Malicious Peers Eviction for P2P Overlays

Hatem Ismail
DEEDS Group, TU Darmstadt
Darmstadt, Germany
hayman@deeds.informatik.tu-darmstadt.de

Daniel Germanus
ENX Association
Frankfurt, Germany
daniel.germanus@enx.com

Neeraj Suri
DEEDS Group, TU Darmstadt
Darmstadt, Germany
suri@cs.tu-darmstadt.de

***Abstract*—P2P networks constitute the foundation for many scalable and fault-tolerant applications. These networks may consist of millions of peers due to their decentralized design. While each peer advertises the same service interface, the actual data provided by common large-scale P2P applications often yields an imbalance, i.e., particular peer subsets are more frequently contacted than the rest of the overlay. Such subsets may refer to peers that replicate critical or popular content.**

As a consequence, Localized Attacks (LA) target these subsets and despite using a relatively small amount of attacking malicious peers, LAs severely impair the overall network's reliability.

Hence, we propose a new two-fold LA countermeasure to detect and evict malicious peers. Our countermeasure has been evaluated in a comprehensive simulation experiment study using a generic LA model that covers a wide range of known LAs such as Sybil, Eclipse or different poisoning attacks. The study shows reliability improvements of up to 97% in the presence of LAs, as well as successful evictions for up to 99% of the cases.

***Index Terms*—P2P, DHT, Localized Attack, Eviction, Detection, Security**

I. Introduction

P2P networks gained a wide popularity due to their scalability, reliability and diversity inherited from their distributed design. Hence, P2P networks are a popular platform for data dissemination applications such as file sharing, video streaming, and online gaming [1]–[3].

For some P2P applications, a subset of peers might experience request rates above average by both, legitimate users as well as attackers due to the criticality or popularity of their stored data, centralized services such as authentication hosted by that particular subset, or their considerable storage/bandwidth provision. Moreover, due to the delay intolerance and fast response time constraints for applications such as video streaming or online gaming, a subset of peers is assigned more responsibilities to support the required QoS such as high level peers in tree based P2P streaming [4], or peers promoted as super peers [5]. Such peer subsets represent vital assets that ensure reliable overlay service provision. Unfortunately, various "Localized Attacks" (LA) target specific peer subsets. LAs continuously evolve and can severely degrade an overlay's reliability and performance due to the perturbations they cause using only a comparably small amount of malicious peers. Examples for LAs are Eclipse attacks (EA), Sybil, index poisoning and DDoS [6]–[12].

In our previous work [13]–[15], we proposed an effective mitigation and detection technique that addresses LAs. In addition, a lot of mitigation techniques have been proposed to counter LAs [15]–[19].

However, the existing mitigation techniques lack of generic applicability, as their efficiency is either bound to specific P2P networks or LA variants. Also, to our knowledge no mitigation technique exists that is, generally applicable for a wide range of LA variants, detects malicious peers, and announces the eviction of previously detected malicious peers to benign ones [17], [20], [21].

The absence of an eviction mechanism leaves overlays exposed to severe security threats since malicious peers are only known to a small fraction of peers. As a consequence, malicious peers can effectively exploit benign peers' partial view of the overlay, e.g., by targeting newly joined peers or peers which are inept to detect their malicious behavior. Moreover, malicious peers can adapt to new adversarial variants to overcome particularly fitted detection techniques. Additionally, detection techniques that are only locally executed by individual peers may be rendered useless from an overlay service perspective.

For the aforementioned reason, we emphasize the necessity of proliferating information about detected malicious peers to the majority of peers in the overlay as a precondition for their eviction. To that end, we propose a novel two-fold eviction mechanism based on the formation of distributed quorums which reliably propagate information about malicious peers.

Contributions

- Assessment of a distributed eviction mechanism that is (i) generally applicable to various LA types (ii) able to effectively evict malicious peers from the overlay and thus, restore reliability and performance requirements, and (iii) capable of propagating the existence of malicious peers to the rest of the overlay.
- Development of an LA model that does not assume a specific LA instance and thereby offers generality and extensibility, and provides the required parameter landscape for evaluation of the eviction mechanism.

The evaluation of the proposed mechanism shows high detection and eviction rates for sophisticated LA variants of up to 99% for up to 10% malicious peers in overlay networks of varied sizes.

In Section II, the related work is discussed that addresses the existing LAs and the proposed mitigation techniques. The system model used for evaluation is presented in Section III. The proposed LA model is discussed in Section IV.

Next, the technical details of the proposed eviction mechanism are provided in Section V. Finally, the impact of the proposed LA and the effectiveness of the eviction mechanism are evaluated in Section VI.

II. Related Work

The severity of LAs along with various countermeasures has been addressed in literature. Drawbacks of existing techniques include the limited applicability to specific LA variants, the need for centralized coordinating peers, sophisticated encryption schemes, or their lack of an eviction mechanism.

A mitigation and detection mechanism is proposed in [22] that focuses on removing suspicious peers from the list of possible candidates to contact. However, the mechanism does not address the eviction of malicious peers from benign peers Routing Table (RT). In [17], the authors introduce a network crawler for KAD, a Kademlia based network, for monitoring and detecting malicious peers. However, no eviction technique is introduced.

A detection mechanism for streaming P2P applications using a belief propagation algorithm is introduced in [23]. In [19], a study on the severity of EAs on KAD is conducted with proposing a mitigation technique based on a trusted cryptographic scheme. Nevertheless, in both studies, no eviction is proposed in addition to the usage of a centralized approach.

In [20], the authors introduce a mitigation mechanism for LAs based on assuring multiple, disjoint paths during lookup initiation. However, the proposed mitigation mechanism uses a cryptographic scheme and no malicious removal mechanism is proposed. Similarly, a self-eviction scheme against false routing information is proposed in [21]. In addition to the absence of a propagating criterion for detected malicious peers, the mechanism is based on strong encrypting technology.

In [24], a stochastic detection and removal scheme is proposed as a countermeasure against pollution attacks. However, the mechanism is only applicable for pollution attacks and in P2P live streaming systems. Furthermore, a mitigation scheme against DDoS attacks in P2P overlays via validating membership information is introduced in [25]. Nonetheless, no eviction mechanism was introduced.

