IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 7, 2022, accepted June 6, 2022, date of publication June 20, 2022, date of current version June 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3184304

Detecting and Mitigating Collusive Interest Flooding Attacks in Named Data Networking

RAMA A. AL-SHARE[1], AHMED S. SHATNAWI[2], AND BASHEER AL-DUWAIRI[1]
[1]Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid 22110, Jordan
[2]Department of Software Engineering, Jordan University of Science and Technology, Irbid 22110, Jordan

Corresponding author: Ahmed S. Shatnawi (ahmedshatnawi@just.edu.jo)

ABSTRACT The large expansion in network services and applications seen in the last few years requires new network architectures to satisfy an increasing number of users and enhance content delivery. Named Data Networking (NDN) has recently appeared as a new paradigm to solve many shortcomings in the current TCP/IP architecture. Its main characteristics like stateful forwarding and in-network caching made NDN networks an efficient environment for data delivery where the data is retrieved based on content names rather than IP addresses. The NDN, by its nature, defends against the well-known Distributed Denial of Service (DDoS) attacks that take place in the traditional TCP/IP architecture. However, a special kind of DDoS attack called Collusive Interest Flooding Attack (CIFA) has appeared to overwhelm the resources of NDN routers by filling their Pending Interest Tables (PIT) with long-lasting malicious entries. The network throughput and consumer satisfaction rate are highly affected by CIFA. A lightweight yet efficient stateless CIFA detection algorithm is proposed in this research utilizing the non-parametric CUSUM algorithm; a change point detection approach that detects the point in time when a transition occurs in the network. The proposed algorithm is characterized by its low computational overhead, highly accurate detection, and quick response. To detect the malicious name prefixes and eliminate the CIFA effect, a mitigation algorithm that uses the average response time vales of all name prefixes is proposed in this research. Experimental results show that this approach detects CIFA after 199.5 ms from when an attack is launched in the large-scale topology. In addition, the mitigation approach effectively reduces the PIT utilization and increases the average consumer satisfaction rate.

INDEX TERMS Collusive interest flooding attack (CIFA), denial of service, detection and mitigation scheme, named data networking (NDN), non-parametric cumulative sum (CUSUM).

I. INTRODUCTION

The Internet has expanded enormously in recent years due to the emergence of new services, applications, and infrastructure [1]. This includes the proliferation of social media applications, streaming video services, cloud computing, fog computing, and the Internet of Things (IoT) [2]. The nature of these services and applications imposes new requirements on the current Internet architecture to satisfy the increased demand of users. Clearly, the traditional TCP/IP architecture has some limitations in dealing with the large amounts of information generated on the Internet. It essentially concentrates on reliability with little focus on Quality of Service (QoS) that is considered the main requirement for video streaming applications [3]. From a security perspective, the TCP/IP architecture tends to secure the channels between network devices using TLS/SSL protocols. However, the data could be modified before or after entering the communication channel; hence, losing the protection at other stages [4].

Named Data Networking (NDN) is one of popular architectures which has recently been proposed to address the limitations of the conventional TCP/IP architecture [5]. Unlike host-centric IP networking that retrieves data based on destination addresses, NDN is Content-Centric Networking (CNN), in which data are retrieved from the network regardless of the destination IP addresses. There are many characteristics for considering NDN as the future Internet architecture in terms of scalability, security, and flexibility. First, the in-network caching property included within its architecture reduces the response time of requests and

The associate editor coordinating the review of this manuscript and approving it for publication was Parul Garg.

eliminates congestion; something that readily enhances the network performance. Second, the use of stateful multipath forwarding strategy enables NDN routers to maintain the state of packets and helps forward them intelligently. Third, NDN has its own authentication properties within the packet format; hence, the authentication is done on the contents instead of the communication channels. Therefore, its built-in features addresses many security problems and defend against the well-known Distributed Denial of Service (DDoS) attacks to which traditional TCP/IP architectures are prone [6].

Meanwhile, a special kind of DoS called Interest Flooding Attack (IFA) affects the throughput and satisfaction rate for NDN architectures [7]. In this type of attack, some malicious consumers try to fill the Pending Interest Table (PIT) space by frequently requesting existent or nonexistent contents until legitimate interests are discarded from the network [8]. Many researchers have tried to detect and mitigate the effect of IFAs using different detection parameters, including timeout statistics and utilization rate calculations of PIT entries [9], [10]. Others have developed PIT management strategies to defend against the attack [11], [12]. Recently, a new variant of interest flooding attacks referred to as "Collusive Interest Flooding Attack" has surfaced in a pretty deceitful way; something that has made it more difficult for researchers to detect and mitigate the attack [13]. In this type, not only malicious consumers contribute to flooding the network, but also the malicious producers who collude with them to fill the PIT table with invalid name prefixes. Often, the CIFA is launched within a short period of time when all consumers send many interest packets simultaneously. These interests stay in the PIT table for a duration very close to the expiration time of the PIT entries, themselves, which causes a flood in the PIT space. Once NDN routers respond to all interests, the PIT returns to its normal state.

In this research, we propose a lightweight and highly accurate CIFA detection mechanism that is based on online sequential analysis. The proposed mechanism employs Cumulative Sum (CUSUM) algorithm for attack detection. The CUSUM was first proposed to monitor the change point detection in a sequence of observations, determining the point when a transition occurs. Here, a normally operating process, requires the presence of a benchmark that reflects a standard network operation. Any deviation from this benchmark would indicate an abnormal ongoing process in the system. In certain situations, this deviation is continuous and small in magnitude, which makes it hard to detect. Therefore, the CUSUM algorithm accumulates these deviations, so they become detectable and readily identifiable by the system. Since CIFA is launched in a relatively short period using multiple interest packets, a large difference between the number of interest and data packets is noticed simultaneously. Therefore, accumulating this difference leveraging the CUSUM algorithm at every sampling period and recognizing the large difference in value can accurately detect CIFA. The proposed detection approach is characterized by low computational overheads, highly accurate detection, and a quick response. Since we use an online sequential approach, the detection time is relatively low.

A. RESEARCH MOTIVATION

The motivation of this work is inherit in the need to prevent new security attacks from affecting the operation of future ICN architectures. Hence, in this study we focus on detecting and mitigating DDoS attacks in NDN; specifically collusive interest flooding attacks. We believe that new network architectures need to be investigated for any security issues to prevent cyber-attacks from occurring or mitigate their effects in the future. Our interest in this research stems from a need for NDNs to be deployed on large scale while warranting that no security issues are encountered. This allows end-users to send and receive their information securely with the lowest possible risk. We examine the effect of CIFA and study its impact on network throughput and satisfaction rate of legitimate consumers through extensive simulations over a realistic network topology.

A few researchers have studied in-depth the effect of CIFA and proposed countermeasures to prevent this type of attack [13], [14]. In this study, we focus on detecting and mitigating CIFA by implementing a change point detection algorithm, which makes a real-time statistical analysis on the traffic involved to identify abnormal activities throughout an NDN. We then compare our detection algorithm with other research efforts to show the effectiveness of the proposed system and report its points of strength. Our main goal in this study is to achieve the lowest detection times in detecting CIFA with low computation overhead to prevent a potential attack during early stages.

B. RESEARCH CONTRIBUTION

Most of the detection mechanisms that were designed to detect non-collusive IFAs were found non-effective against CIFAs; thus, propositions for new detection methodologies are in order. An efficient and lightweight scheme for detecting and mitigating CIFA is proposed in this research. Here, the proposed scheme uses a sequential analysis algorithm called CUSUM to monitor and detect changes in NDN traffic in real-time. The benefits of using online statistical analysis algorithms in detecting flooding attacks are the lower computational overheads and the early-stage detection of an attack. The main contribution of this research is summarized as follows:

1) Simulate CIFA and analyze its effects on NDN networks in terms of network throughput, PIT utilization, input interest rate, and satisfaction rate. ndnSIM is used to simulate CIFA based on a fairly realistic topology, leveraging the large-scale rocketfuel AT&T topology.
2) Present an online statistical analysis scheme to make an early-stage detection of CIFA based on the CUSUM algorithm; an online and non-parametric change point detection algorithm used to detect malicious abnormalities in behavior.

3) Develop a mitigation technique that detects malicious name-prefixes and discards attacking interest packets to enhance the throughput and satisfaction rate of legitimate consumers.

The remainder of this paper is organized as follows: A brief introduction of NDN structure and operation is presented in II. An overview of collusive and non-collusive interest flooding attacks is described in III. We provide a literature review about detecting and mitigating CIFA in IV. Thereafter, we present the proposed detection and mitigation schemes in V and VI, respectively. In VII. we analyze and study the effectiveness of the proposed defense scheme in detecting and mitigating CIFA. Finally, we conclude this paper in VIII.

II. OVERVIEW OF NAMED DATA NETWORKING (NDN)

Named Data Networking (NDN) is one of the well-known Information-Centric Networking (ICN) architectures that uses application-defined content names. In this, the contents are named by the applications and used directly in packet forwarding [15]. There are some unique characteristics that distinguish NDN from other ICN architectures. First, NDN is a consumer-driven architecture, where the communication is initiated from the consumer side, and the data are requested by their names as drawn by the active consumer applications. Second, each data packet sent across the network is cryptographically signed by the content producer, so consumers can verify and trust the sender [16]. The consumers wishing to preserve the privacy of their data have the choice to encrypt the packet's payload [17]. Third, a core component that is part of an NDN router architecture, called Pending Interest Table (PIT), has a stateful forwarding property, where the state of each sent packet is maintained employing intelligent forwarding strategies.

The promise of NDN is to provide a content-based communication architecture allowing the users to request their contents regardless of their physical locations across a network; hence, providing a quick response. Rather than fetching data from a specific naming host, the data is fetched by its content name, where it could be retrieved from multiple physical locations. This principle comes from the fact that most internet traffic is based on a data dissemination approach, where most users use the internet primarily to request or publish the content in question. Over time, NDNs have changed the naming system from named hosts to named contents, where the contents could be any data object, including movies, songs, commands, etc. Unlike the end-to-end TCP/IP paradigm, an NDN offer a hop-to-hop architecture as the packet is forwarded from one NDN router to another commensurate with the available forwarding interfaces where the data could be found [18]. The inherited in-network caching properties, intelligent forwarding strategies, and security features have all improved data delivery speeds, reduced network congestion and harnessed data-level security.

The heart of the internet hourglass architecture concentrates on a universal network layer, which is the IP layer that implements and ensures the global inter-connectivity between end-hosts. The thin waist of this architecture led to a large expansion in the internet by separating the upper and lower layer protocols; hence, allowing them to develop separately without constrains. Initially, the internet was created primarily as a **communication network**, where the end-hosts are the named entities in the communication packets. After the emergence of e-commerce applications and social media networks, the Internet has become a **distribution network**. NDN maintains the same hourglass of the ordinary Internet architecture, but changes the thin waist to allow the construction of distribution-based networks, as described in Figure 1. By removing the constraints on the packet naming system, NDN can name hosts, songs, movies, or any data chunks in the network.

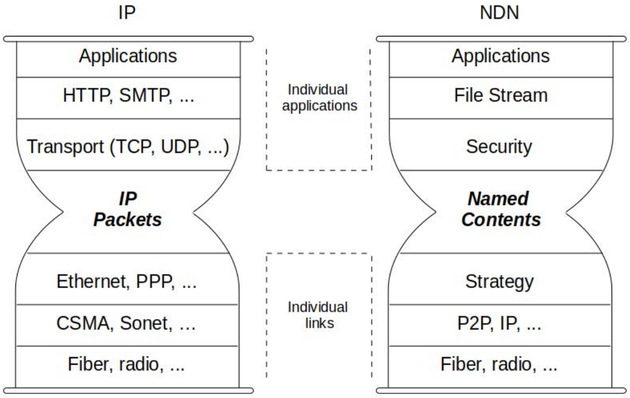


FIGURE 1. The hourglass of IP and NDN architectures.

A. THE NDN STRUCTURE

In NDN networks, clients who request data are called consumers, and servers that render the content are called producers. NDN routers forward consumer requests and store data packets in their cache. There are two types of packets transferred across the network: **interest packets** sent by consumers, which hold information about the requested data, and the **data packets** sent by producers or cache routers that hold the requested data. Generally, interest packets hold information about the content name that is used primarily by NDN routers to forward interests, including order preferences, interest scope, and packet lifetime. Data packets include information about the content name, requested data, and content signature. Figure 2 describe the format of interest and data packets in NDN. Interest packets are forwarded by the core NDN routers based on intelligent forwarding strategies. There are several pre-implemented forwarding strategies defined by an NDN forwarding Deamon (NFD); a core component of an NDN platform; including best route, NCC, and the multicast strategies involved [19]. Here, each strategy has its own merits in terms of the advantages and disadvantages in forwarding the underlying interest packets.

