Mitigating NDN-based Fake Content Dissemination in Opportunistic Mobile Networks

Barun Kumar Saha and Sudip Misra, *Senior Member, IEEE*

Abstract—In this work, we address the problem of Named Data Networking (NDN)-based fake content dissemination in Opportunistic Mobile Networks (OMNs), where nodes have intermittent connectivity and typically lack in end-to-end communication paths. It is important to mitigate the dissemination of such fake contents not only because they waste bandwidth for legitimate communication, but also because such files can be harmful for devices. However, the inherent characteristics of OMNs make such mitigation a challenging task. In this context, we consider a group of nodes, Fake Content Providers (FCPs), who, on receiving content requests, respond with fake contents, rather than the real version. In particular, we consider four different behaviors of the FCPs – referred to as threat scenarios – depending on whether or not they always respond to all requests with fake contents. We analyze these distinct threat scenarios, and characterize the relative performance degradation arising because of them. To mitigate the adverse effects of FCPs, we propose two schemes, wherein the identified FCPs are blacklisted permanently or temporarily, and communication with them is restricted. Results of simulation-based experiments using real-life connection traces show that, compared to the normal scenario with no FCP, the percentage of content requests satisfied decreases by 20–40% in the presence of 40% FCPs in the OMN. Moreover, in the same scenario, the average latency of content satisfaction relatively increases by up to 176% with respect to the normal scenario. However, on using the proposed mitigation schemes, the latency can be reduced by about 13–36% together with up to 9% improvement in the number of interests satisfied.

Index Terms—Opportunistic Mobile Networks, Named Data Networking, content search, fake content provider, blacklist

1 Introduction

OMNs [1]–[4] – a sub-class of Delay Tolerant Networks [5], [6] – are characterized by intermittent connectivity among the nodes (for example, mobile devices carried by human users), and typically lack in end-to-end communication paths. Although communication in OMNs is challenging, the lack of dependency upon network infrastructure makes OMNs promising for use in different scenarios, for example, aftermath of a disaster and cellular traffic offloading. With the growing popularity of OMNs, it is evident that one would be interested in advanced services, for example, content searching, rather than mere unicasting. Consequently, several approaches for content searching (dissemination, in general, for example, [7]–[9]) are proposed till date. In particular, a few such schemes are based on the Named Data Networking (NDN) [10] paradigm – a popular architecture proposed for the information-centric future Internet.

1.1 Motivation and Objectives

One of the burning issues plaguing the online social media today is the viral spread of fake news (contents). They range from fake photos of disaster spread in Twitter [11], [12], fake livestream of spacewalking[1] in Facebook, and politically motivated[2] dissemination of fake stories. Concerned by their impact, many companies, such as Google and Facebook, have adopted new policies to fight fake news. On the other hand, fake contents are often injected as decoys to discourage mass dissemination of copyrighted contents in popular file sharing networks [13]–[15]. Motivated by these aspects, in this work, we address the issue of fake content dissemination in OMNs.

At this point, let us briefly mention what we mean by the term *fake*. Ghali et al. [16] used the term "poisoned" to refer to contents with invalid signature. In contrast, we consider a subjective interpretation of fake. In general, when a user requests a content by its name, but receives something that is largely or completely irrelevant to the expected content, we call it a fake content. Since contents are consumed by users in audio-visual mode, users – as well as third party software and services – typically have the ability to identify whether or not a content received in response to a request is fake (for example, a text file in a different language, an almost completely grainy image, and a corrupted audio file.)

When an FCP in an OMN receives an interest request, it responds back with a fake content either selectively or always. The presence of FCPs in OMNs has several negative impacts, as summarized below.

- Replication of fake contents consumes network bandwidth, which not only results in overhead, but also affects the bandwidth availability for legitimate communication. In other words, an unchecked dissemination of fake contents can lead to denial of service (DoS) attacks.
- In continuation of the above point, the average

• *B. K. Saha and S. Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. E-mail: {barun.kumar.saha, smisra}@sit.iitkgp.ernet.in*

1. http://mashable.com/2016/10/26/facebook-live-iss-station-nasa/
2. http://www.politifact.com/truth-o-meter/article/2016/dec/13/2016-lie-year-fake-news/

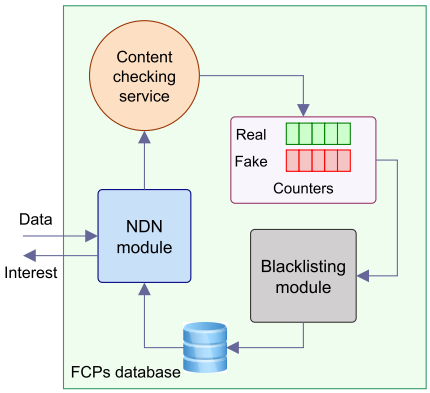


Fig. 1: High level overview of the FCP mitigation scheme.

latency in delivering real (legitimate) contents increases manifold because their transmission in the network is delayed by the fake contents.
- In the worst case, disseminated fake contents may contain worms or viruses that can compromise devices of the users.
- By apparently serving several or all the content requests, FCPs can gain undeserved "popularity" in the network.

