Synergetic Denial-of-Service Attacks and Defense in Underwater Named Data Networking

Yue Li[1], Yingjian Liu[*1,3], Yu Wang[2], Zhongwen Guo[1], Haoyu Yin[1], Hao Teng[1]
[1]Department of Computer Science and Technology, Ocean University of China, Qingdao, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[3]Department of Computer Science, University of North Carolina at Charlotte, Charlotte, USA
{yueli, yhy, tenghaoouc}@stu.ouc.edu.cn {liuyj, guozhw}@ouc.edu.cn wangyu@temple.edu

***Abstract*—Due to the harsh environment and energy limitation, maintaining efficient communication is crucial to the lifetime of Underwater Sensor Networks (UWSN). Named Data Networking (NDN), one of future network architectures, begins to be applied to UWSN. Although Underwater Named Data Networking (UNDN) performs well in data transmission, it still faces some security threats, such as the Denial-of-Service (DoS) attacks caused by Interest Flooding Attacks (IFAs). In this paper, we present a new type of DoS attacks, named as Synergetic Denial-of-Service (SDoS). Attackers synergize with each other, taking turns to reply to malicious *interests* as late as possible. SDoS attacks will damage the Pending Interest Table, Content Store, and Forwarding Information Base in routers with high concealment. Simulation results demonstrate that the SDoS attacks quadruple the increased network traffic compared with normal IFAs and the existing IFA detection algorithm in UNDN is completely invalid to SDoS attacks. In addition, we analyze the infection problem in UNDN and propose a defense method Trident based on carefully designed adaptive threshold, burst traffic detection, and attacker identification. Experiment results illustrate that Trident can effectively detect and resist both SDoS attacks and normal IFAs. Meanwhile, Trident can robustly undertake burst traffic and congestion.**

***Index Terms*—Underwater Sensor Networks, Named Data Networking, Interest Flooding Attack, Denial-of-Service**

I. Introduction

In the past three decades, underwater wireless sensor network (UWSN) has been applied to marine data collection and monitoring [1]–[3]. Unlike terrestrial networks, UWSN faces with some unique challenges, including slow propagation speeds, harsh communication environments, limited energy, node mobility, etc. [4]. These challenges make UWSN more fragile to some security threats, e.g., jamming attack [5], [6], Denial-of-Service (DoS) attack [7], and sybil attack [8].

As a future network architecture, named data networking (NDN) [9], a paradigm of information-centric networking (ICN) [10]–[12], has been widely investigated in recent years. NDN retrieves and forwards each packet based on its content information rather than host identifiers. There are just two types of packet in NDN, *interest* packet (i.e., a request) and *data* packet (i.e., a response to the request). Each router in NDN contains three essential data structures: Pending Interest Table (PIT), Content Store (CS), and Forwarding Information Base (FIB). PIT records information of *interests* that have been forwarded but not yet replied. All duplicate *interests* merge into a single PIT entry, which is deleted only when it is expired or the corresponding *data* packet is received. CS is used to cache the forwarded *data* packets. FIB is a routing table for the incoming *interest* packets according to their name prefixes.

Recently, NDN has been applied to UWSN as Underwater Named Data Networking (UNDN) [13]–[16]. Since NDN supports in-network content caching and location-independent data access, it has unique advantages in dealing with bandwidth usage, multipath forwarding, and node mobility. Therefore, UNDN can achieve more stable delay performance and higher satisfaction ratio than IP-based UWSN. However, although UNDN with suitable MAC protocol performs well in data transmission, it still faces with some security threats of Interest Flooding Attacks (IFAs). IFA floods a large number of malicious *interest* packets that cannot be satisfied to exhaust the memory resource for PIT in routers [17], [18]. It will disturb legitimate *interest* packets sent by normal users and even cause complete denial of services in UNDN [19]. Several countermeasures exploit a typical feature of IFA, high timeout ratio caused by plenty of expired PIT entries, to detect threats in NDN [17], [20]–[23]. However, these methods are based on the assumptions of known, wired connections and sufficient energy supplies, which are not applicable to UNDN. Martin and Rajasekaran [19] proposed a defense method based on timeout ratio and channel measurement for UNDN. To refer conveniently, we denote it as TRD (Timeout Ratio based Defense) in this paper. To the best of our knowledge, TRD is the only currently existing IFA detection method for UNDN. But TRD suffers from infection problem and its evaluation results are only valid with a strong assumption on large node transmission range.

This paper proposes a more severe Synergetic Denial-of-Service (SDoS) attack, in which attackers automatically form attacking networks and answer malicious *interests* in turns with maximum latency before expiration. In such way, not only the *data* packets replied by attackers will pollute the CS, but also excessive high latency *interests* will occupy the PIT and consume memory resources. As a result, a DoS attack

This work was supported partially by the National Natural Science Foundation of China (NSFC) under Grant No. 61572448, No. 61673357, No. 61827810, and by the Key R&D Program of Shandong Province, China under Grant No. 2018GSF120015. This work was partially done when Y. Liu visited the Department of Computer Science, University of North Carolina at Charlotte, with a scholarship from the China Scholarship Council.

* Y. Liu is the corresponding author.

TABLE I. SUMMARY OF DIFFERENT ATTACKS IN NDNS/UNDNS

	FCPA	CPA	IFA	SDoS (this work)
Compromised Roles	Router or Producer	Consumer	Consumer	Consumer and (Router or Producer)
Malicious Packet	Fake *data* for valid *interest* with low latency	Valid unpopular *interest*	Fake *interest*	Fake *interest* and real *data* for it with high latency
Damaged Structure	CS of routers	CS of routers	PIT of routers	PIT, CS and FIB of routers
Attack Mode	Node independence	Node independence	Node independence	Multi-node cooperation
Attack Effects	Cache hit rate decreases & network latency increases	Cache hit rate decreases & network latency increases	PIT surges, eventually to network congestion	Cache hit rate decreases & PIT surges, eventually to net congestion

occurs. In addition, different attackers sending *data* packets with the same prefix are likely to interfere with FIB. TRD cannot be used to detect SDoS attacks because the malicious *interests* are no longer expired. Meanwhile, we discover a new infection problem that an underwater ordinary node (UON) is misidentified as an attacker by the next hop UON because it forwards malicious *interests* before the defense method detects the attack. In order to solve this infection problem and mitigate SDoS attacks, we propose **Trident**, a defense method with adaptive threshold, burst traffic detection, and attacker identification.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, SDoS is the first DoS attack which can damage PIT, CS, and FIB all at the same time. We also discover the unique *infection problem* in UNDN.
- We design and implement Trident, a defense method to mitigate SDoS attacks, which can choose adaptive thresholds, identify attackers, and detect burst traffics.
- We assess the effect of SDoS attacks and Trident by experiments running on Aqua-Sim Next Generation [24]. Simulation results show that the only currently existing IFA detection method for UNDN is completely invalid to SDoS attacks. Trident can not only effectively resist SDoS attacks and IFAs, but also robustly undertake burst traffics and congestions.

The remainder of this paper is organized as follows. Section II discusses the related work. Sections III and IV introduce the attack model and defense method, respectively. Simulation results are presented in Section V. Finally, Section VI concludes the paper with a discussion on future work.

