IN-NETWORK DEFENSE AGAINST DISTRIBUTED DENIAL-OF-SERVICE ON THE INTERNET

by

MINGWEI ZHANG

A DISSERTATION

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2019

DISSERTATION APPROVAL PAGE

Student: Mingwei Zhang

Title: In-Network Defense Against Distributed Denial-of-Service on the Internet

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science by:

Jun Li	Chair
Reza Rejaie	Core Member
Hank Childs	Core Member
Jiabin Wu	Institutional Representative

and

Janet Woodruff-Borden	Vice Provost and Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded September 2019

DISSERTATION ABSTRACT

Mingwei Zhang

Doctor of Philosophy

Department of Computer and Information Science

September 2019

Title: In-Network Defense Against Distributed Denial-of-Service on the Internet

Distributed denial-of-service (DDoS) attacks continue to threaten the availability and integrity of critical Internet infrastructure upon which the society relies more heavily than ever before. The extremely high volume and distributed nature of modern DDoS attacks render traditional "edge-defense" solutions (either victim-side or attack-source-side) less effective. This thesis studies in-network DDoS filtering, i.e. filtering traffic inside the Internet, that aims to address these problems by distributing the workload of filtering DDoS traffic at strategically chosen locations *inside* the Internet. This dissertation conducts a systematic study of three different aspects of an effective and deployable in-network DDoS defense, including: 1) in-network defense incentives, 2) in-network defense filter placement strategies, and 3) in-network defense filter placement algorithm design and evaluation. This dissertation not only shows that the majority of the Internet Service Providers (ISPs) have incentive to participate in in-network DDoS defense, but also examines in-network defense strategies, including proposing a new one, and describes the design and evaluation of an effective in-network filter placement algorithm.

This dissertation includes previously published co-authored materials.

CURRICULUM VITAE

NAME OF AUTHOR: Mingwei Zhang

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
Beijing University of Posts and Telecommunications, Beijing, China

DEGREES AWARDED:

Doctor of Philosophy, 2019, University of Oregon
Bachelor of Science, 2012, Beijing University of Posts and Telecommunications

AREAS OF SPECIAL INTEREST:

Network Security
Internet Routing Security
Advanced Networking Technology

PROFESSIONAL EXPERIENCE:

Internet Data Scientist, San Diego Supercomputer Center (SDSC), 2018 - Present

Graduate Research Assistant, University of Oregon, 2012 - 2018

Kernel Quality Engineering Intern, Redhat, Inc., Beijing, China, 2012

PROFESSIONAL SERVICES:

Reviewer, Transaction of Networking
Reviewer, IEEE Transactions on Networking (ToN)
Reviewer, IEEE Transaction on Dependable and Secure Computing (TDSC)
Reviewer, IEEE Transactions on Network and Service Management (TNSM)

Reviewer, IEEE International Conference on Computer Communications (INFOCOM)

Reviewer, Elsevier Journal of Computer and Telecommunications Networking

Co-Organizer, Oregon Cyber Security Day, 2015 - 2016

Graduate School Advisory Board, University of Oregon Graduate School, 2014

PUBLICATIONS:

Lumin Shi, **Mingwei Zhang**, Jun Li, Peter Reiher, "GENI Experiment Configuration Automated," Technical Report CCSP-TR-2019-02, Computer and Information Science, University of Oregon, 2019

Jun Li, **Mingwei Zhang**, Lumin Shi, Devkishen Sisodia, Samuel Mergendahl, Yebo Feng, Peter Reiher, "Victim-driven, Rule-based, In-network Filtering of Distributed Denial-of-Service Traffic," Technical Report CCSP-TR-2019-01, Computer and Information Science, University of Oregon, 2019

Mingwei Zhang, Lumin Shi, Devkishen Sisodia, Jun Li, Peter Reiher, "On Multi-Point, In-Network Filtering of Distributed Denial-of-Service Traffic," IFIP/IEEE International Symposium on Integrated Network Management, 2019

Konstantinos Arakadakis, Pavlos Sermpezis, Vasileios Kotronis, **Mingwei Zhang**, Alistair King, Alberto Dainotti, Xenofontas Dimitropoulos, "Analysis of BGP prefix hijacking events: a commercial service's view," CoNEXT poster, 2018

Mingwei Zhang, Jun Li, Scott Brooks, "I-seismograph: Observing, Measuring, and Analyzing Internet Earthquakes," IEEE/ACM Transactions on Networking, vol. 99, pp. 1-16, 2017

Lumin Shi, **Mingwei Zhang**, Jun Li, Peter Reiher, "PathFinder: Capturing DDoS Traffic Footprints on the Internet," International Federation for Information Processing (IFIP) Networking 2018.

Jun Li, Josh Stein, **Mingwei Zhang**, Olaf M Maennel, "An Expectation-Based Approach to Policy-Based Security of the Border Gateway Protocol," Global Internet Symposium, 2015

Jun Li, Skyler Berg, **Mingwei Zhang**, Peter Reiher, Tao Wei, "DrawBridge – Software-Defined DDoS-Resistant Traffic Engineering," SIGCOMM Poster and Demo Session, 2014.

ACKNOWLEDGEMENTS

I thank my wife, Yuanyuan Jiang, for her love and support during the longest journey I have ever had in my life. Without her, I would not be able to see to the end of this journey.

I thank my parents for the support they provided for my pursue of this Ph.D. degree. They have been providing nothing but care and encouragement since the beginning, even while they were thousands miles away. I could not imagine the difficulties they must have seeing their only child abroad for this many years.

I sincerely thank my adviser, Professor Jun Li, for his support and encouragement along the way. He and the rest of the network and security research group have been a close family to me during the 7 years of my study. Our team work boosted me to where I am and where I can be in the future.

I thank my other committee members: Hank Childs, Reza Rejaie, Jiabin Wu. I received invaluable advises from Hank Childs and Reza Rejaie, on academic, career, and the understanding of the future paths. Collaboration with Jiabin Wu has been a wonderful experience. With his expertise and advises, I was able to open a brand-new chapter for this dissertation.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

LIST OF ALGORITHMS

Algorithm Page

CHAPTER I

INTRODUCTION: IN-NETWORK FILTERING OF DISTRIBUTED DENIAL-OF-SERVICE ATTACK TRAFFIC

1.1 DDoS Attacks and Mitigation

1.1.1 Distributed Denial-of-Service Attacks. Distributed denial-of-service (DDoS) attacks continue to threaten the availability and integrity of critical Internet infrastructure upon which the society relies more heavily than ever before. The ever-growing connectivity and bandwidth of end-hosts on the Internet, coupled with the skyrocketing number of Internet-connected devices, with a vast number of these hosts and devices compromisable to launch distributed denial-of-service (DDoS) attacks, has made the DDoS attacks easier to launch but harder to defend. Despite years of research and industry efforts that have led to a myriad of defense approaches, the Internet has recently witnessed a sharp increase in the number and scale of distributed denial-of-service (DDoS) attacks. Among the most common DDoS attacks are high-volume DDoS attacks that overwhelm a victim's bandwidth, in which such attacks can reach as high as 1.2 Tbps (Dyn Research (2016)), 1.35 Tbps (Kottler (2018)) or even 1.7 Tbps (Morales (2018)). Figure 1 shows an example of a simple DDoS attack. Attacker invoke compromised machines from multiple autonomous systems (ASes), i.e. AS4, AS5, AS6, to send unwanted traffic to the victim at AS1. The DDoS attack travels through AS2 and AS3 to reach the destination.

1.1.2 DDoS Mitigation by Filtering Traffic. This threat continues to grow, despite years of research and industry efforts which have resulted in a myriad of unique DDoS traffic filtering strategies, ranging from source-end filtering to paid traffic-scrubbing services that reside in the cloud. Many existing DDoS

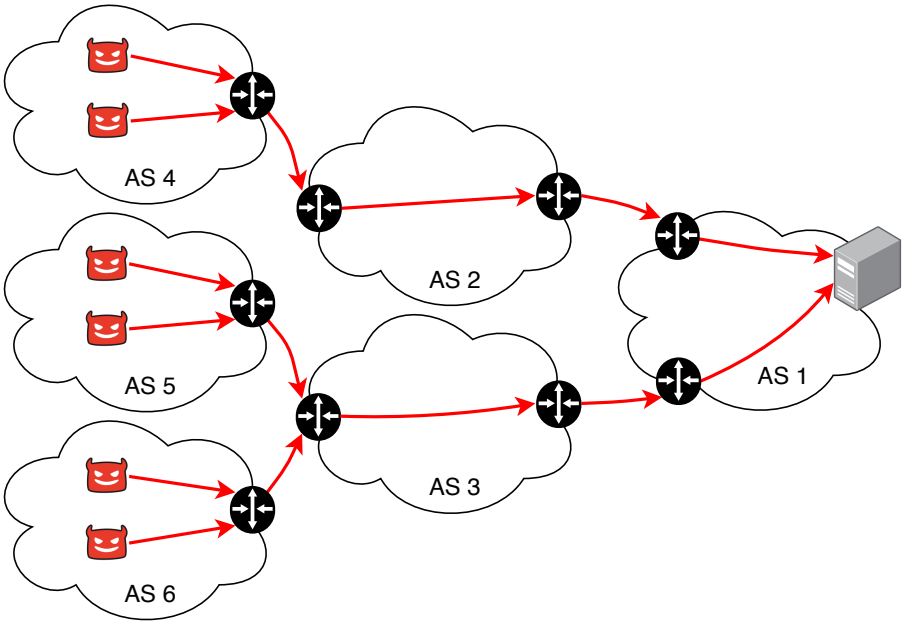


Figure 1. Example of an DDoS attack.

traffic filtering strategies conduct traffic filtering at the edge of the Internet ("edge-defense"), i.e. locations topologically close to or at the attack victims. Such defense does not stop the attack traffic from aggregating to a significant volume causing link congestion before the traffic even reaching the victim.

On the other hand, the Internet has seen a growing proliferation of filtering capabilities throughout. With Access Control Lists (ACLs) built into routers by vendors from day one, broader usage of Border Gateway Protocol (BGP) flow specification, the advent of software-defined networking (SDN), and so on, Internet service providers (ISPs) at the core of the Internet or proxies and firewalls at the edge are equipped and ready to filter traffic, including DDoS traffic. Unfortunately, their filtering of DDoS traffic has mostly been preliminary, such as simply blocking all traffic to a victim, including the legitimate packets. Thus, the Internet today is prepared to filter DDoS traffic, but it holds limited knowledge on how.

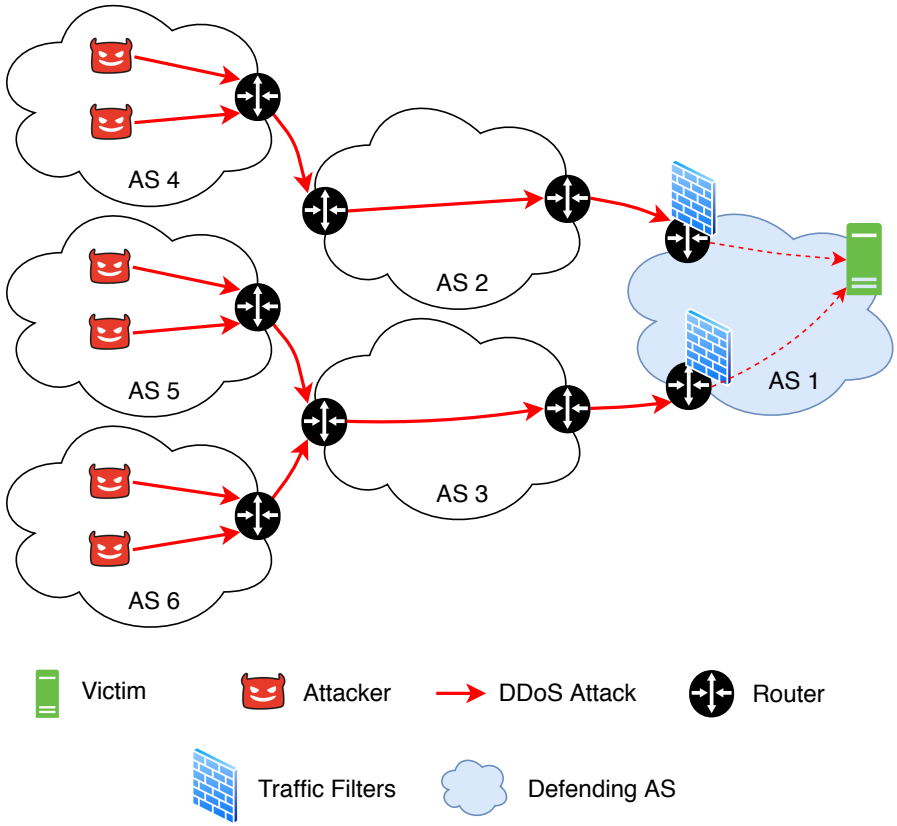


Figure 2. Example of an edge DDoS traffic filtering.

1.1.3 Edge Traffic Filtering. Traditionally, the DDoS defense is considered *edge defense*, in which either the DDoS victim, or a third-party entity is entrusted to conducts the defense, and the defense happens at the edge of the Internet. There are two problems that makes edge defense less effective against current and future DDoS attacks. First, the cost is usually very high for any one single entity to handle Terabit-per-second-level DDoS attack traffic. Making things worse, the attackers can now more easily tap to increasingly popular yet less secure Internet-of-Things devices to launch attacks with record-high volume (Lumbis, Ramdoss, and Miller (2014); Pagiamtzis and Sheikholeslami (2006)). Secondly, even with sufficient investment on handling incoming DDoS traffic at the edge, the

defense, in many cases, can already be late due to traffic congestion that happens before reaching the edge. It is not uncommon to see traffic congestion happen before reaching the victim, and work in Kang, Lee, and Gligor (2013) revealed attacks that triggers congestion around the victim without directly launch attacks at the victim. Edge defense cannot sufficiently handle such cases, and people turn to in-network defense solutions.

Figure 2 shows and example of DDoS traffic filtering happens at the edge of the Internet, specifically at the victim of the attack. The victim in this attack (AS1) places traffic filters at its networks' ingress points to stop the unwanted DDoS traffic from reaching the victim. However, in the scenario when the traffic gets congested before reaching the victim, the victim-side traffic filtering cannot stop the packet loss caused by congestion upstream.

1.1.4 In-network Traffic Filtering. *In-network DDoS defense*, suggested by the name, places the defense efforts inside the Internet, along the paths of the DDoS attack traffic. Instead of defending at the edge, they defend against DDoS attacks before the traffic reaches the victim, often when the DDoS traffic is even further away from the victim's network. These solutions often also distribute the defending workload among a set of defensive collaborators, each in charge of a portion of the traffic. Collectively, the set of collaborators are able to handle a larger volume of DDoS traffic than any individual collaborator. It has the following advantages over edge defense. First, in-network defense allows sharing of the defense load, reducing the defense efforts required at each defending entity. Defenders carry less burden, and can achieve higher overall defense capacity Second, the filtering of DDoS traffic can happen earlier, reducing the traffic load along way to the victim, thus mitigating the traffic congestion on the links before

reaching the victim. Overall, in-network DDoS defense becomes more suitable and efficient than edge defenses in current Internet environment.

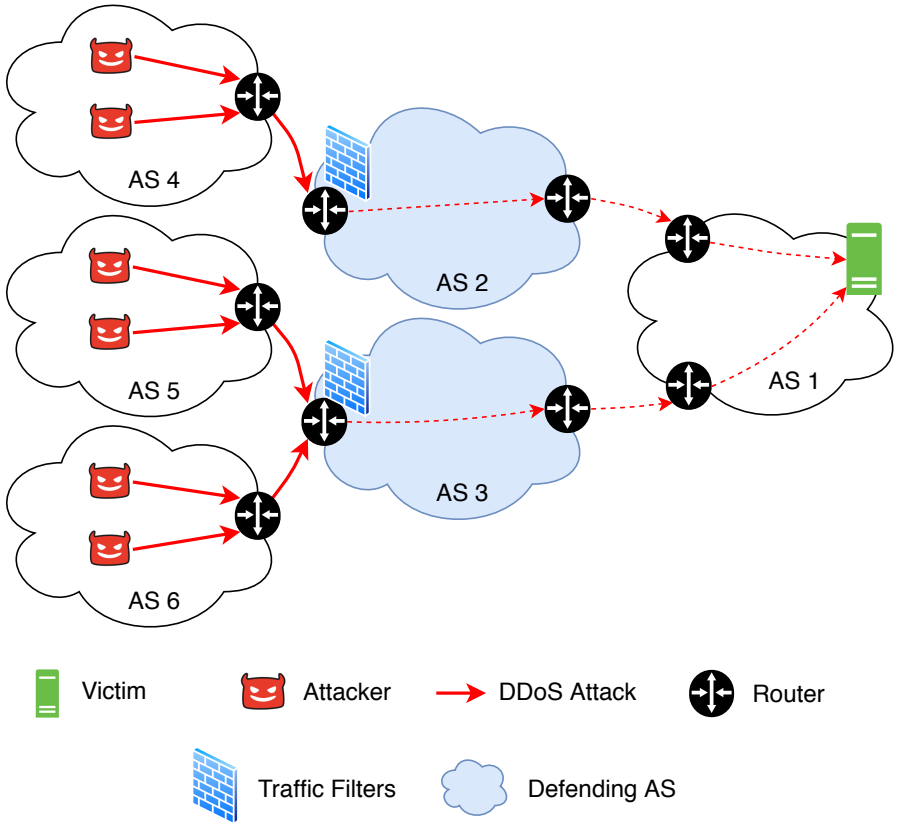


Figure 3. Example of an in-network DDoS traffic filtering

Figure 3 shows and example of in-network DDoS traffic filtering. The victim in this attack requests filters to be placed at its upstream providers, i.e. AS2 and AS3, and the providers place traffic filters at their network to stop the corresponding traffic. In-network traffic filtering stops the traffic early and reduce the traffic stress at the links downstream.

1.2 What's Missing

Although many works have studied topics related to efficient filtering of DDoS traffic, there are still important and yet unresolved issues that makes

effective in-network traffic filtering difficult to achieve. In this section, we will summarize the important related work and describes the missing points. Chapter II describes the related-work in more details.

First, there is a lack of studies that investigate incentives of in-network (or collaborative) DDoS filtering. Although many works have studied incentives for potential defenders to participate defense in different fields of cyber security, such as presented in Bedi, Roy, and Shiva (2011); Bohawek, Hespanha, Lee, Lim, and Obraczka (2007), majority of such studies focus on the interactions between the defenders and the attackers, and leaving the influences of among the defenders unexplored. In the context of in-network traffic filtering, the inter-dependent relationships among defenders is an key factor that affects the incentives, especially when defenders are competitors themselves.

Second, there is a lack of comprehensive surveys and modeling of the filter-placing strategies that in-network traffic filtering solutions can apply. Although many in-network DDoS defense solutions have been proposed, it still remains difficult for a victim to select suitable defense solutions against specific DDoS attacks. The in-network defense solutions (such as DefCOM, PushBack, and MiddlePolice) vary greatly in resource requirements, training data needed, and expected efficiency. Selecting a sub-optimal defense solution could introduce substantial cost and even result in unsuccessful defense. However, there is no quantitative study on how the solutions compare to each other, nor a general model that describes these solutions in a common language. Further, it is also unknown how these solutions perform under insufficient knowledge of the attacks or against intelligent adversaries who can dynamically revise their attack strategies to escape defense. Without a quantitative comparison, it is hard for a DDoS victim to

select the most suitable solution to achieve its defense goal and meet the resource requirements.

Third, there is a lack of design and evaluation of an in-network, DDoS-filtering rule placement algorithm that can effectively deploy traffic filters at optimal locations. Many existing solutions for in-network traffic filtering only focusing on providing communication channels to enable collaboration, but leaving the filter placement strategies unexplored. With placement strategy selected, much work still needs to be done to properly design specific algorithm that implement the selected placement strategy and adjust to inputs from the system in order to achieve maximum performance. Deploying of the filters at the appropriate locations also requires the system to be designed and implemented to work with current deployable technology.

1.3 Dissertation Statement

This dissertation addresses the missing gaps described in Section 1.2. The dissertation statement is as follows:

In-network filtering of the distributed denial-of-service attack traffic is more advantageous than filtering at the edge, can incentivize Internet service providers to adopt, and can be implemented via an effective rule placement algorithm.

1.4 Research Contributions

This research aims to bridge the gaps, and better understand the approaches toward building an effective DDoS traffic filtering systems against DDoS attacks. Overall, this dissertation makes contributions in the following three dimensions: the *incentives* of networks participating in in-network traffic filtering; the modeling and improvement of the *strategies* that can be applied to conducting the filtering; and

lastly the design, and evaluation of an *effective filtering rule placement algorithm.* Detailed contributions are as follows.

Incentives to participate in in-network traffic filtering:

First, we study the underlying incentives for networks on the Internet to defend against DDoS attacks. We propose a game-theoretical model that abstracts the interactions between customer and provider networks via probabilistic provider selection, examines the incentives of ASes to invest in efforts on DDoS defense. Based on the model, we built a large-scale simulation system and examine 1) whether networks on the Internet can be incentivized to participate in in-network traffic filtering, and 2) the affects of a network's topological location, level of competition, and the amount of DDoS traffic it carries for its customer affects its decision on DDoS filtering efforts.

We observe the following patterns from the simulation results. The majority of the provider ASes on the Internet can benefit from providing DDoS defense services to their customers if they can compensate the defense cost by charging for filtering DDoS traffic. The severity of DDoS attacks affects the charge rate a provider can place on its potential customers; if a provider sees higher volume of DDoS traffic going through its potential customers, it would charge higher to achieve its peak profit. The level of competition also drives the charge rate: a provider with low-level competition can charge a high rate while still profitable; a provider that faces strong competitions need to charge less to attract customers for profit.

These observations provide confidence that if in-network collaborative defense mechanisms mature enough provider ASes on the Internet would have incentive to participate in DDoS defense. We believe that such observations can

further help researchers to develop better strategies to devise and deploy DDoS defense solutions.

Strategies to apply for in-network traffic filtering:

Second, with proper understanding of the incentives of networks on DDoS defense, we then study the theoretical strategies on in-network traffic filtering. We introduce a modeling and simulation framework to systematically evaluate in-network DDoS defense algorithms. The framework contains a general model that can describe the attack and defense for various defense algorithms. Using this model, we summarize the existing in-network DDoS defense algorithms into two basic types: PushBack and SourceEnd. A PushBack algorithm employs propagation-based mechanisms to locate suitable defense locations that are close to the victim. A SourceEnd algorithm tries to locate sources of attacks and deploy filtering rules as close to the sources as possible. These two types of algorithms cover most in-network DDoS defense solutions. We then introduce a new type of in-network algorithms that utilizes the topology of the attack sources and ASes *en route* and locates suitable defending ASes in network at critical locations. We call this type of algorithms *StrategicPoints*. We study both the existing PushBack and SourceEnd algorithms and our proposed StrategicPoints algorithms in depth on their defense performance, resource cost, and resiliency against intelligent adversaries. We compare the results using two metrics: DDoS traffic leakage to the victim and DDoS traffic pollution on the Internet. The simulation results provide useful insights into the selection of defense algorithms in response to different attack scenarios. StrategicPoints strikes a balance between DDoS traffic coverage and pollution reduction, and is more effective in cases when resources are not extremely restrictive.

Filtering rule placement algorithm for optimal filtering performance:

Lastly, we design, implement, and evaluate an in-network DDoS traffic filtering rule placement algorithm that is effective against large-scale DDoS attacks and able to locate optimal placement locations for traffic filtering rules. The study is focused on both systematic and algorithmic design of the rule placement algorithm. Specifically, we design an efficient rule placement algorithm that can find optimal placements for a give set of rules. To achieve optimal results, we design a tree-like data structure called H-tree to model the placements of the rules on the DDoS traffic topology toward the victim. As a result, our algorithm can find placement locations that maximize the coverage of the DDoS traffic, minimize the rule space needed for deployment, and maximize the distance away from the victim's network.

1.5 Chapter Relationships

Although we studied in-network DDoS traffic filtering from three different dimensions (i.e. incentives, strategies, and system designs) in separated chapters, the efforts are inherently connected and the results benefits each other in return.

In particular, the relationships between the chapters is described as follows (also shown in Figure 4). The study of the incentives for networks to conduct in-network traffic filtering improves the understanding of the likelihood in-network filter being deployed on the Internet. With confidence of the deployment incentives, we can assume larger number of networks on the Internet can be utilized for in-network filtering purpose. We then conduct study on the survey, modeling, and improving of filter placement strategies which allow the potential DDoS attack victims to better understand the efficacy and drawbacks of different filter placing

mechanisms. Knowing the appropriate strategy for in-network filtering, the algorithm design chapter introduces an effective algorithm that implements the strategy and achieves the filtering of DDoS traffic in in-network style. The existence of well-performing traffic filtering algorithm will further boost the incentives for networks to participate in in-network traffic filtering, completing a full circle of the three dimensions.

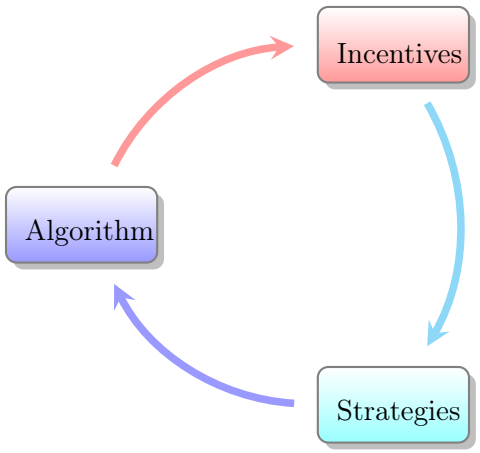


Figure 4. Relationship between chapters

1.6 Scope of this Dissertation

In this dissertation, we focus our efforts on the discovering of the incentives for conducting in-network DDoS traffic filtering, the survey and improvement of strategies for placing traffic filtering rules, and the design and evaluation of an in-network traffic-filtering rule placement algorithm. There are also several related research issues tightly intertwined with DDoS traffic filtering, each with their own significance. We consider these topic orthogonal to our efforts and describe them in the rest of this section.

One topic is the detection of DDoS attacks and the classification of the DDoS attack traffic. Researchers have studied DDoS detection and classification extensively. We believe a victim should also play an active role in detecting and

classifying DDoS traffic, given that it knows better what legitimate traffic it expects than filtering nodes inside the network. Nonetheless, in this dissertation we do not focus on DDoS detection and classification, but refer readers to existing rich literature on this topic. The design of our in-network DDoS traffic filtering system allows users to plugin existing traffic detection and classification system. Albeit the correctness of detection and classification of DDoS traffic affects the mitigation results, we consider these process orthogonal to our filtering system, where we focus on where and how to most effectively deploy filters as filters coming in as input from other components.

