An Adversarial Evaluation of Network Signaling and Control Mechanisms

Kangkook Jee[1], Stelios Sidiroglou-Douskos[2], Angelos Stavrou[3], and Angelos Keromytis[1]

[1] Department of Computer Science, Columbia University
{jikk,angelos}@cs.columbia.edu
[2] Computer Science and Artificial Intelligence Laboratory, MIT
stelios@csail.mit.edu
[3] Department of Computer Science, George Mason University
astavrou@gmu.edu

Abstract. Network signaling and control mechanisms are critical to coordinate such diverse defense capabilities as honeypots and honeynets, host-based defenses, and online patching systems, any one of which might issue an actionable alert and provide security-critical data. Despite considerable work in exploring the trust requirements of such defenses and in addressing the distribution speed of alerts, little work has gone into identifying how the underlying transport systems behave under adversarial scenarios.
In this paper, we evaluate the reliability and performance trade-offs for a variety of control channel mechanisms that are suitable for coordinating large-scale collaborative defenses when under attack. Our results show that the performance and reliability characteristics change drastically when one evaluates the systems under attack by a sophisticated and targeted adversary. Based on our evaluation, we explore available design choices to reinforce the reliability of the control channel mechanisms. To that end, we propose ways to construct a control scheme to improve network coverage without imposing additional overhead.

1 Introduction

The prevalence and effectiveness of large-scale malware phenomena (worms, botnets, web-based malware) has led to the development of several automated defenses that detect new threats and generate various kinds of fixes such as patches, filters. The security literature is rife with distributed security systems [7,5] which assume that reliable, scalable and robust Content Distribution Network (CDN) functionality is universally available. To date, the primary metrics of effectiveness have been propagation time (latency and throughput) and node coverage in the presence of "natural" phenomena such as churn. However, the conspicuous absence of an adversarial analysis, both in terms of performance impact and security guarantees (*e.g.,* susceptibility to man-in-the-middle attacks), is of particular concern as the control channel for security data is a very attractive target for adversaries. This is especially true for systems that make design decisions that favor performance over robustness (*e.g.,* using a centralized tracker in BitTorrent).

We argue that such a narrow view of system performance is inadequate and even dangerous in the presence of malicious adversaries. In other areas of security (spam, honeypots and honeynets, anti-virus), we have seen active targeting of protection mechanisms and, occasionally, their hijacking and use for malicious purposes. Instead, we need to consider system behavior in the presence of intelligent, targeted interference by botnets and other malware. At a minimum, these systems must be able to withstand attacks that seek

to disrupt their primary function: the timely and reliable delivery of security-critical data to all benign participating nodes and users.

To this end, we conduct an evaluation of control channel mechanisms that have been proposed for use in distributing security-critical data at massive scale. Specifically, we evaluate different approaches of centralized, distributed, and hybrid designs in presence of global adversary. We recognize that this is only part of the security-oriented evaluation criteria that such systems should be subjected to; however, we strive for an in-depth analysis of a particular aspect of system behavior rather than a shallower examination of more features. A key contribution of our work is a detailed analysis of existing control channel mechanisms in a number of realistic adversarial scenarios. Rather than limiting our measurements to simple latency and throughput characterizations, metrics of *coverage*, *latency*, and *control efficiency* are considered. We use these to investigate the trade-offs between system performance and resilience to certain type of attacks. Thus, our work explores the spectrum of possible design choices when creating and deploying a distribution mechanism for security-critical data.

As a result of evaluation, we find that centralized designs introduce fragile failure points, centralized entities, or hierarchical indirection, that can cripple performance and reliability when attacked. Distributed mechanisms also cease to function upon failure of nodes more than a certain threshold. Furthermore, the attacker can escalate his impact on distributed mechanisms by taking advantage of heterogeneity of network knowledge among participants. Extending reliability to some extent, the hybrid mechanism still inherits the shortcomings of both centralized and distributed systems. To maximize the reliability benefit of the hybrid mechanism, we explore the design choices available on integrating two contrasting schemes without sacrificing control efficiency.

The road-map of the paper is as follows. After discussing background work (Section 2), the adversarial scenarios are provided in Section 3. In Section 4, we show how we implemented control mechanisms for evaluation. Section 5 delivers evaluation results. Our analysis on these results are presented in Section 6 and the paper is concluded in Section 7.

2 Background

2.1 Control and Signaling Approaches

In contrast to the data transfer channel, the control channel performs its task by signaling small sized management packets to the participating peers. The signaling channel is responsible for: *i)* peer join and leave, *ii)* locating objects *iii)* resource scheduling and allocation *vi)* authentication, integrity, and authenticity and *v)* application specific tasks – for instance, a system for alert distribution raises an alarm of urgent security events using this channel. Traditionally, there were two fundamental but contrasting schools of thought regarding the design and implementation of the signaling mechanisms – centralized and distributed. Recently, there are attempts to leverage the strengths of both distributed and centralized schemes while avoiding some of their weaknesses by using a hybrid approach.

