IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 16, 2017, accepted September 13, 2017, date of publication September 18, 2017,
date of current version October 12, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2754058

Combining Popularity and Locality to Enhance In-Network Caching Performance and Mitigate Pollution Attacks in Content-Centric Networking

GUOZHI ZHANG[1,2], JIQIANG LIU[1], (Member, IEEE), XIAOLIN CHANG[1], (Member, IEEE), AND ZHI CHEN[1]

[1]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China
[2]School of Computer and Engineering, Northwest Normal University, Lanzhou 730070, China

Corresponding author: Jiqiang Liu (jqliu@bjtu.edu.cn)

The work of G. Zhang was supported by the NSF of China under Grant 61672092 and Grant 61363081. The work of J. Liu was supported by the NSF of China under Grant 61672092. The work of X. Chang and Z. Chen was supported by the NSF of China under Grant 61572066.

ABSTRACT Content-centric networking (CCN) aims to improve network reliability, scalability, and security by changing the way that information is organized and retrieved in the current Internet. One critical issue in CCN is in-network cache allocation. It is known that in-network caching mechanisms are vulnerable to distributed denial of service attacks, especially to pollution attacks. That is, a caching mechanism under pollution attacks cannot work well. The past years witnessed kinds of proposals of cache allocation mechanisms. However, none of them could effectively allocate in-network cache while defending against attacks. In this paper, we propose a lightweight non-collaborative cache allocation approach (IFDD), which could not only enhance in-network caching performance in terms of the cache hit ratio and the request processing delay, but also defend against pollution attacks. By lightweight, we mean that IFDD generates low communication overhead (due to non-collaboration) and computational overhead at routers. The key idea behind IFDD is to combine the content popularity with the content locality in making caching decision. Extensive simulation results on ndnSIM platform demonstrate the capability of the proposed approach in improving cache allocation performance while reducing the impact of pollution attacks.

INDEX TERMS Content-centric networking, caching allocation, pollution attacks, locality, popularity.

I. INTRODUCTION

As a new network paradigm, Content-Centric Networking (CCN) [1] has gained significant attention in the past few years. Benefiting from various attractive advantages (e.g. the name routing and the in-network caching), CCN is considered as a potential substitute for current TCP/IP, especially for mobile scenarios such as Mobile Ad-hoc Networking (MANET) and Vehicular Ad-hoc Networking (VANET), etc. It is regarded as the most competitive network architecture for the future Internet.

CCN is similar to Peer-to-Peer (P2P) networks, Web caching systems, and Content Distribution Networks (CDN) from the viewpoint of caching design. P2P and CDN caching mechanisms are usually only for a single type of application services, and the content provider is responsible for content deployment. But CCN, as a more underlying network paradigm, provides caching services for a variety of applications and requires data to be transferred at line-speed. That is, CCN caching design and resource management are more complex, compared to P2P and CDN.

The existing studies [2]–[4] indicated that caching performance is affected by a variety of factors, including router cache capacity, content/interest distribution, network topology and caching strategy. In the past years, various caching mechanisms [5]–[8] were proposed. Most of them [7], [9], [10] were based the content popularity. Here, the popularity refers to the number of times of requesting the content at a router per unit time. For a specific content at a specific router in CCN, the corresponding popularity can be obtained by counting the total number of requests for this content on this router. In the rest of the paper, unless otherwise specified, the popularity of a content is called *the* popularity. When popularity-based cache allocation mechanisms are deployed, the higher the content popularity, the higher caching priority the content.

Studies [11]–[13] indicated that there is the significant difference in consumers' content interests among some large and heterogeneous communities owing to consumers' community attributes and geographic locations. That is, the probability of requesting a given content may vary significantly from

region to region. We use locality to denote the feature of this difference in this paper, namely, the degree of locality dispersion. The higher the locality, the larger the difference in the number of times of requesting this content per unit time among the interfaces of this router.

The feature of locality was ignored in caching strategies on CDN [14], P2P [15], Web caching system [16] and Social Network (SN) [17], [18]. These strategies usually only considered the content popularity. The existing caching strategies in CCN also only exploited the content popularity without the consideration of the *locality*. Traverso *et al*. [19] discussed that the locality is hardly observable even within culturally homogeneous regions, but it could be observed partly in large systems serving different linguistic/cultural communities and even in limited geographical regions.

This paper explores how to combine the popularity and the locality to design cache allocation mechanisms. A novel cache allocation algorithm, named IFDD (Interest-interface Dynamic Degree) based cache allocation, is proposed. In IFDD, the popularity and the locality are both calculated at each router according to the counting of content interests. There is difference in the calculation of these two features. The popularity is obtained only by counting the interests of the content at the router. But the computation of locality still needs counting the content interests at each interface of this router. More details are given in Algorithm1 and Algorithm2 of Section III. With its own calculated popularity and locality, every router could makes caching decision independently with Multiple Attribute Decision Making (MADM) algorithm [20].

The major contributions are summarized as follows:

1) We study a pollution attack in the scenario where popularity-based caching strategy is deployed. This attack may seriously damage the caching advantage and greatly increase the processing delay of consumer requests. The simulation results show that the system under the existing popularity-based cache allocation mechanisms, including simple random strategy (e.g. LCE [21]), is vulnerable to this attack and then these mechanisms could not work effectively.

2) We propose a lightweight non-collaborative cache allocation approach (IFDD). Lightweight means that when IFDD strategy is deployed, not only lower communication overhead is produced but also small calculation overhead is required at routers. The first small overhead is because no cooperation is required among routers. The latter one is because the calculation is completed at each router in the linear polynomial time. To the best our knowledge, IFDD is the first caching approach which integrates the popularity and the locality in making caching decision. Moreover, IFDD is the first caching approach which could not only enhance caching performance, but also defend against the pollution attack. Note that this paper evaluates performance in terms of request processing delay and cache hit ratio.

3) We carry out simulations for evaluating IFDD on ndnSIM platform by varying kinds of system parameters. We found that both the network topology itself and the interest distribution have a considerable impact on network performance. The simulation results suggest that an effective network cache allocation mechanism should consider various system parameters, such as application types and network topology.

The rest of the paper is organized as follows. Section II presents the related work about popularity-based caching strategy and presents the existing locality studies. Section III presents the details of IFDD mechanism. Section IV describes the simulation results to evaluate the performance of our design on ndnSIM platform. Section V concludes the paper and discusses future work.