Another topic of its own significance is IP spoofing, as DDoS bots can spoof their source IP address during a DDoS attack. Fortunately, not only has IP spoofing been studied as a separate topic extensively, it has also been addressed fairly successfully in the real world. In particular, with ever-growing deployment of ingress filtering (Baker and Savola (2004)), nowadays, the number of IP addresses that are still spoofable has reduced to 16.5% (Beverly and Bauer (2005); Matthew Luckie, Ken Keys, Ryan Koga, Rob Beverly, kc claffy (2016)). Also note that many DDoS attacks actually do not employ IP spoofing, either because it is hard to do (e.g. IP address of reflectors in reflector attacks; IP address of bots in TCP-based DDoS such as HTTP floods), can incur extra overhead of the attacker (e.g. to hide a bot from the spoofing detection at their local network), or the attackers simply do not care due to the abundance of DDoS bots. Rather than embedding anti-spoofing mechanisms in the design of our filtering system, we assume IP spoofing to be addressed separately by independent anti-spoofing efforts.

Lastly, although we use volumetric DDoS attacks (i.e. the attacks that uses high volume of junk traffic to overwhelm victim's networking resources) as

an example target for defense, our design of the filtering system is not limited to only this type of attacks. Provided appropriate traffic filters as input, our system is able to locate appropriate locations for placing and applying the filters.

1.7 Road Map

The dissertation is organized to have each chapter address one of the research goals above.

In Chapter II, we survey the work related to DDoS traffic filtering. Section 2.1 overviews the existing traffic filtering solutions for DDoS mitigation. Section 2.2 reviews work related to discovering incentives of deployment for various cyber-security solution. Section 2.3 reviews current literate on strategies of placing traffic filtering rules for DDoS defense. Section 2.4 discusses the existing solutions for conducting in-network traffic filtering.

In Chapter III, we investigate the incentives for networks on the Internet to participate in in-network DDoS defense efforts. Specifically, section 3.2 overviews the problem of discovering incentives for in-network filtering; section 3.3 introduces a game theoretical model of competition among ISPs for customers; section 3.4 describes the simulation setup and algorithms for this study; section 3.5 discusses the simulation results; and section 3.6 summarizes the main conclusions the work.

In Chapter IV, we survey, model, and improve the strategies for the networks to distributes DDoS defense efforts to achieve effective in-network traffic filtering. Specifically, section 4.2 reviews the state-of-the-art related work on in-network DDoS defense algorithms; section 4.3 formally defines our models that describe the Internet, DDoS attacks, and DDoS defenses; section 4.4 introduces a classification of in-network DDoS filtering strategies; section 4.6 examines the simulation results and compares the three defense algorithms against different

metrics; section 4.7 discusses limitations and open issues of this work; and finally section 4.8 summarizes the conclusion and takeaways of this work.

In Chapter V, we introduce the mechanism design, algorithm, and evaluation of a rule placement algorithm that can effectively locate filter deployment locations and filter DDoS traffic. section 5.2 describes our design of the filtering rule placement mechanism; section 5.3 introduces the detailed design of our rule placement algorithm; section 5.4 reviews the evaluation of rule placement algorithm, both using simulation and testbed emulation; section 5.5 summarizes the conclusion and takeaways of this work.

Finally, Chapter VI concludes this dissertation and discusses the future research directions.

1.8 Co-authored Materials and Acknowledgment

1.8.1 Co-authored Materials. Much of the work in this dissertation is from previous collaborative publications. Below is a listing connecting the chapters with the material and authors that contributed. A more detailed elaboration on the division of labor is also provided at the beginning of each chapter.

- Chapter III is mainly based on the work in Zhang, Wu, Li, and Reiher (2019), which is a collaboration between Jiabin Wu, Jun Li, Peter Reiher and myself.
- Chapter IV is mainly based on the work in Zhang, Shi, Sisodia, Li, and Reiher (2019), which is a collaboration between Lumin Shi, Devkishen Sisodia, Jun Li, Peter Reiher and myself.
- Chapter V is mainly based on the work in Li et al. (2019), especially its text on DDoS-filtering rule placement design and evaluation, which is

a collaboration between Jun Li, Lumin Shi, Devkishen Sisodia, Samuel Mergendahl, Yebo Feng, Peter Reiher and myself.

In addition, Chapter II is mainly based on related work sections in (Li et al. (2019); Zhang, Shi, et al. (2019); Zhang, Wu, et al. (2019)).

1.8.2 Acknowledgment. This project is in part the result of funding provided by the Science and Technology Directorate of the United States Department of Homeland Security under contract number D15PC00204. The views and conclusions contained herein are those of the authors and should not be interpreted necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security or the US Government.

CHAPTER II

RELATED WORK

2.1 Overview

At a very high level, the existing DDoS defense can be categorized into two styles Zargar, Joshi, and Tipper (2013): edge traffic filtering, or in-network traffic filtering. Edge filtering mechanisms defend against DDoS attacks within one AS, and usually at the receiving end of the DDoS attack traffic. Single-AS defense solutions, such as work from Sahay, Blanc, Zhang, and Debar (2015) (which redirects attack traffic to middle-boxes close to the victim), RADAR (Zheng et al. (2018)) (which detects and throttles attack traffic at the victim network), SPIFFY (Kang, Gligor, and Sekar (2016)) (which temporarily increases the effective bandwidth of a congested core link and observes the response to detect and mitigate an attack), and Bohatei (Fayaz, Tobioka, Sekar, and Bailey (2015)) (which presents a flexible and elastic DDoS defense system geared towards a single ISP providing customers with DDoS-defense-as-a-service), are easier to deploy and more flexible to implement when compared to in-network defense solutions, especially when network management complexity is reduced by leveraging software-defined networking (SDN). Other industrial systems such as FastNetMon (Odintsov (2019)) and Arbor APS (NETSCOUT (2019)) can detect and filter DDoS traffic. The network operator can also manually connect (e.g. via `ssh`) to local routers to install Access Control Lists (ACLs) to filter the DDoS traffic. However, many edge defense solutions can incur a very high defense cost due to resource requirement in the term of network connection and network devices (Lumbis et al. (2014); Pagiamtzis and Sheikholeslami (2006)), and often fail to mitigate attacks when victims' inbound connections are inundated with DDoS traffic.

The obvious drawbacks with solutions at the edge led to various in-network DDoS defense solutions. The in-network DDoS defense solution, can reduce the amount of resources needed at each collaborating AS and relieve ASes from the heavy burden of network and equipment costs. Methods such as PushBack (Mahajan et al. (2002a)), TVA (Yang, Wetherall, and Anderson (2008)), RAD (Kline, Beaumont-Gay, Mirkovic, and Reiher (2009)), AITF (Argyraki and Cheriton (2005)), DefCOM (Oikonomou, Mirkovic, Reiher, and Robinson (2006)), and StopIt (X. Liu, Yang, and Lu (2008)) have proposed approaches to filtering DDoS traffic in-network, at multiple relevant remote ISPs.

In the rest of this section, we will examine the work related to in-network DDoS traffic filtering from three different angles:

1. section 2.2: the incentives for ASes on the Internet to participate in in-network DDoS traffic filtering,
2. section 2.3: different strategies for placing filters for in-network defense,
3. section 2.4: designs and evaluations of different systems that conduct in-network traffic filtering.

This chapter is derived from related work sections in (Li et al. (2019); Zhang, Shi, et al. (2019); Zhang, Wu, et al. (2019)) that are resulted from collaboration with other co-authors listed in them.

2.2 Discovering Deployment Incentives

There are abundant work studying address cyber-security problems from both incentive and game-theoretical perspectives (Laszka, Felegyhazi, and Buttyan (2014); Manshaei, Zhu, Alpcan, Bacşar, and Hubaux (2013); Papadimitriou (2001); Roy et al. (2010)). Some studies from the angle of interactions between

the attackers and defenders, while some other work explores inter-dependencies and collaborations among defenders. We will survey closely related work in this section.

Numerous research projects have explored using game-theory to study interactions between attackers and defenders in cyber security context. Bedi et al. proposed a model to study optimal firewall settings for DDoS defense against attackers (Bedi et al. (2011)). Bohawek et al. introduced game-theoretic stochastic routing (GTSR) to minimize impact of link and router failures against intelligent attackers (Bohawek et al. (2007)). In Shiva, Roy, and Dasgupta (2010), the authors proposed an holistic architecture that incorporates behaviors of the attackers and decide actions for the defenders, but the work lacks concrete evaluation. Wu et al. (Wu, Shiva, Roy, Ellis, and Datla (2010)) developed a game-theoretical model to study the most effective firewall settings to block DoS/DDoS traffic. All the work above try to model and develop systems to mitigate attacks (which can be intelligent and dynamic) from a single central-controlled entity. However, current cyber attacks on the Internet, especially DDoS attacks, can no-longer be easily mitigated by single AS.

There are studies that investigate the potential collaboration among defenders against cyber-attacks. In Grossklags, Christin, and Chuang (2008b), the authors analyzed how influences among heterogeneous entities could reach different security end-results under five different economic environment. They later studied how the inter-dependent defenders may shift between public good (protection) and private good (insurance) given the choices (Grossklags, Christin, and Chuang (2008a)). Similarly, in Miura-Ko, Yolken, Mitchell, and Bambos (2008), the authors model the impacts of security investment among interdependent organizations using

influence network. However, the lack of quantitative evaluation and conclusion making it less applicable on real-world problems.

Some work study cyber security with a focus on defense incentives. Early work by Huang et al. (Huang, Geng, and Whinston (2007)) analyzed the broken incentive chain that stops ISPs from participating DDoS defense. They argue that the subscription-based pricing model among ISPs at the time, which often incur over-provision and ignores the actual traffic volume pattern, discourages ISPs from participating defense by doing extra work. They suggest that traffic-usage-based pricing model would incentivize ISPs to help filtering out unwanted traffic. Unfortunately, the authors did not consider the potential revenue that the ISPs could have made by *not participating in DDoS defense*; therefore, as the Internet gradually shift to usage-based pricing model, the apparent lack of incentive still persist. Gill et al. (Gill, Schapira, and Goldberg (2011)) argues that efforts in deploying more secure inter-domain routing protocol (i.e. S*BGP) would allow the deployer to attract more inter-domain traffic, and thus generate more revenue, matching results of early discussion in Sami, Katabi, Faratin, and Wroclawski (2004). Different from DDoS defense, securing inter-domain routing does not have negative incentive, i.e. not defending does not introducing extra revenue. However, as the authors suggest, both strong early adopters and simplified protocol are needed for global deployment. Shen et al. (Shen, Yan, and Kantola (2013)) studies the deployment incentive of their previous work using game-theory. They model the deployment problem as a social dilemma, and suggest that ISPs can be incentivized by combining the benefits of achieving public good, and potential punishment for untrustworthy behavior. However, public good itself does not provide strong enough motivation for private companies, and trust assessment and

behavior punishment require enforcement from global central authorities, which is not a realistic assumption under the context of current Internet.

Summary: To summarize, there lacks study that explores the incentives for in-network DDoS filtering, leaving the incentives for deployment of such systems unclear. More specifically, due to the collaborative nature of in-network traffic filtering, the inter-dependent relationships among potential defenders is a key factor when considering the incentives, and should the modeled and examined.

2.3 Strategies for In-network DDoS Traffic Filtering

Table 1. DDoS defense solution categorizations

Work	Single-AS	Multi-AS		
		PushBack	**SourceEnd**	**Other**
RADAR, Sahay et al. 2015, SPIFFY, Bohatei	✓			
ScoreForCore, Mahajan et al. 2002, Yau et al. 2005		✓		
FireCol, DefCOM, AITF, COSSACK, StopIt, D-WARD, Huici et al. 2007, Argyraki et al. 2009			✓	
MiddlePolice, Andersen et al. 2003, Keromytis et al. 2004				✓

Almost all in-network defense solutions require placement strategies to decide where on the Internet to deploy traffic filters or defense measure. We categorize existing in-network defense strategies into three categories: *PushBack*, *SourceEnd*, and *other* (shown in Table 1).

PushBack defense: We define PushBack defense algorithms as those that start defense from a victim AS and expand the defense area to its upstream ASes. Starting from the victim AS, PushBack allows each defending AS to mitigate a portion of the attack traffic, and delegate the rest of the attack traffic to its upstream ASes for further mitigation. The original PushBack style defense propagation is introduced in the *PushBack* paper (Mahajan et al. (2002b)). Although this work considers router-level defense propagation, the basic idea can be applied to AS-level collaboration. Other distributed defense systems that stem from the classic PushBack work, such as the work of Yau et al. (Yau, Lui, Liang, and Yam (2005)) and ScoreForCore (Kalkan and Alagöz (2016)), follow the same PushBack algorithm to defend against DDoS attacks.

SourceEnd defense: Different from a PushBack algorithm where the defense initiated from the victim side, a SourceEnd algorithm attempts to select the ASes that are the sources of the attack or close to the sources, and only fall back on downstream ASes toward the victim if resources run out. D-WARD system (Mirković, Prier, and Reiher (2002)), for example, installs rate-limiting rules at border routers in source networks; COSSACK (Papadopoulos, Lindell, Mehringer, Hussain, and Govindan (2003)) deploys countermeasures at the ASes of attacking sources. Both are early works that employ the SourceEnd strategy. Later work such as AITF (Argyraki and Cheriton (2005)) introduced the idea of propagating the defense from the attacking sources to the victim, thereby providing more flexibility for defense deployment. Specifically, authors of AITF also observed that the current generation of routers have sufficient filtering resources to mitigate DDoS attacks as long as the attack traffic was blocked close to the attacking sources. Furthermore, AITF was also one of the earliest projects to study hardware rule space during

defense. Later work from Huici et al. (Huici and Handley (2007)) and StopIt (X. Liu et al. (2008)) enhance AITF by introducing security measures against DDoS attacks on the defensive infrastructure itself. Unlike PushBack solutions, prior to defending against the attack all SourceEnd solutions need to know the attack topology (i.e., the attack sources and their AS-level routes toward the victim) for each DDoS victim

Systems without clear defense placement strategies: Besides the aforementioned two categories, there also exist systems that provide DDoS defense frameworks without clear placement strategies. Previous work from Keromytis et al. (Keromytis, Misra, and Rubenstein (2004)) and Anderson (Andersen (2003)) introduced authentication nodes at key locations between the sender and the receiver in order to filter out unwanted traffic. More recent work from Liu et al. (Z. Liu, Jin, Hu, and Bailey (2016)) (MiddlePolice) utilizes SDN to measure network congestion status, exchanges measurement among collaborating ASes to discover congested links across the Internet, and then places traffic filters at the routers within congested ASes. However, these works do not clearly state where on the Internet the defense should happen, leaving the decisions to the operators or other algorithms. Without a detailed defense strategy, it is difficult to judge how these systems perform under different DDoS attacks.

Summary: Although many in-network DDoS defense solutions have been proposed, it still remains difficult for a victim to select suitable defense solutions against specific DDoS attacks. The in-network defense solutions (such as DefCOM Oikonomou et al. (2006), PushBack (Mahajan et al. (2002b)), and MiddlePolice Z. Liu et al. (2016)) vary greatly in resource requirements, training data needed, and expected efficiency. Selecting a sub-optimal defense solution could introduce

substantial cost and even result in unsuccessful defense. However, there is no quantitative study on how the solutions compare to each other, nor a general model that describes these solutions in a common language. Further, it is also unknown how these solutions perform under insufficient knowledge of the attacks or against intelligent adversaries who can dynamically revise their attack strategies to escape defense. Without a quantitative comparison, it is hard for a DDoS victim to select the most suitable solution to achieve its defense goal and meet the resource requirements.

2.4 Systems for In-network DDoS Traffic Filtering

In this section, we survey the designs and evaluations of historical and state-of-the-art DDoS traffic filtering systems.

The obvious drawbacks with solutions at the edge led to various in-network DDoS defense solutions. Methods such as PushBack (Mahajan et al. (2002a)), TVA (Yang et al. (2008)), RAD (Kline et al. (2009)), AITF (Argyraki and Cheriton (2005)), DefCOM (Oikonomou et al. (2006)), and StopIt (X. Liu et al. (2008)) have proposed approaches to defend at multiple relevant remote ISPs. A major concern with these approaches is their deployability: as they require either router modification (Pushback, TVA, StopIT) or packet marking (RAD, AITF, DefCOM, StopIT). While router modification is clearly an obstacle for deployment, packet marking is also not practical in the modern Internet and causes switching performance penalty.

In response to the sharp increase in the number and scale of DDoS attacks, the scrubbing center approach, which is readily deployable, has gained popularity in recent years. It redirects all incoming traffic for a customer to scrubbing centers, processes and cleans the traffic there, and then forwards DDoS-free traffic back to

the customer. Companies like Arbor Networks, CloudFlare, or Akamai, all offer cloud-based DDoS-scrubbing service as well as possibly other proprietary on-site solutions (*Akamai DDoS Protection Service* (2016); *DDoS Protection By Arbor Networks APS* (2016)). Currently, scrubbing centers mainly employ proprietary DDoS filtering algorithms, rather then enforcing victim-driven rules or policies (Z. Liu et al. (2016)). Besides incurring extra overhead via DNS or BGP to reroute traffic to scrubbing centers, it has also been shown that a clever attack can bypass the scrubbing centers (Miu et al. (2013); Vissers, Van Goethem, Joosen, and Nikiforakis (2015)).

The advent of SDN, due to its friendliness to deploying rules to filter traffic, has also led to a variety of SDN-based DDoS defense solutions, including (Fayaz et al. (2015); Sahay et al. (2015); Wang, Zheng, Lou, and Hou (2015)). Although both approaches in Sahay et al. (2015) and Wang et al. (2015) populate SDN switches with DDoS-filtering rules, little information is offered how they generate such rules. The Bohatei approach in (Fayaz et al. (2015)), on the other hand, assumes the ISP has a predefined library of defenses specifying a defense strategy for each attack type, and deploys virtual machines of appropriate type, number and location according to the current DDoS type. Much is yet to be done to ensure SDN switches are instructed to deploy the right DDoS traffic filtering rules, at the right time at the right locations, while achieving multiple potentially conflicting objectives for both rule generation and placement. In most of these work, the responsibility of filtering traffic is shared among multiple network devices that are located either within the same network or different networks through collaboration.

Clearly, neither defense at the edge nor in-network defense can address the DDoS problem alone, and researchers, including us, have noticed the disconnection

between the edge and the in-network filtering capabilities. In following the paradigm of filtering *in network* but receiving commands from the edge (i.e. the victim), several approaches have appeared in recent years. There are existing DDoS solutions that allow end-users to play a role in their own However, these works do not address the problem of generating and placing traffic-filtering rules. MiddlePolice (Z. Liu et al. (2016)) diverts traffic to traffic policing units referred as mboxes that uses an information table to police traffic flows; in particular, in order to mitigate DDoS, it enables a DDoS victim to send the mboxes traffic control policies to dictate how bandwidth should be allocated among flows to the victim. SENSS (Ramanathan, Mirkovic, Yu, and Zhang (2018)) instead has the victim send specific filtering requests to SENSS servers running at ISPs in order to filter traffic *en route* toward the victim. Stellar (Dietzel, Wichtlhuber, Smaragdakis, and Feldmann (2018)) uses so called Advanced Blackholing to allow ASes to send *blackholing rules* to their IXPs to filter DDoS traffic toward them. These approaches use different in-network filtering entities, but they are all victim-driven. However, unlike DrawBridge that generates rules for effective DDoS filtering with minimal collateral damage and low filtering overhead, these approaches do not employ a similar mechanism. While in MiddlePolice a victim can issue traffic control policies and in SENSS a victim can issue filtering requests, there is no discussion what policies or requests can achieve the best efficacy of DDoS filtering with minimal collateral damage and low overhead, and neither do they generate such policies or requests. In Stellar, when a victim issues rules, the system can limit their collateral damage by increasing their granularity, but doing so would quickly increase the number of rules. Also, unlike DrawBridge that chooses strategic in-network filtering locations by considering their filtering capabilities and various

attributes, in MiddlePolice a victim is simply bound with certain mboxes, in SENSS a victim simply talks to SENSS servers known to the victim, and in Stellar a victim simply uses available IXPs. Other systems like source-end approaches are less popular due to the difficulties to locate a large number of DDoS sources and deploy defenses against them; example solutions include D-WARD (Mirković et al. (2002)), which installs rate-limiting filters at border routers in source networks, and COSSACK (Papadopoulos et al. (2003)), which deploys countermeasures at the ASes of attacking sources.

Summary: In summary, there currently lacks design and evaluation of in-network traffic filtering systems that both respect flexible user input to control the defense, and can effectively deploy traffic filters at optimal locations. Many existing solutions for in-network traffic filtering only focusing on providing communication channels to enable collaboration, but leaving the filter placement strategies and filter deployment unexplored.

CHAPTER III

INCENTIVES FOR IN-NETWORK DDOS TRAFFIC FILTERING

To study the in-network DDoS traffic filtering, we first need to investigate whether networks have any incentives to participate in such defenses at all. In this chapter, we develop a game-theoretical model that describes the interactions among the Internet service providers and between customers. Through both static and dynamic simulation, we discover that networks on the Internet in general can be incentivized to participate in in-network traffic filtering.

This chapter is directly derived from Zhang, Wu, et al. (2019), resulted from collaboration with other co-authors listed in the manuscript. Mingwei Zhang is the primary author of this work, including co-designing and evaluating the incentive mechanisms for in-network DDoS filtering.

3.1 Overview

Distributed denial-of-service (DDoS) attacks have become increasingly more frequent and powerful. The scale of recent DDoS attacks have reached over one Terabit per second from tens of thousands of unique attack sources. The traditional *edge filtering* solutions can no longer handle the situation, and *in-network filtering* solutions are called upon which involve multiple Internet service providers (ISPs) to collaboratively defend against the attacks. While collaborative defense solutions can be more effective in stopping large-scale attacks from a technical perspective, the incentives of the ISPs to deploy these solutions are left unexplored. Without proper incentives for participation, in-network style traffic filtering solutions cannot be effectively applied to due to the lack of participation from networks that are involved forwarding DDoS traffic.

In this chapter, we develop a game-theoretical model to capture the economic benefits and costs of deploying in-network filtering solutions for the ISPs who are competing for customers. Through large-scale simulations at the Internet level, we have the following observations: the majority of the providers on the Internet have economic incentive to participate in DDoS defense driven by competition; and the severity of DDoS attacks and the level of competitions impact the charge for filtering DDoS traffic a provider can impose on its customers. To our best knowledge, this is the first study examines the incentives for the deployment of in-network filtering solutions.

Scope of This Chapter:

This chapter investigates the incentives for ISPs in-network filter DDoS traffic from economics angle using game-theory and simulations. This chapter creates theoretical base for the further studies on the topic of in-network DDoS traffic filtering. Chapter IV studies the filter placement strategies and chapter V examines different aspects on system design, both of which uses the results from this chapter to support the assumption that multiple ISPs on the Internet collaborate together for DDoS traffic filtering.

3.2 Background

3.2.1 In-network Filtering of DDoS Traffic.

Distributed denial-of-service (DDoS) attack has been plaguing the Internet for more than a decade now. DDoS attacks utilize a large number of attacking sources (e.g. compromised computers) to flood the victims' networks with unwanted traffic to exhaust the victims' network, computation, and other types of resources. Recent DDoS attacks have reached a record-high 1.2 Terabit per second (Kottler (2018)), which can pose severe threat to all online services.

Traditionally, the DDoS defense is considered *edge defense*, in which either the DDoS victim, or a third-party entity is entrusted to conducts the defense, and the defense happens at the edge of the Internet. Multiple problems stops edge defense from effectively filtering DDoS attacks traffic. First, the cost is usually very high for any one single entity to handle Terabit-per-second-level DDoS attack traffic as suggested in Lumbis et al. (2014); Pagiamtzis and Sheikholeslami (2006). Making things worse, the attackers can now more easily tap to increasingly popular yet less secure Internet-of-Things devices to launch attacks with record-high volume. Secondly, even with sufficient investment on handling incoming DDoS traffic at the edge, the defense, in many cases, can already be late due to traffic congestion that happens before reaching the edge. It is not uncommon to see traffic congestion happen before reaching the victim, and work in Kang et al. (2013) revealed attacks that triggers congestion around the victim without directly launch attacks at the victim. Edge defense cannot sufficiently handle such cases, and people turn to in-network defense solutions.

In-network DDoS defense, as suggested by the name, places the defense efforts inside the Internet, along the paths of the DDoS attack traffic. It has the following advantages over edge defense. First, in-network defense allows sharing of the defense load, reducing the defense efforts required at each defending entity. Defenders carry less burden, and can achieve higher overall defense capacity Second, the filtering of DDoS traffic can happen earlier, reducing the traffic load along way to the victim, thus mitigating the traffic congestion on the links before reaching the victim. Overall, in-network DDoS defense becomes more suitable and efficient than edge defenses in current Internet environment.

3.2.2 ISPs Lack Incentives to Filter.

Although technically feasible, it is, however, still unknown if in-network traffic filtering mechanisms are economically beneficial to the collaborators. In edge-defense cases, the defender is either the victim or directly serving the victim, making the incentive of defense clear. However, for ASes involved in-network defense, the incentive on participating in defense in unclear. Filtering DDoS traffic towards its customers would decrease an ISP's profit due to the reduction of the amount of traffic forwarded. If an ISP is the sole provider for its customers, it is always in the ISP's best interest to forward as much traffic as possible as long as revenues exceed costs. This creates a conflict where customers want provider to filter DDoS traffic, while providers want to continue forwarding such traffic to profit. Without proper incentives, it can seem daunting to convince service providers to give up on that portion of profit for the good of others.

3.2.3 Competition Creates Incentives.

Fortunately, as the Internet infrastructure grows, hardly any single ISP can monopolize the entire service market. Instead, ISPs face competition. Customers who pay for the Internet access would expect non-interrupted services and they would naturally choose the ISP who exerts the most efforts on stopping or mitigating incoming DDoS attacks.

Given competition, the benefits of an ISP investing in DDoS defense becomes clear. By investing a higher effort on DDoS defense, an ISP has a higher chance of being selected by customers among its competitors and the right of carrying the traffic for customers yields revenues. However, providing DDoS defense service does not come without its costs. First, an ISP would lose revenues by dropping the DDoS traffic that would have traveled to the customers. Second, deploying new defense solutions is costly, both in terms of equipment and

maintenance costs. The benefits of becoming the winner of the competition provide an incentive to the ISPs to invest efforts. However, it is also possible for an ISP to refrain itself from competition given the potential loss of revenues and the increase of workloads and expenses.

3.2.4 Related Studies. There are abundant work studying address cyber-security problems from both incentive and game-theoretical perspectives surveyed in Laszka et al. (2014); Manshaei et al. (2013); Papadimitriou (2001); Roy et al. (2010).

