DOI 10.12694/scpe.v16i3.1101
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 16, Number 3, pp. 271–287. http://www.scpe.org

SODAP: SELF-ORGANIZED TOPOLOGY PROTECTION FOR SUPERPEER P2P NETWORKS

PAUL L. SNYDER* AND GIUSEPPE VALETTO†

Abstract. Unstructured superpeer overlays are an approach to peer-to-peer systems that enables collective organization and ensures the efficient participation and use of divers peers with heterogenous capabilities and resources. Such overlays are, however, vulnerable to failures and attacks that target the superpeers in an attempt to disrupt the overlay. In this paper, we present *SODAP* (Self-Organized Degree Adaptation Protection), a self-organized strategy for the self-protection of the overlay, based on the local adaptation of a peer's degree in response to disconnections, whether directly detected, or indirectly discovered with the assistance of neighbor peers. When network conditions deteriorate, the SODAP mechanism induces the creation of redundant connections to superpeers, which leads to the adjustment of the entire overlay to a topology that is more resilient to disconnection, while still allowing the system to continue exploiting heterogeneous peer capabilities. When network conditions improve, SODAP responds by reducing peer degrees to reduce redundancy and streamline the topology. We demonstrate this mechanism and evaluate its effectiveness as an extension to *Myconet*, a self-organized superpeer overlay for unstructured peer-to-peer networks.

Key words: P2P, self-organization, self-protection, degree adaptation, topology adaptation

AMS subject classifications. 68M14

1. Introduction.

We present *SODAP* (Self-Organized Degree Adaptation Protection), a new self-protection strategy for peer-to-peer networks based on topology adaptation.

Peer-to-peer (P2P) networks are very large examples of highly decentralized collective systems that can scale to millions of participants. They are by nature open-boundary systems, where the individual peers are largely autonomous, often self-interested, and characterized by (even extreme) diversity in the capabilities and resources among participating peers. P2P networks must also adapt dynamically in the face of ever-changing conditions that may affect individual peers or large portions of the network. This requires decentralized control and self-organization, as central management or supervision is impractical or impossible.

P2P systems typically engage in collective self-organization through the construction of an overlay network; this imposes a topology on top of the often-chaotic underlying network, and acts as an enabler for other services (such as search or routing). Overlays are commonly classified as either structured (typified by Distributed Hash Table-based systems) or unstructured.

Unstructured overlays construct an arbitrary topology that can be optimized for the requirements of a particular application. A common unstructured P2P topology is a hierarchical overlay, based on the concept of *superpeer*. Superpeer overlays attempt to take advantage of peer heterogeneity, using protocols that identify particularly powerful or well-situated peers. Superpeers then differentiate their role and functionality, to improve the global performance and functioning of the overlay. Superpeers act as a backbone for the overlay, and provide services to less-powerful peers; they typically have a greater number of neighbors (hence, higher network degree) than normal peers, and may construct a scale-free topology [19].

Superpeer overlays are more resistent to random failures than overlays that are non-hierarchical [3]. However, they are more vulnerable to malicious attacks targeting specific peers [17]. The higher degree of the superpeers and their specialized role in the overlay increase the effect of these peers' failure; even relatively straightforward attacks—such as *crash attacks*—directed against these nodes may result in the disconnection of many other peers, and the disruption of the overlay at large, with relatively little effort on the part of the attacker [18].

Self-adaptation is an answer to this topological vulnerability. Several superpeer systems use *self-healing* protocols for the maintenance of the overlay that allow them to efficiently rebuild the intended topology after it has been damaged [19, 25]. Self-healing, however, remains purely reactive: it does not proactively *protect* the overlay, nor does it prevent its disruption for the duration of an attack.

*SunGard Consulting Services, New York City, NY, USA
†Fondazione Bruno Kessler, Trento, Italy

Another possibility is to augment the system with *self-protection*. One strategy, as suggested in [11] and [33], is *topology adaptation*, that is, adding another layer of adaptivity that dynamically modifies the rules of the protocol that constructs and maintains the overlay's topology, and in particular makes the choice of peer degree adaptive. By altering peer degree, the overlay topology can shift automatically to an overall degree distribution that is appropriate to the current conditions, and becomes more resilient to malicious attacks, as well as to variable levels of network churn.

A superpeer overlay with topology adaptation has two additional important traits, which, as observed by [12], are often key for a successful collective adaptive system. Not only does it exhibit *heterogeneity* and *temporal diversity*, but it has built-in means to leverage those characteristics for the benefit of the collective. The promotion from peer to superpeer is a mechanism for specialization, which instigates further heterogeneity through beneficial differentiation of roles, functionality and offered services among the system participants. Moreover, topology adaptation is a source of temporal diversity. Each individual peer can significantly change its behavior over time through how it "interprets" the rules of the the topology maintenance protocol, but these effects are not isolated. They also propagate to neighborhoods in the network, and, ultimately, to the whole overlay, which ends up exhibiting different, emergent topological properties at different times based on the circumstances.

Our SODAP topology adaptation protocol is an example of such collective adaptation for the purpose of self-protection. Under SODAP, peers locally adapt their neighbor connections in response to disconnections that are directly detected, or discovered collaboratively via neighbor signalling. We demonstrate and evaluate SODAP on top of Myconet, a self-healing, biologically inspired peer-to-peer overlay protocol [25]. One of the distinguishing characteristics of Myconet is that nodes are not simply divided between peers and superpeers, as in other P2P systems, but can specialize and evolve through a more refined hierarchy of roles, each with distinct functionality and responsibilities towards other peers and the overlay at large.

SODAP represents an advancement with respect to previous topology adaptation approaches, which tend to work in a "binary" mode, that is, make the entire network switch between two rigidly defined topologies. In contrast to those binary approaches, which—by design—choose to ignore peer heterogeneity when under attack, SODAP is able to adapt more smoothly and locally to both degradation and improvements in network conditions. As a consequence, SODAP achieves effective self-protection without giving up the ability to exploit peer heterogeneity.

The rest of this paper is organized as follows: in Section 2, we provide an overview of this field of research, and position SODAP in it. In Section 3 we discuss some technical details on our own previous work, that is, the *Myconet* and *HITAP* protocols that served as the inspiration for SODAP's topology adaptation mechanism, and how that previous work led to the design of the proposed protocol. Section 4 presents in depth the *SODAP* approach to topology self-protection through degree adaptation. Section 5 contains the results of our experimental evaluation of the protocol. Finally, in Section 6 we offer a recap of our results, discuss their significance, and outline some potential directions for future work.

2. Related Work. Peer-to-peer networks represent some of the largest and more widely used distributed systems; for example, content-sharing P2P networks drive very large percentages of Internet traffic [2]. Systems based on superpeer overlays have been used for file sharing (e.g. Gnutella, Kazaa), VOD streaming [21], VOIP (e.g. the original architcture of Skype [1]), and other heavy-duty, large-scale applications.

While P2P networks offer a number of significant benefits to the design and operation of open, complex, and large-scale distributed systems, they are faced with a raft of attacks that can be mounted against them. Surveys of these attacks can be found, for example in [5] and [6], which identify attack classes such as denial-of-service; man-in-the-middle, worms; rational attacks (that is, non-cooperation by network participants); sybil attacks (where nodes forge identities); and eclipse attacks (where colluding nodes attempt to partition the network). Yue *et al.* [31] identify additional classes of attacks, including those targeting the routing process, application-level attacks, and attacks that target the idiosyncratic topological features of a given P2P overlay. Of particular relevance to the self-protection mechanism discussed in this paper are *crash attacks* that aim to disconnect or seriously disrupt the fabric of a P2P network by disabling or eliminating a portion of the peers, such as superpeers, particularly those in topologically sensitive positions.

There are several classes of approaches in the P2P literature to counter those attacks. The conceptually

simplest class is represented by non-adaptive *topology preservation* approaches, which aim to reinforce the network by building a fixed amount of redundancy into the fabric of the overlay. Yang and Garcia-Molina [3] suggest a simple method for increasing resilience using k-redundancy (discussing only the case where $k = 2$). In their approach, multiple superpeers share the duty of providing services to each cluster of leaf peers. The multiple-parent approach is taken in a different direction by ERASP [13], which proposes assigning a fixed number of superpeers for each leaf peer. THUP [28] examines a superpeer topology based on a bimodal degree distribution that is resistant to both churn and denial-of-service attacks, and suggests a join-strategy for peers that enter the network that allow them to approximate such a bimodal distribution. Different topology preservation strategies may be best suited for particular applications: for example, Brinkmeier *et al.* [4] examines topologies that are optimally stable against both peer churn and targeted denial-of-service attacks from the perspective of streaming video applications.

A step beyond building a fixed-topology redundant fabric is represented by adaptive *topology recovery* approaches that enable self-healing following damage to the overlay. This includes protocols such as Myconet, SG-1 [19] and DLPSPN [30], which can quickly heal the P2P overlay network after being disrupted by a successful targeted attack. This is a reactive defense that attempts to minimize the disruption to the overlay and any applications running on top of it.

Proactive and adaptive defense (*i.e.*, self-protection) is the goal of *topology adaptation* approaches. The key feaure of a self-protection strategy is its ability to respond immediately to critical disconnections, and rearrange the P2P fabric in ways that prevent its disintegration, and thwart attacks by rendering them ineffective or too costly. However, topology adaptation imposes itself a cost. as shown by [22], which examines the effect of failures of individual nodes on unstructured P2P networks, and analyzes the efficacy of increasing redundancy, degree of connectivity, and hop-count distance.