II. RELATED WORK

In CCN architecture, a router maintains three main components: Pend Interest Table (PIT), Content Store (CS) and Forward Information Base (FIB). PIT is used to record the unsatisfied interest prefix and the incoming interface index. It could provide a reference interface to the returned content. CS is a key component to cache the returned content and to satisfy the later interest request with the same prefix. The router forwards interests according to FIB information, which is calculated by the routing algorithm.

CCN adopts the ubiquitous caching mechanism to enhance the performance in terms of decreasing the application request processing delay and increasing the cache hit ratio. Thus, the caching strategy is vital for this architecture. A caching strategy consists of two functional modules: allocation and replacement. The former addresses two issues: *WHAT* content should be cached, and *WHERE* a caching location is. The latter is used to determine what contents should be replaced when the cache is full. More details about CCN and cache replacement mechanisms are referred to [1]. The following focuses on the cache allocation strategy.

Most existing strategies in traditional networks adopted the simple random way to determine whether the content should be cached. These strategies include Leave Copy Everywhere (LCE), Leave Copy Down (LCD), Move Copy Down (MCD), Leave Copy Probability (LCP), Randomly Copy One (RC One), Probabilistic caching (Prob Cache) etc. LCE is the default strategy which is available on the ndnSIM platform. In this strategy, replicas of the content are cached on all routers in the path during the transfer of content from producers to consumers.

The caching strategy is closely related to various factors. These factors can be divided into two categories. One is related to the dynamic characteristics of the network, such as quality of service (QoS), router cache capacity, network topology and so on. The other is related to the characteristics of interests.

Based on these factors, the existing caching strategies could be divided into two categories: the collaborative approach and the non-collaborative approach. The collaborative approach usually adopts the network characteristics as the design factors and formulates cache allocation into a

global optimal/suboptimal problem. This approach pursues the overall optimal/suboptimal caching performance usually with higher decision cost. Therefore, some researchers thought that this overhead may offset the benefits of network caching [22]. The non-collaborative approach is mainly based on the simple random caching method or based on statistical method to independently make caching decision. In general, simple random approaches, while with low overhead, may result in a large amount of redundancy in the cache. Thus, they may not be suitable for highly dynamic CCN environments.

The requests of consumers for contents usually conform to a specific statistical model. For example, the Web request frequency on Internet conforms to a similar Zip-like probability distribution model (e.g [23]). This phenomena was also found in the other types of networks including CDN [14], [24], P2P [15], [25] and etc. The existence of such distribution model suggests that popularity could be an important factor in cache allocation. Many popularity-based caching strategies were proposed in the past years. See [7] and references therein. Wang *et al*. [7] proposed a sub-optimal heuristic algorithm based on the popularity and the network node degree. Their simulation results under the Zipf-like distributed data set indicated that both the topology and the popularity of the content are important factors affecting the effectiveness and efficiency of content caching.

Meanwhile, many attack models have been found in CCN. Among of them, pollution attack is a most important one that is a paradigm of Distributed Denial of Service (DDoS). This kind attack works mainly by sending invalid or illegal requests to occupy component resources, such as PIT and CS, leading to the failure of these components. When CS is attacked, attackers can make the router cache filled with the content that is not consistent with the goal of the caching strategy. Under this attack, the caching strategy may fail. Therefore, the network performance is compromised.

Pollution attack is not unique to CCN, and it has been extensively studied in P2P and the other networks. Dhungel *et al*. [26] evaluated the applicability of some possible defenses to the pollution attack, and found that the attack can be devastating. In CCN, Ribeiro *et al*. [27] proposed a mechanism to defend against the pollution attack in which routers randomly checked the content signatures. Guo *et al*. [28] proposed an approach exploiting the diversity of the interest traversing paths within an ISP's point-of-presence network to detect and mitigate the pollution attack.

Our work is close to Guo *et al*. [28]. However, there are two major differences: (1) Besides the capability of defending against attacks, our approach is effective in caching. (2) Our proposed approach is low-cost and non-collaborative, significantly reducing the high *temporal* complexity which usually exists in a collaborative approach. For example, a solution based on complex network theory, or a scheme based on overall network collaboration [7].

III. INTEREST INTERFACE DYNAMIC DEGREE BASED CACHING DECISION STRATEGY

In this section, we first discuss the challenges in designing a caching decision approach. Then we detail how IFDD achieves effective cache allocation and why it has immunity to pollution attacks.

A. CHALLENGES IN CACHE ALLOCATION DECISIONS

From the performance viewpoint, the core of cache allocation decision in CCN is "caching the *specific* content to the *appropriate* location". The solution to this problem faces two challenges:

1) WHAT KIND OF CONTENT SHOULD BE CACHED?

Obviously, the best way is to cache those contents that are most likely to be requested by the other consumers in the future. But it is hard to predict what content will be requested in the future. The popularity and the locality have been used to decide what kind of content to be cached in CDN (e.g. YouTube [12]). However, it is difficult to implement these CDN solutions directly in CCN. There are two major reasons as follows: (i) The CCN network has no centralized control mechanism to calculate the popularity and the locality. (ii) The cache in CCN is more common, dynamic and diverse than in CDN. By common, we mean that caching, the basic network function in CCN, should support all kinds of applications and all users, unlike CDNs only for certain applications and specific users. Therefore, the caching methods proposed in both CDN and similar networks are not fully applicable to CCN.

2) WHERE THE CONTENT SHOULD BE CACHED

The main purpose of in-network caching is to improve the reusability of the content, reduce the processing delay of the consumer request, and save the network bandwidth. A typical way is to save bandwidth overhead but with caching costs. CDN uses the centralized control method to collect the data, calculate the popularity of content, and then coordinate the data distribution on each router. This method cannot be applied to CCN because CCN routers are further away from each other than CDN. Therefore, the overhead of using this method may offset the benefits of in-network caching in CCN.

From the point of security view, the CCN caching mechanism should defend against cache attacks which may disable the caching mechanism. When the CCN architecture was put forward, some researchers (e.g. [29]) pointed out that the CCN may face DDoS attacks, especially pollution attack. The ubiquitous caching mechanism of CCN makes this attack more common and serious.

In the CCN architecture, there are two pollution attack targets: PIT and CS. The PIT attack is to send many invalid requests, so that these requests pending entries for a long time cannot get a response, and eventually exhaust

PIT resources. For this attack, the scholars in [30] and [31] proposed solutions. Our work is mainly for a solution to CS attack. Some scholars [32], [33] responded to this attack by assuming that a mechanism for detecting pollution attacks is ready. We try to solve this problem from another point of view, namely, applying the locality. The advantage of our method is that there is no need for a separate detection mechanism. Thus, our strategy has the natural ability to prevent pollution attack.