Before getting into the details of the NDN forwarding operation, a clear description of the core components of NDN routers is in order. Three entities are maintained inside

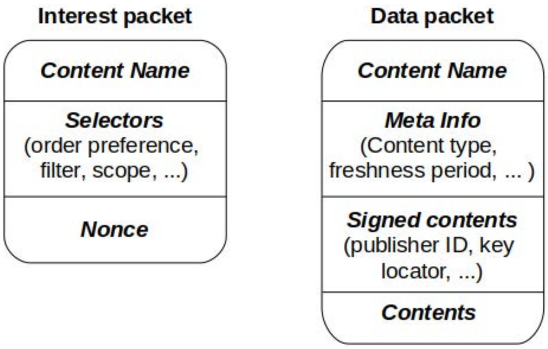

Interest packet
Content Name
Selectors
(order preference, filter, scope, ...)
Nonce
Data packet
Content Name
Meta Info
(Content type, freshness period, ...)
Signed contents
(publisher ID, key locator, ...)
Contents

FIGURE 2. Format of interest and data packets.

each forwarding router, which include the Pending Interest Table (PIT), Content Store (CS), and Forwarding Information Base (FIB). The PIT stores state information of interests that were requested but not fulfilled yet. This information includes the incoming interfaces that requested the contents, along with the requested prefix name. The CS is implemented primarily to enhance the performance of an NDN by storing the more frequently requested data in the router's cache to reduce the response time. Thus, a consumer who requests a previously handled interest will receive a copy of cached data from the NDN routers instead of getting the data from its original sources. The NFD defines a number of cache replacement policies to replace the data cache, and this could vary from one router to another [20]. Finally, the FIB contains a list of known name prefixes along with the interfaces where these prefixes could be located. Specific routing protocols are used to populate the FIB tables.

The naming system of contents in an NDN is fairly flexible, in which the names are chosen based on the consumer application and its requirements. It adopts a hierarchical architecture, where the content name is composed of multiple components separated by a slash character (e.g,, /documents/films/movie.mp4). This naming hierarchy allows users to relate or aggregate the name components with each other. Naming contents in NDNs is one of the essential aspects of application development. End-users have the choice to build their own namespaces for their applications, which enables application developers to make different abstracting methods. Although the management of namespaces is not included in the designed architecture of NDNs, different researchers have implemented various management schemes [21], [22] and proposed secure mapping solutions [23] for namespaces in NDNs.

B. NDN OPERATION

The operation of an NDN is based on a publish/subscribe model, where the data is published by producers and requested by subscribed consumers. Initially, the consumer requests the contents by sending an interest packet that carries the requested content name. When an interest packet is received, an NDN router searches for the requested contents in its Content Store (CS). When a data is found, it will be forwarded to the same interface where the interest packet was received. Otherwise, the router will search its PIT if an entry is found for the asked name prefix. If so, the incoming interface of the interest packet is added to the entry in the PIT. When no existing entry is found, the router will create a new PIT entry for this new interest and forward it again to another NDN router through a strategy determined commensurate with the information stored in the FIB. When the data is found, in some cache routers or data sources, it will be forwarded in a reverse path until it reaches the requesting consumer. If no data is found for an interest packet, the NDN router will simply remove the PIT entry for such contents after an expiration time has been reached. No error messages are transferred in an NDN; thus, the consumer has the choice to re-transmit another interest packet using a specific timeout mechanism.

When a data packet is received, an NDN router searches for a matching entry in its PIT and forwards the packet to the interfaces that requested the contents. Then, the PIT entry will be removed from the PIT after forwarding the data packets involved. To improve content delivery, the data is cached in the CS for future requests. The rule of forwarding data packets is to take the reverse path of interest packets. Here, all the PIT entries related to the requested interest packet will be removed from all routers along the route. The operation of an NDN is described clearly in Figure 3. From the NDN forwarding process, we can see that interest and data packets do not carry any information about end-hosts. They only hold information about the content's names, where forwarding is done based on content information as opposed to destination IP addresses [24].

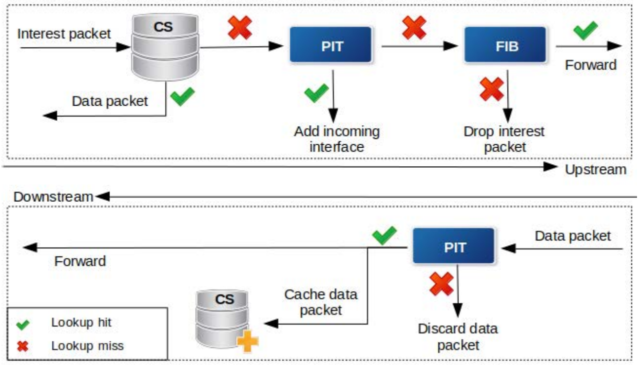


FIGURE 3. Forwarding scheme of NDN.

III. DENIAL OF SERVICE ATTACKS IN NDN

A Denial of Service (DoS) attack is a well-known type of cyber-attack that aims to consume system resources and overwhelm networks. Over the last few years, more attention was drawn to mitigate different types of DoS attacks due to the financial losses sustained and the reputation damage they incur to the businesses involved. Commonly, an attacker uses multiple compromised machines to launch a Distributed DoS (DDoS) attack, severely blocking legitimate users service and adversely impact the business/es involved. In the current TCP/IP architecture, an attacker can easily compromise machines and control them remotely due to the addressable nature of end-hosts [6]. In NDNs, things are totally different since the packets are sent based on content names, not destination IP addresses; hence, launching DDoS attacks could be more difficult in NDN than in ordinary TCP/IP architectures. Nonetheless, there is no guarantee that attackers will not find a way to compromise consumer machines and build their own botnets over an NDN. Hence, we assume that attackers have multiple malicious systems distributed around the network over which they can launch DDoS attacks.

There are two familiar types of DoS attacks commonly identified in NDNs; the Cache Poisoning (CP) and Interest Flooding Attacks (IFA). Cache poisoning attacks aim to forward and distribute fake contents throughout the network by means of compromised routers or end-hosts [25]. Since NDN routers have a special forwarding property, manifested in their ability to serve user requests in case of having copies of the requested content, any compromised router can forward fake contents to consumers. On the other hand, interest flooding attacks target the pending interest tables of NDN routers, where malicious consumers send many interest packets to overwhelm a router's PIT with existing or non-existent content [26]. This type of a flooding attack bears another name, known as non-collusive IFA. Another attack type, dubbed Collusive Interest Flooding Attack (CIFA), is launched with the help of a collusive producer that provides content to malicious consumers. Although collusive and non-collusive IFAs have almost the same impact on NDNs, detecting CIFA is a more difficult process since the malicious interests are satisfied, which makes it more difficult for the defender to distinguish legitimate interests from malicious ones.

A. INTEREST FLOODING ATTACK (IFA)

The aim of IFAs is to overload NDN routers by requesting a large number of unsatisfied contents, exploiting the stateful forwarding feature of an NDN. As the routers maintain a state on every sending interest, an attacker can exhaust system resources and prevent them from handling the valid requests sent by legitimate consumers. There are three common types of IFAs, which are classified depending on the type of contents requested by malicious consumers. The first type requests existing contents, in which an attacker uses a large number of compromised machines to request valid contents; thus, overloading producer resources. The effect in this type is slightly little compared with other IFA types due to the inherent in-network caching property. Every time an attacker requests an existing content from the network, the caching routers will respond from their cache stores, which reduces IFA effects. The second type requests dynamic contents, where dynamic interests are generated to request valid contents from the producer side. This type of attack highly consumes the router's PIT and producer's computational resources since the router will create a PIT entry for every single interest, and the producer will be forced to sign every data packet generated in the network. The last type requests non-existent contents, where unsatisfiable interests are generated by attackers to consume the memory resources of a router's PIT. These interests will stay in the PIT until their expiration time is passed, which causes a large drop in legitimate interests when the PIT is filled with unsatisfied malicious entries.

B. COLLUSIVE INTEREST FLOODING ATTACK (CIFA)

As opposed to IFAs, CIFA attacks aim to fill up the PIT of intermediate NDN routers with valid contents provided by a malicious producer. Here, all interest packets are satisfied, which makes it more difficult for the detector to distinguish malicious interests from legitimate ones. This type of flooding attack is more deceptive than the IFA since the nature of malicious traffic is very close to legitimate traffic. An attacker in CIFA colludes with a malicious server to provide valid and unpopular content to malicious consumers. Every malicious consumer sends a small number of interest packets, requesting different contents from the colluding server. This will force NDN routers to create a new entry in the PIT for every interest packet, causing the PIT to be overloaded and the network throughput to drop. The main idea behind CIFA is to keep the malicious entries in the PIT for as long as possible, so that NDN routers will not be able to serve the legitimate consumers during this period. To achieve this goal, the colluding server will be programmed in a way such that all data packets are sent after a certain time lapse close to the interest lifetime period. Here, the time duration should be long enough to keep the interest packet in the PIT as much as possible. If the malicious producer responds to interest packets in a short period, the attack efficiency will decrease, and the opportunity of accepting new legitimate packets will increase.Therefore, the entries will, as a result, stay in the PIT for a long time, forcing NDN routers to drop all incoming traffic until these entries are satisfied. As such, a CIFA attack prevents legitimate consumers from sending their interest packets and receiving their corresponding data packets. The NDN routers will be overwhelmed by malicious traffic causing the PIT table to be filled with long-duration entries; thus, the legitimate interests will be dropped by the NDN routers till the colluding producer responds to all malicious interests. Apart from consuming PIT resources and decreasing the satisfaction rate of legitimate consumers, the attack also consumes the bandwidth and results in a sudden decrease in network throughput [27].

Usually, the CIFA is launched as periodic pulses with intermittent attacking periods represented by three parameters, include the attack duration T, interest sending interval t, and attack intensity R. Figure 4 describes the attack model for a CIFA attack. First, all the attackers send R interest packets within an interval t and wait for a response from the colluding server. Following this period, the malicious entries will stay in the PIT till colluding servers send their corresponding data packets. At this stage, the malicious consumers will stop sending interest packets and the NDN routers will wait for a response to all interest packets stored in the PIT. When the attacking entries are about to expire in the PIT, the collusive server responds with the corresponding data packets at the end of time T. At this stage, the next attack phase will commence. The periodic nature of a CIFA results in severe degradation in throughput due to the altered traffic status during the attack.

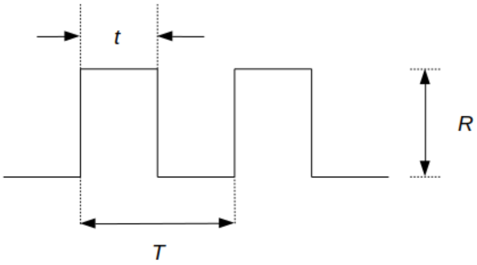


FIGURE 4. CIFA attack model [28].

By launching CIFA attacks, the attacker can achieve better effectiveness than IFA attacks (Type 1 and 2) with the help of the colluding server since the request of interest packets is done on non-legitimate contents that are satisfied within a long period of time. Also, Unlike IFA attacks (Type 3), the attack here is more hidden since all malicious interests are satisfied by the colluding server; therefore, the attack behavior is very similar to the normal legitimate behavior.

To clearly illustrate the idea of a CIFA attack, Figure 5 describes an attack scenario with three malicious consumers and one collusive producer. The malicious consumers MC1, MC2, and MC3 send interest packets to the collusive producer MP. The PIT of bottleneck routers will be filled with these interest packets for a long time, causing legitimate interests to be dropped in the network. In this topology, malicious interest packets result in PIT overload in routers R6 and R8, which would lead to discarding the interest packets sent by the legitimate consumer LC2. As described in the PIT table of router R6, the last three entries are reserved for malicious consumers. Since the colluding server usually responds after some time lapse, all legitimate interests received during this time will be discarded. Here, the main property that differentiates the collusive version of flooding attacks from others relates to the satisfied requests of interests, such that the underlying traffic pattern looks pretty much the same as the legitimate one.

IV. LITERATURE REVIEW

The security of an NDN has received increased attention in the literature, especially that of securing the network from DDoS attacks. In this section, we will present the detection and mitigation architectures proposed over the last few years to defend against IFA and CIFA attacks.

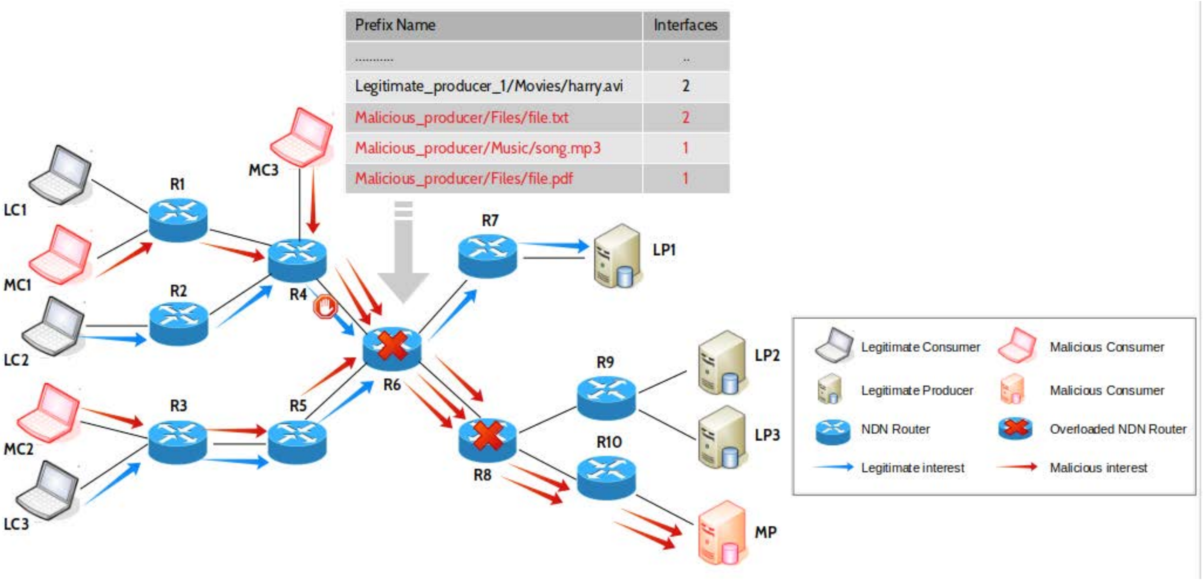
A. REVIEW OF IFA DEFENCE MECHANISMS

Zhi *et al.* [11] used the Gini Impurity concept to calculate the impurity level of different interest names. In each period, the impurity value of interest names received in the current time interval are calculated and compared with the value in the preceding time interval. When a large difference is detected from the normal level of the Gini impurity range, an IFA is detected. To mitigate the attack effect, the authors started out by applying a recognition method to identify malicious interest packets by calculating the Gini impurity variation for each name prefix. Launching IFA decreases the impurity value of a name prefix, which is then deemed as a malicious name prefix. The mitigation is done by limiting the input interest rate of recognized malicious prefixes. The router then notifies the downstream NDN routers about these malicious prefixes by sending a notification packet like the data packet format, listing the prefix name in the content field.

