


RESEARCH

Open Access



# A network-wide view-based detection and mitigation of a sophisticated Interest Flooding Attack

Guang Cheng<sup>1,2,3\*</sup> , Lixia Zhao<sup>1,2,3</sup>, Xiaoyan Hu<sup>1,2,3</sup>, Shaoqi Zheng<sup>1,2,3</sup>, Hua Wu<sup>1,2,3</sup> and Chengyu Fan<sup>4</sup>

\*Correspondence:

[gcheng@njnet.edu.cn](mailto:gcheng@njnet.edu.cn)

An earlier version of the paper [1] was presented by the 1st IEEE International Workshop on Network Meets Intelligent Computations, NMIC@ICDCS. This version has been extended, which added more descriptions of our proposed mechanism DMNWW and some new experiments were done to make our contributions clearer.

<sup>1</sup>School of Cyber Science & Engineering, Southeast University, Nanjing, China

<sup>2</sup>Key Laboratory of Computer Network and Information Integration of Ministry of Education of China, Southeast University, Nanjing, China

Full list of author information is available at the end of the article

## Abstract

Interest Flooding Attack (IFA) is one of the main security threats for the Named Data Networking (NDN). Most of its existing countermeasures enable intermediate routers near the attackers to independently detect the attack and consider the typical attack scenario in which attackers directly send malicious Interests at a constant and relatively high rate after the attack starts. Moreover, they may also throttle legitimate Interests when enforcing the existing defense measures at intermediate routers as it is still difficult for them to distinguish the Interests issued by attackers from those issued by legitimate consumers. Instead, this work aims at a more sophisticated attack scenario in which attackers start the attack at a relatively lower rate but gradually speed up to keep the Pending Interest Tables (PITs) of the victims increasing to finally deplete the PIT resources for legitimate consumers. It is relatively difficult for intermediate routers to independently and timely detect such a sophisticated IFA. To solve this problem, we propose a mechanism to detect and mitigate the sophisticated IFA from the network-wide view, dubbed as DMNWW. In DMNWW, a central controller monitors the network and makes a comprehensive and prompt decision on whether there is an ongoing IFA based on the overall state of the whole network collected from the abnormality information reports sent by the first-hop routers of attackers. Attack sources can be directly located after an IFA is determined, and then the routers directly connected to attackers (i.e., access routers) can take targeted measures based on the located attackers to prevent malicious Interest from entering the network without throttling legitimate Interests. We conduct an experimental study to evaluate the performance of DMNWW, explore the parameter settings of the attack detection algorithm at access routers, and measure the communication overhead of the central controller. The experimental results validate that DMNWW can timely detect and mitigate the sophisticated IFA without throttling requests from legitimate consumers with significantly low communication overhead of the central controller, which will not bring about too much burden to the network.

**Keywords:** Network-wide view, Interest Flooding Attack, Named Data Networking

## 1 Introduction

With the rapid development of network technology and the continuous growth of new types of applications, the communication paradigm of the Internet has gradually transformed from the resource sharing between hosts to the content distribution and retrieval. The traditional host-centric IP networking was designed to connect static end-hosts, which cannot support the requirements of today's applications, such as mobility, efficient content distribution, and others. Hence, researchers have proposed a series of possible next-generation Internet architectures [2].

Named Data Networking (NDN) [3] is one of the most promising future Internet architectures. NDN names the content in the network and transforms the first entity of the network from hosts to named content. NDN communication is driven by a consumer issuing an Interest packet which specifies the name of the desired content segment. Then, intermediate nodes route the Interest by the content name and the matching Data packet with the desired content segment returns along the reverse path of the Interest. NDN supports stateful forwarding. Each NDN router should maintain the state information of each forwarded but not yet satisfied Interest in its Pending Interest Table (PIT). A PIT entry will not be removed unless the corresponding Data packet for the recorded Interest returns or its lifetime expires. This feature brings many advantages to NDN [4]. However, it can also be exploited by attackers to launch a kind of NDN-specific DDoS attack—Interest Flooding Attack (IFA). IFA attackers usually send a great number of spoofed Interests for non-existent content to exhaust routers' PIT resources to make them unable to create new PIT entries for subsequent incoming Interests. Therefore, requests from legitimate consumers will be discarded [5].

The existing mechanisms against IFA [6–12] have one or more of the following features. First, the existing mechanisms mainly enable intermediate routers near the attackers to independently detect and mitigate the attack and focus on the typical IFA scenario in which attackers directly send malicious Interests at a constant and relatively high rate. They may suffer performance degradation to a certain extent when a more sophisticated IFA is launched as an independent decision on attack detection and mitigation may lead to relatively high detection latency, poor sensitivity to low intensity attack, and overreaction. Second, the requests from legitimate consumers may also be throttled as the existing mitigation methods cannot accurately distinguish requests issued by attackers from those issued by legitimate consumers. Third, it is difficult for most existing mechanisms to trace back to attackers since an Interest contains no information about its issuer.

Instead, this work focuses on the more sophisticated IFA scenario proposed in our previous work [13], i.e., attackers start the attack at a relatively lower rate but speed up step by step to keep the PITs of the victims increasing to exhaust their PIT resources, and the changes of router statistics between any two consecutive time intervals during the attack are much more slightly, which is relatively difficult to be timely detected by the existing countermeasures. We propose a mechanism with a central controller to detect and mitigate such sophisticated IFA from the network-wide view, which is referred to as DMNWV. In DMNWV, each access router (i.e., the router directly connected to consumers/attackers) in the network is responsible for detecting the state of its each interface. When an access router finds there is something abnormal on its certain interfaces

but is unsure whether there is an IFA, it will notify the controller and report its abnormal observations according to the controller's requests. Attack detection at access routers can make it easier to locate attackers after an IFA is determined. The controller collects the abnormal information detected by all the access routers that have found something abnormal and detects the attack from the network-wide view based on the overall state of the whole network, aiming to timely detect the attack before the network suffers severe damage. When the controller determines that there is actually an ongoing IFA, it will further locate the attackers and then inform the access routers under attack of their malicious interfaces. Afterwards, access routers can take targeted countermeasures on the identified attackers at source according to the feedback from the controller, which can avoid throttling requests from legitimate consumers.

The remainder of this paper is organized as follows. We analyze the state-of-the-art mechanisms against IFA in Section 2. Section 3 presents the overall framework and design specifications of DMNWV. We conduct an experimental study on DMNWV to evaluate its performance, explore the parameter settings of the attack detection algorithm at access routers, and measure the communication overhead of the controller in Section 4. Finally, we conclude the paper in Section 5.

