People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Ammar Telidji University
Laghouat
Sciences Faculty
Mathematics and Computer Science Department
Doctoral School STIC
Information and Communication Sciences and Technologies

THESIS

submitted in partial fulfillment of the requirements for the degree of

PhD in Computer Science

Option: Distributed and Mobile Computing(**IRM**)

Presented by:

Benarfa Abdelmadjid

Theme:

Communication security in the Future internet architectures

Defended publicly in front of the jury composed of:

Mr.	Youcef Ouinten	President	Professor	Laghouat University
Mr.	Mohamed Benmohammed	Examiner	Professor	Constantine University
Mr.	Mohamed Amine Ferrag	Examiner	MCA	Guelma University
Mr.	Chaker Abdelaziz Kerrache	Examiner	MCA	Laghouat University
Mr.	Noureddine Chaib	Examiner	MCA	Laghouat University
Mr.	Mohamed.B. Yagoubi	Supervisor	Professor	Laghouat University
Mr.	Mauro Conti	Co-supervisor	Professor	Padova University, Italy

year 2021

To my parents,

....

To my brothers and sisters,

....

To my friends,

....

....

I dedicate this modest work

Benarfa Abdelmadjid

Acknowledgments

I want to thank in the first place my God who gave me the health, the physical and intellectual strength and the courage to carry out this work.

I warmly thank my parents who helped me a lot with their advice, their encouragement from my childhood until I arrived at this point, not forgetting my sisters, my brothers and all my family.

Doing a PhD is learning to become a researcher. It's a long and bumpy path that can not be done without much support.

That's why I'd like to thank my PhD supervisor ***M.B YAGOUBI*** *for his supervision, and* ***M. CONTI*** *for all the work meetings and the scientific and personal discussions that we have had, not forgetting all the SPRITZ lab members of Padova university in Italy.*

I would also like to thank the examination committee members for the brilliant comments and suggestions, I also want to thank you for letting my defense be an enjoyable moment, thanks to you.

Thank you to all who have been involved in this work from near and far.

Benarfa Abdelmadjid

ملخص

تهدف العديد من الجهود البحثية الجارية إلى تصميم بنيات الإنترنت المستقبلية المحتملة ، من بينها شبكة البيانات المسماة (NDN) التي تقدم تحولًا من البنية التحتية للإنترنت القائمة على بروتوكول الإنترنت (IP) القائم على المضيف إلى بنية موجهة للمحتوى. ومع ذلك ، فقد حدد الباحثون بعض قيود التصميم في NDN ، والتي من بينها تمكن البعض من بناء نوع جديد من هجوم رفض الخدمة الموزع (DDoS) ، والمعروف باسم هجوم إغراق الفائدة (IFA). في IFA ، يصدر أحد الخصوم طلبات غير مرضية في الشبكة لإشباع جدول الفوائد المعلقة (PIT) لأجهزة توجيه NDN ومنعها من التعامل بشكل صحيح مع حركة المرور المشروعة. كان الباحثون يحاولون التخفيف من هذه المشكلة من خلال اقتراح العديد من آليات الكشف والتفاعل ، لكن جميع الآليات المقترحة حتى الآن ليست فعالة للغاية ، بل على العكس ، تلحق أضرارًا كبيرة بحركة المرور المشروعة. في هذه الرسالة، نقترح آلية جديدة لاكتشاف IFA والتخفيف من حدته ، تهدف إلى تقليل استهلاك الذاكرة لـ PIT عن طريق تقليل حركة المرور الضارة التي تمر عبر كل جهاز توجيه NDN بشكل فعال. على وجه الخصوص ، يستغل بروتوكولنا استراتيجية إدارة فعالة على PIT ، والتي من خلالها يتم إزالة الفائدة الضارة (MIs) المخزنة بالفعل في PIT وإسقاط MIs الواردة الجديدة. بالإضافة إلى ذلك ، يوفر الإجراء المضاد المقترح جدار أمان إضافيًا على حواف الشبكة لاكتشاف وتخفيف الهجوم في أقرب وقت ممكن وتحسين خدمات الشبكة ، أي إشغال أجهزة التوجيه PIT أثناء IFA. لتقييم فعالية عملنا ، قمنا بتنفيذ الإجراء المضاد المقترح على جهاز محاكاة ndnSIM مفتوح المصدر وقارننا فعاليته مع أحدث التقنيات. تظهر النتائج أن الإجراء المضاد المقترح لدينا يقلل بشكل فعال من أضرار IFA من حيث الحفاظ على حركة المرور المشروعة وتوافر أجهزة التوجيه PIT. بالنظر إلى حركة المرور المشروعة، يزداد مقدار الفوائد الحميدة (BIs) المحفوظة من خلال نهجنا من 5 ٪ إلى 40 ٪ فيما يتعلق بالحفاظ المضمون بأحدث الحلول. فيما يتعلق بتوافر PIT للموجهات، يضمن نهجنا ترك 97 ٪ من حجم جدول PIT متاحا للتعامل مع حركة المرور المشروعة.

كلمات مفتاحية : IFA ,هجوم DDos ,NDN , شبكة البيانات المسماة ,احتقان

Résumé

Plusieurs efforts de recherche en cours visent à concevoir de futures architectures Internet potentielles, parmi lesquelles le Named-Data Networking (NDN) introduit un changement de l'infrastructure Internet existante basée sur le protocole Internet (IP) centrée sur l'hôte vers une infrastructure orientée contenu. Cependant, les chercheurs ont identifié certaines limitations de conception dans NDN, parmi lesquelles certaines permettent de créer un nouveau type d'attaque par déni de service distribué (DDoS), mieux connu sous le nom d'attaque par inondation d'intérêt (IFA). Dans l'IFA, un adversaire émet des demandes non satisfaisables dans le réseau pour saturer le tableau des intérêts en attente (PIT) des routeurs NDN et les empêcher de gérer correctement le trafic légitime. Les chercheurs ont tenté d'atténuer ce problème en proposant plusieurs mécanismes de détection et de réaction, mais tous les mécanismes proposés jusqu'à présent ne sont pas très efficaces et, au contraire, endommagent fortement le trafic légitime. Dans cette thèse, nous proposons un nouveau mécanisme de détection et d'atténuation des IFA, visant à diminuer la consommation de mémoire du PIT en réduisant efficacement le trafic malveillant qui passe par chaque routeur NDN. En particulier, notre protocole exploite une stratégie de gestion efficace sur le PIT, à travers laquelle les intérêts malveillants (MI) déjà stockés dans le PIT sont supprimés et les nouveaux MI entrants sont supprimés. En outre, la contre-mesure proposée fournit un mur de sécurité supplémentaire sur les bords du réseau pour détecter et atténuer l'attaque le plus tôt possible et améliorer les services du réseau, c'est-à-dire l'occupation des routeurs PIT pendant l'IFA. Pour évaluer l'efficacité de notre travail, nous avons implémenté la contre-mesure proposée sur le simulateur open-source ndnSIM et comparé son efficacité avec les dernier travaux cités. Les résultats montrent que notre contre-mesure proposée réduit efficacement les dommages IFA à la fois en termes de trafic légitime préservé et de disponibilité des routeurs PIT. Compte tenu du trafic légitime, le flux des Intérêts Bénins (BI) préservés par notre approche augmente d'une manière notable 5% à 40% par rapport à la préservation garantie par les autre solutions cités. Idem en ce qui concerne la disponibilité des routeurs PIT, notre approche garantit que les 97% de la taille du PIT sont laissés libres pour gérer le trafic légitime.

Mots-clés : NDN, IFA, Congestion, DDOS.

Abstract

Several ongoing research efforts aim to design potential Future Internet Architectures, among which Named-Data Networking (NDN) introduces a shift from the existing host-centric Internet Protocol (IP)-based Internet infrastructure towards a content-oriented one. However, researchers have identified some design limitations in NDN, among which some enable to build up a new type of Distributed Denial of Service (DDoS) attack, better known as Interest Flooding Attack (IFA). In IFA, an adversary issues non-satisfiable requests in the network to saturate the Pending Interest Table (PIT) of NDN routers and prevent them from properly handling the legitimate traffic. Researchers have been trying to mitigate this problem by proposing several detection and reaction mechanisms, but all the mechanisms proposed so far are not highly effective and, on the contrary, heavily damage the legitimate traffic. In this thesis, we propose a novel mechanism for IFA detection and mitigation, aimed at decreasing the memory consumption of the PIT by effectively reducing the malicious traffic that passes through each NDN router. In particular, our protocol exploits an effective management strategy on the PIT, through which the Malicious Interest (MIs) already stored in the PIT are removed and the new incoming MIs are dropped. In addition, the proposed countermeasure provides an additional security wall on the edges of the network to detect and mitigate the attack as early as possible and improve the network health, i.e., routers PIT occupancy during IFA. To evaluate the effectiveness of our work, we implemented the proposed countermeasure on the open-source ndnSIM simulator and compared its effectiveness with the state-of-the-art. The results show that our proposed countermeasure effectively reduces the IFA damages both in terms of preserved legitimate traffic and availability of routers PIT. Considering the legitimate traffic, the amount of Benign Interests (BIs) preserved by our approach increases from 5% till 40% with respect to the preservation guaranteed by the state-of-the-art solutions. Concerning the routers PIT availability, our approach guarantees that the 97% of the PIT size is left free for handling the legitimate traffic.

Keywords: NDN; DDoS attack; IFA; Congestion; PIT management.

Contents

List of Figures

List of Tables

General Introduction

The existing Internet architecture was deployed in the early 70's to address the elementary communication requirements which were obligatory at that period. In particular, it was designed to guarantee a reliable host-to-host inter-connectivity. However, over the past few years the Internet has increasingly shown a poor match with its initial design, it started moving towards a content distribution and retrieval paradigm. Today, the Internet produces immense and ever-growing traffic volumes, mainly due to the multiple distribution of the same popular contents, which inevitably introduces high costs for network operators such as additional Content Delivery Networks (CDN) deployment. Numerous solutions have been proposed in recent years to narrow the gap between the Internet design and its current usage (see [8], [9] for an overview). Among the different solutions proposed under the Future Internet Architecture (FIA) program sponsored by National Science Foundation (NSF), Named Data Networking (NDN) [10], which follows the Content-Centric Networking (CCN) approach, uses caching in the network, multi-party communication through replication, and models of interaction that decouple senders and receivers [11], [12]. NDN explicitly addresses the data (content) itself instead of the physical location of the hosts in the network, therefore, converting data into the first-class entity. Instead of creating a direct connection between two hosts to exchange content, the consumer in NDN directly requests the name of the content by issuing an interest. The network then handles the request by efficiently finding and retrieving back the closest copy of the relevant content. This decoupling of time and space among request resolution and content transfer enables NDN to provide security [13], to be more resilient to disruptions, content distribution, mobility [14], and storage [15] as native features belonging to the network architecture.

The work that we present in this thesis mitigates one of the most significant attacks in NDN: the *Interest Flooding Attack* (IFA). This is an NDN-customized DDoS attack, in which the adversaries can send intemperate number of spoofed Interest packets to deplete the memory resources of on-path routers. As a result, all subsequent incoming legitimate requests will be dropped. Due to this, IFA is almost the highest severity attack among all sorts of attacks in NDN.

Motivation and Contribution

Regardless of the substantial quantity of research on NDN security, we identified that the proposed defence mechanisms [16, 17, 18, 19, 20] have one or more of the following limitations. First, the detection of an attack is not robust and accurate, if applied close to the content providers and to the victims and the volume of adversarial traffic is large [21]. Second, the legitimate traffic is likely to be damaged, since most of the proposed countermeasures [17, 16, 20] limit the rate of incoming

traffic and are not able to differentiate between benign and malicious packets, thus resulting in unfair punishments. Third, since each router has to perform first an attack detection and then an attack mitigation, during the first phase, most of the approaches are likely to encounter harmful consequences. Finally, the proposed collaborative mechanisms [19, 17, 16, 21] introduce unnecessary overhead given by the extra messages exchanged among routers.

With the purpose of efficiently defending against IFA, we propose an efficient countermeasure, named as Choose To Kill IFA (ChoKIFA), which mitigates the damages caused by IFA by differentiating the malicious traffic from the legitimate one, and by reducing the former. To distinguish between benign and malicious interests, the proposed protocol exploits the active queue management scheme, i.e., CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows (CHOKe) [22], at each router. ChoKIFA performs a series of operations before an incoming interest enters the PIT of the router. In particular, it allows intermediate routers to control each incoming interest and aims at differentiating and penalizing only the malicious ones, thus preventing the appearance of attack consequences in the network. The initial results depicting the impact of attack and efficacy of our proposed countermeasure is reported in [23]. We further extended the attack scenarios and analysis, and proposed an enhanced version of ChoKIFA, named as *ChoKIFA+: An Early Detection and Mitigation Approach against Interest Flooding Attacks in NDN* [24]. In ChoKIFA+, the edge routers which are connected directly to the consumers provides an additional security wall. This aims to detect the malicious behaviour of the users as-close to the adversary, and react accordingly to reduce the amount of malicious traffic traversing towards the network. ChoKIFA+ further improves the network health during IFA, i.e., in terms of PIT occupancy. In particular, it provides a very large capacity available on router's PIT to process legitimate traffic during an attack. The final aim of ChoKIFA+ is to make routers able to independently detect an attack in progress as-soon and as-close to the adversary as possible, while maintaining the simplicity of forwarding and improving the network health. To this end, the major contributions of this work are as follows.

- We design and implement a detection and mitigation mechanism against IFA by adopting the CHOKe approach. ChoKIFA does not require a global network state information, and without any delay penalizes the malicious traffic by both dropping the new incoming malicious interests and removing the ones already stored in the PIT.

- We design an additional security mechanism on the edge routers, named as ChoKIFA+. This approach aims to quickly identify and block IFA attempts through the edge routers thus further improves the network health during the attack. In particular, in ChoKIFA+, the edge routers detect the malicious behaviour of the users as-close to the adversary, therefore, reduces the amount of malicious traffic traversing towards the network.

- We evaluate the effectiveness of our work by implementing both ChoKIFA and its enhanced version on the ndnSIM[1] simulator [25]. We compare our proposed countermeasure with the state-of-the-art mitigation approaches [16].

[1]ndnSIM implements the NDN protocol stack on NS-3 simulator.

The results show that ChoKIFA+ effectively mitigates the adverse effects of IFA in the network. In particular, ChoKIFA+ is able to guarantee an interest satisfaction rate up to 98% and it shows up to 40% less false positives in comparison with other mitigation approaches. Those ones use a rate limiting approach based on interest satisfaction together with pushback announcements [16].

Organization

The remainder of this thesis is organized as follows. We present the main future internet architectures in chapter 1, and we detailed the NDN architecture in Chapter 2. Chapter 3 reserved to present IFA and the related work on IFA and DDoS countermeasures and state of the art mitigation mechanisms. Our ChoKIFA approach exploiting AQM techniques. is presented in chapter 4. Finally, Chapter 5 describes the proposed protocol including system, adversary model and working methodology of ChoKIFA+, we present the implementation, evaluation and comparison of proposed countermeasure against IFA.

1 Next-generation Internet Architectures

Contents

This chapter is divided into three main parts. The first part describe in brief the current internet architecture and motivates the need for Information-Centric Networking ICNs in general. The second part provides a background on the different proposals for the future Internet. The last part explained in details the proposed future internet architectures, This is followed by comparing, reviewing and analyzing proposed solutions in the literature.

1.1 The current internet

Nowadays, internet is one of the most famous and innovative technologies in the world, born from a study developed by the U.S. government in the early seventies. It represents a product of four decades of evolution and its superb success story is owing to the worldwide spread of TCP/IP. following the sole existing communication system, telephony, TCP/IP protocol was created. This leads to base TCP/IP protocol on host-to-host communication.

Therefore, host-to-host communication model became the principle core of the Internet design. In this design, and to be able to interact with other machines in the network, each machine must have an IP address and follow the TCP/IP protocol.

Although TCP/IP has been doing the process properly, the manner in which people access and use Internet has changed radically and host-to-host communication do not ever meet the current user needs.

1.1.1 Usage

Smartphones, tablets, and laptops allow users to be on-line continuously; thus, sharing contents and retrieving data from Internet became a very frequent activity in our life. Every day, The quantity of data transmitted over the Internet is growing. A latest Cisco research [26] indicates that in 2018, the amount of global IP traffic was 1.5 Zettabyte[1]. Moreover, Cisco has anticipated that the amount of traffic will increment triple by 2022, forming a 100-fold rise from 2005 to 2022 (3-fold rise from 2017 to 2022) [27]. In Fig. 1.1, we assume that the reasons behind this huge amount of data are:

- **Users evolution:** Every year, the number of Internet users increases. Originally, the Internet's users were for the most part academics, researchers and organizations employees. However as of nowadays, 62% of the world population (4.8 billion users) are connected to the Internet [28], [26].
- **Devices evolution:** The number of devices linked to the Internet will be nearly three times the global population, it is foreseen that in 2021. In addition, 3.5 devices per person on average will be in 2021 compared to 2.3 devices in 2016. Quantity of devices is not the sole game-changer, the technology is evolving too. Devices were mostly servers and mainframes, in the early days of the Internet. Later, the Personal Computers and then Laptops were introduced, allowing regular users to connect from anywhere to the Internet. The evolution then saw strong cell phones linking to the Internet, creating the booming smartphone industry. Currently, new concepts are being commercially introduced and used by many users, such as the wearable devices and Internet-connected vehicles.

[1] 1 Zettabyte = 1 billion Terabytes

- **Content evolution:** Since the start of the Web, the sort of content on the Internet has changed due to the creation and use of new technologies in various internet applications. For example, started with texts in web pages, but now Full HD movies can be streamed to regular users (78% of the world's mobile data traffic will be video by 2021 [27]). Not just the type is changing, but also the quality. With all the new technologies in photos and movies, content size is increasing exponentially. Moreover, the use of social networks, blogs, wikis, etc, has allowed users to share and collaborate content on the World Wide Web, which is another huge source of content that is being produced [29].

1.1.2 Design

The architecture of the Internet is based on the TCP / IP model. The IP protocol, used to enable network interconnection, was designed to transmit messages (datagrams) from a source to a destination. It does not have in itself any reliability mechanism (best-effort network) but can take advantage of the end-to-end services (flow control, error control) offered by the TCP transport layer. Network hosts are identified by a unique address, 32-bit (IPv4) or 128-bit (IPv6), called an IP address. It is hierarchical and consists of two parts: the network address and the machine address within this network.

To enable entities to communicate with each other, the hosts of the Internet use the IP protocol and are identified via their IP addresses. The Internet is then based on the client-server model, where clients-host make requests to servers-host. This is the case of the Web where customers send their requests to web servers (eg Facebook.com, Google.com) who return the content (web pages in this case). This predominant client-server model at the beginning of the Internet required some adaptations with the democratization of the Internet. Indeed, beyond textual content (web, email) new multimedia content was consumed via download or streaming site, greatly increasing the volumes of the traffic exchanged. The architecture of the Internet was not originally designed for transporting such amounts of data, all these new changes have brought to light the limits of the current Internet architecture.

1.1.3 Challenges

The present Internet architecture is host-based, highly depending on location, which is allowed by using Internet Protocol (IP) addresses. Today's Internet utilization and needs are quite distinct from its beginnings 1.2. When the Internet was originally designed and implemented, it was mainly used to disseminate research data and share expensive resources, not content [30]. Security was an afterthought, as the need was far less in the earliest years of its existence. Today, Internet users are increasingly mobile and more interested in content, regardless of what server

or source the desired content comes from.

In addition a variety of things are expected to get connected to the Internet, billions of them. These things operate over multiple domains such as construction,transportation, agriculture, energy, weather, health, etc. This phenomenon, is changing the architecture of the Internet. These devices are extremely heterogeneous and have hardware limitations. They have lower power consumption, CPU, and memory usage in order of magnitudes. They generally have various interfaces across distinct communication protocols and have a restricted amount of configuration options. TCP/IP is an end-to-end communication protocol and expects the application layer to provide such services.

Internet traffic is also increasing quickly due to on-demand content such as movies and TV shows from famous content providers such as Netflix and YouTube, which are anticipated to be supplied in a timely, streaming way. A more detailed analysis shows that Netflix (31.6%) and YouTube (18.7%) combined, account for more than 57% of downstream traffic in fixed access [31].

The Internet was not designed for such enormous amounts of content and performance. As a result, many researchers have suggested that the Internet needs a fundamental re-design, such that it is information-centric rather than host-centric.

Figure 1.1 – *IP traffic growth [5]*

1.2 The future internet

Due to the growing focus on content, instead of on hosts, many researchers have suggested that it's necessary to set up a new Internet architecture that responds better to the exponential increase in demand for content.

New networking paradigms like Information-Centric Networking (ICN) provide alternatives to these issues. ICN could be a clean-slate network architecture for future Internet. There are many promising proposals for an ICN design. Any new proposal for the future Internet must provide the following characteristics.

First, must be named-data at the core of the networking, and names be decoupled from content location, applications, storage or media of transport. Decoupling data name and its location gives ICN native support for mobility since the users only need to know the name of the content and not where the content is situated Fig.1.2. Second, supports data security and privacy requirements by enabling digital signature and encryption. This solution is not only agnostic about the source of the content but also gives us the capability of in-network caching for all contents. In-network caching will help to place popular content near the consumer to lower the latency, which will result in a better utilization of the infrastructure and will increase the throughput.

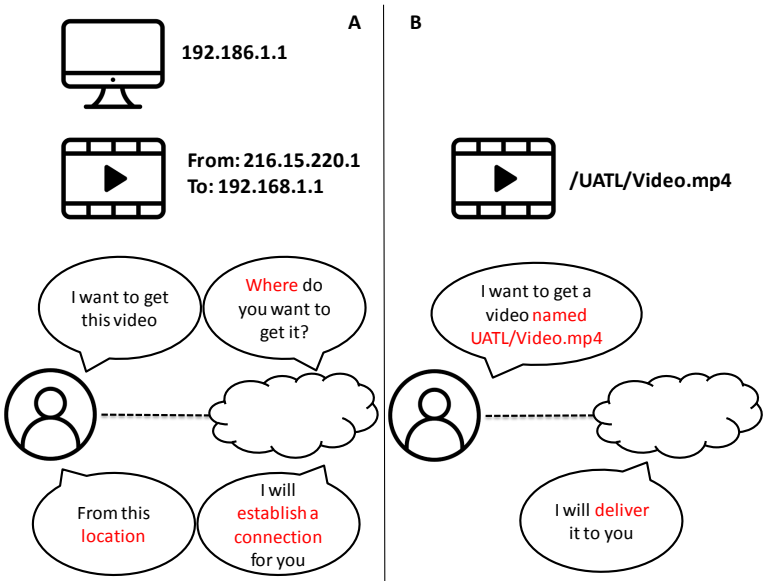

Figure 1.2 – *Communication pattern of (A) IP network and (B) ICN Networking*

1.2.1 Concept, Basic Design and Common Characteristics

In this section, we will review the concepts and terminologies that are common between ICN designs.

- **Naming:**

  Packets are content carriers in the Internet, they travel through the network to transport data between two hosts recognized by IPs only. In addition, packets are transmitted using the receiver's IP which is attached to the packet. However, Because the content in an information-centric network is the focus (rather than hosts), the way by which we identify that content becomes crucial. The relation between hosts and data should be decoupled in order to design a network with content as the main block. Specifically, content should be named in such a way that packets are routed using the name not the IP of the host.

Ideally, a powerful ICN design must build up a scheme that is both efficient and effective in distinguishing between distinct information objects (data, content, services, etc.) The namespace, which is the number of available names, must be enormous in such architecture. In [32] the authors estimate that the namespace must be between 10^{13} and 10^{15} . That's why the naming scheme is a large challenge for the ICN, Naming must provide the following [33]:

(i) Uniqueness: content should be identified in a unique way.
(ii) Persistence: guarantee the validity of the content name.
(iii) Scalability: the name-space scales-up with no restriction on the amount of content.

In general, ICN has two kinds of naming schemes, namely flat and hierarchical [34]. The flat naming scheme identifies a content with a randomly-looking bit string. This scheme uses self certifiable names [35], the name is random bits created from the content itself not the location (e.g., hash of the content). However, the name in this case is not readable. The most popular way to calculate a flat name is to use a cryptographic hash function, for example, on the data itself. one advantage of flat names is that they can be made easier to self-certify. This reduces or eliminates the need for a third-party authentication source, increases effectiveness and could also decrease network traffic overall.

On the other hand, the hierarchical name has a similar structure to the Uniform Resource Identifier (URI) concept of the Internet [36], which consists of many name components (the content provider, content's name and version or segment). Such as UATL/cours/ICN.pdf. the name is structured hierarchically to readily locate the content. However, it is essential to remember that hierarchical names do not necessarily have to be human-readable.

- **Routing:**

  In contrast to the current design of the Internet, where communications are initiated by the source (i.e., IP-based, IP addresses attached to the packet), ICN communications start at the receiver. Since the new packets (Request and Data) are not IP-based. Thus, ICN would have a new method of generating a route through which the data travels from the producer (provider) and the consumer (requester). The routing process has two phases:

  (i) Named-based Routing (request data): routing the request to the data source. (i.e., request Data packet routing).
  (ii) Routing back the data (carrying the data) from the source (or the stored copy) to the requester (i.e., Data packet routing). How the name is resolved can dramatically affect how the data is routed. There are two primary approaches in which the name resolution and data routing characteristics of an

ICN architecture are associated together. One of these ways is a :

- Coupled (or integrated) model: the request is sent directly to the source of the data, while the content takes the reverse path back to the customer. As a result, the data routing path is also determined by the network entity responsible for name resolution; In a coupled approach, a request for information is sent to the publisher of the content or a content storage source containing the requested content (usually via multiple hops across different routers) Then, when the publisher responds with the content itself, the content travels the same path as the initial request, just in reverse order.

- Decoupled (or independent) model: the request would first be forwarded to a dedicated entity for name resolution. This entity will then reply to the consumer with information about data sources. As such, the user can choose the appropriate data source to use. Another option is that the name resolution entity chooses the ideal data source and directs it to send the data back to the user. In a decoupled approach, on the other side, the name resolution service doesn't determine the route between subscriber and publisher in a way that causes the data to travel the reverse path of the request.

- **Caching:**

  With the daily rise in the amount of content requested, the notion of caching has become of paramount significance [37]. Content caching refers to storing data for a specified time at locations other than the source (i.e., usually closer to end-users) such that future requests of the same data can be satisfied much faster [38] [39]. In addition, by implementing an optimized caching strategy, servers would not waste resources responding to the same request from various users. Moreover, the network infrastructure would save resources required for data forwarding.

  ICN architectures can save network resources and enhance the delivery performance by using content caching on routers, particularly in term of the delivery delay. By decoupling information and its location, the named data can be stored anywhere in the network. All nodes, whether it's an end-user device or a router, will be able to cache content that passes through it. What to cache, and for how long, are decisions that influence the efficiency of the network. For instance, content's popularity, size and version are some of the factors that the network designer should consider in optimizing the performance of the cache [37] [40].

  ICN caching can be classified into two types, on-Path and Off-path caching [37] [41]:

  - In on-path caching schemes, content is cached at nodes along the data forwarding paths. In other words, once a network entity receives a request, it

gives priority instead of forwarding the request, addressing the request with cached data, if possible.

