Coordination Supports Security: A New Defence Mechanism Against Interest Flooding in NDN

Hani Salah Julian Wulfheide
Computer Science Department, TU Darmstadt
Hochschul Str. 10, 64289 Darmstadt, Germany
hsalah@cs.tu-darmstadt.de julian.wulfheide@ps.tu-darmstadt.de

Thorsten Strufe
Computer Science Department, TU Dresden
Mommsen Str. 8, 01187 Dresden, Germany
thorsten.strufe@tu-dresden.de

***Abstract*—Named-Data Networking (NDN) is a promising architecture for future Internet. Its design, however, can be misused to perform a new DDoS attack known as the Interest Flooding Attack (IFA). In IFA, the attacker issues non-satisfiable interest packets, aiming to drop legitimate interest packets by overwhelming pending interest tables in NDN routers. Prior defence mechanisms are not highly effective, harm legitimate interest packets, and/or incur high overhead.**

We propose a coordinated defence mechanism against IFAs. We realize our solution by adapting CoMon, a framework that we developed previously to coordinate caching-related decisions in NDN, motivated by its effective, yet affordable, coordination. In our solution, IFAs are detected and mitigated by few routers based on aggregated knowledge of traffic and forwarding states. These routers are selected by a novel heuristic enabling them to observe the entire traffic at an early stage. Extensive simulations confirm the feasibility and effectiveness of our solution.

***Index Terms*—Named-Data Networking; Interest Flooding Attack; Coordinated Defence**

I. Introduction

The current TCP/IP-based Internet was originally designed for reliable host-to-host communications. Today, however, the Internet traffic is dominated by content distribution and retrieval applications. These applications generate massive and ever increasing traffic volumes [1], mainly due to redistribution of popular content, causing high charges for network operators.

Several solutions have been proposed in the last years to narrow this gap between the Internet design and its current usage (see [2] for an overview). Among them, Named-Data Networking (NDN) [3] is widely considered as a promising architecture for future Internet. In essence, NDN shifts the current sender-driven host-centric communication model to a receiver-driven content-centric one.

The work that we present in this paper is stimulated from systematic vulnerability of a new technology that is proposed to potentially replace the current Internet architecture. In particular, we focus on an NDN-tailored DDoS attack coined in the literature with the term *Interest Flooding Attack (IFA)*. The adversary, aiming at flooding the network and obstructing the service received by legitimate users, misuses two design properties of NDN: (*i*) routing based on longest name-prefix match, and (*ii*) storing a forwarding state per interest packet in the so-called Pending Interest Table (PIT). The adversary sends interest packets with unique fake content names targeting name-space(s). As a consequence, one PIT entry is created per interest packet in each NDN router on the path. These entries stay in the PITs till they expire at the end. Succeeding to overload some or all PITs lead to legitimate interest packets being dropped.

Despite the considerable amount of research on IFA, proposed defence mechanisms (e.g. see [4]–[8] and the references therein) have one or more of the following drawbacks: First, attack detection is difficult or inaccurate close to sources, especially for distributed low-rate IFAs, because the observable amount of traffic is relatively small. In contrast, detection close to targets is likely not robust due to the large volume of attack traffic. Second, legitimate traffic can be damaged because proposed reactions do not distinguish legitimate packets from malicious ones. Third, every router is required to perform attack detection and mitigation. Collaborative mechanisms also require routers to communicate with each other. Such requirements cause high communication overhead. Moreover, independent or not well coordinated decisions can result in inaccurate attack detection, overreactions, or inequitable punishments [6].

Our key contribution in this paper is a new defence mechanism against IFA. The mechanism detects and mitigates IFAs in a *coordinated* way. This is done based on *aggregated* and *timely* knowledge of both: (*i*) content access information and (*ii*) forwarding states of interest packets. In practice, we adapt CoMon, a framework for **Co**ordination which is based on *lightweight* **Mon**itoring. This choice is motivated by CoMon's ability to realize coordination that is both efficient and feasible. This was shown in our prior work [9] in which we developed CoMon to coordinate caching-related decisions in NDN in a domain-wide scale.[1]

Our solution assigns monitoring tasks to a small group of routers through which network traffic is expected or enforced to pass. We develop a heuristic to select these routers. The heuristic gives more weight for routers that appear on more routes, yet consider their closeness to clients (thus to sources of attacks). IFAs are detected and mitigated by monitoring routers with the aid of a centralized controller.

[1]We use the terms *domain network* or *domain* for referring to an *autonomous system* (as defined in [10]).

978-1-4673-6770-7/15/$31.00 ©2015 IEEE

Evaluating our solution through extensive simulations, we show that it can counter IFAs effectively, with very low communication overhead.

The remainder of this paper is organized as follows: Section II overviews NDN and IFA. Then, Section III discusses the related work. Next, we overview our solution in Section IV, describe its specifications in Section V, and evaluate it in Section VI. Finally, Section VII concludes the paper.

II. Background

In this section, we give a general review of the NDN architecture and the IFA attack in Section II-A and Section II-B, respectively.

A. Primer on Named-Data Networking

Named-Data Networking (NDN) [3] is one of the projects funded by the American's National Science Foundation for future Internet architectures. It was initiated at Xerox PARC by Van Jacobson and others, aiming mainly at coinciding the host-centric design of the Internet with its current content-centric usage.

In principle, NDN's design is based on four core concepts:

1) *Networking named content:* Each content is identified by a unique hierarchical name (e.g. ***"/org/ieeelcn/papers/comon.pdf"***). Clients access content by its name, rather than locations or host addresses.
2) *On-path caching:* When a content is retrieved, a copy of it is cached in each router along the path from the content provider to the consumer. LRU or LFU is used for content replacement, in case the cache space is full.
3) *Consumer-driven communication model:* Clients use *interest* packets to request named contents. The content itself is delivered inside a *data* packet on the same path through which it was requested, in reverse way. At most one data packet can be retrieved per interest packet.
4) *Content-based security:* Content authenticity and integrity are dealt via a digital signature added to each data packet. The signature is computed by the origin content provider over the content's name and the content itself, thus binding them with each other. The creator's public key can be retrieved from information contained in the packet. This way, the packet's authenticity and integrity can be verified regardless from where the packet is retrieved.
   Interest packets, in contrast, do not include a signature filed. Hence, their origin and integrity are unverifiable.

