Efficiency Study of Sybil Attack on P2P Botnets

Yuhang Luo

A Thesis

In

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science

Concordia University Montreal, Quebec, Canada April

2012

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Yuhang Luo

Entitled: “Efficiency Study of Sybil Attack on P2P Botnets”

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____ Chair
Dr. X. Zhang

____ Examiner, External
Dr. L. Wang (CIISE) To the Program

____ Examiner
Dr. M. Mehmet-Ali

____ Supervisor
Dr. D. Qiu

Approved by: ____
Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

____________20_____

____
Dr. Robin A. L. Drew
Dean, Faculty of Engineering and
Computer Science

Abstract

Efficiency Study of Sybil Attack on P2P Botnets

Yuhang Luo

The main objective of this thesis is to modeling and analysis of Kademlia based Botnets in order to study the efficiency of Sybil attack on such botnets.

We start by researching the structure of Kademlia and specially its look-up procedure, i.e. the process how a node find a desired target node in the Botnet. For the simplicity of analysis, two assumptions are made: *a*) node ID space is full filled; *b*) a Sybil node replies a fake triple when it is queried.With these assumptions, the probability jumping functions and jumping matrices are derived. By adding the distribution of Sybil nodes, we obtain the probability that target nodes are found successfully($P_{success}$).

We then show numerical results of the distribution of $P_{success}$ with different system parameters. From the results, we can obtain some insight on how the parameters affect the efficiency of Sybil attack. Among all these parameters, we find that α, which is known as the number of nodes the initial node requries, is the key parameter of Kademlia based botnet. We also discuss how the triples a node keeps for every distance (k) and the total number of nodes in botnet (n) will affect $P_{success}$.

Based on our model and numerical results, we will draw some conclusions on how to make P2P botnet more robust or more vulnerable in facing Sybil attacking.

Acknowledgments

First of all, I am sincerely and heartily grateful to my supervisor Dr. Qiu, for his patience and guidance through my research in the past two years. One simply could not wish for a better or friendlier supervisor.

Secondly, I owe sincere and earnest thankfulness to my parents Luo, Jian and Chen, Yaqiong who gave my unconditional support and unlimited love. This dissertation would not have been possible without their support.

Last but not the least, it is a great pleasure to thank everyone who helped me write my dissertation successfully. Thanks to my best friends Jie Han, Ming Lei and Sining Liu who helped my a lot. Besides, I would like to show my gratitude to my officemates in 10-235.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

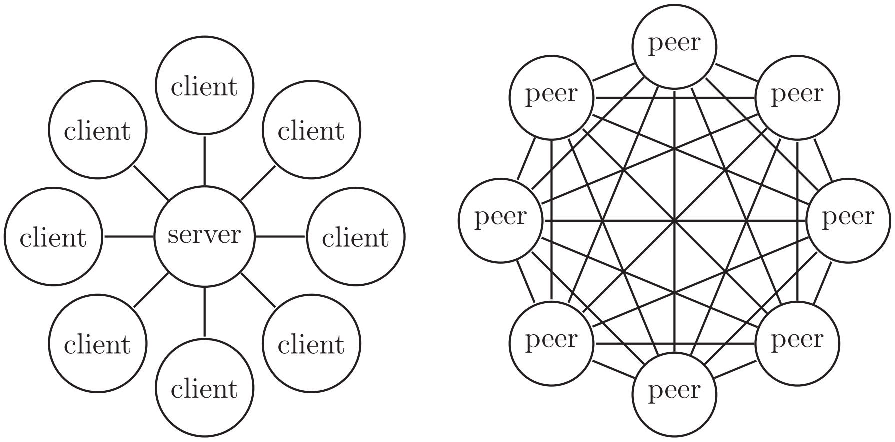
In this information era, Internet usage has been growing significantly in the past twenty years [1, 2]. With the rapid growth of the Internet, malwares have also been developed during the past decades. Among the many malwares, Botnet has been considered to be the most serious threat, because botnet makes it possible for one botmaster to control a network constituted by huge number of compromised systems [3]. The situation gets worse when botnet is combined with Peer-to-Peer(P2P)[4, 5] networks.

1.1.1 Client-Server model and Peer-to-Peer model

Peer-to-Peer(P2P) network is originally designed for sharing computer resources, such as storage, video stream and CPU cycles[6]. Recently, more and more applications use P2P as their communication model. Figure1.1a shows a schematic diagram of Client-server model. Every client connects to server directly, whereas Figure1.1b is the

P2P model that all nodes connect to each other without a central server. There are strengths and weaknesses for both models.

The Client-Server model is widely used from the beginning of Internet. Several well known Internet protocol, such as SMTP, HTTP and FTP, are based on Client-Server model [7]. Also, Internet Relay Chat(IRC) is built on Client-Server model. This model performs well in terms of server maintenance, service assurance in dedicated servers and security. Highly relying on server is the main disadvantage of Client-Server model. More clients lead to high data loading which slows down the connecting speed. And even worse, if a server gets attacked, the whole network may crash down.



(a) Client-Server model

(b) Peer to Peer model

Figure 1.1: Two types of communication model

P2P model partially solved the drawback of Client-Server model because of its structure. In P2P model, every peer acts as both client and server, so more peers lead to high loading and more bandwidths. distributing the information(file, video, etc) in the whole P2P network makes it work as usual even when some peers leave the network. Detailed advantages are listed as following:

Robustness: As a distributed network, P2P model is hard to be attacked. It's possible to shut down one or several peers, but which won't have big influence on P2P network, because it's normal to have peers join or leave dynamically. The distributed Denial-of-Service(DDoS) attack, which is a nightmare for most web sites [8], can not attack P2P network.

Anonymity: Once a P2P network has been built up, all peers are acting similarly to each other. So it is hard to know who controls the network.

Scalability: More peers bring both data loading and bandwidth at the same time. Meanwhile, more peers store more copies of information, which strengthens the network.

Storage: Every peer is a storage device in P2P model, so the storage space is extremely large.

Due to those advantages of P2P model, more and more applications use P2P model as communication module. However, the key problem in P2P model is how to search desired information, that is why the distributed hash table(DHT) is introduced.

1.1.2 Distributed hash table

The earlier P2P systems are not fully decentralized. Since the information is distributed among the whole network, an index web site is needed for searching. Napster[9] is one of those networks. Later, edonkey [10] uses a server to list the ed2k link. Even the edonkey network is decentralized, it somehow relies on the server.

To make a P2P network fully decentralized, distributed hash table has drawn attention of research community. Several research has been done in the field.

Pastry[11] was developed by Antony Rowstron and Peter Druschel. It uses an 128 bit ID space. Each Pastry node is randomly issued an uniform node ID. On top of the DHT, a routing overlay network is designed for measuring the scalability and fault tolerance to reduce the routing cost. Routing overlay network collects information to build several lists, such as leaf nodes list, routing table and a neighbor list. In leaf nodes list, closest nodes are stored together in terms of node ID and direction of the circle. At least two applications are based on Pastry, which are PAST and SCRIBE[12, 13].

Chord was proposed in 2001 by Ion Stoica and his team in 2001 **??**. Structured like Pastry, Chord organizes the ID-key pair on a circle by hashing IP address and keyword of divided information separately. The key is stored in the closest next existed node to avoid issuing key to non-existent nodes. Then method of dichotomic classification was used in routing. Comparing to Pastry, Chord is a less complicated DHT solution.

Content addressable network(CAN)[14] is another well known DHT. “A virtual multi-dimensional Cartesian coordinate space" is the core of design. The virtual multi-dimensional coordinate space can be deemed as virtual address in network layer, independently from all under layers. The routing table consists of IP address and virtual zone pairs. One characteristic of CAN is that the maintaining of state is independent with the network size.

Kademlia[15] was designed in 2002. Kademlia defines distance as the bitwise exclusive of their node IDs. A lot of applications are built on Kademlia, such as KAD, Overnet, BitTorrent[16], Osiris sps, etc. In this thesis, I will mainly focus on Kademlia based P2P botnets. More details about Kademlia will be discussed in Chapter 2.

DHT is a method to map values to nodes in a distributed system. It's widely used in most of the P2P networks today.

1.1.3 Sybil attack

Though there are a lot of advantages for P2P model, it also has some weaknesses. One is that nodes join the network freely and every node stores partial routing table, so nodes leaving has influence on the P2P network. Though every decentralized P2P protocol tries to optimize nodes joining and leaving, multiple leaving is still a potential threat to P2P network. Then, if someone creates large quantities of nodes and has those nodes join P2P network, he may gain a unbalanced high influence, which is known as Sybil attack.

In 2002, John R. Douceur indicated decentralized network is susceptible in facing Sybil attack[17]. It is proved that any decentralized network which remote entities are not specified is vulnerable to Sybil attacks. Then this method is used to against P2P network which people share music or other copyrighted documents and it is proved to be useful to some extend.

1.2 Botnet

A bot, or known as zombie computer, is a personal computer or any online device which is infected unnoticed and controlled by hackers. Botnet is such a network which consists of bots. From its emergence in 1993, botnet has been developed a lot in the past three decades and it became a big threat to the Internet community. Its attacks include(not limited to) DDoS, adware, spyware, spamdexing, click fraud and even stealing confidential information such as driver's license and credit card number[18]. Generally, botnet is categorized into two main classes, centralized botnet and decentralized botnet.

1.2.1 Centralized botnet

The first generation of botnet are Internet Relay Chat(IRC) based network, which is a centralized botnet. The botmaster, as the operator of botnet, sets up an IRC channel to publish command. This IRC server is called command and control(C&C) server. Botmaster uses the botnet to do DDoS attacks, sending spam emails, etc. Some examples are listed below:

EggDrop: [19] is believed to be the first botnet ever, and EggDrop is also a non-malicious botnet. It was designed for managing and protecting an IRC channel.

Agobot: Cooke05thezombie also known as Gaobot, was developed mainly with C++. Agobot is an user friendly software, which means using Agobot requires little or no programming background. It applies features such as harvesting email addresses, spam, DDoS attacks and Click Fraud, etc.

Akbot: is an botnet consisted of over 1 million computers. It is used for gathering data, performing DDoS attacks. The owner of Akbot was caught but released without conviction.

Zeus: uses a Trojan horse as its infection method. Zeus is believed divided into small botnets and sold to individuals. It has been reported of stealing personal data, spam email and stealing credit card information from banks.

Besides IRC based botnet, another centralized botnet is called social network based botnet, that is, their Command and Control process is done through social networks, such as fackook, twitter and some other familiar websites[20].

Asprox botnet[21] was reported in 2008, which is a botnet using its own server and advanced fast-flux network as C&C procedure. It shares the most of important design features as other centralized botnets.

There is a common weakness for all centralized botnets, which is once the channel has been shut, the whole botnet is dead, though the virus is still on victims' computers. Then, those hackers turn to find a more robust communication module for their botnet. Then P2P based botnet comes out.

1.2.2 Decentralized botnet

Hackers are searching for a robust, anonymous and scalable network protocol for their botnet. P2P model seems to meet all the demands. As a result, not only new developed botnets but also traditional botnets tend to use P2P model as its whole or partial communication module. The function of centralized and decentralized botnet are almost the same. Table 1.1 is a time table of P2P model and botnets.

The P2P based botnet is more cryptical and resilient, which makes it harder to be detected or measured. Following are some detected examples of decentralized (P2P based) botnets:

TDL-4: [22] which was find by Kaspersky Lab in 2008. TDL-4 uses a identifier called "bsh parameter" that plays an important rule in its connecting procedure. Moreover, the TDL-4 botnet also uses KAD network to publish commands.

Zeus: [23, 24] Zeus botnet starts as a centralized botnet and recruits mainly by phishing and drive-by downloads. Zeus was used to attack or steal from a lot of famous companies, organizations and even government departments, including Bank of

Date	Name	type	Description
05/1999	Napster	P2P	P2P protocol was used for the first time
11/1999	Direct Connect	P2P	developed Napster model
03/2000	Gnutella	P2P	fully decentralized P2P model
09/2000	eDonkey	P2P	used Multisource File Transfer Protocol
03/2001	Fast Track	P2P	supernodes are used within P2P protocol
07/2001	BitTorrent	P2P	supernodes are used within P2P protocol
09/2003	Sinit	Botnet	random scanning look-up
11/2003	Kademlia	P2P	XOR matrix based P2P protocol
03/2004	Phatbot	Botnet	WASTE based botnet
03/2006	SpamThru	Botnet	a custom backup protocol used
01/2007	Peacomm	Botnet	based on Kademlia

Table 1.1: Time table of P2P networks and botnets

America, ABC, Cisco, Amazon, NASA and even United States Department of Transportation. In October 2011, it is reported by abuse.ch that the new vision of Zeus is to take a Kademlia-like strategy in its communication module [24].