II. RELATED WORK

Security Issues: The principle of IFA [17] in NDN is sending a lot of *interests*, asking for nonexistent content names, to make PIT overloaded. Different from IP-based Distributed DoS (DDoS), IFA targets to attack routers rather than servers [18]. Malicious *interests* with entirely forged names in IFA cannot match any FIB entry in routers and will normally be broadcasted by FIB [9], [18]. They occupy PIT until expiration, consuming the memory resources of routers. Therefore, legitimate *interests* have to be dropped by aggrieved routers. UNDN also faces with the DoS threat based on IFA [19], which not only disturbs communication, but also consumes node energy to reduce the lifetime of whole system.

Unlike IFA damaging network infrastructure, Cache Pollution Attack (CPA) [25], [26] aims to destroy cache balance by sending a large number of elaborate *interest* packets to the network, thereby tricking NDN routers to cache unpopular contents. Content Poisoning Attack is similar to CPA, but legitimate *interests* are responded by corrupted or fake data [27]–[29]. To damage cache, malicious *data* packets sent by compromised routers or bad producers must arrive at the downstream router before the real *data* packets do. In order to distinguish from CPA, we use FCPA to denote Content Poisoning Attack which exploits fake *data*.

In Collusive Interest Flooding Attack (CIFA), malicious content producer responds shortly before the corresponding PIT entries expire. The idea of CIFA is similar to our SDoS, but it is only briefly introduced in [30] without detailed study and analysis. In this paper, we study a specific scenario of DoS attack, SDoS, in UNDN and estimate its damages to both PIT and CS. Moreover, SDoS attackers take turns responding to malicious *interests*, which may interfere with FIB.

Characteristics of security issues and differences among these attacks are summarized in Table I.

Countermeasures of CPA: It is quite difficult to detect CPA precisely because legitimate users may suddenly become interested with certain unpopular contents under some special circumstances. The false alarm rate of many proposed CPA detection methods is relatively high [25], [26], [31]–[35]. To reduce the harm caused by false alarm, most methods mitigate attacks in the same way—forward but do not cache unpopular contents probably requested by attackers. Therefore, these methods can only mitigate the damage to CS. But for SDoS, excessive malicious *interests* still may overflow PIT, which will lead to disrupted communication.

Countermeasures of IFA: Several solutions of detecting and mitigating IFA are based on Interest Satisfaction Ratio (ISR), which is defined as the ratio of satisfied *interests* to total *interests*. Three defense methods with different complexity are proposed in [17]. All these methods set token buckets for each router interface and the best one will set the token bucket size through the ISR of each interface. Routers pass interface restrictions downstream in turn to limit malicious traffic. An approach named Disabling PIT Exhaustion (DPE) is proposed in [20]. The prefix whose ISR exceeds the threshold is considered as malicious and recorded in malicious list. DPE diverts all the malicious *interests* out of PIT. Although DPE can mitigate memory exhaust of PIT, malicious traffic is not filtered. ISR-based methods are invalid to SDoS detection because few malicious *interests* are expired.

Except for ISR, PIT usage or PIT size is also a good indicator of IFA and SDoS attacks. Interest traceback [18] is triggered when PIT size increases at an alarming rate or

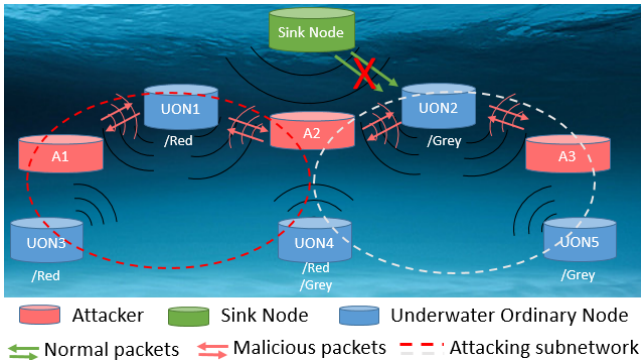


Fig. 1. Example of SDoS Attack.

exceeds a threshold. However, it does not mention clearly how to identify malicious *interest* and it does not consider the impact of burst traffic on PIT size.

In order to detect IFA more accurately, both ISR-oriented and PIT-oriented indicators are used in [21]–[23]. Poseidon [21] not only limits the incoming packet rate on the interface, but also sends out an alert message to its neighbor on the malicious interface. CoMon [22], [23] is a more cooperative approach among routers. It selects a small group of routers to monitor the entire traffic. However, either Poseidon or CoMon requires the transmission of control information between routers, which brings additional security risks because attackers may capture routers and forge malicious control information. These methods still cannot be used to detect SDoS attacks because ISR is a necessary condition for them.

Some security architectures based on cryptography are proposed to resist DDoS attacks in NDN. In InterestFence [36], content servers make a hash-based security label (HSL) for every content to claim their existence. However, malicious list and HSL verification methods stored in routers may become a new target of IFA. Capability-based security enforcement architecture (CSEA) [37] is a distributed lightweight one-time signature method. Routers and content producers verify the authenticity of *interest* by token. However, CSEA has very limited effect on SDoS detection because it relies on reliable content producers, but SDoS attackers can imitate content producers to interact with other attackers.

Countermeasures in UNDN: All solutions mentioned above are not applicable to UNDN due to their assumptions of known, wired connections and sufficient energy supplies. Currently, TRD [19] is the only defense method for UNDN. It adapts to underwater propagation delays and acoustic broadcast median by maintaining a DoS detection table. Like Poseidon, TRD sends alert messages to neighbor nodes after detecting attacks. Since TRD is based on timeout ratio (i.e., $1-\text{ISR}$), it also cannot be used to detect SDoS attacks. Moreover, TRD ignores the infection problem, which affects the detection accuracy and increases the mis-killing rate.

In summary, all the presented solutions have their limitations to be applied to UNDN. ISR-based methods cannot detect SDoS while PIT-based methods cannot identify malicious *interests* accurately. Moreover, many methods rely on secure communication between routers, which cannot be guaranteed in underwater environment. Most algorithms are too complex to be deployed on underwater nodes. In order to effectively resist DoS attacks and protect network communications, our defense method aims to the following security goals: (1) detect SDoS attack & IFA accurately even under burst traffic; (2) solve the infection problem by distinguishing infected UONs and real attackers; (3) can be deployed individually on any UON without interdependence; and (4) have low algorithm complexity to reduce energy consumption.

III. Attack Model

A. SDoS Overview

In this work, we discovery a more severe DoS attack in UNDN which can degrade in-network caching capabilities and block communication while avoiding the existing countermeasure. We call it *Synergetic Denial-of-Service* (SDoS), the combination and distillation of CPA, FCPA and IFA. The most distinguishing feature of SDoS from those three attacks is that attackers take turns responding to malicious *interests* and delay the reply at an appropriate time, as shown in Fig. 1. So that excessive malicious *interests* will occupy PIT for a long time and will be satisfied shortly before they expire.

The key issues of SDoS are how to cooperate among attackers (consumers and producers) to achieve the maximum attack effect at a minimum cost and how to determine the maximum valid delay (MVD) between each pair of attackers. More details will be introduced later, an overview of how SDoS works can be summarized as follows.