Self-protection via topology adaptation is a relatively open-field area of research. In the context of structured P2P overlays, [20] discusses how the Tapestry Distributed Hash Table [32] may identify and neutralize Denial-of-Service (DoS) attacks by modifying its topology in ways that isolate attacking peers in ad hoc clusters. For unstructured superpeer overlays, one of the primary previous works is Keyani *et al.* [11], which discusses the implementation of a binary self-protection strategy that switches a Gnutella overlay from a scale-free to an exponential topology. The switch occurs upon attack detection occuring in a relatively small proportion of the nodes (less than 15%). However, the initial scale-free topology is built using a centralized oracle, which does not provide support for reverting the topology after an attack is complete. Our own previous work, *HITAP* [26] (Hormone-Inspired Topology Adaptation Protection), similarly proposes a binary switch approach, but is fully decentralized and biologically inspired. HITAP uses the diffusion of an "alert hormone" through the nodes, to propagate the signal to switch between a superpeer and a flat topology, in response to targeted attacks against high-degree peers, and to switch back as the hormone level decays. HITAP proved effective in making a superpeer overlay substantially more impervious to those attacks. However, HITAP is limited, due to its binary protection strategy which cannot take advantage of peer heterogeneity during attacks; moreover, it is sensitive to the parameters of the diffusion protocol, which needs tuning to remain effective across multiple network scales, churn levels, and attack sizes.

In contrast with the binary, "all-or-nothing" stance of the works mentioned above, Zweig and Zimmerman [33] have offered a graph-theoretical argument that explores the feasibility and effectiveness of flattening progressively the range of node degrees in an overlay in response to attacks. Thart work has provided us with the inspiration for our SODAP project, which exhibits smoother topology adaptation capabilities. These work seamlessly and locally in both directions, increasing or decreasing the node degrees based on the observed disconnections (or lack thereof) of each node's neighbors.

Finally, it is important to mention works that propose techniques to measure the topological vulnerability of P2P networks. Mitra *et al.* have examined this issue from the perspective of percolation theory [15, 18], measuring network stability as a critical fraction of nodes that must be removed before the network disintegrates, and proposing a framework for analyzing the resilience of P2P networks. This analysis is further advanced using rate equations to analyze the emergence of superpeers over time [16]. Percolation theory is also used by Srivastava *et al.* [27], who examine attacks on superpeer networks with varying levels of degree-degree correlation. Ghedini *et al.* [9] focus on the special problems caused by P2P systems characterized (like superpeer overlays)

by a relatively small number of high-degree nodes. They suggest that considering only actual disconnections (or the use of standard degree and betweenness centrality measures to assess the imminent threat of disconnection) may not be enough to properly assess the state of a network. They suggest examining networks from the perspective of vulnerability as well as actual damage (the proportion of nodes that if killed would result in the disconnection of a network).

These techniques that assess topological vulnerability can be also useful to assess the performance of a given overlay protection strategy. However, they work by taking a snapshot of a topology and examining its instant resistance to attacks. Analysis is typically performed by progressively removing peers from the topology, either at random or through selection via a graph-theoretic metric. These approaches are not well-suited to evaluate the effect of adaptive protection strategies, such as SODAP, which continuously modify the overlay to thwart attacks. The experiments we present in Section 5 examine empirically similar properties as a dynamic feature of the overlay network, and show that SODAP effectively stabilizes the disconnection rate across a wide range of scenarios.

3. Background and Motivation. Below, we discuss two relevant previous works of ours. Myconet (Section 3.1) is a biologically inspired superpeer overlay, which has especially efficient topology construction, maintenance and self-healing features, on top of which we have designed, developed and evaluated SODAP [25]. We also briefly describe (Section 3.2) HITAP, our previous topology adaptation work, since it provided the motivation and informed the design of SODAP. For full details, the interested reader should refer to [26].

3.1. Myconet. Myconet is an unstructured P2P overlay based on a fungal metaphor. Superpeers are considered to be the *hyphae* (root structures) in a fungal mycelium, and non-superpeer leaf nodes are considered to be *biomass* that can be moved among the hyphae as needed. Myconet uses an abstract concept of *capacity* to model the heterogeneity of peer capabilities: higher-capacity nodes make better superpeers, because they are able to maintain connection to (and provide application-specific service to) more of the other peers. To preserve the generality of Myconet, the mapping of capability and resource profiles to this abstract and uniform capacity measure is considered to be an application-dependent exercise.

In Myconet, local protocol dynamics work in a decentralized and self-organized fashion to identify the "best" participants to the overlay (according to the capacity metric) to serve as superpeers from a heterogeneous assortment of peers. In Myconet, superpeers (a.k.a *hyphal peers*) evolve through a series of protocol states, as displayed in Figure 3.1. Each state corresponds to a distinct role, and the peers collaborate, each according to its current role, to build a robust network of interconnections with other hyphal peers. They also perform a self-healing function, as each takes local actions role to reconstruct and maintain the overlay in response to incidental peer failures or crash attacks. When promoted from biomass to the first, *extending* hyphal state, superpeers are mainly focused on exploration, acting as connection points for new biomass peers entering the network. *Branching* hyphae are extending superpeers that have reached their target level of utilization (*i.e.*, they have connected to sufficient biomass peers to reach a level of utilization appropriate to their capacity); they manage the number of extending peers in the network while growing new hyphal interconnections. Finally, *immobile* hyphae have reached levels of full biomass connection and a target number of hyphal interconnections, and are considered to be relatively stable, having remained in the network long enough to transition through all the previous protocol states. When hyphal peers contact each other, superpeers of lower protocol state and lower capacity may transfer a portion of their attached biomass peers towards higher superpeers, in order to saturate the capacity of the latter. Should a hyphal peer for any reason fall below a utilization threshold, it may demote itself to a lower protocol state, possibly reverting to become a biomass peer.

Myconet constructs and maintains a strongly interconnected, superpeer-based overlay that converges quickly to an optimal number of superpeers. It self-heals equally quickly in the face of failures, repairing damage and dynamically adjusting the network topology in response to changing conditions. We have applied Myconet as the overlay substrate for a decentralized service network with clustering and load-balancing, described in [29].

3.2. HITAP. *HITAP* (Hormone-Inspired Topology Adaptation Protection) is a biologically inspired approach to self-protection of a P2P network, switching between two different topology management protocols in response to detected attacks. HITAP has been built on top of the basic Myconet protocol, though its self-protection strategy is also applicable to other unstructured superpeer-based systems.

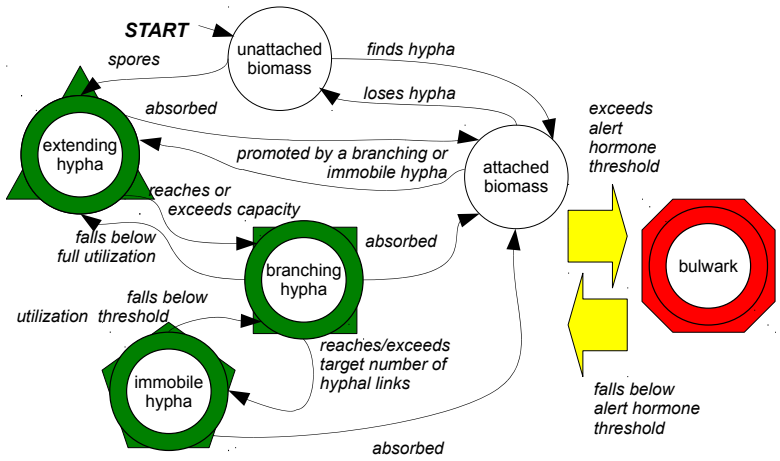


FIG. 3.1. *Protocol state transitions in Myconet (left) with the added bulwark state used by HITAP (see Section 3.2). States are are described in Section 3.*

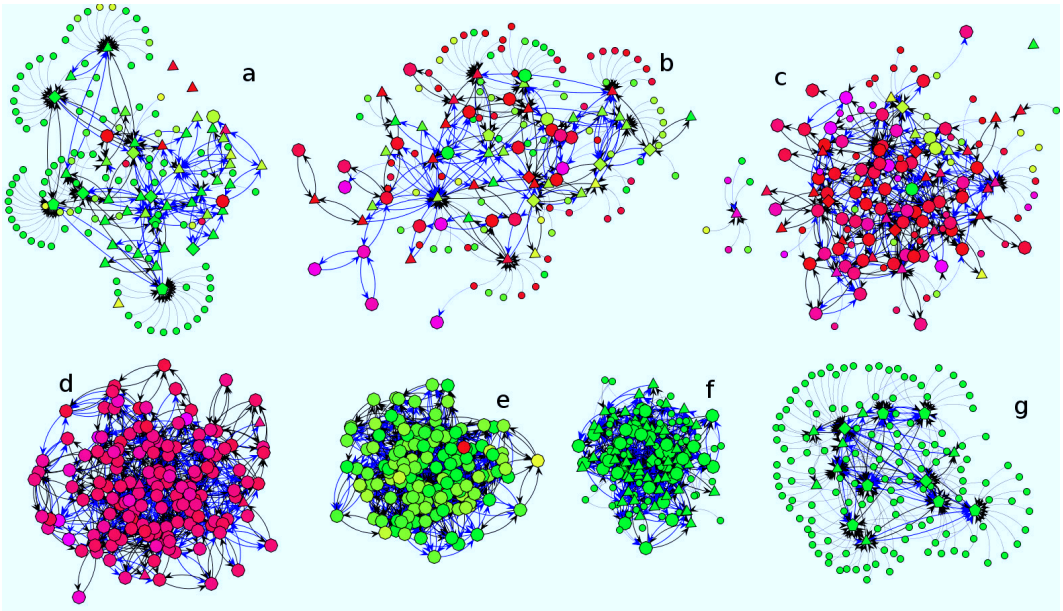


FIG. 3.2. *HITAP attack detection in an example 150-node network with* $c_{max} = 20$ *(see Section 5.1 for a description of this parameter). (a) Two cycles into an attack, some local buildup of alert hormone is occuring. (b) Four cycles into the attack, nodes have passed the transition threshold (red) and switched into bulwark (octagons), causing more nodes to make the jump. (c) Alert hormone levels continue to rise, diffusing through all nodes. (d) Six cycles into the attack, all nodes have switched to the bulwark state. (e) Four rounds after the attack ends, hormone concentrations in most nodes have decayed below the reversion threshold; nodes delay their return by five cycles to reduce thrashing. (f) Nine cycles after the attack ends, alert hormone levels have decayed and the nodes revert to normal operation. (g) Within six more cycles, the nodes have reconstructed the superpeer overlay.*