Waledac: [25] Waledac is a typical e-mail spam botnet, and was taken down by Microsoft in March 2010. Microsoft won a court that taking the ownership of 276 domains which are believed using by Waledac as server [26]. However, Waledac also uses an unknown P2P protocol as backup, so the botmaster might not lose the control to Waledac.

Storm: [27] Storm attracted public attention in earlier 2007, when 8% of all malware which is on Windows computers are occupied by Storm. Later, Microsoft claims that the Malicious software Removal Tool(MSRT) has removed storm from more than $526,000$ personal computers [28, 29]. However, some research do not agree with that result. They believe that it is a botmaster's choice to have a smaller storm botnet [30]. In a word, Storm botnet is a good example of P2P botnet, which attracts a lot of researchers.

Several case studies have been done in the field[19, 31, 32]. Generally, it takes two steps to recruit new bots, *a*) use Trojan horse to infect initial binary. After this binary is installed, the injected bot has the basic functions, such as maintain persistence and join the P2P botnet. *b*) Once it joined P2P botnet, secondary injections will be downloaded to make the bot fully functional. Step a) is done by Trojan email as well as pornography website. The Trojan horse appearances as a video file but in fact it is an executable file. Even more, some Trojan horse disables windows firewall in order to make step b) execute without any warning.

1.3 Related work

Research on botnet are mainly in three categories, which are *a*) Botnet detection; *b*) reverse engineering on specified botnet; *c*) modeling and analysis of botnet.

There are several detection techniques[33, 34, 35]. In [36], Feily categories those techniques as signature-based, anomaly-based, DNS-based, and mining-based. Another research uses method to classify as honeynets based detection and Intrusion Detection System (IDS) based detection[37]. Khan uses data mining technique to detect the traffic of botnet [38]. An ensemble classification approach was proposed to deal with concept-drift and using all historical data. Ping and her team shows that present honeypot technique may be detected by botmaster by checking if bot can send out malicious traffic successfully [39].

Reverse engineering provides details of a single bot. Through which, it helps in understanding the behavior of botnet. Julian analyzed Trojan.Peacomm with PerilEyez malware tool and honeypot[19]. During the two weeks experiment, she researched the two steps of infection and the communication protocol. The Overnet protocol was detected in her case study. Thorsten Holz tracked Storm Worm and gave several measurement results [31]. During over four months detection, the result shows Storm infected machines in Overnet lays lower bound being around $5,000 - 6,000$ bots and around $45,000 - 80,000$ bots for upper bound. In treating Storm as a black box system, Holz indicates it is vulnerable to Sybil attack. The difference from our research is that we analyze the structure of kademlia(which Storm is based on) instead of treating it as black box.

Brett Stone-Gross and his team analyzed botnet takeover. In their research [40], a botnet called Torpig has been taken for ten days. Within those ten days, more than $180,000$ infections were observed. It gives detailed information about how botnet operates. Moreover, the analysis of the decrypted data, it shows that Torpig has a wide targeted list, which includes PayPal, Poste Italiane, E-Trade, Capital One and Chase, as well as several popular credit card companies.

Carlton R. Davis models Sybil attack on Storm using graph theory and birth and death process [41]. He also gives a simulation result shown the relationship between botnet growth rate and Sybil birth rate. Like Holz's work, Carlton also treats Storm as a blackbox. Ping Wang did a systematic study on P2P botnet[42]. Her systematic study fully describes all important characteristic of Storm. In her research, she does not analyze structure neither but she uses some results of [43]. In [43], the hop cost of Kademlia is discussed. The author focuses on k-bucket and results in average bits closer to target node. The difference between our research and theirs is that we give more details about Kademlia structure by using steps distribution.

1.4 Organization of the thesis

The rest of the thesis is organized as following:

- Chapter 2 presents mathematic analysis about structure of kademlia. Firstly, more details about kademlia algorithm and Storm is given. Secondly, the system is analyzed with different parameter α one by one.

- Chapter 3 gives numerical results of possible Sybil attacks. With different parameters, the results show how the efficiency of Sybil attack varies. The results are also compared with the results in [42].

- In chapter 4, we will conclude our work and discuss some possible future work.

Chapter 2

Efficiency Analysis of Sybil attack on P2P Botnets

2.1 Introduction of Kademlia

Kademlia algorithm is used by several P2P network, such as Overnet, KAD network and BitTorrent.

In Kademlia, every node stores a <key, value> pair and a triple list of <IP address, UDP port, Node ID>. The value in the <key, value> pair is a piece of divided information saved in the network, whereas key is an identifier of where that piece of information is stored. In which, key is 160 bit opaque, hashed or partially hashed from value. A <key, value> is stored in the node where ID is closest to the key. The <IP address, UDP port, Node ID> triple list contains some contact information. Detailed information pertaining to triple listing will be presented later in the thesis. Each Kademlia node has a unique node ID and it is randomly chosen from ID space.

The ID space is a binary tree and the quantity of IDs varies from network to network, 160 bits in [15] and 128 bits in [44], etc. For any given node, the ID space has been divided into subtrees. Half of binary tree exclusive of the node itself, constitutes the highest subtree. Half of the rest tree, exclusdes the node itself, constitutes the second highest subtree, et cetera. Kedamlia protocol ensures that each node at least knows one node from each of its subtrees. Figure2.1 shows an example of node ID space and subtrees. Taking node 011 for instance, there are three subtrees for node 011 and it knows at least one node in each of its subtrees.

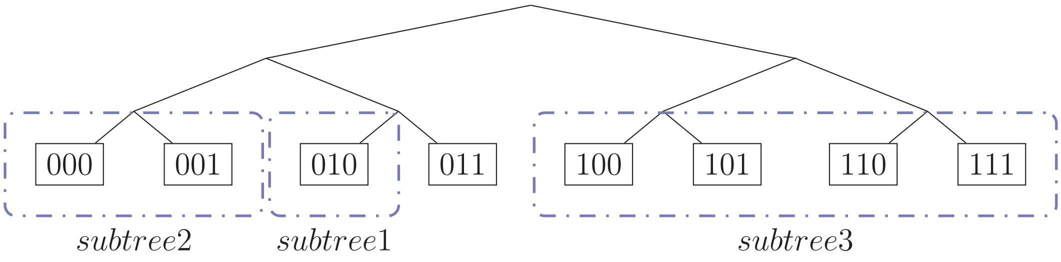


Figure 2.1: An example of nodes ID space and subtrees

In Kademlia, the distance between two nodes, node a and node b, is defined as the bitwise exclusive (XOR) of their nodes IDs, that is, $d(a, b) = a \oplus b$. For example, the distance of node 110 and node 011 is $d(110, 011) = 110 \oplus 011 = 101$. To contact with other nodes, each node holds N lists of <IP address, UDP port, Node ID> triplets. For $0 \leq g < N$, a node keeps maximum k triples of other nodes whose distances to itself are from 2^g to 2^{g+1}. These lists are called k-buckets, where k is a system-wide configurable parameter, and generally $k = 20$[15]. If there are less than k nodes in a distance range $[2^g, 2^{g+1})$, all triples of nodes in that distance range are kept in the list; If there are more than k nodes in the distance range, only k triples are kept. Each k-bucket is recorded by the order of time last seen, most recently seen at the tail and the least recently seen at the head. When node (receiver) receives message from any other nodes (transmitter), the k-bucket is updated following some special rules. If transmitter's triple exists in receiver's k-bucket, the triple gets moved to the tail of the list. If the transmitter's triple does not exists in the related k-bucket, and that k-bucket is not full, the receiver adds transmitter's triple to the tail of the k-bucket. If the related k-bucket is fully filled, the receiver pings the node at the head of k-bucket for more decision parameters. If the node at the head of k-bucket responds, this node gets to move to the tail and the discards the transmitter's triple. Otherwise, the node at head of the k-bucket is deleted from the list, and the transmitter's triple is added to the tail of k-bucket. This procedure ensures the most active nodes' triples are kept. The flowchart of the progress is shown in figure2.2.

There are four remote procedure calls (RPCs) in Kademlia protocol as following:

PING: to determine if a node is online;

STORE: to require a node to store a <key, value> pair;

FIND_NODE: to find a specified node. If the receiver is not the target node, it returns k triples of closest-to-target nodes it knows;

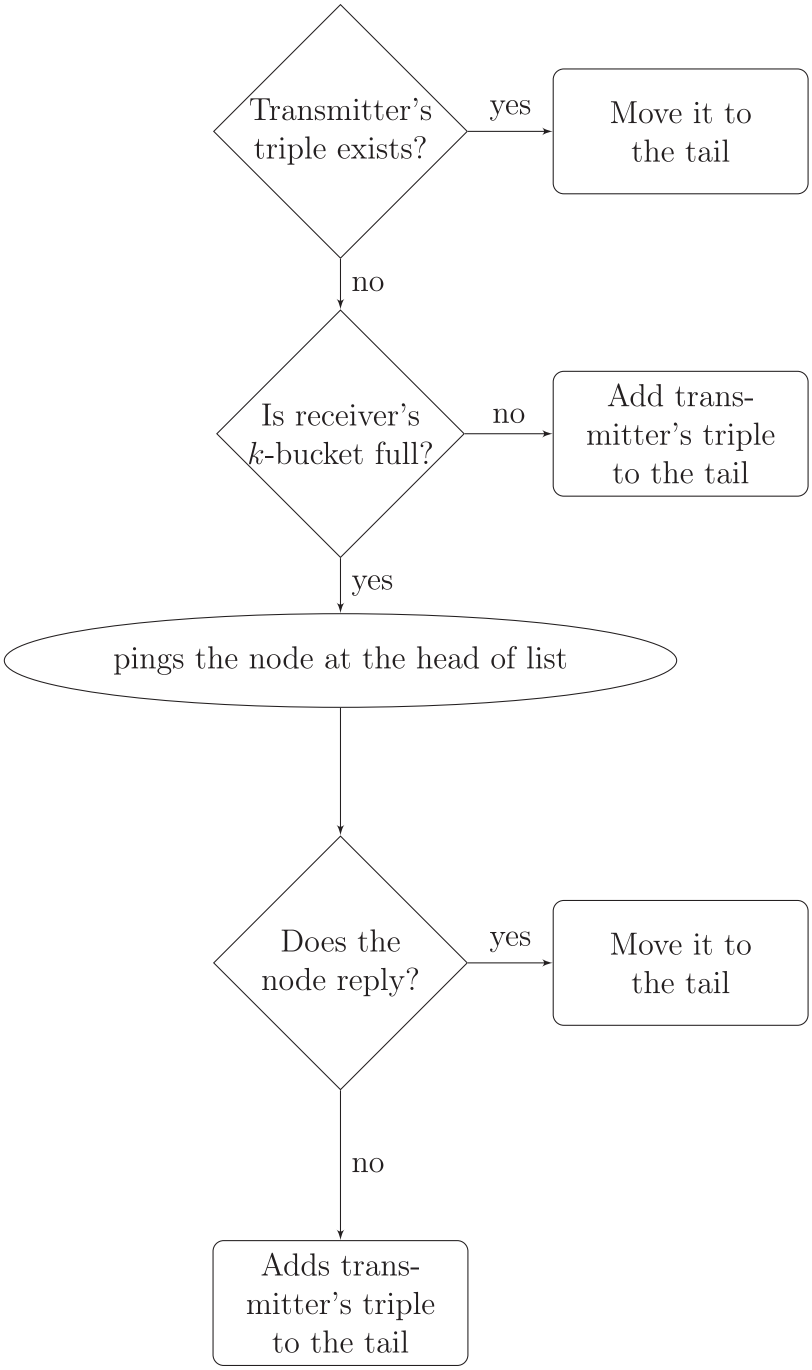

Transmitter's triple exists?
yes
Move it to the tail
no
Is receiver's k-bucket full?
no
Add transmitter's triple to the tail
yes
pings the node at the head of list
Does the node reply?
yes
Move it to the tail
no
Adds transmitter's triple to the tail

Figure 2.2: The criteria progress of updating k-bucket

FIND_VALUE: to find a value stored in some specified nodes. If a node who does not have that value receives this RPC, it returns k triplets of nodes which most likely has that value. If a node who has that value received this RPC, it returns the value.