## 2 Related work

Most existing mechanisms against IFA focus on the typical IFA scenario in which attackers directly send malicious Interests at a constant rate, especially at a relatively high rate. The IFA detection and identification of malicious prefix or/and interfaces in these mechanisms are mainly based on PIT-related statistics, such as the satisfaction ratio of Interests and PIT usage. Afanasyev et al. [6] presented three countermeasures to limit the number of Interests forwarded in the network based on NDN's inherent properties of storing per packet state on each router and maintaining flow balance. Dai et al. [7] proposed *Interest traceback* to trace back to the originators of malicious Interests after detecting an IFA. It detects the attack only based on routers' PIT sizes, which may misjudge small bursts of Interests as IFAs. Vassilakis et al. [9] proposed a mitigation mechanism that allows routers to quickly identify and block attackers by detecting abnormal user behavior. Compagno et al. [8] proposed *Poseidon*, in which a router determines an IFA when both the unsatisfaction ratio and PIT usage of Interests from a certain interface exceed their thresholds respectively. Afterwards, the router will limit the rate of incoming Interests from its malicious interfaces and issue a push-back "alert" message to the node connected to the offending interface. However, the collaboration between routers in *Poseidon* appears only during the mitigation phase. Salah et al. [14, 15] adopted a new framework to assign a predetermined set of routers as monitoring routers which will detect and mitigate an IFA with the help of a central controller. This framework can work efficiently when the network is static. However, the network state is always changing in the real world, such as the distribution of clients and the connections between different nodes, but the monitoring routers in this framework are predetermined.

The mechanisms based on PIT-related statistics may cause misjudgment. For example, the prefix hijacking attack can also lead to a high PIT expiration ratio, which may be mistakenly classified as an IFA. Xin et al. [16] proposed to detect an IFA based on cumulative entropy by monitoring the content request abnormal distribution and then introduced the malicious prefix identification method by relative entropy

theory. Zhi et al. [17] proposed a Gini impurity-based IFA detection mechanism using the statistical properties of the name field in the Interests to detect and mitigate IFAs. These two mechanisms above can quickly detect an IFA and avoid certain misjudgment.

Most existing solutions distinguish Interests issued by attackers from those issued by legitimate consumers by identifying the malicious prefix or/and interfaces. The most frequently used mitigation method against IFA is limiting the rate of incoming Interests from the malicious interfaces or under the malicious prefix or filtering out all the relevant Interests, which can obviously reduce the number of malicious Interests in the network. However, such method may also mistakenly drop requests from legitimate consumers, since Interests coming from the identified malicious interfaces or under the malicious prefix can also be issued by legitimate consumers. Ding et al. [10] presented a retransmission forwarding mechanism to ensure legitimate consumers' requests when defending against an IFA. Wang et al. [18] proposed an approach called *Disabling PIT Exhaustion (DPE)* to decouple all the malicious Interests from PIT, by directly recording their state information (e.g., incoming interface) in the name of each malicious Interest rather than PIT. The authors also introduced a packet marking scheme to enable Data packet forwarding without the help of PIT. These two solutions can ensure that requests from legitimate consumers under malicious prefix or from malicious interfaces can still be satisfied when defense measures are taken after an IFA is detected. However, the complex processing operations should be performed on all the potential malicious Interests, such as changing Interests' names, which will bring about heavy burden to the network due to the large scale of malicious Interests.

Most existing mechanisms against IFA enable intermediate routers near the attackers to independently detect the attack and mainly focus on the typical IFA scenario, in which distributed IFA attackers directly issue spoofed Interests for non-existent content at a constant and fairly high rate after the attack starts. In this scenario, the states of victims, such as the PIT usage and the satisfaction ratio of Interests, will immediately have significant changes after the attack starts, so that an intermediate router can independently and quickly find something abnormal and detect the attack based on its local observations. However, carefully crafted attackers may manage to launch a more sophisticated attack to keep the changes of router statistics much slighter, making it harder for a single intermediate router to timely detect by itself. In our previous work [13], we proposed a more sophisticated IFA scenario in which carefully crafted attackers issue malicious Interests at a relatively lower initial rate at the beginning of the attack. In this way, attackers can make sure that the PIT increasing rate is below a certain level. Afterwards, the attackers speed up step by step to keep the router statistics changing slightly so that it is relatively difficult for an intermediate router near the attackers to timely detect the attack by itself due to the inconspicuous changes to its state information between two consecutive intervals. Finally, the PIT resources of victim routers will be exhausted.

This work aims at the more sophisticated IFA scenario introduced above. We propose DMNWV against such sophisticated IFA with a central controller monitoring the network from the network-wide view, aiming to timely detect the attack at an early stage and then locate the attackers to mitigate the attack at source to avoid throttling legitimate consumers' requests.

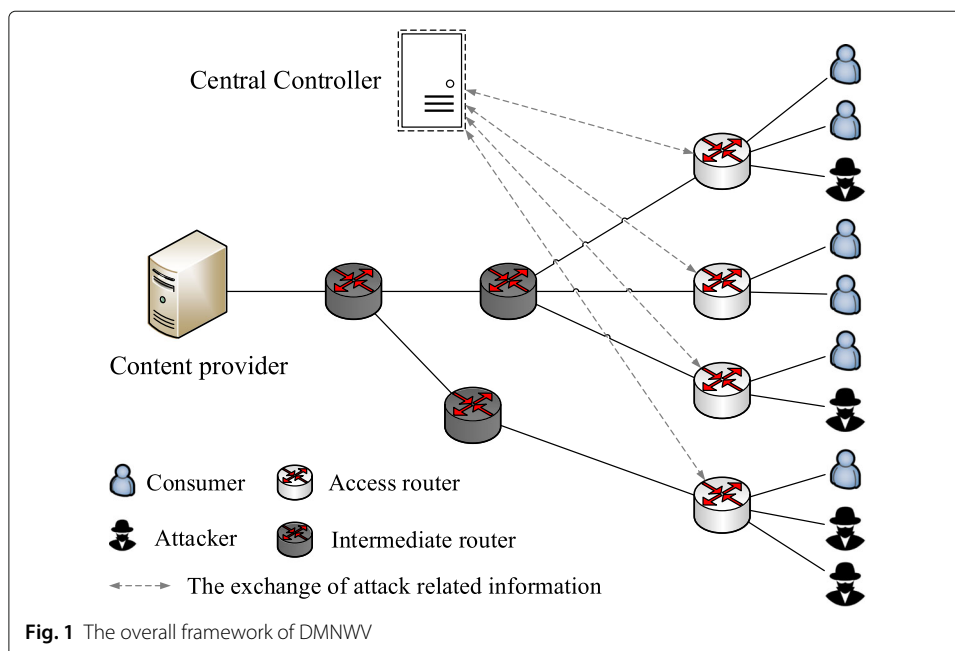
### 3 DMNWV method

In this section, we introduce the overall framework and design specifications of attack detection and mitigation of our proposed DMNWV method.

#### 3.1 Overall framework

In DMNWV, there is a central controller which monitors the network from the network-wide view, aiming to timely detect the sophisticated IFA and then locate the attack sources to take targeted defense measures to avoid throttling requests from legitimate consumers. The overall framework of DMNWV is shown as Fig. 1. The routers directly connected to consumers/attackers are referred to as access routers, and the rest routers are referred to as intermediate routers. The dotted lines represent the interaction of attack-related information between the central controller and all the access routers in the topology.