- **Step (1).** Attackers first eavesdrop on the *interest* or *data* packets propagated in the network and save the prefixes used by the current network for subsequent attacks.
- **Step (2).** Attacking networks are organized according to different target prefixes. An attacker constructs malicious *interests* based on the target prefix and sends them to other attackers in the same attacking subnetwork.
- **Step (3).** Every attacker estimates the MVDs between itself and other attackers in the same subnetwork.
- **Step (4).** Attackers send excessive malicious *interests* to the target network and reply *data* packets according to the MVD estimated in Step (3). If the network changes, the attacker goes back to Step (2) to reorganize the attacking network and continue to attack until the target network is completely paralyzed.

B. Attacking Stages

In SDoS, attackers will perform three stages, as shown in Fig. 2. The first stage is *Networking Stage*, in which attackers discover each other and form attacking subnetworks. The last stage is *Active Attack Stage*, in which attackers perform attacks by sending and responding to malicious interests. The middle stage is *Buffer Stage*, coordinating the progress of different attackers. More details about these stages are now presented.

1) Networking Stage: Routing strategy that broadcasts unmatched prefix *interests* in NDN is not applicable to UNDN because it may overload the network. Here, we assume that UONs use static routes where each node has fixed forwarding prefixes and the un-matched *interests* will be dropped [14].

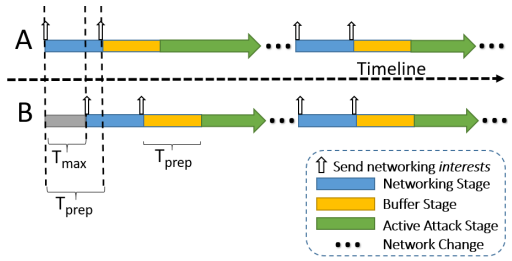


Fig. 2. Stage transformation in SDoS.

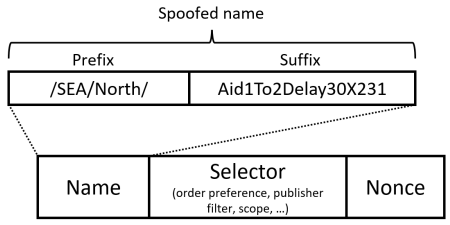


Fig. 3. A malicious *interest* with spoofed name.

Note that there is no specific routing strategy for UNDN at present. Static routes are simple and reasonable, even it is not conducive to launching SDoS attacks.

The packets sent by attackers need to be forwarded by UONs to produce impact. Using static routes, not all UONs can communicate with each other because a prefix which represents a specific network area may not necessarily be forwarded by all UONs. Therefore, communication packets between two attackers may not necessarily be forwarded by the UONs. For example, the network shown in Fig. 1 supports two prefixes (i.e., */Red* and */Grey*), which represent two different network areas. UON1 has responsibility for forwarding *interests* under */Red*, while UON2 is responsible for forwarding *interests* under */Grey*. It can be seen that A1 and A3 cannot communicate with each other by using */Red* or */Grey*. Thus, it is not feasible to preset a single attacking network that contains all attackers.

In SDoS, attackers organize attacking subnetworks according to the target prefixes previously eavesdropped. As shown in Fig. 1, A1 and A2 form an attacking subnetwork against target prefix */Red*, while A2 and A3 form another attacking subnetwork against target prefix */Grey*. Since one attacker (e.g. A2) can eavesdrop on multiple target prefixes, it may locate in multiple attacking subnetworks. In such cases, the attacker may be at different attacking stages in different subnetworks at the same time.

During Networking Stage, an attacker only sends *interest* packets containing networking information (i.e., networking *interests*) at the beginning and the end time points to save energy, and simultaneously collects networking *interests* sent by other attackers. The attacker will set up a Prefix Data Table (PDT), as shown in Table II, for each attacking subnetwork it belongs to. If too many prefixes are detected, the attacker will select several shorter prefixes [36].

2) Active Attack Stage: Attackers in the same attacking subnetwork act as consumer and producer to send malicious packets. Consumer controls the initiative of communication. Malicious *interests* in SDoS have the following characteristics.

TABLE II. PREFIX DATA TABLE FOR SDOS

Prefix	The prefix that constitutes the attacking subnetwork
Stage	The current phase of the attacker
StageTimer	The timer to control the conversion of the stage
Iterator	The pointer to whom the next *interest* will be sent
DelayEstimateTable	Info. of teammates in current attacking subnetwork

TABLE III. ENTRIES OF DELAYESTIMATETABLE

Aid	The ID of teammate attacker
D_{low}	The lower bound of delay estimation range
D_{mid}	The middle value of delay estimation range
D_{up}	The upper bound of delay estimation range
Times	The times of completed delay estimation
MSN	The maximum serial number of *interests* received from the teammate
NetChangeTimer	The timer for sensing network changes

- Request for a nonexistent unique content to avoid being replied by UONs or be satisfied by Cache.
- Appoint a teammate responsible for responding to avoid duplicate responses.
- Indicate the delay in responding to make malicious *interests* occupy PIT.

SDoS concatenates the target prefix and a forged suffix to compose a name field for malicious *interest*, as shown in Fig. 3. We hide the command information of attack coordination in the suffix of name field. Certainly, it can also be hidden in the selector field of *interest*. Rules of spoofed name are illustrated through an example in Fig. 3. Given a spoofed name */SEA/North/Aid1To2Delay30X231*, */SEA/North* is the target prefix used in current attacking subnetwork. In SDoS, each attacker has a unique ID. The attacker chooses the responsible teammate to respond the *interest* by rotation and specifies its ID in the name field of *interest*. *Aid1to2* represents that this malicious *interest* was sent by attacker 1 and should be answered by attacker 2. The iterator field in Table II is a pointer for selecting the responder from the DelayEstimateTable as shown in Table III. Although malicious *interests* are widely spread in the target network, only one attacker responds to each malicious *interest*. Such method will not only save the energy consumption of attackers, but also interfere with the forwarding rules of UONs, because each responder is different for the same prefix.

X231 in Fig. 3 means that the serial number of malicious *interest* is 231. The serial number increases from 1 to ensure that each malicious request is unique and cannot be satisfied by the cache of UONs. Because of multipath forwarding in UWSN, a UON may receive the same *interest* multiple times. Repeated *interests* need only be answered at the first time they are received. Attacker needs to reply a *data* packet when it receives a malicious *interest* from a teammate with a serial number greater than its MSN.

Delay30 in Fig. 3 means that attacker 2 who receives the *interest* needs to wait 30 seconds before sending *data* packet. Delayed reply is a key point of SDoS because it makes malicious *interests* occupy forwarders' resources as much as possible to block the network without timeout. How to determine the MVD of each pair of attackers is a key problem.

