Removing the Blinders: Using Information to Mitigate Adversaries in Adaptive Overlays

David Zage, Charles Killian, and Cristina Nita-Rotaru
Department of Computer Science, Purdue University
{zagedj,ckillian,crisn}@cs.purdue.edu

***Abstract*—The proliferation of peer-to-peer systems has led to the increasing deployment of dynamic, adaptive overlay networks that are designed to preserve application performance goals. While such networks provide increased performance and resiliency to benign faults, they are susceptible to attacks conducted by compromised overlay nodes, especially those targeting the adaptation mechanisms. In this work, we propose a lightweight, general solution to increase the resiliency of adaptive overlay networks. By locally aggregating and correlating network topology with system performance metrics such as latency and bandwidth, each node can check the consistency of the reported information and constrain the attacker's ability to lie about system metrics. As a result, each node can make better adaptation decisions. We demonstrate the susceptibility of adaptation mechanisms to malicious attacks and the utility of our solution through real-life deployments of mature, adaptive overlay-based systems.**

I. Introduction

With the tremendous growth of the Internet, a wide range of applications taking advantage of peer-to-peer (P2P) systems have emerged in recent years. Several studies indicate that P2P systems account for the majority of all Internet traffic [1]–[3]. Many of these applications are built utilizing *adaptive* overlay networks to provide increased performance, increased storage, and fault tolerance to benign failures [4]–[7]. Through the incorporation of *adaptation mechanisms*, the overlay topologies can dynamically change to improve and maintain application-specific performance goals such as bandwidth or latency.

The proliferation of these adaptive P2P applications on public networks and the relocation of functionality to end-systems that are more likely to be compromised than core routers [8] raises questions about how to design and deploy P2P applications in a secure and robust manner [9]. In particular, attacks that exploit the adaptivity mechanisms can be extremely dangerous because they target the overlay construction and maintenance while requiring no additional communication bandwidth on the attacker side. Such attacks can allow an adversary to control a significant portion of the overlay traffic and expedite other attacks including, but not limited to, selective data forwarding and network partitioning. Not only are the attacks damaging from an end-user perspective, but they also have a large economic impact, costing businesses millions of dollars in lost revenue [10]. Reports of subversive attacks designed to disrupt the service provided by Internet businesses are beginning to appear in the media [11].

In this paper, we propose a lightweight solution to increase the resiliency of adaptive overlay networks by using multiple, disparate sources of information to improve the adaptation decisions. Unlike more static systems such as storage systems, adaptive overlays continuously monitor their environment and evolve. These facts allows heuristics, such as each individual node aggregating multiple sources of information (including potentially malicious information) into a unified view used to augment the adaptation process, to work very well. This allows each node to check the consistency of the reported information and constrain the attacker's ability to lie about system metrics.

While it would be ideal to both formally model and deploy our solution in real environments, due to the complexity of the P2P systems we consider, formal analysis often requires augmenting or weakening the specification and its assumptions, thus rendering it less applicable to real-world scenarios. Additionally, it has also been shown that even with formal specification, important implementation decisions are often left unspecified, which greatly affects the survivability of a system in real deployments [12]. Instead, we use real-world deployments which allow us to demonstrate the utility of our defense mechanism and quantify the performance an end-user can expect. We note that the goal of this work is not to create a "perfect" solution, but to improve system robustness in a manner that can be readily incorporated into a variety of overlay networks. Our contributions include:

- We demonstrate the benefit for an attacker to subvert the adaptation mechanisms over other methods of attack by showing the susceptibility of the adaptation mechanisms to attack through real-life deployments on the PlanetLab testbed [13] using End System Multicast (ESM) [4]. Malicious insiders are able to exploit the adaptation mechanisms which fail to take into account the effects of attackers on their environment.
- We provide a solution to increase the resiliency of adaptive systems to attack by aggregating and correlating data-plane and control-plane information into a *path graph* to determine the reliability of received information. The path graph information is incorporated into the adaptation process of the overlay network, constraining the ability of the attacker to lie to honest nodes and increasing the robustness and stability of the network. While network topology has previously been used to detect data-dropping attacks [14], we use it to detect and mitigate a more general set of attacks targeting adaptation.
- We demonstrate the effectiveness of our solution through real-world experiments on the PlanetLab testbed using the tree-based ESM system. We show that even when 30% of the network consists of malicious nodes, we are able to maintain the system bandwidth near that of the optimal level.

- We demonstrate that our solution is a general mechanism that can be readily applied to a variety of protocols without the creation of a formal specification and minimal change to the protocol by using it to add robustness to BulletPrime [15], which uses a mesh-based overlay.

The rest of the paper is organized as follows: We provide an overview of the system and attacker models in Section II and attacks against them in Section III. We propose a defense technique in Section IV, present experimental results demonstrating the effectiveness of our solution in Section V, discuss related work in Section VI, and conclude in Section VII.

II. System and Attacker Model

In this section, we provide an overview of the main components of unstructured overlay networks and our exemplar application, video multicast. We also describe our threat model.

A. System Model

We consider an overlay network providing support for single-source broadcasting applications that are high-bandwidth (hundreds of kilobits to megabits per second) and real-time. The system consists of a set of nodes and a data source communicating via unicast links. All nodes receive data and contribute to the routing process by forwarding data.