Our objectives in this work are two-fold. First, we aim to study the impacts of fake content dissemination in OMNs. In this regard, we consider four scenarios – FCPs always serve fake contents, FCPs serve fake contents based on availability (two variants), and FCPs serve fake and real contents alternatively.

Second, we plan to investigate techniques to mitigate the FCPs exhibiting the above discussed behavior, and reduce the dissemination of fake contents to the extent possible. To this end, we consider two schemes where the non-FCPs blacklist – either permanently or temporarily – the FCPs identified so far and restrict communication with them. Our approach is motivated by the consideration that if a node serves a single fake content item, it is likely to serve several such fake contents.

Figure 1 presents a high level overview of the proposed FCP mitigation scheme. In particular, when a device receives a data message, it invokes a service running in the background to verify whether or not the content is fake. Based on the decision, either the real or fake counter for the corresponding sender is incremented. The blacklisting module takes account of such information, and labels a node either as an FCP or otherwise. These labels are taken into consideration during subsequent communication.

1.2 Contributions

The specific *contributions* of this work are summarized below.

- Considering the effects of fake content dissemination in the content searching process in OMNs, and formulating four distinct threat scenarios pertaining to the different behaviors of the FCPs.
- Theoretically analyzing, as well as experimentally quantifying, the extent of performance degradation (for example, in terms of number of interests satisfied and their latency) in the presence of FCPs using real-life connection traces.
- Proposing schemes for mitigating the adverse effects of the different threat scenarios on the content search performance in OMNs.

1.3 Organization

The remainder of this work is organized as follows. Section 2 presents an overview of NDN and related works. Section 3 presents the system model and underlying assumptions. Section 4 discusses in detail the different behaviors of the FCPs. Section 5 continues the previous discussion by characterizing the relative performances of different threat scenarios. Techniques for mitigating the FCPs are presented in Section 6. We discuss our experimental setup to evaluate the impact of FCPs and effectiveness of the mitigation schemes in Section 7. The results of performance evaluation are discussed in Section 8. Finally, Section 9 concludes this work.

2 Literature Survey

Here, we provide a brief overview of NDN. Subsequently we review the state-of-the-art related to NDN, its adaptation, and different aspects arising out of it.

2.1 Background

Among the different architectures proposed for the information-centric future Internet, NDN [10] has received huge attention. In NDN, every content is identified with a hierarchical *name*. Hosts request any content by creating an *interest* message that carries the name of the concerned content among others. When another host receives such an interest message and has the corresponding content available, it sends out a *data* message that carries the name and the actual content as payload. Each node maintains three entities – content store (CS), pending interests table (PIT), and forwarding information base (FIB) – to facilitate the content searching process. The CS, as the name suggests, is used to cache received contents. The PIT maintains a list of received interests that are not yet satisfied. The FIB, on the other hand, acts as a routing table and stores information on how to "reach" a given content.

OMNs, on the other hand, present an entirely different premise of networking not just because of their host-centric nature, but largely because of their challenging characteristics, as noted in the beginning of Section 1. Consequently, we adapt NDN for OMNs, as discussed in [17]. In particular, we consider that host names are retained in communication since wireless devices usually have a single active networking interface. Moreover,

interest and data messages are replicated (rather than forwarded) in order to improve the chances of their delivery.

2.2 Related Work

Researchers have explored several aspects and applications of NDN, for example, name management [18], post-disaster scenarios [19], and multimedia streaming [20]. NDN has also been adapted for other networks, such as vehicular networks [21] and the Internet of Things [22]. Service function chaining in NDN [23] has also been considered.

Duarte et al. [24] proposed the Probabilistic Interest Forwarding Protocol (PIFP), an NDN-based content searching scheme for OMNs. Interests replication decisions in PIFP are based on a variant of the PRoPHET [25] routing protocol. In particular, an interest is replicated to another node if that has a better chance in coming in "contact" with the concerned named content. Similar to PRoPHET, contact probabilities with names are aged, and transitivity of contacts is taken into account.

Saha and Misra [17] proposed the Content Searching as Regret Minimization (CHARM) scheme for OMNs. In CHARM [17], the nodes compute the regrets associated with replication of interests and lack thereof. An interest is replicated to another node, if the corresponding regret is smaller than the regret associated with its non-replication. The nodes depend on locally available information, such as the number of request hits and inter-contact times, to evaluate the regrets. In contrast to PIFP, CHARM was found to improve the number of content requests satisfied and their latency by different extents. However, neither CHARM nor PIFP address the issue of fake content dissemination in OMNs.

Real-life networks are vulnerable to various threats and attacks, such as DoS [26], distributed DoS (DDoS) [27], vampires [28], and blackholes [29]–[31]. Chatterjee et al. [32] present a survey of many such attacks in the context of NDN. In DoS and DDoS attacks, the objective is to cripple a network by overloading it. Although the inherent design of NDN makes it immune to many traditional types of attacks, it is nevertheless susceptible to new threat categories. For example, the Interest Flooding Attack (IFA) [33]–[35] constitutes a specific form of DDoS attack that exploits some of the flaws of NDN. In particular, in IFA, the adversaries generate content requests that cannot be satisfied, which overloads the PIT of other hosts resulting in the drop of valid interest messages.