Damage to the overlay caused by peer disconnection (whether by attack or churn) results in HITAP in the generation of a marker (metaphorically considered to be an *alert hormone*) that spreads from neighbor to neighbor by diffusion. The increasing concentration of the hormone in a system experiencing a targeted attack causes the network as a whole to adapt its topology and switch to a "flat" mode where all nodes have a nearly uniform degree. While the system remains in that mode, attackers are deprived of major targets and the network becomes much harder to disrupt. As the attack subsides (or is reduced to simply killing random

low-degree nodes), the hormone levels of each node decay. When the local concentration falls below a threshold, nodes will switch their protocol back to normal operation, and start working towards re-establishing a superpeer overlay.

The exact amount of hormone generated by the remaining peers is determined by the number of former neighbors of a failed peer, according to a quadratic relationship. This function places a heavier weight on the failure of high-degree peers (such as those in the immobile or branching states), whereas it is relatively insensitive to the failure of low-degree peers (such as biomass and peripheral extending hyphal peers). Since every neighbor of a failed peer will generate the alert hormone, the failure of large-degree peers will result in the release of large amounts of hormone throughout the network.

Once sufficient amounts of alert hormone have accumulated in a peer, it switches into a new protocol state, termed *bulwark*. Bulwark peers have switched into a defensive posture and act to reduce the network's vulnerability to attack. They do not follow the full Myconet protocol, but simply work to reconfigure their local neighbor relationships into a flat (and hence non-superpeer) network by maintaining a fixed number of connections to other peers. When a peer switches into the bulwark state, it severs connections with all its existing neighbors, causing the cascading generation of yet more alert hormone.

The maximum quantity of alert hormone that may be held at a node is capped, to prevent the build-up of network pockets with very large amounts of hormone. Additionally, peers metabolize the alert hormone over time. The local concentration is periodically reduced by a fixed percentage. Thus, hormone levels will spike following a failure, but will slowly drop back down if additional failures do not continue to generate additional alert signals.

After switching states, a new bulwark peer immediately begins to connect to new neighbors, growing new connections to other peers until it has reached a target range (as well as dropping connections that bring it above that range). If all peers enter the bulwark state, the network will quickly converge to a relatively flat topology, without superpeers. The network thus becomes resilient against targeted attacks, although, while in this mode, the network cannot effectively exploit heterogeneous node capabilities.

Once the network has shifted to such a flat configuration, attacks against particular peers are practically indistinguishable from ordinary churn. Because of this, the failure of a bulwark peer does not cause significant levels of the alert hormone to be generated.

A peer will remain in the bulwark state until its local concentration of alert hormone has dropped below a threshold value, plus an additional period of time. This latency is a customizable parameter, and is designed into the protocol to prevent the network from attempting to switch between modes too quickly, which may result in the overhead of restoring the superpeer topology while waves of hormones are still traversing the network, possibly because an attacker is still trying to disrupt the overlay.

3.3. Motivation. HITAP has demonstrated a decentralized, self-organizing approach to self-protection that is able to detect and mitigate topology-based attacks.

However, HITAP also has a relatively coarse-grained mechanism: the efficiency of the systen is greatly reduced while the network is in the defensive posture, as the overlay is no longer taking advantage of peer heterogeneity. A further challenge is the sensitivity of the hormone-based signaling mechanism used by HITAP to network conditions. Values selected for the protocol parameters—in particular, values that regulate the generation and decay of the alert hormone—have a significant effect on the system's ability to distinguish between attack conditions and normal operations, in which some background level of peer churn is natural; however, there is no single parameter set that is equally effective across a wide combination of churn rates and network sizes. Finally, HITAP tends to fall victim of its own success: when the overlay has assumed the defensive stance of the flat topology, attacks become ineffective but also indistinguishable from background noise and random failures. Therefore, HITAP is not able to detect with certainty that an attack has finished; rather, it "guesses" when to begin rebuilding the superpeer network, based on the decay rate of the alert hormone (another parameter). If the hormone evaporates too quickly, the attack may still be underway, and the network will need to again revert to the bulwark mode, resulting in further inefficiency.

These limitations have prompted us to investigate alternate strategies for overlay self-protection through topology adaptation. The outcome is *SODAP*, where nodes adjust their degrees dynamically to improve resistance to both attacks and network churn while continuing to exploit high-capacity peers as superpeers.

4. Approach. We now describe *SODAP* (Self-Organized Degree Adaptation Protection) as we have implemented it on top of the basic Myconet protocol; however, it is important to underline that its approach is sufficiently general to be applicable also to other superpeer-based P2P overlays. While the inspiration for SODAP's design came from our experimental work developing HITAP, it is based on a quite different approach. Rather than attempting to determine whether an attack is underway, SODAP continually and dynamically adjusts the number of connections to parent superpeers held by leaf peers, based upon disconnection events locally detected by each peer. This has the additional benefit of allowing the network to self-optimize to avoid disconnections due to peer churn.

Algorithm 1 SODAP Superpeer Maintenance Mechanism

```
t_n ← 1
loop
    for i ← 1 ... hops_max do
        fail_i ← max_i[FAILURESRECEIVEDOFSIZE(i)]
        if i < hops_max then
            ANNOUNCEDISCONNECTION(fail_i, i + 1)
        end if
    end for
    f_max ← max_i[fail_i]
    if NODEISSUPERPEER then
        if WASDISCONNECTED then
            t_n ← t_n + 1
        else if f_max > t_n then
            t_n ← t_n + 1
        else if RAND < Prob_decay(t_n) then
            t_n ← t_n − 1
        end if
        if NEIGHBORCOUNT < t_n then
            CONNECTTONEWSUPERPEER
            if CallwasDisconnected then State ANNOUNCEDISCONNECTION(t_n − 1, 0)
            end if
        else if NEIGHBORCOUNT > t_n then
            DISCONNECTFROMRANDOMSUPERPEER
        end if
    end if
end loop
```

Mechanism. The process followed by each SODAP peer is shown in pseudocode in Algorithm 1. A peer P_n keeps a *parent target* (t_n), which is the number of superpeers to which that peer will attempt to maintain connections, whenever it finds itself in the leaf state (*i.e.*, Myconet's *biomass* state). The initial value of t_n is 1, and in perfect network conditions (where no abrupt disconnections are observed) will remain there.

Similar to the mechanism underlying HITAP, the disconnection signal is propagated by the peer that receives it to its neighbors. The SODAP signal is simpler than the hormonal diffusion of HITAP, propagating for a fixed range ($hops_{max}$, the maximum number of hops). Peers receiving this signal track the highest value $fail_i$ observed within a time window for each hop distance $1 \leq i \leq hops_{max}$, as well as the highest value f_{max} observed overall. $hops_{max}$ is a SODAP protocol parameter; through experimentation, we have observed that a value of 2 is sufficient across for a wide range of scenarios in a Myconet overlay.

When a peer detects that it has been disconnected due to the unexpected failure of a neighbor peer, it responds by incrementing its individual t_n value by one. By creating additional links to multiple superpeers, the chance of the node becoming disconnected due to future failures is reduced. Also, since a superpeer's capacity dictates how many leaf peers it can service, after a significant proportion of leaf peers have raised their targets, new superpeers will be dynamically recruited by the regular topology maintenance protocol of Myconet, in order to handle the additional leaf peers connections.

Any peer receiving this failure announcement may also adjust its parent target in response. If the largest failure announcement a peer P_n receives within a given time window is f_{max}, then P_n will increment its own parent target t_n by one if $f_{max} > t_n$.

Similarly, if the largest failure observed is less than P_n's parent target, P_n may adjust its parent target

downward by applying a decay probability function, in an attempt to reduce unnecessary redundancy and improve efficiency. However, that observed failure must be two less than P_n's current target ($f_{max} < t_n + 1$) before P_n will consider taking this action, since if f is only one less than t_n, a reduction of t_n would place the new target at a "danger" level where a failure was actually observed.

The decay function used determines the probabiilty that P_n will decrement its parent target t_n by one. The function used by SODAP is:

$$Prob_{decay}(t_n) = \frac{1}{1 + e^{(t_n/x)-s}}$$

This is a sigmoid function that provides a relatively small probability of decay if the current parent target is low but rapidly increases that probability as that parent target increases. Experiments have demonstrated that values of $x = 2.5$ and $s = 3$ work well across all tested scenarios.