The router model in NDN consists of three data structures: the *Content Store (CS)* temporarily holds data packets passing through the router. The *Pending Interest Table (PIT)* maintains content names of recently received, but still not satisfied, interest packets. Each PIT entry also specifies the incoming interface(s) through which the corresponding interest packet was received. A PIT entry is removed either when the corresponding data packet is received, or if its timeout is caught. The *Forwarding Information Base (FIB)*, acting as a routing table, maintains a list of potential outgoing interfaces for different content names and name-prefixes.

With such a router model, interest and data packets are handled in NDN as follows: Upon receiving an interest packet, the router looks for a matching name in its CS. If found, it forwards the corresponding data packet to the same interface from which the interest packet was received. Otherwise, the router looks for the name in its PIT. If a matching entry is found but the interface from which the interest packet was received is not listed, the new interface is appended to the same entry, and nothing otherwise. This way, NDN routers avoid forwarding duplicate copies of identical interest packets. If no matching PIT entry is found, a new one is created, then the FIB is consulted, and the packet is routed accordingly.

When receiving a data packet, the router first looks for the content name in its PIT. If found, the data packet is cached in the CS, then forwarded to the listed interfaces, and lastly the respective PIT entry is deleted. If no matching PIT entry found, the packet is discarded.

The aforementioned properties of NDN, namely: in-network caching, stateful forwarding plane (using PITs), content-based security, and name-based routing and forwarding (i.e. without host addresses), make the network robust to several types of traditional DDoS attacks. For instance, reflection attacks, bandwidth depletion, black-holing, and prefix hijacking are eliminated or at least mitigated in NDN by design [1]. Furthermore, since NDN does not use a name resolution service, DNS cache poisoning and similar attacks do not represent a threat in NDN.

B. Interest Flooding Attack

Despite the aforementioned security features of NDN, its design opens the door for new types of attacks [11]. Among those, the Interest Flooding attack (IFA) can be mounted by taking advantage of two NDN's properties: (*i*) storing a forwarding state per interest packet in each crossed router, and (*ii*) routing by longest name-prefix match.

In practice, the adversary (through distributed bots) produces a large number of interest packets and inserts them into the network. Such an attack aims at overloading: (*i*) PITs of NDN routers, so they cannot handle legitimate interest packets, and/or (*ii*) the targeted content provider. Note that since contents in NDN are requested by their names, it is difficult to attack specific hosts or routers in the network. Instead, the adversary can easily target name-space(s).

IFAs are classified, according to the interest packets used in the attacks, into three types [4]. In particular, interest packets can be used to request: (*i*) existing, (*ii*) dynamically-generated, or (*iii*) non-existent content. Caches of NDN routers provide a built-in mitigation for type (*i*). The attack in type (*ii*) aims at exhausting resources of origin content providers on serving malicious interest packets, and consuming routers' PITs. However, its impact on PITs is eliminated once the corresponding data packets are received. In type (*iii*), as can be seen in the example in Fig. 1, the adversary targets

a specific name-space[2] (e.g. *"/org/ieeelcn/"*), by producing interest packets using name-prefixes belonging to the targeted name-space (e.g. *"/org/ieeelcn/papers/"*) appended by a unique suffix per packet. The name suffixes are chosen such that the resulted content names are fake, i.e. they will not be satisfied. Consequently, a PIT entry is created per fake interest packet in each router crossed by the packet. Please note that the total number of unique fake interest packets sent during the attack represents an upper bound on the amount of memory that can be exhausted in PITs of victim routers. Please note also that the closer the router to the content provider responsible for the targeted name-space, the more the fake interest packets it receives, thus the larger the impact of the attack on the router's memory. Since malicious interest packets are non-satisfiable, the corresponding PIT entries remain till they eventually expire.

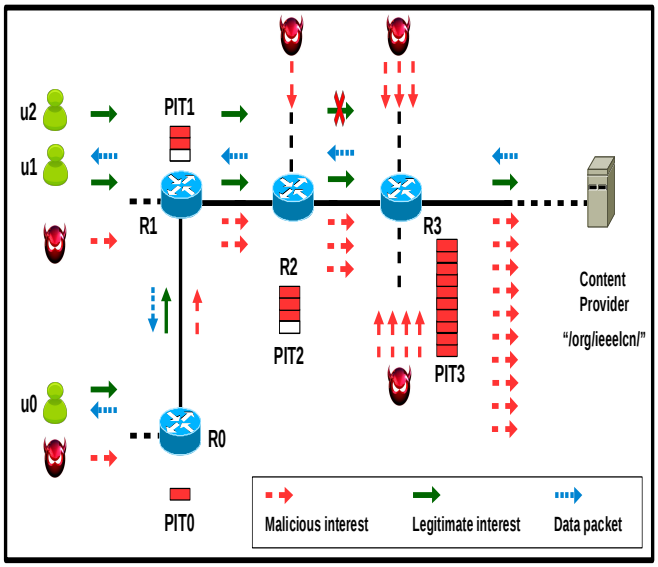

Fig. 1: Example of IFA: All interest packets use the same name-prefix *"/org/ieeelcn/"*, thus are routed toward the same server. At the beginning, u0 & u1, almost concurrently, issue one interest packet each, using the same suffix. Hence, both packets are aggregated, i.e. hold a single PIT entry in each bypassed router. The created entries are eliminated once the corresponding data packet arrives. Then, the five adversaries issue malicious interest packets (10 in total) almost concurrently. Each of those packets uses a unique non-existent suffix. Consequently, a single PIT entry is created per malicious interest packet in each bypassed router, staying till it expires. Next, before the entries corresponding to the malicious interest packets expire, u2 issues a legitimate interest packet. Assuming a PIT capacity of 10, this packet is dropped by R3 because R3's PIT is full.