Lookup is the most important procedure in Kademlia protocol. Lookup procedure is the way a node (initiator) locating k closest nodes to a target node ID. It starts with querying α nodes from the k-bucket list which are closest to target node. Each node of those α nodes returns k closest nodes to the target node in it's k-buckets. Then the initiator chooses and queries new α closest nodes which it has never queried before in the returning list. By repeating described steps, initiator will collect k closest nodes to the target. As k, α is a system-wide parameter.

2.2 Analysis of Sybil attack on botnet

As shown in Holz's research [31], it is possible to catch the encrypt keys, which bring Sybil attack on botnet into reality. However, more detail should be discussed on network level behaviors. Thus our research focus on network module of the botnet. By analysis of the Kademlia strategy, we want to find out how efficient Sybil attack is.

To build a model, several terms are defined as:

- Node I is the initiator node who plans to do lookup. I is its node ID.

- Node T is the target node I is looking for. T is its node ID.
- Define the node space size as 2^{i+1}.
- d is the distance of two nodes and $d_{IT} = I \oplus T$.
- Define tree level distance(TLD) as $D_{IT} = g + 1$ when $d_{IT} \in [2^g, 2^{g+1})$.
- k as in k-bucket, is set to 20.
- $Pr\{x\}$ is the probability that event x occurs.

Besides distance defined by Petar in [15], tree level distance(TLD) is defined in this research. That is because the nodes are deemed equally in a specified subtree, and in which subtree the target node lays, is more important in this research. Then, several assumptions have been made for our model as following:

1. The node ID space is full, thus, all node IDs have been taken.
2. We analyzes two types of sybil attacks.
    - In **random sybil attack**, when receives FIND_NODE or FIND_VALUE, sybil nodes reply with a random fake triples of nodes.
    - In **fake target attack**, when receives FIND_NODE or FIND_VALUE, sybil nodes reply with a fake triple of target node.

3. Initial node (node I) ID starts with 0 and target node (node T) ID starts with 1. Which means, node I and node T are in different halves of binary tree.

When node I starts to locate node T, it first checks the XOR result of their IDs. From our assumption 3, the prefix of I is 0, and the prefix of T is 1, so the prefix of XOR is 1, then I knows T is in the other half of the binary tree, $[2^i, 2^{i+1})$. In that subtree, within those 2^i nodes, node I knows 20 nodes from its k-bucket(as k has been set to 20), which are $B_1, B_2 \dots B_j \dots B_{20}$.

Probability that node T is one of B_j is given by P_k.

$$P_k = \begin{cases} \frac{20}{2^i} & \text{if } i \geq 5, \\ \\ 1 & \text{if } i < 5. \end{cases} \tag{2.1}$$

If $i = 4$ or less, there is only 16 or less nodes in that k-bucket, so node I knows all nodes in that k-bucket, thus the target node must be in the related bucket. So P_k is 1 when i is 4 or less.

With probability $1 - P_k$, node I queries α closest nodes to node T. The α as introduced before, is a system-wide parameter. Three cases, $\alpha = 1$, $\alpha = 2$ and $\alpha = 3$ will be discussed separately.

2.2.1 When $\alpha = 1$

To find node T, node I sends FIND_NODE to 1 closest node in B_i. Before this action, the tree level distance between node T and node I is i. Let B_c to be the closest node

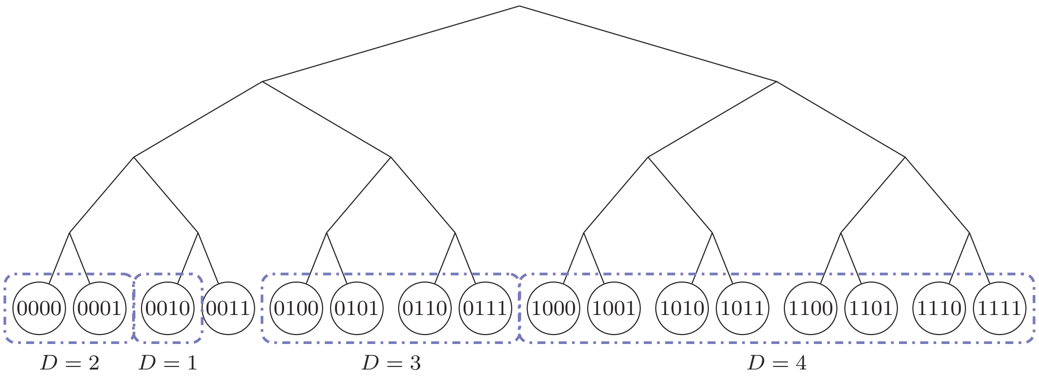


Figure 2.3: Distances form the view of node 0011

in I's k-bucket, and $B_c \in \{B_j\}$. In order to find the probability distribution of the distance between node B_c and node T, we let m be a positive integer in the range of $0 \leq m \leq i$. The probability of that distance being lager than m is calculated first:

$$\begin{aligned} Pr\{D_{B_cT} \geq m\} &= Pr\{D_{B_1T} \geq m, D_{B_2T} \geq m \ldots D_{B_{20}T} \geq m\} \\ &= Pr\{\text{all } D_{B_jT} \geq m\} \qquad j = 1, 2 \ldots 20. \end{aligned} \tag{2.2}$$

Because D_{B_cT} is a distance, it must be a nonnegative integer:

$$Pr\{D_{B_cT} \geq 0\} = 1$$

The distance of two nods is the XOR of their node IDs, so the distance a node to itself is 0, and the tree level distance(TLD) is 0 too. Thus means, $D_{B_cT} = 0$ indicates that B_c is T. Figure 2.3 is a illustration of a subtree with 16 nodes in it. Here we take node 0011 for instance, the TLD has been marked in the figure. Because the distance from node 0000 and node 0001 to 0011 is given by $0000 \oplus 0011 = 0011 = 3$ and $0001 \oplus 0011 = 0010 = 2$, then from our definition of TLD, both TLD are 2. Thinking a subtree with 2^i nodes in it, D_{B_cT} is larger than 1 means that node B_c can be any nodes besides the target nodes, so it is:

$$
\begin{aligned}
Pr\{D_{B_cT} \geq 1\} &= \frac{\binom{2^i-1}{20}}{\binom{2^i}{20}} \\
Pr\{D_{B_cT} \geq 2\} &= \frac{\binom{2^i-2}{20}}{\binom{2^i}{20}} \\
Pr\{D_{B_cT} \geq 3\} &= \frac{\binom{2^i-4}{20}}{\binom{2^i}{20}} \\
&\vdots
\end{aligned}
$$

Using mathematical induction we have:

$$
Pr\{D_{B_cT} \geq m\} = \frac{\binom{2^i-2^{m-1}}{20}}{\binom{2^i}{20}} \qquad m = 1, 2 \ldots i-1. \tag{2.3}
$$

So probability mass function(pmf) can be given as:

$$
\begin{aligned}
Pr\{D_{B_cT} = m\} &= Pr\{D_{B_cT} \geq m\} - Pr\{D_{B_cT} \geq m+1\} \\
&= \frac{\binom{2^i-2^{m-1}}{20} - \binom{2^i-2^m}{20}}{\binom{2^i}{20}} \qquad m = 1, 2 \ldots i-1.
\end{aligned} \tag{2.4}
$$

When node I queries node B_c, node B_c returns $k = 20$ closest nodes $B'_1, B'_2, \ldots B'_{20}$. Among those 20 nodes, let B'_c to be the closet node to T. Following the same process, the probability of the new closest distance m' can be given as:

$$Pr\{D_{B_c'T} \geq m' \mid D_{B_cT} = m\} = \frac{\binom{2^m - 2^{m'-1}}{20}}{\binom{2^m}{20}} \qquad m' = 1, 2 \ldots m-1. \tag{2.5}$$

$$\begin{aligned} &Pr\{D_{B_c'T} = m' \mid D_{B_cT} = m\} \\ &= \frac{\binom{2^m - 2^{m'-1}}{20} - \binom{2^m - 2^{m'}}{20}}{\binom{2^m}{20}} \qquad m' = 1, 2 \ldots m-1 \end{aligned} \tag{2.6}$$

Every time node I queries and checks the closest node, it gets closer to node T. Equation (2.6) gives the probability of closest distance jumping from m to m' when node I queries and checks once. Let $A(j,k)$ to be the distance jumping probability of one query and check, where j is the TLD before jumping and k is after jumping. So $A(j,k) = Pr\{D_{B_c'T} = k \mid D_{B_cT} = j\}$ Then a jumping matrix is given as:

$$\widetilde{A} = \begin{bmatrix} A(1,1) & 0 & \cdots & 0 & 0 \\ A(2,1) & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A(i,1) & A(i,2) & \cdots & A(i,i-1) & 0 \end{bmatrix} \tag{2.7}$$

In matrix $\widetilde{A}$, TLD gets closer in every query and check, thus a jumping that results in a same or greater TLD is impossible, so only the those ones under the diagonal are non-zero elements. Besides , there are some special cases. Element $A(2,1), A(3,1), A(3,2), A(4,1), A(4,2), A(4,3)$ are all zero, that is because when distance is less than or equal to 4, the queried node returns all 16 nodes it knows to node

I, and the target node must be in that 16 nodes. Thus means, the distance jumps to 0 directly. The other special case is element $A(5,4)$. For only 16 nodes within the TLD 4 and k is set to 20, it makes $A(5,4)$ an impossible jump.

Also, from equation 2.1 the probability that B'_c is T is given as:

$$Pr\{D_{B'_cT} = 0 \mid D_{B_cT} = m\} = \begin{cases} \frac{20}{2^m} & \text{if } m \geq 5, \\ \\ 1 & \text{if } m < 5. \end{cases} \tag{2.8}$$

So equation (2.8) gives the jumping function of any TLD to 0. Let j to be the TLD before jumping and $B(j,0)$ to be the probability of jumping from j to 0 after one query and check, so $B(j,0)$ is given by equation 2.8. Then $\widetilde{B}$ is a matrix of $B(j,0)$, which is given as:

$$\widetilde{B} = \begin{bmatrix} B(1,0) \\ \\ B(2,0) \\ \vdots \\ B(i,0) \end{bmatrix} \tag{2.9}$$

By now, matrix $\widetilde{A}$ and $\widetilde{B}$ give all the possibility of jumping with one query and check. To find the distribution of the number of steps needed to reach target node from initial node, matrix $\widetilde{C}$ is introduced. Let $C(j,l)$ be the probability that when tree level distance is j, it needs l steps to reach T. Matrix $\widetilde{C}$ can be calculated from these two following equations:

$$C(j,1) = B(j,0) \qquad j = 1,2\ldots i \tag{2.10}$$

$$C(j,l) = \sum_{x=1}^{j-1} A(j,x) \times C(x,l-1) \qquad l = 1,2\ldots j-1;\, j = 1,2\ldots i \tag{2.11}$$

Then the matrix $\widetilde{C}$ is:

$$\widetilde{C} = \begin{bmatrix} C(1,1) & 0 & \cdots & 0 & 0 \\ C(2,1) & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C(i,1) & C(i,2) & \cdots & C(i,i-1) & 0 \end{bmatrix} \tag{2.12}$$

The last row of $\widetilde{C}$ gives the probability distribution of the number of steps node I will be taking to reach node T. Assume P_{sybil} is the percentage of sybil nodes within the network, and $P_{success}$ is the probability of node I finding T successfully.

$$P_{success} = \sum_{x=1}^{i-1} C(i,x) \times (1 - P_{sybil})^x \tag{2.13}$$

The equation (2.13) is suitable for both attack model. Because of querying only one node in every step, even one sybil node during the querying results in lookup fail. With equation 2.13, it is possible to get a numerical result. In figure2.4, X axis is the percentage of Sybil nodes, and Y axis is the probability of target node found successfully. The total number of nodes is set to 2^{128}. The network shows vulnerability in facing Sybil attack. With 1% of Sybil nodes, $P_{success}$ drops to around 80%, And more, $P_{success}$ fall down shapely to only 10% when there are 10% of Sybil nodes.