P_n will make no adjustment to its parent target t_n if the largest failure announcement f_{max} is equal to or one less than its current target.

If a new peer P_c were always to enter the network with an initial parent target of $t_c = 1$, its chances of disconnection might be relatively high if network conditions are poor. In order to take advantage of current peers' knowledge of current network conditions, a peer that is entering the network and is connecting to a superpeer for the first time will set its parent target to mirror the target of that superpeer. Thus, if P_c connects to a superpeer S with a parent target $t_s = k$, P_c will set its own initial parent target $t_c = k$.

After being disconnected and raising its parent target, a peer P_n then attempts to connect to a new superpeer. It then sends a signal to that superpeer that it is making a reconnection, and also communicates the t_n level to which it was operating when the disconnection occurred. Thus, when a peer with $t_n = 2$ is disconnected by failure, it raises its target to $t'_n = 3$ and, after connecting to a superpeer S, announces to S that operating at $t_n = 2$ was insufficient to prevent disconnection. Following connection to S, P_n will continue—if necessary—to make connections to additional superpeers, until it has reached its t_n target.

SODAP Implementation in Myconet. In order to accommodate the self-protection mechanism of SODAP within Myconet, we had to make only simple modifications to the implementation of the Myconet topology maintenance protocol. We added the functionality for failure detection, failure announcements, and maintenance of parent targets as discussed above; moreover, the behavior of peers in certain Myconet protocol states was adjusted, specifically peers assuming the *biomass* (leaf) role, and the *extending hyphal* role.

When a peer P_n has a parent target of $t_n = 1$, it will operate using the normal Myconet rules. Whenever $t_n > 1$ and P_n is in the *biomass* state, it will attempt to connect to multiple superpeers (rather than the single parent of the basic protocol). If a biomass peer has excess superpeer connections above its t_n target, it will instead drop those connections.

This simple local behavior of biomass peers has global consequences: it will gradually cause the increase (or decrease) of the number of hyphal nodes in the network, since the same peer will now be counted as a client multiple times, by different hyphae. No modifications to the hyphal rules are required in order to achieve this scaling behavior.

One further modification is required for superpeers in the *extending* state. Normally, these peers maintain at least one connection to a higher-state (*branching* or *immobile*) superpeer. However, in network conditions where parent targets greater than 1 are needed to maintain network connectivity, this behavior would make *extending* peers into a topological weak point. This is addressed by having the *extending* nodes maintain a number of connections to higher-state superpeers equal to their current parent target.

A final adjustment to the Myconet rules has also been made in order to improve the distribution of the failure alerts through the network. Under normal rules, only *extending* peers will accept connections from new *biomass* nodes entering the network, since higher-state superpeers are typically at or near their desired utilization levels (pulling leaf peers from lower-state superpeers as needed in order to maintain this level). This behavior would cause failure announcements to be concentrated where *extending* peers are connected. Since the number of extending peers in stable networks may be quite low, those announcements would originate only in few locations in the network. Therefore, we allow all peers to accept connections up to 105% of their target number of clients; in this way, the chance that each peer is likely to see an announcement which is reflective of the current state of the network is much improved.

5. Evaluation. As mentioned in Section 2, the research landscape on the self-protection of superpeer overlays via topology adaptation is relatively sparse. Because of that, a benchmark for comparing and contrasting different topology adaptation approaches does not yet exist. For our evaluation, we report in quantitative ways the self-protection benefit of SODAP by comparing it to the same attack on a similar overlay without the topology adaptation mechanism. This allows an assessment of the level of support and the strength of the defense offered by our self-protection protocol.

We have implemented SODAP on top of Myconet, using an existing, cycle-based simulation (described in detail in [25] and [29]) based on the PeerSim framework [10]. PeerSim is a Java-based platform that has been used extensively in the P2P research literature for evaluating a wide variety of peer-to-peer protocols.[1] Simulation is the premier experimental approach for testing, validating and evaluating P2P protocols at scale, due to the difficulty and expense of building, deploying and testing systems with tens of thousands of nodes (or orders of magnitude more) in the lab or in the field [23]. The behavior of the underlying Myconet toplogy maintenance protocol has also been validated in a live distributed environment at smaller scales [14] with an implementation built using the Protopeer framework [8].

5.1. Experiment Design. The Myconet protocol has a number of parameters, and we have chosen their settings based on values used in our previous work for ease of comparison [25, 26, 29]). The C_n parameter controls the target number of inter-hyphal links that branching and immobile nodes maintain, and influences the level of resilience of the superpeer Myconet overlay to disruptive events. For all experiments, we used $C_n = 5$; a value that has been validated empirically in our previous work as providing a desirable balance between resilience and efficiency at very diverse scales. Myconet peer capacities are assigned using a power-law distribution (capped at a maximum value c_{max}) such that the probability of a node having capacity x (with x in the range $[1 \leq x \leq c_{max}]$) is $Prob\,[c_n = x] = x^{-\alpha}$. We chose $c_{max} = 500$ for experiments with networks of 10,000 peers, as this value is used for evaluating similarly scaled protocols in [19] and [13]. Since very large values of c_{max} relative to the number of peers in the network tend to produce degenerate superpeer topologies (with most nodes clustering around a very few nodes with extremely high relative capacity), we reduced c_{max} to 50 for the 1,000-peer experiments.

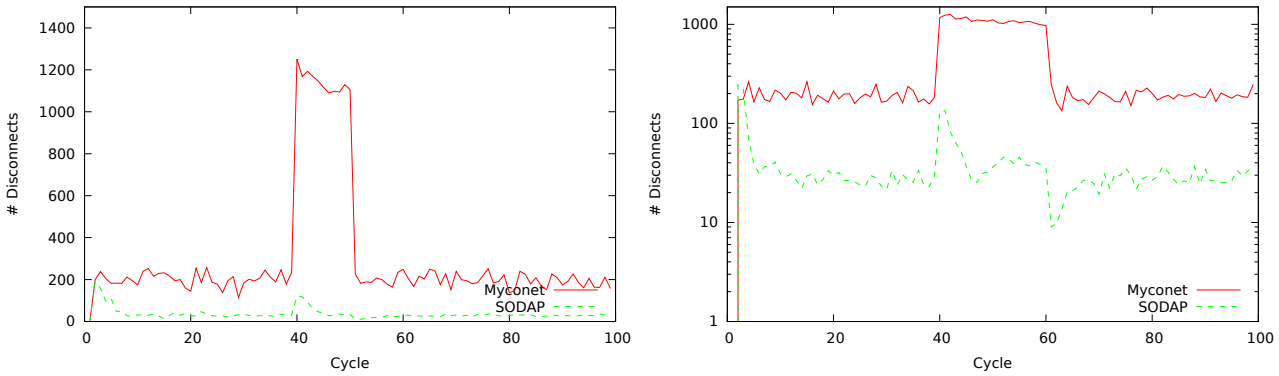
In our experiments, we simulate denial-of-service attacks against the most important peers in the network as used in our previous work [26] (and similar to the attack described in [33]). During this attack, the k highest-degree peers are removed each cycle. Attacks begin at round 40 (after the superpeer topology has been constructed by the Myconet protocol and has stabilized, with easily recognizable, prominent hyphal peers) and continue for either ten or twenty cycles. After these peers are removed, k new peers are added to the network. These peers' capacities are drawn from the same power-law distribution used for the original assignments; this may result in a decrease in the capacity of the most powerful peers over time, as the new peers added are unlikely to be as large.

Churn (the rate at which peers join and leave the network) is a significant factor in the performance of peer-to-peer networks. As SODAP does not explicitly consider the age of peers in its dynamics, the experiments discussed in this section use a simple model of churn, where a fixed percentage of nodes are removed each round, and a similar number are added. While sophisticated models of churn have been examined in the research literature (as discussed in Section 2), SODAP's performance is expected to remain consistent across differing churn models where similar percentages of peers enter and and exit the network within a similar time period. As with the attack scenarios, the peers added during churn may have differing capacities from the peers that were removed.

Our primary metric for the SODAP protocol is the disconnection rate; that is, the number of peers that each cycle lose their connections to the rest of the overlay as the result of another peer's exiting the network, whether from churn or an attack. New peers entering the network for the first time (following an attack or churn operation) are not counted towards this disconnection rate until they have successfully connected to one other peer at least once.

We also use metrics for the average degree of the superpeers (hyphae) and the number of superpeers in the network overall. Where we show the *optimal* number of superpeers in our charts, that number has been

[1] A partial list of the many protocols that have been implemented using PeerSim (including links to publications) can be found at `http://peersim.sf.net/#code`.



(a) 10-round attack (linear y-scale) (b) 20-round attack (log y-scale)

FIG. 5.1. *Disconnection rates for reference experiments (10,000 nodes, 2% churn, attack beginning at cycle 40)*

calculated offline using an oracle: the largest peers in the network are progressively selected until the total capacity of these selected peers is sufficient to service all of the remaining peers in the network.