Due to the aforementioned consequences of type (*iii*), it is considered the most harmful type of IFA. In this paper, we focus on this type only, and associate it with the term IFA from now onwards.

III. Related Work

We restrict the discussion in this section to prior work on NDN's IFA. For a broad overview of research on security of NDN and other ICN architectures, the reader is referred to [11].

[2] This way, the malicious interest packets are routed towards and as close to the origin data provider (the victim) as possible, which increases the attack effectiveness [6].

IFA was discussed for the first time by Lauinger [12]. After that, Gasti et al. [4] detailed IFA's operation and types. Both [12] and [4] suggested tentative defence mechanisms against IFA. Evaluations of those mechanisms, however, were left for future research. Afterwards, several studies evaluated the effectiveness of IFA, and agreed on the necessity to protect routers and content providers in NDN against such an attack. These studies also proposed and evaluated several defence mechanisms against IFA. Developed mechanisms can be classified as either: *autonomous* or *collaborative*.

On the one hand, in the autonomous mechanisms, each individual router detects attacks based on its *local* view of network traffic and/or PIT usage. For instance, an attack is detected when the observed ratio of unsatisfied interest packets (or PIT expiration rate) exceeds a preset threshold. Then, a reaction (e.g. dropping part of suspicious incoming interest packets) is taken by *each* router *independently*. Such mechanisms, although being simple and inexpensive, have three key drawbacks: First, attack detection is difficult or inaccurate close to attack sources, because the amount of received traffic (thus the attack's effect) is small. This is true particularly for distributed low-rate IFAs. Late detection and reaction, in contrast, take place after wasting lots of resources in several areas of the network, and may cannot prevent the attack before it causes a big damage. Secondly, independent attack reactions result in overreactions or inequitable punishments [6]. The third drawback is that applied detection algorithms do not distinguish malicious interests from legitimate ones, thus harm legitimate traffic.

Notable examples of autonomous mechanisms include [7], satisfaction-based acceptance [6], and the autonomous version of Poseidon [5]. Widjaja [13] presented another autonomous mechanism with a unique approach. In particular, it proposes to defend against IFAs by removing the PIT completely from the router model. Instead, routers cache interest packets in the CS as regular data packets. Such a mechanism, however, does not address the aforementioned drawbacks completely. In addition, it makes indexing and forwarding of interest packets more complicated, thus was considered impractical [14].

In the collaborative countermeasures, on the other hand, routers exchange information about their local observations and taken reactions. This information is used by routers to update parameters related to attack detection (e.g. thresholds) and reaction (e.g. percentage of dropped interest packets). By this, routers can detect and mitigate attacks faster (i.e. while they are in progress) and close to their sources. This applies for Interest Traceback [8] and pushback-like mechanisms like: satisfaction-based pushback [6], Cooperative-Filter [14], and the collaborative version of Poseidon [5]. These mechanisms, although outperform their autonomous counterparts, still may suffer from one or more of the aforementioned drawbacks. Furthermore, they result in high communication overhead.

The basic idea of our solution along with a preliminary evaluation were presented in [15].

IV. Solution: Requirements and High-Level Overview

A. Design Requirements

In this paper, we aim to achieve effective defence against IFAs, that can overcome the drawbacks of prior defence mechanisms (see Section I and Section III). Towards this end, the design of our defence mechanism is guided by the following requirements:

R1) IFAs should be detected based on aggregated (can be network-wide) information of packets transmitted and corresponding forwarding states. Such information enables for accurate attack detection, even for low-rate IFAs (which cannot be detected by individual nodes autonomously).

R2) Both the detection and mitigation of IFAs should be performed at an early stage, before malicious interest packets consume lots of resources.

R3) Both duplicate attack detection and overreactions should be avoided.

R4) The mechanism should be able to distinguish between legitimate interest packets and malicious ones, to avoid damaging legitimate traffic.

R5) Overhead of coordination should be low.

B. System Architeture and Operation Primitives

While the intended benefits of fulfilling the requirements above look appealing, realizing a solution that fulfils all of them in practice is challenging. In particular, the distributed nature of the Internet and domain networks, in addition to the huge volume and high dynamics of coordination-related information (i.e. information to be aggregated and disseminated), render such a solution impractical.

In [9], we addressed a similar challenge (coordinating caching-related decisions in a domain-wide scale) by CoMon, a framework for coordination in NDN. Motivated by CoMon's prior results that show its ability to realize effective, yet scalable, coordination, we decided to use its design concepts as a basis for our IFA's defence mechanism.

By adapting CoMon, our solution relies on a small number of NDN routers to monitor network traffic and forwarding states of interest packets. Attacks are detected either by monitoring routers and/or by a centralized controller aggregating observations of all monitoring routers. Reaction to potential attacks is then performed by monitoring routers accordingly.

More particularly, our solution (as in the original CoMon framework) is designed to work within an autonomous domain. Each domain network consists of a set V of routers. As can be seen in Fig. 2, the system architecture has three principal components: a Domain Controller (DC), NDN Routers (NRs), and Monitoring Routers (MRs). In the following, we introduce these components and describe how basically do they work together to provide a coordinated defence against IFAs:

1) *Domain Controller (DC):* Each domain has a controller that collects monitored information about exchanged packets and corresponding forwarding states from a predetermined set of monitoring routers. The DC uses this information to detect attacks on a domain-wide scale, and then shares attack information with monitoring routers. In the current version of our solution, we implement the DC on a single machine. Instead, DC can be implemented on multiple machines to distribute the load and to avoid single-point-of-failure. Such a design is out of the scope of this paper.
2) *NDN Routers (NRs):* These are similar to standard NDN routers (mainly performing routing and caching tasks).
3) *Monitoring Routers (MRs):* A set of $M \subset V$ routers are selected as MRs. In addition to the functions of regular NDN routers[3], MRs monitor interest packets passing through them, and check whether these packets are satisfied (by data packets) or not. Consequently, MRs compute expiration rates of their PIT entries per interface, and determine malicious interfaces and name-prefixes. MRs periodically report summaries of their observations and results to the DC. As mentioned above, the DC in turn sends MRs a feedback summarizing information of attacks ongoing over the entire domain. MRs are also responsible for mitigating potential IFAs.