In [26], the authors propose a detection scheme against Sybil attacks by calculating trust values for each peer joining the overlay. However, the scheme relies on central entities, specific to a single LA variant and no evaluation is provided.

The authors in [27] highlight the basis of conducting the most commonly launched LAs, such as EAs, publish attack and node insertion attack. The common aspect while launching the aforementioned LA variants is intercepting messages destined to the victim via poisoning benign peers RT. However, the paper proposes no mitigation or eviction mechanism.

Next, as discussed in [27] we address a generalized LA model that constitutes the fundamentals of various existing LAs for evaluating our proposed eviction mechanism.

III. System Model

In this section we refer to the system model that constitutes the fundamentals of our framework. We start by describing the overlay model topology. Afterwards, the main protocol model aspects are described.

A. Overlay Network Model

The network is modeled as a directed graph $D = (P, E)$. P is the set of peers $p \in P$ in the overlay network. We denote that N is the overlay size, thus, $N = |P|$. Distinct peers $p, q \in P$ that maintain a neighbor relationship are represented by $e = (p, q) \in E$.

Set P is further classified as follows: benign peers B, malicious peers M, so that $P = B \cup M$, where $B \cap M = \emptyset$.

Malicious peers $m \in M$: refer to peers being controlled by an attacker and may behave maliciously.

Benign peers $b \in B$: that comply to the P2P protocol specifications and have no adversary intentions. Benign Set B can be further classified into two subsets: (i) Victim peers $v \in V$ which refers to peers targeted by the attacker, (ii) Poisoned peers $o \in O$ refer to benign peers that store or propagate malicious information as a consequence of polluting malicious entries. In this case, $B = O \cup V$.

B. P2P Protocol Model

Our abstraction for structured P2P protocols consists of five different aspects detailed below.

1) Address Space: Peers have a unique assigned identifier referred to as the peer's *key*. Typically, keys are generated from an external feature such as the IP address, MAC address, a serial number, or a random number. Keys usually have a length of $w \in \{128, 160, 192\}$ bits and are mapped onto the overlay's *address space* which is used to address resources, such as peers and addressable data tuples.

2) Distance Function: A distance function is defined for peers on the address space. The distance notion is an important feature for many peer operations and the choice of the distance function differs among P2P protocol implementations. For example, Kademlia [28] makes use of the XOR operation to calculate the common prefix length (CPL) using the bitstring representation of two peers' keys.

3) Routing Table: Each peer maintains a routing table (RT) that contains contact information about neighboring peers. Contact information is a tuple that relates keys of peers with their underlay network information (e.g., IP address and port number). RTs vary among protocols and usually store k contact information tuples of peers in w lists for distance ranges $[2^i, 2^{i+1})$ with $i = 0 \ldots w-1$, and k constant. In order to resolve new contact information a lookup call is initiated.

4) Lookup Mechanism: In case the destination peer p_v for a specific message to be sent by peer p_i is not stored in p_i's RT, a lookup call is initiated to *resolve* p_v's contact information. To initiate a lookup, p_i selects α peers from its RT to query them about p_v. We now describe the two main lookup mechanisms used by structured P2P overlays to handle lookup mechanisms.

Convergent Lookups: A commonly applied design best practice are *convergent lookups*, i.e., peer p_i selects a set of known peers with closest possible distance to p_v, and iteratively queries each of them to either return the contact information or to repeatedly *forward* p_i's lookup request to even closer peers until p_v can either be resolved or the lookup is dropped due to a timeout. Due to the structured nature of the overlay, a convergence guarantees low message overhead with a logarithmic amount of hops for resolving a certain lookup.

Divergent Lookups: In [13]–[15], we propose *divergent lookups* to mitigate attacks that make use of convergent mechanisms. Mainly, divergent lookups restrict contacting peers close to the victim, where the notion of closeness is referred to as the peer's proximity. Meanwhile, in [15], the PASS algorithm efficiently defines the address space range that contains peers with high probability of resolving the contact information of p_v.

Unlike a convergent mechanism which is highly susceptible to certain LAs, divergent ones are resilient to such attacks while providing a comparable performance to convergent mechanism.

5) Proximity: Each peer defines a proximity area. Usually, this is a close by and sparsely populated address space region whose extent depends on the overlay size N and the key length w. We define the proximity as the set of peers that is stored in the lowest index i list of the RT, i.e., all lists with indexes less than i are empty.

IV. Localized Attack Model

This section presents the attack model. It is the basis to assess the central contribution of this paper, i.e., the eviction mechanism, which will be presented afterwards.

The novelty of this attack model is its generality as it covers a wide range of existing LAs, e.g., EA, Sybil, index poisoning attacks and DDoS. Hence, the resilience of the proposed eviction mechanism is validated for a diverse set of LAs.

As discussed in Section II, the severity of LAs correlates with the amount of lookup messages that are intercepted by malicious peers and which are meant to resolve the contact information of the victim peer p_v. Our attack model focuses on different adversarial strategies and behaviors that illustrate the trade-off between immediate attack severity and detection hardness. In our model, the amount of malicious peers and their placement in the overlay are referred to as strategies, whereas the behavior refers to the interaction of malicious peers with benign ones that deviates from the specification of the P2P network's protocol. The next three subsections describe the strategies and the behaviors.

A. Malicious Resources

Although using a large amount of malicious peers might increase the LA's severity, this has also drawbacks in terms of an increased detection probability, as well as higher LA cost.

Recent LA studies [14], [15], [17] indicate that malicious insertions of only 5-10% in terms of the overlay size is sufficient to intercept a large majority of lookups requesting p_v's contact information. Therefore, we focus on that percentage range for malicious insertions in the overlay.

B. Malicious Placement

Now, we discuss the strategy for placing malicious peers to maximize lookup interception. As presented in Section III, various lookup mechanisms may be used by P2P networks.

Depending on the particular lookup mechanism in a P2P network, patterns on the lookup request message forwarding among benign peers that try to resolve the contact information of p_v can be determined by the attacker. Hence, an efficient placement focuses on overlay regions that reveal a higher probability of receiving such lookup requests.