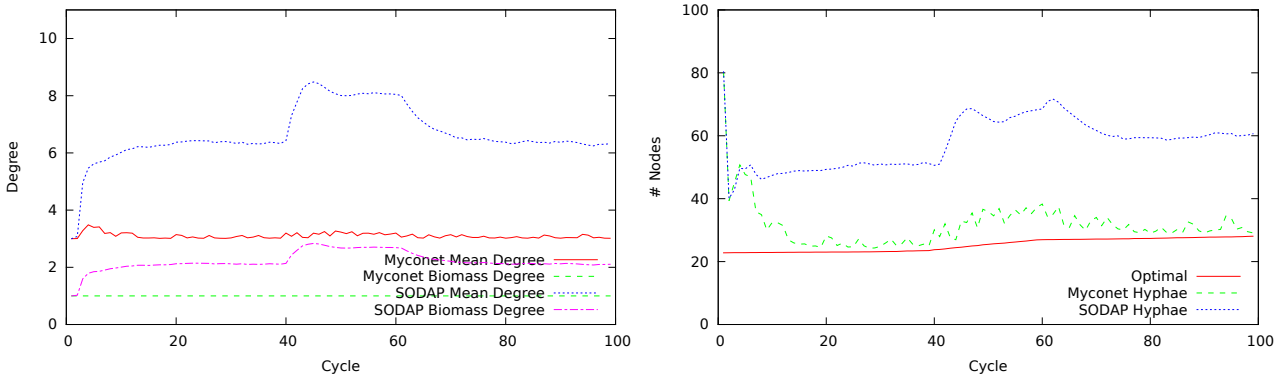
Each experiment discussed in this section consists of 100 independent runs, each lasting 100 cycles; we report the average of a set of metrics over those 100 runs.

5.2. Reference Experiment. We have selected a reference experiment configuration as a baseline for evaluation and discussion, and present the results for that configuration in this subsection. Experimental results discussed in the other subsections of this section vary one parameter of this configuration to facilitate direct comparison of results while examining the performance of SODAP under a range of conditions. The same experiments were also run with the basic Myconet protocol, and results are compared to show the improvements provided by dynamic degree adaptation. Although Myconet was used for expediency and consistency, in these comparisons it is simply a representative superpeer protocol for the maintenance and construction of unstructured superpeer overlays, and we maintain that the results presented hereby can extend to other protocols.

For our reference experiment, we use a network with 10,000 peers, $c_{max} = 500$, $C_n = 5$, and 2% churn per cycle. A single attack is started at round 40 with $k = 2$ (removing the two largest-degree nodes each cycle). Attack lengths of both 10 cycles and 20 cycles were evaluated. Often results for 10 cycles and 20 cycle attacks are similar, so both results are not shown for all experiments.

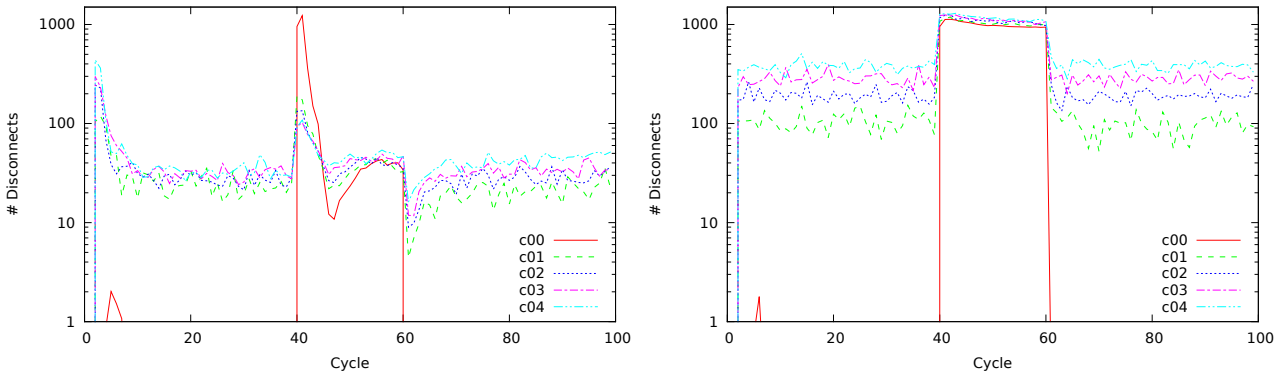
As can be seen in Figure 5.1, during normal network operations with continuous peer churn, SODAP provides a major improvement in the disconnection rate when compared to the basic Myconet protocol, resulting in a reduction of around 90% in the disconnection rate. As can be seen at the start of the experimental run, both basic Myconet and Myconet with SODAP begin with around 200 peers being disconnected as a result of the 2% churn. In response to these detected failures, SODAP then locally increases the parent target of biomass peers, reinforcing the overlay. These effects are shown in Figure 5.2.a, which shows the adaptative response of SODAP: Myconet node degrees do not change, while SODAP degrees quickly increase. The mean degree of biomass peers increases (purple dashed-dotted line) with the increased number of parents, and the overall mean degree of peers in the network increases due the biomass peers as well as to the increased number of of peers that specialize into a superpeer role (blue dotted line). The additional superpeers that promote themselves to handle the additional connections are shown in Figure 5.2.b. This added robustness results in a decrease in the efficiency of the topology: during normal 2% churn, the number of superpeers stays at around 2.5 times the number that would be needed in optimal conditions.

It is during a targeted attack (rounds 40–60) that SODAP shows its merit most clearly. Without degree adaptation, in the basic Myconet network around 11% of all peers are severed each round. Although Myconet's self-healing capabilities may repair the overlay efficiently, the system remains effectively disabled until the attack subsides. SODAP, in comparison, responds quickly as the attack is detected, creating redundant connections that reduce the disconnection rate to one only slightly higher than during normal churn (Figure 5.1).



(a) Mean degree (all peers and biomass peers) (b) Superpeer counts

FIG. 5.2. *Reference experiments (10,000 nodes, 2% churn, 20-round attack beginning at cycle 40)*



(a) SODAP disconnection rate (b) Myconet disconnection rate

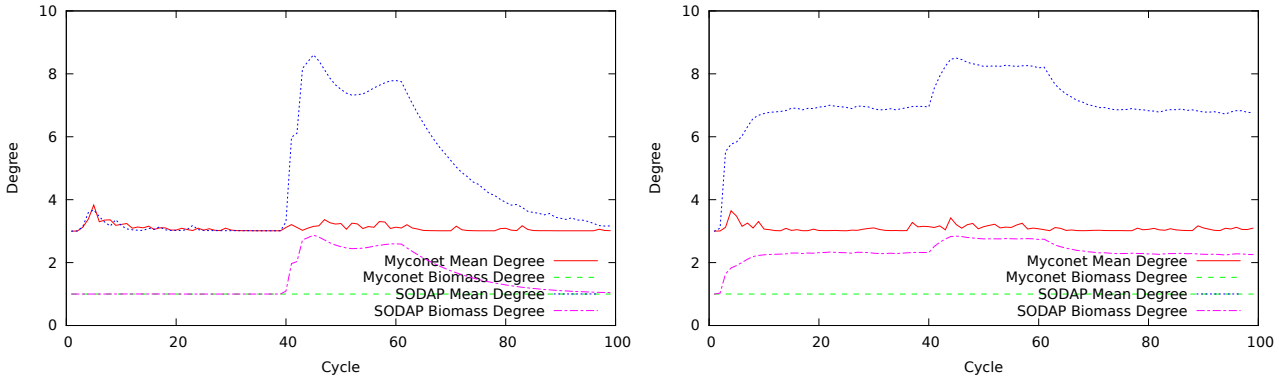
FIG. 5.3. *Disconnection rates varying churn rates (10,000 peers, 20-round attack)*

Following the attack, peers begin to lower their parent targets. This results in an immediate drop on the disconnection rate, which then returns to pre-attack levels as the targets reconverge (shown in log-scale in Figure 5.1.b to provide a better view of these dynamics). Figure 5.2.a shows the peers returning to their pre-target degree levels: the SODAP biomass degree reflects the changes in parent targets (most peers return to a target of 2), while the average degree of the network as a whole (considering superpeers) converges to a mean of just over 6.

5.3. Varying Churn Levels. The performance of SODAP was evaluated under a range of network churn levels. Values from 0% (no churn, with the only network dynamism resulting from attacks) to 4% (which is very considerable churn) were tested. Figure 5.3 shows the effects on the disconnection rate. Both protocol versions remain well-connected when there is no churn, as expected (the small spike near the beginning of the experimental runs for basic Myconet is caused by some nodes being dropped by their neighbors due to normal protocol operations, as the network explores in order to find the most efficient topology). Figure 5.4.a shows that in the absence of churn the parent target values for SODAP nodes remain at one until the attack is detected (Myconet biomass nodes always maintain only a single parent).