Centralized schemes These simple and efficient mechanisms require one or a small set of

centralized entities to coordinate the operations of the entire system. However, the scalability of the system is limited by the network and processing capacity of the control nodes. As a workaround, a hierarchical control network [27] consisting of super-nodes (SNs) was proposed. The control plane is implemented by adding layer of super-nodes which act as the leaders and are in charge of their own sub-networks. Unfortunately, selecting the right super nodes and the size of the clustering for each sub-network is still an open problem. This is further exacerbated in dynamic environments with many joins and leaves. Moreover, akin to the pure centralized solution, each super-node is single point of failure to its own sub-network.

Distributed schemes This class of mechanisms is designed to mitigate the scalability problems of the centralized design. Their design can be accomplished using either structured or unstructured overlay networks. Distributed Hash Tables (DHTs) [25,13,22] is a structured overlay solution that are leveraging the power of consistent hashing [9]. On the other hand, the gossip-based information sharing protocols [18], also known as epidemic or flooding protocols, process requests from clients in unstructured way. The core implementation relies in flooding search requests to peering neighbors. Nevertheless, despite their numerous benefits, distributed solutions also come with their own limitations. DHT-based approaches do not work well in practice [20] as their performance is severely influenced by even a small fraction of slow performing nodes. Moreover, the gossip protocol becomes very costly as the size of network grows and has difficulty in locating information with low availability. In following evaluations, we use DHTs to implement distributed control channel.

Hybrid schemes These signaling mechanisms attempt to combine the advantages of the centralized and the distributed (DHTs) design principles. During normal operations, a hybrid system uses a fast and efficient centralized channel. It can, however, switch to a slower but also more robust distributed channel to resolve capacity overload or even node failures. There is a wealth of recent research on hybrid designs [29,8,10] all of which share the same basic design principles with minor modifications. Moreover, there are systems that attempt to combine two distributed mechanisms of structured (DHTs) and unstructured (gossip protocol) to achieve better search efficiency [12,28].

2.2 Reliability Analysis of Signaling Channels

There exist some previous works that focus on analyzing the reliability of network systems and the security protection of control channels. For reliability of centralized systems, there is work relate to the stability of super-node networks. Yang et al. [27] suggest general guidelines in designing super-node networks and about principles for reliable design. Mitra et al. [15] propose an analytic framework that correlates super-nodes' fraction and their network connectivity with reliability. This work also considers a global adversary of different knowledge and power. It does not, however, address hybrid schemes or provide any comparison between systems. Distributed control systems are designed to be more reliable but they introduce new threats and vulnerabilities which exploit the specifics of each architecture. To counter the shortcomings, researchers proposed a number of implementations [21] and theoretical studies [11,2] with the aim to improve the reliability of distributed systems. Specific example of control channel which supervises the entire

system's operation and becomes a viable target to the adversary is the BitTorrent tracker network. Although originally designed as a centralized control, extensions have been proposed to enhance the reliability of the tracker by having distributed tracker or multiple trackers. Unlike previous studies on BitTorrent [4,19] where the primary interests were performance related factors such as latency and fairness of resource utilization, recent studies [16,17,14] focus more on the system's reliability.

2.3 Secure Message Propagation Systems

The goal of alert distribution systems is to deliver small size messages to many participants under a strict time constraint. Ever since fast, self-replicating worms (for instance Slammer and Nimda viruses) crippled the Internet, there have been many theoretical studies [30,1,26,24] to build an alert distribution system which can compete against such worms. The outcome of this line of research was guidelines regarding how fast the patch propagation should be. However, none of these works consider scenarios of active adversary who also wants to take over the alert propagation processes.

In addition, RapidUpdate [23] is a research performed by research groups of commercial security vendors. It offers a specific solution to their own alert propagation model. The goal of the system is to propagate small sized alert messages (less than 200K) and meet distribution deadlines. Having assistance from peers, the RapidUpdate tries to alleviate the workload of servers/vendors. Another work [7] by a major software vendor quantifies the performance of the world's biggest patch distribution system – Microsoft's Windows update. Based on trace analysis, this work delivers interesting observations on traffic characteristics of patch distribution and end-user's behavioral patterns. Nonetheless, no previous study considers the presence of a sophisticated adversary that attempts to disrupt the operation of the alert distribution network.

