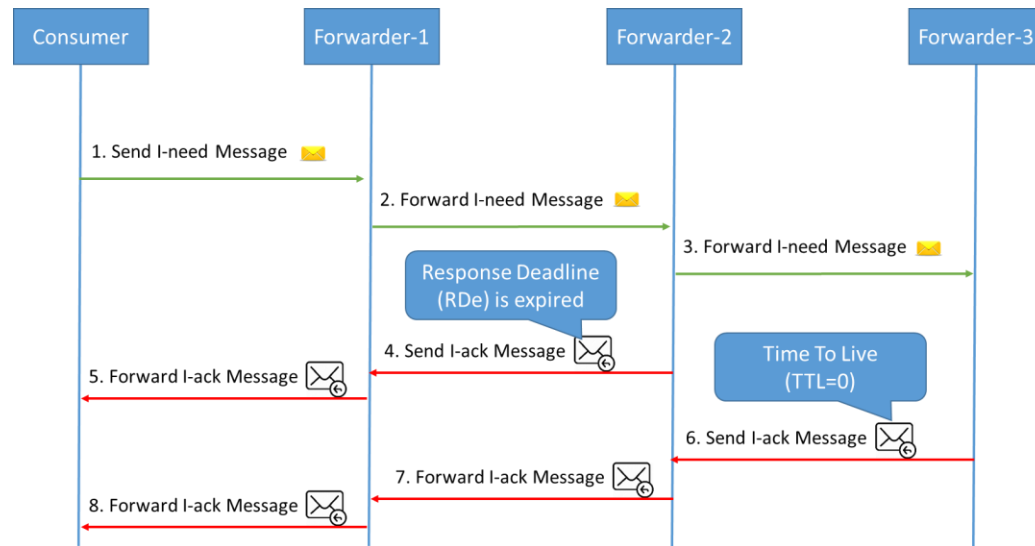
**Figure 13: Flow Diagram of I-thank Module**

#### 4.1.4 I-ack Module

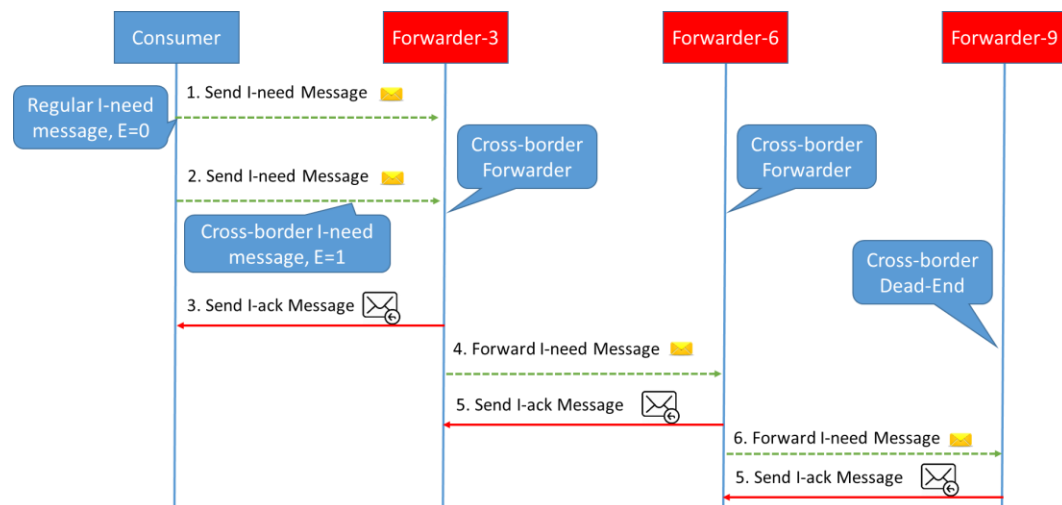
Milgram [Milgram, 1967] chose a socially high-status target person (stockbroker) in an urban area (Boston), meaning that there were plethora of paths toward the target, thus increasing the success rates of the search in the network. On the other hand, it is more difficult to find a low-status target in rural areas [Easley & Kleinberg, 2007]. This phenomenon is taken into account in this dissertation. Because of this, SOR provides three kinds of acknowledgment for collecting information about the I-need message (extending the search space in case no response occurs after the first try): a Send-with-Hope for high social status targets (Figure 14); and a Send-with-Knowledge (Figure 15), and an Extend-with-Knowledge (Figure 16) for low social status targets. These modules guarantee that the I-need message has reached large numbers of nodes in the network and has been neither blocked nor dropped.



**Figure 14: Flow Diagram of I-ack Module (Send-with-Hope)**



**Figure 15: Flow Diagram of I-ack Module (Send-with-Knowledge)**



**Figure 16: Flow Diagram of I-ack Module (Extend-with-Knowledge)**

- **Send-with-Hope** is the default way of sending the I-need message, where the main assumption is that the society is ideal and all nodes will be cooperating to forward the message. The consumer sends the I-need message in the OSN with the Acknowledge field CK=0, meaning that no acknowledgement is needed to be



sent back. The problem with this method is that the consumer does not get any feedback from others and cannot properly guess whether the I-need message reaches its target or not. However, a few mathematical proofs exist to show if the message will reach its destination in a particular time based on some assumptions, for example:

**Conjecture 1:** Based on Milgram's experiment [Milgram, 1967], there is a high chance that  $1/3$  of the message copies will arrive to the target, in a median of 6 steps.

**Lemma 1:** Kleinberg's small-world [J. Kleinberg, 2000] proves that the expected time a message will take to reach its destination is  $O(\log^2 n)$ .

**Lemma 2:** Chip & Van [Martel & Nguyen, 2004] prove that the expected time a message will take to reach its destination is  $O(\log^{3/2} n)$  for both the 2-dimensional model  $O(\log^{1+1/k} n)$  and the  $k$ -dimensional model (for  $k \geq 1$ ).

**Corollary 1.** An I-need message will reach the candidate providers in  $O(\log^2 n)$  at most. This corollary is inherited from Kleinberg's small-world theorems [J. Kleinberg, 2000].

It is assumed that nodes in a given OSN will follow the SOR protocol.

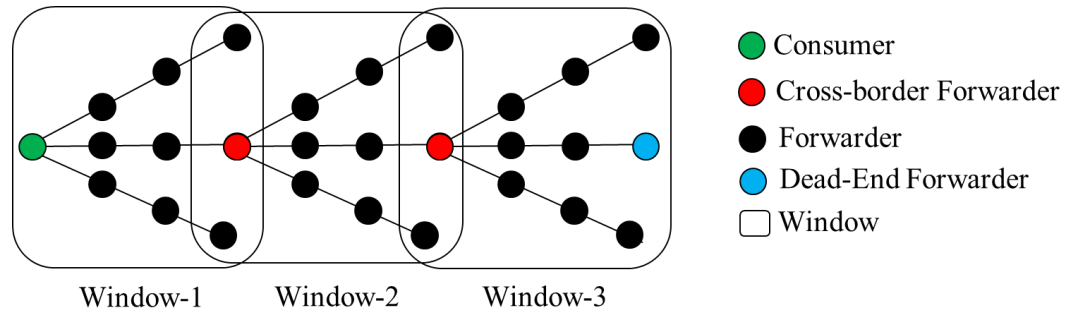
**Assumption 1.** Nodes follow the SOR protocol.

**Proposition 1.** If a node (Forwarder) does not follow the SOR protocol, then the messages (the I-need message and the I-have) will not reach the designated recipient (Provider, Consumer).

**Proof:** Let  $x$ ,  $y$ ,  $z$  be an arbitrary consumer, forwarder, and provider respectively in the graph  $G$ . Assume that there is only a single path between  $x$  and  $z$  through  $y$ . Also assuming the forwarder  $y$  is not participating in forwarding, it is impossible both for the provider  $z$  to receive an I-need message from the consumer  $x$  and for the consumer  $x$  to receive an I-have message from the provider  $z$ .

- **Send-with-Knowledge:** this is not the default method, and it should be used in a society where there is no guarantee that the nodes will participate in the forwarding or there is a suspicion that the propagation rules are not perfect due to 1) the heterogeneity of nodes and their social characteristics, 2) the service being rare, or 3) the service being urgent and sensitive. In this situation, the consumer sends the I-need message in the OSN with Acknowledge field  $CK=1$ , meaning the last node that will drop the I-need message because of the  $TTL=0$  or any other deadline (Response Deadline (RDe) is expired) must send acknowledgment back to the originator (consumer). This kind of method gives the consumer some information about the propagation of the I-need message, which in turn gives some guarantee that the service will be received.
- **Extend-with-Knowledge** is also not the default method and must be used by consumers to extend the search space. The default number of hops that the I-need message can go through is three. However, it can be extended to more hops. As shown in Figure 17, the 3-hops are known as a window, and only three nested windows are allowed. This method is used in societies where the network diameter is long or when the service is not available in the 3-hops search space

and will be available after that window. Special ways of thinking are needed to generate propagation rules for the second and third windows (this detail is intentionally left for the application designers). The consumer can then send the I-need message with  $CK=2$  after two scenarios: 1) the I-need message is sent with  $CK=0$ , the Response Deadline (RDe) expired, and no I-have message has arrived; and 2) the I-need message is sent with  $CK=1$  and no I-have message has arrived. The last node that dropped the I-need message must then do two things: 1) send acknowledgment back to the sender, and 2) forward the message to the next stage with new  $TTL=3$  and  $CK=1$ , and Extended ( $E=1$ ). The extender needs to keep the connectivity, QPs, and SPs in its table and clear them in the message. The Hop Count (HC) helps the receivers to know how many windows this message has passed. The maximum value of HC is 9. The problem with this method is that more network overhead might not be socially acceptable and can cause some privacy risks. However, the extend-with-knowledge provides more confidence and guarantees that the service will be obtained if it is in the OSN. On the second try, the I-need message does not need to be propagated to all neighbors. The consumer chooses some neighbors and sends to them, and the intermediate nodes choose one node and send to it because they know this is a Cross-border I-need message which has  $E=1$ . The Cross-border message needs Cross-border cooperation. Each Cross-border forwarder has to perform a routing computation, or it can just send the message back with a reverse path.



**Figure 17: Windows and forwarders types of Extend-with-Knowledge**

## 4.2 Human Queue Model

A set of queuing models was proposed by Barabási [Barabasi, 2005]. These models are: 1) First-In-First-Out (FIFO) model: this executes the tasks in the order that they were added to the list; 2) The Random model: this executes the tasks in a random order without acknowledgement of priority and arriving time; and 3) The highest-priority-first (HPF) model: this executes tasks with the highest priority first, even if they are added later in the list. Variants of the Barabási priority queuing model have been studied analytically in the literature (as discussed in Chapter 2). This study uses the simple priority queue model proposed by Barabási to model the human queue because of its simplicity and generality as well as being the basis for all other models. Barabási assumed that each individual has a priority list with  $L$  tasks, each task being assigned a priority value  $x_i \in [0, 1]$ , where  $i=1, \dots, L$ , chosen from  $\alpha(x)$  distribution. The priority comes from human decision-making, whenever an individual is presented with multiple tasks and chooses among them based on some perceived priority parameters. His model predicts the time interval between two consecutive actions by the same individual. The

simple priority queue model proposed by Barabási is used in a network of queues instead of one single node as he proposed.

A queue is a set of tasks waiting to be serviced, and queueing theory is the study of waiting times. The main notation in classical queueing theory is  $A/B/C/L/E$ , where  $A$  refers to the inter-arrival time distribution (e.g. exponential inter-arrival times);  $B$  indicates the probability distribution for service time (e.g. exponential distribution service times);  $C$  is the number of parallel servers (for instance, a single server);  $L$  is the queue length (for example, infinite queue size); and  $E$  is the queue discipline (e.g. FCFS). The symbols  $M$ ,  $D$ , and  $G$  refer to Markov (exponential) distribution, Deterministic distribution, and General (arbitrary) distribution, respectively. For example,  $M/M/1/\infty/FCFS$  (or  $M/M/1$  for short) represents a queueing model with exponential inter-arrival times (also called Poisson arrivals), exponential service times, a single server, an infinite queue size, and FCFS queue discipline [Bolch, Greiner, de Meer, & Trivedi, 2006]. Technically, the priority queue is a data structure for storing tasks with a priority. A priority is represented as a floating-point number between  $[0, 1]$ , where 0 is a high priority and 1 is a low priority, or vice versa based on the used system.

### **4.3 Social-based Routing**

Social based routing is the process of exploiting the social characteristics of nodes in OSNs to make a better routing decision by finding the best path from source the (provider) to destinations (consumers). There is a heightened interest in the social characteristics of individuals and how to exploit these characteristics efficiently in social-based routing schemes for OSNs [Boldrini, Conti, & Passarella, 2009; Othman & Khan,

2015; Wei, Liang, & Xu, 2014; Ying Zhu, Xu, Shi, & Wang, 2013]. Before proceeding to routing algorithms, however, a few concepts must first be defined and discussed.

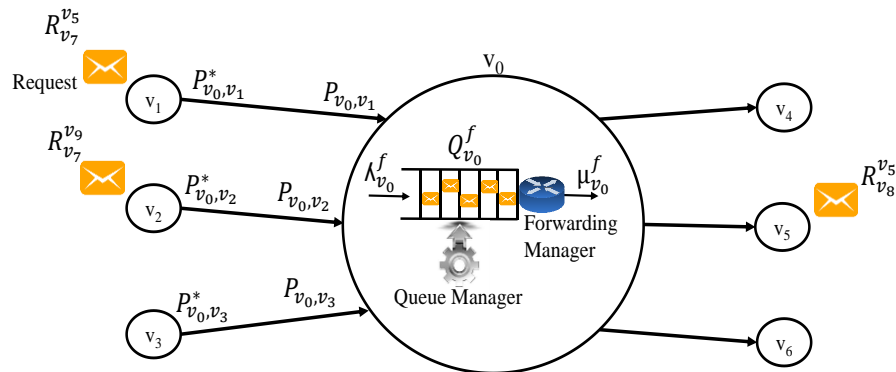
**Queuing Disciplines:** These are ways of governing how messages are buffered while waiting to be transmitted to the next hop or to get a service. The queuing algorithm determines which message is transmitted, serviced, or discarded, directly effecting the latency experienced by a message while traveling to its destination. It is assumed that each SOR node in OSN has two queues: Forwarding and Servicing. Popular queue disciplines in regular networks are first-in-first-out queuing (FIFO), priority queuing (PQ), fair queuing (FQ), weighted fair queuing (WFQ), weighted round-robin queuing (WRR), and deficit weighted round robin queuing (DWRR) [Semeria, 2001]. In online social platforms, each user has a feed (e.g. a Facebook user's wall, a LinkedIn user's timeline or a Google+ user's wall) which contains information from neighbors and is ordered by companies using a particular queueing algorithm. Users read this information by either using last-input-first-out LIFO, or by jumping forward and back.

**Node architecture:** As shown in Figure 18, each node is associated with a queue, a queue manager, and a forwarding manager. The queue (known as a forwarding queue and denoted as  $Q_u^f$ ) is a data structure for temporarily storing messages.

The queue manager utilizes queue disciplines for inserting, dropping, popping, and ordering messages assuming that each node uses only one discipline, either a First-Come-First-Service(FCFS) or a Social Priority (SP). In the future, complex scenarios such as changing queuing behavior with time can be studied.

The Forwarding queue has three parameters: 1) the message arrival rate,  $\Lambda_u^f$ , which is the number of messages arriving at queue  $u$  per unit time; 2) the message forwarding rate,  $\mu_u^f$ , which is the number of messages departing queue  $u$  per unit time; and 3) the forwarding queue length,  $L_u^f(t)$ , which is the number of messages in the forwarding queue of node  $u$  at time  $t$ .

The forwarding manager forwards the messages to the next neighbor based on the forwarding strategies. There are other components in the node model, but this section focuses on the forwarding queue, the queue manager which utilizes queue disciplines for inserting, dropping, popping, and ordering requests, and the forwarding manager which forwards the messages to the next neighbor based on forwarding strategies.



**Figure 18: Node Anatomy**

**Social Priority (SP):** From node  $u$  to node  $v$ , the friendship edge,  $e_u$ , is associated with two values as shown in Figure 18: an In-Social Priority ( $iSP$ ) for forwarding that represents a form of proportionate priority with which  $v$  will treat a message arriving from  $u$ , and an Out-Social Priority ( $oSP$ ) for determining the best path to forward.

Although the value of  $iSP$  is known to  $v$ ,  $v$  can not be expected to reveal it candidly to  $u$ ; therefore,  $u$  continually learns  $oSP$ , which is an estimate of  $iSP$ . If node  $u$  makes a correct estimation, then  $oSP = iSP$ . Gender, Degree, Betweenness, Closeness, and Eigenvector centralities are used as social characteristics of individuals in OSN. To generate social priorities for all potential senders (receivers), each node uses its own set of factors and uses singular value decomposition (SVD) [Othman & Khan, 2015] to generate a SP vector for the immediate in(out) circle made of adjacent neighbors. Estimating social priorities using a matrix factorization technique is discussed in detail in CHAPTER 6.

**Social Priority-based Path Delay (SPPD) Metric:** This subsection, describes the SPPD metric and the information it needs. The objective is to determine the end-to-end delay, which is experienced by a message  $R_s^d$  through paths from a source (provider) node  $s$  to a destination (consumer) node  $d$ . Here, it is assumed that 1) the node can use only one queue discipline (SP or FCFS), and 2) the node  $u$ 's queue parameters ( $\lambda_u^f$ ,  $\mu_u^f$ ,  $L_u^f(t)$ ) are collected from I-need messages. Table 1 summarizes the used parameters.

**Table 1: Queue Parameters**

Parameter	Description
$L_u^f(t)$	the number of requests in forwarding a queue of node $u$ at time $t$ .
$\lambda_u^f$	the number of requests arriving at $u$ 's queue per unit time
$\mu_u^f$	the number of requests departing the $u$ 's



	queue per unit time
$oSP_{u,v}$	Out-social priority
$B_u(t)$	The queue discipline of node u at time t. It is fixed all times.

Given a message  $R_{v_1}^{v_k}$ , a simple path  $P_{v_1, v_k} = (v_1, \dots, v_k)$ , and all parameters of intermediate nodes as depicted in Table 1, how can the expected end-to-end delay that a message will experience through the given path be found? To answer this question, the following two equations are presented:

$$T_{v_i}^f = \frac{L_{i+1}^f(t_0) + T_{v_{i-1}}^f * \kappa_{v_{i+1}}^f}{\frac{\mu_{v_{i+1}}^f}{oSP_{v_{i+1}, v_i}^f} - \kappa_{v_{i+1}}^f} + \frac{oSP_{v_{i+1}, v_i}}{\omega} \quad (1)$$

where  $T_{v_0}^f = 0$  and  $i = 1, 2 \dots v_{k-1}$ ,  $\omega$  is a constant value.

Generally, the end-to-end delay for any number of intermediate nodes in the simple path is computed by the equation below:

$$T_{end-to-end}(v_1, v_k) = \sum_{i=1}^{k-1} T_{v_i}^f + c \quad (2)$$

The equations can be modified to calculate the FCFS queue discipline by assigning one to  $oSP_{v_{i+1}, v_i}$ , meaning that the position of the request will be at the bottom of the queue regardless of the Out-social priority value.

**Routing Algorithms:** The provider collects the connectivity information, social priorities (SPs), and queue parameters (QPs) from the I-need messages and builds a graph

$G = (V, E)$ , as a directed graph where  $V$  is a set of  $n$  nodes and  $E$  is a set of  $m$  edges in the graph. Let  $e_{u,v}$  denote a link (social relationships) of the graph connecting a pair of nodes  $(u, v)$  and let  $P_{u,v}$  denote a path between the source (provider), the node  $u$ , and the destination (consumer) node  $v$ . This path consists of a series of intermediate nodes (forwarders). Based on the available knowledge in the graph, the provider decides which routing algorithm will be used. In SOR, routing is computed using a Topology aware Shortest-Path, Social-Priority-Based, or Queue aware Social-Priority-Based routing algorithm (as shown in Algorithm 1 and Algorithm 2).

#### **4.3.1 Topology aware Shortest-Path-Based Routing Algorithm (CSP)**

In cases when only connectivity information is shared, the CSP algorithm uses only connectivity information collected by the Connectivity Manager (which is responsible for propagating, receiving, and managing connectivity information). It is based on Dijkstra's algorithm. In the study's simulation, it uses only hop counts and emulates classical routing. [Demetrescu & Italiano, 2004].

#### **4.3.2 Social-Priority-Based Routing Algorithm (SPBs)**

In cases between privacy choices, the SPBs algorithm adds an SP collected by a Social Priority Manager (which calculates social priorities for incoming adjacent neighbors and estimates social priorities given by outgoing adjacent neighbors). It uses a Prioritized metric (the minimum sum of the SP between source and destination) to evaluate the best path for a request to travel. It is also based on Dijkstra's algorithm. In

this static version, the queuing load is assumed to be zero (or constant at the time of the snapshot).

### 4.3.3 Queue aware Social-Priority-Based Routing Algorithm (SPB<sub>D</sub>)

In the case of the most open choices, the SPB<sub>D</sub> algorithm adds dynamic queue status information with a special priority collected by both the Queue Manager and the Social Priority Manager, respectively, to evaluate the best path for a message to be assigned. It is simply a modified version of Dijkstra's algorithm.

The time complexity of CSP, SPB<sub>S</sub>, and SPB<sub>D</sub> are  $O((V+E) \log(V))$ ; however, the SPB<sub>D</sub> algorithm is called for each message  $m$ .

---



---

#### Algorithm 1 BestPath (G)

---

**Input:** G (V, E, P, QP) priority values P, queue parameters QPs

```

1: For all s ∈ V in G do
2:   For all d ∈ V in G do
3:     Path ← Single-Source-Dijkstra(G, s, d) {Get the best path from s to d}
4:     Paths ← Path
5:   end for
6: end for
7: return Paths {All pairs best paths in G}

```

---

### Algorithm 1: Best Path

---



---

#### Algorithm 2 Single-Source-Dijkstra (G, s, d)

---

**Input:** G, Source s ∈ V, destination d ∈ V

Dist. [start node] ← 0, Dist. [Others] ← ∞

```

1: While still in G do
2:   u ← Choose the node with the least cost
3:   Remove u from graph or set it visited
4:   For each v ∈ N(u) do {Neighbors of u}
5:     Get-Cost {To compute the cost of edge (u, v)}
6:     Calculate cost between u and v {not visited}
7:     Update Costs
8:     Choose the lowest
9:   end for
10: End While

```

11: return Path

---

## Algorithm 2: Single Source Dijkstra

### 4.4 Experiment Validity

In the literature, to evaluate any routing algorithm two things need to be considered: datasets and a base algorithm to compare these.

#### 4.4.1 Choice of dataset

Three possible datasets that can be used to evaluate the study's proposed routing algorithms are the following:

1. The dataset (Infocom 2006 trace): This contains contacts between devices carried by participants for four days of a conference. The dataset is collected to monitor the presence of people in a conference environment. Its traces contain time-stamped information about the location of each user throughout the period of the conference [F. Li et al., 2013]. Even though, it is used to evaluate some routing algorithms, it is not a real social network and does not contain social features about users. Thus, it cannot be used to evaluate the routing algorithms in this study.
2. Synthetic datasets: These contain social networks generated by special algorithms using distribution (e.g. power laws) such as those proposed by Barabasi [Bu & Towsley, 2002]. Although scientists claim that the generated graph exhibits the small world properties and its topology is well described by power laws, these kinds of datasets may not be the best

way to evaluate the study's proposed algorithms since there are no social characteristics.

3. Real-world OSN datasets: These contain real data about individuals, their social features and their social ties. Three well-known datasets of this type are Facebook, Google+, and Twitter. They are used to evaluate community detection in networks, along with other social properties [Goga, Loiseau, Sommer, Teixeira, & Gummadi, 2015; Gong et al., 2014; Gong et al., 2012; Gonzalez, Cuevas, Motamedi, Rejaie, & Cuevas, 2013; J. Jia, Wang, Zhang, & Gong, 2017; Kairam, Brzozowski, Huffaker, & Chi, 2012; Kong, Liu, & Huang, 2014; J. McAuley & Leskovec, 2014; Pontes et al., 2012; Vesdapunt & Garcia-Molina, 2016; Yang, McAuley, & Leskovec, 2013]. Facebook is the best option, but there is no available online dataset associated with its social features. Twitter is not the best candidate because the underline structure of its network does not reflect that of a real social network. Because of this, Google+ is currently the best option available to evaluate the study's proposed algorithms.