All the access routers in the network are responsible for monitoring their real-time states and detect whether there is something abnormal, such as low satisfaction ratio of received Interests, abnormal distribution of received requests, and excessive speed of incoming Interests. If the access router can independently determine that there is an IFA, it can immediately take defense measures. However, access routers are so close to attackers that the malicious traffic that a single access router can monitor is very small compared to upstream routers close to the content provider. It is very difficult for a single access router to make an independent decision on whether there is an IFA only based on its limited local observations. In DMNWV, we still select access routers to be the monitoring routers since in this way, it is easier to locate attack sources and take defense measures. If the access router finds an abnormality but is unsure whether there is an ongoing IFA only based on its local observations, it can notify the central controller that there exists an abnormality and then wait for the decision from the controller. The access router can report its detailed observations about the detected abnormality to the controller based on the demand of the controller. The monitored and reported information by all access routers



jointly contribute to the network-wide decision on the IFA of the controller. And each access router is also required to take corresponding measures based on the final decision of the controller.

The central controller constantly monitors the network from the network-wide view and is logically connected to all the routers in the network. The controller can collect all the abnormal information reports after receiving attack-related notifications from access routers which have found something abnormal and can also proactively request any information as required. According to all the information reported by access routers which find something abnormal, the controller can learn more about the abnormal traffic in the network and make a comprehensive decision on an IFA more accurately and timely based on the overall state of the whole network compared to IFA detection methods by a single router only based on its local limited observations. If an IFA is determined, the controller will further locate attackers and then notify relevant access routers of its final decision about the attack to make them take defense measures against the IFA.

### **3.2 Attack detection**

The attack detection in DMNWV is comprised of two parts, local attack detection at access routers and network-wide attack detection at the central controller.

In most existing mechanisms against IFA, the attack detection is required to be performed on each router in the network, which will bring about heavy burden to the network. Moreover, it is difficult to locate attackers after an IFA is determined since an Interest contains no information about its issuer to protect users privacy. Obviously, the most efficient way to trace back to attack sources is making good use of access routers that attackers are directly connected to. Therefore, in DMNWV, all the access routers in the network are selected as monitoring routers, which will periodically detect whether there is something abnormal on each of their interfaces.

#### **3.2.1 Local attack detection at access routers**

Most existing mechanisms against IFA are mainly based on PIT-related statistics, such as PIT size (the number of PIT entries), PIT expiration ratio, and the unsatisfaction ratio of incoming Interests. Such mechanisms may cause misjudgement. For example, the prefix hijacking attack can also lead to high PIT expiration ratio, which may be misjudged as an IFA. Moreover, the decision on an IFA may be delayed to some extent in PIT-based mechanisms since they need to wait the malicious Interests to time out, and thus, the adopted PIT-related measurements exceed the set thresholds, respectively.

Therefore, we choose the speed of incoming Interests as the main measurement to detect an IFA at the access routers rather than PIT-related statistics to make the attack detection more timely and accurately. The reason is that in order to launch an effective IFA, attackers should issue a large number of spoofed Interests to make sure that the speed at which a victim adds entries to its PIT is higher than that it removes, so that PIT resources of the victim can eventually be exhausted. For an access router, it is obvious that the speed of incoming Interests on a malicious interface connected to an attacker is certainly different from that on a legitimate interface connected to a legitimate consumer, so the speed of incoming Interests can be used as a measurement to detect an IFA by access routers.

**Algorithm 1** Process of attack detection and mitigation at an access router

---

**Require:** The average speed of incoming Interests  $\bar{v}_i$ , the number of expired Interests  $e_i$  of each interface  $face_i$  on an access router  $R_x$

- 1: **for** each interface  $i$  **do**
- 2:    $Y_n \leftarrow CUSUM(\bar{v}_i)$
- 3:   **if**  $Y_n \geq T_{suspicious}$  AND  $e_i \geq T_{timeout}$  **then**
- 4:     Mark  $face_i$  as suspicious
- 5:     **if**  $face_i$  is the first abnormal interface detected by  $R_x$  **then**
- 6:       Send specific Interest to notify the controller
- 7:     **end if**
- 8:   **end if**
- 9: **end for**
- 10: **if**  $R_x$  receives an attack-related Interest from the controller **then**
- 11:   Validate the Interest
- 12:   Get the command type according to the Interest name
- 13:   **if** the Interest is to request the detected abnormal information by  $R_x$  **then**
- 14:     Reply with the latest detected abnormal information of all interfaces that are marked as suspicious
- 15:   **else**
- 16:     **if** the Interest is to notify that an IFA exists on  $R_x$  **then**
- 17:       Get the malicious interfaces list based on Interest name
- 18:       Mark all interfaces in the malicious interfaces list as malicious and block them
- 19:     **end if**
- 20:   **else**
- 21:     **if** the Interest is to notify that no IFA exists on  $R_x$  **then**
- 22:       Mark all interfaces as normal
- 23:     **end if**
- 24:   **end if**

---

The non-parametric cumulative sum (CUSUM) [19] is one of the change point detection algorithms and is widely used to detect abnormalities, which can accumulate small changes of the applied measurement to achieve amplification effect. So we apply the CUSUM algorithm to the speed of incoming Interests on each interface of each access router to detect an abnormality more timely and accurately.

We define the sequence  $\{X_n\}$  representing the average speed of incoming Interests on an interface of an access router in a series of continuous time window  $\Delta t$ . In normal conditions, the speed at which legitimate consumers send Interests to request their desired content is steady and fluctuates within a normal range. We assume that the upper bound of the average speed at which legitimate consumers send Interests is  $\beta$  and  $\beta = (\alpha + 1)\bar{v}$ , where  $\bar{v}$  is the mean value of the speed of Interests from legitimate consumers observed in normal traffic conditions (where there is no attack or network congestion) and  $\alpha$  is a constant greater than 0 that indicates the percentage above the mean value that is considered an indication of abnormal behavior. It is necessary to transfer  $\{X_n\}$  into a new sequence

$\{Z_n\}$  by  $Z_n = X_n - \beta$ , which must be negative during normal conditions. Further, we can define another sequence  $\{Y_n\}$  as follows:

$$\begin{cases} Y_n = (Y_{n-1} + Z_n)^+, & n > 0 \\ Y_0 = 0, & n = 0 \end{cases} \quad (1)$$

where

$$x^+ = \begin{cases} x, & x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

When the observed average speed of incoming Interests on an interface of an access router is larger than  $\beta$ ,  $Y_n$  becomes larger than 0 and will keep accumulating if the speed of incoming Interests still keeps high. A large value  $Y_n$  indicates that there may be an ongoing IFA on the monitored interface. A threshold  $T_{\text{suspicious}}$  is set for  $Y_n$ .

