UNIVERSITY OF CALIFORNIA
Los Angeles

Building A DDoS-Resilient Internet Architecture

A prospectus for Oral Qualifying Exam (OQE) submitted
in partial satisfaction of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Zhiyi Zhang

2018

The Committee for the Oral Qualifying Exam and the Prospectus of Zhiyi Zhang.

Alexander Afanasyev

Songwu Lu

Leonard Kleinrock

Lixia Zhang, Committee Chair

University of California, Los Angeles

2018

ABSTRACT OF THE PROSPECTUS

Building A DDoS-Resilient Internet Architecture

by

Zhiyi Zhang
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2018

Distributed Denial-of-Service (DDoS) attacks on the Internet have caused serious problems for many years and have become more severe over time. The difficulty in mitigating DDoS attacks comes from the architectural shortcomings of IP which make it easy to attack any IP node from anywhere. Named Data Networking (NDN) changes the network communication model from the delivery of packets between hosts identified by IP addresses to the retrieval of named and secured data packets. Consequently, NDN fundamentally changes the approaches to network security.

In this prospectus, we comprehensively examine the basic properties of the NDN architecture and describe how they make the launch of DDoS attacks more difficult and the attacks less effective. We further explore NDN's architectural properties to develop a new DDoS mitigation solutions. We have developed a preliminary approach is called producer-assisted pushback (PAP), which pushes back DDoS traffic to misbehaving entities, at a much finer granularity than existing DDoS defense mechanisms in IP networks. We evaluated the performance of PAP through extensive simulations and our results show that PAP can effectively push back attack traffic within a few seconds, and ensure over 99% of an attack target's incoming traffic is from legitimate clients. The prospectus also discusses my plan on how to further evolve the PAP design to a mature solution for DDoS mitigation in NDN networks.

TABLE OF CONTENTS

CHAPTER 1

Introduction

Today's Internet runs the TCP/IP protocol stack, which did not have security as a primary design goal. As the Internet plays an increasingly important role in all aspects of our modern society, the world has seen an increasing number of malicious attacks to the Internet through exploiting vulnerabilities in the IP design. The Distributed Denial-of-Service attack (DDoS) has been recognized as one of the biggest threats among all, and new types of DDoS attacks continue to evolve [AAB17a, Kum18].

A number of DDoS countermeasures [MR04] have been proposed over the years. However, instead of being curtailed, DDoS attacks on the Internet seem to have gotten worse. We believe that the root cause of the problem comes from the design of IP itself: it makes DDoS attacks trivial to launch – any node on the Internet can flood packet to any IP address and port number. On top of this, one can generate IP packets with any source IP addresses as there is no source address authentication in general.

Named Data Networking (NDN) [ZAB14], a proposed Internet architecture, changes the basic communication model. Instead of pushing packets to their destination addresses as IP does, in an NDN network users request named data by sending *Interest packets*. In this prospectus, we show that NDN's architectural designs provide a solid foundation for DDoS defense mechanisms. As a simple example, NDN's data pull model assures that no one will receive unsolicited data packets.

Since attackers may still attack an NDN network by flooding interest packets, we want to leverage NDN's stateful forwarding plane to mitigate such attacks. In this prospectus, we first introduce an preliminary approach called Producer-assisted Pushback,*PAP*, to mitigate DDoS attacks by Interest packet flooding. As we show later in this prospectus, PAP achieves

the following desirable goals:

- Fast DDoS detection by utilizing explicit feedback provided by the victim;
- Fine-grained traffic Pushback which only affects the traffic under a specific application-defined namespace,
- Effective reaction by pushing back traffic to the exact misbehaving clients/attackers; and
- Selective rate limiting at client gateway router by monitoring whether an end host follows Interest transmission control (i.e., whether lower down its sending rate).

It is expensive for similar approaches to be deployed in TCP/IP network because the forwarding plane of IP is stateless, which cannot provide rich insights of the traffic. In contrast, NDN's architectural designs including stateful forwarding plane and two-way packet exchange, make it easy for PAP to function over NDN.

PAP has been designed and evaluated by Zhiyi Zhang as a graduate student at UCLA with his partners, and the results showed that PAP can effectively mitigate Interest Flooding that targeted end hosts. However, there is still a long way to go and finally have a mature design to make NDN a DDoS-resilient Internet architecture. For instance, PAP cannot detect DDoS attacks which aim to take specific routers or links down. We leave those unsolved issues as the main research topics for Zhiyi's future work at UCLA.

The rest of this prospectus is organized as follows. We discuss the DDoS and current defense approaches over TCP/IP architecture in Chapter 2 and briefly review the basic concepts of NDN in Chapter 3. We then provide a comprehensive analysis of NDN's architectural properties for DDoS mitigation in Chapter 4. In Chapter 5, we give the threat model, goals, and assumption of our approach and in Section 6, we describe the design, explain our design choices, and provides an description of our implementation in detail of PAP. In Section 7, we demonstrate NDN's DDoS resilience and the performance of PAP with simulation results based on an NS-3 platform. We discuss potential issues in Section 8 and the future work in Chapter 9. We conclude our work in Section 10.

CHAPTER 2

DDoS and Current Mitigation Approaches over TCP/IP Architecture

2.1 DDoS and Vulnerabilities in IP

A number of well-known DDoS attacks such as TCP SYN flooding, ICMP flooding, and UDP flooding exploit the vulnerabilities of TCP/IP architecture. TCP SYN flooding is where an attacker sends the victim a high volume of TCP SYN packets with spoofed source addresses. [KAA14]. Such an attack will eventually consume TCP SYN table space through bogus SYN requests. ICMP flooding and UDP flooding are similar in nature [MR04], and cause damage via sheer throughput volume. Other attacks such as reflection or amplification attacks [Ros14a] abuse services like DNS; these attacks work because the attacker writes the victim's IP address in the source address and thus the attacker can effectively direct traffic to DDoS a victim by requesting large amounts of data on behalf of the victim.

As pointed out in many previous works [JWS03, BCR16, YWA05, HG04, Ros14b], DDoS attacks mainly exploit the following vulnerabilities in IP:

- **Push-model Communication** Any Internet node can send a packet to another Internet node with an IP address.
- **Destination-based Delivery** Packet delivery is solely based on the destination address and there is no validation of source addresses in the routing system, thus source IP addresses can easily be misattributed; reflection attacks and simplification attacks are further enabled by IP spoofing.
- **Stateless Forwarding of IP** There is no state in the forwarding system. Therefore,

DDoS defense mechanisms themselves have to add states at the router level in order to inspect network traffic. Also, because of the stateless forwarding plane, congestion control in TCP/IP architecture is end-to-end and compliance with such controls is dependent on the end host.

Those vulnerabilities make the launch of DDoS attack easy and the mitigation hard.

2.2 Current DDoS Mitigation Approaches

There have been various solutions constructed around defending DDoS attacks. Following the taxonomy defined in the previous works [MR04, GHG14], we take several representative approaches as examples and analyze how those mechanisms work but are limited by the nature of IP networking.

Filter-based and rate limiting approaches such as Ingress Filtering [FS00], Pushback [IB02], Black hole filtering [KM09] and some other recent works [LYL08, YPS04] utilize detection mechanisms to identify offending traffic and control the traffic by filtering it out or applying rate limiting. These mechanisms usually require extra insights on ongoing traffic provided by the forwarding system. However, the packet forwarding in IP routers is stateless and it cannot provide accurate insights of the ongoing traffic. Thus, these mechanisms take the risk of causing collateral damage on the legitimate traffic.

Capability-based approaches like TVA (Traffic Validation Architecture) [YWA05, LLY08] introduce authentication of the packet source into the network system. For example, it does so by embedding authentication information into the IP packet, so that the routing system and servers can distinguish legitimate users from the “bad" ones. However, the primary problem is that network layer doesn’t provide source authentication. Adding authentication as patches creates additional complexity because of the mismatch between the unsecured underlying network architecture and a secured forwarding system.

Ioannidis and Bellovin proposed a realization of router-based Pushback [IB02] based on IP networking. To overcome the difficulty of knowing with certainty whether a packet

actually belongs to a "good" or a "bad" flow, their proposed solution utilizes a heuristics function to detect packets that **probably** belong to an attacker. This is done by using the "congestion signature". Routers can thus preferentially drop the packets. However, it is hard to implement this Pushback defense, due to the difficulty of getting the congestion signature and the lack of forwarding insights provided by the routers.

PAP is similar to the idea of pushing back bad traffic. Compared with [IB02], our approach based on NDN utilizes the victim's feedback and the traffic states provided by NDN at each router would help the system know with certainty the source of the overwhelming traffic. Ultimately, our proposed Pushback learns which clients are not obeying the DDoS control and has the ability to selectively rate limit only those clients.

2.3 Desired Architectural Properties for DDoS Mitigation

Based on the observations of DDoS vulnerability in IP networking itself, [HG04, BCR16] propose to change the existing Internet architecture in order to design greater DDoS resilience at an architectural level. The previous research presents a number of desired features of a DDoS-resistant Internet architecture, some of which include

- Limiting client's reachability of a server based on server's capabilities and wills. For instance, if an end host don't want any traffic, the network should not forward traffic to the host.
- Source address authentication. Source address authentication help the network system to prevent source address spoofing, and thus to prevent reflection attack and amplification attack.
- Separating client and server address space to prevent unwanted traffic. For instance, compared with the traffic from a client to a server, the traffic from client to client and server to server is more likely to be maliciously generated.
- Building symmetric traffic flow and traffic path. This helps to simplify many networking monitoring functions and also to prevent reflection attacks.

In this prospectus, we show that NDN's architectural decisions accord with those features by fundamentally changing the current TCP/IP architecture to using named data as the building block of the network. We elaborate more on this analysis in Chapter 4. We explain how NDN removes some of the inherent vulnerabilities at the core of the current Internet architecture. Furthermore, we utilize NDN's architectural properties to implement hop-by-hop Pushback, enabling PAP to accurately push the offending traffic back and selectively rate limit DDoS suspects.

