AMAR TELIDJI UNIVERSITY OF LAGHOUAT
DEPARTMENT OF MATHEMATICS AND INFORMATICS

INTEREST FLOODING ATTACKS IN NAMED DATA NETWORKING

A DISSERTATION PRESENTED

by

AHMED BENMOUSSA

In partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in Computer Science

Jury members

Mr. Mohamed Bachir Yagoubi	Professor	UATL	President
Mr. Amine Khaldi	M.C.A	UKMO	Examiner
Mr. Akram Boukhamla	M.C.A	UKMO	Examiner
Mr. Noureddine Chaib	M.C.A	UATL	Examiner
Mr. Abdou El Karim Tahari	M.C.A	UATL	Advisor
Mr. Chaker Abdelaziz Kerrache	M.C.A	UATL	Co-Advisor

2021

To my parents, wife, brother, and sisters.

Acknowledgement

I acknowledge the efforts of my supervisors and worldwide collaborators.

ملخص

تعتبر شبكات البيانات المسماة قفزة نوعية لبنية شبكة الأنترنت. حيث تعتمد أساسا على أسماء المعطيات أثناء توجيه الحزم الشبكية بخلاف الشبكة الحالية التي تعتمد على عناوين الخوادم. ضف الى ذلك تعتمد هذه البنية الجديدة على نظام أمن معلوماتي قائم أساسا على حماية البيانات عوض القنوات الشبكية الناقلة لها. وجب الإشارة أنه بالرغم من دمج شبكات البيانات المسماة لنظام أمن معلوماتي الا أنها لا تزال عرضة لبعض الهجمات. أحد أبرز الهجمات التي تستهدفها هي هجوم الإغراق بالطلبات التي نحن بصدد مناقشتها في أطروحتنا. في هذا السياق يأتي أول اقتراح لنا ضد هجمات الإغراق بالطلبات حل قادر على التفريق بين الحالات الناجمة عن الهجمات و الأخرى الناجمة عن الازدحام على مستوى القنوات الشبكية. كما اقترحنا حل ثان له القدرة على كشف و ايقاف هجمات الإغراق بالطلبات العادية ليس هذا فقط بل و ايقاف سيناريوهات هجمات أخرى و هذا دون المساس بطلبات المستخدمين المشروعة. نجاعة حلولنا أثبتت عمليا بمحاكاة مكثفة في ظل العديد من الهجمات المفتعلة. أما مساهمتنا الثالثة في هاته الأطروحة فهي عبارة عن مجموعة من هجمات الإغراق بالطلبات الغير مسبوقة و التي لم تؤخذ بعين الإعتبار من قبل الحلول الموجودة. في نهاية المطاف وقبل أن نختم نقوم بعرض المشاكل التي تواجهها الحلول المقترحة سابقا قبل التطرق الى المتطلبات التي يجب أخذها بعين الإعتبار أثناء تصميم حل ضد هجوم الإغراق بالطلبات.

الكلمات الأساسية: شبكات البيانات المسماة؛ بنية انترنت المستقبل؛ أمن شبكات البيانات المسماة؛ هجمات الحرمان من الخدمة؛ هجوم الاغراق بالطلبات.

Abstract

Named Data Networking (NDN) is a newly proposed Future Internet Architecture (FIA). 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. In this dissertation, we focus on the Interest Flooding Attack and present our contributions to deal with this threat. Our first contribution is a novel congestion-aware solution that differentiates between intentional and unintentional misbehavior due to the network congestion when detecting Interest Flooding Attack. We prove, through simulation evaluation, the efficiency of our solution in congested network scenarios. Our second contribution, named MSIDN, is a cooperative solution that detects and mitigates sophisticated versions of IFA. The proposed solution is capable of stopping the malicious traffic without affecting legitimate users. Through extensive simulation evaluation, we show the efficiency of our solution in different attacking scenarios. Our third contribution is a series of unconsidered adversarial models. We present a multitude of non-conventional IFA scenarios that the actual solutions did not take into consideration. we finish our work by reviewing the requirements for a complete and efficient IFA solution and pinpoint the various issues encountered by IFA detection and mitigation mechanisms before shedding the light on the future research directions regarding IFAs.

Keywords: Named Data Networking (NDN); Future Internet Architecture (FIA); NDN Security; Denial of Service (D)DoS Attacks; Interest Flooding Attack (IFA).

Résumé

Named Data Networking (NDN), ou le réseau des données nommées, est un nouveau type de réseau proposée pour le futur Internet. L'architecture NDN focalise sur la donnée et non pas son emplacement. Contrairement au réseau Internet actuel, l'architecture NDN se base sur le nom de la donnée pour l'acheminement des paquets et non pas les adresses IP. NDN intègre des fonctionnalités de sécurité qui permettent de sécuriser directement la donnée au lieu de sécuriser le canal acheminant la donnée. Toutefois, l'architecture NDN fait face à de nouvelles menaces de sécurité, notamment un nouveau type d'attaque appelé Interest Flooding Attack (IFA), ou l'attaque par inondation de requêtes. Cette thèse a pour objet d'aborder et d'étudier ce nouveau type d'attaque et essayer d'apporter des solutions efficaces pour palier à cette nouvelle menace qu'est l'Interest Flooding Attack. Notre première contribution est une solution capable de détecter et stopper une attaque. Cette solution prend en considération la congestion du réseau comme paramètres afin d'éviter une fausse détection. La phase d'évaluation a montré l'efficacité de notre solution avec un scénario ou le réseau souffre d'une congestion. Notre deuxième solution, MSIDN, est une solution coopérative capable de détecter et de stopper des attaque sophistiqués, et ce sans affecter le trafic des nœuds légitimes. La simulation a prouvé l'efficacité de notre solution dans divers scenarios d'attaque. Et comme troisième contribution pour cette thèse, nous avons présenté des scénarios d'attaques pas encore considérés par les solutions existantes. Nous concluons cette thèse par évoquer les différents problèmes rencontrés par les solutions existantes avant de parler des exigences à prendre en considération quand il s'agit de concevoir une solution efficace. Nous finissons par présenter les futures orientations de recherche concernant IFA.

Mots clés: Réseaux des données nommées; Architecture future d'Internet; Sécurité NDN; Attaques par déni de service; Attaque par inondation de paquets Interest.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
AQM	**A**ctive **Q**ueue **M**anagement
ARI	**A**utoregressive **I**ntegrated
AS	**A**utonomous **S**ystem
CCN	**C**ontent-**C**entric **N**etworking
CDA	**C**ollect **D**etect **A**ct
COMET	**CO**ntent **M**ediator architecture for content-aware n**ET**work
CS	**C**ontent **S**tore
DC	**D**omain **C**ontroller
DDoS	**D**istributed **D**enial **o**f **S**ervice
DONA	**D**ata-**O**riented **N**etwork **A**rchitecture
FIA	**F**uture **I**nternet **A**rchitecture
FIB	**F**orwarding **I**nformation **B**ase
FIFO	**F**irst **I**n **F**irst **O**ut
HMM	**H**idden **M**arkov **M**odel
IFA	**I**nterest **F**looding **A**ttack
ipps	interest packet per second
LFU	**L**east **F**requently **U**sed
ICN	**I**nformation-**C**entric **N**etworking
LRU	**L**east **R**ecently **U**sed
MANET	**M**obile **A**d-hoc **Net**work
MLP	**M**ultilayer **P**erceptron
NDN	**N**amed **D**ata **N**etworking
ndnSIM	NDN Simulator
NFD	**N**DN **F**orwarding **D**aemon
NN	**N**eural **N**etwork
PIT	**P**ending **I**nterest **T**able
PURSUIT	**P**ublish **S**ubscribe **I**nternet **T**echnology
QCF	**Q**uotient-based **C**uckoo **F**ilter
RBF	**R**adial **B**asis **F**unction
SVM	**S**upport **V**ector **M**achine
TLS	**T**ransport **L**ayer **S**ecurity
XIA	**X**tensible **I**nternet **A**rchitecture

Chapter 1

Introduction

1.1 Motivation

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 [Cis], which represents roughly three times the global population, the actual Internet architecture was not designed for such massive numbers of hosts. In addition, the 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. This opened the door to several 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 [Rfc]. Since then, many Future Internet Architectures (FIAs) were proposed like Xtensible Internet Architecture (XIA) [Ana+11], Nebula [And+13], and others. However, the most promising FIA architecture candidate is Information Centric Networking (ICN) [Ahl+12; Xyl+13]. Several architectures were proposed under the umbrella of ICN, like Data-Oriented (and beyond) Network Architecture (DONA) [Kop+07], Publish-Subscribe Internet Technology (PURSUIT), Scalable & Adaptive Internet soLutions (SAIL) [ET11], and COntent Mediator architecture for content-aware nETworks (COMET) [Gar+11]. But the most promising one is Named Data Networking (NDN). NDN is a project funded in 2010 by the US Future Internet Architecture and maintained at UCLA [Zha+10]. NDN takes its roots from the Content-Centric Networking (CCN) [Jac+07].

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. 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 dissertation aims to address IFA and provide efficient solutions to this attack.

1.2 Contributions of this Thesis

The contributions of this thesis can be summarized as follows:

- Provides an in-depth study of IFA and gives a broad analysis and comparison of the solutions that were proposed.
- Design and evaluation of a congestion-aware IFA mitigation solution.
- Design and evaluation of MSIDN, a cooperative solution capable of mitigating sophisticated IFA.
- Identify and analyse several unfaced IFA attacking scenarios against the present literature.
- Pointing out the open issues and providing the challenges and research directions that need to be considered in the future.

1.3 Organization of the Thesis

This thesis is organized as follows: in chapter 2, we introduce the NDN architecture and present its components. We continue by explaining the forwarding process. Following that, we show the security components and we conclude this chapter by detailing the security requirements. Chapter 3 focuses on the Interest Flooding Attack. We start this chapter by talking about the availability threats that target NDN networks. After that, we detail IFA and its variants. We continue this chapter by talking about the related work, in which we present all the relevant works that were authored in the literature. We conclude this chapter by presenting, explaining, and detailing the CDA workflow that every IFA detection and mitigation solution follows.

We present our first contribution in chapter 4. This solution takes into account the network congestion when dealing with IFA to avoid false detection. We start our simulation evaluation by showing how a congested network could mistakenly lead to false detection. We finish our evaluation by proving the efficiency of our proposed solution.

Our second solution is presented in chapter 5. In addition to the conventional IFA, this cooperative solution aims also to detect a sophisticated version of IFA. The proposed solution is capable of mitigating both local and distributed attacks without affecting legitimate users.

We present our third and last contribution in chapter 6, in which we introduce several unfaced IFA attacking scenarios against existing solutions. In this chapter, we show and explain several attacking behaviours that malicious nodes could adopt to counter the existing solutions. Finally, we summarize this thesis in chapter 7, before pointing out the challenges and future research directions that need to be considered.

Chapter 2

Named Data Networking (NDN): Architecture and design overview

NDN is a data-centric Internet architecture designed to replace the host-centric TCP/IP architecture [Jac+09; Zha+10; Zha+14]. 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.

2.1 NDN Protocol Stack

The NDN protocol stack is composed of four different layers: application, network, link, and physical layers [Zha+18a]. The application layer supports the operation of NDN applications. It also embeds transport protocols as system libraries. The role of the network layer is to route NDN packets. It uses the application layer's names to route the packets. The NDN link-layer supports a set of protocols like Ethernet. It can also use virtual links like IP and TCP overlays. Figure 2.1 shows the differences between the TCP/IP and NDN protocol stacks.

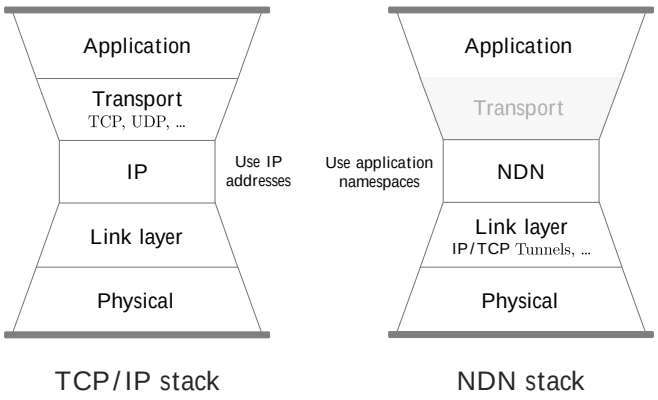

FIGURE 2.1: TCP/IP and NDN Protocol Stacks

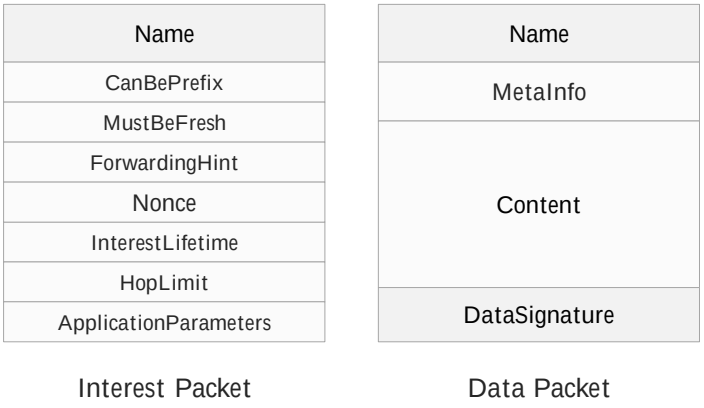

FIGURE 2.2: NDN packets

2.2 NDN Packet Types

The NDN architecture uses two types of packets: *Interest* and *Data*[1] packets. Figure 2.2 illustrates these two NDN packets.

2.2.1 The Interest Packet

NDN consumers use Interest packets to express a need for specific data. To fetch content, data consumers send the name of the desired content in an Interest packet. According to the latest NDN packet specifications [Ndn], every Interest packet is composed of a set of mandatory and optional parameters. Each interest packet must have a *Name*. It represents the name of the content that the consumer is requesting. The *Nonce* field is also mandatory when the Interest packet is transmitted over the network. It consists of a randomly generated 4octets long string. The *Nonce* is used to uniquely identify Interest packets, hence preventing them from looping in the network. The optional parameters that Interest packets may include are: *CanBePrefix* for the Interest packet's name, *MustBeFresh* for the content of the requested Data packet, *InterestLifeTime* represents for how long an NDN router will maintain state for this Interest, *HopLimit* and *ForwardingHint* are used during the forwarding process. The Interest packet may also include application-specific parameters in *ApplicationParameters*. Furthermore, Interest packets can also be signed when needed [Sig].

2.2.2 The Data Packet

Data packets are the response that NDN consumers expect when they send Interest packets. Data packets carry the content requested by a consumer. Each Data packet contains the following fields: the *Name* of the data, the payload of the Data packet held in the *Content* field, and the *DataSignature*, which contains the signature of the producer. Data packets may also contain some optional information in the *MetaInfo*

[1] In this dissertation, we use upper-case "Data" to refer to the NDN's Data packet.

field: The *ContentType*, the *FreshnessPeriod*, which indicates how long the data can stay fresh, i.e., the producer did not produce newer data. The last parameter is the *FinalBlockId* which identifies the last packet of a sequence of fragments [Dat]. Data packets are unique and immutable, i.e., they cannot be modified after being produced.

2.3 Naming Content

NDN adopts a hierarchical naming scheme [Afa+17]. It identifies content with application names. Every Data packet possesses a unique name. It consists of a sequence of components hierarchically ordered. For example, this thesis could be named `/dz/lagh-univ/thesis/phd/Ahmed_Benmoussa`.

2.4 NDN Node Model

Each NDN node maintains three data structures: The *Pending Interest Table (PIT)*, the *Content Store (CS)*, and the *Forwarding Information Base (FIB)*. In this section we detail each component and point out its role.

2.4.1 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.

2.4.2 Content Store (CS)

Each NDN node can store passing Data packets in a local cache [Cao+16]. It enables NDN to offer in-network caching. Requests can be satisfied by an intermediate cache without going down to the source of data. It also reduces time retrieval and saves link bandwidth. Each CS needs 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) [ZLZ15; Din+17].

2.4.3 Forwarding Information Base (FIB)

FIB is used to forward incoming Interest packets to upstream nodes. Unlike IP networks, NDN indexes FIB entries with name prefixes instead of IP addresses [Son+15]. 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 [Cha+17; Sch+16; Mas+20; MM20].

2.5 Routing and Forwarding

NDN routers use application namespace instead of IP addresses to forward packets [Li+18]. Routers update and announce their FIB entries using routing algorithms [Leh+16; Voi+17; Gha+18], or a self-learning mechanism [SNZ17]. Unlike IP routers, NDN routers use stateful forwarding, i.e., routers keep information about the received requests until they are satisfied or timed-out [Yi+13]. The forwarding strategy forwards Interest packets according to the FIB entries, local measurements, or other per-namespace forwarding policies [Shi17]. 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. The NDN forwarding process is illustrated in Figure 2.3.

2.5.1 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 requested data's name with the existing content names in the CS. If the requested content already exists in the local CS, the router sends back the Data packet to the source interface(s), i.e., the sourcing interface(s) of the Interest packet.

If the CS does not own the requested data, the router performs the following actions: first, it checks its PIT for any similar pending request, i.e., the router checks if the name of the Interest packet already exists in the PIT. If the router finds a matching entry, it compares the source interface with the interfaces associated with this pending entry. If the source interface is already recorded, the incoming Interest packet is considered 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 is 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 [AC17; Vus+16].

2.5.2 Data packet forwarding: Receiving data

The consumer receives a Data packet when its request is satisfied by a data producer or in-network cache. When a router receives a Data packet, it checks if it was requested before. The router verifies the existence of a pending Interest with the name

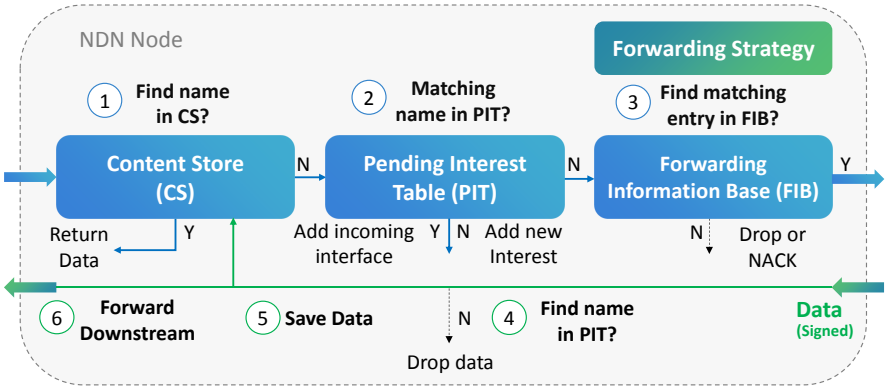

FIGURE 2.3: Interest and data packet forwarding process

of the received Data. If no match exists in the PIT, the router drops the received Data packet. Otherwise, and according to its caching strategy, the NDN router stores (on not) the received Data packet 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.

2.6 Named Data Networking vs TCP/IP

In this section, we introduce a comparison between NDN and TCP/IP in terms of routing and forwarding, multicast, mobility, and security. Table 2.1 summarizes the differences between these two architectures.

2.6.1 Routing and forwarding

Using names instead of IP addresses comes with some advantages: first, compared to IP addresses, names are unbound and have no limit. Second, address translation (NAT) is no longer needed. Third, there is no need to assign and manage IP addresses in local networks. Moreover, NDN networks are immune against packet looping so nodes can take full advantage of multipath forwarding. Figure 2.4 depicts the differences in packet forwarding between NDN and TCP/IP.

2.6.2 Multicast

IP networks support multicast, but setting up multicast groups in IP networks is challenging. Members need to know the group address to join it. Also, the number of nodes in a multicast group is limited. Moreover, multicast groups are not easy to deploy in large networks like the Internet. The NDN architecture comes with a built-in multicast forwarding mechanism with no pre-configuration required. NDN routers aggregate source interfaces requesting the same content in PIT. It allows routers to send the data packet to all requesting interfaces at once, hence performing multicast forwarding [Mas+17].

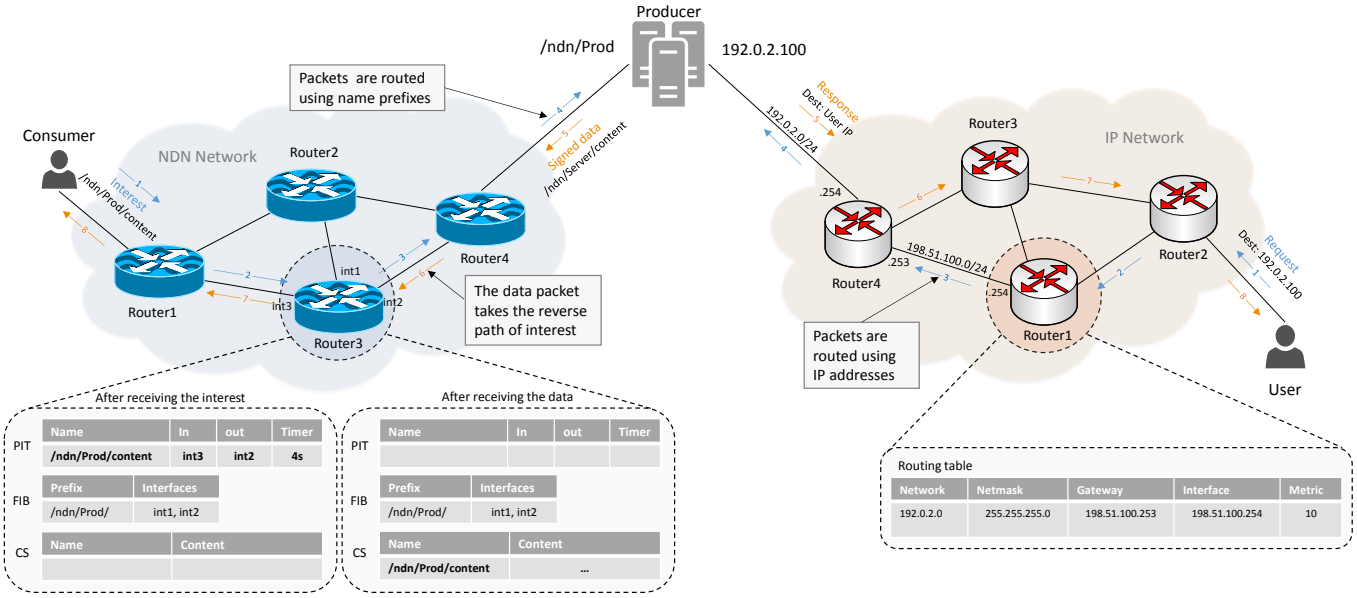

FIGURE 2.4: NDN vs TCP/IP packet forwarding

2.6.3 Mobility

Mobility is another field that NDN handles better than IP networks [Zha+16b; Zha+18b]. The constantly changing network topology and paths make packet routing challenging in mobile networks. The data-centric approach that NDN adopts is beneficial to mobile networks like MANETs [MLZ20; Li+19a]. There is no need to maintain and manage nodes' IP addresses. The cache-store that network nodes offer reduces the delivery time and gives alternative paths to requested data even if the source is not reachable, i.e., data producers.

2.6.4 Security

IP networks rely on communication protocols like TLS [DR08] and IPSec [KF11] to securely send data packets. On the other hand, NDN networks focalize on securing the network's most valuable asset, the data. Unlike IP networks, NDN secures the data instead of securing its path. By choosing this security approach, NDN networks relieve the nodes from maintaining and monitoring secure channels and allow data producers to only focus on securing the data that they produce [Zha+18c].

2.7 NDN Security: Data-centric Security

The NDN design natively embeds security mechanisms and mandates the use of cryptographic signatures in Data packets [Ram+19]. Producers need to digitally sign every Data packet produced. It will enable consumers to verify and validate the authenticity of the received Data packets. It also permits network nodes, i.e.,

TABLE 2.1: Comparison between NDN and TCP/IP

Propriety		**NDN**	**TCP/IP**
Architecture		Data-centric	Host-centric
Identification		Names	IP addresses
Forwarding		Stateful forwarding	Stateless forwarding
Forwarding paths	Request	Multipath forwarding. The request packet could be forwarding through multiple faces	Single path forwarding
	Response	The response packet takes the reverse path of the request	The response packet does not necessarily take the request's path
Caching	Intermediate nodes	All NDN routers offer in-network caching	Routers do not offer data caches
	End nodes	Every NDN node can offer a data cache	Only specific nodes like file servers and P2P nodes
Multicast		Built-in multicast mechanism	Uses multicast address range
Mobility		The NDN nodes are independent of their location	The nodes IP addresses are location dependent.
Security		Secures data packets	Secures communication channels using protocols

routers and repositories, to store Data [Psa+18]. Furthermore, it allows consumers to retrieve and accept Data packets regardless of their source [NZ19].

The NDN architecture integrates some security components to ensure Data security [Zha+18c]. Figure 2.5 illustrates the below detailed security components.

2.7.1 Packet signature

Every Data packet includes a signature field [Zha+16a]. The signature binds the content of the packet to its name [Li+19b]. The signature field contains two components: *SignatureInfo* and *SignatureValue*. The *SignatureInfo* component consists of a *SignatureType*, which represents the cryptographic algorithm used to sign the Data packet, and a *KeyLocator*, which refers to the producer's public key that consumers need to use to authenticate and validate the received Data packet. The *SignatureValue* represents the bits of the generated signature. Although 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 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.

2.7.2 Trust Schema and Access Control

Although that Data packet's signature allows the consumers to validate the received Data and prove its originator, it does not show whether the signer is authorized to produce the Data or not [Yu+15]. Consumers need a mechanism that allows them

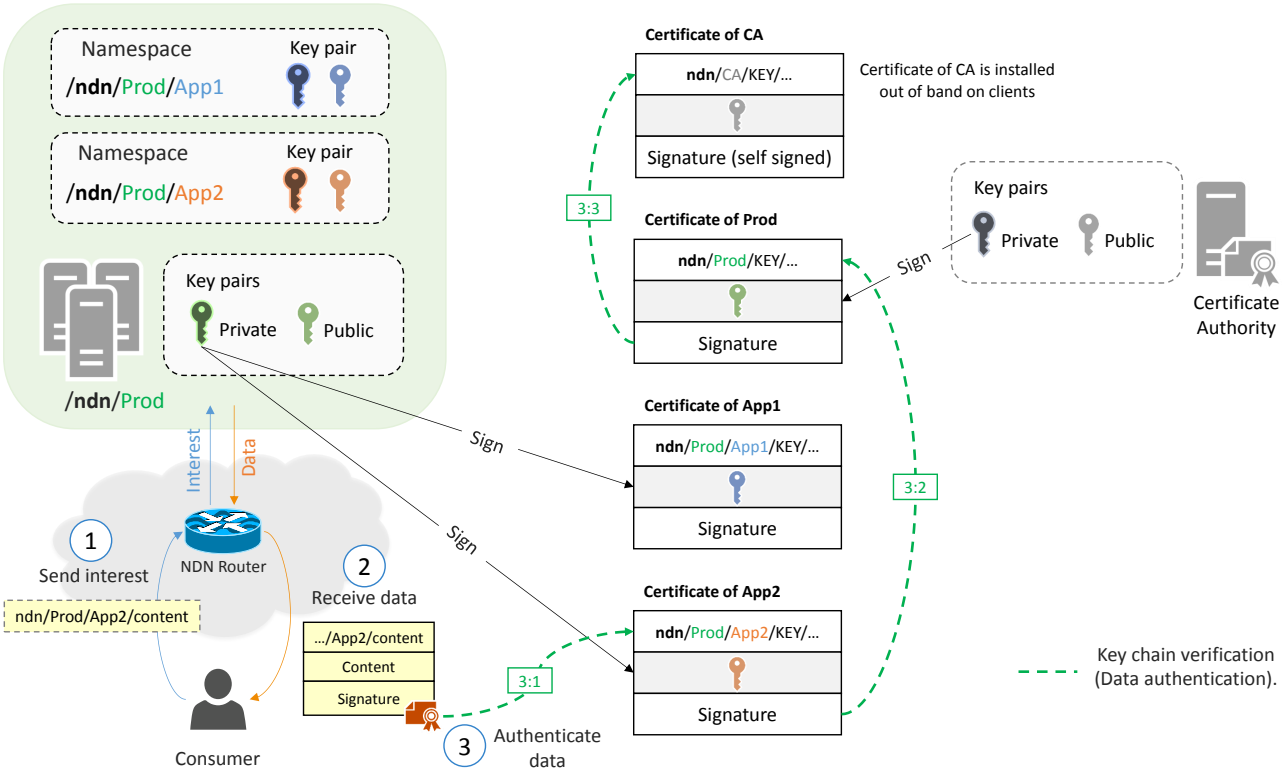

FIGURE 2.5: NDN security components

to check if a producer has the right to generate data under a given namespace, i.e., if a producer's key has the right to sign a Data packet under a given namespace [Sha+17]. Application-based trust policies are used to define which keys are authorized to sign which Data packets. 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 [Zha+18d; Lee+18].

2.7.3 Key pairs and Certificates

Every data producer needs 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 [Afa+16a]. It contains the producer's public key encoded in X509 format [Zha+17]. 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, 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. Every NDN certificate includes a signature field that contains the signature of the issuer.

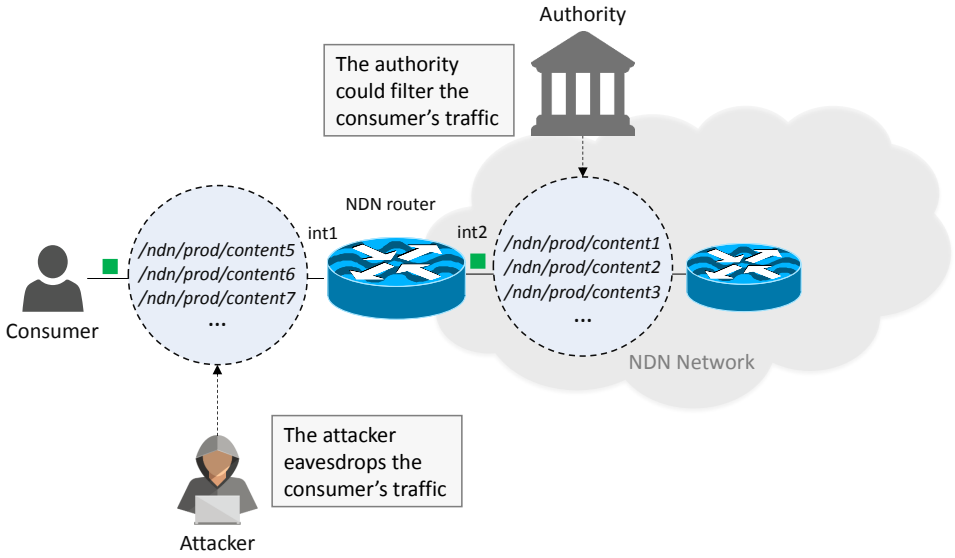

FIGURE 2.6: Example of a privacy attack

2.8 NDN Security Requirements

The following section details the security requirements of NDN networks.

2.8.1 Confidentiality

Information confidentiality is essential to any security system. Network entities should restrict any unauthorized actors from accessing any confidential resources. NDN opens new challenges for data confidentiality. Data packets are more susceptible to breaches because of in-network caching. NDN nodes need to use encryption and implement strict access control schemes to ensure data confidentiality and prevent data leakage [Sha+15].

2.8.2 Privacy

Privacy aims to prevent unauthorized actors from accessing private, personal, or confidential information which could be used to identify and track nodes or individuals. NDN uses application namespaces instead of IP address to route packets in the network layer. This feature opens new challenges to privacy and anonymity. Users' traffic could easily be monitored and filtered, as showed in Fig. 2.6.

2.8.3 Authentication

The process of verifying the identity of a node is called authentication. In scenarios where nodes' identity needs verification, authentication is essential. It represents any additional information that can identify the authorized nodes, like passwords

and certificates. NDN mandates Data packet signatures, which implies the authentication of every data packet before consumption.