#### **4.4.2 Choice of algorithms**

The efficiency (end-to-end routing delay) is generally influenced by the information availability. According to the Stratified Privacy Model, individuals have different privacy options and based on that different information elements can be shared. However, in this study, three algorithms were proposed based on three available information elements. Firstly, in cases where only connectivity information is shared, the

Topology aware Shortest-Path-Based Routing Algorithm (CSP) was used. Secondly, in cases between privacy choices (e.g. priority information is shared), the Social-Priority-Based Routing Algorithm (SPB<sub>S</sub>) was used. Finally, in cases where most open choices where all information is shared, the Queue aware Social-Priority-Based Routing Algorithm (SPB<sub>D</sub>) was used. Although there are other possible cases, this study only focuses on these three cases.

Routing algorithms like Floyd–Warshall [Katz & Kider Jr, 2008; Solomonik, Buluc, & Demmel, 2013], Dijkstra [Brodka, Stawiak, & Kazienko, 2011; Ivanov, Kupriyanov, & Shichkina, 2017; Koutsopoulos, Noutsis, & Iosifidis, 2014] or any other one in the literature [Medhi & Ramasamy, 2017] can be used as a base for the proposed algorithms of this study. Research from this study found that Dijkstra's algorithm is widely used in conventional network routing protocols, most notably IS-IS (Intermediate System to Intermediate System) and Open Shortest Path First (OSPF). Thus, Dijkstra is universally adapted in research as well as practical papers. For this reason, Dijkstra's algorithm was adopted as the base for the study's routing algorithms.

#### **4.5 Experiments on the Efficiency of the Routing Protocol**

An OSN simulator using OMNeT++ (discussed in detail in CHAPTER 7) was designed for this study. Three routing algorithms (CSP, SPB<sub>S</sub>, and SPB<sub>D</sub>) were then implemented in the simulator. A set of experiments were then conducted on each algorithm. Experiments differ in values which were assigned to generated rate, forwarded rate, and served rate parameters. Exponential distributions were assigned to the parameters inter-generate, inter-forward, and inter-serve with different mean values. A

small number of messages for each node in the network to be sent to a randomly chosen destination were chosen in order to avoid heavy traffic network in all experiments.

**Performance Metrics:** The study applies various quantitative interrelating adhered metrics to evaluate the performance of its algorithms. The most important metrics, Delivery Latency, Throughput, and Average hop count have been considered. **Delivery Latency** is the average time needed to finish transmitting messages to their destinations. A delay due to route discovery latency and queuing is known as end-to-end routing delay, and end-to-end total delay is the end-to-end routing delay plus the service delay of message. **Throughput** refers to the number of messages transferred from a source to its destination in a specified amount of time. Typically, throughputs are measured in kbps, Mbps and Gbps, or (in the case of this study) Mpss (Message per simulated second). **Average Hop Count** is the number of intermediate nodes through which a message passes from source to destination in a particular path. These performance metrics were chosen because they are the most common metrics used to evaluate the performance of networks in the literature.

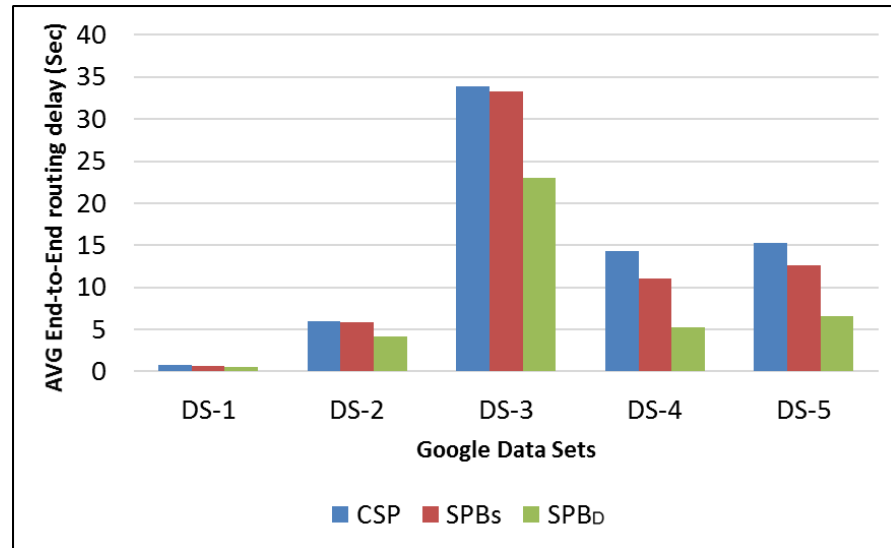
**Datasets:** This study's algorithms are evaluated on a real-world OSN with heterogeneous social characteristics. For evaluation, five datasets downloaded from Jure Leskovec's Website [Jure Leskovec, 2014] were used. The datasets were crawled in 2012 from a popular social network site, plus.google.com. Some of the statistical information of the five datasets is summarized in Table 2, where DS refers to Dataset.

**Table 2: Statistical information of Google+ datasets**

<b>Dataset</b>	<b># nodes</b>	<b># edges</b>
DS-1	54	203
DS-2	116	1024
DS-3	342	4176
DS-4	1648	166291
DS-5	2211	93509

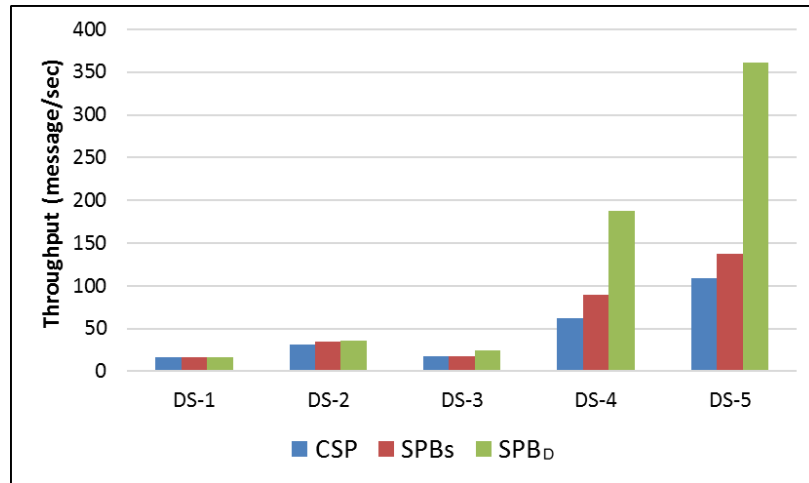
The main Figure 19 compares the average end-to-end routing delay of the three algorithms over five different social graphs of Google Plus. In general, the SPB<sub>D</sub> (Green Color) algorithm performs the best and the CSP (Blue Color) algorithm performed the worst among the three algorithms. In the case of small scale social graphs, however, the SPB<sub>S</sub> (Brown Color) algorithm is very close to the performance of CSP. It performs better in large scale social graphs due to the lack of paths between sources and destinations in small social graphs, as well as the number of paths in case of large scale social graphs.





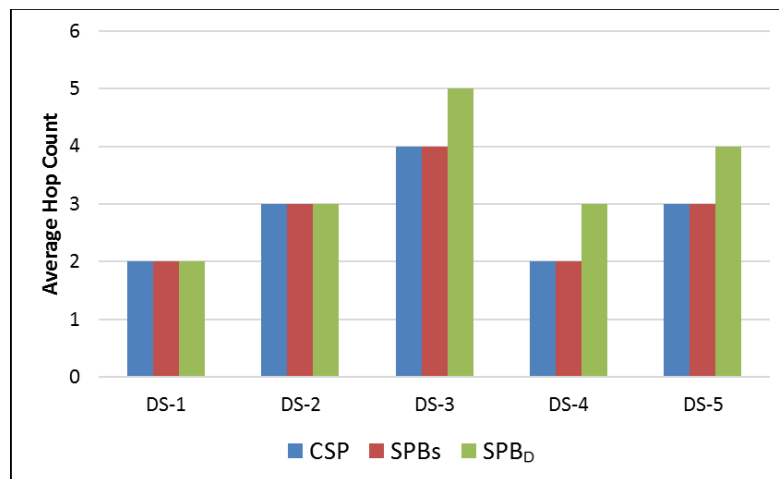
**Figure 19: End-To-End Routing Delay**

Throughput of the three algorithms is compared. It can be noticed, as shown in Figure 20, that the throughput using SPB<sub>D</sub> algorithm with large scale network increases significantly when the node forwarding rate is high, increasing steadily when the node forwarding rate is low. In the case of small scale networks, throughputs are nearly the same, except when the node forwarding rate is high. Here, the throughput using SPB<sub>D</sub> algorithm goes up slightly.



**Figure 20: Network Throughput**

Despite the significant improvements in average end-to-end delay and throughput, the average hop count increases slightly in SPB<sub>D</sub> algorithm and is equal in both CSP and SPB<sub>S</sub>. As shown in Figure 21, in large-scale social graphs, if the forwarding rate of each node is high and the algorithm SPB<sub>D</sub> is used, the average hop count increases.



**Figure 21: Hop Count**

A few other interesting questions related to social routing in OSN involve the strategic location of nodes. The impact of knowledge and node location in OSN is studied, and a few questions to answer here are as follows:

**Does Knowledge Matter?** The choice of a social-based route in a timely and efficient manner is an intricate decision process and requires full and accurate knowledge of individuals in the OSNs. The impact of the imperfect knowledge about the intermediate nodes' social priorities is studied here, along with queue discipline on both end-to-end delays and queue sizes. Understanding this kind of impact is critical, especially with the existence of privacy and security constraints that might cause knowledge imperfection. A simulation study has been conducted to check this impact in different societies. This study's experiments show that for these societies with imperfect knowledge, knowing social priorities has more impact on end-to-end delay than knowing the queue discipline. Furthermore, when perfect knowledge of social priorities is available, end-to-end delays decrease. See [Othman, Khan, & Nafa, 2016a].

**Does Location Matter?** Centrality metrics such as Closeness and Betweenness in an Online Social Network (OSN) determine how much end-to-end delay and queue-load a node can have as a source or as a destination through Social Routing. It was found in this study's experiments that nodes with high Out-Closeness centrality in an OSN suffer from a high end-to-end delay as a target, but not as a source. The study also shows that the cause of this end-to-end delay is that most nodes with high Out-Closeness centrality have a low In-Closeness centrality. Moreover, the increase in the local In-Degree centrality will increase the global In-Closeness centrality, and that the promised level to increase

the In-Closeness centrality of a node is its Friends of Friends-Of-Friends (Level-3). A simulation study is also completed by propagating a set of messages in different societies with different routing schemes and diverse queue disciplines to compare the average end-to-end delays from the source and target perspectives. See [Othman, Khan, & Nafa, 2016b].

#### **4.6 Conclusion**

Experiments have been reported on the efficiency of the routing protocol. The results show that the average end-to-end routing delay is generally influenced by the information availability. In the case of most open choices, near optimum performance is achieved when all information is shared. However, in OSNs, there is no guarantee that all information will be shared. In cases between privacy choices (e.g. priority information is shared) the performance is degraded gracefully. The next chapter, will show that reachability is guaranteed in cases of the most restricted choices (restrictive privacy); where one chooses the null identity, it does not allow for the propagation of connectivity, and shares no queue or priority information.

## CHAPTER 5

### **Study on The Privacy Preservation Ability of SOR**

This chapter proposes three defensive mechanisms for stratified privacy in OSN using SOR. It also introduces Proxima Matrix and Proxima distributions to analyze the degree of anonymity of the service consumer and shows some strategies for securing SOR protocol.

#### **5.1 Privacy Background and Terminology**

To guarantee privacy in OSN, certain requirements must be considered. These requirements are anonymity, unlinkability, unobservability, and pseudonymity [Bai, Liu, & Liu, 2009; Yanes, 2014]. The definitions used by Pfitzmann [Pfitzmann & Hansen, 2010] were adopted for these requirements. *Anonymity* is thus defined as “the state of being not identifiable within a set of subjects, the anonymity set”. *Unlinkability* means hiding the relationships between items (e.g. subjects, messages, events, actions). *Unobservability* is defined as hiding the items themselves. *Pseudonymity* means that pseudonym or alias is used instead of the real identity. There are different techniques to achieve these requirements (e.g. unobservability is mainly achieved by injecting dummy messages into networks). SOR uses special strategies to preserve privacy in OSN (which will be discussed later).

Three types of anonymity have been defined in privacy literature: sender anonymity, recipient anonymity, and relationship anonymity [Diaz, Troncoso, & Serjantov, 2008; Shmatikov & Wang, 2006]. *Sender anonymity* denotes that the identity

of the message sender is hidden, making it impossible to link any message to a particular sender and vice-versa. Similarly, *Recipient anonymity* implies that the identity of message receiver is hidden; thus, it is impossible to link any message to a particular receiver and vice-versa. *Relationship anonymity* implies that the identities of both the message's sender and receiver are hidden, rendering it impossible to prove that a particular sender and a particular receiver are involved in communication with one another. SOR looks at and utilizes three types of anonymity: service consumer anonymity, service provider anonymity, and service forwarder anonymity.

The anonymity metric is used to determine the degree of anonymity a system provides against a specific anonymity attack. However, measuring the *degree of anonymity* is not a trivial task [Tillwick & Olivier, 2005]. Furthermore, according to the literature, there is no consensus metric that should be used to quantify anonymity. Thus, researchers have proposed three basic metrics to quantify anonymity: *Anonymity Set Size*, *Effective Anonymity Set Size*, and *Entropy-based Anonymity Degree* [Ye Zhu & Bettati, 2005]. Many of the anonymity metrics in the literature expand upon one or more of these metrics. Anonymity Set Size is the traditional way of measuring the degree of anonymity, where the size of an anonymity set is used to indicate the degree of anonymity provided by the system (The larger the Anonymity Set Size, the higher the degree of anonymity)[Andersson & Lundin, 2007; Murdoch, 2014; Ye Zhu & Bettati, 2009]. This dissertation introduces a distribution called Proxima that helps to estimate the anonymity set size.

Based on the model of background knowledge that an adversary may use to attack the privacy of individuals in an OSN, anonymity attacks can be broadly classified into two types: a profile attack and a structural attack [Yuan, Chen, & Yu, 2010]. A profile attack is the ability of an attacker to identify individuals in OSN based on profile information. For example, individuals in the United States can be recognized uniquely by zip code, gender, and date of birth [Sweeney, 2000]. The structural attack is the ability of an attacker to identify individuals in an OSN based on structural information. They use two types of structural attack: 1) Degree-based attack: the attack is carried out based on the number of neighbors of a certain individual [Holme, Kim, Yoon, & Han, 2002]. 2) Subgraph-based attack: the attack is carried out based on subgraph information (the victim's directly connected nodes and social ties) of certain individuals in the OSN, where the victim is in the subgraph [Ying & Wu, 2008]. This study focuses only on the Subgraph-based attack, which implicitly includes the Degree-based attack.

Attackers have different abilities and deficiencies, and clear assumptions have to be defined about an attacker when measuring anonymity [Diaz, 2006]. An attacker could be *passive* (listens to the communication and performs traffic analysis) or *active* (adds, removes or modifies messages). He or she could also be *internal* (controls nodes) or *external* (control links); *partial/local* (access to part of the network) or *global* (access to the entire network); *static* (unable to alter their behavior) or *adaptive* (able to modify their behavior); and *permanent* (knows the whole history of a network since it started functioning) or *temporary* (knows information starting from a specific point in time and

have no information prior to that point). This dissertation uses the assumption that the attacker is passive, internal, partial, static, and temporary.

## 5.2 Privacy in Social Networks

A social network consists of nodes, edges, and attributes associated with each node and edge. The attributes could be sensitive or non-sensitive. However, privacy is not just related to those sensitive attributes. Sometimes knowing that the individual is a node in a particular network is a privacy issue (e.g. a sexual relationship network). Zhou and his colleagues [Zhou, Pei, & Luk, 2008] list the pieces of information of an OSN that could be considered privacy of individuals.

- *Node existence*: This is whether a target individual appears in the network. Examples: a social network of millionaires, a sexual relationship network or a disease infection network.
- *Node properties*: This includes the degree of the vertex. Examples: in a financial support network, if the adversary knows the degree of the target then he or she knows how many support sources the target has. The adversary can also guess whether the victim is a community leader or not by knowing his or her distance to the center of the network.
- *Link relationship*: This is whether an edge exists between two target individuals. For example, in a business network, there is a financial transaction. In a phone call network, there is a call.
- *Link weight*: The edge weights can reflect the social priority, trust degree, frequency and duration of communication, and so on.



- *Graph metrics*: This includes the betweenness centrality (the degree an individual lies between other individuals in the OSN equal to the number of shortest paths from all individuals to all others that pass through that individual); closeness centrality (the degree that an individual is near to all other individuals in the OSN and equal to the average shortest distance from each individual to each other individual); path length (the distances between pairs of individuals in the OSN); and reachability (the degree any individual of a OSN can reach other individuals of the OSN).

SOR provides simple mechanisms to battle privacy attacks. The next section discusses the privacy information which might be under attack when SOR is used and how privacy information can be preserved.

### **5.3 SOR Functionality and Privacy Issues**

Most network protocols (e.g., BGP, RIP, OSPF, etc.) require information about network topology/connectivity information to function. SOR is one of them. Service providers in SOR need connectivity information, social priorities (SPs), and queue parameters (QPs) to compute the best path toward the service consumers. However, connectivity information of OSNs is more sensitive than the network topology of the internet and can cause some privacy issues. Thus, any protocol proposed for OSNs must minimize the loss of privacy.

Three fields of the I-need message are responsible for carrying connectivity information to the service providers: the Pathway ID Sequence (CO), the Pathway Social Parameter Sequence (SP), and the Pathway Queue Parameters Sequence (QPs). Each

intermediate node (Forwarder) is required to add itself to the I-need message by its next direct neighbor (This technique is discussed in detail in section 5.5). Each I-need message carries a path from the service consumer to the service provider. Collecting these paths gives partial connectivity information of the OSN. This partial connectivity information can be used by adversaries to violate the privacy of individuals in OSN. However, SOR provides some privacy protection mechanisms that can be used to preserve privacy of individuals in OSN.

#### 5.4 Stratified Privacy Mechanism in SOR

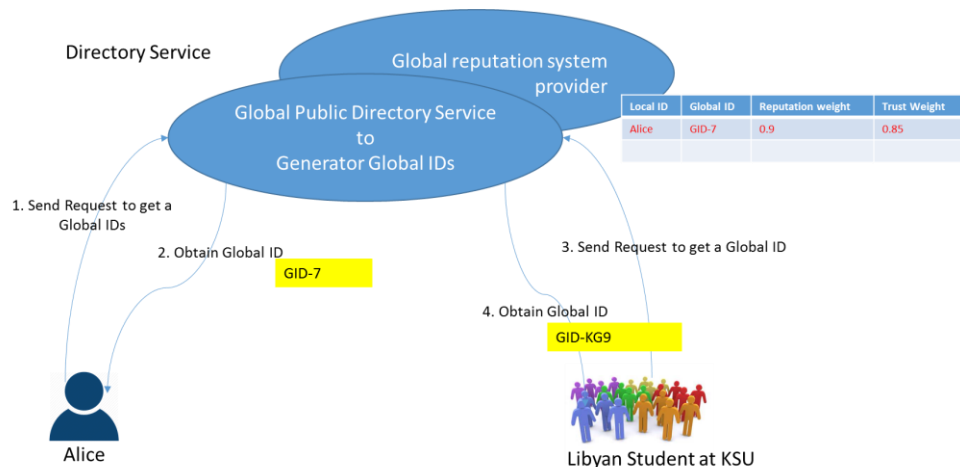
SOR provides three defensive mechanisms: Globally unique Pseudo Identity, Locally unique Pseudo Identity, and Null Identity. The Global, Pseudo, and Null identities can be used by service consumers and forwarders to preserve their privacy by anonymizing the communication. However, this study does not claim that this method can provide absolute privacy preservation. Rather, it seeks to introduce simple stratified privacy preserving mechanisms that can help service consumers, forwarders, and providers to be safe against identity disclosure as one of the main privacy issues when SOR is used in OSN.

Two kinds of societies are defined here: Ideal and factual. The ideal society is a society free from adversaries, so the individuals' Local Identities can be used. The factual society is a society where subsets of nodes could be adversaries, so Global, Pseudo, and Null identities could be used instead of Local Identities.

**Real Identity** is the actual unique label/identifier of the node in a network and only known by its adjacent neighbors. Let  $LID = \{id^{L_1}, id^{L_2}, \dots, id^{L_n}\}$  be the set of local

identities of nodes. Let  $G^L_v(V^L_v, E^L_v)$  denote a graph with local identities collected by a node  $v$ . This type of identity can be added to the I-need message, but privacy will not be guaranteed, because disclosing these identifies beyond direct neighbors allows adversaries to easily identify victims.

**A Globally unique Pseudo Identity** is given by an authorized centralized trusted organization (e.g. Directory Service) and known to everyone in the network. Let  $GID = \{id^G_1, id^G_2, \dots, id^G_n\}$  be the set of global identities of nodes. Let  $G^G_v(V^G_v, E^G_v)$  denote a graph with global identities collected by node  $v$ . Adding this identity to the I-need message could breach the privacy of graph metrics. Figure 22 demonstrates an example of how individuals and groups request global IDs from a directory service. This identity is introduced for certain applications requiring cumulative credit for social or economic benefits. However, because it could be a central point of failure, it is not encouraged to use this kind of identity.



**Figure 22: Steps for getting Node and Group Global IDs**

**A Locally unique Pseudo Identity** is a unique random pseudo ID used place of the Real identity of the node. This mechanism is equivalent to a perturbation-based scheme [Aggarwal & Philip, 2008; He, Liu, Nguyen, Nahrstedt, & Abdelzaher, 2007]. The perturbation-based scheme is used by organizations, such as government agencies or hospitals, to anonymize networks by adding noise (injecting random nodes and edges) to achieve privacy before releasing them to a third-party for different purposes (e.g., analysis) [Cao & Karras, 2012]. The two main differences between the Locally unique Pseudo Identity and a perturbation-based scheme are that 1) this study's chosen mechanism is done locally by each node based on a few predefined rules, whereas the perturbation-based scheme is done globally by an algorithm that matches each node with others in the graph; 2) while the perturbation-based scheme is applied offline and with full knowledge of nodes in the graph, this study's mechanism is performed online with partial knowledge about the network. Let  $PID = \{id^P_1, id^P_2, \dots, id^P_n\}$  be the set of pseudo identities of the nodes. Let  $G^P_v(V^P_v, E^P_v)$  denote a graph with pseudo identities collected by node  $v$ . Adding this identity to the I-need message provides an acceptable level of privacy, making it more difficult for adversaries to identify victims.

**A Null Identity** is a blank ID used by nodes to hide their local identities. This mechanism is equivalent to deleting random nodes and edges in the perturbation-based scheme to achieve privacy [Masoumzadeh & Joshi, 2012]. Let  $G^H_v(V^H_v, E^H_v)$  denote a graph with Null identities that node  $v$  has collected. Adding this identity to the I-need message provides a high Level of privacy. The I-need message with hidden nodes cannot

go more than  $TTL=z$  steps in the network. In this work,  $z=3$ , but based on the application, it can be more or less.

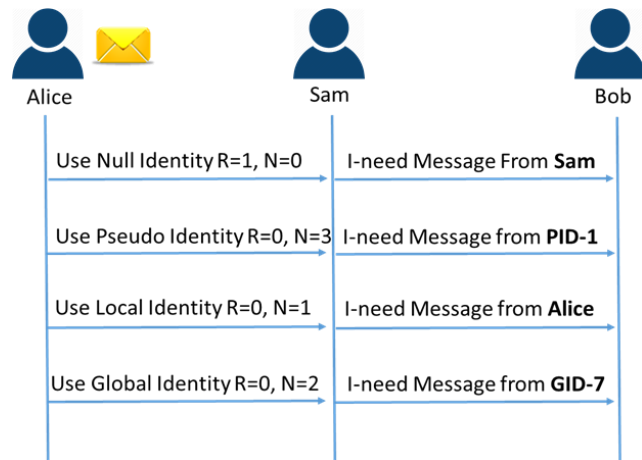
Each node can use one or more of these defensive mechanisms, giving it a mixed graph. For simplicity, all nodes are assumed to be able to use any one of these defensive mechanisms, but not all of them combined.