In this paper, a divergent lookup mechanism is considered that spans the whole address space, i.e., lookups are equally probable of being forwarded to any region in the address space. From our previous work in [13], [15], we make use of *divergent Random Walks* mechanism which is based on random peers selection while restricting only peers within p_v's proximity for forwarding lookup requests. This means that placing malicious peers uniformly across the address space yields equal probability that malicious peers, independent of their location in the address space, intercept lookup requests destined to p_v. Such placement provide a full overview about the generality and suitability of the proposed EM.

Once a malicious peer has been placed, it can launch an LA by performing different adversarial behaviors, which are discussed in the next subsection.

C. Adversarial Behaviors

This subsection highlights various adversarial behaviors of malicious peers as a mean to intercept lookup requests addressed to the victim p_v. Lookup message intercepting LAs follow a threefold approach: (i) poison benign peers' RTs, (ii) malicious collusion, (iii) dynamically alternate adversarial behavior. Each behavior is described below.

1) Poisoning benign peers: Depending only on the inserted malicious peers to intercept lookups destined to p_v is suboptimal due to the limited amount of malicious peers. In order to let benign peers unknowingly partake in the LA execution and promote the interception of lookup messages by malicious peers, benign peers RTs are poisoned, i.e., RT entries pointing to p_v are altered to point towards a malicious peer.

Initially, malicious peers propagate a fake reply regarding p_v by (i) pretending to own p_v's contact information, (ii) advertising the contact information of another malicious peer as the destination contact information. Fake replies are a common practice to achieve RT poisoning.

Once a peer with a poisoned entry towards p_v receives a lookup request, it replies with poisoned information that points to a malicious peer, hence, the lookup initiator forwards the lookup request to. Such adversarial behavior is shown to yield severe impact on the overlay as substantiated in Section VI.

2) Malicious collusion: In order to evaluate the performance of the proposed eviction mechanism in the worst case LA scenarios, we assume that malicious peers are capable of

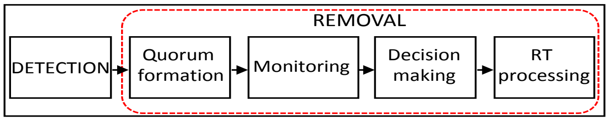


Fig. 1. Eviction process overview

colluding through informing each other about their status. This means that each malicious peer has a periodically updated list about the location and the status (on-line, detected, removed) of other malicious peers.

Moreover, malicious peers are able to exchange messages to inform each other once a peer is detected. As a consequence, malicious peers can alternate between sending fake or correct replies in order to falsify monitoring procedures used by the detection and eviction mechanisms.

3) Alternating behavior: The probability of generating a fake reply is controlled through a configurable parameter for the attacker. We refer to this parameter as FR. Once p_m intercepts a lookup request destined to p_v, p_m sends a fake reply with probability FR to p_i.

In Section VI, we evaluate how FR is a vital parameter for the attacker in terms of detectability and LA severity.

V. Eviction Mechanism

In this section, we describe the technical aspects of the Eviction Mechanism (EM) proposed as a countermeasure against general forms of sophisticated LAs. In order to effectively evict the overlay from malicious peers, an accurate detection scheme must be applied beforehand that allows to detect peers that exhibit possible malicious behavior.

As depicted in Figure 1, the EM is divided into two main blocks, *Detection* and *Removal*. We start by illustrating the detection process that allows peers to locally suspect certain peers based on their lookup replies. Afterwards, the removal process which is responsible for inspecting suspected peers and consequently evict malicious peers is described.

A. Detection Process

The objective of the detection process is to enable peers to locally suspect a given peer according to its lookup reply.

In previous work [14], we introduced a modified lookup mechanism which allows peers to gather more than a single lookup reply in order to compare replies according to (i) the average number of hops for each reply compared to the known average from previous lookups, (ii) the compliance of the replying peers' location with the lookup protocol and (iii) the destination contact information returned in the replies. Once a peer violates any of these detection criteria, this peer is announced to be suspected. The original implementation of the P2P lookup mechanism is based on accepting the first reply that contains the requested information about a given peer p_v. This coerces the lookup initiator p_i to accept the lookup result without being able to validate the results since only a single reply is considered. In turn, whenever a malicious peer receives the lookup request and generates a fake reply, p_i consequently accepts the reply which poisons p_i's RT.

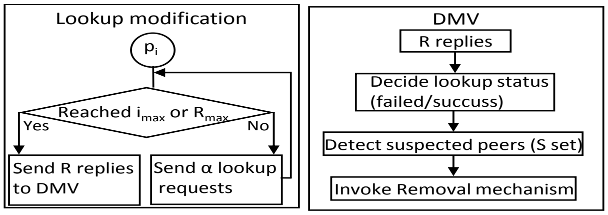


Fig. 2. Detection procedures

The developed modified lookup mechanism which allows p_i to consider a set of replies instead of only the first received reply, is discussed below.

1) Lookup modification: From the original operation of the lookup mechanism, i.e., before introducing the modification, p_i picks α candidate peers from its RT to start forwarding a lookup request for p_v's contact information. Once peer p_r receives such a request, it replies with p_v's contact information if p_r has an entry for p_v in its RT. Otherwise, p_r replies with a list of peers that, according to the routing protocol, have a high chance of knowing p_v. p_i generates α new requests from this list for the next iteration. This process terminates immediately once p_v's address is resolved or i_{max} iterations are reached.

Our lookup modification continues the lookup process until R replies containing p_v's contact information are received or i_{max} iterations are reached, where $R \leq \alpha$. Consequently, p_i will be able to compare multiple replies from different peers and, thus, decide whether certain peers are suspicious.

For this purpose, we make use of a Dynamic Majority Voter [29] (DMV) to process the R received replies. The details of the DMV are discussed below.