3 Application Environment and Adversarial Scenarios

The key element of our work is the evaluation of different mechanisms for implementing a rapid and reliable alert distribution system in the adversarial context. Previous analyses of such systems were largely done without taking into account sophisticated (or, in many cases, even simple) adversaries who might seek to disrupt the operation of the system. Such disruption may, for example, be attempted in parallel with an attack, so as to maximize its impact and minimize the effectiveness of any defenses.

The goal of the adversaries would be to delay distribution and delivery of such alerts, or to prevent their delivery altogether to as large a fraction of the nodes as possible. We consider different adversaries, at varying levels of sophistication and resources. For generality, our evaluation considers the *impact* such adversaries would have on the system, in terms of inhibiting communications to/from some fraction of nodes.

The sophistication of the adversaries in our threat model is determined in terms of their ability to collect reconnaissance on the internal structure of the alert distribution mechanism and focus their attack. Thus, at a high level, we distinguish between two types of adversaries:

- Adversary with **random attack**: Unsophisticated adversaries who can inhibit communication to/from randomly selected nodes. The fraction of nodes they can bring down depends on the level of resources available to them.
- Sophisticated adversaries, who exploit knowledge of the system structure to target nodes such that they maximize the impact of their disruption. We further consider two sub-types of such adversaries:
  1. Adversary with **targeted attack**: Attackers that know and exploit the high-level structure of the network topology. Such attackers, for example, know the identity of and target the super-nodes or other, relatively "fixed" important nodes in the system.
  2. Adversary with **degree dependent attack**: More powerful adversaries that somehow have detailed topology information about a large part or all of the distribution mechanism. Such knowledge includes, for example, the complete connectivity graph of the participating nodes (or a large fraction thereof).

For all type of the above schemes, selected victim nodes are taken out from the system as a consequence of the attacks.

4 Implementation

For our evaluation, three different alert distribution systems were implemented on OverSim [3] network simulation framework. Here, we describe how we implemented the simulation modules. We first talk about the design choices for the signaling channels and the various reliability parameters that we explored. Then, we cover communication models considered for alert distribution systems.

4.1 Control Channel

Centralized System In the case of centralized control, we employed a super node (SN) network. Among many configuration parameters [27] for the SN network, we carefully identified the ones that affect the robustness of the overall network: the size of sub-network (*cluster size*) and number of super-node replicas (*k-redundancy*). The *cluster size* was tested using a range of different values. The same holds for *k-redundancy*. However, in our graphs, we present only the case where *k-redundancy* is two. We did so because other values of *k-redundancy* do not notably change the system's behavior beyond the one captured by the graphs. We configured the rest of the parameters unchanged as these parameters have an effect only on the network performance.

Distributed System For distributed control, we chose Chord [25] to implement a decentralized alert notification system. Chord was selected for two reasons. First, Chord's ring-based routing structure and ID space has been well-studied allowing us to compare our performance results with others when the network is not under attack. This validates our approach beyond the results of a mere simulation. Second, the structural differences among variants of DHT implementations are not discernible in terms of robustness. Indeed, most of the hash-based systems use a common architecture that employs key-based routing [6]. Among many configurable parameters for Chord, we considered *successor*

list size to be the most important one to the reliability and stability of the system. This was varied with different values to see its impact on the system's maintenance cost and reliability.

Hybrid System This model aims to achieve better network performance similar to the centralized systems while maintaining the reliability of the purely decentralized approaches. In hybrid systems, all nodes initially join both a decentralized and a centralized signaling channel. For instance, a super-node in the hybrid network is the centralized entity for its sub-network as well as a regular participant in the DHT channel. Therefore, the hybrid designs inherit all their configuration parameters. Moreover, peers in the hybrid network can utilize the primary (centralized) and secondary (decentralized) signaling channels either in *serial* or *parallel*. In our implementation of hybrid systems, frequent operations such as querying were done first using the centralized and then the decentralized signaling path. This increases performance under normal operations while maintaining robustness in case of attacks. However, for less frequent but more critical functions, such as publishing new information, we used both channels at the same time to increase resilience without severely impacting the performance of the network.

4.2 Models for Alert Distribution

Publish-subscribe model In this model, peers have the option to subscribe to certain classes of security events. *Polling* and *pushing* are available choices to implement this model. For our experiments, we used the *polling* model with 30 seconds of polling interval. This is a cost-effective and easy-to-implement solution, widely adopted by most vendors for their online patching system.