### 5.5 Peer Privacy Request:

The SOR node relies on its adjacent neighbors to help it to be hidden or anonymized by using the aforementioned defensive mechanisms. Particularly, when node  $u$  creates or forwards an I-need message, it requests its adjacent neighbor to hide or anonymize it by using fields Peer Privacy Request (R) and Peer Anonymize Name (N) of the I-need message. Table 3 shows the values of these fields. For example, consider a service consumer “Alice” and her direct neighbor “Sam” (as shown in Figure 23). “Alice” should request “Sam” to hide her by using  $R=1$  and  $N=0$ , to anonymize her using  $R=0$  and  $N=3$ , to disclose her real identity using  $R=0$  and  $N=1$ , or to use her Globally unique Pseudo Identity using  $R=0$  and  $N=2$ .

**Table 3: Privacy Fields of the I-need Message**

<b>Peer Privacy Request (R)</b>	0: Anonymize	1: Hide
<b>Peer Anonymize Name (N)</b>	0: None	1: Local (IP)
	2: Global	3: Random



**Figure 23: Using Defensive Mechanisms by Alice**

It can be assumed that there is a certain level of trust between each node and its adjacent neighbors.

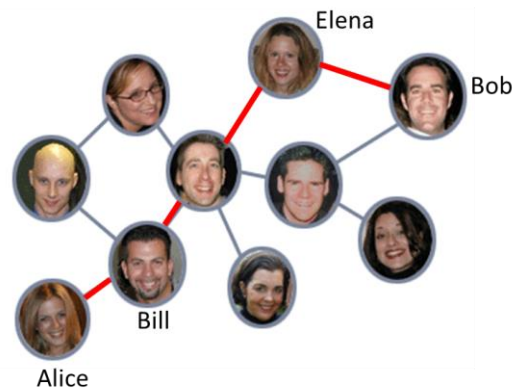
## 5.6 Attacker Model

This section defines the four main actors of any attack in OSN: the assumptions, the attacker's background knowledge, the goal of the attack, and the theoretical attack analysis using Proxima.

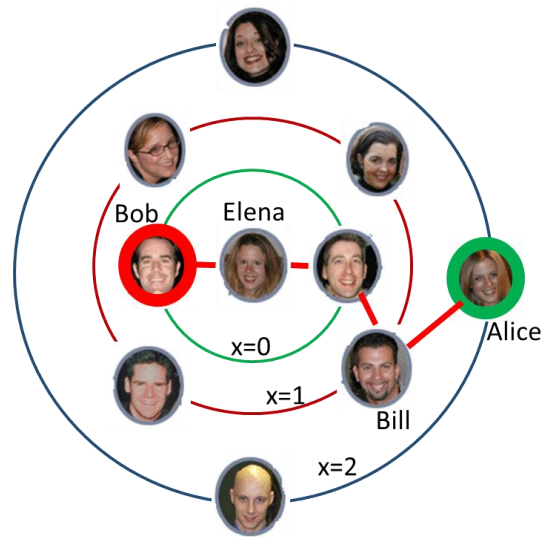
### 5.6.1 Definitions and Assumptions

The nodes that are part of the attack in OSN are classified as: victim node, first-forwarder node, last-forwarder node, and attacker node. The analysis is based on the last-forwarder, where all other nodes are at different circles of the last forwarder. For example, the attacker is the direct friend of the last-forwarder and the victim could be at any level of the last-forwarder node's friend circles. Figure 24 shows a simple social network of ten nodes and eleven edges with Alice as a victim, Bill as the first-forwarder,

Elena as the last-forwarder, and Bob as an attacker. Furthermore, the attack edge (path),  $P$ , which is colored in red, starts at Alice and ends at Bob. Figure 25 shows three circles of Elena's friends: Direct friends, where  $x=0$ ; friend-of-friends, where  $x=1$ ; and friend of friend-of-friends, where  $x=2$ . It also shows the position of the first-forwarder, the victim and the attacker in Elena's circles. This is a way of categorizing and describing attack actors in OSN.



**Figure 24: Illustration of Victim, First-Forwarder, Last-Forwarder, Attacker Nodes and the Attack Path.**



**Figure 25: Elena's Circles of Friends**

**The Victim node** is the sender of the I-need message and the first node of the attack path,  $\mathcal{P}$ . In Figure 24, Alice is the victim who creates the I-need message and hopes to be hidden or anonymized. So, no one except Bill knows about her.

**The First-forwarder node** is the first forwarder of the I-need message and second node in  $\mathcal{P}$ . Here, Bill is the first-forwarder who hides or anonymizes Alice.

**The Last-forwarder node** is the last forwarder of the I-need message to the attacker and the adjacent neighbor of the attacker in  $\mathcal{P}$ . Elena is the last forwarder who gets the I-need message, forwarding it to Bob. She does not know that he is an attacker.

**The Attacker node** is the last receiver of the I-need message and the last node in  $\mathcal{P}$ . It tries to collect information about the network and identifies the senders (service consumers), forwarders, or receivers (service providers) of the I-need messages. Bob is the attacker who gets the I-need message from Elena and tries to identify the owner (Alice).

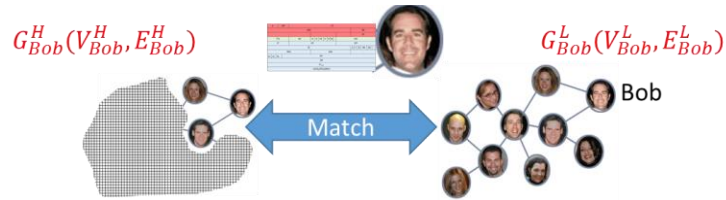


The following are some basic assumptions about an attacker: The attacker is 1) Passive (listens to the communication and reads internal information of nodes); 2) Internal (controls one node that are part of the system); 3) Partial (only sees a limited number of nodes which are direct neighbors and which receive an I-need message through them); 4) Static (unable to alter his or her behavior); and 5) Temporary (knows information starting from a specific point in time and not beyond). These assumptions can be easily satisfied in OSNs.

### **5.6.2 Background Knowledge**

The main part of the attack model is the assumption of background knowledge the attacker has at a particular time. Different background knowledge can cause different attacks with different abilities and different consequences. Previous studies [Korolova, Motwani, Nabar, & Xu, 2008; Martin, Kifer, Machanavajjhala, Gehrke, & Halpern, 2007; Qian, Li, Zhang, & Chen, 2016] had specific assumptions about the attacker's prior knowledge (also referred to as background information, auxiliary information, and background knowledge). Hereafter, these terms of background knowledge interchangeably will be used as such in this study. A set of auxiliary information (personal knowledge, personal communication, public datasets, history sniffing, and social engineering) has been proposed and studied in the literature [Ganta, Kasiviswanathan, & Smith, 2008]. In [Ji, Li, Mittal, Hu, & Beyah, 2015; Narayanan & Shmatikov, 2009; Sharad, 2016], an auxiliary graph is assumed to be accessed by an adversary. Such a graph is used as side information to re-identify individuals in a sanitized graph.

This dissertation, adopts a modified version of the same assumption of the background knowledge (see Figure 26). The Real Identity graph,  $G^L_v(V^L_v, E^L_v)$  is used as an auxiliary graph and the Null Identity graph,  $G^H_v(V^H_v, E^H_v)$  is used as a sanitized graph. However, the Globally unique Pseudo Identity graph,  $G^G_v(V^G_v, E^G_v)$ , and the Locally unique Pseudo Identity graph,  $G^P_v(V^P_v, E^P_v)$ , could be used as a sanitized graph.



**Figure 26: Bob's Background Knowledge (Local and Null Identity Graphs)**

**Definition 1 (Real Identity Graph)**  $G^L_v(V^L_v, E^L_v)$  is a subset of the OSN graph  $G^L_v \subseteq G$ , where each node represents an individual and is associated with a real identity, and each edge represents a social tie. Both the attacker  $v$  and the victim  $u$  nodes are part of the graph and the distance between them is  $d(v, u)$ . The graph  $G^L_v$  can be collected by the attacker.

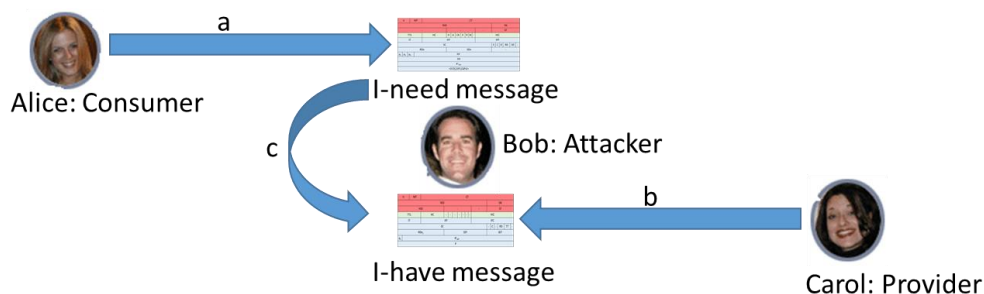
**Definition 2 (Null Identity Graph)**  $G^H_v(V^H_v, E^H_v)$  is a k-star graph, with the attacker having vertex degree k-1 and k-1 adjacent neighbors with degree 1. The k value could be any number less than n, based on the forwarding strategy that the attacker only knows his or her adjacent neighbors. The victim node is hidden  $u \notin V^H$ . The distance between the attacker and the victim node might be derived from the I-need message. It can be assumed that the  $G^H_v$  graph has been collected by the attacker using the information associated with the I-need message. It can also be assumed that the graph belongs to a single I-need message.

Definition 3 (**Disclosing attributes**) are features associated with the I-need message that reveal information about the path that the message comes through. The Has-Hidden (H) attributes tell the attacker if there is a hidden node in the path. The Has-Anonymous (A) attributes inform the attacker if there is an anonymized node in the path. The Pathway ID Sequence (CO) tells the actual length of the path, P. These attributes are assumed to also be a part of the Background Knowledge.

### 5.6.3 Goal of the Attack

The attacker in OSN using SOR protocol desires to learn the *relationship anonymity*: who is exchanging service with whom by linking a) a service consumer to an I-need message; b) a service provider to an I-have message; and c) an I-need message to an I-have message. However, in SOR, *service consumer anonymity* can be achieved by using a Null or a Locally unique Pseudo Identity, making it hard for the attacker to guess the sender's identity. Furthermore, *service provider anonymity* is achieved by using cryptographic techniques. The attacker is able in this version of the protocol to easily connect the I-need message to the I-have message.

Figure 27 illustrates how Bob as the attacker tries to identify a) the I-need message sender, (b) the I-have message sender, and (c) who communicates with whom.



**Figure 27: Goal of Bob as an attacker**

Theoretical attack analysis used in the study focuses on Null Identity and Locally unique Pseudo Identity, but not the Globally unique Pseudo Identity. This was chosen due to the highly improbable event that an attacker could link the service consumer to an I-need message. It could be possible but difficult if the Locally unique Pseudo Identity is used by victims. This belief will be tested in the next section and to prove how and at which level the privacy is preserved.

## 5.7 Theoretical Attack analysis using Proxima

This section analyzes the level of service consumer anonymity achieved by the Pseudo and Null identities. Attacks that attempt to identify the sender of the I-need messages are also considered. The Proxima Matrix and Proxima distributions are introduced to analyze the degree of anonymity of the service consumer.

### 5.7.1 Identity Anonymity of Service Consumer

**Service Consumer Anonymity:** a service consumer is anonymous if his or her identity is unknown with absolute certainty. It is measured by the size of the service consumer anonymity set; the more members are in the set of potential service consumers,

the less the probability that a randomly chosen member of the set is the actual owner of the I-need message.

**A Service Consumer Anonymity Set (S)** is the set of all possible service consumers who could have sent a particular I-need message as observed by the attacker. The larger the anonymity set size, the more anonymity a service consumer enjoys.

The size of the service consumer anonymity set in OSN is different based on the used identity and on how far the I-need message travels. For example, assume that an attacker  $v$  receives an I-need message  $m$  from its adjacent neighbor  $z$  (the last forwarder). The attacker  $v$  is trying to identify the sender  $u$  (service consumer) of  $m$ . If the Locally unique Pseudo Identity is used, then the victim  $u$  could be one of the nodes at distance  $x$  from  $z$ . On the other hand, if the Null Identity is used, the victim  $u$  could be one of the nodes at distance  $x$  from  $z$  and beyond. A Fixed Proxima matrix ( $U^F$ ), an Integrated Proxima matrix ( $U^I$ ), a Fixed Proxima distribution ( $R^F(x)$ ), and an Integrated Proxima distribution ( $R^I(x)$ ), are used to estimate the size of the service consumer anonymity set in OSN, as discussed in the next section. The size of the service consumer anonymity set approximately equals  $R^F(x)$  when the Locally unique Pseudo Identity is used, and approximately equals  $R^I(x)$  when the Null Identity is used.

**Degree of Anonymity of Service Consumer (D):** Generally, the degree of anonymity of a service consumer is quantified by:

$$D = 1 - \frac{1}{S} \quad (3)$$

Where  $1/S$  is the certainty of the adversary on the existence of a link between a service consumer and an I-need message, and  $S$  is the size of the service consumer anonymity

set. For example, assuming there is a set of size  $S = 3$ , the attacker can link an I-need message  $m$  to a service consumer  $v$  only to a certainty of  $1/3 \approx 33\%$ . On the other hand, the degree of anonymity of the victim  $v$  is  $1 - 1/3 \approx 66\%$ . Two variations of the service consumers' degrees of anonymity are defined in OSN: *Fixed Proxima Degree of Anonymity* and *Integrated Proxima Degree of Anonymity*.

**Definition 4 (Fixed Proxima Degree of Anonymity):**  $D^F(x)$ , the degree of anonymity for the service consumer at level (distance)  $x$ , when the Locally unique Pseudo Identity used is quantified as:

$$D^F(x) = 1 - \frac{1}{R^F(x)} \quad (4)$$

where  $x$  is the distance from the last forwarder and  $R^F(x)$  is the estimated service consumer anonymity's set size.

**Definition 5 (Integrated Proxima Degree of Anonymity):**  $D^I(x)$ , the degree of anonymity for the service consumer when the Null Identity is used is quantified as:

$$D^I(x) = 1 - \frac{1}{R^I(x)} \quad (5)$$

where  $x$  is the distance from the last forwarder and  $R^I(x)$  is the estimated service consumer anonymity set size when the Null Identity is used.

The question now is how to estimate the size of the service consumer anonymity set,  $R^F(x)$ , when the Locally unique Pseudo Identity is used; and the size of the service consumer anonymity set,  $R^I(x)$ , when the Null Identity is used. The answer is discussed in the next sections.

### 5.7.2 Proxima Matrices

**Fixed Proxima matrix:** This attempts to figure out how many nodes exist at a distance  $x = \{0, 1, 2, \dots, d\}$  from the given node  $v$  (where  $d$  is the diameter of  $G$ ), given a graph  $G$  and a node  $v \in V$ . When the number of nodes at distance  $x$  of node  $v$  are counted in OSN, a matrix  $U^F \in \mathbb{R}^{dn}$  can be used to represent these numbers. Each cell  $u_{x,v} = |L_{x,v}|$  represents the number of nodes at distance  $x$  from node  $v$ . This matrix shows the exact number of nodes at a given distance from any node in the network.

**Definition 6 (Network Diameter):**  $d$  is the shortest distance between the two most distant nodes in a graph. It can be assumed that this is either known or can be estimated by the attacker by using the following formula [Barabási, 2016]:

$$d \propto \frac{\text{Log}(n)}{\text{Log}(k)} \quad (6)$$

where  $n$  is the total number of nodes in the network and  $k$  is the average degree of the network.

**Definition 7 (Level)** is the set of nodes at distance  $x$  from node  $v$  in  $G$  and denoted by  $L_{x,v}$ :

$$L_{x,v} = \{u \in V : \text{dis}(v, u) = x\} \quad (7)$$

where  $\text{dis}(v, u)$  is the number of edges between  $v$  and  $u$ . For example,  $L_{0,v}$  refers to the set of directly connected neighbors of node  $v$  and are equivalent to node degree  $d_v$ ;  $L_{1,v}$  refers to the set of the friend of friends of node  $v$  and so on.

The algorithm in Figure 28, computes the number of nodes at distance  $x$  from each node in  $G$ . Step 3 gets the number of nodes at distance  $x$  from node  $v$ , step 4 adds the value to a vector  $w$ , and step 6 adds the vector  $w$  to the Proxima matrix  $U^F$ .

---

**Algorithm 1** Computation of  $U^F$

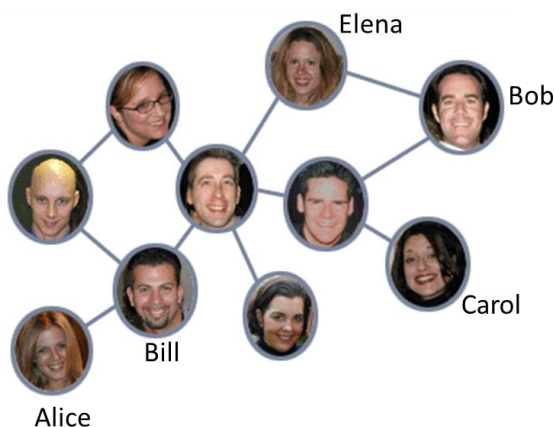
---

**Input:** Social Graph  $G(V, E)$   
**Output:** Proxima Matrix  $U^F$

- 1 **For each** node  $v \in V$  **do**
- 2   **For each** distance  $x = 0 \rightarrow d$  **do**
- 3      $L_{v,x} \leftarrow \text{Get\_number\_of\_nodes\_at\_diatnce}(x);$
- 4      $w \leftarrow \text{Add\_to\_vector}(L_{v,x});$
- 5   **End for**
- 6    $U^F \leftarrow w$
- 7 **End for**
- 8: **Return**  $U^F$

---

**Figure 28: Computation of Proxima Matrix**

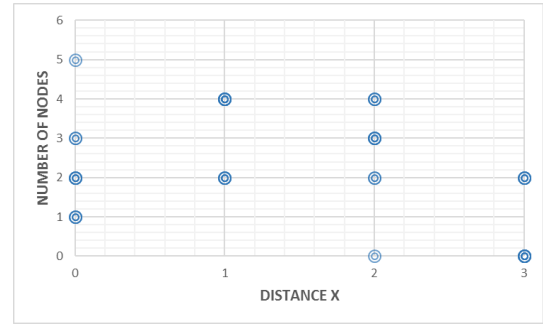


**Figure 29: Small Social Network (SSN)**



$$U^F = \begin{bmatrix} 2 & 2 & 3 & 1 & 5 & 1 & 3 & 2 & 2 & 1 \\ 2 & 4 & 4 & 2 & 4 & 4 & 4 & 4 & 2 & 2 \\ 3 & 3 & 2 & 4 & 0 & 4 & 2 & 3 & 3 & 4 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$

A



B

**Figure 30: (A) Fixed Proxima Matrix of SSN. (B) Scatter Plot of the Fixed Proxima Matrix of SSN.**

The small social network (SSN) in Figure 29 illustrates the Proxima concepts. An SSN consists of ten nodes, eleven edges, an average node degree 2.2, and a network diameter 4. The fixed Proxima Matrix,  $U^F$  of SSN is presented in Figure 30 (A). The scatter plot of matrix  $U^F$  shown in Figure 30 (B): this is where the x axis is the distance  $x$  and the y axis refers to the number of nodes at that distance. The value  $u_{0,1}=|L_{0,1}|=2$  represents the number of nodes at distance 0 from node 1 (Elena). The column  $U^F_{(:,1)}=[2,4,3,0]$  depicts that node 1 (Elena) has two friends at level 0, four friend-of-friends at level 1, three friends at level 2, and no neighbors at level 3. The row  $U^F_{(0,:)}=[2,2,3,1,5,1,3,2,2,1]$  shows the number of friends (node-degree) of each node in  $G$ .

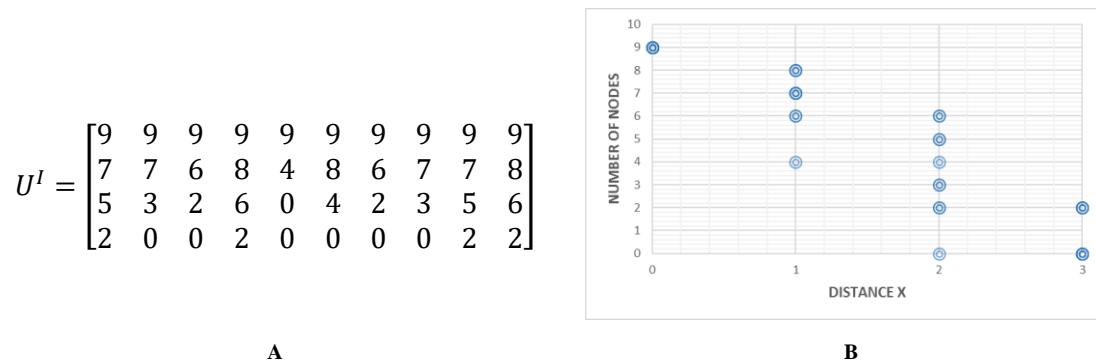
The service consumer anonymity set size can be determined using  $U^F$ . Assume that Bob as an attacker receives an I-need message from his adjacent neighbor Elena (last forwarder) with a Pathway ID Sequence (CO) telling the actual length of the path  $P$ . It can also be assumed that the path length is two and the Locally unique Pseudo Identity is used. Bob knows that the message comes from one of the consumers at level  $x=P-1=1$  of Elena's circle (four potential consumers).

**Integrated Proxima matrix:** Given the fixed Proxima matrix,  $U^F$ , the Integrated Proxima matrix is the summation of its rows. The Integrated Proxima matrix  $U^I \in \mathbb{R}^{dn}$ , where  $n$  is the number of nodes and  $d$  is the diameter of  $G$ , is used to represent the size of the service consumer anonymity set when the Null identity is used. A row  $i$  in  $U^I$  equals the sum of some rows of the fixed matrix  $U^F$ .

$$U^I_{(i),(:)} = \sum_{k=i}^d U^F_{(k),(:)} \quad (8)$$

For example, assume Bob as an attacker receives an I-need message from his adjacent neighbor, Elena (last forwarder), with a path length of two and a hidden service consumer. He knows that the message comes from one of the consumers at level  $x=P-1=1$  of Elena's circle (seven potential consumers).

Figure 31 (A) shows the integrated Proxima Matrix,  $U^I$ , of  $G$  and (B) presents the scatter plot of  $U^I$ .



**Figure 31: (A) Integrated Proxima Matrix of SSN. (B) Scatter plot of the Integrated Proxima Matrix of SSN**

### 5.7.3 Proxima Distributions

It is necessary to find a mathematical expression (model) for the Fixed and the Integrated Proxima matrices that describes (in some sense) the behavior of random variables (the number of nodes at each distance or level). The study uses the polynomial regression model [Ostertagova, 2012] to describe the relationship between the distance variable ( $x$ ) and the number of nodes variable ( $y$ ). Two distributions are proposed here: a Fixed Proxima distribution,  $R^F(x)$ , and an Integrated Proxima distribution,  $R^I(x)$ .

It is noticeable for all real used graphs in Fixed Proxima Matrix that the variable  $y$  (the number of nodes) increases when the variable  $x$  (the distance from a particular node) increases, but only up to a certain point, afterward, as the variable  $x$  continues to increase, the variable  $y$  decreases. This trend is clear in the scatter plots of Figure 30 (A).

**The Polynomial Regression Model for Estimating Proxima Distributions:** The basic polynomial regression model of a dependent (response) variable  $Y$  or  $P(x)$  on an independent (predictor) variable  $x$  can be expressed as:

$$P(x) = a_0 + a_1 * x + a_2 * x^2 + \dots + a_n * x^n \quad (9)$$

Note that the preceding polynomial can be represented as follows:

$$P(x) = \sum_{i=0}^n a_i x^i \quad (10)$$

where  $a_0, a_1, \dots, a_n$  are constants known as the model regression coefficients or parameters.

**Assumptions for Polynomial Regression Model [Poole & O'Farrell, 1971]:** 1)

The behavior of  $Y$  can be explained by a curvilinear additive relationship. 2) The

relationship between the dependent variable  $Y$  and any independent variable  $x$  is linear or curvilinear (specifically polynomial). 3) The independent variables are independent of each other, and the errors are independent and normally distributed. Thus, the aforementioned assumptions have to be tested to check if they are reasonable assumptions to work with.