2.8.4 Integrity

Data integrity prevents authorized and unauthorized nodes from making illegitimate modifications. The NDN architecture mandates the use of cryptographic signatures in every Data packet produced. It enables a native data integrity mechanism in NDN. When the content of a Data packet changes, consumers or network routers can easily detect it. However, this mechanism cannot provide immunity from threats. The data producer can suffer key breaches. The content of Data could also be altered before being signed by the producer.

2.8.5 Non-repudiation

Non-repudiation ensures that an entity cannot deny the production of a packet or its content. The Data packet's signature guarantees a non-repudiation mechanism in NDN. A data producer cannot reject the ownership of a Data packet because it contains its signature. However, the private key of a producer could be stolen by a hacker or accidentally leaked because of human error. The certification authority could also be compromised. Additionally, Interest packets are not signed by default. All these challenges need to be considered when implementing a non-repudiation mechanism.

2.8.6 Availability

System availability consists of ensuring operative and uninterrupted access to authorized nodes. Multipath forwarding and in-network caching are two solid assets that NDN offers to guarantee a high availability level [Yeh+14]. However, NDN Adversary nodes can launch (D)DoS attacks to affect and disturb the system's availability. In NDN networks, routers and producers' resources are potentials targets to (D)DoS attacks [Gas+13].

2.9 Summary

In this chapter, we presented an overview of the NDN architecture and its components. Following that, we discussed the NDN security requirements. In the next chapter, we will briefly discuss the availability attacks that target NDN before detailing the Interest Flooding Attack (IFA).

Chapter 3

Interest Flooding Attack in NDN: Definition, taxonomy, and related work

The present chapter focuses on IFA and provides an in-depth study of this attack. But before diving into IFA, we first present the availability threats that target NDN networks. Availability in NDN is susceptible to various types of attacks. We classify the availability attacks into two categories: first, availability attacks against routers, which groups the different availability attacks that target NDN routers components. Second, availability attacks against producers. Figure 3.1 illustrates the availability attacks in NDN.

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.

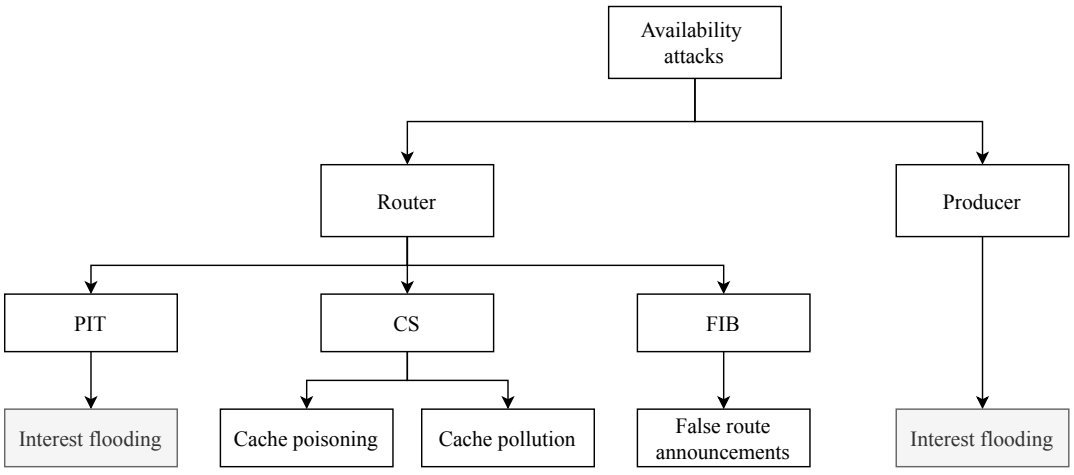

FIGURE 3.1: Availability attacks in NDN

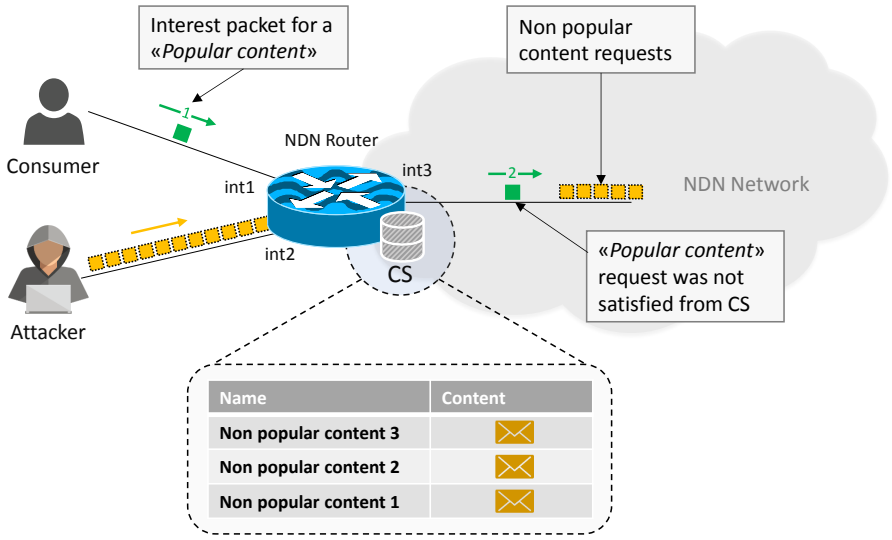

FIGURE 3.2: Example of a cache pollution attack

3.1.1 Availability attacks against the Content Store

Cache pollution attack

NDN nodes use the CS to cache the frequently demanded data, i.e., the most popular content, to reduce time retrieval and network traffic. 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 3.2 shows a scenario of cache pollution attack.

Cache poisoning attack

Another CS-based attack is the cache poisoning attack. The main goal of the attack is to fill the nodes' caches with invalid data. Attackers try to inject Data packets with valid names but altered or malicious content. The cache poisoning attack is slightly difficult to perform. The packet's signature binds the name of the packet with its content, so any changes result in an invalid Data packet. Additionally, the attacker has to control network routers, or perform a man-in-the-middle attack, to inject fake Data packets. Figure 3.3 presents an example of a cache poisoning attack.

3.1.2 Availability attacks against the FIB

False route announcements

The purpose of this attack is to modify the FIB of a target. The adversary tries to add new routes or update existing ones by sending announcement packets containing false routes as shown in Fig. 3.4. The attacker usually uses false route announcements to perform other attacks that target availability like black-holing or privacy like packet interception.

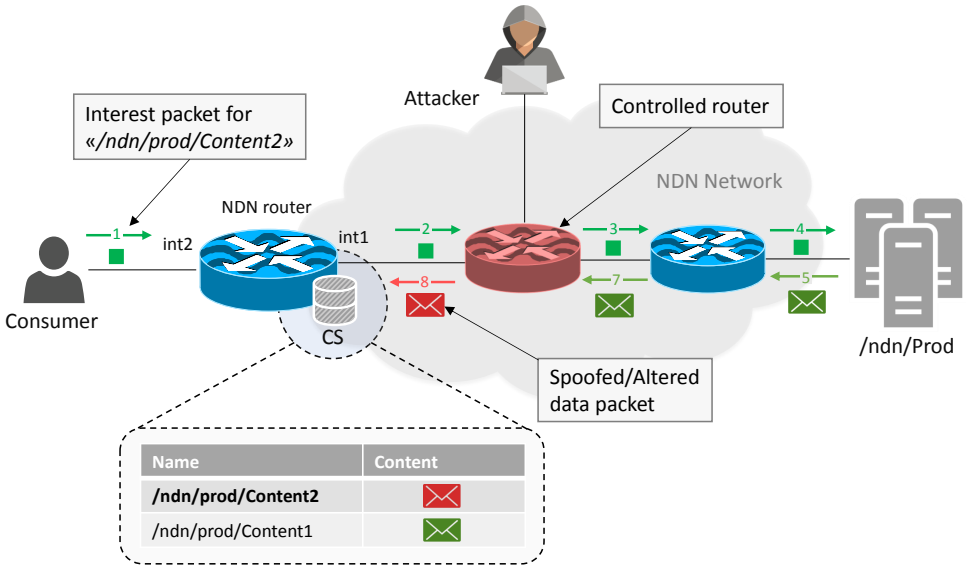

FIGURE 3.3: Example of a cache poisoning attack

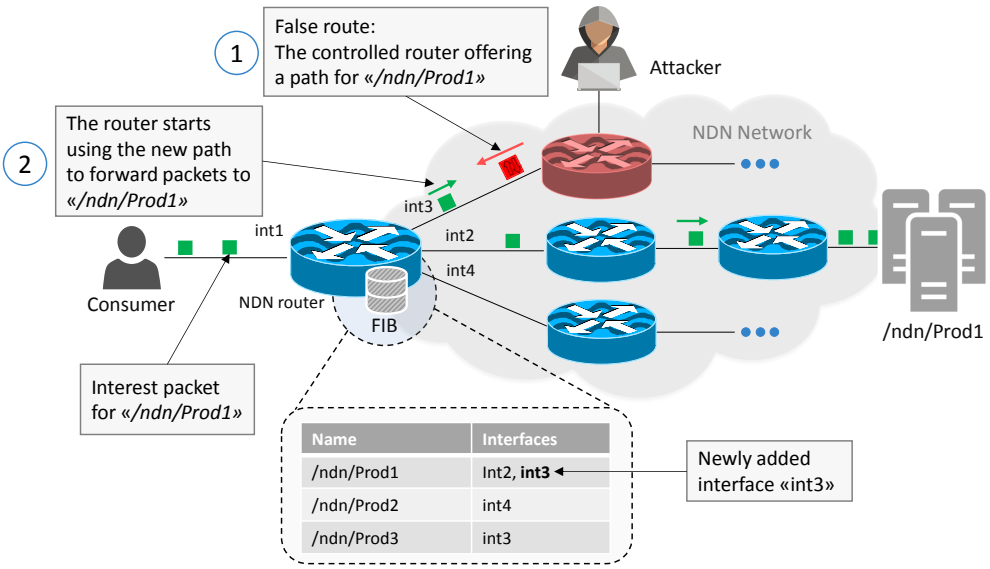

FIGURE 3.4: Example of a false route announcement attack

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 a large number of Interest packets, so it cannot accept legitimate requests. We will discuss IFA in detail in the next section.

3.1.4 Availability attacks against Producers

Producer-based availability attacks are mainly associated with IFA. Attackers could target one or multiple producers by sending a large number of Interest packets towards them. Malicious nodes can lunch attacks against producers using all types of Interest packets mentioned in section 3.3. 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 resource damages due to memory and processing overhead.

3.2 Interest Flooding Attack (IFA)

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 Fig. 3.5. The main goal for attacking routers is to fill 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. It includes memory exhaustion, service overload, or any other hardware-related overhead. IFA 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.

3.3 Taxonomy of IFA Requests

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

3.3.1 Valid data requests

The first type of requests that attackers could use to attack a target with IFA is valid data requests. There are two classes of valid data requests: requesting static data or requesting dynamic data.

Requesting static data

The first type of valid requests that attackers could use to perform IFA is using Interest packets with valid names. Like any other legitimate Interest packet, these

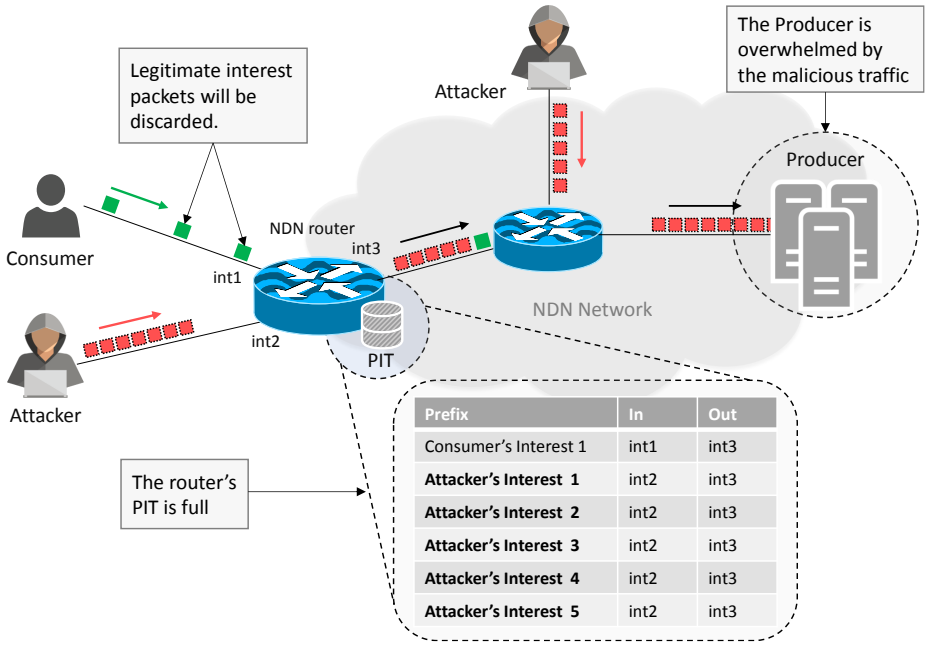

FIGURE 3.5: Example of an Interest Flooding Attack

requests are satisfiable by data producers or network caches. This attacking scenario consists of sending an enormous number of legitimate Interest packets 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 severe damage to the network, especially in cases of a large-scale attack.

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 created when a request arrives. It can range from the current record of a sensor in an engine to the actual stock price values. As their content is time-dependent, dynamic data are usually not stored in caches. Attackers can take advantage of this aspect to lunch IFA against producers. Dynamic data requests are likely to be routed to their producers, which may cause overhead to producers as they use hardware resources to process and sign the content before sending it back.

3.3.2 Invalid data requests

An invalid data request corresponds to any Interest packet with an invalid name, i.e., a forged Interest packet. Attackers can lunch IFA using fake Interest packets.

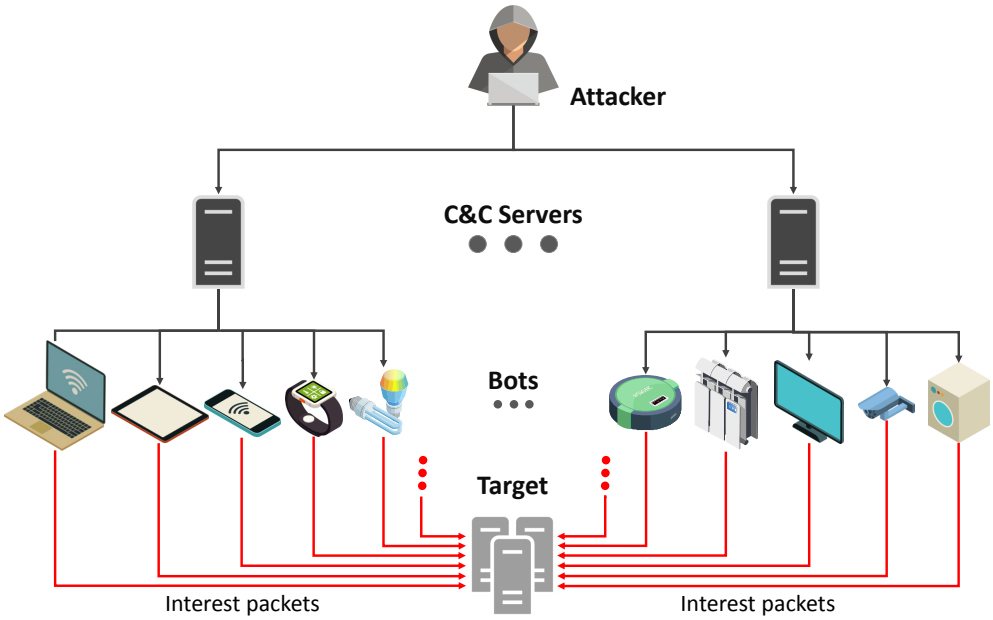

FIGURE 3.6: Example of a botnet

There are two ways of making non-valid Interest packet names: the first method is to choose a completely random name for the Interest packet. As their names are fake, the Interest packets will not reach any producer and affect only network routers. Nevertheless, this 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. It will ensure that the forged Interest packet will reach the targeted producer. For example, if an adversary node wants to target the producer `/Prod`, the content names that it may use are similar to `/Prod/nonce`, where nonce is a random value. This forged request will affect the producers and all the network routers traversed. IFA with invalid data requests is more harmful to the network than an IFA with valid data requests. Routers will not aggregate the received Interest packets and will keep them in PIT until they time out, which fills up their PIT quickly.

3.4 Local and Distributed Interest Flooding Attack

IFA is considered local when one or a small group of locally connected attackers perpetuates them. A distributed IFA occurs when the victim suffers 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, as showed in fig. 3.6. Distributed IFA are hard to stop and present a significant 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.

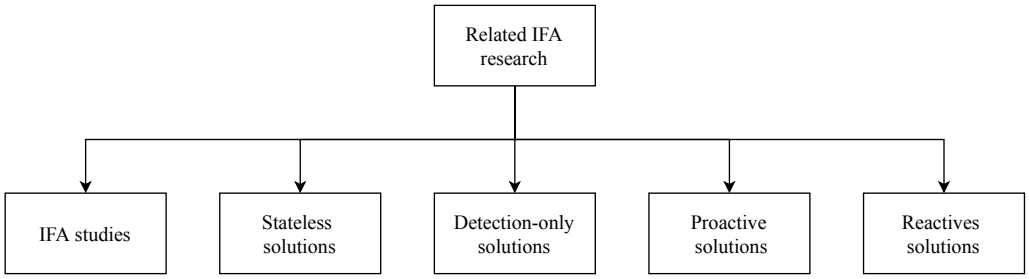

FIGURE 3.7: IFA related research classification

3.5 Other IFA variants

One of the IFA variants is the *"Collusive IFA"*. In this type of IFA, attackers work along with a malicious producer. The goal of this attack is that the malicious producer delays sending back data packets, so pending Interest packets stay longer in routers' PIT and hence occupying their memory. This type of IFA is harder to detect as the malicious producer satisfies attackers' Interest packets.

In another variant of IFA, attackers use a mix of data requests types [Ben+20b]. Attacker lunch attacks by sending Interest packets with valid and invalid names. This type makes it harder for routers to detect them.

3.6 Related Work

IFA was first mentioned in [Jac+09]. The authors of this paper talked about how consumers could overwhelm the network by sending many Interest packets. Following that, other papers studied deeper the IFA problem [WSV13; Gas+13; Afa+13].

Many IFA related papers are present in the literature. In this section, we categorize the IFA related research into five categories. First, IFA studies, where are mentioned the conducted studies about IFA. Second, the stateless solutions category mentions the solutions that took the stateless architecture approach to deal with IFA. The third category is the detection-only solutions category. It mentions existing detection-only IFA mechanisms. And finally, proactive and reactive solutions are grouped in the proactive solutions and reactive solutions categories, respectively. Figure 3.7 illustrates this classification.

3.6.1 IFA Studies

Several papers studied the IFA phenomenon and its impacts on the network. The authors of [Gas+13] first discussed how the NDN paradigm could help preventing some actual IP-related attacks. Then the authors discussed the Interest flooding and cache poisoning attacks before giving some tentative countermeasures. The authors of [Cho+13] conducted a study on the effects of IFA on the network. The authors showed through a series of simulation tests the impacts that IFA could have on the

throughput and the delay. The authors in [VMS13] provided a study about the requirements of an IFA resilient PIT architecture. The authors evaluated three PIT architecture: the default PIT, a hashed names based PIT, and a Bloom filter-based PIT. The authors compared these three architectures according to their memory consumption. Similarly, in [SNO13], the authors presented a hash-based design to reduce memory occupancy.

In [Com+15], the authors discussed the benefits of using network based and application based NACKs. The paper presented the security requirements for implementing secure NACK packets. It showed how producers could use secure application-based NACK packets to help to mitigate IFA. Following that, the authors pointed out the issues that could result from using these NACKs, before explaining how attackers could use them to overwhelm the network. Similarly, a study on the benefits of using NACK was conducted in [WZT+17]. The authors argued that NACK packets help routers from keeping pending Interests in PIT until they time-out. Another study about the benefits of network-based NACK packet was presented in [WGQ19]. The authors suggested to implement them to help mitigating IFA. They defended their proposal by stating that network-based NACK packets reduce the number of pending PIT entries before time out.

In [Sig+17], the authors evaluated three existing collaborative solutions against two variants of IFA: The first uses different name prefixes. The second uses a mix of valid and invalid Interest names. Similarly, in [ASWS15], the authors compared five different mitigation techniques according to their satisfaction ratio. In another context, The authors of [KSS19] conducted a comparison study between several machine learning algorithms for IFA detection: Naïve Bayes, J48 decision tree, multilayer perceptron (MLP) with back propagation, and the radial basis function network (RBF). The results showed that the J48 decision tree gave the most accurate results in detecting IFA.

3.6.2 Stateless solutions

The solutions presented in this category adopted a stateless approach to deal with IFA, i.e., do not store traffic related information in PIT.

The authors of [Gha+17] proposed a new stateless CCN architecture named stateless CCN. The solution modifies the Interest and Data packets to incorporate a new field, called *SupportingName*, which includes the consumer's prefix. The proposed architecture works as follows: when a consumer requests a content, it first advertises its name. Then it sends an Interest packet with the requested content name along with its name. When a producer satisfied the request, it appends with the content packet the consumer's name. Routers use the consumer's name to forward back the content packet. Adopting a stateless forwarding architecture comes with some disadvantages. Every consumer need to be identified, which raises prefix announcement

and management problems. Besides, adopting a stateless architecture reintroduces some TCP/IP based threats, like reflective and privacy attacks. Additionally, it does not benefit from the built-in multicast mechanism.

The solution presented in [AR16] uses cryptographic tokens, called route tokens, to skip the use of PIT. Each Interest and Data packet contains a route token component. Routers and producers use them to forward packets. When a consumer requests data, it sends along with the Interest packet an empty token route. Each receiving router updates the route token with additional information that are necessary 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 still 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.

3.6.3 Detection-only solutions

Some IFA detection-only mechanisms were authored. This subsection provides a summary of all detection-only solution present in the literature.

In [Ngu+15], the authors presented a solution that uses hypothesis testing theory to detect IFA. The authors presented two statistical models to detect IFA. The first model is the *Likelihood Ratio Test (LRT)*, which represents the theoretical optimal scenario where the legitimate traffic is known. The second model is a *Generalized LRT*, which represents the case of unknown legitimate traffic. Routers in this solution periodically record the number of Interest and Data packets associated with each interface and then calculate a packet-loss rate. Routers use the packet-loss rate to model legitimate traffic and then use it 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 detect IFA, which may give some false-positive results. Additionally, routers in this solution do not cooperate to detect IFA.

Another detection-only technique were presented in [Xin+17]. The proposed solution aims to detect collusive IFA. The proposed solution analyses the traffic using a discrete wavelet transform to detect the existence of a collusive IFA traffic. The proposed solution is not cooperative. Besides, the detection process can be resource-consuming in some scenarios. Similarly, the mechanism presented in [SS18] aims to detect collusive IFA. Each router in this solution monitors its PIT usage. When the router suffers a PIT overflow, it checks the incoming rate of its interfaces. If the rate of an interface goes above a predefined threshold, the router examines the cache reference of the requested Data packets. It represents the number of times the Data was satisfied from the router's cache. If this number equals zero, the router recognizes an ongoing attack. The solution relies on the fact that attackers send Interest packets

with different names. However, attackers can easily avoid being detected. Besides, the mechanism does not mitigate attackers. Additionally, the proposed solution can mistakenly penalize legitimate consumers in some scenarios (e.g., live streaming).

3.6.4 Proactive solutions

In this subsection, we present the existing IFA proactive solutions. In [Wan+17], the authors introduced micro-payments into the network to regulate IFA. The solution uses virtual money to reward good nodes and penalize malicious nodes. Authority nodes in the network act as banks to conduct any virtual money-related actions. Initially, every NDN node receives an amount of virtual money. 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 a router receives the Interest packet, it verifies if an entry with the same name exists in PIT. If not, it checks the PIT delay fee sent in the Interest packet; deducts the necessary amount before forwarding the Interest upstream. When the Interest packet reaches a destination, the node checks the content delivery fee. It then subtracts the necessary amount and sends the Data packet downstream. Every traversed router takes an amount from the content delivery fee. The consumer receives the requested Data packet only if the PIT delay and content delivery fees are sufficient to cross all the nodes. Otherwise, a message is sent to the consumer to inform him about the insufficiency of paid fees. In this solution, legitimate traffic is limited by the micro-payment system. Also, legitimate consumers can be penalized in some scenarios like unsatisfied requests or unavailable producers. Besides, deploying and maintaining such a system is extremely hard in large networks. Also, the system relies on central nodes to act as banks, which makes it vulnerable.

Another payment solution was proposed in [ZL19]. It relies on the autoregressive integrated (ARI) and the Hidden Markov Models (HMM). In this solution, edge routers act as banks and distribute virtual money to consumers. When a router receives an Interest packet from an interface, the price that the router charges to forward the incoming interest depends on the number of pending Interests 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. However, the charge/reward mechanism can penalize legitimate traffic. Also, relying only on the satisfaction ratio to predict the consumer's state may lead to false detection.

The authors of [LB14] presented *"Interest Cash"*, a solution that aims to countermeasure the signing overhead resulting from IFA requesting dynamic content. This solution employs a proof-of-work mechanism. The consumer needs to answer a puzzle sent from a producer before it can get the data. When a consumer expresses the need for specific data, it sends an Interest packet to the producer asking for a puzzle. When it receives the request, the producer sends back a meta-puzzle. The consumer computes an l bits long hash that starts with k bits of 0 with the name n, a timestamp

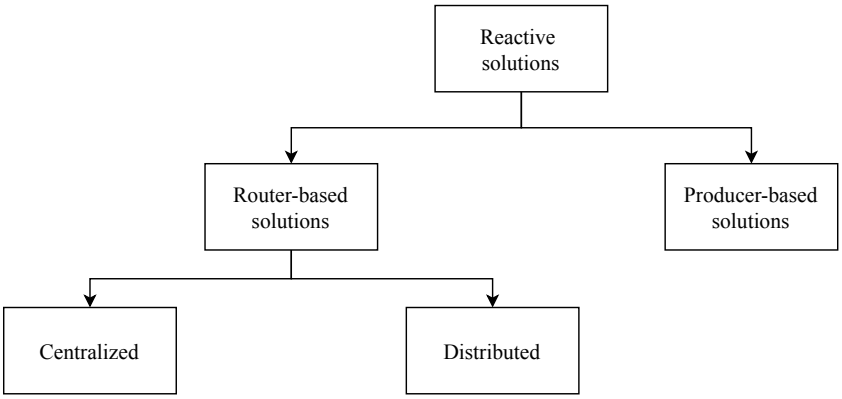

FIGURE 3.8: Classification of reactive solutions

t, and a random string x that the producer sent. After that, the consumer resends the interest packet with the computed value. If it is correct, the producer sends back the data packet. However, this solution does not prevent IFA. Attackers can still perform IFA with interest packets. Similarly, the solution in [TTM20] relies on tokens to prevent IFA. In this solution, consumers need to resolve computational puzzles sent by a proxy to get communication tokens. Edge routers maintain a table containing the information related to the computation puzzles and tokens that it receives from the proxy. When a consumer sends an Interest packet, the edge router verifies if the token exists in the list. If it does, it forwards it and updates the count number of this token. Otherwise, it discards the Interest packet. This solution also employs a Bloom filter-based mitigation technique to counter malicious traffic. 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 the PIT to store incoming Interests from this interface. However, this solution does not stop attackers.

The authors of [Liu+18b] proposed *"BLAM"*, a lightweight Bloom filter-based mitigation technique for IoT devices. In this proactive solution, each IoT node uses a Bloom filter array during the forwarding process. When an Interest packet arrives, and after performing a CS check, the node matches the name of the Interest packet with the Bloom filter array. If an entry exists, the node continues the forwarding process. Otherwise, the node considers the received Interest packet as malicious and discards it. The Bloom filter array is maintained by data producers. Nodes receive Bloom filter array updates within NACK packets. Since each node needs to know every Data packet that producers offer, the proposed solution cannot be deployed on large networks. Additionally, the proposed technique does not stop attackers.

3.6.5 Reactive solutions

Several IFA reactive detection and mitigation solutions have been proposed. Reactive solutions in this subsection are classified into router-based and producer-based solutions. Router-based solutions are further grouped into centralized and distributed solutions as showed in 3.8.

Centralized router-based solutions

Several centralized router-based IFA mitigation solutions were authored. The solution presented in [SWS15] is a centralized mitigation technique. It relies on a Domain Controller (DC) and a group of selected routers named Monitoring Routers (MR) to detect IFA. The MRs constantly monitor the traffic and calculate the PIT usage and the expired interests associated with each interface. When the calculated metrics exceed a certain threshold, the MR checks the name prefixes associated with the expired interests. After that, it calculates the expiration rate of every infected name prefix associated with each interface before sending the collected information to DC. The DC will then decide the infected name prefixes and inform the MRs. After the reception of a message, an MR drops packets with a certain probability. However, the metrics used can lead to false detection in some situations like high legitimate traffic or unavailable producer. Also, the legitimate requests going to infected prefixes can be affected. The authors proposed an enhanced version of this solution in [SS16]. This version aims to detect a collusive IFA. Compared to the previous solution, this one relies only on PIT usage to identify IFA, as expired Interests cannot be considered in a scenario with a malicious producer. However, relying only on the PIT usage may lead to false-positive situations affecting legitimate requests.

Another controller-based mechanism was proposed in [Yin+19]. In this solution, each router of a domain 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 satisfaction ratio and the number of NACK packets exceed their respective thresholds, the controller considers that this router is under attack. The controller sends back to the routers the malicious prefixes. When a router receives an interest packet with a malicious prefix, it blocks the interface. The domain controller maintains three data structures. First, the *Node Info Table (NIT)*, where the controller keeps the information about routers, like the router's name and id. Second, the *Affected Data Table (ADT)*, used to store information about affected routers. Last, the *Attack Info Table (AIT)*, which is used to store detailed information about attacks, including the time, the malicious prefixes, and the list of affected routers. However, The controller in this solution is a SPOF, which makes it a potential target for attackers.

An additional centralized solution was proposed in [Che+19b]. In this solution, edge routers monitor the rate of incoming and timed-out Interest packets of each interface. When these two metrics exceed their respective thresholds, the router sends the collected information to the controller within an Interest packet. When the controller receives the Interest packet, it replies with a Data packet. After that, it starts requesting traffic-related information from this edge router by sending specific Interest packets. Using the received information, the controller analyses the paths that Interest packets will take 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. The controller looks for the source of these Interests and notifies edge routers about the malicious interfaces. When the edge router receives this interest, it blocks the specified interfaces. The controller needs to have a global view of the network. Besides, the proposed solution may consider high traffic rates as malicious, thus block legitimate consumers. Also, the traffic reports that edge routers send to the controller can burden the network. Additionally, the central controller is a SPOF.

The solution presented in [AA20] relies on a domain controller, which manages the detection and mitigation policies, and a content provider. The authors identify three router types: The content provider router (CPR), which connects with the content provider. The entering/departing routers (En/DeR), which represent the routers placed at the extremity of the AS. And the intermediate routers (InR). When the content provider router receives an Interest packet, it checks the CS. If the content does not exist, 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, the CPR considers this Interest packet as malicious and sends a warning message to the controller with the name of this Interest packet. The controller then updates the QCF before informing the routers placed at the extremity of the AS. Each router monitors PIT expiration and occupancy ratios. When these two metrics exceed their respective thresholds, the router deletes fake Interests from PIT and sends back a warning message containing the fake prefix to downstream routers. When an edge router receives the warning message, it applies a restriction on the originating interface. However, the QCF table used by this solution could be used by attackers. Also, the content provider router needs to process all Interest packets of the AS.