CHAPTER 3

Named Data Networking

3.1 Basic Concepts of Named Data Networking

Named Data Networking (NDN) makes named data as the centerpiece of the network architecture. To be more specific, applications name their data at the application layer and NDN directly uses the namespace of applications for network layer data delivery. Moreover, names in NDN are semantically meaningful and hierarchically structured, e.g., a video produced by Alice's device may have the name "`/univ1/cs/alice/video/demo.mp4`", where the character '/' delineates name components, similar to URLs today. In NDN, routing and forwarding the packets is based on *name prefixes*. For instance, Figure 3.1 shows an example for an Interest packet to fetch the video. Each forwarder along the path forwards the Interest packet based on the forwarding table (e.g., "`/univ1`" and "`/uinv1/cs`" are prefixes in the forwarding table) using the longest prefix match.

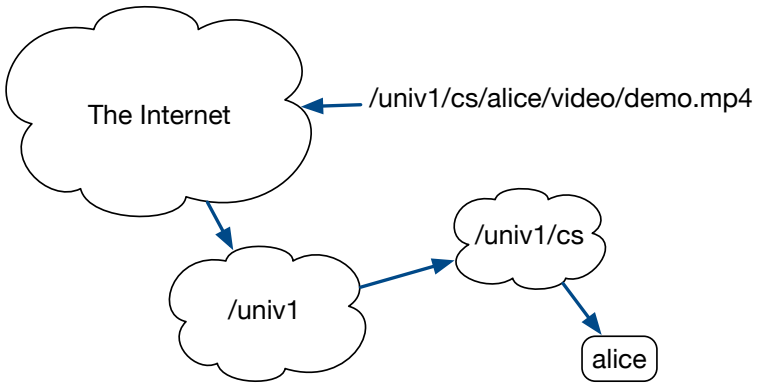


Figure 3.1: Forwarding Based on Prefix

In NDN, applications that produce the data are called *producers* while the ones who consume application data are called *consumers*. Note that an application could be a producer

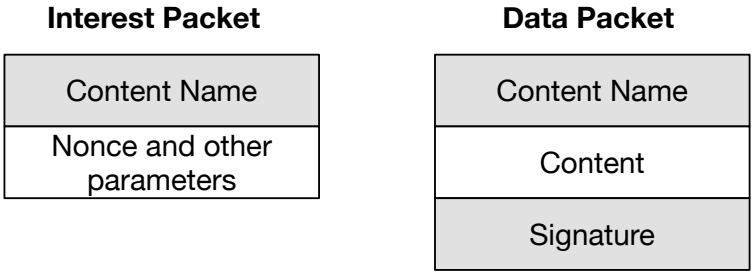


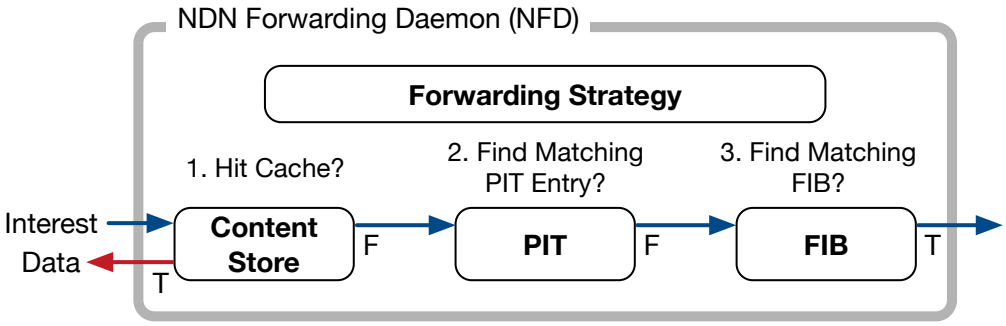
Figure 3.2: Interest packet and Data packet

and a consumer at the same time. Instead of delivering a packet from a source address to a destination address, NDN adheres to using a *pull model* where Data consumers fetch Data from the network in a request/response communication pattern – the request, called **Interest** packet, carries the name of the desired data and fetches the response, called **Data** packet (Figure 3.2). In NDN, forwarders will record the Interest packet along its path to the Data packet and the fetched Data packet will strictly follow, in reverse, the path taken by the corresponding Interest to get back to the requesting consumer.

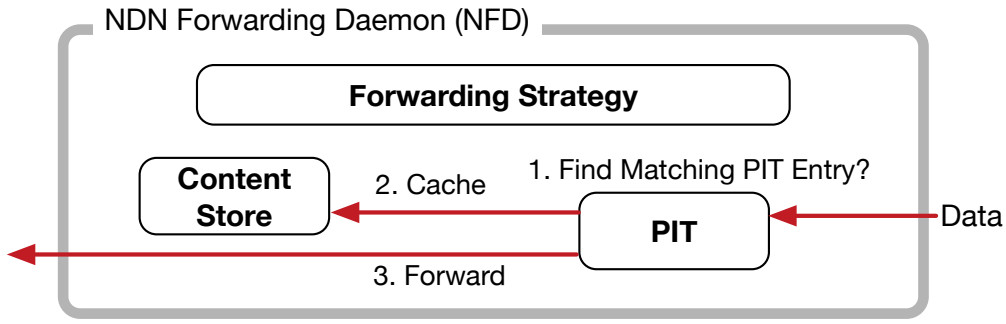
NDN builds communication security [ZZN18] into the architecture by requiring data producers to cryptographically sign all data packets at the time of production and, if needed, encrypting them as well. Securing data packets directly enables routers to cache them as they pass along, and enables consumers to validate Data packet regardless of where and how they are fetched. Importantly, NDN's routing system [HAA13], which is based on NDN's Interest-Data exchange, is also secured such that only an authorized user can register its prefix to a forwarder and the communication between routers is also protected.

NDN Forwarding Daemon (NFD) is an implementation of the NDN forwarding module and each router in NDN runs NFD. It consists of three basic components:

- **Content Store** which is used for Data caching
- **Forwarding Information Base (FIB)** which contains forwarding information, including name prefixes and forwarding interfaces.
- **Pending Interest Table (PIT)** which stores currently unsatisfied Interests and their incoming/outgoing interfaces.



(a) Interest Forwarding



(b) Data Forwarding

Figure 3.3: NDN Forwarding

As shown in Figure 3.3a, when NFD receives an Interest packet, NFD will first check the nonce field in Interest to avoid the loop (omitted in the figure). NFD then looks at the Content Store to see if a desired matching Data packet already exists. If it does exist, NFD will simply return the Data packet on the interface from which the Interest came. Otherwise, the Interest is checked against entries in the PIT. If there is an existing pending/unsatisfied Interest with the same name, the router adds the incoming interface of this Interest to the existing PIT entry and no further forwarding is done. Otherwise, a new PIT entry will be created recording the incoming interface(s) of the new Interest. Forwarding strategy, the decision maker to decide whether, when, and where to forward the Interests, will then determine how to handle the packet. If the forwarding strategy decides to forward packet, the outgoing interface will be recorded in the PIT entry. If it decides to drop the Interest for some reason (e.g., the upstream link is down, there is no forwarding entry in FIB, or extreme congestion occurs) the forwarder can send an Interest NACK to its neighbor(s) it

received the Interest from. NACK is an NDN hop-by-hop feedback mechanism used to report a problem in further forwarding of an Interest. When routers receive such NACK packet, an appropriate action gets triggered in the router based on reason code used in the NACK packet. Note that an Interest NACK is different from an ICMP message; the former goes to the previous hop while the latter is sent to the source host.

As shown in Figure 3.3b, when NFD gets a Data packet, the Data name is used to lookup the PIT for the corresponding Interest entry. Once the matching entry is found, NFD sends the Data packet to all the interfaces from which the Interest was received and removes the entry from the PIT. In case the corresponding PIT entry is not found, the router will simply discard the Data packet. Data will also be cached in NFD's Content Store for future Interests.

3.2 Existing Solutions to NDN Interest DDoS

There have been various existing proposed approaches to mitigate Interest DDoS over NDN. Specifically, [AMM13, CCG13] leverage NDN's stateful forwarding, from which one can compute the "success ratio"(how many Interests get satisfied by Data) to indicate whether there is a fake Interest DDoS or not. [DWF13, SS16] propose approaches to detect Interest flooding by monitoring the PIT size or PIT utilization rate. Also, the previous work mainly focuses on one specific type of attack – fake Interest attack (i.e. Interest towards non-existent data that can never be generated). Moreover, given that routers are the entities that detect attacks, proper threshold values for the detection function are required to be set in advance. These values directly affect the detection accuracy and may be non-trivial to configure when underlying traffic composition is complex.

In comparison, our proposed mechanism takes explicit feedback from the victim and perform accurate Pushback and selective rate limiting to misbehaving consumers. Also, PAP is able to handle valid Interest (i.e. Interest towards existent or dynamically generated data) DDoS and mixed Interest attack scenarios where attackers can send both fake and valid Interests towards the target.

CHAPTER 4

NDN's Architectural Properties for DDoS Defense

In this section, we analyze how NDN's architecture can lead to a inherent resilience to DDoS and provide a better foundation for DDoS defense mechanisms as compared with the current TCP/IP architecture.

4.1 Off By Design

The communication in NDN follows a pull model and an application or a node is considered to be "off by design" for the following reasons: 1. There is no way for a Data packet to come in if there is no Interest for the corresponding name because there is no corresponding Interest path. 2. One Interest can at most bring one Data packet back 3. One cannot send an Interest to an consumer application or an producer application whose name is unreachable from the sender. By simply not announcing the prefix to solicit Interest packets, an application can never be reached by an Interest packet, thus reducing the attack surface for DDoS attackers. With the pull model, an attacker can never DDoS an NDN node by flooding Data packets, thus the network layer DDoS attack can only be carried out by Interest flooding. As Data Packet follows the reverse path to its corresponding Interest, an attacker cannot redirect it to another consumer. Therefore, NDN eliminates reflection DoS attack and distributed reflection DoS (DrDoS) attack.