Moreover, a burst of Interests, where many legitimate consumers send Interests to request their desired content at a relatively high speed at the same time, is very similar to an IFA and may also lead to a large  $Y_n$  value. But a significant difference is that bursts of Interests sent by consumers are all legitimate and satisfiable, and the proportion of expired Interests is extremely limited. While in an IFA, all malicious Interests sent by attackers request for non-existent content and will certainly not be satisfied and then all time out after their lifetime expires. Based on the above difference, an access router will also record the number of expired Interests on each of its interfaces during each time window, which can be a reference to the access router after a large  $Y_n$  value is detected to avoid misjudgement. And another threshold  $T_{\text{timeout}}$  is set.

The process of attack detection and mitigation at an access router is displayed in Algorithm 1. For each interface of an access router, when  $Y_n$  exceeds  $T_{\text{suspicious}}$  and the number of expired Interests exceeds  $T_{\text{timeout}}$ , the access router will judge that this interface may be under an IFA and then send an Interest with specific name */ndn/ddos/flooding/controller/routerId/abnormityNotification* to notify the central controller that something abnormal has been found, where the name component *routerId* refers to the identifier of the access router so that the controller can learn which access router the notification is sent by. Then, the access router will report its latest abnormal observations according to the subsequently received requests from the controller. The reported observations mainly include the time when the report is produced and the collection of the information of incoming Interests on each suspicious interface at the access router. The information of incoming Interests on each suspicious interface includes the identifier of the interface, the prefixes and corresponding average speed of incoming Interests under each prefix.

### 3.2.2 Network-wide attack detection at the central controller

The process of attack detection and mitigation at the central controller is displayed in Algorithm 2. As soon as receiving the attack-related notification from an access router, the controller replies with a Data packet expressing that it has already received the notification and begins to periodically send Interests with specific name */ndn/ddos/flooding/routerId/report/reportSeq* to request the latest observations at the access router whose identifier is *routerId*.

Based on all the already received abnormal observations, the controller makes a comprehensive decision on whether there is an ongoing IFA. As the central controller



monitors the network from the network-wide view, it can observe the overall topology of the network. Moreover, each link in the network has its capacity limit. We express such limit as the number of forwarded Interests out of each interface based on the physical capacity of the corresponding interface (i.e., pending *Interest Limit*) as [6], which will be proportional to the link's bandwidth-delay product (BDP) [20]. The value of Interest limit can be formalized as follows:

$$\text{Interest Limit} = \text{Delay}[s] \cdot \frac{\text{Bandwidth [Bytes/s]}}{\text{Data packet size [Bytes]}} \quad (3)$$

where *Delay* is the expected time for the Interest to be satisfied and *Data packet size* is the size of the returning Data packet.

---

**Algorithm 2** Process of attack detection and mitigation at the central controller

---

**Require:** the factor  $\theta (0 < \theta \leq 1)$

- 1: **if** the controller receives an attack notification Interest from an access router **then**
  - 2:   Validate the notification Interest  
       Get the access router's identity according to the Interest name  
       Begin to periodically send Interests to request the abnormal information detected by the access router
  - 3: **end if**
  - 4: **for** each received Data packet carrying the requested abnormal information from an access router **do**
  - 5:   Parse the Data payload to get the detailed abnormal information
  - 6:   **for** each prefix-speed pair in each interface in the report **do**
  - 7:     Find out the corresponding producer of the prefix  
       Calculate the path from the access router to the producer  
       Add the volume of reported abnormal Interests to the statistics of each link on the path
  - 8:   **end for**
  - 9: **end for**
  - 10: **if** there is one or more link under attack **then**
  - 11:   Determine that there is an IFA
  - 12:   **for** each link under attack **do**
  - 13:     Find out the sources of suspicious Interests on that link, i.e., access routers and their malicious interfaces respectively
  - 14:   **end for**
  - 15:   Send Interests to notify relevant access routers of their malicious interfaces respectively
  - 16: **else**
  - 17:   **if** there is always no link determined as under attack in a certain period of time **then**
  - 18:     Determine that there is no IFA  
       Send Interests to notify relevant access routers of no attack
  - 19:   **end if**
  - 20: **end if**
-

For each report, the controller finds out the corresponding content provider of the reported suspicious Interests and then calculates the paths that suspicious Interests traverse from the access router to the content provider. Afterwards, the controller will further calculate the total number of suspicious Interests transmitted on each link. If the controller finds that there is one or more links on which the number of data requested is going to reach the corresponding link's capacity limit (i.e., *the number of requested data*  $\geq \theta \cdot$  *Interest limit*, where  $\theta$  is a constant and  $0 < \theta \leq 1$ ), it determines that there is an ongoing IFA in the network and then finds out the sources of suspicious Interests on that link, i.e., access routers whose reported suspicious Interests pass through that link and through which interfaces these Interests enter the network (i.e., their corresponding malicious interfaces). Otherwise, if there is always no such link in a certain period of time, the controller determines that there is no an ongoing IFA. Afterwards, the controller notifies relevant access routers of its decision. If an IFA is determined, the controller sends an Interest with the specific name */ndn/ddos/flooding/routerId/attackACK/MaliciousInterfacesList* to notify the access router whose identifier is *routerId* that its interfaces listed in *MaliciousInterfacesList* are malicious. Otherwise, an Interest with name */ndn/ddos/flooding/routerId/noAttack* is issued by the controller to notify the access router whose identifier is *routerId* that it is not under an IFA. After receiving the feedback from an access router (i.e., a Data packet) expressing that it has already received the notification and taken actions according to the controller's decision, the controller will stop requesting the access router's observations.

Note that all the attack-related Interests exchanged between the controller and access routers are signed to avoid bringing new security issues to NDN.

### 3.3 Attack mitigation

After an IFA is detected, most existing mechanisms mitigate the attack by limiting the rate of incoming Interests from the identified malicious interfaces or under the malicious prefix. This method can significantly reduce the number of malicious Interests forwarded in the network, but requests from legitimate consumers may also be mistakenly throttled. Since the Interests from malicious interfaces or under malicious prefix may also be issued legitimate consumers and it is difficult for intermediate routers to accurately distinguish Interests issued by attackers from those issued by legitimate consumers only based on the identified malicious prefix or interfaces.