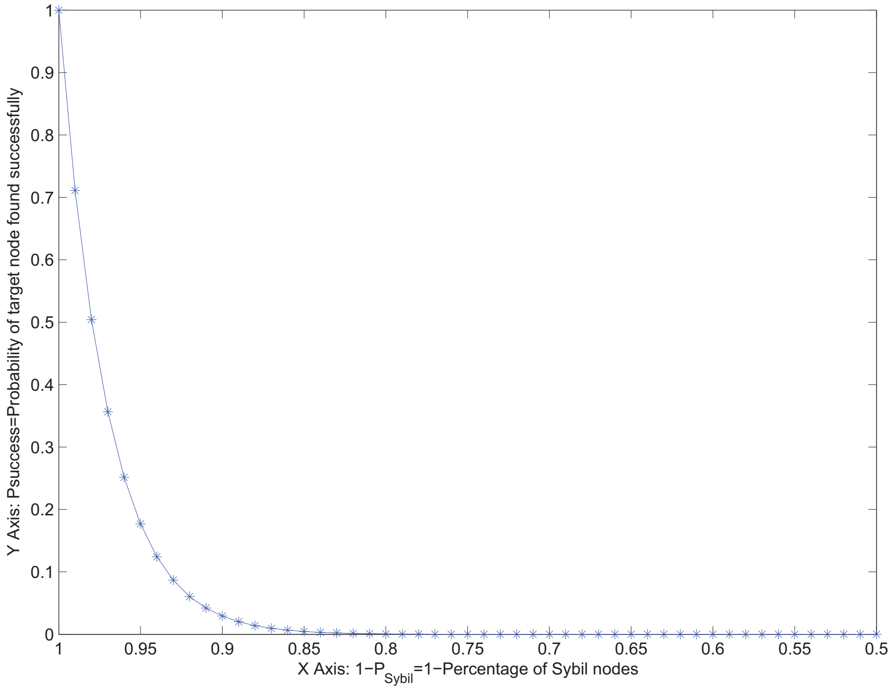


Figure 2.4: The distribution of $P_{success}$ when $\alpha = 1$

2.2.2 When $\alpha = 2$

When $\alpha = 2$, the querying and checking progress is similar to $\alpha = 1$. The only difference is that node I queries 2 nodes every time instead of 1. Let assume B_s is the second closest node in I's k-bucket, so $B_s \in \{B_j\}$. Let positive integer m_s be in the range of $0 \leq m_s \leq m \leq i$. Similar to the case of $\alpha = 1$, to find $Pr\{D_{B_sT}\} = m_s$,

$Pr\{D_{B_sT}\} \geq m_s$ is calculated first. Because the closest node B_c has been taken, that TLD between the second closest node B_s and target node is equal to or greater than m_s means the TLD between all other 19 nodes and the target node are equal to or greater than m_s, which is:

$$\begin{aligned} & Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = m\} \\ & \quad = Pr\{\text{all } D_{B_jT} \geq m_s \mid D_{B_cT} = m\} \qquad j = 1, 2 \ldots 20 \quad B_j \neq B_c. \end{aligned} \tag{2.14}$$

Also, TLD is a nonnegative integer, so we have:

$$Pr\{D_{B_sT} \geq 0\} = 1$$

Recalling the binary tree and figure 2.3, sample space is the total number of nodes i with the closest node B_c taken out. Because the probability distribution of the second closest node depends on the distribution of the closest node, different cases are discussed separately as follows. Firstly, when $D_{B_cT} = 0$, which means node B_c is the target node, the TLD of node B_s and T must be equal to or lager than 1, because the tree level distance of node B_c and T is 0. $D_{B_sT} \geq 2$ happens when all other 19 nodes (including B_s) are placed at where the TLD is larger than or equal to 2, this means, those 19 nodes can be chosen from all the places except 2 nodes, one is node T and the other is its closest neighbor. $D_{B_sT} \geq 3$ means those 19 nodes can be chosen from all nodes except 4 nodes, node T and other three nodes in the same smallest subtree with T, etc. Then we have:

$$\begin{aligned}
Pr\{D_{B_sT} \geq 1 \mid D_{B_cT} = 0\} &= \frac{\binom{2^i-1}{19}}{\binom{2^i-1}{19}} \\
Pr\{D_{B_sT} \geq 2 \mid D_{B_cT} = 0\} &= \frac{\binom{2^i-2}{19}}{\binom{2^i-1}{19}} \\
Pr\{D_{B_sT} \geq 3 \mid D_{B_cT} = 0\} &= \frac{\binom{2^i-2^4}{19}}{\binom{2^i-1}{19}} \\
&\vdots
\end{aligned}$$

From the above equations, it is concluded as:

$$Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = 0\} = \frac{\binom{2^i-2^{m_s-1}}{19}}{\binom{2^i-1}{19}} \qquad m_s = 1, 2 \ldots i \tag{2.15}$$

Secondly, when $D_{B_cT} = 1$, node B_c is not node T but its closest neighbor, so the sample space for choosing node B_s is $2^i - 2$. TLD of node B_s and node T must be equal to or greater than 2. Similar to the case of $D_{B_cT} = 0$, $D_{B_sT} \geq 3$ occurs when those 19 nodes are chosen besides 4 nodes, target node, node B_c and two closest neighbors, etc. It gives:

$$
\begin{aligned}
Pr\{D_{B_sT} \geq 0 \mid D_{B_cT} = 1\} =& 1 \\
Pr\{D_{B_sT} \geq 1 \mid D_{B_cT} = 1\} =& 1 \\
Pr\{D_{B_sT} \geq 2 \mid D_{B_cT} = 1\} =& \frac{\binom{2^i-2}{19}}{\binom{2^i-2}{19}} \\
Pr\{D_{B_sT} \geq 3 \mid D_{B_cT} = 1\} =& \frac{\binom{2^i-2^2}{19}}{\binom{2^i-2}{19}} \\
& \vdots
\end{aligned}
$$

And then we have:

$$
Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = 1\} = \frac{\binom{2^i-2^{m_s-1}}{19}}{\binom{2^i-2}{19}} \qquad m_s = 2, 3 \ldots i \tag{2.16}
$$

From equation (2.15) and (2.16), a more general equation can be written as:

$$
\begin{aligned}
& Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = m\} \\
&= \begin{cases} \frac{\binom{2^i-2^{(m_s-1)}}{19}}{\binom{2^i-2^m}{19}} & m_s \in (m, i],\, m = 0, 1 \\ 1 & m_s \in [0, m],\, m = 0, 1 \end{cases}
\end{aligned} \tag{2.17}
$$

Thirdly, when $D_{B_cT} = 2$, B_s can be placed at everywhere besides 3 places, the target node, the closest node to target node and node B_c. Then the sample space is $2^i - 3$. Taking the same method, it gives:

$$
\begin{aligned}
Pr\{D_{B_sT} \geq 2 \mid D_{B_cT} = 2\} =& 1 \\
Pr\{D_{B_sT} \geq 3 \mid D_{B_cT} = 2\} =& \frac{\binom{2^i-2^2}{19}}{\binom{2^i-3}{19}} \\
Pr\{D_{B_sT} \geq 4 \mid D_{B_cT} = 2\} =& \frac{\binom{2^i-2^3}{19}}{\binom{2^i-3}{19}} \\
Pr\{D_{B_sT} \geq 5 \mid D_{B_cT} = 2\} =& \frac{\binom{2^i-2^4}{19}}{\binom{2^i-3}{19}} \\
& \vdots
\end{aligned}
$$

Then it is summed as:

$$
Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = 2\} = \frac{\binom{2^i-2^{m_s-1}}{19}}{\binom{2^i-3}{19}} \qquad m_s = 3, 4 \ldots i \tag{2.18}
$$

Fourthly, when $D_{B_cT} = 3$, the sample space is $2^i - 2^{(3-1)} - 1$. It is because B_s can not be placed where the distance is less than 3 and B_c has taken one place at the subtree of distance 3. So we have:

$$
\begin{aligned}
Pr\{D_{B_sT} \geq 3 \mid D_{B_cT} = 3\} =& 1 \\
Pr\{D_{B_sT} \geq 4 \mid D_{B_cT} = 3\} =& \frac{\binom{2^i-2^3}{19}}{\binom{2^i-2^{(3-1)}-1}{19}} \\
Pr\{D_{B_sT} \geq 5 \mid D_{B_cT} = 3\} =& \frac{\binom{2^i-2^4}{19}}{\binom{2^i-2^{(3-1)}-1}{19}} \\
Pr\{D_{B_sT} \geq 6 \mid D_{B_cT} = 3\} =& \frac{\binom{2^i-2^5}{19}}{\binom{2^i-2^{(3-1)}-1}{19}} \\
&\vdots
\end{aligned}
$$

To sum up the above equations, we have:

$$Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = 3\} = \frac{\binom{2^i-2^{m_s-1}}{19}}{\binom{2^i-2^{(3-1)}-1}{19}} \qquad m_s = 3, 4 \ldots i \tag{2.19}$$

Cases when $m = 4, 5, 6 \cdots$ are similar to $m = 2$ and $m = 3$, that means node B_c always occupies one place in subtree of TLD m. Then from equation 2.18 and 2.19, we have:

$$
\begin{aligned}
&Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = m\} \\
&= \begin{cases} \dfrac{\binom{2^i-2^{(m_s-1)}}{19}}{\binom{2^i-2^{(m-1)}-1}{19}} & m_s \in (m, i],\ m = 2, 3, 4 \cdots i \\ 1 & m_s \in [0, m],\ m = 2, 3, 4 \cdots i \end{cases}
\end{aligned}
\tag{2.20}
$$

From equation (2.17) and (2.20), it is possible to calculate the probability of $Pr\{D_{B_sT} = m_s \mid D_{B_cT} = m\}$, simply minus $Pr\{D_{B_sT} \geq m_s \mid D_{B_cT} = m\}$ by

$Pr\{D_{B_sT} \geq m_s + 1 \mid D_{B_cT} = m\}$. However, this method is only suitable for $m_s \in (m, i-1]$ and there are some exceptions. *a*) when $m_s = i$ and $m_s > m$, node T is in the highest subtree of node B_s, so $Pr\{D_{B_sT} = m_s \mid D_{B_cT} = m\}$ is equal to $\binom{2^i}{19} / \binom{2^i - 2^m}{19}$ when $m = 0, 1$, and $\binom{2^i}{19} / \binom{2^i - 2^{(m-1)} - 1}{19}$ when $m = 2, 3, 4 \cdots i-1$. *b*) when $m_s = m$ and $m_s \in [2, i-1]$, $Pr\{D_{B_sT} = m_s \mid D_{B_cT} = m\}$ equals to $1 - Pr\{D_{B_sT} \geq m + 1 \mid D_{B_cT} = m\}$. *c*) that two nodes are both 0 or 1 away from node T is impossible. *d*) when $m = i$ occurs, m_s must be equal to m. *e*) due to our assumption, $m_s < m$ is impossible.After all the conditions are discussed, $Pr\{D_{B_sT} = m_s \mid D_{B_cT} = m\}$ is given by:

$$
\begin{aligned}
&Pr\{D_{B_sT} = m_s \mid D_{B_cT} = m\} \\
&= \begin{cases}
\dfrac{\binom{2^i-2^{m_s-1}}{19} - \binom{2^i-2^{m_s}}{19}}{\binom{2^i-2^m}{19}} & m_s \in (m, i-1\,],\ m = 0, 1 \\[2ex]
\dfrac{\binom{2^i-2^{m_s-1}}{19} - \binom{2^i-2^{m_s}}{19}}{\binom{2^i-2^{(m-1)}-1}{19}} & m_s \in (m, i-1\,],\ m = 2, 3, 4 \cdots i-1 \\[2ex]
\dfrac{\binom{2^i}{19}}{\binom{2^i-2^m}{19}} & m_s = i \text{ and } m_s > m,\ m = 0, 1 \\[2ex]
\dfrac{\binom{2^i}{19}}{\binom{2^i-(m-1)-1}{19}} & m_s = i \text{ and } m_s > m,\ m = 2, 3, 4 \cdots i-1 \\[2ex]
\dfrac{\binom{2^i-2^{m_s}}{19}}{\binom{2^i-2^m}{19}} & m_s = i \text{ and } m_s = m,\ m = 2, 3, 4 \cdots i-1 \\[2ex]
0 & m_s = m \text{ and } m \in [0, 1] \\[2ex]
1 & m_s = m = i \\[2ex]
0 & m_s < m
\end{cases}
\end{aligned}
\tag{2.21}
$$