Some projects have explored using game-theory to study interactions between attackers and defenders in cyber-security context, such as in Bedi et al. (2011); Bohawek et al. (2007); Shiva et al. (2010); Wu et al. (2010). However, all work above try to model and develop systems to mitigate attacks (which can be intelligent and dynamic) from a single central-controlled entity, none is directly applicable in the in-network filtering context.

There are studies that investigate the potential collaboration among defenders against cyber-attacks, as presented in Grossklags et al. (2008a, 2008b); Miura-Ko et al. (2008). However, the lack of quantitative evaluation using real-world network topology and attacks making them less applicable on real-world problems.

Some more closely related studies focus on deployment incentives for cyber-security solutions. Early work by Huang et al. Huang et al. (2007) suggested that traffic-usage-based pricing model would incentivize ISPs to help filtering out unwanted traffic. Unfortunately, the authors did not consider the potential revenue that the ISPs could have made by *not participating in DDoS defense*; therefore, as the Internet gradually shift to usage-based pricing model, the apparent lack

of incentive still persist. In Gill et al. (2011) the authors argue that efforts in deploying more secure inter-domain routing protocol (i.e. S*BGP) would allow the deployer to attract more inter-domain traffic. Different from DDoS defense, securing inter-domain routing does not have negative incentive, i.e. not defending does not introducing extra revenue, thus the results are not directly applicable. In Shen et al. (2013) the authors argue that ISPs can be incentivized by combining the benefits of achieving public good, and potential punishment for untrustworthy behavior. However, public good itself does not provide strong enough motivation for private companies, and trust assessment and behavior punishment require enforcement from global central authorities, which is not a realistic assumption under the context of current Internet.

3.3 Game of Traffic: DDoS Defense Investment

When a customer AS selects its provider, the effectiveness of the potential provider's protection against potential DDoS attacks is an important factor as the large-scale attacks becomes more frequent and devastating. As a provider AS, when it invests in DDoS defense, it affects not only the profit of itself, but also could affects its current and potential customers' provider choice, thus also affects the profits of its competitors. In this section, we introduce a game theoretical model to capture the strategic interactions and their effects among the ASes on the Internet.

3.3.1 Network Modeling.

3.3.1.1 Internet Topology. We consider the Internet as a weighted-directed graph $G = \{V, E\}$. V represents the set of all nodes (vertices) in the graph and E represents the set of all edges in the graph. Each node $v_i \in V$ represents an autonomous system (AS) on the Internet. Each edge $e_{i,j} \in E$ represents the inter-AS directional link between two neighboring ASes v_i and v_j.

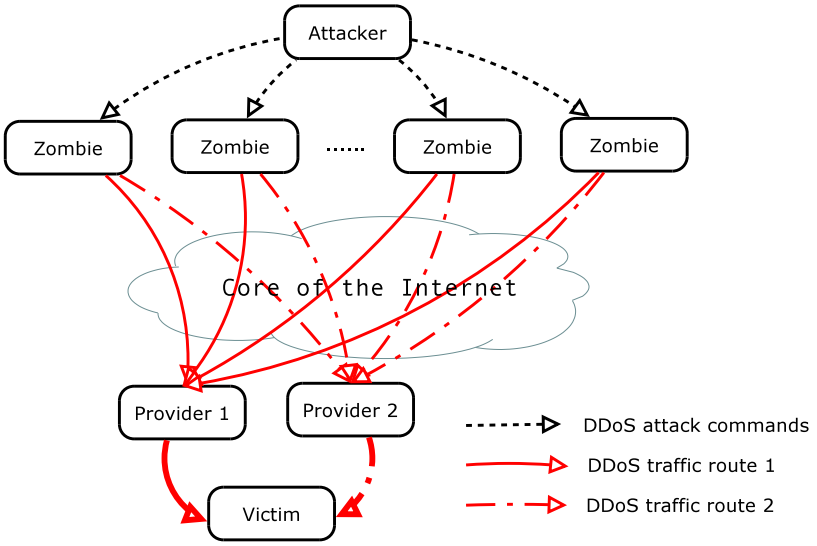


Figure 5. An example of DDoS attack with multiple routes to reach the victim.

ASes on the Internet forms business relationships to establish links (both physically and topologically) to connect themselves with other entities and further with the rest of the Internet. There are three common types of business relationships between the ASes as summarized in Gao (2001): customer-to-provider, peer-to-peer, and sibling-to-sibling. In a customer-to-provider relationship, the provider AS carries traffic from and to the customer AS, and the customer AS pays its provider AS for the traffic the provider carries for it. In peer-to-peer sibling-to-sibling relationships, two ASes forward traffic for each other usually free of charge. In this study, we mainly consider the ASes with customer-to-provider relationship.

For an IP prefix (or IP block) to be reachable, the owner AS must announce the block to the Internet using Boarder Gateway Protocol (BGP). The prefix reachability information is then propagated hop by hop through out the Internet between each pair of neighboring ASes. Upon receiving a propagated paths toward certain prefix, an AS decides whether to use the path and which neighbors it should propagate to.

Based on the AS-level relationship, each AS decides which paths it will take to reach other ASes, and propagates its decisions to its neighboring ASes, the result of which is a fully-connected Internet. Figure 5 shows a simple example of a DDoS attack where the attack traffic can have multiple routes to reach the victim depending on the provider the victim chooses. If the victim chooses provider 1 as its provider, it will announce its prefixes via provider 1, and as a result the rest of the Internet (including DDoS attackers) can reach the victim via provider 1.

3.3.1.2 DDoS Attack Model. We model the DDoS attacks as the following. First, DDoS attacks can originate from any AS $v \in V$ on the Internet and can happen at any time. We define the set of the attack source ASes as

$$SRC = \{v_1^s, v_2^s, .., v_n^s\},$$

where v_i^s is the ith attack source AS, and $SRC \subset V$. A DDoS attacker controls large botnets to launch the attacks, thus the size of the set SRC is usually very large. Second, assume that any AS on the Internet can also be a victim of DDoS attacks.

3.3.1.3 DDoS Defense Model. There are multiple ways to defend against DDoS attacks. The defense can happen at the victim side or any third-party AS that defend for the victim; it can also be carried out by multiple ASes on the paths of the attack traffic. In this study, we allow any AS on the Internet to participate in DDoS defense efforts, assuming that the means of communicating defense details is available to all ASes.

3.3.2 A Game Theoretical Model of Provider Selection. In this subsection, we mathematically describe the likelihood of an AS winning over customers from a game theoretical perspective.

There are two key entities in the model:

– **Customer ASes** who pays for Internet connection to providers but can select which providers to use;

– **Provider ASes** who provides Internet connection as well as DDoS protection service.

The provider ASes who shares potential customers compete with each other for the customers, and the customers chooses their Internet providers independently. We study the worst-case scenario DDoS attacks, i.e. globally-distributed long-term attacks, and consider the attack sources as an input of the model. Peer/sibling ASes do not have influences on the game, and thus are also considered as an input of the model.

3.3.2.1 ***Customer ASes.*** As the Internet infrastructure grows, it is common for customer ASes to have multiple provider ASes to select from. This naturally generates competitions among all the providers ASes. In the context of frequent DDoS attacks, DDoS defense becomes a key criterion for provider selection due to the following reasons. It is a common practice that a provider AS charges a customer AS by the 95% of the peak traffic volume (i.e. 95th percentile bandwidth metering). Frequent DDoS attacks can easily impose significantly higher costs to any downstream customer ASes that have to carry the traffic. A rational customer AS would select the provider AS who provide services with lower cost, with the consideration of the charges involved in forwarding and filtering DDoS traffic by the provider.

3.3.2.2 ***Provider ASes.*** Provider ASes are driven by profit it can make. If a provider faces competition over customers, the provider with a better DDoS defense capability would be more likely to win the competition and carry the traffic for the customers. The losing ASes would then lose all the potential profit

that could have generated from the lost customer ASes. The provider also does not filter traffic for free for customers. In this study, we assume that each provider charges its customers for both forwarding normal traffic and filtering unwanted traffic.

3.3.3 A Game Theoretical Model of Provider Selection. In this subsection, we mathematically describe the likelihood of an AS winning over customers from a game theoretical perspective.

First, let us consider an example where two provider ASes v_1 and v_2 compete for a customer v_c. When under DDoS attack, the customer AS v_c receives total of $T_{c,ddos}$ DDoS traffic and $T_{c,normal}$ normal traffic. Provider v_1 and v_2 each need to make a decision on 1) whether it is providing DDoS filtering service, and 2) how much it would charge for filtering DDoS traffic. A provider charges its customer for forwarding traffic (the rate of which denoted as $r_{forward}$), as well as filtering DDoS traffic if it decides to participate in defense (charges at the rate of r_{filter}). If v_1 decides to participate in defense, it would charge the customer at the rate of

$$r_{1,c} = r_{forward} * T_{c,normal} + r_{filter} * T_{c,ddos},$$

i.e. charge the forwarding of normal traffic and filtering of DDoS traffic separately at different rates. Respectively, if provider v_2 decides not to participate in DDoS defense, it would charge its customer at the rate of

$$r_{2,c} = r_{forward} * (T_{c,normal} + T_{c,ddos}),$$

i.e. charge forwarding of both normal and DDoS traffic at the same forwarding rate.

Based on the charges by its providers (i.e. $r_{1,c}$ and $r_{2,c}$), the customer AS v_c decides v_1 or v_2 as its provider. Naturally, the customer would choose the provider

with lower charge. However, the customer is likely to make errors when evaluating the values $r_{1,c}$ and $r_{2,c}$ due to imperfect estimation of its normal and DDoS traffic. This is called the *bounded rationality* assumption, which is wildly adopted in the recent economics literature to capture the empirical fact that decision makers are not necessarily perfectly rational (see for example in McKelvey and Palfrey (1995) and Goeree, Holt, and Palfrey (2005)). Given the errors and misinformation, the customer is making a probabilistic choice over the two providers instead of a deterministic one. That is, the customer chooses the provider with the higher expected filter rate with a higher probability.

Suppose that the customer's learned charge rate of provider v_1's filtered traffic is given by $r_{1,c} + \epsilon_1$, and that of provider v_2's filtered traffic is given by $r_{2,c} + \epsilon_2$, where ϵ_1 and ϵ_2 are two independent noise terms. A common assumption is that these noise terms are extreme value distributed (usually double exponentially) (see Brock and Durlauf (2001, 2002), Durlauf and Ioannides (2010), Blume, Brock, Durlauf, and Jayaraman (2015), among many others). This assumption will result in that the difference of the two noise terms is logistically distributed, (See McFadden (1973), Anderson, De Palma, and Thisse (1992), Blume (1993), Brock (1993) for discussions on the importance of logistic models in economics.)

$$\text{Prob}(\epsilon_1 - \epsilon_2 \leq x) = \frac{1}{1 + e^{-\lambda x}}$$

, where λ is a parameter that measures the "noisiness" of the two noise terms. As λ increases, the customer's learned values are more precise. As a result, the customer would choose v_1 over v_2 if the customer believes that v_1 charges higher than v_2, and

that translates to

$$r_{1,c} + \epsilon_1 > r_{2,c} + \epsilon_2$$

From this we can derive the probability of v_1 winning the customer as

$$Prob_1(c) = \frac{e^{\lambda r_{1,c}}}{e^{\lambda r_{1,c}} + e^{\lambda r_{2c}}}$$

This calculation can be extended to multiple provider ASes scenarios. We define the set of all potential provider ASes of customer v_c as P_c. The probability of a potential provider $v_i \in P_c$ become the provider is

$$Prob_i(c) \quad = \quad \frac{e^{\lambda r_{i,c}}}{\sum_{v_j \in P_c} e^{\lambda r_{j,c}}} \tag{3.1}$$

For every customer AS v_c, we have

$$sum_{i \in SP_i} Prob_i(c) = 1$$

. Equation (3.1) allows us to calculate the chance of an AS provider winning the right to carry a customer's traffic.

3.3.4 Profit calculation. We assume that each provider AS tries to maximize its profit by taking the competition we described above into consideration.

As an example, let us consider two ASes: v_p and v_c. i_p is a potential provider of v_c, and v_c will make a decision on whether to use v_p as its provider based on the estimated charge $r_{p,c}$. The total amount of normal traffic that v_c needs a provider to carry is given by $T_{c,normal}$, and the total amount of DDoS traffic that is faced by v_c is given by $T_{c,ddos}$. If v_c is chosen to be the provider (winning the competition), The expected profit of v_p made from v_c is thus given by

$$Profit_{p,c} = Prob_p(c) \times r_{p,c}$$

, which equals the probability that v_p is chosen by v_c as the provider times the rate the provider v_p charges. Note that increasing charge rate $r_{p,c}$ has two opposing effects: decreasing v_p's probability of winning the competition due to higher charge, but also increasing v_p's profit on handling the traffic for v_c if selected to be the provider.

The total expected profit of AS_i is the sum of all profits it can get from its customers,

$$Profit_p = \sum_{c \in C_p} Profit_{p,c}$$

, where C_p is the set of all potential customer ASes of v_p.

3.3.5 Cost of defense. Deploying DDoS defense also incurs costs to a provider AS. We define the total defense cost function for an AS as

$$Cost_i = Cost_{equip}(T_{i,filter}) + Cost_{labor}(|C_i|),$$

where $Cost_{equip}(T_{i,filter})$ is the equipment cost for filtering the DDoS traffic for a traffic rate of $T_{i,filter}$, and $Cost_{labor}(|C_i|)$ is the labor cost for maintaining all its customers.

3.3.6 Assumptions. We make a few assumptions in this study. In the rest of this section, we will describe the assumptions and the retionales behind them.

Provider ASes have similar performance features: We assume provider ASes have similar performance features, and the only differentiator during the competition is the decison on whether to particiate in in-network filtering, and if so the rate that it would charge for filtering corresponding DDoS traffic. In this study we focus on the economics aspects of the competition where customers prefer providers with lower charges under DDoS attacks. Other performance metrics such

as bandwidth provided and peering locations are ignored will be considered in the future study.

Providers try to maximize their profit: During the competition, we also assume that each provider tries to maximize their profit, and does not try tactics that might cause loss of profit to gain customers. In the future study, we may include more complex models to capture other actions a provider might take to outperform its competitors.

Costs of in-network fitlering patiticipation are the same for all providers: For simplicity, we also assume the equipment and labor cost functions are the same across all ASes. We assume that C_{equip} increases as the overall traffic to filter increases, and C_{labor} increases when an provider AS has more customer ASes.

3.4 Simulation Design

In the previous section, we have described our modeling of the incentives of ASes on deploying DDoS defense systems. To study the real-world indication of the model, we design a simulation system that allows us to simulate the DDoS defense decisions of ASes on the Internet and explore the outcomes. Designing and implementing a simulation system with more than 60,000 interconnected entities is not a trivial task. In this section, we describe our design of the simulation system and explain how the system can help us explore the defense decision outcomes.

3.4.1 Simulation Setup.

3.4.1.1 Customer-Provider Pairs.

The simulation relies on the full Internet topology to study all possible interactions among the provider ASes and their potential customers on the Internet. The topology include not only the current links (relationships) between ASes, but also all possible/potential relationships that might be established. Ideally, we would like to know: for each

AS, what neighboring (directly connected) ASes does it have and could have, including the relationships with them. However, the relationship information is considered private and often concealed by ASes, even for the current established ones. To best estimate the current and potential relationships between ASes, we use CAIDA AS relationship data (CAIDA (2019); Luckie, Huffaker, Dhamdhere, Giotsas, et al. (2013)) of the past 10 years to compile a relatively comprehensive inter-connection information for the whole Internet.

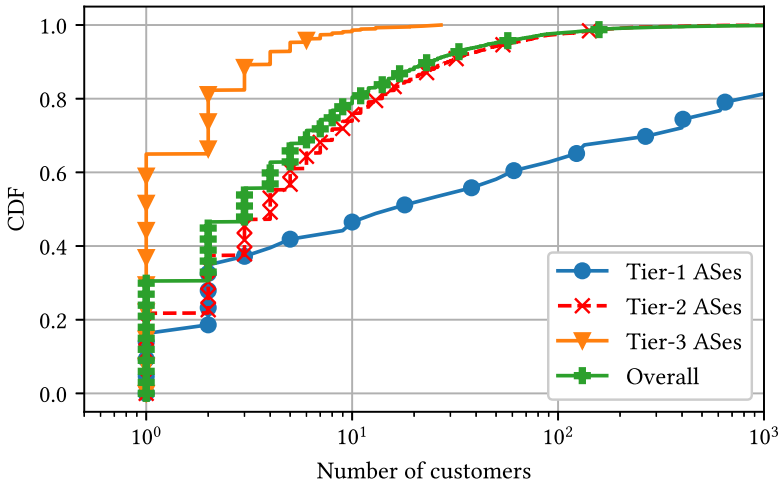
Figure 6 shows the distribution of providers ASes of different tiers with different number of competitors and customers. From both figures we can see that over 80% of the providers have less than 10 competitors/customers, while over 95% providers have less than 100 competitors/customers. We can also see that Tier-3, Tier-2, and Tier-1 ASes have increasingly more customers and competitors. The severity of competitions could potentially affect the filtering participation decisions for providers, and we will study the effects using dynamic simulation in Section 3.5.3.3.

3.4.1.2 Traffic Estimation.

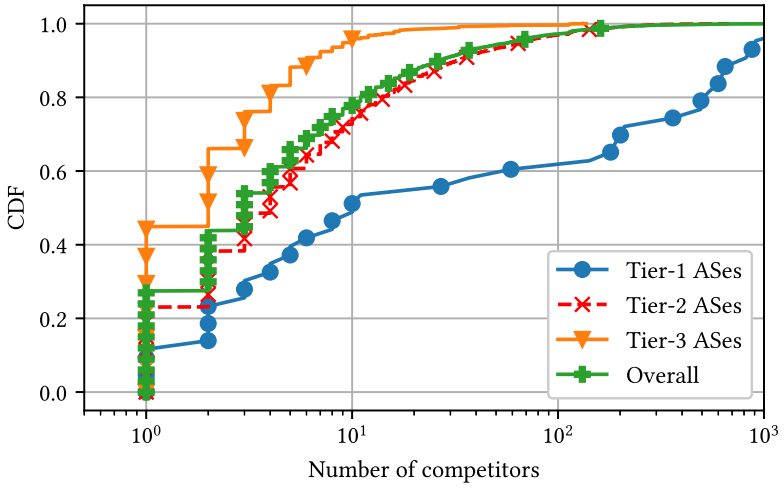
We also use full routing tables from all collectors from RIPE RIS (RIPE RIS (2019)) and RouteViews (University of Oregon (2019)) to construct a best-effort Internet topology. The connected peer routers are considered traffic originators in the simulation for both DDoS traffic and normal traffic. Using that, we estimate the relative amount of traffic for each AS, and use that as base unit for profit calculation (see Section 3.3).

3.4.1.3 Provider AS's Action Options.

Each provider AS has multiple action options when optimizing its profit. One can choose not to participate in defense, and thus charge forwarding both normal and DDoS traffic the same rate. One can also choose to participate in DDoS defense, and charge



(a) CDF of number of customers.



(b) CDF of number of competitors.

Figure 6. CDF of number of customers and competitors for provider ASes in the dataset.

Table 2. Simulation parameters and their value ranges.

parameter	meaning	value range
`ddos_ratio`	Ratio of volume between DDoS and normal traffic	0.0 – 1.0
`do_defense`	Whether an AS participate in DDoS attack	`true` / `false`
`filter_charge`	If defend, how much it charges for filtering traffic	0.0 – 1.0

forwarding normal traffic at regular rate, and charge filtering of DDoS traffic at a fraction of the regular rate, ranging from 0.0 to 1.0. When the filtering charge rate at 0.0, it indicates that the AS provide DDoS defense service free of charge; when the filtering charge rate at 1.0, the provider charges filtering traffic as much as forwarding it; any filtering charge beyond 1.0 would make filtering more expensive than forwarding for the customer, and we do not consider those cases in this study. Table 2 summarizes the parameters and their corresponding value ranges.

3.4.1.4 ***DDoS Traffic Ratio.*** Another important factor that could affect the providers' defense decisions is severity of the DDoS attacks their customers are seeing. We define the term $ddos\_ratio$ as the ratio DDoS attack traffic as compared to normal legitimate traffic to/from a customer AS. In this study, we range the DDoS ratio from 0.5, i.e. the volume of DDoS traffic is half of the normal traffic, to 5.0, i.e. the DDoS traffic is five times as large as the normal traffic.

3.4.2 Static Simulation. In a static simulation, each provider AS makes a decision on defense based on its competitors initial states. Once the decision is made, no more adjustment is considered. Specifically, we study the profit changes for each provider AS on the Internet when it switches from not defending to defending. We consider two scenarios in this study: first AS to participate in

Algorithm 1 Static simulation algorithm.

Require: $V = \{v_i | v_i \ is \ an \ AS\}$ ▷ set of all ASes
Require: $A = \{a_i | a_i \ is \ an \ defense \ option\}$ ▷ set of defense options
Require: $D = \theta$ ▷ set of defense decisions

```
1: procedure StaticSim(V, A, D)
2:     # first-one-to-deploy scenario
3:     for v_i ∈ V do
4:         # calculate options
5:         for a_k ∈ A do
6:             for v_j is a customer of v_i do
7:                 set competitors to no defense
8:                 calculate probability Prob_i(j, a_k)
9:             end for
10:            calculate profit ∑ Profit_{i,j}
11:        end for
12:    end for
13:    # last-one-to-deploy scenario
14:    for v_i ∈ V do
15:        # calculate options
16:        for a_k ∈ A do
17:            for v_j is a customer of v_i do
18:                set competitors to defend at fixed charge
19:                calculate probability Prob_i(j, a_k)
20:            end for
21:            calculate profit ∑ Profit_{i,j}
22:        end for
23:    end for
24: end procedure
```

defense among competitors (or *first-one-to-deploy*); last AS to participate in defense among competitors (or *last-one-to-deploy*). The first scenario reveals how likely the defense solutions can be adopted in the early stage; while the second scenario reveals how defense decisions can drive competitors defense decision. For each provider AS, we exhaust all defense options (from no-defense to defense at different charge rates), and calculate the expected profit if the provider in question option to one of the option under the two scenarios. in each scenario, we also examines the

Algorithm 2 Dynamic simulation algorithm.

Require: $V = \{v_i | v_i \ is \ an \ AS\}$ ▷ set of all ASes
Require: $A = \{a_i | a_i \ is \ an \ defense \ option\}$ ▷ set of defense options
Require: $D = \{a_i | a_i \in A \ and \ v_i \in V\}$ ▷ set of defense decisions
Require: $D' == \theta$ ▷ set of previous decisions
Require: $isConverged$ =`false` ▷ convergence indicator

procedure DYNAMICSYM(V, A, D, D')
  while !isConverged **do**
    `# update defense decisions`
    for $v_i \in V$ **do**
      `# calculate options`
      for v_j is a customer of v_i **do**
        for $a_k \in A$ **do**
          `calculate probability` $Prob_i(j, a_k)$
        end for
      end for
      `# find best action`
      $D_i = a_k \ where \ \max_{a_k \in A} Profit_i$
    end for
    `# test convergence`
    if $D == D'$ **then**
      $isConverged =$ `true`
    else
      $D' \leftarrow D$
      $isConverged =$ `false`
    end if
  end while
end procedure

total number of providers who can at certain charge rate gain profits comparing to not participating in defense.

3.4.3 Dynamic Simulation. The simulation runs multiple rounds of decision making for all ASes. In each round, every AS will choose a defense effort that can optimize its profit based on the knowledge of its competitors' defense efforts from the previous round. We call such a decision rule used by the ASes, the *myopic best response rule*, because each AS chooses the optimal effort level without incorporating its competitors' strategic changes.

Given a configuration, we want to use the simulation to find the Nash equilibrium of the game. The equilibrium (or convergence) state represents the status where there is no individual players would like to change its effort level unilaterally. The detailed steps are as follows.

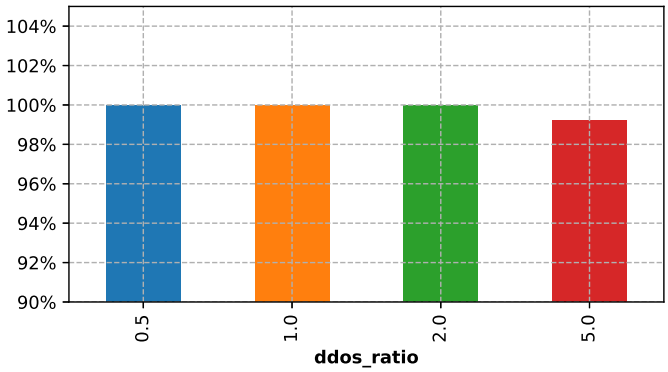
We initiate all ASes by setting their defense preference to no-defense. Based on equation 3.1, each provider AS calculates its probability of being selected by its potential customers for each action choice it has. Using the customer-winning probabilities, the simulation system re-calculate the expected profit of a provider AS given that the AS would choose the option that can maximize the winning probability. After all ASes have updated their decisions, the simulation determines if it has converges, i.e. if any AS have made difference decisions comparing to the previous round. If the simulation has not converged, it will continue the previous procedure and update defense efforts for all ASes. If the simulation has converged, it will then produce the report of the final states for all ASes. See Algorithm 2 for more details.

3.5 Simulation Results

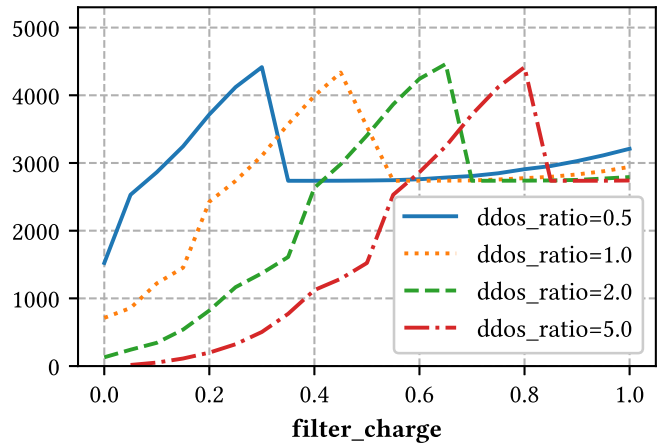
In this section, we will discuss the simulation results. As discussed in 3.4, we divide the simulation into two types:

- *static simulation*, where only the AS under study can change its defense configuration, while other ASes, especially competitors, stays static;
- *dynamic simulation*, where every AS can change its configuration to pursue higher profit.

The static simulation shows a snapshot of ASes' responses under fixed configuration, while the dynamic simulation shows how ASes may update their



(a) Percentage of providers can gain profit.