- Off-path caching schemes replicate data in the network, to improve availability, regardless of the forwarding paths. The network exploits the cached copies situated outside the name resolution request path. In particular, the name resolution component must know the cached copy's locations and manage them as a data provider with the decoupled name resolution and data routing. Off-path caching is not compatible with the coupled model of name resolution and data routing, since the request and the content always follow the same path. Central entities (e.g., ISP) are responsible for developing a caching strategy that strikes a balance between content availability and caching overhead [42].

- **Security:**

  In TCP/IP, security is obtained by encrypting the transmission channel (path of content) plus authenticating the end points of communication. [12] [43]. This model requires the client to trust the server and all the intermediate nodes on the path. There is no way in this model to provide the authenticity of the data itself instead of the path [35], and we have to trust the data container. The content can therefore be given by nodes other than the data provider. Moreover, TCP/IP is the network mainly focuses on forwarding the traffic to the indicated destination. no matter who has sent the packet, this results in a power imbalance between the sender and the receiver. It has no mechanism to reduce redundant or unsolicited traffic . However, in ICN, content can be protected from alteration or eavesdropping and only genuine copies of the data can exist in the network. Also, such communication also decouples the users and the data sources, protecting their privacy during data exchange. Furthermore, Protecting the content provides name-data integrity and authenticity anywhere in the network.

The main differences between the two models TCP/IP and ICN summarizes as follows in table 1.1.

1.3 Main future internet architectures

The idea of moving from a host-to-host network architecture to a content-oriented network is more than a decade old. In fact, The first architecture according to ICN paradigm was introduced in 1999 with TRIAD [1] project. However, a few works have been constructed on this project, probably because the content at that time did not have the same huge weight that it currently takes in Internet traffic. A few years later, several architectures are beginning to shed light on research in the field of content-oriented networks. In this section, we provide a brief introduction and overview of different ICN approaches such us: DONA, PURSUIT, NetInf and NDN.

Property	ICN	IP
ID	Content Name	IP Address
Type of packets	Data Packet/Interest Packet	Several packages
Loop	Essentially without loops using nonce	Not guarantee
Hop count	Can predict several steps	The only predicts a next hop
Forwarding Strategy	YES	NO
Separating routing and forwarding	YES	NO
State	Stateful	Stateless
Security by design	YES	NO
Convergence Time	Little	Much
Load Balancing	YES	NO
Security	Using signature packet is secure	Using extensions like Spec

Table 1.1 – *Comparison between IP and ICN*

1.3.1 Translating Relaying Internet architecture integrating Active Directories (TRIAD) [1]

Content names in TRIAD architecture are human-readable, location-independent names. TRIAD architecture defines a new content layer and bases contents distribution on it. This new layer offers scalable content routing, caching, content transformation and load balancing, integrating naming, routing and transport connection setup.
Content layer not only returns content, but may return a network pointer if the required content is large or indeterminate in length. In order to be compatible with the World Wide Web, the contents are located with the Web Uniform Resource Locator (URL), optionally augmented with cookies. In addition, this network pointer is realized as an HTTP/TCP connection through which the content is read or written.
The content layer is implemented by: (1) content routers that guide the request towards content servers that store the content; (2) content servers that provide content services; and (3) content caches that transparently store come back content closer to the client than the servers themselves.
TRIAD depends on trustworthy directories to authenticate content lookups (but not content itself). It also implements an IP-sec-like mechanism for end-to-end security. For additional security, the authors of [1] suggest that the network be limited to mutually trusting content routers.

1.3.2 Data Oriented Network Architecture (DONA)[2]

DONA is one of the first full ICN designs, developed by the University of California - Berkeley and it is a continuation of the TRIAD architecture. It is mainly based on the use of new content naming mechanisms since it still uses the TCP / IP architecture.

In order to identify the contents, DONA replaces the traditional name resolution system (DNS) with a name resolution system that follows a flat naming scheme. Each content is then designated by the concatenation of two identifiers: (i) the first identifier corresponds to the result of a hash function of the public key of the user who suggested the content (the publisher); (ii) the second identifier is a name chosen by the user. The nodes, called *Resolution Handlers (RH)*, are hierarchical and at each new content recording, the RH warns its peers as well as its parents so as to propagate the information and allow name resolution and routing thereafter.

Each node of the DONA architecture thus has a transmission table allowing it to send the messages to the next node, thus offering the possibility of transmitting the requests to the nearest source. But without an advertisement system to manage the data stored in cache. Once the data is located, it is routed to the destination via point-to-point communications between receiver and source via the current Internet.

Figure 1.3 illustrates the operating principle of the DONA architecture. A user send a FIND message to access data stored in the server (message 1). The node RH1 doesn't know a path to this data, it therefore transmits the FIND request to its parent RH4 (message 2), which transmits it to its peer RH5 (message 3), since it has no rules to access the data, nor parent. RH5 knows the road to access the data thanks to the advertisements previously generated by the server, the request is thus transmitted to the server (message 4 and 5). The server responds to the user by sending the data via a conventional client / server communication of the Internet, the data does not necessarily follow the same path as the request (message 6 to 9). The intermediate nodes (RH1, RH2 and RH3) can decide to store the contents they carry in the cache memory, when the user will send a next request for one of these contents (message 10), RH1 will respond with the copy that he owns (message 11).

The DONA architecture still depends on the current Internet architecture, but brings it a new name resolution that will allow effective distribution of content by alleviating the network load (multiple servers, caching, etc.).

1.3.3 Publish-Subscribe Internet Technologies (PURSUIT) [3]

The PURSUIT architecture [3], which follows its predecessor PSIRP (Publish-

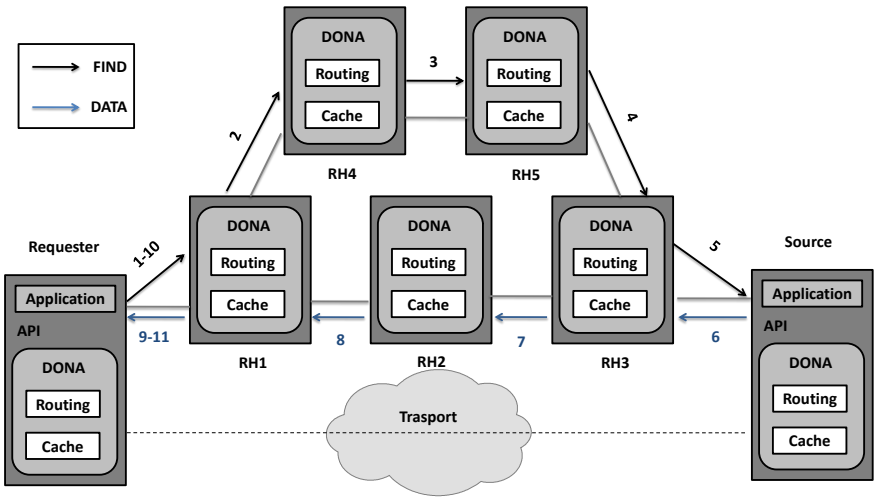

Figure 1.3 – *DONA architecture*

Subscribe Interest Routing Paradigm) [44], was proposed by the European research project of the same name (FP7 EU PURSUIT). This architecture uses a publish-subscribe model to deliver the contents.

PURSUIT is realized based on three components: *Rendez-vous Identifiers (RIds)*, *Topology manager*, and *forwarding*. The Rendezvous is the link between subscriptions and publications. The topology manager manages the topology and generates the paths between publishers and subscribers in order to route the content. Topology managers execute a distributed routing algorithm (OSPF type) in order to discover the different possible routes. Finally forwarding allows to route the data by following the path set up by its topology manager until reaching the subscriber.

To name the information, two identifiers are used: *the scope ID* to define the category of the information, such as a music, an image or a video, and *The Rendez-vous ID* which identifies the object itself by a flat naming. To be more visible and improve its reach, content can be published multiple times with different scope IDs. This means that a content has an ID rendezvous but may have multiple ID scopes. This flat naming mode therefore requires an entity allowing the resolution of names. This is done by *the rendezvous*, which will decide the path to follow for each request.

As shown in Figure 1.4, the publisher advertises new content to its nearest rendezvous node, located through a distributed hash table or Distributed Hash Table (DHT) present in the forwarding entities (messages 1 and 2). When a subscriber is interested in a content, he contacts his local rendezvous node by specifying the desired content (messages 3 and 4). This one will go back in the chain of rendezvous to find the content (messages 5, 6 and 7). Once found, the rendezvous contacts the topology manager to configure it and create the route between the subscriber and the publisher owning the content (messages 8 and 9). For this, the topology manager uses a distributed routing protocol (OSPF type) to calculate the path. Once the publisher is notified of the new route set up (messages 10 and 11), he uses his

forwarding function to transfer the content to the subscriber (messages 12, 13, 14 and 15).

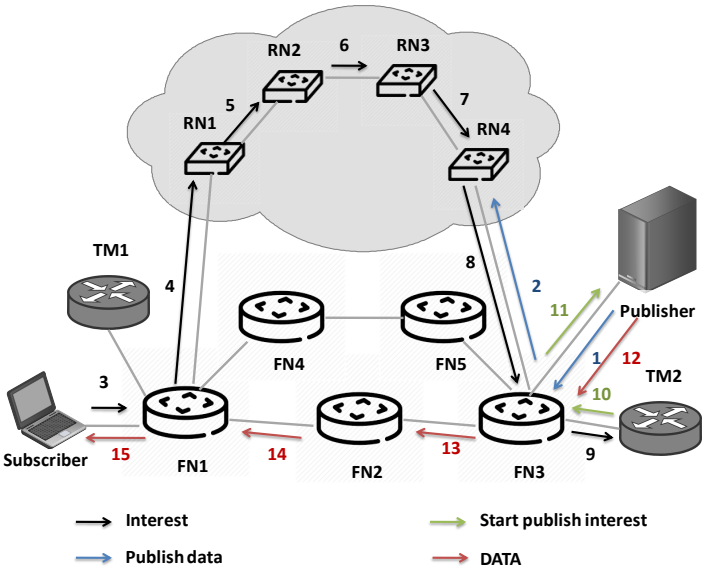

Figure 1.4 – *PURSUIT architecture*

1.3.4 Network of Information (NetInf) [4]

Developed by the *Scalable and Adaptive Internet Solution* (SAIL) project [45] composed of operators, vendors and researchers aiming to develop the networks of the future, NetInf [4] is intended to be a network architecture that can be deployed on a scale of the Internet in a robust and reliable manner. It aims to connect several different technologies to each other in order to be deployed gradually. To do this, elements of the PURSUIT and Content-Centric Networking (CCN) architectures are combined.

NetInf uses the same naming as PURSUIT, which consists of two parts. The first corresponds to the description of the entity that created the object and uses a flat naming, while the second uses a hierarchical name and allows to describe the object.

It uses a multi-level DHT organization based on global name resolvers. These resolvers identify the first part of the name (representing the entity that created the object) in order to know where to send the request to find the content. The second part of the name (describing the object in a hierarchical way) is resolved by a local resolver, which allows to find what content is requested. Global resolvers act as IP routers, routing the request to the appropriate network. In addition, each node of the network has a cache for storing the contents that are transiting.

The NetInf operating mode is shown in Figure 1.5. The publisher announces the contents it has to its local name resolver (message 1) which transmits it to the global

resolver (message 2). In order to find out how to access content, the subscriber sends a message to his local name resolver (message 3). This local resolver will then contact the global resolver to find the publisher (message 4). The global resolver returns the information on the content location to the user (messages 5 and 6) who will then be able to send the request and propagate it to the publisher (messages 7, 8, 9 and 10). Once the request is received, the publisher responds to the message by sending the desired content (messages 11, 12, 13 and 14).

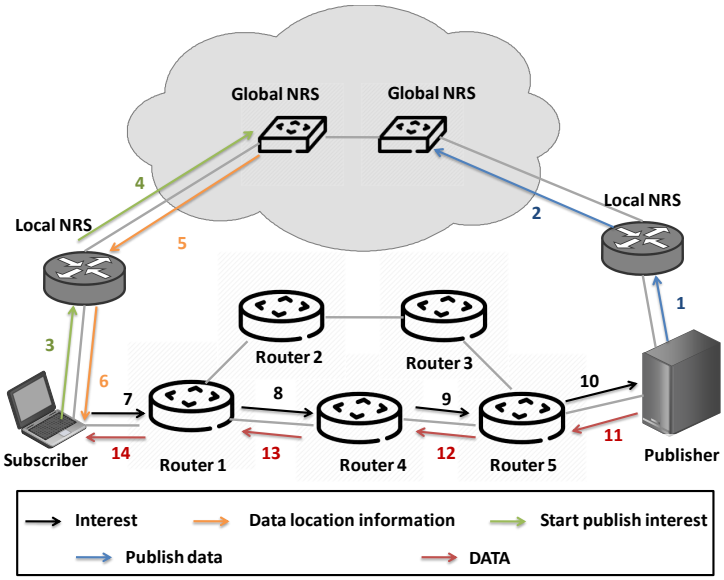

Figure 1.5 – *NetInf architecture*

1.3.5 Named Data Networking (NDN)

The NDN architecture was originally proposed through the NDN project [2] and by Van Jacobson in 2009 [6]. Based on the title Content-Centric Networking CCN, this project was subsequently divided into two other projects (fork): (i) the CCN project, managed by the Palo Alto Research Center (PARC) [3] and owned by the architecture; (ii) the NSF NDN project, managed by UCLA and whose source code is free (Open source). While the two projects share the same architecture, they have subsequently evolved separately according to the different contributors and objectives of the different institutes [4]. We will then primarily use the NDN name. However, we will be able to use the term CCN in an equivalent way.

NDN is a recent concept. However, She received a lot of attention from scientific community. A number of works have investigated the NDN behaviour in distinct environments, such as, Telephony [46], Lighting Control systems [47] or Vehicle-to-Vehicle communication [48] while a number of works has investigated NDN security [49] [43] or privacy [50] and has suggested some enhancement.

[2] https://named-data.net/

[3] http://www.parc.com/

[4] https://named-data.net/project/faq/How$_d$oes$_N$DN$_d$iffer$_f$rom$_C$ontent Centric$_N$etworking$_C$N

the Named-Data Networking architecture, which we will use later in our work, we will allocate the next chapter to present overview, basic principles, characteristics and concept in details. Before that we will compare these architectures by highlighting their commonalities and differences in the next section.

1.4 Comparison Between Information-Centric Networking Architectures

The ICN architectures referred to above have similarities and differences in the techniques used for implementing the key ICN functionalities. Table 1.2 summarize the techniques used by each approach for handling the naming, name resolution and routing, caching, mobility, name data object granularity and security.

The architectures described, have all the same goal: to allow the distribution of massive content across the Internet. They share concepts and objectives but they also vary in the processes that enable them to work. These are set out in Table 1.2 and discussed below:

- **naming:** The naming of data in ICN architectures can be classified into three categories:
  hierarchical naming, flat naming, and a combination of these last two (hybrid naming). With flat naming, each data is directly identified by a name or by the result of a hash function that returns a unique identifier of the content.
  Architectures like DONA or PURSUIT use this type of naming, which consists of calculating the name of the content using deterministic functions. These functions must ensure that each node of the network will get the same value after using the function on the same content, which makes the location of the content with its name independent of the flat structure. This type of naming does not allow to be aggregated and therefore poses the problem of scaling the names resolvers of these architectures because they must keep all the names. If they are derived from hash functions, these names do not convey any semantics for a user who can not memorize them.

  Hierarchical naming, in turn, is separated into several parts to locate the object in a name organization. For example, the "/video/football/algeria/mahrez/free-kick.avi" content name will define a football video of a Algeria national team player named Riyad Mahrez striking the ball allowing the team to win. This naming scheme is similar to that used by URLs. IP addressing is another example of hierarchical naming, in which each IP address is composed of two parts, the network identifier in which is the machine, as well as the identifier of the machine within this network.
  CCN / NDN uses a hierarchical naming scheme similar to URLs, whose semantics can be understood by users. A hierarchical structure of the names makes it possible to aggregate the names (as for example by category type /video or /sports) and thus to limit the number of names to be kept in the

tables of the architecture nodes.

Finally, an architecture like NetInf uses a combination of these two types of naming: flat naming within domains, and hierarchical between different domains. The problem of this type of naming is to move from an intra-domain flat naming, to an inter-domain hierarchical naming.

- **Name resolution:** Name resolution is a key feature in ICNs and is related to data naming. Since the names are independent of the location of the content, the resolution of these names will therefore aim to link a request to a content. This link will then allow requests to be routed to the node(s) having the content, as well as data transmission. Architectures using flat naming such as DONA, PURSUIT or Netinf (intra- domains) require an entity to resolve names. Because each data is uniquely named by a deterministic function, names cannot be aggregated and will require a network-wide resolution system.

  With architectures having hierarchical naming such as CCN/NDN or Netinf (inter-domains), names can be aggregated and it is not mandatory to use a global entity to resolve them, as is the case with CCN/NDN. Netinf still has local resolvers between domains.

- **Routing:** The routing of the data depends on the resolution of names, so it is also depended to the data naming method. When the name resolution is done by a central entity, during a flat naming, as is the case with PURSUIT or NetInf (intra-domain), this entity will calculate the route between the client and the server having the data. Thus, the path is set up for the request.

  CCN / NDN, thanks to hierarchical names, performs hop-to-hop forwarding and aims to propose a routing approach to the closest content (whether it is a cache or a server). The forwarding tables (FIBs) fill up according to the data that the nodes receive and retransmit. When there are no rules to transmit a request, the node sends the request on all its other interfaces and floods the network.

  DONA, although having a flat naming, uses a hop-to-hop transmission system based on hierarchical resolvers. The network root resolver knows all the data available as well as the next hop to access it. It acts a sort of global entity capable of resolving names, while being a node of the network. Thus, the routing of requests is done by going back to the root when the node does not know the next hop to access the content. Then, The content is then transmitted to the user using the IP routing on which DONA is based.

- **Caching:** The caching feature deployed in the core of the network is one of the new principles of ICN architectures. The storage capacities will have a definite impact on the user experience or the total efficiency of the network in delivering the content, whether they are deployed across the infrastructure

or in strategic locations.
Indeed, as closer the content to the user, as faster he will access it and the response time will be shorter. In addition, maintaining local requests will help avoid overloading the core of the network and generating congestion. Not only improving the network performance, but also the economic agreements and partnerships of network operators, will be impacted by the management of these cache memories. By storing the contents within their Autonomous Systems (AS), the operators will be able to reduce the traffic that will be transmitted to the neighboring AS, and thus reduce their operating cost.
Currently, ICN architectures allow nodes to store data locally. Two cases are then possible: (i) a copy of the content is retrieved on the path taken by the request to the source of the content origin, i.e. the server (On-path); (ii) the request is proactively sent to the nearest local copy; which copy is not necessarily on the path between the request sender and the original server (Off-path).
The PURSUIT architecture does not allow cached data to be retrieved because the request is routed through the rendezvous chain and only the response passes through the forwarder network. This would be possible if the forwarders announced the contents they keep cached from the topology manager and the rendezvous.
DONA and NetInf will be able to retrieve data only in a cache located on the path between the client and the server (On-path). It would be possible to recover the data outside of this path in a closer cache if these caches announced the contents they have to the resolvers.
CCN / NDN flooding the network with requests, will be able to recover the data in the cache of the node owning them closest to the client, whether it is on the path to the server or not (On-path and Off-path). Each of the ICN architectures therefore requires to be able to use an announcement system in order to redirect the requests to the nearest caches having the requested copy, thereby allowing a routing called *anycast*. NDN is the only solution allowing to search for data in a cache, but this is done in a hazardous way, flooding the network with requests.

	DONA	PURSUIT	NETinf	NDN
Naming	Flat	Flat	Flat/Hierarchical	Hierarchical
Name Resolution	Resolvers hierarchy	Rendezvous	Local / global resolvers	/
Routing	servers	servers	servers	Local routing based on a routing table
Request Routing	Name Resolution	Name Resolution	Hybrid	Named-based Routing
Content Routing	IP	IP	Reverse path or IP	Reverse path
Caching	Cache everywhere On-path	Impossible	Cache everywhere On-path	Cache everywhere On-path and Off-path
IP/Native	IP	Native	Native	Native
Security	Principal is hash of public key	Packet level authentication for individual packets. Names can be optionally self-certifying	Signature or content hash, PKI independent	Data packets include signatures
Name Data object granularity	objects	objects	objects	packets
Human-readable names	Not human-readable	Not human-readable	Not human-readable	Can be human-readable
Developer Friendly	Closed source	Closed source	Closed source	Open source

Table 1.2 – *Comparison Between Information-Centric Networking Architectures*

1.5 Conclusion

In this chapter, we presented the current Internet architecture, as well as the challenges that have made the current Internet needs a fundamental redesign.

Many new architectures for a future Internet have been designed to be adapted to the needs and new uses (content, but also mobility, security, etc.). These new architectures are based on a new paradigm of content-centric communication and place data at the heart of exchanges. They are also cache networks and network nodes can store data for retransmission. Thus, the data is recovered independently of their locations and this is a fundamental paradigm shift with the architecture of the Internet based on TCP/IP.

Among these content-oriented architectures, we presented the most important of them such as DONA, PURSUIT or Netinf, but it is above all the Named-Data Networking (NDN) architecture which we have particularly studied and used in the course of this work. Indeed, this architecture is the most sophisticated one, and benefits from a large community of researchers and industrialists.

The next chapter aims to present the NDN architecture, their usefulness their fundamental design and their functionalities.

2 Named Data Networking

Contents

In this chapter we present an overview of the NDN architecture. we discuss about: the basic principles that guide the design of NDN and the new architecture of NDN, naming, routing and security.

2.1 Introduction

Numerous solutions have been proposed to narrow the gap between the Internet design and its current usage. One such potential Future Internet Architecture (FIA), sponsored by NSF, is Name Data Networking (NDN) [51]. NDN explicitly addresses the data (content) itself instead of its physical location (i.e., host) in the network, therefore, transforming data into the “first-class" entity. In NDN, consumer directly requests the name of the *content* by issuing an *interest*. The network then handles the request by efficiently finding and retrieving back the closest copy of the relevant content. This decoupling of time and space among request resolution and content transfer enables NDN to provide storage, mobility and security as native features belonging to the network architecture [51].

2.2 NDN Overview

Today’s Internet has shifted from the connections between the hosts to the global distribution and retrieval of contents in huge amounts. This clear difference between the present Internet architecture and its current usage (practice) uncovers all the limits of Internet architecture. To this end, numerous research efforts [8] aims to move the Internet into the future and commence to design and deploy a new Internet architecture called Information Centric Networking (ICN) [8]. In particular, ICN presents the data as a “first class" entity by focusing on *what* is the content rather than *where* the content is located Fig.1.2. Among diverse ICN styles, Name Data Networking (NDN) [51] [6] and Content-Centric Networking (CCN) [11] projects have earned significant recognition in research groups of both academia and industry. The architectures of NDN and CCN are very similar, moreover, their variations do not influence the consideration and description of the thesis.

2.3 NDN Principle

Every one of the proposed ICN designs use different approaches, however all of them center around a similar concept, content delivery rather than host-centric.

Named Data Networking (NDN) [10] is the most promising and popular among ICN architectures, it is a research project in the Future Internet Architecture Program funded by the U.S. National Science Foundation (NSF). Its origin is a previous project, Content-Centric Networking (CCN) [43], which Van Jacobson started at Xerox PARC in 2006 [1] . Its fundamental concepts were outlined in a Google tech talk by Van Jacobson, long before that exactly in 2009 [43].

NDN keeps the same hourglass-shaped architecture, which inherited from the hourglass shape of the IP architecture, this later what makes the original Internet

[1] https://named-data.net/project/faq/

design elegant and powerful. but replaces the host-to-host data delivery model at the thin waist by a receiver-driven data retrieval model as seen in Figure 2.1. This fundamental change leads to the move of communication paradigm from location-centric (Where) to data-centric (What). Given the benefits of NDN architecture, for example, the simplicity, efficiency, accessibility (Open-source implementation CCN and NFD).

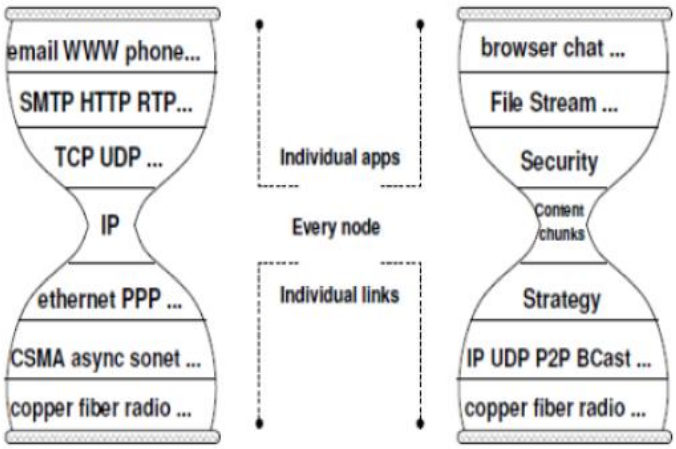

Figure 2.1 – *hourglass architecture [6]*

In order to achieve the objective of addressing the limitations and improving the strengths of the current Internet architecture, the new NDN design must be guided by the following architectural principles [52]:

- Security has to be integrated into the architecture. Security in the present Internet architecture does not meet the requirements of today's increasingly hostile environment. NDN offers the fundamental security building block right in the thin waist by signing all named data.
- The end-to-end principle allows solid apps to be developed against network failures. NDN maintains and extends this principle.
- Network traffic has to be self-regulating. Flow-balanced data delivery is crucial for stable network operation. Since the IP performs open loop data delivery, the transport protocols have been modified to provide unicast traffic balance. NDN designs the flow-balance into the thin waist.
- A separation between the routing and forwarding plane has demonstrated to be essential for the development of the Internet. It allows the forwarding plane to operate while the routing scheme continues to develop over time. NDN adheres to the same principle to allow the implementation of NDN with the best available forwarding technology while we carry out in parallel new routing system research.

2.4 NDN ARCHITECTURE

2.4.1 NDN types of packets

NDN is based on *requester-driven* communication model between two types of hosts; clients are called consumers, while servers are identified as producers. In contrast to IP where content is explicitly exchanged among nodes, NDN clients directly request content pieces from the network without addressing content producer's location. Specifically, when an NDN client wants to consume a specific *content*, it sends out a unique *interest* associated to that content. Each interest is identified by a Uniform Resource Identifier (URI) with a routable name scheme, e.g., `/example.com/video4u/examples.mp3` [51].

The packets types of NDN are indicated in Figure 2.2. An Interest packet consists of the following fields [2]:

- *Name* (required): It contains the prefix of the wanted content.
- *Nonce* (required): which is a randomly number generated by the consumer, which used to detect and avoid looping Interests. It is a random number that uniquely identifies the Interest packet. More specifically, two Interests having the same Name are considered duplicated (or looped) if they also have the same Nonce. Otherwise, they are two distinct Interests for the same content.
- *InterestLifetime:* specifies the time remaining in milliseconds before the Interest expires. The default value of InterestLifetime is 4 seconds.
- *Selector:* since NDN use longest-prefix-matching, an Interest can have several matching Data, this field provides additional descriptions to select the only one desired data to return. Selector contains several sub-fields, for example:
  - MinSuffixComponents, MaxSuffixComponents: referring respectively to the minimum and the maximum number of name components beyond those allowed in the data matching prefix. Default value for MinSuffixComponents is zero and for MaxSuffixComponents is infinite, meaning that any Data whose name starts with the requested prefix is a match;
  - PublisherPublicKeyLocator: which specifies the prefix of the key used to sign the corresponding data packet.
  - Exclude: can be used to choose a list of suffixes (i.e., following name components of the prefix) that the user wants to avoid.
  - ChildSelector: an Interest may match several Data with same prefix's length. The ChildSelector bit expresses a preference for the name component right after the prefix.
  - MustBeFresh: at the point when this bit is set, the user insists in receiving a "fresh" Data. If it is absent from an Interest, the corresponding Data does not have to be fresh.