Note that the strategy proposed in this paper is mainly used to defend against those attacks which aim for popularity-based caching strategies. In this attack model, malicious consumers send a lot of requests for non-popular contents, leading to that the cache is depleted by non-popular contents, and eventually leading to the caching strategy failure. Our simulation results validate the existence of this attack and verify that our approach has a certain immunity to this attack.

B. POPULARITY AND LOCALITY COMPUTATION

This subsection presents how to compute the popularity and the locality to be used in IFDD. The symbols and notations to be used latter are given in TABLE 1.

TABLE 1. Term definitions.

Notation	Description
a	The Prefix of a Content or Interest
$I(a)$	The Interest for Prefix a
$C(a)$	The Content Named a
$Provider(a)$	The Content Provider for Prefix a
$Consumer(a)$	The Content Consumer for Prefix a
R_x	The Router Named x
RF_x^i	The Interface i on R_x
$\mid RF_x \mid$	The Interface Count on R_x
$Stat_x(a)$	The Stat of Prefix a on R_x
$Stat_x^i(a)$	The Stat of Prefix a at RF_x^i on R_x
$P_x(a)$	The Popularity of $C(a)$ on R_x
$Set_x(a)$	The Decision Attribute Set on R_x
$SD_x(a)$	The Standard Deviation of $I(a)$ on R_x
$CV_x(a)$	The Coefficient of Variation of $I(a)$ on R_x
$Degree(x)$	The Degree of R_x
$Capacity(x)$	The Capacity of Cache Store of R_x

The major steps for a CCN router to process interest/content are as follows: When the router R_x receives the interest $I(a)$ at the interface RF_x^i, the prefix a is firstly recorded in PIT. Then R_x searches PIT and judges whether prefix a does exist or not. If a does not exist, R_x creates a PIT entry for $I(a)$ and adds the incoming interface to PIT entry, then forwards $I(a)$ based on FIB. If a exists, the router adds the incoming interface into the PIT entry for interest aggregation. Following this process, when a duplicate interest is received, the router simply discards or aggregates it.

As mentioned earlier, both the popularity and the locality are used to help caching decisions. The popularity $P_x(a)$ can be easily obtained by counting interest with prefix α on R_x. In IFDD, $P_x(a)$ is calculated by using the mechanism proposed in [7]. We now detail how to calculate the locality. We propose to use the *Coefficient* of *Variation* (denoted as $CV_x(a)$) to quantify the locality of content $C(a)$ on R_x. $CV_x(a)$ aims to measure the dispersion of all interest for $C(a)$ over all the interfaces on R_x and is defined as follows:

$$CV_x(a) = \frac{SD_x(a)}{mean_x(a)}$$

where

$$SD_x(a) = \sqrt{\frac{1}{|RF_x|}\sum_{i=1}^{|RF_x|}(Stat_x^i(a) - E_x)^2}$$

is standard deviation of the prefix a on R_x, and

$$mean_x(a) = \frac{1}{|RF_x|}\sum_{i=1}^{|RF_x|} Stat_x^i(a)$$

is mean of the prefix a value on R_x. $CV_x(a)$ could reflect the degree of equilibrium distribution of all interests with prefix a for the content $C(a)$ on R_x. In IFDD, the lower $CV_x(a)$, the higher the locality. Therefore, content $C(a)$ with lower $CV_x(a)$ is more suitable for caching on R_x.

C. Popularity and Locality Computation

This subsection presents the IFDD algorithm and the concrete algorithm of MADM used in IFDD.

The caching decision strategy can be briefly described in three steps: (1) Building statistic table according to the received interest, (2) calculating the fitness ($CV_x(a)$, $P_x(a)$) of content based on $Stat_x(a)$, and (3) making caching decision using MADM algorithm with ($CV_x(a)$, $P_x(a)$).

Algorithm 1 The Process of Router R_x After Receiving an Interest $I(a)$

Require: $I(a)$, RF_x^i
Count $map < I(a), RF_x^i >$ into $Stat_x(a)$
if a is in PIT **then**
  $Aggregrating(a)$ into PIT
  return
else if $Lookup(a)$ in FIB is *true* then
  $Forward(a)$
  $Record(a)$ in PIT
end if

Algorithm 1 and Algorithm 2 describe IFDD algorithm. When R_x receives $I(a)$, it counts the related information of $I(a)$ into $Stat_x(a)$. This information includes prefix, frequency, entering interface, etc. $Stat_x^i(a)$ is to be used in the

Algorithm 2 The Process of Router R_x After Receiving a Content $C(a)$

Require: $C(a)$, $Stat_x^i(a)$
1: **if** a is in PIT **then**
2: $Discarding(C(a))$
3: return
4: **end if**
5: $P_x(a) = Popularity(a)$
6: $CV_x(a) = Stat_x^i(a)$
7: $DecisonResult = MADM(P_x(a), CV_x(a))$
8: **if** $DecisonResult = TRUE$ **then**
9: $Caching(C(a))$
10: $Satisfy\_PIT(a)$
11: **end if**

next process. When R_x receives $C(a)$, it first looks for prefix a at PIT. If not found, $C(a)$ is simply discarded, otherwise proceeds to the next step. After this, R_x forwards $C(a)$ while executing the caching decision to decide whether to cache $C(a)$ on R_x. After searching PIT, $C(a)$ needs to be forwarded if at least one interface RF_x^i receives $I(a)$. In order to judge whether $C(a)$ is cached, the suitability of $C(a)$ cached on R_x needs to be calculated. $CV_x(a)$ of the distribution of the content in each interface is calculated firstly, and then $P_x(a)$ is also obtained. After the above process, the attribute set $Set_x(a) = \{CV_x(a_i), P_x(a_i)\}$ is obtained as the basis of making decision on R_x.

Note that the centrality degree of the router ($Degree_x$), the capacity of the caching store ($Capacity_x$) and the other factors can also be added to make decision. Here, we only use $CV_x(a)$ and $P_x(a)$ as the attributes since IFDD is designed to be a bootstrap and without the aid of the other routers. We will consider these factors in our future research.