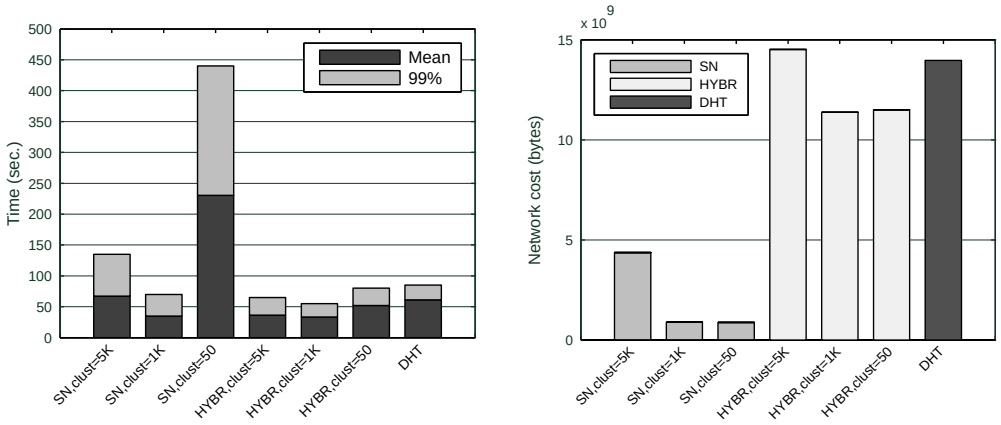
Distributed sensors model In this model, participants with proper permission can be sensors who can detect security incidents and initiate the alert propagation process. This is typical model used to deploy large scale defense posture but it also comes with issues of trust – the security information's integrity and node authentication. For our experiment, only nodes with proper permission can publish new message to subscribers. Their integrity is examined by super-nodes, in the case of centralized and hybrid mechanisms, or peering nodes in charge of the ID segment, for distributed schemes.

5 Evaluation

In the section, we describe evaluation results for the alert distribution systems implemented with three different control mechanisms. First, we explain the evaluation metrics and then we talk about the reason behind the choice of the Oversim simulation framework. Lastly, we discuss our evaluation results with and without global adversaries. For each evaluation instance, all results are averaged over at least 10 iterations.

5.1 Evaluation Metrics

To evaluate the reliability and efficiency of the different control plane mechanisms, we introduce three metrics: *coverage*, *latency*, and *control efficiency*. *Coverage* is measured by enumerating the number of nodes that receive the alert message when the system is



(a) Latency measurement (b) Control cost measurement

Fig. 1. The evaluation results under normal operations. Figure (a) depicts the mean notification time and the completion time for 99% of the nodes and for different control plane mechanisms. Figure (b) presents the total network cost in terms of bytes for the test duration of 600 seconds.

under attack. Alternatively, all alert messages that are not delivered within the duration of the experiment instance are regarded as failures. *Latency* is defined to be the period of time that it takes for alerts to reach each participant from the moment that an alert message is dispatched. *Control efficiency* is the cost to utilize the control mechanisms. This is calculated by summing the total number of bytes required for network operations during the experiment.

5.2 Evaluation Design

To validate the network behavior under adversarial conditions, we implemented three control mechanisms using the OverSim [3] simulation framework. The use of simulation was mandated for the following reasons:

Scalability We were interested in observing the behavior of large scale networks implementing signaling systems in the presence of network-wide malicious attacks. Having tens of thousands of number of participants, Oversim framework enabled us to quantify the design parameters that really influence the behavior of the system.

Global adversary Emulating global adversary in a real-world large-scale testbed is a costly and time-consuming task and it does not allow repetition of experiments. The simulation framework not only helped us to instantiate this size of network but also provided the interface to implement a more precise behavior for the global adversary.

5.3 Evaluation under Normal Operation

To establish a baseline for our experimental results, we first measured the latency and control efficiency of the mechanisms without considering a global adversary. For test topologies of 20,000 nodes, each test instance was measured for 600 seconds of simulation time. Each test instance contained an alert notification event and the same size (40KB) of control messages were propagated to all participants. The size is derived from the average

size of Microsoft patches [7]. SN network, implemented for centralized mechanism, was configured with different *cluster sizes* and *k-redundancy* was fixed to 2 for all test cases. DHT network was used for the distributed control mechanism and its *successor list size* was set to five. Hybrid network inherited parameter from both systems.