Algorithm 1 Binary Delay Estimation

Input: N, D_{low}, D_{up} where $N \geq 1$ and $0 \leq D_{low} \leq D_{up}$
Output: D_{low} as MVD

```
times ⇐ 0
while times++ < N do
    D_mid ⇐ (D_low + D_up)/2
    Delay ⇐ D_mid
    interest ⇐ Forgeinterest(Delay)
    HasResponse ⇐ Send(interest)
    if HasResponse then
        D_low ⇐ D_mid
    else
        D_up ⇐ D_mid
return D_low
```

We propose a binary delay estimation algorithm as shown in Algorithm 1. Similar to the binary search algorithm, we first set the estimation range of MVD: D_{low} to D_{up}. Then we judge whether the estimated value D_{mid} is larger than MVD based on whether the *data* packet can be received, and reset the estimated range accordingly. After N estimations, we get MVD in $(D^n_{low},\ D^n_{up})$. We use D^n_{low} as MVD. The maximum error of estimation is $(D_{up} - D_{low})/2^n$.

At the beginning of Active Attack Stage, the attacker first estimates the delay of all teammates in current attacking subnetwork at a lower data rate, and then sends excessive number of attack *interests* at a higher data rate. Attackers must be able to perceive the network changes in order to reorganize the attacking network. Otherwise, the malicious *interests* sent by attackers cannot be effectively answered. To accommodate for this, we use NetChangeTimer for each teammate, which is reset every time when an *interest* from the teammate is received. Any NetChangeTimer timeout indicates some changes in the attacking subnetwork. Then attackers need to stop attacking and enter into Networking Stage again.

3) Buffer Stage: Buffer Stage is a buffer between Networking Stage and Active Attack Stage. Attackers organize the attacking subnetwork at network initialization or when network changes. Because of the random communication delay and different locations, it is impossible for all attackers to enter into the Networking Stage at the same time. Therefore, malicious *interests* sent by attackers who take the lead in completing Networking Stage will be ignored by those attackers who have not completed the Networking Stage.

Attackers in Buffer Stage do not actively send malicious *interests*, but they receive and reply malicious *interests*. In this way, attackers who take the lead in completing Networking Stage will not send malicious *interests* until they complete their Buffer Stages.

The Buffer Stage and Networking Stage last for the same time duration, set to T_{prep}. We estimate that the maximum time interval between attackers entering the Networking Stage is T_{max}. In order to enable two attackers with the largest interval to connect with each other, we need to set $T_{prep} > T_{max}$. It doesn't matter whether the estimation of T_{max} is accurate or not. A large estimation of T_{max} has no effect on Networking Stage, but it will increase the preparation time to launch an attack. If the estimation of T_{max} is small, the attackers will form two attack subnetworks according to different speed of reflection, which also has little effect on the attack. Fig. 2 illustrates the stage transformation in SDoS. Attacker A first enters into Networking Stage, while attacker B enters into Networking Stage later. Through buffering in the Buffer Stage, attackers can launch attacks smoothly even they enter into Networking Stage at different time.

C. Attack Analysis

1) Attack Conditions: The precondition of SDoS is the same as that of IFA, which can participate in the communication of the target network. SDoS may attack almost all kinds of NDN networks, but it can be launched more easily in UNDN because of the following characteristics: **(a)** *Open wireless environment:* Attackers can easily enter into the network area, eavesdrop and participate in communication. **(b)** *Multiple roles of one node:* UONs are usually responsible for data collection and forwarding. Similarly, an attacker can act as a consumer and a producer simultaneously. **(c)** *Long timer of PIT entries:* Due to the harsh communication environment of UWSN, e.g., long propagation distance, slow transmission speed, high latency, and high bit error rate, the lifetime of PIT entry is relatively long, which provides attackers the opportunity to delay reply. **(d)** *Multipath forwarding:* UWSN often uses multipath transmissions to ensure communication quality, which facilitates the propagation of malicious *interests* [38]. **(e)** *Few countermeasures:* The defense methods proposed to counter DoS attacks in terrestrial NDN are not applicable to UNDN due to their assumptions of known, wired connection and sufficient energy supply.

2) Attack Damages: Like IFA, SDoS floods excessive malicious *interests* with forged names. But these malicious *interests* will be replied, similar to FCPA, by compromised producers or routers. *Data* packets sent by attackers will be stored in CS of routers and squeeze popular contents out, just as CPA does. We set the delay time of replying *interests* to T_{delay}, and the maximum lifetime of *interest* in PIT to T_{live}. The Round-Trip Time (RTT) between producer and the nearest router is set to T_{RTT}. SDoS is equivalent to CPA when $T_{delay} = 0$, and it is equivalent to IFA when $T_{delay} + T_{RTT} > T_{live}$.

In addition, SDoS has a unique ability to interfere with FIB if static routing is not used, because *data* packets replied by attackers with the same prefix actually come from different locations. Energy consumption of nodes is mainly from receiving/sending packets [15]. Packets sent by one attacker are sent and received by at least one UON. In fact, due to broadcasting, UONs send and receive more packets than attackers, which leads to more energy consumed during attacks.

IV. Proposed Countermeasure: Trident

A. Overview

Our countermeasure, **Trident**, contains three modules: *Detecting Attack Threats*, *Identifying Attackers*, and *DoS Restric-*

TABLE IV. SUMMARY OF USED NOTATIONS

Notation	Description
T'	The threshold of PIT size
T''	The threshold for distinguishing infected UONs
A	The magnification factor
θ	The empirical magnification factor
HPS / PS	The historical/current value of the PIT size
$L(i)$	The mean lifetime of interests sent by node i
$P(i)$	The Packet Index of node i
$e(i)$	The number of expired interests sent by node i
$n(i)$	The number of interests sent by node i
Q_p	The set of the nodes send lots of packets
Q_l	The set of the nodes send long lifetime *interests*
T_l	The threshold of classification for Q_l
T_p	The threshold of classification for Q_p
$size(Q)$	The number of nodes in Q

tion. Trident runs periodically on each UON. An overview of Trident is as follows.

- Obtaining the behavior statistics of neighbor UONs, including the number of *interests*, the number of expired *interests*, mean lifetime of *interest* and so on.
- Using an adaptive threshold based on Exponentially Weighted Moving Average (EWMA) to detect the threat of SDoS attack.
- Once attack threat is detected, a classification algorithm based on mean value is used to find suspicious neighbors.
- Putting the suspects in a prison queue, and distinguishing infected UONs from real attackers according to their persistent behavior. Infected UONs will be released from the prison queue to restore innocence.

Before elaborating on Trident, we list frequently used notations in Table IV. We also assume each node contains a unique ID, characterized by its physical and location information.

B. Detecting Attack Threats

Instead of directly detecting attackers by analyzing neighbor nodes, our countermeasure will firstly detect attack threats, through a typical feature of SDoS attack — PIT size of the victim UON, to enhance the efficiency and accuracy. An entry in PIT represents an unfinished *interest*. In the case of stable network transmission, PIT size normally does not fluctuate too much. But it will increase obviously when SDoS attack or IFA occurs. Therefore, Trident can determine whether the UON is under attack threat through the threshold of PIT size, T'.