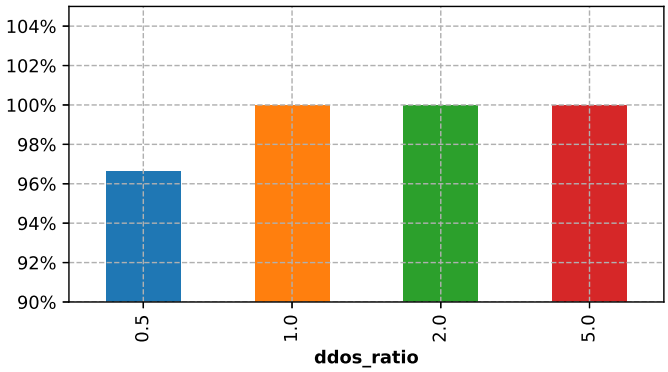


(b) Number of profitable providers.

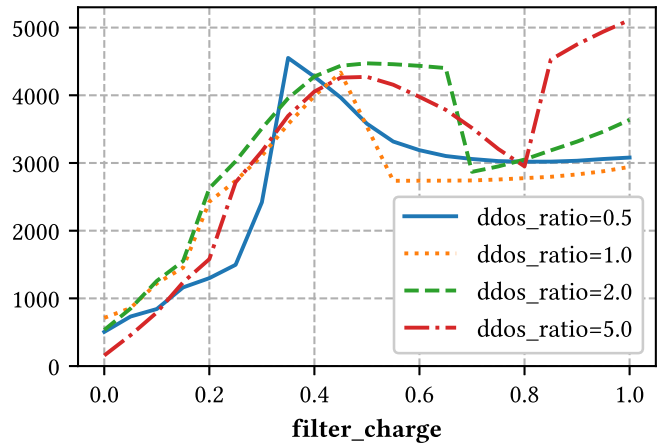
Figure 7. Profitable *first-one-to-deploy* providers.

defense choices and whether the simulation achieves equilibrium. We study these scenarios also with different DDoS-to-normal traffic ratio ($ddos\_ratio$), ranging from 0.5 (i.e. DDoS traffic's volume is half of the normal traffic) to 5.0 (i.e. DDoS traffic is five times as large as normal traffic). Each provider AS who decides to defend also can select different charge rate for processing and filtering DDoS traffic ($filter\_charge$).

3.5.1 Static Simulation. We first examine the incentives of provider ASes participating in in-network DDoS defense. Specifically, we study the profit



(a) Percentage of providers can gain profit.



(b) Number of profitable providers at different charge rate.

Figure 8. Profitable *last-one-to-deploy* providers.

changes for each provider AS on the Internet when it switches from not defending to defending. We consider two scenarios in this study: first AS to participate in defense among competitors (or *first-one-to-deploy*); last AS to participate in defense among competitors (or *last-one-to-deploy*). The first scenario reveals how likely the defense solutions can be adopted in the early stage; while the second scenario reveals how defense decisions can drive competitors defense decision.

Figure 7 shows the results for the *first-one-to-deploy* scenario. We first examine how many providers can make positive profit gain when they switch

from no-defense to defense. Figure 7a shows the percentage of all provider ASes that can gain profit by switching to defense at certain filter charge. It indicates that almost all provider ASes can at least make extra profit at some filter charge. Figure 7b further reveals at what charge does most of the provider ASes can gain profit by switching. It clearly shows that the number of profitable providers peak at different filter charge; and as the DDoS becomes more severe, the peak of filter charge increases.

Figure 8 shows the results for the *last-one-to-deploy* scenario. Figure 7a shows that almost all provider ASes can gain profit at some filter charge, but there are less profitable providers when DDoS volume is low (i.e. $ddos\_ratio = 0.5$). We further study at what charge does most of the provider ASes can gain profit by switching when all competitors are defending with 0.5 filter charge. Figure 7b shows that most of the providers can gain profit when it charges slightly less than the competitors (i.e. 0.4 as opposed to 0.5). Different from the previous scenario, it also shows that the severity of DDoS attacks (i.e. $ddos\_ratio$) does not significantly affect the profitability of an AS when its competitors all defend.

From both results, we can see that when provided freedom to compensate DDoS defense costs by charging for filtering efforts, the majority of the providers on the Internet can find some charge rate that allow the them to gain profits by providing filtering services. If a provider is the first to provide services, the amount of DDoS traffic ratio affects how much it should charge to maximize profits; on the other hand, if a provider joins defense the last, no matter how severe the DDoS attacks are, the profitability is significantly decided by its competitor's choice. In most cases, charging similarly or slightly less comparing to the competitor would result in the best profits for the majority of the providers. In other words, charging

too much would risk losing the customers altogether, while charging too little would make the provider miss large portion of the profit that it could make.

3.5.2 Individual Provider Profit Patterns. The previous study focused on overall statistics of the providers who can gain profit by switching to participating in defense. We also study how each individual AS's profit may change when different $filter\_charge$ is selected.

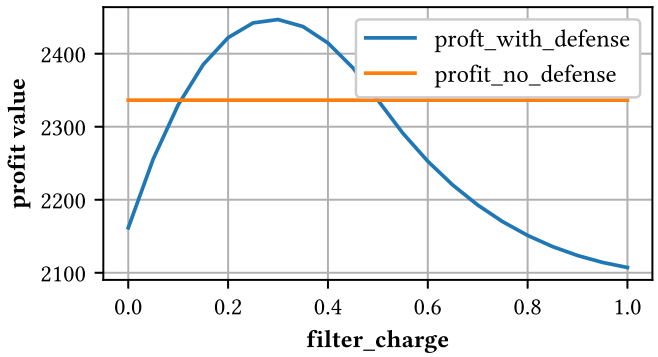
Figure 9 and Figure 10 shows four different types of profit patterns:

- bell-shape profit curve with gain;
- bell-shape profit curve without gain;
- increasing profit curve;
- decreasing profit curve.

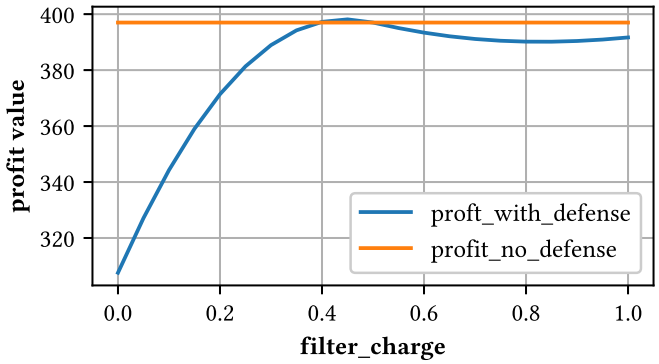
We will discuss these types of profit patterns and their indications in this section.

3.5.2.1 Bell-shape profit curve with gain. Figure 9a shows AS37468's profit value as the $filter\_charge$ changes when it participates in defense. Comparing to the profit baseline when it does not participate in defense, the bell-shaped profit value curve exceed the baseline between 0.1 and 0.5 and peaks at 0.3. The AS procures more profit by increasing the filtering charge when the charge is low (< 0.3), and less profit when the charge is higher (> 0.3). This shows a clear example of diminishing returns Samuelson and Nordhaus (2001), and indicates that the AS has incentive to participate in defense when the charge is set properly.

3.5.2.2 Bell-shape profit curve without gain. Figure 9b shows an similar profit pattern with diminishing returns, but with the peak of the profit when defending lower than the baseline. This figure indicates that the AS in



(a) Bell-shape profit curve (AS37468, 1.0 DDoS ratio).



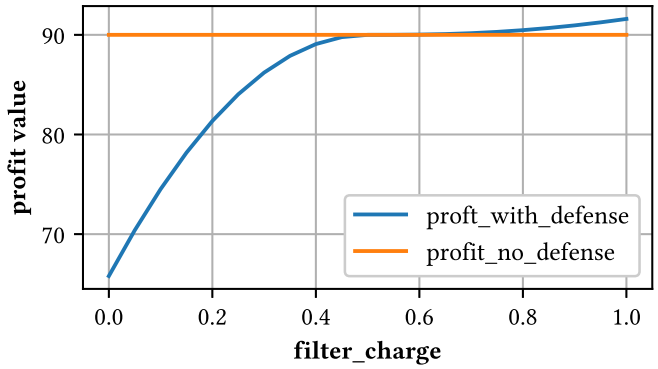
(b) Bell-shape profit curve but unprofitable (AS25227, 1.0 DDoS ratio).

Figure 9. Number of providers gains profit by switching from not-defending to defending.

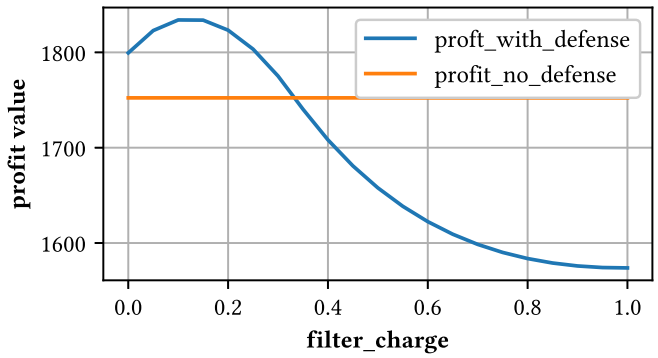
question cannot make enough profit to justify switching to DDoS defense regardless of the $filter\_charge$ choices.

3.5.2.3 Increasing profit curve.

The increasing profit curve (Figure 10a) indicate that the ASes has yet to reach their peak profit even they charges at 1.0 rate. This pattern shows that these ASes face less competitions that compete by charge prices (such as provider ASes that has customers that have no other potential providers).



(a) Increasing profit curve (AS46455, 1.0 DDoS ratio).

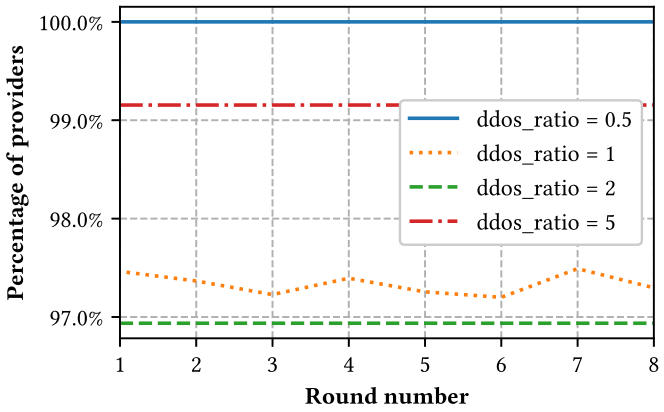


(b) Decreasing profit curve (AS37468, 0.5 DDoS ratio).

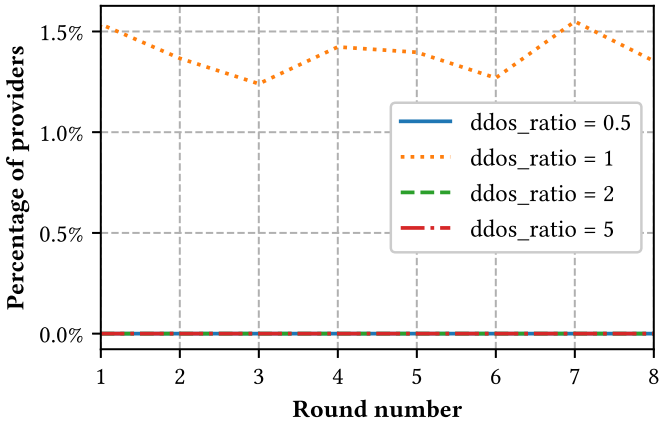
Figure 10. Number of providers gains profit by switching from not-defending to defending.

3.5.2.4 Decreasing profit curve. The decreasing profit curve (Figure 10b, after 0.1), on the other hand, indicate that the ASes has passed their peak profit at low or almost no charge for filtering DDoS traffic. Such ASes tend to have heavy competitions where charge high prices for defense would significantly decrease chances of being selected by its potential customers.

3.5.3 Dynamic Simulation. We further study the decision of provider ASes regarding DDoS defense in a more dynamic environment. In this section, we examine the results for dynamic simulation, where every AS is making



(a) Percentage of provider ASes participate in filtering.



(b) Percentage of provider ASes update charges.

Figure 11. Profitable providers when *no competitors* participate in defense.

decision dynamically based on their competitors' decisions, and the procedure repeats until the decisions converge.

3.5.3.1 Provider AS's choices. At each round, a provider AS can in general decide whether it would participate in DDoS defense or not based on its overall profit calculation. When deciding its options, it calculates the profit *as if*

– it does not participate in defense;

– or it participate in defense and charge $filter\_charge$ amount for processing and filtering DDoS traffic.

The rate ranges between 0.0 to 1.0 with 0.1 as step. The AS will then select the best option that maximize its profit based on calculation introduced in Sec. 3.3.4. Note that its competitors' current configuration (i.e. the results of their previous decision) is incorporated during the profit calculation, and its decision will then also affects its competitors future decision making.

3.5.3.2 Percentage of providers defending. We first examine the number of provider ASes decided to participate in defense, given that each provider AS is try to maximize their profit at each round. Figure 11 shows the summary results for dynamic simulation in terms of provider participation and configuration changes. Figure 11a show the percentage of provider ASes decided to participate in defense at each round of the simulation. It is clear that very high number of provider ASes decided to participate in the defense at the very first round, and the numbers for different DDoS attack scenarios stay high and stable. This indicate that in terms of defense participation, the simulation converges very fast and result in high-level of participation for all provider ASes.

With almost all ASes participate in defense, do they alternate their defense charges? Figure 11b shows that there is very little number of ASes update their $filter\_charge$ configuration, and only appear in one of the simulation where $ddos\_ratio = 1$. Combining this result with results from Figure 11a, we can conclude that given opportunity to freely change and optimize their defense decisions, **the majority of the provider ASes would choose to defense and settle down on their** $filter\_charge$ **rates**.

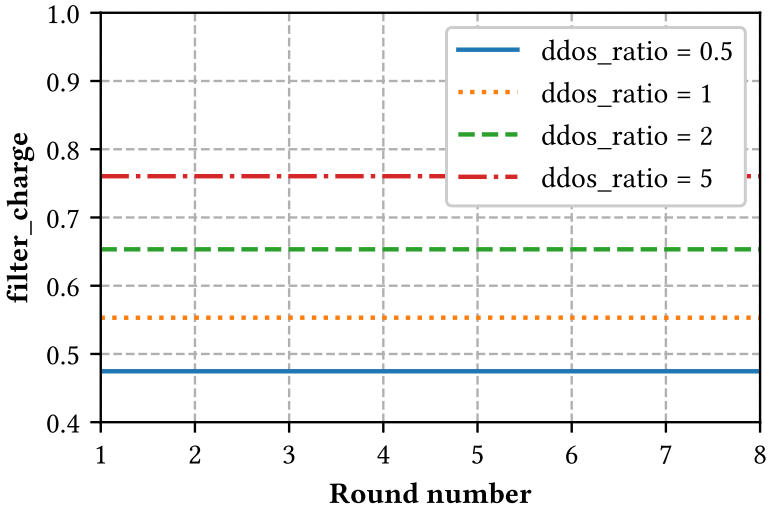
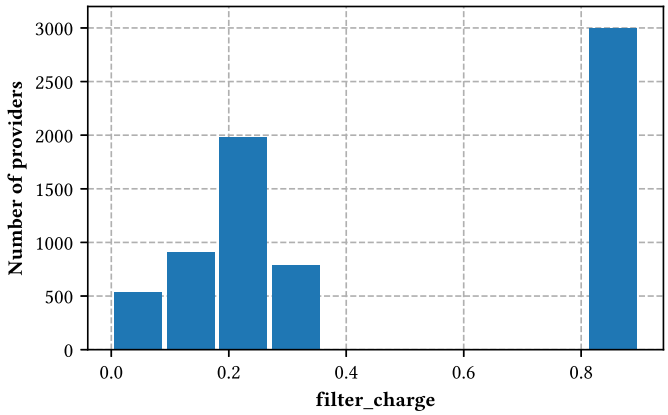


Figure 12. Average charges.

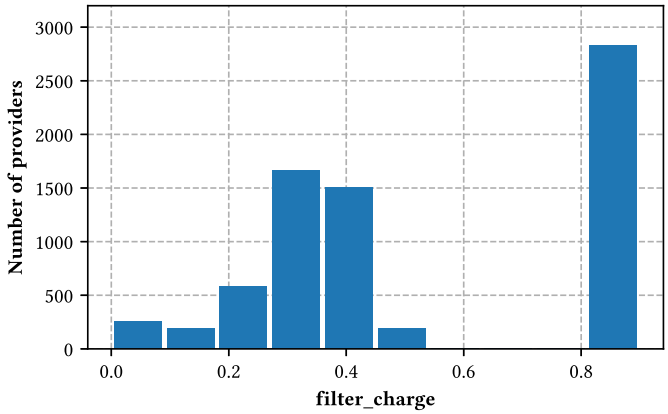
3.5.3.3 Filter charges. Since the majority of the provider ASes would choose to defense and settle down on $filter\_charge$, the next question becomes how much would they charge their customers to maximize their profit? To answer this question, we further dig into the simulation results and examine each individual ASes' optimal charge and the overall distribution of the charge values.

Figure 12 shows the average $filter\_charge$ for all provider ASes that decide to participate in defense under different DDoS traffic ratios. When DDoS attack traffic is relatively low (i.e. < 0.5), the average charge for filtering traffic is around 0.5, meaning a provider AS charges its customer about half of the price for filtering DDoS traffic than forwarding normal traffic. As the DDoS traffic volume increases, the charge also increases to as high as about 0.8 when DDoS attack traffic is five times stronger than normal traffic.

We further take a look at the distribution of the $filter\_charge$ for all provider ASes under different severity of the DDoS attacks. Figure 13 and Figure 14 show the histograms of the number of provider ASes with different



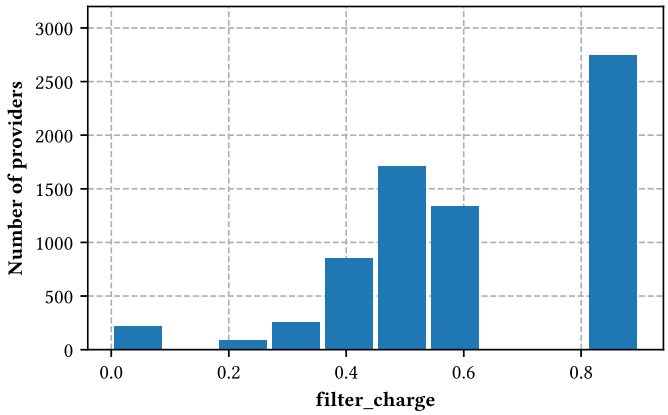
(a) DDoS ratio = 0.5



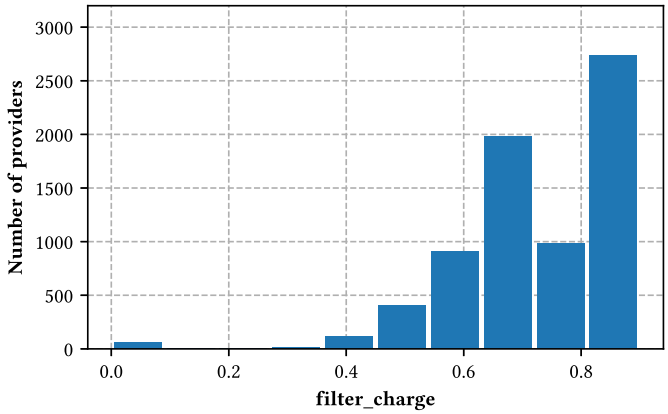
(b) DDoS ratio = 1.0

Figure 13. Provider ASes charge distribution.

$filter\_charge$ when the simulation converges. As we can see from the DDoS ratio increases from 0.5 (Fig.13a), 1.0 (Fig.13b), 2.0 (Fig.14a), to 5.0 (Fig.14b), there are 1) a group of ASes with lower charges that move to higher charges (which also form a bell-shape in the figures), and 2) a consistent number of ASes (around 2,500 to 3,000) that always charges highest rate possible. To understand why there are two groups of ASes that have different charge patterns, we examine every AS in each group to reveal their inner correlations. Specifically, we use Fig.13a as an example, and divide all provider ASes into two groups by their $filter\_charge$: low



(a) DDoS ratio = 2.0



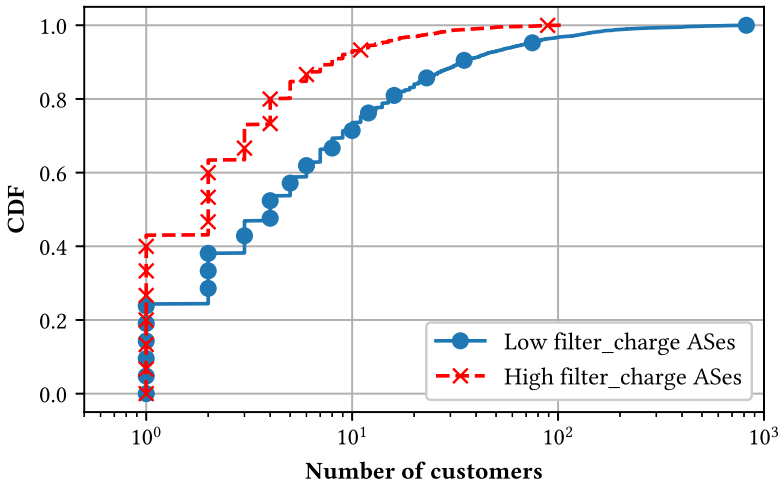
(b) DDoS ratio = 5.0

Figure 14. Provider ASes charge distribution.

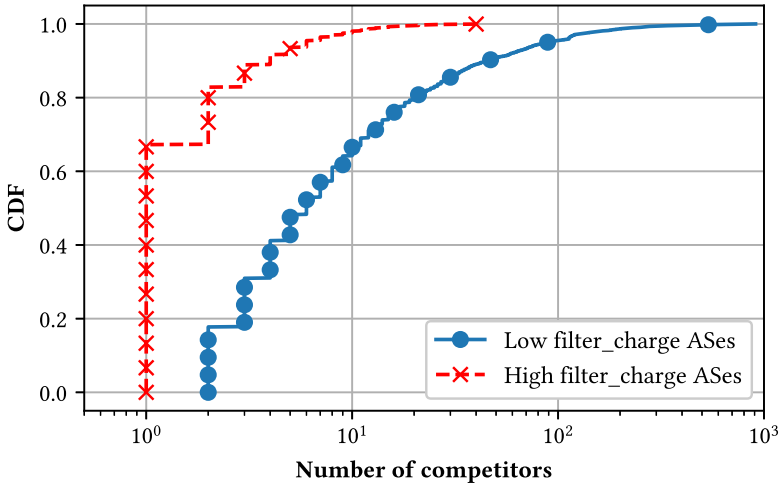
$filter\_charge$ group where $filter\_charge < 0.5$ and high $filter\_charge$ group where $filter\_charge \geq 0.5$.

Figure 15 shows the CDF plots of the number of customers and competitors for each group. It is clear that ASes with higher $filter\_charge$ have less customers (Fig.15a) and less competitors (Fig.15b).

3.5.4 Summary. In this section, we investigate the incentive of provider ASes on the Internet participating in DDoS defense by examining the



(a) CDF of number of customers.



(b) CDF of number of competitors.

Figure 15. CDF of number of customers and competitors for provider ASes.

profit each provider can make under different environment while the customer ASes can freely choose the provider it select to use.

From the static simulation, we learned that most provider ASes can gain profit by provider DDoS defense service to potential customers; providers reach their peak expected profit with different charges for filtering traffic, which is impacted by the severity of the DDoS attacks as well as their competitors' defense decisions.

Further dynamic simulation revealed that for most provider ASes, if they choose to charge DDoS filtering that maximize their profit (assuming no competitors are providing similar services), they all can reach their stable peak profit and achieve a global stable status.

We also discovered that the number of competitors/customers have a strong impact on how much they should charge for DDoS filtering to reach peak profit. An AS can gain higher profit by charging more for DDoS filtering if it has weak competitions; while an AS with strong competitions need to charge less for DDoS filtering in order to win customers and gain higher profit.

3.6 Conclusion

In this chapter, we propose a game-theoretical model that examines the incentives of ASes to invest in efforts on DDoS defense. Based on the model, we built a large-scale simulation system and examine 1) whether networks on the Internet can be incentivized to participate in in-network traffic filtering, and 2) the affects of a network's topological location, level of competition, and the amount of DDoS traffic it carries for its customer affects its decision on DDoS filtering efforts.

We observe the following patterns from the simulation results. The majority of the provider ASes on the Internet can benefit from providing DDoS defense

services to their customers if they can compensate the defense cost by charging for filtering DDoS traffic. The severity of DDoS attacks affects the charge rate a provider can place on its potential customers; if a provider sees higher volume of DDoS traffic going through its potential customers, it would charge higher to achieve its peak profit. The level of competition also drives the charge rate: a provider with low-level competition can charge a high rate while still profitable; a provider that faces strong competitions need to charge less to attract customers for profit.

These observations provide confidence that if in-network collaborative defense mechanisms mature enough provider ASes on the Internet would have incentive to participate in DDoS defense. We believe that such observations can further help researchers to develop better strategies to devise and deploy DDoS defense solutions.

CHAPTER IV

FILTER PLACEMENT STRATEGIES FOR IN-NETWORK DDOS TRAFFIC FILTERING

From Chapter III, we have learned that the majority of the networks on the Internet can be incentivized to participate in in-network traffic filtering. With confidence of the deployment incentives, we can assume larger number of networks on the Internet can be utilized for in-network filtering purpose. Now the question becomes: *Given a set of filtering rules, in what locations should a defender place the rules to achieve the maximum effectiveness on DDoS traffic filtering?*

In this chapter, we survey the existing strategies for placing the rules, summarize the existing solutions by developing a model that can describe all strategies in the same framework, propose a new strategy that can outperform the existing ones, and finally evaluate their performances using simulations based on real-world Internet topology and DDoS attack traces.

This chapter is directly derived from Zhang, Shi, et al. (2019), resulted from collaboration with other co-authors listed in Zhang, Shi, et al. (2019). Mingwei Zhang is the primary author of this work, including co-designing and analyzing the different in-network DDoS filtering strategies.