4.2 The Barrier of Establishing Botnet

Interest packets are forwarded to end hosts based on their names. Unlike IP address which is numerical and of a fixed format, the names under a given prefix in NDN are **not enumerable**. When a producer's name prefix is not widely known (e.g., only known to local network), it is difficult for an outside attacker to "guess" the exact prefix owned by the target host. If Interest name does not match a specific prefix in the forwarding table, the packet will get dropped by the router. As a consequence, an attacker needs to spend more effort to communicate with a host to compromise it in NDN, unlike in TCP/IP where establishing a botnet could simply be done by enumerating IP addresses [AAB17b]. For instance, to reach a producer Alice whose routable prefix is "`/univ1/cs/alice-lastname`", an attacker definitely needs auxiliary information to guess Alice's name prefix. For example, the attacker must know the naming convention of "`/univ1/cs`" (instead of "`/university1/computer-science`"), and also Alice's full name or data name of previous data produced by Alice.

4.3 In-network Cache of Named Data

NDN's content-centric communication model provides enhanced data availability by enabling cache (e.g., the content store in NDN Forwarder) inside the network. Because of the in-network cache in NDN, Data packets carrying static content (e.g. HTML files, CSS files, images) can be cached by routers to satisfy future Interest packets, thus reducing the number of Interests reaching the producer (victim). As shown in previous work [PCP12] and our simulation results in Section 7.1.1, in-network cache can help to mitigate the Interest flooding where attackers send Interests for static or existing Data packets. However, since cache has limited capacity, when attackers are able to send Interests for a large number of Data packets, the effect of cache becomes smaller because the chance for an Interest to hit a cached Data is lower.

4.4 Stateful Forwarding: Interest Aggregation

In NDN, Interests targeting the same piece of the named data will be aggregated by the router and later Interest packets will not be sent out as discussed in section 3. When the corresponding Data is fetched, the router will send a copy to each incoming interface as recorded in the PIT entry. This feature makes it harder for DDoS attackers to flood the same Interest packet or a small set of Interest packets towards the producer. However, if attackers flood a target prefix with a large set of Interest packets or even fake Interests with randomly generated components, Interest aggregation will not be helpful because the chance for a later Interest to hit an existing record becomes very low.

4.5 Stateful Forwarding: Rich Feedback

NDN's stateful forwarding [YAM13] provides rich insight of the ongoing traffic. By observing each Data packet and its corresponding pending Interest entry in the PIT, an NDN forwarder is able to measure the round-trip time, throughput and name reachability of each outgoing interface. As mentioned in [AMM13, CCG13], a forwarder can also learn the Interest satisfaction ratio, namely the proportion of Interests that successfully fetched a Data packet, and thus detect possible fake Interest DDoS attack. Moreover, PIT entry timeouts also offer relatively cheap DDoS attack detection as mentioned in [ZAB14].

As mentioned before, the Pushback in TCP/IP is non-trivial because of stateless forwarding; a router has little knowledge on which downstream interface to use to relay the Pushback (source IP addresses are not trustworthy because of the IP spoofing). In NDN, stateful forwarding, by design, helps forwarders to know exactly which interface the traffic is coming in from and this helps it traceback to misbehaving consumers/attackers and reinforce congestion control.

4.6 NDN Congestion Control

In TCP/IP, congestion control is end-to-end and at the transport layer, where routers and middleboxes have no idea if a sender is abiding by congestion control scheme or not (e.g., DDoS/DoS attack). As a result, attackers can easily flood packets without being punished. Hence, it is impossible for the current congestion control to prevent overwhelming traffic in case of DDoS attack. The network should have the traffic under control independent from whether all hosts behave well. Enabled by the stateful forwarding, NDN builds congestion control at the network layer in a hop-by-hop fashion, thus can help prevent the congestion caused by DDoS. However, congestion control cannot differentiate legitimate traffic from DDoS traffic. Therefore, defense mechanisms are required to specifically handle the DDoS attack.

4.7 Built-in Security in Routing System

As explained in section 3, NDN's built-in routing security allows for authenticated routing announcements. In NDN's routing system, when a user registers a prefix to the routing system, the user needs to prove their ownership of the prefix or if they are authorized to register such prefix, mitigating DDoS attacks based on route hijacking and cache poisoning [GTU14]. For instance, an attacker cannot register a cannot register a prefix under fake identity (e.g., pretend to be /com/google) unless the attacker compromises the routing system or steals the corresponding digital keys. Moreover, routing messages exchanged among the routers are also protected: every Data packet will be signed and routers will only accept the routing announcements signed by trustworthy parties.

4.8 A Summary

NDN's architecture, by its design, leads to DDoS resilience as follows: First, NDN's Interest-Data packet exchange eliminates reflection attack and prevents DDoS attack by flooding Data packets. Second, it becomes harder for attackers to build the "zombie army". Also, NDN

can mitigate DDoS by route hijacking and cache poisoning by its secured routing system and relieve the overload caused by Interests for static and existent Data by Interest aggregation and cache. From the perspective of DDoS defense, NDN provides rich insights of the ongoing traffic. Congestion control can also help to mitigate some congestion when DDoS happens.

One possible attack that NDN architecture does not have a built-in mitigation for is Interest flooding and PAP is to mitigate such type of attack.

CHAPTER 5

Threat Model, Goals, and Assumptions of DDoS Mitigation

In the following section, we use the topology shown in Figure 5.1 to help illustrate our design. In this toy topology, the server (runs NDN producer) is the target of the DDoS attack; it serves Data under the prefix "`/univ1/service/email`" and "`/univ1/service/video`". There are four clients (runs NDN consumers) in the topology where A, B, and C are DDoS attackers while D is a legitimate client. R1, R2, and R3 are three routers that run NDN forwarder. In this prospectus, based on the Data flow, we call the routers towards the server "upstream routers" and routers towards the clients "downstream routers", e.g., R2 and R3 are R1's downstream routers.

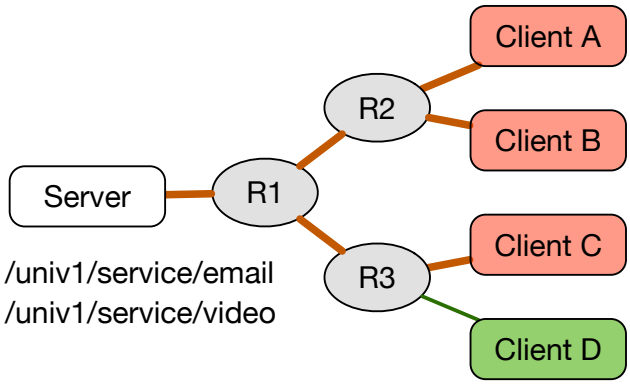


Figure 5.1: A Toy Topology

5.1 Threat Model

DDoS attackers may attempt to attack a target by flooding Interests. Specifically, as pointed by the work [GTU13], we categorize Interest packets sent by attackers based on their target Data packets as follows:

Type A **Interests for static or existing Data packets** This kind of Data packets (e.g., CSS file, video file) can be cached and used to serve future requests.

Type B **Interests for non-existent Data packets that cannot be generated** When attackers send Interests to non-existent Data packets or the Data packets can never be generated, Interest packets cannot be satisfied. This type of Interests is mostly likely to be generated maliciously. One possible way for attackers to generate Type B Interests is to append non-existent name components to valid server prefixes so that the Interest can successfully arrive at the target server.

Type C **Interests for dynamically-generated Data packets**
When an Interest for a dynamically-generated Data arrives, the server needs to process the request (e.g., database query, calculation) before it can generate the Data and reply back. For example, a legitimate Interest name may contain variable components and thus there are no currently existing Data packets to match it. In this case, the server will need to process the request first and then generate a Data packet following with a reply.

Considering whether an Interest packet can finally fetch a Data packet or not, we say the Type A and Type C Interest packets as valid, while Type B Interests as fake. We assume that attackers can generate both valid and fake Interest packet to flood a target server.

5.2 Goals

Type A attack can be naturally mitigated by NDN's Interest aggregation and in-network cache as we discussed in Section 4.3 and Section 4.4. However, as discussed and shown in

Section 7.1.1, when attackers can send Interests to a large number of different names, the effect of Interest aggregation and in-network cache become negligible and countermeasures must be implemented. In contrast, Type B and Type C Interest names would be generated arbitrarily for each packet, and thus hardly any Interests arriving at an NFD would hit an existing PIT entry or a previously cached Data packet. Therefore, NDN architecture itself does not effectively mitigate such attacks.

Our proposed system aims to defend not only from **fake Interest flooding** (Type B), but also from **valid Interest flooding** (Type A and Type C) and **mixed Interest flooding** (mix of Type A, B, and C) in an effective way without affecting legitimate clients and the traffic under another prefix. For example, in our toy topology, when the email service "`/univ1/service/email`" is under attack, the Pushback should be able to limit the DDoS traffic towards the email service only, at the same time, traffic from legitimate client D and the traffic under other prefixes (e.g., "`/univ1/service/video`") should not be affected.

5.3 Assumptions

Our DDoS mitigation solution is based on the following assumptions and we briefly argue that all these assumptions are either reasonable or easy to be realized.

- The server (the victim of DDoS) knows whether it is being flooded and which prefix is under attack. Also, the server can judge whether an Interest packet is a fake or valid. The assumption is reasonable because the victim server inherently has the most accurate feelings and judgments of a DDoS attack.

- We assume that the server knows its available capacity to process incoming Interests at a given time. It is trivial for a server to learn about its own capacity (e.g., based on memory, CPU utilization, etc.) and its setting of the fake Interest tolerance (i.e. how many fake Interests a server can tolerate).

- Core routers are not malicious, and they will help to mitigate the DDoS traffic. Regarding this assumption, it is reasonable because legitimate network providers does

not want DDoS traffic.

- A router knows whether an interface connects to an end client or another router. The information can be obtained by multiple means, for instance, to learn whether an interface is connected to a client or not, the router can check the hop count of incoming packets from that interface, or check whether there are routing protocol messages being passed; the information can also be manually configured by its local autonomous system.