Equation (2.21) follows all the steps introduced in Kademlia protocol, so it gives the most accurate result. However, it is too complicate to do further derivation with Equation (2.21), so the model is simplified in this research. In the original Kademlia protocol, each node of those α queried nodes returns k closest nodes triples in its respective bucket. Then, node I send FIND_NODE to new closest α nodes among those nodes it received. In this research, after node I receives new information from α nodes, it find out the most closest one node ($TLD = m$) and we assume the second closest node is uniformly distributed in $[m, i-1)$. This simplification makes very small influence on final result, since the most important characteristic - query and check two node at one time - has been kept. To find the closest node in the returning information, the triplets returned form those nodes need to be compared. Let node B'_c and node B'_s to be the closest node in the related bucket of node B_c and node B_s respectively. Let m' to be the closest TLD returning from B'_c, whereas m'_s to be the closest TLD returning from B'_s. Then m' and node m'_s are given by equation (2.5). Let B_o to be the new closest node and m_o to be the new closest TLD after one query. Because each of node B'_c and node B'_s returns the triples independently, probability $Pr\{D_{B_oT} \geq m_o\}$ can be find:

$$\begin{aligned}
&Pr\{D_{B_oT} \geq m_o \mid D_{B_cT} = m, D_{B_sT} = m_s\} \\
&\quad = Pr\{D_{B'_cT} \geq m_o \mid D_{B_cT} = m\} \times Pr\{D_{B'_sT} \geq m_o \mid D_{B_sT} = m_s\} \\
&\quad = \frac{\binom{2^m - 2^{m_o - 1}}{20}}{\binom{2^m}{20}} \times \frac{\binom{2^{m_s} - 2^{m_o - 1}}{20}}{\binom{2^{m_s}}{20}} \\
&\qquad m_o \in [1, m-1]
\end{aligned} \tag{2.22}$$

Following the same steps as when $\alpha = 1$, we have:

$$\begin{aligned}
&Pr\{D_{B_oT} = m_o \mid D_{B_cT} = m, D_{B_sT} = m_s\} \\
&\quad = Pr\{D_{B_oT} \geq m_o \mid D_{B_cT} = m, D_{B_sT} = m_s\} \\
&\qquad - Pr\{D_{B_oT} \geq m_o + 1 \mid D_{B_cT} = m, D_{B_sT} = m_s\} \\
&\quad = \frac{\binom{2^m - 2^{m_o - 1}}{20}\binom{2^{m_s} - 2^{m_o - 1}}{20} - \binom{2^m - 2^{m_o}}{20}\binom{2^{m_s} - 2^{m_o}}{20}}{\binom{2^m}{20}\binom{2^{m_s}}{20}} \\
&\qquad m_o \in [1, m-1]
\end{aligned} \tag{2.23}$$

So equation (2.23) is the jumping function for $\alpha = 2$. Different from when $\alpha = 2$, 2.23 gives the jumping form two closest nodes to the newer closest one node. Now thinking about a TLD pair (j, k), in which j is the closest TLD and k is the second closest TLD, after one query the newer closest TLD is o. Then let$A_2(j, k, o)$ to be the probability that after one step checking, a TLD pair (j, k) jumps to the newer closest TLD o, which is given by equation (2.23). So Matrix $\widetilde{A_2}$ is given by:

$$\widetilde{A_2} = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & A_2(j, k, o) & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{2.24}$$

The probability that a given TLD pair (m, m_s) jumps to zero is equal to the probability that either m or m_s jumps to zero i.e. either node B'_c or node B'_s is the target. From equation (2.8), we have:

$$\begin{aligned} & Pr\{D_{B_oT} = 0 \mid D_{B_cT} = m, D_{B_sT} = m_s\} \\ & \quad = Pr\{D_{B'_cT} = 0 \mid D_{B_cT} = m\} + Pr\{D_{B'_sT} = 0 \mid D_{B_sT} = m_s\} \\ & \qquad - Pr\{D_{B'_cT} = 0 \mid D_{B_cT} = m\} \times Pr\{D_{B'_sT} = 0 \mid D_{B_sT} = m_s\} \end{aligned} \tag{2.25}$$

$$\begin{aligned} & Pr\{D_{B_oT} = 0 \mid D_{B_cT} = m, D_{B_sT} = m_s\} \\ & = \begin{cases} \dfrac{20}{2^m} + \dfrac{20}{2^{m_s}} - \dfrac{20 \times 20}{2^m \times 2^{m_s}} & \text{if } m \geq 5 \\ 1 & \text{if } m < 5 \end{cases} \end{aligned} \tag{2.26}$$

Equation (2.26) gives the probability a TLD pair jumping to the newer closest TLD. Let $B_2(j,k,0)$ to be the probability that a TLD pair (j,k) jumps to 0, which is given by Equation (2.26). Then $\widetilde{B_2}$ is a matrix of $B_2(j,k,0)$, it is given as:

$$\widetilde{B_2} = \begin{bmatrix} B_2(1,1,0) & B_2(1,2,0) & \cdots & B_2(1,i,0) \\ 0 & B_2(2,2,0) & \cdots & B_2(2,i,0) \\ \cdots & \cdots & \vdots & \cdots \\ 0 & 0 & \cdots & B_2(i,i,0) \end{bmatrix} \tag{2.27}$$

Then, following the similar steps as $\alpha = 1$, matrix $\widetilde{C_2}$ can be derived. Let $C_2(j,k,l)$ to be the probability that a TLD pair (j,k) takes l steps to reach 0, a three-dimensional matrix $\widetilde{C_2}$ can be calculated by:

$$\begin{aligned} C_2(j,k,1) &= B_2(j,k,0) \\ C_2(j,k,l) &= \sum_{x=1}^{j-1}\sum_{y=x}^{j-1} A_2(j,k,x)C(x,y,\ l-1) \qquad j \le k \le i \end{aligned} \tag{2.28}$$

When $\alpha = 2$, the case is different for two types of attack. The random sybil attack and fake target attack will be discussed separately as following.

In Kademlia, the lookup terminates when the target node has been located or the initiator queried and received from k closest nodes [15]. So in random attack, if one of two queried nodes is sybil node and returns random fake nodes, the lookup procedure does not terminate and the other node is still functional. Thus, only if both nodes are sybil nodes, the lookup procedure fails, so we have:

$$P_{success} = \sum_{x=1}^{i-1} C_2(i,i,x) \times (1 - {P_{sybil}}^2)^x \tag{2.29}$$

Then a numerical result can be given. The total number of nodes has been set to 2^{80}.

In facing random sybil attack, with $\alpha = 2$, the network is stronger than when it is with $\alpha = 1$. It is shown in figure 2.5 that as the number of Sybil nodes increases, $P_{success}$ drops slowly before the percentage of Sybil nodes less than 15%. After there are more than 15% Sybil nodes in the network, $P_{success}$ decreases fast.

In fake target attack, the sybil nodes returns a fake target node, so it means the target node has been located for initiator. Then the $P_{success}$ is given as:

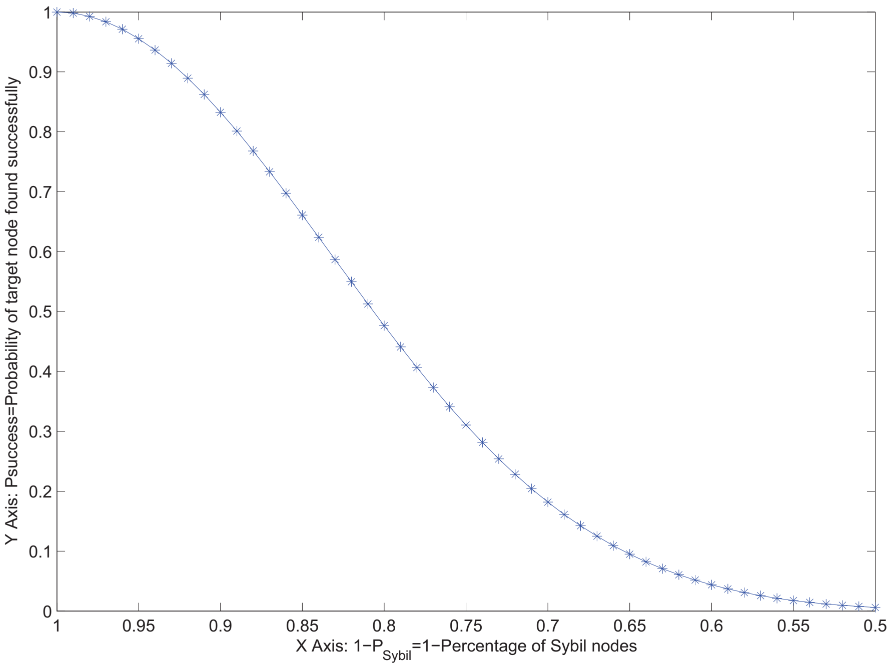


Figure 2.5: The distribution of $P_{success}$ when $\alpha = 2$ for random sybil attack

$$P_{success} = \sum_{x=1}^{i-1} C_2(i,i,x) \times (1 - P_{sybil})^{2x} \tag{2.30}$$

In facing fake target attack, with $\alpha = 2$ worsens the situation. From figure 2.6, $P_{success}$ drops sharply with increasing of sybil nodes, and the curve hits the bottom when there is only 15% sybil nodes in the network.

2.2.3 When $\alpha = 3$

When $\alpha = 3$, the procedure is the similar to $\alpha = 2$. Let node B_t to be the third closest node to node T and the TLD between node B_t and node T is m_t. The jumping

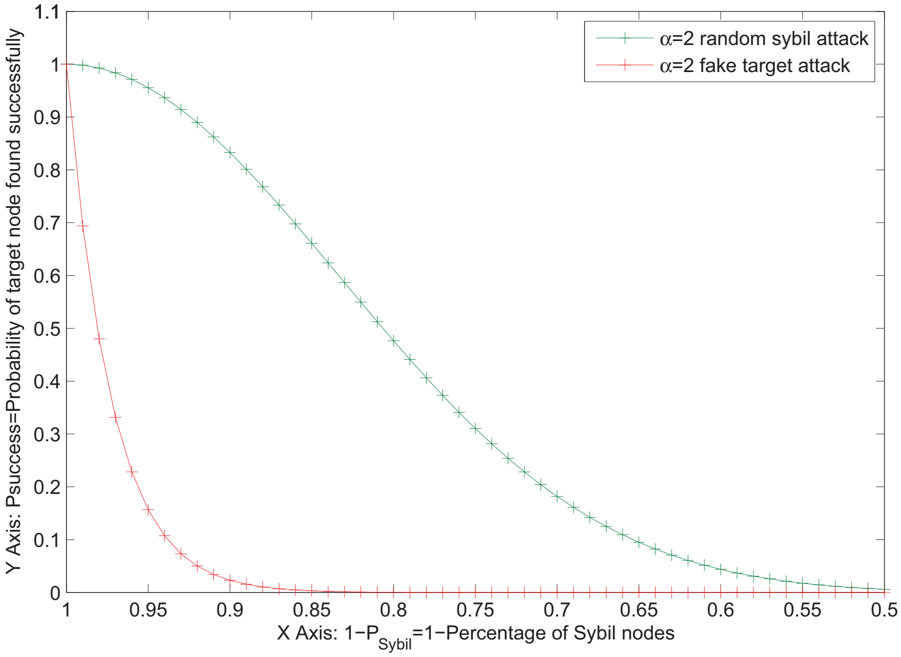


Figure 2.6: $P_{success}$ when $\alpha = 2$ for fake target attack

function is derived as:

$$\begin{aligned} &Pr\{D_{B_oT} \geq m_o \mid D_{B_cT} = m, D_{B_sT} = m_s, D_{B_tT} = m_t\} \\ &\quad = Pr\{D_{B_c'T} \geq m_o \mid D_{B_cT} = m\} \times Pr\{D_{B_s'T} \geq m_o \mid D_{B_sT} = m_s\} \\ &\qquad \times Pr\{D_{B_t'T} \geq m_o \mid D_{B_tT} = m_t\} \\ &\quad = \frac{\binom{2^m - 2^{m_o - 1}}{20}}{\binom{2^m}{20}} \times \frac{\binom{2^{m_s} - 2^{m_o - 1}}{20}}{\binom{2^{m_s}}{20}} \times \frac{\binom{2^{m_t} - 2^{m_o - 1}}{20}}{\binom{2^{m_t}}{20}} \\ &\qquad m_o \in [1, m-1] \end{aligned} \tag{2.31}$$