An early detection and mitigation scheme was first presented by [12]. Here, the proposed scheme is designed to reduce the PIT occupancy rate by enforcing an effective PIT management in the NDN routers. The defense scheme is applied to edge routers; routers that are directly connected to the consumers to defend against IFA and reduce its impact at an early stage. A stateless defense mechanism is proposed leveraging the AQM algorithm so as to effect an active queue management scheme that is able to remove malicious PIT entries and rejects incoming malicious interests received by NDN routers.

A Theil-based Countermeasure (TC) approach was proposed by [29] to detect IFAs. The interest packets are first divided into multiple groups based on the calculated Theil entropy value. The degree of unevenness of interest name distribution is calculated to determine the proportional value to both interest packets within a specific group, and all interest packets of all classified groups. When the Theil entropy value of the incoming interest names starts to gradually decreases, an abnormal activity is detected in the network, denoting the existence of an IFA. Here, a localization approach is used to determine an attacker's location by tracing back malicious interests.

In [30] authors propose a technique, which sets it apart from all others, in applying the IFA detection on the server side rather than applying it on the NDN router. Every content server maintains two main components: a detection component for detecting malicious name prefixes and an HSL generation component to generate a hash value for each content name. The content server periodically monitors the content name's requesting statistics to detect any anomalous requests sent by consumers. Once the content server detects a malicious name prefix, an alarm will be sent to the downstream NDN routers, instructing them to enable HSL validation for incoming interest packets possessing a negative name prefix. The NDN routers then specify whether the subsequent interest packets are fake or real depending on the HSL validation results.



Prefix Name	Interfaces
....	..
Legitimate_producer_1/Movies/harry.avi	2
Malicious_producer/Files/file.txt	2
Malicious_producer/Music/song.mp3	1
Malicious_producer/Files/file.pdf	1

FIGURE 5. CIFA attack scenario.

Authors in [31] propose the use of signed control packets to transfer control commands between NDN routers and content producers to organize the network and detect any suspicious or malicious behaviors.Authors in [Benmoussa *et al.*, 2020] propose the use of signed control packets to transfer control commands between NDN routers and content producers to organize the network and detect any suspicious or malicious behaviors. Producer-based control packets are sent by producers to up-stream routers the minute they are overloaded by consumer requests. Router-based control packets are sent by NDN routers to coordinate with other neighboring routers to regulate the traffic. The mitigation is done on edge NDN routers by specifying legitimate, suspicious, and malicious end-hosts based on their behaviors. A consumer behavior is considered suspicious when the satisfaction rate is low and the number of timed-out interests is high, or when the router receives a producer-based control packet. In this case, a rate limiting approach will be applied in the network. A consumer behavior is classified as malicious when both conditions are met. In this case, those consumers will be blocked by the NDN router. Any transferred packets are cryptographically signed by senders and verified by receivers. Although the proposed approach effectively mitigates IFA and increases consumer satisfaction rate, the communication and computation overheads are rather large as every control packet gets to be signed and transferred between NDN routers.

The work presented by Cheng *et al.* [32] aims at detecting and mitigating a more sophisticated interest flooding attack scenario. Here, an attacker first launches a low-rate IFA, then increases the rate gradually with the primary goal of overwhelming the NDN routers. A central controller which maintains a global overview of the network is used to collect state information from edge NDN routers and detect any anomalous events by injected by consumers. Each edge NDN router is supposed to monitor the state of each interface; so, when a suspicious behavior is detected, a notification is sent to the controller informing it about incidences of suspicious activities. However, abnormal behaviors reported by just one NDN router cannot quite ensure the existence of an IFA attack; hence, the reason why a controller is needed to collect reported behaviors from all routers and analyze them to determine whether an attack is detected in the network, or not, with some certainty. Once an attack is confirmed, the controller will further analyze reports to locate malicious consumers and report them back to the NDN routers.

Pu *et al.* [33] proposed a countermeasure approach called Self-Adjusting Share-Based Countermeasure (SSC), whereby the interest rate is used as a parameter to prevent the IFA attack. A tracing table is maintained in each NDN router to record traffic statistics, including the number of incoming interest and data packets, received negative acknowledgment packets, and expired interest packets. At the end of every observation period, the router collects these statistics and calculates the Interest Unsatisfaction Ratio (IUR) for all interfaces. When the IUR for a specific interface becomes larger than the average IUR, the forwarded interest share value of that interface is reduced with an amount equal to the difference between the two unsatisfaction rate values. Hence, the number of interest packets forwarded by each interface is adjusted dynamically based on the unsatisfaction interest rate. The proposed scheme was able to enhance the PIT utilization rate while mitigating an IFA attack effectively across an NDN.

Researchers in [34] used the features of self-similarity and content name entropy to detect IFA attacks. First, the traffic statistics are collected periodically, including requested contents name, number of times these contents are requested, together with the interest arrival time. This information is divided into multiple data blocks to ease the data processing involved. The Hurst index and content entropy calculation are then calculated for each data series. Here, the Hurst index is used to measure the self-similarity of NDN traffic whereas the information entropy is used to measure the random variation of a data series. As such, a large entropy value with a low Hurst index are leveraged to mark the existence of an IFA attack. Moreover, a non-parametric statistical analysis algorithm commonly referred to as CUSUM is used to detect any abnormal changes across the network by accumulating the small shifts in the Hurst and entropy values.

Authors of [35] proposed an iForest to isolate and detect the abnormal prefixes in NDN. The abnormality score of each prefix was calculated to identify the abnormal prefixes, and all prefixes that maintain an occupancy rate higher than a predetermined threshold are considered malicious prefixes. A rate limitation approach was proposed to mitigate the IFA by sending a notification packet to all downstream routers, preventing them from forwarding the malicious interests.

Researchers in [36] proposed a mitigation algorithm called CHOose to Kill Interest Flooding Attacks (ChoKIFA), where malicious PIT entries and interest packets are detected and discarded by the NDN router. When the PIT size exceeds a predefined minimum threshold value, and the current state of an incoming interest (i.e., prefix name and incoming interface) is similar to a randomly selected interest from the PIT table, the input satisfaction rate will be calculated. If it exceeds some threshold, the incoming interest packet is discarded, and the malicious entry is deleted from the PIT. Otherwise, the incoming interest packet will be dropped with a specific probability, calculated based on the average PIT size.

Authors of [37] proposed a collaborative approach between NDN routers with the help of a centralized controller to detect and mitigate DDoS attacks in NDN. An NDN router connected directly to the content provider is implemented and placed in the network to filter the incoming packets and detect distributed denial of service attacks. Fake interest packets that maintain a legitimate prefix name are detected by mapping all prefixes in the content provider's database into the Quotient-based Cuckoo Filter (QCF). A warning message is sent to the NDN controller reporting the name of malicious interest names. To detect and mitigate IFA, intermediate routers periodically monitor PIT expiration and occupancy rates for each interface. When they both exceed threshold values, all fake entries are deleted from the PIT table, and a warning message is distributed to delete the entries from all downstream routers.

B. REVIEW OF CIFA DEFENCE MECHANISMS

Salah and Strufe [38]were amongst the pioneers who proposed a novel framework for detecting the collusive version of interest flooding attacks. Here, a coordinated monitoring framework called CoMon was introduced to mitigate the effect of CIFA attacks in NDN, as an enhancement leveraging the previously proposed framework [39]. In that earlier version the focus primarily dealt with mitigating the non-collusive interest flooding attacks. Certain NDN routers were chosen based on a location-based greedy algorithm to serve as Monitoring Routers (MRs) to observe the utilization rate of PIT entries and report any malicious name prefixes locally.

Local observations of the PIT utilization rates can only detect high-rate flooding attacks since the monitoring router possesses only local knowledge about the network. However, a Domain Controller (DC), which maintains a global overview of the network topology, was also located to detect low-rate attacks by collecting the PIT utilization information form the MRs and extracting malicious name-prefixes across the NDN network. The mitigation process is performed by discarding the packets coming from the name-prefixes that were marked during the monitoring process.

Xin *et al.* [40] analyzed the spectrum of the traffic and extracted the main properties that distinguish a CIFA traffic from a normal one. Leveraging wavelet transform theory, the authors noticed that the Power Spectral Density (PSD) of the malicious traffic is concentrated mainly around the low frequencies. Based on these findings, the authors suggested a decomposition of the traffic signal using wavelet transform algorithms and a reconstruction of the low-frequency sub-band from the high-scaled wavelet coefficients. The re-construction of the low-frequency band, where the malicious traffic is usually concentrated, makes it easy to detect any sudden signal changes; thus, it easily identifies CIFA attacks in the network. The modulus of the re-constructed sub-band was compared with a specific threshold value to finally decide whether, or not, a CIFA attack is certainly detected across the network.

Liu *et al.* [41] used the prediction error based on particle filtering to detect CIFA by calculating the difference between the predicted and estimated traffic signals; thus, detecting CIFA attacks when the error value exceeds a predefined threshold. It was noticed, in a stark observation, that the fundamental changes in the traffic pattern appeared at the start and end points of a CIFA attack, when the traffic started to become anomalous as it returned to its normal state at the end of an attack. These sudden changes in a signal make it possible to detect an attack easily in an NDN. Initially, a set of particles is defined and calculated using particle filtering algorithms from an observed NDN traffic, representing a one-step prediction process. After that, the estimated traffic value is calculated based on the newly measured weight, and the error value is consequently computed to decide whether the network is actually under a CIFA attack.

Shigeyasu *et al.* [42] applied a distributed approach using the cache reference indicator for detecting and preventing CIFA attacks through the relay Cache Routers (CRs) distributed across NDN networks. As most of the CIFA attacks use different name-prefixes for each interest packet, where legitimate users would not usually request these names, the reference number to these entries in the cache must be very close to zero. As a first step, the PIT state is monitored by relay CRs to detect any abnormal change in the table size. When the PIT is overflowed, and the interest packets start to be discarded from the table, the incoming interest packet rate is calculated for each prefix name. When the incoming rate for a specific prefix exceeds the pre-set threshold, its cache reference number is determined for further detection. Finally, the router will classify the prefixes with small reference values as malicious name-prefixes. It will, also, discard the packets holding such a prefix in their contents.

Nasserala *et al.* [43] focused on enhancing the cache to mitigate an attack as opposed to monitoring the PIT and measuring its utilization rate. In their work, the authors divide the Cache-Store (CS) into multiple sub-caches based on the number of interfaces of the NDN router. The size of each sub-cache is determined based on the transmission rate in each interface, such that the interface with the highest transmission rate maintains the larger sub-cache size. Here, each sub-cache is made responsible for saving the information of the users directly connected to its interface. This separation in the cache could mitigate the effect of CIFA since the sub-cache with the malicious interface that manages the attacking packets will be the only sub-cache affected. Here, all other clean sub-caches will continue to provide other users with their legitimate information, as requested. However, particular consideration in this research is given to the assumption that a router could maintain at most one malicious interface, in an effort to prevent the performance of the proposed solution from any degradation. Further, when legitimate users request contents from an attacking interface, such contents are not found in the other sub-caches. This will inherently lead to requests not being satisfied; a scenario that is found in most cases.

Wu *et al.* [28] adopted the confidence interval and scrolling time window to detect and mitigate CIFA attacks. The authors use two primary features to specify the upper and lower bounds of the confidence interval, the existence times of PIT entries, and the throughput. Since the existence time for interest packets for a CIFA attack is high compared with the normal network operation, it could be a harnessed as a good indicator to detect anomalous entries as opposed to normal ones. Initially, the upper and lower bounds for the confidence interval are determined, which represents the maximum existence time and minimum throughput, respectively. Now, for as long as interest packets continue to be received, the scrolling window continues to move on, and the confidence interval is updated when the current traffic is considered normal (i.e., does not violate the upper and lower bound limits). Otherwise, if the traffic is deemed as anomalous (i.e., some packets violate the upper and lower bound limits), the PIT utilization is measured, recorded, and compared against some threshold value. As a last step, once the utilization rate exceeds the selected threshold, the packets which hold the highest existence time values will be discarded from the PIT.

Existing solutions to detect and mitigate CIFA attacks exhibit limitations and other considerations that could affect detection accuracy. In [38], the placement of monitoring routers statically based on the locations of data sources may not yield a practical solution since the nature of an NDN network is dynamic. Moreover, implementing the CoMon architecture requires a change in interest packet format and communication structure, which is not recommended over NDN networks. On the other hand, in the solution proposed in [40], the false alarm rate is relatively high due to the intermittent nature of CIFA attacks. Further, the method proposed in [41] suffers a high computational overhead, especially when too many particles are selected for particle filtering. Researchers in [43] consider that NDN routers maintain at most one malicious interface; hence, the performance of this approach decreases when the router maintains more than one malicious interface. 1 describes the pros and cons, detection parameters, and the overhead for each proposed approach.

