[Article] Robustness analysis of an unstructured overlay for media communication

Marchetto G., Papa Manzillo M., Torrero L., Ciminiera L., Risso F.

"*Robustness analysis of an unstructured overlay for media communication*"

Published in

IET Communications, vol. 5, n. 4, pp. 409-417

The final published version is accessible from here:

http://dx.doi.org/10.1049/iet-com.2010.0624

Robustness Analysis of an Unstructured Overlay for Media Communication

Guido Marchetto, Marco Papa Manzillo, Livio Torrero
Luigi Ciminiera, Fulvio Risso

Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi, 24 Torino, Italy
E-mail: guido.marchetto@polito.it

August 25, 2010

Abstract

The wide diffusion of NATs (and, in some respect, firewalls) may prevent some applications that require direct end-to-end connectivity (e.g., real-time media) from being able to connect to the remote party. While the solutions currently adopted rely on centralized nodes as third party relays, the DISCOS architecture has been recently proposed and aims at distributing such functionalities across a peer-to-peer overlay. The original paper presented some performance characteristics of the overlay, but the ability to resist to both failures and attacks was not taken into consideration. This paper illustrates the robustness feature of the DISCOS overlay and suggests some minor modifications to the original mechanisms, in order to improve the overall robustness. The key component of DISCOS is its dynamic scale-free topology. Hence, the paper also extends the existing literature concerning the robustness of scale-free networks, which

considers only static graphs.

1 Introduction

The wide diffusion of Network Address Translators (NATs) and (in some respect) firewalls deeply changed networks behavior, thus preventing some applications that require direct end-to-end connectivity (e.g., real-time media) from being able to connect to the remote party. Indeed when two nodes want to establish a media session, both of them can be the initiator: if the session is started by the node outside the network behind NAT, its establishment may fail. To overcome this issue, ad-hoc NAT traversal techniques [1] have been developed. These techniques aim at establishing communication across NATs between two hosts. The *hole punching* [1] aims at establishing direct communication (i.e. without intermediary servers) between nodes. For this technique to be effective, supporting nodes called *rendezvous servers* are needed during session establishment. If hole punching fails, *relaying* [1] is used as alternative. This technique consists in exchanging all the traffic through a centralized relay server. Due to their centralized nature, both rendezvous servers and relays can be considered as single points of failure: the media session fails if they become unavailable. In addition, relays have to forward the traffic related to all the relayed sessions, thus requiring a huge amount of computational resources, which represent an additional scalability problem. DISCOS (DIStributed COnnectivity Service) [2], which has been developed to overcome these issues, integrates rendezvous and relaying functionalities directly into user nodes placed in the open Internet. These nodes organize themselves into a peer-to-peer overlay to share the load due to relayed sessions and to ensure quickly discovery of a rendezvous server or relay when needed. Vice versa, the user nodes with limited connectivity (e.g., behind a NAT) do not participate in the overlay and must only be able to locate an available rendezvous server or relay among the participating peers.

DISCOS has been defined and validated through simulations in an environ-

ment where media sessions are controlled using the Session Initiation Protocol (SIP) [3]. Figure 1 shows such deployment scenario, where user nodes are called User Agents according to the SIP terminology. To underline the double role played by the nodes in the overlay, the peers have been marked as User Agents and *connectivity peers*: as User Agents the nodes are ordinary nodes establishing media sessions; as connectivity peers they are rendezvous and/or relays providing service to other user nodes with limited connectivity. [2] demonstrates that the effectiveness of DISCOS is due to its particular overlay topology. In fact, connectivity peers organize themselves into a scale-free network [4], a topology that can ensure efficient resource lookups if properly exploited.

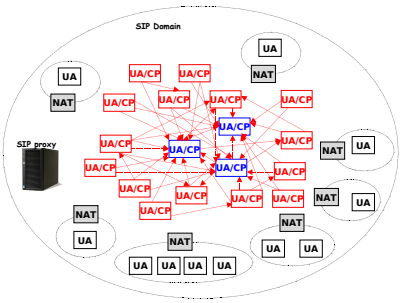

Figure 1: DISCOS overlay topology.

Although an extensive literature (e.g., [5–20]) exists about the robustness of P2P networks, either structured or unstructured, some peculiarities of DISCOS make this technology not completely covered by these studies. First of all, in DISCOS nodes offering the service (i.e., the connectivity peers) are equivalent. Hence, a querying node is interested in locating one of the many available connectivity peers in the shortest time, rather than in locating a specific rare resource. This influences the evaluation of the robustness of the system, which in case of node failures or when attacked by malicious users should continue to ensure these features. Furthermore, DISCOS is specifically designed to result in a scale-free network to offer good lookup performance, thus opening the path for specific attacks aiming at destroying this particular topology or, even more

important, at exploiting this topology to disrupt the service.

This paper focuses on these aspects by investigating the robustness of the DISCOS overlay. Both node failures and malicious actions of some participating peers are considered, in order to analyze how both the overlay topology (and consequently its lookup performance) and the offered connectivity service are affected by these events, which may be common in real networks. We introduce scenarios where peers are selectively removed or where the overlay is under attacks carried out by malicious peers. In particular, we define a Sybil attack where malicious peers try to pollute the overlay data with false information. Our analysis still focuses on a SIP infrastructure, where DISCOS finds its natural deployment, and aims at investigating how the described events affect its operation in terms of connectivity peer lookup performance and media session failure. Although, in general, the robustness of the original mechanisms is good, some of the results presented in our analysis have led to the modification of such mechanisms, in order to improve the resulting overlay performance.

