The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

SELF-DETERMINING FORWARDING SCHEME FOR DEFENDING AGAINST QUERY-FLOODING BASED DDOS ATTACKS IN UNSTRUCTURED PEER-TO-PEER SYSTEMS

A Thesis in

Computer Science and Engineering

by

Kang-Hsien Chou

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2008

The thesis of Kang-Hsien Chou was reviewed and approved* by the following:

Wang-Chien Lee
Associate Professor of Computer Science and Engineering
Thesis Advisor

John J. Metzner
Professor of Computer Science and Engineering

Raj Acharya
Professor of Computer Science and Engineering.
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School

ABSTRACT

A flooding-based search mechanism is commonly used in unstructured peer-to-peer systems, such as Gnutella. However, due to its flooding nature, this mechanism is vulnerable to query-flooding based distributed denial-of-service (DDoS) attacks. Most existing defense techniques only protect networks from network layer DDoS attacks or are unsuitable for peer-to-peer systems. Hence, this thesis proposes a DDoS defense technique aiming at the distributed and dynamic nature of peer-to-peer systems. Each peer in the system can decide to drop or forward a query according to information of the query issuer's past behavior sent along with a received query. This information includes whether or not the query issuer has downloaded a reasonable amount of files during each of the past observation intervals. Verification of the proposed scheme uses real Internet topologies generated from BRITE to simulate query-flooding based DDoS attacks. The simulation results show the effectiveness of the proposed scheme. Moreover, the result shows that the scheme can mitigate query-flooding based DDoS attacks while malicious peers cooperate with each other to cheat defense approaches.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Flooding based Distributed Denial of Service (DDoS) has been a serious problem in the Internet for many years. As file-sharing applications become more and more popular and important today, peer-to-peer systems encounter the same threat. In contrast to the network layer DDoS attacks, peer-to-peer systems may suffer in the application layer from query-flooding based DDoS attacks, which may cause serious damage and prevent legitimate queries from receiving service. Therefore, developing defense mechanisms in peer-to-peer applications is critical.

1.1 Background

Flooding based Distributed Denial of Service (DDoS) attacks cause an overwhelming quantity of traffic from multiple hijacked machines to the targeted victims. The high quantity of traffic that arrives at victims would quickly exhaust some key resources, such as bandwidth, CPU capability, etc. Consequently, the victims do not have enough resources to serve or respond to regular users. While peer-to-peer systems become more and more popular, a similar issue arises in unstructured peer-to-peer systems, which perform flooding based search.

Noted in [9] and [22] is that unstructured peer-to-peer systems are vulnerable to flooding-based DDoS attacks. Unstructured peer-to-peer systems are convenient and very commonly used for searching share-files in today’s Internet. By using the flooding-based search mechanism, a query is broadcast and rebroadcast until certain criteria are satisfied to terminate the flooding process. However, the flooding-based search mechanism makes it very vulnerable to the threat of query-flooding based DDoS attack. Other than DDoS attacks people used to discuss in the network layer, query-flooding based DDoS attacks occur in the application layer. It is performed by multiple compromised peers, who generate as many bogus queries as they can toward the victims. Those bogus queries will be rebroadcast neighbor by neighbor. Bogus queries from malicious peers can exponentially increase throughout the system with the ultimate result of exhausting the system's resources.

Much research work [2][3][5][6][7][9][10][11][13][14][15][21][23][24][29][30][32][33][35] has been done for defending against DDoS attacks. Unfortunately, those approaches focus on the network layer defense and do not work effectively for defending against query-flooding based DDoS attacks in the application layer.

In addition, query-flooding based DDoS attacks do not need to generate specific malicious query content or header field values since their ability to do damage to the system simply lies in the vast amount of traffic. This creates difficulty for finding criteria to identify malicious peers or filter attack queries. DD-POLICE [22] is the most recent effort which focuses on defending query-flooding based DDoS attacks in unstructured peer-to-peer systems. It uses the amount of queries issued by a peer as the criteria to identify malicious peers. Peers exchange messages to identify the peers issuing a large number of queries, and then enforce disconnects. However, legitimate heavy users are adversely affected. Moreover, retrieving information from other peers while they leave the system is difficult. Besides, commanding other peers to perform disconnection operations may be beyond the authority of a peer. Cooperation between malicious peers would also probably cause inefficiency of the mechanism. Hence, a more appropriate solution to deal with query-flooding based DDoS attacks in unstructured peer-to-peer systems remains necessary.

1.2 Basic Ideas

This study borrows an idea from D-WARD [24], a network layer source-end defense technique, to develop the proposed application layer DDoS defense solution. D-WARD monitors the behavior of incoming and outgoing packets at the source end routers and prevents attack packets entering the Internet.

Similarly, the outgoing queries and the size of downloaded data become criteria in the proposed scheme to decide whether a peer's behavior is reasonable or not. Accordingly, a query is forwarded or dropped according to a rate-limiting strategy based on the information of outgoing queries and extent of downloaded data. A framework is proposed to support the defense scheme such that each file provider should provide a certified message periodically to indicate the occurrence of file downloading and resource consumption. A peer should attach those certificates to queries it generates for other peers to examine. This is practical because implementation is in peer-to-peer applications and peers must install specific software to join the system.

By adopting the proposed mechanism, a peer does not need to request information from other peers while processing queries. Moreover, a peer filters queries locally instead of commanding other peers to complete a disconnection. The mechanism provides a solution to defend against query-flooding based DDoS attacks for peer-to-peer systems which are dynamic and distributed in nature.

1.3 Contributions

This thesis proposes a self-determining forwarding scheme to defend against query-flooding based DDoS attacks in unstructured peer-to-peer systems. Due to the lack of criteria for detecting bogus queries, a new criterion for a peer’s behavior is defined to determine if it is harmful to the peer-to-peer system. The proposed mechanism allows each peer to examine a query issuer’s behavior through the information sent along with a query. Based on that, each peer can decide whether to forward or drop a query without communicating with other peers. The proposed mechanism is designed for peer-to-peer systems which have a dynamic and distributed nature. This study develops simulation to evaluate the proposed schemes, and the simulation shows significant performance for defending against query-flooding based DDoS attacks. Meanwhile, since malicious peers may cooperate with each other to cheat defense techniques, the proposed schemes effectively mitigate query-flooding based DDoS attacks despite cooperative behavior exists between malicious peers. The new scheme provides a possible solution while existing techniques may not work effectively under the scenario.