IFDD applies MADM [34] to determine their weights. MADM is usually used to perform multi-criteria decision making, having the following advantages: (1) The weight of decision attributes can be determined objectively, (2) the dimensions of each attribute can be different, and (3) allowing each node to participate in decision-making properties or the ratio between them is not the same. Zanakis *et al.* [35] made a comparative analysis of some concrete algorithms of MADM and discussed the characteristics of these algorithms. Here, we present the concrete algorithm of MADM, which is used in IFDD.

Assuming R_x receives n contents in unit time. However, limited by the cache capacity, only $m(m < n)$ contents can be cached. So, R_x should select most appropriate m contents.

1) The decision-making matrix D_x in R_x is defined as follows:

$$\mathrm{D}_x = \begin{pmatrix} CV_{\mathrm{x}}(a_1) & P_{\mathrm{x}}(a_1) \\ \vdots & \vdots \\ CV_x(a_n) & P_{\mathrm{x}}(a_n) \end{pmatrix}$$

NSD_x is defined to denote the normalized and standardized matrix of D_x. The definition is as follows:

$$\mathrm{NSD}_x = \begin{pmatrix} r_{1,1} & r_{1,2} \\ \vdots & \vdots \\ r_{n,1} & r_{n,2} \end{pmatrix}$$

Here, $r_{i,1} = \frac{1/CV_{\mathrm{x}}(a_i)}{1/\max\limits_{i=1,2,\ldots n} CV_{\mathrm{x}}(a_i)}$, $r_{i,2} = \frac{P_{\mathrm{x}}(a_i)}{\max\limits_{i=1,2,\ldots n} P_{\mathrm{x}}(a_i)}$.

2) We use the *information entropy method* to get the weight of attributes. The column vector of NSD_x is (A_1, A_2). Then, the entropy of A_1 and A_2 for attribute CV_X and P_x will be $E_j = -k \sum_{i=1}^{n} r_{ij} \ln r_{ij}$, $k = 1/\ln n, j = 1, 2$. Define $F_j = 1 - E_j$, then the weight of attribute j will be $w_j = F_j/(F_1 + F_2)$.

(3) We use *simple additive weighting* to get the decision value. The decision of each content will be:

$$v_i = r_{i1}w_1 + r_{i2}w_2, \quad i = 1, 2, \cdots, n$$

Finally, top m contents of v_i will be cached.

D. EFFECTIVENESS OF IFDD IN CACHING DECISION

Some researchers pointed out that the CCN topology is different from traditional networks. However, most CCN studies assume that the CCN topology is a hierarchical tree structure, and if not, it can be converted to a tree structure. Tree topology can simplify the network design. For example, eliminating the loop and then the broadcast storm is avoided. In addition, this makes some spanning-tree algorithms widely used in the network, such as minimum spanning tree (MST) and shortest spanning tree (SPT). In CCN, one of the advantages of a tree topology is that it can be used to determine the interface of interest/content source. If consumers (e.g. communities) are connected to one gateway, they will send all requests to router R_x through the same interface in the tree topology.

Even if the topology is not a tree or hierarchical structure, a good routing algorithm should maintain the stability of routing and avoid routing floating. Interests generated by a particular community/region are different from that of the other communities [28]. That is, the interest requests are clearly imbalanced in geographical distribution. All interfaces on the router can clearly see this feature. In our approach, the popularity of the content is obtained in this direction by counting the interest of the same prefix on an interface. Further, if these interests are mainly sent by a community/region, $CV_x(a)$ of the interest prefix will be very different from those sent by the relatively dispersed consumers.

We use an example to illustrate how the locality is involved in caching decisions. In this example, the working principle of IFDD is compared with LCE and the other popularity-based strategy.

In Figure 1, we assume that consumers connected to $R1$, $R2$, $R4$, $R6$ send same requests $I(a)$ to the provider that is connected to $R11$. $C(a)$ has a higher access probability on routers along the path $R5$ to $R11$. If we use LCE as the caching strategy, $C(a)$ will be cached in each router along the path $R5$ to $R11$. Furthermore, even if the popularity based

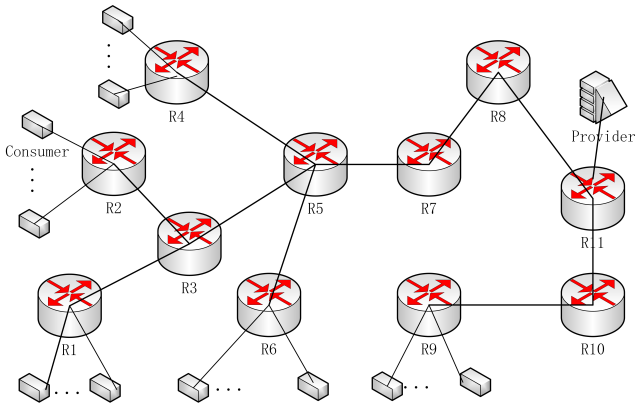

FIGURE 1. A simple example for our core idea.

strategy is used, content $C(a)$ will be cached on most routers along the path from $R5$ to $R11$ since it has higher popularity than other contents. Obviously, this method causes a lot of redundancy. Especially for some routers, it is not entirely reasonable. For example, for router $R7$ and $R5$, if the content $C(a)$ is more popular than the others, and consumers of the content are mostly at downstream routers of the path, the content should be cached on these routers. In this case, $R5$ is the best caching location because if we do not consider the other metrics, $R5$ will satisfy most of the $R1$, $R2$, $R4$, $R6$ requests. In the same way, $R3$ is the best router if only $R1$ and $R2$ request content $C(a)$. In addition, taking into account the content popularity and the probability of request, if $R1$ and $R2$ have higher probability to request the content than the others, $R5$ does not need to cache the content though $R4$ and $R6$ are also possible to request $C(a)$.

In an extreme case, when the distribution of interest is completely free of the locality, the requests follow the uniform distribution on the network. Now, the effectiveness of the algorithm is equivalent to that of the popularity-based strategy. On the other hand, when the locality is more obvious, the benefits of the algorithm will gradually increase.