The paper is organized as follows. We first provide some background information on the DISCOS architecture (Section 2). Then, we briefly review the existing works related to the analyzed topic (Section 3), also describing how DISCOS peculiarities are not completely covered in the available literature on P2P and scale-free robustness. Section 4 presents our analysis methodology and provides some information concerning the adopted simulator. Section 5 specifically discusses the robustness of DISCOS to failures, while Section 6 focuses on attacks. Finally, Section 7 concludes the paper.

2 DISCOS Background

DISCOS (DIStributed COnnectivity Service) [2] is an unstructured gossip-based P2P system for providing connectivity service to nodes with limited connectivity. The decision to adopt an unstructured network was driven by the purpose of the overlay, which is devoted to locate one among the many equivalent peers

offering the service, rather than a specific rare resource for which a structured network may be more appropriate [21]. This section provides a brief overview of DISCOS, firstly focusing on the selection of the overlay topology (i.e., a scale-free) and then presenting its operating principles.

2.1 Overlay topology

The DISCOS overlay is designed to result in a scale-free topology. In such network a few peers called *hubs* have a high in-degree (i.e., they are known by many peers), while the others have a low degree and are usually referred to as *leaves*. In particular, scale-free graphs are characterized by a power-law distribution of the node degree, i.e., the probability for a node to have degree k is given by $P(k) = ck^{-\gamma}$, where c and γ are positive constants.

Results presented in [2] show how this overlay topology offers interesting properties when adopted to provide a connectivity service. First of all, scale free networks feature a short average path length between nodes, thus ensuring an adequate spread of information through flooding with limited depth. This is fundamental in gossip-based overlays to offer good lookup performance together with limited message overhead on the network. In addition, the topology helps in reducing the number of connectivity peers queried by the nodes behind NAT before locating an available servant. Indeed, a connectivity peer already servicing many nodes simultaneously does not accept further requests and has to redirect queries to other connectivity peers it knows. Obviously, the larger the number of known peers, the higher the probability that it can suggest an available one. However, nodes can only maintain a neighbor cache that should be kept small in order to reduce the overhead due to its management. This limits the number of peers known at each instant. [2] describes how a high frequency in cache update can solve this issue in a scale-free network. Hubs receive a large number of gossip messages (often called advertisement messages because their purpose is to advertise the existence of peers in the overlay) as they are the most popular nodes: if these advertisements contain information about lightly

loaded peers (i.e., the leaves), hubs will be able to suggest many different available peers during time. If nodes with limited connectivity start querying the hubs first, the number of contacted peers required to locate an available servant can be significantly reduced.

These properties confirm the results obtained by Adamic et al. [22], which showed how searches in a scale-free topology are extremely scalable and that searches of specific content towards hubs perform better than using pure random walks. DISCOS generalizes these results to the case of a service-oriented overlay where a single "resource" (i.e. the relaying/rendezvous functionality) is provided by many nodes simultaneously.

The described design choices lead the hubs to receive a large portion of the overall advertisement traffic. However, results presented in [2] show that a small advertisement rate is sufficient to achieve high lookup performance. This leads each node to process no more than 48 messages per minute, which is a negligible overhead for modern home-PCs. This is obtained thanks to both the type of service deployed (i.e., the connectivity service, where all peers are equivalent) and the high availability of possible servants at the hubs.

2.2 Operating principles

When a node connects to the network, it implements a mechanism that determines if it can become a connectivity peer (i.e., the peer resides into the public Internet) or it will be a simple client of the overlay. A connectivity peer that joins the DISCOS overlay for the first time will look for a list of active peers registered into a Boostrap Service and it will update its cache of known peers with this information. If the Bootstrap Service is unable to provide any active peer or if the number of peers returned is too low, the joining peer registers itself in the Bootstrap Service. Entries expire after 60 minutes in order to reduce the presence of failed or left nodes in the table. If a peer rejoins the overlay after an offline period, it contacts the Bootstrap Service only if none of the peers it knows from its last visit is still active.

The Bootstrap Service is implemented as a set of multiple bootstrap servers reachable through appropriate DNS SRV entries configured in the DNS. Each bootstrap server stores information about some connectivity peers that register themselves according to the above described policies. Multiple bootstrap servers are deployed for redundancy and load balancing purposes. [2] also describes how different Bootstrap Services can be used to create disjoint overlays because joining peers that fetch nodes from different Bootstrap Services start to exchange advertisement messages with different connectivity peers. This enables the possibility of deploying different DISCOS overlays in different geographical areas of a SIP domain. If a location- aware Bootstrap Service selection policy is adopted (obtained for example by proper DNS configurations), users can find a connectivity peer that is close to them, thus preserving the user-relay latency achieved by current centralized solutions, where different servers can be used at different locations.

After the bootstrap procedure, the new peer starts sending advertisement messages to the known peers. This message flooding aims at two purposes: (*i*) it allows the sender to announce itself to other peers and (*ii*) it enables the sender to share information about the peers in its cache with the rest of the overlay. The whole overlay is maintained by exchanging these advertisement messages among peers in this gossip-based fashion.

Some advertisement policies are adopted to create the scale-free topology. First of all, advertisement messages need to contain highly popular hubs, so that receiving peers can connect to them and hence favor the scale-free construction. However, also lightly loaded leaves need to be spread to enable hubs to offer them to querying users. Hence, advertisement messages include the sender node, the three most popular peers it knows, and the three less popular peers it knows. The popularity of a node is computed autonomously by each peer through a simple approximated metric based on the number of received advertisement messages that contain such node.

When a peer receives an advertisement message, it updates its cache by

increasing the popularity of nodes already present and by inserting the new ones. As previously described, it is important for a node to have both hubs and lowly popular peers in its cache. Thus, also a proper cache management policy is adopted if the cache is full: the node with average popularity is removed before the insertion, resulting in a cache that privileges big hubs and lowly popular peers.