1.4 Synopsis

The rest of the thesis is organized as follows. Chapter 2 describes related work and Chapter 3 presents the proposed model and forwarding schemes. Chapter 4 details the performance of the proposed scheme via simulations and Chapter 5 makes a conclusion of the thesis.

Chapter 2

Related Work

Much work has been done to defend against DDoS attacks. Most efforts involve network layer approaches including IP traceback, filtering, and source-end defense schemes. This chapter reviews some of those network layer techniques, and then introduces two approaches which focus on handling the query flood issue in peer-to-peer systems.

2.1 Network Layer Approaches for Defending against DDoS Attacks

IP spoofing is the most challenging issue for defending against DDoS attacks, and hence, most proposed approaches concentrate of this issue. Forged source IP addresses creates difficulty for identifying attackers and distinguishing legitimate packets from malicious packets. Defense techniques, including IP traceback schemes [3][5][6][7][9][10][13][29][30][32], filtering techniques [11][33][35], and source-end defense systems [21][23][24], have different characteristics and varying degrees of success are described as follows.

2.1.1 Traceback and Authenticated Marking

IP traceback techniques focus on cooperation in the core networks. A router would periodically send out ICMP packets with packets going through it or leave information on the packets going through it. A victim can reconstruct attack paths through the additional information to trace back to attackers.

A series of marking algorithms, including node append, node sampling and edge sampling are proposed in [29] and [30]. The node append algorithm appends each router's address to the end of the packet as it travels from attacker to victim. In the node sampling algorithm, each router has probability, *p*, to write its address to a packet. Edge sampling, instead of marking each node's IP address, writes two adjacent nodes' IP addresses to the IP header to represent an edge. The compressed technique is also proposed to reduce the storage requirement. It makes use of the exclusive-or (XOR) of two adjacent node's IP addresses to represent an edge, and subdivide each IP address to several fragments. The full attack path can be reconstructed by the victim after enough marked packets are received.

However, the marks left by routers may be forged by attackers or malicious routers. Hence, the authenticated marking scheme is proposed in [32]. The authenticated marking scheme assumes victims can have shared secret keys with each router so that techniques of HMAC can be adopted. Moreover, time-released key chains are also proposed as another practical option for authentication.

2.1.2 Detection and Filtering

In addition to discovery of attackers, a common way of responding to DDoS attacks and protecting victims is to filter attack packets once detection indicates the occurrence of attacks. Ingress filter [11] prevents the appearance of IP spoofing. A router with an ingress filter can drop packets with illegal source IP addresses in its subnet. Widespread deployment is necessary for ingress filters to prevent packets with forged IP address from entering the Internet. However, this requirement is hard to fulfill in peer-to-peer systems.

Traceback-based packet filtering [33] is proposed to filter attack packets based on the adoption of traceback algorithms. After victims reconstruct the attack paths, edges of the attack paths could be used as the criteria to filter attack packets. Packets marked with the information of edges in the attack paths are identified as suspicious and filtered. However, since this scheme needs to work with traceback algorithms, use in peer-to-peer systems is difficult.

Instead of using traceback algorithms, [35] proposes a path identification mechanism to distinguish the packets coming from different sources. Each router writes one or two bits in the identification field of the IP header. TTL is used to index which slot in the identification field should be currently marked. The markings of identified attack packets are recorded, and then, a packet filter decides whether to drop a packet according to the information. However, TTL can be easily modified by malicious peers. Furthermore, since no specific destination of queries exists, forwarding in the system is different from packets’ routes in the Internet, and difficulty arises for adopting the mechanism in peer-to-peer systems.

2.1.3 Source-end Defense Systems

While most defense approaches focus on victim or core networks, D-WARD [21][23][24] is proposed to deploy at source-end routers and prevent attack packets from entering the Internet. D-WARD uses observation component to analyze the behavior between incoming and outgoing packets to detect whether or not any anomaly exists in the traffic. Once attack or suspicious traffic is detected, a rate-limiting component would limit the sending rate of those packets.

However, in peer-to-peer systems, attack traffic and normal traffic may come from the same logical link because of flooding searches. Thus, D-WARD is not effective in peer-to-peer systems. However, the idea of observing traffic behavior to control attack traffic inspired development of this study's new DDoS defense scheme for peer-to-peer systems. Moreover, peer-to-peer systems are suitable for a source-end defense approach since implementation of the mechanism would be in peer-to-peer applications and peers are required to install specific software before joining the system.

2.2 Defending against Query-Flooding Attacks in Unstructured Peer-to-Peer Systems

Query flooding is an inherent issue with unstructured peer-to-peer systems. Some research such as [17] and [36] work on improving the performance of peer-to-peer systems deteriorated due to flooding issues. However, they do not focus on defending against malicious query-flooding attacks. The work most related to our study are [8] and

[22]. These are based on different assumptions and hence develop different approaches to defend against query-flooding attacks.

2.2.1 Query-Flooding Based DDoS Attacks and Load Balancing Strategies

Query-flooding based DDoS attacks are first discussed in [8], which suggests that a Gnutella peer-to-peer system is very vulnerable to query-flooding attacks, and its popularity makes addressing issue important. It [8] assumes difficulty distinguishing a high-load of legitimate queries from attack queries. Based on the assumption, several load balancing strategies to mitigate the damage of the threat are proposed. The service guarantee is also defined and used as the evaluation matrix in this work.

Those strategies include incoming allocation strategy and drop strategy. Incoming allocation strategy assigns different query bandwidth for different incoming links. Drop strategy drops excess queries of specific links according to different TTL values. In brief, it [8] does not distinguish attack queries from normal queries but maintains a fair load distribution in peer-to-peer systems. The approach focuses on DoS attack and hence would be less effective while the number of malicious peers increases within DDoS attacks.

2.2.2 Defending P2Ps from Overlay Flooding-Based DDoS Attacks

In [22], Liu et al. express concern for the severity of query-flooding based DDoS attacks in unstructured peer-to-peer systems. It can cause heavy damage in that a small numbers of propagated messages consume a large amount of bandwidth and computing resources. In contrast to the work introduced in the previous section, a distributed approach, DD-POLICE, is proposed to identify the attackers and disconnect them. Identifying a bad peer is according to the amount of queries it issued.