E. The Effectiveness of IFDD against Pollution Attacks

Assume the capability of a group of attackers is represented by $A(G) = \{S(\delta t, N(G), D(G))\}$, where G is a group of attackers, $S(\delta t)$ is the number of requests sent by G per unit of time t, $N(G)$ is the number of nodes in group G, and $D(G)$ is the degree of dispersion of G geographic diversity. If attackers G want to successfully achieve the pollution attack, they must have enough strength in these three aspects: $S(\delta t)$, $N(G)$, $D(G)$. There is no doubt that if the attacker has such ability, our strategy cannot resist pollution attack. Fortunately, it is very difficult for an attacker to have such ability, especially in the third factor $D(G)$. There are significant differences in the complexity of pollution under different caching strategies. We will discuss in the following three different cases:

Case 1: LCE or Similar Strategies: When LCE or a similar strategy is deployed, attackers could attack successfully with only certain $S(\delta t)$. The ability of $S(\delta t)$ enables attackers to send a lot of fake interest to the network. If there are some malicious providers cooperating the attack, they will respond to this interest with fake content. The content will invade and occupy CS along the path from the consumer to the provider. Therefore, in this case it is easy to pollute the cache.

Case 2: Popularity-Based Strategy: When popularity based strategy is used, same as in Case 1, they only have enough $S(\delta t)$. However, in this case, the demand for this ability is much higher than in ***Case 1***. Malicious consumers must send enough invalid interests to achieve a successful attack. The frequency of these requests must be high enough to make the content more *popular*. This case is more difficult than in ***Case 1***. However, it also could be executed effectively if attackers have the ability of $N(G)$.

Case 3: IFDD Strategy: If attackers want to attack, at least two requirements are met: (1) Attackers frequently request the fake content, making it popular, and (2) the geographic location of attackers is sufficiently extensive so that the interest has sufficient degree of dispersion. Indeed, if the attacker has reached these two requirements, our strategy may fail. However, the situation is better than expected. The reason is that the locality on each router is not consistent. Actually, there may be only a small number of routers affected by attacks. In fact, it is difficult for an attacker to want the valid content with high popularity and low locality on most routers at the same time. If an attacker wants to meet this requirement, he must work closely with many of the consumers scattered across the network. If the attacker has such capacity, the entire CCN architecture under the network caching mechanism will face a great security challenge. For such scenario, it is necessary to re-examine the severity of this attack.

IV. SIMULATION AND PERFORMANCE EVALUATION

This section first presents simulation setup. Then IFDD is compared with the other strategies, i.e. LCE and naive popularity-based strategy under various system parameters. The considered parameters include interest distribution, replacing strategy and network topology. In the following, LRU is used as the default replacement mechanism. The metrics considered include cache hit ratios and application request processing delay. We also evaluate IFDD's capability in being resistant to pollution attack.

A. SIMULATION SETUP

We implement the algorithm on ndnSIM platform [36], [37], which is an open experiment environment based on NS3 [38]. The operating system used is ubuntu-12.04-lts. Four real network topologies [39] are used in our simulation, shown in Figure 2 (a)(b)(c)(d) and indexed as *1755, 2914, 3967* and *7018*, respectively. They cover the typical topologies in terms of the number of routers and the topology structure.

It is well known that the traffic model is vital for simulation. Our investigation indicates that there is no public data set for CCN simulation. Many researchers realized the problem and proposed some traffic models either

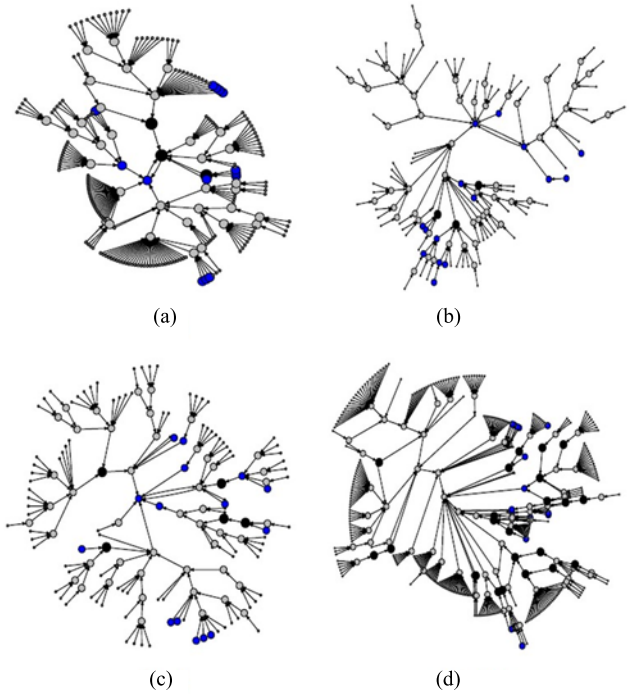

FIGURE 2. The simulation topology. (a) 1221. (b) 1755. (c) 3967. (d) 7018.

theoretically or through experiments. Traverso *et al.* [19] studied the impact of the popularity and the locality under different cache replacement strategies in content caching systems, and introduced a traffic model named *SNM*. *SNM* is different from early traditional caching traffic models. It is a more general and flexible model which can be extended for different requirements of popular caching strategies. Their work provided effective tools for building simulation scenarios. SNM is an important reference model when we design our simulation traffic model.

Generally, consumers and producers are on the edge of the network. Routers are on the core of the network. We follow such scenario and assume that requests for a content conform the Zipf distribution in a region (such as an AS): $P(x) = C/\gamma^{\alpha}$, where α reflects the concentration of the distribution. The higher the α, the more concentrated the content. Note that our proposed mechanism does not assume any specific interest distribution. We compare IFDD, Popularity-based strategy and LCE by setting $\gamma = 0.7$ and varying α from 0.25 to 1.5.

Figure 3-6 show the simulation results in scatter and box plots, in order to intuitively reflect the distribution of cache hits, and get more information from diagram. Each point in the figures represents the number of cache hits in the router cache during the access. For the fixed total number of requests, the cache hit and the cache hit ratio reflect the same thing.

B. THE IMPACT OF INTEREST DISTRIBUTION

This subsection aims to investigate the impact of interest distribution on strategies, that is, the impact of cache size and caching strategy, on cache hits under different parameters of the interest distribution model.

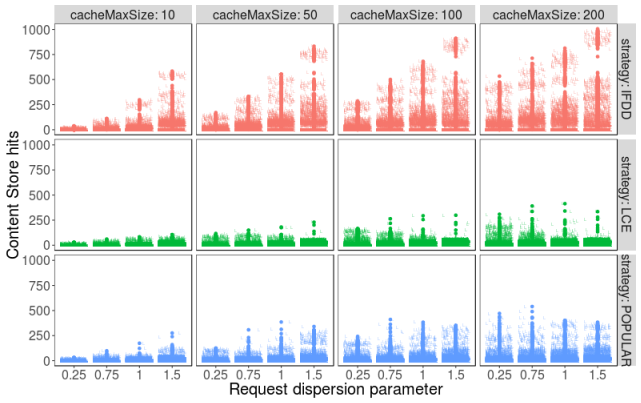

FIGURE 3. Interest Dispersion versus Cache hits under LRU.