Content poisoning [16], [36], [37] constitute another major threat to NDN, in which the hosts return "bogus content" [36] in response to the interests received. Ghali et al. [16] addressed the problem of content poisoning by adversaries in NDN by ranking the contents requested. In the proposed scheme, when users detect fake contents, they request the same content again with instructions for excluding the previously received fake content by using the exclude filter. In particular, the routers rank a given content based on three different information – frequency of exclusion of a content, how recently it was excluded, and at how many network interfaces it was excluded. The authors established the effectiveness of the proposed scheme by simulating different network topologies. DiBenedetto and Papadopoulos [36], on the other hand, proposed that when fake content is detected, a feedback report be sent upstream that marks the corresponding forwarding path as a bad choice.

TABLE 1: Summary of Works Related to NDN and Fake Content.

Studies	Aspects	Shortcomings
PIFP [24]	Maintains likelihood of a node coming in contact with a content	Currently unsuitable node may be a good candidate in the future
Interest Flooding Attacks [33]	Generate content requests that cannot be satisfied	Involves non-existent contents, not fake
Ghali et al. [16]	Probabilistically identify fake content based on consumers' behavior	Routers depend on feedback from consumers, which would involve a long delay in OMNs
DiBenedetto and Papadopoulos [36]	Send feedback upstream via report messages marking the path as bad	Concept of path typically does not hold in OMNs
Bahrami et al. [23]	Verifies end-to-end data integrity in function chaining in NDN	One or more service(s) provided can be fake or they can process originally fake content; data integrity can still be maintained in both the cases

Table 1 summarizes some of the key works in the literature. To *synthesize*, in this work, we address the aspect of content poisoning for NDN-based OMNs. However, the current work is significantly different in terms of the problem addressed and the solution proposed in the literature. First, the existing works address the said problem in the context of NDN. As noted earlier, due to fundamentally different characteristics of OMNs, NDN cannot be used in its present form for OMNs, but must be adapted. Second, while [16] attempts to mitigate the effects of content poisoning, our objective is to identify the malicious nodes, which spread fake contents across the OMN. This is relevant in OMNs, since, as discussed earlier, communication in OMNs retains the node identifiers. Third, the scheme in [16] cannot be ported for OMNs due to the facts that OMNs typically involve large latency in message (information) delivery, and that all the (mobile) nodes in OMNs themselves are routers. On the other hand, the concept of traditional routing paths ceases to exist in OMNs. Consequently, path-based approaches as in say, [36], tend to lack usefulness in OMNs. Fourth, by fake content, we not only refer to those with malformed signatures, but to a larger category, as discussed at the beginning of Section 1.1.

Finally, we also look at different ways in which hosts serve fake contents, which, in turn, gives rise to distinct threat scenarios.

3 System Model and Assumptions

In this work, we assume that content searching in OMNs is based on the NDN paradigm, for example, as in PIFP [24] and CHARM [17]. Here, we do not deal directly with the PIT and FIB components. Rather we look at how the FCPs behave while responding to an interest message.

Let N be the set of nodes in the OMN. Among them, let $F \subset N$ be the set of fake content providers. Let $C = C_R \cup C_F$ be the set of contents available in the OMN. Here, C_R denotes the set of real contents, whereas C_F denotes the set of fake contents. Let I and D, respectively, be the set of all interest and data responses generated in the OMN. Let $\texttt{name}_i$ be the name carried (i.e., content requested) by interest $i \in I$. Then, $data(\texttt{name}_i) \in C$, $\forall i \in I$, where $data(\cdot)$ indicates the data corresponding to a given content name. Moreover, let $I_S \subseteq I$ be the set of interests that are satisfied, i.e., the interests for which the requesting nodes receive the corresponding (real) contents. Finally, let $S_i(t)$ denote the CS of the ith user at time instant t, $i \in N$.

We make the following assumptions in this work.

- Interest messages contain address of the requesting node, whereas data messages contain addresses of both the requesting as well as the responding (content serving) node.
- Alternative versions of any content (having same name but different checksum/hash value) are not available in the OMN.
- On receiving and identifying a fake content, users do not regenerate requests for the concerned content. We argue that the considerable latency, on an average, in receiving a content is deterrent enough to persistently keep requesting the concerned content. However, users can generate requests for a given content several times *independent* of whether or not it was received earlier.
- Nodes are aware of the content names, and therefore, no latency in name resolution is involved.

4 Representation of FCPs

In this Section, we discuss the behavior of FCPs and threat scenarios posed by them in detail.

4.1 Threat Scenario 1: Always Serve Fake Content

In this threat scenario (abbreviated as TS1, henceforth), an FCP serves all incoming requests, irrespective of whether or not it has the corresponding content available. In TS1[3], the behavior of FCPs with respect to content requesting, data reception, and updating routing tables & content advertisement is same as the "normal nodes." However, their behavior differ from that of *bona fide* nodes during **interest reception**. In TS1, an FCP serves *all* incoming interests with fake data. In particular, when an FCP receives an interest message, it responds back by creating a data message as usual. However, the payload of the data message contains garbage or anything but real content corresponding to the request served.