The nodes maintain a logical network consisting of the connectivity between peers, which we refer to as the *control plane* of the system. Every node N that joins the network maintains two sets of nodes, a *peer set* and a *downstream set*, and an upstream node also referred to as N's *parent*. The peer set is a subset of nodes that are currently reachable in the overlay from which performance information is gathered. This set is bootstrapped at join time by contacting the source and is continually updated via information received from the peer set. The downstream set (*i.e.*, children) is the collection of nodes that N is responsible for delivering data to. This set's size is limited by the system parameter termed the *saturation degree*, representing the number of concurrent data streams N is able to support before saturating the bandwidth of the underlying physical link.

In addition to the connectivity between different peers, nodes also maintain information that allows them to decide how data is disseminated. We refer to the dissemination structure of the system as the *data plane*. After a node joins the network, it periodically probes members of its peer set to collect metrics about the performance and structure of the network. Additionally, this probing is used to learn about new potential peers, allowing the nodes to update their peer sets with new members when nodes leave or become unresponsive.

Once the nodes have joined the network and established their respective peer sets, each node will select a parent node from which to receive data, creating a tree-based dissemination structure. As nodes receive new information from their peers, they are able to incorporate the collected performance metrics such as bandwidth (throughput), latency (one-way delay), and round-trip-time (RTT) into adaptive decision functions which allow the node to select a new parent from its peer set if its performance becomes inadequate.

B. Attacker Model

We consider a Byzantine adversary model similar to that in Castro *et al.* [16], with a system size of N and a bounded percentage of malicious nodes f $(0 \leq f < 1)$. The malicious nodes behave arbitrarily and are only limited by the constraints of any cryptographic methods deployed [17]. The set of malicious nodes may collude. We assume a malicious adversary has access to all data at a node as any legitimate user would (insider access), including cryptographic keys stored at a node. Nodes cannot be completely trusted although they are authenticated. We assume that data authentication and integrity mechanisms are deployed and we focus only on attacks targeting adaptivity. Additionally, we assume that the adversary knows what defense mechanisms are deployed in the system and may attempt to subvert these.

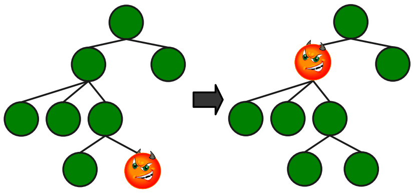


Fig. 1. An example attraction attack in which the malicious node has lied about its performance metrics and caused many of the benign nodes to erroneously select it as a parent.

III. Attacks Against Adaptive Systems

Any adaptive network protocol based on measurements involves periodically observing and estimating the network conditions, followed by making an adaptation decision. For an unstructured multicast overlay, this decision consist of selecting a new parent by weighing the associated costs versus benefits of switching parents.

While many systems employ mechanisms such as data sampling and smoothing to ameliorate the decision process, these techniques are designed to tolerate benign errors. In an adversarial network, compromised nodes can take advantage of the adaptation process by lying about their performance metrics. This allows an attacker to manipulate the path selection of the overlay topology and gain control over the overlay traffic. Previous research has identified three classifications of attacks that target adaptivity: attraction attacks, repulsion attacks, and disruption attacks [18]. In this work, we focus on the attraction attacks. By selecting a representative, damaging attack, we determine a baseline for our technique by which its effectiveness on other types of attacks can be concluded.

The attraction attacks occur when malicious nodes report false performance metrics, creating the illusion of better performance than in reality. The attacker uses these falsified reports to induce a benign node into selecting the malicious node as a parent in the dissemination structure. As can be seen in Figure 1, the malicious node has lied about its performance metrics (*e.g.*, reporting artificially low latency to the source), inducing many benign nodes to connect through it. The node now controls the majority of the traffic in the overlay. The final goal of the attack can be manipulating data, traffic analysis, man-in-the-middle attacks, or selectively dropping packets.

TABLE I
PROBE RESPONSE COMPONENTS

Metric	Usage
Bandwidth received from parent	Data-Plane
Latency from the parent	Data-Plane
Latency from the source	Data-Plane
Time connected to parent (stay time)	Data-Plane
Current parent in the overlay structure	Control-Plane
Number of children	Control-Plane
Path to the source of the data	Control-Plane

Algorithm 1: Path graph procedure to exclude malicious nodes from the adaptation process

```
Input: Potential Adaptation Candidates List ($PACL$) and the Path
       Graph $PG$
Output: Updated $PACL$
foreach $rnode$ in $PACL$ do
    // Inconsistent number of children
    if ($rnode$.numChildren ¡ $PG.rnode$.numChildren) then
        remove $rnode$ from $PACL$;
    // Too many children
    else if ($PG.rnode$.numChildren $>$ SystemSaturationDegree) then
        remove $rnode$ from $PACL$;
    // Inconsistent bandwidth reported
    else if ($rnode$.BW - $PG.rnode$.children.ActualBW $>$
    (SourceRate*.1)) then
        remove $rnode$ from $PACL$;
    // Inconsistent path to the source
    else if ($rnode$.path != $PG.rnode$.path) then
        remove $rnode$ from $PACL$;
    // Too many small stay times
    else if ($PG.rnode$.children.stayTimes $<$ 100 sec) then
        remove $rnode$ from $PACL$;
    else
        keep $rnode$ in $PACL$;
    end
end
```

IV. Leveraging Control and Data-Plane Information to Mitigate Attacks

The attacker's ability to subvert the overlay networks ensues from the fact that the attacker can influence the adaptation process by manipulating the performance metrics. This stems from the assumption that nodes are altruistic and respond with correct metrics to queries. We propose to increase the resiliency of the system by aggregating and correlating both data-plane and control-plane information, allowing each node to use the derived information to make better adaption decisions. The control-plane information is the metrics necessary to manage connectivity while the data-plane metrics are derived from overlay data. One important facet of the solution is that it uses information *already* present in the system, avoiding extra link stress. It should also be noted that many adaptive protocols utilize similar metrics for adaptation [4]–[7], [15], [19], [22], [23], allowing our solution to be readily applied.