2) DMV: From the modified lookup mechanism, peers are compared according to their replies given the aspects described earlier and suspected peers are added to the suspicion set S, where $S \subset R$. Afterwards, the remaining unsuspected peers from the set of R are provided as inputs to the DMV. Note that a suspected reply in this context refers to a suspected peer as replies are only accepted from distinguished peers, i.e., no peer can provide more than a single reply.

The DMV decides whether a valid majority of identical, non-empty replies exists or not. If such majority is found, the lookup reply is considered successful and p_i stores the corresponding contact information. Otherwise, the lookup is declared to be unsuccessful and consequently, p_i initiates a new lookup to resolve p_v's contact information.

In addition, the unmatched minority is added to the suspicious list, i.e., the remaining set of replies that did not constitute the majority. Detailed technical description about the detection process is available in previous work [14].

Once the detection process announces a set of suspected peers, the removal process is invoked to further inspect and accordingly evict malicious peers. Next, the technical concepts of the removal process are described.

B. Removal Process

The basic functionality of the removal process is to further inspect suspected peers and evict peers confirmed to be malicious. In order to do so, a distributed process is required to monitor the suspected peers and then reach a decision about their status. Afterwards, peers that turn out malicious are evicted from peers RT.

The removal process comprises four main procedures. First, *Quorum Formation* defines the criteria of forming a distributed quorum. Second, *Monitoring* handles monitoring the behavior of suspected peers. Third, *Decision Making* is responsible for reaching a decision regarding each suspected peer. Finally, *RT processing* defines the removal criteria about peers confirmed to be malicious.

For more convenience, a list of all abbreviations used within the following sections is provided in Table I.

TABLE I
ACRONYMS DESCRIPTION

Var.	Description	Var.	Description
p_s	Suspected peer	S	Set of suspected peers
p_i	Quorum initiator	p_v	Victim peer
Q	Set of quorum peers	p_q	Peer joined Q
R	number of replied peers	FR	fake reply probability

1) Quorum Formation: Here, we describe how the initiating peer p_i forms a quorum Q, as depicted in Figure 3a. Each process is defined in terms of mechanism and interpretation about the notions behind the development of each process.

Mechanism: Once p_i detects suspicious peers $p_s \in S$ through the DMV, p_i executes the following steps:

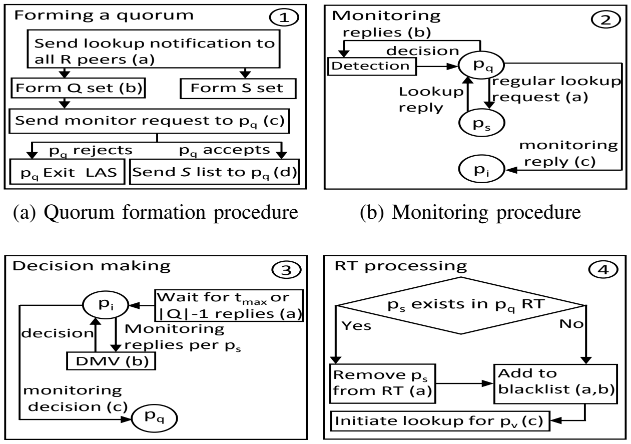
(a) p_i sends a lookup notification to all R peers, recall that R peers contain the set of all peers that replied with p_v's contact information to p_i. A lookup notification is a message containing an acknowledgment that some replies are suspected. The notification message does not convey any information about the identity of the suspected peers, however, its purpose is to alert benign peers about the possibility that they have replied with poisoned entries.

(b) p_i selects only unsuspected peers from the R peers to form a quorum. The set of peers that form a quorum is referred to as Q, including p_i.

(c) p_i sends a quorum joining request to each peer $p_q \in Q$. The joining request contains: (i) a list containing all suspected peers and (ii) a time-stamp that defines when p_q has to start monitoring each p_s.

(d) Each p_q replies to p_i with either an acceptance or rejection to the quorum joining request. p_q may reject a Q_R due to several reasons as p_q might be malicious, already joining another quorum, due to low CPU capabilities or currently experiencing loaded network traffic.

Process interpretation: ***Malicious peers exist in*** Q may provide correct replies according to the FR parameter discussed in Section IV-C. Therefore, colluding malicious peers are capable of informing malicious peers about being suspected.



(a) Quorum formation procedure

(b) Monitoring procedure

(c) Decision making procedure

(d) RT processing procedure

Fig. 3. Removal Procedures

As a countermeasure to restrain suspected malicious peers from changing their behavior if they reveal the identity of any p_q, p_q receives only a list of suspected peers without being informed about other peers in Q or their time-stamps. Malicious peers that are informed about being suspected can behave as follows:

1) Decrease the FR parameter, i.e., behave benignly to avoid being evicted if confirmed malicious.
2) Keep providing fake replies with the same rate, i.e, exploit chances to poison benign peers RT regardless of the risk of being evicted.

Poisoned peers exist in S due to providing suspicious replies to the DMV. For this reason, poisoned peers make use of the notification message to restore their benign state. Once a poisoned peer receives a notification message, it initiates a lookup requesting p_v's contact information in order to re-evaluate its reply to p_i. After a poisoned peer proves its benign state, it receives the suspected set S without joining Q, i.e., it can monitor peers in S and accordingly removes malicious peers based only on its own decision. The main advantage is that poisoned peers are able to restore p_v's correct contact information in addition to removing other malicious entries that might exist in their RT, which allows for higher removal rate of malicious peers.

Churning peers in Q refer to either benign or malicious peers that might leave the overlay or do not complete the removal procedures. Due to the small number of peers that forms a quorum, procedures are executed over a short time frame. Hence, the churning amount of peers do not interrupt the removal process or deviate the final decision of the quorum as evaluated in Section VI.

2) Monitoring procedure: This procedure defines the monitoring basis that p_q follows with each p_s and how results are gathered at p_i, as shown in Figure 3b.

Mechanism: Once p_q accepts joining quorum Q, the following steps are executed in parallel $|S|$ times by p_q.

(a) p_q sends a regular lookup request, using the modified lookup mechanism, to each $p_s \in S$ requesting p_v's contact information according to the time-stamp assigned by p_i in the quorum joining request. Note that each p_q sends a single lookup request to p_s, however, in order to be able to decide about p_s, it sends a lookup request to other $\alpha - 1$ peers, i.e., modified lookup mechanism.