[2] See http://named-data.net/doc/NDN-TLV/current/interest.html

Any node (producer or in-network) receiving the Interest and having the requested data can respond with a Data packet. The Data packet is used to carrier the requested data with some additional fields such [3]:

- Name: contains the full name of the data carried inside;
- MetaInfo: this optional field offers extra information about the Data, such as ContentType and FreshnessPeriod.
  - ContentType: reveals the type of information transmitted in the Data, e.g., public key, payload;
  - FreshnessPeriod: when a Data is stored in router, it is labeled as *fresh*. After its FreshnessPeriod has expired, the router will mark it as stale, making it less prioritized for answering Interest. When this field is absent or equals to Zero, the Data will always be fresh;
  - FinalBlockId: indicates the identifier of the final block in a sequence of fragments.
- Content: contains the requested data;
- Signature: includes the digital signature and supporting information used to verify the integrity of Data. These fields includes two sub-fields: *SignatureInfo* and *SignatureValue*

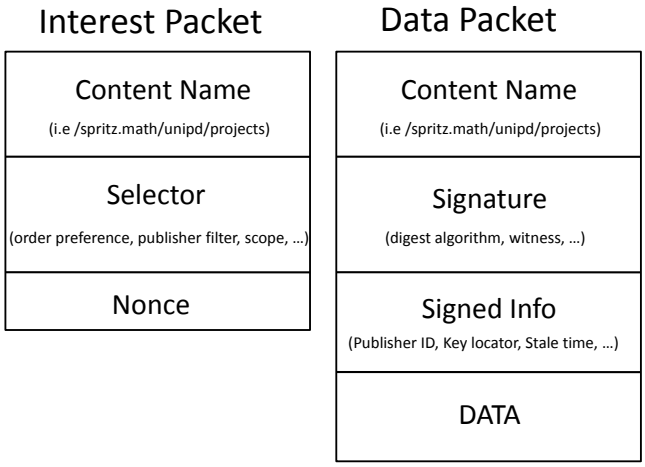

Figure 2.2 – *NDN packet types*

2.4.2 NDN hosts

Figure 2.3 shows the structure of an NDN node. An NDN node contains 3 structures (Content store, Pending Interest Table, Forwarding Information Base) and a number of faces to manage interest and data packets. While faces are quite the same of interfaces in TCP/IP, so their purpose is to connect a node to the rest of the network, the other 3 structures are completely new:

[3]http://named-data.net/doc/NDN-TLV/current/data.html

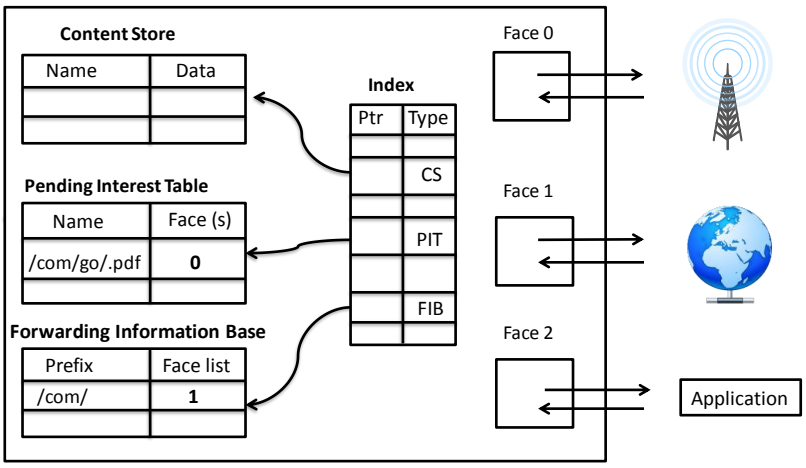

Figure 2.3 – *NDN node*

Content Store: is a buffer memory where it is saved every content received from a face. Its presence allows a node, when an interest arrives, to immediately reply to the interest if the related data packet is in the Content Store Fig.2.52.3.
Pending Interest Table: hereafter PIT, traces interests arrived from a face and forwarded up towards the source of the content until the interests are satisfied Fig.2.4.

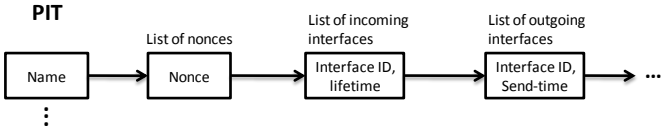

Figure 2.4 – *PIT*

Figure 2.5 – *PIT [7]*

Forwarding Information Base: is used to forward interest packets toward potential source of matching data. It is almost identical to an IP FIB excepts it allows for a list of outgoing faces rather than a single one Fig.2.6.

Even if faces have the same purpose of interface, there are some difference between them. A very important difference depends directly on the fundaments of NDN and allows NDN to take advantage on multiple faces. Because NDN bases its communication on content and it does not include host information on its packets, it does not need to obtain or bind layer 2 identity (MAC address) with layer 3 identity (IP address). This allows NDN to change connectivity faster than TCP/IP, in fact NDN can exchange data as soon as it is physically possible to do so.

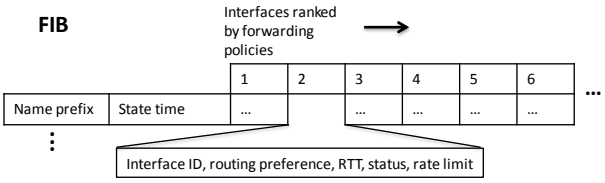

Figure 2.6 – *FIB [7]*

These 3 structures are the ground for the communication in NDN. As explained before, communication is based only on content and not on host; in fact, inside interest and data packets we can not find any trace that refers to host. So, how interest arrives to node that contains the related data packet? And how data packets are forwarded back to the consumer? To answer to these questions we analyze how Content Store, PIT and FIB are used in the communication process.

2.4.3 NDN Forwarding process

NDN stretches additional responsibilities to the routers by enabling router-side content caching and interest aggregation [11]. Each NDN router implements three foremost data structures, named as (i) Forwarding Information Base (FIB), (ii) Pending Interest Table (PIT) and Content Store (CS). Figure 2.7 illustrates the functionality and key components of NDN router. In particular, the FIB is used to perform a lookup operation to determine the interfaces in order to forward the incoming interests (e.g., it includes the interest prefix and the interface number). The second look up table is the PIT, which comprises of all the entries (e.g., interest prefix, arrival interface) of outstanding forwarded interests. Once an interest for a specific content is received, NDN router first checks whether the demanded content already exists in the CS. If the content is not available in the CS, the router looks in the PIT for a pending interest issued for the same content. If there is no entry, the router forwards the interest towards its destination and adds a new entry in the PIT with the associated arrival interface. On the contrary, if there is already a similar entry in the PIT, additional interests requesting the same content will not be forwarded. Later, when the requested content arrives, all the pending interests for it are satisfied just by sending a copy of the content back to all the consumers who issued an interest before. Finally, the requested data packet is forwarded to the consumer just by traversing the reverse trail of the preceding interest [10].

In NDN, not all the interests result in the content being returned. For instance, when an interest encounters either a router that cannot forward it further or a content producer that has no such content. Moreover, no error packets are generated in this case. The PIT entries for such unsatisfied interests in intervening routers are eliminated following a predefined expiration time. In result, a consumer can determine whether to reissue the same interest after a timeout.

2.4.4 Naming

Basing on data oriented communication rather than hosts implies that we need to identify content in some way. Names are used to distinguish one content from another. In NDN, the routers guide the Interests packets toward the potentials data

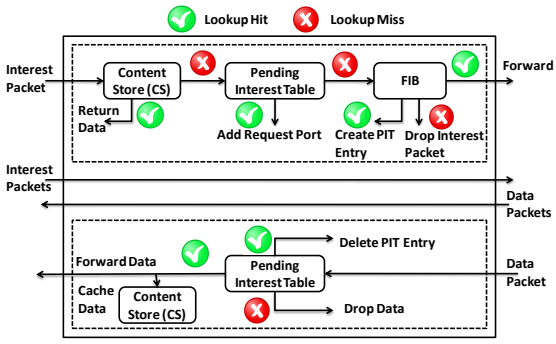

Figure 2.7 – *NDN forwarding process.*

source by exploiting data names. To define and allocate top-level names, a naming system is needed to ensure that the name of the content is unique between providers. However, not all NDN names need to be globally unique. In some local communication, NDN names can be based on local context [43]. Data satisfies an Interest if the data name in the Interest packet is a prefix of the data name in the Data packet [43]. NDN uses a multi-component hierarchical naming scheme. A component can be any string of arbitrary length, the delimiter '/' delineates name components in text representations, can be presented in the Uniform Resource Locator (URL) form. Fig2.8. For example a document shared by the University of Padova could have the following name /unipd/documents/documentA.pdf. This hierarchical structure, in addition to be human readable, allows routing to scale. In particular, as for IP where aggregation is essential in routing scale, this naming scheme encourages the utilization of aggregation (for example we can use /unipd/-documents to refer to all the document published by University of Padua). A user that does not know the entire name of a content can ask for a list of the contents in a namespace and afterwards ask for a specific one. A naming convention must be agreed by both the content providers and users so that they operate smoothly over NDN content names. Furthermore, NDN names are opaque to the network. In other words, the routers do not understand the meaning of a name but recognize boundaries between components in a name. The hierarchical structure of names has three advantages. First, the applications can define their own naming schemes that fits their needs and allow naming schemes to evolve independently from the network. Second, the applications can put the context and relationship of the data elements into the names. Finally, it allows name aggregation which is essential in scaling the routing system. In the example shown in Figure 2.3 go.com could be the autonomous system from which the video originates.

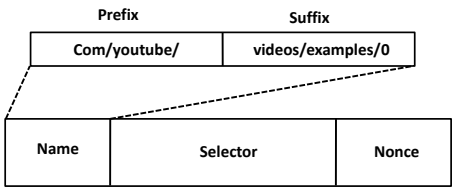

Figure 2.8 – *NDN names.*

2.4.5 Routing

Routing is the feature that will allow network nodes to fill their transmission table, or forwarding. The forwarding functionality consists of consulting, for each node, its forwarding table and then transmitting the message on the appropriate output interface, based on the information from that table. If no routing protocol exists or is deployed, nodes cannot fill their forwarding table and are then forced to flood the network with messages (flooding) by transmitting Interest messages on all interfaces in order to reach the content. Thus, as Interest messages flood into the network, nodes will fill their forwarding table with new entries.
However, such flooding will overload the network and consume its resources (bandwidth, etc.), preventing Internet-wide deployment. In the NDN architecture, messages address names and no longer hosts and this completely changes the problem of routing as known for the Internet. Each entry in a forwarding table will correspond to a content (and not a destination) and indicate the next jump to access it (i.e., the output interface). In addition, the cache features at the heart of the network allow for duplicating content and counting more copies in the network. Thus, a request to a content could be resolved via several nodes that could answer this request.
The routing protocols used in the current Internet based on location of hosts and point-to-point communications can no longer be satisfied to take full advantage of these new caching features. The simple flooding of the network is also not acceptable in order to be able to deploy this architecture one day. It is therefore appropriate to define new routing protocols adapted to the specificities of the Named-Data Networking content-oriented architecture [53] [54].

2.4.6 Caching

In NDN networks, Caching is one of the most important features, in contrast to the IP architecture Data packets are autonomous from the requesters and from the location which they are stored, since each Data packet carries the data name and signature instead than IP source and destination. As a result, a router can store the received Data packets in its Content Store (CS) and use them to fulfill the future requests.
Caching the content in CS is similar to buffer memory in IP routers, the thing that matters is that IP routers cannot reuse the data after forwarding them, whereas NDN routers are able to reuse the data because they are stored in routers CS and identified by persistent names. NDN achieves nearly optimal data delivery for static files. Even dynamic content can benefit from caching in the case of multicast, such as real time teleconferencing or packet loss retransmission [43].
In addition to the Content Store, the NDN architectures allow more permanent storage with very large or high-volume in-network storage, called the Repository.
Caching named data poses different privacy concerns from those of IP. In IP architecture, by analyzing the packet headers and probably the payload, it is possible to identify the requester and the requested data and their location. Nevertheless, in NDN, given that the data name does not have any information of the requester and the source of the requested data, it is more difficult to identify the data source and

the requester. In this manner, the NDN architecture provides some natural privacy protection by making data and users more secure [7].

2.4.7 Transport

The NDN architecture does not have a separate transport layer. The transport protocols functionalities like congestion reliable delivery, and are provided by applications, their supporting libraries, and the forwarding strategy. Segmenting and reassembling segments among application processes is done directly using names at the NDN layer, and data integrity and reliability are directly handled by application processes where the appropriate reliability checking, data signing and trust decisions can be made. Furthermore, the hierarchical namespace allows the required information for transport to be included in the data names, thus removing the need for transport layer information such as sequence and port numbers. A NDN network is designed to operate on top of unreliable packet delivery services, including the highly dynamic connectivity of mobile and ubiquitous computing. In order to provide reliable, resilient delivery, the application itself or its supporting library will monitor the unsatisfied PIT entries within some reasonable period of time and discard or retransmit them if the data is still required. In NDN architecture, the need for a flow balance, together with the ability of nodes to control their own traffic load by limiting the number of pending Interests at each hop can provide effective congestion control across the network. To reduce the congestion in the network, each Interest packet has a limited lifetime. When congestion losses arise, caching will reduce the impact since the retransmitted Interests can be fulfilled by the cached Data packets in routers that precede the packet loss stage. Caches also help data retransmission during congestion. If a router already caches a Data packet before it is dropped at a congestion link, the retransmitted Interest will be satisfied by this cached copy, reducing the latency throughout congestion. Flow control and congestion control are no longer dependent on end hosts. NDN forwarding process helps to reduce significant duplicated Data and Interests. Each Interest has a Nonce fields to detect and remove duplicate Interest. In addition, checking the PIT input before forwarding the Interest helps to reduce redundant Interests for the same content. Checking PIT before forwarding a Data, in the same manner, helps to eliminate unsolicited data traffic. Each FIB entry and PIT entry record several faces, thus support multipath and provide rich connectivity for nodes in its network [7].

2.4.8 Security

Experience in TCP / IP has shown that including a degree of safety in the thin-waist layer is a necessity and this provision has been planned for NDN. Security is one of the NDN stack's inventions. NDN is based on the concept of content-based security: protection and confidence travel with the content itself, rather than being the property of the paths which it travels through. Every data packet is authenticated by digital signatures, transferring protection from hosts to content itself. Currently in IP architecture, security are depends on where (from what host) and how (over what sort of pipe) data is obtained. To trust the data, user must be obtained directly

from the source. Accordingly with the CCN principle, NDN forgets about hosts and simply transfers trust to the contents themselves. i.e. the content is secured instead of channel and connection, so the producers Signing cryptographically every data packet, allows provider to securely join name to content. As result confidence is significantly improved. firstly an end-to-end signature check can be performed to allow a customer to verify the content's authenticity. In addition NDN data is publicly authenticatable, signature are standard public key signature and anyone, not just the endpoints, can verify that name-content binding was signed by a particular key. The signature algorithm, used to sign data packet, is selected by the producer from a large scale of fixed and chosen set to meet required performance in the signature/verification process. Every data packet moreover include all the information necessary to verify the signature and obtain the public key necessary. It may include cryptographic digest, or fingerprint of that public key; a shorthand identifier for the publisher; and a key locator to indicate where the key can be obtained, or containing the key itself. Such limited testing may be surprisingly useful to protect against many forms of network attacks, but it needs more refinement. Anyway, trust, privacy and security area has not been fully covered and this thesis attempts to help resolve a security issue in this context.

Conclusion

In this chapter, we have presented the NDN architecture in the context of CCN networks, which solves many limitations of the current internet architecture. But the future Internet architecture is not immune to some of the attacks that were in the current internet, Among these attacks DDoS.

In the next chapter we will show how a particular DDoS attack can be implemented in a NDN network and we will present how NDN is no resilient to this kind of attack.

3 Interest Flooding Attack (IFA) and Countermeasures

Contents

This chapter aims to introduce a new type of DoS attack, better known as Interest Flooding Attack (IFA). Named-Data Networking design, however, can be misused by this kind of attack. many solution have proposed in the literature. In this chapter we will also present some of them.

3.1 Introduction

NDN routers vary from their IP equivalents in the management of data structures used to store state. In particular, CS and PIT are special to NDN routers and, if there is no mechanism to protect them, they may be exploited by the adversary to carry out DoS / DDoS attacks. In this study, we concentrate on attacks using PIT, in particular, on how to build desires that saturate the PIT of the target router. When described above, the PIT contains information about pending interests. This information is used to guide content back to consumers. If the PIT is completely full, all incoming (un-aggregate) interests will be dropped. A router with a completely full PIT will not be able to accept new interests, until the existing ones are either fulfilled or expire. Flooding a router with many different interests allows the adversary to saturate the PIT. This is exactly the goal of Interest Flooding Attacks (IFA).

3.2 IFA Definition

Similar to IP packets, in the Interest Flooding Attack (IFA) the adversary aims at blocking network services by flooding them with a huge amount of requests for contents that are not available in the network. To achieve his aim, the adversary exploits two essential NDN features [55]: (i) the interest forwarding strategy based on the longest prefix match and (ii) the saving of the forwarded interests into the Pending Interest Table (PIT). The first feature enables the attacker to generate the malicious interests that will flood the routers. As a matter of fact, by exploiting the longest prefix match the attacker uses a prefix name, for which he knows there is already a forwarding rule in the routers FIB, and appends at its end a random value. This way, even though the content requested by the interest is unsatisfiable, the attacker will manage to make the interest transmitted in the network and the routers will save a corresponding entry in their PIT. This data structure is the fundamental component of the second NDN feature exploited by the attacker. Once an entry is added to a PIT, it remains until the expiration time, thus allowing the adversary to easily fill the whole capacity. Note that interest packets in NDN also consumes sensible share in network capacity. Thus, a massive amount of interest packets might lead to congestion in the network and results in legitimate traffic to be dropped. If a well-coordinated IFA targets a specific name prefix, it can concentrate malicious traffic in the certain segments of the network since the routing in NDN is grounded on the name prefix. As the routers in NDN maintains per-packet state for each interest packet in PIT. Therefore, the immense amount of malicious interests can result in exhaustion of routers memory and resources, and prevent them from creating PIT entries for new incoming traffic, resulting in the disrupt of benign users services.

3.3 Categorization IFA's in NDN

Through our study of the various IFA attacks proposed in the literature, We categorize the IFA attacks into three broad categories: (i) content requested IFA attacks This sort of interest flooding attacks called also denial of service against content source (DACS attack). it surges an extensive number of pernicious interests asking for content that does not exist, which ensures that no cache hit can occur at routers until these malicious interests reach the target content source. In this way, it can directly exhaust the resource of the victim. (ii) one other type of attack is totally different of the previous one, called Collusive Interest Flooding Attack. Where the interests can be satisfied by a malicious publisher. (iii) The last category includes a novel advanced attack model for IFA attack in NDN (see Fig 3.2).

3.3.1 Based on content requested

IFAs are categorized on three types based on the type of content requested by adversary [56]: (i) existing or static content, (ii) dynamically generated content, and (iii) non-existent content.

1. In this type, several zombies from multiple locations generate large number of interests for an existing content which propagates through all intervening routers caches. In result, the producer will start dropping interests. This causes interests to remain in the victims PIT until they expire. In particular, this type of attack is quiet restricted since in-network content caching provides a built-in countermeasure (depending on the capability to satisfy interests quickly).

2. In type 2, adversary issues dynamic-generated requests for existing content, therefore, all interests packets are propagated towards the producer(s), resulting in bandwidth consumption and PIT exhaustion. Correspondingly, targeted producer wastes considerable computational resources due to signing the content (i.e., per-packet operation) which is itself expensive

3. In type 3, adversary requests unique non-existent (unsatisfiable) content. These interests, that cannot be collapsed by routers, are routed towards the producer(s). Such interest packets consume memory in routers PIT until they expire due to interest life-time. Therefore, a massive number of non-existent interest packets in the PIT table leads to benign interest (BI) packets being dropped in the network Fig.3.1.

3.3.2 CIFA

As opposed to the previous studies concentrated on the Non-Collusive Interest Flooding Attack (NCIFA), in which adversaries flood the network with non-satisfiable interest packets. [57] study a formerly disregarded version of attack called CIFA. In CIFA Collusive Interest Flooding Attack , malicious clients issue

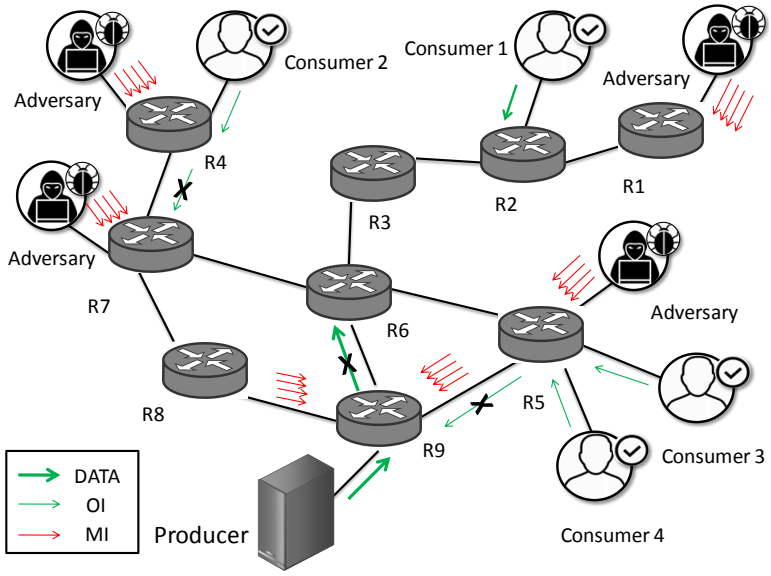

Figure 3.1 – *content requested IFA illustration.*

interest packets that can be satisfied only by a malicious server. The server, in turn, responds with data packets just before expiration of the corresponding PIT entries.

In CIFA, malicious clients and a malicious server (i.e. content provider) collude together to achieve interest flooding. More precisely, malicious clients issue a large number of unique interest packets requesting contents that can be satisfied only by the malicious server, resulting in one PIT entry per interest packet in each NDN router on the path. The malicious server successively answers with data packets just before the corresponding PIT entries expire. Once some PITs on the path are overloaded, legitimate interest packets are dropped. Due to this unique design, CIFA cannot be mitigated by the defence mechanisms previously proposed against NCIFA [55].

the Collusive Interest Flooding Attack (CIFA) requires malicious consumers as well as a malicious producers to connive together to accomplish interest flooding. More precisely, malicious interests issue numerous satisfiable interest packets named with unique content names beginning with the name-space of the malicious producer (e.g. "/malicious domain/"). This outcomes in creating a single PIT entry per each malicious interest packet in each crossed router (like normal IFA). The server, in turn, responds with data packets shortly before the corresponding PIT entries expire. In both IFA and CIFA, succeeding to overflow PITs of some routers leads to drop subsequent interest packets including those belonging to legitimate users.

3.3.3 Advanced IFA

The authors of this paper [58] propose a new advanced attack model for IFA attack in NDN, motivated by the fact that the countermeasures have been proposed to mitigate IFAs, made a simplistic assumptions in attack scenarios undermine their

potential. Like malicious and legitimate Interests belong to trivial disjoint prefix sets and/or use randomly-generated content identifiers to generate requests for non-existent contents. Other thing that all of countermeasures are reactive, i.e., the mitigation of the attack unavoidably begins after a period in which routers collect statistics about the processed traffic. This makes these countermeasures helpless to attackers which may adjust their Interest generation to impact gathered statistics or exploit routers observing time windows.

In general, attackers have the accompanying capacities:

- capacity to produce Interests which both correspond to existent content and do not correspond to any existent content.
- some knowledge about countermeasures put in place by routers.
- capacity to influence different targets at the same time by means of certain Interest types and different prefix names.
- capacity to affect monitoring statistics collected by routers.

IFAs may be targeting either content producers or the network infrastructure or both. So far it has appeared that they can achieve the following effects:

- the exhaustion of routers resources. Especially, they tend to completely possess the PIT space making further Interests be dropped.
- the overload of interests on target producers. In order to be more efficient, such interests should ask for existent contents since those require more computation from the producer, e.g., signatures generation.
- the saturation of network links due to a certain asymmetry in size between Interests and Data packets. In spite of this requires conniving parties performing the attack or requesting for many different existent contents, flooding with Interests upstream links may cause Data packets received downstream to immerse the bandwidth.

3.4 Solutions mitigating IFA in NDN

In this section, we discuss about the approaches and limitations of the state-of-the-art mitigation solutions against IFA. In particular, in order to classify the related work, we refer to the following three IFA mitigation categories based on their functionalities [59]:

- rate limiting-based countermeasures, in which the mitigation relies on throttling down the overall incoming traffic in an autonomous/collaborative manner;
- statistical modeling-based countermeasures, where the mitigation mechanism is based on the statistical information of PIT occupancy;

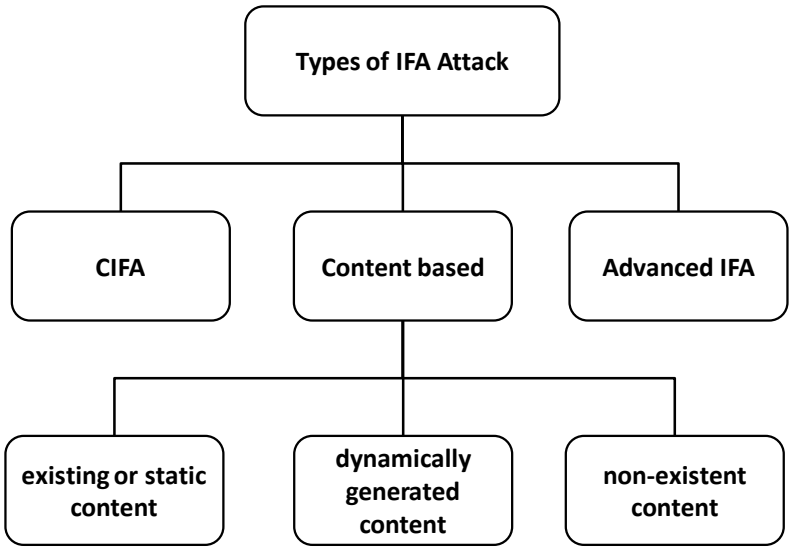

Figure 3.2 – *Types of IFA attacks.*

- the other countermeasures, that include the approaches based on the update of routers structures (i.e., forwarding information).

Below, we comprehensively describe the relevant work under each category (see Fig 3.3).