4.2 Threat Scenario 2: Serve Fake Content if Unavailable

In the second threat scenario (abbreviated as TS2 henceforth), an FCP serves a mix of both real and fake contents. Similar to TS1, in TS2, too, an FCP behaves as a normal node in all events but interest reception. Therefore, here we elaborate upon this aspect only.

In TS2, an FCP begins with zero or more content items available in its CS. Let $S_i(t_0^i)$ be the initial CS of an FCP $i \in F$, where t_0^i indicates the time instant when node i became operational (joined the network). When such an FCP receives an interest message at any time say, t, it checks whether or not the corresponding content – say, $c \in C$ – is available in the CS. In case $c \in S_i(t)$ and $c \in S_i(t_0^i)$, the FCP attempts to send the *real* data to the requesting node. In other words, an FCP in TS2 serves real content only if it was originally available with itself. Otherwise, the FCP sends out a data message with a payload carrying *fake* content.

It may be noted that for any time instant $t > t_0^i$, we have $S_i(t) \neq S_i(t_0^i)$, in general. This is because, an FCP caches the data received in its CS when there is sufficient space. Therefore, FCPs of type TS2 need to maintain a separate list indicating the contents that were originally carried by them, if any.

4.3 Threat Scenario 3: Serve Fake Content Until Real is Available

Next, we consider TS3, a far more realistic threat scenario. In TS3, an FCP serves the fake version of a given content as long as it is unavailable with it. However, when the FCP receives[4] the real version of the concerned content, it changes its behavior by serving the real content. Such switching of behavior holds for all the content items concerned. The difference between TS2 and TS3 is subtle. In TS2, an FCP serves real content only if it was originally available with itself. However, in TS3, an FCP stops catering the fake version of a content as soon as it receives the real one.

4.4 Threat Scenario 4: On-off Attack

In the final threat scenario (abbreviated as TS4), we consider that for a certain time duration, say T_{off}, FCPs serve real content. However, for the immediate T_{on}

3. Wherever appropriate, we also mention this as "an FCP of type TS1."

4. This happens when the FCP itself requests the corresponding content and receives it from some other non-FCP.

duration of time, the FCPs respond to interest messages with fake contents. This cycle of behavior of the FCPs keep on repeating. Without loss of generality we assume that $T_{on} = T_{off} = 2$ hours.

4.5 Motivation Behind Serving Fake Content

TS1, where fake content is spread always, represents a rather hostile or malicious behavior on part of the FCPs. Their apparent objective is to cripple the network operations by launching attacks similar to DoS. On the other hand, serving real content that were originally available with them, FCPs of type TS2 reflect "cleverness" who possibly wish to remain under the radar and prevent exposing their malicious intent. In TS3, we capture a rather realistic scenario. For example, some peer-to-peer sharing platforms may require users to share a minimum volume of data in order to participate. Moreover, users found to share fake files are often banned. Here, a new user may begin with fake files to access the system. However, as soon as he/she receives the desired (real) content, the user replaces the fake files with them. So, although an FCP of type TS3 may not instantly turn into a non-FCP, the extent of fake content served by it over time decreases. On the other hand, TS4 – and to some extent TS2 and TS3 – constitute as stealth attacks that are difficult to detect and mitigate.

5 Analysis of Threat Scenarios

In this Section, we take a deeper look into different aspects of the three threat models considered in this work. The following Lemmas characterize the network performance under the different threat scenarios.

Lemma 1. *Let $|I_S|_k$ be the number of interests satisfied in the presence of k FCPs in the OMN under any threat scenario. Then, $|I_S|_k$ is a monotonically decreasing function of k.*

Lemma 2. *Let $|S_i(t)|$ be the number of items in the CS of an FCP i of type TS3; $\forall i \in F$. Then, $|S_i(t)|$ is a monotonically increasing function of time t.*

Lemma 3. *Let us consider two networks, where the total number of nodes ($|N|$) and the number of FCPs ($|F|$) are the same. Moreover, let both the networks exhibit exactly the same pattern in requesting contents and pairwise contacts. Additionally, we assume that all the FCPs in one of the networks are of type TS2, whereas those in the other are of type TS3. Finally, let I_S^{TS2} and I_S^{TS3}, respectively, be the set of interests satisfied under TS2 and TS3. Then, $|I_S^{TS3}| \geq |I_S^{TS2}|$.*

Lemma 4. *Let us consider FCPs of type TS2, such that $|S_i(t)| = 0$, $\forall i \in F$. Then, TS2 reduces to TS1.*

Theorem 1. *Let $\Delta_k = |I_S^{TS3}|_k - |I_S^{TS2}|_k$ be the difference between the number of interests satisfied under TS3 and TS2 in the presence of k FCPs. Moreover, let $\Delta_{k,k'} = \Delta_{k'} - \Delta_k$, $k' > k$. Then, $\Delta_{k,k'}$ is non-negative.*

Proof: For the sake of legibility, let us introduce a few symbols. Let $a = |I_S^{TS3}|_k$ be the number of interests satisfied in the presence of k FCPs. Similarly, let $b = |I_S^{TS2}|_k$, $c = |I_S^{TS3}|_{k'}$, and $d = |I_S^{TS2}|_{k'}$. Then, the given expression becomes $\Delta_{k,k'} = (c - d) - (a - b)$. In this context, the following inequalities hold.