- We ignore losses of NACK packets (PAP only uses NACK to convey DDoS information). NACK is for a one-hop communication, and because of the soft state of our defense mechanism, if a NACK gets lost, the Pushback will not take place and thus the upstream will continue sending NACKs. To reduce the NACK packet losses, servers and routers can also prioritize NACK packets or add redundancy.

CHAPTER 6

Producer-assisted Pushback: a Preliminary Approach

6.1 Design of PAP

In this section, we first provide the design description of PAP and then explain our design choices in the rest part.

6.1.1 A Design Overview

PAP is designed to be running on each router as a part of the **Forwarding Strategy**. A Pushback is triggered by victim's NACK packet (Figure 6.1). To be more specific, a NACK packet created by the victim server and sent downstream will carry the following information:

R The **reason** code used to notify router whether it is Fake Interest Attack or Valid Interest Attack/Overload.

P The **prefix** under which the overwhelming traffic comes to the victim.

T/C The receiving rate of Fake/Valid Interests that the server can handle currently under the prefix P. We call the number **tolerance** [1] in case of fake Interest attack or **capacity** in case of valid Interest overload. If an application has a zero tolerance for the fake Interest, it can set T = 0.

FL **Fake Interest name list** for fake Interest attack only. The list contains all or sampled Fake Interest names (excluding prefix) under the prefix P that a server received recently (e.g., within the last unit time interval defined by the server).

[1] Given that even legitimate clients may send fake Interests by accident, PAP also allows the application to define its tolerance on the number of fake Interest packets received

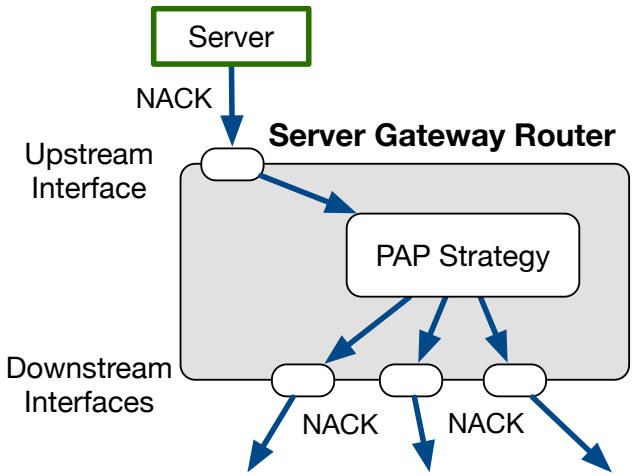


Figure 6.1: PAP System Overview

As shown in Figure 6.2, after receiving a PAP NACK, the router will first check the R field carried in the PAP NACK. If the reason is fake Interest attack, the router will check the FL and find the corresponding pending Interests from PIT. If the reason is valid Interest attack, since attackers also send valid Interest packets, the router cannot distinguish the legitimate traffic from the offending traffic, hence the router will check all the current pending Interests under the prefix P. From those pending Interests, the router will get exact information of their incoming interfaces and perform the **accurate Pushback**. According to the type of the attack, the router will calculate a weight for each of those interfaces based on the number of incoming Interests through the interface. Then the router will use the weight and the total assigned tolerance/capacity derived from the T/C field of the NACK to get a weighted tolerance/capacity for each interface. After that, as shown in Figure 6.1, the router sends each of those interface a new NACK carrying the weighted tolerance/capacity as T/C and a pruned FL if the reason R is fake interest attack.

In this way, all the routers along the Interest sending path will receive and generate new NACKs that will be propagated to further downstream routers. Finally, the rate limiting request that originated from the server will arrive at all gateway routers of the suspected clients.

Different from upstream routers, each gateway router that received a NACK from the upstream will also perform rate limiting. After calculating the weighted tolerance/capacity

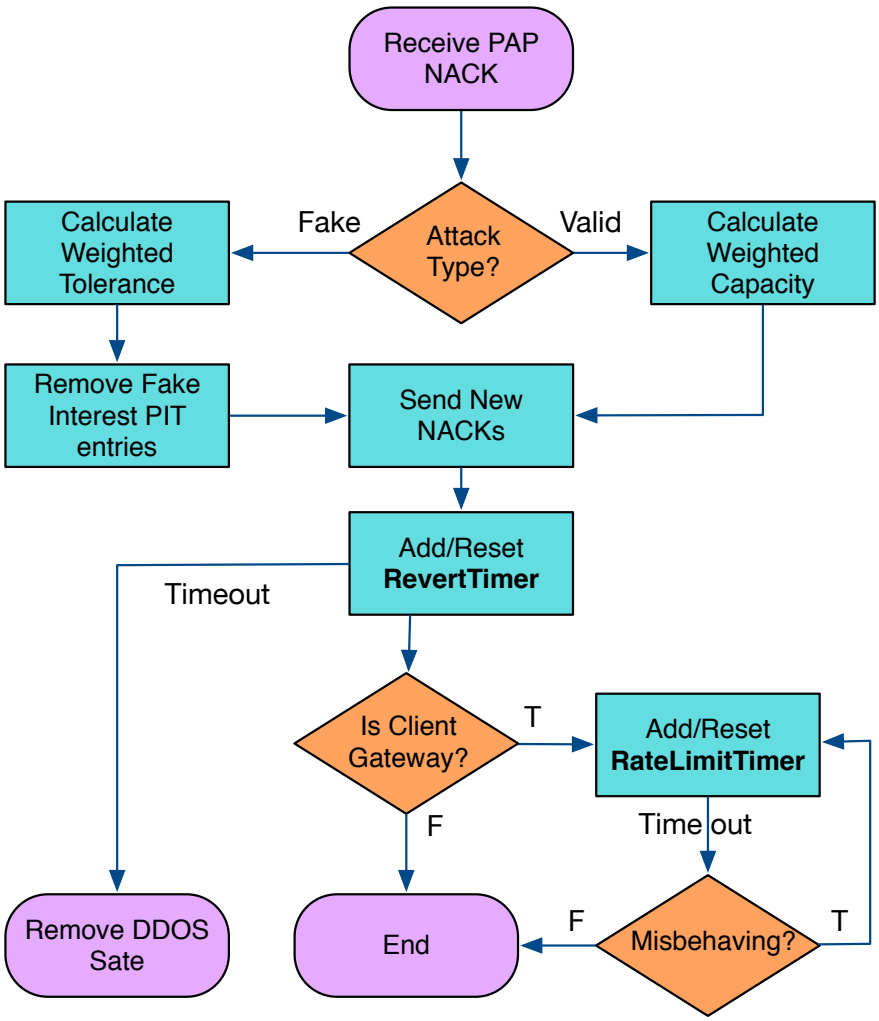


Figure 6.2: Process Logic After Receiving A NACK

for each client interface and sending new NACK packets to each suspect client, the router will begin rate limiting. To be more specific, for a client interface whose weighted tolerance/capacity is larger than zero, the router will randomly drop Interest packets under prefix P in order to control the sending rate from the interface to be less than or equal to the weighted tolerance/capacity. While rate limiting, gateway routers will also monitor the behavior of its clients. After receiving a NACK from the gateway router, legitimate clients will comply and lower down their sending rate of the Interests under the prefix P, while the bots may not obey the rules. As a consequence, the router can perform **selective rate limiting**: the router can notice the misbehaving individuals and further restrict the limit, while for good clients, the router can relax the limit.

In PAP, all the routers maintains a **RevertTimer**, and client gateway routers will maintain an additional timer called **RateLimitTimer**. Notice that both timers do not affect the final result of the Pushback and different routers can set these two timers differently based on their own needs.

- RevertTimer decides how long a router should keep the DDoS Records. Whenever a new NACK arrives, the router should check whether there is an existing RevertTimer for the Pushback with the same reason and same prefix. If yes, the router should update the timer; otherwise, the router should create a new timer.
- RateLimitTimer decides how long it takes for a router to decide whether a individual is well-behaving or not. After the RateLimitTimer, the gateway router will remove the limit of those "good" clients and strengthen the limit of bad ones and reset the RateLimitTimer. Therefore, this timer is periodic until all limits are removed.

Since the design is built to handle Pushback for a variety of prefixes and types of (Fake/Invalid) Interest Attacks, PAP is able to handle complex scenarios where the victim servers may send multiple NACKs containing different prefixes and different reasons.

From the design description, we can see PAP directly makes use of NDN's properties to combat DDoS: 1. Utilizing structured names at the network layer allow routers to perform per-prefix Pushback and monitoring. 2. Stateful forwarding provides exact information of the traffic so that PAP can accurately identify suspect clients.

6.1.2 Pushback Triggered by the Victim

Starting DDoS countermeasures by routers' detection of the underlying traffic poses a risk of denying services to good clients. Usually, such detection requires pre-configured threshold values as discussed in Section 3.2. Moreover, routers need to learn about the server capacities in advance so that defense mechanisms will not control the traffic too strictly (no traffic to the server at all) or too loosely (permitted traffic still overwhelms the server). Therefore, in the PAP system, we let the victim server be responsible for detecting a DDoS attack and

sending PAP NACKs downstream to trigger Pushback. Thus, our system does not require any pre-configured thresholds in routers.

Based on the information carried in PAP NACK packets, the prefix P helps PAP narrow the target traffic scope. The reason R will help PAP to learn the attack type and take different reactions accordingly. The capacity C or tolerance T of the PAP NACK can inform the downstream routers about the percentage volume of traffic that should be controlled. In case of fake Interest attack, the FL can explicitly convey the information of the attack traffic, thus the routers can directly identify the inspect interfaces.

Another benefit of PAP is that the routers don't need to maintain any additional information if there is no reported DDoS attack from servers; all actions in PAP are triggered only after a PAP NACK from the victim is received.

6.1.3 Per-Prefix Pushback

When a server identifies an attack on one of its prefixes, PAP is able to control the flow of Interests for that particular prefix without affecting the traffic under other prefixes. To be more specific, by providing routers the exact prefixes that are under attack, only traffic moving towards those prefixes will be rate limited. This feature is enabled by the hierarchical structure of Data names.