4.1 Overview

Research has shown that distributed denial-of-service (DDoS) attacks on the Internet could often be better handled by enlisting the *in-network* defense of multiple autonomous systems (ASes), rather than relying entirely on the victim's Internet Service Provider at the edge. Less noticed but important is the fact that an in-network defense can also remove DDoS traffic from the Internet early *en route* to the victim, thus decreasing the overall load on the Internet and reducing

chances of link congestion. However, it is not well understood to what degree different in-network defense strategies can achieve such benefits. In this chapter, we model the existing two main categories of in-network DDoS defense algorithms (PushBack, SourceEnd) and propose a new type of algorithm (StrategicPoints). In particular, we compare their effectiveness in minimizing the amount of DDoS traffic that the victim receives, their impact on reducing the DDoS traffic on the entire Internet, and their resiliency against intelligent adversaries and dynamic attacks. We detail how the comparison results vary according to parameters and provide our insights on the pros and cons of these three categories of in-network DDoS defense solutions.

4.2 Background

Researchers have put forward a number of *in-network DDoS defense* solutions to handle very-large-scale DDoS attacks that happen increasingly more frequently on the current Internet. Instead of defending at the edge, they defend against DDoS attacks before the traffic reaches the victim, often when the DDoS traffic is even further away from the victim's network. These solutions often also distribute the defending workload among a set of defensive collaborators, each in charge of a portion of the traffic. Collectively, the set of collaborators are able to handle a larger volume of DDoS traffic than any individual collaborator.

4.2.1 Lacks of Quantitative Comparisons. Although many in-network DDoS defense solutions have been proposed, it still remains difficult for a victim to select suitable defense solutions against specific DDoS attacks. The in-network defense solutions (such as DefCOM (Oikonomou et al. (2006)), PushBack (Mahajan et al. (2002b)), and MiddlePolice (Z. Liu et al. (2016))) vary greatly in resource requirements, training data needed, and expected efficiency. Selecting a

sub-optimal defense solution could introduce substantial cost and even result in unsuccessful defense. However, there is *no quantitative study on how the solutions compare to each other*, *nor a general model that describes these solutions in a common language*. Further, it is also unknown how these solutions perform under insufficient knowledge of the attacks or against intelligent adversaries who can dynamically revise their attack strategies to escape defense. Without a quantitative comparison, it is hard for a DDoS victim to select the most suitable solution to achieve its defense goal and meet the resource requirements.

4.2.2 Existing Filtering Strategies. Almost all in-network defense solutions require placement strategies to decide where on the Internet to deploy traffic filters or defense measure. We categorize existing in-network defense strategies into three categories: *PushBack*, *SourceEnd*, and *other* (shown in Table 1). We will briefly recap the two types strategies here. For interested readers, Chapter II contains for more details and information on solutions with no clear filter placement strategies.

4.2.2.1 PushBack strategy. : We define PushBack defense algorithms as those that start defense from a victim AS and expand the defense area to its upstream ASes. Starting from the victim AS, PushBack allows each defending AS to mitigate a portion of the attack traffic, and delegate the rest of the attack traffic to its upstream ASes for further mitigation. The original PushBack style defense propagation is introduced in the *PushBack* paper (Mahajan et al. (2002b)). Although this work considers router-level defense propagation, the basic idea can be applied to AS-level collaboration. Other distributed defense systems that stem from the classic PushBack paper, such as the work of Yau et al. (Yau et al. (2005)) and

ScoreForCore (Kalkan and Alagöz (2016)), follow the same PushBack algorithm to defend against DDoS attacks.

4.2.2.2 ***SourceEnd strategy.*** : Different from a PushBack algorithm where the defense initiated from the victim side, a SourceEnd algorithm attempts to select the ASes that are the sources of the attack or close to the sources, and only fall back on downstream ASes toward the victim if resources run out. D-WARD system (Mirković et al. (2002)), for example, installs rate-limiting rules at border routers in source networks; COSSACK (Papadopoulos et al. (2003)) deploys countermeasures at the ASes of attacking sources. Both are early works that employ the SourceEnd strategy. Later work such as AITF (Argyraki and Cheriton (2005)) introduced the idea of propagating the defense from the attacking sources to the victim, thereby providing more flexibility for defense deployment. Specifically, authors of AITF also observed that the current generation of routers have sufficient filtering resources to mitigate DDoS attacks as long as the attack traffic was blocked close to the attacking sources. Furthermore, AITF was also one of the earliest projects to study hardware rule space during defense. Later work from Huici et al. (Huici and Handley (2007)) and StopIt (X. Liu et al. (2008)) enhance AITF by introducing security measures against DDoS attacks on the defensive infrastructure itself. Unlike PushBack solutions, prior to defending against the attack all SourceEnd solutions need to know the attack topology (i.e. the attack sources and their AS-level routes toward the victim) for each DDoS victim. However, obtaining this information is not trivial.

4.3 Modeling DDoS Attacks and Defenses

In order to evaluate the performance of different types of in-network defense algorithms, we first construct a model to describe DDoS attacks and their defenses.

In this section, we introduce our general model that describes the Internet, DDoS attacks, and in-network DDoS defense.

4.3.1 Modeling the Internet. The Internet is a well-interconnected network consisting of thousands of autonomous systems (ASes). ASes on the Internet can be represented by the set

$$\mathbb{N} = \{n | n \; is \; an \; AS \; on \; the \; Internet\}.$$

The size of $\mathbb{N}$ is 63332 at the time of writing (Huston, Smith, and Bates (n.d.)). The links or edges between ASes can be represented by the set $\mathbb{L} = \{l_i | i = 1, 2, ..., q\}$, where q is the total number of links on the Internet. The traffic running on the Internet can be summarized into $flow$s, where each flow f represent a set of packets between the *source* of the flow (denoted as $f.src$) and the *destination* of the flow (denoted as $f.dst$) for a transaction. In this study, we focus on using IP addresses to identify an entity involved in a flow. Each flow is also associated with a volume value, denoted as $|f|$, which can be represented by the number of packets or the number of bytes included in the flow.

4.3.2 Modeling DDoS Attacks. The total set of attack sources is represented by

$$\mathbb{A} = \{a | a \in \mathbb{N} \; and \; a \; is \; an \; attack \; source\},$$

where a represents an attack source in a DDoS attack. Similarly, we define the set of victim end-hosts as

$$\mathbb{V} = \{v | v \in \mathbb{N} \; and \; v \; is \; a \; victim \; of \; the \; DDoS \; attack\}.$$

Here, one attacking source represents a machine that is controlled by the attacker and creates unwanted traffic to the victims. Each attacking source can generate multiple flows with varying volume to a victim at any moment during an attack.

The set of total attacking flows is represented as

$$\mathbb{F} = \{f | f.src \in \mathbb{A} \; and \; f.dst \in \mathbb{V}\},$$

where f is the attacking flow, generated by an attack source, sent to a victim. While a DDoS attack can potentially strike multiple targets, for simplicity, the rest of the paper focuses only on a single victim. Each flow traverses through a set of ASes on the Internet before reaching the victim. We denote the number of AS links a flow f must travel through to reach its victim as b_f.

4.3.3 Modeling In-Network DDoS Defense. Different from single-AS edge defense solutions, in-network defenses employ multiple ASes en route of the DDoS attack traffic to filter unwanted traffic. For each defense solution, we denote the set of all ASes that are able to participate on defense, or *defense pool*, as $\mathbb{D}$ and $\mathbb{D} \subset \mathbb{N}$. However, not all ASes in $\mathbb{D}$ will be used by the defense solution. The set

$$\mathbb{S} = \{n | n \in \mathbb{D} \; and \; n \; is \; selected \; for \; defense\}$$

contains all ASes in the network that are not only able to collaborate on defense, but also selected to filter traffic during the defense. The defense algorithms decides which ASes should be utilized for the defense against specific attacks, which takes the following elements into consideration. Finally, we denote

$$\mathbb{L} = \{f | f \in \mathbb{F} \; and \; f \; is \; filtered\},$$

for each defense solution.

4.3.3.1 Resource for Defense. Each defending AS has limited resources in filtering DDoS traffic. Specifically, the main resource limitation is on the number of filtering rules an AS can employ for DDoS defense purposes. We define $Rmax$ as the maximum number of filtering rules that can be installed at

an AS. The defense may also face a limitation on the number of total ASes it can use, which we denote as $Dmax$. $Rmax$ reflects the resource limitation at the *intra-AS* level, while $Dmax$ reflects the limitation at the *inter-AS* level. In practice, the defense algorithms should take both the resource limitation $Rmax$ and scale limitation $Dmax$ into consideration when initiating defenses.

4.3.3.2 Leakage and Pollution. To quantify the effectiveness of a solution, we look at two main metrics: *leakage* and *pollution*; *leakage* represents the total amount of attack traffic that reaches the victim after the defense is in place, and *pollution* represents the total amount of attack traffic that flows through the Internet before it is filtered. The *leakage* metric shows the defense's effectiveness from the victim's perspective, while *pollution* reveals the defense's impact on reducing the overall DDoS traffic on the Internet. DDoS attacks can cause link congestion, therefore a DDoS defense algorithm should not only achieve very low *leakage* to reduce the attack traffic the victim receives, it also needs to further reduce the traffic on the paths toward the victim (*pollution*) to avoid chance of link congestion.

We define *leakage* as

$$leakage = |\mathbb{F}| - \sum_{k \in \mathbb{S}} m_k,$$

where m_k is the total number of attack flows mitigated by AS k in the defending set $\mathbb{S}$. Note that at which AS the filtering happens does not affect the *leakage* metric.

We define *pollution* as

$$pollution = |\mathbb{F}| - |\mathbb{L}|.$$

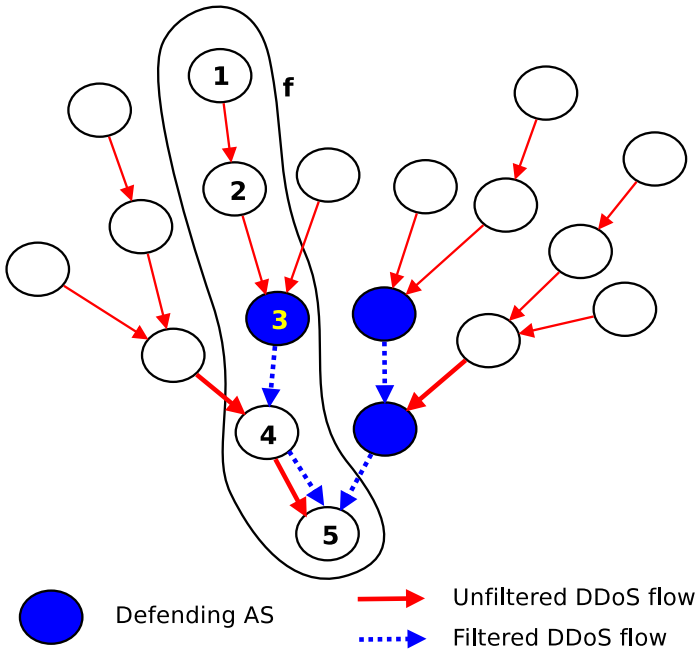


Figure 16. Example of calculating the *pollution* for one attack flow f and an defense measures. The pollution flow f remains to have after defense is 2, between $AS1$ and $AS3$.

The value of *pollution* for a defense measures how well a defense strategy does on limiting the amount of traffic running on the Internet. Clearly, $c_f = 0$ when only the victim's AS stops an attack flow f. If an attack flow f is stopped at the source AS, the value c_f becomes $c_f = b_f$. Figure 16 shows an example of counting the *pollution* for one flow. The attack flow f, of magnitude $|f| = 1$, travels through the AS path 1-2-3-4-5, and AS3 deploys DDoS defense and filters f. In this example, suppose that AS 3 effectively filters the attack flow, the total length of f is $b_f = 4$, and there are two links after AS3 that will not see f, which means $c_f = 2$. Filtered at AS3, f contributes $b_f - c_f = 4 - 2 = 2$ to the overall *pollution* metric. From this example, we can see that the closer a flow is filtered to the attack source, the less it contributes to the overall *pollution*.

Table 3. Notations used to describe the general models in this section.

Symbol	Definition
$\mathbb{N}$	Set of all ASes on the Internet
$\mathbb{A}$	Set of all DDoS attack sources
$\mathbb{F}$	Set of all attacking traffic flows
$\mathbb{V}$	Set of victim end-hosts
$\mathbb{D}$	Set of ASes able to participate in defense
$\mathbb{S}$	Set of ASes utilized in defense
$\mathbb{L}$	Set of filtered DDoS attacking traffic flows
$Rmax$	Maximum number of attack flows an AS can handle
$Dmax$	Maximum number of ASes available for defense
$\|f\|$	Volume of traffic carried by flow f
b_f	Number of AS links between a flow f and its victim
m_k	Number of attack flows mitigated by an AS k
c_f	Number of ASes on the path of flow f after it is filtered
$leakage$	Total amount of attack traffic reach the victim
$pollution$	Total amount of attack traffic on the Internet

4.4 In-network DDoS Traffic Filtering Strategies

In general, the in-network DDoS traffic filtering strategies can be summarized into two major types: **PushBack** strategy that focuses on placing the defense close to the victim network; and **SourceEnd** strategy that distribute defense load among networks close to the attack sources. These types of strategies have been used in various DDoS defense projects (see Chapter II for more). However, there is no quantitative study on how the solutions compare to each other, nor a general model that describes these solutions in a common language. Therefore, we compare these strategies and study their strengths and weaknesses. Furthermore, we propose a new strategy called StrategicPoints strategy which employs ASes at critical locations of the attacks to achieve the high effectiveness with low cost. In order to compare these strategies, we generalize each strategy and study each one individually. In the rest of this section, we will describe the three strategies using the model introduced in Section refsec:modeling.modeling.

Algorithm 3 PushBack Strategy

Require: P ▷ Pool of participating ASes
Require: v ▷ Victim AS
Require: $\mathbb{F}$ ▷ Attack flows

```
1: procedure PushBack Strategy(v, P, F)
2:     P = {v}; B = φ
3:     for v_i ∈ P do
4:         add v_i to B
5:         remove flows v_i.filtered_flows() from F
6:         add all upstream ASes of v_i to P
7:         remove v_i from P
8:         if |F| == 0 or |P| == 0 then
9:             break
10:        end if
11:        sorts ASes in P by the number of flows they can filter
12:    end for
13:    return (B)
14: end procedure
```

4.4.1 PushBack Strategy.

The PushBack strategy propagates the defense workload from the victim AS to upstream ASes. The PushBack strategy expands the defensive area from the victim AS to its upstream neighbors one AS at a time, and further upstream if necessary. PushBack essentially distributes the DDoS defense load (i.e. the deployment of traffic filtering rules) among the set of collaborating upstream ASes. PushBack can be applied recursively, allowing it to cover a larger set of ASes if necessary.

The PushBack strategy runs recursively among the collaborating ASes starting from the victim AS. We assume that all collaborating ASes exchange information about any ongoing defense efforts, and the PushBack strategy knows the filtering status of the attack traffic. The strategy begins by selecting the ASes that are nearest to the victim as defending ASes. In each round, each defending AS installs a number of rules to filter a portion of the attack traffic running through it. If more traffic needs to be filtered, the PushBack strategy will select the next best

AS from the *defense pool* to collaborate on defense, where a defense pool consists of the next available upstream ASes of all current defending ASes, sorted by number of flows they can filter. PushBack selects the AS that can filter the most DDoS traffic at each round of defense propagation. The defense propagation ends when there are no available ASes in the *defense pool*, the number of defending ASes exceed the victim's specified parameter $Dmax$, or all traffic has been successfully handled.

Algorithm 4 SourceEnd Strategy

Require: P ▷ Pool of participating ASes
Require: v ▷ Victim AS
Require: $\mathbb{F}$ ▷ Attack flows

procedure SourceEnd Strategy($v, P, \mathbb{F}$)
    $P = \{v\}; B = \phi$
    for $f \in \mathbb{F}$ **do**
        $n =$ the first collaborating AS on the path of f
        $P = P \cup \{n\}$
    end for
    while $|S| > Dmax$ and $P \neq \phi$ **do**
        sout P based on amount of traffic it can filter
        simulate_filter($P[0]$)
        $B = B \cup \{P[0]\}$
        $P = P - \{P[0]\}$
    end while
    return (B)
end procedure

4.4.2 SourceEnd Strategy. In contrast to the PushBack strategy, the SourceEnd strategy attempts to select ASes that originate the attack traffic for defense, thereby stopping the attack directly at the sources. The SourceEnd strategy intuitively performs better in terms of reducing the overall attack traffic on the Internet (*pollution*). However, it requires more participating ASes and traffic filters to be effective at mitigating the attacks.

Ideally, the SourceEnd strategy should utilize all collaborating ASes that are the closest to the attacking sources. However, facing the maximum available ASes constraint $Dmax$, the SourceEnd strategy will prioritize collaborating ASes and only select the ASes that can filter the most DDoS traffic. We describe the strategy from high level as follows. First, SourceEnd locates the initial defense locations as potential defending ASes, i.e. the ASes that are closest to the attackers. Then it sorts all potential ASes by the amount of traffic each AS can filter. It then adds the top AS to the selected ASes list (S), and removes it from the potential ASes list. The strategy will repeat the previous two steps until $|S| \geq Dmax$, or all flows can be filtered, or there are no ASes available.

Algorithm 5 StrategicPoints Strategy

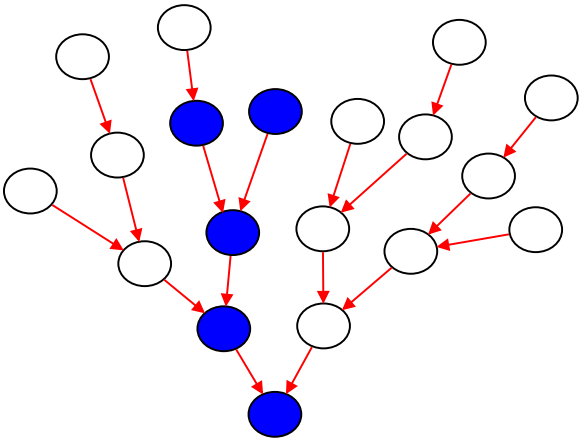
Require: $Dmax$ ▷ Maximum allowed ASes to use
Require: v ▷ Victim AS

```
1: procedure StrategicPoints Strategy(v, Dmax)
2:     B = {v}
3:     while B ≠ D and |B| <= Dmax do
4:         sort B based on the traffic each AS carries
5:         for x ∈ B do
6:             for y ∈ D that is an upstream of x do
7:                 B = B ∪ {y}
8:             end for
9:             if x contains no attack sources then
10:                B = B − {x}
11:            end if
12:            if B = D or |B| > Dmax then
13:                break
14:            end if
15:        end for
16:    end while
17:    return (B)
18: end procedure
```

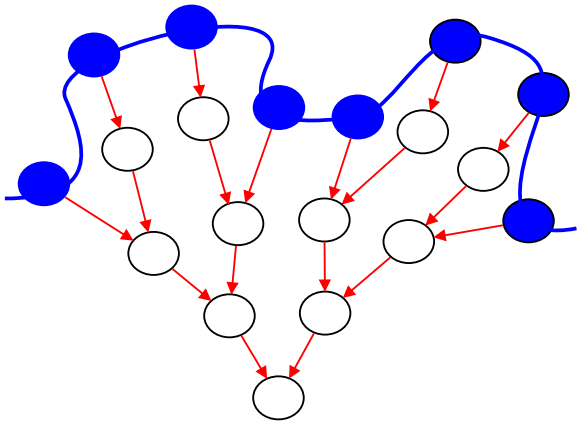
4.4.3 StrategicPoints Strategy. The StrategicPoints strategy, different from PushBack and SourceEnd strategies, employs ASes based on both

traffic and topological information, and tries to deploy defenses at strategic locations inside the network instead of at edges. Although PushBack strategies can utilize a small number of ASes close to the victim to cover most of the DDoS traffic, it suffers from potential heavy pollution. Similarly, while SourceEnd strategy can have minimal pollution on the Internet, it requires a large number of participating ASes to cover the source ASes. The StrategicPoints strategy, selects set of ASes that sits on all attack AS paths, as far into the Internet as possible, to achieve a balance between PushBack and SourceEnd, and target high effective defense with low pollution and low leakage.

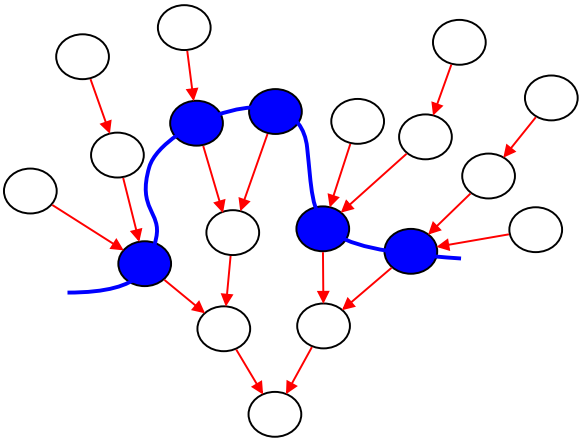
The basic idea of StrategicPoints is to find the ASes that are in strategically important locations in terms of forwarding attack traffic to the victim. Here, we believe the most critical ASes are the participating ASes that 1) observe the most traffic; 2) together consist of a topological cut for all the attack traffic toward the victim; and 3) are closer to the sources if possible. The strategy first collects the statistics of the attack traffic distribution among the ASes. It begins by adding the victim AS to the set of selected defending ASes, B. Then, it builds up B by continuously replacing ASes in this set with their direct upstream ASes until there are no more available ASes in the defense pool or $Dmax$ is surpassed. At each step, we prioritize the ASes by the amount of attack traffic they received, thus pushing the line of defense from the heavily impacted ASes first until all attack traffic is filtered. By doing so, the strategy maintains a set of selected ASes that together consist of a cut of all attack traffic paths toward the victim, and in the meantime, also moves the defense further toward the attack sources. With sorted selection at each step, the strategy also balances the defense workload (the amount of traffic an AS needs to filter) among the defending ASes.



(a) PushBack



(b) SourceEnd



(c) StrategicPoints

Figure 17. Examples of three in-network DDoS defense strategies.

4.4.4 Summary. To summarize, traditional *PushBack* and *SourceEnd* strategies work by deploying traffic filtering rules directly at or close to the victim or the attack sources correspondingly. *StrategicPoints* strategy, on the other hand, selects critical ASes hops away from the victim that carry the most of the attack traffic and cover all critical paths. It is able to push the defense far to sources for the heavily congested links, and maintaining close-to-victim defending ASes for links that are not congested at the moment. This feature further allows the StrategicPoints strategy to better handle dynamic attacks with shifting attack sources without over-fitting towards one attack pattern at any given moment. Figure 17 shows an high-level example of how each strategy works.

4.5 Evaluation Setup

To quantitatively investigate different DDoS defense strategies, we built a simulation framework that simulates the Internet, DDoS attacks, and in-network DDoS defense. The simulation follows the model described in Section 4.3 and implements the DDoS filtering strategies described in Section 4.4, with the following details.

4.5.1 Internet Topology. In this study, we aim to explore the performance and cost of the in-network filtering strategies under extreme stress, i.e. very large-scale DDoS attacks. To simulate large-scale DDoS attacks and defenses, the first step is to construct the Internet topology. We use the full routing table dump data obtained from RouteViews (University of Oregon (2019)) in August 1st 2018 to build the Internet topology. A full routing table contains the AS-level paths toward all the reachable IP prefixes, which reflects the full Internet topology from the perspective of a route collector. By combining the topologies obtained from all 22 RouteViews collectors, we aim to cover as many AS-level links as possible.

Table 4. DDoS attack traces used in simulation.

Trace name	# of sources	# of source ASes
CAIDA-2007 (Hick et al. (2007))	∼4,700	∼1,400
Merit-2016 (Merit Network (2016))	∼2,300	∼1,300

4.5.2 Large-Scale DDoS Attacks. We use two real-world large-scale DDoS attack traces for this study (Table 6). One real-world attack trace is a DDoS attack collected by CAIDA in 2007 (Hick, Aben, Claffy, and Polterock (2007)), and the other is a DDoS attack toward an RADB service collected by Merit in 2016 (Merit Network (2016)). Both traces involve thousands of attack sources originated from thousands of ASes and can be used to evaluate DDoS defense under real conditions.

We use the route collectors in RouteViews (University of Oregon (2019)) as the victims of DDoS attacks. We also assume that the AS-level paths are symmetric, i.e. the AS path from a victim to an attack source is the same as the AS path from the attack source to the victim, allowing us to build attack flows using the AS paths in the routing tables. Note as this assumption is not always true on the Internet, our simulation thus derive results from a constructed topology that is similar to but not exactly the same as a real Internet topology.

4.5.3 In-network DDoS Defenses. In this simulation we assume all ASes are able to participate in the defense and a DDoS filtering strategies chooses the actual defense ASes. The simulation begins with the scenario of no defense. Then we apply the strategies introduced in Section 4.4 to decide ASes to be used for defending against the attack. For each simulation run, we vary the $Dmax$ and $Rmax$ restriction, i.e. the maximum number of ASes available for defense, and the maximum number of defense rules available at each AS (for simplicity this is the

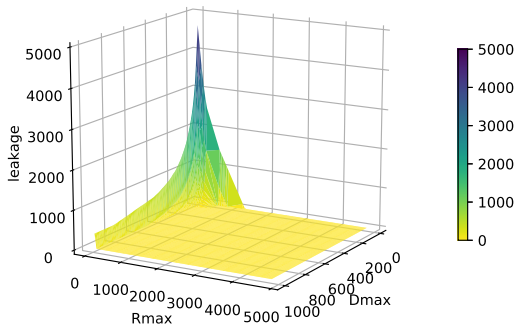
same across all ASes). At the end of each run, the simulation framework collects the simulation results, including values of the metrics introduced in Sec. 4.3): *leakage* and *pollution*.

4.6 Evaluation

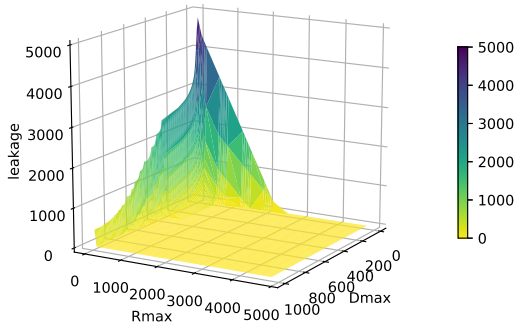
We compared different DDoS defense algorithms in terms of their resource requirements, time to respond to attacks, and resiliency against the intelligent DDoS attackers. Below we report and analyze the results. Refer to Section 4.5.2 for the datasets we use in the evaluation.

4.6.1 *Leakage*. First and foremost, DDoS victims care most about the amount of DDoS traffic still leaking towards them after defense is deployed on the Internet. As we previously defined, we measure *leakage* under different resource constraints to evaluate each algorithm's effectiveness in filtering out the attack flows toward the victim. Specifically, for each algorithm, we run simulations for various combinations of $Rmax$ (i.e. the maximum number of filtering rules each AS has) and $Dmax$ (i.e. the maximum number of defending ASes the defense can utilize).