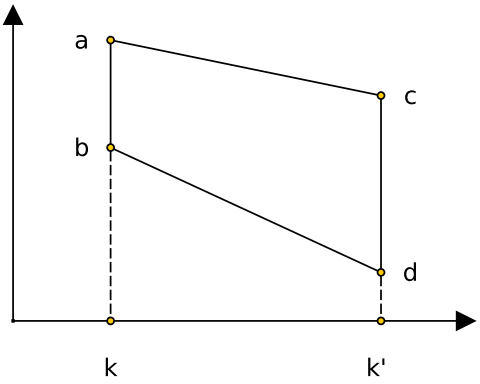


Fig. 2: Construction for the proof of Theorem 1.

- $c \geq d$, from Lemma 3
- $a \geq b$, from Lemma 3
- $a \geq c$, from Lemma 1
- $b \geq d$, from Lemma 1

Based on these inequalities, a trapezium can be constructed, as shown in Figure 2. The x- and y- axes, respectively, indicate the number of FCPs and the number of interests satisfied. Since $\overline{cd}$ constitutes a longer line segment than $\overline{ab}$, the difference $(c - d)$ must be greater than $(a - b)$. Consequently, $\Delta_{k,k'} = \Delta_{k'} - \Delta_k \geq 0$. □

Theorem 1 implies that as the number of FCPs in an OMN increases, the divide between the number of interests satisfied under TS2 and TS3 continuously widens. While the satisfaction rate falls down in both the scenarios, the rate of deterioration for TS3 is lower than TS2.

6 Mitigating FCPs

In this Section, we discuss two schemes for mitigating the previously discussed threat scenarios.

6.1 Blacklist Hosts

We assume that devices of users have a background service running that can detect whether or not any received content is fake. Since a data message also contains the address of the sending node (see assumptions, Section 3), the receiving device is able to identify the concerned FCP. Subsequently, the application *blacklists* the sending node. In particular, each node i maintains a list L_i to which the address of the identified FCP is added. It may be noted that the background service may not always correctly identify a fake content. For example, it may confuse a Belgian flag with the German one. To account for this, we consider that fake contents can be identified with a probability p_I. For instance, certain Android apps can detect whether or not an image has

TABLE 2: Actions of non-FCPs vis-a-vis FCPs.

	Interests	Data
Send	×	✓
Receive	✓	×

been doctored[5]; while another can categorize[6] pictures. It is assumed that the content sharing system interacts with one or more apps in the background, as shown in Figure 1, and increments the counters accordingly. A detailed discussion on such background interaction, however, is scoped out of this work.

6.1.1 Restricting Communication

A rational approach against FCPs of type TS1 would be not to send (replicate) interest messages to the FCPs as well as not to receive data messages created by them. However, a given node can receive interest messages from FCPs and also send data messages to them. These are summarized in Table 2.

The objective of non-FCP nodes here are not to send interests to the FCPs as well as not to receive data messages created by those FCPs. To attain the first objective, a node can simply transmit an empty list of interests. However, in order to refuse receiving data messages offered by another node, we need to augment the *handshaking* scheme between a pair of nodes when they exchange summary vectors (list of identifiers of messages carried by them).

6.1.2 Isolating FCPs

At this point, it may be noted that merely a node's own observations to blacklist FCPs are insufficient here due to the following two reasons.

- Node j (or any other node) itself would be generating only a limited number of content requests.
- Due to the inherent uncertainty in the connection pattern in OMNs, only a small fraction of those content requested by j would be served by i.

Therefore, when two nodes say, i and j, come in contact, they send their respective blacklists to one another and synchronize them. In particular, node i (j) adds all items from L_j (L_i) into L_i (L_j). Following this operation, both i and j have better knowledge about the existing FCPs in the OMN. The complete sequence of communication together with handshaking is illustrated in Figure 3.

Algorithm 1 summarizes the high-level operations performed by a non-FCP node upon the reception of a data message. In particular, on detecting the data item as fake (line numbers 3 and 4), it blacklists the source of the data as an FCP (line numbers 5). However, if the content is not fake or the node failed to identify it as fake, normal operations based on the underlying content searching scheme follows (line number 6).

5. See https://play.google.com/store/apps/details?id=fake_image_detector.coder.genuine.com.fakeimagedetector and https://play.google.com/store/apps/details?id=com.hogdex.photofraudfree