The threshold of each UON may vary with location and forwarding capability. An adaptive threshold T' based on EWMA, the historical value of PIT size, is calculated by

$$T' = A \times HPS_t, \tag{1}$$

where

$$HPS_t = \begin{cases} \alpha \times HPS_{t-1} + (1-\alpha) \times PS_t & t \geq 2 \\ PS_t & t = 1 \end{cases}, \tag{2}$$

and

$$A = \theta + 0.5^{HPS_t - \theta}, \tag{3}$$

where t represents the current detection cycle, $\alpha \in (0, 1]$ is a smoothing constant, PS_t is the current PIT size, and θ is an empirical amplification factor. When the PIT size is small (less than 5 for some isolated UONs), the magnification effect of θ on HPS is not obvious. So we append it with $0.5^{HPS_t - \theta}$, which increases the magnification effect of θ when HPS is small and approaches 0 when HPS is large.

Note that detecting attack threat is just a preliminary process of SDoS detection. Even if the PIT size exceeds the threshold, UON cannot assert that an attack occurs. It may be caused by burst traffic or network congestion. Therefore, further investigation is needed to tell whether UON is really attacked.

C. Identifying Attackers

This module aims to identify attackers in neighbor nodes. Some terminologies are defined first.

Definition 1 (**Mean Interest Lifetime**). *The Mean Interest Lifetime of a node is the average time of interest, sent by the node, staying in PIT during a detection period. It is defined as*

$$L(i) = \frac{\sum_{j=1}^{n(i)} time(j)}{n(i)},$$

where $time(j)$ is the lifetime of j-th interest sent by node i, staying in PIT, and $n(i)$ is the number of interests sent by node i during a detection period.

Definition 2 (**Packet Index**). *Packet Index is the number of the packets sent by node i during a detection period. It is defined as*

$$P(i) = d(i) + n(i) + e(i),$$

where $d(i)$, $n(i)$, $e(i)$ are the number of data packets, interests, expired interests sent by node i during a detection period, respectively. The extra added $e(i)$ will highlight the expired interests which may more likely to be malicious.

Trident records all neighbor nodes as Q_0. It divides all neighbor nodes into two categories according to their Packet Indexes. One category (i.e., Q_p) contains the nodes sending a large number of packets, the other one contains the nodes sending a small number of packets. Trident calculates the threshold T_p by

$$T_p = \frac{\sum P(i), i \in Q_0}{size(Q_0)}, \tag{4}$$

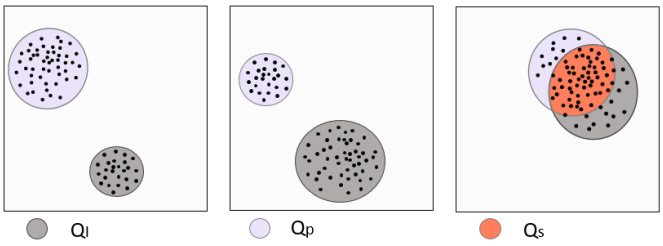
where $P(i)$ is the Packet Index of node i in Q_0, and $size(Q_0)$ represents the number of nodes in Q_0. Through Equ. (4), $Q_p = \{node \in Q_0 \mid P(node) > T_p\}$.

Trident also classifies nodes into two other categories according to Mean Interest Lifetime. Similarly, we can get a set of nodes (i.e., Q_l), sending *interests* that occupy PIT for a long time. The threshold T_l is given by

$$T_l = \frac{\sum L(i), i \in Q_0}{size(Q_0)}, \tag{5}$$

where $L(i)$ is the Mean Interest Lifetime of node i in Q_0. Through Equ. (5), $Q_l = \{node \in Q_0 \mid L(node) > T_l\}$.

Once attack threat is detected, Trident classifies Q_p and Q_l from all neighbor nodes. However, the attack threat may be caused by burst traffic or network congestion. Under burst traffic, some nodes suddenly send a lot of *interests* but these *interests* will be satisfied quickly, generating Q_p and Q_l as illustrated in Fig. 4(a). Under network congestion, the *interests* may stay in PIT for a long time, or even expire, generating Q_p



(a) Burst Traffic. (b) Congestion. (c) DOS Attack.

Fig. 4. Classification of different cases.

and Q_l as shown in Fig. 4(b). But under DoS attacks, Q_p and Q_l will overlap as shown in Fig. 4(c). Therefore, suspicious nodes (i.e., Q_s) sending excessive high latency *interests*, can be identified by $Q_s = Q_p \cap Q_l$.

Although the nodes in Q_s generate malicious behavior by sending lots of high latency *interests*, they may also be UONs that forward malicious packets. Further attacker confirmation and restriction will be introduced in the next subsection.

D. DoS Restriction

This module aims to punish real attacking nodes. We first introduce the infection problem and then propose a simple but effective solution.

We define the **infection problem** as follows. A UON is misidentified as an attacker by its next hop UONs because it forwards malicious interests before detecting the attack threat. This is unavoidable because UONs need some time to verify attacks. Thus, these UONs seem to be infected by real attackers. Furthermore, the infected UONs will continue to infect the next hop UONs. If the infection problem cannot be handled, the normal network communication will be blocked since many UONs are banned as attackers. This infection problem is unique in UNDN, because the attack source cannot be tracked by the incoming interface and edge router as in wired networks. UONs need to determine whether the adjacent nodes are attackers by themselves.

To solve the infection problem, Trident puts all suspicious nodes in a prison queue. Packets sent by suspects in the prison queue will no longer be received, but the packet delivery behavior will still be recorded. A suspect can exit the prison queue after a certain time if there is no more attack behavior recorded. We define Packet Reduce Ratio (PRR) between two consecutive detection periods of node i as:

$$PRR(i) = 1 - \frac{P(i)}{p^{-1}(i)}, \tag{6}$$

where $p^{-1}(i)$ is the Packet Index of node i in the previous detection period. We assume that the Packets Index of a node is proportional to its PIT size because of the flow balance between *interest* and *data* packets in NDN. Therefore, the approximation of Packet Index can be calculated by

$$P(i) \simeq k \times HPS_i,\ k > 0. \tag{7}$$

Infected UONs will have a higher Packet Index at the beginning of attacks because of forwarding malicious packets. Their PIT size will exceed the threshold T', i.e., $p^{-1}(i) > k \times A \times HPS_i$. After detecting and resisting attacks through their own defense methods, their Packet Index will return to normal level. Bring Equ. (7) into Equ. (6), we can get:

$$PRR(i) > 1 - \frac{k \times HPS_i}{k \times A_i \times HPS_i} = 1 - \frac{1}{A_i},$$

where A_i is the magnification factor of node i. Although we do not know the exact value of A_i, we can use A to approximate it because the magnification factors of adjacent nodes are similar. The threshold T'' can be calculated by $T'' = 1 - 1/A$.

If the PRR of a suspect is less than the threshold T'', it will be confirmed as a real attacker and will stay in prison queue for a long time. Otherwise, the suspect is regarded as an infected UON and free from the prison queue.