V. CIFA DETECTION MECHANISM

In this research, we propose a lightweight and highly accurate CIFA detection mechanism that is based on online sequential analysis. The proposed mechanism employs the Cumulative Sum (CUSUM) algorithm for attack detection. The CUSUM was first proposed by Page [44] to monitor the change point detection in a sequence of observations, which determines the point in time when a transition occurs. During any normal process, there should be a benchmark that represents standard network operation. The deviation from this benchmark provides an indication of an anomalous process taking place in the system. In certain situations, this deviation is continuous and small in scale; something that is hard to detect. Consequently, the CUSUM algorithm accumulates such deviations, making them more identifiable and detectable by the system. Since CIFA is launched over a small period using multiple interest packets, a large difference between the number of interest and data packets is readily observable. Therefore, accumulating this difference using the CUSUM algorithm at every sampling period and recognizing the high difference in value can more accurately detect the occurrence of an attack. The proposed detection approach is characterized by its low computational overhead, high detection accuracy, and favorably quick response. Here, as we leverage an online sequential approach, the detection time is expected to be relatively low.

One of the main events that indicate the existence of a CIFA attack is the delayed responses to interest packets. Under normal conditions, the time duration between sending interest packets and receiving their corresponding data packets is much less than the lifetime of interests. In contrast, under a CIFA attack, malicious producers tend to send the corresponding data packets after a long duration of time close to the interest lifetime value. This behavior indicates that the difference between the number of interest packets and data packets at the initial stages of a CIFA attack will be large compared with normal network conditions. Therefore, our proposed detection algorithm uses the large frequent differences between interest and data packets as an indicator to identify an attack. The proposed solution could be seen as a Change Point Detection process, where the times of changes in time series data are detected and identified [45]. Specifically, the change point detection could be used as an anomaly detection system, where the anomalous behaviors are noticed significantly in the network. As with any anomaly detection system, the observed behavior of traffic is compared with the normal profile, which represents legitimate consumer requests in our case, and any variation from the normal baseline will be identified as an attack.

A change detection could be done offline or online, depending on the application that is under investigation. For an offline change detection, the analysis is done on a complete sequence of data where the goal is to provide an accurate estimation of the detected changes in a predefined dataset. However, the offline analysis cannot be applied to real-time data; hence, we cannot take any action when a change is detected. On the other hand, an online change detection can be used with real-time streams; hence, an action could be taken immediately when a change is detected. Further, online change detection saves memory and computational overheads by applying the analysis sequentially when new data streams are observed. In our work, we use a sequential analysis approach using the CUSUM algorithm, where the change point detection is applied online, and the attack is detected in the early stages [44].

A. CUSUM ALGORITHM

In practice, representing user requests in an NDN network by a simple parametric model that could be valid always is rather difficult. Therefore, we tend to use non-parametric methodologies like CUSUM for our change point detection problem. Initially, we will describe the main idea behind the CUSUM algorithm and how it can detect abnormal behaviors in time-series data. Then, we describe our detection system that will be used to detect CIFA attacks.

Let $\{X_i, i = 1, 2, \ldots\}$ be a sequence of observations in a random process. The process mean for X_i is $E(X_i) = \mu_i$, which is the parameter that we aim to monitor during a process initiation. If we set the upper bound of μ_i to μ_0, the normal operating mean should not exceed this value under normal network conditions. The following formula represents the CUSUM control scheme proposed by Page to monitor the process of $\{X_i\}$:

$$S_i = \begin{cases} 0, & i = 0 \\ (S_{i-1} + (X_i - \mu_0))^+, & i = 1, 2, \ldots \end{cases} \tag{1}$$

where $x^+ = max(0, x)$, and μ_0 is the upper bound of a normal process mean. When S_i exceeds the predefined threshold S_{th}, a signal is generated from the system indicating a violation in the observed CUSUM value. The main purpose of defining μ_0 in the formula is to offset the mean of the process in order to prevent the system from deviating toward the abnormal signal level S_{th}. Assuming that the mean of the observations collected from the process is very close to zero such that $\mu_i \ll 1$, the mean of $X_i - \mu_0$ will be negative under normal network conditions. Under abnormal conditions, this value will exceed the upper bound μ_0 and suddenly reach a large positive value. The large increase in the mean of the value $X_i - \mu_0$ is lower bounded by μ_1. This means that μ_1 represents the lower bound of the increase in the mean, which indicates the existence of a change in the time-series data.

B. ATTACK DETECTION PROCEDURE

Let N_i and N_d be the observed number of sent interests and received data packets, respectively, within a sampling period t_s. Define $\Delta n = |N_i - N_d|$, which represents the difference between the number of interest and data packets collected within the sampling period t_s. As we noticed in the attacking behavior of CIFA, this difference will gradually increase when the malicious consumers start to launch an attack at the beginning of each attacking period. Using the CUSUM algorithm, this increase could be detected when we observe changes during normal network operation. One thing that we will do before using the calculated difference directly in the CUSUM algorithm is to normalize it by the average number of data packets. Let $\overline{D}(n)$ be the average number of data packets, which is calculated and updated periodically as follows:

$$\overline{D}(n) = \alpha\overline{D}(n-1) + (1-\alpha)D(n) \tag{2}$$

where α is a constant lying between [0, 1], n is the discrete time value, and $D(n)$ is the count of data packets that have been observed recently. The aim of the normalization process is to make the mean value of Δ very close to zero, making it independent on the amount of traffic or the time of the day. Define $X_n = \Delta n \ / \ \overline{D}(n)$, which represents the normalized difference between the number of interest and data packets collected within a sampling period t_s. Now, the value of X_n will be used in Equation 1 to calculate the CUSUM of process every time a difference is observed at the end of every sampling period. The following function represents the decision that will be taken while monitoring the value S_i:

$$d(S_i) = \begin{cases} 1, & S_i > S_{th} \\ 0, & S_i \leq S_{th} \end{cases} \tag{3}$$

As we can see from the decision function that 1 represents the presence of the attack, which is happened when the value of S_i exceeds the predefined threshold S_{th}, while 0 represents the normal operation where no attack or abnormal behavior is detected.

C. PARAMETER SELECTION

To apply the proposed detection algorithm, suitable sampling periods and threshold values are chosen for the CUSUM algorithm. Selecting appropriate sampling periods for tracing the cumulative sum value is a rather important step

TABLE 1. Comparison of defence mechanisms for CIFA in NDN.

Reference	Description	Detection Parameter	Overhead	Pros	Cons
[38]	Monitor the utilization rate of PIT entries and report the malicious name-prefixes locally through MRs and globally through DC	PIT utilization	Communication, signaling and computation overhead	1) Attack is detected and mitigated at an early stage 2) Applied only on specific routing nodes 3) Detects high-rate and low-rate attacks	1) Hard to change the interest packet structure format 2) Placing the monitoring routers closed to the sources needs to be properly planned
[40]	Construct the low-frequency sub-band from the throughput traffic band using wavelet transform and compare its modulus with a predefined threshold	Power spectral density of the traffic signal	Computation and storage overhead	1) Achieves a detection rate of approximately 90% 2) No change in network structure	1) False alarm rates are high due to the intermittent nature of CIFA attacks 2) The missed detection rate is high
[41]	Calculate the difference error value between predicted and estimated traffic signals and compare this error with a predefined threshold value	Prediction error threshold	Computation and storage overhead	1) Achieves an excellent detection rate of approximately 98.5% 2) False alarm rates and missed detection rates are low 3) No change in network structure	1) Too heavy computations 2) Choosing a lot of particles for particle filtering drains the computational resources of NDN routers
[42]	Monitor the PIT state and calculate the incoming interest rate when the PIT is overflowed. Delete the malicious prefix names that maintain a low cache reference value	PIT utilization and cache reference	Computation overhead	1) Lightweight and effective detection approach 2) Enhance the content acquisition rate up to 50% 3) No change in network structure	1) The attacker can evade this detection approach by increasing the reference number of each malicious prefix in the cache
[43]	Divide the content store into multiple sub-caches based on the number of interfaces to reduce the impact of CIFA	N/A	Computation overhead	1) Reduce the impact of CIFA by 50% 2) Computationally inexpensive solution	1) The performance decreases when the router maintain more than one malicious interface 2) Legitimate user requests are not satisfied if contents are requested from an attacking interface, and these contents are not found in other sub-caches
[28]	Detect the abnormal state of network when any entry violates the confidence interval bounds, and discard the malicious packets when the PIT utilization rate exceeds a threshold value	Existent time of packets and throughput	Computation and storage overhead	1) Applied only on the bottleneck routing nodes 2) Achieves an excellent detection rate with low false alarm rate	1) Since the threshold value is dynamic, the attacker may change his behavior to make the baseline gradually looks similar to the malicious network baseline

to ensure that the detection algorithm achieves detection at low sum values. Meanwhile, choosing high flooding threshold values increases the attack detection time, while selecting low thresholds results in more false alarms when detecting CIFA attacks.

The sampling period t_s is used to collect the number of interest packets forwarded by NDN routers. To map the interest packets to their corresponding data packets, another sampling period t_d is used to collect the number of data packets after the interests are collected during the period t_s. If interest packets have a lifetime value L, where the PIT entry is assumed to expire at the end of this period, then the sampling period of the data packet should be less than L. Further, in order to accurately detect the attack behavior, data packets are expected to be sampled before the end of the attack period T. To detect an attack earlier, thereby increasing the detection accuracy, the sampling period t_d should be much less than $T - t$, where t is the interest sending period as described in Figure 4. Here, sampling periods t_s and t_d with values 125 ms and 62.5 ms,respectively, were chosen to detect a staged CIFA attack, in a manner that ensures the detecting router achieves low detection times.

The proposed detection algorithm, as such, could be applied on both gateway and backbone routers. For backbone routers, high flooding rate is noticed since the traffic is aggregated from different locations, whereas in the case of gateway routers only a small part of the flooding is seen by the detecting router. Hence, it is commonly preferable to use lower threshold values for backbone routers due to their high attack sensitivity and large observed difference values. In order to calculate the threshold value S_{th}, we leveraged the formulas introduced in [46]. The time t_0 was chosen as the designed detection time for backbone routers, whereas $2t_0$ was chosen for gateway routers.

VI. CIFA MITIGATION MECHANISM

In this thesis, we mitigate a CIFA attack by firstly identifying the malicious name prefixes and secondly discarding all incoming interest packets that hold malicious prefixes in their names. After analyzing the CIFA attack behavior, we have noticed that the response time of a malicious request was more than that for a legitimate one. Therefore, we use the Average Response Time (ART) of consumer requests to identify the malicious name prefixes. Such an ART represents the time from when the interests are sent until a response is received by NDN routers. Calculating the ART of each name prefix sent by an NDN router can accurately detect the malicious name prefixes.

Upon receiving each interest packet, the NDN router calculates the response time for each sending request by recording the forwarding time of an interest packet and the receiving time of its corresponding data packet. Then, the prefix name of the requested content will be extracted and the ART for this prefix will be calculated according to Equation 4, where $T^p_{avg}(t-1)$ is the previous average response time of prefix p, T^p_{cur} is the recently measured response time, and $T^p_{avg}(t)$ is the new average response time value.

$$T^p_{avg}(t) = (1-\alpha)\ T^p_{avg}(t-1) + \alpha\ T^p_{cur} \tag{4}$$

When the ART of a prefix name exceeds the threshold value T_{th}, it will be marked as a malicious prefix. A list of malicious prefixes will be constructed by the NDN router, which will be used then to discard the subsequent malicious interests when the CUSUM detection algorithm detects a CIFA attack. A pseudocode for detecting the malicious name prefixes is described in Algorithm 1.

Algorithm 1 Detecting Malicious Name Prefixes

```
Input: Threshold value $T_{th}$
Output: A set of malicious name prefixes $P_m$
Initialize $P_m \leftarrow \phi$
for each interest packet $I$ received by router $R$ do
    Forward interest $I$
    Record the forwarding time $t_f$
end for
for each data packet $D$ received by router $R$ do
    Record the receiving time $t_r$
    Calculate the response time $T^p_{cur}$
    Extract the prefix name $P$
    Calculate $T^p_{avg}(t)$ of prefix $P$ according to 4
    if $T^p_{avg}(t) \leq T_{th}$ then
        $P_m \leftarrow P_m \cup \{P\}$
    end if
end for
```

When an NDN router detects the existence of a CIFA attack in the network, it will drop all incoming interest packets that belong to any malicious name prefix, as described in Algorithm 2. Discarding malicious interests before creating PIT entries in the PIT storage will highly enhance the PIT utilization rate. Also, the throughput of the network is supposed to be enhanced since NDN routers will not forward these interests anymore.

Since we take the average value of the response time for different consumer requests, this will eliminate the problem of delayed packets. The delay resulting from a specific number of packets due to congestion may slightly increase the average response time. However, it will not result in a misjudgment when the threshold value is chosen carefully. However, we set the threshold value in our simulation to 1.5 s, which is almost 37% of the interest lifetime.

A. ALGORITHMS TIME COMPLEXITY

Algorithm 1 has a latency of $2*P*t$ in the worst case. Where P denotes the number of packets received, and t represents the average response time value, which is constant. As a result, This algorithm has linear time complexity of O(P). On the other hand, Algorithm 2 has an O(n*m) time complexity in the worst case, where N denotes the size of the P_m set, and M represents the number of packages in interest. Hence, Algorithm 2 has approximately O(n^2), quadratic time complexity.