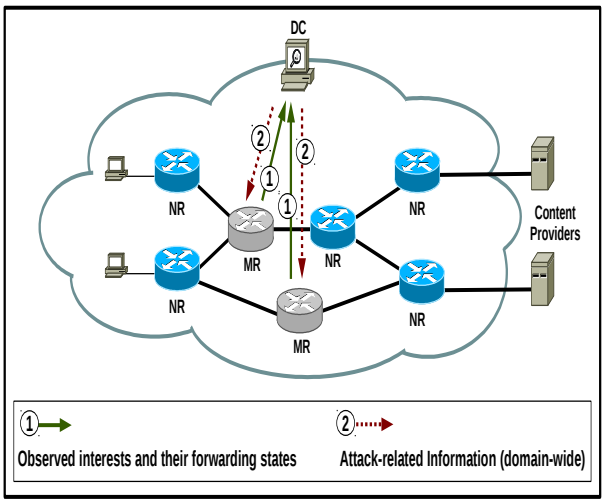

Fig. 2: System architecture (adapted from [9]): "DC" stands for Domain Controller, "NR" for NDN Router, and "MR" for Monitoring Router

To avoid duplicate monitoring and detection of IFAs. MRs do not check interest packets checked previously by another MR. For this purpose, once an interest packet is captured by an MR for the first time, that MR sets a newly added one-bit field in the interest packet, called *"Checked"*, to 1 (0 by default). In addition to this, to avoid overreactions, a reaction decision is taken and performed, per interest packet, only once by the first encountered MR.

[3] We use the term *router* (or *node*) in a generic way for referring to any router $v \in V$ (i.e. either NR or MR). In contrast, we use *NR* for referring to a router u without monitoring capabilities (i.e. $u \in V : u \notin M$).

V. Solution: Design Specifications

We describe in this section the design specifications of our solution that are not discussed in Section IV.

A. Placement of MRs

We argue that the success in detecting IFAs is highly dependent on the amount of traffic that is covered by MRs. However, we also argue that closeness of MRs to clients (thus to potential attack sources) should be also taken into account when placing MRs. That is to say, placing MRs closer to attack sources increases the chance for detecting and reacting to IFAs at an early stage (i.e. before routers waste lots of resources).

The corresponding placement problem can be formulated as follows: *"Given a domain consisting of a set V of routers, which set $M \subset V$ should be selected as MRs such that they together cover the entire traffic, while both $|M|$ and hop counts between MRs and clients are minimized"*.

Such a problem has been shown to be NP-hard [16]. Therefore, we develop a new placement heuristic, called *Placement based on covered Routes and Closeness to Sources (PRCS)*.

The pseudo-code of PRCS is provided in Algorithm 1, and it can be outlined as follows:

Algorithm 1 Placement based on covered Routes and Closeness to Sources (PRCS)

1: $R \leftarrow \varnothing$ ▷ Set of routes
2: $N \leftarrow \varnothing$ ▷ Set of non-gateway routers
3: **for** each $x = 1, 2, ..., X$ **do**
4: Using [17], find r routes $\{P_1^x, P_2^x, ..., P_r^x\}$ from $s(x)$ to $t(x)$
5: $R \leftarrow R \cup \{P_1^x, P_2^x, ..., P_r^x\}$
6: $N \leftarrow N \cup$ non-gateway routers on $\{P_1^x, P_2^x, ..., P_r^x\}$
7: **end for**
8: $M \leftarrow \varnothing$ ▷ Set of MRs
9: **while** $|M| \leq p$ **do** ▷ p: a predetermined value
10: $max \leftarrow 0$
11: **for** each router n in N **do**
12: $W(n) \leftarrow 0$
13: **for** each route r in R **do**
14: $W(n) \leftarrow W(n) + w_r(n)$
15: **end for**
16: **if** $max < W(n)$ **then**
17: $max \leftarrow W(n)$
18: $m \leftarrow n$
19: $C \leftarrow$ routes that n locates on
20: **end if**
21: **end for**
22: $M \leftarrow M \cup \{m\}$
23: $N \leftarrow N - \{m\}$
24: $R \leftarrow R - C$
25: **end while**

1) Using [17], identify the set R of routes (from each consumer router, i.e. source of interest packets, to each content provider), and then identify the set N of non-gateway routers that locate on the identified routes (lines: 3 – 7).
2) Calculate the weight $W(n)$ for each router $n \in N$ (lines: 12 – 15), summing n's partial weights on each route $r \in R$: Considering n's closeness to the beginning of the route (i.e. to the consumer router), its partial weight on r is calculated as follows:

$$w_r(n) = \begin{cases} 1 + \frac{h_r(n)}{l(r)}, & n \ locates \ on \ r \\ 0, & otherwise \end{cases}$$

where $l(r)$ and $h_r(n)$ denote the length of r (i.e. hop count) and n's position on r, respectively. $h_r(n)$ takes the value 0 if n locates next to the gateway node, and incremented by 1 with each hop towards the source otherwise. This idea is inspired from [18].
3) The router with the maximum total weight is then selected as an MR (lines: 16 – 20) and added to the set M (line: 22). The selected router is removed from N (line: 23), and the routes on which it locates are removed from R (line: 24).
4) Repeat step 2 and step 3 till the cardinality of M equals a predetermined value p.