**The Ordinary Least Squares Method** is a method used for estimating the unknown parameters in a linear regression model [Nelson, 1991]. It involves minimizing the sum of the squared errors with respect to the model parameters. Thus, the solution minimizes the sum of the squared errors of  $p(x)$  [Dreesen, Batselier, & De Moor, 2012]:

$$b_w = \sum_j |y_j - p(x_j)|^2 \quad (11)$$

$$b_w = (X^T X)^{-1} X^T Y \quad (12)$$

Using derivatives is not always possible when estimating the parameters of OLS; therefore, iterative methods (gradient descent and Gauss-Newton approximations) are very often used.

**Evaluate accuracy of regression models:** A common way to summarize how well a regression model fits the data is via the coefficient of determination or  $R^2$  [Maydeu-Olivares & Garcia-Forero, 2010]. This can be calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (13)$$

where  $y_i$  is an observed value,  $\bar{y}$  is the average of the  $y_i$ , and  $\hat{y}_i$  is the predicted value (on the line). If the predictions are close to the actual values,  $R^2$  should be close to

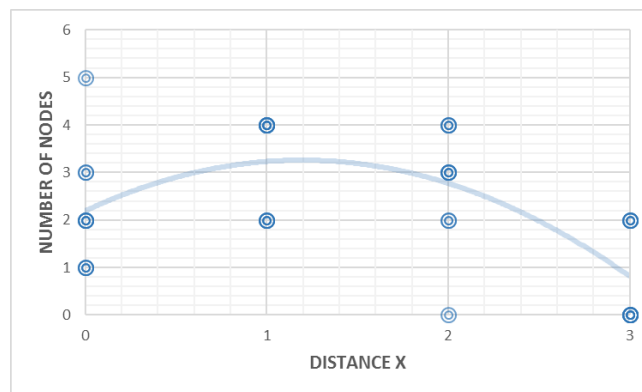
1. On the other hand, if the predictions are unrelated to the actual values, then  $R^2=0$ . In all cases,  $R^2$  lies between 0 and 1 and is interpreted as follows [Helland, 1987]:

- $R^2 = 1$  means that all data points lie on the regression line.
- $R^2 > 0.7$  means that the response variable is well-explained by the predictor variable.
- $0.3 < R^2 < 0.7$  means that the response variable is moderately well-explained by the predictor variable.
- $R^2 < 0.3$  means that the response variable is weakly explained by the predictor variable.
- $R^2 = 0$  means that the response variable has no explanatory effect.

**Fixed Proxima distribution:** The Fixed Proxima distribution of SSN in Figure 29 is:

$$R^F(x) = 2.19 + 1.79x - 0.75x^2 \quad (14)$$

with accuracy  $R^2= 0.43$  (as shown in Figure 32), where the  $x$  axis represents the distance and the  $y$  axis represents the number of nodes at that distance.

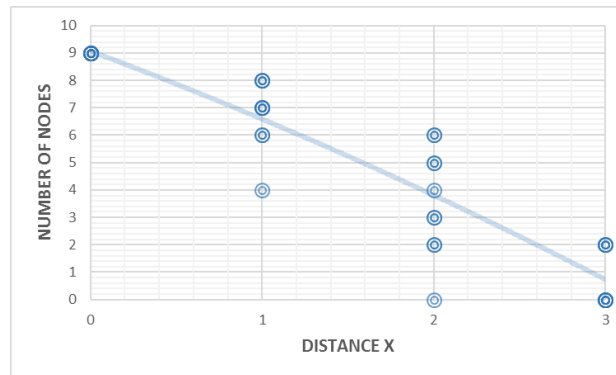


**Figure 32: Fixed Proxima distribution of SSN**

**Integrated Proxima distribution:** The Fixed Proxima distribution of SSN in Figure 29 is:

$$R^I(x) = 9.07 - 2.33x - 0.15x^2 \quad (15)$$

with accuracy  $R^2 = 0.86$  (as shown in Figure 33), where the  $x$  axis represents the distance and the  $y$  axis represents the number of nodes at that distance.



**Figure 33: Integrated Proxima distribution of SSN**

#### 5.7.4 Proxima Degree of Anonymity

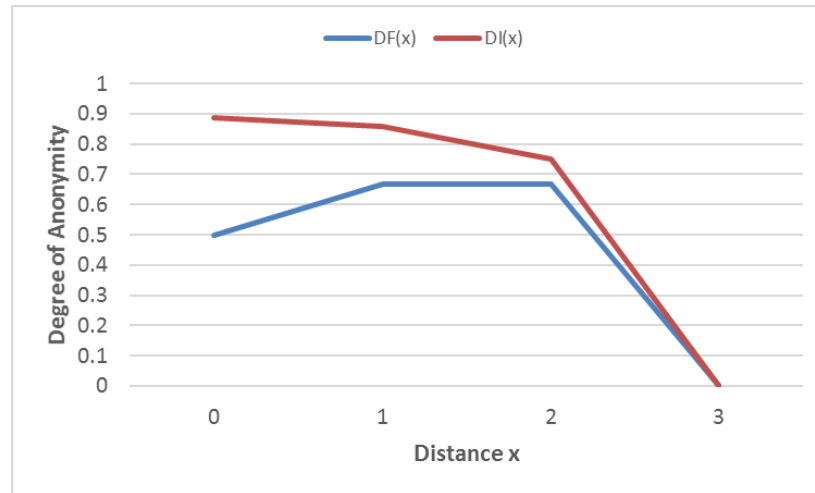
This subsection discusses how the Fixed and Integrated Proxima degrees of anonymity of SSN are calculated.

**Fixed Proxima Degree of Anonymity of SSN:** The Fixed Proxima Degree of Anonymity is defined as:

$$D^F(x) = 1 - \frac{1}{2.19 + 1.79x - 0.75x^2} \quad (16)$$

**Integrated Proxima Degree of Anonymity of SSN:** The Integrated Proxima Degree of Anonymity is defined as:

$$D^I(x) = 1 - \frac{1}{9.07 - 2.33x - 0.15x^2} \quad (17)$$



**Figure 34: Degree of Anonymity vs Distance of Fixed and Integrated Proxima**

As shown in Figure 34, the Integrated Proxima overcomes the Fixed Proxima from distance  $x=0$  to distance  $x=2$ . However, at distance  $x=3$  both Proximas are equal, meaning that if the attacker can guess that the message comes from the last circle of the last forwarder, then no anonymity can be preserved.

**Finding the Global Maximum and Minimum of Proxima Degree of Anonymities Using Higher-Order Derivatives:** It is important for any node in OSN to know the minimum and maximum degree of anonymities, and at which level. That can determine which technique should be used for getting more privacy.

First- and second-order derivatives of the Proxima degree of anonymity functions ( $D^F(x)$  and  $D^I(x)$ ) are used to find their maxima and minima on an interval. The interval is  $[0, d]$  where  $x$  can be any positive integer value greater than or equal to 0 (directly connected friends) and less than or equal to  $d$  (the network diameter). The maximum or minimum over the entire function is called an Absolute or Global maximum or minimum. There is

only one global maximum (and one global minimum), but there can be more than one Relative or Local maximum or minimum. The term extremum (the plural of which is extrema) refers to the points where the function attains a local or global maximum or minimum. The following algorithm was utilized to identify the Absolute extrema for the Fixed Proxima Degree of Anonymity on the interval  $[0, d]$ . The same steps can be used to find the Absolute extrema for Integrated Proxima Degree of Anonymity.

1. Find the first-order derivative of function  $D^F(x)$  at  $x$ :

$$D^F(x)' = \left(\frac{-1}{R^F(x)}\right)' = \frac{(-1)' R^F(x) + (-1)R^F(x)'}{R^F(x)^2} \quad (18)$$

2. Find the critical points by solving the equation  $D^F(x)' = 0$ .
3. Check if the critical points for this function lie within the considered interval  $[0, d]$ .
4. Narrow down which critical points could be the global maxima or minima by applying the second-order derivative:

$$\begin{aligned} D^F(x)'' &= [D^F(x)']' = \left[ \frac{-R^F(x)'}{[R^F(x)]^2} \right] \\ &= \frac{-R^F(x)''[R^F(x)]^2 - 2[R^F(x)']^2 R^F(x)}{[R^F(x)]^4} \end{aligned} \quad (19)$$

5. Substitute the value of each of the critical points one by one in place of  $x$  of  $D^F(x)''$ . If the resulting value is less than 0, the point is a local maximum; if the value is greater than 0, it is a local minimum. If the resulting value is 0, then the test has failed.



6. Find the global maximum of  $D^F(x)$  on the interval  $[0, d]$  by computing the value of  $D^F(x)$  at the points local maximum, 0, and  $d$ . Among these points, the place where  $D^F(x)$  has the largest value must be the global maximum.
7. Find the global minimum of  $D^F(x)$  on the interval  $[0, d]$  by computing the value of  $D^F(x)$  at the points local minimum, 0, and  $d$ . Among these points, the place where  $D^F(x)$  has the smallest value must be the global minimum.

**Global Maximum and Minimum of Proxima Degree of Anonymities of SSN:** Table 4 shows the global maximum and minimum of Fixed Proxima Degree of Anonymity,  $D^F(x)$ , and Integrated Proxima Degree of Anonymity,  $D^I(x)$ , of SSN. The minimum proxima degree of anonymity is at level three and the maximum proxima degree of anonymity is at level one of  $D^F(x)$  and zero of  $D^I(x)$ .

**Table 4: Global Maximum and Minimum of Degree of Anonymities**

	Maxima	Bounded by Distance	Minima	Bounded by Distance
$D^F(x)$	0.69	1.19=1	0	3
$D^I(x)$	0.88	0	0	3

### 5.7.5 Privacy Analysis and Evaluation

This subsection evaluates the proposed defensive mechanisms (Locally unique Pseudo Identity, and Null Identity) by using real datasets from Offline and Online Social Networks.

**Offline Social Networks:** four offline small social networks (used in the social sciences) and two synthetic networks (path and complete graphs) were used to evaluate

the proposed mechanisms on small real social networks. The statistical information of the datasets is summarized in Table 5.

- Davis Southern women social network (DSW): This data was collected by Davis and his colleagues in their study of southern women [Davis, Gardner, Gardner, & Warner, 1941]. The social network consisted of eighteen women and fourteen places (informal social events) observed over a nine-month period. The places are assumed to be common among the friends.
- Zachary's Karate club graph (KC): This social network of a karate club was collected by Wayne W. Zachary for a period of three years from 1970 to 1972 [Zachary, 1977]. The network captures 34 members and 78 pairwise links between them.
- Florentine families graph (FF): This network includes a set of sixteen Italian families. The edges denote a connection by marriage [Wasserman & Faust, 1994].
- Krackhardt kite graph (KK): This simple graph consists of ten nodes and eighteen edges [Wasserman & Faust, 1994].
- Complete graph (COM): This network includes a fully connected network.
- Path graph (PA): This network consists of  $n$  nodes linearly connected by  $n-1$  edges, where  $n=10$ .

**Table 5: Statistical information of the offline social networks**

<b>Network Name</b>	<b>#Nodes</b>	<b>#Edges</b>	<b>Avg. Node Degree</b>	<b>Graph Diameter</b>
Davis Southern women social network (DSW)	32	89	5.5	4(3)
Zachary's Karate club graph (KC)	34	78	4.5	5(4)
Florentine families graph (FF)	15	20	2.6	5(4)
Krackhardt kite graph (KK)	10	18	3.6	4(3)
Complete graph (COM)	25	300	24	1(0)
Path graph (PA)	10	9	1.8	9(8)

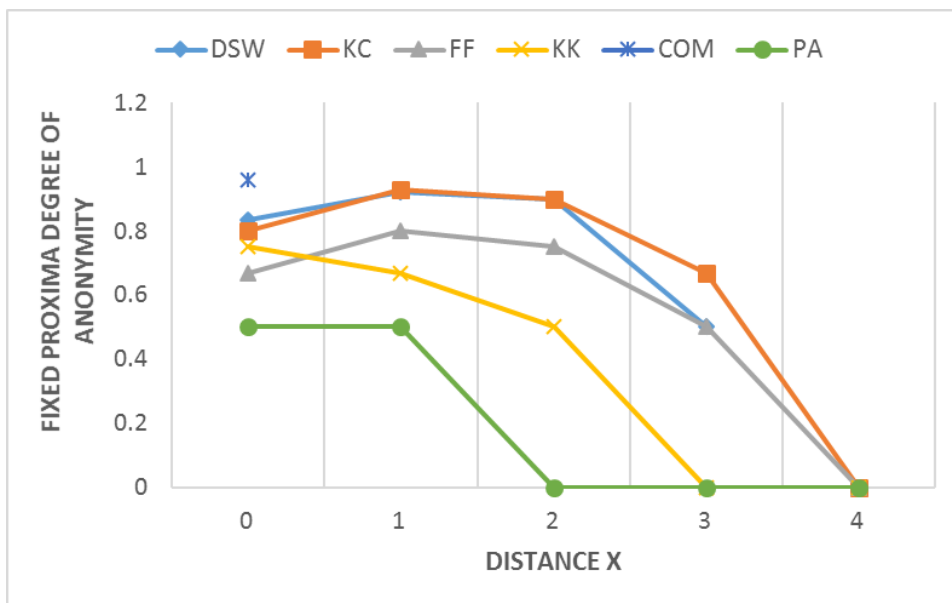
After creating the Fixed and Integrated Proxima matrices (as discussed in Subsection 5.7.2), the polynomial regression model was performed for estimating Proxima Distributions (as discussed in Subsection 5.7.3). The result of this was obtaining the mathematical expressions (models) of Fixed Proxima Distributions in Table 6 and of Integrated Proxima Distributions in Table 7. It was observed that  $R^2$  for both distributions lies between 0.37 and 0.9, meaning that the response variable is well explained by the predictor variable. Following estimating Proxima Distributions, the Proxima degree of anonymity  $D^F(x)$  and  $D^I(x)$  was computed (as plotted in Figure 35 and in Figure 36).

**Table 6: Fixed Proxima Distributions of Offline Social Networks**

<b>Dataset</b>	<b>Mathematical Expression (Model)</b>	<b>R<sup>2</sup></b>
DSW	$R^F(x) = 5.56 + 13.77x - 7.31x^2 + 0.79x^3$	0.65
KC	$R^F(x) = 4.96 + 18.68x - 11.08x^2 + 1.54x^3$	0.57
FF	$R^F(x) = 2.64 + 4.07x - 2.18x^2 + 0.26x^3$	0.61
KK	$R^F(x) = 3.71 - 1.09x + 0.05x^2$	0.37
COM	$R^F(x) = 24.0 - 0.0x + 0.0x^2$	-
PA	$R^F(x) = 1.80 - 0.20x - 0.0x^2$	0.6

**Table 7: Integrated Proxima Distributions of Offline Social Networks**

<b>Dataset</b>	<b>Mathematical Expression (Model)</b>	<b>R<sup>2</sup></b>
DSW	$R^I(x) = 31.00 + 1.35x - 8.56x^2 + 1.65x^3$	0.9
KC	$R^I(x) = 33.32 - 0.61x - 6.80x^2 + 1.23x^3$	0.9
FF	$R^I(x) = 14.01 - 1.05x - 2.01x^2 + 0.36x^3$	0.9
KK	$R^I(x) = 9.05 - 4.35x + 0.55x^2$	0.8
COM	$R^F(x) = 24.0 - 0.0x + 0.0x^2$	-
PA	$R^F(x) = 9.00 - 1.90x + 0.10x^2$	0.9

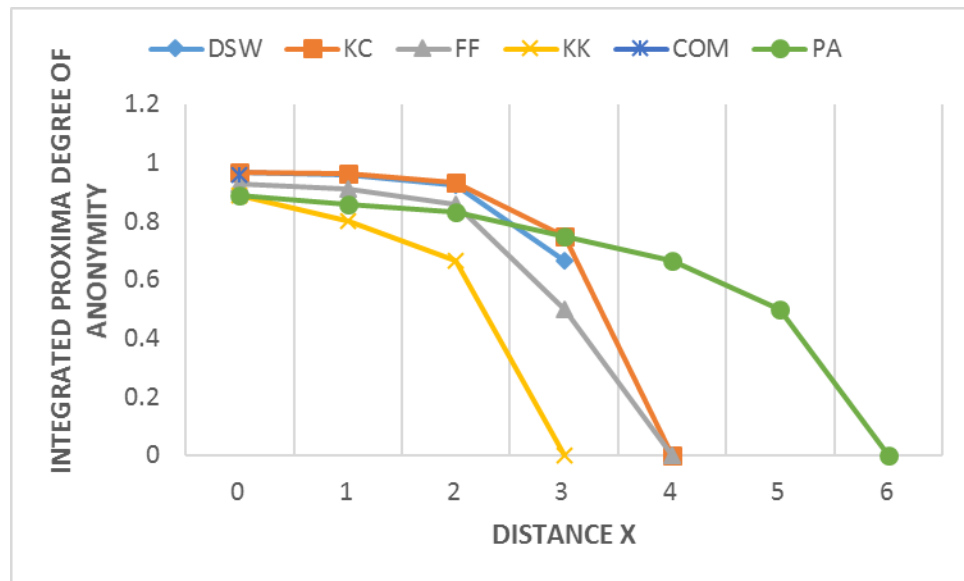


**Figure 35: Fixed Proxima Degree of Anonymity of Offline Social Networks**

Figure 35 compares the Fixed Proxima degree of anonymity ( $D^F(x)$ ) of six Offline Social Networks from  $x=0$  to  $x=d$  (the network diameter). Overall, the  $D^F(x)$  of the fully connected network (COM) is higher than those of other networks and is presented as a point at distance  $x=0$ . Furthermore, the  $D^F(x)$  of DSW, KC, and FF follow a fairly similar peaking pattern at  $x=1$  (Friend-of-Friends). That is because the anonymity set size is large at this distance.

The  $D^F(x)$  of Path graph (PA) is the lowest among the networks. The  $D^F(x)$  of Krackhardt kite graph (KK) is high at distance  $x=0$ , then continuously declines until  $x=d$ . The  $D^F(x)$  increases when the variable  $x$  (the distance from last forwarder) increases, but only up to a certain point (in most networks  $x=1$ ), after which, as the variable  $x$  continues to increase, the  $D^F(x)$  decreases. In addition, the  $D^F(x)$  of networks at distance  $x=d$  is zero,

which means that service consumers at such a distance should be careful when a Pseudo Identity is used.



**Figure 36: Integrated Proxima Degree of Anonymity of Offline Social Networks**

Experiments reveal that the Integrated Proxima degree of anonymity ( $D^I(x)$ ) is high for all Offline Social Networks at distance  $x=0$  (where anonymity set size equals all nodes in the network) as shown in Figure 36. In contrast, the  $D^I(x)$  of all networks begins to decline at  $x=2$ , and drops to zero at  $x=d$ . This means service consumers at this distance should use caution when Null Identity is used.

After finding the Proxima degree of anonymities, the global maximum and minimum of Proxima degree of anonymities (as discussed in Subsection 5.7.4) of Offline Social Networks were also found (as shown in Table 8). If the global maximum or minimum appears at the real value of distance, then one can round the value to the nearest distance  $x$  (as shown in the “Bounded by Distance” columns).

The global maximum degree of anonymity is at the friend-of-friends level when  $x = 1$  for networks DSW, KC, and FF and at the friends level when  $x=0$  for other networks. The global minimum degree of anonymity for all networks is, at its last level,  $x=d$ . This means that nodes at that level can be easily identified by the attacker. However, the global minimum of DSW is not 0 at  $x=d$ , meaning that attackers cannot easily identify nodes at that level. The  $D^F(x)$  of COM network is a single point and that point is the global maximum and minimum of fixed degree of anonymity.

**Table 8: Global Maximum and Minimum of Fixed Proxima Degree of Anonymity**

<b>Dataset</b>	<b>Global Maximum</b>	<b>Bounded by Distance</b>	<b>Global Minimum</b>	<b>Bounded by Distance</b>
DSW	0.92	1.15=1	0.58	3
KC	0.92	1.09=1	0	3.7=4
FF	0.79	1.18=1	0	4
KK	0.73	0	0	3
COM	0.95	0	0.95	0
PA	0.44	0	0	8

Table 9 gives data about the global maximum and minimum of  $D^I(x)$  of six networks: DSW, KC, FF, KK, COM, and PA. As the global maximum of  $D^I(x)$  is when  $x=0$  (the anonymity set size contains all nodes in the network), the global minimum of  $D^I(x)$  is at distance  $x=d$ . Compared with  $D^F(x)$ , the same pattern of networks DSW and COM appears here when the minimum of  $D^I(x)$  is not zero at  $x=d$ .

**Table 9: Global Maximum and Minimum of Integrated Proxima Degree of Anonymity**

<b>Dataset</b>	<b>Maxima</b>	<b>Bounded by Distance</b>	<b>Minima</b>	<b>Bounded by Distance</b>
DSW	0.96	0.08=0	0.60	3
KC	0.96	0	0	3.7=4
FF	0.92	0	0	3.9=4
KK	0.88	0	0	3
COM	0.95	0	0.95	0
PA	0.88	0	0	8

**Online Social Networks:** For analysis, four datasets downloaded from Jure Leskovec's Website [2014] were used. The data sets from 2012 are crawled from a popular social network site, plus.google.com. Each dataset contains a set of nodes (subscribers) and their social connections; they also contain information like user profiles. There is no overlap (same nodes in datasets) among the used datasets. Table 10 summarizes the statistical information of the four data sets, where DS refers to the Dataset. The user-profile information of Google+ datasets include six categories (gender, last name, job titles, institutions, universities, and places lived). See reference [J. J. McAuley & Leskovec, 2012] for more information about the used datasets.



**Table 10: Statistical information of the Online Social Networks**

Dataset	# nodes	# edges	Avg. Node Degree	Graph Diameter
DS-1	54	203	7.5	5 (4)
DS-2	116	1024	17.6	5 (4)
DS-3	1079	46234	85.6	6 (5)
DS-4	342	3361	19.6	6 (5)

After creating the Fixed and Integrated Proxima matrices (as discussed in Subsection 5.7.2) of the OSNs, the polynomial regression model for estimating Proxima Distributions was performed on the data (as discussed in Subsection 5.7.3). Table 11 lists the mathematical expressions (models) of Fixed Proxima Distributions and Table 12 lists the mathematical expressions (models) of Integrated Proxima Distributions.  $R^2$  for both distributions lie between 0.45 and 0.9, which means that the response variable is well explained by the predictor variable. Following estimating Proxima Distributions, the Proxima degree of anonymity  $D^F(x)$  and  $D^I(x)$  is computed as plotted in Figure 37 and in Figure 38.