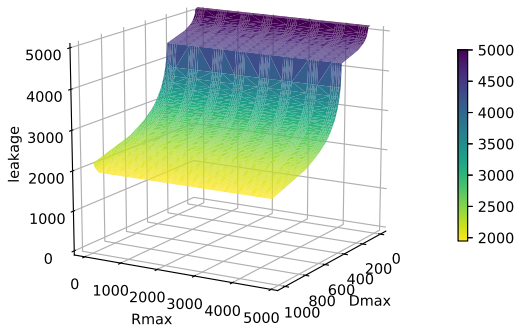
Figure 18 shows the *leakage* simulation results for three algorithms using the CAIDA-2007 dataset, with $Rmax$ ranging from 10 to 5000 (about maximum number of sources of the whole attack) and $Dmax$ ranging from 10 to 1000. It is clear at within the range of the resource limitations in the simulation, both PushBack and StrategicPoints outperform SourceEnd on reducing *leakage*. StrategicPoints performs slightly worse than PushBack when resource limitations are low, and both algorithms perform equally well when resource limitations are high.



(a) PushBack



(b) StrategicPoints



(c) SourceEnd

Figure 18. Resource requirement for reducing *leakage*.

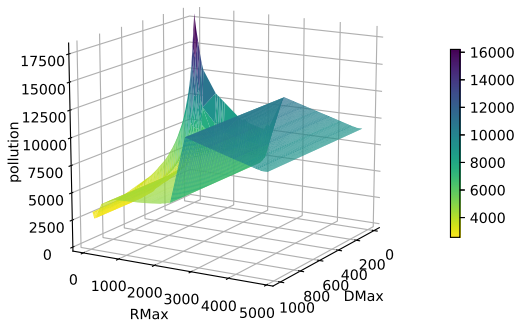
4.6.2 *Pollution.* Since both PushBack and StrategicPoints perform similarly in reducing *leakage*, we further examine their performance in reducing the overall *pollution* on the Internet.

Figure 19 shows simulation results for measuring the *pollution* for the three algorithms. It is clear that as *Rmax* increases, the *pollution* reductions by PushBack become less effective. In fact, given a high enough *Rmax* PushBack utilizes only the victim AS to defend against the attack, leaving a large portion of *pollution* unhandled. This is undesirable when the attack *pollution* is so high that it could not only cause extra burden on the ISPs to forward traffic, but also trigger traffic congestion on the links close to the victim. On the contrary, StrategicPoints although has similar effectiveness in reducing *leakage*, it also greatly reduces the *pollution* caused by the attack on the Internet, thus further reducing the chances of link congestion.

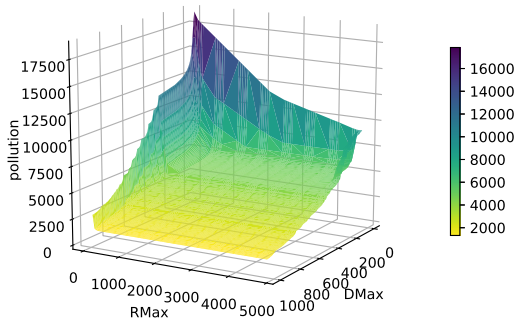
To show the comparison more clearly, we also fix *Rmax* while increasing *Dmax*. As shown in Figure 20, both PushBack and StrategicPoints perform well in reducing *leakage*, while StrategicPoints outperforms PushBack and reduces *pollution* significantly.

4.6.3 Effectiveness against dynamic DDoS attack. One other important metric for an effective DDoS defense algorithm is how effective it is at handling DDoS attacks with dynamic attack sources. Specifically, we want to study how the algorithms perform on a real-world attack trace where there is consistently more attack sources joining and leaving the attack.

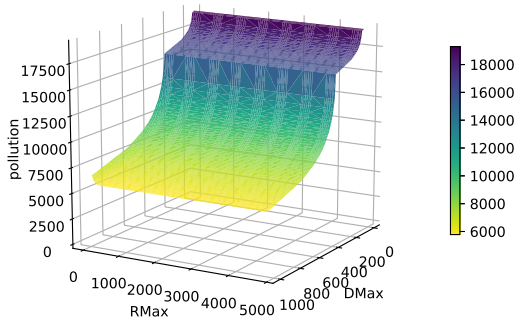
We run three algorithms against the Merit-2016 dataset, using the first 2000 flows (about 15% of the total unique sources) as the training set for locating defending ASes, and then use the same set of ASes for the result of the attack.



(a) PushBack



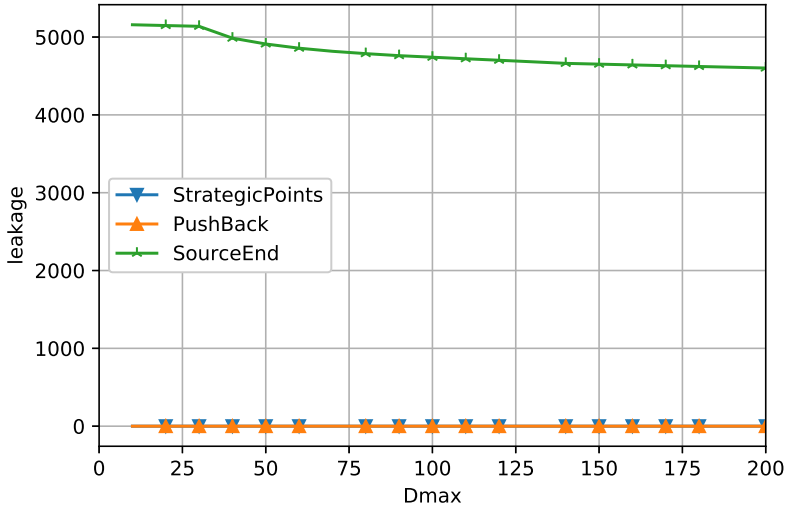
(b) StrategicPoints



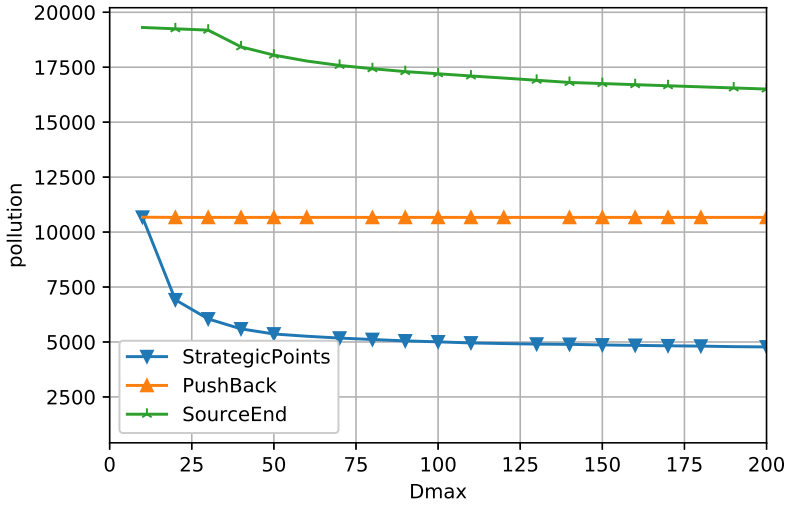
(c) SourceEnd

Figure 19. Resource requirement for reducing *pollution*.

As the new unique attack sources join the attack, the defending ASes' rules space will be gradually filled, and eventually will not be able to handle any more new



(a) *leakage*

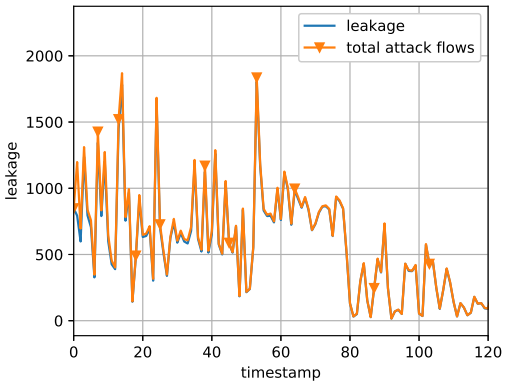


(b) *pollution*

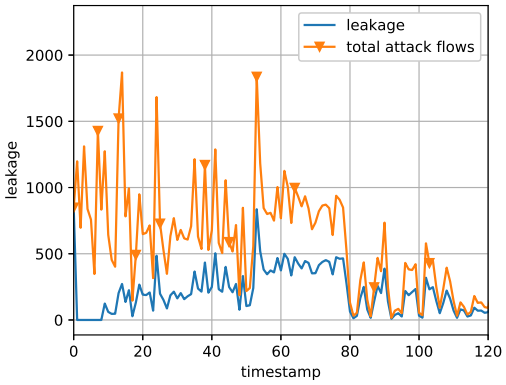
Figure 20. Resource consumption with $Rmax = 3000$

attack flows. Results with larger number of flows shows similar results, and thus are omitted.

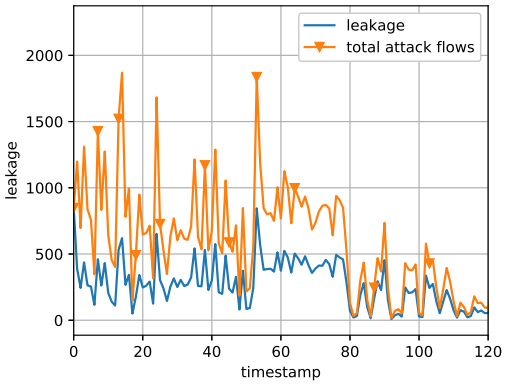
Figure 21 shows that PushBack is very ineffective in dealing with dynamic attack scenarios because it selects ASes that are just able to handle the training flows, allocating no space for new flows and changes of the attacks. StrategicPoints



(a) PushBack



(b) StrategicPoints



(c) SourceEnd

Figure 21. Leakage plot for Merit-2016 trace with strategies using 2000 sources in the first second to locate defending ASes, with both $Dmax = 500$ and $Rmax = 500$.

and SourceEnd both perform much better in handling new flows due to the fact

that they both allocate more overall rule space for defense, allowing defenders to have more flexibility in handling dynamic attacks.

metrics	**PushBack**	**SourceEnd**	**StrategicPoints**
leakage	low	high	**low**
pollution	high	medium	**low**
attack resiliency	medium	low	**high**
key resource	Rmax	Dmax	**Dmax**
when to use	low *Dmax* or *Rmax*	*Dmax* $\approx$ total sources	**all other cases**

Table 5. Algorithms performance summary and usage suggestion.

4.6.4 Summary. In this section, we evaluated the performance of the three algorithms using real-world traces, and summarize the key points as follows.

On reducing the *leakage* of a defense, both StrategicPoints and PushBack perform similarly well when using a reasonable amount of resources. When *Dmax* and *Rmax* are low, PushBack performs slightly better. SourceEnd, on the other hand, is only viable when *Dmax* is very large and close to the total amount of attack source ASes.

Reducing *pollution* is also a very important task in that high *pollution* could cause link congestion, which would still directly affect the quality of service for all the traffic toward the victim. On this aspect, StrategicPoints significantly outperforms PushBack due to the algorithm's tendency to deploy rules farther into the Internet thus closer to the sources. When *Dmax* is very large and close to the total number of source ASes, SourceEnd could also achieve low *pollution*.

When facing dynamic attacks where attack sources join and leave during the attack, it is important that the defense algorithm is flexible and allows deployment of new filtering rules. Due to its design, PushBack always selects a number of ASes that are "just enough" for the defense, thus leaving little or no extra rule space for

new filtering rules to be deployed. On the contrary, StrategicPoints and SourceEnd select defending ASes in a greedy approach, and always fully utilize the available $Dmax$. As a result, ASes selected by either algorithms would have more available rule space to spare for potential future defense rules.

Table 5 summarizes the key attributes of the three algorithms. Based on these results, we believe that PushBack is only suitable when $Dmax$ an $Rmax$ are very low; in all other cases, StrategicPoints can perform best in terms of reducing *leakage* and *pollution*.

4.7 Open Issues

In this section, we discuss our decision to not consider flow volume in our evaluation, the correlation between the three defense placement strategies and existing solutions, and the open issue of IP spoofing which will be addressed in future work. Finally, we derive a conclusion on the necessity of multi-AS over single-AS DDoS defense solutions.

Flow volume: In evaluating the three algorithms, we currently consider the flows have the same weight, and volume information is not included. In fact, after examining the volume information for each sources in the two attack traces, we observe similar volume for the sources with no significant differences among them. The reason behind this decision is that the flow volume in both DDoS trace sets follows a uniform distribution. With that said, our simulation framework in fact can incorporate traffic volume information when needed. However, our simulation framework is able to incorporate traffic volume information, and we plan to expand our evaluation on that direction in the future.

Capturing the essence of existing solutions: We summarize the PushBack, SourceEnd and StrategicPoints strategies in order to capture the essence

of the existing multi-AS DDoS defense solutions. The existing solutions may not follow the exact procedures as we defined. However, we believe our evaluations provide important insights into the three major defense placement strategies.

IP spoofing: For any well-designed DDoS defense system, it needs to consider how to protect victims from spoofed traffic. Although we evaluated the three defense algorithms in non-spoof DDoS attack traces, we plan to evaluate them in scenarios with IP spoofing as our future work.

In-network over edge defense solutions: Edge DDoS defense solutions in general are easy to deploy, but costly in operations (i.e., purchasing large bandwidth links or high performance switches). Edge solutions also cannot prevent large-scale volumetric attacks that congest the inbound links of defense networks. For these reasons, we believe in-network DDoS defense solutions are more suitable for defending against large-scale DDoS attacks of the future.

Deployment for evaluation: This study serves as a pilot study for further real-world evaluation. Ideally, our next step is to conduct real-world deployment for evaluation, and collect results from real-world traffic analysis. However, such evaluation presents the following challenges: 1) it is very difficult (if not impossible) to deploy a large-scale study platform to achieve the scale simulated in this study; 2) it also requires generating a large amount of traffic from multiple vantage points and capturing the traffic passing through ASes on each path, which would result in both high hardware and software requirements at the deployment site; We plan on extending our evaluation to a smaller-scale real-world deployment as our next step.

4.8 Summary

In this chapter, we modeled and evaluated different multi-point, in-networking DDoS traffic filtering algorithms.

After defining a general model for describing DDoS attacks and defense, we categorized existing in-network DDoS filtering algorithms into two basic types, i.e. *PushBack* and *SourceEnd*, that cover the majority of the state-of-the-art research and practice on in-network DDoS filtering. We then introduced *StrategicPoints* algorithms that outperform PushBack and SourceEnd algorithms in most cases.

We designed a simulation framework as a common platform to evaluate major large-scale DDoS attack scenarios and defenses, and evaluated and compared the three types of multi-point, in-network DDoS filtering algorithms in terms of their capability in reducing the DDoS traffic leakage to the victim and the pollution to the whole Internet, as well as their resiliency against dynamic DDoS attacks.

With real-world, Internet-scale DDoS attack traces, our evaluation results show that when having a low number of ASes for defense, PushBack performs slightly better than StrategicPoints and significantly better than SourceEnd in terms of reducing DDoS traffic leakage. As the number of available filtering ASes increases, or the number of available rules space increases, StrategicPoints becomes as effective as PushBack on reducing leakage and significantly outperforms PushBack on reducing pollution caused by DDoS attacks. Finally, we summarized the algorithms and suggested what algorithms to adopt for DDoS defense based on the resource availability of the victim.

CHAPTER V

DESIGN AND EVALUATION OF A DDOS-FILTERING RULE PLACEMENT ALGORITHM

From Chapter III, we have learned that the majority of the networks on the Internet can be incentivized to participate in in-network DDoS traffic filtering. With confidence of the deployment incentives, we can assume a large number of networks on the Internet can be utilized for in-network filtering purpose. Further in Chapter IV, we survey, propose, and evaluate different filter placement strategies for in-network traffic filtering. The *StrategicPoints* strategy we propose has the best performance in reducing both leakage and pollution. In this chapter, we design a concrete StrategicPoints-style DDoS-filtering rule placement algorithm, describe how it works, and finally evaluate its performance under an in-network traffic filtering system.

This chapter is derived from part of (Li et al. (2019)) that describes a DDoS-filtering system called DrawBridge that I participated, especially the latter's text related to DrawBridge rule placement design and evaluation for which I am a primary contributor.

5.1 Overview

As distributed denial-of-service (DDoS) attacks continue to pose a severe threat to network services and users, in-network traffic filtering becomes more preferable due the fact that it filters traffic early upstream, and it utilizes collective filtering capacity from multiple networks. However, as discussed in related work chapter (Chapter II Section 2.4), there is still a lack of a concrete study on the design and evaluation of an effective rule placement algorithm that can achieve in-network filtering on the current Internet.

In this chapter, we present our design and evaluation of an efficient rule placement algorithm that can find satisfactory placements for a give set of rules. Specifically, we design a tree-like data structure called H-tree to model the placements of the rules on the DDoS traffic topology toward the victim. Our algorithm aims to maximize the coverage of the DDoS traffic, minimize the rule space needed for deployment, as well as maximize the distance away from the victim's network.

As part of our efforts to design and evaluate an effective in-network rule placement algorithm, we also briefly introduce a DDoS-filtering system called DrawBridge in which our algorithm runs. Using DrawBridge, we can evaluate the efficacy of the rule placement algorithm in a more realistic environment. Specifically, we evaluate the efficacy of our placement algorithm using both simulation and distributed emulation via DrawBridge. We show that our placement algorithm can find effective locations to place filtering rules, and effectively filter DDoS traffic.

5.2 Design of Rule Placement Mechanism

5.2.1 In-network Filtering System Architecture. The rule placement algorithm is run when an in-network traffic filtering system has obtained a set of filtering rules and need to decide where to place the rules for deployment. We will first introduce system architecture of our in-network filtering system, and describe the functionalities of each component in the picture.

Our designed system, called **DrawBridge**, operates as a subscription service and comprises two main types of players: **DrawBridge subscribers** who subscribe to the DrawBridge service and **DrawBridge providers** who are Internet service providers (ISPs) that support the deployment of traffic filtering rules and

other necessary DrawBridge functionalities. We designed DrawBridge to empower DDoS victims to express their traffic filtering needs in the form of DrawBridge rules (i.e. traffic filtering rules), and to enable ASes on the Internet (i.e. DrawBridge providers) to execute these needs and filter DDoS traffic. This filtering process begins with a DrawBridge subscriber generating traffic filtering rules in response to any DDoS traffic it receives. Every rule will not only indicate how to filter DDoS traffic, but may also include associated meta-data about the rule, including the rule's priority, timestamp, and a timeout value to indicate when the rule should expire. The subscriber then sends the rules to its provider, who decides where the rules should be deployed. The provider who receives the filtering rules from its subscriber runs the rule placement algorithm, through which it decides the optimal locations for the deployment of these rules to maximize its efficiency. Figure 22 shows an example of the basic operation of DrawBridge, where the victim is a subscriber of DrawBridge provider AS 1, and AS 1 is a subscriber of AS 2 and AS 3. The victim informs the controller in AS 1 what traffic to filter, which in turn instructs controllers in AS 2 and AS 3, with the rules eventually placed in two routers in AS 2 and AS 3 to filter DDoS traffic.

A DrawBridge provider can also act as a subscriber to other DrawBridge providers. A DrawBridge subscriber can thus be an end-host, a sub-network, an autonomous system (AS), or an ISP. (An ISP can be composed of one or multiple ASes; below we use AS and ISP interchangeably.) Consequently, all the subscriber-provider pairs form the **DrawBridge network** to serve as the messaging mechanism of DrawBridge. A DrawBridge network is essentially an overlay, where each node is a subscriber or a provider and each link is a subscriber-provider link.

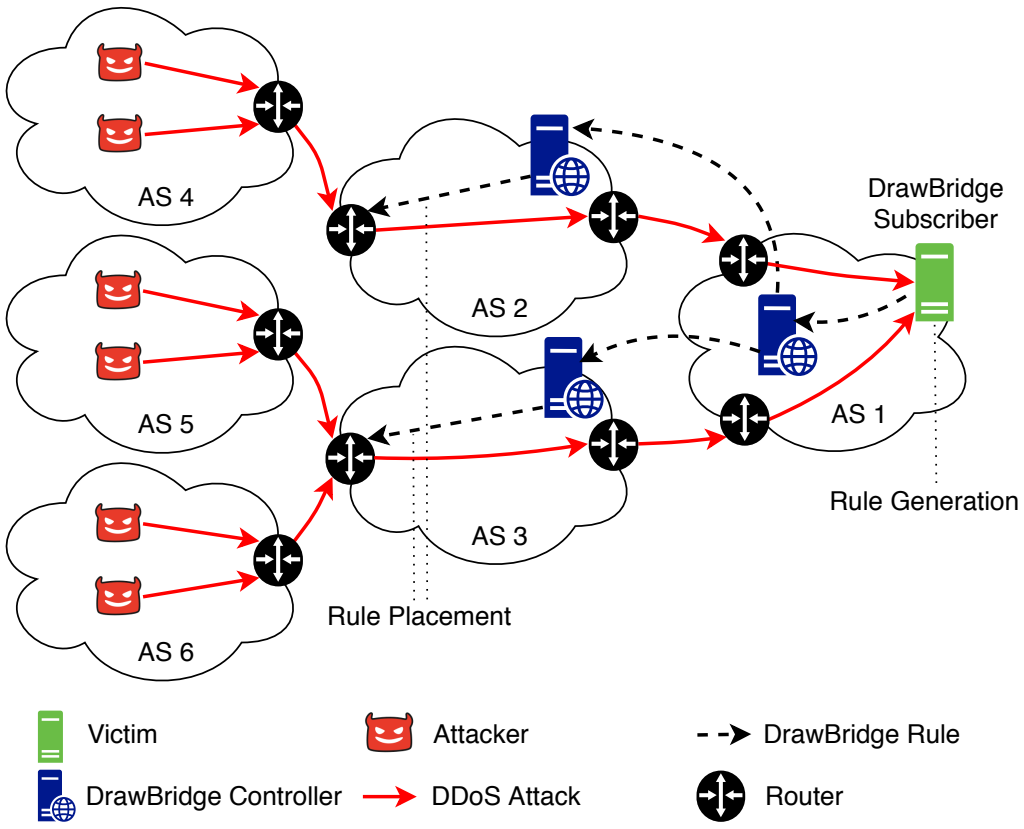


Figure 22. An example of DrawBridge filtering DDoS traffic.

5.2.2 Processing and Placing Filtering Rules.

Once a provider receives a set of rules from a subscriber, it will verify the authenticity of the rules, determine whether and where the rules should be deployed, forward rules to their selected placement locations, and send acknowledgments to the subscriber. The key task here is **rule placement**, which is to select the appropriate locations to place the rules while considering the limited rule space at every AS or switch. We call the AS or the switch that deploys a rule a host of the rule. When selecting the host for a rule, a provider can consider not only itself, but also any DrawBridge provider that may see the DDoS traffic matching the rule. If the provider subscribes to other DrawBridge providers, it can deploy rules remotely (i.e. inter-AS rule placement). After discovering which other DrawBridge-capable ASes can and should deploy the rule in question, the provider then selects an AS and sends the rule to the selected AS, who may either deploy the rule at one or more local switches, or decline to deploy the rule at all. If a suitable inter-AS deployment location is unavailable, and assuming the provider is capable of effectively deploying the rule at one of its own switches (i.e. intra-AS deployment), it merely finds the switch(es) on the path from the attackers to the victim, selects switch(es) closest to the attacking source, and finally deploys the rule at the selected switches.

DrawBridge selects the hosts for a rule as follows. First of all, each filtering rule should be placed at hosts that are on the paths taken by the DDoS traffic matching the rule in question, with as many paths covered as possible (e.g. in Figures 23(b) and 23(c) both paths of DDoS traffic are covered). In the case where no collaborating ASes see traffic matching the filtering rule, the system must place the rule at the victim's AS. On the other hand, if multiple collaborating ASes can capture the corresponding traffic, i.e. there exist multiple DrawBridge-enabled

ASes along the AS path from the source to the victim, filtering rules should be deployed as close to the traffic source (or as far from the victim AS) as possible, thereby limiting the aggregation of the attack traffic before it reaches downstream ASes. Moreover, for each of these paths, there may exist multiple host candidates; if so, the rule should be deployed as close to the traffic source—or as far from the subscriber—as possible, since the volume of DDoS traffic may become too large to handle at a host closer to the subscriber (Figure 23(b)). On the other hand, if two paths converge at an intersection point, it may be better to deploy the rule at a host that is located at or downstream from the intersection, thus only deploying the rule at one host rather than two in order to conserve rule space (Figure 23(c)). Finally, it is possible that a suitable host may not even exist for a given rule, for example, when there are no host candidates on a path of the DDoS traffic, or when the selected host on the path is unavailable due to the lack of space for new rules, further limiting the choices for rule placement (Figure 23(d)). We describe the algorithm for rule placement in Section 5.3.

In order to increase the efficacy of the in-network traffic filtering, as part of the rule placement procedure, the provider will adjust the subscriber's rules to better match the available deployment locations, thus taking advantage of information unavailable to the subscriber— specifically the locations and availability of other providers in the DrawBridge network. For example, the provider can choose to aggregate two rules into one if this aggregation will result in better use of available deployment locations. This type of rule aggregation takes advantage of the intersection between two given AS-level paths by taking two rules with different corresponding source ASes (and thus different AS-level paths), discovering the paths' intersection, and aggregating to a single rule that must now

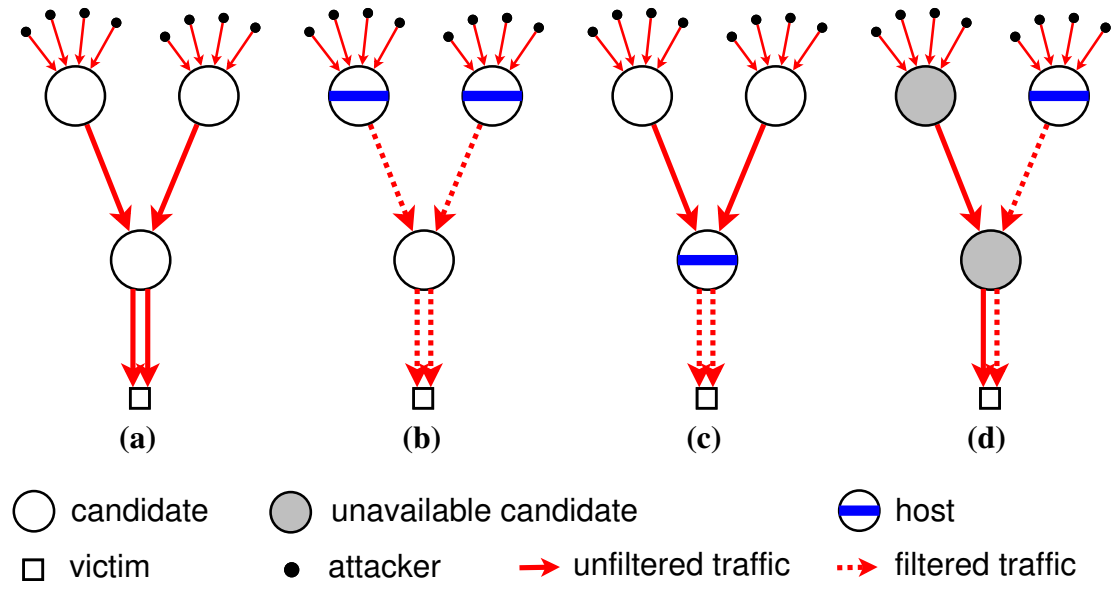


Figure 23. Example scenarios for rule placement.

be deployed at an AS in the intersecting set. This aggregation will never introduce collateral damage, nor will it cause decreased coverage from the subscriber's original ruleset. Rather, the cost of this second-phase aggregation is merely an increase in potential collateral damage as smaller-prefix rules are aggregated into larger prefixes that cover more unknown or unseen sources.