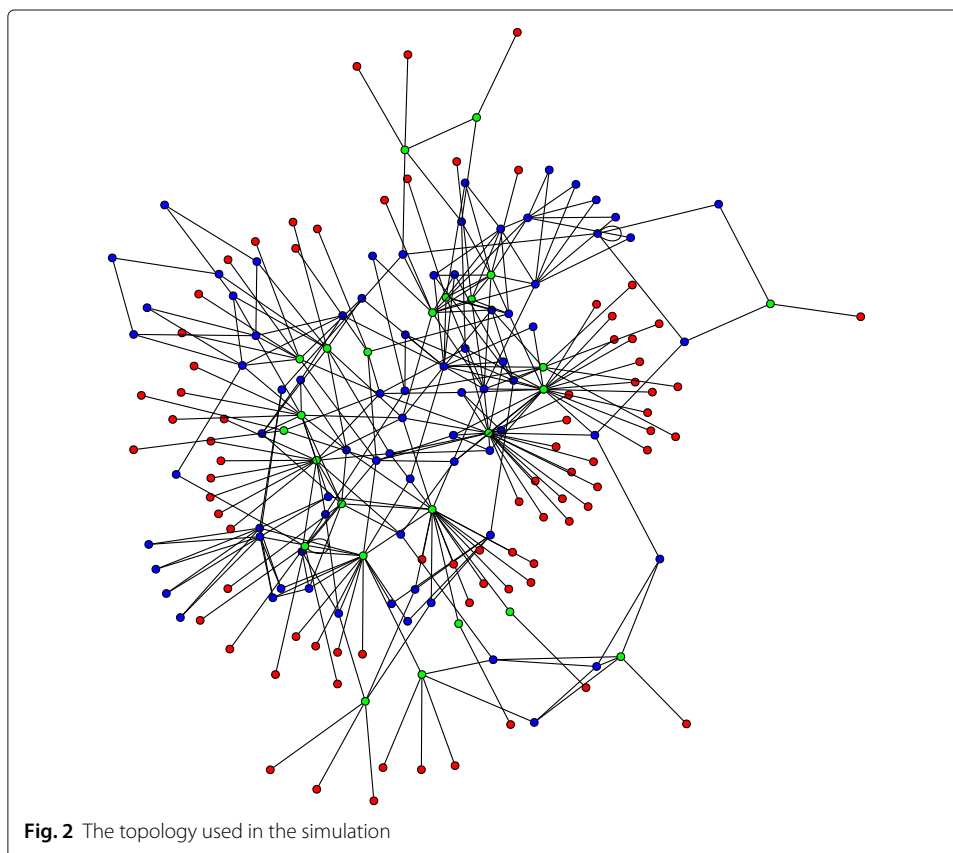
In view of the problem above, the attack mitigation in DMNWV is performed at access routers that attackers are directly connected to and through which malicious Interests enter the network. As soon as receiving the attack-related notification from the controller expressing that there is an IFA on its certain interfaces, an access router will immediately block the nodes directly connected to its malicious interfaces determined by the controller, i.e., dropping all the incoming Interests from its malicious interfaces. Mitigating an IFA at source can directly prevent malicious Interests from entering the network and avoid throttling the requests from legitimate consumers, since for an access router, the node directly connected to a malicious interface must be an attacker and Interests from the malicious interfaces are all issued by attackers.

#### 4 Performance evaluation and result discussion

In this section, we present the experimental studies on DMNWV. We evaluate the performance of DMNWV, explore the parameter settings of the attack detection algorithm at access routers, and measure the communication overhead of the controller.

We use the open-source ndnSIM [21], a NS-3 based NDN simulator, to run our simulations. The topology we used is based on a modified version of Rocketfuel's AT&T topology [22], which is shown as Fig. 2. The topology consists of 182 nodes, including 80 leaf nodes shown as red (i.e., consumers), 25 gateway nodes shown as green (i.e., access routers, which are directly connected to consumers), and 77 intermediate nodes shown as blue (i.e., intermediate routers, which are directly connected to other routers). Moreover, we additionally create a new node serving as the central controller, which is connected to a randomly selected intermediate router. The central controller will not participate in the routing of packets between consumers and producers.

In our experiments, 40% of the consumers are randomly selected as attackers, and we randomly pick either an intermediate node or a gateway node as the content provider. Before the attack starts, attackers do as what legitimate consumers do, i.e., send satisfiable Interests at the same speed as legitimate consumers. The initial attack speed of attackers is 1/3 of the speed of Interests from legitimate consumers. Each simulation is repeated for 10 runs to randomize the results to get an average result. The detailed parameter settings are shown in Table 1.



**Table 1** Parameter setting

Parameter	Default value
Maximum PIT size	2000 PIT entries
The lifetime of Interests	1 s
Size of each content item	1100 bytes
Forwarding strategy	BestRoute
Rate of legitimate consumers	40 Interests per second
Delay in (3)	300 ms
Simulation time	300 s
Duration of attack	60–240 s
Time window ( $\Delta t$ )	1 s
$T_{\text{suspicious}}$	$3 \cdot \bar{v}$
$T_{\text{timeout}}$	$0.2 \cdot \bar{v} \cdot \Delta t(s)$
Factor $\alpha$ for access routers	0.5
Factor $\theta$ for the controller	0.7

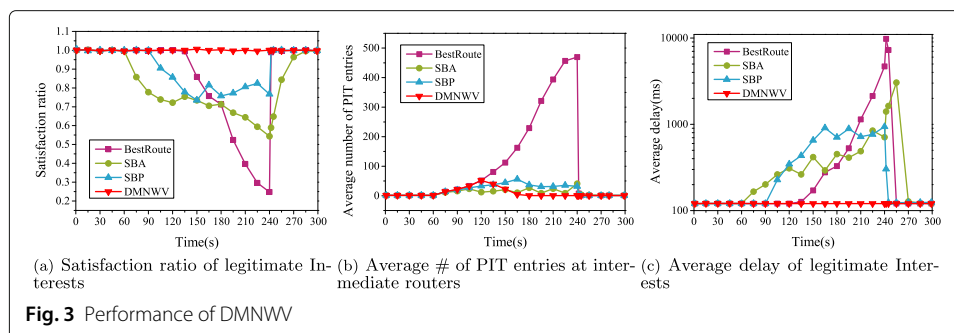
**4.1 Performance evaluation**

**4.1.1 Performance of DMNWW**

In this part, we evaluate the performance of DMNWW and compare it with *Satisfaction-based Interest acceptance (SBA)* and *Satisfaction-based pushback (SBP)* presented in [6], and *BestRoute* strategy which represents the state of the network with no defense mechanism. The attack speed is 3% higher per second, the factor  $\alpha$  is set to 0.5 and the time window is set to 1 s. The results are shown as Fig. 3.

We quantify the performance of DMNWW from the following three aspects: (1) the satisfaction ratio of legitimate Interests, (2) the average number of PIT entries at intermediate routers, and (3) the delay of legitimate Interests (time interval between the first Interest sent and the received Data packet, i.e., including the time of Interest retransmissions).

When there is no defense mechanism, the number of PIT entries keeps gradually increasing after the attack starts. In the early stage of the attack, though the PIT usage at intermediate routers becomes larger than that in the normal condition, there is still no router’s PIT resources exhausted. Therefore, the satisfaction ratio and delay of legitimate Interests keep unchanged. But with the increase of the attack speed, malicious Interests continue to accumulate in the PITs of routers under attack. Finally, the PIT resources of victim routers will be exhausted by malicious Interests, which makes them unable to create new PIT entries for subsequently incoming legitimate Interests, so the satisfaction



ratio of legitimate Interests begins to decline and the delay of legitimate Interests begins to increase after the attack lasts for a period of time.