Then the pmf is possible to draw as:

$$
\begin{aligned}
&Pr\{D_{B_oT} = m_o \mid D_{B_cT} = m, D_{B_sT} = m_s, D_{B_tT} = m_t\} \\
&= \frac{\binom{2^m - 2^{m_o - 1}}{20}\binom{2^{m_s} - 2^{m_o - 1}}{20}\binom{2^{m_t} - 2^{m_o - 1}}{20} - \binom{2^m - 2^{m_o}}{20}\binom{2^{m_s} - 2^{m_o}}{20}\binom{2^{m_t} - 2^{m_o}}{20}}{\binom{2^m}{20}\binom{2^{m_s}}{20}\binom{2^{m_t}}{20}} \qquad (2.32) \\
&\qquad m_o \in [1, m-1]
\end{aligned}
$$

Here equation (2.32) gives the probability a TLD triplet of three closest nodes jumping to the newer closest TLD. Let(j, k, h) to be the a possible TLD triplet, o is the newer closest TLD, $A_3(j, k, h, o)$ is the probability that after one step checking, the TLD triplet jumps from (j, k, h) to the newer closest TLD o. Then $\widetilde{A_3}$ is the matrix of $A_3(j, k, h, o)$, which is given by equation (2.32):

$$
\widetilde{A_3} = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & A_3(j,k,h,o) & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \qquad (2.33)
$$

The probability that a TLD triplet jumps to 0 is given as:

$$
\begin{aligned}
&Pr\{D_{B_oT} = 0 \mid D_{B_cT} = m, D_{B_sT} = m_s, D_{B_tT} = m_t\} \\
&= \begin{cases} \dfrac{20}{2^m} + \dfrac{20}{2^{m_s}} + \dfrac{20}{2^{m_t}} - \dfrac{20 \times 20}{2^m \times 2^{m_s}} - \\[2ex] \dfrac{20 \times 20}{2^m \times 2^{m_t}} - \dfrac{20 \times 20}{2^{m_s} \times 2^{m_t}} + \dfrac{20 \times 20 \times 20}{2^m \times 2^{m_s} \times 2^{m_t}} & \text{if } m \geq 5 \\[2ex] 1 & \text{if } m < 5 \end{cases} \qquad (2.34)
\end{aligned}
$$

Then, following the same steps as tracking $\widetilde{A_3}$, the probability of TLD (j, k, h) jumping to 0 is given by equation 2.34. In which j, k, h are m, m_s, m_t respectively. Then Matrix $\widetilde{B_3}$ can be find as:

$$\widetilde{B_3} = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & B_3(j,k,h,0) & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{2.35}$$

Let $C_3(j, k, h, l)$ to be the probability that a TLD triplet (j, k, h) takes l steps to reach 0, then four-dimensional matrix $\widetilde{C_3}$ can be calculated by,

$$C_3(j,k,h,1) = B_3(j,k,h,0) \tag{2.36}$$

$$C_3(j,k,h,l) = \sum_{x=1}^{j-1}\sum_{y=x}^{j-1}\sum_{z=y}^{j-1} A_3(j,k,h,x) \times C(x,y,z,\ l-1) \quad j \leq k \leq h \leq i \tag{2.37}$$

Finally, $P_{success}$ for both attack can be derived. In random sybil attack, the probability of target nodes found successfully $P_{success}$ is given as:

$$P_{success} = \sum_{x=1}^{i-1} C_2(i,i,i,x) \times (1 - {P_{sybil}}^3)^x \tag{2.38}$$

In fake target attack, the the probability of target nodes found successfully is:

$$P_{success} = \sum_{x=1}^{i-1} C_2(i,i,i,x) \times (1 - P_{sybil})^{3x} \tag{2.39}$$

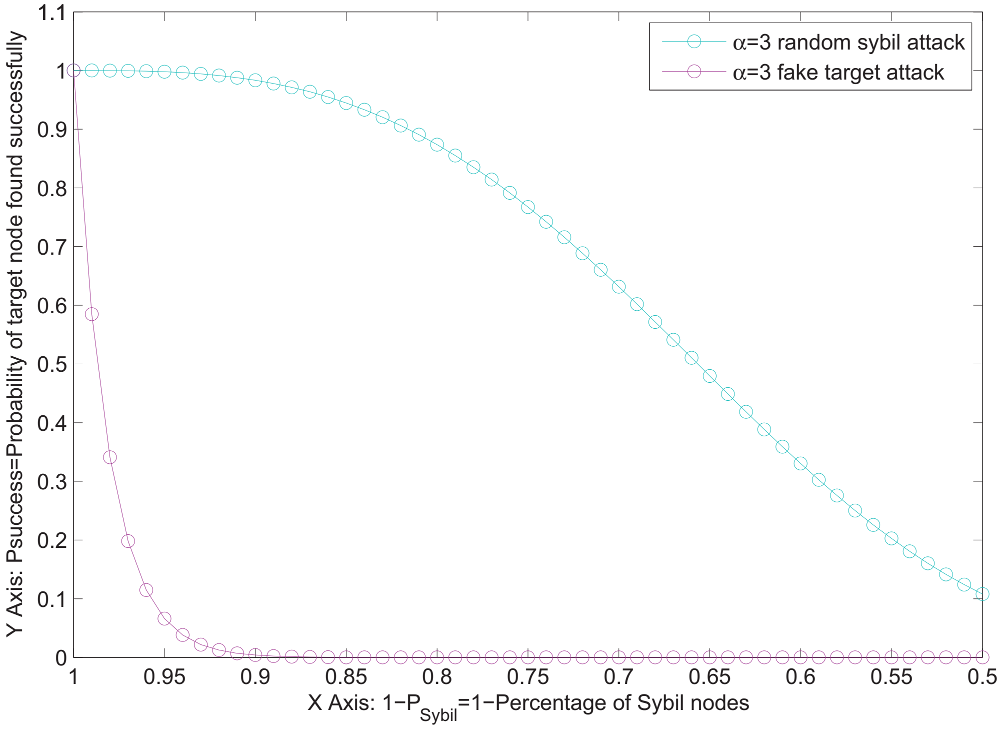


Figure 2.7: The distribution of $P_{success}$ when $\alpha = 3$

Figure 2.8: $P_{success}$ when $\alpha = 3$ for two attack model

When $\alpha = 3$, the network is pretty strong in facing random sybil attack. Form figure 2.8, to drop the $P_{success}$ to 80%, Sybil nodes must occupy 25% of total nodes. However, if the percentage of Sybil nodes keep increasing, 10% more Sybil node makes $P_{success}$ drop to less than 50%.

Just the opposite, when $\alpha = 3$, the network is more vulnerable in facing fake target attack. It shows in figure 2.8, only 5% of sybil nodes almost destroy the botnet. More results discussing will be introduced in next chapter.

Chapter 3

Numerical Results

From the model analyzed in Chapter 2, some numerical results are given in this chapter. In which, the look-up step distribution is shown first. Then the relationship of $P_{success}$ and three system parameters will be analyzed. Followed by a discussing of average steps to find the target node. This chapter ends up with comparison with other's work in the literature.

3.1 Step distribution

Different from other's work, step distribution has been taken into consideration in our work. Step distribution is the probability of the number of steps taken to find the target node. Through this distribution, structure characteristic of P2P based botnet is unfolded in the results. Figure 3.1 shows the step distribution when $\alpha = 1, 2, 3$ and the total nodes and parameter k has been set to 2^{40} and 20 respectively. From the figure, we see that lager α results in less steps, which also reduces the risk of attacking.

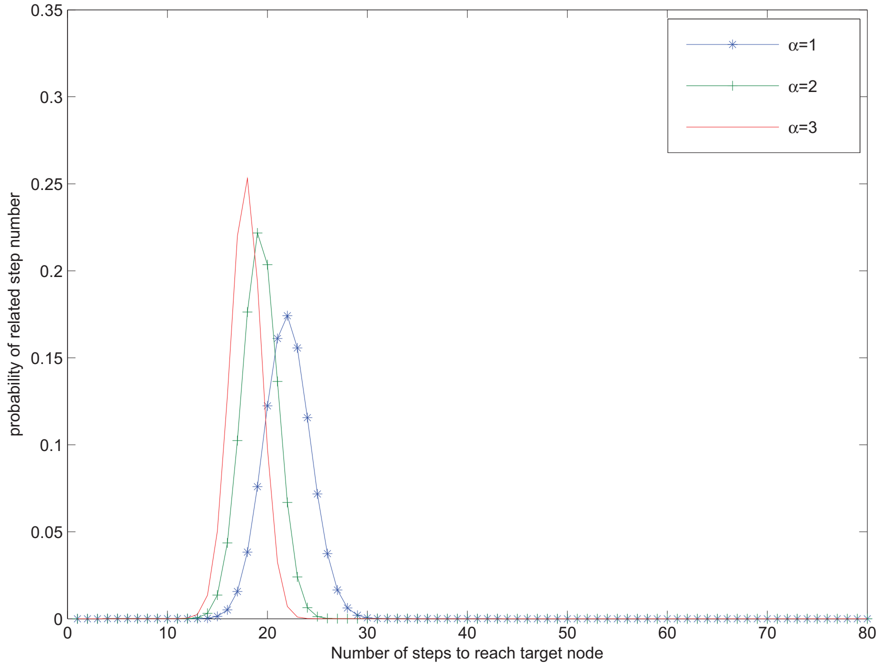


Figure 3.1: The distribution of the number of steps when $\alpha = 1, 2, 3$ separately.

3.2 $P_{success}$ and parameters

After analysis of the details about Kademlia, we want to figure out how efficient the Sybil attack is, and further more, which factor makes a P2P based botnet robust or vulnerable. These question will be answered through our analysis. During our research, we find that three parameters may affect $P_{success}$. Those three parameters are α (the number of nodes queried in every step), k (triple list a node kept in one bucket) and the total number of nodes n.

3.2.1 $P_{success}$ and α

Figure 3.2 shows $P_{success}$ when α is $1, 2$ or 3 for both attack model respectively. The total number of nodes are set to 2^{80} and k is equal to 20 for all α. Let's call them botnet A1, A2 and A3 when α is equal to $1, 2$ and 3 separately.

In facing random sybil attack, when there are 5% Sybil nodes in the botnet, $P_{success}$ of botnet A1 drops to less than 40%, whereas it of botnet A2 and A3 stay over 95%. When Sybil nodes occupy 10% of the network, botnet A1 has a $P_{success}$ only at 11.28%, and $P_{success}$ of botnoet A2 is 83.37% and starts to drop faster, but botnet A3 still has a $P_{success}$ over 98%. When the Sybil nodes is getting more than 10% for A2 and 20% for A3, $P_{success}$ decreases rapidly. $P_{success}$ reaches 20% when there are around 30% Sybil nodes for A1 and as many as 55% are needed for A3. A1, however, reaches zero when there are around 25% Sybil nodes.

The other way around, a larger α worsens the situation in facing fake target attack. Only 15% for $\alpha = 2$ and 10% for $\alpha = 3$ kills the botnet. This is because in every step, initiator queries α nodes and in fake target attack, and if any one of those α is sybil nodes, the lookup procedure terminates. So a lager α results in more chance of sybil nodes being choosing, which makes the botnet weaker.

From figure 3.2 and what has been discussed above, it is apparently that α is a key parameter in Kademlia based botnet. When $\alpha = 1$, it resembles Chord [45] network. Kademlia was designed to be more reliable in dealing with node fail, so α

was introduced. It is successfully in facing random sybil attack (which acts like massive node fail), but reacts badly in fake facing target attack.