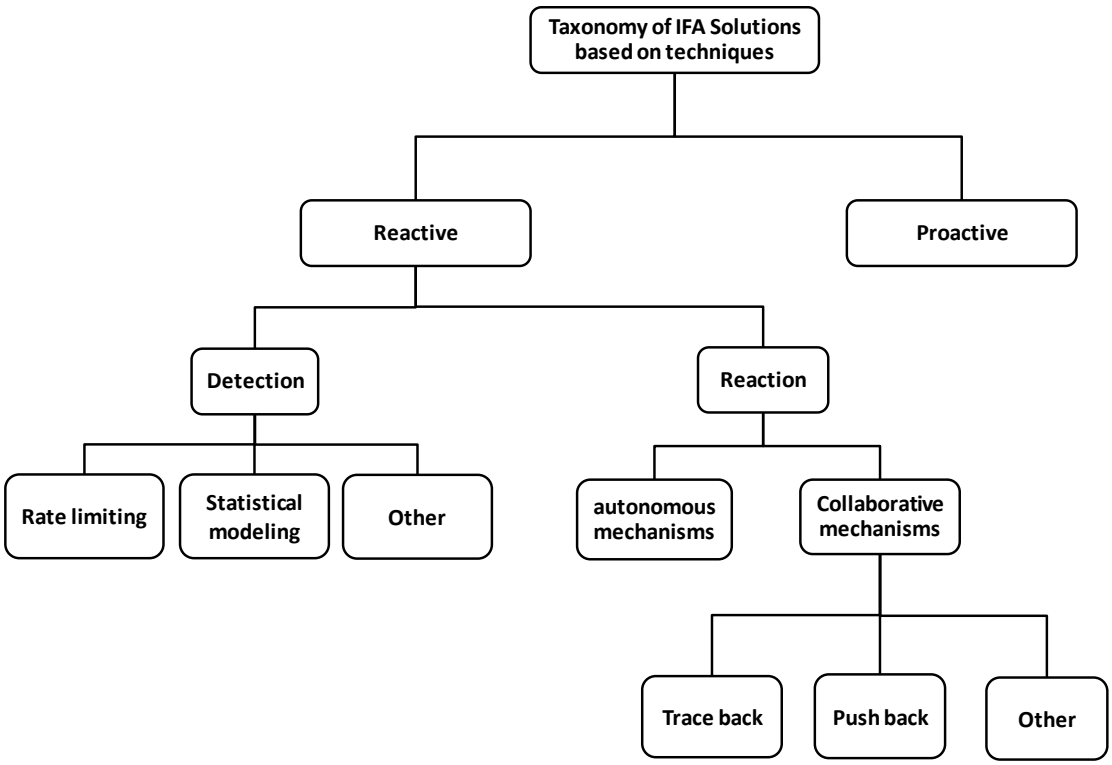

Figure 3.3 – *Types of IFA solutions.*

3.4.1 Proactive approaches

As the name suggests, anticipation of an attack is the main feature of proactive countermeasures. In this way, when malicious users start an attack, the attack will have no effect (or slight effect) on the network.

Wang et al.[18] proposed a mitigation mechanism called DPE (Disabling PIT Exhaustion) which diverts all the malicious interests out of the PIT. In particular, the state information of the malicious interests is recorded separately in the name of malicious-list instead of PIT. It introduces also a packet marking scheme to help with data packets forwarding without PIT. In DPE, the quantity of lapsed Interests under each prefix is observed whenever an Interest arrives, and every router keeps a malicious list (meant as m-list). Whenever the number of expired Interests under certain prefix surpasses its pre-defined threshold, this prefix is recorded in the m-list (each prefix recorded in m-list is considered as being assaulted by IFA). Each vindictive prefix can be reserved in m-list amid the Decay Time (DT), which is defined as the duration that each recorded malicious prefix can be kept in m-list. For every Interest experiencing a NDN routers DPE includes a name segment named as Interface-List, which is put as the last segment of the Interest name to record the incoming interfaces list. For each Interest packet whose prefix is recorded in the m-list, the router getting it would never again record it in PIT, yet check it by adding its incoming interface of this router into its Interface-List component. Every one of the checked Interest packets is called as m-Interest. When m-Interest goes through routers along the way to data producer, the incoming interface of each router is affixed to the Interface-list of the Interest one by one. whenever an Interest packet arrives at a NDN router, The router checks if this Interest is set apart by the Interface-List component or the prefix of this Interest can be found in m-list. If so, this Interest is sent dependent on FIB without recorded in PIT after its incoming interface of this router is appended into its Interface-List component (that is, if this Interest travels through several routers, the incoming interfaces of these routers are recorded into Interface-List component one by one consecutively). Although this method ensures PIT not being overloaded, malicious interests will still be forwarded, which may lead to the network congestion and poor bandwidth utilization. Other disadvantages include extra packet overhead and processing load.

The authors of DPE [18] propose another solution [60] aiming to balance the detecting accuracy and delay, they investigate the main cause of detecting delay and propose the mTBAD mechanism, which maintains an m-list to record malicious Interests entries by combining the DPE and the negative acknowledgments (NACK). A lightweight monitor is equipped to issue m-NACK packets to inform the attacked router and update its m-list. The mTBAD mechanism collaborates DPE with NACK to achieve ultra-low total detecting delay. However, the thing that matters is that the mTBAD mechanism keeps up m-list utilizing m-NACK packets while DPE maintains m-list by tallying Interest lapse rate. Furthermore, m-NACK is propelled by rate-limit and NACK. As opposed to these works, m-NACK has malicious entry field and could be applied to refresh m-list. When an Interest arrives at a router, first the router matches it with local cache, then matches it with m-list, third lookups PIT, and finally forward the Interest to other routers with the

help of FIB. Such a procedure is equivalent to what it does in DPE, however, in the mTBAD mechanism, a producer has a monitor which detects malicious Interests in time. Furthermore, when the monitor receives the Interest, it makes a decision about whether the Interest can be satisfied, if yes, it issues responding Data. Otherwise, the monitor regards the Interest as malicious one and sends m-NACK labeled with malicious entry filed to inform attacked routers.

This work [61] presents a hybrid algorithm for proactive detection of DoS attacks and adaptive mitigation reaction in NDN. In the detection phase, a combination of multiobjective evolutionary optimization algorithm and particle swarm optimization in the context of the Radial basis function RBF neural network has been applied in order to improve the accuracy of DoS attack prediction.

This paper [62] proposes Interest Cash, an application-based countermeasure against Interest Flooding for dynamic content. Interest Cash requires a content consumer to solve a puzzle before it sends an Interest. The content consumer should provide a solution to this puzzle as cash to get the signing service from the content provider. The key idea is that the content consumer has to do extra computation before the content provider serves it. The extra work is to solve a puzzle. The content consumer has to append the solution to this puzzle as cash to the content name in Interest. The content provider only signs the content for the content consumer who offers a valid cash. Since we increase the cost to send a valid Interest, we can prevent potential adversaries from sending too many Interests for dynamic content. At the same time, normal users are hardly affected because the extra computation is not a problem if they do not send Interests too fast. The drawback of Interest Cash is to generate puzzle for each Interest that may generate a great overhead.

3.4.2 Reactive approaches

We are now addressing the proposed reactive countermeasures. They are characterized by the Detection and Reaction phases. Detection may be based on Rate limiting, Statistical modeling or many Other techniques. The Reaction phase may be either Autonomous or Collaborative. In the first example, routers depend mainly on local measurements (e.g. PIT consumption, interest rate unsatisfied, sum of bandwidth used to forward content). In the second example, the neighboring routers are working together to decide if the attack is in progress and how to react.

3.4.2.1 Rate Limiting-Based Countermeasures

Afanasayev et al. [16] proposed four different methods to deal with IFA. The first method introduces a simple limit on the interfaces, based on the physical capacity of the links and resulting in an under-utilization of the network. The second method is an adaptation of the token bucket algorithm [63] providing per-interface fairness. In particular, the algorithm regulates the number of outgoing interests by limiting the assigned tokens to a specific outgoing interface. The major drawback of this method is that tokens are assigned without discriminating between BIs and MIs. Thus, not all MIs are dropped, while some BIs could. The third method is based on the per-interface ratio between the interests sent and the corresponding data packets received, which is also defined as the “satisfaction-based interest

acceptance". In this method, tokens are fairly distributed among all the incoming interfaces according to their interest satisfaction rates. The drawback of this method is that the router decision to forward or discard an interest packet relies on a router's local statistics, which is the router's interest satisfaction rate. Due to this reason, the probability of legitimate interests being forwarded declines as the number of hops/routers between the consumer and the producer increases [16]. The last method is a collaborative approach defined as "satisfaction-based pushback". In this case, each router sets an explicit limit value for each incoming interface, and announce this value to all downstream routers. This method has shown to be more effective than the previous ones, but the legitimate stream is still influenced, especially when the path is long. Moreover, it creates unnecessary signalling overhead in the network.

Similar to the works of Afanasyev et al. [16], other detection and reaction solutions have been proposed so far based on an independent or a collaborative approach. For the first case, the attack detection is based on network traffic analysis and/or PIT usage [17, 56], while the reaction consists in reducing the incoming/outgoing traffic, independently on each router. Vassilakis et al. [20] also proposed a similar mechanism that relies on the anomalous behavior of the consumers, to detect the attack, and on the reduction of the requesting rate of the traffic from the detected nodes, to block the attack. In addition to the traditional IFA detection criteria, Benmoussa et al. [64] considered also the network congestion to avoid the false positives generated by unintentional IFA phenomenon. Hence, to detect whether there is an intentional or an unintentional IFA, the authors use three parameters: the ISR (i.e., the number of interests issued over the number of data packets received); the incoming interest rate (i.e., the amount of interest packets arriving at a particular interface of an NDN router in a given amount of time); the network congestion, measured considering the number of timed-out interests and of NACK packets. Considering the collaborative component of the above-mentioned mechanisms, this consists in an information exchange among intermediate routers, which inevitably generates a signalling overhead. Moreover, some of those mechanisms [17, 19, 20] send a "push-back alert" to the downstream interfaces to reduce the data rate. In [19], the authors proposed a collaborative countermeasure known as "interest traceback", which detects an attack when the PIT size increases and produces artificial spoofed data packets for each interest stored in the PIT. Eventually, during the attack, the data packets trace the interests generators. The limitation of the approaches includes an excessive amount of additional traffic in the network, that results in the depletion of bandwidth and performance.

In summary, the main limitations of the rate limiting-based countermeasure are the following ones:

- lack of differentiation between benign and malicious traffic;
- non elimination of the interests already stored in the PIT;
- additional overhead due to the massive message exchange among the routers.

3.4.2.2 Statistical Modeling-Based Countermeasures

In [65], the authors face IFA through a detection scheme focused on Statistical Hypothesis Testing Theory, that relies on the Neyman-Pearson bi-criteria approach by providing a test that does not depend on router characteristics or measured values. Such framework considers two cases:

a scenario in which all traffic parameters are known (in this context, the optimal test is designed and its statistical performance is given);

a linear parametric model proposed to estimate unknown parameters and to design a practical test. However, the evaluation only uses a simple binary tree graph of eight clients and one adversary. It is difficult to analyze the efficacy of the scheme for larger networks or during distributed attacks. The authors in [66] also leverage the hypothesis testing theory to develop a generalized likelihood ratio test, adapted to evolve IFA attacks, especially, in the context of coupling NDN with IP, which can hardly be addressed by traditional solutions.

In [67], the authors proved the feasibility of IFA in a real NDN deployment. To this aim, the authors proposed a comprehensive set of 18 NFD metrics. For each metric, a micro detector is designed to capture any abnormal variation from the metrics normal behavior. The relevance of the micro detector design was evaluated through its performance against IFA in a testbed. To assess the generality of the approach, this should be tested with other attacks and larger topologies.

In [68], the authors proposed a Theil-Based Countermeasure (TC) to detect the distributions of BIs and MIs in the NDN routers and identify an IFA. Each NDN router records the name of each interest packet received. The router can use the statistical distribution of the names of the interest packets to detect the IFA. When an adversary launches an IFA, the occurrence frequency of the fake names in the MIs will increase significantly. The shift in the value of the Theil entropy can be used to determine the networks situation. In addition, the Theil entropy can divide the interest packets into groups based on a preset rule to evaluate the contribution of intra-group and inter-group differences. Then, under IFA, the TC detects the attack based on the change of intra-group heterogeneity. When an IFA is detected, a trace-back method on MIs can be used to find the adversary's position and avoid further attempts. Despite that solution induces additional signalling overhead, its evaluation uses only a simple binary tree graph with eight clients and two adversaries. The effectiveness of the scheme for larger networks or during distributed attacks is difficult to analyze.

In summary, this category also has some drawbacks:

- the attack detection is difficult close to attack sources, resulting in late detection and no reaction;
- the detection takes place after wasting a huge amount of resources in several regions of the network, and it likely cannot prevent the attack before causing a severe damage;

- mitigation mechanisms are also unable to remove malicious traffic from the PIT;
- the applied detection algorithms do not distinguish MIs from BIs. Hence, they may also harm legitimate traffic.

3.4.2.3 Other Countermeasures

This category of DoS mitigation includes approaches that change routers structures, such as PIT and CS. Wang et al.[18] proposed a mitigation mechanism called Disabling PIT Exhaustion (DPE) which diverts all the MIs out of the PIT. In particular, the state information of the MIs is recorded separately in the name of malicious-list instead of PIT. It introduces also a packet marking scheme to help with data packets forwarding without PIT. Although this method prevents the PIT from being overloaded, MIs will still be forwarded, which may lead to the network congestion and poor bandwidth utilization. Other disadvantages include extra packet overhead and processing load.
InterestFence [69] detects IFA based on content servers rather than routers to guarantee accurate detection. All content items with the same prefix within a content server have a Hash-based Security Label (HSL) to claim their existence, and a HSL verification method is securely transmitted to related routers to help filtering and cleaning IFA traffic in transit. InterestFence consists of three key functional entities: InterestFence-enabled router, InterestFence-enabled content server and the communication between them.
In [70], the authors proposed a Charging/Rewarding mechanism based on Hidden Markov Model (HMM) to defend against IFA. The approach is based on the external characteristic parameters of the consumers to establish HMM and then detect the malicious user. In particular, the HMM is established by the Baum-Welch algorithm. In order to limit the malicious users and stimulate the legitimate consumer, the edge router will charge consumers when providing forward service and reward legitimate users.
However, such mechanisms will still be responsible for forwarding MIs, resulting in network congestion and starvation of legitimate clients. In addition, the mechanisms also put additional processing burden on the routers and increase packet overhead.

3.4.3 autonomous mechanisms

In the case of a successful IFA, the victim's router can easily detect the attack by observing whether its PIT is full or if the bandwidth allocated to the forwarding of content is very limited. However, the router on the path to the victim may not be able to detect an attack in progress.

3.4.4 collaborative mechanisms

Collaborative mechanisms allow routers to exchange information about their status, with the aim of reacting to mitigate the attack as soon as possible once has been identified. Ideally, routers should withdraw all interests from the adversary, while at the same time forwarding the interests of honest users. When routers rely on collaborative detection mechanisms, they can not only share details about the presence of the attack, but also the (locally detected) properties of the attack. Many collaborative approaches have been proposed to counter IFA, like traceback, pushback and others.

3.5 Mitigating DDoS through Active Queue Management Schemes in IP

To mitigate DDoS attacks in IP, the congestion handling techniques, such as AQM, have gathered significant attention from the research community [71, 72, 73, 74]. AQM methods are classified into two categories according to their functionality and to the type of traffic they are able to handle [75]. The first category aims to provide fairness during network congestion, when the incoming traffic consists of only responsive flows (i.e., to which server responds). Random Early Detection (RED) [76], BLUE [77], and Adaptive Virtual Queue (AVQ) [78] approaches belong to the first category. On the other hand, the second category provides fairness when the incoming traffic consists of both responsive and unresponsive flows, such as CHOKe [22], Stochastic Fair Blue (SFB) [79], and Fair Random Early Detection (FRED) [80].

RED tries to acquire a better route queue stability by estimating the level of congestion in the router buffer and drops packets accordingly, by using an exponentially weighted moving average (EWMA) of the queue length. One of the RED limitations is that it is unable to identify unresponsive flows. Thus, it requires a significant parameters tuning to attain optimal results. To address this limitation, several methods, based on the idea of (or function with) RED, have been proposed: CHOKe [22], xCHOKe [81], RECHOKe [71] and FRED [80]. In particular, CHOKe is a stateless technique that tries to handle unresponsive flows by identifying and penalizing them through the drop of their packets.

In this thesis, we propose a solution that exploits an Active Queue Management (AQM) mechanism to defend against IFA in NDN. Our solution is the first one that removes the MIs from the PIT as a mitigation strategy; that works both as a detection and reaction approach; that avoids the drawbacks of the first and second categories of the state-of-the-art and that does not add new additional structures to routers, like the third category of the existing solutions.
In Table 3.1, we extend the review of the existing IFA solutions [82] by providing a detailed comparison of the existing countermeasures along with their functional details.

Table 3.1 – *Comparison of IFA countermeasures in NDN.*

	Type of solution		Detection			Detection mechanism	Mitigating mechanism	Stateless	Remove fake interest	Topology tested
	Reactive	Proactive	Per router	Per interface	Per prefix					
Poseidon [17]	✓	✗	✗	✓	✗	ISR + PIT usage	Push back	✗	✗	AT&T
Traceback [19]	✓	✗	✓	✗	✗	Thresholds	Spoofed data	✗	✗	Rocket fuel topology
Tocken bucket [16]	✓	✗	✓	✗	✗	Token	Trace back	✗	✗	ISP topology
Simple limits [16]	✓	✗	✓	✗	✗	PIT usage	Rate limit	✗	✗	ISP topology
Satisfaction based [16]	✓	✗	✗	✓	✗	ISR based	Rate limit	✗	✗	ISP topology
mTBAD [60]	✓	✗	✗	✓	✗	Thresholds	m-NACK	✗	✗	AT&T
Satisfaction pushback [16]	✓	✗	✗	✓	✗	Thresholds	Push back	✗	✗	AT&T
TDM [83]	✓	✗	✓	✗	✗	Thresholds	Rate limit	✗	✗	AT&T
Congestion-Aware IFA [64]	✓	✗	✗	✓	✗	ISR + Interest rate + congestion	Rate block	✗	✗	AT&T
Hypothesis Testing [65]	✓	✗	✗	✓	✗	Hypothesis Testing	✗	✗	✗	testbed
Reliable Detection [66]	✓	✗	✗	✓	✗	Hypothesis Testing	✗	✗	✗	testbed NDN with IP
Security Monitoring [67]	✓	✗	✗	✓	✗	18 metrics + monitoring plane	✗	✗	✗	testbed
Theil-Based [68]	✓	✗	✗	✓	✓	Theil entropy	Traceback	✗	✗	Small tree
InterestFence [69]	✓	✗	✗	✓	✓	hash-based security label (HSL) filter	✗	✗	✗	Tree
Charge/Reward [70]	✓	✗	✗	✓	✓	HMM	✗	✗	✗	Tree
Decoupling DPE [18]	✗	✓	✗	✗	✓	Malicious list	Decoupling	✗	✗	AT&T
ChoKIFA [23]	✓	✗	✗	✓	✓	AQM technic	Remove fake interest	✓	✓	AT&T
ChoKIFA+ AQM technic Edge router	✓	✗	✗	✓	✓	AQM technic + Edge router	Remove fake interest + rate limit	✓	✓	AT&T

Conclusion

In the next chapter, we take a footstep in the direction of identifying and differentiating MIs from BIs during IFA, in order to selectively stop the malicious traffic. In particular, we exploit an AQM algorithm [22] (i.e., CHOose to Kill malicious Interest, CHOose to keep genuine Interest for IFA - ChoKIFA) to stabilize routers PIT, by dropping the incoming MIs and removing the ones already stored in the PIT, without any global knowledge of the network (i.e., state information).

4 ChoKIFA: A New Detection and Mitigation Approach against Interest Flooding Attacks in NDN

Contents

This chapter is dedicated to the description of the new solution proposed in the context of attacks IFA in NDN networks.

4.1 Introduction

One of the key goals of NDN is “security by design", this thesis addresses the most significant NDN-tailored DDoS attack: the *Interest Flooding Attack* (IFA) [56]. In IFA, adversary aims at flooding the network and blocking the network services received by legitimate users via abusing two fundamental NDN features [21], i.e., (i) forwarding grounded on the longest name-prefix match, and (ii) maintaining the record of outstanding forwarded interests in so-called *Pending Interest Table (PIT)* for efficient multicasting. In particular, the adversary issues unique requests for unsatisfiable content name targeting the name-space(s). As a consequence, one PIT entry is created for each request in each on-path NDN router. These entries stays in the PITs till they expire at the end. Succeeding to overload some or all PITs which leads to legitimate interest packets being dropped [56].

4.2 Motivations

Regardless of the substantial quantity of research on NDN security, we identified that the proposed defence mechanisms for IFA [56, 16, 17, 19, 21] have one or more of the following limitations. First, the legitimate traffic is likely to be damaged, since most of the proposed countermeasures [17, 16, 20] limits the rate of incoming traffic and are not able to differentiate between legitimate and malicious packets, thus resulting in unfair punishments. Second, since each router has to perform first an attack detection and then attack mitigation, during the first phase (i.e., inaccurate), most of the approaches are likely to encounter harmful consequences. Finally, the proposed collaborative mechanisms [19, 17, 16, 21] introduces unnecessary overhead given by the extra messages exchanged among routers.

We propose an efficient mechanism, named as Choose To Kill IFA (ChoKIFA), which mitigates the damages caused by IFA by differentiating the malicious traffic from the legitimate one, and by reducing the former, without any collaborative communication or global network monitoring. In order to do so, ChoKIFA exploits the Active Queue Management (AQM) scheme, i.e., CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows (CHOKe) [22], and without any delay penalizes the malicious traffic by dropping both the new incoming malicious interests and removing the ones already stored in the PIT. Thus, routers are able to independently detect and mitigate the attack in progress as-soon and as-close to the adversary as possible, while maintaining the simplicity of forwarding.

4.3 Mitigation of IFA exploiting AQM

In this thesis, we take a footstep in the direction of identifying and differentiating malicious packets from the benign traffic during IFA. By exploiting the phenomena of AQM [22], we design an algorithm, i.e., CHOose to Kill malicious Interest, CHOose to keep genuine Interest for IFA (ChoKIFA) which aims to provide fairness among the benign interest packets that pass through the router. In particular,

ChoKIFA utilizes the PIT state which forms adequate statistics regarding the incoming and outgoing interest packets and use it to identify and drop malicious interest packets.

4.3.1 System and Adversary Model

In our system model, we consider the topology illustrated in Figure 4.1, as used by various authors [19, 16]. Multiple benign consumers (C) issues Benign Interests (BIs) for existing content towards a producer (P) which is publishing the content under specific name prefix ($prefix$). BIs and the corresponding content packets traverse multiple routers (R) before being satisfied by P. Each router $r_i^j \; \epsilon \; |R|$ has the default settings of NDN [10], where j is the interface of i-th router.

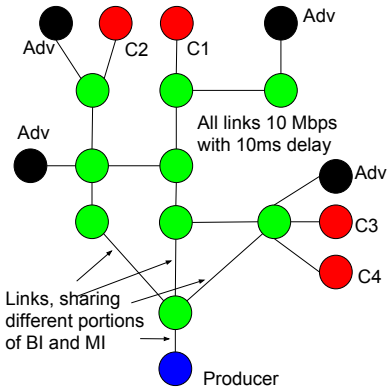

Figure 4.1 – *Topology considered.*

We assume that adversary (Adv) generates massive amount Malicious Interests (MIs) which have bogus names to request non-existing content (i.e., type three, see Section 3.3). The aim of Adv is to saturate R's PIT, in particular, by rapid generation of large numbers of MIs [16, 17]. Once the PIT is completely full, incoming BIs are being dropped. Apart from that, this has more than a few consequences. First, the sending rate of MIs is not dependent on the allocated bandwidth [56]. Secondly, MIs cannot be replied back by the $R's$ caches. Lastly, if created sophisticatedly (i.e., with a random component at the end of each name-prefix such as $prefix/Rnd$) MIs are never collapsed until the interests decay. All these effects allows Adv to efficiently fill up $R's$ PIT, which makes the attack more damaging than type one and type two IFA. In addition, without the loss of generality, we assume that Adv is capable to corrupt set of C (i.e., botnet), through which it triggers the attack [16, 17]. Lastly, the percentage of bots is taken 50% with the ratio of C in the whole network [16].

4.3.2 ChoKIFA: CHOose to Kill Interest Flooding Attack

In this section, we present the details our proposed mitigation mechanism for IFA. In order to be effective in defending against IFA, ChoKIFA exploits traffic flow as an attribute to differentiate and penalize the MIs from BIs. Unlike IP, where traffic flow measurement relates to the accountable attributes such as source/destination address, interface number, packets/bytes counts forwarded (source to destination), backward (destination to source) counts and so on. In NDN, following content

oriented communication model, traffic flow is centered around series of packets that corresponds to specific piece of data [84]. Considering this, we design the three novel attributes to compare incoming traffic flow at each router: (i) name-prefix match, (ii) interface match, and (iii) level of interest satisfaction ratio, i.e., rate between incoming interests to outgoing content, denoted as $\delta(r_i^j)$. In particular, $\delta(r_i^j) > 1$ denotes that the number of content packets received at router r_i^j is less than the number of interests forwarded from the same interface.

In order to mitigate IFA, ChoKIFA dynamically computes the actual size of the PIT, denoted as $\rho_{size}(r_i^j)$, at each instance. Further, ChoKIFA marks two thresholds on the PIT size, a minimum threshold ($\rho_{th}^{min}(r_i^j)$) and a maximum threshold ($\rho_{th}^{max}(r_i^j)$), as well as, a threshold for interest satisfaction ratio, denoted as $\delta_{th}(r_i^j)$. For each interest arriving at r_i^j, if the actual PIT size is less than the $\rho_{th}^{min}(r_i^j)$, the interest gets stored in the router's PIT. If all the interests requested by C are satisfied by P or router's cache, then PIT size should not reach up to $\rho_{th}^{min}(r_i^j)$, frequently. In case of IFA, when the actual PIT size is greater than $\rho_{th}^{min}(r_i^j)$ and less than $\rho_{th}^{max}(r_i^j)$, each new incoming interest is compared with the randomly selected interest from PIT, named as *drop interest candidate*. If both the interests have the same traffic flow then both are dropped. This choice is motivated by the fact that all the entries in PIT are likely to be occupied by MIs (i.e., under IFA). On the other side, when the PIT size goes more than $\rho_{th}^{max}(r_i^j)$, all the new incoming interest are being dropped. This leads the PIT occupancy back to below $\rho_{th}^{max}(r_i^j)$.

The key attributes to identify the traffic flow of each new incoming interest are three subsequent conditions: (i) if it holds the same prefix as of drop interest candidate, (ii) if it is coming from the same incoming interface as of *drop interest candidate*, and (iii) if both the above conditions holds true, then router compares if the current $\delta(r_i^j)$ exceeds $\delta_{th}(r_i^j)$. In contrast, if the new incoming interest is not having the same traffic flow as of drop interest candidate then the randomly selected interest is remained stored in PIT, and the incoming interest is dropped/accepted with the probability (P_b) which depends on the average PIT size ($\rho_{avg}(r_i^j)$), as illustrated in Equation 5.1 [76].

$$P_b = \frac{P_{max} * (\rho_{avg}(r_i^j) - \rho_{th}^{max}(r_i^j))}{(\rho_{th}^{max}(r_i^j) - \rho_{th}^{min}(r_i^j))}, \tag{4.1}$$

here P_{max} denotes the maximum probability[1]. As the average PIT size varies from $\rho_{th}^{min}(r_i^j)$ to $\rho_{th}^{max}(r_i^j)$, the interest dropping probability P_b varies from 0 to P_{max}. In particular, the interest dropping probability is computed by exploiting the mechanism of packet dropping probability of Random Early Detection (RED) [76]. A detailed flow chart of ChoKIFA is given in Figure 5.1.