6. https://play.google.com/store/apps/details?id=com.ldqstudio.image_recognizer

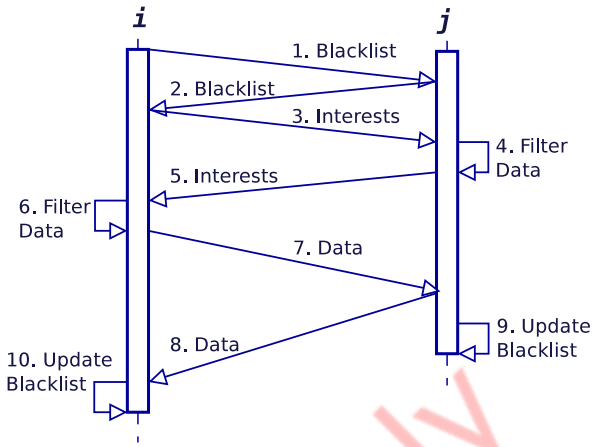


Fig. 3: Illustration of the communication sequence between two nodes i and j. The different steps involve the following processing. 1 & 2: mutually exchange the blacklists; 3 & 5: send interest messages, if any, to one another; 4 & 6: filter data messages to send to the other node; 7 & 8: exchange data messages; 9 & 10: update respective blacklists based on information received from the other node. Note that there would be a zeroth step before step 1 where the nodes exchange routing information, if any, based on the underlying search scheme. This is not shown here.

Algorithm 1: Message reception by non-FCP node i

Inputs:
- L_i: Blacklist to store addresses of FCPs
- d: Data item

```
1 upon event <data reception>:
2     if this host requested data d then
3         Invoke background service to check
          whether d is fake
4         if d is fake then
5             Add d.createdBy() to L_i
6         else
              // Usual processing upon data
                 reception based on the
                 underlying search scheme
7         end
8     end
```

6.2 Dynamic Blacklists

The aforementioned static blacklisting scheme is suitable to deal with TS1, where FCPs always serve fake contents. However, it does not take into account the dynamic behavior of FCPs as considered in TS3, where, given enough time, an FCP can possibly turn into a non-FCP. To this end, we consider a dynamic version of the node blacklisting scheme.

We consider that each node j maintains a *transactions table*, T_j, where each row of the table is a three-tuple $\langle i, n_R^j(i), n_F^j(i)\rangle$. Here, i indicates the address of the ith node; $n_R^j(i)$ and $n_F^j(i)$, respectively, denote the number of real and fake content items received by node j from i. Initially, every node begins with an empty table. When a node say, j, comes in contact with i, it creates a row for node i in the table and initializes both $n_R^j(i)$ and $n_F^j(i)$ with zero. Let us consider that at a later time instant, j requests a content, which is served by i. If the received content is real, j sets $n_R^j(i)$ to 1; otherwise, it sets $n_F^j(i)$ to 1. When j comes in contact with another node say, k, j synchronizes its record for i in T_j, as shown in (1).

$$\left.\begin{aligned} n_R^j(i) &= n_R^j(i) + n_R^k(i) \\ n_F^j(i) &= n_F^j(i) + n_F^k(i) \end{aligned}\right\} \quad (1)$$

Node k does a similar update to its T_k, in turn.

It may be noted that neither $n_R^j(i)$ nor $n_F^j(i)$, in general, indicate accurate counts. This is because any pair of nodes j and k can meet several times and repeatedly update their transactions tables following (1). Thus, $n_R^j(\cdot)$ nor $n_F^j(\cdot)$ are non-decreasing with respect to time.

In reality, what we need is a logical inference – whether or not $n_R^j(i)$ is greater than $n_F^j(i)$ – that can help label a given node as an FCP or otherwise. In fact, if a node serves more fake contents than real, then, based on (1), the count $n_F^j(\cdot)$ would likely be amplified. Similarly, for a node say, f, that has served several real contents, the count $n_R^j(f)$ would keep on increasing and eventually exceed $n_F^j(f)$. Now even if $n_F^j(f)$ is non-zero (i.e., f has served at least one fake content), not blacklisting f would be beneficial because a larger set of real contents can be received at a cost of a few fake ones. Based on the above, at any instant of time t, a non-FCP node j assigns a label $\ell_i^j(t)$ to node $i \neq j$, as shown in (2).

$$\ell_i^j(t) = \begin{cases} \text{Blacklist}, & \text{if } n_F^j(i) > n_R^j(i) \\ \text{Whitelist}, & \text{otherwise} \end{cases} \quad (2)$$

As long as a label indicates that a node is blacklisted, message replication actions are performed based on Table 2. However, if the label indicates that the node be whitelisted, normal procedure based on the underlying content searching scheme is followed.

In a different context, Saha et al. [38] considered the possibility of providing incentives for exploratory actions of users. Here, as an example, let us consider a utility function that is measured by the difference in number of real and fake contents served. As Figure 4 shows, the utility increases when the number of fake contents served increases, whereas the utility decreases with increasing number of fakes. Therefore, the rational decision for any node should be to share more real contents than fake to improve their utility values.

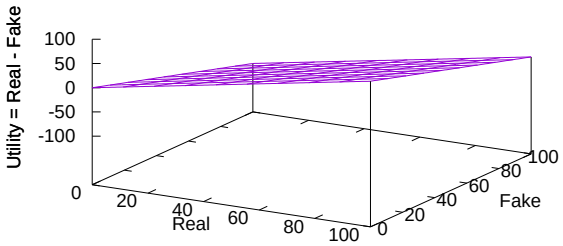


Fig. 4: Variation in utility of a node.