User nodes with limited connectivity can access the service by locating an available connectivity peer. When entering the SIP domain, they contact the Bootstrap Service to obtain some connectivity peers. Then they start a random walk search to locate a SIP relay, required to be reachable from the outside. Whenever they have to be involved in a media session, they try the hole-punching procedure using their SIP relay as rendezvous node. If hole-punching fails, they also have to locate a media relay. In both searches, they direct their random walks to the most popular nodes they know (generally the hubs) in order to exploit the scale-free topology.

3 Related work

The robustness to attacks and failures of P2P networks has been extensively analyzed in literature. The behavior of structured networks under high churn is studied, for example, in [5–10], which model the effects of churn on DHTs, investigate how these reduce their lookup performance, and propose methodologies to create structured networks resilient to high churn. On the other side, [11–13] study the negative effect of nodes suddenly leaving unstructured overlays, also proposing solutions to overcome these limitations. Existing work on P2P networking properly covers also security aspects: among others, [14–20] analyze several types of threats for P2P overlays, ranging from Sybil to DoS attacks, propose solutions for peer authentication, and describe methodologies to improve the security of P2P networks.

Specific security aspects of P2P overlays for media communications are con-

sidered in [23, 24]. In particular, these papers deal with possible attacks and malicious actions in structured overlays designed to decentralize SIP, developed in the context of the P2PSIP working group [25] in IETF. P2PSIP proposes to use a DHT to decentralize all the functionalities proper of the SIP protocol, such as registration and user location. Similarly to DISCOS, P2PSIP also proposes to integrate rendezvous and relaying functionalities into the user agents. However, these aspects did not receive particular attention as the P2PSIP work mainly focused on the decentralization of the SIP proxy and registrar servers. In [26], one of the IETF-draft on this topic, the authors simply propose for users behind NAT to perform a connectivity service lookup by randomly selecting a target key and then exploring the DHT to reach this key: if the node is available, it will offer the connectivity service to the querying user. Previous publications on DISCOS [2] also compared the DISCOS scale-free architecture with the random exploration of a flat topology, which exactly maps the technique proposed in [26]. The obtained results showed that the number of peers contacted to find an available relay is sensibly lower in DISCOS than in the structured approach. Furthermore, the ratio between the performance obtained by the two policies increases with the network size, thus demonstrating the scalability properties of the DISCOS architecture. This is not surprising as DISCOS is specifically designed for offering connectivity service, while the structured solution proposed in [26] has the main aim of locating specific resources (i.e., the SIP clients). Hence, DISCOS is somewhat orthogonal to P2PSIP, which may adopt such an approach to offer connectivity service in the distributed SIP architecture it defines.

The presented studies about the robustness of P2P systems to both churn and attacks are somehow related to DISCOS due to its P2P nature. However, its specific aim and operating principles require the analysis of additional aspects not covered in these papers. First of all, in DISCOS peers offering the service can be considered equivalent. Hence, the metric for the overlay performance is the blocking probability (i.e., the probability for a communication to be blocked

because none of the many available connectivity peers is found), rather than the capability of the system to reach arbitrarily rare resources. Furthermore, we have to consider that DISCOS is thought for a multimedia communication network, where nodes connect to the system and then remain online for a long time to exploit the offered services (e.g., make or receive phone calls). Hence, churn is relatively low and does not affect sensibly the overlay operation. Here we are mainly interested in node departures due to failure events, when groups of nodes suddenly leave the overlay, e.g., because of routing or connectivity problems. Finally, in DISCOS we have to deal with the scale-free topology, which is the key to offer high lookup performance. Hence, we need specific analysis concerning the resilience of this type of network to node failures, as well as studies about its robustness to attacks specifically designed to destroy or exploit this topology.

The robustness of scale-free networks has been analyzed in literature. In particular, the effect that these events have on the network topology is considered, pointing out which kind of operations may lead to the network destruction (i.e., the loss of its scale-free nature and, consequently, of the features of this topology). [27] compares a scale-free overlay with an exponential network, i.e., a network where the probability $P(k)$ that a node of the network is connected to k other nodes decays exponentially as k grows. The paper shows how random deletions of nodes in the exponential network result in a sensible growth of the diameter with consequent increase of the distance between nodes, while the scale-free network has been proven to be robust to random node deletion due to its inhomogeneous nature. In practice, a random node deletion policy will select mostly leaves and hence it does not have any sensible impact on the overall topology. However, results change significantly when nodes are removed in decreasing order of degree. While the behavior of the exponential network remains almost the same, in the case of the scale free network the diameter grows rapidly, doubling the values obtained in the previous experiment when just 5% of the nodes were removed. This leads to the conclusion that scale-free networks

are sensible to hub failures and attacks explicitly targeted to hub jeopardizing. These results are confirmed by [28, 29], which further investigate the scale-free behavior when node are deleted randomly or in decreasing order of degree. [28] proves that the fraction of nodes that have to be randomly removed in order to disrupt a scale-free network with $\gamma = 2.5$ is greater than 0.99, while [29] estimates the critical fraction of hubs that must be removed to make the scale-free network collapse.

The limitation of these studies is that they all consider static (or quasi-static) scale-free networks where nodes, once joined the network, neither leave nor change their neighbors. On the contrary, DISCOS is a dynamic infrastructure where peers join and leave the overlay during time. In addition to that, the gossip-based mechanism used for the overlay maintenance provides the capability to create links between peers dynamically, thus enabling the topology to be updated during time. For this reason, we expect DISCOS to be more robust than a static scale-free network.