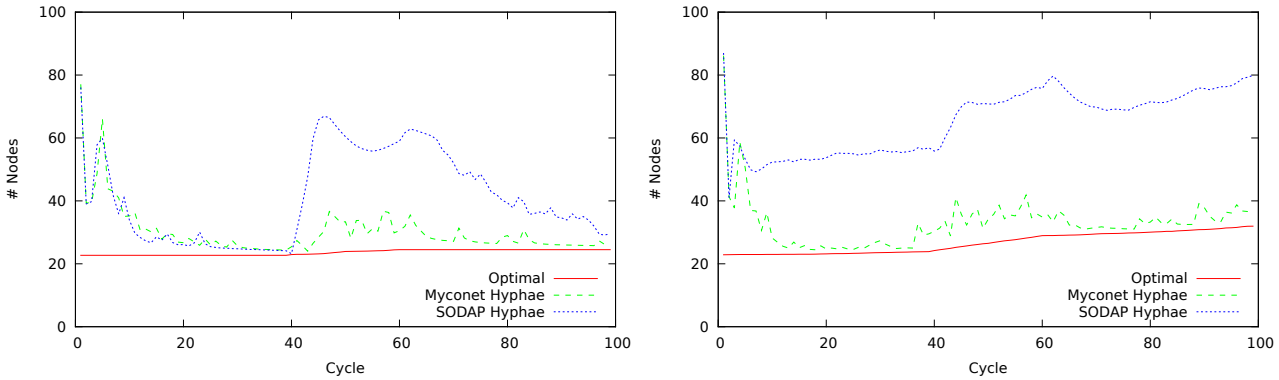
Log-scale Figures 5.3.a and 5.3.b show how the basic Myconet protocol is much more sensitive than SODAP to increasing churn. Myconet's disconnection rate remains roughly equivalent to the churn rate: a four-times increase in the churn rate results in a four-times increase in disconnections. highlights. In contrast, the SODAP disconnection rate stabilizes at around 0.2–0.3% across all evaluated churn levels.

The effect of the attack on the two protocols is quite different. Myconet experiences a 10-11% disconnection



(a) 0% churn (b) 3% churn

FIG. 5.4. *Mean node degrees, varying churn rate (10,000 nodes, 3% churn, 20-round attack beginning at cycle 40)*



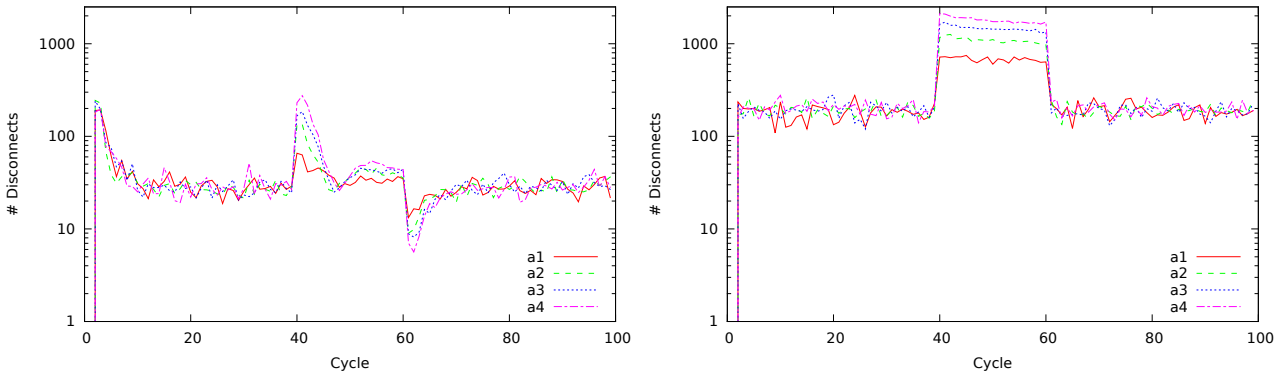
(a) 0% churn (b) 3% churn

FIG. 5.5. *Superpeer counts, varying churn rate (10,000 nodes, 2% churn, 20-round attack beginning at cycle 40)*

rate as long as the attack continues. The targeted attack also hits the no-churn SODAP network hard at first, with around 10% of the nodes being disconnected for the first two rounds; within a few cycles, though, this drops greatly as SODAP increases the parent targets, converging to around 0.3% (Figure 5.3.a). Networks that have been subjected to higher levels of churn are strengthened against the attack: even with a churn level of 1%, the first two rounds of attack result in a SODAP disconnection rate of less than 2%, and a churn level of 4% results in the attack disconnecting around 1% of nodes for the first two rounds. At all churn levels, SODAP converges to a during-attack disconnection rate of around 0.3–0.4%, slightly higher than the rate during normal operation.

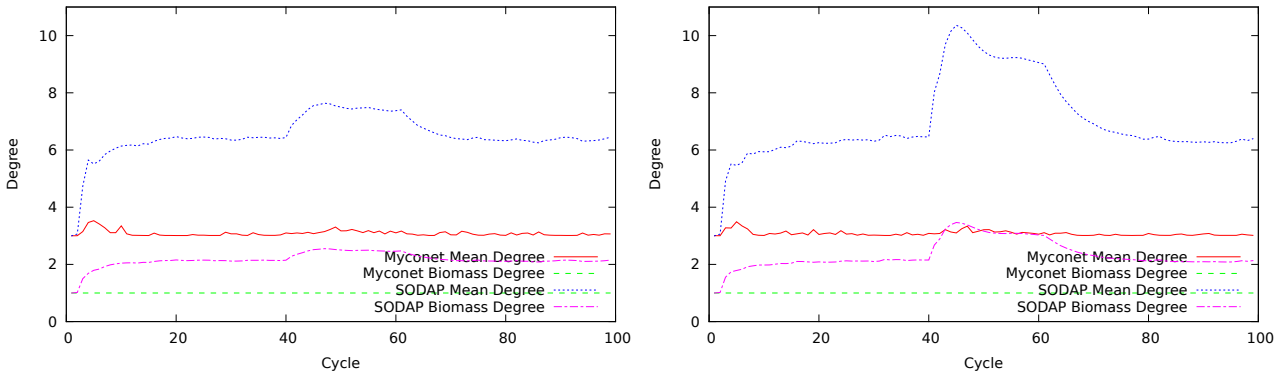
Following the attack, basic Myconet returns to its pre-attack disconnection rate, while SODAP's disconnection rate is actually somewhat improved, particularly at higher churn levels. The reason for this can be seen in Figures 5.4 and 5.5: the biomass target is elevated during the attack (to a mean of just under 3 in both scenarios shown), but then falls off gradually following the attack. The fall-off is much slower for the 0% churn case due to the sigmoid shape of the decay function. A single parent is insufficent to protect from disconnection during churn, but is the ideal state when churn is not present, and the decay rate to a single parent is relatively slower due to the increased risk. The capacity of largest nodes in the network also decreases, on average (shown by the increasing number of peers required to achieve optimality, the solid red line in Figure 5.5). Thus, a proportionally higher number of superpeers must be promoted and maintained.

5.4. Attack Scenarios. We also evaluated the effects of attacks that target a varying number of the highest-degree superpeers (from 1 to 4 per cycle). As Figure 5.6.a shows, the effect of increasing attack size on



(a) SODAP disconnection rate (b) Myconet disconnection rate

FIG. 5.6. *Disconnection rates, varying attack size* $1 \leq k \leq 4$ *(10,000 peers, 20-round attack)*



(a) Attack $k = 1$ (b) Attack $k = 4$

FIG. 5.7. *Mean node degrees, varying attack size k (10,000 nodes, 2% churn, 20-round attack beginning at cycle 40)*

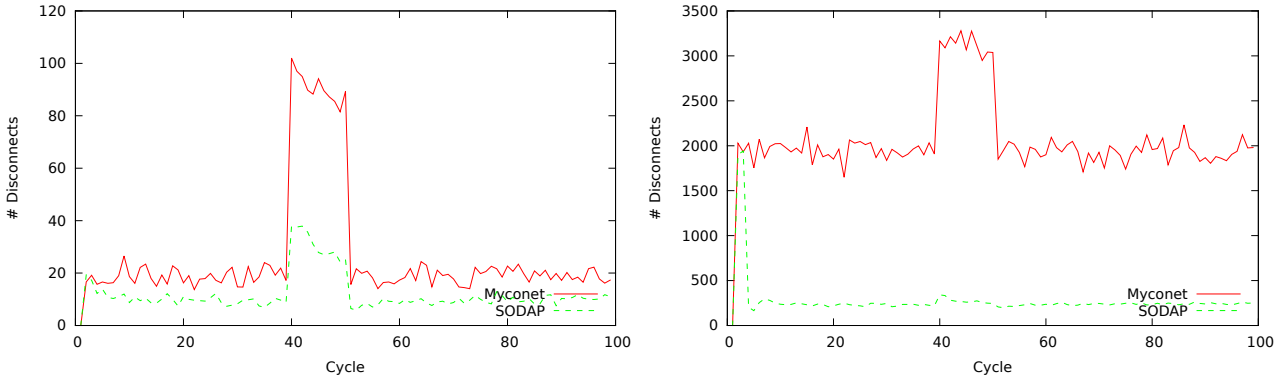
Myconet with SODAP is most visible at the beginning of the attack, determining the size of the initial spike (around 0.5% for an attack size of $k = 1$, up to around 2% for $k = 4$). As the attack continues and the network degree adapts to avoid disconnection, this drops to levels slightly above the disconnection rate for background churn (from around 0.25% for $k = 1$ to around 0.35% for $k = 4$). The behavior of basic Myconet is much less resilient; an attack of $k = 1$ already results in a disconnection rate of around 5%, up to nearly 11% for for $k = 4$.

Following the attack, basic Myconet quickly returns to pre-attack disconnection rates, while SODAP shows much larger dips for larger attacks before stabilizing. This reflects the additional network connections that were established in order to maintain connectivity.

The greater increase in degree by SODAP for larger attacks is shown in Figure 5.7. In response to an attack of $k = 1$, the average network degree increases to around 7.5, returning to pre-event levels following the end of the attack. An attack of $k = 4$, on the other hand, is extremely significant, as many nodes are likely to be affected by the abrupt removal of this many high-degree superpeers. To protect against this, SODAP initially raises the mean node degree to around 10, stabilizing at around 9 as the attack continues. After the attack ends, the average degree drops as the parent targets decay.