Distributed router-based solutions

The majority of IFA solutions that exist in the literature are distributed router-based solutions. The authors of [Afa+13] proposed three solutions based on a token bucket approach. The first method consists of distributing the tokens equally among all interfaces. The second technique, called *"Token bucket with satisfaction-based interest acceptance"* consists of distributing the tokens according to interfaces' interest acceptance, i.e., the ratio of satisfied interest packets. The third token bucket-based technique proposed is named *"token bucket with satisfaction-based push-back"*. In this solution, the interest limit value also depends on the interface's satisfaction ratio. When the router calculates the limit values, it announces the calculated limit to their downstream neighbor routers. The receiving routers will then adapt their sending rates according to the received value.

The solution in [Com+13] is called *"Poseidon"*, which proposed two metrics to detect IFA: satisfaction ratio and PIT usage. The router calculates these metrics per interface and compares them with predefined thresholds. When the calculated values of an

interface exceed their thresholds, the router suspects malicious traffic. After that, it limits the incoming traffic and updates the thresholds according to a predefined scaling factor. If the scaling factor equals two, the newly calculated threshold value will equal half of its old value, which will penalize the incoming traffic from this interface. After that, the router sends an alert message to the router connected with this interface to inform it about the new reduced rate. Nevertheless, interface-based rate-limiting used can also impact legitimate consumers.

The authors of [Dai+13] proposed a solution that takes advantage of NDN's symmetric routing to trace back the originator of spoofed Interest packets, i.e., the Data packet takes the reverse path of the Interest. A Router monitors the size of its PIT to detect IFA. When the size exceeds a certain threshold, it triggers the trace-back process by generating a spoofed Data packet and sending it back to the originator of the spoofed Interest packet. The spoofed Data packet contains the name of the forged Interest, a spoofed content, and a special bit to indicate that this data is a spoofed Data packet. When an edge router receives a spoofed Data packet, it limits the rate of this interface by dropping incoming interest packets. However, Attackers can use the spoofed Data packets to overwhelm the network.

The authors of [Wan+13] proposed a solution called *"Disabling PIT Exhaustion(DPE)"*. This solution decouples the malicious from legitimate traffic to prevent PIT exhaustion. Routers monitor the number of timed-out Interest packets under each prefix to detect IFA. When this number reaches a threshold, the router considers the prefix under attack and stores this prefix in a list called *m-list* along with a decay time. When the router receives an Interest packet, it checks if its prefix already exists in the m-list. If an entry is found, the router adds the name of the source interface, i.e., the interface from where the interest packet came, to the name of the interest packet within a component called *Interface list*. After that, the router forwards the Interest packet without storing it in PIT. This solution limits the effects of an IFA but does not stop it. Also, it affects legitimate requests. Furthermore, attackers can use forged names to fill the m-list therefore, targeting the router's resources and making the solution inefficient.

The solution presented in [Wan+14a] relies on fuzzy logic to detect IFA. Every router on the network monitors the PIT occupancy and expiration rates. The first parameter represents the number of pending Interests to the maximum PIT capacity. The PIT expiration rate represents the number of timed-out Interests to the total number of pending PIT entries. The router calculates these values in real-time and takes them as entries for the detection algorithm. If the output identifies an IFA, the router sends an alert message to other routers. When a router receives an alert message, it checks for the malicious prefix, i.e., the prefix that has the highest number of pending and/or timed-out Interest packets and sends it back to the sourcing interfaces. After that, the router sends a pushback message containing the malicious

prefix to the sourcing interfaces. 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.

The author of [Wan+14b] proposed a technique called *"Threshold-based Detection and Mitigation (TDM)"*. This solution monitors timed-out Interest packets to detect IFA. It expands the FIB to include four additional fields: a counter for Interest packets *Ci*, another counter *Fs* to detect if an attack is still happening, a mode indicator *M* that has two values (ordinary or malicious), and a *Capacity* that a router uses when it applies rate-limiting. When an Interest packet times out before a Data packet returns, the router changes the mode value of this prefix in FIB to malicious and increments the counter *Ci*. If its value goes above a predefined threshold, the router sets the value of *Fs* to zero, enables the *Capacity* field associated with this prefix, and applies rate limiting. The rate-limiting used in this solution is associated with FIB entries, which affects all the traffic of this entry. Additionally, namespace rate-limiting can affect legitimate requests to this namespace.

The authors in [KGZ15] proposed a solution 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 received Interest and Data packets and the number of satisfied and timed-out Interests. When the detector module of a router detects malicious traffic, the router sends an alert message to all interfaces that sourced the malicious traffic. The alert message contains the new reduced rate, which is related to the unsatisfied rate of each interface. Additionally, the alert message comprises the generation timestamp and the reduction period. When a router receives an alert message, it reduces the traffic rate of the interface according to the obtained value and the unsatisfied rate of each interface. The neural network detection process used by this solution can consume a lot of the router's resources. Furthermore, rate-limiting can affect legitimate consumers.

Another machine learning-based detection mechanism was proposed in [Zhi+19]. This solution relies on the *Support Vector Machine (SVM)* to detect IFA and the *Jensen-Shannon divergence* to detect the malicious prefix. 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 SVM classifier. If the detector classifies it as an anomaly, the router declares that an IFA is happening. Following that, the router calculates the Jensen-Shannon divergence value between two sets of Interest prefixes to extract the malicious prefixes. After that, the router informs downstream routers about the malicious prefixes. The recipient routers stop forwarding Interest packets with these prefixes. This solution also affects legitimate Interest packets going to blocked prefixes. Additionally, the detection process may consume a lot of resources.

The solution presented in [Vas+15] relies on expired Interest packet statistics to detect IFA. Edge routers monitor the number of expired PIT entries of each interface. It classifies the interfaces into three categories: normal, suspicious, or malicious. If an edge router classifies an interface as suspicious, it reduces its traffic. On the other hand, if the interface's behavior is malicious, the edge router blocks it and sends a notification message to other routers to inform them about the blocked consumer. Identifying NDN consumers is not an easy task which makes this solution hard to apply.

The authors of [Din+16] presented a solution based on the vector space model and Markov chains. The solution uses the PIT occupancy ratio, expired PIT entries, and the number of Interest packets as metrics for detection. When the PIT size reaches an alarming level, the router calculates a state for each received Interest packet using these three parameters and a value α. Three Interest states are used: normal, unknown, and risk. When a router receives an Interest packet from another router, it 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 packet. Otherwise, the router attaches to the Interest packet its state info and an ID that identifies the edge router that sourced the Interest packet. To prevent legitimate requests from being discarded, the proposed solution stores a copy of non-satisfied Interests in a specific cache, so if the Interest packet with the same name arrives, the router considers the request as legitimate and forwards it. However, attackers can use this propriety to target routers. Additionally, The process of Interest packet state checking consumes the router's resources.

The solution presented in [Xin+16] uses cumulative and relative entropy to detect and mitigate IFA. In this solution, routers monitor the name of incoming Interest packets for every interface and compute its cumulative entropy. When this latter exceeds an upper bound, the router confirms that an attack is undergoing. Afterward, routers use a relative entropy algorithm to detect malicious prefixes. Routers also trigger a pushback mechanism that sends a spoofed Data packet to the originators of the malicious requests. When an edge router receives a spoofed Data packet, it applies rate limiting on the 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 packets as a tool to drown the network.

Similarly, the authors of [Hou+19] proposed a statistical solution that uses the *Thiel* index to detect IFA. The proposed technique relies on the fact that when IFA is undergoing, the occurrence of forged names increases, which decreases the Thiel entropy. Routers monitor the Thiel entropy value to detect if there is an attack. Every router records the names of incoming Interest packets and calculates their entropy to decide whether IFA is happening. As mitigation action, the proposed technique adopts the pushback mechanism. When the edge router receives the forged Data

packet, it blocks the interface. 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 resources, especially in the case of a massive attack.

Another statistical solution was proposed in [ZLL18]. It uses Gini impurity to detect IFA. A router monitors the name of received interest packets and calculates their probability. Then, it uses Gini impurity to measure the dispersity of the set and compares it with a threshold. When the metric exceeds a known normal range, the router affirms that 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. Once the malicious prefix is detected, the router applies rate-limiting on the prefix and sends a notification message containing the malicious prefix to downstream nodes. The proposed solution may lead to false-positive detection.

The solution presented in [NWN18], called *"FROG"*, is a hop count-based IFA detection technique. It aims to detect and mitigate IFA with existing data. Edge routers store the hop count of each received Data packet associated with every interface. Routers calculate the mean and variance of the recorded set of each interface. When the mean and variance are respectively higher and smaller than the normal, the router affirms that the consumer connected to this interface is malicious. The authors rely on the fact that malicious traffic is satisfied by producers and not intermediate caches, which makes the mean higher and the variance lower than the normal traffic. As mitigation action, the router blocks the malicious interface. This solution may mistakenly block legitimate users in scenarios where the legitimate requests need to be satisfied by the producer. Besides being a non-cooperative solution, the proposed solution is unable to detect attacks against routers.

The authors of [Ben+19a] presented a solution named *"ChoKIFA (Choose to Kill IFA)"*. It relies on *Active Queue Management (AQM)* to decouple malicious from legitimate traffic. For each received Interest packet, the router first checks the size of its PIT. If the size is above the minimum threshold, but below the maximum thresholds, the router performs the following actions. First, it randomly picks a pending Interest and matches its name prefix with the received Interest. Second, it compares their source interfaces. Third, the router checks if the satisfaction ratio of the interface that sourced the incoming Interest packet is under a predefined threshold. If all three conditions are met, the router drops both Interest packets. Otherwise, it applies a dropping probability on the Interest to decide whether it forwards or drops the incoming interest. On the other hand, when the size of the PIT exceeds the maximum threshold, the router drops all incoming Interest packets. The proposed solution is a traffic control mechanism, which does not stop IFA. In addition, attackers can timely orchestrate attacks to fill the target's PIT so the router will start dropping incoming Interest packets. An enhanced version of this technique was proposed in [Ben+20a]. In this version, the mitigation process does stop attackers. When an edge

router receives an Interest packet, it checks the PIT size and the satisfaction ratio. If both metrics exceeded their maximum thresholds, the router blocks the interface. However, legitimate traffic is still penalized by the mechanism and it increases as it gets close to the core network.

Similarly, the authors of [ZLW20] proposed a reputation-based early detection mechanism to counter collusive IFA. A router monitors the PIT usage of every interface. Each interface has a minimum and a maximum threshold that the router uses when dropping Interest packets. If the PIT usage is under the minimum threshold, the router forwards all Interest packets. If it is above the minimum and below the maximum threshold, the router drops incoming Interest packets with a probability. Otherwise, it discards every incoming Interest packet from this interface. The router defines the minimum and maximum thresholds according to the interface's reputation value, which is calculated from the satisfaction ratio and the average RTT of the interface. This solution takes into consideration that attackers estimate the maximum RTT when the producer sends back Data packets. It ensures that the malicious Interest packets will stay longer in routers' PIT. The proposed solution is a traffic control mechanism, which does not stop attackers. Finally, the metrics used to calculate the interface reputation value are not reliable to give precise results, which may lead to dropping a large portion of legitimate Interest packets, especially in core routers.

In [PPB19], the authors presented a share-based mechanism. It introduces a new structure called *Operation Trace Table (OPT)*. The router uses this table to record the traffic statistics associated with each interface. Each entry has seven components: source interface, number of Interest and Data packets, number of expired and NACK packets, and the share of this interface. The latter represents the number of allowed Interest packets. The router calculates, for each interface, the related unsatisfaction ratio. It represents the number of expired and NACK packets to the total number of received Interest packets within an observation window. After that, the router compares it with the average unsatisfaction ratio. If it surpasses this value, the router reduces the difference from the interface's share and distributes it equally among all interfaces. If the router receives an Interest packet from an interface that reached its allowed share, it forwards the Interest to another outgoing interface to discover unknown paths. The proposed solution does not mitigate attackers. In addition to that, sending Interest packets to other paths may overwhelm upstream links and routers. Attackers can take advantage of this feature to inflict additional damage on the network.

The authors presented *"iForest"* [Che+19a]. An isolation forest-based IFA mitigation technique. Routers record for each name prefix, four metrics, the number of Interest and Data packets, the number of PIT entries, and the number of timed-out Interest packets. After that, routers randomly select some prefixes from the recorded list. Following that, the routers randomly pick a metric and selects a value from the minimum and maximum values of all items. Then, the routers construct a search binary

tree, which the authors named *iTree*, based on the selected value. The router constructs other iTrees to form an iForest. The routers determine the average traverse path of prefixes to calculate an abnormal score for each prefix. If the abnormal score of a prefix exceeds a predefined threshold, the router considers it as malicious. As mitigation action, the router sends a notification message with the malicious prefix to downstream routers. When a router receives a notification message, it limits the traffic of the malicious prefix. The namespace-based rate-limiting that this solution adopts also penalizes legitimate Interest packets. Besides, the detection process can consume a lot of resources, especially for core routers. Additionally, this technique can take a long time before it detects an attack.

The authors in [Tan+13] presented a two-phase detection mechanism. During the first phase, called the *rough detection* phase, routers calculate the satisfaction ratio associated with each interface. When the rate exceeds its threshold, the router starts the *accurate identification* phase. During this phase, the router counts the number of expired Interests under each prefix. If the number exceeds a predefined threshold, the router considers the prefix as malicious and starts blocking incoming Interest packets going to this prefix. Likewise, the authors of [Shi+16] presented IFBN, an Interest flow balancing method that focuses on the number of requests. A Router calculates a reputation value for each one of its interfaces. This value depends on its satisfaction ratio. If an interface has a low reputation value, the router checks the number of PIT entries associated with this interface. If the number of pending Interests is high, the router concludes that this interface is sending malicious requests and stops recording incoming Interest packets in PIT. These two solutions do not prevent attackers from sending malicious Interest packets. Additionally, they may lead to blocking legitimate Interest packets as they rely only on the satisfaction ratio.

Producer-based solutions

There are some producer-based reactive solutions in the literature. The solution proposed in [Zha+19] is named *"Fine-grained Interest Traffic Throttling (FITT)"*. It relies on producers' messages to initiate IFA mitigation. When a producer needs to release its resource that suffers overhead due to high demand or malicious traffic, it sends a specific NACK packet to the gateway router. The generated NACK carries four different parameters. The first parameter is *RSN*, or the reason, which represents the nature of the traffic, fake or valid. Second, the *PREF* parameter, which contains the affected prefix. Third, capacity *C* expresses the traffic rate that the producer can support under the prefix PREF. The fourth and last parameter is *FakeList*. It incorporates all the fake names received under the affected prefix. When a router receives the NACK, it first checks the *RSN* parameter. If it equals fake, the router matches the pending Interests with the received *FakeList* and inspects the source interface. After that, and for each source interface, the router generates a new NACK containing a new FakeList, i.e., fake Interest names received from this particular interface, along

with *RSN* and *PREF*. If the value of parameter *RSN* equals valid, the receiving router will distribute equally the capacity *C* between the source interfaces. Following that, the router informs the downstream node about the new capacity *C*. If an edge router receives a NACK with *RSN* equal to fake, it blocks the source interfaces for some time. However, the fake Interest names list can be huge, especially in a large distributed attack. Additionally, the proposed technique relies on producers' feedback to initiate the mitigation process, which leaves routers vulnerable to attacks.

A new version of [Wan+13], named *"lightweight m-list table based attack detecting mechanism (mTBAD)"* was proposed in [Liu+18a]. Similar to the previous solution, routers use the *m-list* and packet marking during the forwarding process. However, this solution introduced a new producer-based monitor to reduce the detection delay. When a producer receives a malicious interest packet, it responds with an *m-NACK* to inform the downstream routers about the malicious request. When a router receives the m-NACK packet, it checks the m-list whether a similar entry exists or not. Although it reduces the detection delays, this solution still does not stop attackers. Additionally, producers may consider some legitimate requests as malicious.

The authors in [Don+20] proposed *"InterestFence"*. In this solution, data producers generate content names using a hash function before signing and publishing them to the network. When a data producer detects an abnormal number of forged Interest packets, it sends an alarm message to downstream routers. The alarm message is a NACK that contains the prefix under attack and cryptographic information that routers use to verify incoming Interest packets. Each router in the system maintains a list called *m-list"*, where it stores the affected prefixes, associated hash information, and a TTL. When the router receives an Interest packet, it first checks if the prefix is under attack in the m-list. If it exists, the router performs a hash verification with the information received from the producer. If there is a match, the router forwards the Interest packet. Otherwise, the router considers the Interest packet as forged and drops it. This solution does not stop attackers. Also, routers need to process every received Interest packet, which may consume resources due to the name verification process, especially in large attacks. Besides, 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.

The solution presented in [Wu+20] intends to mitigate collusive IFA. The proposed solution analyses the network traffic inside fixed time windows. Routers monitor the throughput and PIT usage during each time window to detect malicious traffic. When these two metric goes above their thresholds, the router judges the traffic abnormal. As mitigation action, the router starts to delete PIT entries that existed for the longest time. However, this solution may delete legitimate requests. Also, it may lead to false-positive detection in some scenarios.

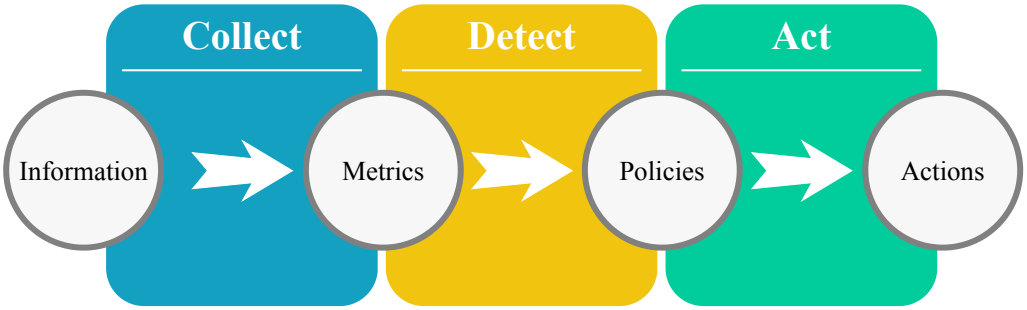

FIGURE 3.9: CDA workflow

3.7 CDA Workflow: Collect-Detect-Act

In this section, we present and explain CDA. CDA is a workflow that every IFA detection and mitigation solution follows. CDA consists of three elemnts: *"Collect"*, *"Detect"* and *"Act"* as illustrated in Fig.3.9.

3.7.1 Collect

The first activity of the CDA workflow is *Collect*. During this activity, solutions collect a set of traffic related information. The nature of this information varies from one solution to another. There are two types of collection metrics, router-based metrics and producer-based metrics. Some solutions collect only router-based information like [Afa+13], [SWS15], and [Ben+19b], while others collect producer-based information like [Zha+19] and [Liu+18a]. On the other hand, solutions may collect both router-based and producer-based information like [Ben+20b]. Table 3.1 presents the detection parameters used by existing solutions and table 3.2 summarizes all the metrics that existing solutions gather during the *Collect* activity.

Router-based collection metrics

1. **The traffic rate**: The first router-based metric used in the collection phase is the incoming traffic rate. It represents the rate of incoming Interest packets of each interface. When the sending rate of an interface reaches a certain predefined threshold, the router suspects the source interface. The interface traffic rate is usually used with other metrics.

2. **The satisfaction ratio**: The second metric that routers collect to detect eventual IFA is the satisfaction ratio. It corresponds to the number of received Data packets to the number of requested Interest packets in a given period. The router monitors the traffic of each interface and periodically calculates its satisfaction ratio.

$$Satisfaction\ Ratio_i = \frac{\#\ of\ Data\ Packets\ Received_i}{\#\ of\ Interest\ Packets\ Issued_i} \tag{3.1}$$

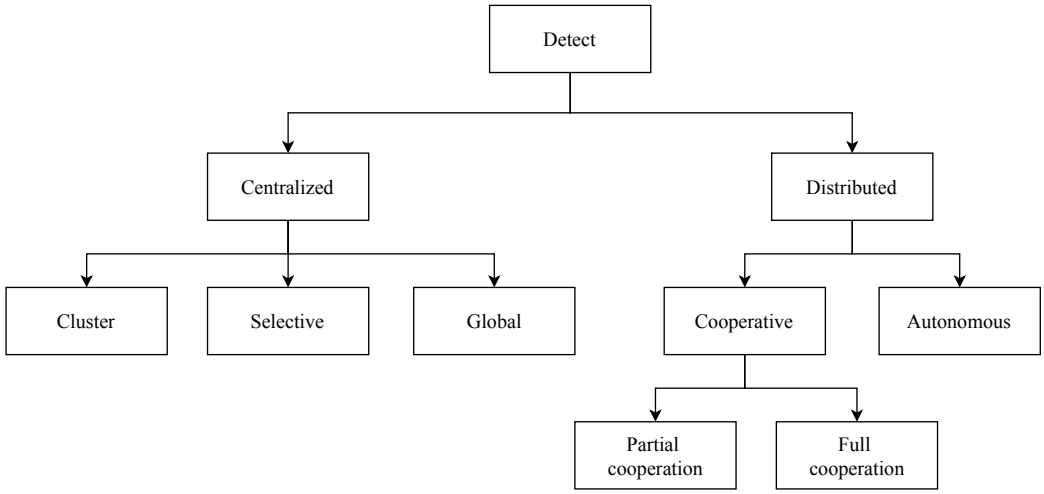

FIGURE 3.10: Classification of detection methods

3. **PIT usage**: Another traffic metric that routers collect is PIT usage. This metric represents the occupancy ratio of each interface in the PIT, i.e., the number of pending Interest packets generated by an interface to the total number of PIT entries. Routers monitor their PIT to check the occupancy ratio of each interface.

4. **Number of packets**: Routers may count the number of a variety of packets during IFA detection, including the number of received Interest and Data packets, the number of timed-out pending Interest packets, and the number of NACK packets.

5. **Interest prefixes** This metric represents the name prefixes of requested data. Routers could store statistics about the name prefixes of received Interest packets to detect attacks.

Producer-based collection metrics

The producers may collect a variety of metrics that could help in detecting IFA. It includes PIT-related metrics and the incoming traffic rate. Moreover, producers collect hardware and service-related metrics like processing overhead, memory consumption, and service overload.

3.7.2 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 [SS16] and [Che+19b], and distributed, like [Com+13] and [Dai+13]. Figure 3.10 shows the different detection approaches that IFA solutions may adopt. The detection parameters used by existing solutions are summarized in 3.3.

TABLE 3.1: Collection parameters used by existing solutions

Ref	Router statistics	Producer statistics	Statistics storage	Router classes	Modified PIT	Modified FIB	New structure	Central node
[Afa+13]	Yes	No	No	01	Flags: Inqueue, Fwd	No	No	No
[Com+13]	Yes	No	No	01	No	No	No	No
[Dai+13]	Yes	No	No	01	No	No	No	No
[Wan+13]	Yes	No	Yes	01	No	No	Affected prefixes list	No
[Liu+18a]	Yes	Yes	Yes	01	No	No	Affected prefixes list	No
[Wan+14a]	Yes	No	Yes	01	No	No	No	No
[Wan+14b]	Yes	No	Yes	01	No	Yes	No	No
[KGZ15]	Yes	No	No	01	No	No	No	No
[Zhi+19]	Yes	No	Yes	01	No	No	No	No
[Vas+15]	Yes	No	No	01	No	No	No	No
[SWS15] [SS16]	Yes	No	Yes	• Monitoring • Normal	No	No	No	Yes
[Yin+19]	Yes	No	Yes	01	No	No	ADT and AIT	Yes
[Ngu+15]	Yes	No	Yes	01	No	No	No	No
[Xin+17]	Yes	No	No	01	No	No	No	No
[SS18]	Yes	No	Yes	01	No	No	No	No
[Din+16]	Yes	No	Yes	01	No	No	Unsatisfied interests cache	No
[Xin+16]	Yes	No	Yes	01	No	No	No	No
[Hou+19]	Yes	No	Yes	01	No	No	No	No
[ZLL18]	Yes	No	Yes	01	No	No	No	No
[NWN18]	Yes	No	Yes	01	No	No	No	No
[Ben+19a], [Ben+20a]	Yes	No	No	01	No	No	No	No
[ZLW20]	Yes	No	No	01	No	No	No	No
[Zha+19]	Yes	Yes	Yes	01	No	No	No	No
[PPB19]	Yes	No	Yes	01	No	No	Operation Trace Table	No
[Che+19b]	Yes	No	Yes	• Monitoring • Normal	No	No	No	Yes
[Che+19a]	Yes	No	Yes	01	No	No	No	No
[Ben+19b]	Yes	No	No	01	No	No	No	No
[Don+20]	No	Yes	Yes	01	No	No	Affected prefixes list	No
[Tan+13]	Yes	No	No	01	No	No	No	No
[Shi+16]	Yes	No	No	01	No	No	No	No
[Ben+20b]	Yes	Yes	No	01	No	No	No	No
[AA20]	Yes	No	Yes	01	No	No	Quotient based Cuckoo filter	Yes
[Wu+20]	Yes	No	No	01	No	No	No	No
[TTM20]	Yes	No	No	01	No	No	Bloom filter	No
[Wan+17]	No	No	No	01	No	No	No	No
[ZL19]	Yes	No	No	01	No	No	No	No
[Liu+18b]	No	No	No	01	No	No	Bloom filter array	No

TABLE 3.2: 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**	
[Afa+13]	✓								
[Com+13]		✓		✓					
[Dai+13]		✓							
[Wan+13]				✓					
[Liu+18a]									Malicious interests
[Wan+14a]		✓		✓					
[Wan+14b]				✓					
[KGZ15]	✓		✓	✓	✓	✓			
[Zhi+19]	✓	✓						✓	
[Vas+15]				✓					
[SWS15]		✓		✓					
[SS16]		✓							
[Yin+19]	✓						✓		
[Ngu+15]					✓	✓			
[Xin+17]						✓			
[SS18]		✓	✓						
[Din+16]		✓		✓		✓			
[Xin+16]								✓	
[Hou+19]								✓	
[ZLL18]								✓	
[NWN18]									
[Ben+19a], [Ben+20a]	✓	✓							
[ZLW20]	✓								
[Zha+19]									Resource exhaustion
[PPB19]	Unsatisfaction ratio								
[Che+19b]			✓	✓					
[Che+19a]		✓		✓	✓	✓			
[Ben+19b]	✓		✓	✓			✓		
[Don+20]									Number of forged interests
[Tan+13]	✓			✓					
[Shi+16]	✓	✓							
[Ben+20b]	✓			✓					Resource exhaustion
[AA20]		✓		✓					
[Wu+20]		✓	✓						
[TTM20]			✓						
[Wan+17]									
[ZL19]	✓								
[Liu+18b]									

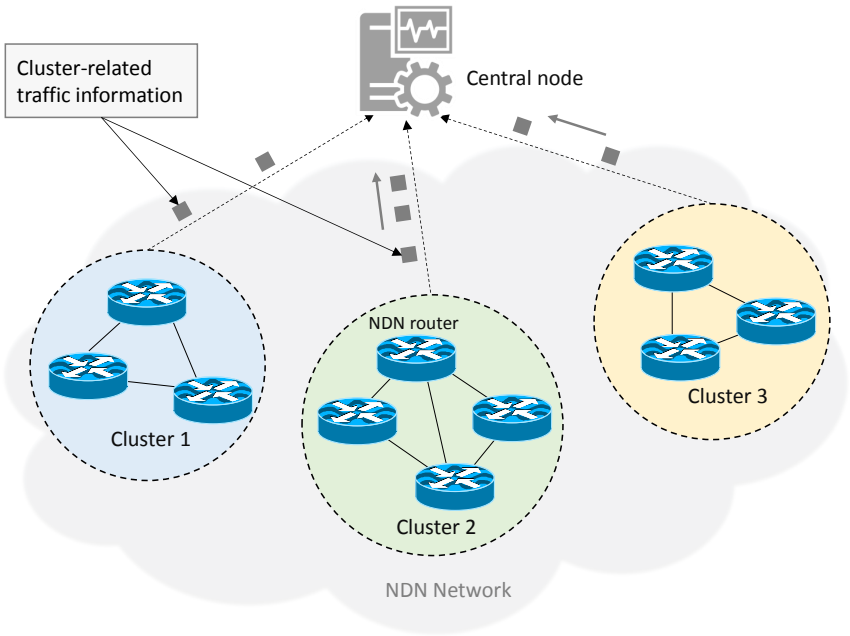

FIGURE 3.11: Cluster-based centralized detection

Centralized detection

The centralized detection architecture relies on a central node to detect 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.

1. **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 necessary actions to mitigate the attack. Figure 3.11 illustrates the cluster-based centralized detection.

2. **Selective nodes**: The second centralized-based detection topology relies on 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 chosen routers are responsible for sending the information that the central node needs to evaluate whether the traffic is malicious or not, as illustrated in Fig. 3.12. After that, the central node sends back to the selected routers the proper actions to take.

3. **Global**: The third and last centralized-based detection topology is the global

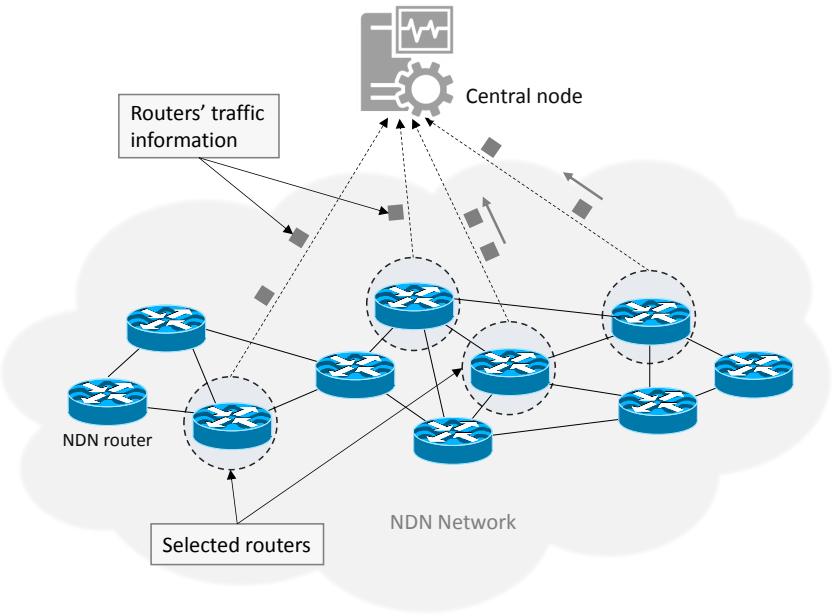

FIGURE 3.12: Selective nodes centralized detection

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 as shown in Fig. 3.13. 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.

Distributed detection

The distributed detection is divided into autonomous detection and cooperative detection.

1. **Autonomous detection**: The first distributed detection approach is 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.

2. **Cooperative detection** The second distributed detection approach that network nodes can adopt is cooperative detection. Network nodes share traffic-related information and take collective actions to mitigate IFA. Cooperative detection is further classified into partial cooperation and full cooperation.

    - *Partial cooperation*: In a partial cooperation design, network nodes collaborate only on the mitigation action. Each node analyzes its local traffic to detect IFA and then takes the proper action to mitigate it. After that, the node informs its neighbors about its action to cooperates in blocking the attack and stopping its growth.

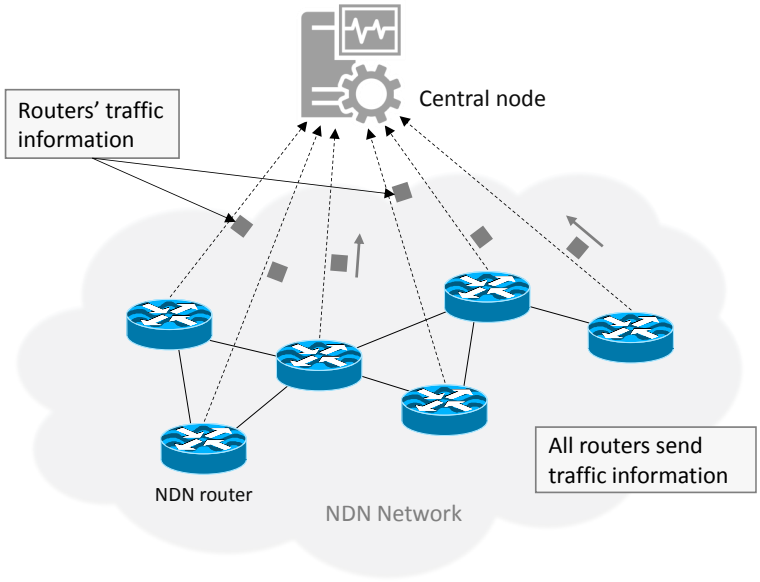

FIGURE 3.13: Global nodes centralized detection

- *Full cooperation*: In full cooperation distributed detection, the nodes fully cooperate to detect and mitigate an IFA, i.e., the nodes share information and jointly decide on the actions to take.