DD-POLICE has three steps: 1) neighbor list exchange, 2) neighbor traffic monitoring, and 3) bad peer reorganization. Each peer maintains a neighbor list including all its logical neighbors. Two neighboring peers exchange their neighbor lists periodically. Once a peer receives queries from one of its neighbors more than a preset threshold, the neighboring peer is marked as a suspicious peer. According to the neighbor list, this peer can communicate with the neighboring peers of the suspicious peer to check how many queries are from them. Consequently, the number of queries originally issued from the suspicious peer can be discerned. If the number is over the preset threshold, then the suspicious peer would be disconnected by all its neighboring peers.

However, frequent neighbor list exchanges and traffic inquiries may result in heavy overhead in the system. Additional messages may also be used by attackers to consume system resources. The authority to command other peers to perform disconnection may possibly be used by attackers. Besides, simply using the volume of queries as the criteria to identify attackers ignores the needs of legitimate heavy users. Moreover, the mechanism may not be efficient to defend against cooperation between malicious peers.

2.3 Summary

Many different approaches have been proposed for defending against DDoS attacks. Unfortunately, most of them defend against attacks at the network layer, and thus do not meet the needs of peer-to-peer systems. Some effort has focused on defending against query-flooding based DDoS attacks, and DD-POLICE makes some initial contributions. However, the assumptions and models of these existing techniques may not be appropriate for peer-to-peer systems. The concepts of D-WARD including source-end defense, observation and rate-limiting model contribute to develop a new approach. The authenticated marking scheme provides the idea for preparing the framework to support the mechanism proposed in this study.

Chapter 3

Design of Self-Determining Forwarding Scheme

This chapter describes the design of the proposed self-determining forwarding scheme for defending against query-flooding based DDoS attacks in unstructured peer-to-peer systems. The observations of attack behavior are described first, and then, the framework is introduced. These are followed by detailed explanations of each component.

3.1 Observation of Attack Behavior

To defend against query-flooding attack, the key is to prevent bogus queries from being forwarded in peer-to-peer systems. Hence, an essential element is to distinguish bogus queries from legitimate ones. Unfortunately, bogus queries do not contain specific malicious query content or header field values, and hence identifying them is difficult. Observing the number of queries from a peer may be a possible solution. In [22], number of queries issued by a peer is used as the criteria to decide whether or not it is an attacker, ignores the possibility of legitimate heavy users.

In flooding attacks, attackers would overwhelm victims' CPU, memory, and network resources by sending large numbers of spurious requests [25]. The resource consumption between the attacker end and the victim end is quite different for a service request, and for that reason an attacker can efficiently overwhelm the victim [2][15][25].

In other words, an attacker may only expend a small amount of resources to send a packet or a request while the victim needs to allocate much more resources to respond to the request, even though the attacker never plans to spend its own resource to receive responses. For example, an attacker may send out many TCP connection requests to a victim machine. Each request causes the targeted machine to instantiate data structures from limited resources to remain a self-open TCP connection. Once the targeted machine's resources are exhausted, no more TCP connections can be established and thus regular services to other users is not possible. Moreover, the source IP address is spoofed, so the attacker would not receive any reply from the victim. It does not consume the attacker's resources beyond sending the request, while the victim is forced to consume more computation power, bandwidth, and resources to serve the request.

Another well-known example is smurf attack, ICMP echo floods, which floods a targeted system via spoofed broadcast ping messages. In the attack, an attacker would send a large number of ICMP echo requests to IP broadcast address, but containing the spoofed source address of the victim. All the hosts in the Internet receiving the ICMP echo request would reply to the victim and hence flood it. Obviously the cost of resources consumed in attackers and victims are highly unbalanced. While attackers can expend all their resources on sending attacks, the damage to the victims is multiplied. Regarding to unbalanced resource consumption, D-WARD [24] observes incoming and outgoing traffic at source end to detect attacks.

While the query-flooding based DDoS attacks in unstructured peer-to-peer systems are discussed, similar situation remains. The attackers try to flood the targeted victims with relatively little resource costs to the attackers to generate attack queries, but the attack consumes much more of the system's resources since these attack queries broadcast between peers and exponentially increases through the system. Since attackers never plan to respond to query-hit and download files, all their resources could be used to generate bogus queries and cause great damage.

Table 1: Peer behavior and impact

Scenario	Cause	Impact
Few queries + No/little file downloading	1. The peer searches and downloads files infrequently. 2. The Peer searches files infrequently and does not find files.	Harmless
Few queries + Heavy file downloading	1. The peer searches files infrequently but downloads files with large size.	Harmless
Many queries + No/little file downloading	1. The peer searches many files but does not find them or does not want to download them. Although it does not hurt the system intentionally, generating too many queries in a time of period would cause congestion in the system. 2. This is a malicious peer who sends large volumes of bogus queries.	Harmful
Many queries + Heavy file downloading	1. The peer searches many files and downloads them. 2. This is a malicious peer who tries to download large volumes of files to make behavior looks reasonable. However, the ability to generate bogus queries is limited by its bandwidth consumption.	Harmless

According to the discussion above, a relationship between outgoing and incoming traffic should exists. In other words, the ratio between outgoing queries and incoming traffic from downloading files could be used as the criteria to decide whether a peer's behavior is reasonable or not. A similar idea is also mentioned in [18], which suggests that query should send no more data than data retrieved. Different scenarios of query issuing and downloading are described in Table1. If a peer generates queries infrequently, no matter how many files it downloads, it is not considered harmful to the system. However, if the peer generates queries frequently but downloads no or only few files, harm may accrue to the system even though this may not be its intention. The most interesting scenario occurs whenever a peer generates queries frequently while downloading large volumes of files. This could be considered reasonable behavior, even though this is a malicious peer, the ability to attack the system is limited because the file downloading already consumes its bandwidth.

3.2 Framework

The discussion in the previous section provides a criterion for deciding whether or not to forward a query. If a query's generator does not have reasonable behavior, then a peer can choose to drop the query while receiving it. However, the characteristics of unstructured peer-to-peer systems make it a major challenge for a peer to know if the query generator has reasonable incoming traffic for file downloading. First, the distributed nature in which there is no central server makes it difficult for cooperation. Second, peer-to-peer systems are highly dynamic since peers join and leave dynamically. Collecting information from certain peers while they leave the system would be very difficult.