**Table 11: Fixed Proxima Distributions of Online Social Networks**

Dataset	Mathematical Expression (Model)	$R^2$
DS-1	$R^F(x) = 7.5 + 18.8x - 9.2x^2 + 1.03x^3$	0.45
DS-2	$R^F(x) = 18.3 + 85.2x - 53.8x^2 + 7.8x^3$	0.68

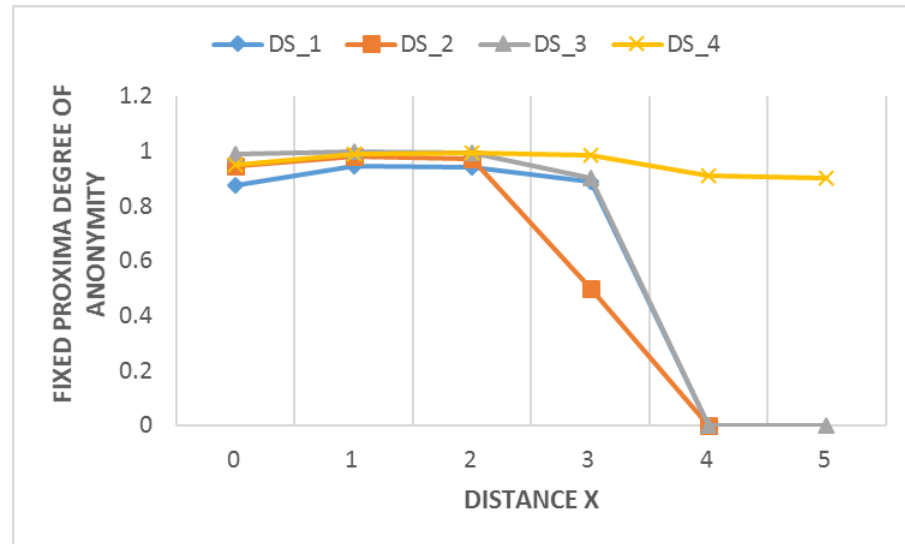
DS-3	$R^F(x) = 85.6 + 2812.3x$ $- 3239.1x^2 + 1362.09x^3$ $- 249.8x^4 + 16.8x^5$	0.86
DS-4	$R^F(x) = 19.6 - 41.3x + 231.8x^2 - 142.7x^3$ $+ 30.6x^4 - 2.22x^5$	0.70

**Table 12: Integrated Proxima Distributions of Online Social Networks**

Dataset	Mathematical Expression (Model)	$R^2$
DS-1	$R^I(x) = 53.12 - 0.58x - 9.05x^2 + 1.49x^3$	0.8
DS-2	$R^I(x) = 116.03 + 2.58x - 31.26x^2 + 5.86x^3$	0.9
DS-3	$R^I(x) = 1078.0 + 1514.74x$ $- 2594.79x^2 + 1211.68x^3 - 233.56x^4$ $+ 16.23x^5$	0.9
DS-4	$R^I(x) = 341.00 - 33.47x + 70.57x^2 - 75.53x^3$ $+ 20.47x^4 - 1.69x^5$	0.9

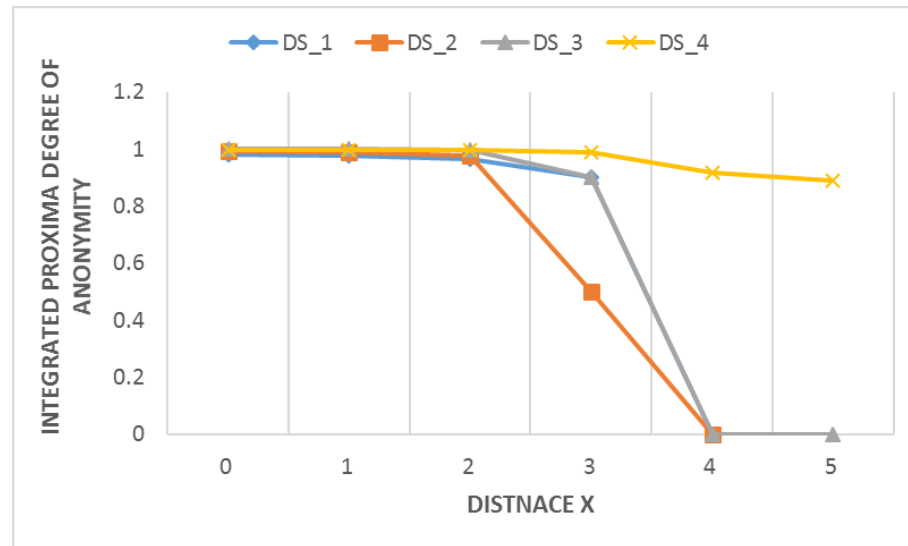
Figure 37 compares the Fixed Proxima degree of anonymity ( $D^F(x)$ ) of four Online Social Networks from  $x=0$  to  $x=d$  (the network diameter). Overall, the  $D^F(x)$  of all networks is high when  $x \leq 2$ , and declining after that point. Furthermore, the  $D^F(x)$  of most networks follows a fairly similar pattern at  $x=d$ , but the DS-4 steadily declines, with its

value being high at  $x=d$ . Nodes in network DS-4 reveal a higher  $D^F(x)$  than other nodes in other networks.



**Figure 37: Fixed Proxima Degree of Anonymity of Online Social Networks**

The study's experiments revealed that the Integrated Proxima degree of anonymity ( $D^I(x)$ ) is high for all Online Social Networks at the distance  $x=0$  (where the anonymity set size contains all nodes in the network), as shown in Figure 38. In contrast, the  $D^I(x)$  of all networks begins to decline at  $x=2$ , dropping to zero at  $x=d$  except DS-4. This means service consumers at such distance should be cautious when a Null Identity is used (except for those in network DS-4).



**Figure 38: Integrated Proxima Degree of Anonymity of Online Social Networks**

After finding the Proxima degree of anonymities, the global maximum and minimum of Proxima degree of anonymities (as discussed in Subsection 5.7.4) of Online Social Networks was also found (as shown in Table 13). The table shows the global maximum and minimum of  $D^F(x)$  and at which distance. If the global maximum or minimum appears at the real value of distance, then one can round the value to the nearest distance  $x$  (as shown in the “Bounded by Distance” columns).

The global maximum degree of anonymity is at the friend-of-friends level when  $x = 1$  for networks DS-1, DS-2, and DS-3, and at the distance  $x=2$  for DS-4. The global minimum for all networks is at its last level where  $x=d$ . This means that nodes at that level can be easily identified by the attacker. However, the global minimum of DS-1 and DS-4 is not 0, meaning that attackers cannot easily identify nodes at distance  $x=4$ .

**Table 13: Global Maximum and Minimum of Fixed Proxima Degree of Anonymity**

<b>Dataset</b>	<b>Maxima</b>	<b>Bounded by Distance</b>	<b>Minima</b>	<b>Bounded by Distance</b>
DS-1	0.94	1.3=1	0.30	4
DS-2	0.98	1.01=1	0.08	4
DS-3	0.99	1	0	5
DS-4	0.99	2	0.90	5

Table 14 provides data about the global maximum and minimum of  $D^I(x)$  of four networks: DS-1, DS-2, DS-3 and DS-4. The global maximum degree of anonymity of  $D^I(x)$  is at  $x=0$  (the anonymity set size contains all nodes in the network), and the global minimum degree of anonymity of  $D^I(x)$  is at distance  $x=d$ . Compared with  $D^F(x)$ , the same pattern of networks DS-1 and DS-4 appears when the minimum of  $D^I(x)$  is not 0 at  $x=d$ .

**Table 14: Global Maximum and Minimum of Integrated Proxima Degree of Anonymity**

<b>Dataset</b>	<b>Maxima</b>	<b>Bounded by Distance</b>	<b>Minima</b>	<b>Bounded by Distance</b>
DS_1	0.98	0	0.26	4
DS-2	0.99	0.04=0	0.18	4
DS-3	0.99	0	0	5
DS-4	0.99	0	0.88	5

In summary, the global maximum and minimum of the Fixed Proxima degree of anonymity for all networks (Online and Offline) are at distances  $x=1$  and  $x=d$  respectively. The global maximum and minimum of the Integrated Proxima degree of anonymity are at distances  $x=0$  and  $x=d$  respectively. The figures of Offline and Online social networks clearly indicate that the Fixed Proxima degree of anonymity declines sharply after  $x=2$  for most networks. This means that the sender of messages at distance  $x>2$  might be easily identified by the attacker. The reason behind this decline is that when the distance between the attacker and its victim increases, the number of potential senders increases, but only up to  $x=2$ , after which, as the distance continues to increase, the number of potential senders decreases.

## 5.8 Validity of Privacy Requirements

This subsection offers proof for some of the properties provided by the proposed protocol.

Each node can choose to be hidden or anonymized. The SOR protocol provides a Null Identity and a Locally Unique Pseudo Identity for nodes to be hidden or anonymized, respectively.

**Definition 8 (Anonymized):** when node  $v$  creates or forwards an I-need message  $m$ , it requests its adjacent neighbor  $u$  to be anonymized by using the fields Peer Privacy Request (R) and Peer Anonymize Name (N) of the I-need message. Assuming that  $u$  will

not misbehave, instead it should use a Locally Unique Pseudo Identity in place of  $v$ 's real identity.

**Definition 9 (Hidden):** when node  $v$  creates or forwards an I-need message  $m$ , it requests its adjacent neighbor  $u$  to be hidden by using the fields Peer Privacy Request (R) and Peer Anonymize Name (N) of the I-need message. This is assuming that  $u$  will not misbehave, but rather use Null Identity instead of  $v$ 's real identity.

**Assumption 1:** Assuming there are two adjacent nodes  $v$  and  $u$  in a network, if node  $v$  sends an I-need message and requests its adjacent neighbor  $u$  to be hidden or ammonized, then  $u$  will not misbehave and will either hide, or ammonize  $v$  based on its request.

**Definition 10 (Information leakage of Connectivity information):** SOR uses five different messages: I-need Message (InM), I-have Message (IhM), I-thank Message (ItM), I-like/dislike message (IdM) and I-Ack Message (IaM). The I-need message can leak information about individuals in OSNs. The attributes associated with the I-need message that could reveal information about the path that the message comes through are: Has-Hidden (H), The Has-Anonymous (A), and The Pathway ID Sequence (CO). The Has-Hidden (H) field tells if there is a hidden node in the path. The Has-Anonymous (A) field shows if there is an anonymized node in the path. The Pathway ID Sequence (CO) field tells the identities of the sender and the forwarders. These fields can cause information leakage about Connectivity information. However, the SOR provides a couple of privacy options (Null Identity and a Locally Unique Pseudo Identity) for nodes to be protected against this kind of information leakage. The field Path (P) of the I-have

message can also leak some information about the provider. However, that field can be encrypted to protect the privacy. Using globally unique IDs of the I-need and I-have message can be a potential privacy leak and help the attackers to trace paths that the messages take. However, the ID switch ids technique can be used to mitigate this privacy issue. When node  $u$  receives an I-need message with NID, it first keeps this ID locally in its FoT table, then associating a new NID to the I-need message, and finally forwarding it to the next neighbor. In this case, attackers cannot trace messages and privacy is preserved.

**Property 1:** In SOR protocol, if node  $v$  sends an I-need message  $m$  to node  $u$  and it chooses to be hidden from all others in  $S_u = \{\text{neighbors of node } u\}$  except for  $u$ , then it will be hidden.

**Proof:** Let  $u$  be an adjacent neighbor of the source node  $v$  and let  $u$  have a set of neighbors,  $S_u$ , being aware that  $v \in S_u$ . Now let  $z \in S_u$  be an attacker. In the worst-case scenario: 1) there is an edge between  $v$  and the attacker  $z$  and 2)  $v$  knows the real identities of the nodes in  $S_u$ . Also assume that there is no external information node  $z$  can collect about node  $v$ . Let  $u$  sends the message  $m$  to the attacker  $z$  after hiding  $v$ . The certainty of node  $z$  that  $v$  sends  $m$  is  $\frac{1}{|S_u|}$ , since  $v$  has chosen to be hidden. According to the rule, the Has-Hidden (H) field of the I-need message will be set to one, meaning that some nodes have chosen to be hidden. In this case, the attacker will suspect all nodes at distance one, two, and more form to the last forwarder (node  $u$ ). That mean the size of  $S_u$  will dramatically increase. Then the attacker certainty approaches zero when  $S_u$  is very



large.  $\lim_{S_u \rightarrow \infty} \frac{1}{S_u} \approx 0$ . Using assumption 1, it is known that  $u$  will not disclose information about who sent  $m$ . ■

**Property 2:** In SOR protocol, if node  $v$  sends an I-need message  $m$  to node  $u$  and it chooses to be anonymized from all others in  $S_u = \{\text{neighbors of node } u\}$  except for  $u$ , then it will be anonymized.

**Proof:** Let  $u$  (an adjacent neighbor of the source node  $v$ ) have a set of  $S_u$  neighbors, being aware that  $v \in S_u$ . Now let  $z \in S_u$  be an attacker who has an edge to node  $v$  and who knows the real identities of the nodes in  $S_u$  (worst-case scenario), assuming that there is no external information node  $z$  can collect about node  $v$ . Let  $u$  send the message  $m$  to the attacker  $z$  after hiding  $v$ . The certainty of node  $z$  that  $v$  sends  $m$  is  $\frac{1}{|S_u|}$ , since  $v$  chose to be hidden. According to the rule, the Has-Anonymous (A) field of the I-need message will be set to one, meaning that some nodes have chosen to be anonymized. In this case, the attacker will suspect that all nodes at distance one from the last forwarder (node  $u$ ). That means the size of  $S_u$  is equal to the adjacent neighbors of node  $u$  and based on that the certainty of the attacker can be determined. According to [1,2], the minimum number of neighbors in a social network is 16 and 24. Then the certainty of the attacker is between  $\frac{1}{16} \approx 0.06$  and  $\frac{1}{24} \approx 0.04$ . By using assumption 1, it is known that  $u$  will not disclose information about who sent  $m$ . ■

**Property 3:** The anonymity that SOR provides cannot be broken even if the attacker collects information from a set of I-need messages.

**Proof:** Assume node  $v$  sends an I-need message  $m_1$ , requests service  $a_1$  and chooses to be hidden. Then it sends another I-need message  $m_2$ , requests service  $a_2$  and chooses to be anonymized. Finally, it sends an I-need message  $m_3$ , requests service  $a_3$  and chooses to use its real identity. Assume that these three I-need messages go through an adjacent neighbor  $u$  who has a set of  $S_u$  neighbors. Let  $z \in S_u$  be an attacker who has an edge to  $v$  and knows the real identities of the nodes in  $S_u$  (worst-case scenario), assuming there is no external information node  $z$  can collect about node  $v$ . Let  $u$  send the message  $m_1$  to the attacker  $z$  after hiding  $v$ , the message  $m_2$  to the attacker  $z$  after anonymizing  $v$ , and the message  $m_3$  to the attacker  $z$  with  $v$ 's real identity. The privacy intended in  $m_1$  can not be compromised by the information in  $m_2$  because there is no correlation between  $m_1$  and  $m_2$  even though both came through  $u$ . The privacy intended in  $m_1$  cannot be compromised by the information in  $m_3$ , even though  $m_3$  is associated with the real identity. There is no extra information  $z$  can collect from  $m_2$  and  $m_3$  which can help to increase or decrease the certainty of  $z$  that  $v$  sent  $m_1$ . Then, the anonymity that SOR provides cannot be broken by collecting information from a set of I-need messages. ■

**Property 4:** In SOR protocol, if the content of the I-need message encrypted, then the sender identity of the I-need message will be disclosed and cannot be hidden or anonymized.

**Proof:** Assume that node  $v$  sends an I-need message  $m$  requesting service  $a$  and encrypted  $m$  by using its private key. Then, any node  $u$  in the network needs to get the public key of the sender to be able to read  $m$ 's content. Knowing the public key of the

sender is as same as knowing its identity. Thus, the Null Identity and a Locally Unique Pseudo Identities cannot help to hide or anonymize  $v$ . ■

Each node has its own Forwarding Table (FoT) which maintains an entry for each I-need message and its I-have, I-thank, and I-ack messages. The FoT table is used by all nodes (consumers, forwarders, or providers) to trace the I-need message and its responses messages (I-have, I-thank, and I-ack). These tables are presumed to be encrypted.

**Property 5:** In SOR protocol, if all nodes decide to be hidden then reachability is guaranteed.

**Proof:** Assume that node  $v$  sends an I-need message  $m$  to its neighbor node  $u_0$  and chooses to be hidden (by assumption 1,  $u_0$  will not disclose information about the sender). It can also be assumed that all next forwarders have chosen to be hidden and no information about them is associated with  $m$ , considering that  $m$  goes through  $n$  step through a null path  $(v, \rightsquigarrow, u_n)$ . Presume that  $u_n$  sends an I-have message  $Q$  back toward node  $v$ . Here, all the forwarders  $(u_{n-1}, u_{n-2}, \dots, u_0)$  in the null path can use their FOT table to forward  $Q$  back to  $v$ . This will guarantee reachability. ■

## CHAPTER 6

### **Social Priority**

This chapter presents a quantity that determines the levels of importance of social ties in Online Social Networks (OSNs). This quantity is known as Social Priority (SP), and it is used between individuals in Offline Social Networks to rank tasks. Barabási introduced a model [Barabasi, 2005] that show that tasks are selected for execution according to rules which depend on the priorities of the tasks. He discussed how SP reflects the delay that tasks take to get a service in a human queue. In real life, two individuals do not use same set of social factors to assign the priority. However, despite the importance of this quantity, no theoretical or empirical research is currently available to estimate its value. This chapter explores the factors that drive individuals to rank their social ties. Specifically, a set of social characteristic-based metrics are introduced here, along with a set of social factors (e.g., Gender, Degree, Closeness, Betweenness, Eigenvector centralities) to estimate SP between two individuals. In the proposed framework, individuals are free to choose their primary feature set. To evaluate the stud's framework, it is tested on public real-world datasets (from the social media Google Plus). The study finds that social priorities in large communities tend to cluster toward the lower SP of the right end, while in small groups they tend to be normally distributed.

## 6.1 Related Work

The most relevant work to the study's framework is trying to 1) discover the importance of actors in Email Communication Networks by using various weighting schemes [Pawel Lubarski & Morzy, 2012; Paweł Lubarski & Morzy, 2014]; 2) incorporate weighted and directed influence edges in a social graph to improve search using an influence weight equation [Hangal et al., 2010]; 3) determine whether a work-item notification generated for a person needs his/her attention by way of using a machine-learning classifier [Mukherjee & Garg, 2013]; 4) characterize the effect of misalignment between priorities of both the task sender and task the receiver by using a cost function as an average priority-weighted sojourn time of a task in the queue [Sharma, Jagannathan, & Varshney, 2014]; and 5) find signs of links in the underlying social networks by using a logistic regression classifier [Leskovec, Huttenlocher, & Kleinberg, 2010a]. However, unlike previous frameworks, this study's framework a) utilizes the available social characteristics of individuals in OSN; b) combines multi-dimensional Social Priorities into a single-dimension Social Priority by using a singular value decomposition (SVD), and c) supports bidirectional Social Priority for both individuals at the end of the social tie. Finally, while most of proposed models are not extensible, the proposed framework can use a large number of metrics while the SVD narrows these metrics to one-dimension (dimensionality reduction).

## 6.2 Social Characteristics

In online social networks (OSNs) such as Facebook, Google+, and Twitter, individuals are associated with a set of social characteristics which could be personal,

such as basic descriptors (e.g. gender, relationship status), or based on personal interests (e.g. favorite music, places, players); structural Graphs (e.g. centralities, tie strength, social communities); or social interactions (e.g. likes, shares, posts, comments, pokes, Tweets and Retweets) [Gross & Acquisti, 2005; Hugo Liu & Maes, 2005; Shakimov, Lim, Cox, & Cáceres, 2008; Viswanath, Kiciman, & Saroiu, 2012]. These characteristics can be extracted either directly from individuals' profiles (such as basic information), or indirectly by using methods (i.e. machine-learning techniques) such as influence, trust/distrust, support/opposition, and friend/foe. These social characteristics are of great value for designing efficient solutions for the edges and weights of nodes in OSNs. The weight of a node can represent its importance in the network. Five social characteristics are used in this dissertation. It is beyond the scope of this work to discuss the details of social characteristics and to understand the importance of social actors. To learn more about these fields, the following references cover these in greater detail: [Borgatti, 2005] [Dequiedt & Zenou, 2014; Faust, 1997; Freeman, 1978; Friedkin, 1991; Friedl & Heidemann, 2010; Kang, Papadimitriou, Sun, & Tong, 2011; Valente, Coronges, Lakon, & Costenbader, 2008; Zafarani, Abbasi, & Liu, 2014] and [Brubaker & Cooper, 2000; Helmhout, 2006; Lamb & Kling, 2003; Nass, Steuer, & Tauber, 1994; Shoib, Nandhakumar, & Rowlands, 2009].

### **6.2.1 Centralities and Gender**

This subsection discusses the main known centrality measures in OSN, which reflect social aspects of how people connect, communicate and respect each other.

**Degree centrality** is the number of edges that a node has. Therefore, it reflects the importance of a node. Directed networks have incoming edges (in-degree) and outgoing edges (out-degree). In a social network context, in-degree is often interpreted as a form of popularity, and out-degree as gregariousness [Yan & Ding, 2009; Zafarani et al., 2014].

**Closeness centrality** measures the mean geodesic distance between a node and all of its reachable nodes. It identifies the node location. It also refers to how near a node is to all other nodes in the network. In a social network context, this describes how fast this node can reach everyone in the network. This measure plays an important role in how information is propagating throughout the network [Yan & Ding, 2009].

**Betweenness centrality** measures how important a node is by counting the number of shortest paths that pass through it. Therefore, it measures the load of a given node. In a social network context, it means how likely a given node is to being the most direct path between two individuals in the network, along with how it can influence the flow of information between them[Yan & Ding, 2009].

**Eigenvector centrality** measures the importance of a node in a network by seeing the importance of the other nodes connected to it. Google's Page Rank is a variant of the Eigenvector centrality measure. The assumption to calculate Eigenvector centrality is that each node's centrality is the sum of the centrality values of the nodes connected to them [Spizzirri, 2011].

**Gender** is a personal attribute that is not easily changed and is more about a personal sense of who a person is (e.g., man, woman). The focus of this study is only on male and female.

These social characteristics are utilized to generate Social Priority between individuals in OSNs. The Social Priority can be used to improve the performance of routing and request forwarding processes in OSNs [Othman & Khan, 2015]. This study uses only five social characteristics due to the availability and accessibility discussed in the next section.

### **6.2.2 Social Characteristics' Availability and Accessibility**

Subscribers (individuals) of OSNs can control the privacy settings and restrict visibility of sensitive social characteristics such as name, gender, birth date and so on. Furthermore, for privacy reasons, many subscribers do not give real and true information about themselves— sometimes they do not give any information except for mandatory information (such as an e-mail address). Thus, the efficiency and accuracy of any proposed model is determined by the availability of subscribers' information [Gross & Acquisti, 2005; Humbert, Studer, Grossglauser, & Hubaux, 2013; Mondal, Liu, Viswanath, Gummadi, & Mislove, 2014]. The more information that can be accessed, the higher efficiency and accuracy that we can be obtained. The focus of this study is on structural social characteristics (centrality measures) that can be easily computed from the available datasets. These measures reflect social meanings and can be used to determine the Social Priority between individuals. Based on Stanley Milgram's observation [Milgram, 1967] that “certain kinds of communication are strongly



conditioned by sex roles”, gender is included in the metrics. The gender was given in the used datasets, as will be discussed later in the social priority computation section.

The levels of access to social characteristics of OSNs are categorized into five levels from low to high: 1) *Abstract graph level*, which is easy to get and provides partial information about subscribers; 2) *Anonymous social level*, which is easy to get and provides encoded information about subscribers; 3) *User social level*, which is only available to the account owner; 4) *System social level*, which can be seen by OSNs’ administrators who own the data and can see the full picture; and 5) *Crystal ball social level*, which is ideal and gives not only subscribers’ social characteristics but also an accurate estimate of their behaviors and actions both now and in the future.

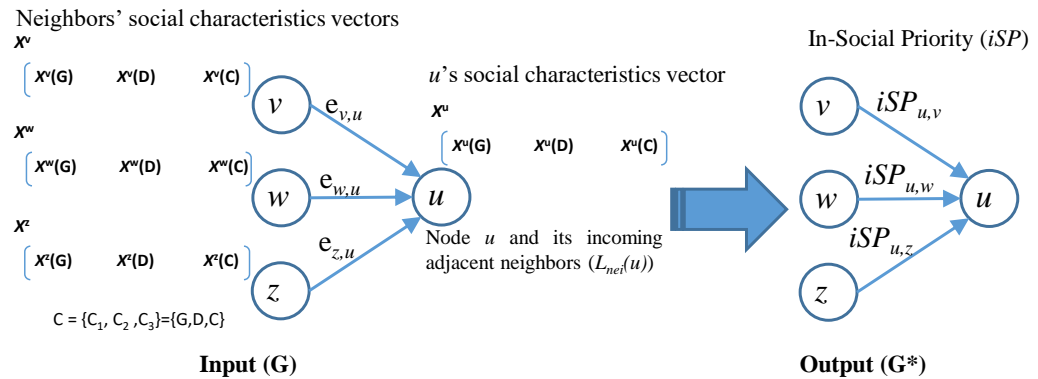
## 6.3 Preliminaries

### 6.3.1 The Used Notations

$C = \{C_1, C_2 \dots C_k\}$  denotes the set of social characteristics defined in the Online Social Network (OSN). Each  $C_j$ ,  $1 \leq j \leq k$ , takes its values from a finite/infinite domain  $D_j$ . For example, if  $C_j$  was gender, then the finite domain is  $D_j = \{\text{male, female}\}$  and if  $C_j$  was a centrality measure then the infinite domain  $D_j = [0,1]$ .  $G = (V, E)$  is a directed social graph where  $V$  is a set of  $n$  nodes and  $E$  is a set of  $m$  links between the nodes. The link  $e_{v,u}$  denotes a friendship (social tie) between its endpoints  $(v, u)$ , as shown in Figure 39.