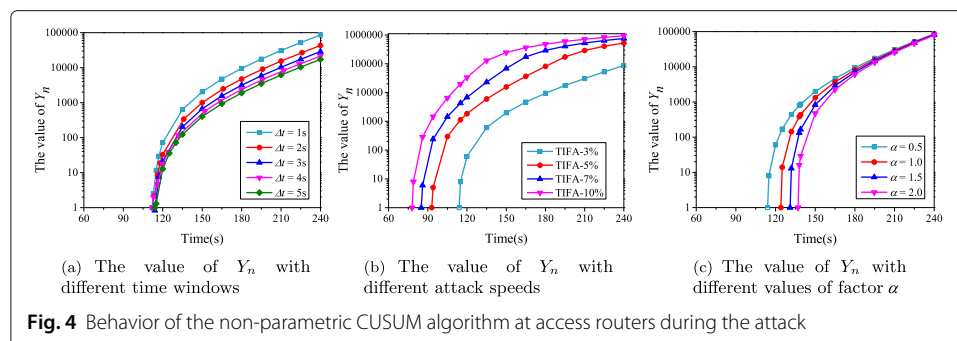
In SBA and SBP, the probability that a router accepts the received Interest is based on the satisfaction ratio of incoming Interests on the arrival interface of the received Interest. After the sophisticated IFA starts, the satisfaction ratio of incoming Interests on intermediate routers' interfaces which malicious Interests pass by begins to decline. Therefore, some legitimate Interests will be dropped mistakenly and the delay of legitimate Interests becomes larger, even when there is still no router's PIT resources exhausted by malicious Interests in the early stage of the attack. Though the PIT usage is improved, it is still higher than that in the normal condition. Since there are still a proportion of malicious Interests successfully forwarded and some requests from legitimate consumers mistakenly dropped, such Interests will pend in routers' PITs until their lifetime expires.

However, in DMNWV, the satisfaction ratio and average delay of legitimate Interests are always the same as those before the sophisticated IFA starts. Though the PIT usage at intermediate routers becomes larger at the beginning of the attack, it is still relatively low and will return to its normal level after the attack is detected. Since DMNWV can detect the sophisticated IFA timely before the victims' PITs are overwhelmed and then mitigate the attack at source, i.e., directly dropping all the Interests from the malicious interfaces at access routers (i.e., attackers), which can directly prevent malicious Interests from entering the network and will not throttle the requests from legitimate consumers.

**4.1.2 Exploration on the parameter settings of the non-parametric CUSUM algorithm at access routers**

In this part, we explore the parameter settings of the attack detection algorithm at access routers. Figure 4 shows the impact of different parameters on the behavior of the non-parametric CUSUM algorithm on a malicious interface at an access router during the sophisticated IFA, including the time window, the factor  $\alpha$ , and the attack speed. The default value of time window is 1 s, the default value of factor  $\alpha$  is 0.5, and the attack speed is 3% higher per second by default.

Figure 4a shows the value of  $Y_n$  under different time windows. The value of time window ranges between 1 to 5 s. It can be seen that the value of  $Y_n$  is always equal to zero when the attack speed is relatively low in the early stage of the attack. With the increase of the attack speed, the average speed of incoming Interests on a malicious interface at an access router becomes larger and the value of  $Y_n$  begins to continuously accumulate and keeps increasing. Since the value of  $Y_n$  is calculated and accumulates at the end of each time



window, the smaller the time window is, the more frequently the value of  $Y_n$  accumulates, and the faster the value of  $Y_n$  grows. Figure 4b shows the value of  $Y_n$  under different attack speeds. The attack speed is set to 3%, 5%, 7%, and 10% higher per second, respectively. It is obvious that the faster the attackers speed up, the larger the value of  $Y_n$  is at the same time and the earlier an access router can find an abnormality on the malicious interface. Figure 4c presents the value of  $Y_n$  while the factor  $\alpha$  ranges between 0.5 and 2.0. The smaller the factor  $\alpha$  is, the smaller the value of  $\beta$  is and the earlier the value of  $Y_n$  becomes larger than zero and begins to keep increasing. Since the attackers gradually speed up, the attack speed becomes significantly larger than the mean value of the speed of Interests from legitimate consumers (i.e.,  $\bar{v}$ ) after the attack lasts for a period of time and the difference in the value of  $Y_n$  under different values of the factor  $\alpha$  becomes smaller.

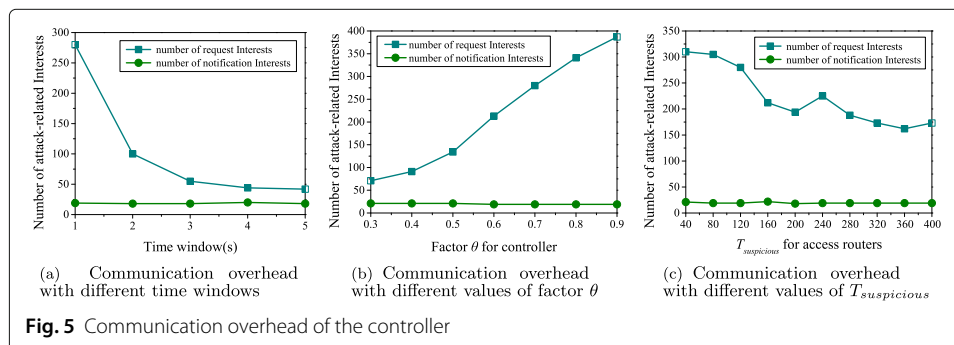
### 4.1.3 Communication overhead of the controller

In this part, we measure the communication overhead of the central controller and evaluate how three different factors impact the overhead, including the time window, the factor  $\Theta$  for the controller, and the value of  $T_{suspicious}$  for access routers.

We express the communication overhead of the controller in the form of the number of attack-related Interests sent by the controller in our simulations, which mainly consists of two parts, (1) request Interests sent to access routers under attack to retrieve their detected abnormal information, respectively, the responses of which should be further parsed and analyzed by the controller, and (2) notification Interests sent to notify access routers of the controller’s decision, respectively.

**Request Interests sent by the controller:** Figure 5a shows the impact of time window on the number of request Interests sent by the controller. The time window ranges from 1 to 5 s. With the increase of time window, the number of request Interests sent by the controller decreases. The reason is that after receiving an attack-related notification from an access router, the larger time window is, the less frequently the controller sends Interests to request the detected abnormal information and the less Interests the controller sends.

