Interest Flooding Attacks in Named Data Networking: Survey of Existing Solutions, Open Issues, Requirements and Future Directions

AHMED BENMOUSSA, CHAKER ABDELAZIZ KERRACHE, and NASREDDINE LAGRAA, Laboratoire d'Informatique et de Mathématiques, Université Amar Telidji de Laghouat, Algeria
SPYRIDON MASTORAKIS, University of Nebraska at Omaha, USA
ABDERRAHMANE LAKAS, College of Information Technology, United Arab Emirates University, UAE
ABDOU EL KARIM TAHARI, Université Amar Telidji de Laghouat, Algeria

Named Data Networking (NDN) is a prominent realization of the vision of Information-Centric Networking. The NDN architecture adopts name-based routing and location-independent data retrieval. Among other important features, NDN integrates security mechanisms and focuses on protecting the content rather than the communications channels. Along with a new architecture come new threats and NDN is no exception. NDN is a potential target for new network attacks such as Interest Flooding Attacks (IFAs). Attackers take advantage of IFA to launch (D)DoS attacks in NDN. Many IFA detection and mitigation solutions have been proposed in the literature. However, there is no comprehensive review study of these solutions that has been proposed so far. Therefore, in this paper, we propose a survey of the various IFAs with a detailed comparative study of all the relevant proposed solutions as counter-measures against IFAs. We also review the requirements for a complete and efficient IFA solution and pinpoint the various issues encountered by IFA detection and mitigation mechanisms through a series of attack scenarios. Finally, in this survey, we offer an analysis of the open issues and future research directions regarding IFAs.

CCS Concepts: • **Networks** → **Denial-of-service attacks**; • **Security and privacy** → **Denial-of-service attacks**.

Additional Key Words and Phrases: Named Data Networking (NDN), Interest Flooding Attack (IFA), Survey.

1 INTRODUCTION

A long time has passed since the creation of the Internet. Back then, the goal was to interconnect pairs of hosts. With the constant growth of the connected devices that will reach almost 30 billion in 2023 [1], which represents roughly three times the global population, the actual Internet architecture was not designed for such massive numbers of hosts. In addition, Internet usage itself has changed. Users are interested in the content to retrieve rather than its source. In addition, security was an afterthought when the Internet was created, which opened the door to security and privacy issues. Taking all these challenges in mind, the need for a new, suitable, and secure Internet architecture is essential. The research community started the discussion on a new architecture about three decades ago [25]. Since then, many Future Internet Architectures (FIAs) were proposed like Xtensible Internet Architecture (XIA) [16], Nebula [17], and others. However, the most promising FIA architecture candidate is

Authors' addresses: Ahmed Benmoussa, ah.benmoussa@lagh-univ.dz; Chaker Abdelaziz Kerrache, ch.kerrache@lagh-univ.dz; Nasreddine Lagraa, n.lagraa@lagh-univ.dz, Laboratoire d'Informatique et de Mathématiques, Université Amar Telidji de Laghouat, Laghouat, Algeria; Spyridon Mastorakis, smastorakis@unomaha.edu, University of Nebraska at Omaha, Omaha, Nebraska, USA; Abderrahmane Lakas, alakas@uaeu.ac.ae, College of Information Technology, United Arab Emirates University, Al Ain, UAE; Abdou el Karim Tahari, k.tahari@lagh-univ.dz, Université Amar Telidji de Laghouat, Laghouat, Algeria.

0360-0300/2022/6-ART $15.00
https://doi.org/10.1145/3539730

Information Centric Networking (ICN) [11, 100]. Several architectures were proposed under the umbrella of ICN, like Data-Oriented Network Architecture (DONA) [47], Publish-Subscribe Internet Technology (PURSUIT) [36], Scalable & Adaptive Internet soLutions (SAIL), COntent Mediator architecture for content-aware nETworks (COMET) [37], and MobilityFirst [75]. But the most promising information-centric architecture is Named Data Networking (NDN). NDN is a project funded by the US Future Internet Architecture in 2010 and maintained at UCLA [107]. NDN takes its roots from the Content-Centric Networking (CCN) [43].

NDN adopts a content-driven communication approach where packet forwarding is based on data names rather than IP addresses. NDN also provides features such as in-network caching, built-in multicast, mobility support, and native security mechanisms. NDN focuses on securing the content rather than the communication channels. NDN mandates the use of data signatures, which permits users to retrieve any available piece of content no matter where it comes from, as long as the signature can be verified. Although NDN integrates security mechanisms, it is still not immune to certain new security and privacy issues [46, 85]. One of these network threats is related to Interest Flooding Attacks (IFAs). IFA is a new type of attack that adversaries use to launch (D)DoS attacks in NDN.

This survey provides an in-depth study of IFA and gives a broad analysis of IFA solutions that have been proposed so far before pointing out the open issues and providing the research directions that need to be considered in the future. Before we dive deep into the subject, we first review existing surveys and discuss the importance and novelty of our survey compared to the literature.

1.1 Related Surveys

Several surveys about NDN and ICN security issues have been authored. Some of them briefly discussed IFA, while others detailed IFA. In the following subsection, we summarize and explain the differences between each authored survey. The authors of [6] surveyed several ICN attacks and grouped them into four different categories: naming, routing, caching, and miscellaneous. The survey briefly discussed IFA and the number of cited solutions is very limited. Similarly, the authors of [85] classified ICN threats into three categories: security, privacy, and access control. The authors defined IFA as a DoS attack and reviewed some countermeasures. However, the authors did not compare in detail the reviewed works. The survey in [58] discussed security threats and vulnerabilities in ICN. It classified the attacks into three categories: security in content, routers, and caches. It discussed IFA and presented several IFA solutions. Despite lacking many relevant works, this survey did not detail the cited solution, and the provided review does not mention their drawbacks. The authors of [5] presented a survey about DoS attacks in CCN and classified them into three categories: flooding, forced computation, and cache/content manipulation. The authors discussed IFA and gave a detailed review of some countermeasure solutions. However, this survey did not talk about their drawbacks. Also, this survey does not compare the cited works. The authors in [57] stated the benefits of the CCN paradigm on some security and privacy threats. Then, they outlined the actual challenges that need to be corrected. They talked about IFA and mentioned few countermeasure solutions without giving any comparison. The survey in [15] compared four FIA architectures: Nebula, NDN, MobilityFirst, and XIA in terms of integrity, confidentiality, availability, authentication, trust, access control, and anonymity. This paper shortly discussed IFA and the number of cited works is very limited. Similarly, the survey [27] provides a short review about some NDN security issues before briefly discussing IFA without mentioning any solution. The authors of [22] classified NDN attacks according to the targeted layer. They classified IFA as an application layer attack and presented a few countermeasure solutions. However, the list of covered solutions is poor and misses recent and relevant works. In another context, the authors of [46] presented different security and privacy issues in Vehicular NDN (VNDN). This paper shortly discussed IFA and mentioned only two countermeasure solutions. The work presented in [38] focuses on IFA and privacy attacks in NDN/CCN. It classified solutions into simple techniques, Anomaly/Attack detection, and PIT-less routing. The provided analysis and the list of IFA

Table 1. Related Surveys

Ref	Year	Main focus	Limitations	Covered IFA solutions
[5]	2015	DoS attacks in CCN	• Does not compare cited IFA solutions. • Does not talk about the drawbacks of cited solutions.	09
[6]	2015	ICN attacks	• Briefly talks about IFA. • The number of IFA solution is limited.	04
[27]	2015	Security issues in NDN	• Do not mention IFA countermeasure solutions.	00
[57]	2016	Security and privacy challenges in CCN	• Does not talk about the drawbacks of cited solutions. • Did not provide a comparison between cited IFA solutions.	08
[85]	2017	ICN attacks	• Does not compare in detail cited IFA solutions	08
[15]	2018	Security and privacy in FIA architectures	• IFA was very shortly discussed. • Very limited IFA solutions.	03
[38]	2018	IFA and privacy in NDN/CCN	• Does not compare cited solutions. • The paper misses relevant works.	06
[46]	2018	Security and privacy issues in VNDN	• Briefly discuss IFA. • The number of cited solutions is very limited.	02
[69]	2018	IFA in NDN	• The survey misses a lot of relevant works. • The comparison given is very limited. • Does not mention the drawbacks of solutions.	10
[48]	2019	Security threats in NDN	• The survey misses some recent relevant works. • Does not compare in details the cited solutions. • Does not talk about the drawbacks of every solutions.	21
[58]	2019	Security threats in ICN	• Does not detail cited solution. • Does not mention solutions' drawbacks. • Misses many recent relevant research.	10
[70]	2019	IFA in NDN	• Does not compare between the cited solutions. • Misses many recent relevant research.	06
[22]	2020	Security threats in NDN	• Briefly discuss IFA. • The number of cited solutions is very limited.	05
Our	2021	IFA in NDN		43

work are limited. It lacks recent and relevant works. The authors of [69] and [70] focused on IFA and gave a short comparison between some existing solutions. However, the authors did not review in detail cited works. Also, these two surveys miss many recent and relevant works. The survey in [48] focuses on IFA and cache/content attacks. It described IFA and its variants before reviewing and classifying several solutions. Although it provides a long list of IFA solutions, this survey does not cover the full spectrum of IFA works and misses many recent and relevant works. Also, the authors did not review in detail cited solutions. We summarized the above surveys in Table 1.

1.2 Motivation and Goal of the paper

Although the previously authored surveys talk about IFA and some countermeasure solutions, they only scratch the surface of the subject. These surveys fail to cover all the aspects of the attack. Therefore, the authors only focused on presenting the solutions without deep analysis and comparison. In addition, the mentioned surveys do not study the full spectrum of IFA variants and scenarios. They generally describe the conventional version of the attack. Moreover, the present surveys do not cover all the works present in the literature. For instance, the

majority of recent (and relevant) works were not considered by these surveys. Considering all these limitations, there is a need for a survey that provides an in-depth study of IFA to bridge this gap. There is a need for a paper that provides a basic understanding of the attack, its variants, its characteristics, and the different techniques used to conduct such attacks. Furthermore, there is a need to provide a systematic and detailed review of all the countermeasure works present in the literature. In addition, a comparative study needs to be conducted to unravel the strength and weaknesses of each solution. Finally, there is a need to pinpoint the actual open issues and the lessons learned for future research directions.

1.3 Our Contributions

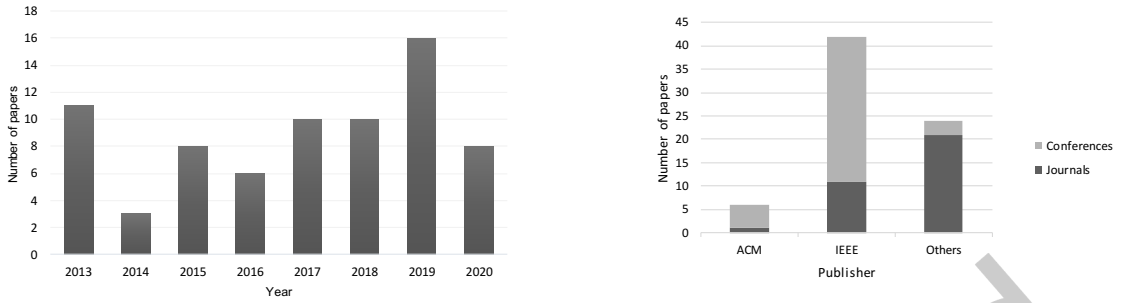
Existing surveys in the literature are not IFA specific or do not discuss and compare in detail related IFA solutions. To the best of our knowledge, our work is the first attempt that comprehensively and systematically review all the aspects of the Interest Flooding Attack. The highlights of the various aspects covered in this work are summarized below:

- Our survey provides a comprehensive discussion on state-of-the-art IFA research and an exhaustive research taxonomy of IFA by considering its working principles and approaches.
- Our survey reviews all the relevant IFA countermeasure solutions. It discusses the drawbacks of each one and presents a workflow that every IFA countermeasure solution follows.
- Our survey provides a comprehensive comparison of existing solutions. This comparison takes into consideration different aspects: the type of collected information, the calculated metrics, the detection parameters, and the defensive actions.
- Our survey provides an extensive list of non-conventional attack scenarios that existing works cannot handle and existing literature has not considered. Il also presents directions for future research associated with designing efficient and robust IFA solutions.

1.4 Methodology of Survey on Interest Flooding Attack in Named Data Networking

In this subsection, we present the methodology that we utilized during our review of the state-of-the-art Interest Flooding Attack. The process that we adopted is described below:

- **Surveyed Databases** To achieve our objective and cover all the relevant IFA-related research, we have collected papers on domain-relevant electronic databases, including *ACM Digital Library*, *IEEE Xplore*, *Science Direct*, *Springer Link*, *Wiley Online Library*, and *Hindawi*. Furthermore, we have collected articles related to this domain from *arXiv*, *Google Scholar*, and *ResearchGate*. Figure 1(b) illustrates the distribution of collected IFA works depending on their publisher.
- **Filtering and Paper Selection Process** The filtering process begins by categorizing each paper depending on its nature: a study paper, a survey paper, or a solution. The study papers category includes any IFA based study authored. It may include studies on the effects of IFA or a comparative study between some solutions, ...etc. Regarding survey papers, we chose to cite any survey that mentions (briefly or extensively) IFA, whether it is NDN specific or an ICN survey. However, for the authored solutions, we chose a different approach. Before we cite a paper in this survey, we first read the proposed technique and then verify if we have already reviewed a similar solution. If a similar solution exists, we compare the authors of these two papers. If they are the same, we choose to cite only the most recent one (e.g., the authors presented it in a conference paper and then published the same solution in an article paper). However, if the authors are different, we choose to cite the oldest research paper. On the other hand, If a given paper presents an enhanced or a modified technique to another published solution, we cite them consecutively. During our research, we collected 97 IFA based works. In the end, we chose 72 papers. Figure 1(a) illustrates the distribution of cited works since the research community started studying IFA.



(a) Distribution of cited works per year

(b) Distribution of cited works per publisher

Fig. 1. Statistics on cited IFA research papers

1.5 Organization of the Survey

We organize this survey in a top-down manner. We begin with an overview of the NDN architecture in section 2. We talk about the key features of this architecture before concluding this section with its security components. Following that, we introduce the availability attacks that target the NDN architecture in section 3. We discuss each attack and finish this section by briefly presenting the Interest Flooding Attack. In section 4, we dive into the main subject of this survey and detail IFA. We show the characteristics of the attack and its variants before giving an overview of the IFA related studies present in the literature. Section 5 focuses on IFA solutions that were authored and gives a detailed discussion on each solution. Section 6 introduces the Collect-Detect-Act workflow (CDA). This section presents an in-depth study of state-of-the-art solutions from different perspectives. Section 7 discusses unconsidered attacking scenarios. This section proposes several attacking scenarios that were not considered by existing solutions. Section 8 highlights the lessons learned and future research directions that need to be considered when designing solutions. Section 9 concludes this survey.

2 NAMED DATA NETWORKING: ARCHITECTURE DESIGN

NDN is a data-centric Internet architecture designed to replace the host-centric TCP/IP architecture [44, 106, 107]. NDN falls under the umbrella of Information Centric Networking (ICN) where the focus is on the data rather than its location. NDN defines two entities: *Producer* and *Consumer*. Producers generate and offer content for the consumers to request. To fetch for a content, consumers send the name of the desired data into an *Interest* packet. NDN adopts a hierarchical naming scheme to identify content [9]. According to the latest NDN packet specifications [3], every *Interest* packet must have a *Name*. It represents the name of the content that the consumer is requesting. *Interest* packets can also contain optional parameters. Furthermore, *Interest* packets can also be signed when needed [4]. NDN's *Data*[1] packets are the response that consumers expect when they send *Interest* packets. Each *Data* packet must contain the *Name*, the content, and a signature [2]. NDN *Data* packets identify four principal content types. First is the *BLOB* type, which represents the default content type. The second is the *LINK* type, which is used to include a list of producers. This packet is used for forwarding purposes. The third is the *KEY* type, used to represent a certificate. Finally, the *NACK* type designates application-based *NACK* packets.