Therefore, a framework is proposed as illustrated in Figure 1. A file provider should send a receipt message periodically to the peer who is downloading files from it. The receipt message contains the information of the volume downloaded during an observation interval. The receipt could be evidence to show a peer's activity and attach to queries issued in the future. A peer could decide whether or not to forward a query according to the attached receipts. By this way, a peer does not need to retrieve information from other peers who may not currently be in the system, when receiving a query.

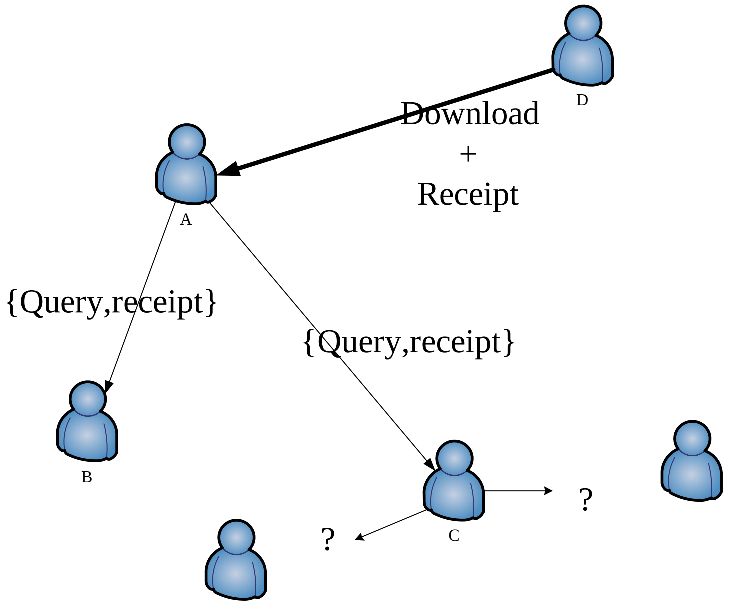

Figure 1: Framework of the self-determining forwarding scheme

A fundamental shortcoming of the framework is that the receipt may be forged and provide incorrect information. Hence, a mechanism to authenticate receipts is needed. Message Authenticated Codes (MAC) could be used to authenticate receipts. HMAC [16] are commonly used for two-party message authentication through a shared secret key.

When party *A* sends a message *M* to party *B*, *A* appends the message with the MAC of *M* using key *K*. When *B* receives the message, it can check the validity of the MAC. This technique is the one adopted in the Authenticated Marking Scheme in [32] to authenticate the mark left by routers in the networks. Different approaches for key agreement and distribution are proposed and some of them are designed for peer-to-peer systems such as [20]. Hence, the assumption is that each peer shares a unique secret key with different peers.

3.3 Receipt Format

The receipt format appears in Figure 2. The first four bytes are used to record the IP address of the file downloader, and it attaches the receipt upon issuance of a query. Including the field can prevent the receipt from being stolen and replayed. The second four bytes are used to record the IP address of the file provider. The third four bytes indicate the size of data has been transmitted during the observation interval. The following four bytes are a time stamp indicating the time of the receipt generation. It can provide the effective time of this receipt and prevent it from being replayed. The last eight bytes are used for HMAC following the design in [32].

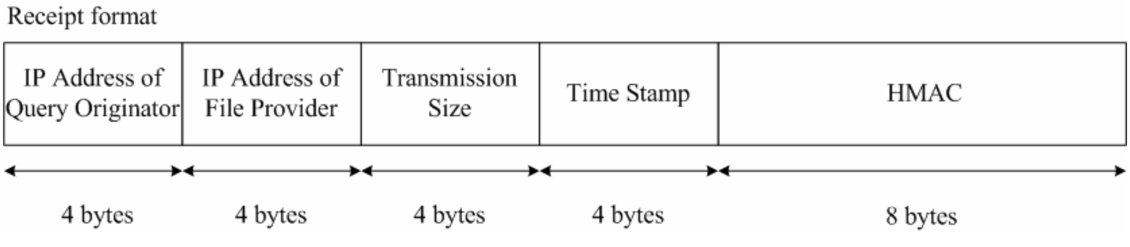

Figure 2: Content and format of a receipt

3.4 Rate-limiting Strategy

To successfully flood peer-to-peer systems requires a large volume of queries remaining in the system to occupy the resources such as computation capacity and bandwidth. Consequently, malicious peers would send out a large number of bogus queries in each time interval. Therefore, given abovementioned framework, designing a rate-limiting strategy to drop malicious queries could mitigate the query-flooding based DDoS attacks. Rate-limiting techniques are commonly used in defending DDoS attacks, and different variations occur in different schemes. It is used in [24] to prevent attack or suspicious packets from entering the Internet. Another technique [8] also adopts a similar idea of dropping queries in order to balance loads in peer-to-peer systems.

If peers make bad decisions, they may process bogus queries generated by other malicious peers and broadcast to neighboring peers. If all those neighboring peers do the same, bogus queries increase exponentially in the system. If peers make good decisions, they can at the very beginning prevent bogus queries from increasing and being broadcast, and minimize the effect of query flooding. If most bogus queries are dropped at the source end, neighboring peers of a malicious peer, damage would be controlled most effectively.

The strategy of rate limit is shown in Figure 3. A pre-determined value allows a small number of queries from different peers to be forwarded without receipts in each time interval. This occurs because some peers do not issue queries frequently, or they initialize a search just during the time interval and have not begun to download files. Otherwise, if queries generated by a peer exceed the assigned value, those queries should be forwarded according to a query issuer's behavior. If a peer downloads more data during the observation interval, that peer would be given more weight for its queries to be forwarded. A *balance factor*, *e*, is assigned to decide the ratio between outgoing query traffic and incoming traffic of data downloaded in a peer.