[1]We take the value of maximum probability (P_{max}) to be one.

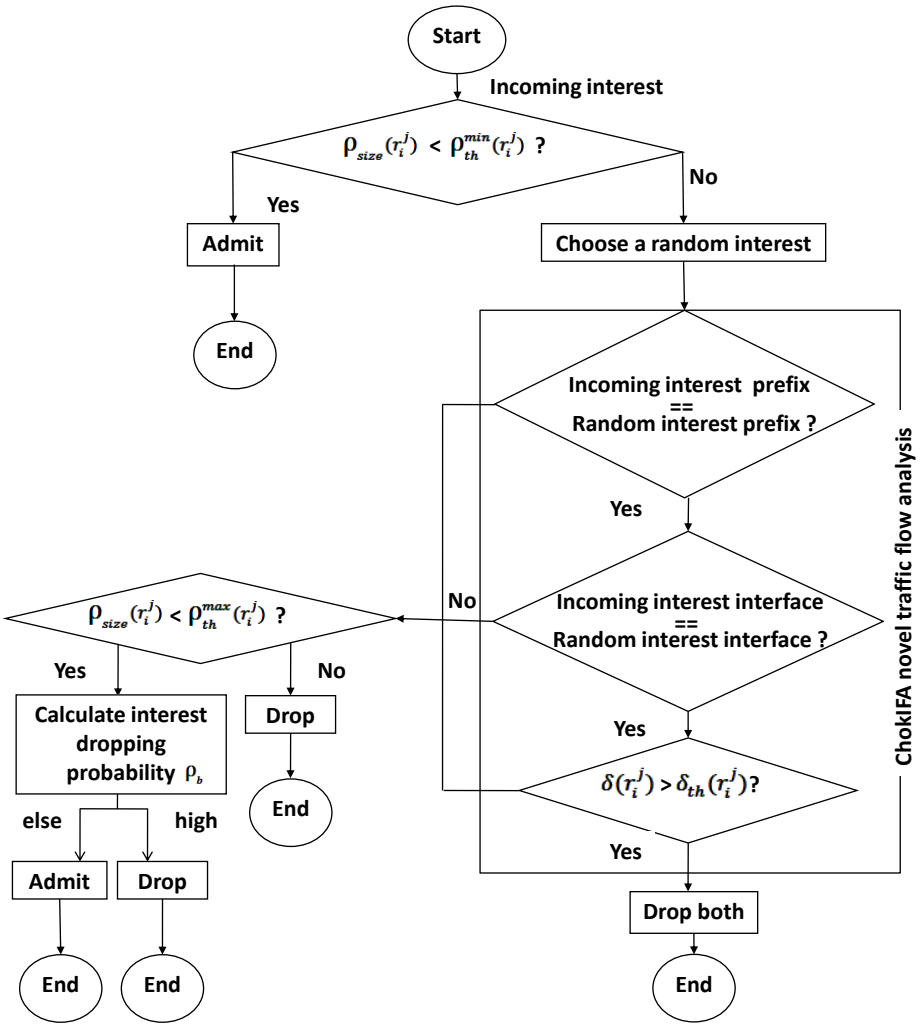

Figure 4.2 – *ChoKIFA algorithm flowchart.*

4.3.3 Parameters setting

The parameters, $\rho_{avg}(r_i^j)$, $\rho_{th}^{min}(r_i^j)$ and $\rho_{th}^{max}(r_i^j)$ are essential as they directly impact on the interest dropping probability. Below we illustrate few rules for parameter's setting which give effective performance for ChoKIFA under variety of traffic conditions while mitigating the attack.

Ensure adequate calculation of the average PIT size: ChoKIFA calculates the average PIT size using an exponential weighted moving average (EWMA). The use of EWMA for calculating $\rho_{avg}(r_i^j)$ makes sure that the short term increase in PIT size which may result from a burst of benign incoming interests (e.g., which are not satisfied due to network congestion/delay from the producer) do not result in the significant increase of average PIT size. Equation 5.2 illustrates the calculation of the $\rho_{avg}(r_i^j)$ where w_ρ is the weight factor for calculating EWMA and $\rho_{size}(r_i^j)$ is the current/actual PIT size [76].

$$\rho_{avg}(r_i^j) = (1 - w_\rho) * \rho_{avg}(r_i^j) + w_\rho * \rho_{size}(r_i^j). \tag{4.2}$$

Note that the calculation of average PIT size can be made particularly efficient

when w_ρ is set as a negative power of two[2]. If w_ρ is too large, then the averaging procedure will not filter out the temporary congestion of PIT.

Setting a minimum threshold for the PIT size: The optimal value of $\rho_{th}^{min}(r_i^j)$ depends on the desired level of $\rho_{avg}(r_i^j)$ and default network conditions. In case, the typical traffic is fairly bursty and congested, then the $\rho_{th}^{min}(r_i^j)$ should be correspondingly large to allow PIT utilization to be maintained at an acceptably high level.

Setting $\rho_{th}^{max}(r_i^j) - \rho_{th}^{min}(r_i^j)$ sufficiently large to avoid global synchronization: The optimal value of $\rho_{th}^{max}(r_i^j)$ depends in the part of maximum average delay that can be allowed to interest (e.g., round trip time for interest to retrieve data) and total size of PIT. A useful rule of thumb ChoKIFA implements is to set $\rho_{th}^{max}(r_i^j)$ more than thrice of $\rho_{th}^{min}(r_i^j)$ [76], since the mitigation mechanism works efficiently when max-min is larger than the typical increase in average PIT size.

4.4 Evaluation

We evaluate the effectiveness of our proposed approach in the presence of IFA and state of the art mitigation approaches which implements interest rate limiting based on the simple limit, interface fairness using token bucket, satisfaction ratio and with limit announcement technique [16]. To this end, we perform extensive simulations using the open-source ndnSIM [85] simulator. We evaluate the impact of IFA against ChoKIFA over three metrics which have been widely used in the related work [17, 16, 21, 19]. First, the PIT usage which indicates the available capacity of the routers to process benign traffic. Second, the percentage of BIs and MIs dropped by the network during IFA and with the proposed countermeasure. Third, we compare the efficiency our proposed countermeasure with existing mitigation approaches in terms of Interest Satisfaction Ratio (ISR) of benign users and legitimate traffic which is intended to measure the benign traffic received by users. Precisely, the lower the ISR refer, the greater amount of false positives made by the mitigation approach while distinguishing between the MIs and BIs.

Test Setup: We ran our simulations (with 100 seconds of simulation time for each experiment) on two different network topologies: a tree topology [19] (see Figure 4.1) and a more realistic large-scale ISP-like topology, i.e., AS-7018 [86] (see Figure 4.3). The selection of tree topology is because it represents one of the worst case to defend IFA [16], while the larger ISP topology reflects the performance of mitigation approach when deployed on the real Internet. The topology consist of a single P and number of consumers, including four honest clients (C) and four adversaries (Adv) connected with multiple ICN routers. Adv requests for non-existing content (i.e., MI), which exhibits distinct suffix ($/good/rnd$) compared to valid content ($/good/data$) with frequency of 1000 interests/second. C requests the interests (BI) for valid content which are entitled to P at a rate of 30 interests/second. The total PIT size of R, i.e., 600 kilobytes, thus we set the $\rho_{th}^{max}(r_i^j)$ and $\rho_{th}^{min}(r_i^j)$ equal to 3/4 and 1/8 of total PIT capacity (i.e., 450 and 75), respectively.

[2]In our simulations, we take w_ρ equal to 0.001.

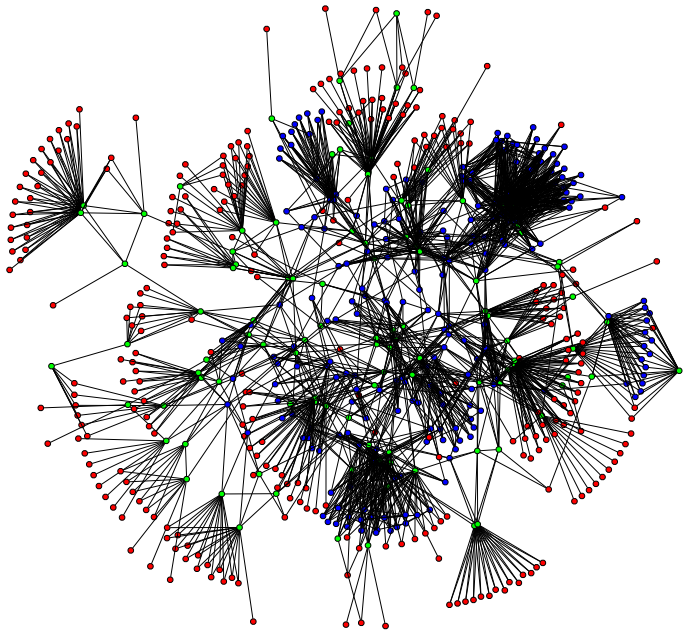

Figure 4.3 – *Internet-like topology: 296 clients (red), 108 gateways (green), 221 backbone (blue).*

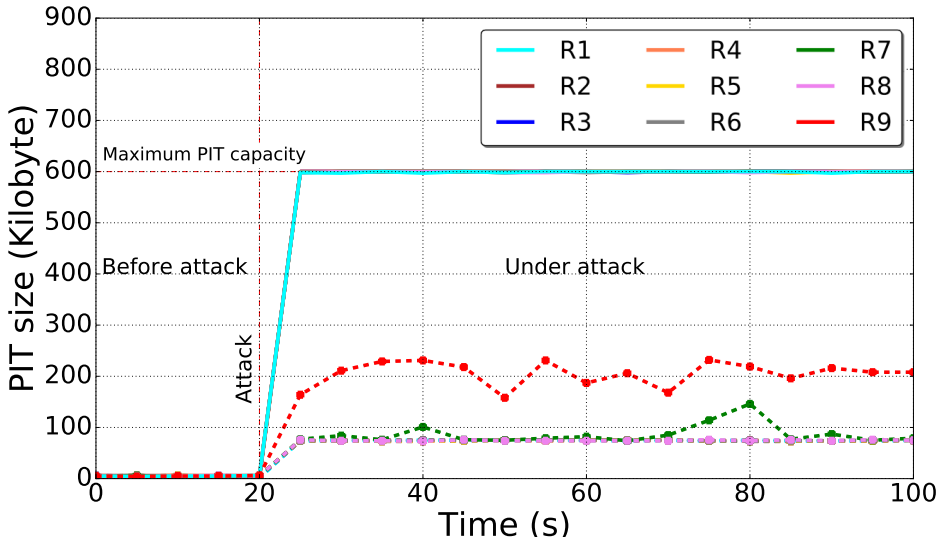

Figure 4.4 – *PIT usage, base-line (solid lines) and ChoKIFA (dotted lines).*

4.4.1 Small-scale simulation

In this section, we present the results of tree topology to evaluate the impact of attack and effectiveness of ChoKIFA. Figure 4.4 reports PIT usage of all the routers as a function of the simulation time under IFA for the base-line scenario (i.e., with no countermeasure) and, when the proposed countermeasure is active. In our simulations to evaluate and compare ChoKIFA under IFA, adversaries launches the attack at different time, i.e., starting from the 20th second, while the benign users starts to request for existing content from the beginning (see Figure 4.4). Because of the design of CHoKIFA, approach allows the IFA to fill the PIT of all the routers till 75 kilobyte before being able to start traffic flow comparison, i.e., minimum

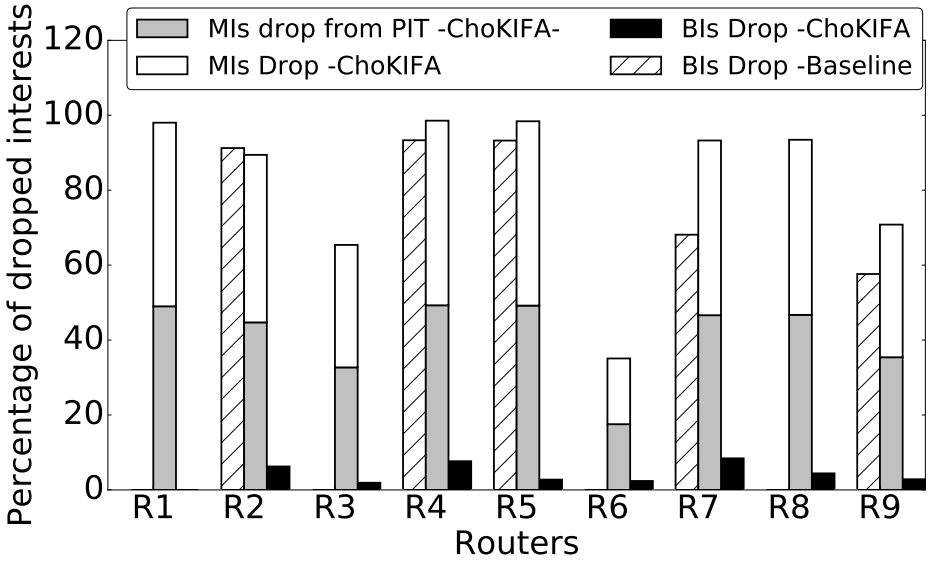

Figure 4.5 – *BI and MI drop, base-line and ChoKIFA.*

threshold of PIT. In contrast, after exceeding the minimum threshold, ChoKIFA's traffic flow comparison and interest dropping probability does not allow PITs to exceed certain level (i.e., slightly higher than 75) which depends on the dropping probability related to average PIT size. Results show (see Figure 4.4) that gateway node to producer attains slightly higher PIT size than the rest of routers since it receives aggregated amount of malicious traffic from the whole network.

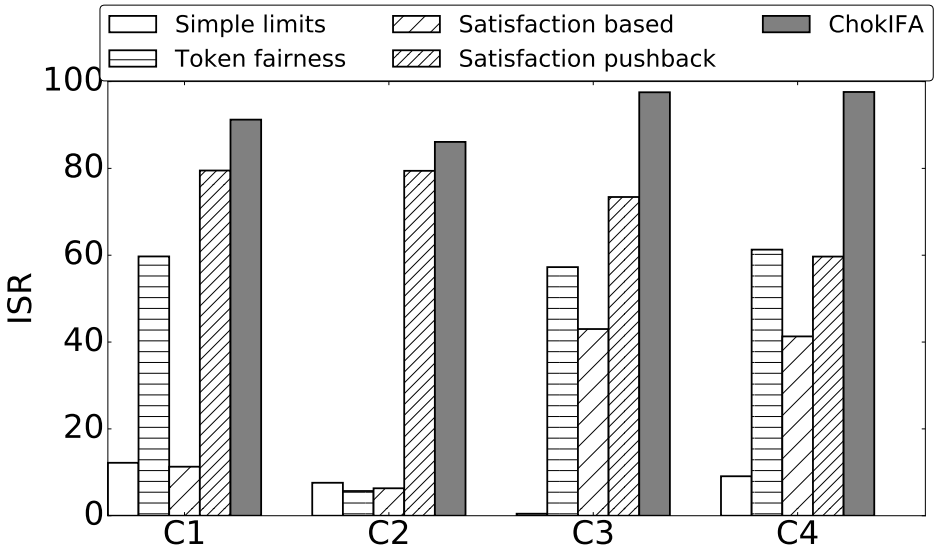

Figure 4.6 – *Benign consumers ISR in small topology.*

Figure 4.5 reports effectiveness of ChoKIFA under IFA, in terms of legitimate (BI) and malicious traffic (MI) drop. It shows the percentage of total BIs and MIs dropped over total received at each router, respectively. In particular, the legitimate traffic is slightly affected (only 4% of BIs are dropped on an average) with the use of ChoKIFA, while in base-line 90% of BIs are dropped. Because the PIT is filled up with MIs, therefore, the drawn random interest from PIT is also MI with the very

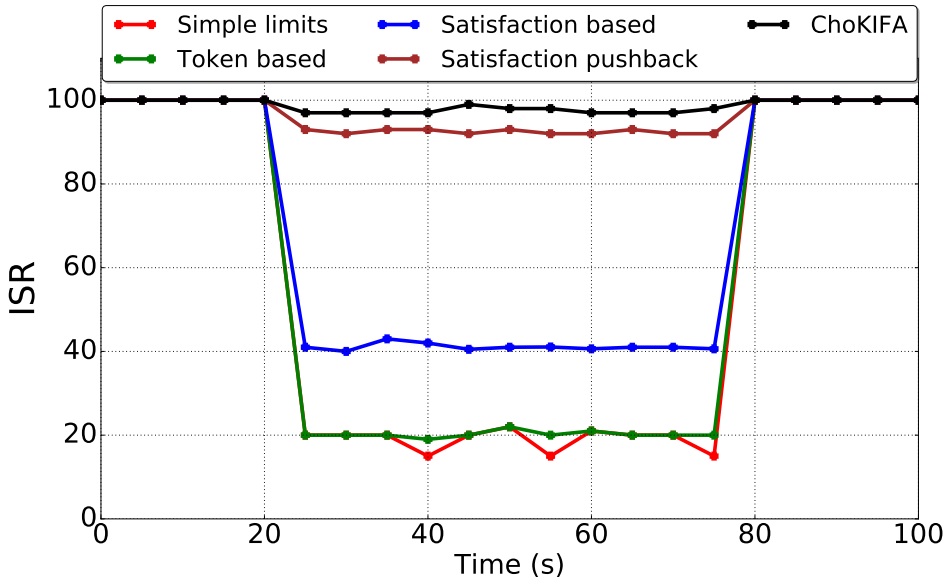

Figure 4.7 – *Global legitimate ISR in AS-7018.*

high probability, and in consequence ChoKIFA drops only MIs, i.e., both incoming and already stored in the PIT (see Figure 4.5).

Figure 4.6 reports the ISR of benign users which can be achieved when enabling ChoKIFA. We also compare these results with four different mitigation approaches [16]. The first three approaches are lightweight and stateless nevertheless not effective in legitimate ISR. Results show (see Figure 4.6) that Satisfaction-based pushback is slightly effective than previous methods but it also induces unnecessary signaling overhead by sending rate limiting announcements continuously in the whole network [16]. In particular, Figure 4.6 reports that ChoKIFA outperforms all four approaches in terms of all benign users ISR, remarkably. In particular, ChoKIFA is able to main 97% of all benign users ISR, moreover, induces 20 to 60% less false positives comparing to all four approaches while mitigating the attack.

4.4.2 Large-scale simulation

In this section, we evaluate the performance of ChoKIFA by implementing a real ISP-like topology (AS 7018) which is measured by the Rocket fuel project [86] (see Figure 4.3). To study the performance of ChoKIFA in ISP-like topology and under a range of conditions, we varied the percentage of adversary in the network and the frequency with which adversary is sending malicious interests.

There are 625 nodes involved in the topology, which are separated into three categories: clients, gateways, and backbones. The 296 nodes (red) are classified as clients for having degree less than four, while 108 nodes that directly connects to clients are classified as gateways (green). And the remaining 221 nodes (blue) are classified as backbones. The larger ISP topology reflects how our mitigation methods would perform when deployed on the real Internet. To study the performance of our proposed mitigation mechanism under a range of conditions, we varied the percentage of adversary in the network and the frequency with which adversary is sending malicious interests.

Figure 4.7 confirms that rate limiting approaches [16] are not able to maintain acceptable ISR for benign users in bigger topology as well. In particular, the result shows the percentage of global ISR of all legitimate interests generated in the network, where ChoKIFA maintains almost 97% of ISR during the attack. Note that the attack duration, in this case, is from 20 to 80 seconds. Figure 4.8 shows the ISR percentage of legitimate interests when we varied the percentage of attackers in the network, precisely, the values ranged from 6% attackers to over 50% attackers in the network. The results are as expected â- for ChoKIFA and all four state of the art mitigation algorithms. As the number of attackers in the network increases, the lower is the ISR ratio for legitimate interests. For instance, in the case of the token bucket with per interface fairness, only 3 attackers can halve the quality of service for the remaining 13 legitimate users. While the two intelligent attack mitigation algorithms also show a decline in legitimate service quality as the percentage of attackers increases. Although ChoKIFA outperforms all mitigation algorithms and shows a very minor reduction in ISR ratio (i.e., approximately 3%) even when the attacker's percentage is raised more than 50%.

Figure 4.9 shows the aggregated legitimate ISR ratio when we increased malicious interest sending rate from 100 interests/second to 10000 interests/second. The result shows that ChoKIFA remains almost unaffected even with huge amount of increase in malicious interest frequency, while among all state of the art approaches only Satisfaction-based pushback shows satisfactory results.

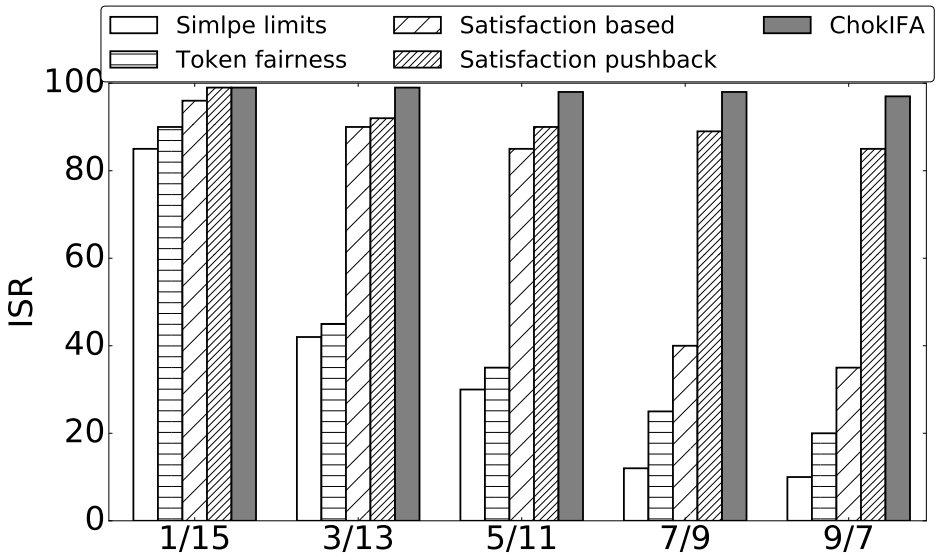

Figure 4.8 – *Legitimate ISR with increasing adversary.*

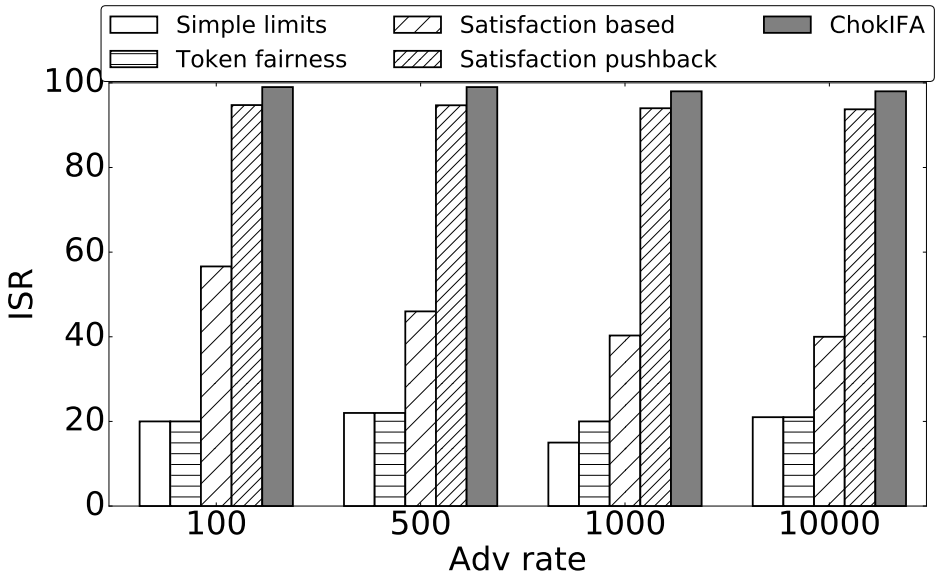

Figure 4.9 – *Legitimate ISR with increasing malicious traffic.*

Conclusion

In this chapter, we address the interest flooding-based DDoS over NDN, which is explicitly named as IFA. More specifically, we have found that several proposed countermeasures, that adopt detection and reaction mechanisms based on interest rate limiting, are not highly effective and also damage the legitimate traffic.

In our solution, we exploited an active queue management scheme to propose an efficient detection and mitigation mechanism against IFA, which stabilizes the router PIT. The proposed approach penalizes the adversarial traffic by dropping malicious interests generated during the IFA.

To better highlight the contributions of our solution, a simulation-based analysis under the NDNsim simulator has made. We implemented the proposed protocol on the open-source ndnSIM simulator and compared it with the state-of-the-art. The results report that our proposed protocol effectively mitigates the adverse effects of IFA and shows significantly less false positives in comparison to the state-of-the-art IFA mitigation approaches.

5 ChoKIFA+: An Early Detection and Mitigation Approach against Interest Flooding Attacks in NDN

Contents

5.1 Introduction

In this chapter we will present an improvement of the previous ChoKIFA solution, the extended work is called ChoKIFA+, the simulation results that we obtained using the NDNSim simulator through defined simulation scenarios show better performance.

5.1.1 Motivation and Contribution

IFA has been extensively addressed by previous works, either in terms of detection or in terms of mitigation [16, 17, 18, 19, 20]. However, all the existing solutions are affected by one or more among the following limitations:

- A variable effectiveness of the detection approach according to the location: the attack detection is not robust and accurate, if applied close to content providers and victims, and if the volume of adversarial traffic is large [21];
- The amount of damage applied on the legitimate traffic: the legitimate traffic is likely to be damaged, since most of the proposed countermeasures [17, 16, 20] limit the rate of incoming traffic and is not able to differentiate between Benign Interests (BIs) and Malicious Interests (MIs), thus resulting in unfair punishments;
- The overhead introduced on the routers that both detect and mitigate the attack: during the attack detection, most of the approaches used by routers are likely to encounter harmful consequences;
- The network overhead caused by collaborative approaches: the proposed collaborative mechanisms [19, 17, 16, 21] introduce unnecessary overhead due to the extra messages exchanged among routers.

. To improve the detection and mitigation against IFA, by overcoming the above-mentioned limitations of the state-of-art solutions, we propose an efficient countermeasure, named as Choose To Kill IFA (ChoKIFA), which mitigates the damages caused by IFA by differentiating the malicious traffic from the legitimate one, and by reducing the former. To distinguish between BIs and MIs, the proposed protocol applies on each router an active queue management scheme, i.e., CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows (CHOKe) [22]. For each incoming interest, ChoKIFA evaluates several conditions before saving it in the router PIT by differentiating and penalizing only the malicious ones, thus preventing the propagation of the attack in the network. The preliminary results concerning the effectiveness of ChoKIFA were reported in [23]. In this chapter, we propose an enhanced version of ChoKIFA, named as *ChoKIFA+: An Early Detection and Mitigation Approach against Interest Flooding Attacks in NDN*. In ChoKIFA+, the edge routers, which consumers are directly connected to, provide an additional security wall to detect the attack as early as possible and to limit the damage of the overall network. With respect to ChoKIFA, ChoKIFA+ further improves the network health status by reducing the routers PIT occupancy. The major contributions of this work are as follows:

- Design and implementation of a detection and mitigation mechanism against IFA, named as ChoKIFA [23], that relies on the CHOKe approach. ChoKIFA penalizes the malicious traffic by both dropping the new incoming MIs and removing the ones already stored in the PIT, without delay and without requiring information about the global network state;
- Design of an additional security mechanism applied on edge routers, named as ChoKIFA+. This approach aims to quickly identify and block IFA at the edge routers, thus improving the network health during the attack;
- Evaluation of the efficiency and effectiveness of ChoKIFA and ChoKIFA+, both implemented on the ndnSIM[1] simulator [25], by comparing them with the state-of-the-art mitigation approaches [16]. The results show that ChoKIFA+ effectively mitigates the IFA effects by guaranteeing an interest satisfaction rate (ISR) up to 98% and by reducing up to 40% the number of false positives. Moreover, comparing to ChoKIFA, ChoKIFA+ further improves the network health during IFA by reducing the PIT size up to 99 %.