[1]In this paper, we use upper-case *Data* to refer to the NDN's specific content packet.

2.1 NDN Node Model

The NDN protocol stack is composed of four different layers: application, network, link and physical layers [104]. The application layer supports the operation of NDN applications and it also embeds transport protocols as system libraries. The network layer's role is to route *Interest* and *Data* packets. It uses the application layer's names to route the packets. The NDN link-layer supports a set of protocols like Ethernet and can use virtual links like IP and TCP overlays. Each node with the NDN stack maintains three data structures:

2.1.1 Pending Interest Table (PIT). PIT is a data structure where are stored all the not yet satisfied *Interest* packets. Every pending *Interest* packet is stored in PIT until a *Data* packet returns or it times out. Each PIT entry contains the following fields: the name of requested *Data* (*Interest* packet's name), the incoming interface(s), outgoing interface(s) and an expiry timer.

2.1.2 Content Store (CS). Each NDN node can store the passing *Data* packets in a local cache [23]. This enables NDN to offer in-network caching. Requests can be satisfied by an intermediate cache without going down to the source of the *Data* packet, which reduces time retrieval and saves links bandwidth. Each CS need to implement a caching policy to maintain its size and keep the most relevant and popular data. Many caching policies can be used, including but not limited to, FIFO (First In First Out), LRU (Least Recently Used) and LFU (Least Frequently Used) [33, 108].

2.1.3 Forwarding Information Base (FIB). FIB is used to forward incoming *Interest* packet to upstream nodes. Unlike IP networks, NDN's FIB entries are indexed with name prefixes instead of IP addresses [83]. Every FIB entry is composed of a name prefix and a list of next hops. According to its forwarding strategy, routers can forward *Interest* packets to one or multiple hopes, hence enabling multi-path forwarding [24, 59, 60, 74].

2.2 Routing and Forwarding

NDN routers use application namespace instead of IP addresses to forward packets [54]. Routers update and announce their FIB entries using routing algorithms [41, 51, 89], or a self-learning mechanism [78]. Unlike IP routers, NDN routers use stateful forwarding, i.e., routers keep information about the received requests until they are satisfied or timed-out [101]. The forwarding strategy forwards *Interest* packets according to the FIB entries, local measurements or other per-namespace forwarding policies [77]. The forwarding strategy is also responsible for choosing the destination interfaces. NDN routers could also use multi-path forwarding to ensure priority, load-balancing and avoid failed links. Every *Interest* packet brings no more than one *Data* packet, and each response takes the reverse path of its corresponding request.

2.2.1 Interest packet forwarding: Requesting data. When a router receives an *Interest* packet, first it checks if the requested data can be satisfied from the local CS (i.e., it matches the received data name with the existing content names in the CS). If the requested data already exists in the local CS, the router sends back the *Data* packet to the source interface, i.e., from where the *Interest* packet came from. If the CS does not hold the requested *Data* packet, the router performs the following actions: first, it checks the PIT for any similar pending request, If an entry with a similar name already exists in the PIT, the router matches the source interface with all the interfaces associated with this pending entry. If the interface already exists, the incoming *Interest* packet is considered as a duplicate packet and will be discarded. Otherwise, the router adds the source interface to the list of interfaces associated with the pending *Interest* i.e., aggregates the incoming *Interest* packets requesting the same data. On the other hand, when no pending *Interest* with the same name exists in the PIT, the router creates a new entry for this *Interest* packet. Once the PIT lookup process finished, the router sends the *Interest* packet to its upstream neighbor(s) according to its forwarding strategy. NDN routers can also send a NACK packet to their

downstream nodes when the *Interest* packet cannot be satisfied, e.g., no matching entry in FIB, the upstream links are down [7, 90].

2.2.2 Data packet forwarding: Receiving data. When a consumer's request is satisfied from a data producer or in-network cache, a *Data* packet is sent back. When a router receives a *Data* packet, it checks if it was requested before by verifying the existence of a pending *Interest* with the name of the received data. If no match exists in the PIT, the router drops the received *Data*. Otherwise, and according to its caching strategy, the NDN router stores (or not) the received *Data* before sending it downstream to all the interfaces associated with the pending *Interest* in PIT. The aggregation of source interfaces gives NDN routers a built-in multicast mechanism.

2.3 NDN Security: Data-centric Security

The NDN design natively embeds security mechanisms and mandates the use of signatures in *Data* packets [71]. Producers need to digitally sign every *Data* produced. This will enable consumers to verify and validate the authenticity of the received *Data*. Also, it permits to network nodes, i.e., routers and repositories, to store *Data* [67]. Furthermore, it allows consumers to retrieve and accept *Data* packets regardless of their source [62]. The NDN architecture integrates some security components to ensure data security [113].

2.3.1 Packet signature. Every *Data* packet includes a signature field [105]. The signature binds the content of the packet to its name [52]. The Signature field contains two components: *SignatureInfo* and *SignatureValue*. The *SignatureInfo* component embeds the name of the producer's public key and the cryptographic algorithm used to sign the *Data*. The *SignatureValue* represents the bits of the generated signature. Althouth NDN does not mandate the use of signatures in *Interest* packets, however, in some scenarios where the authenticity of *Interest* packets is needed, signatures are required, e.g., a router sends a new route announcement, sending a command packet to an IoT device, etc. The *Interest* packet's signature field includes additional components compared to the *Data* packet's signature. It includes a *SignatureNonce*, a *SignatureTime* and/or a *SignatureSeqNum*. These components are used to add uniqueness to the signature.

2.3.2 Trust Schema and Access Control. Although that *Data* packet's signature allows the consumers to validate the received *Data* and proof its originator, it does not show whether the signer is authorized to produce the data or not [103]. Consumers need a mechanism that allows them to check if a producer has the right to produce *Data* under a given namespace, i.e. if a producer's key has the right to sign a *Data* packet under a given namespace [76]. Application-based trust policies are used to define which keys are authorized to sign which *Data*. Besides, applications can also implement access control policies to protect the content of a *Data* packet, through encryption, and permit only authorized nodes to decrypt it [50, 112].

2.3.3 Key pairs and Certificates. Every data producer need at least one cryptographic key pair. Producers use private keys to sign *Data* packets and consumers use public keys to verify them. Each public key is embedded in a digital certificate. The NDN certificate is a *Data* packet signed by an issuer [8]. It contains the producer's public key encoded in X509 format [111]. The name of an NDN certificate follows a specific naming convention: */SubjectName/KEY/KeyId/IssuerId/Version*, where *SubjectName* is the prefix to which the key is bound to, i.e., the namespace to which the key can operate under. Followed by the keyword *KEY* is the *KeyId*, which represents the value of the key. The value can be an 8-byte long random value, SHA-256 digest of the public key, a timestamp or a numerical identifier. After that comes the information about the issuer and the version of the certificate. Like any NDN *Data* packet, certificates include also a signature field, i.e., signature of the issuer.

3 AVAILABILITY ATTACKS IN NDN

Availability in NDN is susceptible to various types of attacks. In this section, we detail the availability attacks that target NDN routers and producers.

3.1 Availability Attacks Against Routers

Every NDN router component is a potential target for attackers. This subsection explains the availability attacks that routers can deal with.

3.1.1 Availability attacks against the Content Store.

(a) *Cache pollution attack*: To reduce time retrieval and network traffic, NDN nodes use the CS to cache the frequently demanded data. The cache pollution attack consists of altering the popularity of stored content. The attacker tries to evict popular content from caches by continuously requesting non-popular content. Figure 2(a) shows a scenario of cache pollution attack.

(b) *Cache poisoning attack*: Another CS-based attack is the cache poisoning attack. The goal of the attacker is to fill routers' caches with invalid *Data*. The attacker tries to inject *Data* packets with valid names but altered or malicious content. Unlike cache pollution, cache-poisoning attacks are slightly hard to perform as it implies the modification of the content. The packet's signature binds the name to the content, so any change on the content results in an invalid *Data* packet. However, the attacker can still spread malicious *Data* packets in the network, as routers tend to not verify *Data* authenticity at wire rate [39]. Attackers can perform cache poisoning attacks in two ways: (1) with malicious producers that send poisoned *Data*. In this scenario, the malicious producers could also cooperate with malicious consumers to flood the network with invalid *Data* [64]. (2) With malicious consumers. In this attacking scenario, the malicious nodes either perform a man-in-the-middle attack or control compromised routers (e.g., edge routers) as shown in figure.2(b).

3.1.2 Availability attacks against the FIB. The availability attack that targets router's FIB is the false route announcements. The purpose of this attack is to inject false routes into the network. The attacker needs to take control of the router to perform such an attack. The adversary, which controls an edge router, injects false paths to redirect legitimate requests to a malicious provider, as shown in Figure.2(c). In a different scenario, adversarial consumers can also send false routes in a mobile network, like Vanet, to perform other attacks like black holing or packet interception [46].

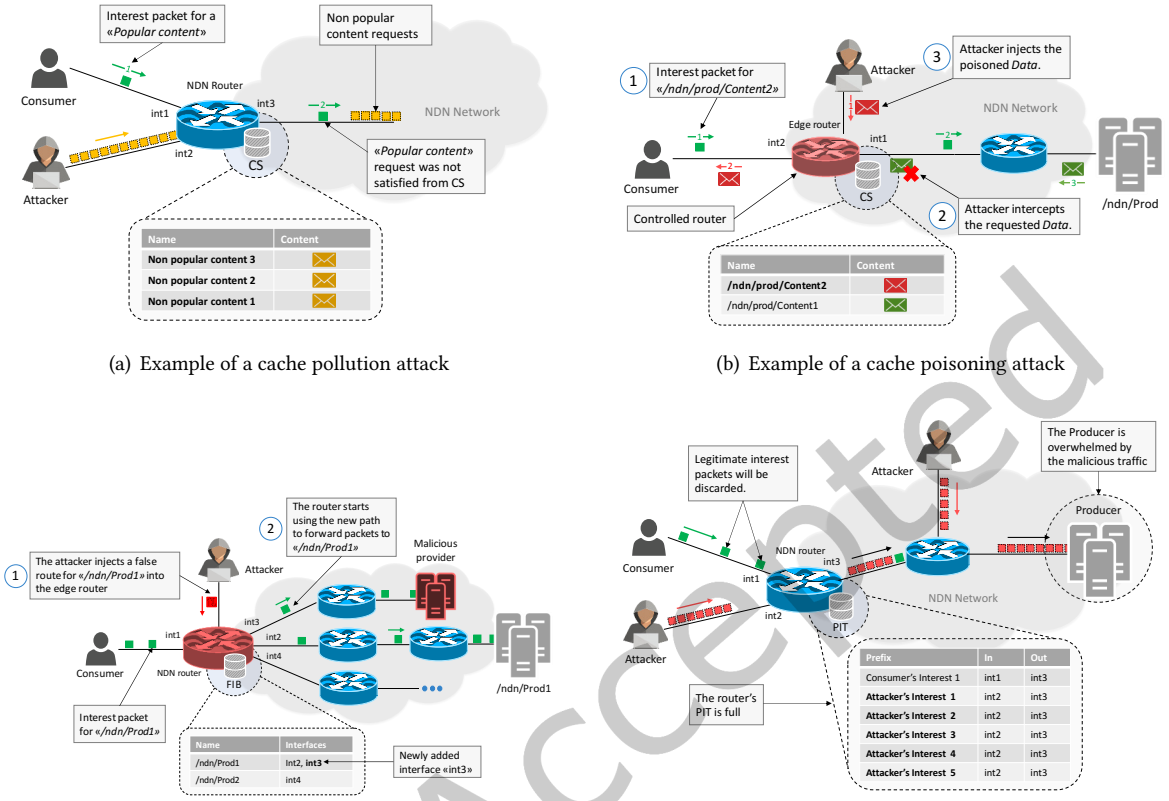
3.1.3 Availability attacks against the PIT. The main attack that targets PIT availability is the Interest Flooding Attack (IFA). An attacker target PIT's availability by sending to it a large number of *Interest* packets so it cannot accept legitimate requests. We discuss IFA in detail in Sec.4.

3.2 Availability attacks against Producers

Producer-based availability attacks are mainly associated with IFA. Attackers could target one or multiple producers by sending a big amount of *Interest* packets towards them. Malicious nodes can lunch attacks against producers using all types of *Interest* packets mentioned in 4.2. The damage that IFA can inflect on producers depends on the nature of *Interest* packets used during the attack. Depending on the severity of IFA, producers can suffer resources damages due memory and processing overhead.

4 INTEREST FLOODING ATTACK

The Interest Flooding Attack consists of flooding the network with a massive number of *Interest* packets to drown network routers and/or data producers, as shown in Figure 2(d). The main goal for attacking routers is to fill



(a) Example of a cache pollution attack

(b) Example of a cache poisoning attack

(c) Example of a false route announcement attack

(d) Example of an Interest Flooding Attack

Fig. 2. Availability Attacks in NDN

their PIT with unnecessary *Interest* packets so there will be no remaining entries for incoming legitimate *Interest* packets, which results in a denial of service. Additionally, data producers could also suffer a DoS from IFA. This includes memory exhaustion, service overload or any other hardware related overhead. attacks could be local, launched by one or a small group of nodes, or distributed, initiated by a large group of nodes often controlled by one master node.

4.1 Local and Distributed Interest Flooding Attack

IFA is considered local when one or a small group of locally connected attackers perpetuates them. Another form of IFA is distributed. A distributed IFA occurs when the victim suffers from massive traffic originating from a large number of distributed nodes. These groups of nodes, called botnets, are usually constituted of compromised systems managed by an attacker through one or multiple botnet controllers. Distributed IFA are difficult to stop and present a big threat as it floods the target with a large number of requests, resulting in a devastating attack. Botnet networks could implement hundreds of thousands to millions of nodes.

4.2 Taxonomy of IFA Requests

Attackers could perform IFA using two types of *Interest* packets. The first type of *Interest* packets carry valid data requests, and the second type carries invalid data requests.

4.2.1 Valid data requests. The first type of requests that attackers could use to attack a target with IFA is valid data requests. This type of data requests is further divided into two classes: requesting static data or dynamic data.

(1) *Requesting static data*: The first type of valid requests that attackers could use to perform IFA on a target is using *Interest* packets with valid names. Like any other valid *Interest* packet, these requests are satisfiable by data producers or network caches. The attack consists of sending an enormous number of *Interest* packets to the network to choke network routers and data producers. Static data are generally stored in intermediate caches or repositories, which reduces the traffic heading to producers, i.e., the request will be satisfied before reaching its producer. However, even that static data are likely to be found in network caches, the attack can still inflict serious harm to data producers especially in its debut, i.e., when data is not stored in caches. Besides, the malicious traffic induced by the attack can cause serious problems to the network especially in case of a large-scale attack.
(2) *Requesting dynamic data*: Dynamic data represent any content that producers generate on-demand. Unlike static content, dynamic data changes over time and needs to be produced when a request arrives. The optional *FreshnessPeriod* field in *Data* packets indicates for how long the content could be considered as fresh. Dynamic *Data* can range from the current record of a sensor in an engine to the actual stock price values. As their content are time-dependent, dynamic *Data* are usually not stored in caches. Attackers can take advantage of this aspect to lunch IFA against producers. In fact, dynamic data requests are normally not satisfied by network caches and are likely to be routed to their producers. This may cause high damage to producers as they use hardware resources to process and sign the content before sending it back.

4.2.2 Invalid data requests. An invalid data request corresponds to any *Interest* packet with an invalid content name, i.e., forged *Interest* packet. Attackers can lunch IFA using fake *Interest* packets. There are two different ways of forging an *Interest* packet's name: the first method is to choose a completely random name for the *Interest* packet. As their names are random, the *Interest* packets will not reach any producer and will affect only network routers. Nevertheless, the first type of forged *Interest* packet could heavily affect the network, especially the routers near the source of the attack. The second method of forging *Interest* names is to append a random series of characters to a real producer's prefix. This ensures that the forged *Interest* packet will reach the targeted producer. For example, if an adversary wants to target a producer *"/Prod"*, the *Interest* packets' names that the attacker may use are similar to *"/Prod/nonce"* where nonce is a random value. This type of forged *Interest* packet will affect producers and all the network routers traversed by the forged *Interest* packet. IFA with invalid data requests is more harmful to the network compared to an IFA with valid data requests. This is because routers will not aggregate the *Interest* packets and will stay in PIT until they time-out. NACK packets help reduce the number of pending invalid PIT entries. However, the attack can still do harm to the network. Additionally, attackers can use this mechanism to flood the network with NACK packets.