Algorithm 2 Mitigating CIFA Attacks

```
Input: A set of malicious name prefixes $P_m$
while receiving interest packets do
    for each interest packet $I$ do
        Extract the prefix name $P$
        if $P \in P_m$ then
            reject interest $I$
        else
            forward interest $I$
        end if
    end for
end while
```

VII. PERFORMANCE EVALUATION

As described earlier, CIFAs are launched by multiple malicious consumers along with a collusive producer that together consume the PIT resources of intermediate NDN routers, which primarily affect the forwarding of legitimate interest packets. Extensive simulations were done to first describe the effect of CIFA on throughput, PIT utilization rate, and satisfaction rate of NDN networks, and second to evaluate the performance of the proposed algorithms in detecting and mitigating CIFA.

A. SIMULATION ENVIRONMENT

Before getting into the details of CIFA effect and performance evaluation results, this section presents the simulation environment that was used to launch CIFA and implement detection and mitigation systems. The ndnSIM, release 2.8, was used to conduct CIFA to evaluate the performance of the proposed CUSUM algorithm in detecting and mitigating the attack. ndnSIM is an open-source platform used to simulate the large-scale NDN-related experiments; an extension from the Network Simulator 3 (NS3) [47]. One of the main characteristics of ndnSIM is the realistic integration with NDN prototypes and the NDN Forwarding Daemon (NFD), which ensures that simulations are reliable and applicable to work in real environments without any problems.Teble 2 mentions the list of modules used to implement the detection and mitigation mechanisms:

TABLE 2. Module names used to implement the detection and mitigation mechanisms using NDNSim.

Module name	New/Existed	Description
Forwarder	Existed	Implement forwarding mechanism
AppHelper	Existed	Install applications on nodes
StackHelper	Existed	Install NDN stack
Malicious-Producer	New	Represent the colluding server
Malicious-Consumer	New	Send malicious interests
Detection-Module	New	To detect CIFA attack
Mitigation-Module	New	To mitigate CIFA attack
L3Protocol	Existed	Tracing interest and data packets
RttMeanDeviation	Existed	Calculate RTT of interest packets

Two simulation topologies with different number of nodes were used to conduct the attack, the small-scale binary tree topology and a modified version of the large-scale rocketfuel AT&T topology. The rocketfuel topology represents a real network architecture, which provides accurate results in simulating CIFAs. As described in Figure 6, the binary tree consists of 16 user nodes, 8 gateway nodes, and 11 backbone nodes. Figure 7 describes the rocketfuel topology, which consists of 130 user nodes, 33 gateways, and 13 backbones. In the conducted topology, 25% of user nodes were selected randomly as malicious consumers to ensure the even distribution of attackers across the network. Two backbone nodes were selected randomly as legitimate and colluding servers.

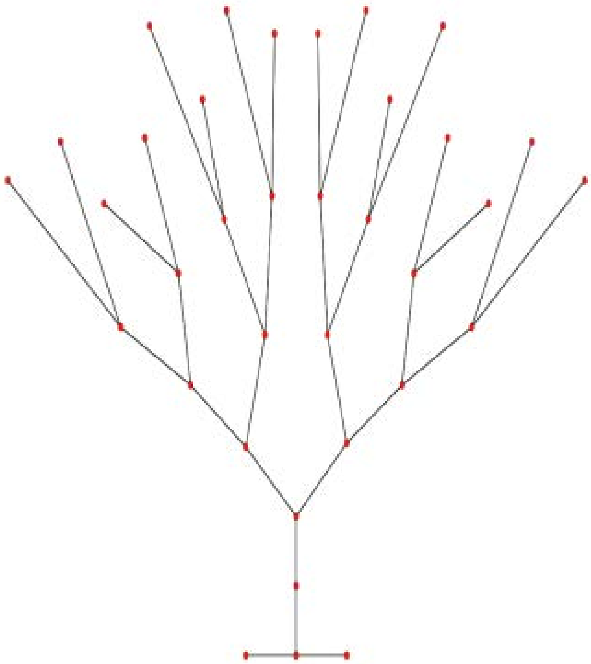


FIGURE 6. Binary tree topology.

To differentiate the interests coming from malicious or legitimate consumers, malicious interests start with the prefix /malicious/node_name, while the legitimate ones have the prefix /legitimate/node_name. For legitimate users, interests are sent at a constant rate, and the content distribution follows the Zipf-Mandelbort distribution. Usually, the sending rate of malicious users is very close to the rate for legitimate users, in which the effect of CIFAs becomes hidden by the attackers. In our simulation, we used the attacking parameters [4, 1, 50] and [4, 1, 10] in the binary and rocketfuel topologies, respectively. For the legitimate interest rate, we used a rate of 50 interests/s for a binary topology and 15 interests/s for the rocketfuel. A detailed description of the simulation parameters is summarized in Table 3.

B. CIFA IMPACT

As an initial step before detecting CIFA, we analyzed the traffic behavior of CIFA to understand the attacking nature and identify anomalous traffic patterns. In this section, an evaluation of the effect of CIFA is described in terms

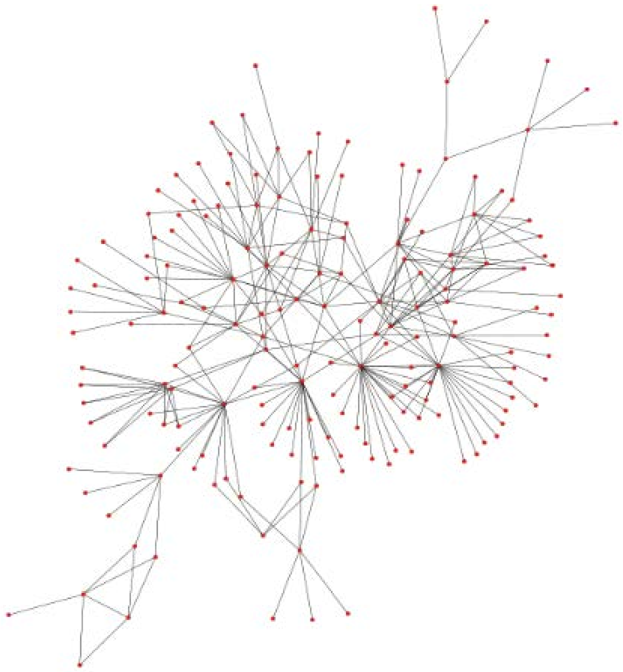


FIGURE 7. Rocketfuel AT&T topology.

TABLE 3. Simulation parameters description.

Parameter	Value
PIT size	200 entries
Interest lifetime	4s
Data packet size	1100 bytes
Legitimate request prefix	legitimate/node-name/...
Malicious request prefix	malicious/node-name/...
Simulated time	80s
Normal user request time	0-80s
Malicious user request time	40-80s
Attack sending rate (Binary)	[4, 1, 50]
Attack sending rate (AT&T)	[4, 1, 10]
Normal sending rate (Binary)	50 interests/s
Normal sending rate (AT&T)	15 interests/s

of Throughput, PIT Utilization Rate (PUR), Input Interest Rate (IIR), and Satisfaction Rate (SR). The simulation environment described in Section VII-A was implemented on ndnSIM to launch the attack. Following subsections describe in details the effect of CIFA on NDN.

Due to the large number of interest packets sent by malicious consumers at specific period, the PIT space of intermediate routers could be fully utilized, especially when the number of received interest packets exceeds the maximum number of entries in the PIT table. Figures 8a and 8b illustrate the PIT utilization rate for three backbone and one gateway routers in the binary and rocketcfuel topologies, respectively. As seen in the figures, the PIT utilization at normal operating conditions for the network did not exceed 12% for the two topologies. At second 41, the utilization started to increase severely in the network and reached 100% for some backbone routers due to the large sent interests during this period. At second 44, when the collusive producer started to send data packets, the utilization returned to its normal state for one to two seconds and then, started to increase once more. It is evident from the figures that the backbone routers reached higher utilization rates compared with gateway routers. Hence, it is readily surmised that backbone routers are more prone to CIFAs than other NDN routers.

The second metric that clearly describes the effect of CIFA is the network throughput. Figures 8c and 8d describe the throughput for three backbone and one gateway routers in the binary and rocketcfuel topologies, respectively. In all cases, we can see that the throughput was changing frequently across different attack periods. First, when the PIT was fully utilized at the beginning of each period, the throughput was largely reduced. When the collusive producer responded to the requests at the end of the period, the throughput returned to its normal state. One can immediately notice that the effect was quit dramatic on the bottleneck routers since all network traffic was passing through it. As seen in Figure 8c, the node bb-2 in the binary topology was severely affected by the attack since its throughput decreased from 4500 to 2000 Kbps. For the other nodes like bb-5 and gw-3, the effect was dismal.

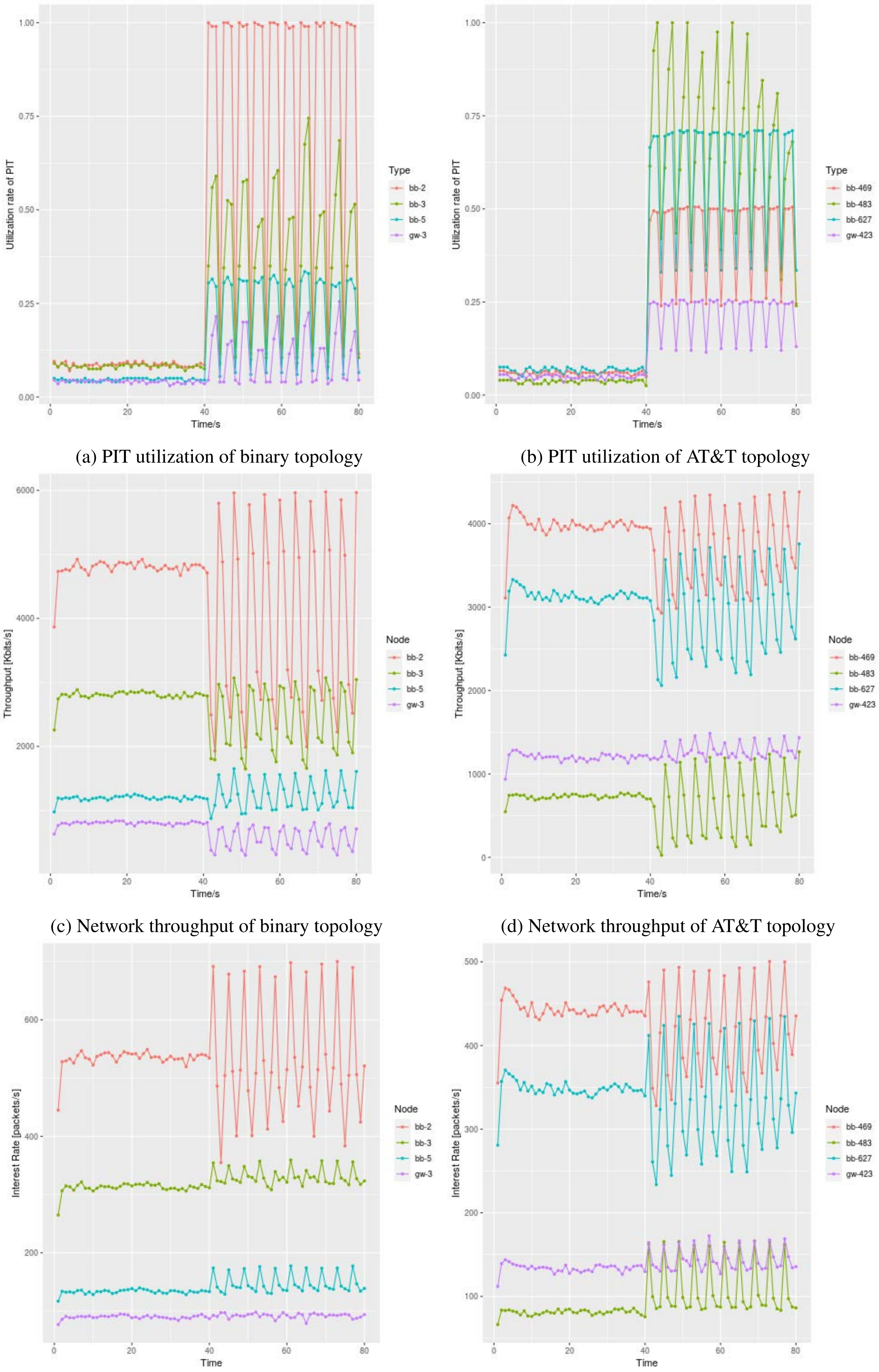
The input interest rate gives us an indication of received interest packets per second on NDN routers. As seen in Figures 8e and 8f, the IIR was changing frequently from second 40 to 80. In bb-2 of the binary topology, a normal rate would be around 530 packets/s under stable operating conditions. Meanwhile, during an attack it is shown to fluctuate periodically between 400 and 700 packets/s. Furthermore, it is evident that gw-3 has a stable IIR over the entire simulation period, which is attributed to the fact that the gateways do not hold the traffic of the entire network.

Another parameter that describes a CIFA impact is the satisfaction rate of the interest packets. Figures 8g and 8h describe the SR of legitimate consumers in the binary and rocketcfuel topologies, respectively. As seen from the figures, the satisfaction rate was between 80% and 100% at the beginning of the simulation. At second 41, it started to decrease gradually until reaching nearly 10% in the rocketfuel topology and 25% in the binary topology at the end of the attack period. This severe reduction results from a drop of legitimate interest packets sent by legitimate consumers. When the router reaches its full utilization rate, it starts to drop the received interest packets from all legitimate consumers, which largely reduces the SR. At the end of the attack period, the satisfaction rate starts to increase since the PIT entries of malicious consumers already start to be satisfied.