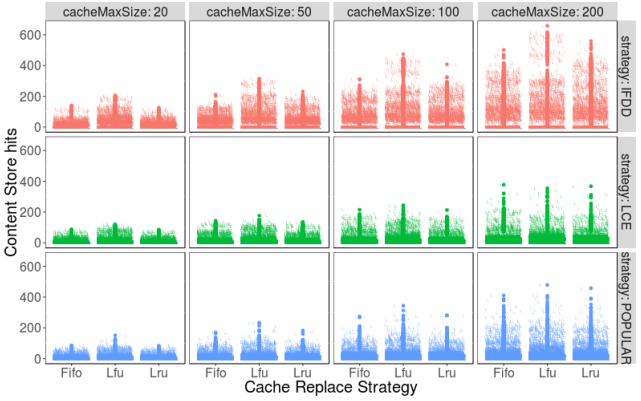

FIGURE 4. Replacement Strategy versus Cache hits.

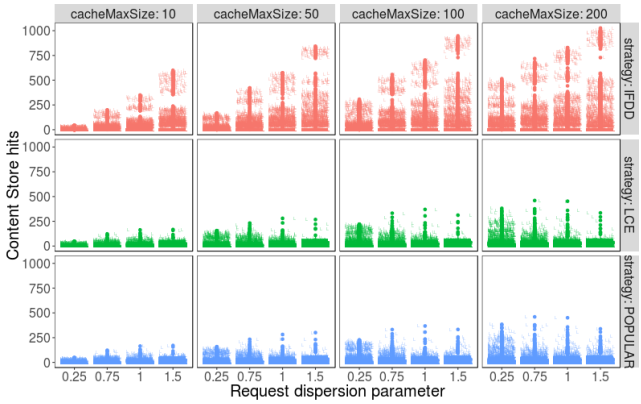

FIGURE 5. Interest Dispersion versus Cache hits under Lfu.

Simulation results in Figure 3 show that when using IFDD, the cache hit is higher than the other strategies. We also observe that, regardless of LCE or IFDD or POPULAR strategy, the cache hit with the increasing cache capacity increases significantly. However, the effect of the increase under IFDD is more significant. This suggests that the IFDD strategy is more sensitive to the increase in cache capacity.

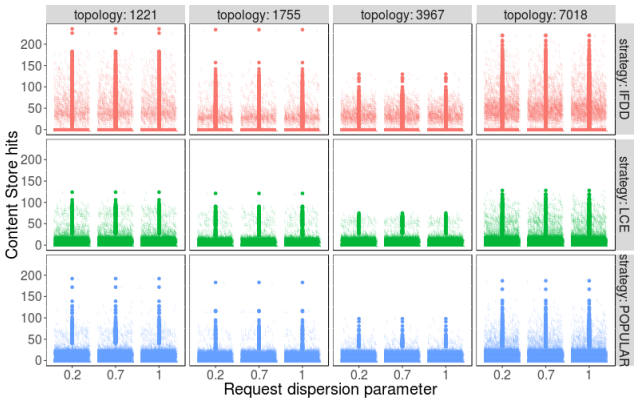

FIGURE 6. Topology versus Cache hits.

Fayazbakhsh *et al*. [22] found that the content should be cached in the network core routers, but Psaras *et al.* [40] came to the opposite conclusion. Our simulation results indicate that the cache hit is very sensitive to the degree of dispersion of interest, especially in the IFDD. With the increase in the locality of interest, replicas are more centralized towards the edge of the network when IFDD is applied. With the α reduction, interests tend to be scattered. In this case, the probability that the content is cached to core routers increases and the caching strategy will gradually fail.

Although the individual router cache hits may be reduced due to the decrease in the number of replicas, in general significantly higher than LCE and POPULAR. When α is increased, IFDD will have a higher cache hits than the others. The simulation results demonstrate the effectiveness of our algorithm.

C. THE IMPACT OF CACHING REPLACEMENT STRATEGY

This subsection aims to investigate the effect of cache replacement on the performance of caching allocation mechanisms. Three common cache replacement strategies are considered: LRU (Least Recently Used), LFU (Least Frequently Used) and FIFO (First-In-First-Out). Evaluations are carried out under different cache capacities. Figure 4 shows the results. We observe that:

1) Different allocation strategies show significant differences in performance under the same capacity. IFDD produces the best cache hits, and the cache capacity does have a critical impact on the cache hits. The cache hit increases when the cache capacity is increasing. This suggests that IFDD is more sensitive to cache capacity in terms of the cache hit, which is exactly what we are pursuing in the caching design.
2) The performance difference of the three replacement strategies is not obvious. As the default replacement strategy in ndnSIM, LRU does not show a more obvious performance than a simple FIFO. In other words, the effect of the replacement strategy on caching performance is much smaller than the allocation strategy. It seems that the replacement strategy designed for the traditional caching mechanism may not be suitable for the CCN, and we may need to design a new replacement strategy for CCN. Our experiment results confirm that the design of the caching strategy is very meaningful for CCN in-network caching mechanism. We also repeat the simulations of Section IV.B but under Lfu. Figure 5 shows the results, indicating that there is less performance difference between Lfu and LRU.
3) For IFDD and POPULAR, the cache hit is higher under Lfu than under the other two replacement strategies. But for LCE, the effect of a replacement mechanism is less. One possible reason is that LCE is not related to the content popularity. When the router caches a content according to a decision algorithm, this content is not necessarily the most popular. Therefore, if the replacement strategy is LFU-like, the content may soon be replaced, making the allocation strategy less effective. This suggests that there may be a close relationship between the cache allocation strategy and the replacement strategy. It may be better to design the cache replacement strategy and the allocation strategy together for much better network performance in term of cache hit ratio. We leave such investigation and design for future work.

D. The Impact of Topologies

In a small-scale network, the content is less and the distance of routers is relatively closer. In this case the caching strategy cannot play an effective role. When the network size becomes larger and the distance between the consumer and the provider increases, the network cache is far from satisfying the needs of the content. Then the caching decision strategy gradually highlights the advantages.