4.3 IFA Studies

IFA was first mentioned in [44]. The authors of this paper talked about how consumers could overwhelm the network by generating a huge number of *Interest* packets. Following that, several papers studied the IFA phenomenon and its impacts on the network. The authors of [39] discussed IFA and cache poisoning attacks before giving some tentative countermeasures. The authors of [29] conducted a study on the impact of IFA, through a series of simulation tests, on the throughput and the delay. The authors in [88] evaluated three PIT

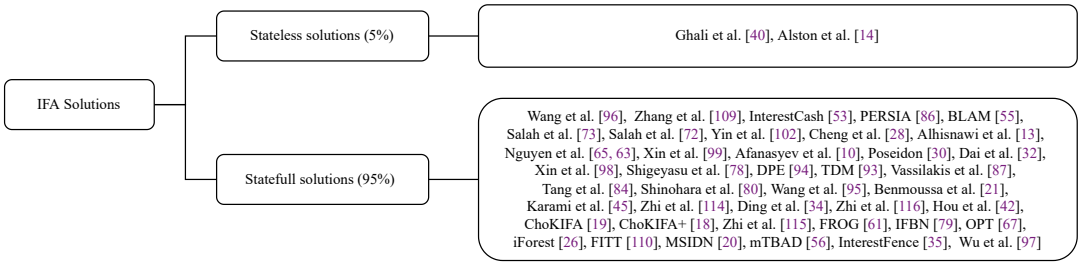


Fig. 3. Taxonomy of IFA solutions

architectures against IFA according to their memory consumption: the default PIT, a hashed names-based PIT, and a bloom filter-based PIT. Similarly, the authors of [82] presented a hash-based design to reduce memory occupancy. On the other hand, authors in [66] conducted a study on the collusive IFA, where attacker cooperate with a malicious provider to overwhelm the network. In [31], the authors discussed the benefits and issues of using network-based and application-based NACKs to help mitigate IFA. Similarly, the authors of [92] argued that NACK packets help routers from keeping pending *Interests* in PIT until they time out. Another similar study was presented in [91]. The authors also recommend the usage of NACKs to remove pending requests and hence mitigate attacks. In [81], the authors evaluated three collaborative solutions against two variants of IFA: The first uses different name prefixes. The second uses a mix of valid and invalid names. Similarly, in [12], the authors compared five techniques according to their satisfaction ratio. In another context, The authors of [49] conducted a comparison study between several machine learning algorithms to detect IFA.

5 RELATED WORKS ON INTEREST FLOODING ATTACKS

Many IFA solutions are present in the literature. In this section, we categorize the IFA related works into two main categories, as illustrated in Figure.3. The first category, named the stateless category mentions the solutions that took a stateless architecture approach to deal with IFA. The second is the stateful solutions category. This category is further classified into proactive and reactive solutions subcategories.

5.1 Stateless solutions

The solutions in this category chose a stateless approach to deal with IFA (i.e., do not store traffic-related information in PIT). The authors of [40] proposed a new architecture named *Stateless CCN*. The solution modifies the *Interest* and *Data* packets to incorporate a new component, called *Supporting Name*, which includes the consumer's prefix. Routers use the consumer's name to forward back the content packet. In this architecture, every consumer needs to be identified, which raises prefix announcements and management problems. Besides, this architecture introduces some TCP/IP based threats, like reflective and privacy attacks.

In a different approach, the solution proposed in [14] uses cryptographic tokens to route packets. Each receiving router updates these token, called route tokens, with additional information to forward the *Data* backward. Symmetric keys are used to ensure the integrity and confidentiality of route tokens. Although this solution prevents memory consumption due to PIT overhead, the proposed technique may consume a lot of processing resources, especially with large traffic. Attackers can use it as a tool to inflict high damage to routers, and the damage gets even higher as it gets closer to the core network.

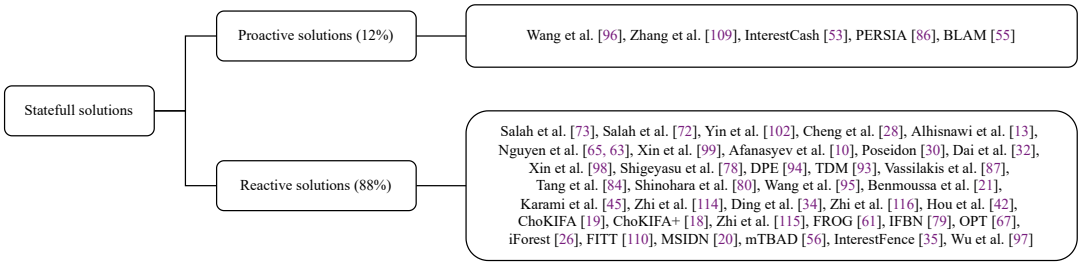


Fig. 4. Taxonomy of stateful solutions

5.2 Stateful solutions

The last main category of our classification groups the stateful solutions authored in the literature. The stateful solutions category is divided into: proactive and reactive solutions. Figure 4 illustrate the different stateful IFA solutions.

5.2.1 Proactive solutions. A small group of proactive countermeasure solutions are present in the literature. The authors of [96] introduced micro-payments into the network to regulate IFA. Authority nodes in the network act as banks to conduct any virtual money-related actions. To request a *Data* packet, consumers need to add a prepayment to the *Interest* packet, which includes a PIT delay and content delivery fees. When the request reaches a destination, the node checks its content delivery fee, deducts the necessary amount, and sends the content downstream. Every traversed router takes an amount from the content delivery fee. The consumer receives the requested *Data* packet only if the fees are sufficient to cross all the nodes. The micro-payment system limits legitimate traffic. Also, legitimate consumers can be penalized in scenarios like unsatisfied requests or unavailable producers. Additionally, deploying and maintaining such a system is extremely hard in large networks, given that the system relies on central nodes to act as banks. To overcome this problem, the authors in [109] proposed a payment solution that relies on auto-regressive integrated (ARI) and the hidden Markov models (HMM). When a router receives an *Interest* packet, the price that the router charges to forward the incoming *Interest* depends on the number of pending requests associated with this interface. The router uses the satisfaction ratio as a parameter for the Hidden Markov Model to predict the consumer's state, legal or bad. Besides the fact that legitimate traffic is penalized by the charge/reward mechanism, relying only on the satisfaction ratio to predict the consumer's state may lead to false detection.

Another proactive solution was presented in [53]. It employs a proof-of-work mechanism to countermeasure the signing overhead resulting from IFA requesting dynamic content. When a consumer expresses the need for specific data, he needs to answer a puzzle sent by the producer and re-sends the *Interest* packet with the computed value (result). If it is correct, the producer sends back the *Data*. To reduce the communication delay resulting from retrieving/answering puzzles, the solution in [86] relies on tokens that consumers get when they solve computational puzzles. When a consumer sends an *Interest* packet, the edge router verifies if the token exists in its table. If it does, it forwards the *Interest* and updates the count number of this token. Furthermore, core routers monitor the loss rate of each interface. When it reaches a threshold, the router suspects malicious traffic and starts using a bloom filter, instead of PIT, to store incoming *Interest* from this interface. Similarly, the authors of [55] proposed BLAM, a lightweight bloom filter-based mitigation technique for IoT devices. In this solution, each IoT node uses a bloom filter array during the forwarding process. When an *Interest* arrives, the node matches its name with the bloom filter array. If an entry exists, the node continues the forwarding process. Otherwise, the node considers the received *Interest* as malicious and discards it. The proposed solution cannot be

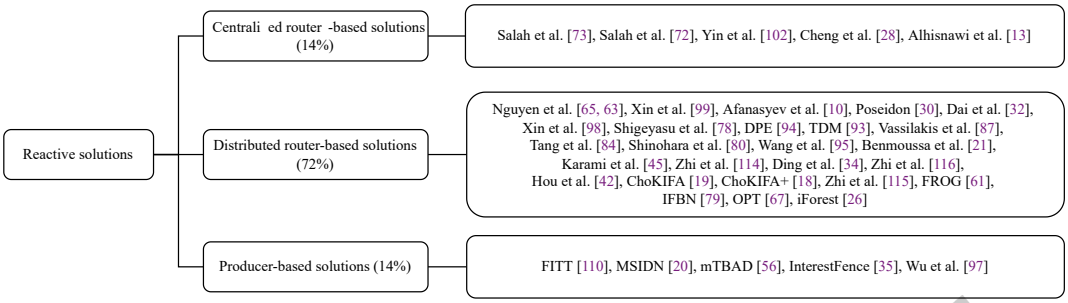


Fig. 5. Taxonomy of reactive solutions

deployed on large networks as it requires knowing all the *Data* packets that producers offer. Similar to the last three techniques, this solution does not prevent IFA. Attackers can still flood the network with *Interest* packets.

5.2.2 Reactive solutions. Several IFA reactive solutions have been proposed. Reactive solutions are classified into router-based and producer-based solutions. Router-based solutions are further grouped into centralized and distributed solutions as showed in Fig 5.

Centralized router-based solutions Several centralized router-based IFA mitigation solutions exist in the literature. The solution presented in [73] relies on a Domain Controller (DC) and a group of selected routers named Monitoring Routers (MR) to detect IFA. The MRs calculate the PIT usage and the expired *Interests* associated with each interface. When these metrics exceed a certain threshold, the MRs send the name prefixes and their respective expiration rates to the DC. The DC will then decide on the infected name prefixes and inform the MRs. However, the metrics used can lead to false detection in the case of high legitimate traffic or unavailable producers. In addition, the legitimate requests going to infected prefixes can be affected. The authors proposed an enhanced version in [72] capable of detecting collusive IFA. Compared to the previous solution, this mechanism relies only on PIT usage, which leads to false-positive situations affecting legitimate requests.

Unlike the previous two, the solution in [102] relies on every router to collects and sends to the domain controller the number of *Interest*, *Data*, and NACK packets. The controller calculates the satisfaction ratio associated with each router. If the metrics exceed their respective thresholds, the controller considers that this router is under attack and sends back the malicious prefixes. However this solution relies on the controller to calculate the metrics, which makes it a SPOF. Also, the traffic that routers send may overwhelm the network. To overcome this problem, the solution proposed in [28] relies only on edge routers. when the rate and the number of timed-out *Interests* of a router's interface exceed their respective thresholds, it informs the controller. The latter inspects the received traffic information and checks whether the links will reach their capacity. If one or more links are likely to reach their capacity, the controller determines that an IFA is happening and informs the routers on the interfaces to block. The proposed solution may consider high traffic rates as malicious.

The solution presented in [13] has the particularity to rely not only on a domain controller but also a content provider. When the content provider router (CPR) receives an *Interest*, it verifies the name of the *Interest* against the FIB and the Quotient-based Cuckoo filter (QCF), which represents an updated list of fake *Interest* names. If the name exists in the QCF and no entry is found in FIB, it considers this *Interest* as malicious and sends its name within a warning message to the controller. The controller then updates the QCF before informing the edge routers. When an edge router receives the warning message, it restricts the originating interface. However, the QCF table could be used by attackers. Also, the content provider router needs to process all *Interests* of the

AS, which is resource-consuming.

Distributed router-based solutions The majority of IFA solutions that exists in the literature are distributed router-based solutions. In [65], the authors presented a solution that uses hypothesis testing theory to detect IFA. Routers calculate a packet-loss rate associated with each interface to model the legitimate traffic. Then this model is used to detect IFA. Regardless of being a detection-only mechanism, the proposed solution relies only on the number of *Interest* and *Data* packets statistics to identify IFA, which may give false-positive results. Routers can alleviate this problem if they cooperate to detect IFA. The authors also employed this technique to detect collusive IFA in [63]. Similarly, the proposed solution in [99] analyses the traffic using a discrete wavelet transform in an uncooperative way to detect collusive IFA. The detection process can be resource-consuming in some scenarios. These two solution do not act to mitigate attacks.

The authors of [10] proposed a solution based on the token bucket. It consists of distributing the tokens according to interfaces' satisfaction ratio. When a router calculates the limit values of an interface, it announces this limit to the neighboring routers connected to it. The receiving routers will then adapt their sending rates according to the received value. In addition to the satisfaction ratio, the solutions proposed in [30], use the PIT usage of each interface. The router suspects malicious traffic when these two metrics exceed their thresholds. As a mitigation action, the router limits the incoming traffic and penalizes the interface with reduced thresholds. Then, it sends an alert message to the router connected to this interface to inform it about the reduced rate. Similar to the previous solution, the rate limit employed can impact legitimate consumers. Similarly, routers in [32] monitor the size of their PIT. When the size exceeds a certain threshold, the router generates a spoofed *Data* packet and sends it back to the originator of the spoofed *Interest.* Then, the edge router limits the rate of the source interface. Another pushback-based mechanism was proposed in [98]. In this solution, routers monitor the name of incoming *Interests* and compute the interface's cumulative entropy. When This latter exceeds an upper bound, the router confirms that an attack is undergoing. Afterward, the router sends a spoofed *Data* to the originators of the malicious prefix. When an edge router receives the spoofed *Data*, it applies rate limiting on source interfaces. Besides the resource exhaustion of the detection process, legitimate interfaces can be penalized by rate-limiting in some scenarios. Another drawback is that attackers can use the spoofed *Data* as a tool to drown the network.

The authors of [94] proposed a solution that decouples the malicious traffic from legitimate traffic to prevent PIT exhaustion. Routers monitor the number of timed-out *Interest* packets under each prefix. When the router considers the prefix under attack, it stores this prefix in a list called "m-list". The router forwards the *Interest* packet without storing it in PIT if the prefix exists in the m-list. This solution limits the effects of an IFA but does not stop it. Also, it affects legitimate requests. Furthermore, attackers can target the router's resources by forging names to fill the m-list, which makes the solution inefficient. Similarly, the mechanism presented in [93] monitors the timed-out *Interest* packet to detect IFA. It expands the FIB to include four additional fields. When an *Interest* times out before a *Data* returns, the router changes the mode value of this prefix to malicious and increments a related counter Ci in FIB. If the Ci value goes above a predefined threshold, the router applies rate limiting on this prefix. The rate-limiting used in this solution is associated with FIB entries, which affects all the traffic of this entry. The solution in [87] also relies on the number of expired PIT entries. Edge routers classify each interface, according this metric, into three categories: normal, suspicious, or malicious. The edge router can reduce or block the traffic depending on the interface's behavior. It also notifies other routers if it blocks a consumer. However, identifying NDN consumers is not an easy task, which makes this solution hard to apply.

The authors in [84] presented a two-phase detection mechanism. A router calculates the satisfaction ratio of each interface. If it exceeds its threshold, the router counts the number of expired *Interests* under each prefix. If the number exceeds its threshold, the router considers the prefix as malicious and starts blocking *Interests* going to this prefix. Similarly, in [80], routers calculate a reputation value for each interface depending on its

satisfaction ratio. If an interface has a low reputation value, the router checks if the number of PIT entries is high. In this case, it discards incoming *Interests*. These two solutions do not prevent attackers. Additionally, they may lead to blocking legitimate *Interest* as they rely only on the satisfaction ratio.

The solution presented in [95] relies on fuzzy logic to detect IFA. Every router on the network monitors the PIT occupancy and PIT expiration rates. The router takes these values as entries for the detection algorithm. If an IFA is detected, an alert message containing the malicious prefix is sent to downstream routers. When an edge router receives the pushback message, it limits the traffic going to the malicious prefix. The detection mechanism is resource-consuming. Besides, namespace-based rate-limiting can also affect legitimate requests and may lead to false-positive detection in some scenarios. To avoid false detection, the work presented in [21] takes network congestion as a parameter when detecting IFA. Edge routers classify a consumer as suspicious when its incoming rate and satisfaction ratio are above and below their respective thresholds. After that, the router checks the network congestion to avoid false detection. It compares the number of timed-out *Interests* with NACK packets. If the network is not congested, the router classifies the consumer as malicious and blocks it. However, the congestion detection process employed is not cooperative, which may give false results in some scenarios.