B. Maximizing Traffic Coverage

While PRCS considers the fraction of routes covered by selected MRs, few MRs may do not achieve full traffic coverage. Furthermore, traffic in NDN-like networks can be filtered by caches or PITs, or dropped by a router (e.g. reacting to an attack) before it is intercepted by an MR. To maximize the amount of traffic covered by MRs, CoMon incorporates two techniques, namely: (*i*) *MR-Aware Routing (MAR)* and (*ii*) *Forward-Till-Be-Monitored (FTBM)*. When both MAR and FTBM are enabled, each interest packet (thus the corresponding data packet) is *enforced* to pass through an MR. In the following, we overview those two techniques.[4]

With MAR, each interest packet is first routed towards an MR (e.g. closest MR in the basic version of MAR), and secondly routed from the designated MR towards the original target. This two-phase routing results in extra hops, unless the designated MR locates on the default route.

As for FTBM, it eliminates the effects of the aforementioned filters. More precisely, when a router receives an interest packet that is not monitored yet (i.e. $Checked = 0$) and finds a matching data packet in its cache or a matching PIT entry, or decided to drop the packet, the router: (*i*) looks for the closest MR in its FIB, say ***"MRx"***, (*ii*) adds the prefix ***"/MRx/served"*** or ***"/MRx/dropped"*** (correspondingly) to the original name, and then (*iii*) forwards the packet accordingly, i.e. to ***"MRx"***. The designated MR, in turn, performs detection- and monitoring-related tasks on the received packet and drops it afterwards.

The additional overhead caused by FTBM is measured by the number of hops traversed by the interest packet since it was served till reaching the designated MR. Note that, since the interest is already consumed, this overhead does not apply for the corresponding (larger size) data packet. Furthermore, *served* and *dropped* interest packets are not stored in PITs.

[4] For more details about the algorithmic design and hop count overhead of MAR and FTBM (shown to be low), the reader is referred to [9].

C. Detection of IFAs

IFAs are *basically* detected by each MR independently. The detection algorithm can be outlined as follows:

1) MR m continuously monitors: the utilization of its PIT, arriving interest packets, and corresponding PIT entries.
2) m calculates the PIT utilization ratio $U(m,q)$. This ratio represents the maximum number of PIT entries observed during observation window q, divided by the PIT space.
3) m calculates the PIT expiration rate $E(m_f,q)$. This rate is calculated per incoming interface f over observation window q, as follows:

$$E(m_f,q) = \frac{e(m_f,q)}{e(m_f,q) \ + \ s(m_f,q)}. \qquad (1)$$

   $e(m_f,q)$ and $s(m_f,q)$ represent the corresponding counts of expired and satisfied PIT entries, respectively. Both values consider only PIT entries that correspond to interest packets not monitored before (i.e. $Checked = 0$). The two detection parameters above are similar to the ones which have been used in [14]. However, aiming to achieve better detection, our detection algorithm employs them differently (as we detail in this section).
4) m triggers the reaction function at end of q if: $E(m_f,q)$ is positive and $U(m,q)$ exceeds a threshold τ. In this case, m identifies the name-prefixes of expired PIT entries. Identified name-prefixes as well as the interface f are then considered *infected*. The MR then calculates the PIT expiration rate $E(m_f^j,q)$ for each infected name-prefix j (calculated as in Eq. 1, but only for entries with prefix j). Next, m sends a report to the DC including its observations and results (Fig. 2).

To minimize false positives, due to transient failures in accessing the network or in delivery of packets, τ should be given a large value. However, such a setting reduces the sensitivity of the detection function for low-rate IFAs. To address this drawback, we include an additional domain-wide detection in our defence mechanism, performed by the DC. The corresponding algorithm consists of the following steps:

1) The DC aggregates monitoring information that it received from MRs at the end of q.
2) For each name-prefix j, the DC calculates the ratio $Q(j,q)$ which represents the maximum number of corresponding expired PIT entries divided by the PIT space. $Q(j,q)$ considers only PIT entries that correspond to interest packets monitored first by the MR which sent the report. This way, $Q(j,q)$ represents an upper bound on the PIT entries that can be occupied in any subsequent router by interest packets with name-prefix j.
3) If $Q(j,q)$ exceeds a threshold γ, j is considered *infected*.
4) The DC informs MRs about infected name-prefixes. Each MR, in turn, calculates the expiration rate for each of those name-prefixes on *all* incoming interfaces, and then triggers the reaction function accordingly.

D. Reaction Against Potential IFAs

When triggered, the reaction lasts along the next observation window. The pseudo-code of the reaction function is provided in Algorithm 2. In essence, the PIT expiration rate is used to determine the probability of rejecting (i.e. dropping) incoming interests packets. We adapted this idea, motivated by its simplicity, from *satisfaction-based acceptance* [6]. Our solution, however, is generic, i.e. can incorporate any other reaction strategy. The function uses a uniform probability distribution model, and it is applied in our algorithm on each infected interface per name-prefix (line: 6), while [6] disregards the name-prefix granularity.

Before applying the strategy above, the algorithm first (line: 5) excludes (i.e. directly accepts) interest packets that: (*i*) are checked earlier by another MR, (*ii*) do not belong to an infected name-prefix, or (*iii*) satisfied, i.e. returned a data packet, earlier. These exceptions aim at fulfilling the requirements R3 and R4 (Section IV-A). In particular, by directly accepting previously monitored interest packets (exception (*i*)), duplicate detection and overreactions are avoided. As for exceptions (*ii*) and (*iii*), they are included to avert dropping legitimate interest packets. More precisely, both interest packets that do not belong to infected name-prefixes (exception (*ii*)) as well as those that have been satisfied earlier (exception (*iii*)) should be accepted directly, since they are surely not part of an attack.