(b) After p_q processes the received replies via the DMV, p_q decides accordingly whether p_s is malicious or not.

(c) p_q sends a monitoring reply to p_i containing the DMV decision about each p_s, i.e., a boolean that depicts the monitoring result.

Process interpretation: The purpose of using regular lookups is to provide anonymity to the monitoring procedure so as malicious peers can not differentiate between lookups intercepted from normal peers and those intercepted from p_q. Moreover, p_i assigns different time-stamps to each p_q so that p_s does not receive multiple lookup requests simultaneously more than its expected in-going bound to assure p_s does not detect any abnormal behavior.

Through the detection mechanism, p_i uses the DMV to process lookup replies that include p_s's reply, and hence, decide whether p_s is malicious or not. Notably, if the DMV does not suspect p_s, this might be inferred that p_s was in a poisonous state.

3) Decision making procedure: In this procedure, we discuss the procedure that allows to reach a decision about p_s, as illustrated in Figure 3c.

Mechanism: p_i reaches a decision about p_s through executing the following steps:

(a) p_i waits for either timeout t_{max} or receives $|Q| - 1$ monitoring replies.

(b) p_i inputs the received monitoring replies about each $p_s \in S$ separately to its DMV.

(c) Finally, p_i sends the DMV results to each $p_q \in Q$, we refer to this message as "monitoring decision reply".

Process interpretation: Timeout t_{max} assures malicious peers do not block the removal process since a malicious peer in Q might not send a monitoring reply to p_i. t_{max} is set according to the latest time-stamp assigned to any peer in Q.

The monitoring decision message includes p_i's decision about each p_s, i.e., p_s is malicious or was poisoned with malicious entry. Consequently, each $p_q \in Q$, proceeds to the "RT processing" procedure.

4) RT processing: Here we define the criteria of removing malicious peers from RTs as depicted in Figure 3d. Moreover, we highlight how information about malicious peers are propagated through the overlay.

Mechanism: After p_q compares the received decision from p_i about each p_s with its own decision, the following steps are executed when p_q decides to proceed with removing p_s. Note that different comparison cases are discussed in details in the procedure interpretation.

(a) In case p_s exists in p_q's RT, p_q removes p_s and adds it to a blacklist to assure no further contact with p_s.

(b) Otherwise, in case p_s does not exist in p_q's RT, p_s is added to p_q's blacklist to hinder adding p_s in its RT in the future.

(c) p_q initiates a new lookup for p_v's contact information. During different lookup iterations, no replies or candidates suggestions will be accepted if such peers exist in p_q's blacklist.

Process interpretation: p_q compares the monitoring decision received from p_i about each p_s with its own decision about p_s concluded during the monitoring procedure, i.e., p_q checks whether both decisions match or not.

Recall that p_q's possible decisions about a suspected peer are: 1) Benign, 2) Suspicious. Similarly, for p_i, possible states reported in the monitoring decision reply are: 1) Poisoned, 2) Malicious. For simplicity, we state here the consequent decisions according to the possible combinations.

p_i: Poisoned, p_q: Benign: In this case, p_i states that p_s was poisoned, which confirms p_q's status that p_s is benign. As a result, no further action is taken about p_s.

p_i: Poisoned, p_q: Suspicious: This denotes that either p_s was poisoned during the monitoring period of p_q, or p_s is malicious but the majority of monitoring replies was correct due to low FR at p_s. At this point, p_q initiates new lookup request to p_s asking for p_v's contact information to decide whether to remove p_s or not before deciding to initiate a quorum to monitor p_i.

p_i: Malicious, p_q: Benign: This case refers to p_s being malicious according to the majority of monitoring replies reported to p_i. However, due to low FR, p_q reached to a decision that p_s is benign. p_q initiates a new lookup in order to re-monitor p_s before deciding to start monitoring p_i. In fact such approach assures malicious peers do not exploit the removal procedure. Meanwhile, p_i further proceeds to the removal steps.

p_i: Malicious, p_q: Suspicious: As both p_i and p_q confirms p_s being malicious, p_i and p_q proceed to the removal steps.

VI. Evaluation

In this section, we provide an evaluation of the proposed EM against general attack model that constitutes the basis of various specific LAs. We first start with case study 1 "LA impact" that evaluates the impact of launching a severe LA on a P2P overlay. Case study 2 "EM evaluation" assesses the performance and effectiveness of EM.

First of all, the simulation environment, parameters and metrics used for evaluation are introduced. Afterwards, each case study is presented with results discussion and interpretation. Finally, a summary that highlights the main results and conclusion about EM is provided.

A. Simulation environment

Case studies were conducted using the OMNeT++ simulator [30] and OverSim [31] which provides various P2P protocol implementations. Each simulation experiment was running for 4 hours. Moreover, for confidence interval measurements, each simulation was scheduled for 10 repetitions. In Table II, the simulation parameters used in the experiments are provided.

B. Simulation model

In order to validate the scalability of our approach, EM is assessed using different overlay sizes, i.e., $N = 5k, 10k, 20k, 30k$. Different malicious insertion ratios $MI = 5\%, 10\%$ and fake reply probabilities $FR = 50\%, 80\%$ are

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Maximum iterations (i_{max})	10
Number of victims ($\|V\|$)	1
Maximum received lookup replies (α)	9
Key length (w)	128
Lookup	Divergent Random Walks [13]
Malicious Insertion ratio (MI)	$5\%, 10\%$
Overlay size (N)	$5k, 10k, 20k, 30k$
Fake Reply probability (FR)	$50\%, 80\%$

used to represent the impact of varying amounts of malicious peers with different probabilities of generating fake replies, resulting in 16 different overlay configurations.

1) System workload: Our target is to base our evaluation on launching an LA on P2P networks with special set of peers that are more frequently contacted and offer special services to the overlay. For this reason, simulations are based on a "Service Overlay Network" where 80% of lookup requests are addressed to the victim. In general, lookups are sent on average every 10 seconds with 5 seconds standard deviation.