Limited value : a pre-determined value to allow a small number of a certain peer's queries without receipt to be forwarded in an observation interval

A_{source} : total size of query from a certain query originator in an observation interval

$A_{download}$: total size of data downloaded by the query originator in an observation interval

e (balance factor) : positive fraction representing the reasonable ratio between incoming data traffic and outgoing query traffic

```
if (amount of queries from a specific peer < limited value)
    $P = 1$
else
    $P_{next} = \min(1, P_{previous} \times \frac{A_{download}}{A_{source}} \times e)$
```

Figure 3: Rate-limiting strategy

3.5 Forwarding Scheme I

Based on the proposed framework and rate-limiting strategy, a forwarding scheme is shown in Figure 4. In the forwarding scheme, each peer should prepare a source table to record the source IP address of a query, the number of queries from a source peer coming in during the observation interval, and the total size of queries from a source peer coming in during the observation interval. The table would be cleaned periodically. Through the table, the rate-limiting strategy proposed in previous section can have enough information to decide whether forward a query or not.

```
1     for each query q
2         update source_table
3         forwarded_number = number of queries from a peer, p,have
                                  been forwarded in the observation interval
4         if (forwarded_number < limited value)
5             forward q
6             forwarded_number++
7     else
8         if (receipt.MAC is valid)
9             calculate the forwarding probability for q
10            if (q is forwarded)
11                forwarded_number++
12        else drop q
```

Figure 4: Forwarding Scheme

If the number of queries from a source peer forwarded during the observation interval is not over the limited value, queries from this source peer can be directly forwarded and the amount would accumulate (line 1-5). Otherwise, the query with valid receipts should be forwarded according to the rate-limiting strategy proposed in previous section. If MAC of the query's receipt is not valid, the query would be dropped directly (line 6-11).

3.6 Forwarding Scheme II

Through the abovementioned scheme, unreasonable query generation can be prevented from being forwarded in the system. However, scheme I may suffer from variations of sybil attack. In sybil attack, a malicious peer cooperating with other malicious peers, holds multiple identities to cheat the system. A malicious peer that owning multiple identities could forge receipts which would damage the proposed scheme.

A current common solution against sybil attack is resource testing [19][26]. These tests include checks for computing capability, bandwidth, and so on. Hence, cheating can be detected if a malicious peer claims to be downloading a large volume of files by using other malicious peers' identities to forge receipts. However, the solution if used here could cause vulnerability, since each malicious peer can pretend to be different cooperative peers to generate a reasonable amount of queries with forged receipts.

Thus, we modify the forwarding scheme I described in the previous section. Instead of using each query's originator to index the source table, clustering querys' originators and receipts' originators to index another table, cluster_table. During an observation interval, all related peers are clustered into the same entry. Since a malicious peer intends to generate large volumes of bogus queries in an observation interval, those high-volume incoming bogus queries with limited receipt originators would be more easily clustered in the same category than those of legitimate peers.

In the clustering algorithm shown in Figure 5, while a query is being clustered, all the existing entries containing an identity the same as the query's originator or the receipts' originators should be combined into one category. The query's information is also added to that category.

```
for each query q
    add q's originator and all receipts' issuers as members $m_1,..,m_n$ into
    an entry $c$ in the cluster_table
    for each entry in the cluster_table
        if it contains any member of entry $c$
            combine the entry to entry $c$
            accumulate recorded query size
```

Figure 5: Clustering algorithm

3.7 Summary

The observations of peer behavior and characteristics of peer-to-peer systems determine the criteria to mitigate the flood of bogus queries. Thus, proposed framework, rate-limiting strategy and forwarding schemes defend against query-flooding DDoS attacks in unstructured peer-to-peer systems.

Chapter 4

Performance Evaluation

The experiments adopt a simulation model similar to [22]. This chapter describes the simulation setup and experiment design, followed by a discussion of the results.

4.1 Simulation Setup

BRITE is used to generate the network topology with 5,000 peers and to set the bandwidths. BRITE collects data from GT-ITM, Inet, NLANR AS, and CAIDA's Skitter map to generate real-network topologies.

In the simulation, each legitimate peer sends 0.3 queries per minute. To observe the service received, 10,000 queries are issued. The average query size is 105.6 bytes [1]. Each malicious peer generates as many bogus queries as it is capable [8][22]. Hence, the number of attack queries sent by a compromised peer per minute, Q_b, is given by

$$Q_b = \min\{20{,}000, \text{the capacity of the link}\}$$

Evaluation of whether or not the proposed schemes mitigate the query-flooding based DDoS attacks uses a metric, service rate, defined in [8]. This metric measures the ratio of queries that can be serviced in the system. In a peer-to-peer environment, using Q_n denotes the total number of forward services for all legitimate queries while no DDoS attacks occur, and Q_a denotes the total number of forward services for all legitimate queries while DDoS attacks occur, then the service rate S is given by

$$S = \frac{Q_a}{Q_n} \times 100\%$$

Four experiments are designed to evaluate the performance of the proposed scheme for defending against query-flooding based DDoS attacks. They are described as follows.

Experiment I:

This experiment evaluates the average service rate in the system with *k* random peers, where *k* ranges from 1 to 100, are malicious peers. All peers except malicious ones are downloading files within assigned bandwidths. Three different balance factors are used to observe their impact.

Experiment II:

This experiment evaluates the average service rate while legitimate heavy users exist in the system. 10 peers, 0.2% of the system, are randomly selected as legitimate heavy users and they issue high volumes of queries (100 queries per minute) while downloading files with fully engaging bandwidths.

Experiment III:

This experiment evaluates the average service rate with Scheme II. No cooperation occurs between malicious peers in this experiment.

Experiment IV:

This experiment evaluates the average service rate with Scheme II, while *k* malicious peers cooperate with each other. Each bogus query uses a different forged source IP address and attaches *n* forged receipts with information of cooperative malicious peers, where *n* ranges from 1 to 10 [28].

4.2 Results

Figure 6 shows the performance of proposed Scheme I defending against query-flooding based DDoS attacks with various balance factors, *e*, which means different ratios between outgoing query traffic and incoming traffic of downloaded data. The service rate quickly drops to 40 percent when only 10 malicious peers appear, and then further degrades with the increase of the number of malicious peers. Proposed Scheme I, with different balance factors, could remain over 80 percent service rate while query-flooding attacks occur. No obvious differences appear when different balance factors are adopted, because all malicious peers can not forge receipts. Hence, only a limited number of bogus queries can be forwarded during an observation interval during which a peer receives them. All other bogus queries would be dropped because of a lack of valid receipts.