Algorithm 2 Reaction against potential IFAs

```
1:  J ← infected name-prefixes observed on f during q
2:  procedure REACTION(f, J)
3:      while receiving interest packets on f do
4:          for each interest packet I do
5:              if Checked = 0 & I ∈ J & I ∉ A then
                        ▷ A: all satisfied interests with prefixes ∈ J
6:                  Drop I with probability P(E(m_f^j, q))
                        ▷ P(a) = a  (∀ a ∈ [0,1])
7:              end if
8:          end for
9:      end while
10: end procedure
```

The storage overhead of the set J (infected name-prefixes observed during the previous observation window) is obviously small. As for the storage overhead of the set A (content names of previously satisfied interest packets), it can be reduced by considering only the names which have been requested during few last observation windows.

VI. Evaluation

Our evaluation consists of two simulation studies: In Section VI-A we evaluate PRCS (our algorithm for placing MRs) with respect to both the fraction of routes covered by monitoring routers and closeness of monitoring routers to clients. Next, we evaluate both the effectiveness and messaging overhead of our defence mechanism in Section VI-B.

In both studies, we fed the simulator with real ISP topologies measured by the Rocketfuel project [19]. At the beginning of each simulation run, the simulator randomly picks $\lceil 70\% \rceil$ of

the nodes as consumer routers, and three of the rest as gateway routers (through which content providers are accessed). We experimented with three topologies, and obtained very similar results with all of them. Due to space constraints, we discuss the results of one topology only: the Exodus ISP (AS 3967) topology, consisting of 79 nodes and 147 bidirectional edges (Fig. 3).

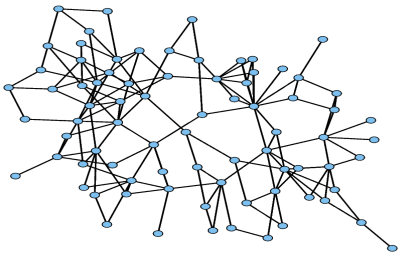

Fig. 3: AS 3967 topology: 79 nodes and 147 bidirectional edges

We repeated each experiment 20 times. In the figures below, we plot the average values with the corresponding 95% confidence intervals.

A. PRCS: Routes Coverage and Closeness to Sources

Evaluation metrics: Following the discussion about placing monitoring routers in Section V-A, we evaluate PRCS with respect to: (*i*) the fraction of covered routes against the number of MRs, and (*ii*) closeness of MRs to consumer routers (i.e. from potential attackers).

Results: Fig. 4 plots the CDF of covered routes as a function of the number of MRs, as ranked by: (*i*) PRCS, (*ii*) the popular betweenness centrality (BC) algorithm, and (*iii*) random placement. It can be seen that with PRCS only 13 MRs, i.e. less than 16% of the routers, are sufficient to cover all the routes. That is to say, with those MRs, it is guaranteed that on each route locates at least one MR. Please note that the aggregate knowledge of packets that can be collected from MRs is equivalent to the aggregate knowledge of all consumer routers. In contrast, the same number of routers as ranked by BC and random placement cover only about 73% and 53% of the routes, respectively. These results confirm the superiority of PRCS.

In Fig. 5, we plot the CDF of the distances between MRs and consumer routers. More precisely, for each route, we measure the number of hops after the consumer router (towards the gateway) till the closest MR. We then normalize this distance by the hop count of the entire route. As shown, roughly 15% of MRs are consumer routers, i.e. the distance to the closest MR is 0. It can be also seen that this distance, for about half of the routes, is less than 0.4 the route length, i.e. locate closer to the consumer router than to the gateway router. For 80% of the routes, this distance increases up to 0.6 the route length.

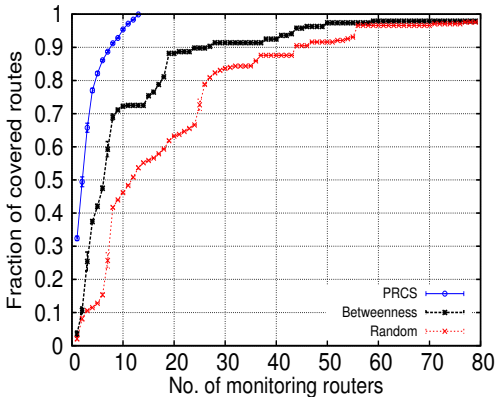

Fig. 4: CDF of covered routes as a function of the number of monitoring routers

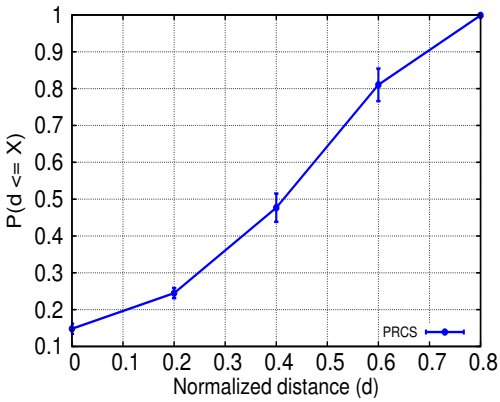

Fig. 5: CDF of hops between consumer router and closest monitoring router (normalised by the route length)

Based on the above results, it is sensible to conclude that PRCS, to a high extent, achieves the goals for which it was developed (Section V-A). That is, it enables to select few routers that *jointly* cover traffic entirely, relatively early.

B. Defense Against IFAs: Effectiveness and Feasibility

Setup: We implemented IFA and our solution in ndnSIM [20]. At the beginning of each simulation run, top $\lceil 10\% \rceil$ PRCS-ranked nodes are selected as MRs. Also, 25% of the clients are selected randomly as attackers. Each run lasts for nine minutes: the attack starts at the beginning of minute 2 (second 61) and stops at end of minute 6 (second 360).