This paper focuses on the robustness of the DISCOS overlay and consequently brings a double contribution to the existing work. First of all, it extends the results available in literature about the robustness of P2P networks by analyzing how node failures and attacks affect the achieved blocking probability, which is the performance metric of interest in an overlay designed to support media communications as peers can be considered equivalent. Second, it extends the robustness analysis of scale-free networks presented in [27–29] by considering both the dynamicity of the network and attacks specifically designed to compromise a service offered over this topology.

4 Analysis presentation and background

This section introduces and gives some background information on the robustness analysis operated on the DISCOS overlay, which investigates by simulation the DISCOS behavior when failures occur or when some peers attack the overlay

trying to corrupt the information stored in the caches of the other peers. This study is carried out using an ad-hoc simulator implementing DISCOS. Details on the simulator and on some parameters adopted are given in Section 4.2.

4.1 Robustness analysis overview

Given the importance of the scale-free topology for the proper operation of DISCOS, our analysis starts from the investigation of the consequences that failures and attacks may have on the overlay topology. We need to verify that the network both maintains its scale-free topology and is not partitioned into isolated clusters, thus making some connectivity peers mutually unreachable (i.e., when a lookup starts at a connectivity peer and there are other peers that cannot be reached through the information stored in the caches of such peer and its neighbors). This may be deleterious for nodes with limited connectivity, which then are no longer able to propagate their service requests to connectivity peers in separated clusters and consequently see the portion of the network they can explore to find an available servant to be sensibly reduced. The latter evaluation is performed by measuring the number of disjoint clusters generated by failure events.

In addition to that, we evaluate the DISCOS effectiveness in quickly locating reliable connectivity peers, which is directly connected to the maintenance of the scale-free topology. In particular, we measure the number of peers contacted to locate an available connectivity peer and the average call blocking probability due to a fail in locating an available peer. We compare these results with the ones obtained during normal operation (i.e., without failures and attacks). The location of both SIP and media relays is considered.

Our analysis concerning the robustness of DISCOS to failures (presented in Section 5) considers node deletions, similarly to the existing literature on scale-free networks described in the previous section, as well as failures of the Bootstrap Service, which is the key component to allow peers to join the overlay. Furthermore, Section 6 investigates the behavior of DISCOS when attacked by

some malicious nodes. In particular, we define and analyze an attack aiming at disrupting the offered service.

4.2 Simulation background

We developed a custom, event-driven simulator implementing DISCOS. This supports the following four operations: node arrival/departure, media session set up/teardown, SIP relay lookup (triggered when a node with limited connectivity joins the network), and media relay lookup (that occurs when a node requires a relay to perform a media session). In addition to that, we defined some events specifically designated for our robustness study: failure events of both some nodes and the Bootstrap Service and the arrival of malicious nodes whose operation differs from the traditional one as their aim is to attack the DISCOS overlay.

According to [30], about 74% of Internet hosts are placed behind a NAT. In [1] it has been estimated that 20% of the sessions involving such hosts requires a relay, otherwise the communication is not possible. These data are used to determine the percentage of nodes involved in the DISCOS overlay and the amount of sessions requiring a media relay lookup. In essence, a node joining the SIP domain is labeled as behind NAT with probability $p_{nat} = 0.74$ and as connectivity peer joining the DISCOS overlay with probability $p_{cp} = 0.26$. Furthermore, whenever a media session occurs, we label it as requiring a media relay with probability $p_{mr} = 0.2$.

Node lifetime and call length distributions are obtained empirically after analyzing Skype traffic coming from/to the network of the University campus [31]. Node arrivals are modeled using a Poisson process with average arrival rate of 100 nodes per minute that, in conjunction with the Skype node lifetime distribution, results in a SIP domain population of about 30000 nodes in the steady state. This population guarantees the significance of the obtained results and meets our memory and CPU constraints. We also adopted Poisson distributed call occurrences with an average rate of 2000 sessions per minute. This value,

coupled with the Skype call length distribution, leads to about 98% of nodes simultaneously involved in a media session. This allows us to analyze the behavior of the architecture in a worst case scenario. The simulated time is fixed to ten days, which allows us to analyze the network in the steady state (reached after about 1.5 days).

The Bootstrap Service is modeled as a single well-known node capable to store information related to 20 peers of the overlay. Vice versa, according to definitions provided in [2], peers in the overlay deploy a cache capable to contain up to 10 entries and peers sent advertisement messages every 60 minutes using a TTL equal to 2, meaning the flooding depth is limited to 2 hops. [2] shows how these parameters guarantee good performance together with reasonable overhead on the network.

5 Failures of peers and Bootstrap Service

This section studies how DISCOS reacts to failure events. Two different failures are analyzed: a failure at some peers in the overlay which makes them suddenly unavailable[1] and a failure at the Bootstrap Service which causes its reset and the consequent impossibility for new joining peers to enter the network.

5.1 Effects of failures on the scale-free topology

We first investigate the effect of node failures. Since scale-free networks are known to be resilient to random perturbations [27,28], we focus on hub failures. The reason for such an analysis is that hubs disappear only seldom, but the effect of their failure might be catastrophic for the whole overlay because of their crucial role. However, we also have to consider that the dynamic nature of the DISCOS overlay, where nodes join and leave the system, offers opportunities

[1] Node departures could have similar effects. However, as highlighted in Section 3, the relatively low churn-rate characterizing VoIP networks reduces the significance of these events that, as implicitly shown in [2], do not affect the overlay operation. Therefore, we prefer to explicitly refer to node failures, which are the unpredictable events that could actually affect the DISCOS performance.

for recovering from hub failures that do not exist in the static scale-free networks studied so far [27, 28]. Therefore, another goal of this analysis is to investigate the healing capabilities of DISCOS, which may be powerful even under these critical failure patterns.