Algorithm 2 Trident

Input: $\theta > 1.0$

1: $HPS_t \Leftarrow \alpha \times HPS_{t-1} + (1-\alpha) \times PS_t$
2: $A \Leftarrow \theta + 0.5^{HPS_t - \theta}$ and $T' \Leftarrow A \times HPS_t$
3: **if** $(PIT.size \geq T')$**or**$(SuspiciousNodes > 0)$ **then**
4: $T_p \Leftarrow GetT_p()$ and $T_l \Leftarrow GetT_l()$
5: **for** $i = 1$ **to** $DetectionTableEntries$ **do**
6: **if** $InPrison(Node[i])$ **then**
7: **if** $PPR(i) < T''[i]$ **then**
8: Prison($Node[i]$) and Reduce($T''[i]$)
9: **else**
10: Acquit($Node[i]$) and $SuspiciousNodes - -$
11: **continue**
12: **if** $(P(i) \geq T_p)$ **and** $L(i) \geq T_l)$ **and** $(PIT.size \geq threshold)$ **then**
13: Prison($Node[i]$) and $T''[i] \Leftarrow 1 - \frac{1}{A}$
14: $SuspiciousNodes + +$

E. Algorithmic Complexity

The overall algorithm of Trident is shown in Algorithm 2.

1) Space Complexity: During each detection cycle, Trident records the number of packets sent by each neighbor node, the number of satisfied and expired *interests*, and the total lifetime of these *interests*. When there are n neighbor nodes, the space complexity is $O(n)$.

2) Time Complexity: Time complexity of T' in Equ. (1) and that of T'' are both $O(1)$. Time complexity of T_p and T_l are $O(n)$. In Algorithm 2, Q_s can be classified through a loop from line 5 to 14. Thus, time complexity of Q_s is $O(n)$. Therefore, the overall time complexity of Trident is $O(n)$.

V. Simulations

In this section, we present quantified evaluations on the effectiveness of SDoS attack and Trident. Simulations are implemented on Aqua-Sim Next Generation [24], an event-driven UWSN simulator based on NS-3 [39]. Each experiment lasts one hour, including 14 ordinary nodes and 1 sink node in a $12,000m$ by $10,000m$ area. The network topology is shown in Fig. 5. The *interests* sent by sink follow the Zipf-like distribution. There are three content prefixes in our experiment as shown in Table V. We use BroadcastMac protocol with reduced backoff [19]. Detailed parameter settings of SDoS and Trident are shown in Table VI. In order to fully verify the effectiveness of our work, we design three sets of experiments.

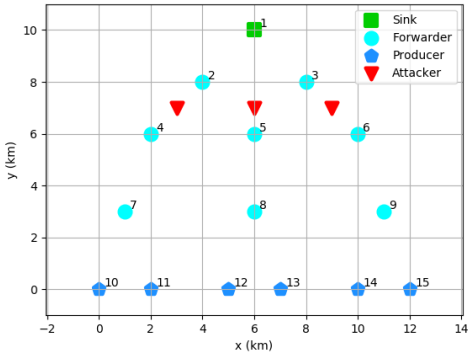


Fig. 5. Network topology.

TABLE V. PREFIX INFORMATION

interest Prefix	Consumer	Forwarder	Producer
/SEA/WEST/	1	2, 4, 7	10, 11
/SEA/NORTH/	1	2, 3, 5, 8	12, 13
/SEA/EAST/	1	3, 6, 9	14, 15

A. Attack Impact

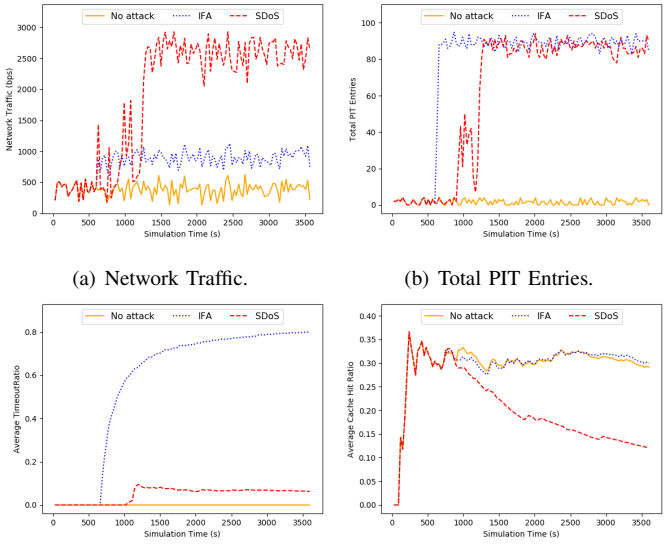
The purpose of this set of experiments is to compare the attack impacts between SDoS and IFA. We do not simulate CPA and FCPA for comparisons because they are relatively less harmful, and their defense methods are almost invalid to SDoS and IFA, as previously mentioned in Section II. We deploy 3 arranged attackers in the area of target network, as shown in Fig. 5. The data rate of attacker's modem is set to $1,440$ bps, which is twice that of UONs. Attackers launch SDoS attacks and IFAs at the point of $600s$, respectively. The main performance metrics are network traffic and the total number of PIT entries. Other indicators include the average timeout ratio and cache hit ratio of node 2 and node 3, since the attackers disguise to be their roles. The experimental results are shown in Fig. 6.

In Fig. 6(a), the average network traffic of the system is 386 bps without attack, increased to 521 bps with IFA, and increased to $2,194$ bps with SDoS. This is because the malicious *interests* (320 $bits$) in SDoS will accompanied by a *data* packet ($1,024$ $bits$). With the same number of malicious *interests*, extra network traffic generated by SDoS could be 4.2 $((1,024+320)/320)$ times that of IFA. In Fig. 6(b), the total PIT entries are very high and similar for both two attacks. However, the *interest* timeout ratio of SDoS is less than 0.1, which is close to normal level, while the timeout ratio for IFA reaches to 0.8, as shown in Fig. 6(c). This means SDoS is difficult to be detected by defense methods based on interest timeout ratio. Fig. 6(d) shows that the cache hit ratio will continuously decline under SDoS, polluting cache seriously.

Note that almost all indicators fluctuate during the period of $600s$ to $1,300s$ under SDoS attack. Because attackers are preparing for formal attacks by organizing the attack network and estimating the MVD.

B. Defense Effect

The purpose of this set of experiments is to compare the effectiveness of different defense methods. Since the only existing defense method for UNDN in [19] is based on timeout



(a) Network Traffic. (b) Total PIT Entries.

(c) Average Timeout Ratio. (d) Average Cache Hit Ratio.

Fig. 6. Comparisons of attack impacts.

ratio, we call it TRD. Experiments are carried out based on the first set of experiments. The empirical amplification coefficient θ of Trident is set to 1.5.

Firstly, we test two defense methods to resist IFAs. The network traffic and total PIT entries fluctuate dramatically with TRD, as shown in Fig. 7(a) and Fig. 7(b), respectively. We can see from Fig. 7(c), whether TRD is used or not, the *interest* timeout ratios are similar. This is because TRD does not solve the infection problem so that constantly kill infected UONs with high false alarms (see Fig. 7(d)). Therefore, the TRD cannot counter IFA effectively. In contrast, the network traffic and total PIT entries with Trident almost remain at the same level as no attack occurs except for a short initialization period. Moreover, Trident solves the infection problem and make FAR (False Alarm Ratio) very low. Therefore, Trident can resist IFA quickly and effectively.