In the selected four topologies, the number of nodes is 278 (Topo-1221), 162 (Topo-1755), 920 (Topo-3697), 624 (Topo-7018) respectively. Among them, Topo-1755 and Topo-3697 are similar. They both have long request paths and smaller node degrees. But Topo-1221 and Topo-7018 have higher node degrees.

In Figure 2, blue nodes represent content providers which are randomly selected. Black nodes are gateway nodes and gray nodes are trunk nodes. Consumers are at the edge of network, and small dots are used to represent them. The simulation compares the cache hit ratio among the four topologies respectively.

The simulation results in Figure 6 show that under different topologies, the cache hit is directly related to the concentration of interest. However, the cache hit in both Topo-1221 and Topo-7018 is significantly better than in Topo-1755 and Topo-3697. The main reason is that the node degree is higher in Topo-1221 and Topo-7018.

The result shows that the caching strategy has a certain dependence on the network topology. For those topologies which have relatively higher node degree, the caching strategy can achieve better performance. This further confirms that node degree could be used as the basis for heuristic algorithm, which was also used in [7].

E. THE CACHING PERFORMANCE UNDER POLLUTION ATTACK

As mentioned previously, one of the advantages of our approach is its ability in being immune to the cache pollution attack. We design a simulation scenario to verify the effectiveness of our approach in defending against the pollution attack.

This simulation assumes that there exists a certain percentage of malicious consumers and producers. The locations of these nodes are completely random. The proportion of the total number of malicious nodes accounted for 10%. In the absence of the attack, these nodes are not different from the normal nodes.

In our simulation, the attack starts at the 20th second from the beginning of the simulation and lasts 10 seconds. These malicious nodes send requests at a higher rate twice than normal nodes. That is, these contents are more "popular" so that they have a higher caching potential. Malicious providers respond to these requests with useless contents. These contents are more likely to occupy the cache to achieve the goal of the attack.

Note that PIT attack (See in [30]) is much easier to be detected than CS attack. When the PIT is attacked, the interest requests from attackers will occupy the PIT entry to achieve the purpose of the attack, even requests are not satisfied. That is, we can find these pollution attacks by examining the satisfaction rate of these PITs.

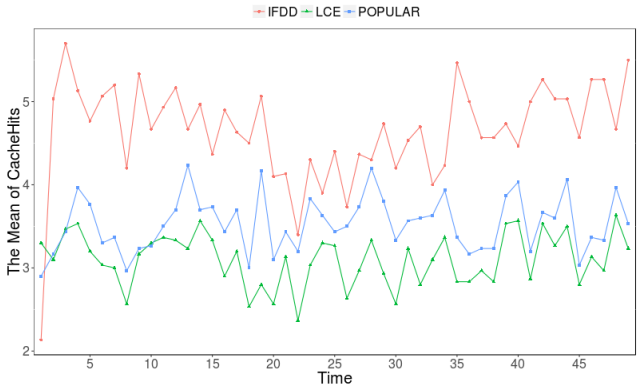

FIGURE 7. Time versus Cache hits under Cache Pollution.

However, it is much more difficult to detect CS attack than PIT attack. Figure 7 indicates that, under either LCE or POPULAR, when the attack occurs during the period 20-30, the cache hit does not decline significantly. In this case, if some malicious providers cooperate with the attacker, the attack will cause CS to cache a large amount of useless content. At the same time, the cache hit may not fall, and may even be higher. So, we cannot, from the cache hit, determine whether the content is from the attack or not. This result indicates that for CS pollution attack, immune mechanisms are less costly and easier to implement than detection and recovery mechanisms. IFDD could achieve such immunity.

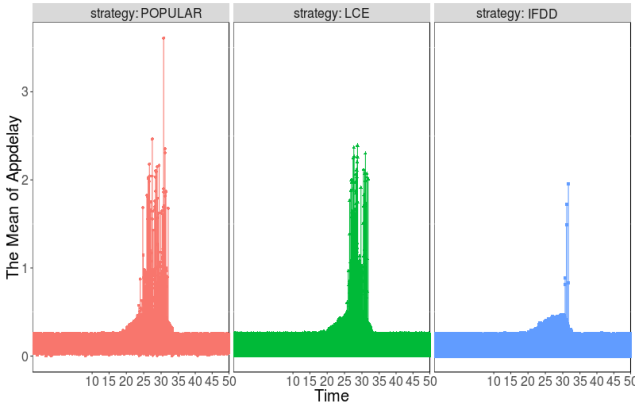

FIGURE 8. Time versus interest request delay under cache pollution.

Figure 8 shows that CS attack could lead to the great increase in the request processing delay of the normal consumers. In addition, when the attack is over, the request delay quickly restores to normal. Obviously, IFDD can effectively reduce the harm caused by this attack.

V. CONCLUSION

This paper explored the combination of content popularity and locality to design a lightweight non-collaborative cache allocation approach. The proposed approach could significantly improve caching performance in terms of decreasing dramatically request processing delay and increasing cache hit ratio. Meanwhile, it also could defend against pollution attacks from malicious nodes. The simulation results verify the effectiveness and efficiency of IFDD.

Future work includes the analysis of the network cache attack models. In addition, we plan to investigate the potential relationship between the cache replacement and the caching decision strategy in CCN for better caching performance. Note that this paper only investigates the effect of the traditional cache replacement strategies, which were not designed for CCN. In the future work, we plan to explore whether we could design an appropriate cache replacement mechanism which could cooperate with cache allocation mechanism to improve CCN performance. Note that MADM is a very basic tool to solve the problem under question. This paper exploited a concrete implementation of MADM in IFDD. Future work will investigate the effect of different concrete algorithms of MADM on the effectiveness of IFDD.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. Int. Conf. Emerg. Netw. Experim. Technol.*, 2009, pp. 117–124.
[2] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Comput. Netw.*, vol. 57, no. 16, pp. 3128–3141, 2013.
[3] I. Abdullahi, S. Arif, and S. Hassan, "Survey on caching approaches in information centric networking," *J. Netw. Comput. Appl.*, vol. 56, pp. 48–59, Oct. 2015.
[4] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.