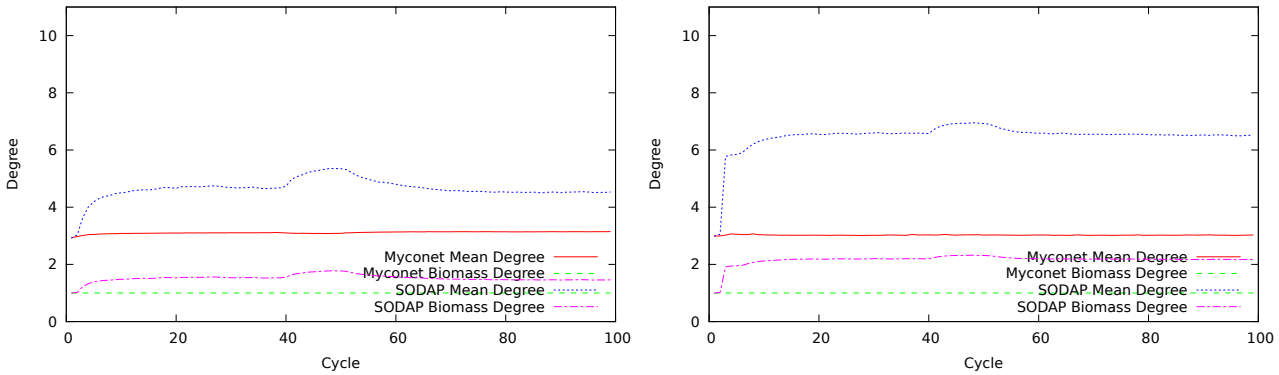
5.5. Network Scale. The performance of SODAP at different network scales was also evaluated, from 10^3 to 10^5. These results are shown in Figures 5.8 and 5.9 (results for 10^4 are found in Section 5.2, above).

For networks of 1,000 nodes (with $c_{max} = 50$) and 2% churn, the basic Myconet protocol stabilizes at a disconnection rate of around 2%. SODAP reduces this to about 1%. SODAP also reduces the increase in disconnection rate during an attack, from around 9% for basic Myconet to around 3.5%, which is comparable



(a) 1,000 nodes (b) 100,000 nodes

FIG. 5.8. *Disconnection rates, varying network size (2% churn, 10-round attack)*



(a) 1,000 nodes (b) 100,000 nodes

FIG. 5.9. *Mean node degrees, varying network size k (2% churn, 10-round attack)*

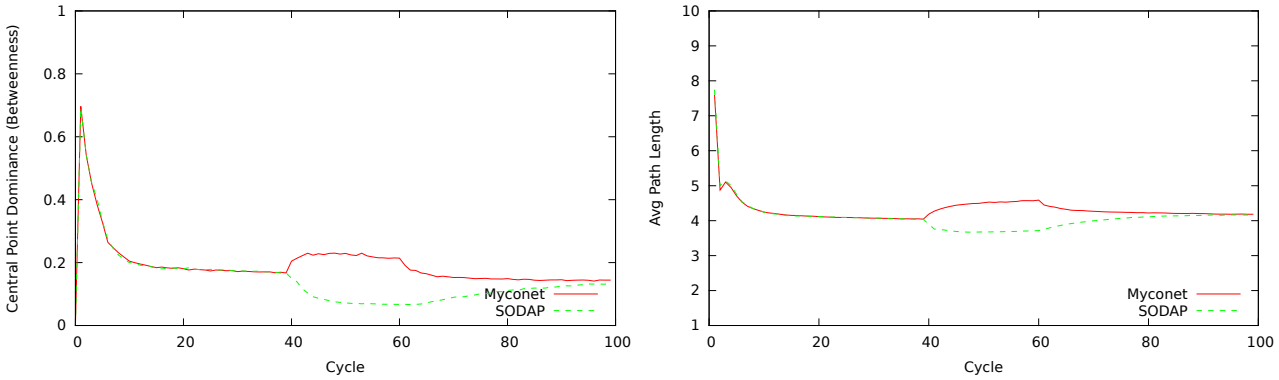
to the 10,000 node case.

At 100,000 nodes, Myconet's disconnection rate remains proportional at 2%, while Myconet with SODAP begins at this level but quickly drops to around 0.25%. Targeted attacks on the two largest superpeers ($k = 2$) have, of course, a much lesser effect at this larger scale. During such an attack, basic Myconet jumps to a rate of around 3% (versus 11% for 10^4). SODAP's response is comparable to the smaller network, rising slightly to around 0.3%, as compared to 0.25% for 10^4.

The effect that SODAP has on node degrees is shown in Figure 5.9. For networks of 10^3 nodes, a mean parent target of around 1.5 is selected, which rises to close to 2 during the attack. This is reflected in the overall difference in mean degree for all nodes in the network, which is around 4.7 (compared to 6.2 for 10^4). At 10^5, the results are slightly higher than those for 10^4, with a mean parent target of just above 2 and mean degree around 6.5.

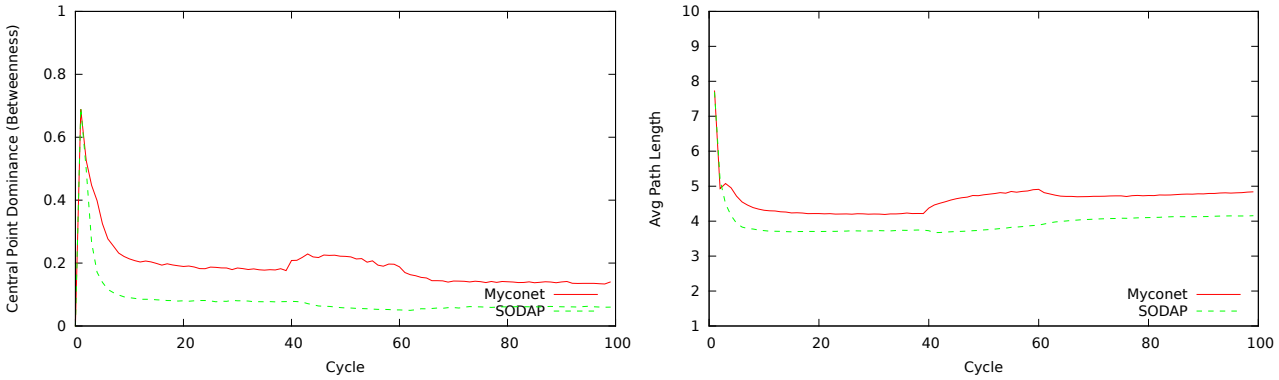
5.6. Other Metrics. Figures 5.10 and 5.11 show the central point dominance (CPD) of the largest connected component (using betweenness centrality) [7] and average path lengths for experimental runs with 10^3 nodes.

As can be see in Figure 5.10.a, without churn, these metrics remain relatively close until the targeted attack begins at round 40. At this point, the degree of the leaf peers rises when SODAP is used, resulting in a drop in the level of centralization in the network. Similarly, Figure 5.10.b shows that this results in slight drop in average path length, as peers' parent targets increase. Without SODAP, the average path length increases during the attack.



(a) Central Point Dominance (Betweenness) (b) Average Path Length

FIG. 5.10. *Graph metrics for network with 1,000 nodes, no churn, 20-round attack beginning at cycle 40*



(a) Central Point Dominance (Betweenness) (b) Average Path Length

FIG. 5.11. *Graph metrics for network with 1,000 nodes, 2% churn, 20-round attack beginning at cycle 40*

With 2% churn, the difference in central point dominance is readily apparent in Figure 5.11.a; with SODAP, it is lower due to the higher average degree of the leaf peers. CPD drops slightly further during the attack, and slowly returns to pre-attack levels after its conclusion as the parent targets decay. Figure 5.11.b shows average path lengths; with SODAP, path lengths remain around 4 before, during and after the attack (though it can be seen that the level following the attack is slightly higher due to the decreased maximum capacity from targeted removal of those peers). Without SODAP's degree adaptation, the path length rises during the attack, and settles to a slightly higher level after it ends.

6. Conclusions. In this paper, we have proposed the *SODAP* strategy for the self-organized protection of unstructured superpeer overlays from topology attacks. SODAP effectively protects the overlay from targeted attacks against high-degree nodes, which can be a significant vulnerability of superpeer-based P2P networks.

We evaluated our approach using a modified version of the Myconet superpeer overlay, and compared its performance with the non-adaptive approach. The experimental results show that, relative to the non-degree-adaptive approach, SODAP provides a very significant reduction (around 90%) in peer disconnection rate for networks under attack, across a range of network sizes and conditions.

A key advantage of SODAP is its ability to seamlessly adapt in both directions, increasing node degrees in response to detected disconnections, but also decreasing them in the absence of failures. SODAP improves previous work in this area, and is effective across multiple network scales, churn levels, and attack sizes, as well as adapting smoothly (through the exploratory reduction in degree of a small number of peers) to improvements in network conditions without requiring the reversion of the entire topology.

SODAP's self-protection mechanism has a smaller parameter space than previous work, making it more flexible and easier to apply and tune. An in-depth examination of the the effects of varying SODAP's rules and parameters can be found in [24].

The protocol focuses on disconnections (rather than attempting to guess whether an attack is underway), an approach that has the advantage of being directly, locally observable. Since SODAP nodes attempt to minimize disconnections no matter what their cause, the problem of possible false positives and false negatives does not need to be considered.