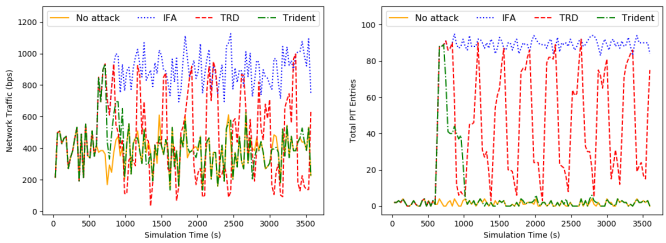
Then, we use two defense methods to resist SDoS attacks, as shown in Fig. 8. All indicators of UNDN with TRD are the same as no defense used, which indicates SDoS attacks completely avoid the TRD. On the contrary, after a short time fluctuation, most indicator of UNDN with Trident almost remains at the same level as no attack occurs. Therefore, Trident is able to effectively detect and resist SDoS attacks. It is worth noting that the FAR of Trident increases slightly at the time point around $1,400s$, as shown in Fig. 8(d). At that time, SDoS just finishes delay estimation and starts an all-out attack. The UONs forward some malicious packets before Trident activation and is misidentified as attackers by downstream nodes. However, Trident solved this infection problem by PRR and FAR returns to normal immediately after the next round of detection. Therefore, with the protection of Trident, neither SDoS nor IFA can cause serious damages.

C. Robustness of Trident

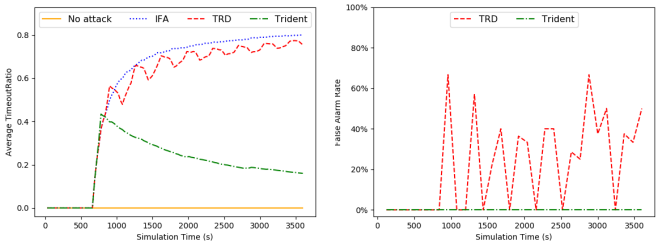
Now, we test the robustness of Trident. As mentioned previously, the burst traffic and network congestion also can

TABLE VI. PARAMETER SETTINGS.

TranX Range	**Rate of Normal Modem**	***interest* Packet Size**	***data* Packet Size**	***interest* Lifetime**	α	**Stage Timer**	**NetChange Timer**	**Estimate Times**	D_{low}	D_{mid}	D_{up}
$3,500m$	$720\ bps$	$320\ bits$	$1,024\ bits$	$60s$	$e^{-1/20}$	$150s$	$180s$	3	$10s$	$40s$	$70s$

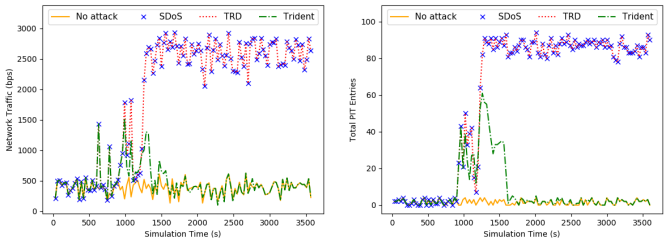


(a) Network Traffic. (b) Total PIT Entries.

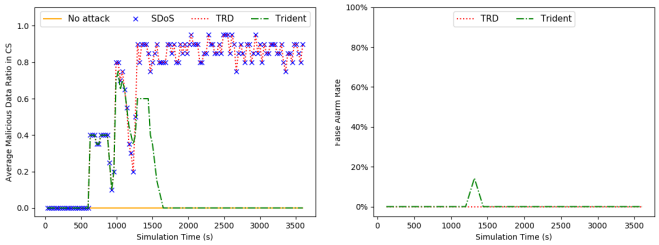


(c) Average Timeout Ratio. (d) False Alarm Ratio.

Fig. 7. Two defense methods against IFA.



(a) Network Traffic. (b) Total PIT Entries.

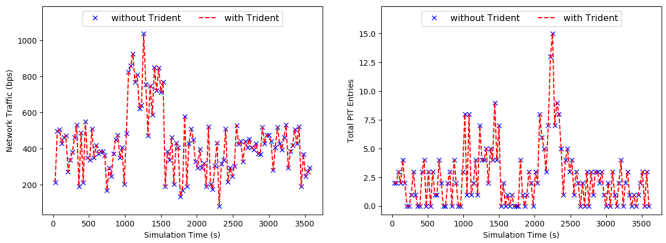


(c) Average Malicious Data Ratio. (d) False Alarm Ratio.

Fig. 8. Two defense methods against SDoS.

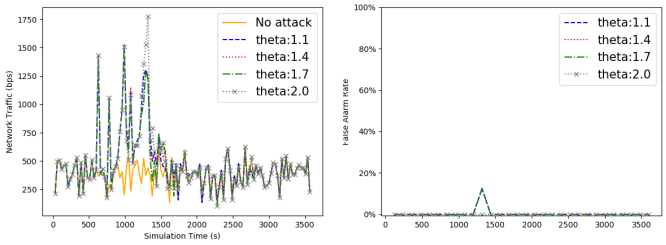
cause PIT size increasing rapidly and trigger Trident. We design a simulation to test whether the Trident is robust enough under such situations.

We set burst traffic that the sink sends twice as many *interests* during the period of $1,000s$ and $1,500s$, and set congestion that half of the *interests* will be timed out during the period of $2,000s$ and $2,500s$. In Fig. 9, the traffic and the total PIT entries have surged in these two periods, respectively. With Trident, all the indicators are the same as normal, which prove that Trident is robust to both burst traffic and network



(a) Network Traffic. (b) Total PIT Entries.

Fig. 9. Robustness of Trident under burst traffic and network congestion.



(a) Network Traffic. (b) False Alarm Ratio.

Fig. 10. Effect of Trident with different θ.

congestion. This is because the Q_p and Q_l have no intersection under such situations, as we mentioned in Section IV-C.

In addition, we test the effect of Trident with different θ, as shown in Fig. 10. The network traffic returns normal after about $1,500s$, while the FAR keeps normal. This shows that θ is not the determinant factor of Trident. Trident with small θ is more sensitive, while Trident with large θ has lower FAR. We just need to set it within a reasonable range.

VI. CONCLUSIONS

In this paper, we introduce a more severe SDoS attack in UNDN, in which attackers synergize with each other and take turns to reply to malicious *interests* with maximum time delay. To the best of our knowledge, SDoS is the first DoS attack which can damage PIT, CS, and FIB all at the same time. We also discover the unique infection problem in UNDN. In addition, we design and implement Trident, a defense method to mitigate SDoS attacks, which can choose adaptive thresholds, identify attackers, and detect burst traffics. Simulation results show that the existing IFA detection method for UNDN is completely invalid to SDoS attacks. Trident can not only effectively resist both SDoS attacks and IFAs, but also be robust under burst traffic and network congestion. In future, we plan to further study secure and efficient forwarding rules for UNDN. We will extend the SDoS attack to other NDN networks and test its damage to FIB with various routing rules.

REFERENCES

[1] J. G. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, "Shallow water acoustic networks," IEEE Commun. Mag., vol. 39, no. 11, pp. 114-119, 2001.

[2] Y. Wang, Y. Liu, and Z. Guo, "Three-dimensional ocean sensor networks: A survey, " Journal of Ocean University of China, vol.11, no. 4, pp.436-450, 2012

[3] C. Zhang, Y. Liu, Z. Guo, G. Sun, and Y. Wang, "Minimum cost localization problem in three-dimensional ocean sensor networks," in Proc. of IEEE ICC, 2014.