In this chapter, we evaluate our ChoKIFA+ solution against all the three IFA types, having routers and legitimate traffic the primary victims of the attack. Using a valid name *prefix*, there are many ways for an adversary to generate unsatisfiable interests: by appending a random value to the name *prefix* (such interests are propagated towards the producer and are never satisfied), by replacing the `PublisherPublicKeyDigest` field with a random value (no public key would match this value, therefore, will never be satisfied) or by setting the `InterestExclude filter` to exclude all existing content starting with */prefix* (the interest can never be satisfied as it concurrently requests and excludes the same content).

We propose a solution that exploits an Active Queue Management (AQM) mechanism to defend against IFA in NDN. Our solution is the first one that removes the MIs from the PIT as a mitigation strategy; that works both as a detection and reaction approach; that avoids the drawbacks of the first and second categories of the state-of-the-art and that does not add new additional structures to routers, like the third category of the existing solutions.

5.1.2 Mitigating DDoS through Active Queue Management Schemes in IP

To mitigate DDoS attacks in IP, the congestion handling techniques, such as AQM, have gathered significant attention from the research community [71, 72, 73, 74]. AQM methods are classified into two categories according to their functionality and to the type of traffic they are able to handle [75]. The first category aims to provide fairness during network congestion, when the incoming traffic consists of only responsive flows (i.e., to which server responds). Random Early Detection (RED) [76], BLUE [77], and Adaptive Virtual Queue (AVQ) [78] approaches belong

[1] ndnSIM implements the NDN protocol stack on NS-3 simulator.

to the first category. On the other hand, the second category provides fairness when the incoming traffic consists of both responsive and unresponsive flows, such as CHOKe [22], Stochastic Fair Blue (SFB) [79], and Fair Random Early Detection (FRED) [80].

RED tries to acquire a better route queue stability by estimating the level of congestion in the router buffer and drops packets accordingly, by using an exponentially weighted moving average (EWMA) of the queue length. One of the RED limitations is that it is unable to identify unresponsive flows. Thus, it requires a significant parameters tuning to attain optimal results. To address this limitation, several methods, based on the idea of (or function with) RED, have been proposed: CHOKe [22], xCHOKe [81], RECHOKe [71] and FRED [80]. In particular, CHOKe is a stateless technique that tries to handle unresponsive flows by identifying and penalizing them through the drop of their packets.

5.2 IFA Mitigation through an Active Queue Management Scheme

In this thesis, we take a footstep in the direction of identifying and differentiating MIs from BIs during IFA, in order to selectively stop the malicious traffic. In particular, we exploit an AQM algorithm [22] (i.e., CHOose to Kill malicious Interest, CHOose to keep genuine Interest for IFA - ChoKIFA) to stabilize routers PIT, by dropping the incoming MIs and removing the ones already stored in the PIT, without any global knowledge of the network (i.e., state information).

5.2.1 Approach Overview

The fundamental idea behind ChoKIFA is to exploit the PIT state, which provides adequate statistics regarding the incoming and outgoing interest packets, and use it to identify and drop MIs. When an interest arrives at a router, ChoKIFA randomly draws an interest from the PIT and compares it with the incoming one. If both interests belong to the same traffic flow[2], then they are both dropped. Otherwise, the randomly drawn interest is left stored in the PIT and the incoming one is stored in the PIT according to a probability that depends on the level of PIT occupancy. The intuition of ChoKIFA is that, during IFA, the router PIT is likely to have more entries filled with MIs[3]. Thus, it is more likely that MIs will be chosen for the comparison with the incoming interests and that the malicious traffic will be stopped.

5.2.2 Adversary Model

Our adversary follows the third IFA type, as illustrated in Chapter 3, by generating a massive amount of MIs, requesting non-existing contents. We assume that the

[2]The NDN traffic flow measurement differs from the IP one and we present the comparison between them in Section 5.2.3.

[3]Recall that unsatisfiable interests refer to non-existing contents and saturate the PIT.

adversary aims to saturate the routers PIT through the rapid generation of a large number of MIs [16, 17, 18], so that, when the PIT is full, the incoming BIs are dropped. In comparison to the first and the second IFA types, the third one has a bigger impact for the following reasons:

- the MIs referring to non-existing contents are kept stored in the router PIT;
- the sending rate of MIs does not depend on the bandwidth allocated by the R to content packets, or on the capability of the adversary to receive content;
- MIs cannot be satisfied by the contents saved in router caches;
- if generated in a smart way (e.g., with a random component at the end of each name), MIs never collapse until the interests decay.

Without loss of generality, we assume that the adversary is able to corrupt a set of C (i.e., botnet) and use them to send the attack. This assumption is realistic and justified by the current scenarios of DDoS attacks [87]. The adversary issues MIs referring to a $prefix$, which is registered by P. To make each MI reach P, the adversary attaches a random value to each interest, i.e. $prefix/Rnd$ where Rnd is a random string. Moreover, the 50% of C in the whole network are used in the botnet [16]. Lastly, similar to C, the adversary starts sending MIs at time t. Table 5.1 summarizes the notations used in the thesis.

Table 5.1 – *Summary of the notations used.*

Notation	Meaning
Adv	adversary
C	consumer
P	producer
R	set of all routers
MI	malicious interest
BI	benign interest
r_i	i-th router, $r\epsilon\|R\|$
r_i^j	j-th interface of router i
$\delta(r_i^j)$	interest satisfaction rate
$\delta_{th}(r_i^j)$	threshold for interest satisfaction rate
$\rho_{avg}(r_i^j)$	average PIT size
w_ρ	weight factor for EWMA
$\rho_{size}(r_i^j)$	actual PIT size
$\rho_{th}^{min}(r_i^j)$	minimum threshold for PIT size
$\rho_{th}^{max}(r_i^j)$	maximum threshold for PIT size
P_b	interest drop probability
P_{max}	maximum drop probability

5.2.3 ChoKIFA: CHOose to Kill Interest Flooding Attack

To be effective against IFA, a mitigation approach has to differentiate between MIs and BIs. To this purpose, ChoKIFA relies on the traffic flow to differentiate and penalize the MIs from BIs.

Unlike in the IP architecture, where the traffic flow is measured through accountable attributes (e.g., source/destination address, interface number, number of packets/bytes sent forward and backward [88]), in NDN, the traffic flow is based on a content oriented communication model [84]. In particular, the traffic flow carries the content name, together with further information used to transport the multiple chunks or segments of the same content. Considering this, we identified the following three attributes to analyze the NDN traffic flow reaching each router:

- name prefix;
- interface;
- ISR, i.e., the rate between incoming interests and outgoing content, denoted as $\delta(r_i^j)$ and used to measure routers capability to satisfy interest on a particular interface [17, 25].

In particular, if $\delta(r_i^j) > 1$, it means that the number of content packets received by r_i^j is less than the number of interests forwarded from the same interface.

ChoKIFA is a justified stateless algorithm that does not require any specific data synchronization between intermediate routers and traffic analyzers. Compared to the existing rate limiting IFA approaches [16], ChoKIFA performs few operations and it not only restricts the rate of new incoming MIs, but also remove MIs which are already stored in PIT during attack. A detailed flowchart of ChoKIFA is given in Figure 5.1, i.e., explicitly presented in solid lines.

To mitigate IFA, ChoKIFA relies on the dynamic computation of the PIT size, denoted as $\rho_{size}(r_i^j)$, and on its evaluation against three thresholds: $\rho_{th}^{min}(r_i^j)$, $\rho_{th}^{max}(r_i^j)$ and $\delta_{th}(r_i^j)$ [17]. For each interest arriving at r_i^j, if the PIT size is less than the $\rho_{th}^{min}(r_i^j)$, the interest gets stored in the router's PIT. During a normal network trend, all interests sent by C are satisfied by P or by a router's cache. Moreover, there are no interests requested for non existing contents and there is no massive network traffic going to R. In this scenario, the PIT size almost never reaches $\rho_{th}^{min}(r_i^j)$. However, when there is either a default delay in the network (e.g., due to congestion or packet loss) or an ongoing IFA, the PIT size can reach $\rho_{th}^{min}(r_i^j)$. In case of network congestion, the content retrieval is delayed and the PIT entries are kept longer in the PIT, while, in case of IFA, the massive amount of MIs sent by the adversary to saturate the PIT. Thus, the choice of the $\rho_{th}^{min}(r_i^j)$ threshold is critical, since it has to consider both the legitimate busty traffic and the default delay (more details in Section 5.2.5.1). When the PIT size is greater than $\rho_{th}^{min}(r_i^j)$ and lower than $\rho_{th}^{max}(r_i^j)$ (i.e., $\rho_{th}^{min}(r_i^j) < \rho_{size}(r_i^j) < \rho_{th}^{max}(r_i^j)$), each new incoming interest is compared with an interest that is randomly selected from the PIT and named as *drop interest candidate*, to verify whether they belong to the same traffic flow, which means:

- having the same prefix;
- coming from the same interface;
- checking if the current $\delta(r_i^j)$ exceeds $\delta_{th}(r_i^j)$, in case both previous conditions are verified.

If both interests have the same traffic flow, then both are dropped, since most of the PIT entries are likely to be occupied by MIs under IFA. Otherwise, the randomly selected interest is kept stored in the PIT, and the incoming interest is dropped with the probability (P_b), which depends on the average PIT size ($\rho_{avg}(r_i^j)$), as illustrated in Equation 5.1 [76].

$$P_b = \frac{P_{max} * (\rho_{avg}(r_i^j) - \rho_{th}^{max}(r_i^j))}{(\rho_{th}^{max}(r_i^j) - \rho_{th}^{min}(r_i^j))}, \tag{5.1}$$

P_{max} denotes the maximum probability[4]. Since the average PIT size varies between $\rho_{th}^{min}(r_i^j)$ and $\rho_{th}^{max}(r_i^j)$, the interest dropping probability P_b varies between 0 and P_{max}. This means that an interest is dropped with a probability equal to 1, if it arrives when the average PIT size exceeds $\rho_{th}^{max}(r_i^j)$, otherwise it is accepted and stored in PIT. In particular, the interest dropping probability is computed by exploiting the mechanism of packet dropping probability of RED [76]. Algorithm 1 illustrates the functionality of the proposed approach at the core routers.

5.2.4 Edge Router Functionality

The enhanced version of ChoKIFA (i.e., ChoKIFA+) provides an additional security wall on the edges of the network to further improve the network health (i.e., the PIT occupancy) during IFA. In ChoKIFA+, the edge routers detect the malicious behaviour of the users close to the adversary, therefore, they reduce the amount of malicious traffic towards the core routers. In particular, the edge routers detect the adversary location and interface so that they are able to stop the malicious traffic coming from it. The flowchart in Figure 5.1 illustrates the functionality of edge routers (i.e., in dotted lines). In particular, an edge router monitors the statistics about the ISR for each consumer, considering the threshold $\delta_{th}(r_i^j)$ to discriminate between legitimate and malicious traffic. For each interest arriving at the edge router, if the current PIT size $\rho_{size}(r_i^j)$ is greater than $\rho_{th}^{min}(r_i^j)$, the edge router compares the current ISR for that interface (i.e., $\delta_{size}(r_i^j)$) with the threshold $\delta_{th}(r_i^j)$. If $\delta_{size}(r_i^j)$ is greater than $\delta_{th}(r_i^j)$, the consumer is considered malicious and the interface is blocked until the $\delta_{size}(r_i^j)$ ranges under threshold. The Algorithm 2 illustrates the procedure followed by the edge router when receiving an interest.

5.2.5 Parameters Setting

In ChoKIFA, the values of the parameters $\rho_{avg}(r_i^j)$, $\rho_{th}^{min}(r_i^j)$ and $\rho_{th}^{max}(r_i^j)$ are essential to make decisions about the desired average PIT size, which directly impacts the interest dropping probability. A proper choice of these parameters ensures the handling of small bursts of benign traffic, which might happen in case of network congestion or content retrieval delay. We will now illustrate the rules applied to

[4]We take the value of maximum probability (P_{max}) to be one.

Algorithm 1: Processing incoming packet to core router.

```
input : packet,PITsize,Minth,Maxth
begin
    if packet = Content then
        process packet as Content
    end
    if packet = Interest then
        if (PITsize < Minth) then
            Accepte(Interest)
        else
            Select the old interest (RI) from PIT
            if (PrÃ©fixe(Interest) = PrÃ©fixe(RI)) & (Interface(Interest) =
              Interface(RI)) & Ratio(Interface(Interest)) ≥ th then
                Drop(interest,RI)
            else
                if (PITsize > Maxth) then
                    Drop(Interest)
                else
                    Calculate drooping probability (RED) DPRED
                    if DPRED then
                        Accepte(Interest)
                    else
                        Drop(Interest)
                    end
                end
            end
        end
    end
end
```

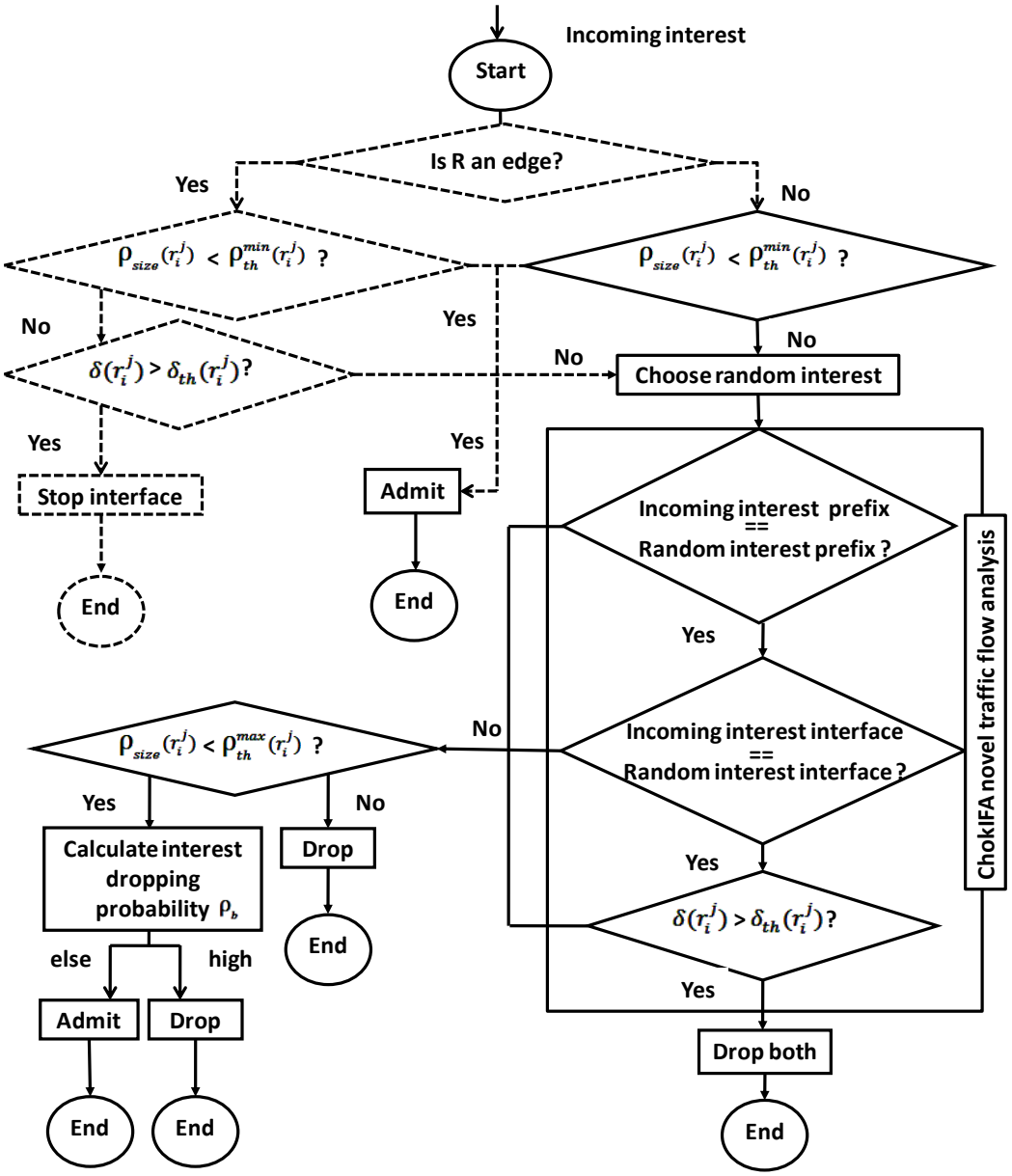

Figure 5.1 – *ChoKIFA+ algorithm flowchart.*

choose the best parameters values for both keeping an effective performance of the network and a mitigation of the attack.

5.2.5.1 Average PIT Size Calculation

To calculate the average PIT size $\rho_{avg}(r_i^j)$, ChoKIFA uses EWMA, that guarantees that a short term increase in PIT size, due to a burst of BIs, does not result in the significant increase of the average PIT size. The Equation 5.2 illustrates the calculation of $\rho_{avg}(r_i^j)$, where w_ρ is the weight factor for calculating EWMA and $\rho_{size}(r_i^j)$ is the current/actual PIT size. Then, the average PIT size, used for the interest dropping probability, is updated as follows [76],

$$\rho_{avg}(r_i^j) = (1 - w_\rho) * \rho_{avg}(r_i^j) + w_\rho * \rho_{size}(r_i^j). \tag{5.2}$$

Algorithm 2: Processing incoming packet to edge router.

```
input : packet,PITsize,Minth,Maxth
begin
    if packet = Content then
        process packet as Content
    end
    if packet = Interest then
        if (PITsize < Minth) then
            Accepte(Interest)
        end
        else
            if Ratio(Interface(Interest)) ≥ th then
                stop interface
            end
            else
                process as ChoKIFA
            end
        end
    end
end
```

Note that the calculation of the average PIT size can be made particularly efficient when w_ρ is set to a negative power of two. If w_ρ is too large, then the averaging procedure will not filter out the temporary congestion of PIT [76].

5.2.5.2 PIT Size Minimum Threshold

The optimal value for $\rho_{th}^{min}(r_i^j)$ depends on the desired level of the average PIT size and on the default network conditions. In case the default traffic is often congested, the $\rho_{th}^{min}(r_i^j)$ threshold should be large enough to allow the PIT usage under acceptable levels.

5.2.5.3 PIT size Maximum Threshold

The $\rho_{th}^{max}(r_i^j)$ threshold affects the maximum number of interests that can be stored in the PIT and its value is critical due to the consequences it might generate. If $\rho_{th}^{max}(r_i^j)$ is too low, the attack might be restricted or ineffective, since the MIs will not be stored in the PIT. However, a low $\rho_{th}^{max}(r_i^j)$ threshold may also damage the ChoKIFA+ performance detection algorithm, making it report a large number of false positives. This happens because the ChoKIFA+ detection algorithm requires MIs to be stored in the PIT and to allow the comparison between the incoming malicious traffic with the one already stored in the PIT. On the other side, if $\rho_{th}^{max}(r_i^j)$ is too high, the adversary could store even a larger amount of MIs and,

consequently, delay the detection of the attack and the sensitivity of the algorithm.

5.2.5.4 Setting $\rho_{th}^{max}(r_i^j)$ and $\rho_{th}^{min}(r_i^j)$ to Avoid Global Synchronization

The optimal value for $\rho_{th}^{max}(r_i^j)$ depends also from the maximum average delay that can be allowed to interest (e.g., round trip time for interest to retrieve data) and from the PIT total size. A useful rule used by ChoKIFA is to set the value of $\rho_{th}^{max}(r_i^j)$ more than three times the value of $\rho_{th}^{min}(r_i^j)$ [76], since the mitigation mechanism works efficiently when $\rho_{th}^{max}(r_i^j)$-$\rho_{th}^{min}(r_i^j)$ is larger than the typical increase in average PIT size.

5.3 Evaluation

In this section, we evaluate the effectiveness and efficiency of our proposed approach under IFA, by comparing ChoKIFA with ChoKIFA+ and with the state of the art IFA mitigation approaches [16]. In particular, among those we chose to compare with the ones that implement an interest rate limiting based on interface fairness, ISR and limit announcement technique. To this end, we implemented the protocol and performed extensive simulations using the open-source ndnSIM [85] simulator.[5]

5.3.1 Simulations Setup

We ran the simulations on two different network topologies, both for 100 seconds: a tree topology [19] and a more realistic large-scale ISP-like topology (i.e., AS-7018 [86]). We chose the tree topology because it provides one of the worsts scenarios to apply a defence against IFA [16], while we chose the larger ISP topology to evaluate the performance of the mitigation approach, when deployed on the real Internet. Table 5.2 illustrates the other network parameters we used for the simulation setup.

5.3.2 Evaluation Metrics

To evaluate the impact of IFA and to compare our solution with the state-of-the-art, we adopted the following metrics, which have been widely used in related work [17, 18, 21]:

- the PIT usage: this indicates the available capacity of the routers to process benign traffic during an attack;
- the percentage of BIs and MIs dropped by the network during IFA: this measures the attack impact and the effectiveness of the countermeasure;

[5] ndnSIM implements the NDN protocol stack on NS-3 simulator.

Table 5.2 – *Parameters for simulation*

Parameters	Value
Interest sending rate for C (interests/second)	30
Interest sending rate for Adv (interests/second)	1000
Interest size (kilobyte)	1
Number of C	8
Number of P	1
Number of malicious nodes	4
Number of benign nodes	4
Number of routers	9
Link capacity (Mbps)	10
Link delay (ms)	10
Interest life time (sec)	1
R Total PIT size (kilobyte)	600
Min. threshold for PIT size (kilobyte)	1/8 of PIT (75)
Max. threshold for PIT size (kilobyte)	3/4 of PIT (450)
Weight factor (w_ρ) for EWMA	0.001
P_{max}	1
Interest satisfaction ratio threshold	3
Simulation time (sec)	100
Simulator version	ndnSIM 2.1
Operating system	Ubuntu 16.04

- the ISR of benign users: this is intended to measure the benign traffic received by users during IFA. The lower the ISR, the greater the amount of false positives generated by the mitigation approach when distinguishing between MIs and BIs;
- any additional delay encountered by the clients, when the proposed mitigation approach is active:

5.3.3 Evaluation on a Small Scale Topology

As a small scale topology [19, 16], we used the one illustrated in Figure 5.2, that has multiple benign consumers (i.e., C) retrieving the desired content from a producer (i.e., P), that publishes contents under a specific name prefix (i.e., $prefix$). We assume that C sends a BI for a content specified as $prefix/data$, which can be satisfied by P after traversing multiple routers R. On the contrary, the adversary sends requests for non-existing content (i.e., MI), which exhibits a distinct suffix ($/good/rnd$) compared to the one of the existing contents ($/good/data$). Each router has the NDN default features [10] and it is referred as $r_i^j \; \epsilon \; |R|$, where j is the interface of i-th router. In addition, each router performs caching and uses *best route* as forwarding strategy.

In this section, we used the tree topology to evaluate:

- the impact of the three IFA types on the routers PIT occupancy and on the ISR of benign consumers;

- the effectiveness of ChoKIFA and ChoKIFA+ in terms of routers PIT occupancy and ISR of benign consumers;
- the effectiveness of ChoKIFA+ against the three IFA types in terms of routers PIT occupancy and ISR of benign consumers.

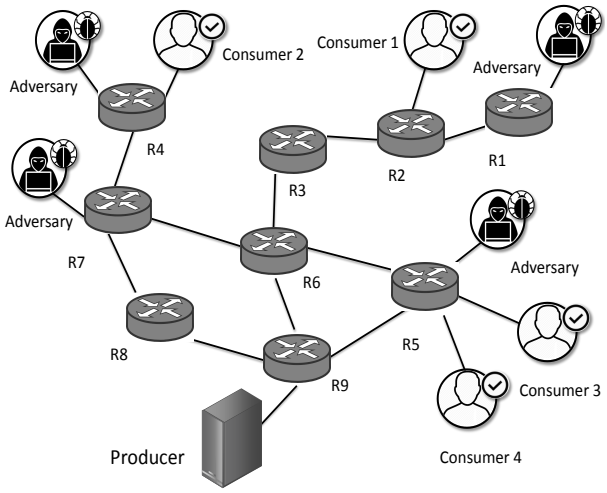

Figure 5.2 – *Small scale topology used for simulations.*

5.3.3.1 Impact of Interest Flooding Attacks

Figure 5.3 shows the impact of the three IFA types on the routers PIT occupancy with no active countermeasure. On the y-axis, we plot the average PIT usage (i.e., 600 kilobytes is the maximum PIT size) and the variation of its usage under IFA. At 20 sec, the adversary starts the attack by issuing MIs with a rate equal to 1000 interests/second. As shown in Figure 5.3, the impact of the third IFA type is more significant than the impact of the other two IFA types. This is motivated by the in-network caching, that provides an intrinsic defence against the first and the second IFA type, while in the third IFA type, the interests are kept stored in the PIT until their lifetime has passed. Before the 20th second, only legitimate consumers send BIs with an interest sending rate equal to 30 interests/second. Since those interests refer to existing contents, they are all satisfied and the PIT average occupancy is equal to zero.

Figure 5.4 shows the impact of all IFAs types on the consumers ISR with no active countermeasure. This quantifies the quality of service observed by legitimate users, while the network is under IFA. Under the first IFA type, consumers have an ISR between 20% and 30% higher than the ISR under the second IFA type. This result was expected, since requests for static contents are satisfied by all on-path-router's caches. On the contrary, the requests for dynamic contents are routed to the producer. Finally, the third IFA type generates the worst ISR, since the network drops almost 90% of the legitimate traffic. For the rest of simulations, we focus on the IFA where adversary generates unsatisfiable interests, i.e, the third IFA type.

5.3.3.2 Comparison between ChoKIFA and ChoKIFA+ Effectiveness

Figure 5.5 compares the effectiveness of ChoKIFA and ChoKIFA+ against the third IFA type in terms of PIT average usage. In the simulations, the adversaries start the attack at the same time as the benign users which send requests for existing content from the beginning of the simulation. Following the ChoKIFA algorithm, the PIT size needs to be filled with a certain amount of MIs before the drop of the malicious traffic starts. Thus, in both ChoKIFA and ChoKIFA+, the PIT size reaches a greater value than the minimum threshold, after which our approach starts treating each new incoming interest separately, identifying first its traffic flow (i.e., malicious or benign), and then deciding whether to drop the interest or not. However, with respect to ChoKIFA, ChoKIFA+ keeps the average PIT usage significantly lower because of the security wall introduced at the edge routers. Under IFA, the edge routers readily detect the attack by monitoring the flow of consumers over each interface.

Figure 5.6 compares the performance of ChoKIFA and ChoKIFA+ under IFA in terms of ISR observed by consumers. The legitimate traffic is slightly affected by the attack as, on average, only the 4% of BIs are dropped.