3.7.3 Act

The last activity of the CDA workflow is "Act". After collecting the traffic-related information and detecting the existence of an IFA, solutions act by taking mitigation actions to stop the attack. Table 3.4 summarizes the mitigation parameters used by existing solutions.

Router-based mitigation actions

1. **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 malicious traffic, it first checks the source interface of this traffic. Following that, the router reduces the incoming traffic from this interface.

2. **Block** The second router-based mitigation action consists of blocking the traffic of an interface. Network routers, when necessary, can block their interfaces for a period. The blocking action is usually considered 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, i.e., the routers that connect consumers to the network. The

TABLE 3.3: 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
[Afa+13]	All routers	No	Autonomous	No	No	1*sec*	Non existent	No	No	01	-
[Com+13]	All routers	No	Autonomous	No	No	60*ms*	Non existent	No	No	02	-
[Dai+13]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[Wan+13]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[Liu+18a]	All routers Producers	No	Cooperative	No	No	-	Non existent	No	No	01	-
[Wan+14a]	All routers	No	Cooperative	Fuzzy logic	No	real-time	-	No	No	02	Normal Malicious
[Wan+14b]	All routers	No	Autonomous	No	No	-	Non existent	No	No	02	-
[KGZ15]	All routers	No	Autonomous	RBF NN	No	-	All types	No	No	02	-
[Zhi+19]	All routers	No	Autonomous	SVM	No	time period	Non existent	No	No	-	-
[Vas+15]	Edge routers	No	Autonomous	No	No	-	Non existent	No	No	02	Legitimate Suspicious Malicious
[SWS15]	Mon routers and DC	Selective nodes	Autonomous	No	No	Window q	Non existent	Yes	No	02	-
[SS16]	Mon routers and DC	Selective nodes	Autonomous	No	No	Window q	Collusive	Yes	No	01	-
[Yin+19]	DC	Global nodes	No	No	No	-	Non existent	Yes	No	02	-
[Ngu+15]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[Xin+17]	Edge routers	No	Autonomous	No	No	-	Collusive	No	No	01	-
[SS18]	All routers	No	Autonomous	No	No	-	Collusive	No	No	02	-
[Din+16]	Edge routers	No	Autonomous	No	No	Windows L	Non existent	No	Interest state and router ID	01	-
[Xin+16]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[Hou+19]	All routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[ZLL18]	All routers	No	Autonomous	No	No	Time Δt	Non existent	No	No	01	-
[NWN18]	Edge routers	No	Autonomous	No	No	Time window	Existent	No	No	02	-
[Ben+19a], [Ben+20a]	All routers	No	Autonomous	No	No	-	Non existent	No	No	03	-
[ZLW20]	All routers	No	Autonomous	No	No	-	Collusive	No	No	02	-
[Zha+19]	Producers	No	Autonomous	No	No	-	All types	No	No	Producer based	Normal Suspicious
[PPB19]	Edge routers	No	Autonomous	No	No	Window ω	Non existent	No	No	01	-
[Che+19b]	Mon routers DC	Selective nodes	No	No	No	-	Non existent	Yes	No	03	-
[Che+19a]	All routers	No	Autonomous	No	No	time interval	Non existent	No	No	02	-
[Ben+19b]	Edge routers	No	Autonomous	No	Yes	-	Non existent	No	No	04	Legitimate Suspicious Malicious
[Don+20]	Producers	No	Autonomous	No	No	-	Non existent	No	No	Producer based	-
[Tan+13]	All routers	No	Autonomous	No	No	-	Non existent	No	No	02	-
[Shi+16]	All routers	No	Autonomous	No	No	-	Non existent	No	No	02	-
[Ben+20b]	All routers Producers	No	Cooperative	No	No	-	All types	No	No	03	Normal Suspicious Harmful
[AA20]	All routers	No	Cooperative	No	No	-	Non-existent	No	No	02	-
[Wu+20]	All routers	No	Autonomous	No	No	Time window	Collusive	No	No	02	-
[TTM20]	Core routers	No	Autonomous	No	No	-	Non existent	No	No	01	-
[Wan+17]	All routers	No	Cooperative	No	No	-	All	No	PIT delay& content fees	-	-
[ZL19]	Edge routers	No	Autonomous	No	No	-	Non existent	No	No	-	Legal Bad
[Liu+18b]	All nodes	No	Autonomous	No	No	-	Non existent	No	No	-	-

blocking periods may vary during the time. Routers could increase the blocking period when the previous amount of time did not help to stop the attack.

Producer-based mitigation actions

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

1. **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).
2. **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 blocking actions against network interfaces.

3.8 Summary

In this chapter, we started by briefly discussing the availability attacks that target NDN. After that, we presented the Interest Flooding Attack and explaining how attackers perform such an attack. Following that, we showed different IFA variants. Later, we detailed and mentioned the drawbacks of all the relevant IFA related work. Finally, we presented the CDA workflow. In the next chapter, we will present our first contribution against IFA, a novel congestion-aware solution.

TABLE 3.4: Mitigation parameters used by existing solutions

Ref	Mitigation method	Mitigation actors	Mitigation actions	Specific packets	Specific namespace	Control information	Signed interests
[Afa+13]	Autonomous	All routers	Rate-limit	No	No	No	No
[Afa+13]	Cooperative	All routers	Rate-limit	Yes	-	Rate limit announcements	No
[Com+13]	Cooperative	All routers	Rate-limit	Yes	`/pushback/alerts/`	Reduced rate	No
[Dai+13]	Cooperative	All routers	Rate-limit	Yes	No	Spoofed data	No
[Wan+13]	Cooperative	All routers	Forwards interest without using PIT	Yes	No	Interfaces list	No
[Liu+18a]	Cooperative	All routers Producers	Sends Nack	Yes	No	Malicious interest	No
[Wan+14a]	Cooperative	Edge routers	Rate-limit (prefix)	Yes	`/ALERT/IFA/`	Malicious prefix	No
[Wan+14b]	Autonomous	All routers	Rate-limit (prefix)	No	No	No	No
[KGZ15]	Cooperative	All routers	Rate-limit	Yes	`/pushbackmessage/ alert/`	Reduced rate	No
[Zhi+19]	Cooperative	All routers	Block (prefix)	Yes	Empty	Malicious prefixes	No
[Vas+15]	Autonomous	Edge routers	Rate-limit Block	Yes	-	Blocked user	No
[SWS15] [SS16]	Cooperative	Mon routers	Rate-limit (prefix)	Yes	-	Infected prefixes	No
[Yin+19]	Cooperative	Edge routers	Block	Yes	`/CTRL/`	Malicious prefixes	No
[Din+16]	Cooperative	All routers	Rate-limit (prefix)	No	No	No	No
[Xin+16]	Cooperative	All routers	Rate-limit	Yes	No	Spoofed data	No
[Hou+19]	Cooperative	All routers	Block	Yes	No	Spoofed data	No
[ZLL18]	Cooperative	All routers	Rate-limit (prefix)	Yes	No	Malicious prefix	No
[NWN18]	Autonomous	Edge routers	Block	No	No	No	No
[Ben+19a]	Autonomous	All routers	Rate-limit	No	No	No	No
[ZLW20]	Autonomous	All routers	Rate-limit	No	No	No	No
[Ben+20a]	Autonomous	Edge routers	Rate-limit Block	No	No	No	No
[Zha+19]	Cooperative	All routers Producers	Rate-limit Block	Yes	No	RSN, PREF, C FakeList	No
[PPB19]	Autonomous	Edge routers	Rate-limit	No	No	No	No
[Che+19b]	Cooperative	Edge routers	Block	Yes	`/ndn/ddos/flooding/`	Interfaces list	Yes
[Che+19a]	Cooperative	All routers	Rate-limit (prefix)	Yes	-	Malcious prefix	No
[Ben+19b]	Autonomous	Edge routers	Block	No	No	No	No
[Don+20]	Cooperative	All routers Producers	Rate-limit	Yes	No	Affected prefixes	No
[Tan+13]	Autonomous	All routers	Block (prefix)	No	No	No	No
[Shi+16]	Autonomous	All routers	Rate-limit	No	No	No	No
[Ben+20b]	Cooperative	Edge routers	Rate-limit (prefix) and Block	Yes	`/ndn/PCIP` `/ndn/RCIP`	Affected prefix	Yes
[AA20]	Cooperative	Edge routers	Rate-limit	Yes	- Malicious prefix	No	
[Wu+20]	Autonomous	All routers	Delete interests	No	No	No	No
[TTM20]	Autonomous	Core routers	Bloom filter	No	No	No	No
[Wan+17]	Cooperative	All routers	Rate-limit	Yes	No	Insufficient fees	No
[ZL19]	Autonomous	Edge routers	Rate-limit	No	No	No	No
[Liu+18b]	Autonomous	All nodes	Rate-limit	No	No	No	No

Chapter 4

A novel congestion-aware Interest Flooding Attacks detection mechanism in Named Data Networking

Network congestion is crucial when it comes to detecting IFA because it could lead to false detection of attacks. The existing solutions did not consider network congestion. Assuming that a path between a data consumer C and some producer P is congested, it will lead routers to mistakenly consider legitimate consumers and traffic as malicious. In this chapter, we present a solution that integrates network congestion as a parameter. It will ensure the propagation of more precise and reliable information about the attack.

4.1 Design Overview

The proposed solution is an autonomous solution deployable at edge routers. It consists of monitoring two network metrics, the satisfaction ratio, and the incoming traffic rate, along with the network congestion to detect ongoing attacks. During the congestion detection phase, routers further monitor two other network metrics: the number of timed-out Interest and NACK packets.

4.2 Satisfaction Ratio

The first parameter used in this solution is the *satisfaction ratio*. Every router monitors its interfaces and periodically calculates the satisfaction ratio associated with each interface. Routers will then use the calculated satisfaction ratio to detect malicious interfaces. When the satisfaction ratio of an interface is low, the router will consider this traffic as suspicious.

The satisfaction ratio could lead to false detection of a malicious interface (e.g., when the network is congested or the data producer is not reachable). Therefore, we induced network congestion and interest rate parameter within our proposed solution to get reliable detection results.

4.3 Incoming Interest Rate

The incoming Interest rate represents the number of Interest packets arriving at a specific interface in a given amount of time. Routers use this parameter to detect abnormal rates. Every router monitors the incoming rate of each interface. When the rate exceeds the average threshold level, the routers will consider it suspicious and starts investigating. Router calculate the average rate of each interface independently, and it is obtained as follow:

$$NewAvg_i = (1 - \alpha)OldAvg_i + \alpha \times inr_{it} \tag{4.1}$$

In eq. 4.1, $NewAvg_i$ is the newly calculated average of interface i, $OldAvg_i$ represents the actual average associated with interface i, while inr_{it} is the incoming rate to interface i in time t. Furthermore, the used skew factor (α) is calculated as:

$$\alpha = \begin{cases} 0.1, & \text{if } inr_{it} \in [Low_{th}, Avg_i] \\ 0.00001, & \text{if } inr_{it} \in [Avg_i, Up_{th}] \end{cases}$$

where, Avg_i represents the average rate of interface i and inr_{it} is the incoming rate to interface i at time t. Further, Up_{th} is the upper bound, which is given as $avg + 10\%Avg$ while Low_{th} represents the lower bound threshold with value of $avg - 50\%Avg$.

In equation 4.1, we chose these values of α after a series of tests. The main motivation of these tests was to identify and chose a value that doesn't influence the average when an Interest Flooding Attack is happening. At the same time, a value that takes into consideration the traffic piques that go above the average sometimes.

We also decided to take into consideration only the 10% (above average) part. So if an attack is happening, it will not make the average value increases. For the lower bound α, we chose to take only the 50% below average and exclude the other part so it will not make the average rate decreases very quickly (e.g., when an interface is not active). Figure 4.1 shows an example of rate average and upper/lower thresholds.

4.4 Network Congestion

The third and last parameter of our proposed solution is network congestion. The network congestion could lead to false results upon detection of an IFA. When a

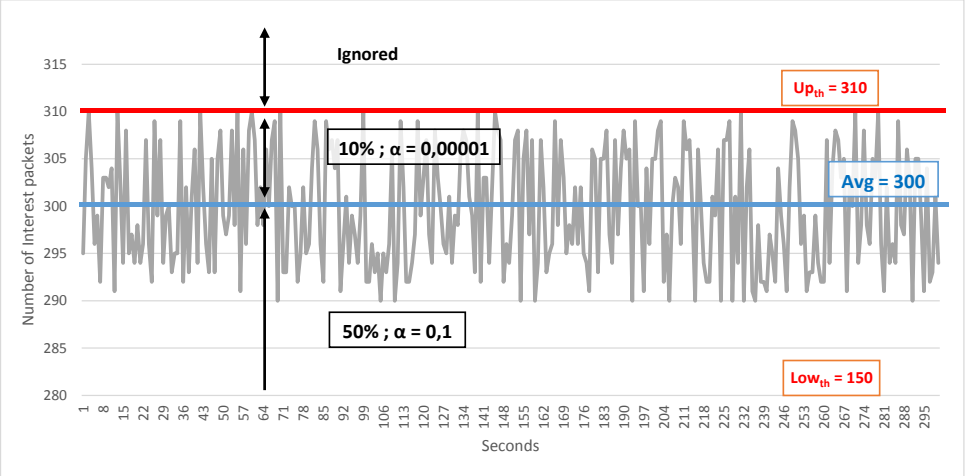

FIGURE 4.1: Average rate calculation example

router adds an Interest packet to PIT, it assigns with this Interest a lifetime (default 4 seconds). This newly added Interest will stay in PIT as long as it is not yet satisfied (Data packet returned) or timed out (no Data packet returned).

When the network is congested, Interest and Data packets take a long time to reach a destination. This delay leads pending Interests in PIT to timeout, i.e., the pending interests times out before they are satisfied.

We took a simple approach to detect network congestion. The detection process is done locally by every router. In our solution, we chose two parameters to identify a congested network:

- *Number of timed-out Interests*: When network congestion occurs, the number of timed-out Interest packets increases. This is due to the delay as the pending Interests take a longer time to be satisfied.
- *Number of NACK packets*: NACK packets are used to carry a response to an Interest packet that is not satisfied. When the network is congested, the number of NACK packets arriving at a time decreases because of the delay.

We compare these two parameters to get the state of the network (congested or not). If the numbers are not identical or nearly identical, the network is considered congested (The Interest packets are timed out, and no NACK packets are returned). Another parameter that could be used to detect network congestion in NDN is the PIT occupation. When a network is congested, Interest packets take a longer time to be satisfied, thus, staying longer in PIT of NDN routers.

4.5 Detection and Mitigation Process

The routers monitor the incoming rate of their interfaces. The average is calculated periodically by calling the *Average* procedure presented in algorithm 1.

Algorithm 1 Rate average calculation

```
1: procedure AVERAGE(int OldAvg, int inr)
2:     Avg ← OldAvg
3:     if ((Avg − (50%Avg)) ≤ inr ≤ Avg) then
4:         Avg ← 0.9 × Avg + 0.1 × inr
5:     else if (Avg ≤ inr ≤ (Avg + 10%Avg)) then
6:         Avg ← 0.99999 × Avg + 0.00001 × inr
7:     end if
8:     return Avg
9: end procedure
```

When the incoming traffic rate of an interface i exceeds its average rate, the router will consider it suspicious and starts to investigate. The router first checks the satisfaction ratio of this interface by calling the procedure *SatisfactionRatio* presented in algorithm 2. If the satisfaction ratio is low, the router checks the network state, following the algorithm introduced in 3, to test if the network is congested or not.

Algorithm 2 Satisfaction ratio calculation

```
1: procedure SATISFACTIONRATIO(interface i, time t)
2:     float ratio ← 0
3:     Integer int_nbr ← Number of Interest packets sent by interface (i) in periode t
4:     Integer data_nbr ← Number of Data packets received by interface (i) in periode t
5:     ratio ← data_nbr/int_nbr
6:     return ratio
7: end procedure
```

Algorithm 3 Testing Network Congestion

```
1: procedure ISCONGESTED
2:     bool congested ← false
3:     Integer timeoint_nbr ← Number of timed-out Interests sent by all interfaces
4:     Integer nack_nbr ← Number of NACK received by all interfaces
5:     if (timeoint_nbr >> nack_nbr) then    ▷ # of timed-out Interest is much bigger than # of Nack
6:         congested ← true
7:     end if
8:     return congested
9: end procedure
```

If all three parameters are satisfied, i.e., the incoming rate is below average, the satisfaction ratio is low, and the network is not congested, the router considers this interface as malicious and blocks it as showed in algorithm 4.

Algorithm 4 Detecting an Attacker

```
1: procedure ISATTACKER(interface i, int inr)
2:     bool attacker ← false
3:     int Avg ← GetAvg(interface i)          ▷ Returns average rate of interface (i)
4:     int ratio ← SatisfactionRatio(interface i)
5:     bool congested ← IsCongested
6:     if (inr > Avg) && (ratio is low) && (congested = false) then
7:         Block the interface (i)
8:     end if
9: end procedure
```

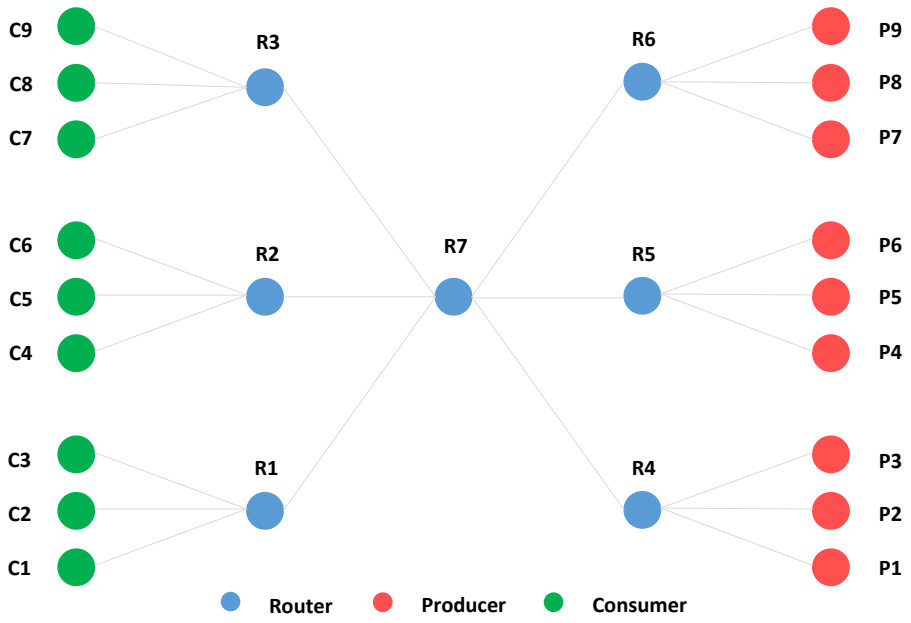

FIGURE 4.2: Simulated topology

4.6 Performance Evaluation

To evaluate the performance of the proposed solution, we conducted extensive simulations using ndnSIM [MAZ17]. ndnSIM is the simulation platform for the NDN paradigm, embedded into ns3. We considered a topology with seven routers, nine producers, and nine consumers, as depicted in figure 4.2. The attacker model used in our simulation continuously sends a huge amount of Interest packets with constant bit-rate (10000 Interests packets/s).

Further, consumers request data as follows: *consumer1* requests data from *producer9* using namespace `/dest9/`. *consumer2* requests data from *producer8* using namespace `/dest8/` and, so on. Simulation details are provided in table 4.1.

First, we start our simulations by showing the effects of a congested network on the overall performance. Afterward, we show that the network congestion leads to false detection of IFA. In addition, we prove that existing solutions relying on the Interest satisfaction ratio are not effective when a network is congested. Finally, we simulate

TABLE 4.1: Simulation settings

Parameter	**Value**	
Simulation time	300s, 150s	
Number of nodes	25	
Data Packet size	1,024Bytes	
CS size	100	
Cache replacement strategy	LRU	
Forwarding Strategy	Best Route	
Average rate	300 ipps	
Rate of legitimate users	100 to 500 ipps	
Rate of attackers	10000 ipps	
Number of Attackers	01	
Link	**Throughput**	**Delay**
Router – Router	100Mbps	10ms
Router – Producer Router – Consumer	10Mbps	10ms
Congested link	<1Mbps	100ms

an attack and show how the network is affected and how our proposed solution solves the false detection decisions due to the congested network.

4.6.1 Simulation of a legitimate traffic

The first simulation scenario that we conducted represents a scenario where all consumers send legitimate traffic. We use its results to compare them with other scenarios. Figure 4.3 depicts the satisfaction ratios associated with every consumers within the network. The results show a satisfaction ratio of nearly 100% for every consumer. That means that nearly every Interest packet sent by consumers was satisfied (Data packet returned).

4.6.2 Simulation of a Congested Network

In this scenario, we simulate a congested network to see its impact on the NDN network. For this, we restricted the throughput of the link between Router1 and Router7 to a maximum of $1Mbps$. Then, we introduced a delay of $100ms$ in this particular link. The simulation results for this scenario are represented according to the number of timed-out Interest packets, the number of dropped packets, and the satisfaction ratio.

Timed-out interest packets

We start by showing how the network congestion impacts the number of timed-out Interest packets. Figure 4.4 and 4.5 highlights the impact of the network congestion on consumers 1, 2 and 3 as they are directly connected to Router1. The results show

FIGURE 4.3: Satisfaction ratios of consumers

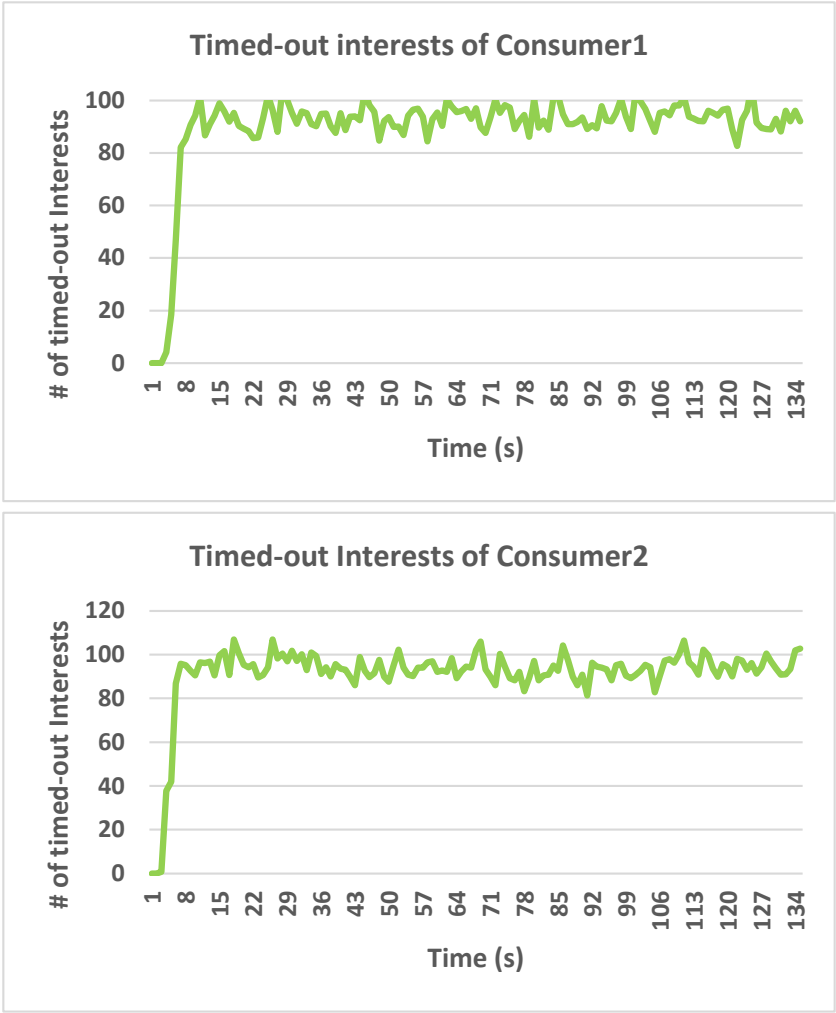

FIGURE 4.4: Number of timed-out Interest packets issued by consumers 1 and 2 (rate=100ipps)

that more than 90% of issued Interest packets timed out, which is a huge number. Further, the effect of network congestion on the PIT of Router1 is depicted in figure 4.6, suggesting that about 90% of pending Interests timed out. This is due to the delay that resulted from the congested link.

Dropped packets

We show how the network congestion impacted Router7 (Core Router). Figure 4.7 depicts the number of Dropped packets by the Core Router. It represents a considerable number compared to the total number of packets passing through the Core Router (1600 packets/s). The results show that 40% of the packets were dropped. All the network packets (Interests and Data packets) are passing through the core router, which means that 40% of whole network traffic was dropped due to the congested link.

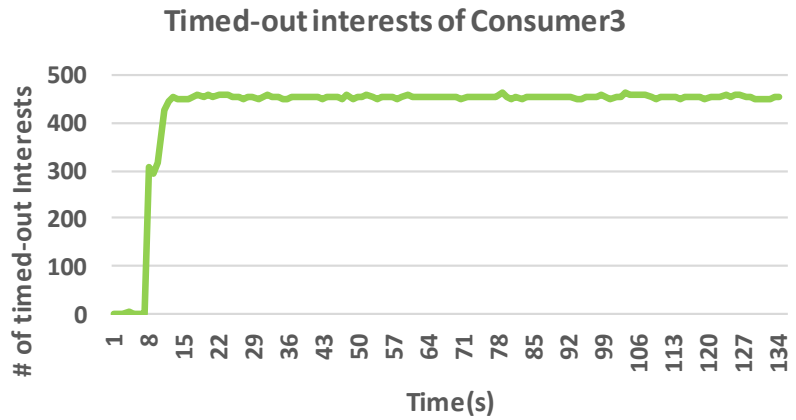

FIGURE 4.5: Number of timed-out Interest packets issued by Consumer 3 (rate=500ipps)

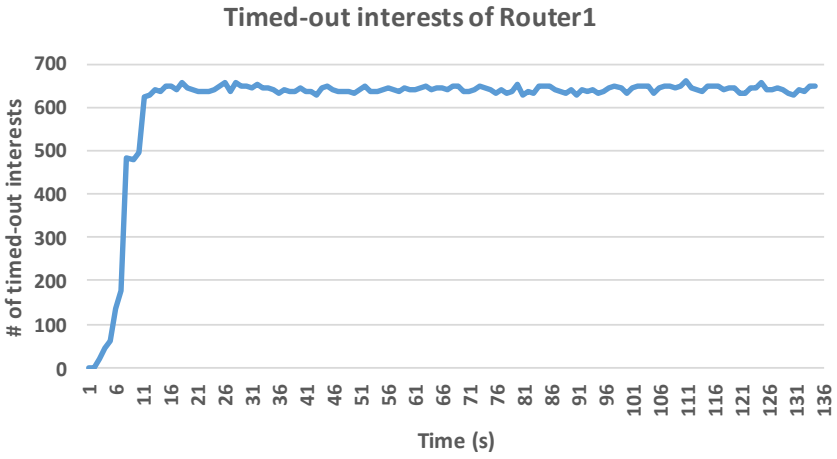

FIGURE 4.6: Number of timed-out Interest packets recorded by Router1

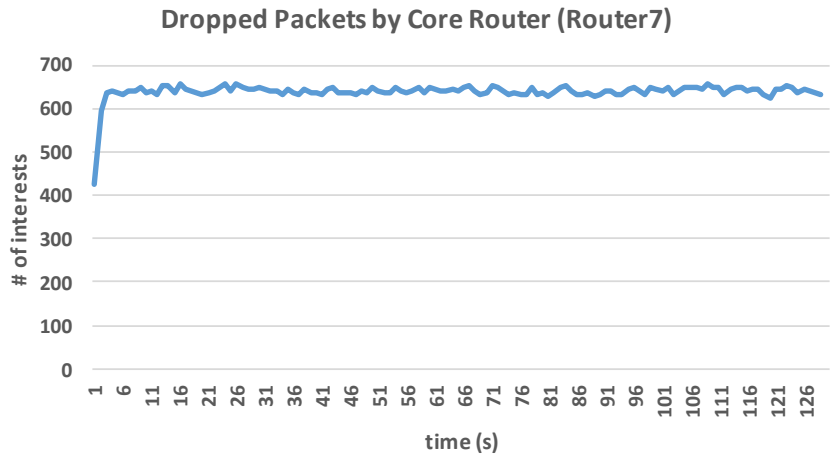

FIGURE 4.7: Number of dropped packets by Router7 (Core Router)

Satisfaction ratio

In this subsection, we analyze the impacts of a congested network on the satisfaction ratio. We prove that the solutions that rely on the satisfaction ratio may give false results when detecting IFA.

Figure 4.8 compares the number of Interest packets sent with Data packets received of the three consumers who are directly connected with Router1. We can see that a congested network can significantly decrease the satisfaction ratios of a router's interfaces. For example, Consumer1 who requests 100 *ipps* is receiving an average of only 5 Data packets/s which means a satisfaction ratio of only 5%.

Figure 4.9 show the satisfaction ratio values associated with Consumers 1,2 and 3. The results show that relying only on the satisfaction ratio when detecting IFA may give false results when a network is congested.

In this subsection, we showed how a congested network impacts the statistics of a router and may lead to mistakenly detecting IFA. The other nodes in the network were not affected by the network congestion as their traffic do no pass through the affected link.

4.6.3 Our Solution During IFA

To test our solution against IFA, we simulated an attacking scenario where Consumer3 sends 10000 forged ipps. The malicious node uses the namespace `/dest7/attack` to request data from Producer7 who satisfies Interest packets under the namespace `/dest7/legit`. The goal is that the Interest packets arrive at destination (Producer7) without being satisfied. They will stay in PIT tables until they time out.