Since the host(s) for a rule must be on the path(s) of the DDoS traffic matching the rule, DrawBridge needs to discover such path(s) in order to discover the host(s). Specifically, a provider must be able to discover the AS-level path(s) of the traffic, and a provider's DrawBridge controller needs to know the switch-level path(s) inside the provider. This task is difficult since the routes between any two given nodes on the Internet are often asymmetric, meaning that between two nodes, the path taken by traffic traveling in one direction is not the same as that taken by traffic flowing in the opposite direction (He, Faloutsos, Krishnamurthy, and Huffaker (2005)). The subscriber, for example, cannot base on the sources of

its DDoS traffic to determine the paths taken by the DDoS traffic. For the former, DrawBridge utilizes existing path inference solutions (such as described in (Burch and Cheswick (2000); Gong and Sarac (2008); Katz-Bassett et al. (2010); Mao, Qiu, Wang, and Zhang (2005); Savage, Wetherall, Karlin, and Anderson (2000); Shi, Zhang, Li, and Reiher (2018); Snoeren et al. (2001))). Certainly no path inference system is perfect, but even semi-accurate path inference will serve the purposes of DrawBridge. For the latter, the DrawBridge controller (which is an SDN controller of the provider) can directly extract the path information using the provider's internal topology and routing information.

A provider then conducts the rule placement procedure as follows. It will iterate through the set of rules in increasing order of their priority. For each rule, it will discover the paths of the DDoS traffic matching the rule, identify host candidates on these paths, and choose which host candidates, if selected as the set of hosts of the rule, would cover as many paths as possible (thus the best efficacy possible), have the furthest possible distance from the subscriber (thus least possible to handle an overwhelming amount of DDoS traffic), and also have space for the rule at each of them. Once the provider chooses the hosts for the rules, it then can try to deploy the rules at them. If hosts are switches inside the provider, the DrawBridge controller of the provider then can use SDN to place rules at those switches. If hosts are other DrawBridge providers, the provider in question then uses the DrawBridge network to send a rule installation message to every host to install the rules for that host.

Upon receiving a rule request from one of its subscribers, a DrawBridge provider further distributes the rule request to *its* provider(s), and so on. Each provider who receives a rule request must determine whether it sees any traffic

that matches the rules so it can send a confirmation to the subscriber who sent the rule request. Each provider also waits to receive confirmations from its immediate upstream providers before sending a confirmation to the requesting subscriber. After receiving confirmations from upstream providers, the original DrawBridge provider has enough information to run the rule placement algorithm in Section 5.3 to select the appropriate deployment locations.

5.3 Algorithmic Design for Rule Placement

In Section 5.2, we discussed the systematic design of the components involved in a in-network filtering systems, and described the rule-placement procedure in high level. In this section, we dig deeper into the design of the rule placement algorithm, and try to answer the following question: *given a set of traffic filtering rules, how can we find the best locations to place the rules so that the DDoS can be most effectively mitigated?* Here, we assume that a set of rules are provided as input, and the rule generation procedure is out of the scope of this dissertation.

5.3.1 Problem Formulation.

We assume a DrawBridge provider needs to place a set of rules $R=\{r_i|i=1, \ldots, n\}$, where r_i is a rule with a priority of y_i. Each rule $r_i \in R$ has a set of host candidates $C_i=\{c_{ij}|j=1, \ldots, |C_i|\}$, where c_{ij} is d_{ij} hops away from the subscriber and has space to accommodate s_{ij} rules. We represent a placement solution of R as a vector of host sets: $P(R)=<H_1, H_2, ..., H_n>$, where $H_i=\{h_{ik}|k=1, \ldots, |H_i|\}$ $(i=1, \ldots, n)$ is the set of hosts for r_i and h_{ik} is d_{ik} hops away from the subscriber. We define the following for $P(R)$:

efficacy of $P(R)$: For rule $r_i \in R$, its host set H_i may not cover all the paths that need rule r_i to filter the DDoS traffic. Assuming e_i is the fraction of paths that

are covered by H_i, we have

$$e(P(R)) \sum_{i=1}^{n} y_i * e_i.$$

$d(P(R))$: average deployment distance of $P(R)$: As discussed in Section 5.2.2, the further away a host is from the subscriber, the less likely it will have to handle a large amount of DDoS traffic. We can define the average deployment distance of H_i as

$$d(H_i) = \frac{\sum_{k=1}^{|H_i|} h_{ik}}{|H_i|},$$

and

$$d(P(R)) = \frac{1}{n} \sum_{i=1}^{n} d(H_i).$$

$s(P(R))$: rule space overhead of $P(R)$: As every host of a rule needs a copy of the rule, we use the number of hosts of a rule to represent its storage overhead. We thus have

$$s(P(R)) = \sum_{i=1}^{n} |H_i|,$$

as r_i is deployed at $|H_i|$ hosts.

We therefore define the rule placement problem as a following multi-objective optimization problem:

With a set of rules R=$\{r_i|i = 1, ..., n\}$ and a given topology of host candidates, how may a DrawBridge provider find a placement solution $P(R)$ such that among all possible placement solutions over the topology, $P(R)$ has the maximal $e(P(R))$, minimal $s(P(R))$, and maximal $d(P(R))$?

As with the multi-objective optimization problem for rule generation, we solve the rule placement problem defined above by formulating multiple single-objective optimization problems and seeking the solution to each of them as a Pareto optimal solution to the multi-objective rule placement optimization problem.

Like the solution to rule generation, this solution is again Pareto optimal since an objective cannot be improved upon without degrading other objectives. In a similar way that rule generation allows the subscriber more freedom to choose what objectives to optimize for when generating rules, rule placement allows the provider more freedom to choose what objectives to optimize for when placing rules in the Internet (e.g., maximize $e(P(R))$, minimize $s(P(R))$, or maximize $d(P(R))$), depending on the circumstances.

Assuming a rule placement solution cannot exceed storage overhead S and the selected hosts must be at least D hops away from the subscriber on average (we leave the choice of S and D to be out of the scope of this study), we focus on the following single-objective problem:

With a set of rules R=$\{r_i|i = 1, ..., n\}$ and a given topology of host candidates, find a placement solution $P(R)$ that maximizes $e(P(R))$ whereas $s(P(R))\leq S$ and $d(P(R))\geq D$.

5.3.2 H-tree Data Structure. For every rule to place we introduce a tree data structure called H-tree. It is rooted at the subscriber, where every other node represents a host candidate of the rule, with the leaf nodes representing the candidates furthest from the subscriber. It encompasses all the paths covered by all the host candidates. Every node on the H-tree is also associated with an s value that indicates how many rules the node can further accommodate. Every node on the H-tree also records how many rules the node can further accommodate. Figure 24 shows two example H-trees for two rules, respectively.

Once a DrawBridge provider discovers the AS-level (if an inter-AS topology) or switch-level (if an intra-AS topology) paths that the DDoS traffic matching the rule would use, as discussed in Section 5.2.2, it can check which DrawBridge

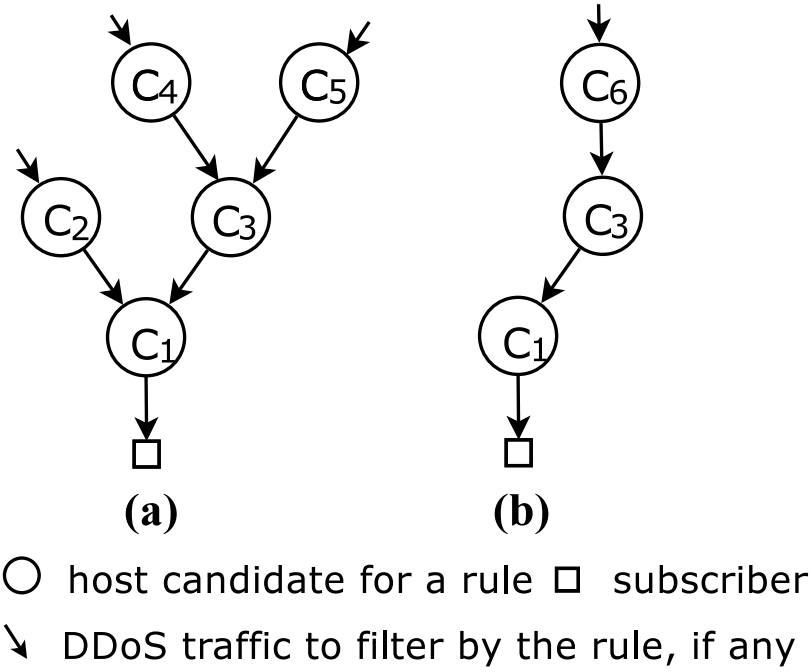


Figure 24. Two example H-trees.

providers or switches are on each path, respectively, and use them to construct the H-tree of the rule.

The H-tree for a subscriber's rule has several important properties related to selecting the hosts of the rule:

- Along every path to the subscriber only one node needs to be selected as a host to cover the path.
- A cut set of the tree covers all paths to the subscriber.
- The cut set using all the leaf nodes is the cut set furthest from the subscriber.
- H-trees for different rules may have host candidates in common; nodes closer to the subscriber are more likely to be a common host candidate for multiple rules.

5.3.3 Rule Placement Algorithm. We now describe how a provider finds the placement solution $P(R)$ for a set of rules R=$\{r_i|i = 1, ..., n\}$. The

provider will iterate through the rules according to their priority (starting with the rule with the highest priority), identify the set of hosts to use for the current rule, and append the set to $P(R)$ (initially empty). The $P(R)$ at the end is then the solution. The core operation at each iteration is to process the H-tree of the current rule, say r_i, and discover a cut set of the H-tree to be the set of hosts for rule r_i, with the following steps:

(i.) The provider determines the allowed maximum size of the cut set. Denote R'=$\{r_j | j = 1, ..., i-1\}$ to be set of rules already handled, $P(R')$ the placement of R', and $s(P(R'))$ the storage overhead of $P(R')$. Given that the storage overhead for $P(R)$ cannot be more than S, the cut set then cannot be larger than L=S-$s(P(R'))$. (If L is zero, the provider will end the iteration and return $P(R')$ as $P(R)$.)

(ii.) The provider discovers the cut set H_i of the H-tree that, among all the cut sets with no more than L nodes, is the furthest from the subscriber. The procedure is as follows: if the cut set with all the leaf nodes (which is the furthest cut set) has no more than L nodes, it then returns this cut set; otherwise, it will recursively replace sibling nodes with a parent node to construct a new, smaller cut set, until it obtains a cut set with no more than L nodes.

(iii.) For every host candidate in H_i that has no space for rule r_i, say X, the provider will traverse the path from X toward the subscriber, discover the first host candidate, say Y, that has space for rule r_i, and replace X and all other Y's descendants in H_i with Y. In case no node exists to replace X, the provider simply removes X from H_i. Note that the algorithm does not require hosts to reveal information of available rule space in their networks, which can be considered sensitive information that ASes would not be willing to share. When a host AS is

asked to deploy a rule r_i, it can simply express whether it is willing to deploy r_i without revealing any sensitive information about the inner workings of its network.

(iv.) If H_i may shorten the average deployment distance such that $d(<P(R'), H_i>))<D$, the provider will recursively remove the host candidate in H_i that has the shortest hops from the subscriber, until $d(<P(R'), H_i>))\geq D$.

As a result, for every rule r_i, the provider obtains a set of hosts H_i that covers the maximal number of paths of the DDoS traffic targeted by r_i, respects the space constraints, and has the largest possible distance from the subscriber. The rule placement algorithm thus finds an optimal solution to the aforementioned single-objective rule placement problem. As detailed in the next subsection, it is polynomial-time in the worst case, and therefore is scalable in terms of the number of rules and host candidates. Furthermore, the algorithm can easily be adapted to solve the rule placement problem for the other potential objectives (minimize $s(P(R))$ or $d(P(R))$).

Let us use the H-tree in Figure 24(a) as an example. Assuming S is large enough to allow cut sets of any size, the provider will identify cut set $H=\{C_2, C_4, C_5\}$ composed of all leaf nodes. If C_2, C_4 and C_5 all have space, the provider will use all three of them. If, say, C_5 has no space but C_3 does, C_3 will replace C_5 as well as C_4, resulting in $H=\{C_2, C_3\}$. If neither C_3 nor C_1 has space, no replacement of C_5 exists, we simply remove C_5 and result in $H=\{C_2, C_4\}$.

5.3.4 Algorithm Complexity Analysis. The best-case scenario complexity of the rule placement algorithm is $O(n)$, where n is the number of rules. In the best case, the number of hosts closest to the sources for all rules will be within S (storage overhead), so the algorithm takes linear time ($O(1)$) to traverse

the H-tree. In other words, for each rule, traversing the H-tree will take $O(1)$ and thus doing this for n rules will lead to a complexity of $O(n)$.

The worst-case scenario complexity is $O(\sum_{i=1}^{n} |C_i|)$. This is because in the worst case, the algorithm will traverse through each host in the H-tree in order to meet the four aforementioned criteria. In other words, traversing the tree for each rule i will take $O(|Ci|)$ and doing this for n rules will lead to $O(\sum_{i=1}^{n} |C_i|)$.

5.4 Rule Placement Efficacy Evaluation

In Section 5.2, we discussed the systematic design of the components involved in a in-network filtering systems, and described the rule-placement procedure in high level. In Section 5.3, we dig into details of the design and complexity of our placement algorithm. In this section, we will evaluate the performance of the placement algorithm.

5.4.1 Evaluation Setup. We built a simulation to measure the performance of rule-placement algorithm against real-world, large-scale DDoS attacks, which we replay using three captured real-world DDoS attack traces that are of different sizes and attack dynamics (Table 6): RADB-2016 (Merit Network (2016)) with the DNS protocol and ∼16,000 DDoS sources, Booter1-2015 (Santanna et al. (2015)) with the DNS protocol and ∼4,500 DDoS sources, and CAIDA-2007 (Hick et al. (2007)) with the ICMP protocol and ∼7,000 DDoS sources. Moreover, we also deployed and evaluated the placement algorithm using real traffic and physical equipment on GENI testbed (Berman et al. (2014)).

Table 6. DDoS attack traces used for evaluation.

Name	Protocol	approx. # of sources
RADB-2016	DNS	16,000
Booter1-2015	DNS	4,500
CAIDA-2007	ICMP	7,000

5.4.2 Static Rule Placement. We first study the performance of rule placement algorithm in a static attack environment where the attack sources does not change over time, and the placement algorithm only need to be conducted once. We evaluate rule placement algorithm against a number of distinct deployment profiles (shown in Table 7), which represent different rates of AS participation in the in-network filtering. We set deployment rates for ASes in tiers 1, 2, and 3, where the total number of ASes in each tier is 89, 8442, and 47052, respectively. Full deployment of DrawBridge is clearly unrealistic, but we use these profiles as a baseline. The "victim only" profile represents a scenario in which the entire network consists of a single subscriber and its provider, which means that all rules must be deployed at local switches controlled by the provider.

Table 7. Deployment profiles for rule placement.

Name	Tier 1	Tier 2	Tier 3	Total #
Full Deployment	100%	100%	100%	55583
Tier 1 only	100%	0%	0%	89
Top-centered	100%	50%	0%	4310
Middle-centered	0%	80%	20%	9410
Bottom-centered	0%	20%	80%	39330
Victim only	0%	0%	0%	1

We first evaluate the rule placement success rate, i.e. the percentage of rules for which suitable locations are found. Figure 25 depicts the success rate under each profile. The first and most obvious trend displayed is that the success rate for all profiles either remains stable or generally increases as we increase the per-AS rule limit from 1 to 1000. Clearly, more available rule space at deploying ASes results in fewer rule deployment failures due to exhausted space. Another trend is the impact of a higher overall deployment rate for DrawBridge. Overall, the placement success rate increases with a higher deployment rate of the DrawBridge

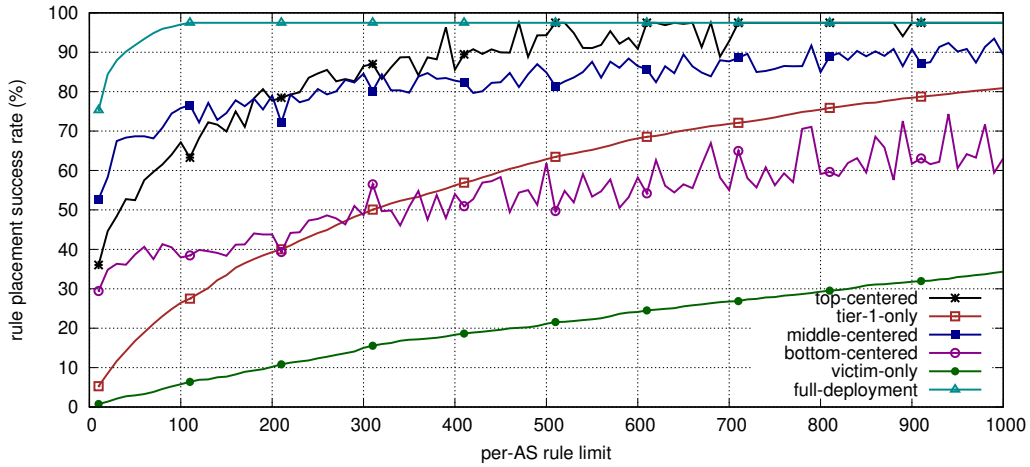


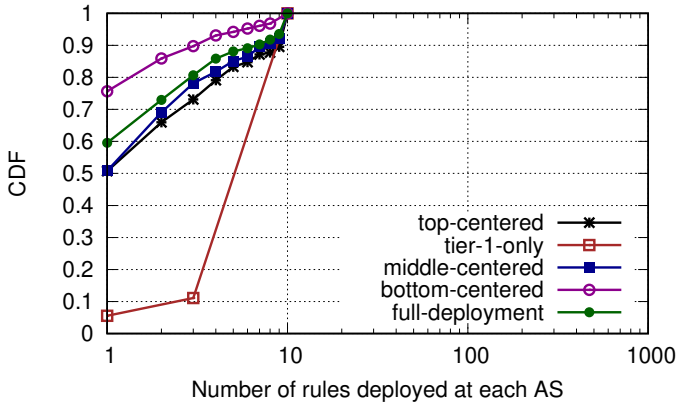
Figure 25. Rule placement success rates.

system itself, though increasing the deployment rate for some AS tiers has different effects than for others. As expected, the lowest success rate belongs to the victim-only profile, while the highest rate is achieved by the full deployment profile. The four profiles in between generally perform much better than the victim-only profile, and slightly or moderately worse than the full-deployment profile, where the top-centered profile is the only profile of these four to reach nearly 100% success rate, and generally performs better than the others. The middle-centered profile is not far behind, however, and actually reaches higher success rates than the top-centered profile when the number of rules per AS is low. The tier-1-only profile is the most sensitive to the per-AS rule limit, as with only 89 tier-1 ASes each AS faces pressure to deploy more rules than other profiles; it thus has a lower success rate than other profiles (except for victim-only) when the per-AS rule limit is low, but gradually improves as the limit gets higher.

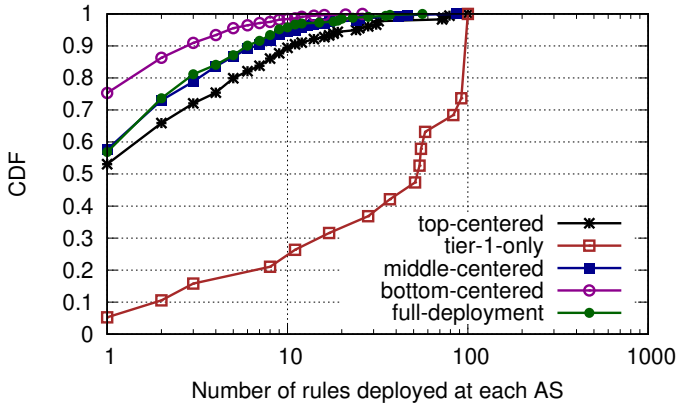
Next, we examine how many rules are placed at each participating AS under five different deployment profiles while each AS can only deploy at most 10, 100, or 1000 rules (Figure 26). Three figures show similar trends, and we take

Figure 26a as an example. Across all deployment profiles except the tier-1-only profile, approximately 60% or more of ASes that participate in the defense must deploy only a single rule, approximately 95% or more of ASes deploy no more than 10 rules, and thus a very small percentage of ASes deploy more than 10 rules. In these cases, increasing the per-AS rule limit past 100 would have little effect, since even a per-AS limit of 10 results in very few (10%) ASes with between 10 and 100 rules deployed. For the tier-1-only profile, the rules are more spread out among all ASes, but note this profile corresponds to the smallest actual number of ASes. Overall, we can see that rather than employing a defense evenly distributed among all ASes, DrawBridge takes advantage of the fact that for any given attack, a small number of ASes are in especially advantageous locations and can contribute disproportionately to the defense.

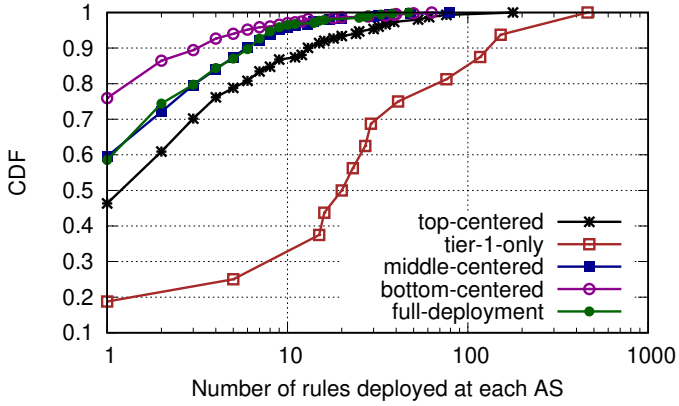
5.4.3 Dynamic Rule Placement in Simulation. We also evaluate the overall efficacy of the rule placement algorithm within the overall DrawBridge system as we defend in real time against real-world DDoS attack traces with multiple rounds of rule placement for continuous incoming sets of rules. The total number of deployed rules never exceeds the budget set by the subscriber while the coverage needs to be above a threshold. Figure 27 shows two representative time series for defense against two replayed DDoS attacks with dissimilar dynamics (CAIDA-2007 and RADB-2015). For each attack, we show the number of DDoS flows filtered by DrawBridge at each second, as well as the number of flows that arrive at the victim when no filtering is performed; although not shown, no legitimate flows are ever filtered. The rule placement algorithm is dynamically and continuous executed to place ever changing sets of rules for the maximum filtering



(a) 10 rules per AS

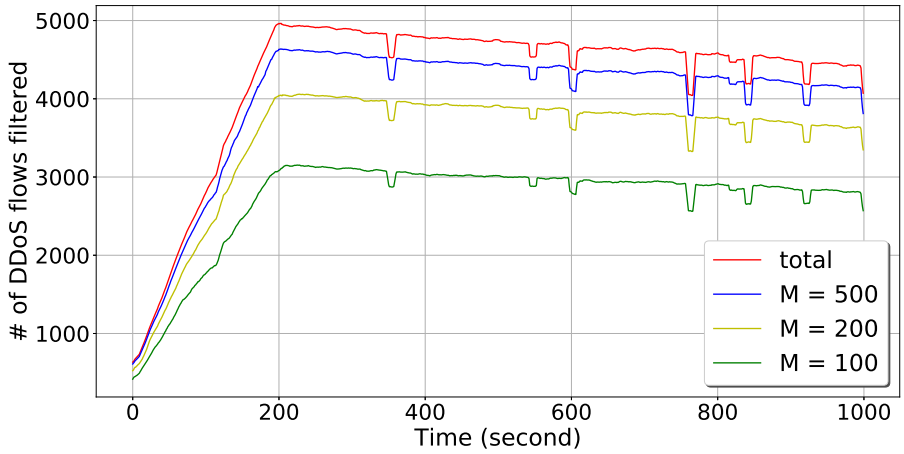


(b) 100 rules per AS

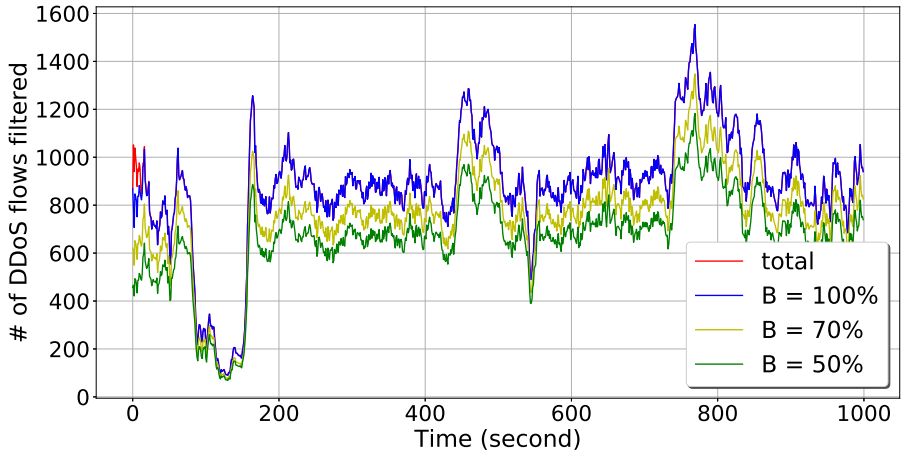


(c) 1000 rules per AS

Figure 26. The distribution of the number of rules deployed at each DrawBridge-participating AS.



(a) CAIDA-2007 DDoS attack under rules for maximal coverage with various rule budgets.



(b) RADB-2015 DDoS attack under rules for minimal number of rules with various required DDoS coverage.

Figure 27. Time series of DrawBridge's filtering of DDoS flows. The "total" curve shows DDoS flows w/o filtering.