Though the implementation in this paper is built on top of Myconet, SODAP is applicable to any unstructured peer-to-peer network that uses a superpeer topology, independent of the protocol that is used for superpeer selection. One key property of SODAP is that it exploits heterogeneity that typically occurs among overlay participants, in particular at large scale, and is designed to enable the superpeer network to continue taking advantage each peer's individual capabilities, even when under targeted attack. The selected, high-degree superpeers are leveraged in SODAP itself, as they act as both connection points for reconnecting peers and conduits for announcing these reconnections to a relatively large number of other peers; in this way, a few disconnection messages are able to spread through a system that is both attempting to minimize disconnections and minimize the degree of leaf peers, which helps improve efficiency.

6.1. Future Work. We plan to compare this adaptive mechanism against topology preservation overlays that use a fixed number for the parent target value of leaf peers. We also plan to test the performance of SODAP outside of a simulation environment and in a live network environment, using our recent Protopeer-based implementation of Myconet.

Balancing the upward and downward pressure on node degrees results in a disconnection rate that is very low, but is not zero. Some inefficiency (in the form of actual disconnected nodes) is inevitable, and in fact needed to keep the SODAP system stabilized at a particular level of parent targets. A possible improvement may be to have nodes adjust as a result of "near" disconnections (*i.e.*, if they were reduced to a single neighbor within the time window). This may allow a larger percentage of nodes to avoid disconnection at the price of slightly higher average degree.

The new topologies constructed by the local actions of peers (increasing the degree of leaf peers by adding additional superpeer connections) are influenced by the underlying overlay protocol. One direction for future experiments is to apply SODAP's strategy to a superpeer-based overlay other than Myconet. Another is to examine additional neighbor-selection strategies for constructing the reinforced topologies.

SODAP focuses on self-protection from attacks that result in disconnection of peers. Many other types of attacks against peer-to-peer networks have been examined in the literature, such as those discussed in the attack taxonomy in [31]. Direct attacks against learning mechanisms or the self-organized behavior itself are also possible: for example, peers might report false values for disconnections, or for their current parent targets. We would like to examine the effect of these different types of attacks on our strategy. Moreover, within the bounds of the discussed attack itself, we also plan to perform more detailed analysis of the trade-offs between sensitivity to attacks and the inefficiency induced by the increase in node degrees.

REFERENCES

[1] S. Baset and H. Schulzrinne, *An analysis of the skype peer-to-peer internet telephony protocol*, in INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, April 2006, pp. 1–11.

[2] N. Basher, A. Mahanti, A. Mahanti, C. Williamson, and M. Arlitt, *A comparative analysis of web and peer-to-peer traffic*, in Proceedings of the 17th international conference on World Wide Web, ACM, 2008, pp. 287–296.

[3] B. Beverly Yang and H. Garcia-Molina, *Designing a super-peer network*, in Data Engineering, 2003. Proceedings. 19th International Conference on, 2003, pp. 49–60.

[4] M. Brinkmeier, G. Schafer, and T. Strufe, *Optimally DoS resistant P2P topologies for live multimedia streaming*, Parallel and Distributed Systems, IEEE Transactions on, 20 (2009), pp. 831–844.

[5] C. Diwakar, *Security threats in peer to peer networks*, Journal of Global Research in Computer Science, 2 (2011).

[6] M. Engle and J. I. Khan, *Vulnerabilities of P2P systems and a critical look at their solutions*, Tech. Report TR2006-11-01, Kent State University Medianet Lab, 2006.

[7] L. Freeman, *A set of measures of centrality based on betweenness*, Sociometry, 40 (1977), pp. 35–41.

[8] W. Galuba, K. Aberer, Z. Despotovic, and W. Kellerer, *Protopeer: a p2p toolkit bridging the gap between simulation and live deployement*, in Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 60.

[9] C. G. Ghedini and C. H. Ribeiro, *Rethinking failure and attack tolerance assessment in complex networks*, Physica A: Statistical Mechanics and its Applications, 390 (2011), pp. 4684–4691.

[10] M. Jelasity, A. Montresor, G. P. Jesi, and S. Voulgaris, *The Peersim simulator*. http://peersim.sf.net.

[11] P. Keyani, B. Larson, and M. Senthil, *Peer pressure: Distributed recovery from attacks in peer-to-peer systems*, in Revised Papers from the NETWORKING 2002 Workshops on Web Engineering and Peer-to-Peer Computing, London, UK, UK, 2002, Springer-Verlag, pp. 306–320.

[12] P. Lewis, H. Goldingay, and V. Nallur, *It's good to be different: Diversity, heterogeneity, and dynamics in collective systems*, in Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2014 IEEE Eighth International Conference on, Sept 2014, pp. 84–89.

[13] W. Liu, J. Yu, J. Song, X. Lan, and B. Cao, *Erasp: An efficient and robust adaptive superpeer overlay network*, in Progress in WWW Research and Development, Y. Zhang, G. Yu, E. Bertino, and G. Xu, eds., vol. 4976 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 468–474.

[14] D. Lucia, *Mycocloud: Improving QoS by managing elasticity of services in decentralized clouds*, master's thesis, 2013.

[15] B. Mitra, M. Afaque, S. Ghose, and N. Ganguly, *Developing analytical framework to measure robustness of peer-to-peer networks*, in Distributed Computing and Networking, S. Chaudhuri, S. Das, H. Paul, and S. Tirthapura, eds., vol. 4308 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 257–268.

[16] B. Mitra, A. Dubey, S. Ghose, and N. Ganguly, *Formal understanding of the emergence of superpeer networks: A complex network approach*, in Distributed Computing and Networking, K. Kant, S. Pemmaraju, K. Sivalingam, and J. Wu, eds., vol. 5935 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 219–230.

[17] B. Mitra, F. Peruani, S. Ghose, and N. Ganguly, *Analyzing the vulnerability of superpeer networks against attack*, in Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007, pp. 225–234.

[18] B. Mitra, F. Peruani, S. Ghose, and N. Ganguly, *Measuring robustness of superpeer topologies*, in Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing, PODC '07, New York, NY, USA, 2007, ACM, pp. 372–373.

[19] A. Montresor, *A robust protocol for building superpeer overlay topologies*, in Proceedings of the 4th IEEE International Conference on Peer-to-Peer Computing, Zurich, Switzerland, Aug. 2004, IEEE, pp. 202–209.

[20] P. Perlegos, *DoS defense in structured peer-to-peer networks*, Tech. Report UCB/CSD-04-1309, University of California, Berkeley, 2004.

[21] N. Ramzan, H. Park, and E. Izquierdo, *Video streaming over p2p networks: Challenges and opportunities*, Image Communication, 27 (2012), pp. 401–411.

[22] K. Samant and S. Bhattacharyya, *Topology, search, and fault tolerance in unstructured P2P networks*, in System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, 2004, pp. 6 pp.–.

[23] G. Shi, Y. Long, H. Gong, C. Wan, C. Yu, X. Yang, and H. Zhang, *A high scalability p2p simulation framework with measured realistic network layer support*, in Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International, IEEE, 2008, pp. 311–318.

[24] P. L. Snyder, *Modeling and Engineering Self-Organization in Complex Software Systems*, PhD thesis, Drexel University, 12 2013.

[25] P. L. Snyder, R. Greenstadt, and G. Valetto, *Myconet: A fungi-inspired model for superpeer-based peer-to-peer overlay topologies*, in Self-Adaptive and Self-Organizing Systems, 2009. SASO '09. Third IEEE International Conference on, 2009, pp. 40–50.

[26] P. L. Snyder, Y. Osmanlioglu, and G. Valetto, *Biologically inspired attack detection in superpeer-based P2P overlay networks*, Bio-Inspired Models of Networks, Information, and Computing Systems, (2012), pp. 99–114.

[27] A. Srivastava, B. Mitra, F. Peruani, and N. Ganguly, *Attacks on correlated peer-to-peer networks: An analytical study*, in Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, 2011, pp. 1076–1081.

[28] K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Fujii, and A. Takahara, *Thup: A p2p network robust to churn and dos attack based on bimodal degree distribution*, IEEE Journal on Selected Areas in Communications, 31 (2013), pp. 247–256.

[29] G. Valetto, P. L. Snyder, D. J. Dubois, E. Di Nitto, and N. M. Calcavecchia, *A self-organized load-balancing algorithm for overlay-based decentralized service networks*, in Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on, 2011, pp. 168–177.

[30] T. Yi-Hong, L. Xi-Dao, L. Ya-Ping, and Z. Bi-hai, *DLPSPN: New efficient super-peer network based on double-loop Petersen graph*, in Network Computing and Information Security (NCIS), 2011 International Conference on, vol. 2, IEEE, 2011, pp. 201–205.

[31] X. Yue, X. Qiu, Y. Ji, and C. Zhang, *P2P attack taxonomy and relationship analysis*, in Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on, vol. 02, feb. 2009, pp. 1207–1210.

[32] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, *Tapestry: A resilient global-scale overlay for service deployment*, Selected Areas in Communications, IEEE Journal on, 22 (2004), pp. 41–53.

[33] K. A. Zweig and K. Zimmermann, *Wanderer between the worlds - self-organized network stability in attack and random failure scenarios*, in Self-Adaptive and Self-Organizing Systems, 2008. SASO'08. Second IEEE International Conference on, IEEE, 2008, pp. 309–318.

Edited by: Giacomo Cabri and Emma Hart
Received: Dec 23, 2014
Accepted: Jul 15, 2015