For example, consider the server in Figure 5.1 and a client A who is now accessing server's video service "`/univ1/service/video`". A is a good citizen without malicious intent. However, A doesn't know that its laptop has been compromised and there is a malicious program (a bot) running which is used to carry out DDoS attack on server's email service "`/univ1/service/email`". In TCP/IP, DDoS mitigation may simply filter out Alice's packets whose destination is server's IP address because there is no finer granularity (routers cannot understand better granularity). Obviously, in this case, the defense will also deny the service of the video service. However, in our case, the victim can clearly say it is "`/univ/service /email`" that is under attack and so the downstream routers will only limit the traffic directed to the email service. With per-prefix Pushback, only bot traffic from client A will be blocked

and A can continue using the video service without being affected.

6.1.4 Rate Limiting at Client Gateway Routers

In our design, only client gateway routers (edge routers) play the role of rate limiting. This is because, on one hand, we cannot trust a client's device to take actions - it could be compromised as well. On the other hand, an upstream router should not perform rate limiting for the following reasons: 1. When the traffic volume under a target prefix increases, upstream routers cannot tell whether it is because of the misbehaving downstream routers or because new clients have joined. 2. When legitimate clients are behind a downstream router, upstream router actions will also hurt legitimate clients.

6.1.5 Selective Rate Limiting by Client Monitoring at Client Gateway Routers

After a client gateway router sends out PAP NACKS to suspect clients, we believe that the legitimate clients will obey the DDoS control and lower their sending rate of Interests accordingly while attackers may not abide by this. This is when client monitoring comes into the picture. In PAP, after starting rate limiting, a client's gateway router will start checking whether a client has changed its behavior following the NACK or not, i.e., whether it lowers down its sending rate to the required value under the specified prefix. Accordingly, the router can relax or increase the limit. In this way, PAP is able to ensure that eventually almost all the server's incoming traffic is from legitimate clients after several rounds of monitoring. Notice that RateLimitTimer decides the time of each monitoring.

It is possible that the bots may intelligence to analyze and attempt to circumvent the NACK, but PAP already succeeds if the bots cannot increase Interest sending rate, thus greatly reducing the damages. Essentially, PAP forces bad entities to comply. Since each router starts the limit at a different time because of the various round-trip time (RTT) between the victim server and each client gateway router, even though an attacker can reset its sending rate after the router removes the limit, there are no longer bursts of traffic and the attack becomes less harmful.

6.2 Implementation of PAP

We implement PAP [ZV18] in C++ over ndnSIM [AMZ12], which is a NDN simulation platform based on NS-3.

6.2.1 Accurate Pushback by Stateful Forwarding

The symbols used in this section are listed in Table 6.1.

Table 6.1: Notation Table

w_i	weight for the ith interface
NL	number of names listed in the NACK's FL
NP	number of all matched PIT entries
NF_j	number of incoming interfaces in the jth PIT entry
T_i	weighted tolerance for the ith interface
C_i	weighted capacity for the ith interface
T_{NACK}	tolerance value derived from the PAP NACK
C_{NACK}	capacity value derived from the PAP NACK

Stateful forwarding provides accurate information about which Interest is from which downstream interface. In case of Fake Interest attack, the FL carried in a NACK helps routers to know exactly the interfaces from which listed Interest packets are coming from and thus can traceback to attackers, avoiding the "guesswork". Legitimate clients who are sending valid Interest will not be affected. More precisely, a router can know per prefix traffic from each incoming router interface and calculate the *weighted tolerance* for each face. To calculate the weight w_i for face i, we have

$$w_i = \frac{1}{NL} \sum_{j=1}^{NP} \frac{1}{NF_j} \tag{6.1}$$

To calculate the weighted tolerance T_i for the relayed NACK for face i, we have

$$T_i = T_{NACK} w_i \tag{6.2}$$

For example, if a router receives a NACK with Fake Interest tolerance 50 and router observes (through the PIT entries and the FL in NACK) that 20% of the total fake Interests came from interface A while rest 80% came from interface B, the weighted tolerance for interface A will be 10 and that for interface B will be 40. This is fair because 80% of the Fake Interest traffic comes from interface B and so we want to drop more packets coming along that path than interface A. Notice that all these calculations are specifically for the prefix P. All fake Interest PIT entries are removed after sending the NACK downstream.

As for valid Interest Attack, a NACK doesn't have FL. Even with an explicit list of valid Interest Names, the router cannot differentiate Interest packets sent by attackers from those from legitimate clients. In this case, PAP will first check all matched pending Interests and simply Pushback to all clients who have sent Interests under prefix P because at this moment, PAP has no clue which clients are legitimate. To calculate the weight w_i for face i, we have

$$w_i = \frac{1}{NP} \sum_{j=1}^{NP} \frac{1}{NF_j} \tag{6.3}$$

where j is matched PIT table entry whose name is under prefix P and incoming interfaces contain face i. To calculate the weighted capacity C_i for the relayed NACK for face i, we have

$$C_i = C_{NACK} w_i \tag{6.4}$$

After the Pushback to all clients, client performance monitoring (Section 6.2.3) will help to further identify attackers from legitimate clients. Unlike in fake Interest Attack, routers will not remove any PIT entry after sending the NACK downstream.

6.2.2 Limiting the Size of PAP NACK

One obvious question is how to efficiently send a NACK containing a list of thousands of names to downstream routers. Two techniques could be used to address this issue: Bloom filter and Sampling.

Bloom filter Bloom filter [Blo70] is space-efficient and fits perfectly in our scenario. The victim server can add fake names into a bloom filter rather than to a list data structure

and then replace FL with the bloom filter in the NACK to be sent. In this case, the PAP NACK packet size will be constant instead of $O(n)$ where n is the size of FL. After gateway router receives the NACK, it can easily test whether a pending Interest is in the bloom filter by doing non-cryptographic hash of the Interest name. Even though bloom filter is a probabilistic data structure, we can achieve an acceptable error rate by properly setting the length of the filter.

Sampling The goal of the name list is to notify the router which Interests are fake so that the router can check states of those Interests and identify upstream interfaces to further send NACKs. Therefore, it is sufficient for the victim to sample the name list because the sampled name list is sufficient enough to reveal those downstream interfaces. If the victim samples the list in a totally random way, the proportion of Interests from the same face will keep the same as the list that is not sampled, which will not change the Pushback weight distribution of interfaces at routers.

6.2.3 Selective Rate Limiting

When a PAP NACK arrives at a client gateway router, the router will calculate the weighted tolerance/capacity for each involved end hosts according to Equation 6.2 and Equation 6.4. For any end host who has a weighted tolerance/capacity, the router will randomly forward only a permitted number of Interest packets under specified prefix upstream. This permitted number is the value of weighted tolerance/capacity.

When both Valid Interest Attack and Fake Interest Attack happens, the client gateway router will take the smaller limit of the two calculated using the Fake Tolerance and the Valid Capacity; the algorithm is shown as Algorithm 1. Whenever tolerance/capacity is in a fraction, we randomly round up or round down to an integer using Algorithm 2.

As discussed in Section 6.1.5, a legitimate client is supposed to adjust its sending rate to be lower than the tolerance/capacity value when they receive a NACK. If a client limits its sending rate accordingly, after a predetermined time period, the router will remove the limit on the corresponding interface; otherwise, the router will reset the timer and halve the

Algorithm 1 Rate Limiting on Interface

```
function doRateLimit(ValidDDoSRecord, FakeDDoSRecord, Face)
    if ValidDDoSRecord.limitList.contains(Face) then
        Limit_1 = ValidWeight[Face] × tolerance
    end if
    if FakeDDoSRecord.limitList.contains(Face) then
        Limit_2 = FakeWeight[Face] × capacity
    end if
    return min(Limit_1, Limit_2)
end function
```

Algorithm 2 Precise Limit Rounding

```
function limitRounding(Limit)
    int l-int = getIntegerPart(Limit)
    double l-fractional = getFractionalPart(Limit)
    Limit = l-int + rand() < l-fractional? 1:0
    return Limit
end function
```

permitted Interest number. This applies to both Fake Interest Attack and Valid Interest Attack. Algorithm 3 shows how client monitoring works under fake Interest Attack. The algorithm is the same in case of valid Interest attack.

6.2.4 Timers in PAP

If a router receives a NACK again before the RevertTimer expires, it does not recalculate the weighted tolerance for the faces for whom the information about this prefix (included in the NACK) is already stored in the router (in DDoS Records). The reason is to avoid “overlooking" those attackers who are already being limited caused by earlier NACKs; think of the scenario where few attackers are closer to the victim than the other attackers in our toy topology 5.1. We assume that client A and B are closer to the server compared to

Algorithm 3 Selective Client Monitoring

```
function onLimitTimeout(FakeDDoSRecord)
    for any inFace in the FakeDDoSRecord.limitList do
        if FakeDDoSRecord.followDDoSControl(inFace) then
            FakeDDoSRecord.removeLimit(inFace)
        end if
    end for
    if FakeDDoSRecord.limitList is empty then
        Router.removes(FakeDDoSRecord)
    else
        tolerance /= 2; Reset timer
    end if
end function
```

client C and D. When routers receive the first NACK from the server who is under a fake Interest attack, R2 will limit the traffic from A and B while R3 will limits C (D is sending valid Interests). Because A and B are closer to the server, R2 will start rate limiting first. Before R3 receives the NACK, the server sends the second NACK and since now A and B are under control, the FL in the second NACK may not contain fake Interests from A and B. In this case, if routers recalculate weighted tolerance for all the faces after receiving the second NACK, there will not be any limit on A and B anymore, allowing attack traffic again from them. To avoid such situation, when new NACK arrives in a short time (RevertTimer) before the previous one, a router will keep the existing weights (and limits) and only calculate weighted tolerance/capacity for interfaces which do not have a weight before.