2) Simulation Churn Models: In order to simulate churning rate of peers, a **Pareto (P-500)** is used where the average life-time and dead-time of peers is 500 seconds. The choice of such distribution is due to the realistic experimental results provided for P2P overlays in [32].

C. Evaluation Metrics

1) ***Lookup Success Ratio*** *(LSR):* the ratio of successful lookups to the total number of lookups initiated to the victim only. LSR assesses the reliability of the network.

2) ***Message Complexity*** *(MC):* the overhead exerted on the system due to lookups initiation, malicious existence and EM procedures execution.

3) ***Poisoned Replies*** *(PR):* the average number of poisoned replies per lookup. This metric is used to assess the impact of poisoning benign peers RT on the victim's service provision.

4) ***Malicious ratio per RT*** *(MRT):* the average ratio of malicious entries in benign peers RT. This metric evaluates the impact of malicious peers insertions.

D. Case Study 1: LA Impact

In this study, we assess the impact of launching an LA with the proposed adversarial behaviors discussed in Section IV and how poisoning benign peers RT can severely degrade the system's reliability.

Discussion: As shown in Figure 4a, LSR shows negligible values that average below 1% due to LA impact, i.e., more than 99% of lookups initiated to resolve p_v's contact information fail. MC overhead is depicted in Figure 4b where values are in the range of 8 to 13. For $MI = 10\%$, MC is slightly lower, 8-9 messages, compared to values observed for $MI = 5\%$.

In Figure 4c, the average number of poisonous replies per lookup is remarkably high as ranges are between 78% and 90%. PR values For $MI = 5\%$ are higher than in $MI = 10\%$ with an average of about 6%. Finally, Figure 4d depicts the average MRT. For $MI = 5\%$, values ranges between 20%-22%. For $MI = 10\%$, higher existence of malicious peers is observed in peers RT where values average between 25%-26%.

Interpreting the results: Due to the malicious existence and poisoned peers which propagate malicious information, lookup requests are almost completely intercepted as depicted in Figure 4a. Consequently, p_v's service provision is markedly degraded as 99% of lookups initiated to p_v fail.

From [13], the average MC overhead using divergent Random Walk mechanism is in the range of 11-13. However, as indicated for particular LA configurations, MC indicates less overhead as indicated in Figure 4b. The reason is that benign peers' RT entries pointing to p_v are poisoned with malicious peers that fake storing p_v's contact information. Accordingly, such peers reply with malicious information and the whole lookup is *falsely* resolved in the first or second iteration at most which lowers MC value. For the same reason, MC is lower for $MI = 10\%$ as more malicious peers intercept the lookup request than for $MI = 5\%$ and thus, reply with fake replies which accelerates collecting α replies which terminates the lookup process. In fact, such abnormal MC is one of the criteria used by the detection mechanism to suspect malicious peers.

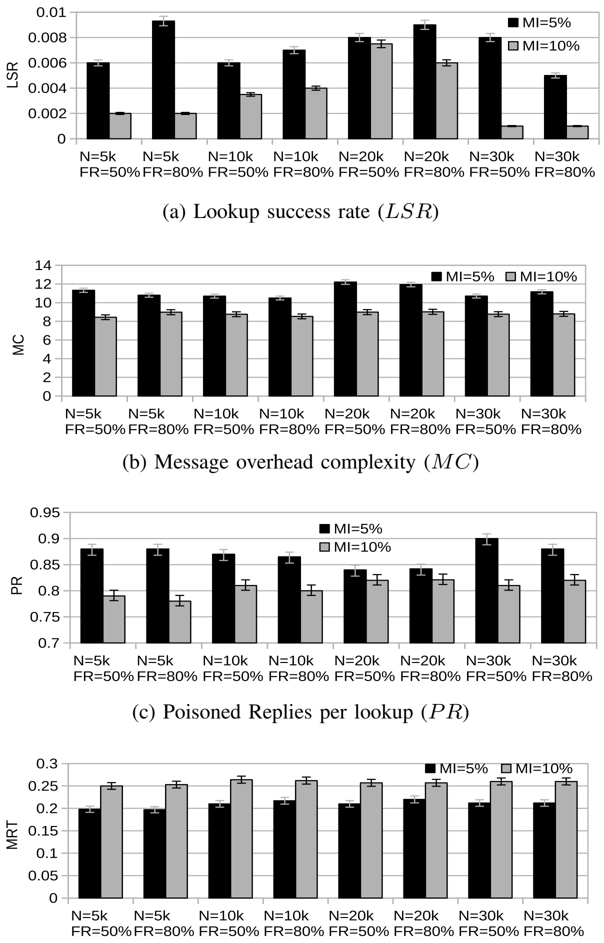
PR highlights the impact of poisoning benign peers entries towards p_v as an application of the adversarial behavior of malicious peers as discussed in Section IV. As shown in Figure 4c, a smaller amount of malicious peers as $MI = 5\%$ yields a higher probability for poisoned peers to receive lookup requests than for $MI = 10\%$ where a larger amount of malicious peers intercept the lookup request. Regardless of the selected value of FR, a large fraction of benign peers is poisoned due to the severe impact caused by small MI. Such a large fraction of poisoned replies per lookup is the main reason of the resulting severe degradation in LSR.

As illustrated in Figure 4d, malicious peer insertions in the range of 5%-10% is capable of polluting 20%-26% of benign peers RT. The reason for that is the propagation of entries pointing to malicious peers. Consequently, whenever a peer selects α different peers from its RT for the first lookup iteration, or replies to a lookup request with a list of possible candidates, malicious peers are selected with high probability. The severity of launching LA various forms can be concluded from the small amount of required malicious insertion to completely intercept messages destined to the victim.

E. Case Study 2: EM Evaluation

Now, we evaluate the performance of the proposed EM in terms of the detection and propagation effectiveness. For comparability reasons, the same set of metrics and evaluation criteria are used.

Results discussion: As depicted in Figure 5a, activation of EM results in high LSR rates in the range of 90% to 97%. LSR values are relatively comparable to the case where lookup requests are not intercepted by malicious peers through different iterations in [13].