C. ASSESSMENT OF DETECTION SCHEME

For all the sequential change point detection algorithms, there are two main performance metrics that should be considered:

- **Detection Time (DT):** The delay from when the attack starts until it is detected by the detecting router.



(a) PIT utilization of binary topology

(b) PIT utilization of AT&T topology

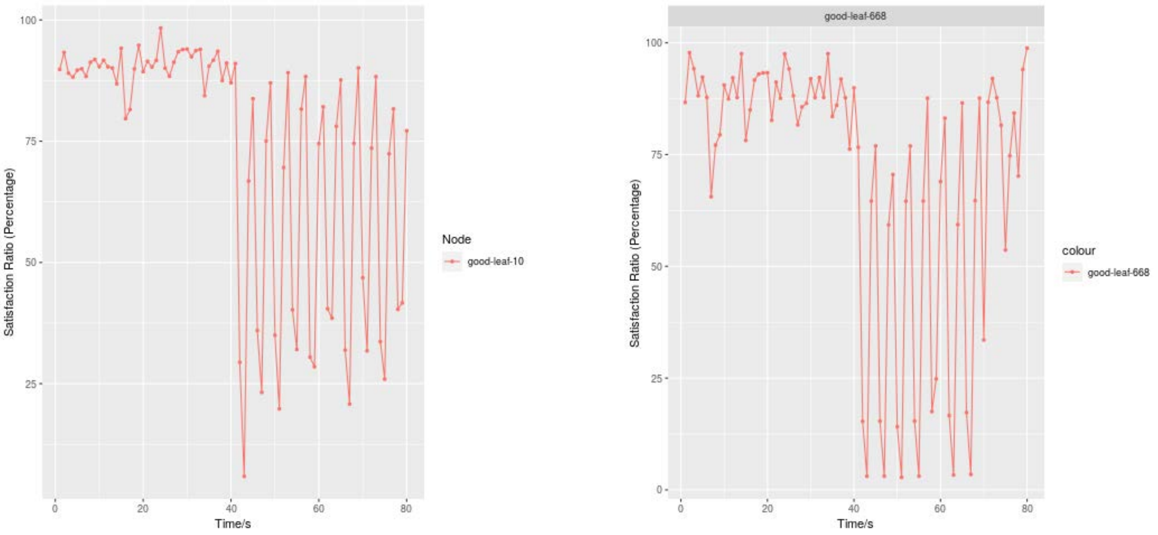
(c) Network throughput of binary topology

(d) Network throughput of AT&T topology

(e) Input interest rate of binary topology

(f) Input interest rate of AT&T topology

FIGURE 8. Evaluating the effect of CIFA on PIT utilization rate, network throughput, input interest rate, and consumer satisfaction rate of both binary and AT&T rocketfuel topologies.



(g) Satisfaction rate of binary topology

(h) Satisfaction rate of AT&T topology

FIGURE 8. ***(Continued.)*** **Evaluating the effect of CIFA on PIT utilization rate, network throughput, input interest rate, and consumer satisfaction rate of both binary and AT&T rocketfuel topologies.**

- **False Alarm Time (FAT):** The measured delay without false alarms under normal conditions when there is no CIFA attack.

We always tend to keep the first metric as short as possible while keeping the second metric as long as possible when detecting CIFA attack. The optimal detection algorithm maintains the lowest detection time among all other detection algorithms subject to some false alarm time.

In this section, we present the detection results and analysis for the CUSUM algorithm. We choose the routers gw-8 and bb-4 from the binary topology, and gw-423 and bb-627 from the rocketfuel topology as detecting routers. Figures 9 and 10 describe the cumulative sum value S_i for backbone and gateway routers, respectively. The red line represents the threshold value that should not be exceeded during normal network operation, which is 0.2 for gateways and 0.1 for backbones. As we explained earlier, see Section V-A, under normal conditions the mean values should not exceed the upper mean bound, which must always be around the zero value. In contrast, under abnormal conditions, the value S_i will still increase until exceeding the upper mean bound reaching a large positive value. From Figure 9, we see that all S_i values are zeros for gw-423, whereas there are three values in gw-8 exceed the threshold value.

In Figure 10, all the values are below the threshold value, except for one point which exceeds the threshold in the second 32.3 in bb-4. Since in gateway routers the flooding rate is very small, the difference between the number of data and interest packets is also small. Therefore, false alarms could be raised faster in gateways than in backbone routers.

To study the effect of increasing or decreasing the upper mean bound at the detection time and false alarm time, we simulated the network under different μ_0 values. Tables 4 and 5 presents the results of the DT and FAT for different μ_0 values in the binary and rocketfuel topologies, respectively. In most cases, when we increase the value of μ_0, the attack detection times increase since more time will be spent to reach the upper bound of the mean. As seen in the table, the DT increases from 199.8 ms to 520.3 ms at gw-423 when we change μ_0 from 0.05 to 0.2, as the DT increases from 87 ms to 499.5 ms for bb-627.

TABLE 4. **Results of DT after applying the CUSUM on binary topology for different μ_0 values.**

Detecting router	$\mu_0 = 0.1$	$\mu_0 = 0.2$	$\mu_0 = 0.4$
gw-8 ($2t_0$)	612 ms	**387 ms**	449.5 ms
bb-4 (t_0)	224.5 ms	**237 ms**	549.5 ms

TABLE 5. **Results of DT after applying the CUSUM on AT&T topology for different μ_0 values.**

Detecting router	$\mu_0 = 0.05$	$\mu_0 = 0.1$	$\mu_0 = 0.2$
gw-423 ($2t_0$)	199.8 ms	**242.5 ms**	520.3 ms
bb-627 (t_0)	87 ms	**199.5 ms**	499.5 ms

Figure 11 describes the effect of increasing μ_0 values on attack detection times for the rocketfuel topology. To clearly illustrate this relation, we plotted the values of S_i for different μ_0 values at gw-423, as described in Figure 12. For $\mu_0 = 0.05$, the algorithm took two sampling periods to detect the attack, as opposed to four sampling periods which had to pass before alerting against an attack for $\mu_0 = 0.2$. Increasing μ_0 value means that the upper bound for the normal mean increases, which requires more sampling periods to reach this upper bound. The accumulative sum will increase

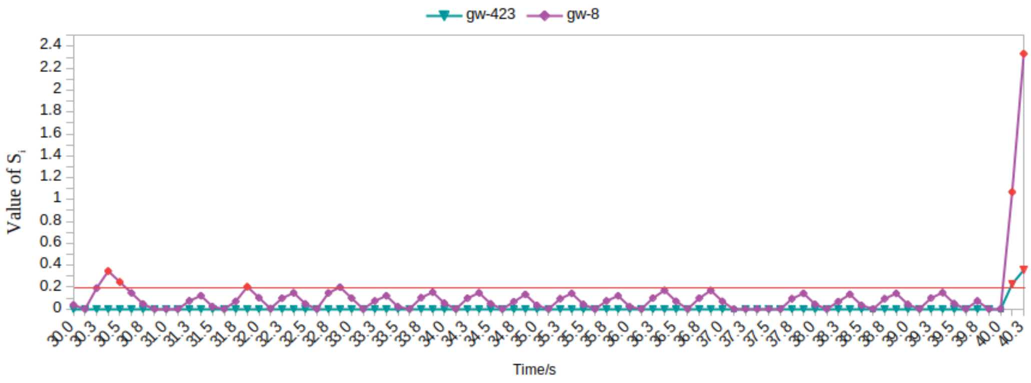


FIGURE 9. The Cumulative sum values at gateway routers.

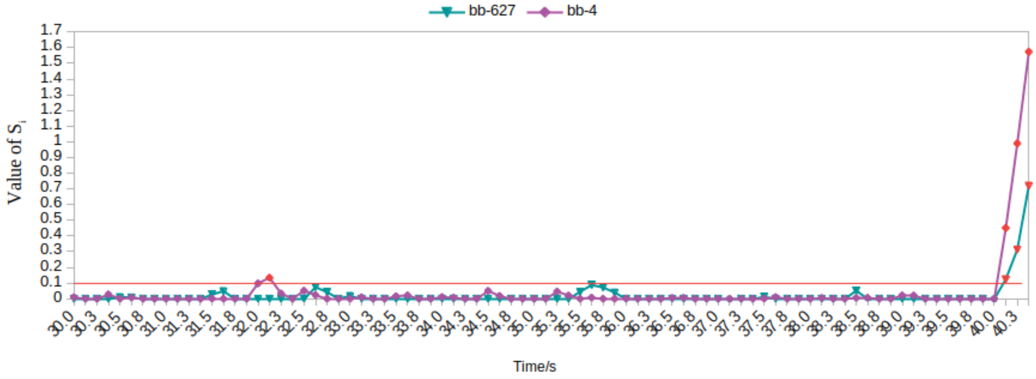


FIGURE 10. The Cumulative sum values at backbone routers.

gradually after the attack is launched to reach a value more than the threshold value that we set.

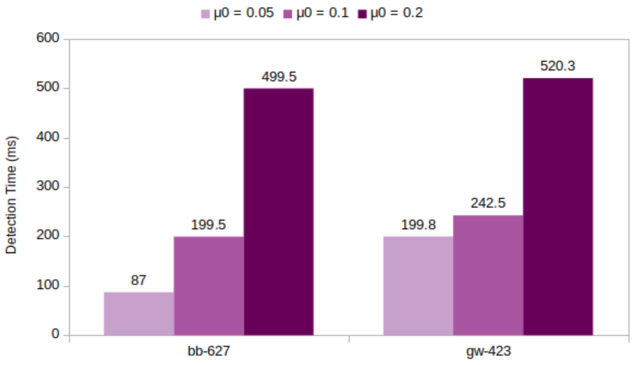


FIGURE 11. Detection time values for different μ_0 values in the rocketfuel topology.

Table 6 describes the false alarm time values for all detecting routers after applying the CUSUM algorithm on both topologies. Whenever we increase the μ_0 value, the FAT also increases. This is because increasing the upper mean bound of the CUSUM reduces the chance of wrongly considering a CIFA attack when there is no attack; hence increasing the FAT. We can see that when the μ_0 value is 0.1 in the binary topology, the largest time without false alarms is 15.3687 s at gw-8 and 25.604 s at bb-4. Whereas when the μ_0 value is 0.05 in the rocketfuel topology, the largest time without false alarms is 1.7122 s at bb-627. This means that the number of false alarms becomes very high when we set the upper mean bound to very small value. For $\mu_0 = 0.4$ and $\mu_0 = 0.2$, there are no false alarms since the value of the upper mean bound is very high. In our work, the least DT with the largest FAT is achieved when $\mu_0 = 0.2$ in the binary topology and $\mu_0 = 0.1$ in the rocketfuel topology.

TABLE 6. Results of FAT after applying the CUSUM on AT&T and binary topologies for different μ_0 values.

Topology	Detecting router	$\mu_0 = 0.1$	$\mu_0 = 0.2$	$\mu_0 = 0.4$
Binary	gw-8 ($2t_0$)	15.3687 s	–	–
	bb-4 (t_0)	25.604 s	–	–
Topology	Detecting router	$\mu_0 = 0.05$	$\mu_0 = 0.1$	$\mu_0 = 0.2$
AT&T	gw-423 ($2t_0$)	–	–	–
	bb-627 (t_0)	1.7122 s	31.5935 s	–

We study the performance of the proposed CUSUM algorithm in terms of Detection Time (DT), CPU Utilization (CPU-U). Since the CUSUM is an online sequential analysis approach where the traffic is monitored in real-time, any abnormal activities in the network would be detected

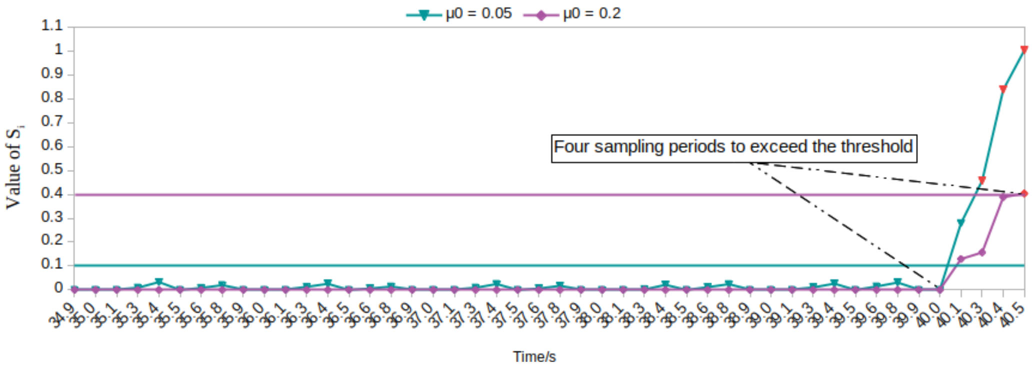


FIGURE 12. The Cumulative sum values at gw-423 for different μ_0 values.

quickly by the NDN router. Therefore, the time for detection is expected to be relatively low. In addition, as the CUSUM algorithm analyzes the traffic by accumulating the differences between interest and data packets, one of its characteristics is the associated low-computational overhead; hence, the reason for the low CPU utilization as compared with other detection algorithms.

To show the effectiveness of the CUSUM detection algorithm, the results were compared with the algorithm proposed in [28], which uses the rolling time window algorithm along with confidence intervals for detecting CIFA. Therein, the authors used two primary parameters to detect the CIFA attack, including the waiting time for PIT entries and network throughput. For each period, an NDN router will update the threshold values based on the confidence interval, which are calculated at the end of each window. The mean and variance of the two selected parameters will be used during each window to forecast the confidence interval for the next window. When the waiting time and throughput exceed the threshold values determined in the previous sliding window, the NDN router will output an alarm signaling the existence of a CIFA attack. The following subsections present the results for the two algorithms under consideration in terms of detection time and CPU utilization.