CHAPTER 7

Evaluation of PAP

The network topology that we use for our experimentation and evaluation is shown in figure 7.1, where there are four Autonomous Systems (ASes) with meshed connections. The nodes "`/univ1/cs/server`" is the target server and it belongs to "`/univ1/cs`" subnet. We assume the server is globally reachable, which means all users, even outside the server's local network, have means to learn the name. For sake of simplicity, we will just call the server *univ1-server* in rest of the section. In our experiments, we simulate DDoS attacks on univ1-server and there are 60 attackers located across all the ASes shown in the figure. We also add 12 legitimate clients into the topology. Each attacker's Interest sending rate is 100 Interests/s while legitimate clients will send 20 Interests per second.

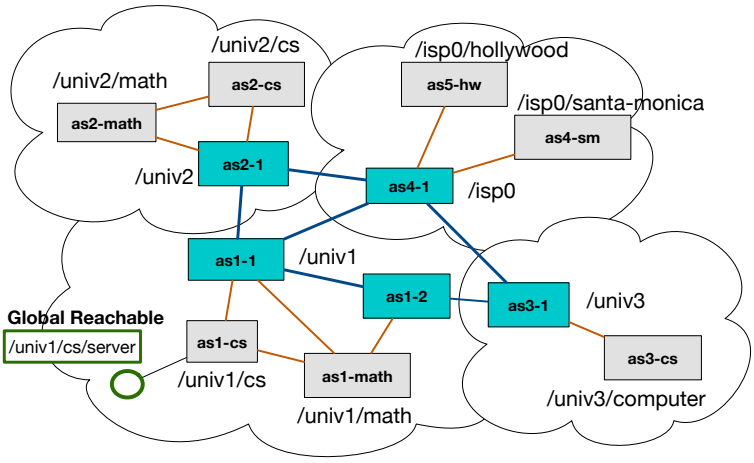


Figure 7.1: Four Meshed ASes Topology

Our results are shown in the figures shown in this section, where in each figure, the red line with star points represent the traffic sent from **attackers**, the green line with circle points represent the traffic sent by **legitimate clients**, the yellow line with triangle points

represent the traffic that received by the **server gateway router**, and the blue line with rectangle points represent the traffic that finally arrives at the **victim server**. The attackers will start flooding Interests from the third second.

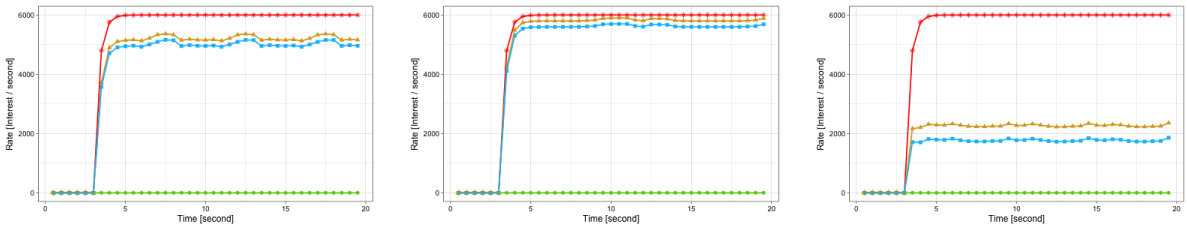
We first demonstrate NDN's DDoS resilience to Type A Interest attack with the help of Interest Aggregation and caching. After validating that NDN's architectural components do help mitigate the traffic overwhelm right from the start of a DDoS attack, we then move on to evaluate the performance of PAP, our proposed approach to fight against DDoS in NDN. PAP will get triggered when DDoS attack reaches a scale where Interest Aggregation and caching will have no significant effect on attack traffic. To simulate this, we'll simply disable caching in our experiments and choose a large range of available Data names to reduce the effect of Interest Aggregation. We evaluate PAP for all three types of attacks: Fake (Type B) Interest flooding, Valid (Type A and Type C) Interest flooding and Mixed Interest flooding where the attackers will flood the target with both fake Interests and valid Interests. We also evaluate PAP when there are more than one victim servers that are under attack. The simulations results show that after the DDoS starts, PAP can effectively control the traffic to the victim as expected within seconds (less than 2 seconds under our simulation settings), and ensure that over 99% of the attack target(s) incoming traffic is from legitimate clients (about 10 seconds under our simulation settings).

7.1 NDN's Inherent DDoS Resilience

7.1.1 Interests Aggregation and Cache in Type A Interest Attack

Since there is no DDoS defense mechanism deployed yet, there is difference between attacker and legitimate clients and thus we disabled legitimate clients for this simulation. We have each attacker send Type A Interests towards univ1-server at the rate of 100 Interests/s.

We first disabled cache in all routers so that the result will only be affected by Interest Aggregation. As shown in Figure 7.2a, Interest Aggregation can withhold traffic from attackers to the server, saving the victim from being overloaded. We then introduced cache to see



(a) Range=500, Cache=0 (b) Range=1000, Cache=0 (c) Range=500, Cache=200

Figure 7.2: Interest Aggregation and In-network Cache

how it can suppress traffic even more. As shown in Figure 7.2c, compared with the scenarios without cache (Figure 7.2a), it is apparent that the number of Interests reaching as1-1 and univ1-server decreases as we increase the caching capacity (i.e. NFD's Content Store size), which is because intermediate nodes along the path will serve future same Interests with cached Data (the freshness of cached Data is 4 seconds in our simulation).

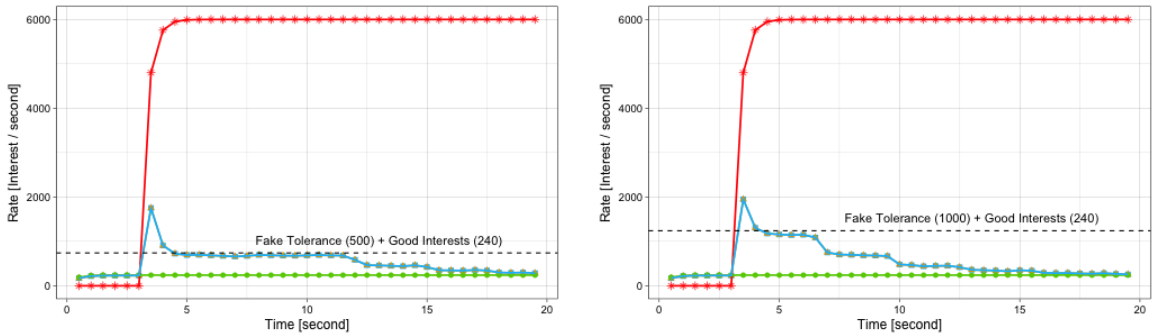
The difference between Figure 7.2a and Figure 7.2b indicates that the effect of Interest Aggregation and cache is lower when an attacker uses a bigger set of Interest names. This is because larger the set, smaller the chance of two Interests carrying the same name and smaller the chance to hit a previous cached Data packets.

7.2 Evaluation of PAP

7.2.1 PAP: Fake Interest Attack

We first study PAP's performance against Fake (Type B) Interest flooding attack. We use the same toy topology as shown in Figure 7.1 and still we have 60 attackers. Notice that we also let 12 legitimate clients send valid (Type A or C) Interests with a reasonable sending rate of 20 Interests/s. We set the server's Fake Interest Tolerance (T) to be 500 and 1000 Interests/s for Figure 7.3a and Figure 7.3b, and we set the RateLimitTimer to be 3 seconds.

As shown in Figure 7.3a and Figure 7.3b, initially, there are only legitimate clients sending Interests in the scenario. After 3 seconds, attackers start their attack by sending Type B Interests at the rate of 100 Interests/second. Without any defense mechanism, univ1-server



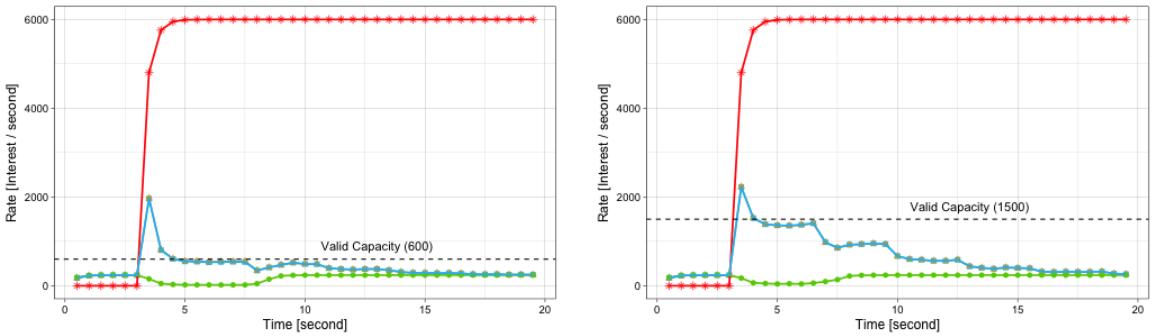
(a) Fake Interest Attack, T=500 (b) Fake Interest Attack, T=1000

Figure 7.3: Fake Interest Attack after Employing PAP

should now receive over 6000 Interests every second. By employing PAP, the attack traffic reaching the sever quickly goes down within the tolerance (500/1000) plus the Interests from legitimate clients (240) as depicted by the plot, meaning traffic from legitimate clients doesn't get affected. The results demonstrate that PAP performs accurate Pushback.

Importantly, as shown, the traffic on victim drops periodically, which confirms the selective rate limiting function of PAP: PAP will detect who is not abiding by the DDoS control placed on sending rate and strengthen their limits. Theoretically, the time period is supposed to be 3 seconds by our setting of RateLimitTimer. Figure 7.3b follows the theory, but Figure 7.3a's victim traffic remains constant for a longer time (about 10 seconds). This happens because client gateway routers receives further NACKs during the effect time of PAP after the first few NACKs. Later NACKs will reset the timer and thus defer the "drop". The reason why there are later NACKs is because of the rounding explained in Algorithm 2. After the first few NACKs triggered by the server, ideally, the traffic should be under control. However, because of the rounding, it is quite possible that majority of the fractional tolerances/capacities are rounded up (as explained in 6.2.3) and so overall traffic violates the set thresholds again. To avoid this, victims can actually set the tolerance in the NACK they send to be a bit lower than what it can actually tolerate or handle.