During the system lifetime, each node periodically probes its peer set to retrieve their performance metrics and subsequently receives responses for a fixed interval of time. The contents of the probe responses along with where the data is utilized are listed in Table I. Each interval, nodes use the gathered information, combined with their own locally measured performance, to determine if they should change parents. If a malicious insider responds with falsified information, it can bias the adaptation decision and cause incorrect choices.

As part of the solution, a node receives probe responses and aggregates this information into a new, in-memory graph structure called the *path graph*. The intuition behind this is that by combining information from multiple sources, we make it possible to detect inconsistencies across related peers, allowing us to identify potential foul play. Even though unstructured overlays have no tight topological invariants such as those present in structured overlays, the system forms a stable topology which can be examined for such inconsistencies. By correlating data into a path graph at each node, it becomes possible to check the consistency of the reported information and constrain the attacker's ability to lie about system metrics.

An example path graph is presented in Figure 2. The root of the path graph is the data source and all other vertices contain information pertaining to a specific node. Each of the edges represent a link in the overlay topology. Since only a subset of the entire overlay is probed by each node, nodes referred to, but for which information is not directly obtained, are represented by placeholders. For example, in Figure 2, the ellipses depict nodes whose existence was discovered by examining the path to the source used by other probed nodes.

Given the path graph, benign nodes can prevent many unnecessary adaptations by avoiding adaptation decisions induced by falsified data using Algorithm 1. During a node's adaptation process, when it is considering a new potential parent p, it will check the sanity of p's contextual information contained in the path graph. The possible parent p will be avoided if any inconsistencies are found using Algorithm 1. For example, if the highlighted node in Figure 2 reports having one child while the path graph contains two, the node is potentially malicious and should not be considered as a viable potential parent. Our solution is conservative in nature, resetting the path graph after each adaptation attempt to maintain data freshness and never removing potential parents on the basis of incomplete information (*e.g.*, the path graph containing fewer children than the parent node reports). Even if a node accidentally avoids a good potential parent, nodes have multiple potential parents to choose from and no parent is permanently avoided, allowing the system to tolerate occasional false positives caused by events such as topology changes without detrimental effect.

Discussion. As stated previously, the goal of the solution is to add robustness to overlay networks by constraining the ability of the attacker through a lightweight, generally applicable technique. Our technique introduces minimal link stress since it focuses on using information already being exchanged between nodes and minimal storage requirements since it only stores the path graph at each node during the adaption process. This is especially valuable in instances where extra computational resources or centralized points of trust are unavailable. Also, it should be noted that while the path graph forms a tree structure based on the data-plane information provided, the construction of the graph is agnostic of the type or function of overlay network, allowing it to be applied to P2P systems with other connectivity models.

Through the use of data-plane and control-plane infor-

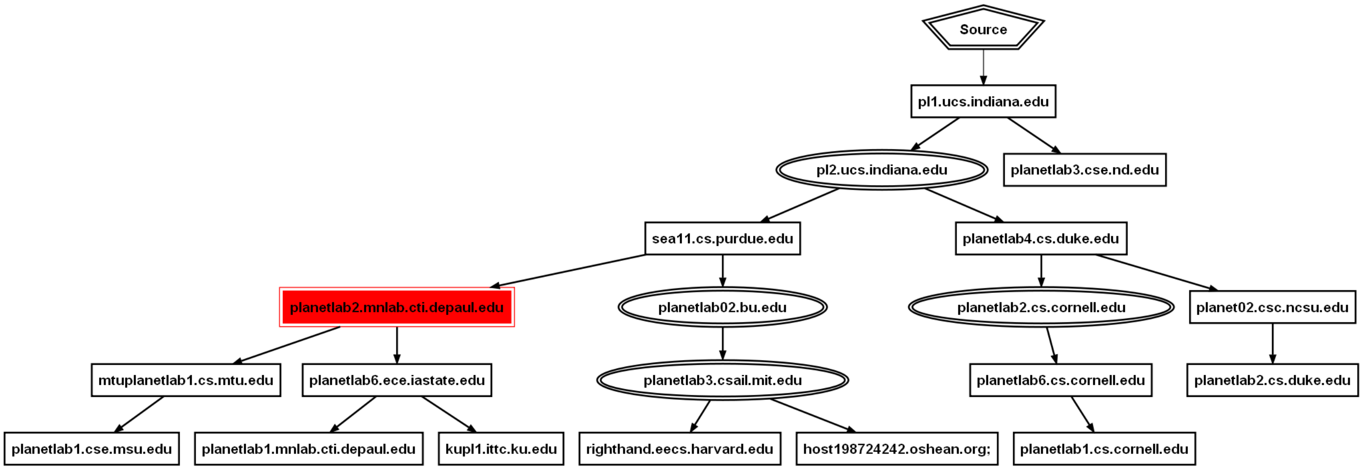


Fig. 2. The aggregation of data-plane and control-plane information into a graph structure used to improve adaptation decision quality. The square nodes represent peers from which data has been received while the ellipses are placeholders used to complete the graph structure. The highlighted node represents a malicious node reporting it has only one child.

mation, the room for malicious activity has been reduced, but malicious nodes are still occasionally chosen as parents throughout the system lifetime. Each benign node only takes into account *its own* path graph, not sharing potentially helpful information learned through the data aggregation process. Thus, benign nodes may have to experience malicious activity they could have avoided with collaboration. However, using only local information prevents abuse of our technique by collaborating malicious nodes. To have a comprehensive solution and allow the system under attack to return to operational system performance experienced in a benign environment, a complimentary removal mechanism must be incorporated as part of a comprehensive solution to malicious activity. Our technique should be considered a useful tool for system designers to enhance the robustness of the system without adding extra complexity. For example, it could be applied in conjunction with defense techniques such as those discussed in [24] to improve the effectiveness of both schemes.