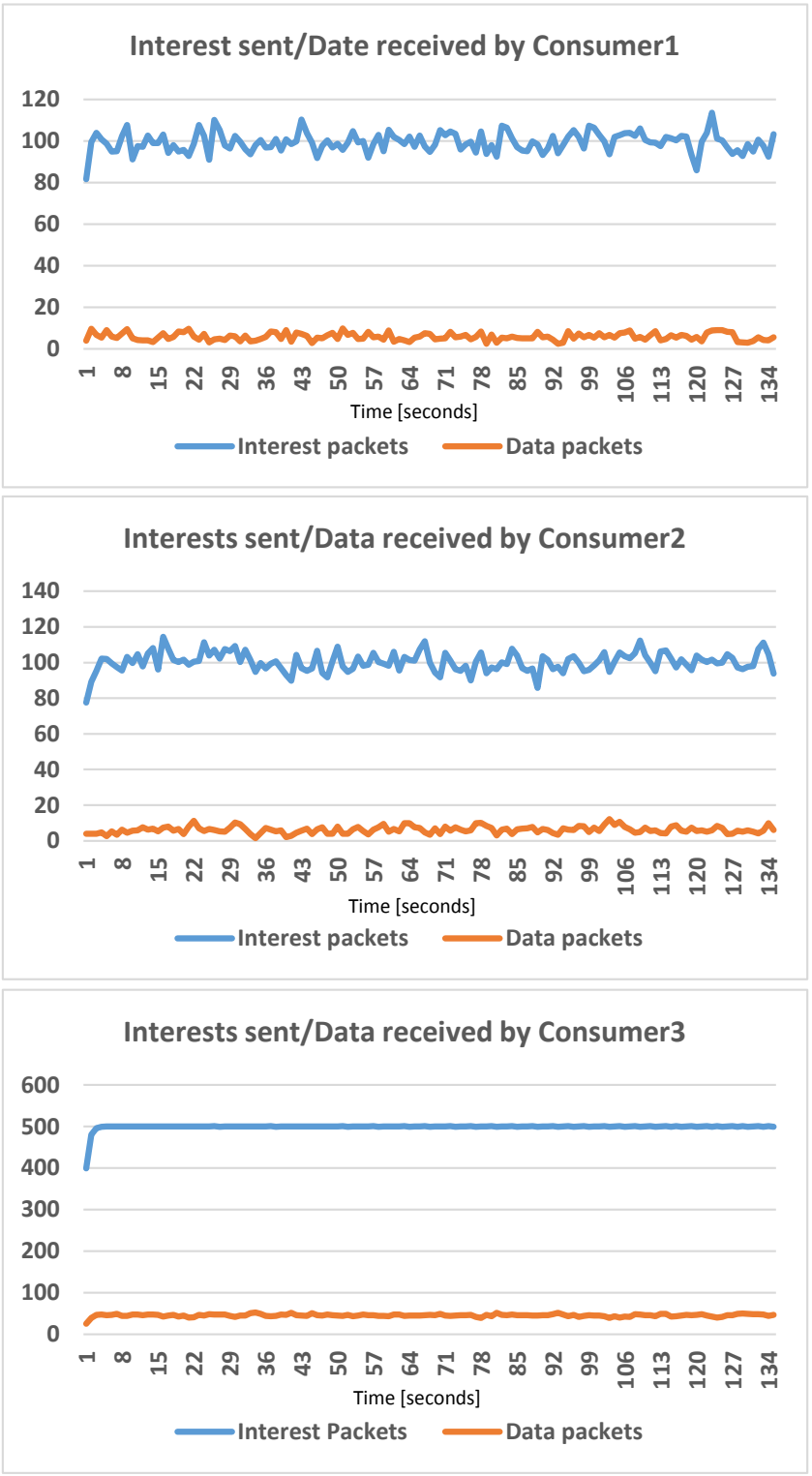

FIGURE 4.8: Number of Interest and Data packets sent and received by Consumers 1,2 and 3

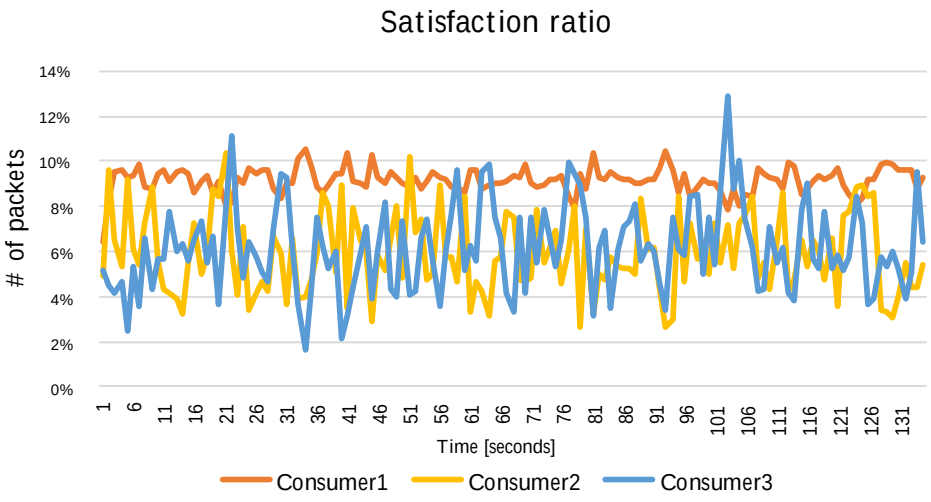

FIGURE 4.9: Satisfaction ratios of consumers 1, 2 and 3 during the congested network scenario

We recorded a considerable number of NACK packets during this attacking scenario. It is the result of non-existing data. The producers, when they receive an Interest packet for non-existing data, they respond with NACK packets. Figure 4.10 shows the number of NACK packets received by the Router1 (Attacker's Gateway). The number of NACK packets starts to rise after the attack is launched (at $t = 10$ second). The number of received NACK packets stabilized at 10000 packets/s, which equals the number of Interest packets sent by the attacker. However, when the attacker is mitigated (blocked), the number of NACK packets drops down to 0 NACK packet/s after 3 seconds.

The other parameter that changed during the attack is the timed-out Interest packets. Figure 4.11 shows the number of timed-out Interest packets recorded by Router1. We also noted that some legitimate Interest packets timed out. It is the result of the malicious traffic that burdens the network. Data packets take a longer time to return, which results in timed-out pending Interest packets. However, the number is negligible compared to the number of timed-out Interest packets related to the attacker. The number of timed-out Interest packets recorded by Router1 starts to rise from $t = 10s$ (Attack launch) until it stabilizes under 10000 timed-out $ipps$. When the attacker is mitigated ($t = 30s$), this number starts to go down until it reaches 0 $ipps$ after 3 seconds.

Figures 4.10 and 4.11 shows that the number of NACK and timed-out interest packets are identical. It was a confirmation for our solution that the network is not congested, which helped to stop the attack and mitigate its initiator.

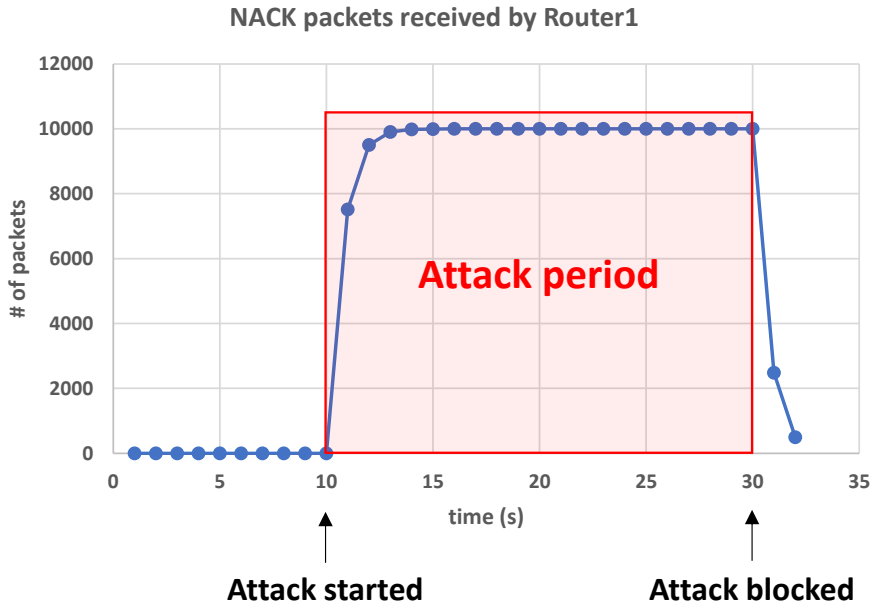

FIGURE 4.10: Number of NACK packets received by Router1

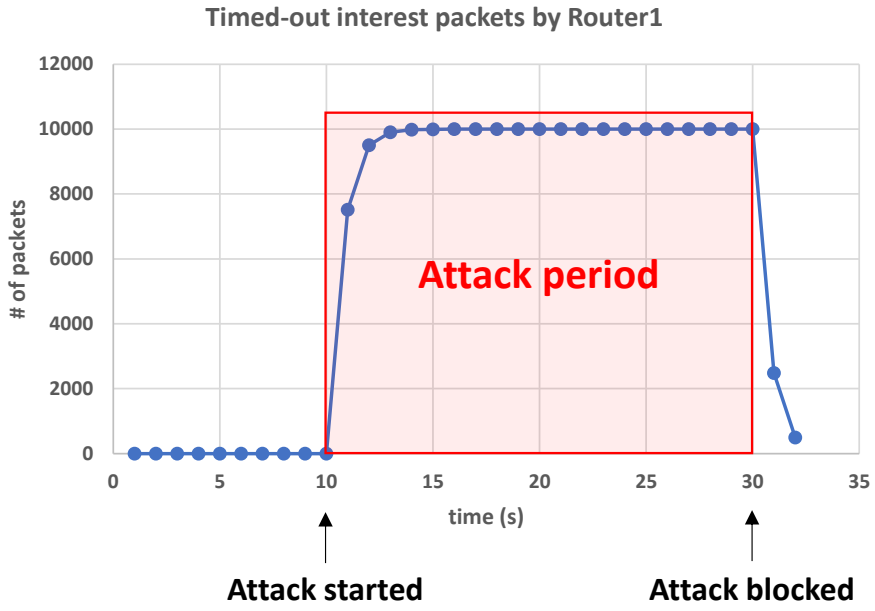

FIGURE 4.11: Number of timed-out Interest packets received by Router1

4.7 Summary

In this chapter, we presented a solution that takes the network congestion as a parameter to detect and mitigate IFA to avoid false alarms. We showed, through simulation results, that network congestion may lead routers to false-positive detections. It could also consider legitimate traffic as malicious. Finally, we showed how this solution mitigated an attack within a congested network. In the next chapter, we will present our second solution against IFA, named MSIDN.

Chapter 5

MSIDN: Mitigation of Sophisticated Interest Flooding-based DDoS Attacks in Named Data Networking

Various mitigation solutions exist in the literature. However, legitimate users and their traffic are usually affected by these solutions. To face this problem, we present in this chapter a lightweight mechanism capable of mitigating sophisticated interest flooding-based DoS and Distributed (DDoS) attacks in NDN. Our solution, named MSIDN, aims to mitigate attacks at the source of communication without affecting legitimate users.

5.1 Design Overview

MSIDN relies on the collaboration of data producers and network routers to mitigate different types of Interest flooding-based (D)DoS attacks. The proposed solution classifies routers into three categories:

1. Lower-edge routers: The Lower-edge routers connect end-nodes (consumers) to the network.
2. Upper-edge routers: The Upper-edge routers are responsible for connecting data producers to the NDN network.
3. Core routers connect the network routers.

Figure 5.1 illustrates the above-mentioned router categories in our system.

5.2 Interest flooding-based DDoS attack scenarios

Without loss of generality, this solution considers the following two DDoS attack scenarios.

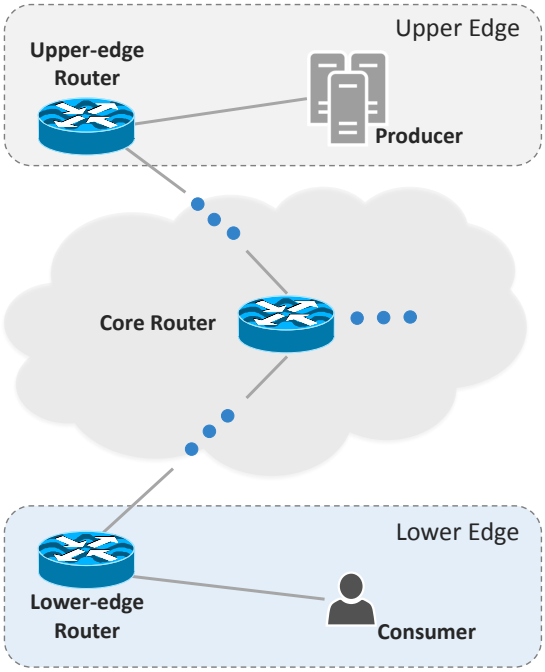

FIGURE 5.1: Architecture of the proposed solution

5.2.1 Botnet with high per-device sending request rate

The first distributed attacking scenario that we consider is a DDoS with an aggressive sending rate. In this scenario, an attacker who controls a botnet will employ the controlled bots to target a producer with a high sending rate. Figure 5.2 illustrates this scenario of DDoS attacks.

5.2.2 Botnet with normal per-device request rate

In the second scenario, botnet members send a small number of requests per second. Assuming that an attacker who controls a very large botnet (e.g., IoT botnet) wants to take down a specific server. There is no need for the commanded devices to send a large number of Interest packets to inflict damage to the target. For example, a botnet of a hundred thousand devices sending each, only ten Interest packets per second will flood a target with a total number of one million Interest packets per second. In this scenario, detecting attacks is much harder than in the first scenario. It is hard for routers to detect malicious or controlled devices because of their low sending rates. Figure 5.3 depicts this scenario of DDoS attacks.

5.3 Interest flooding-based DDoS Traffic Classification

To understand attackers' behavior, we categorize the network traffic using three parameters: the satisfaction ratio. The second parameter is the speed of incoming interest packets. And the third parameter is the number of timed-out interest packets.

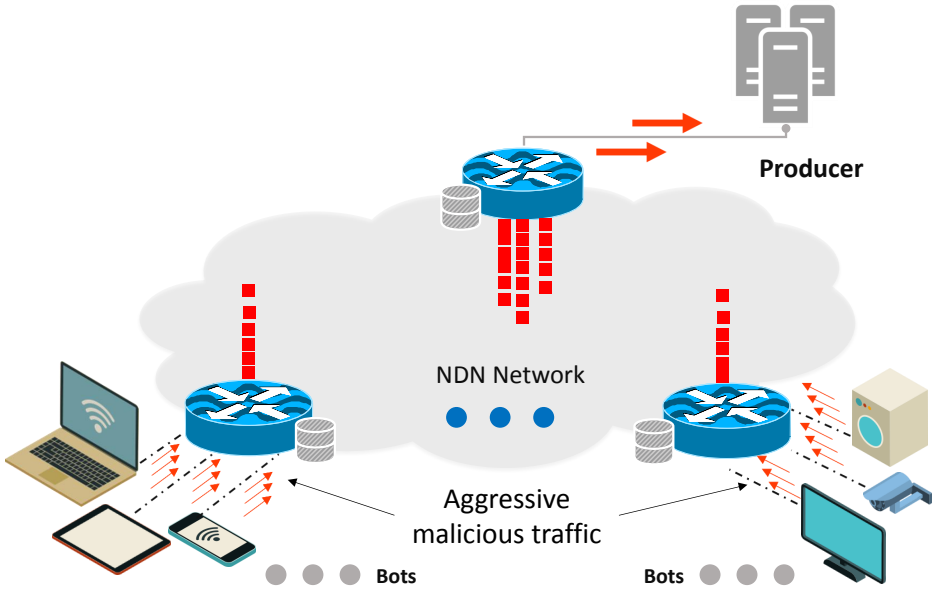

FIGURE 5.2: In this DDoS scenario, the attack initiator orders the controlled bots to send aggressive traffic to the target (Sending Interest packets with a high sending rate).

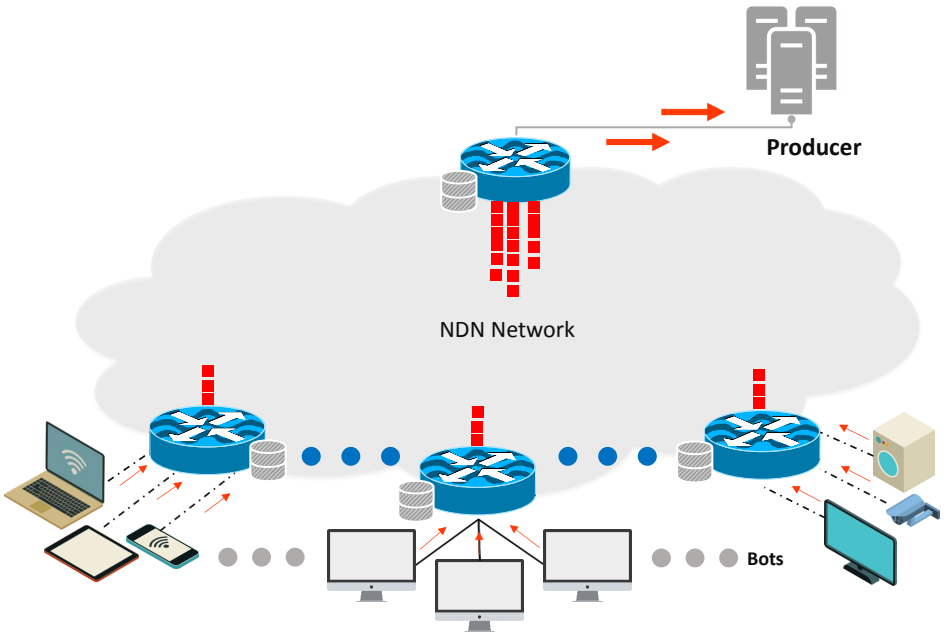

FIGURE 5.3: In this DDoS scenario, the attack initiator, who is in control of a very large botnet, orders the bots to target a victim with normal traffic (Sending Interest packets with a regular sending rate).

TABLE 5.1: Interest flooding-based DDoS traffic classification

	Normal rate scenario			Aggressive rate scenario		
	Satisfaction ratio	Incoming rate	Timed-out interests	Satisfaction ratio	Incoming rate	Timed-out interests
Static-valid data requests	High	Low to normal	Low	Normal	High	Low to normal
Dynamic-valid data requests	Normal to high	Low to normal	Low	Normal	High to very high	Low to normal
Invalid data requests	Low	Low to normal	High	Very low	Very high	Very high

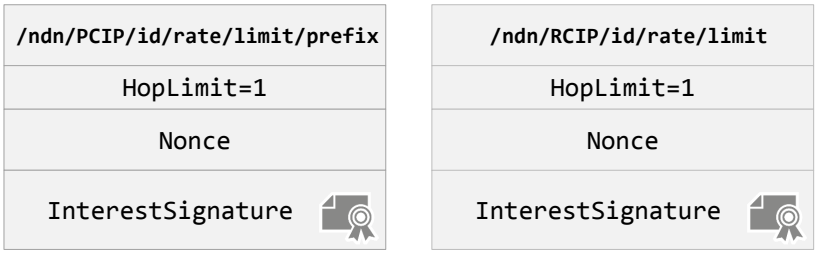

FIGURE 5.4: Control Interest Packets: PCIP (Left) and RCIP (Right)

These three parameters are compared against all data request types mentioned in section 3.3 as well as our two attacking scenarios mentioned in section 5.2. Table 5.1 summarizes this network traffic classification.

5.4 Control Interest Packet

The proposed solution uses signed Interest packets. Producers and network routers use them to carry specific information. These information will then be used by routers to mitigate an ongoing attack. The solution uses two signed interest packets that we named, *Producer-based Control Interest Packet* or PCIP, and *Router-based Control Interest Packet* or RCIP.

5.4.1 Producer-based Control Interest Packet (PCIP)

Every data producer can send a PCIP when needed. PCIP operates under the namespace `/ndn/PCIP/` and has a name similar to `/ndn/PCIP/id/rate/limit/prefix` where `id` is the sender's identifier, `rate` is a keyword, `limit` is an operation, and `prefix` is a namespace that the producer sends. Every PCIP is cryptographically signed by its sender and has a *HopLimit* parameter equals to 1.

5.4.2 Router-based Control Interest Packet (RCIP)

RCIP is created and signed by network routers. RCIP operates under the namespace `/ndn/RCIP/`. The name of an RCIP is identical to `/ndn/RCIP/id/rate/limit/`, and its *HopLimit* parameter is equal to 1. Every RCIP is cryptographically signed by its sender. Figure 5.4 details the structure of PCIP and RCIP.

5.5 Hop-by-hop Signing and Verification

MSIDN adopts a hop by hop signing and verification process. It means that every PCIP and RCIP exchanged in the network is re-signed by the router that sends it. The signing of the packets is carried out for two reasons:

1. It is faster. We assume that every router in our system has the public keys of its neighbors. The signature verification process is local. There is no need to fetch the public key from the originator of a PCIP/RCIP when received (i.e., Interest packet's *KeyLocator*). This process ensures a fast verification process.
2. When a producer sends a PCIP, the receiving router needs to get the producer's public key to authenticate the received PCIP. This process can add additional overhead to the originating producer when it is under attack. By choosing a hop by hop approach, we reduce the charge of producers.

Hop by hop signing and verification process is guaranteed by the `id` component of the name. Every network router and data producer has a unique `id`. Data producers operate under the namespaces `/ndn/PCIP/id` whereas network routers operate under the namespace `/ndn/RCIP/id` and `/ndn/PCIP/id`. We build a trust schema based on these two namespaces. Every node of the system has full control over its namespace, and only this node can sign and send packets under the mentioned namespace.

For example, if the `id` of an Upper-edge router is *UR-1*, this router has full control on the namespaces `/ndn/PCIP/UR-1`, and `/ndn/RCIP/UR-1`. It is the only node that can create and send packets under these two namespaces. Only Upper-edge and core routers can operate under two namespaces. Data producers generate packets only on the PCIP namespace. On the other hand, lower-edge routers are permitted to generate neither PCIP nor RCIP packets. For every namespace, nodes generate a pair of public and private keys necessary for signing packets under these namespaces. Besides, routers possess the public keys of their neighbors so they can verify the authenticity of the received PCIP and RCIP packets instantly. Figure 5.5 illustrate our hop-by-hop signing/verification process.

5.6 Producer-based (D)DoS mitigation Process

When a data producer is overwhelmed by requests, it sends a PCIP to the upper-edge router. The PCIP's name will contain the affected namespace/service. If a producer works under the namespace `ProducerNamespace/`, the prefix that a producer sends in a PCIP could be the whole producer's namespace or just a specific service. For example, a producer could send `ProducerNamespace/web`, `ProducerNamespace/app1`, or any other affected service.

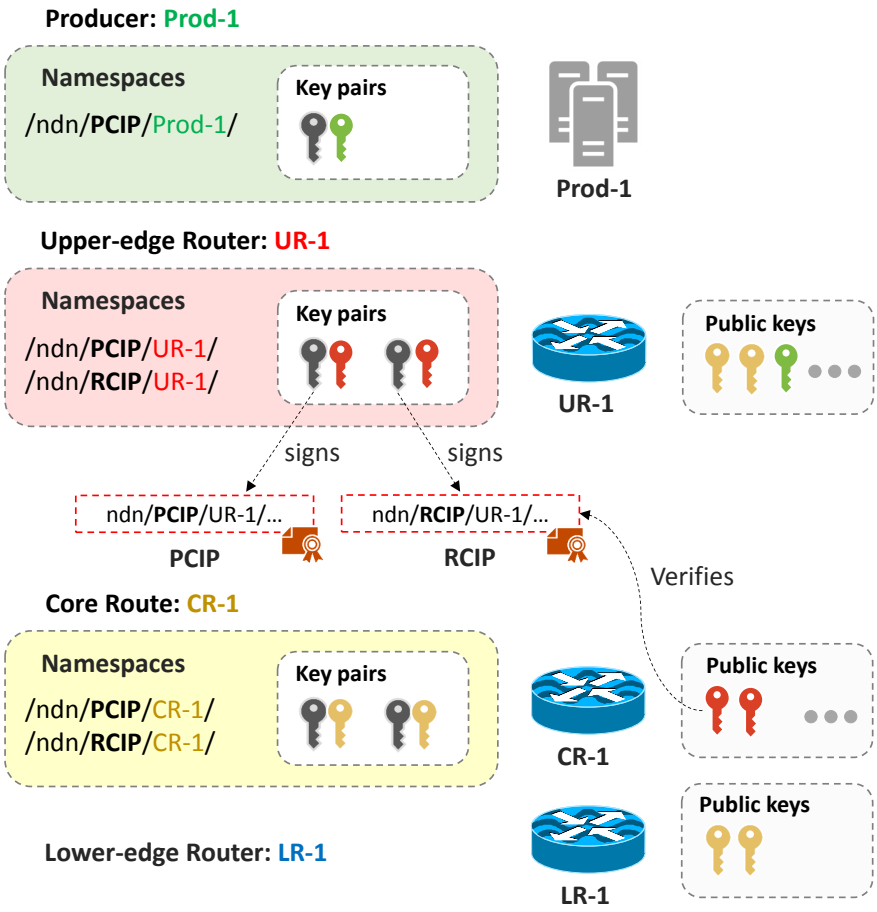

FIGURE 5.5: Overview of MSIDN signing/verification process

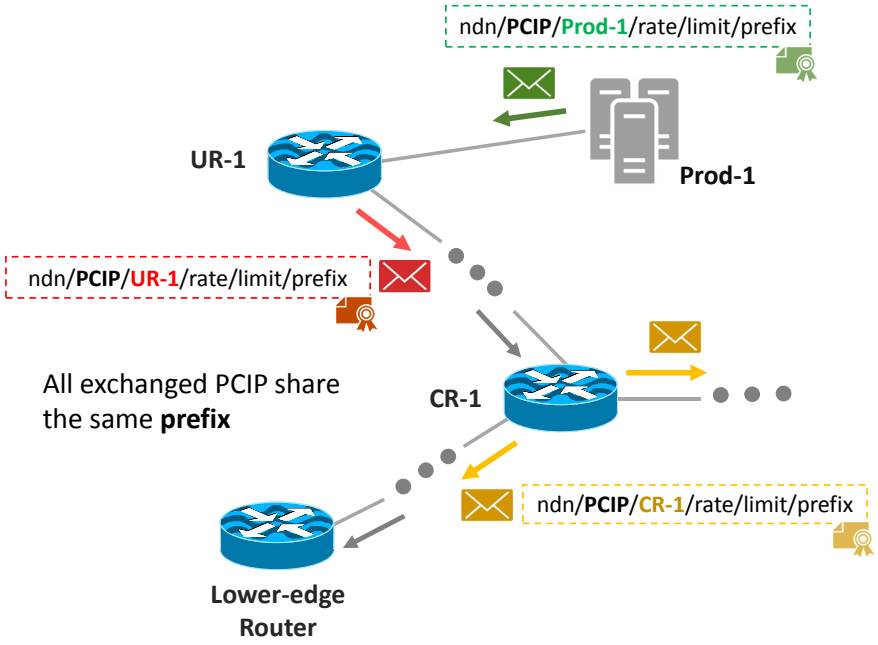

FIGURE 5.6: PCIP Forwarding

When an upper-edge router receives a PCIP, it first verifies the authenticity of the PCIP by verifying the producer's signature. If the signature is verified successfully, the router will limit the traffic going to the prefix mentioned in the PCIP. In other words, the router will apply a limit on every Interest packet that has a name similar to the prefix sent in the PCIP. After controlling the traffic going to the affected producer, the upper-edge router creates an identical copy of the received PCIP. It adds its `id` to the name and signs it with its private key before sending it to the routers directly connected to it. We summarize the above-discussed process in Algorithm 5.

When a core router receives a PCIP, it applies the same process as upper-edge routers. First, it verifies the authenticity of the received PCIP with the sender's public key. Then, it applies a rate-limiting on the traffic destined to the received `prefix`. Once the traffic is controlled, the router creates a new copy of PCIP with its `id` and signs it. Finally, the newly created PCIP is forwarded to the neighboring routers (see Fig. 5.6).

When a PCIP reaches a lower-edge router, i.e., its final destination, the router limits the traffic according to the received `prefix`, the same way as the upper-edge and core routers.

The producer-based trigger process relieves data producers from the number of requests they may receive when they are under attack. It also reduces the charge of the network.

Algorithm 5 ProducerControl

```
Begin
    Input Interface int, Interest pcip
    Output Interest i
 1: if Verify(pcip) == True then                    ▷ Interest packet signature is valid
 2:     if CoreRouter OR UpperedgeRouter then
 3:         ControlTraffic (int, prefix)
 4:         CreateInterest (i)
 5:         i.name ← "/ndn/PCIP/id/rate/limit/prefix"
 6:         i.hoplimit ← 1
 7:         Sign (i)
 8:         Send Interest (i) to all interface except (int)
 9:     else
10:         ControlTraffic (int, prefix)    ▷ Limit forwarding Interest packets with name =
    "prefix" to interface (int)
11:     end if
12: else
13:     Drop(pcip)
14: end if
End
```

5.7 Router-based (D)DoS Mitigation Process

Network routers use RCIPs to send additional information to other routers. The impact of a DDoS attack on a network could be enormous. Routers could suffer a lot from the traffic that they receive during a DDoS attack. And the effects may last longer, even after the attack is contained. In these situations, routers exchange RCIPs to regulate network traffic. The name of an RCIP contains the `id` of the sending router and the `/rate/limit` command.

RCIP is a one-hop packet which means that it is not forwarded. Every router sends its own RCIP. If a router wants to regulate the traffic that arrives from a specific interface, it sends an RCIP to the router connected to this interface. The recipient will then reduce the traffic going to this interface. For a more sophisticated solution, a router could send the amount of traffic that it can handle at a specific time. Figure 5.7 shows an example of routers exchanging RCIPs.

Algorithm 6 RouterControl

```
Begin
    Input Interface int, Interest rcip
 1: if Verify(rcip) == True then                    ▷ Interest packet signature is valid
 2:     ControlTraffic (int)          ▷ Limit forwarding all interest packets to interface (int)
 3: else
 4:     Drop(rcip)
 5: end if
End
```

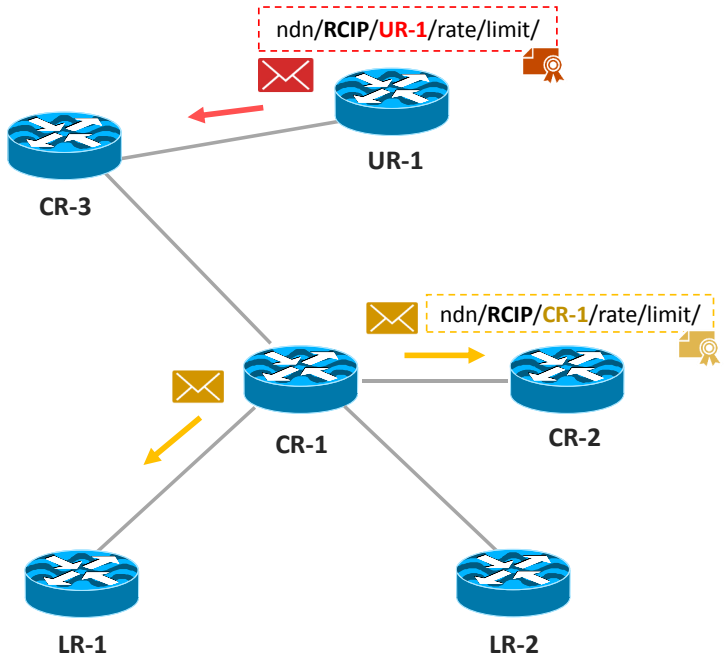

FIGURE 5.7: RCIP Forwarding

5.8 Blocking Malicious Nodes

In our proposed scheme, lower-edge routers are responsible for blocking malicious nodes at the source. According to the network-based and producer-based information that they receive, in addition to the information gathered locally, lower-edge routers decide whether to block or not a final node. The Traffic classification in Table 5.1 helps lower-edge routers to decide a convenient action.

The lower-edge routers should also differentiate between a consumer who is performing an aggressive attack and a consumer who is legitimately sending traffic with a high request rate (e.g., watching a high-quality live video stream). Lower-edge routers classify consumers according to their behavior: normal behavior, suspicious behavior, or harmful behavior.

Lower-edge routers classify consumers to have suspicious behavior if one of these conditions are true:

- Low packet satisfaction ratio and a high number of timed-out pending Interest packets.
- Requested data from a producer that sent a PCIP.

On the other hand, lower-edge routers classify a consumer as a legitimate user when the satisfaction ratio is high, the number of timed-out Interest packets is low, and no PCIP involving this consumer was received.