A Social Priority Graph  $G^*$  is a directed and weighted social graph  $G^*(V, E, SP)$ , where  $V$  and  $E$  are the same as in the directed social graph  $G$ , and  $SP$  is the Social

Priority value as weights on edges (as demonstrated in Figure 40). Each node (individual)  $u \in V$  is associated with the items below: 1) a social characteristic vector  $x^u \in \mathbb{R}^k$ , where each element of the vector  $x^u(j) \in D_j$ , and the notation  $x^u(j)$  denotes the value of  $j^{\text{th}}$  social characteristic associated with node  $u$ ; 2) a social characteristics matrix  $A^u \in \mathbb{R}^{d \times k}$ , where  $d=|L_{\text{nei}}(u)|$  is the number of  $u$ 's adjacent neighbors and  $k$  is the number of social characteristics associated with nodes in the set  $L_{\text{nei}}(u)$ .  $A^u_{(i),(\cdot)} \in \mathbb{R}^{1 \times k}$  denotes the  $i$ -th row of matrix  $A^u$  which corresponds to a vector  $x^i$  of an adjacent neighbor  $i$   $A^u_{(\cdot),(j)} \in \mathbb{R}^{d \times 1}$ . This denotes its  $j$ -th column which corresponds to the values of social characteristic  $C_j$  for node  $u$ 's adjacent neighbors.  $A^u_{(i),j}$  refers to the  $i^{\text{th}}$  adjacent neighbor's social characteristic value of  $C_j$ .



**Figure 39: Social Characteristic Vectors and In-Social Priorities**

### 6.3.2 Social Priority

Social Priority (SP) is a quantifiable social property that characterizes the importance and levels of reciprocation between two individuals. This property can be used as an indicator of information flow to determine which path (a chain of intermediate

nodes) of individuals has the strongest influence on the destination to get a fast response. The maximum Social Priority value is zero and the minimum is one, so the Social Priority value between two nodes lies in  $[0, 1]$ . Generally, two types of Social Priorities have been investigated: 1) **In-Social Priority** which is the Social Priority between nodes  $(v, u)$  on edge  $e_{v,u}$  given by a node  $u$  for its in-coming adjacent neighbor  $v$ , where  $iSP_{u,v}$  denotes In-Social Priority; 2) **Out-Social Priority** is an estimated Social Priority  $oSP_{u,v}$  of In-Social Priority  $iSP_{u,v}$  which is given by node  $u$ . If node  $v$  makes an optimal estimation, then  $oSP_{u,v} = iSP_{u,v}$ . The SP notation will be used instead of Social Priority,  $iSP$  instead of In-Social Priority, and  $oSP$  instead of Out-Social Priority. SP is a generalized social priority that might be either  $iSP$  or  $oSP$ .

### 6.3.3 Problem Definition

Given a node  $u$ , as shown in Figure 39 with 1) its social characteristics vector  $x^u$  which contains only three social characteristics Gender ( $G$ ), Degree ( $D$ ), and Closeness ( $C$ ) as  $\{C_1, C_2, C_3\} = \{G, D, C\}$ ; and 2) a set of incoming adjacent neighbors,  $L_{nei}(u) = \{v, w, z\}$ , associated with their social characteristic vectors  $x^v, x^w, x^z$ , how can one find the In-Social Priority  $iSP = \{iSP_{u,v}, iSP_{u,w}, iSP_{u,z}\}$  that node  $u$  will give to its neighbors?

### 6.3.4 Complexity of the Problem

Estimating the exact Social Priority between two individuals in OSNs is not easy; it cannot even be easily estimated in Offline Social Networks. However, at least OSNs offer some interaction, communication, and collaboration datasets between individuals that implicitly reflect a lot of information about the relationship between them. Because

of privacy, each individual can estimate the Social Priority of his or her direct neighbors, and the estimated value can be disseminated in the OSNs based on the application. To simplify the problem, one can assume that all individuals in an OSN use the same social metrics, but with different social factors. Each social characteristic is represented as a metric and associated with a social factor. Social factors can be different from one individual to another, from a community to another, and from a nation to another. Research from psychology and sociology [Tajfel, 1969] can help to infer the value of the social factors. In this work, all social factors are binary.

## 6.4 General Social Priority Framework

This section introduces the core concepts of the proposed framework. It also shows how two adjacent individuals give Social Priority to each other. While five social characteristics are used to determine this (Gender, Degree, Closeness, Betweenness, and Eigenvector centralities), other social characteristics can be used based on the application and availability. For each social characteristic, a few local social metrics are proposed in the following section.

### 6.4.1 Social Metrics

**Gender-based social metric** ( $SP_G$ ): Node  $u$  gives node  $v$  an SP value based on gender (male or female) using the following rules which may be different from culture to culture as mentioned in [Dale, Osili, Mesch, & Ackerman, 2015; Roohani, 2015] and as Milgram observed in [Milgram, 1967]. 1) Male gives female high SP. 2) Male gives male low SP. 3) Female gives male low SP. 4) Female gives female high SP. The next

equation explains the above four rules where gender social factor  $\alpha_G = 1$ , which means high Social Priority. In the equation, gender (male, female) is coded as (0, 1).

$$SP_G(u, v) = \begin{cases} \alpha_G, & A_{(v,G)}^u = 1, x^u(G) = 0 \\ 1 - \alpha_G, & A_{(v,G)}^u = 0, x^u(G) = 0 \\ 1 - \alpha_G, & A_{(v,G)}^u = 0, x^u(G) = 1 \\ \alpha_G, & A_{(v,G)}^u = 1, x^u(G) = 1 \end{cases} \quad (20)$$

**Degree-based social metric (SP<sub>D</sub>):** Node  $u$  gives its incoming neighbor  $v$  SP value based on an in-degree metric. Thus, the incoming neighbor ( $v$ ) with a high in-degree centrality should be treated with higher Social Priority by the receiver node ( $u$ ), and vice versa. The next equation explains how the in-degree-based SP value is given where the degree social factor  $\alpha_D = 1$ .

$$SP_D(u, v) = \begin{cases} \alpha_D, & A_{(v,D)}^u \geq x^u(D) \\ 1 - \alpha_D, & \text{Otherwise} \end{cases} \quad (21)$$

**Closeness-based social metric (SP<sub>C</sub>):** Node  $u$  gives its incoming neighbor  $v$  a Social Priority value based on closeness centrality. The incoming neighbor ( $v$ ) with a lower closeness centrality should be treated with lower Social Priority by the receiver node ( $u$ ), and vice versa. The next equation shows how Social Priority value is given where the social factor  $\alpha_C = 1$ . For example, node  $u$  gives a high Social Priority for node  $v$  if  $v$ 's closeness centrality value is higher than  $u$ , and it gives a low Social Priority otherwise.

$$SP_C(u, v) = \begin{cases} \alpha_C, & A_{(v,C)}^u \geq x^u(C) \\ 1 - \alpha_C, & \text{Otherwise} \end{cases} \quad (22)$$

**Betweenness-based social metric (SP<sub>B</sub>):** Node  $u$  gives its incoming neighbor  $v$  a Social Priority value based on Betweenness centrality. The incoming neighbor ( $v$ ) with a

high Betweenness centrality should be treated with a higher Social Priority by the receiver node ( $u$ ), and vice versa. The next equation represents how the Social Priority value is given by node  $u$  to its incoming neighbor node  $v$ . In the equation, the social factor  $\alpha_B = 1$ , meaning a high Social Priority. Node  $u$  gives a low Social Priority for node  $v$  if  $v$ 's Betweenness centrality value is less than  $u$ , and it gives a high Social Priority otherwise.

$$SP_B(u, v) = \begin{cases} \alpha_B, & A_{(v,B)}^u \geq x^u(B) \\ 1 - \alpha_B, & \text{Otherwise} \end{cases} \quad (23)$$

**Eigenvector-based social metric (SP<sub>E</sub>):** Node  $u$  gives its incoming neighbor  $v$  a Social Priority value based on Eigenvector centrality. The incoming neighbor ( $v$ ) with a high eigenvector centrality should be treated with a higher priority by the receiver node ( $u$ ), and vice versa. The next equation presents how node  $u$  gives Social Priority to its directly connected incoming neighbor where the social factor  $\alpha_E = 1$ . Node  $u$  gives a low Social Priority for node  $v$  if  $v$ 's Eigenvector centrality value is less than  $u$ , and it gives it a high Social Priority otherwise.

$$SP_E(u, v) = \begin{cases} \alpha_E, & A_{(v,E)}^u \geq x^u(E) \\ 1 - \alpha_E, & \text{Otherwise} \end{cases} \quad (24)$$

This metric begins with the case where all social factors ( $\alpha_G$ : Gender,  $\alpha_D$ : Degree,  $\alpha_C$ : Closeness,  $\alpha_B$ : Betweenness,  $\alpha_E$ : Eigenvector) are binary; that is, where they are based on values of the kind “low/high” or “0/1”. However, these factors can be learned by 1) social science and psychology studies of communities and cultures as in [Tajfel, 1969]; 2) data mining, machine learning, and data analysis techniques; and 3) manually assigning binary or non-binary values in range [0, 1] to these social factors.

## 6.5 Social Priority Computation

As depicted in Figure 40, to generate the SP values, a node  $u$  needs to collect the available social characteristics of its neighbors and construct the *social characteristic matrix* of node  $u$  ( $A^u$ ) which temporally stores *social characteristic vectors* of node  $u$ 's incoming adjacent neighbor. The Social Priority metric is used to construct the Social Priority Matrix ( $SPA^u$ ). Each entry of each row vector denotes a given Social Priority based on a particular social metric. The singular value decomposition (SVD), which is a dimensionality reduction technique, is used for getting social priorities of incoming adjacent neighbors.

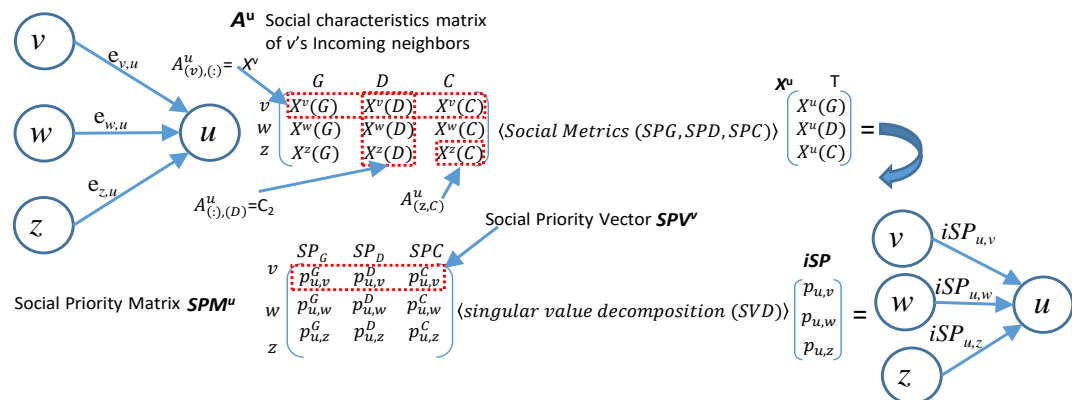


Figure 40: Steps of Social Priority Computation

### 6.5.1 Constructing Social Priority Matrix (SPA)

The first stage, involves collecting social characteristic values for each individual from a social Google+ dataset. Most individuals miss their personal social characteristics. Gender is the only one available for most individuals and because of this, records (individuals) with missing gender values are discarded. The social characteristic vectors of individuals are filled by these social characteristic values as described in Algorithm 3

step (4), based on the assumption that social characteristics vectors are known for adjacent neighbors. Each individual exchanges its social characteristics vector with its adjacent neighbors and, in turn, each individual stores these social characteristics vectors,  $x^u$ , and its social characteristics matrix  $A^u$ ,  $u \in V$ . Each node  $u$  in the social graph gives different social priorities to its adjacent incoming neighbors based on the defined social metrics, and stores them in its matrix  $SPA^u$  (see steps (5-10) in Algorithm 3). For instance, in step (6) node  $u$  gives its incoming adjacent neighbor  $v$  a gender-based Social Priority  $P_G$  using the Social Priority metric  $SP_G$ , along with a degree-based Social Priority  $P_D$  as in step (7), and so on. Step (11) shows how generated *Social Priority Vectors*,  $SPV^v$  for node  $v$  are stored as a row in the *Social Priority Matrix*  $SPA^u$  of the node  $u$ . Steps 5 to 12 are repeated for all of  $u$ 's incoming adjacent neighbors. A mathematical tool is needed to decompose the  $SPA^u$  matrix. SVD, which is a technique that produces the best (in the least-squares sense) reduced-rank approximation to the original data, is used [Klema & Laub, 1980].

### 6.5.2 Social Priority Matrix decomposition

This step aims to decompose the  $SPA^u$  matrix to acquire the social priorities of the incoming adjacent neighbors by using SVD. The goal of the SVD algorithm is to find a representation of the study's matrix  $SPA^u$  of social priorities as a product of lower-rank matrices. The  $SPA^u$  is a matrix of  $d$  adjacent neighbors (rows) with a  $k$  social characteristics vector (columns). As illustrated in step 13 in Algorithm 3, the singular value decomposition, SVD ( $SPA^u$ ), where  $u$  is the node index, is defined as:



$$SVD(SPM^u) = U \times S \times V^T \quad (25)$$

where  $U$  and  $V^T$  are  $d \times r$  and  $k \times r$  orthonormal matrices of singular vectors respectively and  $S$  is an  $r \times r$  diagonal matrix of singular values. The first singular value  $s_1$  is chosen from the diagonal entries  $(s_1, s_2, s_3, \dots, s_r)$  of  $S$  because it is the closest approximation to the original matrix [Kalman, 1996]. The diagonal matrix  $S$  has the property that  $s_r > 0$  and  $s_1 > s_2 > s_3 > \dots > s_k$ . Let  $r=1$  get the first column of  $U$ , which represents the social priorities given by node  $u$  for its incoming adjacent neighbors. In most cases,  $s_1$  represents the big portion of data with less loss. Each incoming edge is assigned a Social Priority from the column  $U_1$  (as shown in Algorithm 3 steps 14-16). At step 15, one is subtracted from the SP to reverse the value due to the use of SP to represent task's (request) position in the human queue. The study's previous assumption was that the maximum (the lowest) Social Priority value is one and the minimum (the highest) is zero, meaning that the tasks with  $SP=0$  would be on top of the human queue; and with  $SP=1$ , they would be at the end (bottom) of the human queue. Step (18) returns Social Priority Graph  $G^*$ . Usually, using SVD does not guarantee that the elements in the singular vectors will be in the interval  $[0,1]$ ; thus, the matrix columns are normalized.

---

**Algorithm 3** In-Social Priority (*iSP*) Computation

---

**Input:** A Social Graph  $G(V, E)$ , a set of social characteristics vectors

**Output:** Social Priority Graph  $G^*(V, E, P)$

```

1 For each node  $u \in V$  do
2   initialize Social Priority matrix  $SPA^u$ 
3    $L_{nei}(u) \leftarrow$  Get_incoming_adjacent_neighbors_of ( $u$ )
4    $A^u \leftarrow$  Fill_from_social_characteristics_vectors ( $L_{nei}(u)$ );
5   For each in-neighbor  $v \in L_{nei}(u)$  do
6      $p_{u,v}^G \leftarrow SP_G(x^u(G), A_{(v,G)}^u, \alpha_G)$ ;
7      $p_{u,v}^D \leftarrow SP_D(x^u(D), A_{(v,D)}^u, \alpha_D)$ ;

```

```

8    $p_{u,v}^C \leftarrow \text{SP}_C(x^u(C), A_{(v,C)}^u, \alpha_C)$ ;
9    $p_{u,v}^B \leftarrow \text{SP}_B(x^u(B), A_{(v,B)}^u, \alpha_B)$ ;
10   $p_{u,v}^E \leftarrow \text{SP}_E(x^u(E), A_{(v,E)}^u, \alpha_E)$ ;
11   $\text{SPA}^u \leftarrow \langle v, p_{u,v}^G, p_{u,v}^D, p_{u,v}^C, p_{u,v}^B, p_{u,v}^E \rangle$ ; {Add vector  $\text{SPV}^v$  to the
12  end for
13   $U_r \times S_r \times V_r^T \leftarrow \text{SVD}(\text{SPA}^u)$ ; {Matrix decomposition,  $r=1$ }
14  For each in-neighbor  $v \in L_{\text{in}}(u)$  do
15     $e_{u,v} = 1 - U_1$  {opposite Social Priority,  $i\text{SP}_{u,v}$ }
16  end for
17 end for
18: return graph  $G^*(V, E, P)$  {Social Priorities as weight on edges}

```

---

### Algorithm 3: In-Social Priority Computation

## 6.6 Analysis of social priority in some sample real networks

As previously mentioned, the benefit of using Social Priority on edge as a weight into Social Routing and Forwarding on different large scale Online Social Networks (OSNs) was evaluated in [Othman & Khan, 2015]. This section first introduces the datasets used in the study's analysis, then looks into how the proposed framework generates the social priorities between individuals in OSNs.

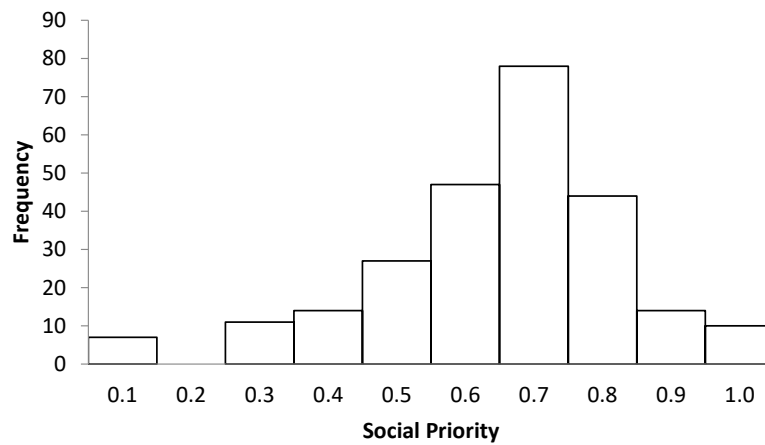
### 6.6.1 Dataset Descriptions

For the study's analysis, four data sets downloaded from Jure Leskovec's Website were used (as shown in Table 2 in section 4.4). Each dataset contains a set of nodes (subscribers) and their social connections; it also includes information such as gender. There is no overlap (same nodes in datasets) between the used datasets. The user-profile information of Google+ datasets includes six categories (gender, last name, job titles, institutions, universities, and places lived). See the following reference [J. J. McAuley & Leskovec, 2012] for more information about the used datasets.

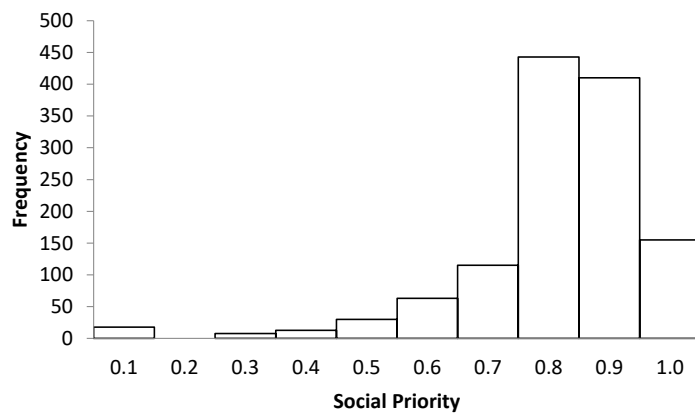
### 6.6.2 Analysis

Histograms of Social Priorities of the used datasets were generated using the study's proposed framework (shown in Figure 41, Figure 42, Figure 43, and Figure 44). The list of social priorities was grouped into intervals based upon their values, and the frequency for each interval (number of edges/ties) was found, as shown in the histogram where the horizontal axis is labeled Social Priority intervals and the vertical axis is labeled frequency.

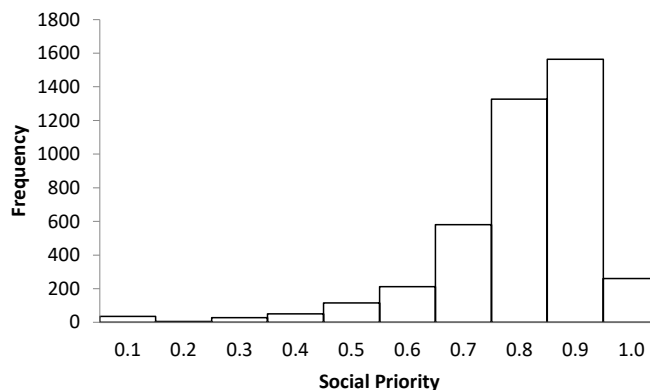
In the four datasets, the number of edges with low social priority values between individuals in small OSNs is large, while in large OSNs it is small. For instance, in the small dataset (DS-1) the distribution of social priorities on edges is normally distributed, and in the other large datasets (DS-2, DS-3, and DS-4) the social priorities are clustered around 0.9 and 1.0. This reflects the reality that individuals in large communities are normally grouped because of beneficial reasons (and not because of social relationships), and thus, the social priorities between them are low. To interpret this phenomenon, one can examine the individual level in the next subsection.



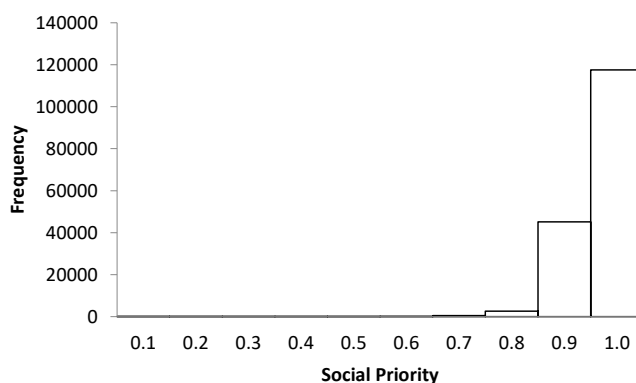
**Figure 41: Histograms of Social Priorities of DS-1**



**Figure 42: Histograms of Social Priorities of DS-2**



**Figure 43: Histograms of Social Priorities of DS-3**



**Figure 44: Histograms of Social Priorities of DS-4**

## 6.7 Discussion

This section focuses on one kind of individual-to-individual evaluation: Social Priority. According to research done for this study, this study is the first work that attempts to calculate Social Priority between individuals based on the social characteristics of individuals, which can be seen as a complementary to analyzing the sent and received emails as in [Paweł Lubarski & Morzy, 2014]. A computational framework is presented here to analyze how people give Social Priority to each other;

five social metrics have been proposed to estimate Social Priority. It can be assumed that individuals employ the same algorithm with the same social metrics, but with different social factor values.

However, this framework can be improved upon by adding the content of the task. In some cases, people give tasks high priority even if they give the requester a low priority, because of the importance or benefits (social credit, money, or reciprocity) they expect to get (in either the present or the future) when they respond. By combining the requester (who is asking) with the task content (type of task), it should improve performance over using just one of these strategies. The other kind of improvement is by adding an Indirect Social Priority, which is given a person who doesn't have a direct connection to the person, such as a friend of a friend who has not been met but has been talked about.

Furthermore, Social Priority needs to be changed (increasing or decreasing) with time after interactions between individuals. This change happens on the whole path, not just the source and the destination, as described in SOR [Othman & Khan, 2015]. Everyone involved in sending the task has to estimate the given Social Priority changes. Social and linguistic sentiment analysis techniques could be used for classifying task contents and individuals (requestors and providers) respectively. Finally, more computational frameworks and techniques that can analyze individual-to-individual evaluation, such as giving Social Priority, are still urgently needed with this new era of social platforms.

## 6.8 Conclusion

Barabási assumed that each individual has a priority list with  $L$  tasks, each task being assigned a priority value  $x_i \in [0, 1]$ , where  $i=1, \dots, L$ , chosen from  $\alpha(x)$  distribution. To the best of our knowledge no one has tried to introduce a framework to generate priority between individuals based on social factors. Barabási and others rather use different distributions to generate priority values. Motivated by the fact that humans execute their tasks based on a perceived priority [Min, Goh, & Kim, 2009; J Gama Oliveira & Vazquez, 2009; Vázquez et al., 2006], a social characteristic-based framework is introduced that can rank direct neighbors for Online Social Network (OSN) using Singular Value Decomposition (SVD). The derived ranks are specifically called Social Priorities (SP). The study's framework serves to determine the position of the task in the to-do-list (human queue). Datasets from Google+ are analyzed. The study has found that the ties in larger communities tend to have lower SPs while those in smaller groups tend to be normally distributed.