Finally, Figure 5.7 and Figure 5.8 illustrate the effectiveness of ChoKIFA+ in terms of PIT usage and ISR encountered by the consumers when the three IFA types are active. Thanks to the additional security wall introduced at the edge routers, ChoKIFA+ can readily recognize an attack and guarantee good network performances: an average PIT usage close to zero and a consumer ISR around 98%.

5.3.4 Evaluation on a Large Scale Topology

As a large scale topology, we considered the AS 7018 topology, measured by the Rocket fuel project [86] and shown in Figure 5.9. The topology involves 625 nodes, separated into three categories: clients, gateways, and backbones. 296 nodes are classified as clients due to a degree less than four, while the 108 nodes connected to clients are classified as gateways. The remaining 221 nodes are classified as backbones. A large ISP topology reflects how our mitigation methods would perform when deployed on the real Internet. To study the performance of our proposed mitigation strategy under a range of conditions, we varied the percentage of adversaries in the network and the frequency with which they send MIs.

5.3.4.1 ChoKIFA+ Effectiveness and Comparison with the State-of-the-Art

Figure 5.10 shows how ChoKIFA, ChoKIFA+ and four different IFA mitigation approaches perform with respect to the average PIT usage under the third IFA type. As discussed in Section 3.4, the four approaches, proposed in [16] and available online[6], are:

- *simple limits*,
- *token based*,

[6]https://github.com/cawka/ndnSIM-ddos-interest-flooding

- *satisfaction based* and
- *satisfaction pushback*.

In our simulation, the adversaries start launching the attack at the same time as the benign users start sending requests, i.e. from the beginning. As shown in the figure, ChoKIFA attains slightly higher PIT size than the *satisfaction pushback*. This is because all routers allow the PIT to be filled till the minimum threshold is reached. On the other side, ChoKIFA+ detects and blocks the malicious traffic at the edge routers, and stops the MI from being stored in the PIT.

Figure 5.11 illustrates the performance of all the approaches in terms of consumers ISR under an attack that starts at second 20 and ends at second 80 of the simulation. As shown in the figure, the rate limiting approaches [16] are not able to maintain an acceptable ISR for benign users in a large topology. In particular, only the *satisfaction pushback* provides a reasonable ISR, but it is overcome by both ChoKIFA and ChoKIFA+.

Figure 5.12 shows the ISR percentage for legitimate interests generated in the network, under a varying number of adversaries in the network (i.e., from 6% of adversaries to over 50% of adversaries in the network). For any mitigation technique, the ISR ratio for legitimate interests decreases (i.e., aggregated for all benign users) together with the increase of the number of adversaries. However, while the *token based* algorithm faces an ISR worsening already for 3 adversaries and 13 legitimate users, the other approaches have worse performances only for a higher number of adversaries. In particular, ChoKIFA+ outperforms all mitigation algorithms and shows a very minor reduction in ISR ratio (i.e., approximately 3%) even when the adversary percentage is raised more than 50%.

Figure 5.13 shows the aggregated legitimate ISR ratio under a varying adversary interest sending rate (i.e., from 100 interests/second to 10000 interests/second). As shown in the figure, both ChoKIFA and ChoKIFA+ are almost unaffected, even with a high increase in the adversary interest sending rate, while, among the state-of-the-art approaches, only the *satisfaction pushback* shows satisfactory results.

In Figure 5.14 and Figure 5.15, we show the impact of the ChoKIFA+ mitigation strategy under the three IFA types in a large topology. Considering Figure 5.14, ChoKIFA+ proves to be able to keep roughly the 90% of the PIT size avaiable for handling new traffic. Similarly, ChoKIFA+ is able to maintain almost the 97% of the global ISR.

Finally, we also analyzed the performance of ChoKIFA+ in terms of delay encountered between the interest sending and the corresponding data receiving. In particular, Figure 5.16 shows the additional delay introduced for each consumer by ChoKIFA+ with respect to the default delay measured in the network, without any attack. As shown in the figure, ChoKIFA+ introduces an extra delay equal to 10.9 microseconds on average.

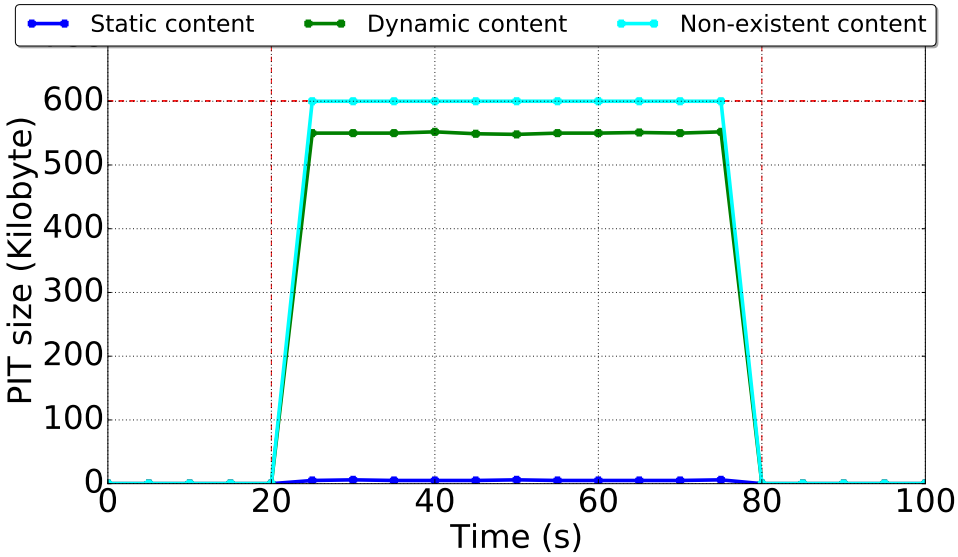

Figure 5.3 – *Impact of the three IFA types on the PIT usage in a small scale topology.*

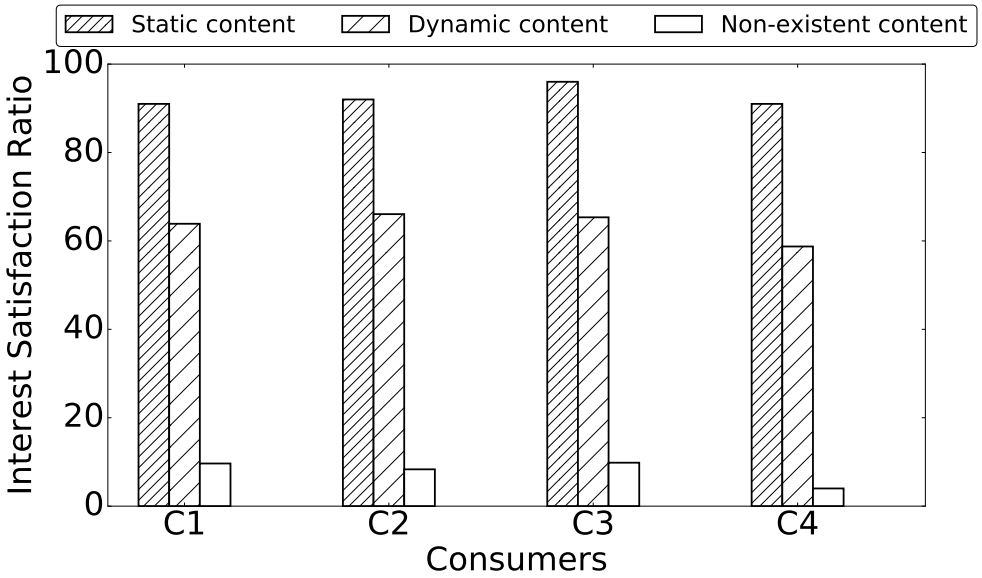

Figure 5.4 – *Impact of the three IFA types on consumers ISR in a small scale topology.*

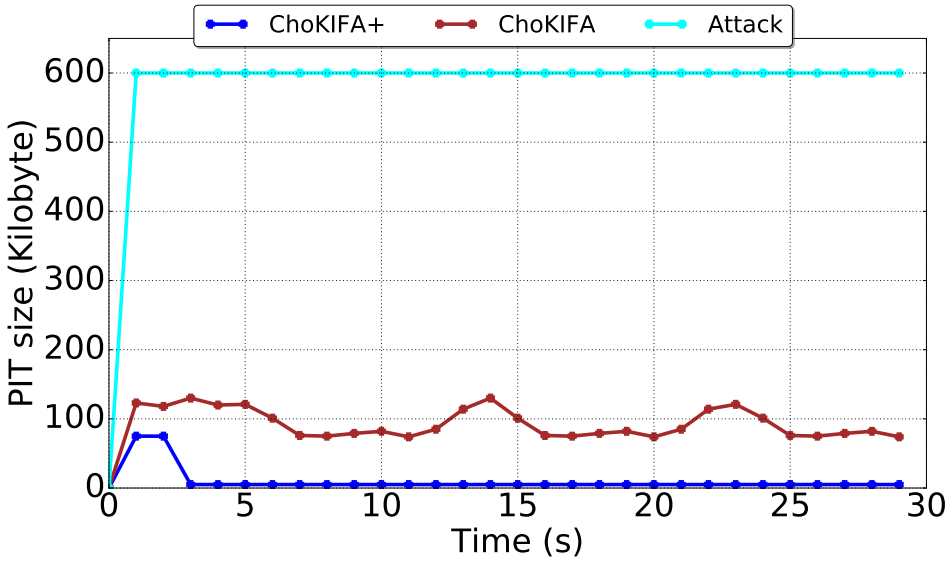

Figure 5.5 – *Effectiveness of ChoKIFA and ChoKIFA+ on the PIT usage in a small scale topology under the third IFA type.*

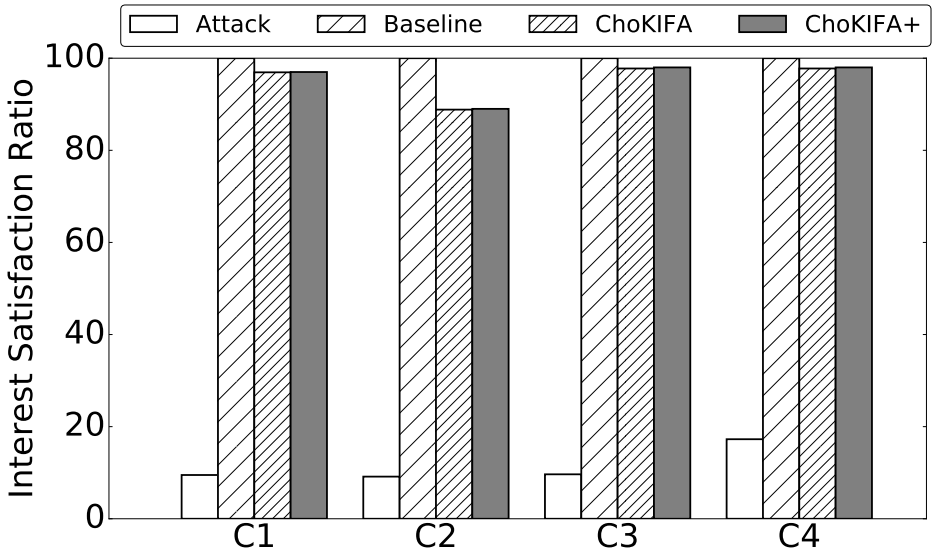

Figure 5.6 – *Effectiveness of ChoKIFA and ChoKIFA+ on consumers ISR in a small scale topology under the third IFA type.*

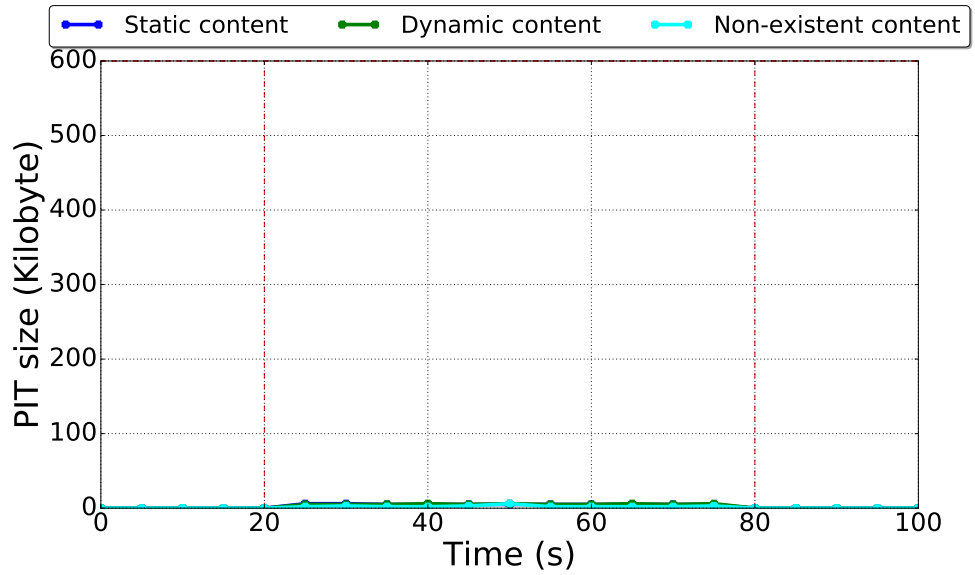

Figure 5.7 – *Effectiveness of ChoKIFA+ ont he PIT usage in a small scale topology under the three IFA types.*

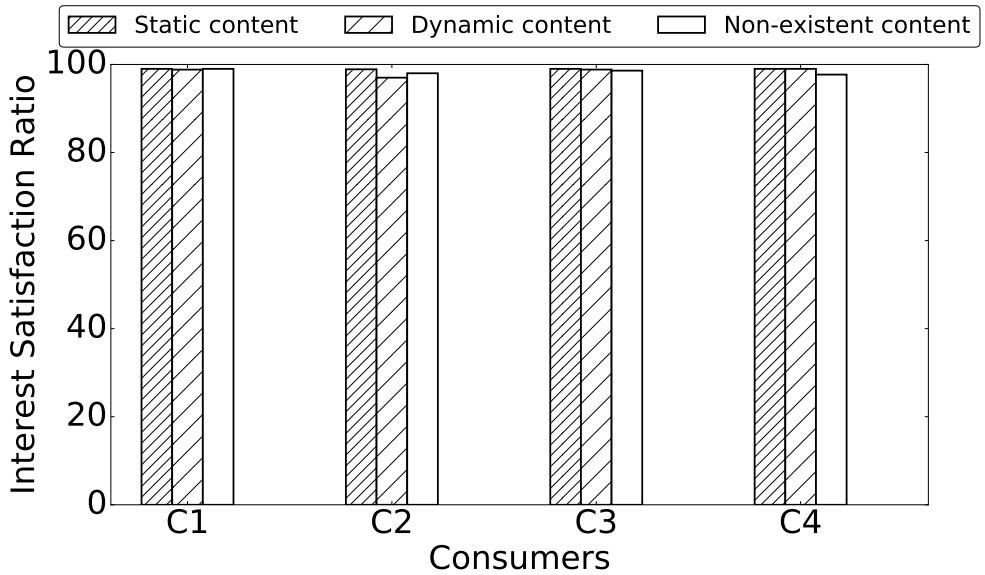

Figure 5.8 – *Effectiveness of ChoKIFA+ on consumers ISR in a small scale topology under the three IFA types.*

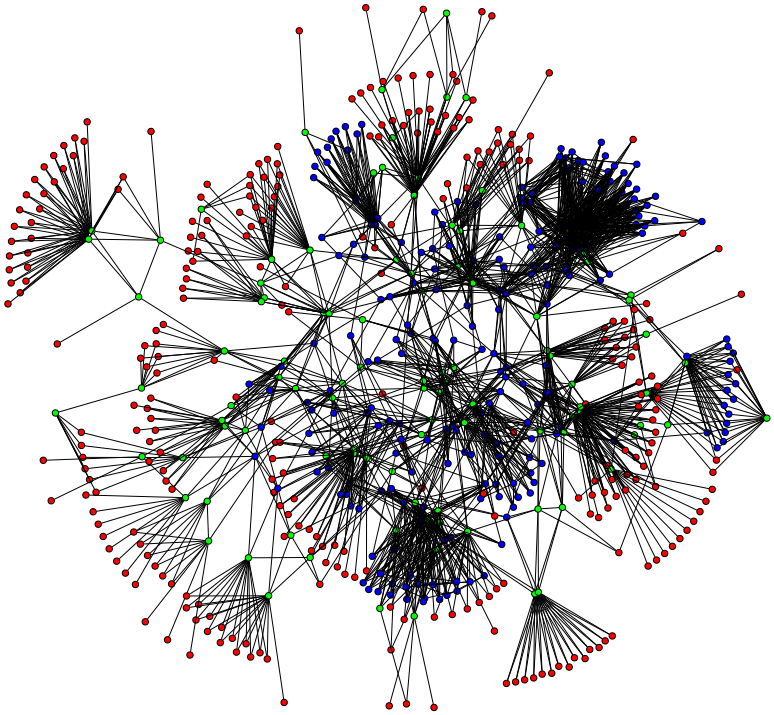

Figure 5.9 – *Large scale topology used for simulations.*

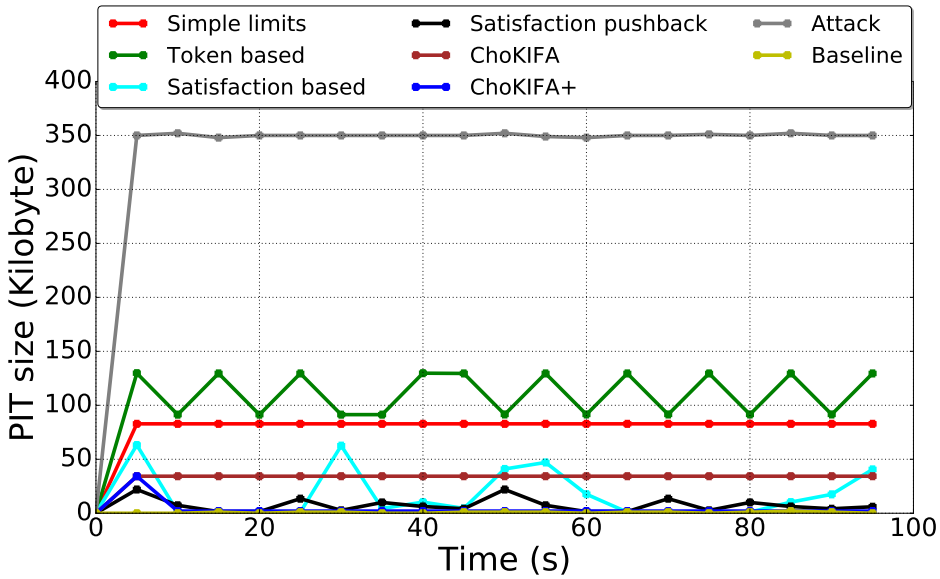

Figure 5.10 – *Effectiveness of ChoKIFA, ChoKIFA+ and the existing solutions on the PIT usage in a large scale topology under the third IFA type.*

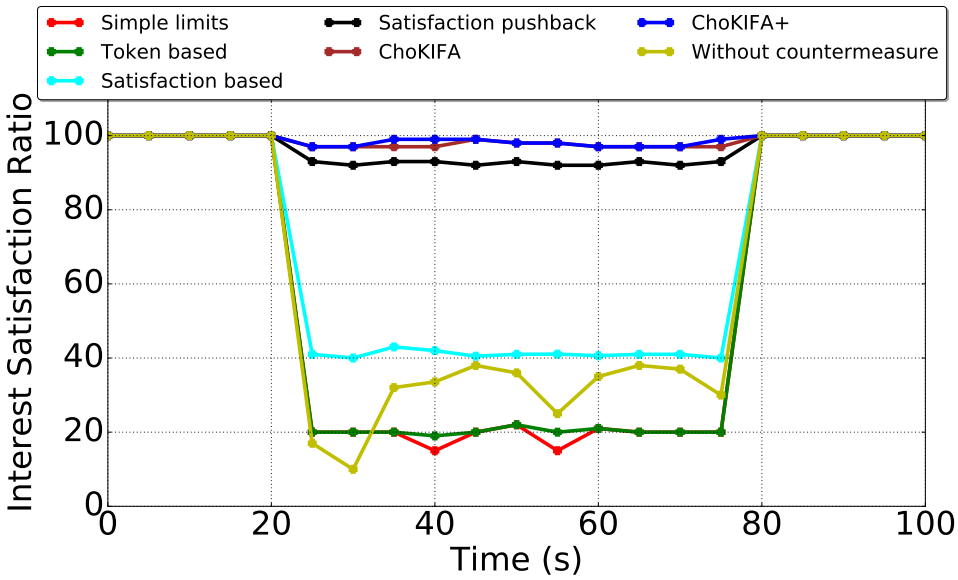

Figure 5.11 – *Effectiveness of ChoKIFA, ChoKIFA+ and the existing solutions on consumers ISR in a large scale topology under the third IFA type.*

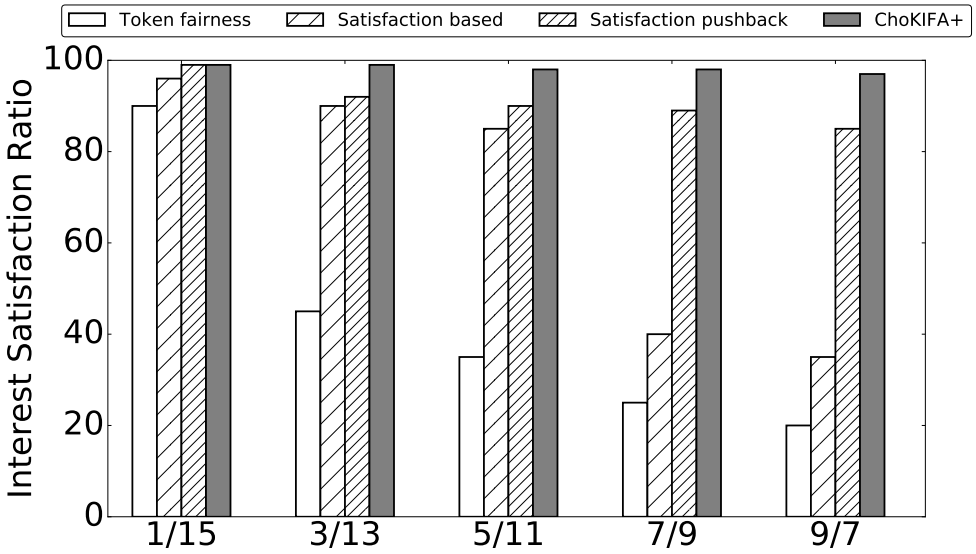

Figure 5.12 – *Consumers ISR with different mitigation approaches and an increasing number of adversaries in a large scale topology.*

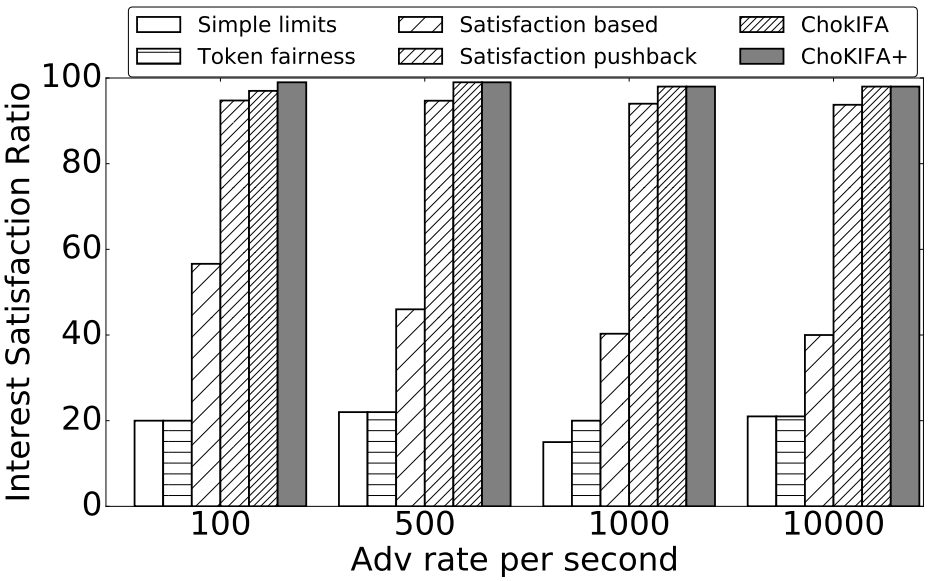

Figure 5.13 – *Consumers ISR with different mitigation approaches and an increasing MI sending rate in a large scale topology.*

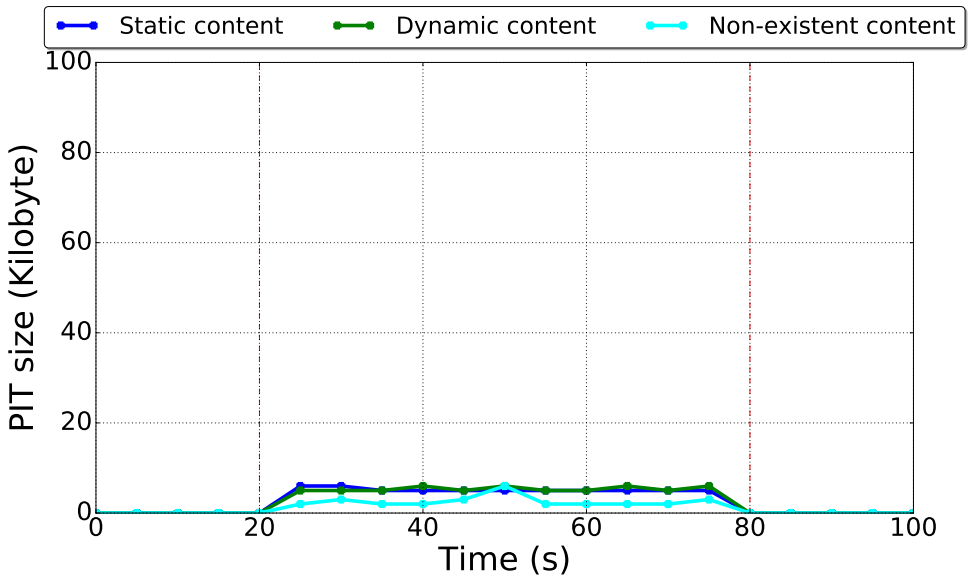

Figure 5.14 – *Effectiveness of ChoKIFA+ on the PIT usage under the three IFA types.*

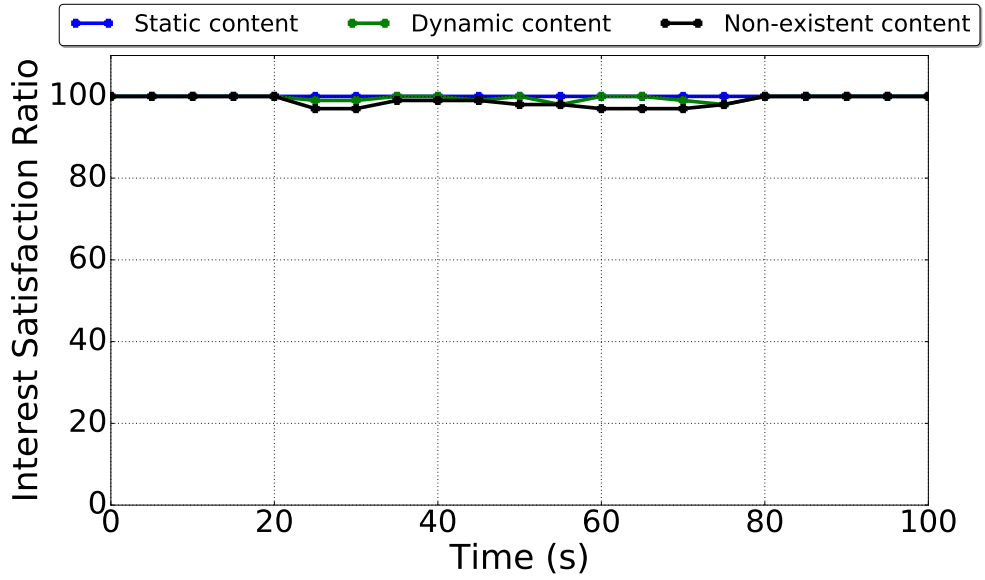

Figure 5.15 – *Effectiveness of ChoKIFA+ on consumers ISR under the three IFA types.*

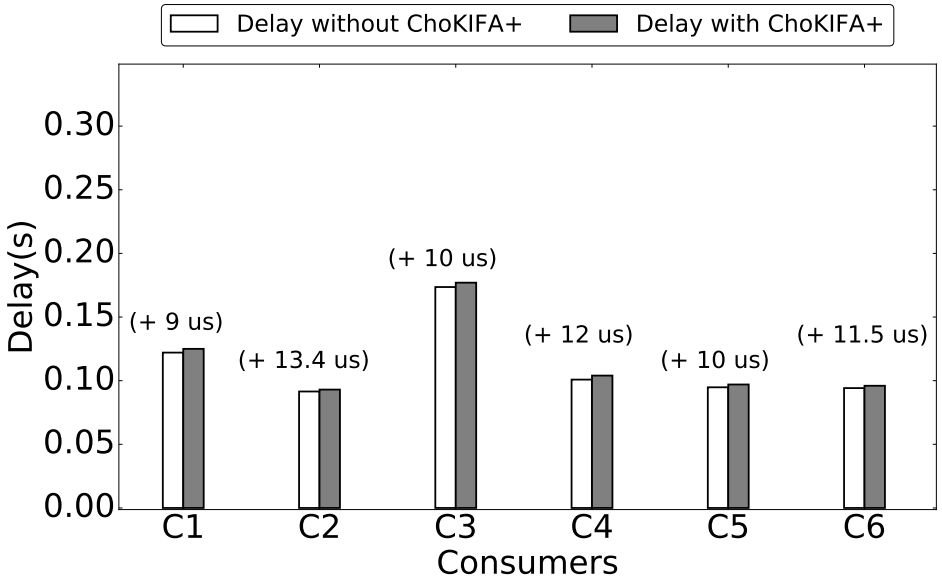

Figure 5.16 – *Additional delay time in µs introduced by ChoKIFA+.*

Conclusion

In this thesis, we have addressed the interest flooding-based DDoS over NDN, which is explicitly named as IFA. More specifically, we have found that several proposed countermeasures, that adopt detection and reaction mechanisms based on interest rate limiting, are not highly effective and also damage the legitimate traffic.