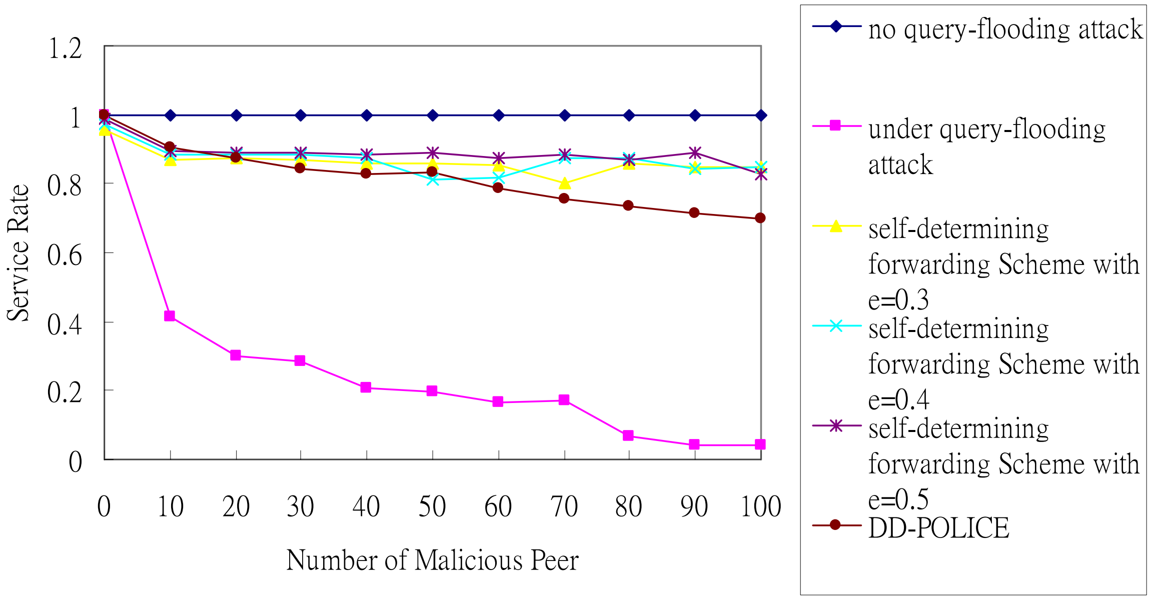

Figure 6: Service rate while scheme I is adopted

Another curve represents the performance of DD-POLICE, which disconnects a peer who is identified for sending too many queries. Before malicious peers are detected, some volumes of bogus queries are already forwarded in the system. This is the major reason for loss of service rate. No significant difference between the proposed Scheme I and DD-POLICE appears. However, no legitimate users in this experiment send a large number of queries.

Figure 7 shows the performance of defending against query-flooding attacks while 10 heavy peers, 0.2% of all peers, exist in the system. The service rate would largely drop while DD-POLICE is adopted, because all the heavy peers are disconnected since they issue a large number of queries, and their queries can not receive service. Nevertheless, the service rate remains high when proposed scheme is adopted. Queries form those heavy peers can be serviced because their receipts shows their reasonable behavior.

Figure 8 shows that Scheme II maintains a high service rate when no cooperation exists between malicious peers. It is because only a limited number of bogus queries could still be forwarded in each clustered category. All other bogus queries would be dropped since they do not contain valid receipts. Although some legitimated queries originated from different peers may be clustered into the same entry and accumulated, these queries can still be serviced because their originators do not issue queries frequently and continue to download data.

Figure 9 demonstrates the performance of Scheme II for defending against query-flooding based DDoS attacks while cooperation exists between malicious peers. Choosing a balance factor, *e*=0.4, would be a better choice, because while dropping bogus queries, it allows more legitimate queries to be forwarded. Although the service rate degrades while the number of malicious peers increases, it can maintain over 50 percent service rate when fewer than 30 malicious peers operate in the system, and improve more than 10 percent service rate when fewer than 60 malicious peers operate in the system. Forged receipts can still allow some volume of bogus queries to be forwarded. When the number of malicious peers increases, more bogus queries are serviced, which explains the major reason for the degradation in service rate.