6.3 Why not Blacklist Fake Contents?

Blacklisting contents, rather than hosts, does not seem to be an effective strategy primarily due to two reasons. First, a host can identify the concerned content as fake only after receiving it, which involves a long delay in OMNs. However, it is not sufficient just for the node to identify it – other nodes in the network must be informed about the same. In turn, it again takes a considerable time to spread the news to the other nodes in the OMN. Therefore, by the time most of the nodes receive the information, the concerned fake content has perhaps already transmitted several times across the network. It may be noted that even synchronizing the balcklists of all the nodes takes time. However, when FCPs are blacklisted, the spreading of data messages created by them is prevented. On the other hand, when a specific content created by an FCP is blacklisted, that does not help in preventing the transmission of other possible fake contents created by that same FCP. Second, by allowing any host to blacklist any content across the network, we would be enabling possibly arbitrary censorship of any content that someone may not like. This can seriously affect users' right to free speech, and can bias the communication for/against certain entities.

6.4 Collusion

We also consider the aspect of collusion among the FCPs of type TS3. In particular, we assume that under such a collusive scheme, when an FCP i comes in contact with another FCP j, i increments $n_R^i(j)$ by k, where k is a positive integer. Similarly, FCP j increments the value of $n_R^j(i)$ by k. In other words, the FCPs attempt to present each other as bonafide and popular agents catering to several content requests. In practice, the value of k should not be too large so that signs of anomaly become clearly visible to the genuine nodes.

Note that, in practice, the above scheme can be realized in different ways. If the protocol implementation is not secure, such colluding nodes can directly tamper with the content counters and increment them at wish. However, such an approach would not work if the concerned protocol and devices employ tamper-proof hardware. Nevertheless, in such a case, node i

can always request for k arbitrary items, and node j can always provide them back to i. Since the two nodes collude, i is not bothered by the fact that those k received items could be fake; it simply increments its counter by the value k.

7 Performance Evaluation

In this Section, we describe the experimental setup used for investigating the impact of FCPs on content searching in OMNs. Subsequently, we discuss the metrics used to evaluate the search performance.

7.1 Simulation Setup

We used the Information Centric Opportunistic Network Environment (ICONE) simulator [24] to implement the three threat scenarios and schemes for their mitigation. ICONE is based on the popular ONE simulator [39], and additionally, provides features related to NDN. We used CHARM [17] as the underlying content searching scheme. This was largely motivated by the fact that CHARM, in general, can outperform PIFP [24], another popular NDN-based search scheme for OMNs. Atop CHARM, we added the behaviors pertaining to TS1, TS2, and TS3, as discussed in the earlier Sections. Thus, the nodes in the network simulation would search and route contents following the functionality of CHARM. The only differences were that the FCPs would respond with fake contents, and the non-FCPs would restrict communication with FCPs.

To simulate connectivity among the users, we used the Infocom'05 [40] real-life connection traces. We considered all the 41 users from the trace; the connection events from the first 26 hours were considered. Other typical settings were similar to [17]. In particular, the interest generation interval was 45–95 seconds. Content requests were generated for about the first 19 hours that led to the creation of 1004 interest messages. The sizes of payloads were in the range 250–350 KB. A list of $|C| = 500$ unique content names were considered. We followed a skewed distribution where these 500 content items distributed among the 14 nodes considered in the simulation. To some extent, this was motivated by the Pareto principle – a few nodes hold most of the contents. Moreover, no two nodes had the same (real) content in their CS.

Our primary experiment was designed to capture the effects of FCPs on different performance metrics in the absence and presence of the mitigation schemes. Accordingly, we varied the percentage of FCPs from zero to 50% and noted the results. FCPs were chosen randomly from the 41 nodes during each execution of the simulations. Since the proposed schemes involved randomized algorithms, we considered 15 run instances of all scenarios, took their ensemble average, and computed the corresponding 95% confidence interval. Unless otherwise stated, we assumed $p_I = 1.0$.

7.2 Evaluation Metrics

We used the following metrics to evaluate the content search performance [17].

- **Interests satisfied**: The percentage of interests satisfied is evaluated as $\frac{|I_S|}{|I|} \times 100$, where I_S is the set of satisfied interests. This metric serves as the primary indicator of content search performance. The higher the percentage of interests satisfied, the better it is.
- **Average latency**: For any satisfied interest $i \in I_S$, let t_r^i and t_s^i, respectively, indicate the time instants when the content was requested and served. Then, the average latency of interests satisfaction is evaluated as $\frac{\sum_{i \in I_S} t_s^i - t_r^i}{|I_S|}$. An efficient content searching scheme attempts to reduce the average latency to the extent possible.
- **Data received ratio**: Let $n_R(D)$ be the number of times the data messages were replicated. Then, the normalized data received ratio is defined as $\frac{n_R(D)}{|I|}$. A moderate value of this metric is desirable. However, whether or not this ratio constitutes as overhead – or to what extent – would depend on the context.
- **Normalized size of the CS**: At any time instant t, this is measured as $\frac{\sum_{f \in F} |S_f(t)|}{|F| \times |C|} \in [0, 1]$. In general, a larger value of the metric indicates better capability of the network to satisfy content requests. However, given the intermittent connectivity in OMNs and the experimental setup, we typically expect a low value of this metric.