Unlike the previous solutions, the authors in [45] proposed an AI-based mechanism to mitigate IFA and cache poisoning using Radial Basis Function (RBF) neural networks. It uses a set of statistics as inputs, such as the number of satisfied and timed-out *Interests*. When the detector module signals malicious traffic, the router sends an alert message to source interfaces. The alert message contains the new reduced rate, the generation timestamp, and the reduction period. However, the rate-limiting can affect legitimate consumers. Another machine learning-based detection mechanism was proposed in [114]. In this solution, a router constantly collects the entropy of *Interest* names, the satisfaction ratio, and the PIT usage of interfaces before using them as entries for the support vector machine (SVM) classifier. If the detector classifies it as an anomaly, the router declares that an IFA is happening. Following that, the router extracts the malicious prefixes using the Jensen-Shannon divergence. Then, it informs downstream routers about the malicious prefixes. This solution affects also the legitimate *Interest* packets going to blocked prefixes. Similar to the previous solution, the detection process may consume a lot of resources.

The authors of [34] presented a solution based on the vector space model and Markov chains. When the PIT size reaches an alarming level, the router determines a state for each received *Interest* (normal, unknown, and risk). Upon receiving an *Interest*, a router first checks its state. If the previous (i.e., received from the upstream router) and the actual (i.e., calculated by the router) are both equal to risk, the router discards the *Interest*. To prevent legitimate requests from being discarded, the proposed solution stores a copy of unsatisfied *Interests* in a specific cache, so if an *Interest* with the same name arrives, the router considers it as legitimate. However, attackers can use this propriety to target routers. Additionally, the process of state checking consumes router's resources.

A statistical solution, that uses Gini impurity, was proposed in [116]. A router monitors the name of received *Interests* and calculates their probability. Then, it uses Gini impurity to measure the disparity of the set and compares it with a threshold to determine if an IFA is happening. After that, the router uses Gini impurity to compare the actual set with a previously recorded set to detect the malicious prefix. The proposed solution may lead to false-positive detection. Similarly, An entropy-based solution was proposed in [42]. Based on the fact that when IFA happens, the occurrence of forged names increases, and the Thiel entropy decreases. Every router records the names of incoming *Interest* packets and calculates their entropy to decide whether IFA is happening. The proposed solution can mistakenly consider legitimate traffic as malicious as it relies only on the statistical distribution of *Interest* names. Furthermore, storing *Interest* names may consume a lot of resources, especially in the case of a massive attack.

The authors of [19] presented a solution that relies on Active Queue Management (AQM). For each received *Interest*, the router randomly picks a pending *Interest* and matches its name with the received *Interest*. Then, it compares their source interfaces and checks the satisfaction ratio of the source interface. If it is under a threshold, the router drops both *Interests*. Otherwise, it drops the received *Interest* with a probability. However, attackers can timely orchestrate attacks to fill the target's PIT so the router will start dropping incoming *Interests*. The authors modified this mechanism to include blocking [18]. When an edge router detect that the PIT occupancy and the satisfaction ratio of an interface are above thresholds, it blocks it. However, legitimate traffic is still penalized by the mechanism and it increases as it gets close to the core network. Similarly, the authors of [115] use a reputation-based early detection mechanism to counter collusive IFA. When an interface's PIT usage is between the minimum and the maximum thresholds, the router drops incoming *Interests* with probability. The router defines the minimum and maximum thresholds according to the satisfaction ratio and the average RTT of an interface. The metrics used may lead to dropping a large portion of legitimate *Interest*, especially in core routers. Similar to the previous two, this solution is a traffic control mechanism, which does not stop attackers. Another solution that aims to counter collusive IFA was proposed in [61]. It relies on the fact that malicious traffic is satisfied only by producers, which makes the mean hop count of each received *Data* high and its variance low. Hence, edge routers store the hop count of each *Data* and calculate the mean and variance of the recorded set. Following these values, the router affirms if a consumer is malicious or not. This solution may lead to mistakenly blocking legitimate users in scenarios where the legitimate requests need to be satisfied by the producer. Being a non-cooperative solution, the proposed solution is unable to detect attacks against routers. Another mechanism against collusive IFA was presented in [79]. Each router monitors its PIT usage. If the rate reaches a threshold, the router examines the number of times the *Data* was satisfied from the router's cache. If it equals zero, the router recognizes an ongoing attack. However, attackers can avoid detection since this solution relies on the fact that attackers send *Interests* with random names. Besides being a detection-only mechanism, it can also lead to false-positive detection in some scenarios (e.g., live streaming).

In [68], the authors presented a mechanism that uses a new structure called Operation Trace Table. The router uses it to record the traffic statistics associated with each interface. When the unsatisfaction ratio of an interface is above the average satisfaction ratio, the router reduces the interface's share and distributes it equally among all interfaces. If the router receives an *Interest* from an interface that reached its allowed share, it forwards it to another outgoing interface to discover unknown paths. The proposed solution does not mitigate attackers. In addition, sending *Interests* to other routes may overwhelm upstream links and routers.

Each router in [26] randomly selects some prefixes from a recorded list containing calculated metrics for each prefix. Then, it constructs a group of search binary trees from the minimum and maximum values of the selected prefixes. After that, it calculates the average traverse path of chosen prefixes and uses it to calculate an abnormal score for each prefix. Finally, it notifies downstream routers with the malicious prefixes. The namespace-based rate-limiting penalizes legitimate *Interest*. Besides, the detection process can consume a lot of resources, especially for core routers. Additionally, this technique can take time to detect attacks.

Producer-based solutions Some producer-based reactive solutions were proposed in the literature. The solution proposed in [110] relies on producers' messages to initiate IFA mitigation. When a producer needs to release its resource, it sends a specific NACK packet to the gateway router. The generated NACK carries essentially the nature of the traffic, fake or valid, and all the fake names received under the affected prefix. When a router receives the NACK, it notifies neighboring routers of the fake names. If an edge router receives a NACK with traffic equal to fake, it blocks the source interfaces for some time. The fake names list used can burden routers in case of massive distributed attacks. Additionally, the proposed technique relies on producers' feedback to initiate the mitigation process, which leaves routers vulnerable to attacks. To overcome this limitation, The authors of [20] proposed a solution that relies on both producers and routers to mitigate IFA. Besides detecting

conventional distributed IFA, it also detects distributed IFA where attackers adopt regular sending rates. When a producer is overwhelmed with requests, it sends a signed *Interest* asking routers to limit forwarding requests to the affected prefix. When received, the router stops forwarding *Interests* before informing neighboring routers. An edge router classifies a consumer as harmful when it keeps sending requests to the infected prefix and has a low satisfaction ratio along with a high number of timed-out *Interests*. Although the proposed mechanism can mitigate different types of distributed IFA without affecting legitimate traffic, it still may unintentionally block legitimate consumers.

A new version of [94] was proposed in [56]. In this solution, a new producer-based monitor mechanism was introduced to reduce the detection delay. When a producer receives a malicious *Interest*, it responds with a NACK to inform downstream routers. When a router receives the NACK, it checks whether a similar entry exists or not in the m-list. Although it reduces detection delays, this solution still does not stop attackers. Additionally, producers may consider some legitimate requests malicious. Similarly, each router in [35] maintains an m-list where it stores the affected prefixes received from producers. When the router receives an *Interest*, it checks its existence in the m-list. If it matches the information received from the producer, the router forwards the *Interest*. Otherwise, it drops it. Routers need to process every received *Interest*, which may consume resources due to hash verification. Also, the m-list can occupy a lot of memory in case of a massive attack with a large number of prefixes. Additionally, routers need to process application-based NACK packets.

Unlike the previous solutions, the mechanism presented in [97] mitigates collusive IFA. Routers monitor the interfaces' throughput and PIT usage during fixed time windows to detect malicious traffic. When these metrics go above their thresholds, the router starts to delete the oldest PIT entries. However, this solution may delete legitimate requests.

6 CDA WORKFLOW: COLLECT-DETECT-ACT

In this section, we provide an in-depth analysis of each authored solution. Our comparison depends on the information solutions collect, the metrics and parameters they use during detection, and the actions they take to counter IFA. To this end, we present CDA, which is a workflow that every solution follows. The CDA workflow consists of three activities *Collect*, *Detect* and *Act*.

6.1 Collect

The first activity of the CDA workflow is *Collect*. During this activity, solutions collect a set of traffic-related information to detect IFA. The nature of this information varies from one solution to another. Some solutions rely only on router-based information [10], [73], [21], while others rely on producer-based information [110],[56]. On the other hand, to detect IFA, some solutions rely on both router-based and producer-based information like [20].

6.1.1 Collection parameters. Besides the source of collected traffic, router-based or/and producer-based solutions use a variety of parameters during the *collect* phase. We conducted a quantitative and qualitative comparison based on the parameters used by existing solutions to collect traffic information. The first column of Table 2 specifies proposed solutions. The next two columns indicate if a solution collects router or/and producer information during this phase. The fourth column shows whether a solution stores or not the collected traffic information. The fifth column provides the routers categories used by a solution. Routers are classified depending on their role in the system. The default value equals one. The following two columns specify whether a solution modifies the PIT and FIB tables. The "New structure" field indicates if a solution introduces a new data structure. The last column specifies the solutions that rely on a central node during the Collect phase.

6.1.2 Key observations of our comparison. The fundamental observations of our comparative study on the information that solutions collect to detect attacks are as follows: The first thing that can be noted from Table 2

Table 2. Collection parameters used by existing solutions

Ref	Router statistics	Producer statistics	Statistics storage	Router classes	Modified PIT	Modified FIB	New structure	Central node
[10]	Yes	No	No	01	Flags: Inqueue, Fwd	No	No	No
[30]	Yes	No	No	01	No	No	No	No
[32]	Yes	No	No	01	No	No	No	No
[94]	Yes	No	Yes	01	No	No	Affected prefixes list	No
[56]	Yes	Yes	Yes	01	No	No	Affected prefixes list	No
[95]	Yes	No	Yes	01	No	No	No	No
[93]	Yes	No	Yes	01	No	Yes	No	No
[45]	Yes	No	No	01	No	No	No	No
[114]	Yes	No	Yes	01	No	No	No	No
[87]	Yes	No	No	01	No	No	No	No
[73], [72]	Yes	No	Yes	• Monitoring • Normal	No	No	No	Yes
[102]	Yes	No	Yes	01	No	No	ADT and AIT	Yes
[65]	Yes	No	Yes	01	No	No	No	No
[99]	Yes	No	No	01	No	No	No	No
[79]	Yes	No	Yes	01	No	No	No	No
[34]	Yes	No	Yes	01	No	No	Unsatisfied *Interests* cache	No
[98]	Yes	No	Yes	01	No	No	No	No
[42]	Yes	No	Yes	01	No	No	No	No
[116]	Yes	No	Yes	01	No	No	No	No
[61]	Yes	No	Yes	01	No	No	No	No
[19], [18]	Yes	No	No	01	No	No	No	No
[115]	Yes	No	No	01	No	No	No	No
[110]	Yes	Yes	Yes	01	No	No	No	No
[68]	Yes	No	Yes	01	No	No	Operation Trace Table	No
[28]	Yes	No	Yes	• Monitoring • Normal	No	No	No	Yes
[26]	Yes	No	Yes	01	No	No	No	No
[21]	Yes	No	No	01	No	No	No	No
[35]	No	Yes	Yes	01	No	No	Affected prefixes list	No
[84]	Yes	No	No	01	No	No	No	No
[80]	Yes	No	No	01	No	No	No	No
[20]	Yes	Yes	No	01	No	No	No	No
[13]	Yes	No	Yes	01	No	No	Quotient based Cuckoo filter	Yes
[97]	Yes	No	No	01	No	No	No	No
[86]	Yes	No	No	01	No	No	Bloom filter	No
[96]	No	No	No	01	No	No	No	No
[109]	Yes	No	No	01	No	No	No	No
[55]	No	No	No	01	No	No	Bloom filter array	No
Distribution	87%	10%	51%	NA	2%	2%	23%	10%

is that the majority of authored solutions rely on router information to detect IFA. Routers are the heart of a network. Each connection session passes through a router. That explains the reason of the deployment of the solutions on the routers. It permits gathering a maximum of traffic information, which helps detect ongoing attacks. Routers are favored over producers when deploying solutions, as shown in this study. Producers do not

have a global view of a network. They are limited to the communication details that they receive. Solutions that rely only on producers' information are not efficient to detect attacks against routers. That explains why only one solution chose to counter IFA using session information that producers collect. As NDN uses application-layer names at the network level, routers and producers can cooperate to share information. Some solutions took the hybrid approach. These solutions rely on both routers and producers to gather information. It is the case of three authored solutions.Another key observation regards information storage. Our comparison shows that half of the solutions record connection statistics. Stored information help in detecting attacks with an additional precision level as it implicates the use of more data. It also permits detecting behavioral changes in some cases. However, this process implies using additional storage capacity. In addition, it slows routers at high network peaks, especially those close to the core network. That explains why researchers are divided on the use of information storage. Our comparative study found that only two solutions adopted personalized PIT and FIB tables. Personalized tables occupy more space in memory as they store additional information, and the memory space they occupy grows considerably for core routers. Furthermore, connection processing times increase as routers perform more checks before forwarding packets, which leads to higher network delays. That explains why solutions do not use personalized tables. Nevertheless, solutions may employ a completely new structure, as shown in this comparative study. We notice that almost a quarter of solutions uses a new data structure. They are usually associated with PIT. Using a new data structure comes with a trade-off. It supplies additional information for routers but at the same time occupies more space and introduces processing and computational overhead. Finally, our study shows that some solutions rely on a central node to collect traffic information. They represent 10% of authored solutions. Router classes are mainly associated with centralized solutions. They describe the role that routers occupy in a system. We will discuss centralized solutions further in the next subsection.

6.1.3 Collection metrics. Routers and producers collect a variety of traffic-related statistics. Table 3 summarizes the metrics that existing solutions gather during the Collect phase. The first column of the table specifies proposals. The second column represents the rate of incoming *Interest* packets of each interface. When the sending rate of an interface reaches a threshold, the router suspects malicious behavior. Solutions usually couple the traffic rate with other metrics. The third column corresponds to the ratio of received *Data* packets to the total number of requested *Interest* packets in a given period. Routers/producers calculate the satisfaction ratio of each interface separately. The next column denotes the occupancy ratio of each interface in PIT, i.e., the number of pending *Interest* packets generated by an interface to the total number of PIT entries. The following four columns of the table show the nature of packets being counted during this phase. Solutions may collect the number of received *Interest* and *Data* packets, the number of timed-out pending *Interest* packets, and the number of NACK packets. The ninth column shows whether a solution stores the name of received *Interest* packets. The last column of the table shows whether the solution collects producer-based metrics, including but is not limited to processing overhead, memory consumption, and service overload.

6.1.4 Key observations of our comparison. The key observations of the performed comparative study on metrics used by solutions are as follows: First, we notice that the most used detection metric is the number of timed-out *Interest* packets. It was used by 35% of existing solutions. Many solutions consider timed-out pending PIT entries as an alarm for an ongoing attack. Attackers usually use non-valid requests to overwhelm networks, and malicious *Interests* stay in PIT until they time out. That is why solutions usually consider the high number of timed-out *Interests* as a potential attack. However, this metric may lead to false positives, e.g., non-available producers. Another observation from Table 3 states that the second most frequently used metric is PIT occupancy. The goal of adversary nodes when they perform IFA is to fill the target's PIT with invalid requests so it cannot accept legitimate *Interests*. That justifies the use of PIT occupancy as a metric to detect attacks. However, this metric is not reliable in every scenario, as we will see in the next section. Our comparative study found that the satisfaction ratio is the third most used metric with 30% of solutions. Attackers usually perform IFA with invalid requests.