(a) Lookup success rate (LSR)

(b) Message overhead complexity (MC)

(c) Poisoned Replies per lookup (PR)

(d) Malicious ratio per RT (MRT)

Fig. 4. LA impact

Figure 5b provides the average message overhead exerted by EM. Measurements average between 41 and 53 which is higher than MC values provided in case study one.

In Figure 5c, the average number of poisoned peers is provided. PR averages between 2%-7%, which show a significant decay compared to case study one.

As shown in Figure 5d, the average number of malicious peers per RT remarkably decreases as values average between 0.2%-0.6% regardless of MI. Figure 5e depicts the average ratio of peers per Q. Note that data collection time starts at $t = 300s$ and the EM is set to be triggered after 1800 sec. The ratio of benign peers increases from 10% to 90% as the EM is continuously triggered while malicious peers ratio decreases from 72% to 3%.

Interpreting the results: A significant increase in LSR is shown in Figure 5a due to the effect of EM. As malicious peers are evicted and poisoned peers are restoring correct entries towards p_v, the number of successful lookups increases. This denotes that the reliability of p_v's service provision can be effectively restored when EM is activated.

As shown in Figure 5b, MC during the different removal procedures depicts reasonable overhead given the amount of lookups initiated and the restored reliability. As poisoned peers continuously attain correct RT entry towards p_v and malicious peers are evicted due to EM, the number of suspected peers in S decreases. Consequently, less overhead is exerted on the overlay due to EM, which is the reason MC maintains a steady average even when MI increases. Given that the number of peers in Q depends on α, MC decreases when choosing less value for α.

As depicted in Figure 5c, PR decreases as an effect of the notification message sent to peers during the quorum formation procedure. Besides, less peers are subject to poisoning as EM evicts malicious peers from the overlay. Accordingly, more poisoned peers are able to restore their benign status.

Moreover, EM allows poisoned peers to recover independent of the FR parameter as the large number of initiated quorums allows a large fraction of poisoned peers to exist in S, although FR determines the ratio of malicious to poisoned peers that might be suspected by the detector. The number of quorums initiated when a high detection rate is observed reaches up to 96% from the total number of initiated lookups. Once a large fraction of malicious peers are evicted from the overlay, average quorum initiated settles around 0.6%.

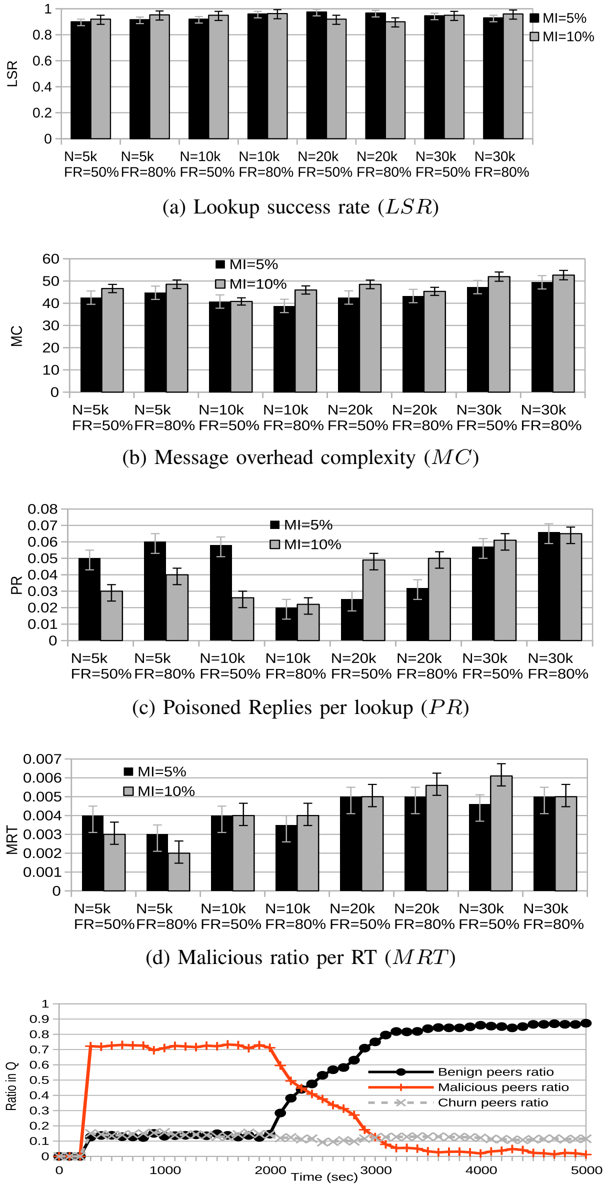
As illustrated in Figure 5d, MRT decreases as a result of evicting malicious peers at high rates as 96% of lookups initiated trigger EM. In addition, malicious peers are effectively evicted since the monitoring procedure is designed such that malicious peers are not aware of the monitoring peers or the monitoring timings, which yields no gain for malicious peers to lower FR value.

As shown in Figure 5e, the ratio of benign peers continuously increases in Q due to the effect of malicious peers being evicted and poisoned peers restoring their benign status. Subsequently, malicious peers existence in the quorum decreases around 0.5%. Regarding churning peers, out of $\alpha = 9$ peers that may form a quorum, the number of churning peers average around one peer per Q due to the effect of P-500 churn model. Also, as malicious peers target maximizing lookups interception, their in-going bound expects high rate of receiving lookup requests. For this reason, removal procedures are executed in a small time-frame as time-stamps are closely assigned to quorum peers and hence, churning peers do not affect the eviction process.

F. Summary

In case study one, it was shown that LAs severely impact on the overlay as LSR values drop below 1% due to malicious interception. Moreover, PR significantly increases to 90% while MRT averages between 20%-26%.

A remarkable enhancement in the overlay's performance due to EM is observed as LSR values increase to 97% while PR and MRT dropped to 2% and 0.2%, respectively. Simultaneously, MC values average around 41-53 message for different network sizes and various malicious insertions. EM stable performance on different overlay size yields its ability to be deployed in large-scale applications that host thousands-millions of users.