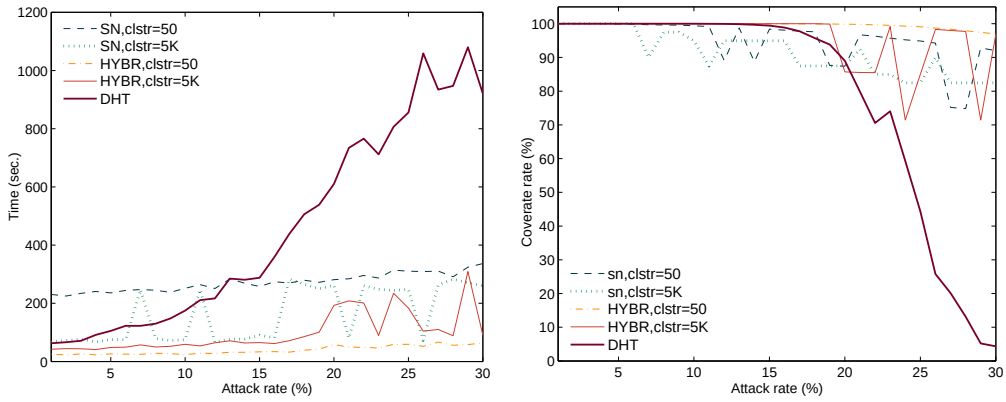
Latency measurement The latency results are shown in Figure 1(a). On the X-axis, from the left to right, we have results for the SN network, Hybrid network, and DHT network. The SN network and hybrid network are configured with different cluster sizes. For each bar, the dark portion represents the average time for notification and the gray part represents the time until 99% of the nodes are notified. Large variance was observed for the latency results of SN network. With different cluster sizes, mean latency ranged from 35 to 230 seconds. Populated sub-networks (lower-layer) accounted for delays in the case of large cluster size (5,000). For smaller cluster size (50), having more super-nodes made the upper-layer network the bottleneck. In contrast, for hybrid network, we observed small variance in latency and less delays. This is because the secondary, distributed channels masked the errors or failures of the primary channel. Mean latencies ranged only from 33 to 51 seconds. Not having a secondary channel, the DHT network took longer than the worst case of hybrid network. However, the latency remained relatively low (61 seconds).

Control cost measurement Figure 1(b) represents the control cost of different mechanisms to propagate alert messages of the same size (40KB). SN network, thanks to its simple implementation, required the least amount of packets to maintain its control channel and signaling operations. However, in the case of larger cluster size (5,000), many number of network errors and retries introduced rapid increase in cost. DHT network required larger amount of control traffic to maintain its distributed data structures. Hybrid network with large cluster size (5,000) required even more and was the most expensive control channel due to excessive numbers of network errors from its primary channel. However, with the proper choice of cluster size, hybrid network could spare its control cost to become a more efficient solution than the DHT network.

5.4 Evaluation of Adversarial Scenarios

In adversarial scenarios, we again used the topology of 20,000 nodes with longer simulation duration of 1,200 seconds to carefully observe the system's reaction to malicious activities. Nodes that could not be notified within this time duration were regarded as a delivery failure. Two different cluster sizes were plotted for SN network and hybrid network – 50 to represent a small cluster size and 5,000 for large cluster size. During the experiment, the alert propagation event was triggered at 100 seconds of the simulation time and the attack from the adversary was launched five seconds prior to the event.

Random attack evaluation In this attack scenario, the adversary randomly selects its victims varying its attack ranges (0% $\sim$ 30%). The latency and coverage results against this attack are shown in Figure 2(a) and Figure 2(b) respectively. For DHT network, both latency and coverage were most severely impacted by this attack. Having acceptable latency from its initial stage, DHT network's latency steadily increased. Network failures that impacted coverage started approximately around 17 $\sim$ 18%. The coverage results dropped rapidly from that point onwards. SN network showed better results than DHT



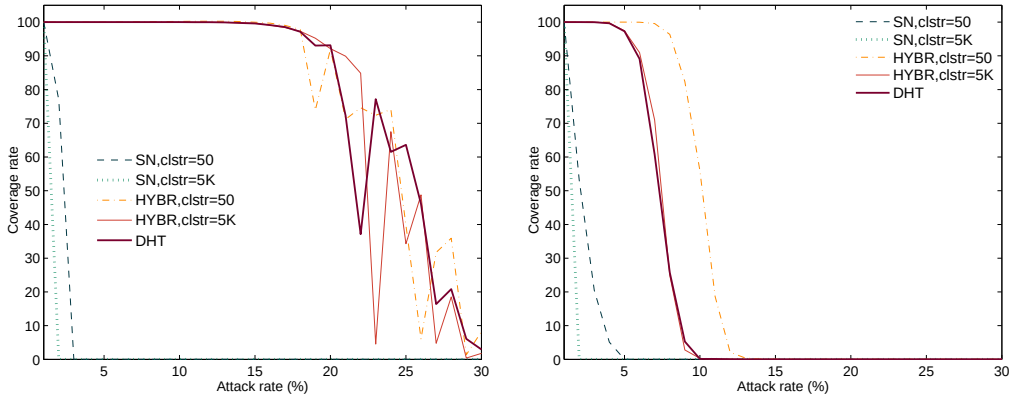
(a) Random attack latency

(b) Random attack coverage

Fig. 2. Figures (a) and (b) illustrate the latency and coverage for the random attack scenario respectively and for different attack intensities. In (a), Y-axis shows average notification time and in (b), Y-axis shows the percentage of nodes which successfully received the alert message.