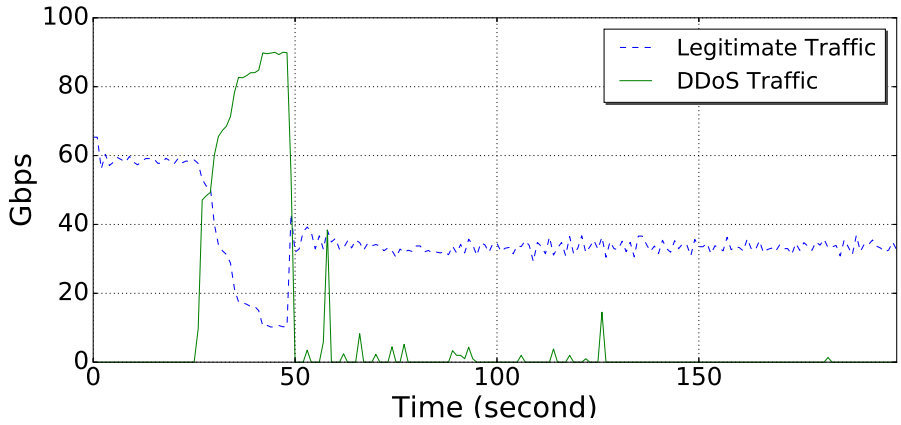
efficacy. The filtering efficacy results reflects the performance of the rule placement algorithm.

More specifically, Figure 27a applies rules that are generated based on source addresses of the traffic toward maximal DDoS coverage under *zero* collateral

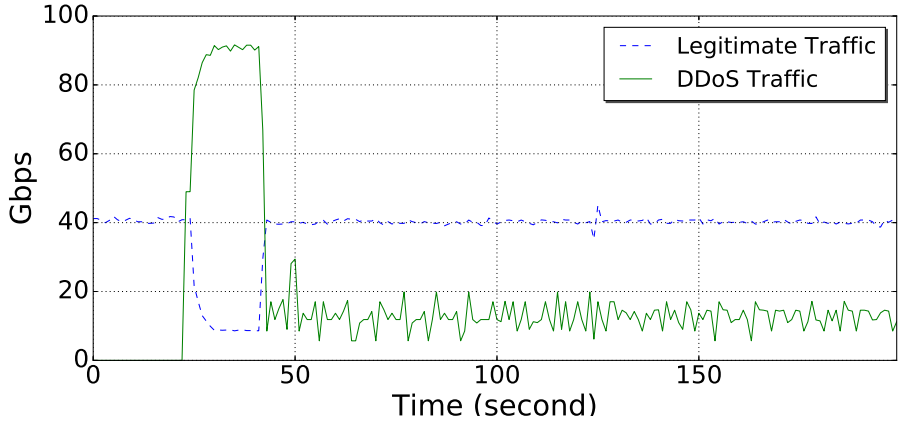
damage requirement *and* three different rule budgets (100, 200, and 500, which represent roughly 1.5%, 3%, and 7%, respectively, of the total approximately 7,000 DDoS sources). Here, as expected, the more effective filtering is achieved with a higher value for the rule budget, and even with a tight budget of 100 source-based rules that is only 1.5% of DDoS sources, 60-70% of DDoS flows will be filtered. Figure 27b instead applies rules that are generated toward minimum number of rules under *zero* collateral damage requirement *and* three different requirements on minimum DDoS coverage (100%, 70%, and 50%). The generation and placement of rules tracks very closely the spikes in the attack traffic, demonstrating the overall accuracy of DrawBridge's rule generation and placement algorithms. In particular, with rules required to cover 100% DDoS, although initially not all DDoS flows are filtered, it takes only about 13 seconds for DrawBridge to begin filtering *all* DDoS flows at every second afterwards.

5.4.4 Dynamic Rule Placement in Real Network. To further evaluate the placement algorithm under more realistic environment and using real network traffic, we deployed the system on the GENI (Global Environment for Network Innovations) testbed (Berman et al. (2014)), and emulated DDoS attacks to measure the traffic filtering performance. Based on an Internet topology that consists of all Internet ASes as of June 2018, we chose a subgraph of 1 tier-1 AS, 18 tier-2 ASes, and 31-tier3 ASes as a deployment topology of DrawBridge where each of the total 50 ASes is a DrawBridge-participating AS and they form a DrawBridge network. We attach a local machine to one of the 50 ASes as a DrawBridge subscriber.

Each of these 50 ASes is supported with two virtual machines provided by GENI. The first virtual machine for each AS runs a Ryu controller as an SDN



(a) Traffic time series under rules for minimal collateral damage with a rule budget of 150 rules to cover 100% of DDoS traffic.



(b) Traffic time series under rules for maximal DDoS coverage without any collateral damage and with a rule budget of 200 rules.

Figure 28. Volume of legitimate and DDoS traffic over time before and during traffic filtering on GENI.

controller, an Open vSwitch (Pfaff et al. (2015)) as an SDN switch that can deploy OpenFlow rules, and a DrawBridge controller on the Ryu controller, where the Open vSwitch is populated with a forwarding table by running the OSPF routing protocol (as in (Moy (2017))) over the deployment topology. The second virtual machine for each AS acts as an end-host in the AS that can generate benign traffic toward a destination from different IP addresses of the AS. More, in order

to emulate large-scale DDoS attacks on the deployment topology, we installed a DDoS agent on each AS's second virtual machine. It can receive commands about a variety of DDoS attacks from a bot master that we deployed on GENI and generate DDoS traffic toward a victim at a scheduled time from different IP addresses of an IP prefix from the AS.

Our procedure is as follows: we first bootstrap the DrawBridge network by providing each AS's local DrawBridge controller with a routing table that contains a mapping from each destination to the next-hop controller, thus allowing messages to flow between DrawBridge controllers.

The system runs smoothly on this platform with good performance and low network overhead. It also runs fast with rule generation at 105 milliseconds on average and rule placement mainly subject to network latency. The network overhead is no more than 10 kilobytes each round for rule deployment.

We launch an emulated 100-Gbps DDoS attack toward the subscriber from roughly 1000 unique source addresses, together with 40- to 60-Gbps legitimate traffic to the subscriber from ∼200 sources. In this case, with the subscriber's generated rules, we run the placement algorithm as part of the system's filtering pipeline, and deploy input rules at desired locations to filter the DDoS traffic in question.

Figure 28 shows the defense in two different scenarios. In the first scenario (Figure 28a) where the defense begins at second 48, it takes *only approximately 3 seconds* for the filtering of DDoS traffic to reach 100%. Since we are using source-based filtering, and the number of unique attack sources (1000) is relatively high compared to the rule budget of 150, some collateral damage has to happen, preventing the volume of legitimate traffic since second 48 from recovering to

the level seen before the attack; nonetheless, the legitimate traffic does recover relative to the sharp dip to about 11 Gbps while DDoS is at its peak. In the second scenario (Figure 28b), we increase the rule budget to 200 and require *zero* collateral damage; although we no longer filter as much of the DDoS traffic as the first scenario, we filter enough to relieve the link congestion, while all the legitimate traffic can continue to flow at its previous rate.

5.5 Conclusion

In-network DDoS traffic filtering requires not only the participation from network providers on the Internet, but more importantly the effective decisions from the defender on how to utilize the participants in order to achieve the maximum efficacy of defense. There lacks an effective placement algorithm that can achieve high performance on both reducing the traffic to the victim and on flowing on the Internet unfiltered.

In this chapter, we present our design of an efficient rule placement algorithm that can find optimal placements for a give set of rules. To achieve optimal results, we design a tree-like data structure called H-tree to model the placements of the rules on the DDoS traffic topology toward the victim. As a result, our algorithm can find placement locations that maximize the coverage of the DDoS traffic, minimize the rule space needed for deployment, and maximize the distance away from the victim's network.

As part of our efforts to build an effective in-network filtering system, we also briefly introduce our design of the DrawBridge system in this Chapter. Through DrawBridge, we can evaluate the efficacy of the placement algorithm in more realistic environment. We evaluate the efficacy of our placement algorithm using both simulation and distributed emulation via DrawBridge. We show that

our placement algorithm can find effective locations to place filtering rules, and effectively filter high-volume DDoS traffic.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Distributed denial-of-service (DDoS) attacks continue to threaten the availability and integrity of critical Internet infrastructure upon which the society relies more heavily than ever before. As the attack's frequency and severity continues to increase, the current defense paradigm and solutions fail to stay ahead. Traditional "edge-defense" solutions suffer from high-filtering cost and early congestion problems; and it has become a never-ending arms-race between the ever-growing strength of botnets and the defender's invested infrastructure. In the mean time, "in-network-defense" solutions has promising features to address the problems, but are less explored and yet put into practice. This dissertation has studied the in-network defense solutions from both theoretical aspects and systematical aspects, and come up with the following conclusions:

- ISPs inside the Internet can be incentivized to participate in the filtering DDoS traffic;
- There are two major types of existing in-network defense strategies (PushBack and SourceEnd); our proposed StrategicPoints strategy outperforms the existing ones in most cases;
- We designed an in-network traffic filtering rule placement algorithm that can locate optimal filtering rule deployment locations and achieve effective traffic filtering.

Specifically, we have made the following contributions and observations in each section.

In Chapter III, we proposed a game-theoretical model that examines the incentives of ASes to invest in efforts on DDoS defense. Based on the model, we built a large-scale simulation system that can simulate the path propagation procedure on the Internet with consideration of DDoS defense efforts of each provider AS. Through simulation, we observed the following patterns from the simulation results. The majority of the provider ASes on the Internet can benefit from providing DDoS defense services to their customers if they can compensate the defense cost by charging the filtering of DDoS traffic. The severity of DDoS attacks affects the charge rate that a provider can place on its potential customers; if a provider sees a higher volume of DDoS traffic going through its potential customers, it would charge higher to achieve its peak profit. The level of competition also drives down the charge rate. These observations provide confidence that provider ASes on the Internet can have incentive to participate in DDoS defense.

In Chapter IV, we modeled and evaluated different multi-point, in-networking DDoS traffic filtering algorithms. We categorized existing in-network DDoS filtering algorithms into two basic types, i.e. PushBack and SourceEnd, then introduced StrategicPoints algorithms that outperform PushBack and SourceEnd algorithms in most cases. Through simulation, we evaluated and compared the three types of multi-point, in-network DDoS filtering algorithms in terms of their capability in reducing the DDoS traffic leakage to the victim and the pollution to the whole Internet, as well as their resiliency against dynamic DDoS attacks. With real-world, Internet-scale DDoS attack traces, our evaluation results show that StrategicPoints is as effective as PushBack on reducing leakage and significantly

outperforms PushBack and SourceEnd on reducing the pollution caused by DDoS attacks.

In Chapter V, we introduced our design and evaluation of an in-network DDoS traffic filtering rule placement algorithm that is effective against large-scale DDoS attacks and able to locate satisfactory placement locations for traffic filtering rules. Specifically, we designed a tree-like data structure called H-tree to model the placements of the rules on the DDoS traffic topology toward the victim. Our algorithm aims to maximize the coverage of the DDoS traffic, minimize he rule space needed for deployment, as well as maximize the distance away from the victim's network. As part of our efforts to design and evaluate an effective in-network rule placement algorithm, we also briefly introduced a DDoS-filtering system called DrawBridge in which our algorithm runs. Using DrawBridge, we evaluated the efficacy of the rule placement algorithm in a more realistic environment. Specifically, we evaluated the efficacy of our placement algorithm using both simulation and distributed emulation via DrawBridge. We showed that our placement algorithm can find effective locations to place filtering rules, and effectively filter DDoS traffic.

6.2 Future Work

There are multiple future directions that the work of dissertation can further develop into.

We can further expand the incentive study to connect it to real-world network operations. By incorporating operational information, such as pricing models, per-network available resources, and real-world inter-network relationships, we can improve study to more closely model the Internet operations. We can also study the interactions between customers and providers, as opposed to only

provider-provider interactions, to expand the scope of this model. The expansion of the study should allow researchers and operators to better design and deploy in-network DDoS filtering solutions.

Second direction is to further evaluate our in-network traffic-filtering rule placement algorithm in real-world production-level network operations. Although we have evaluated our algorithm under both simulated and testbed environment, an evaluation from a real setting can be more convincing for its efficacy against real-world DDoS attacks.

A third direction can be explored to focus on the inter-operatability between filtering rule placement and other components for DDoS traffic filtering. Such components can be DDoS detection software, traffic classification software and algorithms, different traffic forwarding/filtering infrastructure, etc.

REFERENCES CITED

Akamai DDoS Protection Service. (2016). http://www.akamai.com/us/en/solutions/products/cloud-security/ddos-protection-service.jsp.

Andersen, D. G. (2003). Mayday: Distributed Filtering for Internet Services. *USENIX Symposium on Internet Technologies and Systems*, *4*.

Anderson, S. P., De Palma, A., & Thisse, J. F. (1992). *Discrete Choice Theory of Product Differentiation.* MIT press.

Argyraki, K., & Cheriton, D. R. (2005). Active Internet Traffic Filtering: Real-time Response to Denial-of-service Attacks. *USENIX Annual Technical Conference*.

Baker, F., & Savola, P. (2004). *RFC 3704: Ingress filtering for multihomed networks* (Tech. Rep.).

Bedi, H. S., Roy, S., & Shiva, S. (2011). Game theory-based defense mechanisms against DDoS attacks on TCP/TCP-friendly flows. *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 129–136.

Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., ... Seskar, I. (2014). GENI: A federated testbed for innovative network experiments. *Computer Networks*, *61*(0), 5 - 23.

Beverly, R., & Bauer, S. (2005). The Spoofer project: Inferring the extent of source address filtering on the internet. *USENIX Sruti*, *5*, 53–59.

Blume, L. E. (1993). The statistical mechanics of strategic interaction. *Games and Economic Behavior*, *5*(3), 387–424.

Blume, L. E., Brock, W. A., Durlauf, S. N., & Jayaraman, R. (2015). Linear social interactions models. *Journal of Political Economy*, *123*(2), 444–496.

Bohawek, S., Hespanha, J. P., Lee, J., Lim, C., & Obraczka, K. (2007). Game theoretic stochastic routing for fault tolerance and security in computer networks. *IEEE Transactions on Parallel and Distributed Systems*, *18*(9), 1227–1240.

Brock, W. A. (1993). Pathways to randomness in the economy: emergent nonlinearity and chaos in economics and finance. *Estudios Economicos*, 3–55.

Brock, W. A., & Durlauf, S. N. (2001). Discrete choice with social interactions. *The Review of Economic Studies*, *68*(2), 235–260.

Brock, W. A., & Durlauf, S. N. (2002). A multinomial-choice model of neighborhood effects. *The American Economic Review*, *92*(2), 298–303.

Burch, H., & Cheswick, B. (2000). Tracing anonymous packets to their approximate source. *USENIX Large Installation System Administration Conference (LISA)*, 319-327.

CAIDA. (2019). *As relationships dataset.* http://www.caida.org/data/as-relationships/.

DDoS Protection by Arbor Networks APS. (2016). http://www.arbornetworks.com/ddos-protection-products/arbor-aps.

Dietzel, C., Wichtlhuber, M., Smaragdakis, G., & Feldmann, A. (2018). Stellar: Network attack mitigation using advanced blackholing. *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*.

Durlauf, S. N., & Ioannides, Y. M. (2010). Social interactions. *Annual Review of Economics*, *2*(1), 451–478.

Dyn Research. (2016). *Dyn analysis summary of Friday october 21 attack.* https://dyn.com/blog/egypt-leaves-the-internet/.

Fayaz, S. K., Tobioka, Y., Sekar, V., & Bailey, M. (2015). Bohatei: Flexible and elastic DDoS defense. *24th USENIX Security Symposium*.

Gao, L. (2001). On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, *9*(6), 733–745.

Gill, P., Schapira, M., & Goldberg, S. (2011). Let the market drive deployment: A strategy for transitioning to BGP security. *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*, 14–25.

Goeree, J. K., Holt, C. A., & Palfrey, T. R. (2005). Regular quantal response equilibrium. *Experimental Economics*, *8*(4), 347–367.

Gong, C., & Sarac, K. (2008). A more practical approach for single-packet IP traceback using packet logging and marking. *IEEE Transactions on Parallel and Distributed Systems*, *19*(10), 1310-1324.

Grossklags, J., Christin, N., & Chuang, J. (2008a). Secure or insure?: a game-theoretic analysis of information security games. *Proceeding of the 17th international conference on World Wide Web*, *7*(1), 209–218.

Grossklags, J., Christin, N., & Chuang, J. (2008b). Security and insurance management in networks with heterogeneous agents. *Proceedings of the 9th ACM conference on Electronic commerce*, 160–169.

He, Y., Faloutsos, M., Krishnamurthy, S., & Huffaker, B. (2005). On routing asymmetry in the internet. *IEEE Global Telecommunications Conference*, *2*, 6-pp.

Hick, P., Aben, E., Claffy, K., & Polterock, J. (2007). *The CAIDA DDoS Attack 2007 Dataset.*

Huang, Y., Geng, X., & Whinston, A. B. (2007). Defeating DDoS attacks by fixing the incentive chain. *ACM Transactions on Internet Technology*, *7*(1), 5–es.

Huici, F., & Handley, M. (2007). An Edge-to-Edge Filtering Architecture Against DoS. *ACM SIGCOMM Computer Communication Review*, *37*.

Huston, G., Smith, P., & Bates, T. (n.d.). *CIDR Report.* (`https://www.cidr-report.org/as2.0/`, last accessed on 2019-01-28)

Kalkan, K., & Alagöz, F. (2016). A distributed filtering mechanism against DDoS attacks: ScoreForCore. *Computer Networks*, *108*.

Kang, M. S., Gligor, V. D., & Sekar, V. (2016). SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks. *NDSS*.

Kang, M. S., Lee, S. B., & Gligor, V. D. (2013). The Crossfire Attack. *2013 IEEE Symposium on Security and Privacy (SP)*, 127–141.

Katz-Bassett, E., Madhyastha, H. V., Adhikari, V. K., Scott, C., Sherry, J., Van Wesep, P., ... Krishnamurthy, A. (2010). Reverse traceroute. *USENIX Symposium on Networked Systems Design and Implementation*, 219-234.

Keromytis, A. D., Misra, V., & Rubenstein, D. (2004). SOS: An architecture for mitigating DDoS attacks. *IEEE Journal on Selected Areas in Communications*, *22*.

Kline, E., Beaumont-Gay, M., Mirkovic, J., & Reiher, P. (2009, December). RAD: Reflector attack defense using message authentication codes. *IEEE Annual Computer Security Applications Conference*.

Kottler, S. (2018). February 28th ddos incident report. *GitHub Engineering*. https://githubengineering.com/ddos-incident-report/.

Laszka, A., Felegyhazi, M., & Buttyan, L. (2014). A Survey of Interdependent Information Security Games. *ACM Computing Surveys*, *47*(2), 1–38.

Li, J., Zhang, M., Shi, L., Sisodia, D., Mergendahl, S., Feng, Y., & Reiher, P. (2019). *Victim-driven, Rule-based, In-network Filtering of Distributed Denial-of-Service Traffic* (Tech. Rep. No. CCSP-TR-2019-01). University of Oregon.

Liu, X., Yang, X., & Lu, Y. (2008). To filter or to authorize: Network-layer DoS defense against multimillion-node botnets. *ACM SIGCOMM Computer Communication Review*.

Liu, Z., Jin, H., Hu, Y.-C., & Bailey, M. (2016). MiddlePolice: Toward enforcing destination-defined policies in the middle of the Internet. *ACM SIGSAC Conference on Computer and Communications Security*.

Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., et al. (2013). As relationships, customer cones, and validation. *Proceedings of the 2013 conference on Internet measurement conference*, 243–256.

Lumbis, P., Ramdoss, Y., & Miller, D. (2014). CAT 6500 and 7600 Series Routers and Switches TCAM Allocation Adjustment Procedures. *Cisco TechNotes*. Retrieved from "http://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/117712-problemsolution-cat6500-00.html"

Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., & Shenker, S. (2002a). Controlling high bandwidth aggregates in the network. *SIGCOMM Computer Communication Review*, *32*(3).

Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V., & Shenker, S. (2002b). Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, *32*.

Manshaei, M. H., Zhu, Q., Alpcan, T., Bacşar, T., & Hubaux, J.-P. (2013). *Game theory meets network security and privacy* (Vol. 45) (No. 3).

Mao, Z. M., Qiu, L., Wang, J., & Zhang, Y. (2005). On as-level path inference. *Proceedings of the 2005 International Conference on Measurement and Modeling of Computer Systems*, 339-349.

Matthew Luckie, Ken Keys, Ryan Koga, Rob Beverly, kc claffy. (2016). *Spoofer source address validation measurement system.* https://spoofer.caida.org. (Accessed: 2019-05-15)

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. *P. Zarembka (ed.), FRONTIERS IN ECONOMETRICS*, 105-142.

McKelvey, R. D., & Palfrey, T. R. (1995). Quantal response equilibria for normal form games. *Games and Economic Behavior*, *10*(1), 6–38.

Merit Network, I. (2016). *A DDoS event against the RADb service.* https://www.impactcybertrust.org/dataset_view?idDataset=576.

Mirković, J., Prier, G., & Reiher, P. (2002). Attacking DDoS at the source. *10th IEEE International Conference on Network Protocols*.

Miu, T., Hui, A., Lee, W., Luo, D., Chung, A., & Wong, J. (2013). Universal DDoS mitigation bypass. *Black Hat USA*.

Miura-Ko, R. A., Yolken, B., Mitchell, J., & Bambos, N. (2008). Security Decision-Making among Interdependent Organizations. *2008 21st IEEE Computer Security Foundations Symposium*, 66–80.

Morales, C. (2018). *NETSCOUT Arbor Confirms 1.7 Tbps DDoS Attack; The Terabit Attack Era Is Upon Us.* `https://www.netscout.com/blog/asert/netscout-arbor-confirms-17-tbps-ddos-attack-terabit-attack-era`.

Moy, J. (2017). *OSPF Version 2.* https://tools.ietf.org/html/rfc2328.

NETSCOUT. (2019). *Arbor availability protection system.* `https://www.netscout.com/product/arbor-availability-protection-system`.

Odintsov, P. (2019). *FastNetMon community - very fast DDoS analyzer with sflow, netflow, mirror support.* `https://github.com/pavel-odintsov/fastnetmon`. GitHub.

Oikonomou, G. C., Mirkovic, J., Reiher, P. L., & Robinson, M. (2006). A Framework for a Collaborative DDoS Defense. *ACSAC*, *6*.

Pagiamtzis, K., & Sheikholeslami, A. (2006). Content-addressable memory (CAM) circuits and architectures: A tutorial and survey. *IEEE Journal of Solid-State Circuits*, *41*.

Papadimitriou, C. H. (2001). Algorithms, games, and the Internet. *Proceedings of the 33rd annual ACM symposium on Theory of computing*, 749–753.

Papadopoulos, C., Lindell, R., Mehringer, J., Hussain, A., & Govindan, R. (2003). COSSACK: Coordinated Suppression of Simultaneous Attacks. *DARPA Information Survivability Conference and Exposition*, *2*.

Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A., Rajahalme, J., ... Casado, M. (2015). The design and implementation of open vswitch. *12th USENIX Symposium on Networked Systems Design and Implementation*, 117-130.

Ramanathan, S., Mirkovic, J., Yu, M., & Zhang, Y. (2018). Senss against volumetric ddos attacks. *Proceedings of the 34th Annual Computer Security Applications Conference*.

RIPE RIS. (2019). *RIPE RIS raw data.* https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data.

Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., & Wu, Q. (2010). A Survey of Game Theory as Applied to Network Security. *2010 43rd Hawaii International Conference on System Sciences*, 1–10.

Sahay, R., Blanc, G., Zhang, Z., & Debar, H. (2015). Towards Autonomic DDoS Mitigation using Software Defined Networking. *Workshop on Security of Emerging Networking Technologies*.

Sami, R., Katabi, D., Faratin, P., & Wroclawski, J. (2004). Practice and Theory of Incentives in Networked Systems (PINS): Workshop Report.

Samuelson, P. A., & Nordhaus, W. D. (2001). *Microeconomics, ISE Editions.* McGraw-Hill Education, New York.

Santanna, J., van Rijswijk-Deij, R., Hofstede, R., Sperotto, A., Wierbosch, M., Zambenedetti Granville, L., & Pras, A. (2015, May). Booters - an analysis of ddos-as-a-service attacks. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 243-251.

Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2000). Practical network support for IP traceback. *Proceedings of the 2000 ACM Conference on SIGCOMM*, 295-306.

Shen, Y., Yan, Z., & Kantola, R. (2013). Game Theoretical Analysis of the Acceptance of Global Trust Management for Unwanted Traffic Control. *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*(October 2015), 935–942.

Shi, L., Zhang, M., Li, J., & Reiher, P. (2018, May). Pathfinder: Capturing ddos traffic footprints on the internet. *IFIP Networking*.

Shiva, S., Roy, S., & Dasgupta, D. (2010). Game theory for cyber security. *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*(APRIL), 34.

Snoeren, A. C., Partridge, C., Sanchez, L. A., Jones, C. E., Tchakountio, F., Kent, S. T., & Strayer, W. T. (2001). Hash-based IP traceback. *ACM SIGCOMM*, 3-14.

University of Oregon. (2019). *Route Views project.* http://www.routeviews.org.

Vissers, T., Van Goethem, T., Joosen, W., & Nikiforakis, N. (2015). Maneuvering around clouds: Bypassing cloud-based security providers. *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, 1530–1541.

Wang, B., Zheng, Y., Lou, W., & Hou, Y. T. (2015, April). DDoS attack protection in the era of cloud computing and software-defined networking. *Computer Networks*.

Wu, Q., Shiva, S., Roy, S., Ellis, C., & Datla, V. (2010). On modeling and simulation of game theory-based defense mechanisms against DoS and DDoS attacks. *Proceedings of the 2010 Spring Simulation Multiconference on - SpringSim '10*, 10.

Yang, X., Wetherall, D., & Anderson, T. (2008). TVA: A DoS-limiting Network Architecture. *IEEE/ACM Transactions on Networking*, *16*, 1267–1280.

Yau, D. K., Lui, J., Liang, F., & Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking*, *13*.

Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys and Tutorials*, *15*.

Zhang, M., Shi, L., Sisodia, D., Li, J., & Reiher, P. (2019, April). On Multi-Point, In-Network Filtering of Distributed Denial-of-Service Traffic. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 180-188.

Zhang, M., Wu, J., Li, J., & Reiher, P. (2019). A Game Theoretical Analysis of Distributed Denial-of-Service Defense Incentive. *unpublished*.

Zheng, J., Li, Q., Gu, G., Cao, J., Yau, D. K., & Wu, J. (2018). Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis. *IEEE Transactions on Information Forensics and Security*.