(a) Lookup success rate (LSR)

(b) Message overhead complexity (MC)

(c) Poisoned Replies per lookup (PR)

(d) Malicious ratio per RT (MRT)

(e) Peers ratio in Q

Fig. 5. EM performance

VII. Conclusion & Future Work

In this work [1], we propose a malicious Eviction Mechanism (EM) for P2P overlays that can efficiently perform against various LA models using decentralized quorums to monitor, decide and accordingly propagate and evict malicious peers. EM shows in comprehensive simulation experiment studies a high malicious removal rate of up to 99% while restoring the overlays reliability to 97%. The studies were conducted using a generic LA model that includes an established attack variety including Sybil, Eclipse attack, and more.

As on-going work, we are evaluating the performance of EM on out-going LAs on super-P2P networks, where malicious peers target intercepting out-going messages from super peers. Moreover, we plan to assess the EM performance on real P2P networks using Planet-lab.

References

[1] N. Good, et. al, "Usability and Privacy: A Study of Kazaa P2P File-sharing," *In Proc. SIGCHI*, pp. 137–144, 2003.

[2] N. Ramzan, et. al, "Video streaming over P2P networks: Challenges and opportunities," *Signal Process-Image*, vol. 27, pp. 401–411, 2012.

[3] L.Fan, et. al, "Design issues for peer-to-peer massively multiplayer online games," *In Proc. IJAMC*, vol. 4, pp. 108–125, 2010.

[4] C. Liang, et al., "Topology optimization in multi-tree based P2P streaming system," *In Proc. ICTAI*, pp. 806–813, 2009.

[5] A. Yahyavi, et. al, "Peer-to-peer Architectures for Massively Multiplayer Online Games: A Survey," *In Proc. CSUR*, vol. 46, pp. 9:1–9:51, 2013.

[6] A. Singh et al., "Defending against Eclipse Attacks on Overlay Networks," *In Proc. SIGOPS*, pp. 115–120, 2004.

[7] J. Dinger, et. al, "Defending the sybil attack in p2p networks: Taxonomy, challenges, and a proposal for self-registration," *ARES*, p. 8, 2006.

[8] J. Kong, et. al, "The evaluation of index poisoning in bittorrent," *In Proc. ICCSN*, pp. 382–386, 2010.

[9] J. Liang, et. al, "The Index Poisoning Attack in P2P File Sharing Systems," *In Proc. INFOCOM*, pp. 1–12, 2006.

[10] S. Zargar, et al., "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 2046–2069, 2013.

[11] X. Sun, et. al, "On the feasibility of exploiting P2P systems to launch DDoS attacks," *Peer Peer Netw Appl*, vol. 3, pp. 36–51, 2010.

[12] H. Koo, et. al, "A DDoS attack by flooding normal control messages in Kad P2P networks," *In Proc. ICACT*, pp. 213–216, 2012.

[13] D. Germanus et al., "Mitigating Eclipse Attacks in Peer-to-Peer Networks," *In Proc. CNS*, 2014.

[14] H. Ismail, et. al, "Detecting and Mitigating P2P Eclipse Attacks," *In Proc. ICPADS*, 2015.

[15] D. Germanus, et. al, "PASS: An Address Space Slicing Framework for P2P Eclipse Attack Mitigation," *In Proc. SRDS*, 2015.

[16] A. Singh et al., "Eclipse Attacks on Overlay Networks: Threats and Defenses," *In Proc. INFOCOM*, pp. 1–12, 2006.

[17] T. Cholez et al., "Detection and Mitigation of Localized Attacks in a widely Deployed P2P Network," *Peer Peer Netw Appl*, pp. 155–174, 2013.

[18] Q. Li, et. al, "An Enhanced Kad Protocol Resistant to Eclipse Attacks," *In Proc. NAS*, pp. 83–87, 2014.

[19] L. Maccari, et. al, "Avoiding eclipse attacks on Kad/Kademlia: an identity based approach," *In Proc. ICC*, 2009.

[20] I. Baumgart, et. al, "S/Kademlia: A practicable Approach towards Secure Key-based Routing," *In Proc. ICPADS*, pp. 1–8, 2007.

[21] C. Lu, et al., "A Safety Algorithm of P2P Routing based on Multiple-Encryption Detecting Technology," *In IAES*, pp. 5815–5823, 2013.

[22] T. Cholez, et al., "Efficient DHT attack mitigation through peers' ID distribution," *In Proc. IPDPSW*, pp. 1–8, 2010.

[23] R. Gaeta, et al., "Identification of malicious nodes in peer-to-peer streaming: A belief propagation-based technique," *In Proc. TPDS*, vol. 24, no. 10, pp. 1994–2003, 2013.

[24] Y. Li et al., "Stochastic analysis of a randomized detection algorithm for pollution attack in P2P live streaming systems," *Performance Evaluation*, pp. 1273–1288, 2010.

[25] X. Sun et al., "Preventing DDoS attacks on internet servers exploiting P2P systems," *Computer Networks*, vol. 54, pp. 2756–2774, 2010.

[26] D. Sharma et al., "Performance Analysis of Sybil Decline: Attack Detection and Removal Mechanism in Social Network," *In Proc. IJCSIS*, vol. 13, p. 165, 2015.

[27] T. Locher et al., "Poisoning the Kad network," *In Proc. LNCS*, vol. 5935, pp. 195–206, 2010.

[28] e. a. P. Maymounkov, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," *In Proc. IPTPS*, pp. 53–65, 2002.

[29] F. Belli, et. al, "Comparative Analysis of Concurrent Fault-Tolerance Techniques for Real-Time Applications," *In Proc. ISSRE*, 1991.

[30] G. Pongor, "OMNeT: Objective Modular Network Testbed," *In Proc. MASCOTS*, pp. 323–326, 1993.

[31] I. Baumgart et al., "OverSim: A Flexible Overlay Network Simulation Framework," *In Proc. INFOCOM*, pp. 79–84, 2007.

[32] Y. Zhongmei, et al., "Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks," *ICNP*, pp. 32–41, 2006.

[1] Research supported in part by EC H2020-CIPSEC GA #700378