network in terms of both metrics. It is interesting to note that the centralized mechanism with a little redundancy configuration (*k-redundancy*=2) showed better coverage results than the distributed system. In the DHT network, by distributing certain amount of connections to all participants, each node's failure had some influences on the system's connectivity. This resulted in network disintegration and gradual deterioration of latency beyond a certain threshold. This result is consistent with the observation that DHT network's performance is severely influenced by even a small fraction of slow performing nodes [20]. In the SN network, failures of all SN replicas for a sub-network significantly deteriorate system's latency and coverage. But, in the case of random attack, probability to hit all replicas in the same group is exponentially low in regards to *k-redundancy* parameter. Irregular spikes in its latency and coverage results indicates this type of failures where *k-redundancy* is two. Hybrid-network, by having dual channels, showed improved coverage and latency results. While the hybrid network showed smoother results than the SN network overall, systems with smaller cluster size had better latencies and reduced traffic irregularities.

Targeted attack evaluation In this attack, the adversary takes one step further by targeting nodes of *explicit* importance – super-nodes for the SN network and the hybrid network. After selecting all available target nodes, the attacker randomly select the rest of her victims. DHT does not expose any *explicit* targets. Thus, all the victims are selected randomly. The attack in this case becomes identical to the random attack. Coverage result against this attack are presented in Figure 3(a). Unlike the DHT network, whose results didn't changed much from the random attack result, the SN network is seriously impacted by this attack. Having all super-nodes eliminated, the system stopped being operational from the very initial stage of attack, less than or around 4%. Similarly, not having the benefits of its primary SN channel, latency and coverage results of the hybrid network soon converged to that of the DHT network.



(a) Targeted attack coverage (b) Degree dependent attack coverage

Fig. 3. Figure (a) and (b) present coverage results for the targeted attack and degree dependent attack respectively and for different attack intensities. Y-axis shows the percentage of nodes who successfully received the alert message.

Degree dependent attack evaluation In this attack, the attacker can identify nodes not only of *explicit*, but also of *implicit* importance. For this, she considers each node's topological significance. Super nodes maintain more state acting as defaults routes for the their clusters and thus are higher priority targets. In Figure 4(a), we depict the connection distribution for the hybrid network (of cluster size 50). We present the number of connections for the super-nodes and regular nodes using different colors. This Figure illustrates how the attacker chooses its victims for degree dependent attack with different attack rates of 1% and 4%. With respective dotted and dashed lines, the nodes with number of connections above the lines will be the victims.

The coverage result against the attack is presented in Figure 3(b). Similar to the targeted attack, SN network's coverage deteriorated from the initial stage of the attack. By choosing nodes with higher connectivity, this attack was highly effective in crippling the DHT network. DHT's coverage starts to drop around 7%~8%. In the case of hybrid network, the coverage was also impacted by the attack. The outcome for a large cluster size (5,000) with few super-nodes, does not show much difference from the DHT network's result. The small cluster size (50) performed better and extended coverage about 4%, because it was able to distribute the SN connections more evenly across the network curtailing the reachability failures due to the attacks.

Quantifying the behavior of the different signaling mechanisms when under different attack scenarios allowed us to make this observation: hybrid network is the efficient solution for both adversarial and normal situations with the following benefits. *i)* latency-wise, it was an efficient solution with less configuration sensitivity. *ii)* with the proper choice of cluster size, the system consumes reasonable amount of control cost which is higher than SN but less than DHT system. *iii)* Under all type of attacks, it showed the best resilience in terms of coverage and latency. Another interesting observation is that SN network, even

with less network connections, could show better results than DHT network against the random attack. However its reliability benefit is immediately cancelled by sophisticated and targeted attackers.

6 Analysis

Our evaluation shows that the hybrid network gained the number of reliability benefits by adding a constant number (two) of SN connections to the DHT network. This result indicates that the number of connections and the way it connects participants can seriously impact the reliability of the system. Unfortunately, the number of network connections is constrained by both system and network resources. Therefore, we want to explore the design space that can enhance reliability by only improving the way it connects participants. To that end, we investigate different ways of implementing control systems by using the same number of connections. More concretely, we extend DHTs with the fixed number (two) of connections in different ways to observe how these influence the coverage result. The number of connections is the same one used from the previous evaluations. Of course, this parameter can have significant impact on coverage results. However, for all proposals, we want to demonstrate how we can add connections under the same constraints and maximize the coverage benefits.

To further enhance the system's behavior when attacked, we leverage the benefits of DHT's internal structure with a modified routing table. This technique, which exploits finger-table, is implemented for Chord and is also applicable to other DHT systems.

6.1 Chord Connection Types

The Chord maintains two types of logical network connections. One for the *successor lists* and the other for the *finger table entries*.