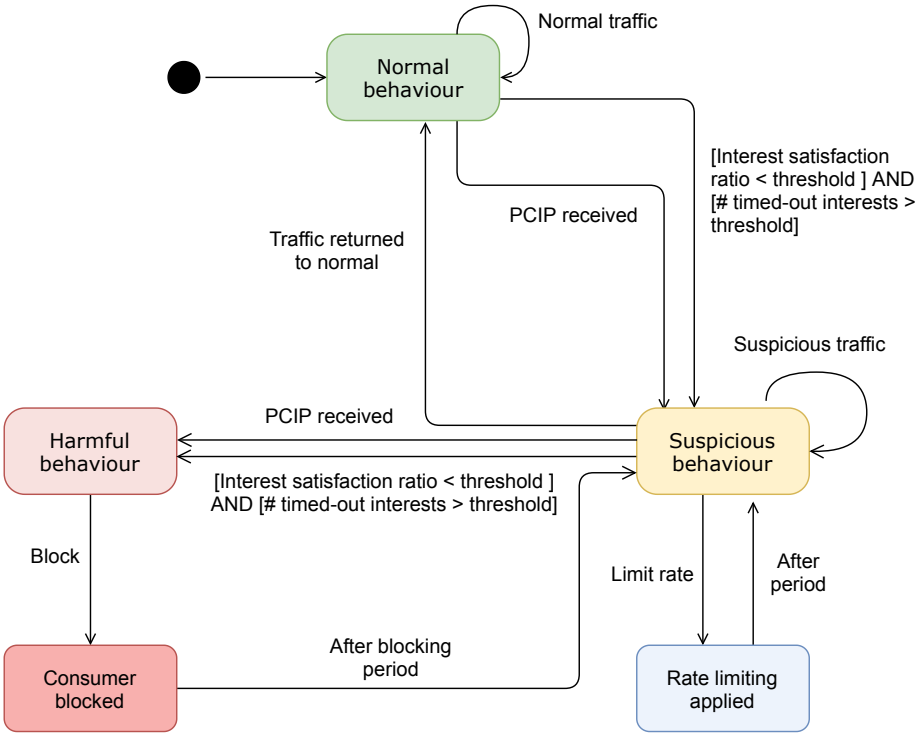

FIGURE 5.8: Consumer behavior classification state chart

Lastly, a consumer is considered to have harmful behavior when every condition of a suspicious profile is satisfied. Routers classify every consumer with harmful behavior as malicious and block them. The state diagram in Figure 5.8 illustrates the different states of a consumer. Algorithm 7 details the classification process.

5.9 Performance Evaluation

We evaluate the performance of MSIDN using ndnSIM. In our simulations, we used a topology with three core routers connecting three upper-edge routers and three lower-edge routers. Upper-edge routers are connected with three producers. On the other hand, lower-edge routers are connecting six consumers to the network. Our network topology is shown in Fig. 5.9). The simulation parameters used are listed in Table 5.2.

5.9.1 Impacts of IFA on the network

In this section, we study the impact of IFA on the network. We conducted several IFA scenarios. In the first scenario, the attackers use valid data requests. In the second scenario, attackers use invalid data requests, and in the third scenario, the attackers use a mix of valid and invalid requests to perform an IFA. Attackers use different sending rates. The network contains four attackers (as shown in Fig. 5.9): C2, C3, C4, and C6. These attackers target one victim, which is P3 in this scenario.

Algorithm 7 Consumer Classification

Begin

for each i **do**
    Normal(int_i) ▷ all interfaces (int_i) are initially classified as Normal
end for
for each i **do**
    if SatisfactionRatio(int_i)<threshold AND
TimedoutIntrest(int_i)>threshold **then**
        Suspicious(int_i) ▷ Classify interface (int) as suspicious
        RateLimit(int_i) ▷ Apply rate limiting on interface (int)
    end if
end for
if PCIP received **then**
    for each int_i requested "prefix" **do**
        if IsSuspicious(int_i) **then**
            Harmful(int_i) ▷ Classify interface (int) as harmful
            Block(int_i) ▷ Block interface (int) for a period
        else
            Suspicious(int_i)
            RateLimit(int_i)
        end if
    end for
end if

End

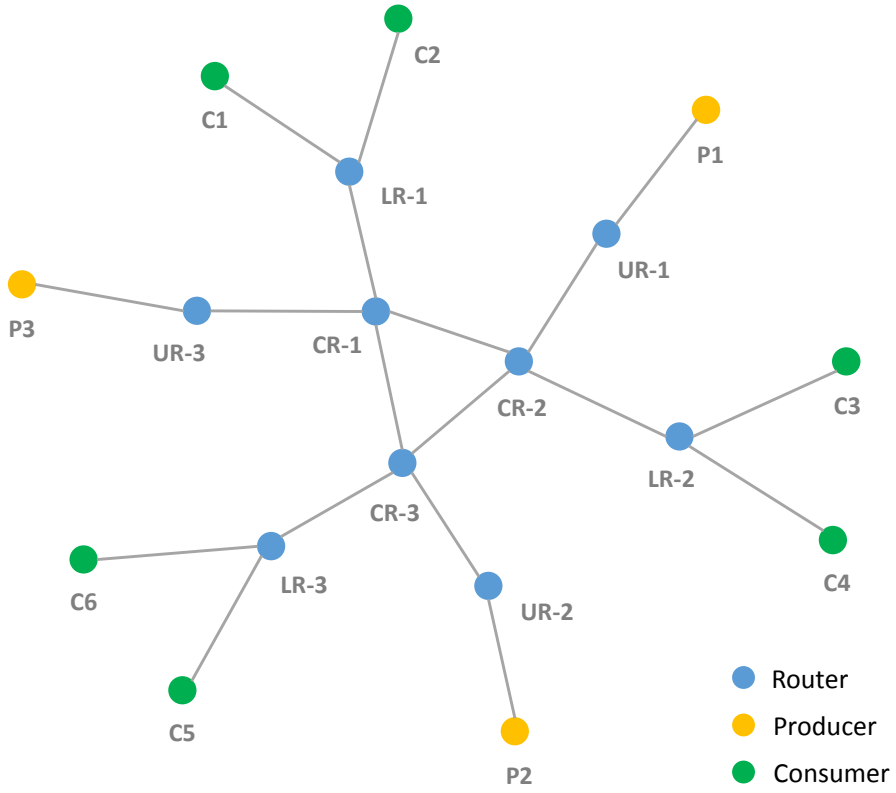

FIGURE 5.9: Network topology used in simulation

TABLE 5.2: MSIDN simulation settings

Parameter	**Value**
Data packet size	1,024 Bytes
CS size	100
Cache replacement strategy	LRU
Forwarding strategy	Best route
Rate of legitimate users	100 interests/sec
Rate of attackers	1500, 3000, 5000 and 10000 interests/sec
Number of attackers	04
Link	**Throughput**
Core router – Core router	1Gbps
Core router - Upper-edge router Core router - Lower-edge router	100Mbps
Upper-edge router - Producer	100Mbps
Lower-edge router - Consumer	10Mbps

Scenario 1.1: valid data requests

In this scenario, the attackers C3 and C4 target the producer P3 with 1500 valid ipps. Furthermore, C2 and C6 attack P3 with sending rate that equals 3000 valid ipps. Whereas legitimate users C1 and C5 send 100 ipps to P2 and P1, respectively.

Figure 5.10 shows the satisfaction ratios of all Lower-edge routers (LR) (consumers' gateway) and the Upper-edge router (UR), UR-3 (target's gateway). As depicted in this figure, LR-1 was not affected by the network traffic. It recorded a satisfaction ratio of nearly 100%, as for LR-3. On the other hand, LR-2 was clearly affected by the malicious traffic. It reached a satisfaction ratio of only 46%. It can be explained by the distance between this router and the target P3. LR-2 is the farthest router to P3. That is why the traffic was further affected as compared to LR-1 and LR-3.

The second parameter that we compared during our simulation is the number of dropped layer2 packets (link-layer packets). Figure 5.11 shows the number of dropped packets by URs. The results show a considerable number of dropped packets. For instance, LR-3 dropped an average of 1800 packets/sec.

Scenario 1.2: valid data requests

The results of the second scenario with valid requests are shown in Fig. 5.12. In this scenario, attackers C3 and C4 target P3 with 3000 ipps. C2 and C6 attack P3 with 5000 ipps. The results show that network routers were affected by malicious traffic. The routers were more affected than they were in the previous scenario. The satisfaction ratio of these routers decreased significantly. It can be explained by the number of Interest packets that the attackers sent. Producer P3 couldn't satisfy all the incoming requests. Because of that, the number of satisfied Interest packets decreased. The

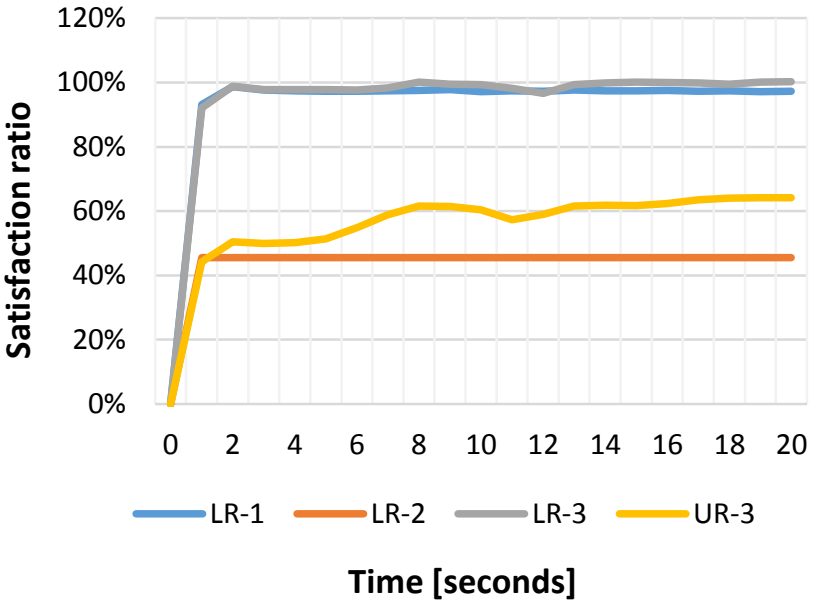

FIGURE 5.10: Satisfaction ratios of all Lower-edge routers and the Upper-edge router connected with the target P3.
Settings: Attack rate equals 1500 and 3000 ipps. Rate of legitimate users equals 100 ipps.

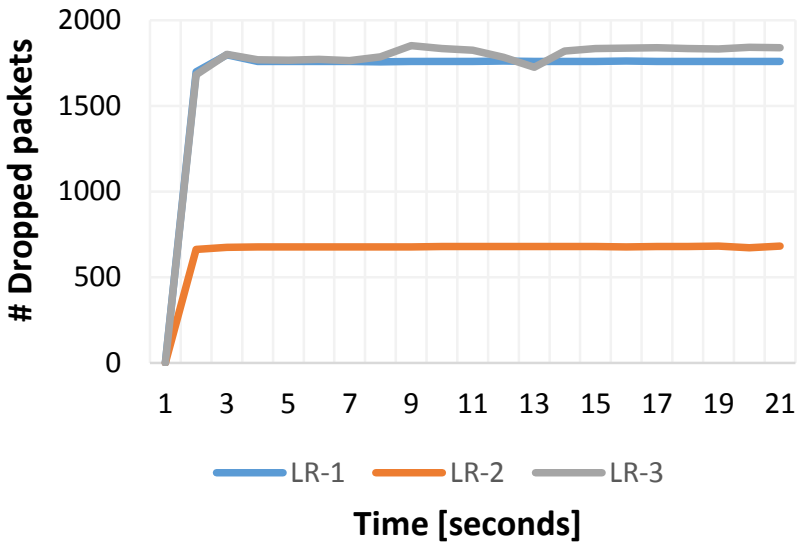

FIGURE 5.11: Number of dropped packets by Lower-edge routers.
Settings: Attack rate equals 1500 and 3000 ipps. Rate of legitimate users equals set to 100 ipps.

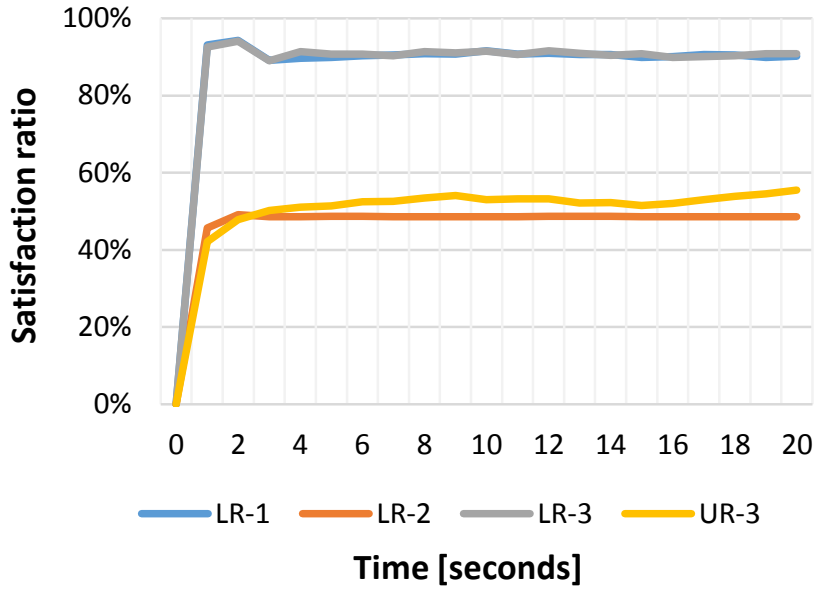

FIGURE 5.12: Satisfaction ratios of all lower-edge routers and the Upper-edge UR-3.
Settings: Attack rate equals 3000 and 5000 ipps. Rate of legitimate users is set to 100 ipps.

attack affected two routers. The first router is UR-3, which is the P3's gateway. The second router is LR-2, which received the highest number of requests.

The second attacking scenario also affected the number of dropped packets. Figure 5.13 shows the number of registered dropped packets by lower-edge routers. It is the result of the incoming malicious traffic. The network links couldn't handle the overhead caused by the malicious traffic, which led routers to drop packets.

Scenario 2.1: invalid data requests

We aim to study the effects of an IFA with invalid requests on legitimate users and the targeted producer. In essence, we overwhelmed PIT tables of network routers, and the producer P3, to examine its effects on legitimate users. In this first attacking scenario, the attackers C2 and C6 target the producer P3 with 3000 invalid ipps. The attackers C3 and C4 attack P3 with 1500 invalid ipps. In this scenario, we recorded the number of NACK packets. Figure 5.14 shows the number of NACK packets received by LRs and UR-3. The number of NACK packets is similar to the number of invalid Interest packets sent by attackers. The producer P3 responded to these invalid interests with NACK packets.

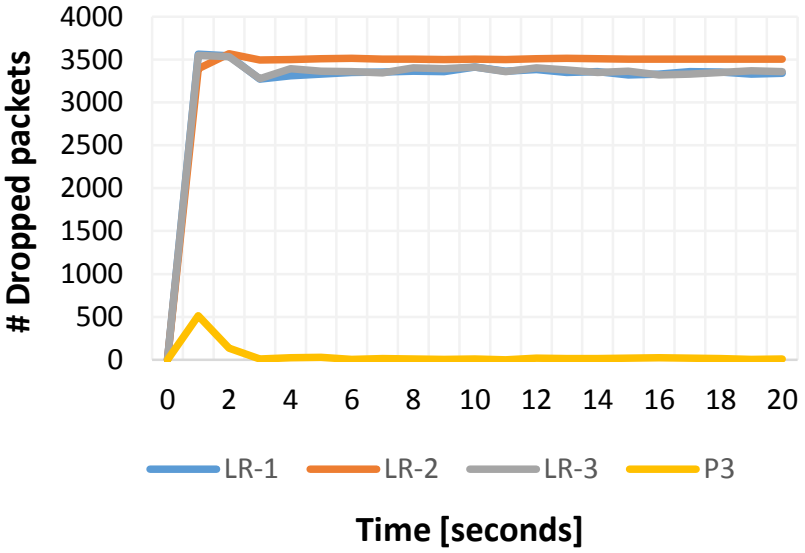

FIGURE 5.13: Number of dropped packets by lower-edge routers and the producer P3.
Settings: Attackers rate equals 3000 and 5000 ipps. Rate of legitimate users is set to 100 ipps

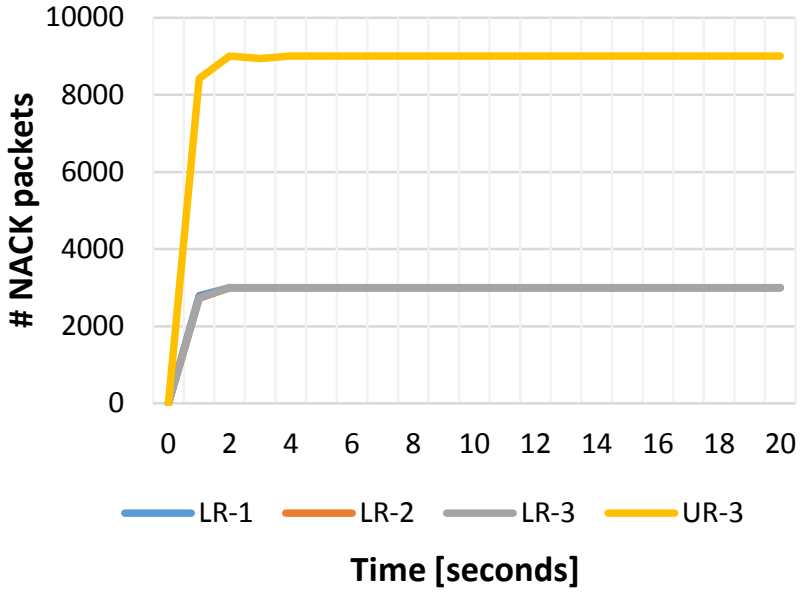

FIGURE 5.14: Number of NACK packets.
Settings: Attack rate equals 1500 and 3000 ipps. Rate of legitimate users is set to 100 ipps.

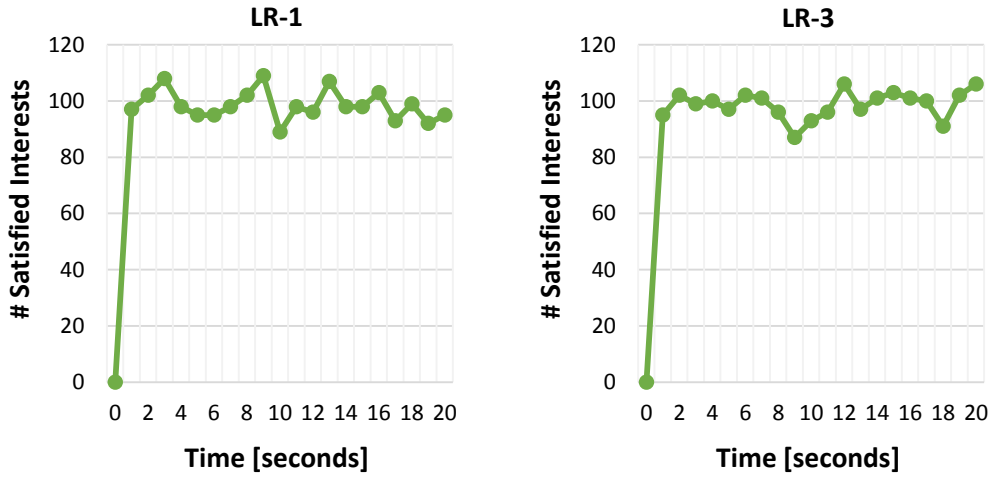

FIGURE 5.15: Number of satisfied Interest packets.
Settings: Rate of attackers: 5000 and 10000 ipps. Rate of legitimate users equals 100 ipps

Scenario 2.2: invalid data requests

The second attacking scenario simulated used 5000 and 10000 invalid ipps. The attackers C2 and C6 send 10000 ipps, whereas C3 and C4 used an attacking rate of 5000 ipps. Figure 5.15 shows that legitimate users connected with LR-1 and LR-3 were not affected. All the Interest packets were satisfied even with a large amount of malicious traffic. Like the previous scenarios, we also noticed a considerable number of NACK packets. The number of registered NACK packets in this scenario equals to the number of invalid Interest packets, which is 30000 packets/sec.

Scenario 3.1: mixed valid/invalid data requests

In these tests, we used mixed requests to perform an IFA. In the first scenario, C2 and C6 attack the producer P3 with 3000 valid ipps. On the other hand, C3 and C4 target P3 with 1500 invalid ipps. Figure 5.16 shows the recorded satisfaction ratio results. It shows that the router LR-2 had a satisfaction ratio of 0%. It is the result of the invalid malicious traffic sent by C3 and C4. All the outbound traffic of LR-2 was not satisfied, which explains why the satisfaction ratio of LR-2 equals 0%.

We also recorded many dropped packets during this attacking scenario, as shown in Fig. 5.17. The routers LR-1 and LR-3 dropped an average of 1700 packets/sec.

Scenario 3.2: mixed valid/invalid data requests

In this scenario, C2 and C6 send 5000 valid ipps. C3 and C4 attack P3 with 3000 invalid ipps. Figure 5.18 shows the satisfaction ratios of lower-edge routers and UR-3. LR-2 registered a satisfaction ratio of 0%. It is the result of the invalid traffic

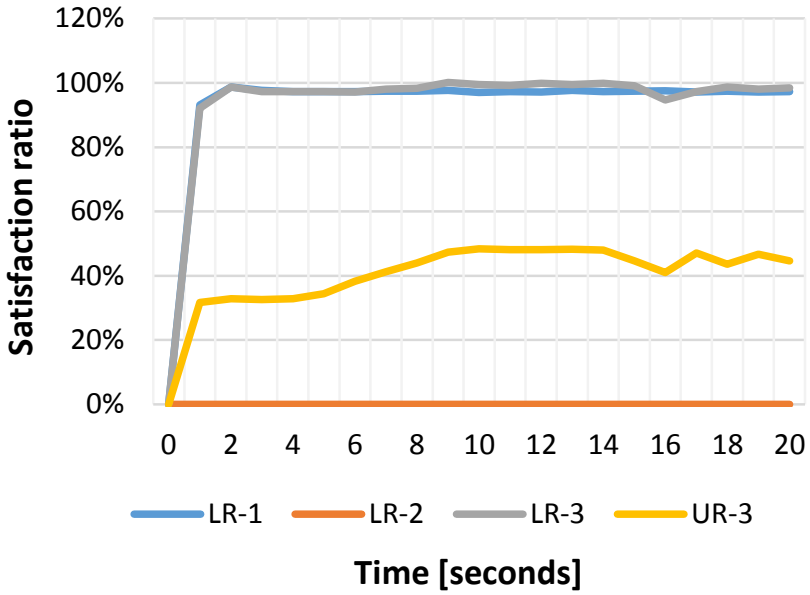

FIGURE 5.16: Satisfaction ratios.
Settings: Attackers rate equals 1500 of invalid and 3000 of valid ipps. Rate of legitimate users is set to 100 ipps

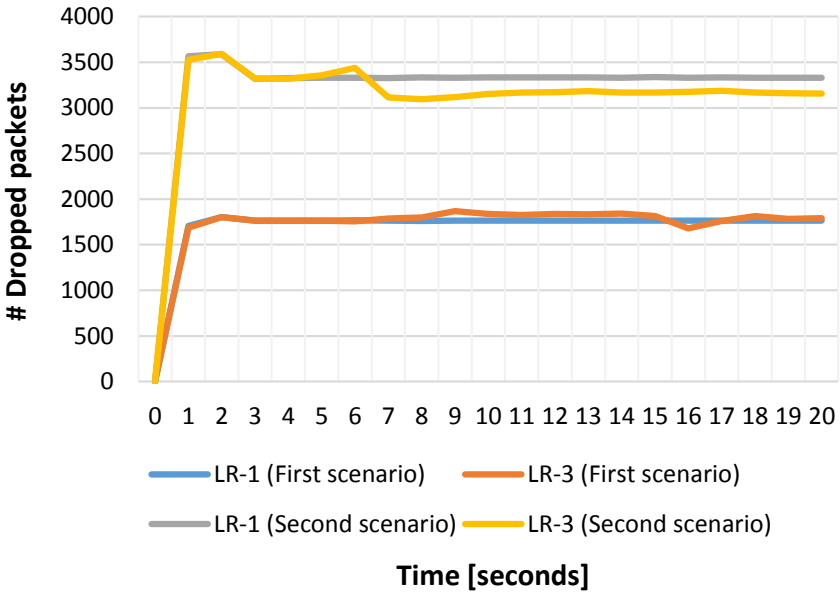

FIGURE 5.17: Number of dropped packets.
Settings: Rate of attackers: 1500 of valid and 3000 invalid ipps (scenario 1). 3000 of valid and 5000 invalid ipps (scenario 2). Rate of legitimate users equals 100 interests/sec.

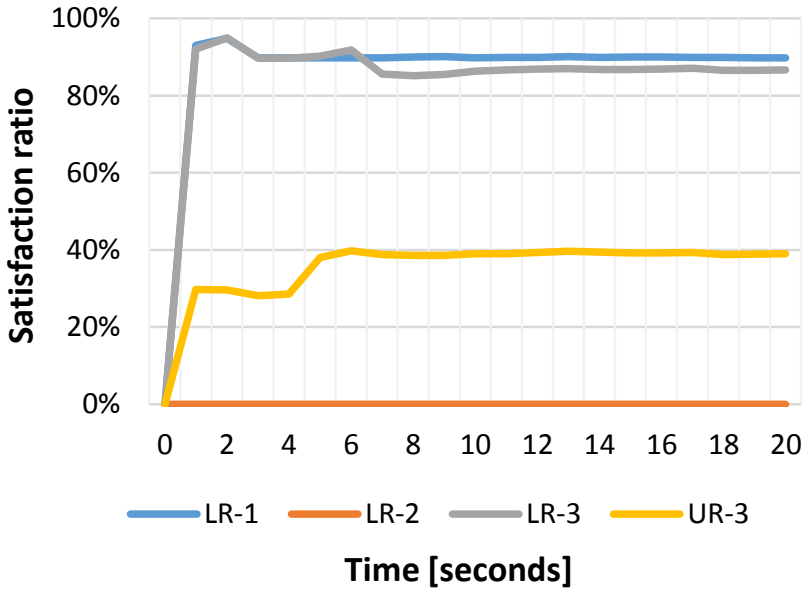

FIGURE 5.18: Satisfaction ratios.
Settings: Attackers rate equals 3000 invalid and 5000 valid ipps. Rate of legitimate users is set to 100 ipps

sent by C3 and C4. The satisfaction ratios of LR-1 and LR-3 were slightly affected compared to the previous scenario. LR-1 and LR-3 registered a satisfaction ratio of 90 and 87 %, respectively. It is due to the amount of malicious traffic sent to the network. UR-3 recorded a satisfaction ratio of nearly 40%. UR-3 received both valid and invalid requests, which is why its satisfaction ratio value was affected.

As mentioned in the previous scenario, many dropped packets were recorded in this scenario. LR-1 and LR-3 dropped a considerable number of packets, as shown in Fig. 5.17. LR-1 and LR-3 dropped an average of 3300 and 3200 packets/sec, respectively.

During the simulation, the PIT of network routers and the producer P3 were not affected by the malicious traffic. The NDN Forwarding Daemon (NFD) [Afa+16b], which ndnSIM uses as a forwarding engine, does not apply size-limiting on PIT. The size of the PIT keeps increasing until the memory runs out [Red]. That is why all legitimate Interest packets were satisfied. We could not simulate the scenario where legitimate Interest packets are dropped because of PIT running out of space.

5.9.2 MSIDN During IFA

In this subsection, we evaluate our proposed solution to test its effectiveness against IFA.

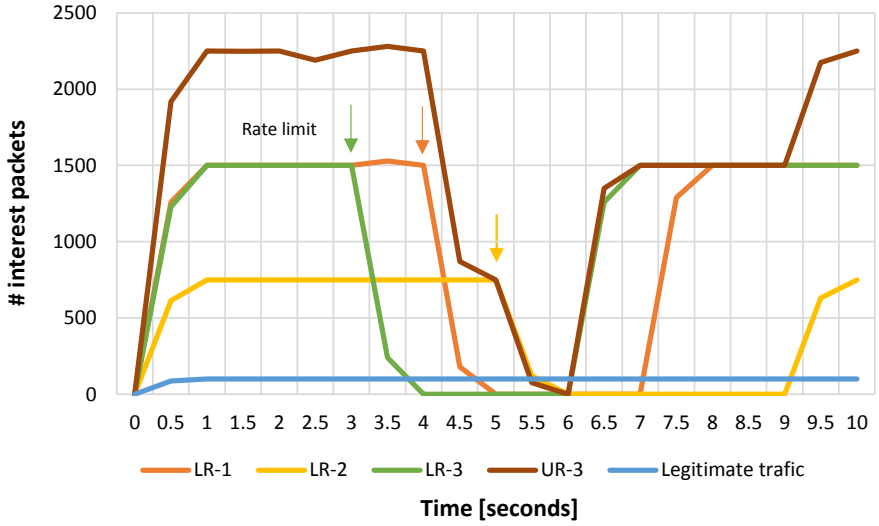

FIGURE 5.19: Rate limiting the traffic going to P3 (the victim)
Settings: Rate of attackers: 1500 and 3000 invalid ipps. Rate of legitimate users equals 100 interests/sec

Scenario1: IFA with invalid requests

The first scenario of IFA that we tested against MSIDN was IFA with invalid requests. Figure 5.19 depicts the results of rate-limiting applied by routers, on requests going to P3, to block the attack. The routers start to block the malicious traffic after the reception of PCIP. The first router that mitigated the malicious traffic going to P3 was LR-3 on $s = 3$. He applied a block on the sourcing interface. Then, LR-1 and LR-2 on $s = 4$ and $s = 5$, respectively. The malicious traffic decreased until it was completely blocked by lower-edge routers.

Legitimate traffic of routers LR-1 and LR-3 was not affected by rate-limiting as shown in Fig. 5.19. Only traffic heading to the victim P3 was limited and blocked.

Scenario2: IFA with mixed requests

The second scenario of IFA that we tested against our proposed solution was an IFA with mixed requests. Attackers target the producer P3 with valid and invalid Interest packets. Figure 5.20 shows the number of timed-out Interest packets recorded by lower-edge routers. When routers apply rate-limiting on the traffic going to the victim P3, the number of timed-out Interest packets starts to decrease. This confirms that the invalid requests going to the victim were blocked.

Figure 5.21 depicts the number of Interest packets that producer P3 received during the simulation. The number of Interest packets starts to increase until it stabilizes at 3750 ipps. At the time $s = 4$, the number of received Interest packets began to decrease. The reason is that the router LR-3 started to block the traffic going to P3.

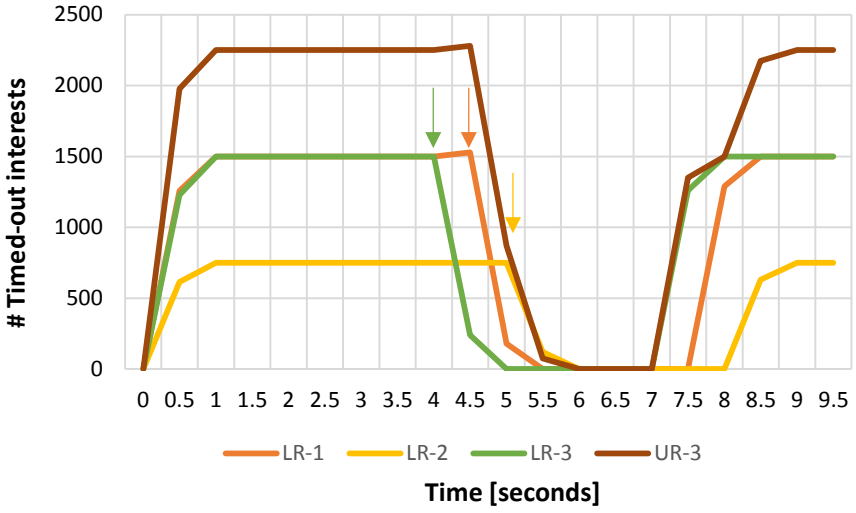

FIGURE 5.20: Number of timed-out interest packets
Settings: Rate of attackers: 1500 valid and 3000 invalid ipps. Rate of legitimate users equals 100 ipps

The number decreased until it reached zero when all lower-edge routers blocked the traffic going to P3. Figure 5.21 also confirms that every Interest packet (valid or invalid) going to P3 was blocked by the routers.