Table 3. Collected metrics by existing solutions

	Router-based metrics								Producer-based metrics
Ref	**Satisfaction ratio**	**PIT usage**	**Traffic rate**	**Timed-out *Interests***	**Number of *Data* packets**	**Number of *Interests***	**Number of NACK**	***Interest* prefixes**	
[10]	✓								
[30]		✓		✓					
[32]		✓							
[94]				✓					
[56]									Malicious *Interests*
[95]		✓		✓					
[93]				✓					
[45]	✓		✓	✓	✓	✓			
[114]	✓	✓						✓	
[87]				✓					
[73]		✓		✓					
[72]		✓							
[102]	✓						✓		
[65]					✓	✓			
[99]						✓			
[79]		✓	✓						
[34]		✓		✓		✓			
[98]								✓	
[42]								✓	
[116]								✓	
[61]									
[19],[18]	✓	✓							
[115]	✓								
[110]									Resource exhaustion
[68]	Unsatisfaction ratio								
[28]			✓	✓					
[26]		✓		✓	✓	✓			
[21]	✓		✓	✓			✓		
[35]									Number of forged *Interests*
[84]	✓			✓					
[80]	✓	✓							
[20]	✓			✓					Resource exhaustion
[13]		✓		✓					
[97]		✓	✓						
[86]			✓						
[109]	✓								
[55]									
Distribution	30%	33%	15%	35%	7%	12%	5%	10%	NA

That is why solutions consider low satisfaction-ratio levels as a sign of ongoing attacks. We also notice that solutions adopt either the satisfaction ratio or the numbers of timed-out *Interest*, as they both rely on the fact that the adversary floods the network with invalid requests. However, coupling these two metrics can help reduce false-positive detections that result from the use of the number of timed-out *Interest* packets. The next most used metric is the traffic rate. It was used by 15% of authored solutions. Adversary nodes send malicious requests at a high pace to overwhelm a network. That is why IFA is usually associated with high traffic rates. We also found that solutions always associate the traffic rate with another metric. Our comparison shows that some solutions count the number of received packets. This includes the number of *Interest* packets, *Data* packets, and NACK packets. For instance, we notice that the solutions, which use the number of *Interest* packets, usually compare it

with timed-out *Interests*. We also remark that the solutions which collect the number of *Data* packets always compare it with the number of *Interests*. On the other hand, only two solutions keep statistics of NACK packets. One used it to detect network congestion and the other to exchange information. The last router-based metric of our comparative study regards *Interest* prefixes. This metric was essentially used by statistical-based solutions. They use this metric to calculate an entropy to detect prefixes under attack. Finally, our last observation regards the number of metrics used by each solution. We found that 39% of authored solutions rely only on one metric, and 34% of solutions use two metrics to detect IFA. It is considered insufficient to detect IFA in different situations. We will discuss this point further in the following sections.

6.2 Detect

The second activity of the CDA workflow is *"Detect"*, in which solutions process the collected information to detect anomalies and ongoing attacks. There are two main IFA detection architecture designs, centralized, like [72] and [28], and distributed, like [30] and [32]. Before we detail the detection parameters that solutions may adopt during the *Detect* phase, we first discuss detection architecture designs. There are two architecture designs: centralized and distributed.

6.2.1 Centralized detection design. The centralized detection architecture relies on a central node to detect an IFA. The central node periodically receives traffic-related information from routers to analyze and decide whether an IFA is happening or not. The centralized detection is further categorized into cluster-based detection, selective nodes detection and global detection.

(a) *Cluster-based*: The first centralized detection topology is cluster-based centralized detection. The central node groups the network routers into different clusters. The cluster will then collect its traffic information before sending it to the central node. After that, the central node analyzes the received information to detect attacks. When an IFA is detected, it sends back to the clusters the actions to take in order to mitigate the attack. None of the existing solution adopted a cluster-based detection approach.
(b) *Selective nodes*: The second centralized-based detection topology is using selective nodes. The central node selects a group of routers from the network. The selection criteria are essential to ensure a global view of the network. That is why the central node needs to select the most relevant routers to cover the maximum of the network's traffic. The selected routers are responsible for sending the information that the central node needs in order to evaluate whether the traffic is malicious or not. After that, the central node sends back to the selected routers the proper actions to take. Solutions [73],[72], and [28] adopted the selective centralized detection mode.
(c) *Global*: The third and last centralized-based detection topology is the global topology. Every router in the network is part of the system. That is, every network router collects and sends the information of the traffic passing through it to the central node. Similar to other centralized detection topologies, the central node analyzes the information that it receives from the network routers before deciding the proper actions to take. The solution presented in [102] chose the global centralized approach as its detection mode.

6.2.2 Distributed detection design.

(a) *Autonomous detection*: The first distributed detection approach is the autonomous detection. It means that network routers and data producers collect and detect IFA locally. In this approach, the network nodes do not share traffic or attack information with each other. The majority of existing works adopted the autonomous detection approach like [21],[19], and [61].
(b) *Cooperative detection*: The second distributed detection approach that the network nodes can adopt is the cooperative detection. Network nodes share traffic-related information and take cooperative actions to mitigate IFA. The cooperative detection is further classified into partial cooperation and full cooperation.

(1) *Partial cooperation*: In a partial cooperation design, the network nodes collaborate only on the mitigation action, like [10], [114] and [20]. Each node analyzes its local traffic to detect an IFA and then takes the proper action to mitigate it. After that, the node informs its neighbors about the action and cooperates to block the attack and stop its growth.
(2) *Full cooperation*: In a full cooperation distributed detection, the nodes fully cooperate to detect and mitigate IFA, i.e., the nodes share information and jointly decide on the actions to take. Solutions presented in [56] and [96] adopted the full cooperation detection mode.

6.2.3 Detection parameters. Proposed solutions employ multiple parameters to detect attacks. Table 4 details the parameters used by each mechanism. The first column of the table specifies proposals. The second column defines the entities that are responsible for detecting attacks. The following two columns indicate the architecture approach that solutions use to detect IFA, centralized or distributed. Centralized detection may employ the entire routers or just a selection of them. In distributed detection, routers work solemnly or cooperatively to detect attacks. The fifth column shows whether a solution employs AI to detect attacks. The sixth column specifies if a solution considered network congestion when detecting IFA. The "Check interval" field represents the time that a solution waits for before it performs another detection check. The eight parameters determines the nature of IFA that a solution detect. The following two fields indicate exchanged packets during the Detect phase. The first parameter specifies if a solution introduces a new packet. The second parameter shows whether a solution modifies a packet (essentially an *Interest* packet). "The number of thresholds" field represents the set of thresholds that a solution uses to detect attacks. The final parameter shows the categories that a solution uses to classify consumers.

6.2.4 Key observations of our comparison. The key observations of our comparative study on the parameters that solution use during the detection phase are as follows: First, we found that 89% of solutions use routers as the detection actor. It includes both router-based and hybrid solutions. This number was predictable, as we previously noticed, in Table 2, that the majority of solutions rely on routers' information. We also observed that all centralized solutions rely on the central node to decide whether IFA exists or not. Second, we notice that 94% of authored solutions adopted the distributed detection design, and only 10% chose the centralized architecture. Centralized solutions require additional deployment phases. It implies the exchange of control information with system nodes. Additionally, deployment complexity grows, as the network gets bigger. Furthermore, the traffic information that system nodes send to the central node may burden the network, especially in large deployments. That explains why a small portion of existing countermeasure solutions adopted the centralized architecture. Nevertheless, we found that two solutions chose a hybrid approach to detect IFA. Another key observation regards the use of AI to detect IFA. Although it can help increase detection precision and reduce false positives, only 10% of authored solutions used an AI-based algorithm. AI-based solutions require additional computational and memory resources, which may penalize some nodes. That explains why researchers usually do not choose to rely on AI. Our comparative study found that most proposals are solutions against the third type of IFA. As discussed earlier, it represents the most harmful IFA type, as it implies the use of invalid requests. The first type of IFA, which uses valid requests, causes less harm compared to the third type. That explains why 80% of solutions chose to counter IFA with non-existent content, and only one was authored for the first type. However, some solutions counter both types. It is the case of 10% of solutions. These solutions usually rely on both router-based and producer-based information to detect attacks. On the other hand, solutions against collusive IFA represent only 13%. It shows that the research community did not study in detail this IFA variant. Our last observation from this comparative study points to control packets that solutions exchange during detection. Solutions adopted two approaches: the first one implies the creation of new specific packets. This approach was exclusively used by centralized solutions. The second approach consists of sending control information within exchanged *Interest* packets. It modifies *Interest* packets, usually their name, to include additional information for neighboring routers.

Table 4. Detection parameters used by existing solutions

Ref	Detection actors	Centralized detection	Distributed detection	AI-based	Congestion aware	Check interval	IFA type	Specific packets	Modified packets	Number of thresholds	Consumer classes
[10]	All routers	No	Autonomous	No	No	1*sec*	Non existent	No	No	01	-
[30]	All routers	No	Autonomous	No	No	60*ms*	Non existent	No	No	02	-
[32]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[94]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[56]	All routers Producers	No	Cooperative	No	No	-	Non existent	No	No	01	-
[95]	All routers	No	Cooperative	Fuzzy logic	No	real-time	-	No	No	02	Normal Malicious
[93]	All routers	No	Autonomous	No	No	-	Non existent	No	No	02	-
[45]	All routers	No	Autonomous	RBF NN	No	-	All types	No	No	02	-
[114]	All routers	No	Autonomous	SVM	No	time period	Non existent	No	No	-	-
[87]	Edge routers	No	Autonomous	No	No	-	Non existent	No	No	02	Legitimate Suspicious Malicious
[73]	Mon routers and DC	Selective nodes	Autonomous	No	No	Window q	Non existent	Yes	No	02	-
[72]	Mon routers and DC	Selective nodes	Autonomous	No	No	Window q	Collusive	Yes	No	01	-
[102]	DC	Global nodes	No	No	No	-	Non existent	Yes	No	02	-
[65]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[99]	Edge routers	No	Autonomous	No	No	-	Collusive	No	No	01	-
[79]	All routers	No	Autonomous	No	No	-	Collusive	No	No	02	-
[34]	Edge routers	No	Autonomous	No	No	Windows L	Non existent	No	*Interest* state and router ID	01	-
[98]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[42]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[116]	All routers	No	Autonomous	No	No	Time Δt	Non existent	No	No	01	-
[61]	Edge routers	No	Autonomous	No	No	Time window	Existent	No	No	02	-
[19], [18]	All routers	No	Autonomous	No	No	-	Non existent	No	No	03	-
[115]	All routers	No	Autonomous	No	No	-	Collusive	No	No	02	-
[110]	Producers	No	Autonomous	No	No	-	All types	No	No	Producer based	Normal Suspicious
[68]	Edge routers	No	Autonomous	No	No	Window ω	Non existent	No	No	01	-
[28]	Mon routers DC	Selective nodes	No	No	No	-	Non existent	Yes	No	03	-
[26]	All routers	No	Autonomous	No	No	time interval	Non existent	No	No	02	-
[21]	Edge routers	No	Autonomous	No	Yes	-	Non existent	No	No	04	Legitimate Suspicious Malicious
[35]	Producers	No	Autonomous	No	No	-	Non existent	No	No	Producer based	-
[84]	All routers	No	Autonomous	No	No	-	Non existent	No	No	02	-
[80]	All routers	No	Autonomous	No	No	-	Non existent	No	No	02	-
[20]	All routers Producers	No	Cooperative	No	No	-	All types	No	No	03	Normal Suspicious Harmful
[13]	All routers	No	Cooperative	No	No	-	Non-existent	No	No	02	-
[97]	All routers	No	Autonomous	No	No	Time window	Collusive	No	No	02	-
[86]	Core routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[96]	All routers	No	Cooperative	No	No	-	All types	No	PIT delay& content fees	-	-
[109]	Edge routers	No	Autonomous	No	No	-	Non existent	No	No	-	Legal Bad
[55]	All nodes	No	Autonomous	No	No	-	Non existent	No	No	-	-
Distribution	NA	10%	NA	7%	2%	NA	NA	10%	5%	NA	NA

Using control packets may introduce delays to the network, as routers need to process them. Solutions that rely on control packets during detection represents 15% of the overall solutions.

6.3 Act

The last activity of the CDA workflow is "Act". After collecting the traffic related information, and detecting the existence on an IFA, solutions act by taking mitigation actions to stop the attack. Table 5 summarizes the mitigation parameters used by existing solutions. Before we detail the mitigation actions that solutions use to stop attacks, we first discuss the mitigation parameters that solutions may employ during this phase.

6.3.1 Mitigation parameters. Table 5 lists the parameters that solutions may use during the mitigation process. The first column specifies proposals. The second column corresponds to the way a solution work to mitigate attackers, autonomous or cooperative. System nodes may cooperate or work independently to stop IFA. The third column specifies the nodes that take the action against attacks. The following field represents the action that a solution adopts to counter IFA. Mitigation actions are detailed in the following subsection. The next two columns indicate whether a solution uses special packets during the mitigation process. The "Control information" field lists the information that a solution shares during the mitigation process. The last column specifies whether a solution uses signed *Interest* packets or not.

6.3.2 Router-based mitigation actions.

(a) *Rate-limiting*: The first mitigation action that routers usually take after detecting an IFA is rate limiting. It consists of reducing the amount of traffic allowed from a given interface. When a router detects a malicious traffic, it first checks the source interface of this traffic. Following this, the router reduces the incoming traffic from this interface. The router may apply rate-limiting on the whole interface's traffic or just for a given namespace. The vast majority of existing IFA countermeasure solution act by limiting the rate of incoming traffic.

(b) *Block*: The second router-based mitigation action consists of blocking the traffic of an interface. The network router, when necessary, can block an interface for a period. The blocking action is usually considered as a second-level reaction. When rate limiting does not suffice to throttle an attack, the router considers blocking the interface for a period. Interface blocking is particularly used by edge routers. The blocking periods may vary during time. Routers could increase the blocking period when the previous amount of time did not help to stop the attack. Blocking actions could be prefix-based (i.e., applied on a specific prefix) as used in [84] and [114], or on interface-base (i.e., block all traffic of an interface) as used in [21] and [42].

6.3.3 Producer-based mitigation actions. The producer-based mitigation actions are essentially associated with blocking. Data producers, when needed, may block the incoming network traffic. Producer-based blocking action can be temporary or permanent.

(a) *Temporary block*: Temporary blocking consists of blocking the network traffic for a limited period. To throttle attacks, data producers use temporary blocking against malicious traffic. The blocked traffic could be the whole traffic of an interface or just a portion of the traffic (e.g., the network traffic heading to a particular service) as used in [20].

(b) *Permanent block*: Data producers may also use permanent blocking against consumers. Producers can implement a security policy to permanently block consumers when needed (i.e., blacklisting consumers). Producers do not use permanent block against network interfaces.