- **Successor list** maintains the list of neighboring nodes. It is an important parameter that influences DHT's reliability and its default size is set to five. Having a $O(\log n)$ size of this connection provably guarantees the stability of the system which indicates success rate of lookup request.
- **Finger table** is a core data structure that implements $O(\log n)$ routing of Chord. The upper limit of its size is logarithmic to the size of hash space (in the case of Chord, this is set to 2^{160}).

Unlike previous proposals which naïvely added SN connection to DHT connections, we implement a hybrid network utilizing existing slots of finger tables. SN connections can be replaced with immediately preceding entries in the finger table. This does not increase the required state per node or the total number of connections, but this costs additional hops for lookup activities due to some sub-optimal entries.

Label	**Control mechanisms**
DHT	Chord with successor list size of 5 (default).
DHT S-list	Extend *DHT* by adding 2 connections to successor list. Successor list size is set to 7.
HYBR	Hybrid mechanism that naïvely integrates 2 additional SN connections to *DHT*. This is the hybrid network used from previous evaluations.
HYBR F-table	This extends *HYBR* by integrating 2 additional SN-connections with the finger table.

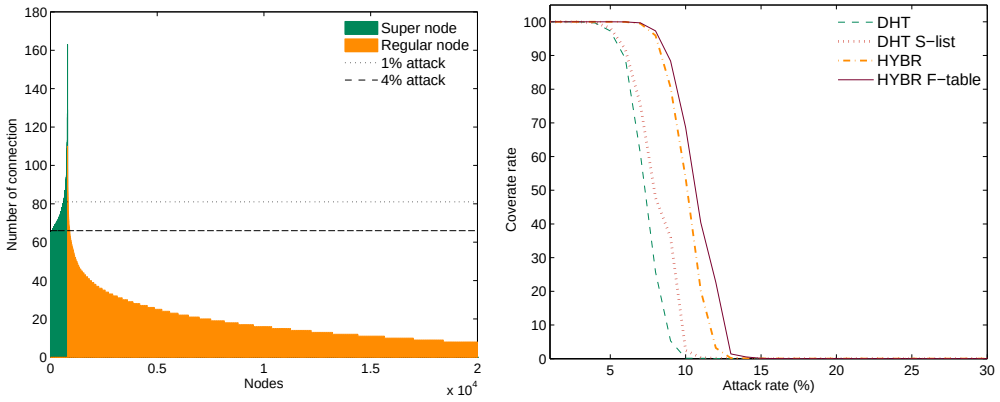
Table 1. Control mechanisms and their labels

6.2 Evaluation of Network Coverage

We measured the performance of our proposed modifications in terms of coverage. To that end, we present our experimental results from the degree dependent attack by varying its attack rate (0% ~ 30%) for a network of 20,000 participating nodes. The cluster size for hybrid network was set to 50 to make the effect of SN connections more pronounced. With larger cluster sizes, thus smaller SN connections, we expect coverage results similar to that of a DHT network. Table 1 details specific configurations and their labels used for evaluations. The evaluation results are presented in Figure 4(b).

The last three configurations (*DHT S-list*, *HYBR*, *HYBR F-table*) from Table 1 are implemented with the same number (two) of additional connections to the original Chord DHTs (*DHT*). The result for *DHT S-list* shows the limited effects of the two additional success list entries. From Figure 4(b), this improves coverage only by 2 ~ 3%. The result for *HYBR* shows better coverage (5 ~ 6%) than *DHT S-list*. Although SN connections replace connections assigned to the successor list, the structural benefits offered by the SN network are far greater. This is apparent in the *HYBR F-table*, by harnessing the reliability benefits of both successor list entries and SN connections, the coverage is boosted by 7 ~ 9%. Furthermore, penalty for having two sub-optimal entries in its finger table is negligible and requires only a small amount of additional lookup calls (3.4%).

The experimental results present interesting insight about the trade-offs between network structure and their impact on reliability. We can deduct that additional entries in the list have limited effect. Thus, it is better to consider other avenues of adding connections in order to enhance system's reliability. Modifying the finger table can be an option to consider because it increases coverage without deteriorating its original functionality.



(a) Distribution of incoming connection (b) Integrations under the degree dependent attack

Fig. 4. (a) enumerates connections for nodes in a hybrid network with cluster size of 50. The dotted and dashed lines show the impact of the degree dependent attack. The attacker choose victims with number of connection above the lines. (b) presents the coverage for different modifications and for the degree dependent attack. In X-axis we vary the attack intensity while Y-axis shows the alert success rate.