V. Experimental Results

We demonstrate through experimental results the susceptibility of adaptation mechanisms to malicious attacks and the utility of the path graph defense at mitigating the effects of the attack in the context of the ESM overlay multicast system [4] . We selected ESM because of its maturity, extensive deployment, and the advanced set of adaptation techniques it employs. Our experiments show that, although ESM employs an advanced set of adaptation mechanisms, it is unable to mitigate the attacks posed by a malicious adversary. Our defense technique is able to reduce the impact of malicious nodes without adding to the link stress in the system.

A. Overview of ESM and Narada

ESM is a multicast system used for broadcasting live events such as academic conferences (*e.g.*, INFOCOM). We provide a high-level description below and direct the reader to [4] for more details. ESM uses an application level multicast protocol, Narada, that builds an overlay tree for distributing content.

A key component of Narada is the use of adaptivity mechanisms to dynamically change the multicast tree to improve application performance or maintain it when network conditions change. More specifically, this adaptivity serves to improve suboptimal overlay meshes. Narada employs both passive observation and probing of peers to collect the data used to make the adaptation decision. Once the data has been received at a node, an extensive set of mechanisms is employed to improve the quality of the data and subsequent adaptation decision, including but not limited to, data sampling, data smoothing, decision randomization, and hysteresis.

B. Experimental methodology

To study the effects of the attacks and the defense technique under real-world conditions, we conducted experiments on the widely-used PlanetLab [13], [25] Internet testbed. PlanetLab provides a research platform for large-scale distributed experimentation of P2P systems over the Internet. To mitigate the possible limitations of using a testbed, such as those discussed by Spring et al. [26], multiple experiments were conducted at different times and on different days of the week. Further, experimental nodes were selected randomly for different experiments to validate the statistical significance of the results and nodes were chosen to span multiple operational and administrative domains. Each experiment was conducted multiple times and the results were averaged.

The baseline configuration for the experiments consists of a 30 minute deployment of 100 nodes in which the nodes join after the experiment begins and leave before it ends, with an average participation time of 26 minutes. As in previous ESM deployments [27], nodes are probed every seven seconds, each node probes 30 peers, the saturation degree of benign nodes is four, and the source streaming rate is 480Kbps. All experiments use these parameters unless otherwise noted. Experiments with larger systems (> 100 nodes), higher streaming rates (> 480Kbps), longer run times (> 30 minutes), and higher saturation degrees (> 4 children) were conducted and resulted in performance similar to those presented here.

As mentioned in Section V-A, ESM employs a set of methods to improve the quality of adaptation decision. For the experiments that incorporate the defense technique, we

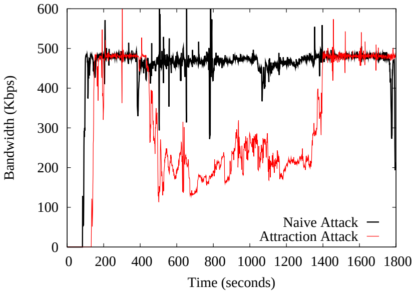


Fig. 3. Average system bandwidth for a 100 node ESM overlay deployed on PlanetLab with 30% malicious nodes. The graph depicts a naïve data-dropping attack and the more powerful attraction attack.

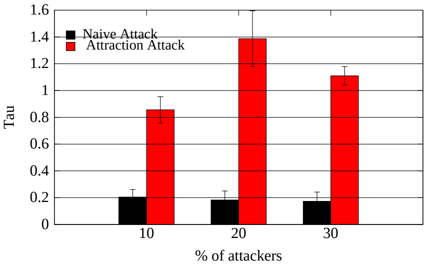


Fig. 4. Attack strength for different percentages of malicious nodes for ESM deployments of 100 nodes on PlanetLab using the naïve and attraction attacks.

integrate them into the ESM decision process *prior* to the preexisting data cleansing techniques. As the two sets of techniques are orthogonal, they do not conflict with each other. Additionally, using our technique first allows for the removal of much of the malicious data during an adaptation decision, allowing the existing data cleansing techniques to take place in a environment similar to that of a benign system.

C. The System Under Attack

We study the effect different percentages of malicious nodes have on the overlay topology if they lie about their performance metrics to gain beneficial positions in the dissemination structure. While ESM has several mechanisms designed to tolerate *benign* errors, it has no built-in mechanisms to defend against malicious nodes and is thus prone to instability and poor performance when under attack.

To determine the efficacy of subverting the adaptivity of the system, we examine the following scenarios:

• **Naïve Attack**: This is an attack method in which the malicious nodes simply obtain random positions in the overlay and, after an initial starting period, drop 90% of the data. Ninety percent was selected since it bypasses ESMs mechanisms for tolerating benign errors.