[5] D. D. Hernandez, J. G. Reinoso, and I. Vidal, "SFP: Statistical filtering policy for caching in content-centric networking," *Comput. J.*, vol. 58, no. 8, pp. 1763–1775, 2015.
[6] K. Kvaternik, J. Llorca, D. Kilper, and L. Pavel, "A methodology for the design of self-optimizing, decentralized content-caching strategies," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2634–2647, Oct. 2015.
[7] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 95–107, Jan. 2016.
[8] C. Bernardini, T. Silverston, and O. Festor, "MPC: Popularity-based caching strategy for content centric networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2013, pp. 3619–3623.
[9] J. Li *et al.*, "Popularity-driven coordinated caching in named data networking," in *Proc. 8th ACM/IEEE Symp. Archit. Netw. Commun. Syst.*, Oct. 2012, pp. 15–26.
[10] W. Li, Y. Li, W. Wang, Y. Xin, and T. Lin, "A popularity-driven caching scheme with dynamic multipath routing in CCN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 633–638.
[11] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft, "Track globally, deliver locally: Improving content delivery networks by tracking geographic social cascades," in *Proc. 20th Int. Conf. World Wide Web*, Mar./Apr. 2011, pp. 457–466
[12] A. Brodersen, S. Scellato, and M. Wattenhofer, "YouTube around the world: Geographic popularity of videos," in *Proc. Int. Conf. World Wide Web*, 2012, pp. 241–250.
[13] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of Facebook photo caching," in *Proc. 24th ACM Symp. Oper. Syst. Principles*, 2013, pp. 167–181.
[14] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Comput. Commun.*, vol. 25, no. 4, pp. 376–383, 2002.
[15] D. Xu, S. S. Kulkarni, C. Rosenberg, and H. K. Chai, "Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution," *Multimedia Syst.*, vol. 11, no. 4, pp. 383–399, 2006.
[16] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the placement of Web server replicas," in *Proc. IEEE INFOCOM*, vol. 3. 2001, pp. 1587–1596.
[17] G. Liu, H. Shen, and H. Chandler, "Selective data replication for online social networks with distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 8, pp. 2377–2393, Aug. 2013.
[18] C. Kong and X. Cao, "Disseminating authorized content in interest-centric opportunistic social networks," in *Proc. ICCCN*, Aug. 2015, pp. 1–8.
[19] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Unravelling the impact of temporal and geographical locality in content caching systems," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1839–1854, Oct. 2015.
[20] G. H. Tzeng and J. J. Huang, *Multiple Attribute Decision Making: Methods and Applications*. Boca Raton, FL, USA: CRC Press, 2011, pp. 1–531.
[21] N. Laoutaris, S. Syntila, and I. Stavrakakis, "Meta algorithms for hierarchical Web caches," in *Proc. IEEE Int. Conf. Perform., Comput., Commun.*, Apr. 2004, pp. 445–452.
[22] S. K. Fayazbakhsh *et al.*, "Less pain, most of the gain: Incrementally deployable ICN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 147–158, 2013.
[23] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1. 1999, pp. 126–134.
[24] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proc. 1st ACM SIGCOMM Internet Meas. Workshop*, 2001, pp. 169–182.
[25] Y. Gu, L. Chen, and K. M. Tang, "A load balancing method under Zipf-like requests distribution in DHT-based P2P network systems, Web information systems and mining," in *Proc. Int. Conf. WISM*, 2009, pp. 656–660.
[26] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena, "The pollution attack in P2P live video streaming systems: Measurement results and defenses," in *Proc. SIGCOMM P2P-TV Workshop*, 2010, pp. 323–328.
[27] I. Ribeiro, A. Rocha, C. Albuquerque, and F. Guimarães, "On the possibility of mitigating content pollution in Content-Centric Networking," in *Proc. IEEE Conf. Local Comput. Netw.*, Sep. 2014, pp. 498–501.
[28] H. Guo, X. Wang, K. Chang, and Y. Tian, "Exploiting path diversity for thwarting pollution attacks in named data networking," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2077–2090, Sep. 2016.
[29] M. Aamir and S. M. A. Zaidi, "Denial-of-service in content centric (named data) networking: A tutorial and state-of-the-art survey," *Secur. Commun. Netw.*, vol. 8, no. 11, pp. 2037–2059, 2015.
[30] A. Compagno *et al.*, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking," in *Proc. IEEE 38th Conf. Local Comput. Netw. (LCN)*, 2013, pp. 630–638.
[31] 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. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 963–968.
[32] M. J. Xie, I. Widjaja, and H. N. Wang, "Enhancing cache robustness for content-centric networking," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2426–2434.
[33] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in Named Data Networking," *Comput. Netw.*, vol. 57, no. 16, pp. 3178–3191, 2013.
[34] S. Greco, J. Figureueira, M. Ehrgott, *Multiple Criteria Decision Analysis*. New York, NY, USA: Springer, 2005.
[35] S. H. Zanakis, A. Solomon, N. Wishart, and S. Dublish, "Multi-attribute decision making: A simulation comparison of select methods," *Eur. J. Oper. Res.*, vol. 107, no. 3, pp. 507–529, 1998.
[36] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," Univ. California, Los Angeles, CA, USA, Tech. Rep. 4, 2012.
[37] S. Mastorakis *et al.*, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," NDN, USA, Tech. Rep. NDN-0028, 2015.
[38] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the NS-3 simulator," *SIGCOMM Demonstration*, vol. 14, p. 527, Aug. 2008.
[39] N. Spring, R. Mahajan, T. Anderson, and D. Wetherall, "Measuring ISP topologies with rocketfuel," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, pp. 133–145, Feb. 2002.
[40] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2920–2931, Nov. 2014.

GUOZHI ZHANG received the B.S. and M.A.Sc. degrees from Northwest Normal University, Lanzhou, China, in 1999 and 2006, respectively. He is currently pursuing the Ph.D. degree with the Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University. He is also a Lecturer with the School of Computer and Engineering, Northwest Normal University. His research interests include network architecture, network protocol design, and network security.

JIQIANG LIU (M'14) received the B.S. and Ph.D. degrees from Beijing Normal University in 1994 and 1999, respectively. He is currently a Professor with the School of Computer and Information Technology, Beijing Jiaotong University. He has authored over 80 scientific papers in various journals and international conferences. His main research interests are trusted computing, cryptographic protocols, privacy preserving, and network security.

XIAOLIN CHANG (M'12) received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology in 2005. She is currently a Professor with the School of Computer and Information Technology, Beijing Jiaotong University. Her current research interests include cloud data center and network security.

ZHI CHEN is currently pursuing the Ph.D. degree with the Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University. His research interests include cloud computing and information security.