5.10 Summary

In this chapter, we presented MSIDN. A solution that is capable of mitigating simple and sophisticated Interest flooding-based (D)DoS attacks. MSIDN can mitigate malicious traffic and reduces network overhead without affecting legitimate requests. In addition, the proposed solution also blocks the attack initiators, i.e., malicious consumers. In the next chapter, we will discuss unfaced IFA scenarios.

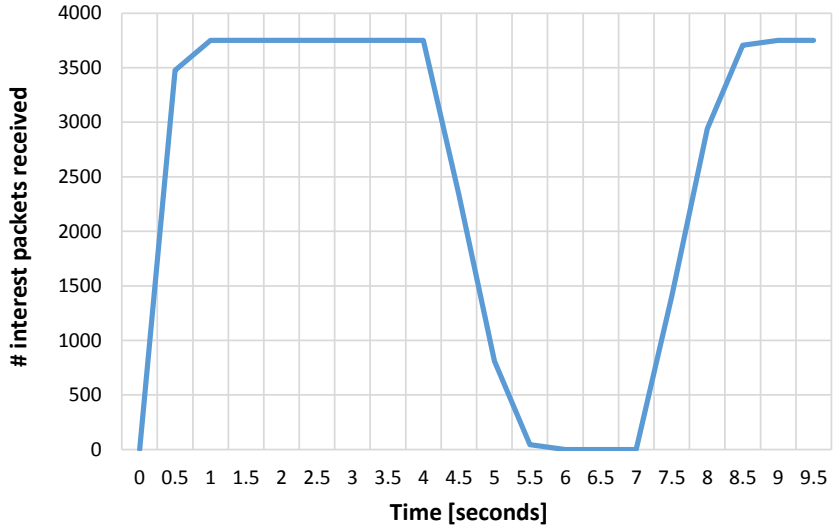

FIGURE 5.21: Number of received data requests by P3 (the victim)
Settings: Rate of attackers: 1500 valid and 3000 invalid ipps. Rate of legitimate users equals 100 ipps

Chapter 6

Unconsidered adversarial models

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 chapter, we present and explain several attacking scenarios against the existing solutions. The attacking scenarios are grouped into affecting centralized solutions and affecting distributed solutions.

6.1 Attacking scenarios against non cooperative solutions

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

6.1.1 Targeting neighboring consumers in a non-cooperative solution

The attacking scenario illustrated in fig.6.1 shows how an attacker could take advantage of a non-cooperative solution to affect legitimate consumers. In this scenario, the router *R1* takes defensive action against its interface *int1* because it reached its thresholds due to the malicious traffic, which implicitly penalizes the traffic of the legitimate consumers connected to the router *R2*.

6.1.2 Targeting distant consumers in a non-cooperative solution

Similarly, the scenario in Fig. 6.2 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 led to affecting the legitimate consumers. The attacker was able to penalize the distant consumers *Consumer1* and *Consumer2*.

6.1.3 Targeting legitimate consumers behind a switch

Another attacking scenario that targets legitimate consumers is presented in fig. 6.3. This scenario shows that non-cooperative solutions are susceptible to penalize legitimate consumers. The malicious traffic sent by the attacker will push the router to

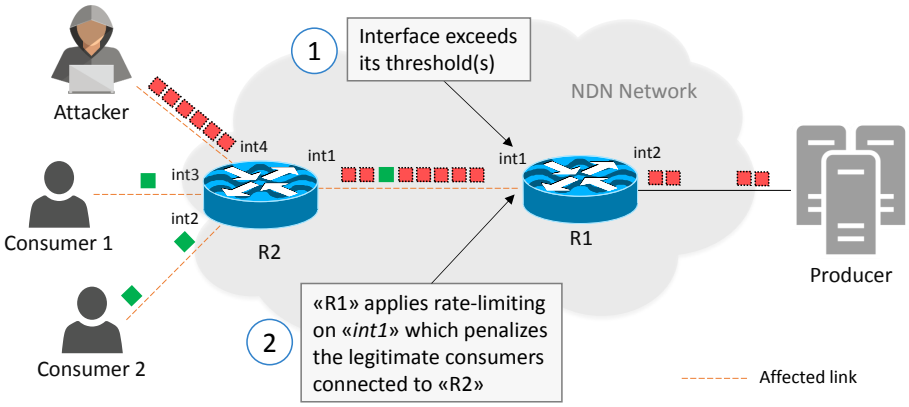

FIGURE 6.1: Attacking scenario against neighboring consumers in a non-cooperative solution

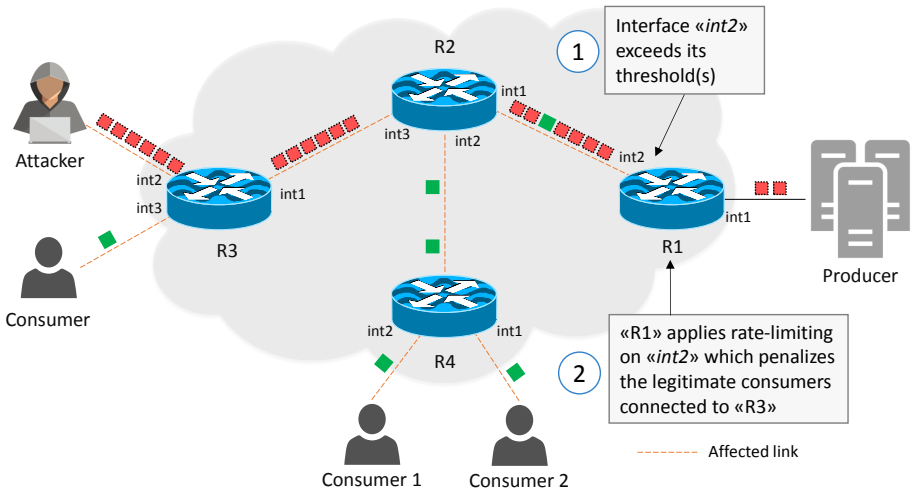

FIGURE 6.2: Attacking scenario against distant consumers in a non-cooperative solution

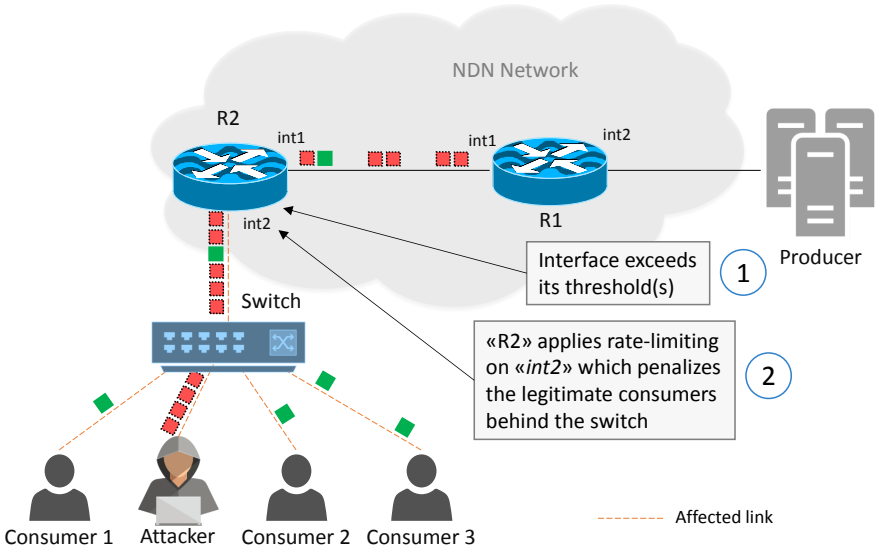

FIGURE 6.3: Attacking scenario against legitimate consumers behind a switch

take defensive action (i.e., rate-limit or block) against its interface *int2*, which leads to penalizing all the consumers connected to the switch.

6.2 Attacking scenarios against cooperative solutions

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

6.2.1 Countering alert-based solutions with a compromised edge router

In this scenario, the adversary, who is in control of an edge router, can continue flooding the network because the edge router will ignore all received solution-based alert messages as shown in 6.4. Routers could reduce the impact of the attack when each router takes a defensive action. However, in a solution where only the edge router takes defensive actions, the attacker will continue flooding the network.

6.2.2 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. 6.5. In this scenario, the attacker forge an alert message and send it to the router *R2* to push it to take a defensive action against the legitimate consumers connected to it. The attacker can conduct such an attack only if the solution use non-signed alert messages, or is in control of a router.

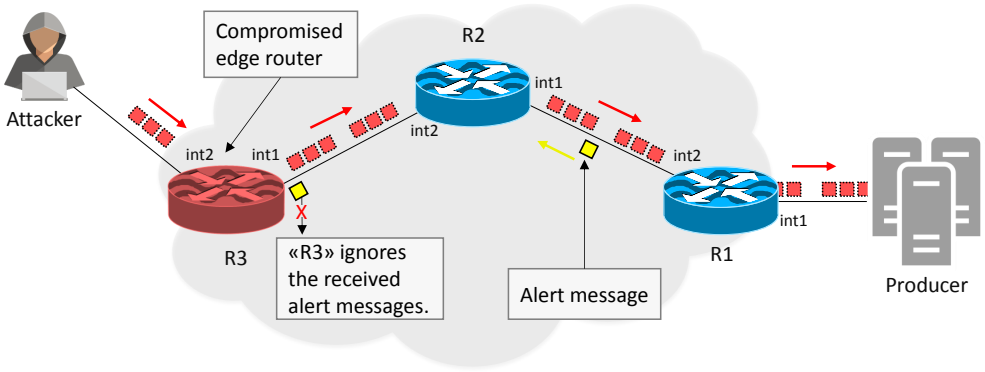

FIGURE 6.4: Countering alert-based solution with a compromised edge router

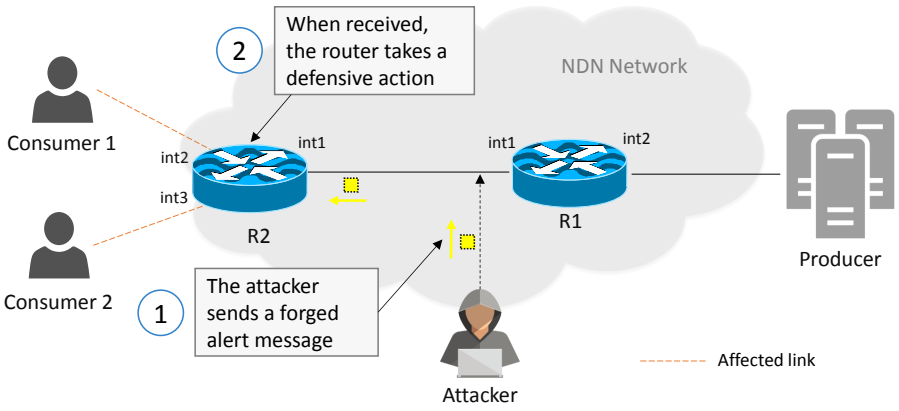

FIGURE 6.5: Attacking scenario against legitimate consumers with alert message

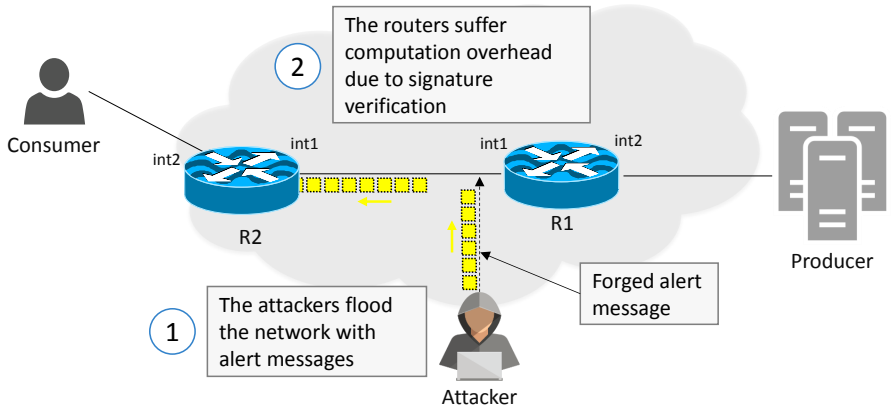

FIGURE 6.6: Targeting routers with forged alert messages

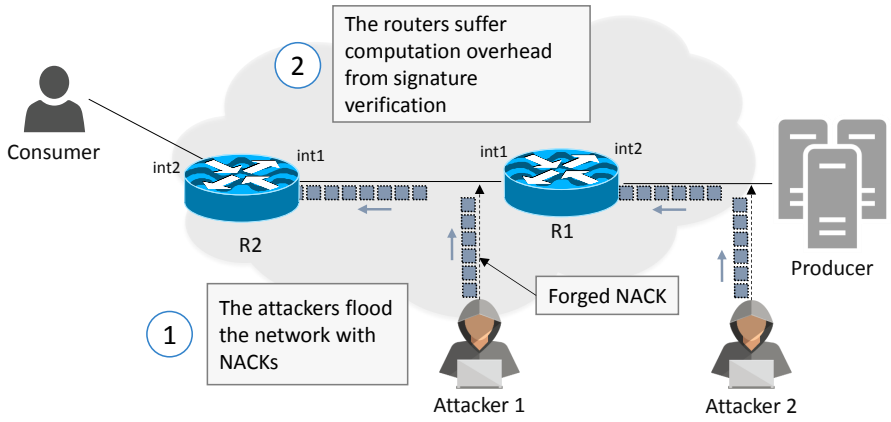

FIGURE 6.7: Targeting routers with forged NACK packets

6.2.3 Targeting routers resources with forged alert messages

Another way of using solution alert messages is to target routers' resources. As shown in 6.6, the attacker continually sends forged alert messages to the router *R2* to stress it with signature verification and leads it to computation overhead.

6.2.4 Targeting routers resources with forged NACK packets

Similar to the previous scenario, attackers can also lunch attacks against the routers that rely on NACK packets to send solution-based messages like [Liu+18a; Don+20]. To do so, attackers flood the targeted router with forged NACK packets to penalize him with computation overhead due to signature verification.

6.2.5 Flooding the network with solution-based spoofed data packets

Some solutions like [Dai+13] use spoofed Data packets to counter malicious nodes. Attackers can take advantage of this feature to flood the network as shown in 6.8. In

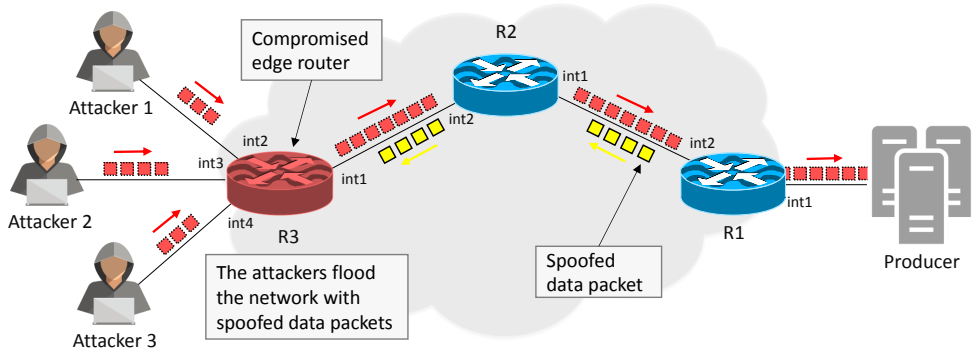

FIGURE 6.8: Flooding the network with solution-based spoofed Data packets

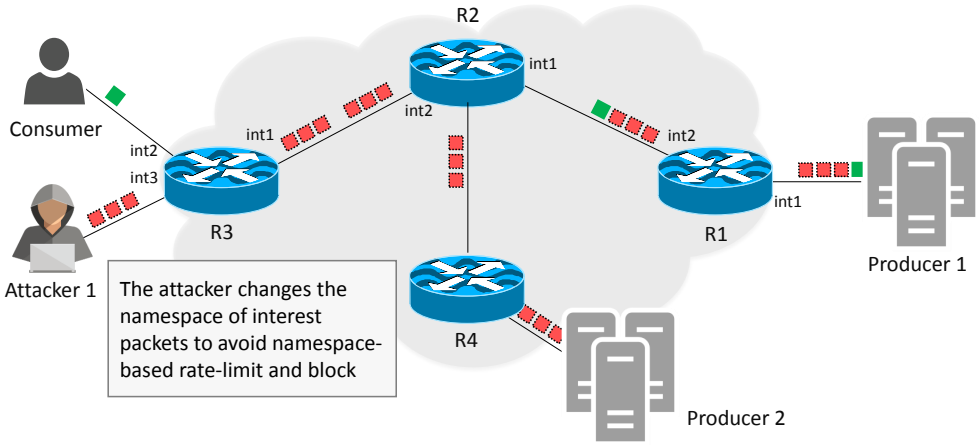

FIGURE 6.9: Attacking scenario against prefix-based solutions

this scenario, the attackers who are in control of an edge router flood the network with forged Interest packets. As a defensive action, the router sends back spoofed Data packets so the edge router can take proper actions. As the edge router is controlled by the attackers, it will take no action against the malicious nodes.

6.2.6 Countering prefix-based solutions

As mitigation action, some solutions like [Tan+13; Che+19a] 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. 6.9.

6.2.7 Affecting legitimate traffic in a prefix-based solutions

Another attacking scenario against prefix-based solutions is illustrated in 6.10. The goal of the attacker in this scenario is to affect legitimate traffic heading to a specific producer. To do so, the attacker flood the network with forged Interest packets with

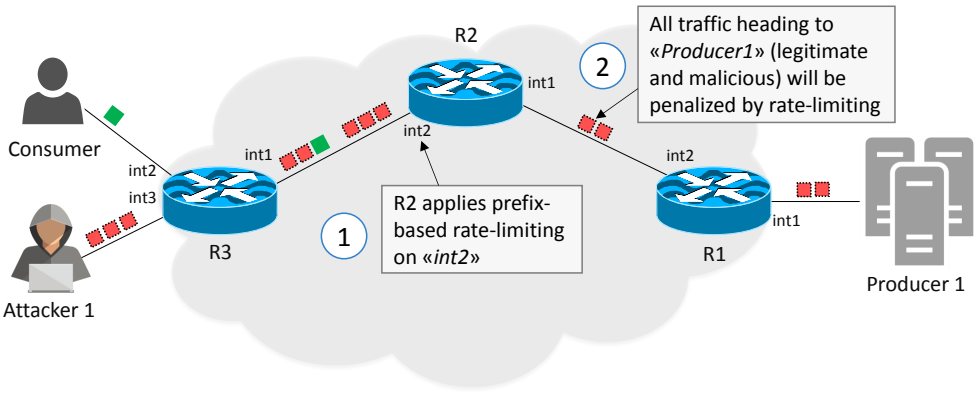

FIGURE 6.10: Targeting legitimate traffic in a prefix-based solution

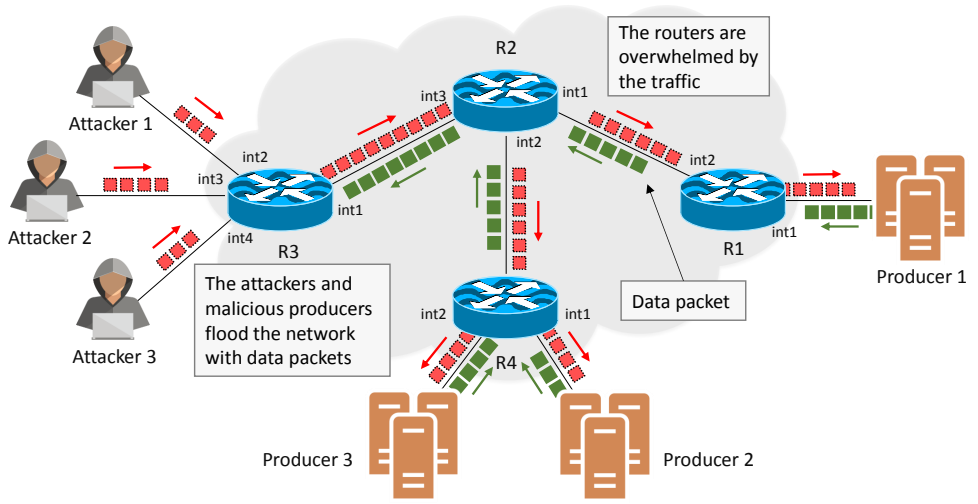

FIGURE 6.11: Targeting the network with a distributed collusive attack

the targeted prefix to push the routers to take defensive action against this prefix, which leads also to penalizing the legitimate traffic.

6.2.8 Targeting the network with a distributed collusive attack

Figure 6.11 illustrates a distributed collusive attack. In this scenario, attackers work with a distributed group of malicious producers to overwhelm the network. This attack generates high traffic and introduces a significant delay to the network, which makes it even more difficult to mitigate for existing solutions.

6.2.9 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

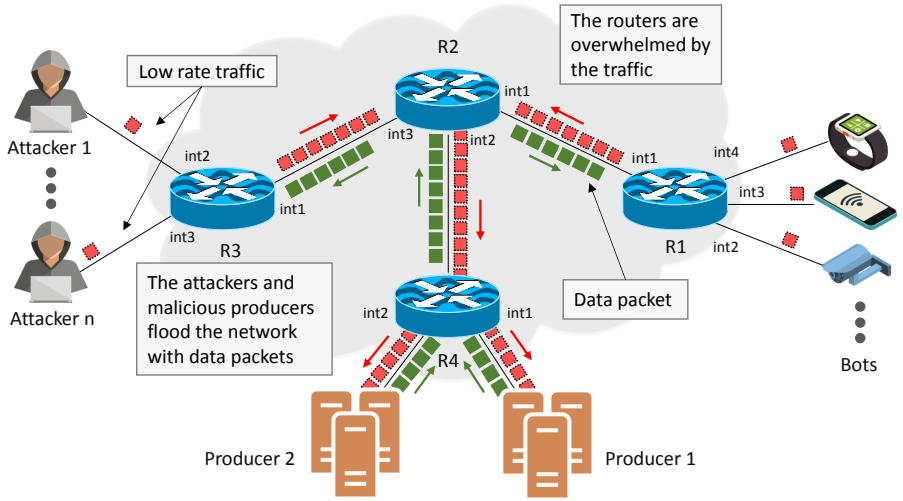

FIGURE 6.12: Targeting the network with a low-rate distributed collusive attack

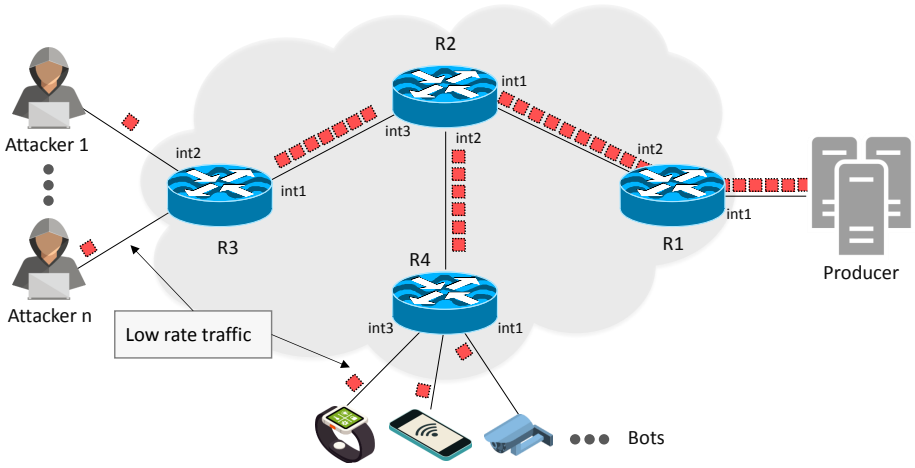

FIGURE 6.13: Targeting the network with low-rate distributed IFA

request data from malicious producers with low sending rates, as shown in 6.12. Existing collusive solutions like [SS16] rely on the high traffic rate to detect a collusive attack which makes them inefficient against this attacking scenario.

6.2.10 Targeting the network with low-rate distributed IFA

The attacking scenario depicted in 6.13 represents a distributed IFA with low sending rates. This attacking solution works against all solutions that use the traffic rate and PIT usage as detection metrics. Attackers overcome these solutions by adopting a low sending rate to keep flooding the network with malicious Interest packets.

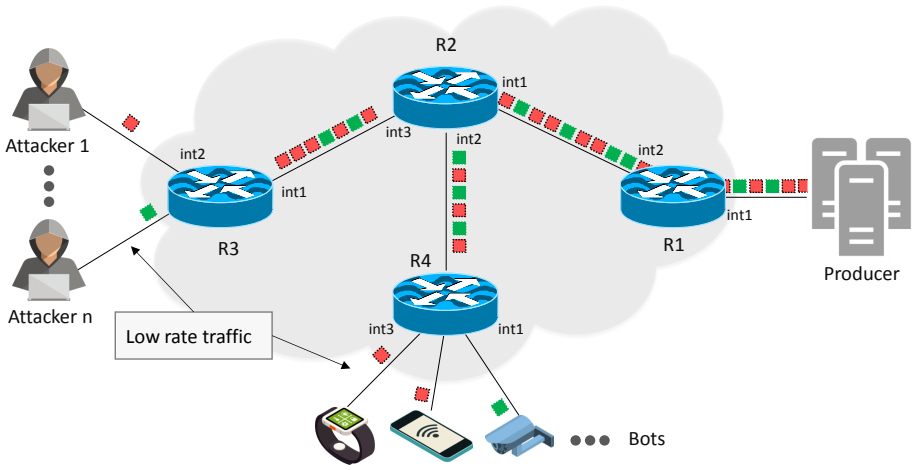

FIGURE 6.14: Targeting the network with low-rate mixed distributed IFA

6.2.11 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 6.14. In these particular scenarios, attackers and controlled nodes flood the network with valid and invalid Interest packets. It permits attackers to counter other detection metrics like the satisfaction ratio and timed-out Interest packets.

6.2.12 Scenario of a Smart IFA

In this scenario, attackers adopt smart behaviors 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 continues the attack. This attacking scenarios could be local (Fig. 6.15) or distributed (Fig. 6.16).

6.3 Summary

In this chapter, we presented several unfaced adversarial models. The attacking scenarios that we detailed in this chapter were not considered by the existing solutions. The next chapter concludes this thesis.

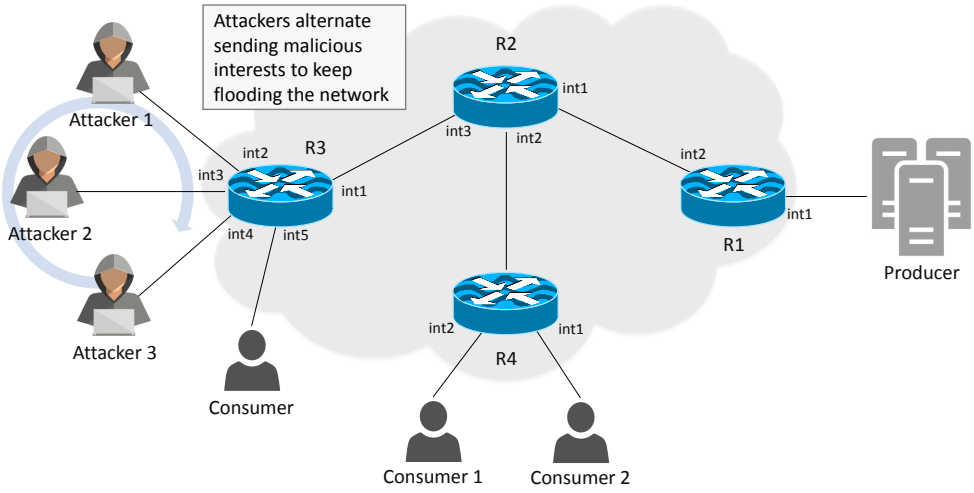

FIGURE 6.15: Local IFA: case of smartly behaving attackers

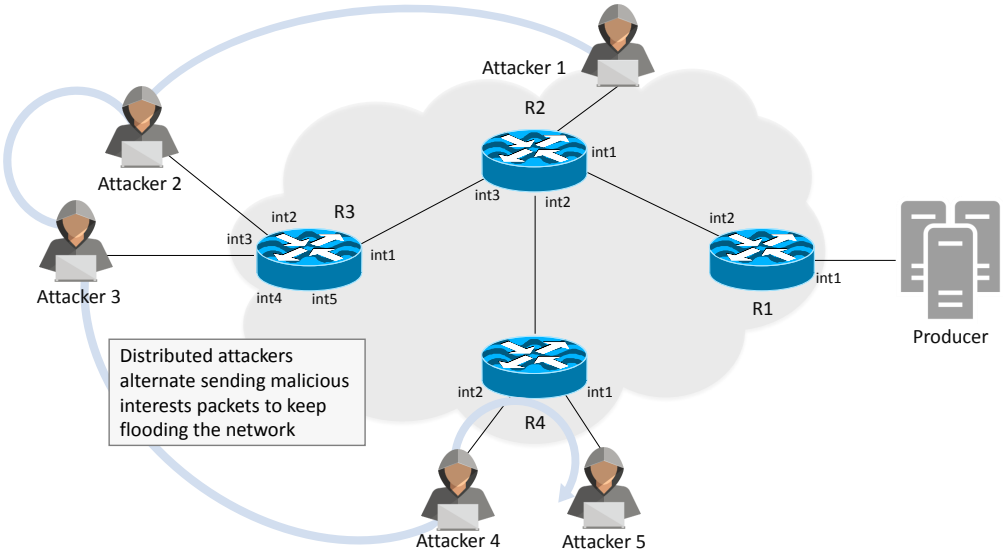

FIGURE 6.16: Distributed IFA: case of smartly behaving attackers

Chapter 7

Conclusion

The NDN architecture provides intrinsic security where security techniques are used to secure the data instead of focusing on communication security. This phenomenon solves some challenges of the communication security problems inherited from the traditional Internet architecture. Despite the intrinsic security provided by NDN, there are still security and privacy threats faced by NDN.

This dissertation focused on the Interest Flooding Attack. In this work, we briefly discussed the availability attacks that target NDN. Following that, we detailed IFA, explained its aspects, and showed its variants. After that, we reviewed and classified all the relevant IFA related work present in the literature.

As a first contribution, we proposed a solution to detect and mitigate the Interest Flooding Attack. Our proposal takes into consideration the network congestion and takes it as a parameter to avoid false alarms regarding the behavior of the routers. We showed, through simulation results, that network congestion can lead routers to mistakenly consider a legitimate user as malicious. Finally, we presented a scenario of an IFA and showed how the attacker was mitigated.

We also presented another solution, named MSIDN. This solution is capable of mitigating simple and sophisticated Interest flooding-based (D)DoS attacks. MSIDN can mitigate malicious traffic and reduces network overhead without affecting legitimate requests. In addition, the proposed solution also blocks the attack initiators, i.e., malicious consumers, as shown through the simulation results.

Finally, we presented several non-conventional IFA scenarios that the present solutions did not take into consideration.

7.1 Future Research Direction

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. After studying IFA, we conclude that the challenges and

future direction of IFA solutions need to focus on five main aspects. First, IFA solutions need to be designed to detect multiple types and variants of IFA. Second, routers and producers need to cooperate for better results. Third, the solution must precisely mitigate attackers and malicious traffic without affecting legitimate consumers. Fourth, the proposed solutions have to be resource-friendly. Lastly, newly designed solutions must be scalable and easy to deploy.

7.1.1 Intelligent Detection

As shown in chapter 6, the vast majority of existing IFA solution is explicit and can detect a specific type of IFA. Additionally, these solutions cannot even detect this single IFA when attackers adopt different scenarios, which leads to flooding the network with malicious traffic and also affecting the legitimate traffic. Newly designed solutions need to be more reliable and intelligent to cope with a variety of IFA scenarios.

7.1.2 Broad Cooperation

Cooperation between routers and producers is essential to build a reliable IFA solution. It will ensure having a better view of the network, which implies getting more information and helps to better understand ongoing attacks. Also, the cooperation between routers and producers helps in reducing false positive detection.