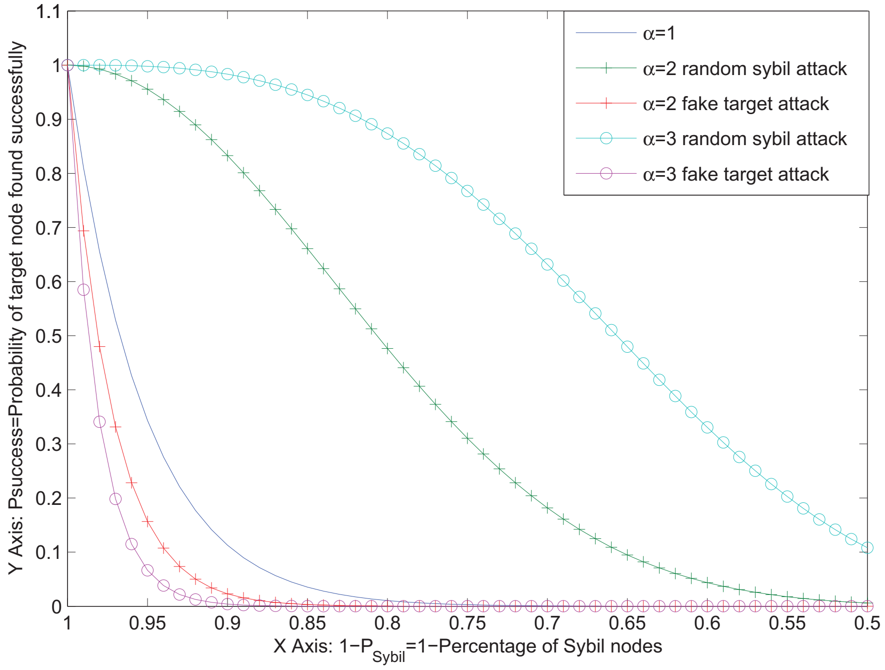


Figure 3.2: The distribution of $P_{success}$ when $\alpha = 1, 2, 3$ for two attack model separately.

3.2.2 $P_{success}$ and the total number of nodes

Besides α, total number of nodes n also has been discussed frequently. Figure 3.3, figure 3.4 and figure 3.5 shows relationship between $P_{success}$ and total number of nodes n. α is 1 in figure 3.3, 2 in figure 3.4 and 3 in figure 3.5. Parameter k equals to 20 in all three figures. All three figures show that the larger botnet is, the weaker it is. In figure 3.3, when 5% Sybil nodes are in the botnet, $P_{success}$ of botnet with $n = 2^{128}$

drops sharply to less than 20%, whereas it of botnet with $n = 2^{40}$ decreases to 60%. It also shows from figure 3.4 and figure 3.5 that as the increasing number of Sybil nodes, $P_{success}$ of botnet with lager n decreases faster than those with smaller n.

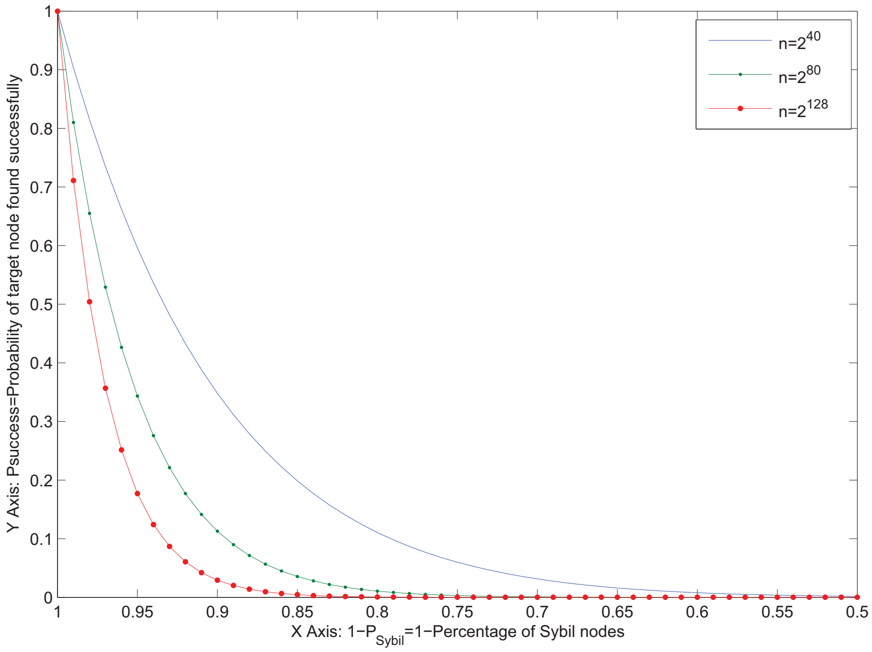


Figure 3.3: The distribution of $P_{success}$ when total number of nodes $n = 2^{40}, 2^{80}, 2^{128}$ separately. α has been set to 1 and k equals to 20.

In recent years, larger botnet has barely been detected. Other than improving of network security, more people believe that hackers choose to limit the size of their botnet. What has been discussed above may partially explain the reason. We only give the results for random sybil attack, but the results for fake target attack are similar.

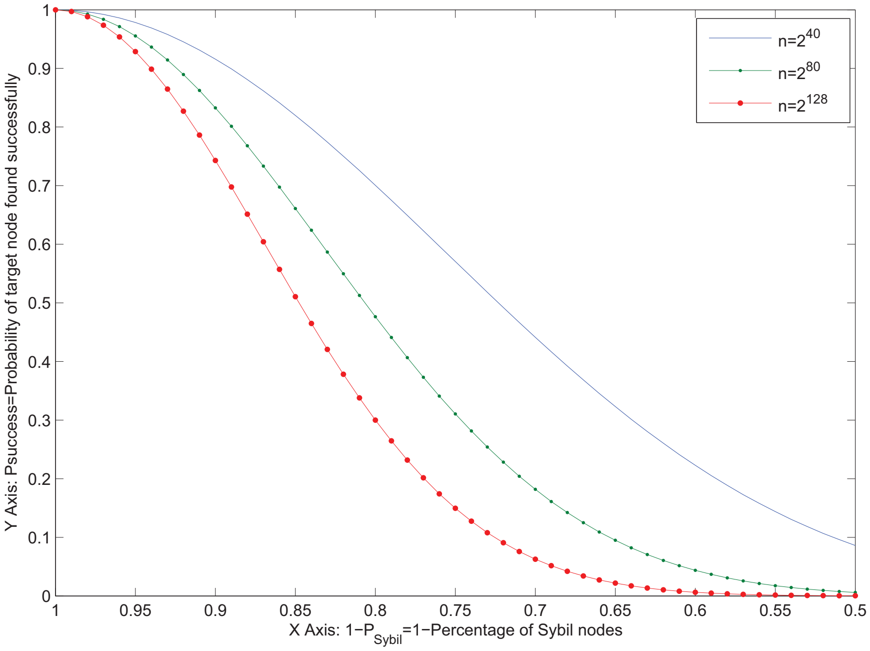


Figure 3.4: The distribution of $P_{success}$ when total number of nodes $n = 2^{40}, 2^{80}, 2^{128}$ separately. α has been set to 2 and k equals to 20.

3.2.3 $P_{success}$ and parameter k

k known form k-bucket, is the number of triples a node keeps for distance within every $[2^g, 2^{g+1})$. Figure 3.6 shows the relationship between distribution of $P_{success}$ and different k. It the figure, three pairs of α and k are showed together. Parameter n has been set to 2^{40}. All three pairs shows that the botnet with $k = 100$ is better than it with $k = 20$. Specially for $\alpha = 3$, Botnet with $k = 100$ holds 90% of success when 37% of total nodes are Sybil nodes, while the one with $k = 40$ holds the same percentage when there are 33% Sybil nodes. To hold $P_{success} = 70\%$, there are 35% Sybil nodes

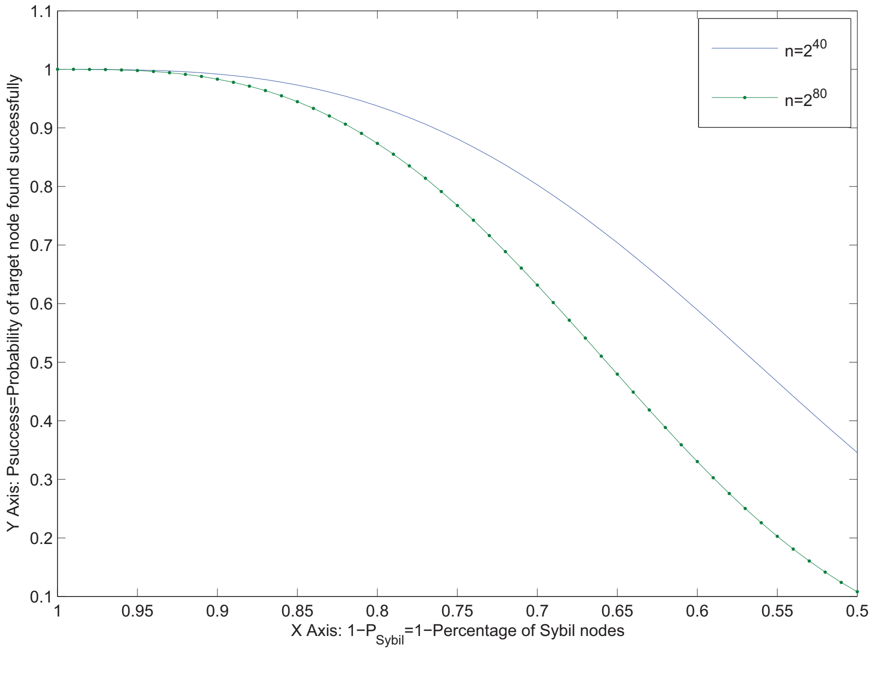


Figure 3.5: The distribution of $P_{success}$ when total number of nodes $n = 2^{40}, 2^{80}$ separately. α has been set to 3 and k equals to 20.

maximum for botnet with $k = 40$ and 40% Sybil nodes maximum for the one with $k = 100$. It is not a big improvement from $k = 20$ to $k = 100$. However, For such small improvement, every nodes are required to keep 80 triples more for each distances.

In a word, a larger k only strengthen botnet to a little extent, but the cost is comparatively high.

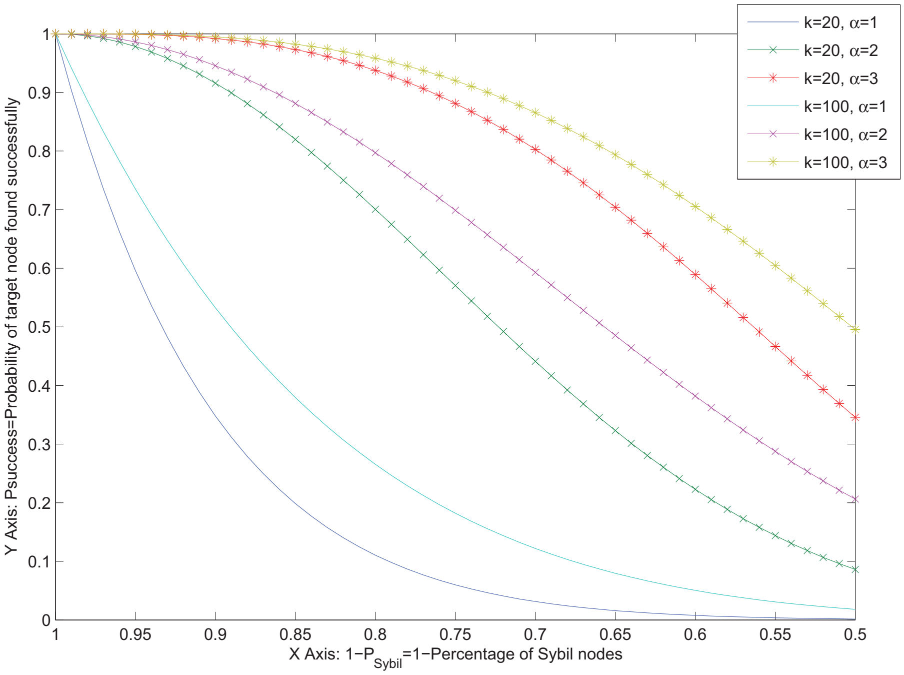


Figure 3.6: The distribution of $P_{success}$ when $k = 20, 100$ and $\alpha = 1, 2, 3$ respectively. Parameter n equals to 40.

3.3 Result Comparing

A familiar research to our study is Ping and her team's work [42]. They investigated in several aspects, such as network construction, C&C mechanisms, mitigation approaches, etc. Also, they analytically studied random Sybil attack on stormnet, which uses Kademlia-based protocol as it's communication module. $P_{success}$ is defined as the probability of a bot receiving real command, which is the same as our $P_{success}$, and it is given by: [42]:

$$P_{success} = (1 - \frac{n_{sybil}}{nsybil + n})^{l_{tz}} \tag{3.1}$$

Where l_{tz} is the length of a search path within the botnet. This l_{tz} is derived from average steps one node find the other [43].