6.3.4 Key observations of our comparison. The fundamental observations of our comparative study on mitigation parameters are as follows: We first note that the number of solutions that cooperate to mitigate attacks represents 54%. The number may seem low as cooperation helps to better contain attacks, especially in the case of distributed IFA. However, cooperation implies a pre-configuration phase and the exchange of solution-based information. Scalability is also a challenge for cooperative solutions, especially in large deployments compared to the autonomous approach. We also notice from our comparison that autonomous mitigation is usually associated with

Table 5. Mitigation parameters used by existing solutions

Ref	Mitigation method	Mitigation actors	Mitigation actions	Specific packets	Specific namespace	Control information	Signed *Interests*
[10]	Autonomous	All routers	Rate-limit	No	No	No	No
[10]	Cooperative	All routers	Rate-limit	Yes	-	Rate limit announcements	No
[30]	Cooperative	All routers	Rate-limit	Yes	`/pushback/alerts/`	Reduced rate	No
[32]	Cooperative	All routers	Rate-limit	Yes	No	Spoofed *Data*	No
[94]	Cooperative	All routers	Forwards *Interest* without using PIT	Yes	No	Interfaces list	No
[56]	Cooperative	All routers Producers	Sends Nack	Yes	No	Malicious *Interest*	No
[95]	Cooperative	Edge routers	Rate-limit (prefix)	Yes	`/ALERT/IFA/`	Malicious prefix	No
[93]	Autonomous	All routers	Rate-limit (prefix)	No	No	No	No
[45]	Cooperative	All routers	Rate-limit	Yes	`/pushbackmessage/alert/`	Reduced rate	No
[114]	Cooperative	All routers	Block (prefix)	Yes	Empty	Malicious prefixes	No
[87]	Autonomous	Edge routers	Rate-limit Block	Yes	-	Blocked user	No
[73], [72]	Cooperative	Mon routers	Rate-limit (prefix)	Yes	-	Infected prefixes	No
[102]	Cooperative	Edge routers	Block	Yes	`/CTRL/`	Malicious prefixes	No
[34]	Cooperative	All routers	Rate-limit (prefix)	No	No	No	No
[98]	Cooperative	All routers	Rate-limit	Yes	No	Spoofed *Data*	No
[42]	Cooperative	All routers	Block	Yes	No	Spoofed *Data*	No
[116]	Cooperative	All routers	Rate-limit (prefix)	Yes	No	Malicious prefix	No
[61]	Autonomous	Edge routers	Block	No	No	No	No
[19]	Autonomous	All routers	Rate-limit	No	No	No	No
[115]	Autonomous	All routers	Rate-limit	No	No	No	No
[18]	Autonomous	Edge routers	Rate-limit Block	No	No	No	No
[110]	Cooperative	All routers Producers	Rate-limit Block	Yes	No	RSN, PREF, C FakeList	No
[68]	Autonomous	Edge routers	Rate-limit	No	No	No	No
[28]	Cooperative	Edge routers	Block	Yes	`/ndn/ddos/flooding/`	Interfaces list	Yes
[26]	Cooperative	All routers	Rate-limit (prefix)	Yes	-	Malcious prefix	No
[21]	Autonomous	Edge routers	Block	No	No	No	No
[35]	Cooperative	All routers Producers	Rate-limit	Yes	No	Affected prefixes	No
[84]	Autonomous	All routers	Block (prefix)	No	No	No	No
[80]	Autonomous	All routers	Rate-limit	No	No	No	No
[20]	Cooperative	Edge routers	Rate-limit (prefix) and Block	Yes	`/ndn/PCIP` `/ndn/RCIP`	Affected prefix	Yes
[13]	Cooperative	Edge routers	Rate-limit	Yes	No	No	
[97]	Autonomous	All routers	Delete *Interests*	No	No	No	No
[86]	Autonomous	Core routers	Bloom filter	No	No	No	No
[96]	Cooperative	All routers	Rate-limit	Yes	No	Insufficient fees	No
[109]	Autonomous	Edge routers	Rate-limit	No	No	No	No
[55]	Autonomous	All nodes	Rate-limit	No	No	No	No

edge routers, and solutions that employ core routers are always cooperative. The next observation regards the actions that authored solutions chose to mitigate attacks. Our study found that 66% of them apply rate-limiting. Countermeasure solutions chose this action for two reasons: (1) When applied on a core interface, it does not stop all the traffic. (2) Avoid penalizing legitimate consumers connected to edge routers after behavioral changes. (3) Avoid penalizing the traffic of a specific producer in case of prefix-based rate limiting. Routers usually employ rate limiting for a period. On the other hand, 29% of proposals adopted blocking as a mitigation action. We first notice that interface-based blocking is always associated with edge routers, which is understandable. Core routers do not block the whole interface's traffic. However, all routers adopted prefix-based blocking. Our comparative study found that cooperative solutions employ control packets during mitigation. Solutions use them to exchange information. We found that the most shared information regards rate limiting. Routers, especially core routers, inform neighboring nodes after the application of traffic limitation. Some solutions also send the applied rate. The second most shared information within cooperative solutions is the affected/malicious prefix. It is associated with prefix-based actions. Routers use this information to ask other routers to limit/stop forwarding packets to that prefix. Our last observation regards the nature of packets used by cooperative solutions to share information. System nodes usually use *Interest* packets to exchange information between them. However, we discovered that only 14% of countermeasure solutions used signed *Interest* packets to share information, which is very low. Using non-signed *Interest* packets constitutes a threat to the reliability of a solution, as we will see in the following section.

7 UNCONSIDERED IFA SCENARIOS

All existing IFA solutions may lack the detection of attacks in some particular scenarios, which attackers can take advantage of to flood the network and/or penalize legitimate consumers. In this section, we present and explain several unconsidered IFA scenarios.

7.1 Attacking scenarios against non cooperative solutions

This subsection groups several attacking scenarios that target non cooperative IFA solutions.

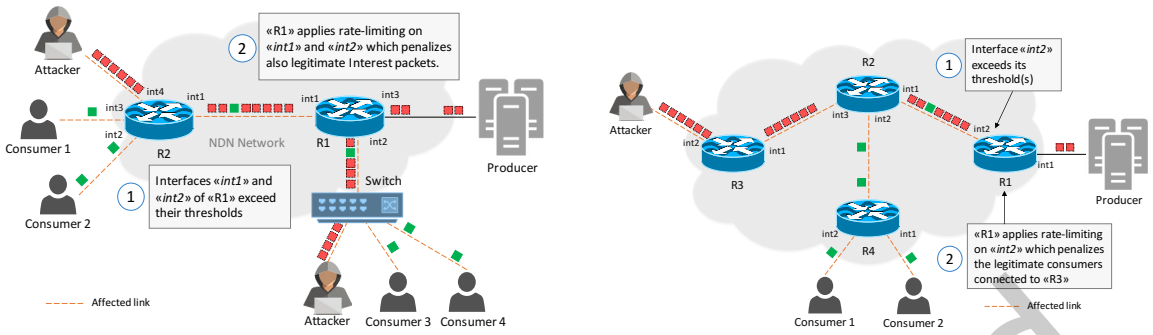
7.1.1 Targeting neighboring consumers in non-cooperative solutions. Targeting neighboring consumers in non-cooperative solutions. The attacking scenario illustrated in Fig. 6(a) shows how an attacker could take advantage of a non-cooperative solution to affect legitimate consumers. In this scenario, Router *R1* takes a defensive action against its interfaces *int1* and *int2* because they reached their respective thresholds due to the malicious traffic. As a result, Router *R1* will penalize legitimate consumers connected to Router *R2* and those behind the switch.

7.1.2 Targeting distant consumers in non-cooperative solutions. Similarly, the scenario in Fig. 6(b) shows that attackers can also affect a distant legitimate consumer in a case of a non-cooperative solution. In this scenario, The router *R2* applied rate limiting on its interface *int2* in response to the malicious generated by the attacker, which lead to affecting the legitimate consumers. The attacker was able to penalize the distant *consumer1* and *consumer2*.

7.2 Attacking scenarios against cooperative solutions

This subsection groups several attacking scenarios that target cooperative IFA solutions.

7.2.1 Countering alert-based solutions with a compromised edge router. This attacking scenario counters solutions that rely on edge routers to stop attackers (e.i., edge routers are the only mitigation actors). Attackers can get around by taking control of the edge router that they are connected to. Adversary nodes will have the ability to continue flooding the network because the edge router will ignore all received solution-based alerts, as shown



(a) Attacking scenario against neighboring consumers in non-cooperative solutions

(b) Attacking scenario against distant consumers in non-cooperative solutions

Fig. 6. IFA Scenarios against non-cooperative solutions

in 7(a). Routers could reduce the impact of the attack when they take defensive actions. However, in a solution where the edge router is the only mitigation actor, attackers will continue flooding the network.

7.2.2 Targeting legitimate consumers with alert messages. Some IFA solutions use alert messages to exchange information and action decisions. Attackers could take advantage of this feature to target legitimate consumers as shown in Fig. 7(b). In this scenario, the attacker forges an alert message and sends it to Router *R2* to push it to take defensive action against the legitimate consumers connected to it. The attacker can conduct such attacks only if the solution uses non-signed alert messages.

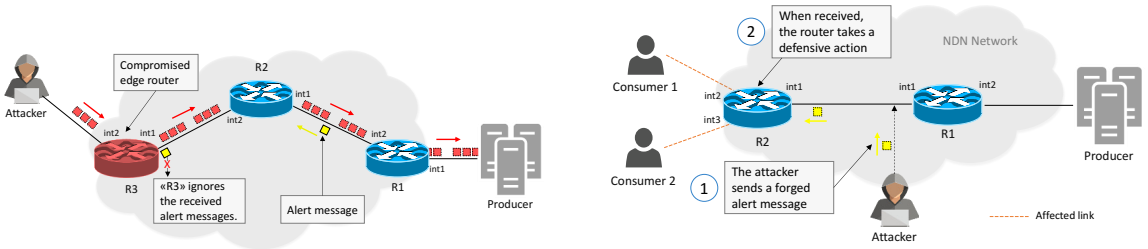
7.2.3 Targeting routers resources with forged alert messages. Another way of using solution-based alert messages is to target routers' resources. As shown in Fig. 7(c), the attacker continually sends forged alert messages to Router *R2* to stress it with signature verification and leads it to computation overhead.

7.2.4 Targeting routers resources with forged NACK packets. Similar to the previous scenario, attackers can also lunch attacks against routers that rely on NACK packets to send solution-based messages like [35, 56]. To do so, attackers flood the targeted router with forged NACK packets to penalize him with computation overhead due to signature verification. Figure 8(a) illustrates this attacking scenario.

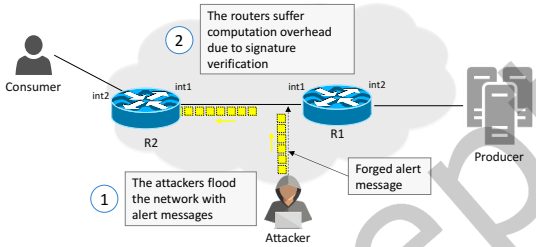
7.2.5 Flooding the network with solution-based spoofed Data packets. Some solutions like [32] use spoofed *Data* packets to counter malicious nodes. Attackers can take advantage of this feature to flood the network as shown in Fig. 8(b). In this scenario, the attackers who are in control of the edge router flood the network with forged *Interest* packets. Router *R1* sends back spoofed *Data* packets to edge Router *R3* to stop adversary nodes. Router *R3* will take no action against the malicious nodes as it is controlled.

7.2.6 Countering prefix-based solutions. As mitigation action, some solutions like [26, 84] apply rate-limiting or blocking on name prefixes that routers consider as malicious or under attack. However, attackers can easily overcome this restriction by changing the prefix of forged *Interest* packets to keep flooding the network, as shown in Fig. 9(a).

7.2.7 Affecting legitimate traffic in prefix-based solutions. Another attacking scenario against prefix-based solutions is illustrated in Fig. 9(b). The goal of the attacker in this scenario is to affect legitimate traffic heading to a

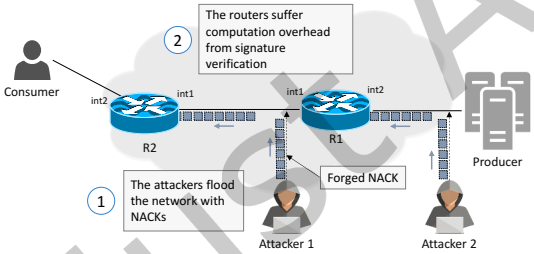


(a) Countering alert-based solutions with a compromised edge router (b) Attacking scenario against legitimate consumers with alert message

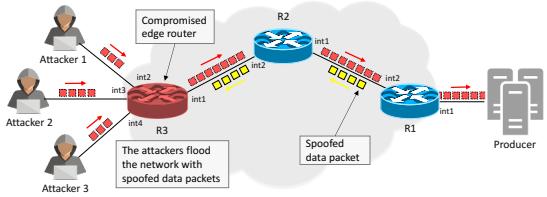


(c) Targeting routers with forged alert messages

Fig. 7. IFA scenarios against alert messages based solutions



(a) Targeting routers with forged NACK packets

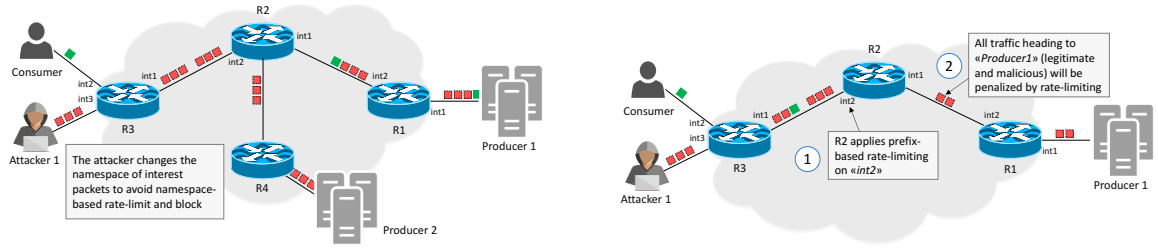


(b) Flooding the network with solution-based spoofed *Data* packets

Fig. 8. IFA scenarios with forged packets

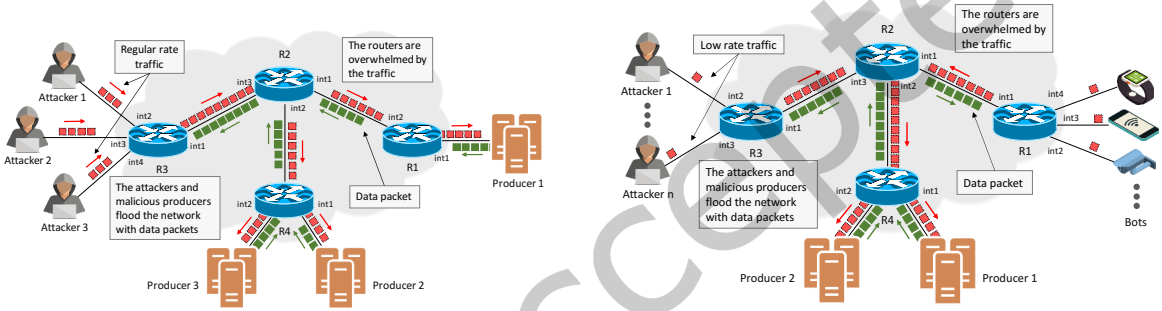
specific producer. To do so, the attacker flood the network with forged *Interest* packets with the targeted prefix to push routers to take defensive actions against this prefix, which penalizes also legitimate traffic.

7.2.8 Targeting the network with a distributed collusive attack. Figure 10(a) illustrates a scenario of a distributed collusive attack. Compared to the collusive attack presented in the literature, in this scenario, attackers work with a distributed group of malicious producers to overwhelm the network. Additionally, adversary nodes send with



(a) Targeting prefix-based solutions (b) Targeting legitimate traffic in prefix-based solutions

Fig. 9. IFA scenarios against prefix-based solutions



(a) Targeting the network with a distributed collusive attack (b) Targeting the network with a low-rate distributed collusive attack

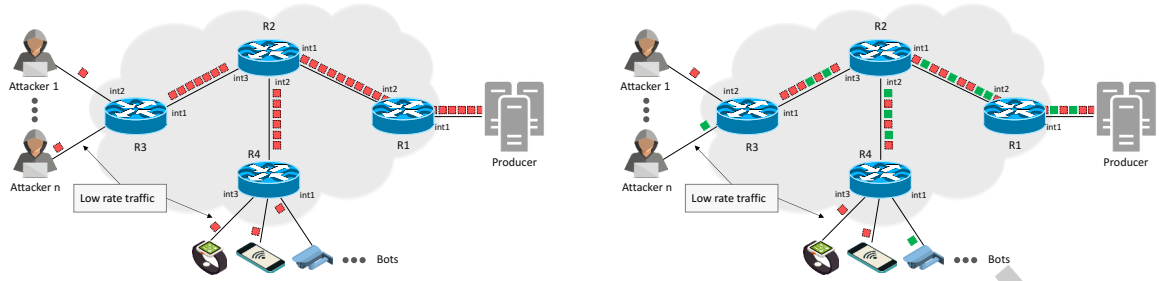
Fig. 10. Distributed collusive IFA scenarios

regular rates. Existing solutions rely on metrics like PIT usage and traffic rate, which are linked to the aggressive nature of adversary nodes. That makes it difficult for existing solutions to detect this attacking scenario. Because of its distributed nature, this attacking scenario generates high traffic and introduces important delays to the network even with regular sending rates.