Once client gateway router stops receiving further NACKs (no timer reset) and since attackers won't limit their sending rate, the tolerance is halved every 3 seconds until it finally merges legitimate client traffic line, meaning all Interests received by the victim are



(a) Valid Interest Attack, C=600 (b) Valid Interest Attack, C=1500

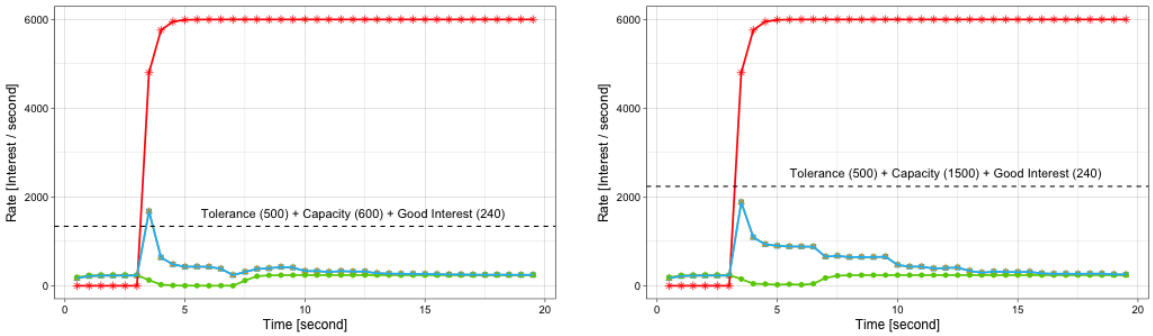
Figure 7.4: Valid Interest Attack after Employing PAP

only from legitimate clients.

7.2.2 PAP: Valid Interest Attack

We use the same topology and simulation settings as we simulate Fake Interest attack, but now attackers send valid Interests to univ1-server. The simulation results of the valid Interest Attack are shown in Figure 7.4a and Figure 7.4b, where we set the server's capacity of handling Interests under the prefix to be 600 Interests/s and 1500 Interests/s respectively.

Different from the fake Interest attack simulations, at the beginning of the Pushback, since both legitimate clients and attackers send out valid Interest packets, as we discussed in Section 6.1, the router cannot tell good traffic from bad traffic and thus both legitimate clients and attackers will be limited. After receiving the DDoS NACK, legitimate clients will abide by the control placed and lower down their sending rate until the router determines them to be legitimate and free the limits, which explains why the green line goes down in the first several seconds of the attack and then back to the normal later. As for attackers, PAP will halve the limit and similar to the plots for Fake Interest attack, we notice that at the end of the Pushback, almost all the Interests received by the victims are from legitimate clients.



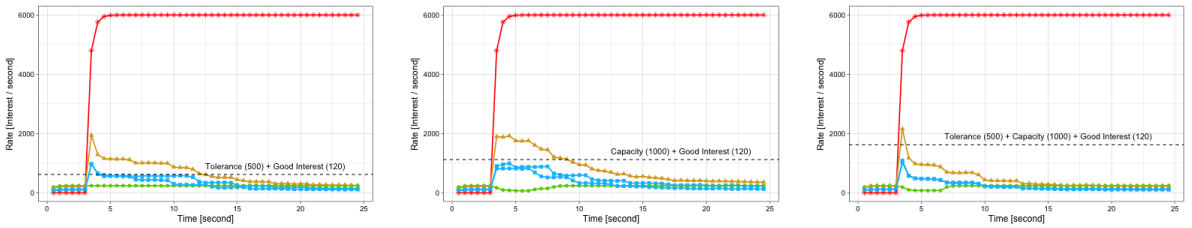
(a) Mixed Interest Attack, C=600, T=500 (b) Mixed Interest Attack, C=1500, T=500

Figure 7.5: Mixed Interest Attack after Employing PAP

7.2.3 PAP: Mixed Interest Attack

Figure 7.5a and Figure 7.5b show how PAP handle mixed Interest attack where attackers will send both fake and valid Interests towards the server. We set the valid Interest capacity and fake Interest tolerance to be 600 and 500 respectively for Figure 7.5a and 1500 and 500 for Figure 7.5b.

Regarding the two plots, one obvious difference from the fake and valid Interest attack scenarios is that, after the attack starts, PAP will limit the traffic to be much lower than the black line (the tolerance plus the capacity plus the good Interests). This is because when mixed DDoS takes place, the router will perform both fake Interest Pushback and valid Interest Pushback; given the valid Interest Pushback will calculate the weighted capacity based on all the matched pending Interests (including fake interests when a mixed DDoS takes place), the weighted capacity will be shared by both valid Interest senders and fake Interest senders. Besides, the gateway router will take the smaller value from the limits for Fake Interest and Valid Interest. As a consequence of aforementioned reasons, the limited traffic rate will be smaller than single type attack scenario, but this is not a problem because we achieve the goal to save the victim. Furthermore, after a short time period, PAP will adjust the limit to the misbehaving clients only and legitimate clients will recover. As shown, in the end, PAP will only pass legitimate traffic to the server.



(a) Two Targets, Fake Interest Attack, T=500

(b) Two Targets, Valid Interest Attack, C=1000

(c) Two Targets, Mixed Interest Attack, T=500, C=1000

Figure 7.6: Fake Interest Attack, Valid Interest Attack and Mixed Interest Attack with Two Victims after Employing PAP

7.2.4 PAP: Two Victim Servers

Previous plots indicate that PAP works well with all types of Interest attacks, keeping the traffic well below the thresholds. In previous simulations, there is one server that is under attack. We also evaluated PAP when attackers attack two servers at the same time with three different types of Interest attacks. The topology settings are the same as the previous simulation and we set the fake Interest tolerance to be 500 and valid Interest capacity to be 1000 for both servers. We let attackers flood two servers at the same time using fake Interest attack, valid Interest attack, and mixed Interest attack respectively and the results are shown in Figure 7.6a, Figure 7.6b, and Figure 7.6c.

The simulation results show that when multiple Pushbacks take place, PAP can effectively control the DDoS traffic from both attacks at the same time. For each victim server, the incoming Interests are controlled in the similar way as that when there is only one victim server under attack. Two servers' incoming traffic lines go below the threshold less than 2 seconds after the attack and soon (about 10 seconds after the attack) merge the legitimate client traffic line.

CHAPTER 8

Discussion

8.1 Authenticity of Victim's PAP NACK

PAP relies on victim's explicit feedback to start a Pushback, thus the authenticity of the feedback is of vital importance. Attackers may send fake NACKs to deny services of good traffic. However, such fake NACKs can easily be detected by the gateway routers of those attackers.

Given a gateway router knows the interface it received the NACK from, the router can get the corresponding prefix(es) of that interface from its FIB. To validate a NACK, the router can simply check whether the NACKed prefix is under the prefix(es) from the FIB or not. If yes, the gateway router can trust the NACK and start a Pushback, otherwise, the NACK should be ignored and the router should be careful because the end host behind this interface could be compromised. For instance, the router connects to a server in one hop and the FIB entry of the corresponding interface is "`/univ1/example`". If a router receives a NACK from that interface to trigger a Pushback for prefix "`/univ1/cs/server`", the NACK should be dropped.

8.2 Misbehaving Client Gateway Routers

In PAP system, the client gateway router will perform selective rate limiting. It is possible that attackers can deploy illegitimate routers (e.g., free WiFi access point), and those routers may not follow PAP and will not limit misbehaving clients at all. Given it is not a good choice for upstream routers to drop packets as discussed in Section 6.2.3, in this case, local network

operators can explicitly mark edge routers who can perform the selective rate limiting. In this case, the real edge router will treat the evil routers as normal clients and control.

8.3 Server Side: a Network Congestion or a DDoS Attack

The pushback is triggered by server's NACK packet. The server may use the incoming Interest packet number or other measures to decide when to send out a NACK. In cases of network congestion, the server may sense the similar changes as when there is a DDoS attack, e.g., the incoming Interest packet number exceeds the threshold. One issue is how to differentiate a network congestion from a DDoS attack. On the one hand, to minimize the damage of a possbile DDoS attack, the detection/reaction should be as soon as possible. On the other hand, the server may not be able to distinguish the two in a short time.

Actually, in PAP, it does not matter if a pushback is because of a DDoS or a network congestion. Assuming the pushback is triggered by a congestion, the server will send a NACK packet to the gateway router. Routers will then take actions to ensure the server's incoming traffic is within a expected max value. In this case, the pushback will not hurt much on legitimate users for the following reasons. 1. In case of a network congestion, the incoming traffic volume will not exceed the threshold too much. Therefore, the pushback will only drop a small number of packets, which rarely affect users. 2. A network congestion is usually short lived, so the pushback will also function shortly corresponding because there are less NACK packet. 3. Even if a end user is limited, the user will follow the limit and thus will soon recover from the rate limit.

8.4 User Side: a DDoS Attack or a Network Failure

When a end host sends out an Interest packet but there is no Data back, the user should be able to differentiate a DDoS attack targeted on the server or there is a network failure. In NDN, if there is a network failure, the router which cannot further forward the packet will generate a NACK packet and reply the Interest. Different from the NACK packet used in

PAP, such NACK packets carry the reason code why the packet cannot be further forwarded, e.g., link failure. In cases of a DDoS attack, the end host will either receive a PAP NACK indicating there is a DDoS attack that targeted the server or receive a Data packet.

CHAPTER 9

Future Work: Building a DDoS-resilient Internet Architecture

9.1 Improve the Design of PAP

PAP is a preliminary design with several unsolved issues. We leave these issues as the future work to make PAP a mature design.

9.1.1 Rate Limit Fairness in PAP

In the current implementation, PAP will distribute the capacity/tolerance into each downstream interfaces based on the incoming traffic of each interface. In other words, the more incoming traffic from an interface, the larger a weighted capacity/tolerance will be assigned. In cases when all the users behind that interface are attackers, it does not make sense to assign a large capacity/tolerance value. Even though after some time, the selective rate limiting will correct the limit to ensure that all attackers will be strictly limited, at the early period of the defense, the rate limit distribution among all the end hosts may not be fair.