• **Attraction Attack**: This is the attraction attack discussed in Section III. In this attack, the malicious nodes report having the best bandwidth (480Kbps), smallest latency (0ms), and no saturation. After an initial starting period which the malicious nodes use to optimize their positions in the overlay network, they drop 90% of the data.

In both attack scenarios, other secondary actions besides dropping data could be performed. However, dropping data visibly demonstrates the severity of the attacks. The nodes only drop data for a ten minute interval to demonstrate the performance degradation is caused solely by the attack since the system bandwidth returns to the source rate after the attack.

Figure 3 shows the two attacks scenarios in which the naïve attack has little effect on the bandwidth of the system. Even when 30% of the network is comprised of malicious nodes, the average bandwidth only drops by 20Kbps to 460Kbps. This is not the case for the attraction attack, in which the average bandwidth of the system markedly decreases as more malicious nodes are introduced The difference between the attacks is due to the fact that without supplementary mechanisms such as attacking the adaptation mechanisms, most of the malicious nodes are unable to obtain advantageous locations in the overlay and are rendered ineffective. This insight elucidates the need to make these mechanisms more robust.

Attack Strength with Different Percentages of Malicious Nodes. To quantitatively compare experiments with different percentages of malicious nodes and possible defense techniques, we utilize an augmented form of the relative strength of attack measure τ [24].

The relative strength of a particular attack as is defined as:

$$\tau = \frac{B_{norm} - B_{adv}}{B_{norm} \times N_{adv}} \times 100 \qquad (1)$$

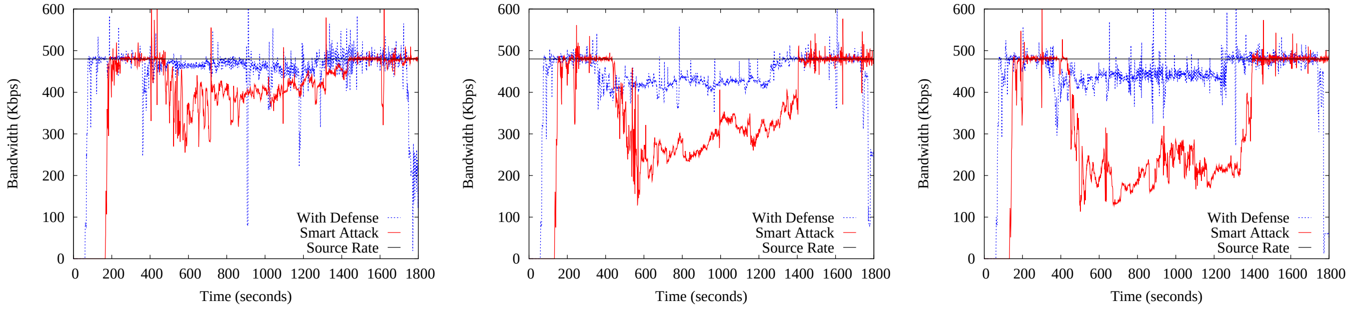
where B_{norm} and B_{adv} represent the average throughput in the absence and presence of adversaries respectively, and N_{adv} is the number of adversaries. Intuitively, τ represents the amount of damage an attack created in the system normalized by the number of adversaries. The greater the performance degradation observed in the system (the difference between B_{norm} and B_{adv}), the higher the value of τ. Figure 4 depicts τ varying over the percentage of attackers for both attacks. As expected, the naïve attacks resulted in very low τ values as they were largely ineffective. However, we can see the attraction attack has significant impact on the performance of the system for even a small percentage of malicious nodes (10%) and thus has a high τ value. Increasing the percentage of malicious nodes yields higher τ values, with the maximum effectiveness for the attacker occurring when 20% of the network was malicious. With percentages greater than 20%, the average system bandwidth continues to decrease, but relative measures like τ experience diminishing returns as each individual malicious node is less effective.

D. Mitigating Attacks Using the Path Graph

To demonstrate the effectiveness of using the path graph to improve the adaptation decision quality and mitigate the effects of malicious activity, we use the following scenarios:

• **Smart Attack**: We use this name to denote a scenario in which a percentage of the nodes is malicious and performs a smarter version of the attraction attacks described in Section V-C. Specifically, not only do the nodes report incorrect performance metrics, but they also attack the defense scheme itself by lying about the data collected in our path aggregation scheme (*e.g.*, reporting fewer children than in actuality).

• **With Defense**: We use this name to denote a scenario in which an attacker performs the Smart Attack while our path graph defense technique presented in Algorithm 1 is enabled.



(a) 10% Malicious (b) 20% Malicious (c) 30% Malicious

Fig. 5. Average system bandwidth for 100 node ESM overlay deployed on PlanetLab under different percentages of attackers. The graphs show the degradation of the system performance and the utility of using our defense technique to mitigate the effects of the attack.

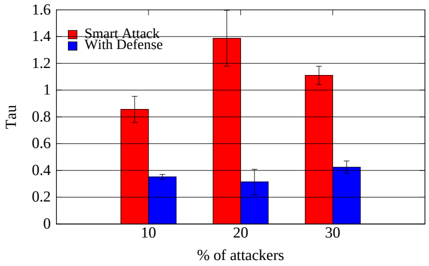


Fig. 6. Attack strength for different percentages of malicious nodes for ESM deployments of 100 nodes on PlanetLab using our defense technique.