1) DETECTION TIME

One of the most important metrics in detecting any type of attack is the detection time, which, in our case, is the elapsed time between the start of a CIFA attack and its detection by the NDN router. The sooner we detect an attack, the sooner can respond to it an instate the necessary steps for mitigation. In this, we always attempt to achieve the lowest detection times as per our detection approach to speed the underlying response and mitigate the consequential effect across the network.

Table 7presents the detection time values in backbone routers for different detection algorithms. We can see from the table that CUSUM algorithm achieved the lowest detection time of 199.5 ms. Since the CUSUM algorithm can track small shifts in time-series data, the underlying attack is detected early on during the detection phase. In contrast, the rolling time window algorithm detects an attack based on two detection parameters: the throughput and PIT entries waiting time. Hence, the more significant detection times than those for the CUSUM algorithm.

TABLE 7. Detection time values of different detection detection schemes.

Detection method	Confidence interval [28]	Wavelet analysis [40]	CUSUM
Detection time	1001.8 ms	400ms	199.5ms

2) CPU UTILIZATION

As stated previously, one of the notable CUSUM characteristics is the low computational overhead compared with other statistical analysis algorithms. Therefore, the CPU utilization of the system was measured under both the CUSUM and rolling time window algorithms. Figure 13 illustrates the CPU utilization for both algorithms during 30 seconds of simulation time for the rocketfuel topology. As seen in the figure, the average CPU utilization for the CUSUM algorithm is around 45.5% as it is around 46% for the rolling time window algorithm. This is because the analysis in the CUSUM detection algorithm is done periodically on the traffic without tracking every sending interest in the network. In contrast, the rolling time window algorithm keeps a state for every PIT entry and calculates the PIT entries waiting times along with the network throughput upon receiving the interest packets to calculate the confidence interval levels.

D. ASSESSMENT OF MITIGATION SCHEME

Few research papers have adequately contributed to the mitigation of CIFA attacks. The coordinated monitoring architecture proposed by [38] used PIT utilization as an indicator to identify malicious name prefixes in the network. The mitigation was accomplished by discarding the interest packets that belong to the names marked by the monitoring NDN routers. [42] used cache reference value to identify malicious name prefixes since attackers usually request unpopular contents that are not requested by legitimate consumers. When the incoming rate of a name prefix exceeds the selected

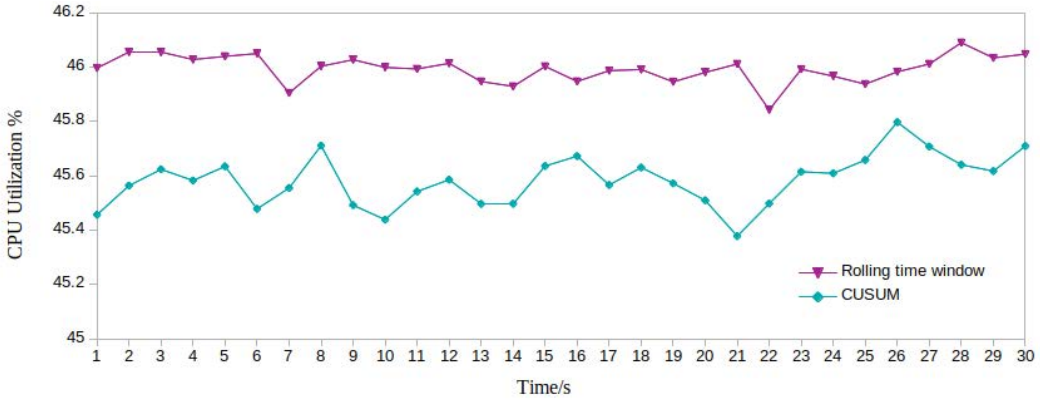


FIGURE 13. CPU utilization for the CUSUM and rolling time window algorithms on rocketfuel topology.

threshold value, its cache reference will be calculated for further analysis. Then, the NDN routers will discard the packets that belong to malicious name prefixes, which represent the names that maintain small reference values. However, an attacker may increase the requests on malicious content names to increase the cache reference value and, hence, bypass the detection algorithm.

Authors of [28] used the PIT occupancy rate along with the waiting time value of PIT entries to mitigate an attack. The NDN router periodically calculates the PIT usage when a CIFA attack is detected in the network. The PIT usage is calculated by subtracting the number of received data packets from the number of incoming interest packets for all interfaces and dividing the results by the PIT size. When the PIT occupancy rate exceeds a predefined threshold, N entries will be deleted from the PIT storage, where N is the number of PIT entries that maintain the most waiting time values. Here, the NDN router tries to reduce the PIT occupancy rate to a specific level, depending on a threshold value. Although the large waiting time values indicate the presence of a CIFA attack, deleting a specific number of malicious PIT entries may not significantly enhance the satisfaction rate and PIT utilization in the network. However, an attacker is still able to maintain the malicious entries in the PIT space. Our mitigation algorithm uses the average response time of consumer requests as an indicator to detect the malicious name prefixes and discard all incoming malicious interests in the network, as described in Section VI.

To evaluate the performance of our proposed mitigation algorithm, we compared our work with the one proposed by [28]. The performance of the proposed mitigation algorithm is measured in terms of Satisfaction Rate (SR), Network throughput, and PIT Utilization Rate (PUR). The following subsections provide a detailed description for these metrics along with the simulation results.

1) CPU UTILIZATION

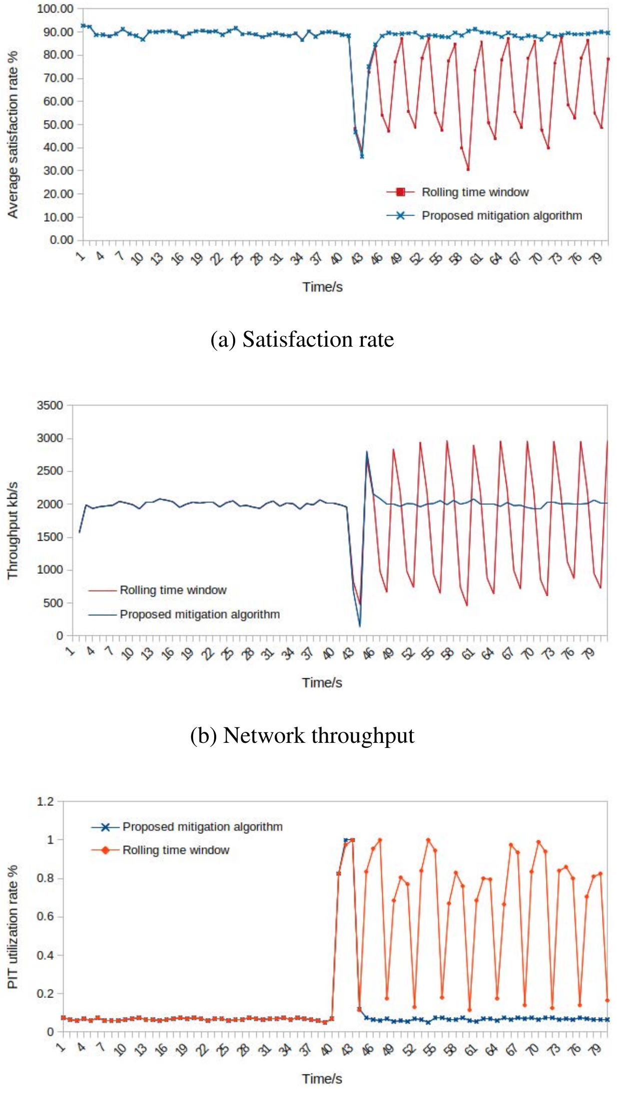
As stated previously, one of the notable CUSUM characteristics is the low computational overhead compared with other statistical analysis algorithms. Therefore, the CPU utilization of the system was measured under both the CUSUM and rolling time window algorithms. Figure 13 illustrates the CPU utilization for both algorithms during 30 seconds of simulation time for the rocketfuel topology. As seen in the figure, the average CPU utilization for the CUSUM algorithm is around 45.5% as it is around 46% for the rolling time window algorithm. This is because the analysis in the CUSUM detection algorithm is done periodically on the traffic without tracking every sending interest in the network. In contrast, the rolling time window algorithm keeps a state for every PIT entry and calculates the PIT entries waiting times along with the network throughput upon receiving the interest packets to calculate the confidence interval levels.

2) SATISFACTION RATE

Studying the satisfaction rate of user requests before and after taking the mitigation measures can accurately evaluate the effectiveness of the proposed mitigation algorithm. As described in Figure 8, the satisfaction rate of consumers under normal conditions was very close to 100%, while it reached 10% under a CIFA attack. The degradation in consumer satisfaction is the result of dropping legitimate interest packets when the PIT storage of NDN routers is fully utilized with malicious long-life entries. We use the following formula to calculate the average satisfaction rate in NDN, which will be used to evaluate our proposed algorithm in mitigating CIFA attacks:

$$S_i = \frac{\sum_{t=0}^{s} I^{s}_{InputInterests}}{\sum_{t=0}^{s} I_{InputInterests}} \tag{5}$$

where S_i is the Satisfaction rate of user i, $I^{s}_{InputInterests}$ is the number of input satisfied interests, and $I_{InputInterests}$ is the number of input interests within s seconds period. When there is no attack, most of the interest packets will be satisfied in the network since the PIT utilization will be in its normal state of all NDN routers. In the best case, the number of satisfied interests will be equal to the number of input interests and the satisfaction rate will be 1. When a CIFA attack is

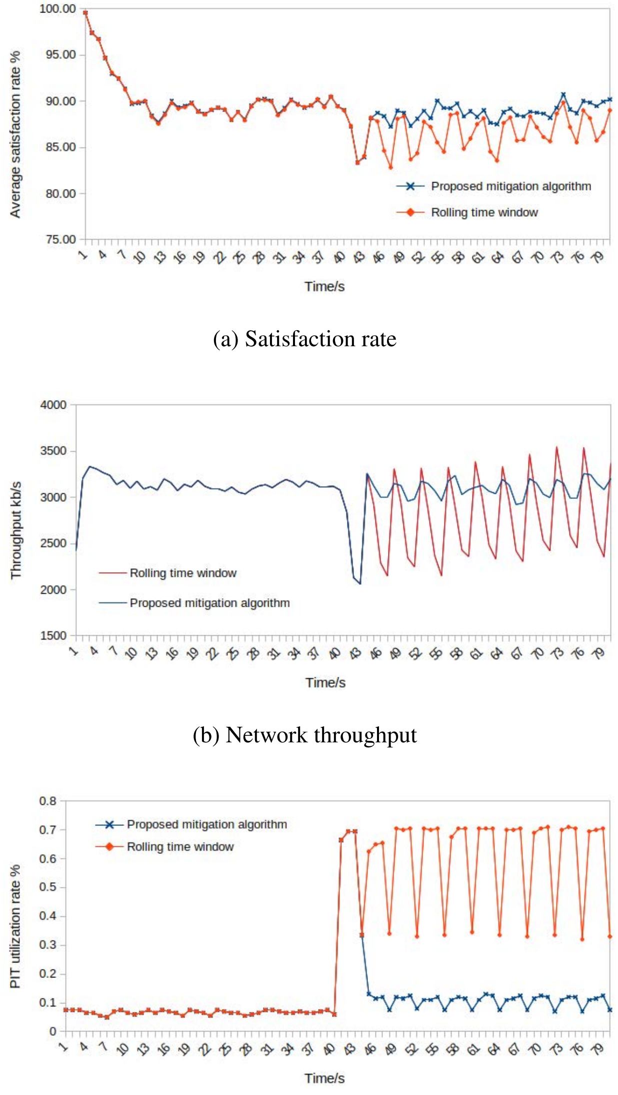


(a) Satisfaction rate

(b) Network throughput

(c) PIT utilization rate

FIGURE 14. Performance evaluation of the proposed mitigation algorithm based on the Average Response Time (ART) in binary topology.



(a) Satisfaction rate

(b) Network throughput

(c) PIT utilization rate

FIGURE 15. Performance evaluation of the proposed mitigation algorithm based on the Average Response Time (ART) in AT&T topology.

launched, the number of satisfied interests will decrease and the satisfaction rate will degrade gradually to reach a very low value. The average satisfaction rate in the NDN network could be calculated as follows, where n is the total number of legitimate consumers in the network:

$$S_{avg} = \frac{\sum_{i=0}^{n} S_i}{n} \tag{6}$$

After executing the mitigation measures to reduce the effect of CIFA attacks, the satisfaction rate is supposed to increase reaching a value very close to 1. It is always better to maintain high satisfaction rates for legitimate consumers, so they can send and receive their data effectively in the network. Figures 14a and 15a describe the average satisfaction rate for all legitimate consumers for both our proposed mitigation algorithm and rolling time window algorithm in the binary and AT&T topology, respectively. After mitigating the CIFA attack, the average satisfaction rate increased to 87.7% for our proposed algorithm and to 77.2% for the rolling time window algorithm in the binary topology. In the rocketfuel AT&T topology, the average SR increased to 89.4% for our algorithm and to 88.4% for the rolling time window algorithm. Once all the malicious interest packets are eliminated from the network after an attack detection, the NDN router is able to handle the legitimate interests properly in the network. In the rolling time window algorithm, only a specific number of PIT entries are deleted when the PIT usage exceeds a threshold value. Hence, the NDN router will still create new PIT entries even for a short time, which will not make a large increase in the satisfaction rate. In contrast, our proposed mitigation algorithm detects the malicious name prefixes and discards all incoming malicious interests, which inherently prevents the NDN router from making new PIT entries for those malicious interests.