7.2.9 Targeting the network with a low-rate distributed collusive attack. Another variant of the distributed collusive attack is the low-rate distributed collusive IFA. In this scenario, a large distributed number of attackers or infected bots request *Data* packets from malicious producers with low sending rates, as shown in Fig. 10(b). Similar to the previous scenario, existing collusive solutions rely on high traffic metrics to detect collusive attacks. It makes them inefficient against this attacking scenario.

7.2.10 Targeting the network with low-rate distributed IFA. The attacking scenario depicted in 11(a) represent a distributed IFA with low sending rate. This attacking solution works with all solutions that use the traffic rate and PIT usage as detection metrics. Attacker overcome these solution by adopting a low sending rate to keep flooding the network with malicious *Interest* packets.

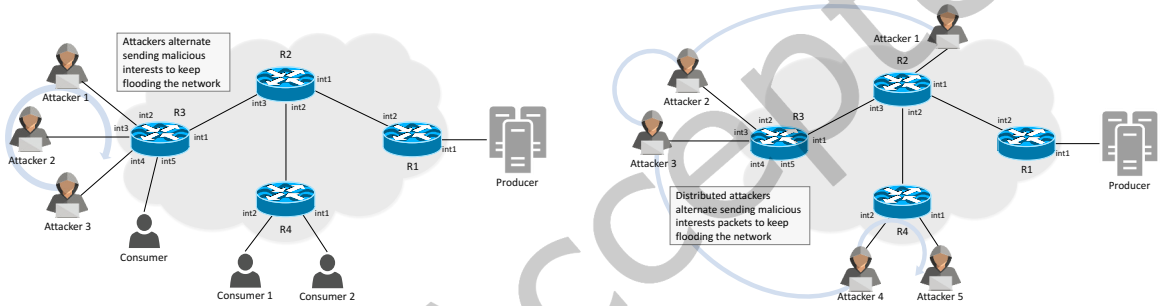
7.2.11 Targeting the network with low-rate and mixed distributed IFA. An even more hard-to-detect attacking scenario than the previous one is illustrated in 11(b). In these particular scenarios attackers and controlled nodes



(a) Targeting the network with low-rate distributed IFA

(b) Targeting the network with low-rate mixed distributed IFA

Fig. 11. Low-rate distributed IFA scenarios



(a) Local IFA: case of smartly behaving attackers

(b) Distributed IFA: case of smartly behaving attackers

Fig. 12. Smart IFA Scenarios

flood the network with valid and invalid *Interest* packets. This permits to attacker to counter other detection metrics like the satisfaction ratio and timed-out *Interest* packets.

7.2.12 Scenario of a Smart IFA. In this scenario, attackers adopt a smart behaviour when launching IFA to avoid mitigation and keep flooding the network. The adversary nodes cooperate by sending malicious traffic one after another. The goal of attackers is to keep flooding the network in case an attacker gets mitigated. Another more sophisticated scenario is when an attacker keeps sending forged *Interest* packets and before reaching solution's thresholds it stops and another attacker continue the attack. This attacking scenarios could be local as showed in Fig. 12(a) or distributed as illustrated in Fig. 12(b).

8 LESSONS LEARNED AND FUTURE DIRECTIONS

In this survey, we reviewed the literature on IFA depending on the information they collect, the metrics and parameters they use, and the actions they take. Based on our analysis, we found that IFA still represents a threat to NDN networks. The research community needs to conduct significant work to cover all its aspects. In this section, we list the key lessons learned and highlight research directions.

8.1 Lessons Learned

In the following, we list the fundamental lessons learned from this survey that could potentially help security researchers:

(1) **Start by considering other IFA variants instead of conventional scenarios**. Existing works generally focus on a specific type of IFA. However, as outlined in section 7, multiple IFA scenarios are still untreated by state-of-the-art works. Solutions need to widen their field of action and go off the beaten path to cover the full spectrum of the attack.
(2) **Consider additional parameters to enhance the detection accuracy**. As stated in this survey, existing solutions may classify legitimate requests as malicious, which leads to penalizing honest consumers. To this end, solutions need to consider additional parameters and proprieties during the detection process.
(3) **Rethink the solution's algorithm to adapt to behavioral changes**. As outlined in Sec. 7, existing solutions are incapable of detecting attackers when they adopt different behaviors from those expected by the algorithm.

8.2 Future Research Directions

Accurate detection, embedded security, and autonomous adaptation are some of the research goals that need to be achieved while designing IFA solutions. In the following, we outline potential research directions:

- **Accurate detection and mitigation -** Many effective countermeasure solutions are available. However, authored solutions are explicit and generally detect specific attacking scenarios. Additionally, these solutions cannot even stop the scenario that they are intended to detect when some conditions change, which leads to flooding the network with malicious traffic. In addition, false-positive rates are tightly linked to detection conditions, which leads to unfairly blocking consumers and their legitimate traffic, as shown in Sec.7. Newly designed solutions need to be more reliable. They need to revisit detection parameters and adopt intelligent approaches to cope with different IFA scenarios. For example, broad cooperation between network nodes could be considered, and concepts like federated learning and hybrid-based solutions could be of help.
- **Embedded security mechanisms -** This research pathway suggests that solutions should embed security mechanisms to ensure reliable and undisrupted interconnectivity. The following prerequisites need to be part of a designed solution: (1) The solution-based information that nodes exchange needs to be securely validated and verified. (2) The solution needs to guarantee a secure and reliable transmission mechanism to ensure the integrity and authenticity of this information. (3) The solution needs to implement trust policies between participating entities. The implementation of these requirements will ensure a healthy environment among system nodes.
- **Autonomous adaptation -** The existing solutions can face a stable attacking scenario. However, an intelligent attacker can alter between legitimate and malicious behavior to avoid detection. Hence, this research direction recommends that newly designed solutions should be able to autonomously adapt to different situations, including the intelligent attacker scenarios. Additionally, solutions need to adapt their parameters following network situations, especially in high traffic peaks.
- **Hybrid approaches -** The following research pathway proposes to use hybrid solutions to detect and mitigate IFA. For example, a router may combine two solutions: one for conventional situations, in which it applies predefined rules, and another for non-conventional situations to detect behavioral changes. Another example of a hybrid approach consists of a modular solution. In this hybrid approach, tasks are distributed among the nodes of a system. Each node represents a module of the solution. Nevertheless, implementing hybrid solutions is challenging. It requires coordination between system nodes, which rises interoperability concerns. Another challenge consists of finding a balance between performance and resource consumption.

- **Deployment-friendly -** Another aspect of a reliable IFA solution relies on its deployment. Newly designed solutions need to be able to scale at any level without affecting the overall system. Additionally, solutions need to offer easy deployment mechanisms for recently added nodes without adding significant load on nodes' resources. A solution, as good as it is, needs to have a minimum impact on a device's resources so it can work smoothly even in cases of high traffic peaks. Newly designed solutions need to have a low resource consumption fingerprint so attackers cannot take advantage of it to take down network nodes.

9 CONCLUSION

Many solutions were authored since the proposal of the first IFA countermeasure mechanism. However, the proposed mechanisms still do not cover the full spectrum of potential IFAs. This survey broadly discussed IFA. Following that, it detailed, classified and compared the state-of-the-art solutions. After that, the survey discussed open issues through the presentation and analysis of several non-conventional attacking scenarios that were not considered before. Finally, the survey discussed the lessons learned and research directions by providing the requirements for a more robust and efficient IFA solution.

ACKNOWLEDGMENTS

This work is partially supported by National Science Foundation awards CNS-2104700, CNS-2016714, and CBET-2124918, the National Institutes of Health (NIGMS/P20GM109090), the Nebraska University Collaboration Initiative, and the Nebraska Tobacco Settlement Biomedical Research Development Funds.

REFERENCES

[1] Cisco Visual Networking Index: Forecast and Trends 2018–2023. *2020.* Retrieved Oct 2, 2020 from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
[2] NDN Data packet. *2020.* Retrieved Oct 3, 2020 from https://named-data.net/doc/NDN-packet-spec/current/data.html
[3] NDN Packet Format Specification version 0.3. *2020.* Retrieved Sep 17, 2020 from https://named-data.net/doc/NDN-packet-spec/current/interest.html
[4] Signed Interest Packet. *2020.* Retrieved Sep 17, 2020 from https://named-data.net/doc/NDN-packet-spec/current/signed-interest.html
[5] Muhammad Aamir and Syed Mustafa Ali Zaidi. 2015. Denial-of-service in content centric (named data) networking: a tutorial and state-of-the-art survey. *Security and Communication Networks* 8, 11 (2015), 2037–2059.
[6] Eslam G AbdAllah, Hossam S Hassanein, and Mohammad Zulkernine. 2015. A survey of security attacks in information-centric networking. *IEEE Communications Surveys & Tutorials* 17, 3 (2015), 1441–1454.
[7] Hila Ben Abraham and Patrick Crowley. 2017. Controlling strategy retransmissions in named data networking. In *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE, 70–81.
[8] Alexander Afanasyev, J Alex Halderman, Scott Ruoti, Kent Seamons, Yingdi Yu, Daniel Zappala, and Lixia Zhang. 2016. Content-based security for the web. In *Proceedings of the 2016 New Security Paradigms Workshop*. 49–60.
[9] Alexander Afanasyev, Xiaoke Jiang, Yingdi Yu, Jiewen Tan, Yumin Xia, Allison Mankin, and Lixia Zhang. 2017. NDNS: A DNS-like name service for NDN. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–9.
[10] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. 2013. Interest flooding attack and countermeasures in Named Data Networking. In *2013 IFIP Networking Conference*. IEEE, 1–9.
[11] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. 2012. A survey of information-centric networking. *IEEE Communications Magazine* 50, 7 (2012), 26–36.
[12] Samir Al-Sheikh, Matthias Wählisch, and Thomas C Schmidt. 2015. Revisiting countermeasures against ndn interest flooding. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. 195–196.
[13] Mohammad Alhisnawi and Mahmood Ahmadi. 2020. Detecting and Mitigating DDoS Attack in Named Data Networking. *Journal of Network and Systems Management* (2020), 1343–1356.
[14] Aubrey Alston and Tamer Refaei. 2016. Neutralizing interest flooding attacks in named data networks using cryptographic route tokens. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*. IEEE, 85–88.
[15] Moreno Ambrosin, Alberto Compagno, Mauro Conti, Cesar Ghali, and Gene Tsudik. 2018. Security and privacy analysis of national science foundation future internet architectures. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 1418–1442.

[16] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David G Andersen, John W Byers, et al. 2011. XIA: An architecture for an evolvable and trustworthy Internet. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. 1–6.

[17] Tom Anderson, Ken Birman, Robert Broberg, Matthew Caesar, Douglas Comer, Chase Cotton, Michael J Freedman, Andreas Haeberlen, Zachary G Ives, Arvind Krishnamurthy, et al. 2013. The nebula future internet architecture. In *The Future Internet Assembly*. Springer, 16–26.

[18] Abdelmadjid Benarfa, Muhammad Hassan, , Eleonora Losiouk, Alberto Compagno, Mohamed Bachir Yagoubi, and Mauro Conti. 2020. ChoKIFA+: an early detection and mitigation approach against interest flooding attacks in NDN. *International Journal of Information Security* (2020).

[19] Abdelmadjid Benarfa, Muhammad Hassan, Alberto Compagno, Eleonora Losiouk, Mohamed Bachir Yagoubi, and Mauro Conti. 2019. ChoKIFA: A New Detection and Mitigation Approach Against Interest Flooding Attacks in NDN. In *International Conference on Wired/Wireless Internet Communication*. Springer, 53–65.

[20] Ahmed Benmoussa, Abdou el Karim Tahari, Chaker Abdelaziz Kerrache, Nasreddine Lagraa, Abderrahmane Lakas, Rasheed Hussain, and Farhan Ahmad. 2020. MSIDN: Mitigation of sophisticated interest flooding-based DDoS attacks in named data networking. *Future Generation Computer Systems* 107 (2020), 293–306.

[21] Ahmed Benmoussa, Abdou el Karim Tahari, Nasreddine Lagaa, Abderrahmane Lakas, Farhan Ahmad, Rasheed Hussain, Chaker Abdelaziz Kerrache, and Fatih Kurugollu. 2019. A Novel Congestion-Aware Interest Flooding Attacks Detection Mechanism in Named Data Networking. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–6.

[22] Madhurima Buragohain and Sukumar Nandi. 2020. Demystifying Security on NDN: A Survey of Existing Attacks and Open Research Challenges. In *The" Essence" of Network Security: An End-to-End Panorama*. Springer, 241–261.

[23] Jianxun Cao, Dan Pei, Xiaoping Zhang, Beichuan Zhang, and Youjian Zhao. 2016. Fetching popular data from the nearest replica in NDN. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–9.

[24] Kevin Chan, Bongjun Ko, Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. Fuzzy interest forwarding. In *Proceedings of the Asian Internet Engineering Conference*. 31–37.

[25] Lyman Chapin, Dr. David D. Clark, Robert T. Braden, Russ Hobby, and Dr. Vinton G. Cerf. 1991. Towards the Future Internet Architecture. RFC 1287. https://rfc-editor.org/rfc/rfc1287.txt

[26] Jing Chen, Guanglin Xing, Mengtian Cui, Hong Huo, and Rui Hou. 2019. Isolation Forest Based Interest Flooding Attack Detection Mechanism in NDN. In *2019 2nd International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 58–62.

[27] Shuoshuo Chen and Fabrice Mizero. 2015. A survey on security in named data networking. *arXiv preprint arXiv:1512.04127* (2015).

[28] Guang Cheng, Lixia Zhao, Xiaoyan Hu, Shaoqi Zheng, Hua Wu, Ruidong Li, and Chengyu Fan. 2019. Detecting and Mitigating A Sophisticated Interest Flooding Attack in NDN from the Network-Wide View. In *2019 IEEE First International Workshop on Network Meets Intelligent Computations (NMIC)*. IEEE, 7–12.

[29] Seungoh Choi, Kwangsoo Kim, Seongmin Kim, and Byeong-hee Roh. 2013. Threat of DoS by interest flooding attack in content-centric networking. In *The International Conference on Information Networking 2013 (ICOIN)*. IEEE, 315–319.

[30] Alberto Compagno, Mauro Conti, Paolo Gasti, and Gene Tsudik. 2013. Poseidon: Mitigating interest flooding DDoS attacks in named data networking. In *38th annual IEEE conference on local computer networks*. IEEE, 630–638.

[31] Alberto Compagno, Mauro Conti, Cesar Ghali, and Gene Tsudik. 2015. To NACK or not to NACK? negative acknowledgments in information-centric networking. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–10.

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

[33] Ikram Ud Din, Suhaidi Hassan, Muhammad Khurram Khan, Mohsen Guizani, Osman Ghazali, and Adib Habbal. 2017. Caching in information-centric networking: Strategies, challenges, and future research directions. *IEEE Communications Surveys & Tutorials* 20, 2 (2017), 1443–1474.

[34] Kun Ding, Yun Liu, Hsin-Hung Cho, Han-Chieh Chao, and Timothy K Shih. 2016. Cooperative detection and protection for interest flooding attacks in named data networking. *International Journal of Communication Systems* 29, 13 (2016), 1968–1980.

[35] Jiaqing Dong, Kai Wang, Wei Quan, and Hao Yin. 2020. InterestFence: Simple but efficient way to counter interest flooding attack. *Computers & Security* 88 (2020), 101628.

[36] Nikos Fotiou, Pekka Nikander, Dirk Trossen, and George C Polyzos. 2010. Developing information networking further: From PSIRP to PURSUIT. In *International Conference on Broadband Communications, Networks and Systems*. Springer, 1–13.

[37] Gerardo García, Andrzej Beben, Francisco J Ramón, Adrián Maeso, Ioannis Psaras, George Pavlou, Ning Wang, Jarosław Śliwiński, Spiros Spirou, Sergios Soursos, et al. 2011. COMET: Content mediator architecture for content-aware networks. In *2011 Future Network & Mobile Summit*. IEEE, 1–8.