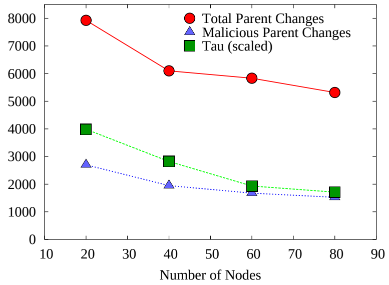


Fig. 7. The effect of using different peer set sizes on our defense mechanism for ESM deployments of 400 nodes on PlanetLab when 20% of the nodes are malicious and performing an attraction attack. The attack strength τ has be scaled to be visible on the same graph as the number of parent changes.

Figure 5 shows both the severity of the attacks and the ability of our solution to mitigate their effect. As can be seen, the bandwidth of the original system is greatly reduced when the malicious nodes are dropping data. However, using the path graph to improve adaptation decisions, the system maintains performance (average bandwidth) close to the optimal source rate of 480Kbps. For example, in Figure 5(c), the average system bandwidth using the path graph is approximately 430Kbps, or 200Kbps higher than the undefended deployment. While there is some degradation in the system performance, this is in accordance with our goal of creating a lightweight, broadly-applicable solution discussed in Section IV.

Figure 6 presents the effect of the path graph defense on the relative strength of the attacks, τ. It confirms the intuition that constraining the ability of the attacker to subvert the adaptation of the system diminishes the strength of the attack for all percentages of malicious nodes. For the smaller percentages of malicious nodes, τ is reduced to levels near those of the naïve attacks. This demonstrates our solution greatly increases the robustness of the adaption mechanism and the system.

False Positives. As each node performs its probe cycle independently and the overlay topology continuously adapts, inconsistencies can occur in nodes' path graphs which will cause them to avoid otherwise valid parents. As the majority of the overlay is stable, nodes have multiple potential parents, and no node is permanently banned from overlay, the system is able to tolerate these false positives without detrimental effect.

E. Effect of the Peer Set Size on the Defense Technique

We examine the effect of changing the number of peers each node probes to gather information from as this can impact the utility of the path graph. If too few peers are probed, the path graph will be too incomplete and ineffective at augmenting the adaptation decision. On the other hand, if too many peers are probed, computational resources are used for little gain.

Using the Chuang and Sirbu scaling law [28], we can analyze the average amount of topological structure information an individual node aggregates into the path graph when probing different numbers of peers. The average number of links $L(N)$ used to reach n random receivers in a k-ary multicast tree of size M is approximated as follows:

$$L(N) \approx \frac{\ln(1-\frac{n}{M})}{\ln(1-\frac{1}{M})} \times \left(c - \frac{\ln\left(\frac{\ln(1-\frac{n}{M})}{M\ln(1-\frac{1}{M})}\right)}{\ln k}\right) \qquad (2)$$

where c is an additive constant set to $c = 1$. As we can see from the analytical calculations and experimental results presented in Table II, the scaling law is able to accurately estimate the size of the path graph. As the number of probe responses increases, each node will receive greater amounts of information about the graph structure. However, as shown in both Table II and Figure 7, as more nodes are probed, the amount of unique information gathered and the improvement of our solution slows as each node begins to receive large amounts of duplicate data. Not only does further increasing the probe set size incur extra overhead for minimal gain, this extra information is often superfluous as each node is only concerned with the areas around the nodes it is evaluating as a potential parent. These facts lead us to conclude that a peer set of 30 nodes is sufficient for most deployments, with the exception being those anticipated to have large numbers of nodes (≥ 400). For such systems, the peer set size should be

TABLE II
APPROXIMATE NUMBER OF NODES IN A RANDOMLY CHOSEN PATH GRAPH FOR A 400 NODE DEPLOYMENT

Number of Probes	*Path Graph Size*	
(% of Systems Size)	Estimated	Actual
20 (5%)	64	78
40 (10%)	110	120
60 (15%)	150	155
80 (20%0	186	179

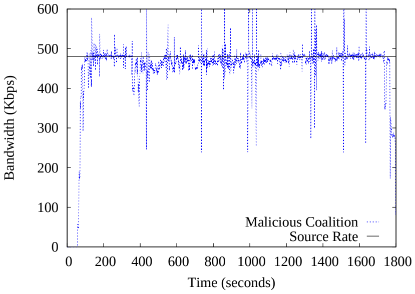


Fig. 8. Resilience of our technique to malicious coalitions attempting to bypass the defense technique while conducting an attraction attack on a representative ESM deployment of 100 nodes containing 20% malicious.

dynamically based on the scaling law to receive information from at least 35% of the network.

F. Malicious Coalitions

All defense mechanisms and protocols resilient to insiders have limitations regarding the number of attackers they can tolerate. We now consider the constrained collusion model presented in Section II in which the faulty nodes are part of the same coalition. We use the following scenario:

- **Malicious Coalition**: We use this name to denote the scenario in which a coalition of colluding attackers attempts to bypass the defense itself by strategically lying about system metrics to have members of the malicious coalition selected as parents high in the dissemination structure.