The analysis is performed by periodically removing the most popular nodes in the overlay in order to try to destroy the network. To determine which peers must be deleted, peers are first sorted by decreasing degree values, then the most popular ones are dropped from the overlay. We consider three simulation scenarios, characterized by the removal of the 1%, 5%, and 10% most popular nodes at each failure event. We select these values because are comparable to the values that [29] identified as able to destroy a static scale-free network.

Figure 2 plots the average in-degree distribution registered on the overlay when hub removal events periodically occur. This figure do not consider overlay partitioning, which is however discussed in the next section. Simulation results show how these percentages of hub failures do not affect the scale-free characteristic of the overlay. This is a major advantage of DISCOS with respect to static scale-free networks and shows how, although the overlay may degrade after the single hub removal event, peers are able to restore the original topology thanks to the advertisement messages exchanged to share the overlay knowledge: they continue to evaluate node popularity and start to favor connections to the most popular nodes survived, which further increase their popularity and become the new hubs.

Analogous considerations may be done for the failure of the Bootstrap Service. Concerning this failure event, we prefer to analyze a reset of the Bootstrap Service instead of making it temporarily unavailable. This is done because if the Bootstrap Service is unreachable for a while and a peer joins the overlay for the first time, the peer keeps on trying to access the Bootstrap Service until it is successful, without introducing significant changes in the overlay behavior. On the contrary, if the joining peer finds the Bootstrap Service empty after a reset, it registers itself into the Bootstrap Service and does not connect to anyone

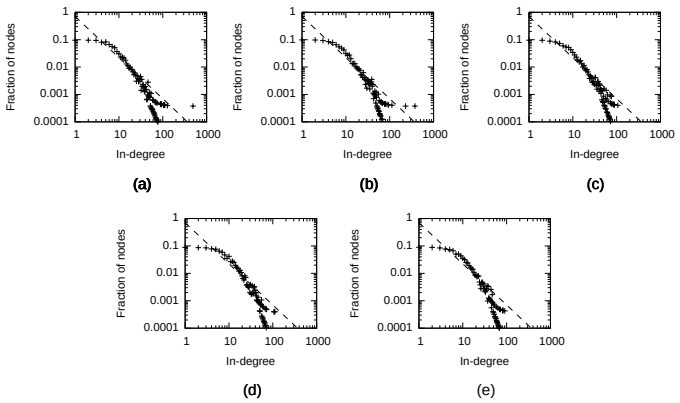

Figure 2: Average in-degree distribution during time: (a) DISCOS normal operation; (b) failure of the 1% most popular nodes; (c) failure of the 5% most popular nodes; (d) failure of the 10% most popular nodes; (e) Bootstrap Service reset. The dotted line is a power-law distribution $P(k) = 0.7k^{-1.5}$.

else, thus potentially modifying the overlay topology. Figure 2(e) shows how the overlay rewiring in DISCOS maintains the scale-free topology also in this case.

5.2 Overlay partitioning

While Figure 2 confirms the maintenance of the scale-free topology, it does not consider if the original overlay breaks into multiple separated clusters of nodes after failures.

In the case of hub removals, multiple disconnected sub-graphs may come up after the deletion. This is due to the fact that, being hubs so popular, the remaining peers can reach each other thanks to the hubs they know. When the hubs are eliminated, a node may not be able to send advertisement messages to other peers, since they are part of separated clusters. Separated clusters may be originated also after a Bootstrap Service reset. In fact, peers arriving just after the reset event have the perception to be the first nodes of the overlay and do not have the chance to connect to anyone else already present. This potentially triggers the creation of a new overlay.

Table 1: Overlay partitioning

Failure event	**Max number of components**	**Max overlay reconnection time**
1% of nodes	1	-
5% of nodes	2	53 min
10% of nodes	7	164 min
Bootstrap Service	2	753 min
Bootstrap Service*	2	98 min

Concerning the failures of nodes, simulation results show that only the deletion of a remarkable fraction of hubs causes relevant fragmentation. Indeed, no overlay partitioning is observed when 1% of nodes are removed, while the creation of no more than two separated clusters is observed when failure events remove 5% of nodes simultaneously. Finally, we observed the creation of at most 7 components when 10% of nodes are removed at the same time. The overlay reconnects within different time bounds thanks to the joining of both new peers and peers that have already been part of the overlay in the past. As we described in Section 2, the former contact the Bootstrap Server when joining, while the latter try to connect first to peers they have in their cache. Hence, if in the Bootstrap Service or in the cache of rejoining peers there are two or more peers being part of different clusters, these can reconnect in a unique component. Results about both the maximum number of observed clusters and the maximum amount of time necessary to observe again a unique network are summarized in Table 1.

The overlay behavior after a Bootstrap Service reset is analyzed by simulating a single reset after the steady state is reached. As expected, after the Bootstrap Service reset the overlay breaks in two components. In this case, the action of rejoining peers is of little help because most of the peers of the second overlay have joined DISCOS after the failure event. Only a particular event can reconnect the two components: a rejoining peer that has in its cache another peer which previously rejoined to DISCOS and for some reason (e.g., peers it

knew were no longer in the network) is in the new overlay; this happens only seldom, as confirmed by the maximum overlay reconnection time reported in Table 1. This result suggests that some modification should be introduced in the original DISCOS mechanisms to improve this specific aspect. In particular, to ensure a faster overlay reconnection, we modified the behavior of peers that join the overlay multiple times. When such peers rejoin the overlay, they do not only reconnect to the active peers they discovered before leaving, but also check the Bootstrap Service to get other peers in the overlay. In this way, rejoining peers are able to know peers in both the components: indeed they know some peers of the original overlay (the ones already in their cache) and new peers that are part of the new component (the ones they got from the Bootstrap Service). This simple modification reduces the maximum overlay reconnection time to less than 100 minutes, as reported in the bottom row of Table 1 (identified with an asterisk symbol).