We experimented with a uniform PIT capacity of 5000 entries in each router. Each legitimate client requests existing contents with a rate of 100 interest packets per second (ipps). This rate keeps the average PIT utilization low and does not cause PIT overflow as long as there is no attack ongoing. In contrast, each attacker requests non-existent contents at higher rates; in particular, we experimented with three attack rates: $\{500, 1000, 10000\}$ ipps. We configured the observation window of MRs to 10 seconds, which equals five times the default PIT's expiration time in ndnSIM. As for the thresholds τ and γ, we set them to 0.3 and 0.5, respectively. We also used a uniform data packet's size of 1100 bytes, and disabled content caching.

Evaluation metrics: We evaluate the effectiveness of our mechanism against IFAs by two metrics: (*i*) the *satisfaction ratio of legitimate interest packets*, and (*ii*) *domain-wide PIT usage* measured as a ratio of the overall PIT space. The first parameter is meant to evaluate the quality of service received by legitimate clients during the attack period, and it has been widely used in the related work. As for the second parameter, we use it to evaluate the protection which can be provided by our defence mechanism for PITs, the direct target of IFAs.

Regarding the *messaging overhead* of our defence mechanism, we measure it by normalizing the number of bytes used by defence-related messages (Fig. 2) over the number of bytes used by regular data packets.

Results: We measure the satisfaction ratio of legitimate interest packets that can be achieved when enabling our defence mechanism. We also compare these results to: (*i*) a system without defence, (*ii*) a system incorporating satisfaction-based acceptance (SBA), and (*iii*) a system incorporating satisfaction-based pushback (SBP). As discussed in Section III, both SBA and SBP were proposed and evaluated in [6]. The first is very lightweight and simple but not highly effective, while the second has been shown to be effective.

Fig. 6 – Fig. 8 plot the results for the aforementioned three attack rates, respectively. We can see that the satisfaction ratio of legitimate interest packets during the attack period improves significantly with our mechanism: from about 60% to about 98% under attack rate of 500 ipps, from about 48% to above 90% under attack rate of 1000 ipps, and from about 22% to about 66% under attack rate of 10000 ipps. It is also important to note that our mechanism do not cause packet drops, i.e. enables for full satisfaction, when no attack exists.

Fig. 6 – Fig. 8 also show that our mechanism outperforms both SBA and SBP, remarkably. The utility of SBA almost disappears under high attack rates, and the same holds for SBP under massive attack rates. Please note that the impact of SBA even becomes negative under massive attack rates (the satisfaction ratio is lower than when there is no defence enabled at all).

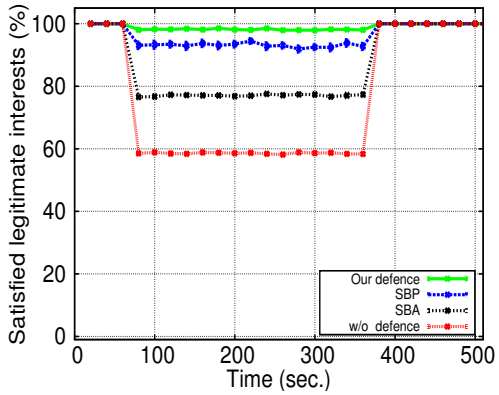

Fig. 6: Satisfaction ratio of legitimate interests under attack rate of 500 ipps (attack period: sec. 61 – sec. 360)

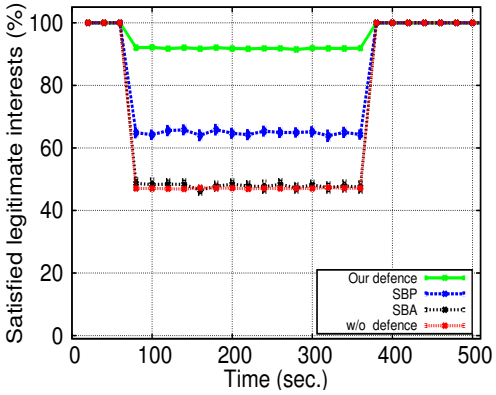

Fig. 7: Satisfaction ratio of legitimate interests under attack rate of 1000 ipps (attack period: sec. 61 – sec. 360)

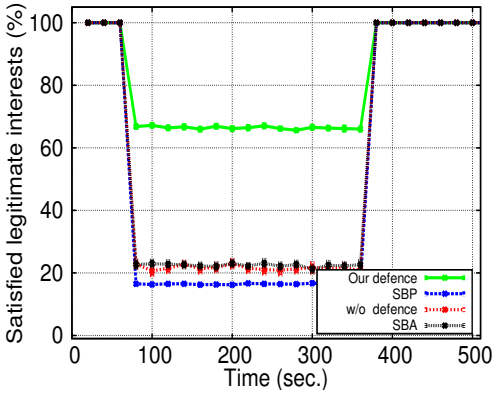

Fig. 8: Satisfaction ratio of legitimate interests under attack rate of 10000 ipps (attack period: sec. 61 – sec. 360)

The effectiveness of our mechanism against IFAs is also confirmed by the results of the second metric: Fig. 9 shows that our defence mechanism lowers the *global* PIT usage during the attack period remarkably.

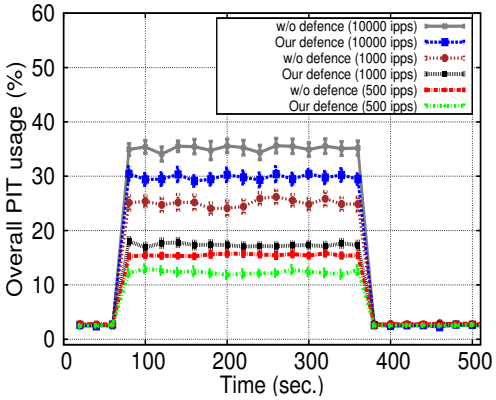

Fig. 9: Global PIT usage under three attack rates (attack period: sec. 61 – sec. 360)

In particular, the global PIT usage is reduced from about 16% to about 12% under attack rate of 500 ipps, from about 25% to about 17% under attack rate of 1000, and from about 35% to about 30% under attack rate of 10000 ipps. Please note that this reduction is measured over *all* PITs. The reduction

values per router differ by the amount of malicious interest packets each router receives (increases for routers which locate closer to targeted content providers).