To subvert our defenses, the attacker must control multiple nodes in the same section of the overlay, allowing for coordinated lying between parents and children. This allows the parent node to attract benign nodes with less chance of inconsistencies appearing in the benign node's path graph. However, the greater the number of malicious nodes that must be connected to each other, the fewer the number of benign children that can be connected to a malicious node without creating inconsistencies in the path graph (*e.g.*, more children than the saturation degree), thereby lessening the effectiveness of the malicious nodes. Also, the malicious children are constrained to accurately reporting their path or they risk identifying themselves or their parent as being malicious. From Figure 8, we can see that our solution is able to mitigate the attack, even when malicious coalitions are actively trying to bypass the path graph detection. In fact, the relative attack strength τ is reduced by one-third of that seen in Figure 6 to 0.19. This implies that for maximum effectiveness in bypassing our solution, the malicious nodes should work alone or in small groups. However, from Section V-D, our solution mitigates much of the effect of this strategy.

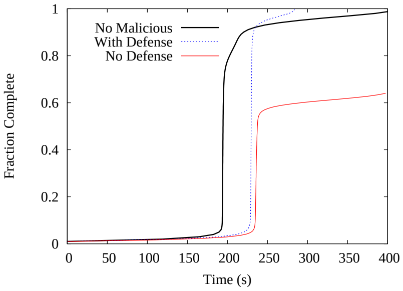


Fig. 9. Degradation of the system performance and the utility of using our defense technique to mitigate the effects of the attack in BulletPrime deployment of 100 nodes on ModelNet containing 20% malicious nodes.

G. Defending Other Systems

To show the general applicability of our defense technique as discussed in Section IV, we apply it to BulletPrime [15]. BulletPrime is a mesh-based P2P file sharing application similar to BitTorrent. While BitTorrent assumes nodes are greedy and hence uses a tit-for-tat incentive mechanism, BulletPrime assumes nodes are altruistic and work together to achieve the best performance. BulletPrime incorporates mechanisms based on bandwidth, data received, and the number of outstanding blocks to adapt the number of peers maintained, which peers to contact, and the number of blocks to request from each peer. Thus, while BulletPrime exceeds BitTorrent performance by optimizing the global system bandwidth through adaption mechanisms, BulletPrime is susceptible to attacks against its adaptation mechanisms. Using our defense technique to correlate the bandwidth and number of blocks from multiple neighbors, we can make minor changes to the adaption mechanisms and information exchanged to add robustness to malicious nodes while incurring minimal overhead.

We present the CDF for the file transfer completion for 100 node deployments of BulletPrime on ModelNet [30], with each node downloading a 50Mb file. ModelNet is a network emulator which accurately emulates specified loss, latency, and bandwidth limitations for unmodified applications running on physical hardware. We observe from Figure 9 that when malicious nodes report having all of the pieces and good bandwidth, they are able to prevent many of the benign nodes from receiving the last block(s). This prevents nearly half of the nodes from completing the transfer. By using our technique, we allow all of the nodes to complete the file transfer quickly rather than dragging on indefinitely by making better adaption decisions and selecting altruistic peers.

VI. RELATED WORK

In this section, we provide an overview of previous research in two main areas related to our work: attacks exploiting adaptivity and overlay defense techniques.

- *Attacks Exploiting Adaptivity.* Previous work demonstrated attacks with severe effects on TCP throughput by manipulating the TCP adaptation mechanism's perception of network congestion [31]. The attack was generalized as a form of low-rate ROQ attack by Guirguis *et al.* [32]. Our work assumes a stronger adversarial model in an overlay network. Other research has demonstrated the vulnerability of the adaptation mechanism in distributed virtual coordinate systems [**?**], where malicious nodes influence the calculation of the virtual coordinates by reporting false metrics. By using outlier detection to avoid using falsified data during coordinate updates (a form of adaptation), much of the effect of the malicious nodes is mitigated. Our work presents a complimentary approach to mitigating malicious activity based on the data present without the need to determine outlier detection thresholds.

• *Attacks and Defenses in Overlay Networks.* The problem of malicious attackers was previously studied in the context of structured overlay networks, where solutions often enforces constraint invariants [**?**]. As unstructured overlay networks do not have such constraints, the proposed solutions are not directly applicable. Our work considers malicious attackers and presents results in the context of a real system in real deployments over the Internet. Malicious insiders have also been studied before in the context of multicast networks by Walters *et al.* [18] and Xie and Zhu [14]. Walters *et al.* [18] provide a solution for mitigating the attacks based on outlier detection coupled with a reputation system. Xie and Zhu [14] use a random sampling based scheme to statistically determine if any nodes are d ropping messages. Both of these works assume a source of trust in the network which we do not. Secondly, our technique does not rely on detecting the effects of the attack, but instead limits the attacker's ability to influence the adaptation process, mitigating much of the effect of the attack before it begins.

VII. Conclusions

In this paper, we focus on insider attacks against adaptation mechanisms in adaptive, unstructured overlays. We demonstrated the utility for an attacker in subverting adaptation mechanisms over other methods of attack. We proposed a technique to mitigate the impact of the attacks by aggregating and correlating network topology and application metrics at each node, allowing the nodes to use the derived information to make better adaptation decisions. Our solution is lightweight, scalable, and improves the adaptation process and the overall stability of the system while limiting the effect of malicious nodes. Through experiments run on the PlanetLab Internet testbed using the ESM, we demonstrate our technique is effective in mitigating the attacks and raises the bar for the attacker without adding additional link stress or overhead in the system. Even for overlays containing up to 30% malicious nodes, our technique is able to maintain system performance near the optimal level and double that of an undefended system. We show the general applicability of our technique by using it to add robustness to BulletPrime. Based on these results, our technique should be considered a useful tool in the toolbox of system designers for enhancing the robustness of an adaptive, unstructured overlay system.

References

[1] K. Cho, K. Fukuda, H. Esaki, and A. Kato, "Observing slow crustal movement in residential user traffic," in *CoNEXT*, 2008.