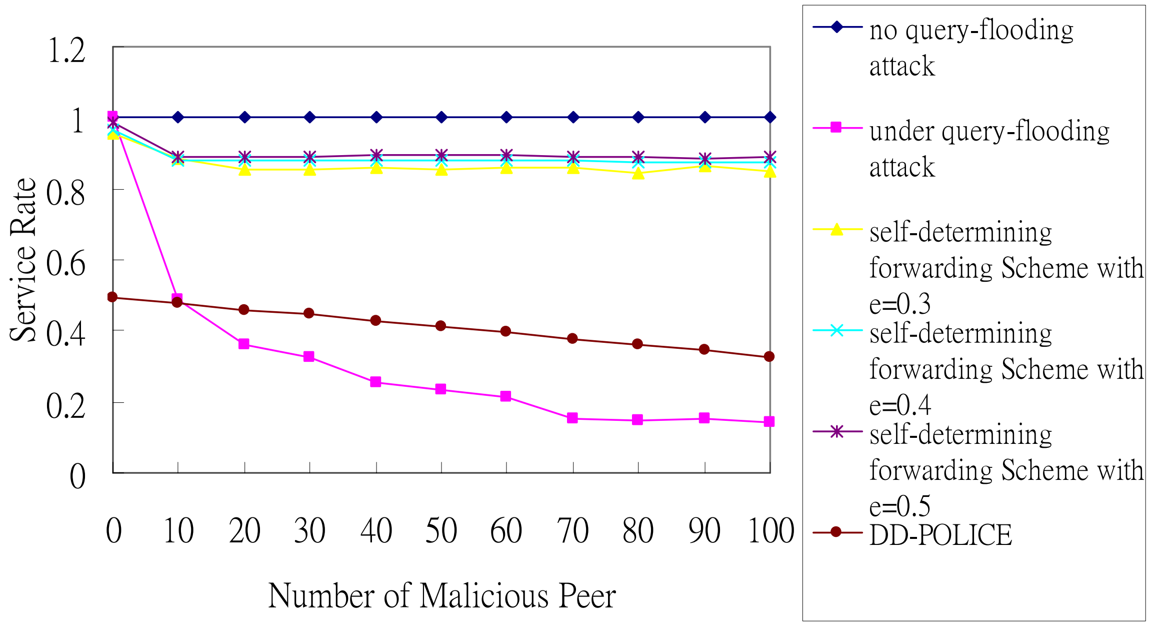

Figure 7: Service rate within 10 legitimate heavy users when Scheme I is adopted

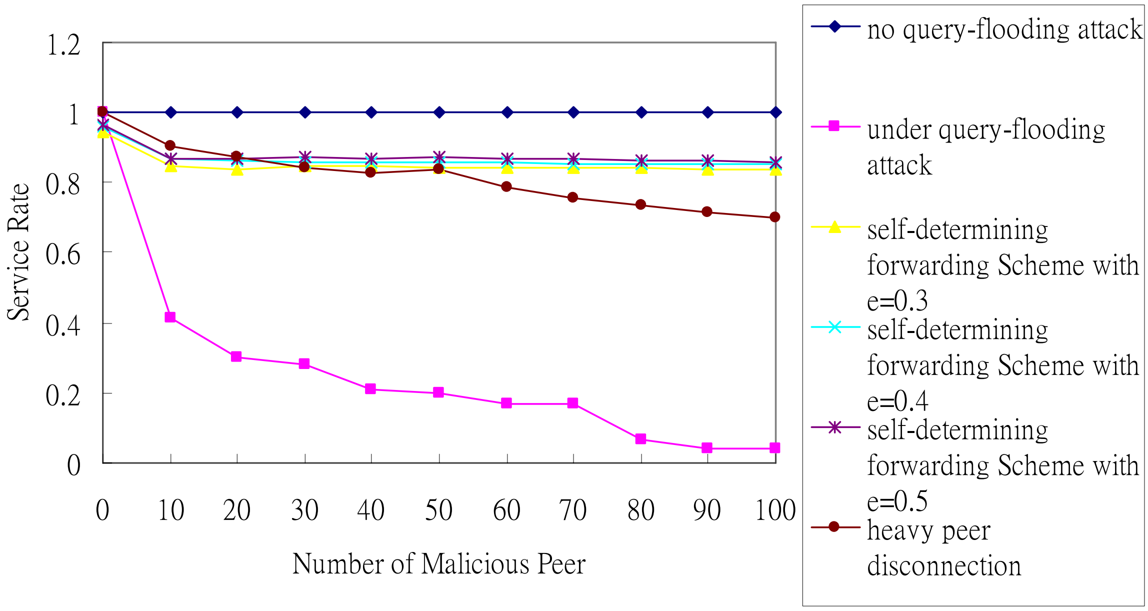

Figure 8: Service rate without cooperation among malicious peers when Scheme II is adopted

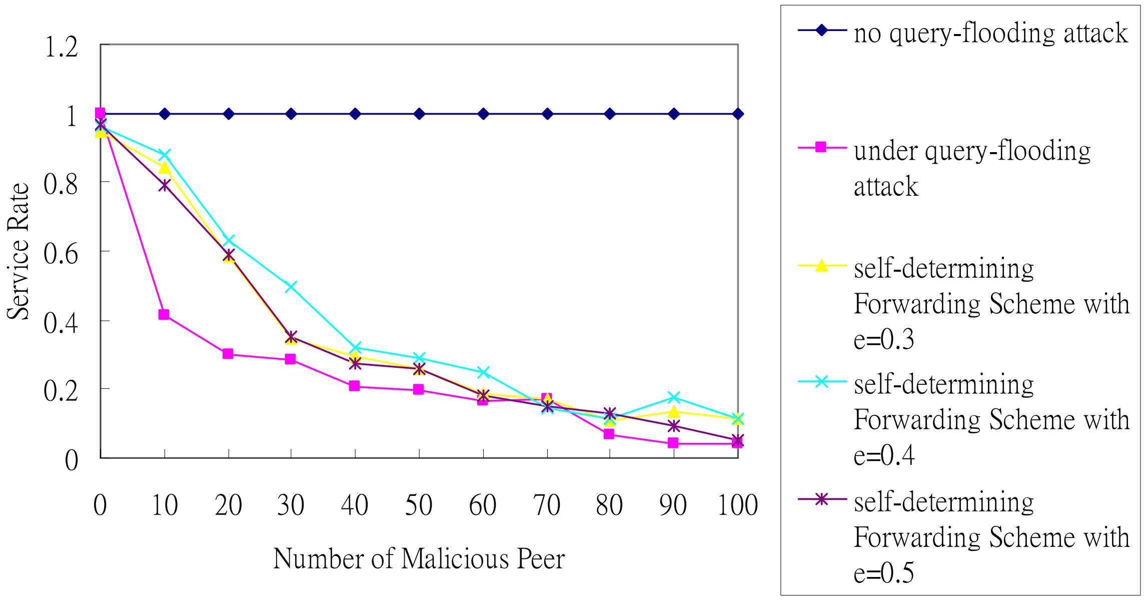

Figure 9: Service rate with cooperation among malicious peers when Scheme II is adopted

Chapter 5

Conclusion

The thesis presents a self-determining message forwarding scheme, a novel approach to defend against query-flooding based DDoS attacks, in unstructured peer-to-peer systems. Since the malicious peers may optimize the use of their own bandwidth to broadcast bogus queries, it is a challenge to defend such attacks. This thesis observes that the behavior of query originators could help determine whether or not a query is bogus and proposes a new scheme to effectively defend against query-flooding based DDoS attacks in unstructured peer-to-peer systems.

The scheme is based on an assumption that each peer has a unique identity which would not be stolen. This can be achieved through key exchange. In the scheme, each file provider should sign receipts to the peers who download files from it. As such, peers that would like to originate queries could attach receipts to show their reasonable behavior. Subsequently, a majority of bogus queries can be dropped and not forwarded in the peer-to-peer system. However, malicious peers will try to cooperate with each other. In other words, they may share identities, so they can use different identities to send lower volumes of bogus queries to prevent detection. Thus, an enhanced scheme is proposed to cluster queries according to receipt originators instead of query originators.

The simulation uses BRITE, which is based on the real-network data, to generate the network topology. The simulation results show a significant improvement in legitimate query service rate during query-flooding based DDoS attacks. Moreover, while legitimate heavy users exist, the service rate still maintains reasonable performance. Thus, the proposed scheme is effective on dealing with the query-flooding issue in peer-to-peer systems.

Bibliography

1 W. Acosta and S. Chandra. Trace Driven Analysis of the Long Term Evolution of Gnutella Peer-to-Peer Traffic. Proceedings of the 8th International Conference on Passive and Active Measurement (PAM '07), Louvain-la-neuve, Belgium, April 2007, pp. 42-51.