5.3 Blocking Probability

The modification of the overlay topology and the partitioning of the network in disjoint clusters have a direct impact on the blocking probability (i.e., the probability for a user with limited connectivity to fail in locating an available relay) achieved in the DISCOS overlay, which can be considered the performance metric of real interest in our analysis as its degradation is directly perceived by users: when a lookup fails, they cannot be involved in media sessions or even join the SIP domain.

The results of the lookup performance analysis on the DISCOS overlay confirm what has been observed during the previous tests: the effects of the discussed failures do not have a relevant impact on the overlay behavior. We did not observe any significant modification of the blocking probability when the 1% and the 5% most popular peers are removed from the overlay, as well as when the Bootstrap Service fails. Instead, the average number of peers that a node needs to contact for locating an available SIP relay (evaluated along the

entire simulation) experiences a maximum of 3% growth (from 2.06 peers to 2.12 peers) when the 10% most popular peers are periodically removed from the overlay. These events have even minor impact on the location of media relays, which are required by a smaller portion of users. This confirms the robustness of DISCOS, which, thanks to its dynamic nature, can reconstruct the original scale-free topology (hence providing high lookup performance) even after serious failure events.

6 Attack analysis

This section investigates the behavior of DISCOS when malicious peers attack the overlay. Concerning the possible attacks, we can distinguish between actions aiming at disrupting the scale-free topology of the overlay and actions aiming at disrupting the service itself. The first type of attack includes the attempt by malicious peers to isolate some hubs, as well as actions aiming at destroying the scale-free topology of the overlay. From the point of view of this work, the former can be considered equivalent to the deletion of some hubs, which has been discussed in the previous section. Instead, the latter is effective only if a high percentage of peers are malicious. Since DISCOS is a closed overlay where all participating peers have to be part of a SIP domain, and hence are authenticated and controlled, this event is unlikely to occur. Hence, we focus on attacks aiming at disrupting the service, i.e., attacks where malicious peers do not attempt to modify the scale-free topology, but try to spread fake advertisement messages including peers that do not offer the service. In particular, we define an ad-hoc Sybil attack that is expected to make the overlay inefficient if successful.

6.1 Attack overview

According to [32], a Sybil attack consists in forging a large number of false identities and in using them to gain influence in the overlay. When applied to DISCOS, this consists in propagating the information related to some malicious

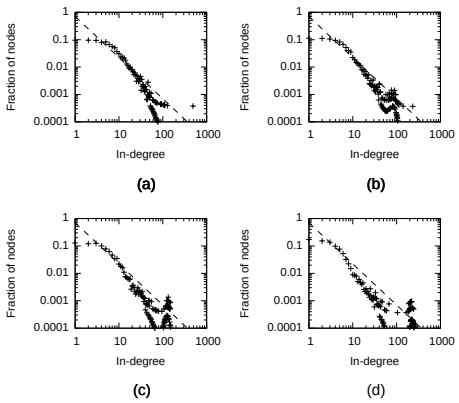

Figure 3: Average in-degree distribution computed during time: (a) DISCOS normal operation; (b) 5% of malicious peers; (c) 10% of malicious peers; (d) 30% of malicious peers. The dotted line is a power-law distribution $P(k) = 0.7k^{-1.5}$.

nodes across the overlay, so that they can gain the reputation of hubs. The attack is successful if malicious nodes replace the original hubs and return other malicious peers when queried for service. Indeed, this may result in lookup performance loss or even in lookup failures as in DISCOS lookups are directed to hubs with high probability.

Specifically, the attack consists in trying to push the popularity of a restricted number of malicious peers that we call "black-hole" peers, in order to replace hubs with them. When queried for the service, the "black-hole" peers answer negatively and suggest only other "black-hole" peers as search alternatives. To make the popularity of "black-hole" peers growing fast, they are designed to stay in the overlay forever and are supported by malicious peers that join the overlay and start announcing them instead of adopting the advertisement strategy defined for DISCOS. By doing this, it is possible to poison the caches of the peers in the overlay, making them believe that the "black-hole" peers are hubs. Hence, also the "poisoned" peers start announcing the "black-hole" peers, thus further propagating the overlay corruption. We also assume malicious nodes to know the lifetime of entries in the Bootstrap Service, which is constant and equal to 60 minutes, as described in Section 2. This is a realistic assumption (malicious

nodes can cooperate in collecting statistics to estimate this value with reasonable accuracy) and allows malicious peers to further increase the effectiveness of their attack: malicious peers try to re-register themselves on the Bootstrap Service when their entries are elapsing, thus potentially leading the Bootstrap Service to be entirely populated of attackers; these are then communicated to new users, which hence result poisoned since the beginning of their activity in the overlay.

During simulation, we assume that 30 "black-hole" peers are present in the overlay, i.e., 0.1% of the overlay population. This value is comparable to the percentage of highly popular nodes (i.e., whose in-degree is close to 100) present in the overlay during normal operation, as can be observed in Figure 3(a). This is done to effectively poison the overlay, which potentially replaces all ordinary hubs with these "black-hole" nodes. Furthermore, we consider three different percentages of malicious peers announcing "black-hole" nodes, namely, 5%, 10% and 30% of the overall overlay population.

The effect of the attack on both the overlay topology and the blocking probability is analyzed in the following. Unlike the failure analysis, overlay partitioning is not considered in this case as there is no reason for this type of attack to generate separated clusters.