As a future work, the defense should act more clever to assign proper limit to each consumer and the main difficulty is how to distinguish a legitimate user from a malicious attacker even before the selective rate limiting takes place.

9.1.2 Ensuring the Acceptable Service in Pushback

The defense requires that proper clients of the victim back off in their sending rates. If their backoffs essentially break their applications, even temporarily, the defense has failed,

since service will be denied at the higher application level. To ensure the acceptable service for each user, a possible solution (which actually cannot work) is to set a minimize sending rate. However, because the defense cannot tell the legitimate users from the attacker until selective rate limiting starts, the minimize sending rate may also protect attackers at the very beginning. Also, how to properly set the minimize sending rate is non-trivial.

We leave this as a future work to protect the acceptable service for each legitimate user.

9.2 DDoS by Collusion of Servers and Attackers

In this prospectus, we show that PAP works well to mitigate Interest flooding whose target is an innocent application, but DDoS attacks like crossfire may aim to flood links and routers where evil servers collaborate with attackers to generate a large amount of traffic.

9.2.1 Threat Model

Low-bandwidth links in current networking topologies present a serious threat to Internet usability because they are the bottlenecks in any given network. An attacker has the option to flood all such links which is a highly effective and efficient DDoS technique called crossfire [KLG13] to shut down almost any connectivity to the victim. In NDN, evil servers could collaborate with attackers to generate a large amount of traffic and attack specific links or routers. In this case, an evil server will never send a NACK packet to its gateway router and thus PAP does not fit.

Since maliciously generated Interests will fetch Data packets (e.g., Data packets carrying junk content) from evil servers, there is no “fake" Interests from the perspective of routers. As we discussed in Section 4.6, in this case, NDN’s hop-by-hop congestion control can help to prevent the congestion caused by such attacks but may not be able to distinguish legitimate clients from attackers. We leave the defense approaches to such attacks as our future work.

9.2.2 Related Work

[SS16] proposes an approach to detect such attack using the utilization rate of PIT of the router. If the utilization rate of PIT exceeds the pre-configure threshold, the defense will be triggered. As mentioned in Chapter 6, such defense mechanism requires proper settings of the threshold value and also may take the risk of hurting legitimate users. Also, the proposed approach by [SS16] cannot stop attackers from sending more Interest packets because there is no enforcement.

9.3 Blackholing the Victims

9.3.1 Related Work

Remote-triggered Blackhole filtering is an effective technique for the mitigation of denial-of-service attacks. Also, [KM09] propose a mechanism called Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding, which can also filter the traffic based on the source addresses.

9.3.2 Blackholing over NDN

Once a server has determined it is under DDoS attack, another possible approach, as a last counter, is to change the registered prefix of itself. Thus, the DDoS traffic targeting the previous prefix will be dropped along the path. For example, when the server "`/univ1/service`" is under attack, the server can send a registration command to its router to re-register its prefix to be "`/univ1/service-backup`". The idea is similar to the Black hole filtering [KM09], but since NDN uses names instead of IP addresses, ASes don't need to reserve backup IP addresses for prefix changing. However, simply blackholing the victim will also disrupt services for legitimate clients, thus we need to figure out how to effectively notify "good" clients of the new prefix. One possible approach is to utilize client monitoring and only notice the new prefix to clients that follow the DDoS control. We leave this as another possible future work.

CHAPTER 10

Conclusion

The vulnerabilities at the core of the TCP/IP architecture enable DDoS and make countermeasures difficult to implement. Instead of continually adding extra functionalities as patches to the current TCP/IP architecture, a DDoS-resilient architecture can be a better choice.

Our analysis shows that NDN's architectural design leads to inherent DDoS resilience. By reviewing existing DDoS attack types for IP and potential flooding attacks over NDN, we learned that the only remaining volume-style of network attack is Interest flooding. Therefore, by utilizing NDN's stateful forwarding and structured names, we developed PAP as a preliminary work that is capable of actively responding to all three types of Interest flooding. Evidence from our effort to mitigate the Interest flooding suggests that NDN provides a solid foundation for DDoS defense.

REFERENCES

[AAB17a] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. "Understanding the mirai botnet." In *USENIX Security Symposium*, 2017.

[AAB17b] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. "Understanding the mirai botnet." In *USENIX Security Symposium*, 2017.

[AMM13] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. "Interest flooding attack and countermeasures in Named Data Networking." In *IFIP Networking Conference, 2013*, pp. 1–9. IEEE, 2013.

[AMZ12] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, et al. "ndnSIM: NDN simulator for NS-3." *University of California, Los Angeles, Tech. Rep*, **4**, 2012.

[BCR16] Hitesh Ballani, Yatin Chawathe, Sylvia Ratnasamy, Timothy Roscoe, and Scott Shenker. "Off by default!" 2016.

[Blo70] Burton H Bloom. "Space/time trade-offs in hash coding with allowable errors." *Communications of the ACM*, **13**(7):422–426, 1970.

[CCG13] Alberto Compagno, Mauro Conti, Paolo Gasti, and Gene Tsudik. "Poseidon: Mitigating interest flooding DDoS attacks in named data networking." In *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, pp. 630–638. IEEE, 2013.

[DWF13] Huichen Dai, Yi Wang, Jindou Fan, and Bin Liu. "Mitigate ddos attacks in ndn by interest traceback." In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pp. 381–386. IEEE, 2013.

[FS00] P. Ferguson and D. Senie. "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing." BCP 38, RFC Editor, May 2000. `http://www.rfc-editor.org/rfc/rfc2827.txt`.

[GHG14] Moti Geva, Amir Herzberg, and Yehoshua Gev. "Bandwidth distributed denial of service: Attacks and defenses." *IEEE Security & Privacy*, **12**(1):54–61, 2014.

[GTU13] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. "DoS and DDoS in Named Data Networking." In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, July 2013.

[GTU14] Cesar Ghali, Gene Tsudik, and Ersin Uzun. "Needle in a haystack: Mitigating content poisoning in named-data networking." In *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, 2014.

[HAA13] AKM Hoque, Syed Obaid Amin, Adam Alyyan, Beichuan Zhang, Lixia Zhang, and Lan Wang. "NLSR: named-data link state routing protocol." In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 15–20. ACM, 2013.

[HG04] Mark Handley and Adam Greenhalgh. "Steps towards a DoS-resistant internet architecture." In *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pp. 49–56. ACM, 2004.

[IB02] John Ioannidis and Steven M Bellovin. "Implementing Pushback: Router-Based Defense Against DDoS Attacks." In *NDSS*, volume 2, 2002.

[JWS03] Cheng Jin, Haining Wang, and Kang G Shin. "Hop-count filtering: an effective defense against spoofed DDoS traffic." In *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 30–41. ACM, 2003.

[KAA14] Samad S Kolahi, Amro A Alghalbi, Abdulmohsen F Alotaibi, Saarim S Ahmed, and Divyesh Lad. "Performance comparison of defense mechanisms against TCP SYN flood DDoS attack." In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on*, pp. 143–147. IEEE, 2014.

[KLG13] Min Suk Kang, Soo Bum Lee, and Virgil D Gligor. "The crossfire attack." In *Security and Privacy (SP), 2013 IEEE Symposium on*, pp. 127–141. IEEE, 2013.

[KM09] W. Kumari and D. McPherson. "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)." RFC 5635, RFC Editor, August 2009.

[Kum18] Mohit Kumar. "Biggest-Ever DDoS Attack (1.35 Tbs) Hits Github Website." `https://thehackernews.com/2018/03/biggest-ddos-attack-github.html`, March 2018. [Online; posted 2018-03-01].

[LLY08] Xin Liu, Ang Li, Xiaowei Yang, and David Wetherall. "Passport: Secure and Adoptable Source Authentication." In *NSDI*, volume 8, pp. 365–378, 2008.

[LYL08] Xin Liu, Xiaowei Yang, and Yanbin Lu. "To filter or to authorize: Network-layer DoS defense against multimillion-node botnets." In *ACM SIGCOMM Computer Communication Review*, volume 38, pp. 195–206. ACM, 2008.

[MR04] Jelena Mirkovic and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." *ACM SIGCOMM Computer Communication Review*, **34**(2):39–53, 2004.

[PCP12] Ioannis Psaras, Wei Koong Chai, and George Pavlou. "Probabilistic in-network caching for information-centric networks." In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 55–60. ACM, 2012.

[Ros14a] Christian Rossow. "Amplification Hell: Revisiting Network Protocols for DDoS Abuse." In *NDSS*, 2014.

[Ros14b] Christian Rossow. "Amplification Hell: Revisiting Network Protocols for DDoS Abuse." In *NDSS*, 2014.

[SS16] Hani Salah and Thorsten Strufe. "Evaluating and mitigating a collusive version of the interest flooding attack in NDN." In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pp. 938–945. IEEE, 2016.

[YAM13] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. "A case for stateful forwarding plane." *Computer Communications*, **36**(7):779–791, 2013.

[YPS04] Abraham Yaar, Adrian Perrig, and Dawn Song. "SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks." In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pp. 130–143. IEEE, 2004.

[YWA05] Xiaowei Yang, David Wetherall, and Thomas Anderson. "A DoS-limiting network architecture." In *ACM SIGCOMM Computer Communication Review*, volume 35, pp. 241–252. ACM, 2005.

[ZAB14] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. "Named data networking." volume 44, pp. 66–73. ACM, 2014.

[ZV18] Zhiyi Zhang, Vishrant Vasavada, et al. "GitHub repository: PAP Implementation." `https://github.com/Zhiyi-Zhang/NDN-DoS-Simulation`, 2018.

[ZZN18] Zhiyi Zhang, Haitao Zhang, Eric Newberry, et al. "Security in Named Data Networking." Technical Report NDN-0057, NDN, 2018.