Figure 5b shows how the factor  $\Theta$  for the controller impact the number of request Interests sent by the controller. The value of  $\Theta$  ranges from 0.3 to 0.9. In DMNwV, the controller determines that a link is under an IFA when the number of suspicious Interests transmitted on the link reaches a certain percentage (i.e., the factor  $\Theta$ ) of the link’s capacity limit based on all the reported abnormal information. Therefore, for a link which malicious Interests transmit on, the larger the value of  $\Theta$  is, the later the controller will



**Fig. 5** Communication overhead of the controller

determine that the link is under attack and the more Interests the controller should send to request the abnormal information detected by access routers.

Figure 5c shows the number of request Interests sent by the controller under different values of  $T_{\text{suspicious}}$ , which ranges from 40 to 400 (i.e., from 1 to 10 times the rate of Interests from legitimate consumers). In DMNWV, when no attack-related notification from an access router is received, no Interests will be issued by the controller. The controller begins to periodically send request Interests to obtain the abnormal information detected by an access router after the access router detects something abnormal and notifies the controller. It is obvious that the smaller the value of  $T_{\text{suspicious}}$  is, the earlier an access router under an IFA detects something abnormal and sends a notification to the controller, and the earlier the controller begins to request the detected abnormal information. Therefore, in general, the number of request Interests sent by the controller decreases with the increase of the value of  $T_{\text{suspicious}}$  on the whole. However, in some cases, a larger value of  $T_{\text{suspicious}}$  may lead to more request Interests sent by the controller, which is shown as Fig. 5c when the value of  $T_{\text{suspicious}}$  is set to 240. The reason is that in DMNWV, after an IFA is determined, the controller will further locate attackers and notify access routers under attack of their malicious interfaces, respectively. And for an access router, the abnormal information of the interfaces that have already been determined as malicious by the controller will no longer be reported to the controller. In such case, the total number of suspicious interfaces whose abnormal information will be reported to the controller decreases, and thus, the total number of suspicious Interests reported to the controller becomes smaller. And the controller needs to send more request Interests to obtain the newly detected abnormal information and determines that a link is under attack until the number of suspicious Interests transmitted on the link becomes large enough. Therefore, even more request Interests should be sent by the controller though the value of  $T_{\text{suspicious}}$  is set larger in some cases.

**Notification Interests sent by the controller:** As can be seen from Fig. 5, the total number of notification Interests sent by the controller keeps stable under different parameter settings. In DMNWV, after an IFA is determined, the controller will further locate attack sources. For each access router that is determined as under attack, the controller will notify the access router of all its malicious interfaces by sending only one notification Interest, even when there is more than one malicious interfaces. However, in some cases, for an access router under attack, there are some interfaces that are judged as suspicious later after the previously detected and reported suspicious interfaces have already been determined as malicious by the controller. In such case, the access router should notify the controller again to report the abnormal information of its newly detected suspicious interfaces. Then, the controller should send another notification Interest to the access router after the newly reported suspicious interfaces are determined as malicious. However, the probability of such case is limited since distributed IFA attackers always try to simultaneously start the attack and the differences of attack on different malicious interfaces are limited. In general, the total number of notification Interests sent by the controller is approximately equal to the number of access routers under attack and may be slightly larger in some cases.

As can be seen from the above, the proposed mechanism, DMNWV, can work efficiently with significantly low communication overhead of the controller. In our experiments, the total number of request Interests sent by the controller is smaller than 400 and

the number of notification Interests sent by the controller is approximately equal to 20 during the entire simulation run, while the rate of Interests from legitimate consumers is 40 Interests per second. Therefore, the work of the central controller will not bring about too much burden to the network.

#### 4.2 Result discussion

As can be seen from the experimental results presented above, DMNWV can timely detect the sophisticated IFA and locate attack sources, and then mitigate the attack at source without throttling legitimate consumers' requests with relatively low communication overhead of the central controller.

However, there are still some limitations of DMNWV. It is obvious that in DMNWV, the cooperation between the central controller and access routers plays a key role in the performance of DMNWV. A good cooperation can make DMNWV work as efficiently as possible to timely detect and mitigate the attack to protect the network from being seriously damaged as well as keep the communication overhead of the controller significantly low at the same time. However, improper interaction of attack-related information between the controller and access routers can also make DMNWV suffer performance degradation to some extent. For example, as shown in Fig. 5c, the parameter setting that  $T_{\text{suspicious}}$  is 240 while the factor  $\Theta$  is 0.7 leads to the phenomenon that there are a small proportion of attackers difficult to be located and the communication overhead of the controller is greater. Moreover, when a significantly high-rate IFA is launched, the performance of DMNWV may also be negatively impacted since the attack-related Interests exchanged between the controller and access routers may be dropped by the victim routers whose PIT resources will be immediately exhausted by malicious Interests after the high-rate IFA starts.

In our future work, the current design and implementation of DMNWV introduced in this paper can be further improved from the following two aspects. First, a reasonable and comprehensive method of parameter settings should be proposed to improve the cooperation between the controller and access routers to make DMNWV work as efficiently as possible. Among all the parameters in DMNWV, our future work will mainly focus on the following two parameters of them, (1) the value of  $T_{\text{suspicious}}$  which determines when an access router under attack will notify the controller mainly based on the Interest speed of its each interface and what the scale of suspicious Interests is at that time, and (2) the factor  $\Theta$  which determines when the controller will make a decision that a link is under attack based on the link's capacity limit and all the reported abnormal information from access routers. These two parameters play the major role in the effectiveness of the cooperation between the controller and access routers. Second, DMNWV should also be improved to work efficiently enough when faced with a relatively high-rate IFA.

## 5 Conclusion

In this paper, we propose a mechanism, called DMNWV, to detect and mitigate a more sophisticated IFA from the network-wide view based on a central controller, aiming to timely detect the attack and locate attackers before it causes great damage to the network and then mitigate the attack at source without throttling the requests from legitimate consumers.



In DMNWW, each access router is responsible for detecting whether there is something abnormal on each of its interfaces. When an access router finds an abnormality but is unsure whether there is an IFA, it will send an Interest with specific name to notify the controller and then report the abnormal information of its suspicious interfaces based on the controller's requests. The central controller collects all the abnormal information detected by access routers and makes a comprehensive decision on whether there is an ongoing IFA based on the overall state of the network. If an IFA is determined, the controller will further locate the attackers and notify the access routers under attack of their malicious interfaces respectively. Afterwards, access routers can refuse to accept any Interest from its malicious interfaces determined by the controller, which can directly and immediately prevent malicious Interests from entering the network as well as avoid throttling requests from legitimate consumers. The experimental studies validate that DMNWW can timely detect the sophisticated IFA and accurately locate attackers before the attack causes great damage to the network with quite low communication overhead of the central controller, and legitimate consumers can still retrieve the desired content. In our future work, we will improve the cooperation between the central controller and access routers and also enable DMNWW to work efficiently enough when faced with a relatively high-rate IFA.

#### Abbreviations

NDN: Named data networking; PIT: Pending interest table; DDoS: Distributed denial of service; IFA: Interest flooding attack; DMNWW: Detecting and mitigating the attack from the network-wide view; DPE: Disabling PIT exhaustion; CUSUM: Cumulative sum; BDP: Bandwidth-delay product; SBA: Satisfaction-based Interest acceptance; SBP: Satisfaction-based pushback

#### Acknowledgements

Not applicable.