6.2 Overlay topology

Although the attack is not designed to destroy scale-free networks, the behavior of malicious peers may have negative effects on the overlay topology, thus causing a decrease in the DISCOS performance. Figure 3 portrays the weighted mean of the in-degree distributions related to normal operation, compared to the distributions obtained assuming the above mentioned percentages of malicious peers announcing "black-hole" nodes (i.e., 5%, 10% and 30% of the overall overlay population). The figure shows how the in-degree distribution changes due to the attack. It can be seen that one peak is present in correspondence of large in-degree values, which grows proportionally to the number of attackers. This

can be motivated with the abnormally large popularity of "black-hole" peers, which is obtained by pushing "black-hole" peers popularity through the overlay in conjunction with their infinite lifetime. However, we can argue that the overlay topology slightly changes and can still be considered a good approximation of a scale-free network.

6.3 Blocking probability

Although the topology is not sensibly affected, the DISCOS operation is vulnerable to the presented Sybil attack. In fact, cache poisoning leads nodes to know, announce, and contact malicious and "black hole" peers with high probability, potentially only these nodes. This is confirmed by the sensible worsening we observed in the lookup performance: after a few hundreds of minutes from the beginning of the attack, only about 20% of users with limited connectivity can locate a SIP relay when entering the SIP domain, while the call blocking probability (i.e., the probability for an incoming call requiring a media relay to be rejected because the system fails in locating the relay) assumes values close to 1 even when the percentage of malicious peers is only 5% of the overlay population. This is clearly due to the behavior of the "black-hole" peers, which do not provide the required service and announce only other "black-hole" nodes. Consequently, nodes with limited connectivity keep on searching until an available servant is found.

To prevent this performance decrease, we slightly modified the operation of the Bootstrap Service: instead of deregistering peers after a fixed entry lifetime of 60 minutes (we remember the reader that this value is constant in the current version of DISCOS), the Bootstrap Service adopts a statistical process to determine the deletion time, so that there is no way for malicious nodes to estimate the lifetime of their entry and, consequently, to force their re-registration when the lifetime elapses. To circumvent this countermeasure, malicious peers could poll the Bootstrap Service at high rate in order to increase their probability to be registered. However, this can be avoided by deploying simple techniques

based on well-known anti-DoS solutions. Hence, it is not considered in this paper.

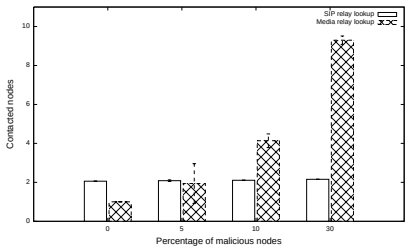

Figure 4: Average number of contacted peers to locate a SIP/media relay.

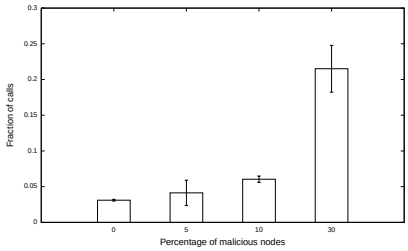

Figure 5: Call failure probability.

Figure 4 shows the average number of contacted peers to locate a relay when the abovementioned Bootstrap Service operation is introduced in the network. It can be observed how this Bootstrap Service modification leads SIP relay lookups to undergo negligible effects due to attacks. The performance of media relay lookups is sensibly improved, as well. In fact, the average number of contacted peers to locate a media relay experiences a significant growth (reaching a value greater than 9) only when the attackers are 30% of the overall population. This situation is unlikely to verify in real networks, as it would require the involvement of a large number of nodes.

These results regards the average complexity of relay lookups, but it is important to investigate also how this Bootstrap Service modification impacts on the call failure probability, i.e., the probability for a relayed call to be refused as the system cannot find a media relay. Figure 5 plots the call failure probability

for different percentages of malicious peers in the overlay. Again, we observe a significant growth (from about 0.03 to about 0.22) only when 30% of peers are malicious.

This performance improvement is due to the modified operation of the Bootstrap Service, which now is able to offer non-malicious nodes to incoming users, thus reducing cache poisoning and hence contrasting the action of malicious peers. In particular, we observed above how SIP relay lookups are almost unaffected by attacks with this Bootstrap Service modification. In fact, since users perform SIP relay lookups as soon as they enter the SIP domain, this modification allows them to base SIP relay searches on the non-malicious nodes just retrieved from the Bootstrap Service.

7 Conclusion

Overlays for media communication have been receiving an increasing interest due to their potential to avoid scalability issues of centralized solutions. In particular, the DIStributed COnnectivity Service (DISCOS) architecture distributes rendezvous and relay functionalities among peers in order to offer *connectivity service* (i.e., NAT traversal functionalities) to the users of a SIP domain. Previous work on DISCOS covered aspects related to its overlay topology and operating principles, showing how a gossip-based unstructured overlay resulting in a scale-free network offers high performance and efficiency. This work, instead, focuses on the robustness of DISCOS to failures and attacks.

We considered different failure rates of the participating peers, as well as the possible failure of the Bootstrap Service (a system supporting node joining in DISCOS). Furthermore, we defined a Sybil attack aiming at disrupting the offered connectivity service.

Although our analysis demonstrated the overall robustness of the system, it also highlighted two critical points. First, we observed a significant decrease of the overlay performance when the system is under attack. Second, we noticed

that the failure of the Bootstrap Service may lead the overlay to partition in separated clusters, with consequent lookup performance decrease. However, our simulation results showed how slight modifications of the Bootstrap Service operation and the peer joining process solve this issue and allow DISCOS to offer reasonable performance also in presence of these critical events.

The paper also shows how the dynamicity of nodes improves the robustness of scale-free networks. In fact, unlike its static counterpart, the DISCOS dynamic scale-free overlay is robust also to hub deletions. This result has a more general value and may be of interest in other contexts where scale-free networks cover an important role, as well.

References

[1] Bryan Ford, Pyda Srisuresh, and Dan Kegel. Peer-to-peer communication across network address translators. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, Anaheim, CA, 2005. USENIX Association.