Some possible extensions to the study's framework are 1) combining the requester (who is asking) with the task content (type of task); 2) including indirect Social Priority, which is given to people who are just heard about (a friend of a friend) but with whom no direct connection made; and 3) including dynamic Social Priority, which is not fixed but changes overtime.

## CHAPTER 7

### **Online Social Network Simulator**

This chapter defines OMNeT++, describes the general architecture of the study's Online Social Network Simulator, and introduces the parameters, rules and methods of internal flow of messages inside the SOR node.

#### **7.1 OMNeT++**

OMNeT++ is not a network simulator itself but is rather a simulation environment for discrete event-driven simulations. It is an extensible, modular, component-based C++ simulation library and framework used primarily for building network simulators [Varga, 2001]. OMNeT++ could be used to simulate different routing protocols. A number of network simulators exist such as ns-2, ns-3, OMNET++, SWAN, OPNET, etc. However, OMNeT++ was chosen because it provides an infrastructure for writing different simulations.

It is used to build an Online Social Network Simulator for SOR protocol to validate and verify SOR protocol, as well as to evaluate the performance of its routing algorithms. The simulator has two parts: the Simulator and the Online User Interface.

#### **7.2 Simulator Architecture**

The simulator is designed in a way that makes it flexible for future improvements. It consists of three layers (User Interface Layer, Intermediate Layer, and Workforce



Layer), as shown in Figure 52. The internal architecture of the node consists of five independent components, as shown in Figure 50.

**Receptionist (recip)** is responsible for sending/receiving messages and message updates from other nodes in the network. The receptionist is the only component that can receive external messages from neighbors. Figure 45 shows the parameters and gates of the receptionist component.

<b>receptionist</b>
Parameters:
address
Gates:
port[]
oute
outf
inr
dirIn @directIn

**Figure 45: Receptionist Parameters**

**Generator (gen)** generates new messages to a given destination. The destination node could be generated randomly or read from a file. Figure 46 shows the parameters and gates of the generator component.

<b>lgenerator</b>
Parameters:
laTime
address
Gates:
out

**Figure 46: Generator Parameters**

**Router (rut)** runs different routing algorithms (as discussed in CHAPTER 4) to get the best/shortest paths to all reachable destinations in the network. Figure 47 shows the parameters and gates of the router component.

```

lrouter
Parameters:
--
Gates:
ing
inf
outr

```

**Figure 47: Router Parameters**

**Forwarder (fwd)** keeps the untargeted messages in its forwarding queue while the Receptionist is busy. Figure 48 shows the parameters and gates of the component.

```

lforwarder
Parameters:
laTime
Gates:
in
out

```

**Figure 48: Forwarder Parameters**

**Executer (exe)** serves the targeted messages either immediately after arrival or holds them in its service queue until there is a chance to serve them. Figure 49 shows the parameters and gates of the component.

```

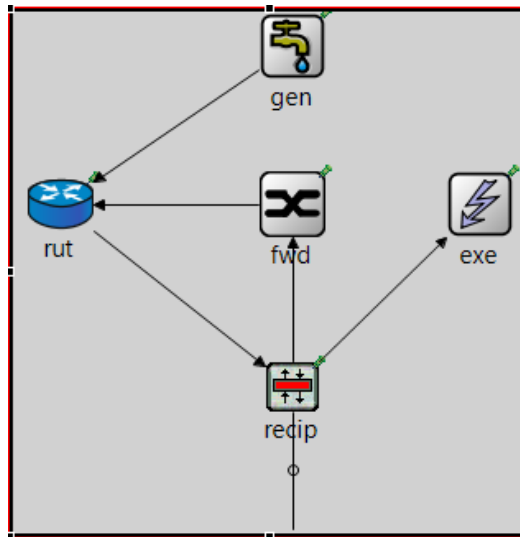
lexecuter
Parameters:
laTime
Gates:
in

```

**Figure 49: Executer Parameters**

The internal interconnection between those components (as depicted in Figure 50) is done as follows: The receptionist is directly connected to the Forwarder, Executer, and Router. The type of connection is outgoing to the Forwarder and Executer, and ingoing from the Router. There is no direct connection with the Generator, which in turn has a directly outgoing connection with the Router. There is a connection from the Forwarder

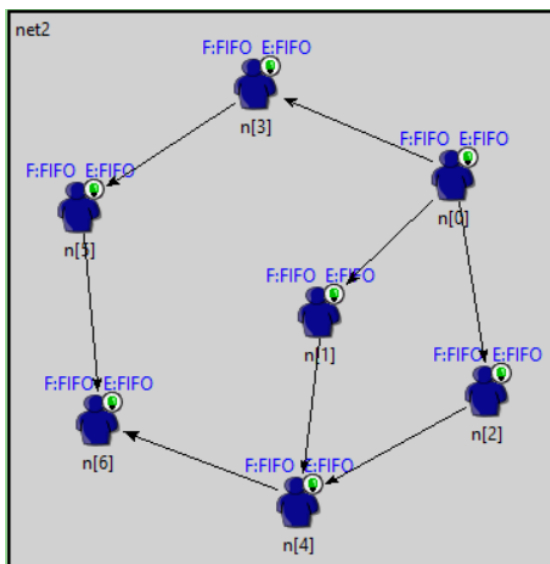
to the Router. Finally, the Receptionist has direct connections to all other nodes' directly connected Receptionists.



**Figure 50: The internal interconnection of node's components**

The flow of messages between components happens in a particular manner where the Receptionist either receives messages externally through its direct neighbors or internally from the Router. In the former case, based on the destination field associated with the message, the Receptionist makes one of two decisions: either send the message to the Executer if the current node is the destination, or to the Forwarder if the current node is not the destination of the message. In the latter case, the Receptionist just sends the received message to one of its directly connected neighbors. While the Executer has no outgoing link to other components (which means no outgoing flow), the Forwarder has a link. After receiving the untargeted message from the Receptionist, it pushes the untargeted message at a particular position in its forwarding queue based on a specific criteria (as discussed in CHAPTER 6), and after a specified time (specific distribution)

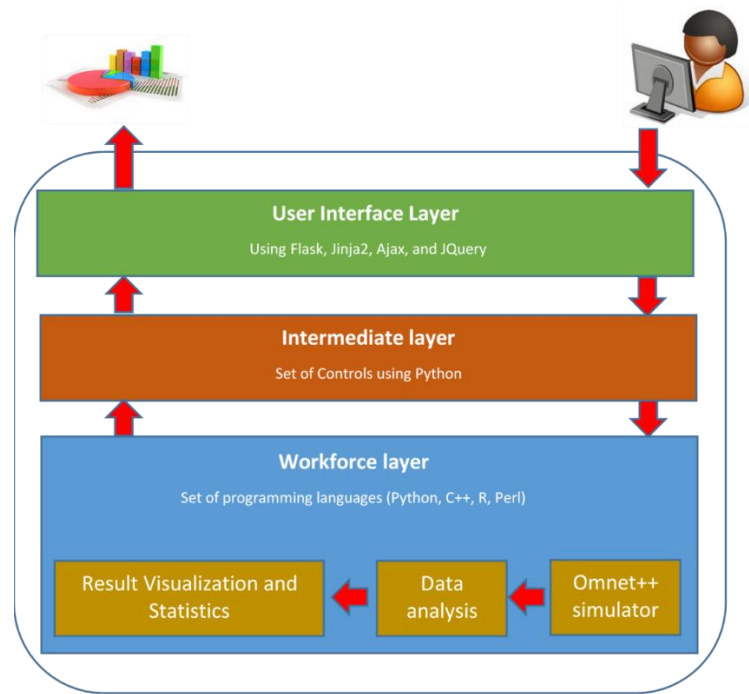
pops it from its queue and sends it to the Router. The Router also has two ways of receiving messages: 1) from the Forwarder, or 2) from the Generator. The Router looks into its routing table or dynamically tries to get the path to the request destination as well as the port to next directly connected node. It then sends the message to the Receptionist. Figure 51 presents a small social network of seven nodes and their social ties. Each node is associated with two queues: Forwarding (F) and Servicing (E).



**Figure 51: Small Social Network**

### 7.3 Online User Interface

After designing and implementing the Social Network Simulator, it was noticed that in order to run the study's simulator without any complexity, a friendly simple web-based user interface is needed for users (e.g., social scientists). The domain name of the website is <http://www.osimulator.com/> (as shown in Figure 53). The interface is designed and implemented using Python, Flask, Jinja2, Ajax, and JQuery (as shown in Figure 52).



**Figure 52: Online Social Network Simulator architecture**

A user needs to 1) connect to a server (Amazon Web Services, Microsoft Azure, etc.) to run the simulator; 2) create a new folder or connect to an existing one to save his/her datasets and results; 3) upload a social graph or generate a random graph; 4) configure some parameters (e.g. the number of messages to be generated by each node, the routing algorithm, the queue type and so on.); and 5) run the simulator and get the results (Total\_Delay, Total\_Delay\_statistics\_ByNode, Network\_TotalDelay, HopCount, HopCount\_Statistics\_ByNode, Network\_MaxHopCount, Forward\_Queue\_Lengths, FrwdQueue\_statistics\_ByNode, Network\_MaxForwardQueue, Service\_Queue\_Lengths, ServQueue\_statistics\_ByNode, Network\_MaxServiceQueue). See the two figures below.

## Five simple steps to run the simulator and get results!

**Choose a Machine**

Amazon Web Services(Not available)  
 Microsoft Azure(Not available)  
 Server at Kent State University  
 Salem's Laptop(Not available)

Sometimes the port to connect to the machine will be closed. To wake it up, send a request.

Server at Kent State University is available!

**Create a New Folder Or Connect to Existing one**

You have to create a new folder to contains your networks and result files. Or, connect to your existing folder.

**Enter Folder Name**

**Enter Four Digit Number**

For future use, you should remember the folder name and the four digit number.

You are connected to your folder and there are some networks in your folder. Please check the table below

**Generate or Upload a Network**

**Name of Network:**

**Number of nodes:**

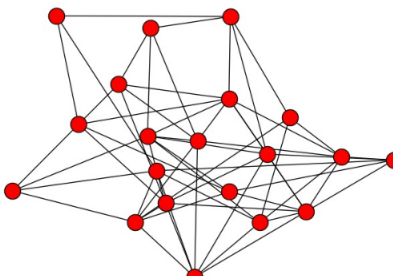
  

**Number of Edges:**

**List of networks in your folder:**

Network Name	Number of Nodes	Number of Edges	File Type
<input checked="" type="radio"/> net7	20.0	60.0	Excel File
<input type="radio"/> Qw	10.0	30.0	Excel File
<input type="radio"/> net1	54.0	120.0	Excel File



The topology of network [net7] has been plotted.

**Figure 53: Online User Interface (connecting to a server)**

**Parameters Configuration**

This configuration is for SOR Protocol ver 1. For more information about the parameters check this paper.  
[SOR: A Protocol for Requests Dissemination in Online Social Networks](#)

**Number Of Messages to Be Generated:**

**Routing algorithm:**  
 ▼

**Forwarding Queue Type:**  
 ▼

**Servicing Queue Type:**  
 ▼

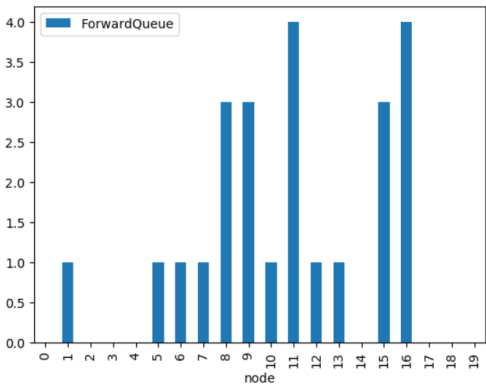
**Run and Get Results!**

Run Online for small networks  
 Run Offline for large scale networks[Not available, but it is coming soon!]

The result is ready. Look below.

**The Result Will be Shown Here**

[Download Results as Microsoft Excel File](#)



node	ForwardQueue
1	1.0
2	0.0
3	0.0
4	0.0
5	1.0
6	1.0
7	1.0
8	3.0
9	3.0
10	1.0
11	4.0
12	1.0
13	1.0
14	0.0
15	3.0
16	4.0
17	0.0
18	0.0
19	0.0

**Figure 54: Online User Interface (parameters configuration and result)**

## CHAPTER 8

### Conclusions and Future Work

A Social Online Routing (SOR) protocol for social routing in OSNs has been proposed in the preceding chapters. This final chapter begins by reviewing the conclusions, and ends with introducing some limitations of the study and future research lines.

#### 8.1 Contribution

A protocol for social routing in OSNs can help individuals to interact with people not directly connected to them (for example, looking for somebody to write a recommendation for a job at a particular company, searching for tutors, or looking for babysitters). However, the design and implementation of protocol for OSNs is not an easy task and requires privacy, security, and performance goals to be achieved. To address some of these design issues this research *first* proposes Social Online Routing (SOR) protocol for social routing in OSNs. The protocol includes the following (as described in Chapter Three):

1. Five messages (I-need Message, I-have Message, I-thank Message, I-like/dislike message, and the I-Ack Message) for carrying routing information.
2. Five tables to store routing information and policies (Messages Table, Forwarding Table, and Routing Table, Self-Interest Table, and Peer-like/dislike Table).



3. Four forwarding modules to exchange messages between nodes (I-need, I-have, I-thank, and I-ack).
4. Two Attribute-based languages for message propagation.
5. Three routing algorithms (Topology aware Shortest-Path-Based routing algorithm, Social-Priority-Based routing algorithm, Queue aware Social-Priority-Based routing algorithm) for social routing.
6. Four anonymization techniques for stratified privacy (Real Identity, Globally unique Pseudo Identity, Locally unique Pseudo Identity, and Null Identity).

*Second*, Chapter Four presents the reachability and efficiency (end-to-end routing delays) of SOR. The study has found that social routing can be achieved with different end-to-end routing delays by using the disclosed information elements. In the case of most open choices both reachability and near optimum routing performance are guaranteed. In the case in between privacy choices, the performance degrades gracefully. It has been shown that reachability is also guaranteed in the case of the most restricted choices of privacy.

*Third*, the privacy-preserving properties of SOR protocol are analyzed. The modes of user choices and degree of anonymity of service consumer are also examined. A Proxima matrix and a Proxima distribution are proposed for measuring the degree of anonymity. It has been found that the degree of anonymity increases when the distance between the victim and the attacker increases, but only up to a certain point; afterward, as

the distance continues to increase, the degree of anonymity decreases. This trend is clear in the scatter plots found in Chapter Five.

*Fourth*, a simulator was designed and built in order to evaluate this proposed protocol. Using this simulator and a real-world OSN with heterogeneous social characteristics, a set of experiments on each algorithm was conducted to evaluate the performance of the study's algorithms by using various quantitative interrelating adhered metrics. This simulator was described in Chapter Seven.

*Finally*, motivated by the fact that humans execute their tasks based on a perceived priority, a social characteristic-based framework is introduced that can rank direct neighbors for an Online Social Network (OSN) using Singular Value Decomposition (SVD). Datasets from Google+ are analyzed. The study found that ties in larger communities tend to have lower SPs while those in smaller groups tend to be normally distributed. This framework was presented in Chapter Six.

## **8.2 Limitations and Future Work**

Beyond the contributions of this dissertation, the researcher of this study still aims to achieve a number of research goals in the future.

### **8.2.1 Incentivization**

Individuals will contribute to SOR only based on their personal utility gain. The key question is how individuals in OSNs can be motivated to use SOR. One answer to this question may be that incentive mechanisms (such as a point system or micropayment scheme) are needed. Such mechanisms motivate and encourage individuals to share some information, to accept messages and queue them, and to participate in forwarding and routing. Frameworks from Game theory (a branch of applied mathematics) can be used in this research direction.

### **8.2.2 Misbehaving**

The following research questions need to be investigated: 1) What can encourage a selfish sender (service consumer) to create good propagation rules, and discourage the selfish consumer from creating inappropriate (wide) propagation rules? 2) What mechanism can prevent forwarders from tampering with the propagation rules?

### **8.2.3 Privacy of advertisement**

The I-like/dislike message can progress by one step (only to an adjacent neighbor); because of that, encryption is not needed. The I-have message can go more than one hop and it can be encrypted. However, the I-need message can also go more

steps in the network and is not encrypted. The question then is if there as any case (e.g. application) that requires the I-need message to be encrypted. SOR protocol cannot support that. This is a research line requiring further investigation.

#### **8.2.4 Security**

The security of any protocol is a precondition with minimal requests: *confidentiality* (only the service provider and service consumer should be able to open the contents of I-have and I-Ack messages), *authentication* (the service provider and service consumer confirm their identities), and *message integrity* (the service consumer and service provider want to ensure that the messages, which go beyond one step, are not altered in transit by forwarders without detection). A hybrid cryptosystem which consists of both asymmetric and symmetric algorithms is needed to secure SOR. However, in the literature, there are plenty of public/private keys (symmetric encryption, asymmetric encryption) that are based on mechanisms such as digital signatures, digital certificates, digital envelopes, and chains of trusted nodes known as the “Web of Trust”. These mechanisms can be readily used in SOR. This is another research line needing to be investigated.

#### **8.2.5 Social Priority**

Some possible extensions to the social characteristic-based framework are 1) combining the requester (who is asking) with the task content (type of task); 2) including indirect Social Priority, which is given to people who are just heard about (a friend of a friend) but with whom no direct connection is made; and 3) including dynamic Social

Priority, which is not fixed but changes overtime. This research line also needs to be investigated.

### **8.2.6 Reachability**

When the Real Identity, globally unique Pseudo Identity, and locally unique Pseudo Identity are used, reachability in SOR is guaranteed. This becomes more difficult to achieve, however, when the Null Identity is used. For example, if a set of nodes chooses to be hidden and are looking for the same service (e.g. babysitters), then the service provider can only send an I-have message back to one of the senders, while the others will not be able to receive it. This research line is one among many requiring further investigation.

### **8.2.7 Routing Loops Prevention**

One of the routing issues that any protocol must be aware of are routing loops. For example, in regular networks, the RIP (Routing Information Protocol) uses three different mechanisms (Split Horizon, Route Poisoning, and Holddown) to prevent routing loops. In SOR, they can be prevented based on the used identity; in cases where Real Identity, globally unique Pseudo Identity, and locally unique Pseudo Identity are used, one of the known mechanisms in regular networks (e.g. decreasing the hop count) can be used. However, when using Null Identity, preventing routing loops becomes harder to accomplish because neither the known mechanisms in regular networks nor the hop count decreasing technique (for privacy reasons) can be used to prevent routing loops. Thus, this is another research line needing more study.

### **8.2.8 Application**

Lastly, for various services, various fields (domain objects) should be defined and used. Application designers need to define some attributes, such those in the United Nations Standard Products and Services Code (UNSPSC) [Dumas, O’Sullivan, Heravizadeh, Edmond, & Ter Hofstede, 2003], over OSN to exchange services. The standard data fields of SOR are compatible with it. UNSPSC is an open, global, multi-sector standard for efficient, accurate classification of products and services[Sicilia, Manouselis, & Costopoulou, 2006]. It enables the procurement to deliver on cost-effectiveness demands, allowing for full exploitation of electronic commerce capabilities. However, a product naming scheme for services is difficult and names must be distinct. This is a research line needing to be investigated.

## REFERENCES

- Aberdeen, D., Pacovsky, O., & Slater, A. (2010). The learning behind gmail priority inbox. Paper presented at the LCCC: NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds.
- Adamic, L., & Adar, E. (2005). How to search a social network. *Social networks*, 27(3), 187-203.
- Aggarwal, C. C., & Philip, S. Y. (2008). A general survey of privacy-preserving data mining models and algorithms *Privacy-preserving data mining* (pp. 11-52): Springer.
- Anderson, P., Kourtellis, N., Finnis, J., & Iamnitchi, A. (2010). On managing social data for enabling socially-aware applications and services. Paper presented at the Proceedings of the 3rd Workshop on Social Network Systems.
- Andersson, C., & Lundin, R. (2007). On the fundamentals of anonymity metrics. Paper presented at the IFIP International Summer School on the Future of Identity in the Information Society.
- Ayday, E., Raisaro, J. L., Hubaux, J.-P., & Rougemont, J. (2013). Protecting and evaluating genomic privacy in medical tests and personalized medicine. Paper presented at the Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society.
- Bai, K., Liu, Y., & Liu, P. (2009). Prevent identity disclosure in social network data study. Paper presented at the ACM CCS.

- Banerjee, A., & Basu, S. (2008). A social query model for decentralized search. Paper presented at the Proceedings of the 2nd Workshop on Social Network Mining and Analysis. ACM, New York.
- Barabasi, A.-L. (2005). The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039), 207-211.
- Barabási, A.-L. (2016). *Network science*: Cambridge university press.
- Barbella, D., Kachergis, G., Liben-Nowell, D., Sallstrom, A., & Sowell, B. (2007). Depth of field and cautious-greedy routing in social networks. Paper presented at the International Symposium on Algorithms and Computation.
- Bertocco, M., Ferraris, F., Offelli, C., & Parvis, M. (1998). A client-server architecture for distributed measurement systems. *IEEE transactions on instrumentation and measurement*, 47(5), 1143-1148.
- Blanchard, P., & Hongler, M.-O. (2007). Modeling human activity in the spirit of barabasi's queueing systems. *Physical Review E*, 75(2), 026102.
- Bolch, G., Greiner, S., de Meer, H., & Trivedi, K. S. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*: John Wiley & Sons.
- Boldrini, C., Conti, M., & Passarella, A. (2008). Exploiting users' social relations to forward data in opportunistic networks: The HiBOP solution. *Pervasive and Mobile Computing*, 4(5), 633-657.
- Boldrini, C., Conti, M., & Passarella, A. (2009). Social-based autonomic routing in opportunistic networks *Autonomic Communication* (pp. 31-67): Springer.



- Borgatti, S. P. (2005). Centrality and network flow. *Social networks*, 27(1), 55-71.
- Brock, D. L. (2001). The electronic product code (epc). Auto-ID Center White Paper MIT-AUTOID-WH-002.
- Brodka, P., Stawiak, P., & Kazienko, P. (2011). Shortest path discovery in the multi-layered social network. Paper presented at the Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on.
- Brubaker, R., & Cooper, F. (2000). Beyond "identity". *Theory and society*, 29(1), 1-47.
- Bu, T., & Towsley, D. (2002). On distinguishing between Internet power law topology generators. Paper presented at the INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.
- Cao, J., & Karras, P. (2012). Publishing microdata with a robust privacy guarantee. *Proceedings of the VLDB Endowment*, 5(11), 1388-1399.
- Castro, M., Druschel, P., Ganesh, A., Rowstron, A., & Wallach, D. S. (2002). Secure routing for structured peer-to-peer overlay networks. *ACM SIGOPS Operating Systems Review*, 36(SI), 299-314.
- Clauß, S., & Schiffner, S. (2006). Structuring anonymity metrics. Paper presented at the Proceedings of the second ACM workshop on Digital identity management.
- Dale, E., Osili, U., Mesch, D., & Ackerman, J. (2015). Where Do Men and Women Give? Gender Differences in the Motivations and Purposes for Charitable Giving.

- Davis, A., Gardner, B. B., Gardner, M. R., & Warner, W. L. (1941). *Deep South: A Sociological Anthropological Study of Caste and Class*: University of Chicago Press.
- Demetrescu, C., & Italiano, G. F. (2004). Engineering shortest path algorithms. Paper presented at the International Workshop on Experimental and Efficient Algorithms.
- Dequiedt, V., & Zenou, Y. (2014). Local and consistent centrality measures in networks.
- Dezsö, Z., Almaas, E., Lukács, A., Rácz, B., Szakadát, I., & Barabási, A.-L. (2006). Dynamics of information access on the web. *Physical Review E*, 73(6), 066132.
- Diaz, C. (2006). Anonymity metrics revisited. Paper presented at the Dagstuhl Seminar Proceedings.
- Diaz, C., Seys, S., Claessens, J., & Preneel, B. (2002). Towards measuring anonymity. Paper presented at the International Workshop on Privacy Enhancing Technologies.
- Diaz, C., Troncoso, C., & Danezis, G. (2007). Does additional information always reduce anonymity? Paper presented at the Proceedings of the 2007 ACM workshop on Privacy in electronic society.
- Diaz, C., Troncoso, C., & Serjantov, A. (2008). On the impact of social network profiling on anonymity. Paper presented at the International Symposium on Privacy Enhancing Technologies Symposium.
- Dodds, P. S., Muhamad, R., & Watts, D. J. (2003). An experimental study of search in global social networks. *science*, 301(5634), 827-829.