In our solution, we exploited an active queue management scheme to propose an efficient detection and mitigation mechanism against IFA, which stabilizes the router PIT. The proposed approach penalizes the unresponsive flows generated by adversarial traffic by dropping malicious interests generated during the IFA. We implemented the proposed protocol on the open-source ndnSIM simulator and compared it with the state-of-the-art. The results report that our proposed protocol effectively mitigates the adverse effects of IFA and shows significantly less false positives in comparison to the state-of-the-art IFA mitigation approaches.

Conclusion and Perspectives

In this thesis, we address the interest flooding-based DDoS over NDN, which is explicitly named as IFA. More specifically, we have found that several proposed countermeasures, that adopt detection and reaction mechanisms based on interest rate limiting, did not prove very successful and also penalize the legal traffic as well.

We have used an AQM approach to provide an effective mechanism for IFA detection and mitigation, which stabilizes the router PIT. The proposed approach penalizes the traffic generated by adversarial users by dropping MIs generated during the IFA. We provided, to the best of our knowledge, the first countermeasure based on an AQM technique and the first one that can remove the fake interests from the PIT.

In this thesis we have provided a working solution to counter this type of attacks. We've applied the protocol on an open-source ndnSIM simulator and compared it with the state-of-the-art in order to measure the efficacy of our approach. The findings show that our proposed protocol mitigates the adverse effects of IFA successfully and shows significantly less false positives in comparison to the state-of-the-art IFA mitigation approaches. We have used two typologies to understand and illustrate effects of IFA on the network. A simple topology has been used as a testing topology to understand how we can perform a devastating IFA and then we have verified results on a large scale topology.

We have demonstrated, in the both typologies, that IFA is a realistic threat; in particular, we have shown that, in the simple topology, an adversary with limited resources can reduce the amount of bandwidth allocated for content objects.

Then we have introduced a new mechanism for detecting and mitigating IFA based on an AQM technique. Our technique implements two mechanism to counter an attack: a local (ChoKIFA) and an Edge-router countermeasure (ChoKIFA+). While the first relies only on local metrics to detect and thwart an attack, the second exploits the edge routers to obtain better performance, in particular to detect as soon as possible an IFA. We have shown that the benefits of ChoKIFA are significant. Our experiments have suggested that local countermeasure provides a very interesting results. In the simple topology, the worst case restores more than 95% of the available bandwidth during the attack. In the large topology, results have shown that ChoKIFA restores more than 94% of the bandwidth. Simulation with Edge-router ChoKIFA+ countermeasure have revealed very important proprieties. Even if is not able to improve performance in term of ISR, they do not decrease it. In fact, in our simulation on simple topology, results of ChoKIFA+ are comparable with results of ChoKIFA.

Experiments with ChoKIFA+ countermeasure reveal that in the large topology we obtain a better performance that local countermeasure specially in term of PIT size. While have the same performance than local countermeasure in term of ISR. In fact, Edge-router ChoKIFA+ countermeasure are more effective in a large topology and this confirms that our solutions should perform very well on real network.

ChoKIFA and ChoKIFA+ are a first effort to counter IFA in NDN based on AQM. Even if has revealed very good performances and remove fake interests, we are aware that it needs a further refinement to improve its performance and to adapt it to all network topology. In the following section we present our plans for future works.

Future Works

Even if ChoKIFA and ChoKIFA+ are working solutions, we need more experiments and tests to improve and generalize it. In the following we describe our future steps to obtain a full working solution, with good performance in all typologies and with different attacks (more adversaries and different rates). Future works will be dedicated to the following areas:

- *Extend IFA investigation:* we want to conduct more experiment to study more deeply the effect of IFA with different parameters (e.g. adversary controls more consumers) on different typologies. If primary results tell us that IFA has very similar effects on the two used topology, we can not exclude a priory that for some other (larger) topologies IFA can reveal new different effects.

- *Other IFA types:* we want also to focus our attention on the other types of IFA and do more experiments to study more deeply the effect of other kinds of IFA like CIFA and Advanced IFA.

- *New metrics:* we want to improve the effectiveness of ChoKIFA+ by introducing new metrics for early detection of IFA. In particular we are studying solutions that consider also outgoing metrics and name space of interest to better identify fake interests.

- *Blockchain Reaction algorithm:* We plan to design more sophisticated reaction algorithms based on Blockchain technique and improve ChoKIFA+ performance.

Bibliography

[1] David R Cheriton and Mark Gritter. Triad: A new next-generation internet architecture. 2000. (Cité pages v, 5, 12, and 13.)

[2] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. *ACM SIGCOMM Computer Communication Review*, 37(4):181–192, 2007. (Cité pages v, 5, and 14.)

[3] Nikos Fotiou, Pekka Nikander, Dirk Trossen, and George C Polyzos. Developing information networking further: From psirp to pursuit. In *International Conference on Broadband Communications, Networks and Systems*, pages 1–13. Springer, 2010. (Cité pages v, 5, and 14.)

[4] Christian Dannewitz, Dirk Kutscher, BöRje Ohlman, Stephen Farrell, Bengt Ahlgren, and Holger Karl. Network of information (netinf)–an information-centric networking architecture. *Computer Communications*, 36(7):721–735, 2013. (Cité pages v, 5, and 16.)

[5] T Barnett, S Jain, U Andra, and T Khurana. Cisco visual networking index (vni), complete forecast update, 2017–2022. *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018. (Cité pages viii and 8.)

[6] Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D Thornton, Diana K Smetters, Beichuan Zhang, Gene Tsudik, Dan Massey, Christos Papadopoulos, et al. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 157:158, 2010. (Cité pages viii, 17, 24, and 25.)

[7] Abdelali Kerrouche. *Routing Named Data in Information-Centric Networks*. PhD thesis, 2017. (Cité pages viii, 28, 29, and 32.)

[8] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys Tutorials*, 16(2):1024–1049, Second 2014. (Cité pages 1 and 24.)

[9] Jianli Pan, Subharthi Paul, and Raj Jain. A survey of the research on future internet architectures. *IEEE Communications Magazine*, 49(7):26–36, 2011. (Cité page 1.)

[10] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014. (Cité pages 1, 24, 29, 51, and 72.)

[11] Van Jacobson et al. Networking named content. In *ACM International Conference on Emerging Networking Experiments and Technologies*, pages 1–12, 2009. (Cité pages 1, 24, and 29.)

[12] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36, 2012. (Cité pages 1 and 12.)

[13] Zhiyi Zhang, Yingdi Yu, Haitao Zhang, Eric Newberry, Spyridon Mastorakis, Yanbiao Li, Alexander Afanasyev, and Lixia Zhang. An overview of security support in named data networking, Revision 2, April 8, 2018. (Cité page 1.)

[14] Alberto Compagno, Mauro Conti, and Muhammad Hassan. *An ICN-Based Authentication Protocol for a Simplified LTE Architecture*. Springer International Publishing, Cham, 2018. (Cité page 1.)

[15] Guoqiang Zhang, Yang Li, and Tao Lin. Caching in information centric networking: A survey. *Comput. Netw.*, 57(16):3128–3141, November 2013. (Cité page 1.)

[16] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. Interest flooding attack and countermeasures in named data networking. In *IFIP Networking Conference, 2013*, pages 1–9. IEEE, 2013. (Cité pages 1, 2, 3, 42, 43, 47, 50, 51, 54, 57, 58, 62, 63, 65, 66, 71, 72, 74, and 75.)

[17] Alberto Compagno, Marco Conti, Paolo Gasti, and Gene Tsudik. Poseidon: Mitigating interest flooding ddos attacks in named data networking. In *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, pages 630–638. IEEE, 2013. (Cité pages 1, 2, 43, 47, 50, 51, 54, 62, 65, 66, and 71.)

[18] Kai Wang, Huachun Zhou, Yajuan Qin, Jia Chen, and Hongke Zhang. Decoupling malicious interests from pending interest table to mitigate interest flooding attacks. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 963–968. IEEE, 2013. (Cité pages 1, 41, 45, 47, 62, 65, and 71.)

[19] Huichen Dai, Yi Wang, Jindou Fan, and Bin Liu. Mitigate ddos attacks in ndn by interest traceback. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 381–386. IEEE, 2013. (Cité pages 1, 2, 43, 47, 50, 51, 54, 62, 71, and 72.)

[20] Vassilios G Vassilakis, Bashar A Alohali, ID Moscholios, and Michael D Logothetis. Mitigating distributed denial-of-service attacks in named data networking. In *Proceedings of the 11th Advanced International Conference on Telecommunications (AICT), Brussels, Belgium*, pages 18–23, 2015. (Cité pages 1, 43, 50, and 62.)

[21] H. Salah, J. Wulfheide, and T. Strufe. Coordination supports security: A new defence mechanism against interest flooding in ndn. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pages 73–81, Oct 2015. (Cité pages 1, 2, 50, 54, 62, and 71.)

[22] Rong Pan, Balaji Prabhakar, and Konstantinos Psounis. Choke-a stateless active queue management scheme for approximating fair bandwidth allocation. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 942–951. IEEE, 2000. (Cité pages 2, 46, 48, 50, 62, and 64.)

[23] Abdelmadjid Benarfa, Muhammad Hassan, Alberto Compagno, Eleonora Losiouk, Mohamed Bachir Yagoubi, and Mauro Conti. Chokifa: A new detection and mitigation approach against interest flooding attacks in ndn. In *International Conference on Wired/Wireless Internet Communication*, pages 53–65. Springer, 2019. (Cité pages 2, 47, 62, and 63.)

[24] Abdelmadjid Benarfa, Muhammad Hassan, Eleonora Losiouk, Alberto Compagno, Mohamed Bachir Yagoubi, and Mauro Conti. Chokifa+: an early detection and mitigation approach against interest flooding attacks in ndn. *International Journal of Information Security*, pages 1–17, 2020. (Cité page 2.)

[25] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, et al. ndnsim: Ndn simulator for ns-3. *University of California, Los Angeles, Tech. Rep*, 4, 2012. (Cité pages 2, 63, and 66.)

[26] Cisco. The zettabyte era: Trends and analysis. Technical report, Cisco, June 2017. (Cité page 6.)

[27] Cisco. Cisco visual networking index: Forecast and methodology. Technical report, Cisco, June 2017. (Cité pages 6 and 7.)

[28] worldometers. https://www.worldometers.info/world-population/. Technical report, 2019. (Cité page 6.)

[29] Hesham Farahat. *Proactive Caching to support mobility in Named Data Networks*. PhD thesis, Queenâs University Kingston, Ontario, Canada, 2017. (Cité page 7.)

[30] Erik Nygren, Ramesh K Sitaraman, and Jennifer Sun. The akamai network: a platform for high-performance internet applications. *ACM SIGOPS Operating Systems Review*, 44(3):2–19, 2010. (Cité page 7.)

[31] Sandvine. Global internet phenomena. Technical report, October. (Cité page 8.)

[32] Matteo D'Ambrosio, Christian Dannewitz, Holger Karl, and Vinicio Vercellone. Mdht: a hierarchical name resolution service for information-centric networks. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 7–12. ACM, 2011. (Cité page 10.)

[33] Gabriel M Brito, Pedro Braconnot Velloso, and Igor M Moraes. Information-centric networks. *Inf-Centric Netw*, pages 13–22, 2013. (Cité page 10.)

[34] Ali Ghodsi, Teemu Koponen, Jarno Rajahalme, Pasi Sarolahti, and Scott Shenker. Naming in content-oriented architectures. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 1–6. ACM, 2011. (Cité page 10.)

[35] Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox. Information-centric networking: seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, page 1. ACM, 2011. (Cité pages 10 and 12.)

[36] Larry Masinter, Tim Berners-Lee, and Roy T Fielding. Uniform resource identifier (uri): Generic syntax. 2005. (Cité page 10.)

[37] Andriana Ioannou and Stefan Weber. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Communications Surveys & Tutorials*, 18(4):2847–2886, 2016. (Cité page 11.)

[38] Wenjie Li, Sharief MA Oteafy, and Hossam S Hassanein. Rate-selective caching for adaptive streaming over information-centric networks. *IEEE Transactions on Computers*, 66(9):1613–1628, 2017. (Cité page 11.)

[39] George Xylomenos, Christopher N Ververidis, Vasilios A Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V Katsaros, and George C Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 16(2):1024–1049, 2013. (Cité page 11.)

[40] Giuseppe Rossini and Dario Rossi. A dive into the caching performance of content centric networking. In *2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 105–109. IEEE, 2012. (Cité page 11.)

[41] Ibrahim Abdullahi, Suki Arif, and Suhaidi Hassan. Survey on caching approaches in information centric networking. *Journal of Network and Computer Applications*, 56:48–59, 2015. (Cité page 11.)

[42] Sherin Abdelhamid, Hossam S Hassanein, Glen Takahara, and Hesham Farahat. Caching-assisted access for vehicular resources. In *39th Annual IEEE Conference on Local Computer Networks*, pages 28–36. IEEE, 2014. (Cité page 12.)

[43] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009. (Cité pages 12, 17, 24, 30, and 31.)

[44] Dmitrij Lagutin, Kari Visala, and Sasu Tarkoma. Publish/subscribe for internet: Psirp perspective. *Future internet assembly*, 84:75–84, 2010. (Cité page 15.)

[45] Marcus Brunner. Scalable & adaptive internet solutions (sail). *Future Internet Assembly (FIA), Ghent, Belgium*, 2010. (Cité page 16.)

[46] Van Jacobson, Diana K Smetters, Nicholas H Briggs, Michael F Plass, Paul Stewart, James D Thornton, and Rebecca L Braynard. Voccn: voice-over content-centric networks. In *Proceedings of the 2009 workshop on Re-architecting the internet*, pages 1–6. ACM, 2009. (Cité page 17.)

[47] Jeff Burke, Paolo Gasti, Naveen Nathan, and Gene Tsudik. Securing instrumented environments over content-centric networking: the case of lighting control and ndn. In *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 394–398. IEEE, 2013. (Cité page 17.)

[48] Lucas Wang, Ryuji Wakikawa, Romain Kuntz, Rama Vuyyuru, and Lixia Zhang. Data naming in vehicle-to-vehicle communications. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 328–333. IEEE, 2012. (Cité page 17.)

[49] Fabio Angius, Mario Gerla, and Giovanni Pau. Bloogo: Bloom filter based gossip algorithm for wireless ndn. In *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pages 25–30. ACM, 2012. (Cité page 17.)

[50] Steven DiBenedetto, Paolo Gasti, Gene Tsudik, and Ersin Uzun. Andana: Anonymous named data networking application. *arXiv preprint arXiv:1112.2205*, 2011. (Cité page 17.)

[51] Lixia Zhang et al. Named data networking. *ACM SIGCOMM CCR*, 44(3):66–73, 2014. (Cité pages 24 and 26.)

[52] Alberto Compagno. *Poseidon:Mitigating Interest Flooding DDoS Attacks in Named Data Networking*. UNIPD Thesis, 2012. (Cité page 25.)

[53] Cheng Yi, Jerald Abraham, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. On the role of routing in named data networking. In *Proceedings of the 1st ACM Conference on Information-Centric Networking*, pages 27–36. ACM, 2014. (Cité page 31.)

[54] Haowei Yuan, Tian Song, and Patrick Crowley. Scalable ndn forwarding: Concepts, issues and principles. In *2012 21st International Conference on computer communications and networks (ICCCN)*, pages 1–9. IEEE, 2012. (Cité page 31.)

[55] Hani Salah, Julian Wulfheide, and Thorsten Strufe. Coordination supports security: A new defence mechanism against interest flooding in ndn. In *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pages 73–81. IEEE, 2015. (Cité pages 36 and 38.)

[56] Paolo Gasti, Gene Tsudik, Ersin Uzun, and Lixia Zhang. Dos and ddos in named data networking. In *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pages 1–7. IEEE, 2013. (Cité pages 37, 43, 50, and 51.)

[57] Hani Salah and Thorsten Strufe. Evaluating and mitigating a collusive version of the interest flooding attack in ndn. In *Computers and Communication (ISCC), 2016 IEEE Symposium on*, pages 938–945. IEEE, 2016. (Cité page 37.)

[58] Salvatore Signorello, Samuel Marchal, Jérôme François, Olivier Festor, and Radu State. Advanced interest flooding attacks in named-data networking. In *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, pages 1–10. IEEE, 2017. (Cité page 38.)

[59] Reza Tourani, Satyajayant Misra, Travis Mick, and Gaurav Panwar. Security, privacy, and access control in information-centric networking: A survey. *IEEE communications surveys & tutorials*, 20(1):566–600, 2017. (Cité page 39.)

[60] Gang Liu, Wei Quan, Nan Cheng, Kai Wang, and Hongke Zhang. Accuracy or delay? a game in detecting interest flooding attacks. *Internet Technology Letters*, 1(2):e31, 2018. (Cité pages 41 and 47.)

[61] Amin Karami and Manel Guerrero-Zapata. A hybrid multiobjective rbf-pso method for mitigating dos attacks in named data networking. *Neurocomputing*, 151:1262–1282, 2015. (Cité page 42.)

[62] 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*, page 2. ACM, 2014. (Cité page 42.)

[63] Jayakrishna Kidambi, Dipak Ghosal, and Biswanath Mukherjee. Dynamic token bucket (dtb): A fair bandwidth allocation algorithm for high-speed networks. *J. High Speed Netw.*, 9(2):67â87, October 2000. (Cité page 42.)

[64] Ahmed Benmoussa, Abdou el Karim Tahari, Nasreddine Lagaa, Abderrahmane Lakas, Farhan Ahmad, Rasheed Hussain, Chaker Abdelaziz Kerrache, and Fatih Kurugollu. A novel congestion-aware interest flooding attacks detection mechanism in named data networking. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE, 2019. (Cité pages 43 and 47.)

[65] Tan Nguyen, Remi Cogranne, and Guillaume Doyen. An optimal statistical test for robust detection against interest flooding attacks in ccn. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 252–260. IEEE, 2015. (Cité pages 44 and 47.)

[66] Tan Nguyen, Hoang-Long Mai, Rémi Cogranne, Guillaume Doyen, Wissam Mallouli, Luong Nguyen, Moustapha El Aoun, Edgardo Montes De Oca, and Olivier Festor. Reliable detection of interest flooding attack in real deployment of named data networking. *IEEE Transactions on Information Forensics and Security*, 14(9):2470–2485, 2019. (Cité pages 44 and 47.)

[67] Tan Nguyen, Hoang-Long Mai, Guillaume Doyen, Remi Cogranne, Wissam Mallouli, Edgardo Montes de Oca, and Olivier Festor. A security monitoring plane for named data networking deployment. *IEEE Communications Magazine*, 56(11):88–94, 2018. (Cité pages 44 and 47.)

[68] Rui Hou, Min Han, Jing Chen, Wenbin Hu, Xiaobin Tan, Jiangtao Luo, and Maode Ma. Theil-based countermeasure against interest flooding attacks for named data networks. *IEEE Network*, 33(3):116–121, 2019. (Cité pages 44 and 47.)

[69] Jiaqing Dong, Kai Wang, Yongqiang Lyu, Libo Jiao, and Hao Yin. Interestfence: Countering interest flooding attacks by using hash-based security labels. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 527–537. Springer, 2018. (Cité pages 45 and 47.)

[70] Xin Zhang and Ru Li. A charging/rewarding mechanism-based interest flooding attack mitigation strategy in ndn. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 402–407. IEEE, 2019. (Cité pages 45 and 47.)

[71] Changwang Zhang, Jianping Yin, Zhiping Cai, and Weifeng Chen. Rred: robust red algorithm to counter low-rate denial-of-service attacks. *IEEE Communications Letters*, 14(5), 2010. (Cité pages 46, 63, and 64.)

[72] Visvasuresh Victor Govindaswamy, Gergely Záruba, and G Balasekaran. Rechoke: a scheme for detection, control and punishment of malicious flows in ip networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 16–21. IEEE, 2007. (Cité pages 46 and 63.)

[73] X. Jiang, J. Yang, G. Jin, and W. Wei. Red-ft: A scalable random early detection scheme with flow trust against dos attacks. *IEEE Communications Letters*, 17(5):1032–1035, May 2013. (Cité pages 46 and 63.)

[74] Harkeerat Bedi, Sankardas Roy, and Sajjan Shiva. Mitigating congestion based dos attacks with an enhanced aqm technique. *Computer Communications*, 56:60 – 73, 2015. (Cité pages 46 and 63.)

[75] Harkeerat Bedi, Sankardas Roy, and Sajjan Shiva. Mitigating congestion-based denial of service attacks with active queue management. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 1440–1445. IEEE, 2013. (Cité pages 46 and 63.)

[76] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking*, 1(4):397–413, 1993. (Cité pages 46, 52, 53, 54, 63, 67, 69, 70, and 71.)

[77] Wu-chang Feng, Kang G Shin, Dilip D Kandlur, and Debanjan Saha. The blue active queue management algorithms. *IEEE/ACM Transactions on Networking (ToN)*, 10(4):513–528, 2002. (Cité pages 46 and 63.)

[78] Srisankar S Kunniyur and Rayadurgam Srikant. An adaptive virtual queue (avq) algorithm for active queue management. *IEEE/ACM Transactions on Networking (ToN)*, 12(2):286–299, 2004. (Cité pages 46 and 63.)

[79] Wu-chang Feng, Dilip D Kandlur, Debanjan Saha, and Kang G Shin. Stochastic fair blue: A queue management algorithm for enforcing fairness. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1520–1529. IEEE, 2001. (Cité pages 46 and 64.)

[80] Dong Lin and Robert Morris. Dynamics of random early detection. In *ACM SIGCOMM Computer Communication Review*, volume 27, pages 127–137. ACM, 1997. (Cité pages 46 and 64.)

[81] Parminder Chhabra, Shobhit Chuig, Anurag Goel, Ajita John, Abhishek Kumar, Huzur Saran, and Rajeev Shorey. Xchoke: Malicious source control for congestion avoidance at internet gateways. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 186–187. IEEE, 2002. (Cité pages 46 and 64.)

[82] Sandesh Rai, Kalpana Sharma, and Dependra Dhakal. A survey on detection and mitigation of distributed denial-of-service attack in named data networking. In *Advances in Communication, Cloud, and Big Data*, pages 163–171. Springer, 2019. (Cité page 46.)

[83] Kai Wang, Huachun Zhou, Hongbin Luo, Jianfeng Guan, Yajuan Qin, and Hongke Zhang. Detecting and mitigating interest flooding attacks in content-centric network. *Security and Communication Networks*, 7(4):685–699, 2014. (Cité page 47.)

[84] S. Oueslati, J. Roberts, and N. Sbihi. Flow-aware traffic control for a content-centric network. In *2012 Proceedings IEEE INFOCOM*, pages 2417–2425, March 2012. (Cité pages 52 and 66.)

[85] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012. (Cité pages 54 and 71.)

[86] Neil Spring et al. Measuring ISP Topologies with Rocketfuel. *IEEE/ACM Trans. Netw.*, 2004. (Cité pages 54, 57, 71, and 74.)

[87] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. Dos and ddos in named data networking. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, July 2013. (Cité page 65.)

[88] N. Brownlee, C. Mills, and G. Ruth. Traffic flow measurement: Architecture. 1997. (Cité page 66.)

Acronymes

AQM	Active queue management
BIs	Benign Interests
CCN	Content Centric Network
CDN	Content Delivery Network
CHOKe	CHOose and Kill for unresponsive flows
ChoKIFA	Choose To Kill IFA
CIFA	Collusive Interest Flooding Attack
CS	Content Store
DACS	Denial of service against content source
DT	Decay Time
DNS	Domain Name System
DONA	Data Oriented Network Architecture
DoS	Denial of Service
DPE	Disabling PIT Exhaustion
EWMA	ExponentiallyWeightedMoving Average
FIB	Forwarding Information Base
FN	Forwarding Node
ICN	Information Centric Network
IFA	Interest Flooding Attack

IP Internet Protocol

ISP Internet Service Provider

ISR interest satisfaction ratio

MIs Malicious Interests

NACK Negative ACKknowledge

NDN Named Data Networking

NDNsim Named Data Simulator

NDNsim Named Data Simulator

NACK Negative acknowledgments

NetInf Network of Information.

NFD NDN Forwarding Daemon

PIT Pending of Interest Table

PURSUIT PUblish SUbscribe Internet Technology

QoS Quality of Service

RED Random Early Detection

RSI Relative Strength Index

RENE REndezvous NEtwork

RFC Request For Comment

RH Resolution Handler

RId Rendezvous Identifier

RN Rendezvous Node

RTT Round Trip Time

TCP Transmission Control Protocol

TDM Threshold-based Detection and Mitigation

TM Topology Management

TTL Time To Live