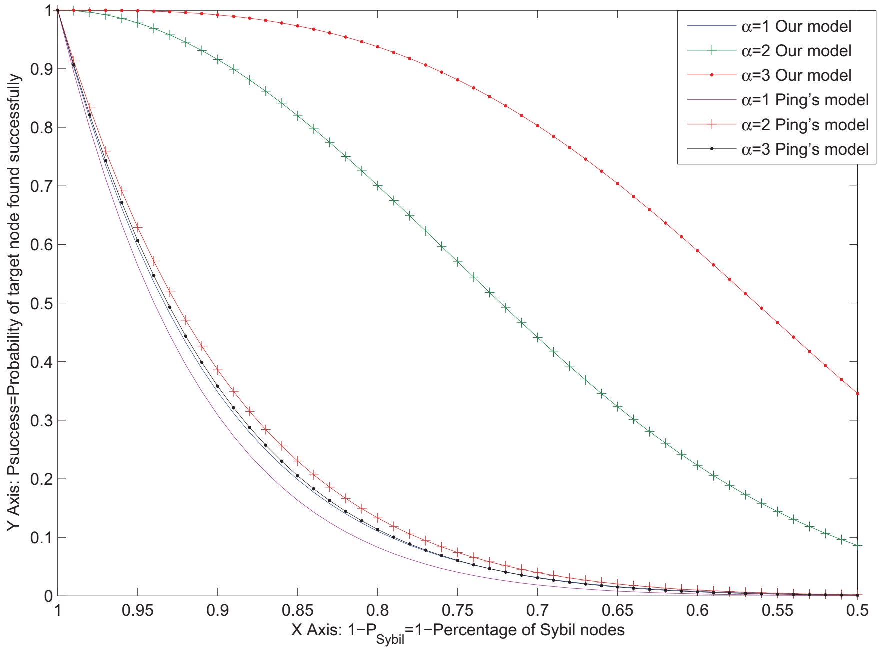


Figure 3.7: Comparing of results. $\alpha = 1, 2, 3$ for both cases and $k = 20$, $n = 2^{40}$

Ping's work is on random sybil attack. To compare two models, average steps uses in Ping's model is calculated from our model, in order to make two model comparable. Figure 3.7 shows when $\alpha = 1$, our result are similar, whereas totally different when $\alpha = 2, 3$. Actually, all Ping's three results are similar to our result when α equals to 1. It is because that Ping uses a normalized formula for different α, in which an average of steps are used as mentioned before. Then Ping's work ignores α the key structure characteristic of Kademlia. However, our research investigate the details of look-up procedure. For deriving the $P_{success}$, the steps probability distribution was involved instead of an average steps, thus how parameter α strengthen the botnet is represented. So our results for $\alpha = 2, 3$ are more accuracy than Ping's work.

3.4 Summary

In this chapter, several results from our research has been shown. From the results, three parameters are discussed. Then, an comparing to other's results has been given and it shows our research is more accuracy then others' work.

Chapter 4

Conclusion and future work

4.1 Conclusion

In chapter 2, we proposed a Kademlia based botnet model and two assumptions have been made. *a*) node ID space is full filled; *b*) Two types of attack, random sybil attack and fake target attack, have been proposed.Then the relationship between parameter α and steps distribution has been analyzed. First of all, a jumping function is derived. Then through which, a jumping matrix is then derived, followed by a steps distribution matrix. Later, by solving the matrix, we obtain the steps distribution. And with P_{sybil} introduced, the probability of finding target node successfully is achieved for both attack model. It shows that the botnet is weak when $\alpha = 1$ whereas pretty strong when $\alpha = 3$ in random sybil attack. A totally opposite result for fake target attack. Kademlia based botnet is vulnerable in facing fake target attack when $\alpha = 1, 2$ or 3. Those results gives clue on how to attack botnet more efficiently.

In chapter 3, more detailed results have been given. In the first place, it is shown that there are huge differences with different αs. When α increases, the probability of finding target node successful grows significantly. It also shows that when α is larger, the initiator takes less steps to reach target node. Then, it shows that the less nodes in the network, the stronger the network is. Also, a bigger parameter k do strengthen the botnet, but the cost is comparably high. Last but not the least, a comparing work has been done. Comparing with [42], our work shows that the Kademlia based botnet is hard to be attacked in facing random sybil attack, whereas it is vulnerable in facing fake target attack. Our work is a successive research of Ping's work, where we introduced details about $\alpha = 2, 3$ and a new attack model - fake target attack.

4.2 Future work

Although most of the important is captured by the model used in this thesis, the model can be developed. One possible future work is attempt of making the model more realistic. In the model, we assume the node ID space is full to simplify the calculation. However, in real botnet, it is not full and the nodes are distributed following special distribution. If that distribution can be found , more accurate results can be given. Moreover, nodes fail or leaving can be added to the model in order to make the model more realistic.

Because the jumping function is too complicated, we simplify it by only keeping the key characteristic of α. So another possible future work is to track with the original jumping function to gain a more accuracy results.

Bibliography

[1] K. Coffman and A. Odlyzko, "The size and growth rate of the internet," *First Monday*, vol. 3, 1998. (Cited on page 1.)

[2] M. Hasim and A. Salman, "Factors affecting sustainability of internet usage among youth," *Electronic Library*, vol. 28, 2010. (Cited on page 1.)

[3] P. Barford and V. Yegneswaran, *Malware*. 233 Spring Street, New York, NY, USA: Springer, 2007. (Cited on page 1.)

[4] M. Singh, "Peering at peer-to-peer computing," *IEEE Internet Computing*, vol. 5, pp. 4–5, 2001. (Cited on page 1.)

[5] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *SIGCOMM 2004 Conference on Computer Communications*. (Cited on page 1.)

[6] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, 2004. (Cited on page 1.)

[7] J. Davidson, *An Introduction to TCP/IP*. 233 Spring Street, New York, NY, USA: Springer, 1989. (Cited on page 2.)

[8] W. Zhou, “Keynote iii: detection and traceback of ddos attacks,” in *2008 8th IEEE International Conference on Computer and Information Technology.* (Cited on page 3.)

[9] G. Matthew, “Napster opens pandora ¡¯ s box : Examining how file-sharing services threaten the enforcement of copyright on the internet,” *Ohio State Law Journal*, vol. 63, 2002. (Cited on page 4.)

[10] W. Saddi and F. Guillemin, “Measurement based modeling of edonkey peer-to-peer file sharing system,” in *Managing Traffic Performance in Converged Networks. Proceedings 20th International Teletraffic Congress, ITC20 2007.* (Cited on page 4.)

[11] A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” 2001. (Cited on page 4.)

[12] A. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel, “Scribe: The design of a large-scale event notification infrastructure,” in *In Networked Group Communication*, pp. 30–43, 2001. (Cited on page 4.)

[13] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, “Scribe: A large-scale and decentralized application-level multicast infrastructure,” *IEEE Journal on Selected Areas in Communications JSAC*, vol. 20, 2002. (Cited on page 4.)

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *IN PROC. ACM SIGCOMM 2001*, pp. 161–172, 2001. (Cited on page 4.)

[15] P. Maymounkov and D. Mazieres, "A peer-to-peer information system based on the xor metric," in *In 1st International Workshop on Peer-to-peer Systems (IPTPS'02), 2002*. (Cited on pages 5, 15, 19 and 37.)

[16] S. Kaune, R. Rumi'n, G. Tyson, A. Mauthe, C. Guerrero, and R. Steinmetz, "Unraveling bittorrent's file unavailability: Measurements and analysis," in *Proceedings of the 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P 2010)*. (Cited on page 5.)

[17] J. Douceur and J. S. Donath, "The sybil attack," in *International workshop on Peer-To-Peer Systems. Retrieved 31 March 2011*, pp. 251–260, 2002. (Cited on page 6.)

[18] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*. (Cited on page 6.)

[19] J. B. Grizzard and T. Johns, "Peer-to-peer botnets: Overview and case study," in *In USENIX Workshop on Hot Topics in Understanding Botnets HotBots¡¯07*, 2007. (Cited on pages 7, 10 and 11.)

[20] E. Kartaltepe, J. Morales, S. Xu, and R. Sandhu, “Social network-based botnet command-and-control: Emerging threats and countermeasures,” in *Applied Cryptography and Network Security. Proceedings 8th International Conference, ACNS 2010*. (Cited on page 7.)

[21] R. Borgaonkar, “An analysis of the asprox botnet,” in *Proceedings of the 2010 Fourth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*. (Cited on page 7.)

[22] “Tdl-4: The ’indestructible’ botnet?,” June 2011. (Cited on page 8.)

[23] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang, “On the analysis of the zeus botnet crimeware toolkit,” in *2010 Eighth Annual International Conference on Privacy, Security and Trust (PST)*. (Cited on page 8.)

[24] “Zeus gets more sophisticated using p2p techniques,” October 2011. (Cited on pages 8 and 9.)

[25] G. Tenebro, “W32.waledac threat analysis,” tech. rep., Symantec Lab. (Cited on page 9.)

[26] “Microsoft gets legal might to target spamming botnets,” Aguest 2010. (Cited on page 10.)

[27] "Storm botnet storms the net," September 2007. (Cited on page 10.)

[28] "Microsoft: We took out storm botnet," April 2008. (Cited on page 10.)

[29] "Guessing at compromised host numbers," Sempteber 2007. (Cited on page 10.)

[30] "Microsoft didn¡¯t crush storm, counter researchers," April 2008. (Cited on page 10.)

[31] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm," in *LEET'08 Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. (Cited on pages 10, 11 and 18.)

[32] D. Dittrich, F. Leder, and T. Werner, "A case study in ethical decision making regarding remote mitigation of botnets," in *Financial Cryptography and Data Security. FC 2010 Workshops, RLCPS, WECSR, and WLC 2010. Revised Selected Papers*. (Cited on page 10.)

[33] A. Spognardi, A. Lucarelli, and R. Di Pietro, "A methodology for p2p file-sharing traffic detection," in *Proceedings. Second International Workshop on Hot Topics in Peer-to-Peer Systems*. (Cited on page 11.)

[34] M. Jelasity and V. Bilicki, "Towards automated detection of peer-to-peer botnets: on the limits of local approaches," *EET'09 Proceedings of the 2nd USENIX confer-*

ence on Large-scale exploits and emergent threats: botnets, spyware, worms, and more, 2009. (Cited on page 11.)

[35] (Cited on page 11.)

[36] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Proceedings - 2009 3rd International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009*. (Cited on page 11.)

[37] H. R. Zeidanloo, M. J. Zadeh, Shooshtari, P. V. Amoli, M. Safari, and M. Zamani, "A taxonomy of botnet detection techniques," in *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*. (Cited on page 11.)

[38] L. Khan, M. M. Masud, J. Gao, J. Han, and B. Thuraisingham, "Peer to peer botnet detection for cyber-security: A data mining approach," in *CSIIRW'08 - 4th Annual Cyber Security and Information Intelligence Research Workshop: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead*. (Cited on page 11.)

[39] P. Wang, L. Wu, R. Cunningham, and C. Zou, "Honeypot detection in advanced botnet attacks," *International Journal of Information and Computer Security*, vol. 4, pp. 30–51, 2010. (Cited on page 11.)

[40] B. Stone-Gross, M. Cova, B. Gilbert, R. Kemmerer, C. Kruegel, and G. Vigna, "Analysis of a botnet takeover," *IEEE Security & Privacy*, vol. 9, 2011. (Cited on page 12.)

[41] C. Davis, J. Fernandez, S. Neville, and J. McHugh, "Sybil attacks as a mitigation strategy against the storm botnet," in *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)*. (Cited on page 12.)

[42] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "A systematic study on peer-to-peer botnets," in *Proc Int Conf Comput Commun Networks ICCCN*. (Cited on pages 12, 13, 50 and 54.)

[43] D. Stutzbach and R. Rejaie, "Improving lookup performance over a widely-deployed dht," 2006. (Cited on pages 12 and 51.)

[44] C. Davis, S. Fernandez, J.M.and Neville, and J. McHugh, "Sybil attacks as a mitigation strategy against the storm botnet," in *2008 3rd International Conference on Malicious and Unwanted Software (MALWARE)*. (Cited on page 15.)

[45] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Computer Communication Review*. (Cited on page 45.)