#### Authors' contributions

The authors have contributed jointly to the manuscript. The authors have read and approved the final manuscript.

#### Funding

The research work leading to this article is supported by the National Key Research and Development Program of China (2018YFB1800602, 2017YFB0801703), the National Natural Science Foundation of China (61602114), the CERNET Innovation Project (NGIICS20190101, NGII20170406), and JSPS KAKENHI (JP19H04105).

#### Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>School of Cyber Science & Engineering, Southeast University, Nanjing, China. <sup>2</sup>Key Laboratory of Computer Network and Information Integration of Ministry of Education of China, Southeast University, Nanjing, China. <sup>3</sup>Key Laboratory of Computer Network Technology of Jiangsu Province, Nanjing, China. <sup>4</sup>Computer Science Department, Colorado State University, Fort Collins, USA.

Received: 13 September 2019 Accepted: 23 April 2020

Published online: 08 July 2020

#### References

1. C. Guang, Z. Lixia, H. Xiaoyan, Z. Shaoqi, W. Hua, Ruidong L., Chengyu F., in *First IEEE International Workshop on Network Meets Intelligent Computations, NMIC@ICDCS 2019, Dallas, TX, USA, July 7-9, 2019*, Detecting and Mitigating A Sophisticated Interest Flooding Attack in NDN from the Network-Wide View (IEEE, 2019), pp. 7–12. <https://doi.org/10.1109/NMIC.2019.00007>
2. J. Pan, S. Paul, R. Jain, A survey of the research on future Internet architectures. *IEEE Commun. Mag.* **49**(7), 26–36 (2011)
3. L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, c. claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking. *Comput. Commun. Rev.* **44**(3), 66–73 (2014)

4. C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, L. Zhang, A case for stateful forwarding plane. *Comput. Commun.* **36**(7), 779–791 (2013)
5. P. Gasti, G. Tsudik, E. Uzun, L. Zhang, in *Proc. ICCCN, Nassau, Bahamas*, DOS and DDOS in named data networking, (2013), pp. 1–7. <https://doi.org/10.1109/icccn.2013.6614127>
6. A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, L. Zhang, in *Proc. IFIP Networking Conference, Brooklyn, New York, USA*, Interest flooding attack and countermeasures in named data networking (IEEE, 2013), pp. 1–9. <https://ieeexplore.ieee.org/document/6663516>
7. H. Dai, Y. Wang, J. Fan, B. Liu, in *Proc. INFOCOM Workshops, Turin, Italy*, Mitigate DDOS attacks in NDN by interest traceback (IEEE, 2013), pp. 381–386. <https://doi.org/10.1109/infcomw.2013.6970722>. <https://ieeexplore.ieee.org/document/6970722>
8. A. Compagno, M. Conti, P. Gasti, G. Tsudik, in *Proc. LCN, Sydney, Australia*, Poseidon: mitigating interest flooding DDOS attacks in named data networking (IEEE, 2013), pp. 630–638. <https://doi.org/10.1109/lcn.2013.6761300>. <https://ieeexplore.ieee.org/document/6761300>
9. V. G. Vassilakis, B. A. Alohal, I. Moscholios, M. D. Logothetis, in *Proc. AICT, Brussels, Belgium*, Mitigating distributed denial-of-service attacks in named data networking (IARIA, 2015), pp. 18–23. [https://www.researchgate.net/publication/302878160\\_Mitigating\\_Distributed\\_Denial-of-Service\\_Attacks\\_in\\_Named\\_Data\\_Networking](https://www.researchgate.net/publication/302878160_Mitigating_Distributed_Denial-of-Service_Attacks_in_Named_Data_Networking)
10. K. Ding, Y. Liu, H. Cho, H. Chao, T. K. Shih, Cooperative detection and protection for interest flooding attacks in named data networking. *Int. J. Commun. Syst.* **29**(13), 1968–1980 (2016)
11. J. Tang, Z. Zhang, Y. Liu, H. Zhang, in *Proc. GreenCom, Beijing, China*, Identifying interest flooding in named data networking, (2013), pp. 306–310
12. K. Wang, H. Zhou, Y. Qin, H. Zhang, Cooperative-filter: countering interest flooding attacks in named data networking. *Soft Comput.* **18**(9), 1803–1813 (2014)
13. L. Zhao, G. Cheng, X. Hu, H. Wu, J. Gong, W. Yang, C. Fan, in *Proc. HotICN, Shenzhen, China*, An insightful experimental study of a sophisticated interest flooding attack in ndn, (2018), pp. 121–127. <https://doi.org/10.1109/hoticn.2018.8605965>
14. H. Salah, J. Wulfheide, T. Strufe, in *Proc. INFOCOM Workshops, Hong Kong, China*, Lightweight coordinated defence against interest flooding attacks in NDN, (2015), pp. 103–104. <https://doi.org/10.1109/infcomw.2015.7179364>
15. H. Salah, J. Wulfheide, T. Strufe, in *Proc. LCN, Clearwater Beach, FL, USA*, Coordination supports security: a new defence mechanism against interest flooding in NDN, (2015), pp. 73–81. <https://doi.org/10.1109/lcn.2015.7366285>
16. Y. Xin, Y. Li, W. Wang, W. Li, X. Chen, in *Proc. Globecom, Washington, DC, USA*, A novel interest flooding attacks detection and countermeasure scheme in NDN, (2016), pp. 1–7. <https://doi.org/10.1109/glocom.2016.7841526>
17. T. Zhi, H. Luo, Y. Liu, A Gini impurity-based interest flooding attack defence mechanism in NDN. *IEEE Commun. Lett.* **22**(3), 538–541 (2018)
18. K. Wang, H. Zhou, Y. Qin, J. Chen, H. Zhang, in *Proc. Globecom Workshops, Atlanta, GA, USA*, Decoupling malicious interests from pending interest table to mitigate interest flooding attacks, (2013), pp. 963–968. <https://doi.org/10.1109/glocomw.2013.6825115>
19. H. H. Takada, U. Hofmann, Application and analyses of cumulative sum to detect highly distributed denial of service attacks using different attack traffic patterns. *IST INTERMON Newsletter.* **7**, 1–14 (2004)
20. A. Afanasyev, N. Tilley, P. L. Reiher, L. Kleinrock, Host-to-host congestion control for TCP. *IEEE Commun. Surv. Tutor.* **12**(3), 304–342 (2010)
21. The NDN simulator ndnSIM. <http://ndnsim.net/>
22. N. Spring, R. Mahajan, D. Wetherall, in *Proc. SIGCOMM, New York, NY, USA*, Measuring ISP topologies with Rocketfuel, (2002), pp. 133–145. <https://doi.org/10.1145/964725.633039>

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---