[38] Paolo Gasti and Gene Tsudik. 2018. Content-Centric and Named-Data Networking Security: The Good, The Bad and The Rest. In *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE, 1–6.

[39] Paolo Gasti, Gene Tsudik, Ersin Uzun, and Lixia Zhang. 2013. DoS and DDoS in named data networking. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–7.
[40] Cesar Ghali, Gene Tsudik, Ersin Uzun, and Christopher A Wood. 2017. Closing the floodgate with stateless content-centric networking. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–10.
[41] Chavoosh Ghasemi, Hamed Yousefi, Kang G Shin, and Beichuan Zhang. 2018. MUCA: New routing for named data networking. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 289–297.
[42] Rui Hou, Min Han, Jing Chen, Wenbin Hu, Xiaobin Tan, Jiangtao Luo, and Maode Ma. 2019. Theil-Based Countermeasure against Interest Flooding Attacks for Named Data Networks. *IEEE Network* 33, 3 (2019), 116–121.
[43] Van Jacobson, Marc Mosko, D Smetters, and Jose Garcia-Luna-Aceves. 2007. Content-centric networking. *Whitepaper, Palo Alto Research Center* (2007), 2–4.
[44] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. 2009. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 1–12.
[45] Amin Karami and Manel Guerrero-Zapata. 2015. A hybrid multiobjective rbf-pso method for mitigating dos attacks in named data networking. *Neurocomputing* 151 (2015), 1262–1282.
[46] Hakima Khelifi, Senlin Luo, Boubakr Nour, and Sayed Chhattan Shah. 2018. Security and privacy issues in vehicular named data networks: An overview. *Mobile Information Systems* 2018 (2018).
[47] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. 2007. A data-oriented (and beyond) network architecture. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. 181–192.
[48] Naveen Kumar, Ashutosh Kumar Singh, Abdul Aleem, and Shashank Srivastava. 2019. Security Attacks in Named Data Networking: A Review and Research Directions. *Journal of Computer Science and Technology* 34, 6 (2019), 1319–1350.
[49] Naveen Kumar, Ashutosh Kumar Singh, and Shashank Srivastava. 2019. Feature selection for interest flooding attack in named data networking. *International Journal of Computers and Applications* (2019), 1–10.
[50] Craig A Lee, Zhiyi Zhang, Yukai Tu, Alex Afanasyev, and Lixia Zhang. 2018. Supporting virtual organizations using attribute-based encryption in named data networking. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 188–196.
[51] Vince Lehman, AKM Mahmudul Hoque, Yingdi Yu, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2016. A secure link state routing protocol for NDN. *Tech. Rep. NDN-0037* (2016).
[52] Yanbiao Li, Zhiyi Zhang, Xin Wang, Edward Lu, Dafang Zhang, and Lixia Zhang. 2019. A secure sign-on protocol for smart homes over named data networking. *IEEE Communications Magazine* 57, 7 (2019), 62–68.
[53] Zhaogeng Li and Jun Bi. 2014. Interest cash: an application-based countermeasure against interest flooding for dynamic content in named data networking. In *Proceedings of The Ninth International Conference on Future Internet Technologies*. 1–6.
[54] Zhuo Li, Yaping Xu, Beichuan Zhang, Liu Yan, and Kaihua Liu. 2018. Packet forwarding in named data networking requirements and survey of solutions. *IEEE Communications Surveys & Tutorials* 21, 2 (2018), 1950–1987.
[55] Gang Liu, Wei Quan, Nan Cheng, Bohao Feng, Hongke Zhang, and Xuemin Sherman Shen. 2018. BLAM: Lightweight Bloom-filter based DDoS mitigation for information-centric IoT. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–7.
[56] Gang Liu, Wei Quan, Nan Cheng, Kai Wang, and Hongke Zhang. 2018. Accuracy or delay? A game in detecting interest flooding attacks. *Internet Technology Letters* 1, 2 (2018), e31.
[57] Roman Lutz. 2016. Security and privacy in future internet architectures-benefits and challenges of content centric networks. *arXiv preprint arXiv:1601.01278* (2016).
[58] Elisa Mannes and Carlos Maziero. 2019. Naming content on the network layer: a security analysis of the information-centric network model. *ACM Computing Surveys (CSUR)* 52, 3 (2019), 1–28.
[59] Spyridon Mastorakis, Abderrahmen Mtibaa, Jonathan Lee, and Satyajayant Misra. 2020. ICedge: When Edge Computing Meets Information-Centric Networking. *IEEE Internet of Things Journal* 7, 5 (2020), 4203–4217.
[60] Abderrahmen Mtibaa and Spyridon Mastorakis. 2020. NDNTP: A Named Data Networking Time Protocol. *arXiv preprint arXiv:2007.07807* (2020).
[61] Yoshimichi Nakatsuka, Janaka L Wijekoon, and Hiroaki Nishi. 2018. FROG: A Packet Hop Count based DDoS Countermeasure in NDN. In *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 00492–00497.
[62] Eric Newberry and Beichuan Zhang. 2019. On the Power of In-Network Caching in the Hadoop Distributed File System. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 89–99.
[63] Tan Nguyen, Hoang-Long Mai, Rémi Cogranne, Guillaume Doyen, Wissam Mallouli, Luong Nguyen, Moustapha El Aoun, Edgardo Montes De Oca, and Olivier Festor. 2019. Reliable detection of interest flooding attack in real deployment of named data networking. *IEEE Transactions on Information Forensics and Security* 14, 9 (2019), 2470–2485.
[64] Tan Nguyen, Xavier Marchal, Guillaume Doyen, Thibault Cholez, and Rémi Cogranne. 2017. Content poisoning in named data networking: Comprehensive characterization of real deployment. In *2017 IFIP/IEEE Symposium on Integrated Network and Service*

Management (IM). IEEE, 72–80.

[65] Tan N Nguyen, Rémi Cogranne, Guillaume Doyen, and Florent Retraint. 2015. Detection of interest flooding attacks in named data networking using hypothesis testing. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–6.

[66] Bin Pang, Ru Li, Xin Zhang, Jinshan Shi, and Manxin Huang. 2017. Research on interest flooding attack analysis in conspiracy with content providers. In *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, 543–547.

[67] Ioannis Psaras, Onur Ascigil, Sergi Rene, George Pavlou, Alex Afanasyev, and Lixia Zhang. 2018. Mobile data repositories at the edge. In {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*.

[68] Cong Pu, Nathaniel Payne, and Jacqueline Brown. 2019. Self-Adjusting Share-Based Countermeasure to Interest Flooding Attack in Named Data Networking. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 142–147.

[69] Sandesh Rai and Dependra Dhakal. 2018. A survey on detection and mitigation of interest flooding attack in named data networking. In *Advanced Computational and Communication Paradigms*. Springer, 523–531.

[70] Sandesh Rai, Kalpana Sharma, and Dependra Dhakal. 2019. A survey on detection and mitigation of distributed denial-of-service attack in named data networking. In *Advances in communication, cloud, and big data*. Springer, 163–171.

[71] Sanjeev Kaushik Ramani, Reza Tourani, George Torres, Satyajayant Misra, and Alexander Afanasyev. 2019. NDN-ABS: Attribute-Based Signature Scheme for Named Data Networking. In *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 123–133.

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

[73] Hani Salah, Julian Wulfheide, and Thorsten Strufe. 2015. Coordination supports security: A new defence mechanism against interest flooding in NDN. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*. IEEE, 73–81.

[74] Klaus Schneider, Cheng Yi, Beichuan Zhang, and Lixia Zhang. 2016. A practical congestion control scheme for named data networking. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 21–30.

[75] Ivan Seskar, Kiran Nagaraja, Sam Nelson, and Dipankar Raychaudhuri. 2011. Mobilityfirst future internet architecture project. In *Proceedings of the 7th Asian Internet Engineering Conference*. 1–3.

[76] Wentao Shang, Zhehao Wang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. 2017. Breaking out of the cloud: Local trust management and rendezvous in Named Data Networking of Things. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. 3–13.

[77] Junxiao Shi. 2017. *Named data networking in local area networks*. Ph. D. Dissertation. The University of Arizona.

[78] Junxiao Shi, Eric Newberry, and Beichuan Zhang. 2017. On broadcast-based self-learning in named data networking. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 1–9.

[79] Tetsuya Shigeyasu and Ayaka Sonoda. 2018. Distributed Approach for Detecting Collusive Interest Flooding Attack on Named Data Networking. In *International Conference on Network-Based Information Systems*. Springer, 76–86.

[80] Ryoki Shinohara, Takashi Kamimoto, Kazuya Sato, and Hiroshi Shigeno. 2016. Cache control method mitigating packet concentration of router caused by interest flooding attack. In *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 324–331.

[81] Salvatore Signorello, Samuel Marchal, Jérôme François, Olivier Festor, and Radu State. 2017. Advanced interest flooding attacks in named-data networking. In *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–10.

[82] Won So, Ashok Narayanan, and David Oran. 2013. Named data networking on a router: Fast and DoS-resistant forwarding with hash tables. In *Architectures for Networking and Communications Systems*. IEEE, 215–225.

[83] Tian Song, Haowei Yuan, Patrick Crowley, and Beichuan Zhang. 2015. Scalable name-based packet forwarding: From millions to billions. In *Proceedings of the 2nd ACM conference on information-centric networking*. 19–28.

[84] Jianqiang Tang, Zhongyue Zhang, Ying Liu, and Hongke Zhang. 2013. Identifying interest flooding in named data networking. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. IEEE, 306–310.

[85] Reza Tourani, Satyajayant Misra, Travis Mick, and Gaurav Panwar. 2017. Security, privacy, and access control in information-centric networking: A survey. *IEEE communications surveys & tutorials* 20, 1 (2017), 566–600.

[86] Reza Tourani, George Torres, and Satyajayant Misra. 2020. PERSIA: a PuzzlE-based InteReSt FloodIng Attack Countermeasure. In *Proceedings of the 7th ACM Conference on Information-Centric Networking*. 117–128.

[87] Vassilios G Vassilakis, Bashar A Alohali, I Moscholios, and Michael D Logothetis. 2015. Mitigating distributed denial-of-service attacks in named data networking. In *Proceedings of the 11th Advanced International Conference on Telecommunications (AICT), Brussels, Belgium*. 18–23.

[88] Matteo Virgilio, Guido Marchetto, and Riccardo Sisto. 2013. PIT overload analysis in content centric networks. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. 67–72.

[89] Ivan Voitalov, Rodrigo Aldecoa, Lan Wang, and Dmitri Krioukov. 2017. Geohyperbolic routing and addressing schemes. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 11–18.

[90] Satyanarayana Vusirikala, Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2016. Hop-by-hop best effort link layer reliability in named data networking. *NDN, Technical Report NDN-0041* (2016).
[91] Kai Wang, Dongchao Guo, and Wei Quan. 2019. Analyzing NDN NACK on Interest Flooding Attack via SIS Epidemic Model. *IEEE Systems Journal* (2019).
[92] Kai Wang, Yude Zhao, Xiangrong Tong, et al. 2017. On the urgency of implementing Interest NACK into CCN: from the perspective of countering advanced interest flooding attacks. *IET Networks* 7, 3 (2017), 136–140.
[93] Kai Wang, Huachun Zhou, Hongbin Luo, Jianfeng Guan, Yajuan Qin, and Hongke Zhang. 2014. Detecting and mitigating interest flooding attacks in content-centric network. *Security and Communication Networks* 7, 4 (2014), 685–699.
[94] Kai Wang, Huachun Zhou, Yajuan Qin, Jia Chen, and Hongke Zhang. 2013. Decoupling malicious interests from pending interest table to mitigate interest flooding attacks. In *2013 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 963–968.
[95] Kai Wang, Huachun Zhou, Yajuan Qin, and Hongke Zhang. 2014. Cooperative-Filter: countering Interest flooding attacks in named data networking. *Soft Computing* 18, 9 (2014), 1803–1813.
[96] Licheng Wang, Yun Pan, Mianxiong Dong, Yafang Yu, and Kun Wang. 2017. Economic levers for mitigating interest flooding attack in Named Data Networking. *Mathematical Problems in Engineering* 2017 (2017).
[97] Zhijun Wu, Wenzhi Feng, Meng Yue, Xinran Xu, and Liang Liu. 2020. Mitigation measures of collusive interest flooding attacks in named data networking. *Computers & Security* 97 (2020), 101971.
[98] Yonghui Xin, Yang Li, Wei Wang, Weiyuan Li, and Xin Chen. 2016. A novel interest flooding attacks detection and countermeasure scheme in NDN. In *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–7.
[99] Yonghui Xin, Yang Li, Wei Wang, Weiyuan Li, and Xin Chen. 2017. Detection of collusive interest flooding attacks in named data networking using wavelet analysis. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 557–562.
[100] George Xylomenos, Christopher N Ververidis, Vasilios A Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V Katsaros, and George C Polyzos. 2013. A survey of information-centric networking research. *IEEE communications surveys & tutorials* 16, 2 (2013), 1024–1049.
[101] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2013. A case for stateful forwarding plane. *Computer Communications* 36, 7 (2013), 779–791.
[102] Gubei Yin, Junhua Tang, Futai Zou, Yue Wu, and Jianhua Li. 2019. Controller Based Detection Scheme of Interest Flooding Attack in Named Data Networking. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 1628–1633.
[103] Yingdi Yu, Alexander Afanasyev, David Clark, KC Claffy, Van Jacobson, and Lixia Zhang. 2015. Schematizing trust in named data networking. In *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. 177–186.
[104] Haitao Zhang, Yanbiao Li, Zhiyi Zhang, Alexander Afanasyev, and Lixia Zhang. 2018. NDN host model. *ACM SIGCOMM Computer Communication Review* 48, 3 (2018), 35–41.
[105] Haitao Zhang, Zhehao Wang, Christopher Scherb, Claudio Marxer, Jeff Burke, Lixia Zhang, and Christian Tschudin. 2016. Sharing mhealth data via named data networking. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 142–147.
[106] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.
[107] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D Thornton, Diana K Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, Christos Papadopoulos, et al. 2010. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC* 157 (2010), 158.
[108] Meng Zhang, Hongbin Luo, and Hongke Zhang. 2015. A survey of caching mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials* 17, 3 (2015), 1473–1499.
[109] Xin Zhang and Ru Li. 2019. An ARI-HMM based Interest Flooding Attack countermeasure in NDN. In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 10–15.
[110] Zhiyi Zhang, Vishrant Vasavada, Eric Osterweil, Lixia Zhang, et al. 2019. Expect More from the Networking: DDoS Mitigation by FITT in Named Data Networking. *arXiv preprint arXiv:1902.09033* (2019).
[111] Zhiyi Zhang, Yingdi Yu, Alex Afanasyev, and Lixia Zhang. 2017. NDN certificate management protocol (NDNCERT). *NDN, Technical Report NDN-0050* (2017).
[112] Zhiyi Zhang, Yingdi Yu, Sanjeev Kaushik Ramani, Alex Afanasyev, and Lixia Zhang. 2018. NAC: Automating access control via Named Data. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 626–633.
[113] Zhiyi Zhang, Yingdi Yu, Haitao Zhang, Eric Newberry, Spyridon Mastorakis, Yanbiao Li, Alexander Afanasyev, and Lixia Zhang. 2018. An overview of security support in named data networking. *IEEE Communications Magazine* 56, 11 (2018), 62–68.
[114] Ting Zhi, Ying Liu, Jiushuang Wang, and Hongke Zhang. 2019. Resist Interest Flooding Attacks via Entropy–SVM and Jensen–Shannon Divergence in Information-Centric Networking. *IEEE Systems Journal* (2019).
[115] Ting Zhi, Ying Liu, and Jun Wu. 2020. A Reputation Value-Based Early Detection Mechanism Against the Consumer-Provider Collusive Attack in Information-Centric IoT. *IEEE Access* 8 (2020), 38262–38275.

[116] Ting Zhi, Hongbin Luo, and Ying Liu. 2018. A Gini impurity-based interest flooding attack defence mechanism in NDN. *IEEE Communications Letters* 22, 3 (2018), 538–541.