[2] W. John, S. Tafvelin, and T. Olovsson, "Trends and differences in connection-behavior within classes of internet backbone traffic," in *PAM*, 2008.

[3] H. Schulze and K. Mochalski, "Internet study 2008/2009," Ipoque, Tech. Rep., 2009.

[4] Y. hua Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Keynote Address of SIGMETRICS*, 2000.

[5] F. Dabek, J. Li, E. Sit, J. Robertson, M. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," in *NSDI*, 2004.

[6] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming," in *INFOCOM*, 2005.

[7] A. Nandi, A. Ganjam, P. Druschel, T. Ng, I. Stoica, H. Zhang, and B. Bhattacharjee, "SAAR: A shared control plane for overlay multicast," in *NSDI*, 2007.

[8] "2010 CyberSecurity watch survey." [Online]. Available: http://www.cert.org/archive/pdf/ecrimesummary10.pdf

[9] M. Sher and T. Magedanz, "A vulnerabilities analysis and corresponding middleware security extensions for securing NGN applications," *Computer Networks*, vol. 51, pp. 4697–4709, 2007.

[10] Codenomicon, "IPTV security and reliability challenge," Codenomicon, Tech. Rep., 2008.

[11] J. Louderback, "Inside the attack that crippled Revision3," May 2008, http://revision3.com/blog/2008/05/29/.

[12] C. Killian, J. Anderson, R. Jhala, and A. Vahdat, "Life, death, and the critical transition: Finding liveness bugs in systems code," in *Proc. of NSDI*, 2007.

[13] "Planetlab." [Online]. Available: http://www.planet-lab.org/

[14] L. Xie and S. Zhu, "Message dropping attacks in overlay networks: Attack detection and attacker identification," *ACM Trans. Inform. Syst. Se.*, vol. 11, no. 3, pp. 1–30, 2008.

[15] D. Kostić, A. C. Snoeren, A. Vahdat, R. Braud, C. Killian, J. W. Anderson, J. Albrecht, A. Rodriguez, and E. Vandekieft, "High-bandwidth data dissemination for large-scale distributed systems," *ACM Trans. Comput. Syst.*, vol. 26, pp. 1–61, 2008.

[16] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," in *OSDI*, 2002.

[17] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, pp. 198–208, 1983.

[18] A. Walters, D. Zage, and C. Nita-Rotaru, "A framework for securing measurement-based adaptation mechanisms in unstructured multicast overlay networks," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 1434 – 1446, 2008.

[19] V. Venkataraman, K. Yoshida, and P. Francis, "Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast," in *ICNP*, 2006.

[20] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, "Chainsaw: Eliminating trees from overlay multicast," in *IPTPS*, 2005.

[21] "Azureus BitTorrent client." [Online]. Available: http://azureus.sourceforge.net/

[22] S. Kaune, T. Lauinger, A. Kovacevic, and K. Pussep, "Embracing the peer next door: Proximity in kademlia," in *P2P*, 2008.

[23] O. Abboud, A. Kovacevic, K. Graffi, K. Pussep, and R. Steinmetz, "Underlay awareness in P2P systems: Techniques and challenges," in *IPDPS*, 2009.

[24] A. Walters, D. Zage, and C. Nita-Rotaru, "Mitigating attacks against measurement-based adaptation mechanisms in unstructured multicast overlay networks," in *ICNP*, 2006.

[25] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: an overlay testbed for broad-coverage services," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 3–12, 2003.

[26] N. Spring, L. Peterson, A. Bavier, and V. Pait, "Using PlanetLab for network research: Myths, realities, and best practices," *ACM SIGOPS OSR*, vol. 40, pp. 17–24, 2006.

[27] Y. hua Chu, A. Ganjam, T. E. Ng, S. G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," in *USENIX ATC*, 2004.

[28] G. Phillips, S. Shenker, and H. Tangmunarunkit, "Scaling of multicast trees: comments on the Chuang-Sirbu scaling law," *SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 41–51, 1999.

[29] S. Fahmy and M. Kwon, "Characterizing overlay multicast networks and their costs," *IEEE/ACM Trans. Netw.*, vol. 15, pp. 373–386, 2007.

[30] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostić, J. Chase, and D. Becker, "Scalability and accuracy in a large-scale network emulator," in *OSDI*, 2002.

[31] A. Kuzmanovic and E. Knightly, "Low-rate TCP-targeted DOS attacks: the shrew vs. the mice and elephants," in *SIGCOMM*, 2003.

[32] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the transients of adaptation for RoQ attacks on internet resources," in *ICNP*, 2004.

[33] D. Zage and C. Nita-Rotaru, "On the accuracy of decentralized network coordinate systems in adversarial networks," in *CCS*, 2007.

[34] L. Mathy, N. Blundell, V. Roca, and A. El-Sayed, "Impact of simple cheating in application-level multicast," in *INFOCOM*, 2004.

[35] T. Ngan, D. Wallach, and P. Druschel, "Incentives-compatible peer-to-peer multicast," in *P2PECON*, 2004.

[36] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz, "Listen and whisper: security mechanisms for BGP," in *NSDI*, 2004.

[37] X. Hu and Z. M. Mao, "Accurate real-time identification of IP prefix hijacking," in *S&P*, 2007.

[38] I. Avramopoulos and J. Rexford, "Stealth probing: efficient data-plane security for IP routing," in *USENIX ATC*, 2006.