7 Conclusions

We evaluated alert distribution systems implemented using three control channel mechanisms under different adversarial scenarios. Our evaluation enabled us to draw a number of interesting insights regarding the reliability of the signaling channel. The pure distributed system (DHTs), designed to be robust under attacks, suffers in terms of network performance including latency and coverage. In the case of random attack, DHTs reliability turned out to be worse than that of a super-node based centralized design. To alleviate this, we proposed the integration of centralized and the distributed designs. Our approach consists of structural changes that enable us to seamlessly integrate a SN network and a DHT network. We evaluated a hybrid network design that offered the best coverage and reliability under all type of attack scenarios. We believe that with proper engineering choices, we can further enhance the system's reliability.

8 Acknowledgements

This work was supported by the NSF through Grant CNS-06-27473, 09-37060 to the Computing Research Association for the CIFellows Project,by ONR through MURI Contract N00014-07-1-0907, and by AFOSR through MURI Contract FA9550-07-1-0527. Any opinions, findings, conclusions or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the US Government, ONR, AFOSR, or the NSF.

References

1. J. Aspnes, N. Rustagi, and J. Saia. Worm versus alert: Who wins in a battle for control of a large-scale network? *Lecture Notes in Computer Science*, 2007.
2. B. Awerbuch and C. Scheideler. Towards a scalable and robust dht. *Theory of Computing Systems*, 2009.
3. I. Baumgart, B. Heep, and S. Krause. Oversim: A flexible overlay network simulation framework. *Proc. of IEEE GI*, 2007.
4. A. Bharambe, C. Herley, and V. Padmanabhan. Analyzing and improving a bittorrent network's performance mechanisms. *Proc. IEEE INFOCOM*, 2006.
5. M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham. Vigilante: end-to-end containment of internet worms. *Proc. of SOSP*, 2005.
6. F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica. Towards a common api for structured peer-to-peer overlays. *Proc. of IPTPS*, 2003.
7. C. Gkantsidis, T. Karagiannis, and M. VojnoviC. Planet scale software updates. *Proc. of SIGCOMM*, 2006.
8. L. Hui-shan, X. Ke, X. Ming-wei, and C. Yong. S-chord: Hybrid topology makes chord efficient. *Proc. of ICN*, 2005.
9. D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. *Proc. of STOC*, 1997.
10. S. Ktari, A. Hecker, and H. Labiod. Exploiting power-law node degree distribution in chord overlays. *Proc. of NGI*, 2009.
11. J. Li, J. Stribling, R. Morris, M. Kaashoek, and T. Gil. A performance vs. cost framework for evaluating dht design tradeoffs under churn. *Proc. IEEE INFOCOM*, 2005.
12. B. Loo, R. Huebsch, I. Stoica, and J. Hellerstein. The case for a hybrid p2p search infrastructure. *Proc. of IPTPS*, 2004.
13. P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. *Proc. IPTPS*, 2002.

14. D. Menasche, A. Rocha, B. Li, D. Towsley, and A. Venkataramani. Modeling content availability in peer-to-peer swarming systems. *SIGMETRICS Perform. Eval. Rev.*, 2009.
15. B. Mitra, F. Peruani, S. Ghose, and N. Ganguly. Analyzing the vulnerability of superpeer networks against attack. *Proc. of CCS*, 2007.
16. G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher. Availability in bittorrent systems. *Proc. IEEE INFOCOM*, 2007.
17. M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in bittorrent. *Proc. of NSDI*, 2007.
18. B. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 1987.
19. D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. *Proc. of SIGCOMM*, 2004.
20. S. Rhea, B. Chun, J. Kubiatowicz, and S. Shenker. Fixing the embarrassing slowness of opendht on planetlab. *Proc. of WORLDS*, 2005.
21. S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a dht. *Proc, of the USENIX Annual Technical Conference*, 2004.
22. A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001.
23. D. Serenyi and B. Witten. Rapidupdate: Peer-assisted distribution of security content. *Proc. IPTPS*, 2008.
24. S. Shakkottai and R. Srikant. Peer to peer networks for defense against internet worms. *Proc. of InterPerf*, 2006.
25. I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 2001.
26. M. VojnoviC and A. Ganesh. On the race of worms, alerts, and patches. *IEEE/ACM Transactions on Networking*, 2008.
27. B. Yang and H. Garcia-Molina. Designing a super-peer network. *Proc. of ICDE*, 2003.
28. M. Zaharia and S. Keshav. Gossip-based search selection in hybrid peer-to-peer networks. *Proc. of IPTPS*, 2006.
29. Y. Zhu, H. Wang, and Y. Hu. A super-peer based lookup in structured peer-to-peer systems. *Proc. of PDCS*, 2003.
30. C. Zou, W. Gong, and D. Towsley. Worm propagation modeling and analysis under dynamic quarantine defense. *Proc. of WORM*, 2003.