7.1.3 Precise Mitigation

Precise mitigation is another important aspect that needs to be considered when designing a new solution. As shown in chapter 6, legitimate consumers could suffer from existing solutions in some scenarios. It leads to unfairly blocking legitimate traffic and even blocking consumers in some scenarios. That is why IFA solutions have to be designed to distinguish between the bad traffic and the good traffic to precisely mitigate adversary nodes and stop the spread of the malicious traffic.

7.1.4 Resource Friendly

Another essential aspect that should not be neglected when designing a reliable IFA solution is its impact on resources. The 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 traffic peaks. Also, the solution has to have a low resource consumption fingerprint so attackers cannot take advantage of it to take down network nodes.

7.1.5 Scalability

The last aspect of a reliable IFA solution is its scalability. Newly designed solutions need to able to scale at any level without affecting the overall functioning of the

system. Additionally, solutions need to offer easy to deploy mechanisms for newly added nodes.

Appendix A

Performance comparison

In this chapter, we compare existing IFA solutions in terms of simulation parameters. We also compare the evaluation metrics used to validate the proposed solution.

A.1 Simulation Parameters

In the following section, we list and explain all the simulation parameters used to evaluate the proposed solutions. Table A.1 summarizes the simulation environment adopted by the existing IFA solutions.

A.1.1 Simulator

The most common simulator used to evaluate IFA solutions is the NDN network simulator (ndnSIM) [MAZ17]. ndnSIM is an open-source network simulator based on ns-3 and designed to conduct NDN-based simulation studies. Some solutions evaluated their work using *Matlab*. On the other hand, one existing IFA solution was evaluated using *OMNeT++*, which is another network simulator.

A.1.2 Network topology

Several network topologies were used by existing works during the evaluation part. The most commonly used topologies are the Rocketfuel topologies[SMW02].

A.1.3 Links bandwidth

This metric represents the bandwidth chosen for the links during the simulation.

A.1.4 Network delay

It represents the network delay values chosen during the evaluation.

A.1.5 Forwarding strategy

It shows the strategy adopted by routers to forward Interest packets.

A.1.6 Dishonesty ratio

This metric equals the number of malicious nodes deployed during the simulation to the total number of nodes.

A.1.7 Number of producers

It represents the number of producers deployed in the network.

A.1.8 Rate of consumers

This parameter equals the frequency at which legitimate consumers send interest packets.

A.1.9 Rate of attackers

It represents the number of malicious Interest packets sent by attackers in a period.

A.1.10 Nature of malicious interest

It shows the nature of Interest packets used by malicious nodes during the attack.

A.1.11 PIT size

This parameter denotes the size of routers PIT adopted during the simulations. It is represented in terms of memory size or the number of PIT entries.

A.1.12 Interest lifetime

It represent the value of the *InterestLifetime* field chosen for the evaluation.

A.1.13 Intermediate Cache

This simulation parameter states whether or not routers use the local CS to satisfy incoming Interest packets.

A.1.14 CS size

Equals the capacity of a router's CS in terms of the number of entries, i.e., data that it supports.

A.1.15 CS strategy

This parameter represents the cache replacement strategy adopted by routers.

A.1.16 Data size

It represents the size of the Data packet chosen during the simulation.

TABLE A.1: Simulation Parameters

Ref	Simulator	Network topology	Links bandwidth	Network delay	Forwarding strategy	Dishonesty ratio	Number of producers	Rate of consumers	Rate of attackers	Nature of malicious interest	PIT size	Interest lifetime	Intermediate Cache	CS size	CS strategy	Data size
[Afa+13]	ndnSIM	Binary tree Rocketfuels AS 7018	10Mbps	$80-330ms$	-	$6-50\%$ 40%	01	-	-	Non-existent	-	-	No	-	-	1100Bytes
[Com+13]	ndnSIM	Rocketfuel AS 7018	-	-	-	50%	01	30ipps	01int/min	Non-existent	120KB	$4s$	-	-	-	-
[Dai+13]	-	Rocketfuel AS 1755	-	-	-	-	01	-	1000ipps	Non-existent	-	$500ms-8s$	-	-	-	-
[Vas+15]	ndnSIM	Random	-	-	-	5%	-	20ipps	1000ipps	Non-existent	-	$200ms-1s$	-	-	-	1KB
[Wan+14a]	ndnSIM	Rocketfuel AS 7018	1-100Mbps	$5-70ms$	Best route	40%	01	20ipps	400ipps	Non-existent	100 entries	$1s$	Yes	500 entries	LRU	1KB
[SWS15]	ndnSIM	Rocketfuel AS 3967	-	-	-	25%	-	-	500-10000ipps	Non-existent	5000 entries	-	No	-	-	1100Bytes
[SS16]	ndnSIM	Rocketfuel AS 3257	-	-	-	25%	01 malicious	100ipps	200-5000ipps	Non-existent	5000 entries	$2s$	No	-	-	1100Bytes
[Wan+13]	ndnSIM	Rocketfuel AS 7018	1-100Mbps	$5-70ms$	Best route	40%	01	10-1000ipps	100ipps	Non-existent	Unlimited	-	Yes	500 entries	LRU	1KB
[Wan+14b]	Matlab	-	-	-	-	-	-	Poisson distribution	Poisson distribution	Non-existent	-	-	-	-	-	-
[KGZ15]	ndnSIM Matlab	DFN-like Rocketfuel AS 7018	-	-	-	$20-50\%$	02-05	100-600ipps	400-3000ipps	Existent and non-existent	-	-	-	-	-	-
[Din+16]	ndnSIM	Small tree Rocketfuel AS 7018	1-100Mbps	$5-70ms$	-	$16-25\%$	01	67-1000ipps	2000-20000ipps	Non-existent	10000 entries	$1s$	No	-	-	-
[Ngu+15]	ndnSIM Matlab	Binary tree	-	-	-	-	01	Poisson distribution	-	-	-	-	-	-	-	-
[Xin+17]	ndnSIM	China Telecom	-	-	-	15%	01	50ipps	Random distribution	-	200 entries	$4s$	No	-	-	-
[Xin+16]	ndnSIM	Small tree	100Mbps	$10ms$	Best route	75%	01	200ipps	20ipps	Non-existent	50 entries	$1s$	No	-	-	1KB
[ZLL18]	ndnSIM	Small tree	-	-	-	25%	01	50ipps	100ipps	Non-existent	200 entries	$1s$	-	-	-	-
[NWN18]	ndnSIM	Rocketfuel AS 1221	1-100Mbps	$5-70ms$	-	95%	10	100ipps	100ipps	Existent	-	-	-	-	-	1KB
[Ben+19a], [Ben+20a]	ndnSIM	Small topology Rocketfuel AS 7018	-	-	-	$6-50\%$	01	30ipps	100-10000ipps	Non-existent	600KB	-	-	-	-	-
[Zha+19]	ndnSIM	Meshed topology	-	-	-	20%	01	40ipps	100-6000ipps	Existent and non-existent	-	-	Yes	200 entries	-	-
[Hou+19]	ndnSIM	Binary tree	-	-	-	25%	01	200ipps	1000-2000ipps	-	50 entries	$1s$	-	-	-	-
[PPB19]	OMNeT++	Small tree	-	-	-	-	-	-	-	-	-	-	-	-	-	-
[ZL19]	ndnSIM	Small Tree Large network	-	-	-	$35-40\%$	01	-	-	-	-	-	-	-	-	-
[AR16]	-	Internet-like	300Mbps	-	-	30%	-	5ipps	-	-	-	-	-	-	-	500Bytes
[Che+19b]	ndnSIM	Modified Rocketfuel AS 7018	-	$300ms$	Best route	40%	01	40ipps	13ipps then $3-10\%$ higher	Existent	2000 entries	$1s$	-	-	-	1100Bytes
[Liu+18a]	ndnSIM	GEANT	155Mbps-20Gbps	$10ms$	Best route	50%	01	250ipps	200ipps	-	-	-	Yes	1000 entries	-	4KB
[Liu+18b]	-	GEANT	100Mbps-20Gbps	-	-	40%	01	20ipps	200ipps	-	-	-	-	-	-	4KB
[Che+19a]	-	Binary tree	-	-	-	12%	01	100ipps	2000ipps	Non-existent	300 entries	$500ms$	-	-	-	-
[Zhi+19]	ndnSIM	Binary tree	10Mbps	$10ms$	-	75%	01	500ipps	100ipps	Non-existent	200 entries	$1s$	-	-	-	-
[Yin+19]	ndnSIM	Binary tree Small topology	-	$10ms$	-	$45-50\%$	02	5-10ipps	100ipps	Non-existent	100 entries	-	-	-	-	-
[WGQ19]	Matlab	Rocketfuel AS 7018	1-100Mbps	$5-70ms$	-	-	-	-	-	-	-	-	-	-	-	-
[Ben+19b]	ndnSIM	25 nodes topology	1-100Mbps	$10-100ms$	Best route	$10-25\%$	09	100-500ipps	10000ipps	Non-existent	-	-	Yes	100 entries	LRU	1KB
[Don+20]	-	Small topology	100Mbps-1Gbps	$20-30ms$	-	10%	01	100ipps	1000ipps	Non-existent	-	$2s$	-	-	-	-
[ZLW20]	ndnSIM	25 nodes topology	1-100Mbps	$5-66ms$	-	55%	-	500ipps	500-2000ipps	-	100 entries	$2s$	-	-	-	512Bytes
[Shi+16]	ndnSIM	Binary tree	10Mbps	$10ms$	-	25%	01	1000ipps	10000ipps	Non-existent	275 entries	$1s$	-	-	-	-
[SS18]	-	12 nodes topology	1Mbps	$1ms$	-	60%	02	10ipps	320-2030ipps	Non-existent	50 entries	-	Yes	10 entries	FIFO	1KB
[Tan+13]	ndnSIM	100-to-1 topology	10Mbps	$1-10ms$	-	$10-90\%$	01	200ipps	200ipps	Non-existent	Unlimited	$2s$	-	-	-	-
[Ben+20b]	ndnSIM	18 nodes topology	10Mbps-1Gbps	$1-10ms$	Best route	66%	03	100ipps	1500ipps-10000ipps	Existent and non-existent	Unlimited	-	Yes	100 entries	LRU	1KB

A.2 Evaluation Metrics

In this section, we detail all the evaluation metrics used by existing IFA works to validate the proposed solutions. Table A.2 summarizes the evaluation metrics used by each existing solution.

A.2.1 Satisfaction ratio

As described in 2, the satisfaction ratio equals the number of Data packets returned to the total number of Interest packets issued.

A.2.2 PIT usage

This metric is also called PIT occupancy. The PIT usage of an interface represents the number of pending Interest packets sourced by this interface to the total capacity of the PIT, as mentioned in 3.

A.2.3 Number of PIT entries

This metric represents the number of pending PIT entries of an interface. Some solutions use this metric to detect abnormal traffics. This metric is compared to the usual activity of an interface to detect malicious traffic.

A.2.4 Number of Interest packets

Unlike the number of PIT entries metric, which includes only pending Interests, this metric represents the total number of Interest packets issued by an interface.

A.2.5 Number of Data packets

It represents the total number of Data packets received by an interface in a period.

A.2.6 Number of satisfied interest packets

It represents the total number of interest packets that resulted in a data packet.

A.2.7 Number of dropped packets

This metric represents the total number of dropped packets recorded for a given interface.

A.2.8 Number of timed-out interest packets

It represents the number of timed-out Interest packets of a given interface. An Interest packet times-out when its *InterestLifetime* reaches zero before a response comes back.

TABLE A.2: Simulation Evaluation Metrics

Ref	Satisfaction ratio	PIT usage	# of PIT entries	# of interests	# of Data	# of satisfied interests	Dropped packets	# of timed-out interests	# of NACK packets	Traffic rate	Interest drop ratio	Delay	False positive ratio
[Afa+13]	✓												
[Com+13]		✓			✓								
[Dai+13]		✓	✓										
[Vas+15]		✓					✓						
[Wan+14a]	✓	✓										✓	
[SWS15],[SS16]		✓				✓							
[Wan+13]		✓											
[Wan+14b]				✓									
[KGZ15]	✓	✓			✓								✓
[Din+16]	✓	✓											
[Xin+17]		✓								✓			
[Xin+16]		✓											
[ZLL18]			✓										✓
[NWN18]	✓												
[Ben+19a], [Ben+20a]	✓	✓					✓						
[Zha+19]										✓			
[Hou+19]		✓			✓								
[PPB19]		✓				✓							
[ZL19]	✓		✓										
[AR16]											✓	✓	
[Che+19b]	✓		✓									✓	
[Liu+18a]			✓										✓
[Liu+18b]		✓										✓	✓
[Che+19a]		✓			✓								
[Zhi+19]	✓	✓		✓		✓	✓						
[Yin+19]	✓			✓	✓				✓				
[Ben+19b]	✓			✓			✓	✓	✓				
[ZLW20]	✓	✓									✓	✓	
[Shi+16]				✓	✓		✓						
[SS18]	✓												
[SS18]	✓												
[Ben+20b]	✓					✓	✓	✓	✓				

A.2.9 Number of NACK packets

Represents the number of NACK packets recorded by a router for one or all its interfaces.

A.2.10 Traffic rate

As already defined in 1, the traffic rate of an interface equals the frequency of incoming interest packets.

A.2.11 Interest drop rate

This metric represents the number of dropped Interest packets, due to PIT saturation, to the total number of Interest packets issued (by an interface) or received by a router.

A.2.12 Delay

It represents the time that an Interest packet takes to reach a destination.

A.2.13 False positive ratio

This metric represents the number of wrongly classified events as malicious to the total number of events recorded.

References

[AA20] Mohammad Alhisnawi and Mahmood Ahmadi. "Detecting and Mitigating DDoS Attack in Named Data Networking". In: *Journal of Network and Systems Management* (2020), pp. 1343–1356.

[AC17] Hila Ben Abraham and Patrick Crowley. "Controlling strategy retransmissions in named data networking". In: *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. IEEE. 2017, pp. 70–81.

[Afa+13] Alexander Afanasyev et al. "Interest flooding attack and countermeasures in Named Data Networking". In: *2013 IFIP Networking Conference*. IEEE. 2013, pp. 1–9.

[Afa+16a] Alexander Afanasyev et al. "Content-based security for the web". In: *Proceedings of the 2016 New Security Paradigms Workshop*. 2016, pp. 49–60.

[Afa+16b] Alexander Afanasyev et al. "NFD developer's guide". In: *NDN, Technical Report* (2016).

[Afa+17] Alexander Afanasyev et al. "NDNS: A DNS-like name service for NDN". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2017, pp. 1–9.

[Ahl+12] Bengt Ahlgren et al. "A survey of information-centric networking". In: *IEEE Communications Magazine* 50.7 (2012), pp. 26–36.

[Ana+11] Ashok Anand et al. "XIA: An architecture for an evolvable and trustworthy Internet". In: *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. 2011, pp. 1–6.

[And+13] Tom Anderson et al. "The nebula future internet architecture". In: *The Future Internet Assembly*. Springer. 2013, pp. 16–26.

[AR16] Aubrey Alston and Tamer Refaei. "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. 2016, pp. 85–88.

[ASWS15] Samir Al-Sheikh, Matthias Wählisch, and Thomas C Schmidt. "Revisiting countermeasures against ndn interest flooding". In: *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. 2015, pp. 195–196.

[Ben+19a] Abdelmadjid Benarfa et al. “ChoKIFA: A New Detection and Mitigation Approach Against Interest Flooding Attacks in NDN”. In: *International Conference on Wired/Wireless Internet Communication*. Springer. 2019, pp. 53–65.

[Ben+19b] Ahmed Benmoussa et al. “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. 2019, pp. 1–6.

[Ben+20a] Abdelmadjid Benarfa et al. “ChoKIFA+: an early detection and mitigation approach against interest flooding attacks in NDN”. In: *International Journal of Information Security* (2020).

[Ben+20b] Ahmed Benmoussa et al. “MSIDN: Mitigation of sophisticated interest flooding-based DDoS attacks in named data networking”. In: *Future Generation Computer Systems* 107 (2020), pp. 293–306.

[Cao+16] Jianxun Cao et al. “Fetching popular data from the nearest replica in NDN”. In: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2016, pp. 1–9.

[Cha+17] Kevin Chan et al. “Fuzzy interest forwarding”. In: *Proceedings of the Asian Internet Engineering Conference*. 2017, pp. 31–37.

[Che+19a] Jing Chen et al. “Isolation Forest Based Interest Flooding Attack Detection Mechanism in NDN”. In: *2019 2nd International Conference on Hot Information-Centric Networking (HotICN)*. IEEE. 2019, pp. 58–62.

[Che+19b] Guang Cheng et al. “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. 2019, pp. 7–12.

[Cho+13] Seungoh Choi et al. “Threat of DoS by interest flooding attack in content-centric networking”. In: *The International Conference on Information Networking 2013 (ICOIN)*. IEEE. 2013, pp. 315–319.

[Cis] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*. 2020. URL: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdfl.

[Com+13] Alberto Compagno et al. “Poseidon: Mitigating interest flooding DDoS attacks in named data networking”. In: *38th annual IEEE conference on local computer networks*. IEEE. 2013, pp. 630–638.

[Com+15] Alberto Compagno et al. “To NACK or not to NACK? negative acknowledgments in information-centric networking”. In: *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2015, pp. 1–10.

[Dai+13] Huichen Dai et al. "Mitigate ddos attacks in ndn by interest traceback". In: *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2013, pp. 381–386.

[Dat] *NDN Data packet*. 2020. URL: https://named-data.net/doc/NDN-packet-spec/current/data.html.

[Din+16] Kun Ding et al. "Cooperative detection and protection for interest flooding attacks in named data networking". In: *International Journal of Communication Systems* 29.13 (2016), pp. 1968–1980.

[Din+17] Ikram Ud Din et al. "Caching in information-centric networking: Strategies, challenges, and future research directions". In: *IEEE Communications Surveys & Tutorials* 20.2 (2017), pp. 1443–1474.

[Don+20] Jiaqing Dong et al. "InterestFence: Simple but efficient way to counter interest flooding attack". In: *Computers & Security* 88 (2020), p. 101628.

[DR08] Tim Dierks and Eric Rescorla. "RFC 5246-the transport layer security (TLS) protocol version 1.2". In: *The Internet Engineering Task Force (IETF)* (2008).

[ET11] Thomas Edwall and Benoit Tremblay. "SAIL Project". In: (2011).

[Gar+11] G. García et al. "COMET: Content mediator architecture for content-aware networks". In: *2011 Future Network Mobile Summit*. 2011, pp. 1–8.

[Gas+13] Paolo Gasti et al. "DoS and DDoS in named data networking". In: *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2013, pp. 1–7.

[Gha+17] Cesar Ghali et al. "Closing the floodgate with stateless content-centric networking". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2017, pp. 1–10.

[Gha+18] Chavoosh Ghasemi et al. "MUCA: New routing for named data networking". In: *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE. 2018, pp. 289–297.

[Hou+19] Rui Hou et al. "Theil-Based Countermeasure against Interest Flooding Attacks for Named Data Networks". In: *IEEE Network* 33.3 (2019), pp. 116–121.

[Jac+07] Van Jacobson et al. "Content-centric networking". In: *Whitepaper, Palo Alto Research Center* (2007), pp. 2–4.

[Jac+09] Van Jacobson et al. "Networking named content". In: *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM. 2009, pp. 1–12.

[KF11] Suresh Krishnan and Sheila Frankel. "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap". In: *IETF RFC 6071, Ericsson* (2011).

[KGZ15] Amin Karami and Manel Guerrero-Zapata. "A hybrid multiobjective rbf-pso method for mitigating dos attacks in named data networking". In: *Neurocomputing* 151 (2015), pp. 1262–1282.

[Kop+07] Teemu Koponen et al. "A data-oriented (and beyond) network architecture". In: *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. 2007, pp. 181–192.

[KSS19] Naveen Kumar, Ashutosh Kumar Singh, and Shashank Srivastava. "Feature selection for interest flooding attack in named data networking". In: *International Journal of Computers and Applications* (2019), pp. 1–10.

[LB14] Zhaogeng Li and Jun Bi. "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*. 2014, pp. 1–6.

[Lee+18] Craig A Lee et al. "Supporting virtual organizations using attribute-based encryption in named data networking". In: *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE. 2018, pp. 188–196.

[Leh+16] Vince Lehman et al. "A secure link state routing protocol for NDN". In: *Tech. Rep. NDN-0037* (2016).

[Li+18] Zhuo Li et al. "Packet forwarding in named data networking requirements and survey of solutions". In: *IEEE Communications Surveys & Tutorials* 21.2 (2018), pp. 1950–1987.

[Li+19a] Tianxiang Li et al. "Distributed Dataset Synchronization in Disruptive Networks". In: *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE. 2019, pp. 428–437.

[Li+19b] Yanbiao Li et al. "A secure sign-on protocol for smart homes over named data networking". In: *IEEE Communications Magazine* 57.7 (2019), pp. 62–68.

[Liu+18a] Gang Liu et al. "Accuracy or delay? A game in detecting interest flooding attacks". In: *Internet Technology Letters* 1.2 (2018), e31.

[Liu+18b] Gang Liu et al. "BLAM: Lightweight Bloom-filter based DDoS mitigation for information-centric IoT". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2018, pp. 1–7.

[Mas+17] Spyridon Mastorakis et al. "ntorrent: Peer-to-peer file sharing in named data networking". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2017, pp. 1–10.

[Mas+20] Spyridon Mastorakis et al. "ICedge: When Edge Computing Meets Information-Centric Networking". In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 4203–4217.

[MAZ17] Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. "On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation". In: *ACM Computer Communication Review* (July 2017).

[MLZ20] Spyridon Mastorakis, Tianxiang Li, and Lixia Zhang. "DAPES: Named Data for Off-the-Grid File Sharing with Peer-to-Peer Interactions". In: *arXiv preprint arXiv:2006.01651* (2020).

[MM20] Abderrahmen Mtibaa and Spyridon Mastorakis. "NDNTP: A Named Data Networking Time Protocol". In: *arXiv preprint arXiv:2007.07807* (2020).

[Ndn] *NDN Packet Format Specification version 0.3*. 2020. URL: https://named-data.net/doc/NDN-packet-spec/current/interest.html.

[Ngu+15] Tan N Nguyen et al. "Detection of interest flooding attacks in named data networking using hypothesis testing". In: *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE. 2015, pp. 1–6.

[NWN18] Yoshimichi Nakatsuka, Janaka L Wijekoon, and Hiroaki Nishi. "FROG: A Packet Hop Count based DDoS Countermeasure in NDN". In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2018, pp. 00492–00497.

[NZ19] Eric Newberry and Beichuan Zhang. "On the Power of In-Network Caching in the Hadoop Distributed File System". In: *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 2019, pp. 89–99.

[PPB19] Cong Pu, Nathaniel Payne, and Jacqueline Brown. "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. 2019, pp. 142–147.

[Psa+18] Ioannis Psaras et al. "Mobile data repositories at the edge". In: {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*. 2018.

[Ram+19] Sanjeev Kaushik Ramani et al. "NDN-ABS: Attribute-Based Signature Scheme for Named Data Networking". In: *Proceedings of the 6th ACM Conference on Information-Centric Networking*. 2019, pp. 123–133.

[Red] *Redmine: NFD*. https://redmine.named-data.net/issues/1301. [Online; last accessed Sep 27, 2019].

[Rfc] *RFC1287*. 1991. URL: https://tools.ietf.org/html/rfc1287.

[Sch+16] Klaus Schneider et al. "A practical congestion control scheme for named data networking". In: *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 2016, pp. 21–30.

[Sha+15] Wentao Shang et al. "Ndn-ace: Access control for constrained environments over named data networking". In: *NDN Project, Tech. Rep. NDN-0036, Revision 1* (2015).

[Sha+17] Wentao Shang et al. “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*. 2017, pp. 3–13.

[Shi+16] Ryoki Shinohara et al. “Cache control method mitigating packet concentration of router caused by interest flooding attack”. In: *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE. 2016, pp. 324–331.

[Shi17] Junxiao Shi. “Named data networking in local area networks”. PhD thesis. The University of Arizona., 2017.

[Sig] *Signed interest packet*. 2020. URL: https://named-data.net/doc/NDN-packet-spec/current/signed-interest.html.

[Sig+17] Salvatore Signorello et al. “Advanced interest flooding attacks in named-data networking”. In: *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2017, pp. 1–10.

[SMW02] Neil Spring, Ratul Mahajan, and David Wetherall. “Measuring ISP topologies with Rocketfuel”. In: *ACM SIGCOMM Computer Communication Review* 32.4 (2002), pp. 133–145.

[SNO13] Won So, Ashok Narayanan, and David Oran. “Named data networking on a router: Fast and DoS-resistant forwarding with hash tables”. In: *Architectures for Networking and Communications Systems*. IEEE. 2013, pp. 215–225.

[SNZ17] Junxiao Shi, Eric Newberry, and Beichuan Zhang. “On broadcast-based self-learning in named data networking”. In: *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE. 2017, pp. 1–9.

[Son+15] Tian Song et al. “Scalable name-based packet forwarding: From millions to billions”. In: *Proceedings of the 2nd ACM conference on information-centric networking*. 2015, pp. 19–28.

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

[SS18] Tetsuya Shigeyasu and Ayaka Sonoda. “Distributed Approach for Detecting Collusive Interest Flooding Attack on Named Data Networking”. In: *International Conference on Network-Based Information Systems*. Springer. 2018, pp. 76–86.

[SWS15] Hani Salah, Julian Wulfheide, and Thorsten Strufe. “Coordination supports security: A new defence mechanism against interest flooding in NDN”. In: *2015 IEEE 40th Conference on Local Computer Networks (LCN)*. IEEE. 2015, pp. 73–81.

[Tan+13] Jianqiang Tang et al. “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. 2013, pp. 306–310.

[TTM20] Reza Tourani, George Torres, and Satyajayant Misra. "PERSIA: a PuzzlE-based InteReSt FloodIng Attack Countermeasure". In: *Proceedings of the 7th ACM Conference on Information-Centric Networking*. 2020, pp. 117–128.

[Vas+15] Vassilios G Vassilakis et al. "Mitigating distributed denial-of-service attacks in named data networking". In: *Proceedings of the 11th Advanced International Conference on Telecommunications (AICT), Brussels, Belgium*. 2015, pp. 18–23.

[VMS13] Matteo Virgilio, Guido Marchetto, and Riccardo Sisto. "PIT overload analysis in content centric networks". In: *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. 2013, pp. 67–72.

[Voi+17] Ivan Voitalov et al. "Geohyperbolic routing and addressing schemes". In: *ACM SIGCOMM Computer Communication Review* 47.3 (2017), pp. 11–18.

[Vus+16] Satyanarayana Vusirikala et al. "Hop-by-hop best effort link layer reliability in named data networking". In: *NDN, Technical Report NDN-0041* (2016).

[Wan+13] Kai Wang et al. "Decoupling malicious interests from pending interest table to mitigate interest flooding attacks". In: *2013 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2013, pp. 963–968.

[Wan+14a] Kai Wang et al. "Cooperative-Filter: countering Interest flooding attacks in named data networking". In: *Soft Computing* 18.9 (2014), pp. 1803–1813.

[Wan+14b] Kai Wang et al. "Detecting and mitigating interest flooding attacks in content-centric network". In: *Security and Communication Networks* 7.4 (2014), pp. 685–699.

[Wan+17] Licheng Wang et al. "Economic levers for mitigating interest flooding attack in Named Data Networking". In: *Mathematical Problems in Engineering* 2017 (2017).

[WGQ19] Kai Wang, Dongchao Guo, and Wei Quan. "Analyzing NDN NACK on Interest Flooding Attack via SIS Epidemic Model". In: *IEEE Systems Journal* (2019).

[WSV13] Matthias Wählisch, Thomas C Schmidt, and Markus Vahlenkamp. "Backscatter from the data plane–threats to stability and security in information-centric network infrastructure". In: *Computer Networks* 57.16 (2013), pp. 3192–3206.

[Wu+20] Zhijun Wu et al. "Mitigation measures of collusive interest flooding attacks in named data networking". In: *Computers & Security* 97 (2020), p. 101971.

[WZT+17] Kai Wang, Yude Zhao, Xiangrong Tong, et al. "On the urgency of implementing Interest NACK into CCN: from the perspective of countering advanced interest flooding attacks". In: *IET Networks* 7.3 (2017), pp. 136–140.

[Xin+16] Yonghui Xin et al. "A novel interest flooding attacks detection and countermeasure scheme in NDN". In: *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2016, pp. 1–7.

[Xin+17] Yonghui Xin et al. "Detection of collusive interest flooding attacks in named data networking using wavelet analysis". In: *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE. 2017, pp. 557–562.

[Xyl+13] George Xylomenos et al. "A survey of information-centric networking research". In: *IEEE communications surveys & tutorials* 16.2 (2013), pp. 1024–1049.

[Yeh+14] Edmund Yeh et al. "Vip: A framework for joint dynamic forwarding and caching in named data networks". In: *Proceedings of the 1st ACM Conference on Information-Centric Networking*. 2014, pp. 117–126.

[Yi+13] Cheng Yi et al. "A case for stateful forwarding plane". In: *Computer Communications* 36.7 (2013), pp. 779–791.

[Yin+19] Gubei Yin et al. "Controller Based Detection Scheme of Interest Flooding Attack in Named Data Networking". In: *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE. 2019, pp. 1628–1633.

[Yu+15] Yingdi Yu et al. "Schematizing trust in named data networking". In: *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. 2015, pp. 177–186.

[Zha+10] Lixia Zhang et al. "Named data networking (ndn) project". In: *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC* 157 (2010), p. 158.

[Zha+14] Lixia Zhang et al. "Named data networking". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73.

[Zha+16a] Haitao Zhang et al. "Sharing mhealth data via named data networking". In: *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 2016, pp. 142–147.

[Zha+16b] Yu Zhang et al. "A survey of mobility support in named data networking". In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2016, pp. 83–88.

[Zha+17] Zhiyi Zhang et al. "NDN certificate management protocol (NDNCERT)". In: *NDN, Technical Report NDN-0050* (2017).

[Zha+18a] Haitao Zhang et al. "NDN host model". In: *ACM SIGCOMM Computer Communication Review* 48.3 (2018), pp. 35–41.

[Zha+18b] Yu Zhang et al. "Kite: Producer mobility support in named data networking". In: *Proceedings of the 5th ACM Conference on Information-Centric Networking*. 2018, pp. 125–136.

[Zha+18c] Zhiyi Zhang et al. "An overview of security support in named data networking". In: *IEEE Communications Magazine* 56.11 (2018), pp. 62–68.

[Zha+18d] Zhiyi Zhang et al. "NAC: Automating access control via Named Data". In: *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE. 2018, pp. 626–633.

[Zha+19] Zhiyi Zhang et al. "Expect More from the Networking: DDoS Mitigation by FITT in Named Data Networking". In: *arXiv preprint arXiv:1902.09033* (2019).

[Zhi+19] Ting Zhi et al. "Resist Interest Flooding Attacks via Entropy–SVM and Jensen–Shannon Divergence in Information-Centric Networking". In: *IEEE Systems Journal* (2019).

[ZL19] Xin Zhang and Ru Li. "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. 2019, pp. 10–15.

[ZLL18] Ting Zhi, Hongbin Luo, and Ying Liu. "A Gini impurity-based interest flooding attack defence mechanism in NDN". In: *IEEE Communications Letters* 22.3 (2018), pp. 538–541.

[ZLW20] Ting Zhi, Ying Liu, and Jun Wu. "A Reputation Value-Based Early Detection Mechanism Against the Consumer-Provider Collusive Attack in Information-Centric IoT". In: *IEEE Access* 8 (2020), pp. 38262–38275.

[ZLZ15] Meng Zhang, Hongbin Luo, and Hongke Zhang. "A survey of caching mechanisms in information-centric networking". In: *IEEE Communications Surveys & Tutorials* 17.3 (2015), pp. 1473–1499.