[2] L. Ciminiera, G. Marchetto, F. Risso, and L. Torrero. Distributed connectivity service for a SIP infrastructure. *Network, IEEE*, 22(5):33–40, 2008.

[3] G. Camarillo, J. Rosenberg, H. Schulzrinne, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP session initiation protocol. In *IETF Request for Comment 3261*, 2002.

[4] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. *Review of Modern Physics*, 74:47–92, June 2001.

[5] Sean Rhea, Sean Rhea, Dennis Geels, Dennis Geels, Timothy Roscoe, Timothy Roscoe, John Kubiatowicz, and John Kubiatowicz. Handling churn in a dht. In *In Proceedings of the USENIX Annual Technical Conference*, 2003.

[6] H. Shen and C.-Z. Xu. Locality-aware and churn-resilient load-balancing algorithms in structured peer-to-peer networks. *Parallel and Distributed Systems, IEEE Transactions on*, 18(6):849 –862, june 2007.

[7] O. Herrera and T. Znati. Modeling churn in p2p networks. In *Simulation Symposium, 2007. ANSS '07. 40th Annual*, pages 33 –40, march 2007.

[8] Song Fu, Cheng-Zhong Xu, and Haiying Shen. Random choices for churn resilient load balancing in peer-to-peer networks. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1 –12, april 2008.

[9] Giang Ngo Hoang, Hung Nguyen Chan, Khang Nguyen Van, Thu Le thi Xuan, Thang Nguyen Manh, and Vinh Vu Thanh. Performance improvement of chord distributed hash table under high churn rate. In *Advanced Technologies for Communications, 2009. ATC '09. International Conference on*, pages 191 –196, oct. 2009.

[10] Junfeng Xie, Zhenhua Li, Guihai Chen, and Jie Wu. On maximum stability with enhanced scalability in high-churn dht deployment. In *Parallel Processing, 2009. ICPP '09. International Conference on*, pages 502 –509, sept. 2009.

[11] R. Baldoni, S. Bonomi, A. Rippa, L. Querzoni, S.T. Piergiovanni, and A. Virgillito. Evaluation of unstructured overlay maintenance protocols under churn. In *Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on*, pages 13 – 13, july 2006.

[12] Ruggero Morselli, Bobby Bhattacharjee, Michael A. Marsh, and Aravind Srinivasan. Efficient lookup on unstructured topologies. *Selected Areas in Communications, IEEE Journal on*, 25(1):62 –72, jan. 2007.

[13] F.E. Bustamante and Y. Qiao. Designing less-structured p2p systems for the expected high churn. *Networking, IEEE/ACM Transactions on*, 16(3):617 –627, june 2008.

[14] J. Dinger and H. Hartenstein. Defending the sybil attack in p2p networks: taxonomy, challenges, and a proposal for self-registration. In *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*, page 8 pp., april 2006.

[15] J. Liang, N. Naoumov, and K. W. Ross. The index poisoning attack in p2p file sharing systems. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –12, april 2006.

[16] Zhongwen Li, Xiaochen Xu, Liang Shi, Jian Liu, and Chen Liang. Authentication in peer-to-peer network: Survey and research directions. In *Network and System Security, International Conference on*, volume 0, pages 115–122, 2009.

[17] K.P.N. Puttaswamy, Haitao Zheng, and B.Y. Zhao. Securing structured overlays against identity attacks. *Parallel and Distributed Systems, IEEE Transactions on*, 20(10):1487 –1498, oct. 2009.

[18] K.R.B. Butler, S. Ryu, P. Traynor, and P.D. McDaniel. Leveraging identity-based cryptography for node id assignment in structured p2p systems. *Parallel and Distributed Systems, IEEE Transactions on*, 20(12):1803 –1815, dec. 2009.

[19] M. Srivatsa and Ling Liu. Mitigating denial-of-service attacks on the chord overlay network: A location hiding approach. *Parallel and Distributed Systems, IEEE Transactions on*, 20(4):512 –527, april 2009.

[20] M. Sanchez-Artigas and P. Garcia-Lopez. On routing in distributed hash tables: Is reputation a shelter from malicious behavior and churn? pages 31 –40, sept. 2009.

[21] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making gnutella-like p2p systems scalable. In *Proceedings of ACM SIGCOMM '03*, pages 407–418, 2003.

[22] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. Search in power-law networks. *Physical Review E*, 64(4), 2001.

[23] J. Seedorf. Security challenges for peer-to-peer sip. *Network, IEEE*, 20(5):38 –45, sept.-oct. 2006.

[24] D. Chopra, H. Schulzrinne, E. Marocco, and E. Ivov. Peer-to-peer overlays for real-time communication: security issues and solutions. *Communications Surveys Tutorials, IEEE*, 11(1):4 –12, quarter 2009.

[25] Peer-to-peer session initiation protocol (p2psip) ietf working group.

[26] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne. Resource location and discovery (reload) base protocol. Internet Draft draft-ietf-p2psip-base-06, Internet Engineering Task Force, November 2009. (Work in progress).

[27] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, July 2000.

[28] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85:4626–4628, November 2000.

[29] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Breakdown of the internet under intentional attack. *Physical Review Letters*, 86:3682–3685, April 2001.

[30] Martin Casado. Peering through the shroud: The effect of edge opacity on IP-Based client identification. In *Proceedings of the int. Symp. On Networked Systems Design and Implementation*, 2007.

[31] Dario Bonfiglio, Marco Mellia, Michela Meo, and Dario Rossi. Detailed analysis of skype traffic. *IEEE Trans. Multimedia*, 11(1):117–127, 2009.

[32] John R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer-Verlag, 2002.