3) THROUGHPUT

As described in Section VII-B, the throughput was not stable during the CIFA attacking period. As seen in Figures 14b and 15b, after applying our mitigation algorithm the throughput of the network returned to its normal state after almost three seconds from when the attack was launched. The NDN router was able to identify the malicious name prefixes and started rejecting all subsequent malicious interests coming from the attacking side in 3 seconds. Since the NDN router discarded all the malicious interests, no PIT entries were created, something that would allow the legitimate interests to be forwarded normally in the network. The average throughput was 1970.4 Kbps after applying our mitigation measures, as it was 1828.6 Kbps after applying the rolling time window algorithm.

4) PIT UTILIZATION RATE

The PIT space should not be fully utilized under normal network conditions. For example, the normal utilization rate in the experiment that was done in Section VII-B was less than 15%. Under a CIFA attack, however, the utilization rate reached 100% for some backbone nodes. After taking the mitigation measures, the PIT utilization rate is supposed to decrease reaching a value very close to the normal utilization rate in the network. The following formula represents the PIT utilization rate for an NDN router, where $P_{entries}(t)$ is the number of PIT entries at time t, and P_{size} is the PIT size:

$$PUR(t) = \frac{P_{entries}(t)}{P_{size}} \tag{7}$$

As seen in Figures 14c and 15c, the PIT utilization rate returned to its normal state when we applied our mitigation measures. The average utilization rates for our proposed mitigation algorithm were 10.04% and 11.15% in the binary and rocketfuel AT&T topology, respectively. Meanwhile, the average utilization rates for the rolling time window algorithm were 36.87% and 33.89%. Since our proposed algorithm identifies the malicious name prefixes and discards the interest packets before creating new PIT entries, the PIT utilization rate is highly reduced to reach the normal utilization rate value.

As a result, our proposed detection scheme is able to detect CIFA attacks within 199.5 ms in the large-scale topology based on the non-parametric CUSUM algorithm. The network throughput, average satisfaction rate, and PIT utilization rate returned to their normal state after applying our mitigation measures based on the average response time of consumer requests. The NDN router was able to identify the malicious name-prefixes and discard all malicious interest packets coming from different attacking sources. Finally, Our proposed defense scheme was implemented based on a statistical analysis process, which uses lightweight detection and mitigation algorithms and consumes fewer resources as compared with the works of other researchers.

VIII. CONCLUSION

This article proposed a resilient scheme based on the non-parametric CUSUM algorithm to detect and mitigate CIFA attacks. After studying the behavior of CIFA and analyzing the sequence of user requests, we found that the difference between sent interest packets and received data packets is large within a short period of time. This abnormal traffic pattern enabled us to use the statistical analysis algorithms to detect the point in time where an attack was launched. The CUSUM is a change point detection algorithm, where the abnormal changes in the time-series data is detected. Our proposed solution uses the frequent large differences between interest and data packets as an indicator to identify CIfA attacks. In addition, we used the average response time of interest packets as a parameter to identify malicious name prefixes to mitigate the effects of an attack. Our proposed detection scheme was able to detect an attack within 199.5 ms ms in the large-scale topology. After mitigating a CIFA attack, the average satisfaction rate, throughput, and PIT utilization rates returned to the normal network baseline.

IX. FUTURE WORK

As future work, we are going to implement a defense scheme against the Improved Collusive Interest Flooding Attack (I-CIFA), proposed by [13], which has a much higher impact than normal IFA attack and higher concealment than CIFA attack. In this type, the PIT capacity of NDN routing nodes is probed before starting the attack. Every attacking node has a different request mode, which makes the I-CIFA attack detection more difficult for detecting engines than the CIFA attack. Further, this new type reduces the attack cost of the attacker and causes more damage in NDN networks.

REFERENCES

[1] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 26–36, Jul. 2011.

[2] A. Kobusińska, C. Leung, C.-H. Hsu, S. Raghavendra, and V. Chang, "Emerging trends, issues and challenges in Internet of Things, big data and cloud computing," *Future Gener. Comput. Syst.*, vol. 87, pp. 416–419, Oct. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18311270

[3] A. Djama, B. Djamaa, and M. R. Senouci, "TCP/IP and ICN networking technologies for the Internet of Things: A comparative study," in *Proc. Int. Conf. Netw. Adv. Syst. (ICNAS)*, Jun. 2019, pp. 1–6.

[4] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, "An overview of security support in named data networking," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 62–68, Nov. 2018.

[5] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, doi: 10.1145/2656877.2656887.

[6] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *Proc. 22nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2013, pp. 1–7.

[7] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *Proc. IFIP Netw. Conf.*, May 2013, pp. 1–9.

[8] M. Wählisch, T. C. Schmidt, and M. Vahlenkamp, "Backscatter from the data plane—Threats to stability and security in information-centric network infrastructure," *Comput. Netw.*, vol. 57, no. 16, pp. 3192–3206, 2013, information Centric Networking. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613002259

[9] K. Wang, H. Zhou, Y. Qin, J. Chen, and H. Zhang, "Decoupling malicious interests from pending interest table to mitigate interest flooding attacks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 963–968.

[10] K. Ding, Y. Liu, H.-H. Cho, H.-C. Chao, and T. K. Shih, "Cooperative detection and protection for interest flooding attacks in named data networking," *Int. J. Commun. Syst.*, vol. 29, no. 13, pp. 1968–1980, Sep. 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.2883

[11] T. Zhi, H. Luo, and Y. Liu, "A Gini impurity-based interest flooding attack defence mechanism in NDN," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 538–541, Mar. 2018.

[12] A. Benarfa, M. Hassan, A. Compagno, E. Losiouk, M. B. Yagoubi, and M. Conti, "ChoKIFA: A new detection and mitigation approach against interest flooding attacks in NDN," in *Wired/Wireless Internet Communications*, M. Di Felice, E. Natalizio, R. Bruno, and A. Kassler, Eds. Cham, Switzerland: Springer, 2019, pp. 53–65.

[13] Z. Wu, W. Feng, J. Lei, and M. Yue, "I-CIFA: An improved collusive interest flooding attack in named data networking," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102912. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212621001356

[14] R.-T. Lee, Y.-B. Leau, Y. J. Park, and M. Anbar, "A survey of interest flooding attack in named-data networking: Taxonomy, performance and future research challenges," *IETE Tech. Rev.*, pp. 1–19, 2021, doi: 10.1080/02564602.2021.1957029.

[15] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, and L. Zhang, "A brief introduction to named data networking," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 1–6.

[16] C. Katsis, A. Singla, and E. Bertino, "Real-time digital signatures for named data networking," in *Proc. 7th ACM Conf. Information-Centric Netw.*, New York, NY, USA, Sep. 2020, pp. 149–151, doi: 10.1145/3405656.3420227.

[17] L. Wang, Z. Zhang, M. Dong, L. Wang, Z. Cao, and Y. Yang, "Securing named data networking: Attribute-based encryption and beyond," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 76–81, Nov. 2018.

[18] V. Stocker, G. Smaragdakis, W. Lehr, and S. Bauer, "The growing complexity of content delivery networks: Challenges and implications for the internet ecosystem," *Telecommun. Policy*, vol. 41, pp. 1003–1016, Mar. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308596116302476

[19] N. Aloulou, M. Ayari, M. F. Zhani, L. Saidane, and G. Pujolle, "Taxonomy and comparative study of NDN forwarding strategies," in *Proc. 6th Int. Conf. Commun. Netw. (ComNet)*, Mar. 2017, pp. 1–8.

[20] J. H. Ran, N. Lv, D. Zhang, Y. Y. Ma, and Z. Y. Xie, "On performance of cache policies in named data networking," in *Proc. Int. Conf. Adv. Comput. Sci. Electron. Inf.*, 2013, pp. 668–671.

[21] P. F. Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt, and M. Wählisch, "The missing piece: On namespace management in NDN and how DNSSEC might help," in *Proc. 6th ACM Conf. Inf.-Centric Netw.*, New York, NY, USA, 2019, pp. 37–43. [Online]. Available: https://doi.org/10.1145/3357150.3357401

[22] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang, "NDNS: A DNS-like name service for NDN," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2017, pp. 1–9.

[23] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "SNAMP: Secure namespace mapping to scale NDN forwarding," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 281–286.

[24] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: A survey," *Comput. Sci. Rev.*, vol. 19, pp. 15–55, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013715300599

[25] H. Im and D. Kim, "An overview of content poisoning in NDN: Attacks, countermeasures, and direction," *KSII Trans. Internet Inf. Syst. (TIIS)*, vol. 14, no. 7, pp. 2904–2918, 2020.

[26] S. Al-Sheikh, M. Wählisch, and T. C. Schmidt, "Revisiting countermeasures against NDN interest flooding," in *Proc. 2nd ACM Conf. Inf.-Centric Netw.*, New York, NY, USA, Sep. 2015, pp. 195–196, doi: 10.1145/2810156.2812604.

[27] J. Lrt, L. Yu Beng, Y. Park, and M. Anbar, "Capturing collusive interest flooding attacks signal: A novel Malaysia's state named-data networking topology (MY-NDN)," *J. Eng. Sci. Technol.*, vol. 17, pp. 997–1009, 04 2022.

[28] Z. Wu, W. Feng, M. Yue, X. Xu, and L. Liu, "Mitigation measures of collusive interest flooding attacks in named data networking," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101971. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820302443

[29] R. Hou, M. Han, J. Chen, W. Hu, X. Tan, J. Luo, and M. Ma, "Theil-based countermeasure against interest flooding attacks for named data networks," *IEEE Netw.*, vol. 33, no. 3, pp. 116–121, May 2019.

[30] J. Dong, K. Wang, W. Quan, and H. Yin, "InterestFence: Simple but efficient way to counter interest flooding attack," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101628. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404819301750

[31] A. Benmoussa, A. E. K. Tahari, C. A. Kerrache, N. Lagraa, A. Lakas, R. Hussain, and F. Ahmad, "MSIDN: Mitigation of sophisticated interest flooding-based DDoS attacks in named data networking," *Future Gener. Comput. Syst.*, vol. 107, pp. 293–306, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X19328729

[32] G. Cheng, L. Zhao, X. Hu, S. Zheng, H. Wu, R. Li, and C. Fan, "Detecting and mitigating a sophisticated interest flooding attack in NDN from the network-wide view," in *Proc. IEEE 1st Int. Workshop Netw. Meets Intell. Computations (NMIC)*, Jul. 2019, pp. 7–12.

[33] C. Pu, N. Payne, and J. Brown, "Self-adjusting share-based countermeasure to interest flooding attack in named data networking," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2019, pp. 142–147.

[34] Y. Xu, T. Xu, and X. Xu, "Research on detection method of interest flooding attack on content centric network," *Comput., Mater. Continua*, vol. 64, no. 2, pp. 1075–1089, 2020. [Online]. Available: http://www.techscience.com/cmc/v64n2/39347

[35] G. Xing, J. Chen, R. Hou, L. Zhou, M. Dong, D. Zeng, J. Luo, and M. Ma, "Isolation forest-based mechanism to defend against interest flooding attacks in named data networking," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 98–103, Mar. 2021.

[36] A. Benarfa, M. Hassan, E. Losiouk, A. Compagno, M. B. Yagoubi, and M. Conti, "ChoKIFA+: An early detection and mitigation approach against interest flooding attacks in NDN," *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 269–285, Jun. 2021.

[37] M. Alhisnawi and M. Ahmadi, "Detecting and mitigating DDoS attack in named data networking," *J. Netw. Syst. Manage.*, vol. 28, no. 4, pp. 1343–1365, Oct. 2020.

[38] H. Salah and T. Strufe, "Evaluating and mitigating a collusive version of the interest flooding attack in NDN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 938–945.

[39] H. Salah, J. Wulfheide, and T. Strufe, "Coordination supports security: A new defence mechanism against interest flooding in NDN," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Oct. 2015, pp. 73–81.

[40] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, "Detection of collusive interest flooding attacks in named data networking using wavelet analysis," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2017, pp. 557–562.

[41] L. Liu, W. Feng, Z. Wu, M. Yue, and R. Zhang, "The detection method of collusive interest flooding attacks based on prediction error in NDN," *IEEE Access*, vol. 8, pp. 128005–128017, 2020.

[42] T. Shigeyasu and A. Sonoda, "Distributed approach for detecting collusive interest flooding attack on named data networking," in *Advances in Network-Based Information Systems*, L. Barolli, N. Kryvinska, T. Enokido, and M. Takizawa, Eds. Cham, Switzerland: Springer, 2019, pp. 76–86.

[43] A. Nasserala, I. V. Bastos, and I. Monteiro Moraes, "Cache nFace: A simple countermeasure for the producer-consumer collusion attack in named data networking," *Ann. Telecommun.*, vol. 74, nos. 3–4, pp. 125–137, Apr. 2019.

[44] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, nos. 1–2, pp. 100–115, 1954. [Online]. Available: http://www.jstor.org/stable/2333009

[45] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. vol. 104. Upper Saddle River, NJ, USA: Prentice-Hall, 1993, [Online]. Available: hftp://ftp.irisa.fr/local/as/mb/k11.pdf

[46] H. Wang, D. Zhang, and K. G. Shin, "Detecting SYN flooding attacks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 3, Jun. 2002, pp. 1530–1539.

[47] A. Alexander, I. MOISEENKO, and L. Zhang, "ndnSIM: NDN simulator for NS-3," Tech. Rep. NDN-0005, 2012.

• • •