[4] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "The challenges of building scalable mobile underwater wireless sensor networks for aquatic applications," IEEE Netw., vol. 20, no. 3, pp. 12-18, 2006.

[5] M. Zuba, Z. Shi, Z. Peng, J.-H. Cui, and S. Zhou, "Vulnerabilities of underwater acoustic networks to denial-of-service jamming attacks," Secur. Commun. Networks, vol. 8, no. 16, pp. 2635-2645, 2015.

[6] P. Xiao, M. Kowalski, D. McCulley, and M. Zuba, "An experimental study of jamming attacks in underwater acoustic communication," in Proc. of 10th Int'l Conf. on Underwater Networks & Systems, 2015.

[7] Y. Dong, H. Dong, and G. Zhang, "Study on Denial of Service against underwater acoustic networks," J. Commun., vol. 9, no. 2, pp. 135-143, 2014.

[8] X. Li, G. Han, A. Qian, L. Shu, and J. Rodrigues, "Detecting Sybil attack based on state information in underwater wireless sensor networks," in Proc. of 21st International Conf. on Software, Telecommunications and Computer Networks, 2013.

[9] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, Kc Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," ACM SIGCOMM Comput. Commun. Rev., vol. 44, no. 3, pp. 66-73, 2014.

[10] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," IEEE Commun. Surv. Tutorials, vol. 16, no. 2, pp. 1024-1049, 2014.

[11] B. Nour, K. Sharif, F. Li, and Y. Wang, "Security and privacy challenges in information centric wireless IoT networks," IEEE Security & Privacy, to appear.

[12] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, and Y. Wang, "A survey of internet of things communication using ICN: A use case perspective," Computer Communications, vol. 142143, pp. 95-123, 2019.

[13] S. H. Bouk, S. H. Ahmed, and D. Kim, "NDN goes deep: Foreseeing the underwater named data networks," in Proc. of ACM Symposium on Applied Computing, 2017.

[14] C. Partridge, S. Nelson, V. Shurbanov, D. Ryder, and B. Thapa, "NDN in Large Detached Underwater Sensing Arrays," in Proc. of IEEE Globecom Workshops, 2016.

[15] G. Xing, Y. Chen, L. He, W. Su, R. Hou, W. Li, C. Zhang, and X. Chen "Energy Consumption in Relay Underwater Acoustic Sensor Networks for NDN," IEEE Access, vol. 7, pp. 42694-42702, 2019.

[16] M. Kuai, T. Haque, X. Hong, and Q. Yu, "A Named-Data Networking Approach to Underwater Monitoring Systems," in Proc. of 10th Int'l Conf. on Underwater Networks & Systems, 2015.

[17] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in Proc. of IFIP Netw. Conf., 2013.

[18] H. Dai, Y. Wang, J. Fan, and B. Liu, "Mitigate DDoS attacks in NDN by interest traceback," in Proc. of IEEE INFOCOM Workshops, 2014.

[19] R. Martin and S. Rajasekaran, "Data centric approach to analyzing security threats in Underwater Sensor Networks," in Proc. of IEEE OCEANS, 2016.

[20] K. Wang, H. Zhou, Y. Qin, J. Chen, and H. Zhang, "Decoupling malicious *interests* from Pending Interest Table to mitigate Interest Flooding Attacks," in Proc. of IEEE Globecom Workshops, 2013.

[21] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating interest flooding DDoS attacks in Named Data Networking," in Proc. of IEEE LCN, 2013.

[22] H. Salah, J. Wulfheide, and T. Strufe, "Lightweight coordinated defense against interest flooding attacks in NDN," in Proc. of IEEE INFOCOM Workshops, 2015.

[23] H. Salah, J. Wulfheide, and T. Strufe, "Coordination supports security: A new defense mechanism against interest flooding in NDN," in Proc. of IEEE LCN, 2015.

[24] R. Martin, S. Rajasekaran, and Z. Peng, "Aqua-Sim Next generation: An NS-3 based underwater sensor network simulator," in Proc. of 10th Int'l Conf. on Underwater Networks & Systems, 2017.

[25] Y. Gao, L. Deng, A. Kuzmanovic, and Y. Chen, "Internet Cache Pollution Attacks and Countermeasures," in Proc. of IEEE ICNP, 2006.

[26] L. Deng, Y. Gao, Y. Chen, and A. Kuzmanovic, "Pollution attacks and defenses for Internet caching systems," Comput. Networks, vol. 52, no. 5, pp. 935-956, 2008.

[27] S. Dibenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in Proc. of IEEE INFOCOM, 2016.

[28] D. Kim, J. Bi, A. V. Vasilakos, and I. Yeom, "Security of Cached Content in NDN," IEEE Trans. Inf. Forensics Security, vol. 12, no. 12, pp. 2933-2944, Dec. 2017.

[29] T. Nguyen, X. Marchal, G. Doyen, T. Cholez, and R. Cogranne, "Content Poisoning in Named Data Networking: Comprehensive characterization of real deployment," in Proc. of IFIP/IEEE International Symposium on Integrated Network and Service Management, 2017.

[30] H. Salah and T. Strufe, "Evaluating and mitigating a Collusive version of the Interest Flooding Attack in NDN," in Proc. of IEEE Symposium on Computers and Communication, 2016.

[31] H. Park, I. Widjaja, and H. Lee, "Detection of cache pollution attacks using randomness checks," in Proc. of IEEE ICC, 2012.

[32] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," in Proc. of IEEE INFOCOM, 2012.

[33] A. Karami and M. Guerrero-Zapata, "An ANFIS-based cache replacement method for mitigating cache pollution attacks in named data networking," Computer Networks, vol. 80, pp. 51-65, 2015.

[34] T. Kamimoto, K. Mori, S. Umeda, Y. Ohata, and H. Shigeno, "Cache protection method based on prefix hierarchy for content-oriented network," in Proc. of IEEE CCNC, 2016.

[35] L. Yao, Z. Fan, J. Deng, X. Fan, and G. Wu, Detection and Defense of Cache Pollution Attacks Using Clustering in Named Data Networks, IEEE Trans. Dependable Secur. Comput., pp. 1-1, 2018.

[36] J. Dong, K. Wang, Y. Lyu, L. Jiao, and H. Yin, "InterestFence: Countering interest flooding attacks by using hash-based security labels," in Proc. of 18th International Conf on Algorithms and Architectures for Parallel Processing, LNCS-11337, 2018.

[37] Q. Li, P. P. C. Lee, P. Zhang, P. Su, L. He, and K. Ren, "Capability-Based Security Enforcement in Named Data Networking," IEEE/ACM Trans. Netw., vol. 25, no. 5, pp. 2719-2730, Oct. 2017.

[38] W. Yang, Y. Qin, and Y. Yang, "Analysis of malicious flows via SIS epidemic model in CCN," in Proc. of IEEE INFOCOM, 2018.

[39] ns-3 — a discrete-event network simulator for internet systems. [Online]. Available: https://www.nsnam.org/. [Accessed: 01-Aug-2019].