2 T. Anderson, T. Roscoe, and D. Wetherall. Preventing Internet Denial-of-Service with Capabilities. Proceedings of the 2nd Workshop on Hot Topics in Networks (HotNets II), Cambridge, MA, November 2003, pp. 39-44.

3 A. Belenky and N. Ansari. IP Traceback with Deterministic Packet Marking. IEEE Communications Letters, volume 7, Issue 4, April 2003, pp. 162-164.

4 BRITE. http://www.cs.bu.edu/brite/

5 H. Burch and B. Cheswick. Tracing Anonymous Packets to Their Approximate Source. Proceedings of the 14th Usenix Conference on System Administration, New Orleans, Louisiana, Dec 2000, pp. 319-328.

6 A. C. Snnoeren, C. Partidge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Single-packet IP Traceback. IEEE/ACM Transactions on Networking (TON), Volume 10, Number 6, December 2002, pp. 721-734.

7 A. C. Snoeren, C.Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Hash-based IP Traceback. Proceedings of the ACM SIGCOMM, San Diego, CA, August 2001, pp. 3-14.

8 N. Daswani and H. Garcia-Molina. Query-flood DoS Attacks in Gnutella. Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02), Washington, DC, November, 2002, pp. 181-192.

9 D. Dean, M. Franklin, and A. Stubblefield. An Algebraic Approach to IP Traceback. ACM Transaction on Information and System Security (TISSEC), Volume 5, Issue 2, May 2002, pp. 119-137.

10 T. Doeppner, P. Klein, and A. Koyfman. Using Router Stamping to Identify the Source of IP Packets. Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS '00), Athens, Greece, November 2000, pp. 184-189.

11 P. Ferguson. Network Ingress Filtering: Defending Denial of Service Attacks which Employ IP Source Address Spoofing. RFC 2267, January 1998.

12 K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. Proceedings of the 19th ACM Operating Systems Principles Symposium (SOSP '03), Bolton Landing, NY, October 2003, pp314-329.

13 ICMP Traceback (itrace). IETF working group, http://www.ietf.org/html.charters/itrace-charter.html.

14 J. Ioannidis and S.M. Bellovin. Implementing Pushback: Router-Based Defense against DDoS Attacks. Proceedings of the Network and Distributed Systems Security Symposium (NDSS '02), San Diego, CA, February 2002, pp. 6-8.

15 A. Juels and J. Brainard. Client Puzzles: A Cryptographic Countermeasure against Connection Depletion Attacks. Proceedings of the Networks and Distributed Network System Security Symposium (NDSS '99), San Diego, CA, March 1999, pp. 151-165.

16 R H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. Internet RFC 2104, February 1997.

17 T. Lau and D. Siu. Distributed Ranking over Peer-to-Peer Networks. Proceedings of the 13th International Conference on World Wide Web (WWW '04), New York, NY, May 2004, pp. 356-357.

18 J. Li, B. Loo, J. Hellerstein, F. Kaashoek, D. Karger, and R. Morris. On the Feasibility of Peer-to-Peer Web Indexing and Search. Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, February 2003, pp. 207-215.

19 B. N. Levine, C. Shields and N. B. Margolin. A Survey of Solutions to the Sybil Attack. Technical Report 2006.

20 Patrick P. C. Lee, John C. S. Lui, and David K. Y. Yau. Distributed Collaborative Key Agreement and Authentication Protocols for Dynamic Peer Groups. IEEE/ACM Transactions on Networking (TON), Volume 14, Issue 2, April 2006, pp. 263-276.

21 J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang. SAVE: Source Address Validity Enforcement Protocol. Proceedings of the IEEE INFOCOM, Las Vegas, NV, June 2002, pp. 1557-1566.

22 Y. Liu, X. Liu, C. W, and L. X. Defending P2Ps from Overlay Flooding-Based DDoS. Proceedings of the IEEE International Conferences on Parallel Processing (ICPP '07), Xian, China, September 2007, p. 28.

23 J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the Source. Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP '02), Paris, France, November 2002, pp. 312-321.

24 J. Mirkovic, G. Prier, and P. Reiher. DWARD: A Source-End Defense against Flooding Denial-of-Service Attacks. IEEE transactions on Dependable and Secure Computing, Volume 2, Issue 3, September 2005, pp. 216-232.

25 D. Moore, G. M. Voelker, and S. Savage. Inferring Internet Denial-of-Service Activity. Proceedings of the 10th Conference on USENIX Security Symposium, Washington, DC, August 2001, pp. 9-22.

26 J. Newsome, E. Shi, D. Song and A. Perrig. The Sybil Attack in Sensor Networks: Analysis and Defenses. Proceedings of the Information Processing in Sensor Networks (IPSN '04), Berkeley, CA, April 2004, pp. 259-268.

27 S. Sen and J. Wang. Analyzing Peer-to-Peer Traffic Across Large Networks. IEEE/ACM Transactions on Networking (TON), Volume 12, Issue 2, April 2004, pp. 219-232.

28 S. Saroiu, P. Gummadi, and S. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Proceedings of Multimedia Computing and Networking (MMCN '02), San Jose, January 2002, pp. 156-170.

29 S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Network Support for IP traceback. ACM/IEEE Transactions on Networking (TON), Volume 9, Issue 3, June 2001, pp. 226-237.

30 S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. Proceedings of the ACM SIGCOMM, Stockholm, Sweden, August 2000, pp. 295-306.

31 K. Sripanidkulchai. The Popularity of Gnutella Queries and its Implications on Scalability. White paper, Carnegie Mellon. University, February 2001.

32 D. Song and A. Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. Proceedings of the IEEE INFOCOM, Anchorage, Alaska, April 2001, pp. 878-886.

33 M. Sung and J. X. IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks. Proceedings of the 10th

IEEE International Conference on Network Protocols (ICNP ‘02), Paris, France, November 2002, pp. 302-311.

34 The Gnutella Protocol Specification. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

35 A. Yaar, A. Perrig, and A.Song. Pi: A Path Identification Mechanism to Defend against DDoS Attacks. Proceedings of the IEEE Symposium on Security and Privacy (S&P ‘03), Berkeley, CA, May 2003, pp. 93-107.

36 H. Zhuge, X. Chen and X. Sun. Preferential Walk: Towards Efficient and Scalable Search in Unstructured Peer-to-Peer Networks. Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, May 2005, pp. 882-883.