As for the messaging overhead of the proposed mechanism (Fig. 10), we can see that it is very marginal both when no attack exists as well as during the attack period. We can also see that the overhead increases with the attack rate due to increase of the number of defence-related messages exchanged between the DC and MRs.

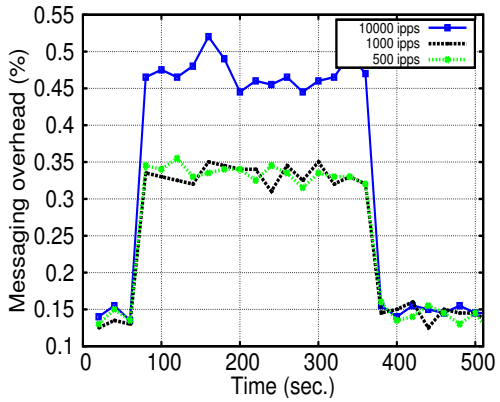

Fig. 10: Messaging overhead under three attack rates (attack period: sec. 61 – sec. 360)

All over all, the results above suggest that our defence mechanism is both effective and feasible. With a relatively high attack rate (25% of the nodes issue malicious interest packets that can fill PITs very fast), above 90% of legitimate interest packets are satisfied, with a negligible messaging overhead. Increasing that attack rate 10 times (10000 ipps) lowers the satisfaction ratio to about 65%. Nevertheless, with this result our defence mechanism still significantly outperforms a state-of-the-art solution which was considered to be effective. Last but not least, the effectiveness of our mechanism against massive attack rates can be improved by involving consumer routers in the reaction against potential IFAs (possibly commanded by the DC). We leave the design details and evaluation of this idea for future work.

VII. Summary

We presented a defence mechanism for the so-called Interest Flooding Attack in NDN. Attack detection is basically performed by a small number of routers, from which a network-wide knowledge of traffic and forwarding states can be acquired, with the aid of a centralized controller. Reactions against potential attacks are also assigned to those routers.

The proposed mechanism fulfils the preset design requirements (Section IV-A): Table I summarizes which design element or feature enables to fulfil which requirement.

Through extensive simulations, we have shown that the proposed mechanism is highly effective against IFAs and incurs only low communication overhead.

Our agenda for future work include designing and implementing the domain controller in a distributed way for load balancing and fault tolerance. In addition, we plan to extend our solution in two directions: performing attacks' detection and mitigation over multiple domains, and adapting it for other NDN-tailored attacks, e.g. pollution of cache stores.

TABLE I: Mapping design elements and features to the requirements

	R1	R2	R3	R4	R5
Full coverage: MAR & FTBM	✓				
Using knowledge of DC	✓				
Preferring MRs near clients		✓			
Only first MR reacts		✓	✓		
Each packet is checked once			✓		
Detection per name-prefix				✓	
Accept satisfied interests				✓	
Relying on few routers					✓
No explicit coordination					✓

Acknowledgement

This work was supported by the Federal Ministry of Education and Research (BMBF), Germany, under the "An Optic's Life" project (no. 16KIS0025). We thank the authors of [6] for sharing their implementation with us. We also thank the anonymous reviewers for their valuable comments.

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2013–2018," *CISCO White paper*, 2014.
[2] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Communication Magazine*, 2013.
[3] V. Jacobson *et al.*, "Networking named content," in *CoNEXT*, 2009.
[4] P. Gasti *et al.*, "DoS and DDoS in Named Data Networking," in *IEEE ICCCN*, 2013.
[5] A. Compagno *et al.*, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking," in *IEEE LCN*, 2013.
[6] A. Afanasyev *et al.*, "Interest flooding attack and countermeasures in Named Data Networking," in *IEEE IFIP Networking*, 2013.
[7] D. Goergen *et al.*, "Security monitoring for content-centric networking," in *Springer data privacy management and autonomous spontaneous security*, 2013.
[8] H. Dai *et al.*, "Mitigate DDoS Attacks in NDN by Interest Traceback," in *INFOCOM Workshops*, 2013.
[9] H. Salah and T. Strufe, "CoMon: An Architecture for Coordinated Caching and Cache-Aware Routing in CCN," in *IEEE CCNC*, 2015.
[10] J. Hawkinson and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS) - RFC 1930," 1996.
[11] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A Survey of Security Attacks in Information-Centric Networking," *IEEE Communications Surveys & Tutorials (accepted for publication)*.
[12] T. Lauinger, "Security & scalability of content-centric networking," *Master thesis, TU Darmstadt and Eurecom*, 2010.
[13] I. Widjaja, "Towards a flexible resource management system for content centric networking," in *IEEE ICC*, 2012.
[14] K. Wang *et al.*, "Cooperative-Filter: countering Interest flooding attacks in named data networking," *Springer Soft Computing*, 2014.
[15] H. Salah, J. Wulfheide, and T. Strufe, "Lightweight Coordinated Defence Against Interest Flooding Attacks in NDN - (poster paper)," in *INFOCOM WKSHPS*, 2015.
[16] K. Suh *et al.*, "Locating network monitors: complexity, heuristics, and coverage," *Elsevier Computer Communications*, 2006.
[17] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Informs' Management Science*, 1971.
[18] D. T. Ha *et al.*, "On the effectiveness of structural detection and defense against P2P-based botnets," in *IEEE/IFIP DSN*, 2009.
[19] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *SIGCOMM*, 2002.
[20] A. Afanasyev *et al.*, "ndnSIM: NDN simulator for NS-3," *University of California, Los Angeles, Tech. Rep*, 2012.