- Dreesen, P., Batselier, K., & De Moor, B. (2012). Weighted/Structured Total Least Squares problems and polynomial system solving. Paper presented at the ESANN.
- Dumas, M., O'Sullivan, J., Heravizadeh, M., Edmond, D., & Ter Hofstede, A. (2003). Towards a semantic framework for service description Semantic issues in e-commerce systems (pp. 277-291): Springer.
- Easley, D., & Kleinberg, J. (2007). The small-world phenomenon. *Networks*, Spring.
- Eppstein, D., Goodrich, M. T., Löffler, M., Strash, D., & Trott, L. (2013). Category-based routing in social networks: Membership dimension and the small-world phenomenon. *Theoretical Computer Science*, 514, 96-104.
- Farzad, B., Olver, N., & Vetta, A. (2008). A priority-based model of routing. *Chicago Journal of Theoretical Computer Science*, 1.
- Faust, K. (1997). Centrality in affiliation networks. *Social networks*, 19(2), 157-191.
- Feuz, K. D., & Allan, V. H. (2012). Simulating Pedestrian Route Selection with Imperfect Knowledge. Paper presented at the ICAART (2).
- Feuz, K. D., & Allan, V. H. (2013). Group Formation and Knowledge Sharing in Pedestrian Egress Simulation. Paper presented at the ICAART (1).
- Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3), 215-239.
- Freudiger, J., Raya, M., Félegyházi, M., Papadimitratos, P., & Hubaux, J.-P. (2007). Mix-zones for location privacy in vehicular networks. Paper presented at the ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS).

- Friedkin, N. E. (1991). Theoretical foundations for centrality measures. *American journal of Sociology*, 1478-1504.
- Friedl, D.-M. B., & Heidemann, J. (2010). A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, 2(6), 371-385.
- Fujii, T., Ren, Y., Hori, Y., & Sakurai, K. (2009). Security Analysis for P2P Routing Protocols. Paper presented at the Availability, Reliability and Security, 2009. ARES'09. International Conference on.
- Ganta, S. R., Kasiviswanathan, S. P., & Smith, A. (2008). Composition attacks and auxiliary information in data privacy. Paper presented at the Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.
- Gavoille, C. (2000). A survey on interval routing. *Theoretical Computer Science*, 245(2), 217-253.
- Gavoille, C. (2001). Routing in distributed networks: Overview and open problems. *ACM SIGACT News*, 32(1), 36-52.
- Gilbert, E., & Karahalios, K. (2009). Predicting tie strength with social media. Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.
- Giordano, S., & Stojmenovic, I. (2004). Position based routing algorithms for ad hoc networks: A taxonomy *Ad hoc wireless networking* (pp. 103-136): Springer.
- Goga, O., Loiseau, P., Sommer, R., Teixeira, R., & Gummadi, K. P. (2015). On the reliability of profile matching across large online social networks. Paper presented

at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Gong, N. Z., Talwalkar, A., Mackey, L., Huang, L., Shin, E. C. R., Stefanov, E., . . .

Song, D. (2014). Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2), 27.

Gong, N. Z., Xu, W., Huang, L., Mittal, P., Stefanov, E., Sekar, V., & Song, D. (2012).

Evolution of social-attribute networks: measurements, modeling, and implications using google+. Paper presented at the Proceedings of the 2012 Internet Measurement Conference.

Gonzalez, R., Cuevas, R., Motamedi, R., Rejaie, R., & Cuevas, A. (2013). Google+ or

google-?: dissecting the evolution of the new osn in its first year. Paper presented at the Proceedings of the 22nd international conference on World Wide Web.

Gross, R., & Acquisti, A. (2005). Information revelation and privacy in online social

networks. Paper presented at the Proceedings of the 2005 ACM workshop on Privacy in the electronic society.

Hamel, A., Grégoire, J.-C., & Goldberg, I. (2011). The misentropists: New approaches to

measures in tor. Centre for Applied Cryptographic Research (CACR).

Han, X.-P., Zhou, T., & Wang, B.-H. (2008). Modeling human dynamics with adaptive

interest. *New Journal of Physics*, 10(7), 073010.

Hangal, S., MacLean, D., Lam, M. S., & Heer, J. (2010). All friends are not equal: Using

weights in social graphs to improve search.

- He, W., Liu, X., Nguyen, H., Nahrstedt, K., & Abdelzaher, T. (2007). Pda: Privacy-preserving data aggregation in wireless sensor networks. Paper presented at the IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications.
- Helland, I. S. (1987). On the interpretation and use of  $R^2$  in regression analysis. *Biometrics*, 61-69.
- Helmhout, J. M. (2006). *The Social Cognitive Actor: A multi-actor simulation of organisations*: University Library Groningen][Host].
- Holme, P., Kim, B. J., Yoon, C. N., & Han, S. K. (2002). Attack vulnerability of complex networks. *Physical Review E*, 65(5), 056109.
- Huang, Y., & Garcia-Molina, H. (2004). Publish/subscribe in a mobile environment. *Wireless Networks*, 10(6), 643-652.
- Hui, P., & Sastry, N. (2009). Real world routing using virtual world information. Paper presented at the Computational Science and Engineering, 2009. CSE'09. International Conference on.
- Humbert, M., Studer, T., Grossglauser, M., & Hubaux, J.-P. (2013). Nowhere to hide: Navigating around privacy in online social networks. Paper presented at the European Symposium on Research in Computer Security.
- Iribarren, J. L., & Moro, E. (2009). Impact of human activity patterns on the dynamics of information diffusion. *Physical review letters*, 103(3), 038702.

- Ivanov, P., Kupriyanov, M., & Shichkina, Y. (2017). Methods for constructing optimal routes in DTN networks. Paper presented at the Soft Computing and Measurements (SCM), 2017 XX IEEE International Conference on.
- Ji, S., Li, W., Mittal, P., Hu, X., & Beyah, R. (2015). Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. Paper presented at the 24th USENIX Security Symposium (USENIX Security 15).
- Jia, J., Wang, B., Zhang, L., & Gong, N. Z. (2017). AttrInfer: Inferring user attributes in online social networks using markov random fields. Paper presented at the Proceedings of the 26th International Conference on World Wide Web.
- Jia, S., St Juste, P., & Figueiredo, R. J. (2013). A multidimensional heuristic for social routing in peer-to-peer networks. Paper presented at the Consumer Communications and Networking Conference (CCNC), 2013 IEEE.
- Jure Leskovec, A. K. (2014). Stanford Large Network Dataset Collection. Retrieved from <http://snap.stanford.edu/data/index.html>
- Kabir, M. A., Han, J., Yu, J., & Colman, A. (2012). SCIMS: a social context information management system for socially-aware applications. Paper presented at the International Conference on Advanced Information Systems Engineering.
- Kairam, S., Brzozowski, M., Huffaker, D., & Chi, E. (2012). Talking in circles: selective sharing in google+. Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.

- Kalman, D. (1996). A singularly valuable decomposition: the SVD of a matrix. *The college mathematics journal*, 27(1), 2-23.
- Kang, U., Papadimitriou, S., Sun, J., & Tong, H. (2011). Centralities in Large Networks: Algorithms and Observations. Paper presented at the SDM.
- Karsai, M., Kaski, K., Barabási, A., & Kertész, J. (2012). Universal features of correlated bursty behaviour Supplementary Informations.
- Katz, G. J., & Kider Jr, J. T. (2008). All-pairs shortest-paths for large graphs on the GPU. Paper presented at the Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware.
- Kesdogan, D., Egner, J., & Büschkes, R. (1998). Stop-and-go-mixes providing probabilistic anonymity in an open system. Paper presented at the International Workshop on Information Hiding.
- Kleinberg, J. (2000). The small-world phenomenon: An algorithmic perspective. Paper presented at the Proceedings of the thirty-second annual ACM symposium on Theory of computing.
- Kleinberg, J. (2006). Complex networks and decentralized search algorithms. Paper presented at the Proceedings of the International Congress of Mathematicians (ICM).
- Kleinberg, J. M. (2000). Navigation in a small world. *Nature*, 406(6798), 845-845.
- Klema, V., & Laub, A. (1980). The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2), 164-176.



- Kong, L., Liu, Z., & Huang, Y. (2014). Spot: Locating social media users based on social network context. *Proceedings of the VLDB Endowment*, 7(13), 1681-1684.
- Korolova, A., Motwani, R., Nabar, S. U., & Xu, Y. (2008). Link privacy in social networks. Paper presented at the Proceedings of the 17th ACM conference on Information and knowledge management.
- Kossinets, G., & Watts, D. J. (2006). Empirical analysis of an evolving social network. *science*, 311(5757), 88-90.
- Koutsopoulos, I., Noutsi, E., & Iosifidis, G. (2014). Dijkstra goes social: Social-graph-assisted routing in next generation wireless networks. Paper presented at the European Wireless 2014; 20th European Wireless Conference; Proceedings of.
- Lamb, R., & Kling, R. (2003). Reconceptualizing users as social actors in information systems research. *MIS quarterly*, 197-236.
- LARSON, R. Q. (1987). PERSPECTIVES QN QUEUES: SOCIAL JUSTICE AND THE PSYCHOLQGY OF QUEUEING.
- Lattanzi, S., Panconesi, A., & Sivakumar, D. (2011). Milgram-routing in social networks. Paper presented at the Proceedings of the 20th international conference on World wide web.
- Leskovec, J., & Horvitz, E. (2008). Planetary-scale views on a large instant-messaging network. Paper presented at the Proceedings of the 17th international conference on World Wide Web.

- Leskovec, J., Huttenlocher, D., & Kleinberg, J. (2010a). Predicting positive and negative links in online social networks. Paper presented at the Proceedings of the 19th international conference on World wide web.
- Leskovec, J., Huttenlocher, D., & Kleinberg, J. (2010b). Signed networks in social media. Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.
- Li, F., Jiang, H., Wang, Y., Li, X., Wang, M., & Abdeldjalil, T. (2013). SEBAR: Social energy based routing scheme for mobile social delay tolerant networks. Paper presented at the Performance Computing and Communications Conference (IPCCC), 2013 IEEE 32nd International.
- Li, S., Su, L., Suleimenov, Y., Liu, H., Abdelzaher, T., & Chen, G. (2014). Centaur: Dynamic message dissemination over online social networks. Paper presented at the 2014 23rd International Conference on Computer Communication and Networks (ICCCN).
- Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., & Tomkins, A. (2005). Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33), 11623-11628.
- Linnolahti, J. (2004). QoS routing for P2P networking. Paper presented at the HUT T-110.551 Seminar on Internetworking.
- Liu, H., & Maes, P. (2005). Interestmap: Harvesting social network profiles for recommendations. *Beyond Personalization-IUI*, 56.

- Liu, H., Pardoe, D., & Liu, K. Audience Expansion for Online Social Network Advertising.
- Liu, L., & Jing, Y. (2012). A Survey on Social-Based Routing and Forwarding Protocols in Opportunistic Networks. Paper presented at the Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on.
- Lubarski, P., & Morzy, M. (2012). Measuring the importance of users in a social network based on email communication patterns. Paper presented at the Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).
- Lubarski, P., & Morzy, M. (2014). @ Rank: Personalized Centrality Measure for Email Communication Networks State of the Art Applications of Social Network Analysis (pp. 209-225): Springer.
- Malmgren, R. D., Stouffer, D. B., Motter, A. E., & Amaral, L. A. (2008). A Poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences*, 105(47), 18153-18158.
- Martel, C., & Nguyen, V. (2004). Analyzing Kleinberg's (and other) small-world models. Paper presented at the Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing.
- Martin, D. J., Kifer, D., Machanavajjhala, A., Gehrke, J., & Halpern, J. Y. (2007). Worst-case background knowledge for privacy-preserving data publishing. Paper presented at the 2007 IEEE 23rd International Conference on Data Engineering.

- Masoumzadeh, A., & Joshi, J. (2012). Preserving structural properties in edge-perturbing anonymization techniques for social networks. *IEEE Transactions on Dependable and Secure Computing*, 9(6), 877-889.
- Maydeu-Olivares, A., & Garcia-Forero, C. (2010). Goodness-of-fit testing. *International encyclopedia of education*, 7(1), 190-196.
- Mcauley, J., & Leskovec, J. (2014). Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1), 4.
- McAuley, J. J., & Leskovec, J. (2012). Learning to Discover Social Circles in Ego Networks. Paper presented at the NIPS.
- Medhi, D., & Ramasamy, K. (2017). *Network routing: algorithms, protocols, and architectures*: Morgan Kaufmann.
- Meng, M., Xu, H., Wu, X., d'Auriol, B. J., Jeong, B.-S., Lee, S., & Fan, X. (2007). PBR: Priority Based Routing in Multi-Sink Sensor Networks. Paper presented at the ICWN.
- Merugu, S., & Ghosh, J. (2003). Privacy-preserving distributed clustering using generative models. Paper presented at the Data Mining, 2003. *ICDM 2003*. Third IEEE International Conference on.
- Michlmayr, E., Pany, A., & Kappel, G. (2007). Using taxonomies for content-based routing with ants. *Computer Networks*, 51(16), 4514-4528.
- Milgram, S. (1967). The small world problem. *Psychology today*, 2(1), 60-67.
- Min, B., Goh, K.-I., & Kim, I.-M. (2009). Waiting time dynamics of priority-queue networks. *Physical Review E*, 79(5), 056110.

- Mondal, M., Liu, Y., Viswanath, B., Gummadi, K. P., & Mislove, A. (2014). Understanding and specifying social access control lists. Paper presented at the Symposium On Usable Privacy and Security (SOUPS 2014).
- Mtibaa, A., May, M., Diot, C., & Ammar, M. (2010). Peoplerank: Social opportunistic forwarding. Paper presented at the INFOCOM, 2010 Proceedings IEEE.
- Mukherjee, D., & Garg, M. (2013). Which work-item updates need your response? Paper presented at the Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on.
- Murdoch, S. J. (2014). Quantifying and measuring anonymity Data Privacy Management and Autonomous Spontaneous Security (pp. 3-13): Springer.
- Narayanan, A., & Shmatikov, V. (2009). De-anonymizing social networks. Paper presented at the 2009 30th IEEE symposium on security and privacy.
- Nass, C., Steuer, J., & Tauber, E. R. (1994). Computers are social actors. Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.
- Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the Econometric Society*, 347-370.
- Oliveira, J. G., & Barabási, A.-L. (2005). Human dynamics: Darwin and Einstein correspondence patterns. *Nature*, 437(7063), 1251-1251.
- Oliveira, J. G., & Vazquez, A. (2009). Impact of interactions on human dynamics. *Physica A: Statistical Mechanics and its Applications*, 388(2), 187-192.

- Ostertagova, E. (2012). Modelling using polynomial regression. *Procedia Engineering*, 48, 500-506.
- Othman, S., & Khan, J. I. (2015). SOR: A Protocol for Requests Dissemination in Online Social Networks. Paper presented at the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction.
- Othman, S., Khan, J. I., & Nafa, F. (2016a). Does Knowledge Matter?: Efficiency of Routing Delay under Imperfect Knowledge in Online Social Networks. Paper presented at the International Conference on Computer Games, Multimedia & Allied Technology (CGAT). Proceedings.
- Othman, S., Khan, J. I., & Nafa, F. (2016b). Does Location Matter? The Efficiency of Request Propagation Based on Location in Online Social Networks. Paper presented at the International Conference on Social Computing and Social Media.
- Paykin, J., & Zdancewic, S. (2015). A linear/producer/consumer model of classical linear logic. arXiv preprint arXiv:1502.04770.
- Pfitzmann, A., & Hansen, M. (2010). A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management.
- Pontes, T., Magno, G., Vasconcelos, M., Gupta, A., Almeida, J., Kumaraguru, P., & Almeida, V. (2012). Beware of what you share: Inferring home location in social networks. Paper presented at the Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on.

- Poole, M. A., & O'Farrell, P. N. (1971). The assumptions of the linear regression model. *Transactions of the Institute of British Geographers*, 145-158.
- Qian, J., Li, X.-Y., Zhang, C., & Chen, L. (2016). De-anonymizing social networks and inferring private attributes using knowledge graphs. Paper presented at the Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on.
- Rajkumar, A., & Sharma, N. M. (2008). A distributed priority based routing algorithm for dynamic traffic in survivable WDM networks. *International Journal of Computer Science and Network Security*, 8(11), 192-198.
- Ramana, K. S., Chari, A., & Kasiviswanth, N. (2010). Trust based security routing in mobile adhoc networks. *International Journal on Computer Science and Engineering*, 2(2), 259-263.
- Reiter, M. K., & Rubin, A. D. (1998). Crowds: Anonymity for web transactions. *ACM transactions on information and system security (TISSEC)*, 1(1), 66-92.
- Roohani, A. (2015). Male and Female Social Actor Representation in Four Corners 4: A Critical Discourse Perspective. *Iranian Journal of Research in English Language Teaching*, 2(2), 23-35.
- Schnettler, S. (2009). A structured overview of 50 years of small-world research. *Social networks*, 31(3), 165-178.
- Schulten, E., Akkermans, H., Botquin, G., Dörr, M., Guarino, N., Lopes, N., & Sadeh, N. (2001). The e-commerce product classification challenge. *IEEE Intelligent systems*, 16(4), 86-89.

- Schurgot, M. R., Comaniciu, C., & Jaffres-Runser, K. (2011). Beyond traditional DTN routing: social networks for opportunistic communication. arXiv preprint arXiv:1110.2480.
- Schwartz, B. (1978). Queues, priorities, and social process. *Social Psychology*, 3-12.
- Semeria, C. (2001). Supporting differentiated service classes: queue scheduling disciplines. *Juniper networks*, 11-14.
- Serjantov, A., & Danezis, G. (2002). Towards an information theoretic metric for anonymity. Paper presented at the International Workshop on Privacy Enhancing Technologies.
- Shakimov, A., Lim, H., Cox, L. P., & Cáceres, R. (2008). Vis-à-Vis: Online social networking via virtual individual servers. submitted for publication.
- Sharad, K. (2016). Change of Guard: The Next Generation of Social Graph De-anonymization Attacks. Paper presented at the Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security.
- Sharma, A., Jagannathan, K., & Varshney, L. R. (2014). Information overload and human priority queuing. Paper presented at the 2014 IEEE International Symposium on Information Theory.
- Shmatikov, V., & Wang, M.-H. (2006). Measuring relationship anonymity in mix networks. Paper presented at the Proceedings of the 5th ACM workshop on Privacy in electronic society.



- Shoib, G., Nandhakumar, J., & Rowlands, B. (2009). A social actor understanding of the institutional structures at play in information systems development. *Information Technology & People*, 22(1), 51-62.
- Shokri, R., Theodorakopoulos, G., Le Boudec, J.-Y., & Hubaux, J.-P. (2011). Quantifying location privacy. Paper presented at the Security and privacy (sp), 2011 IEEE Symposium on.
- Sicilia, M.-A., Manouselis, N., & Costopoulou, C. (2006). Quality in metadata: a schema for e-commerce. *Online Information Review*, 30(3), 217-237.
- Socievole, A., De Rango, F., & Marano, S. (2013). Face-to-face with Facebook friends: using online friendlists for routing in opportunistic networks. Paper presented at the 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC).
- Solomonik, E., Buluc, A., & Demmel, J. (2013). Minimizing communication in all-pairs shortest paths. Paper presented at the Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on.
- Spizzirri, L. (2011). Justification and application of eigenvector centrality. *Algebra in Geography: Eigenvectors of Network*.
- Suthaputchakun, C., & Sun, Z. (2011). Priority based routing protocol in vehicular ad hoc network. Paper presented at the Computers and Communications (ISCC), 2011 IEEE Symposium on.
- Sweeney, L. (2000). Simple demographics often identify people uniquely. *Health (San Francisco)*, 671, 1-34.

- Syverson, P. (2009). Why I'm not an entropist. Paper presented at the International Workshop on Security Protocols.
- Tajfel, H. (1969). Social and cultural factors in perception. *Handbook of social psychology*, 3, 315-394.
- Thorup, M., & Zwick, U. (2001). Compact routing schemes. Paper presented at the Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures.
- Tillwick, H., & Olivier, M. (2005). Towards a framework for connection anonymity. Paper presented at the Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries.
- Travers, J., & Milgram, S. (1969). An experimental study of the small world problem. *Sociometry*, 425-443.
- Türkes, O., Scholten, H., & Havinga, P. (2013). RoRo-LT: social routing with next-place prediction from self-assessment of spatiotemporal routines. Paper presented at the Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC).
- Tyler, J. R., Wilkinson, D. M., & Huberman, B. A. (2005). E-mail as spectroscopy: Automated discovery of community structure within organizations. *The Information Society*, 21(2), 143-153.

- Valente, T. W., Coronges, K., Lakon, C., & Costenbader, E. (2008). How correlated are network centrality measures? *Connections (Toronto, Ont.)*, 28(1), 16.
- Varga, A. (2001). The OMNeT++ discrete event simulation system. Paper presented at the Proceedings of the European simulation multiconference (ESM'2001).
- Vazquez, A. (2005). Exact results for the Barabási model of human dynamics. *Physical review letters*, 95(24), 248701.
- Vázquez, A., Oliveira, J. G., Dezsö, Z., Goh, K.-I., Kondor, I., & Barabási, A.-L. (2006). Modeling bursts and heavy tails in human dynamics. *Physical Review E*, 73(3), 036127.
- Vazquez, A., Racz, B., Lukacs, A., & Barabasi, A.-L. (2007). Impact of non-Poissonian activity patterns on spreading processes. *Physical review letters*, 98(15), 158702.
- Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., & Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *ACM Sigmod Record*, 33(1), 50-57.
- Vesdapunt, N., & Garcia-Molina, H. (2016). Updating an existing social network snapshot via a limited api. Retrieved from
- Viswanath, B., Kiciman, E., & Saroiu, S. (2012). Keeping information safe from social networking apps. Paper presented at the Proceedings of the 2012 ACM workshop on Workshop on online social networks.
- Wagner, I., & Eckhoff, D. (2015). Technical privacy metrics: a systematic survey. *arXiv preprint arXiv:1512.00327*.

- Wan, Z. (2012). Priority based Dynamic Packet Assignment for Multipath Routing in Multihop Networks. *Journal of Networks*, 7(11), 1876-1883.
- Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications* (Vol. 8): Cambridge university press.
- Watts, D. J., Dodds, P. S., & Newman, M. E. (2002). Identity and search in social networks. *science*, 296(5571), 1302-1305.
- Wei, K., Liang, X., & Xu, K. (2014). A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues. *IEEE Communications Surveys & Tutorials*, 16(1), 556-578.
- Xiang, R., Neville, J., & Rogati, M. (2010). Modeling relationship strength in online social networks. Paper presented at the Proceedings of the 19th international conference on World wide web.
- Xu, Z., Min, R., & Hu, Y. (2003). HIERAS: a DHT based hierarchical P2P routing algorithm. Paper presented at the Parallel Processing, 2003. Proceedings. 2003 International Conference on.
- Yan, E., & Ding, Y. (2009). Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology*, 60(10), 2107-2118.
- Yanes, A. (2014). Privacy and Anonymity. arXiv preprint arXiv:1407.0423.
- Yang, J., McAuley, J., & Leskovec, J. (2013). Community detection in networks with node attributes. Paper presented at the Data Mining (ICDM), 2013 IEEE 13th international conference on.

- Ying, X., & Wu, X. (2008). Randomizing Social Networks: a Spectrum Preserving Approach. Paper presented at the SDM.
- Yuan, M., Chen, L., & Yu, P. S. (2010). Personalized privacy protection in social networks. *Proceedings of the VLDB Endowment*, 4(2), 141-150.
- Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 452-473.
- Zafarani, R., Abbasi, M. A., & Liu, H. (2014). *Social media mining: an introduction*: Cambridge University Press.
- Zhou, B., Pei, J., & Luk, W. (2008). A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 10(2), 12-22.
- Zhu, Y., & Bettati, R. (2005). Anonymity vs. information leakage in anonymity systems. Paper presented at the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05).
- Zhu, Y., & Bettati, R. (2009). Information leakage as a model for quality of anonymity networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(4), 540-552.
- Zhu, Y., Xu, B., Shi, X., & Wang, Y. (2013). A survey of social-based routing in delay tolerant networks: positive and negative social effects. *IEEE Communications Surveys & Tutorials*, 15(1), 387-401.