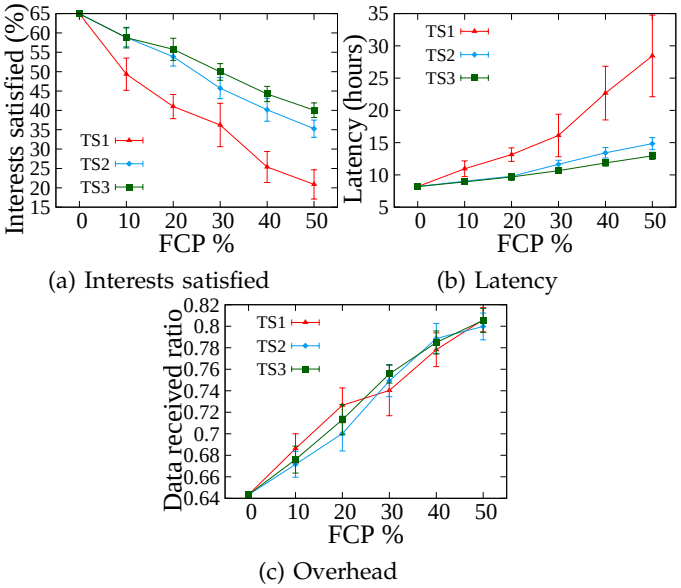
8 Results

In this Section, we present the simulation results and related discussions.

8.1 Comparison of the Threat Scenarios

Figure 5 depicts the overall effect upon the content search performance in OMNs under the three different threat scenarios considered in this study. The percentage of FCPs in the network are shown along the x-axis. The 0% scenario corresponds to the case when no user in the OMN shared fake contents.

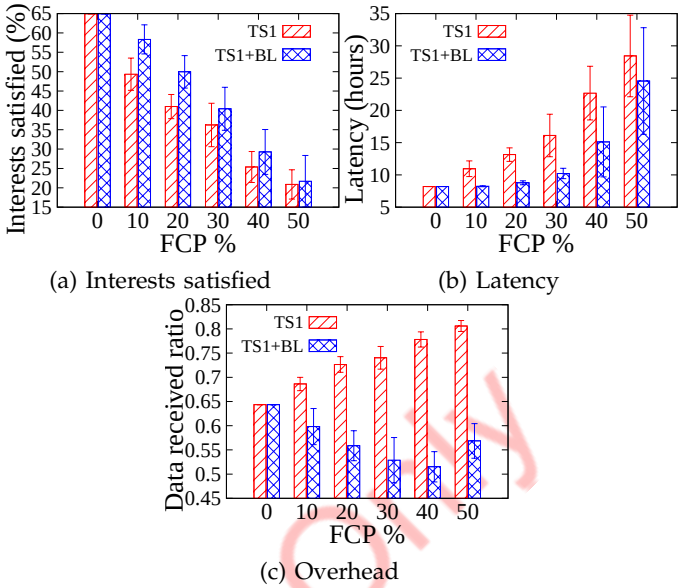
The Figure highlights the general trend that as the number of FCPs increased, the percentage of interests satisfied fell down sharply. Moreover, the average latency of interests satisfaction as well as the data received ratio both increased substantially. In particular, when there was no FCP in the network, about 65% of the content requests were satisfied, as shown in Figure 5 (a). In TS1, this reduced to 50% in the presence of merely 10% FCPs in the OMN. In a rather worse scenario where half the nodes were FCPs, only about 20% of the interests – a 40% difference – could be satisfied when TS1 was considered. However, the performance was relatively better when TS2 and TS3 were considered. For example, in TS2, around 35% requests were satisfied – about 15% more



(a) Interests satisfied (b) Latency

(c) Overhead

Fig. 5: Comparative performance of TS1, TS2, and TS3.



(a) Interests satisfied (b) Latency

(c) Overhead

Fig. 6: Effects of TS1 and its mitigation using blacklisting.

than in TS1 – when there were 50% FCPs in the OMN. With TS3, additional 5% of the interests were served. Figure 5 (a) also verifies the claims made in Theorem 1 (the number of interests satisfied decrease with the number of FCPs) and Theorem 3 (TS3 is relatively less harmful than TS2).

TS1 not only resulted in the least number of satisfied interests, but also increased their average latency by several hours, as indicated in Figure 5 (b). This was far more severe when TS1 was considered. In particular, the average latency had a relative increase of a massive 247% in the presence of 50% FCPs than when no FCP was present. However, the picture was not that gloomy when TS2 and TS3 were considered.

On the other hand, Figure 5 (c) indicates that in terms of the data received ratio, all the three threat scenarios performed very closely. It may be noted here that when there was no FCP in the network, the data received ratio was about 0.64. However, introduction of FCPs increased this ratio. In particular, with 50% FCPs in the network, the said ratio is about 0.80. However, in that particular case, the percentage of interests satisfied ranges from 20–40%, which is far below from the 65% obtained in the absence of FCPs. Therefore, the data overhead ratio here constitutes as an overhead. This arises due to the reason that as the number of FCPs increased, more fake contents circulated in the network. Such transmission and storage redundancies do not offer any utility to the users. All these aspects taken together indicate the need for mitigating FCPs in OMNs.

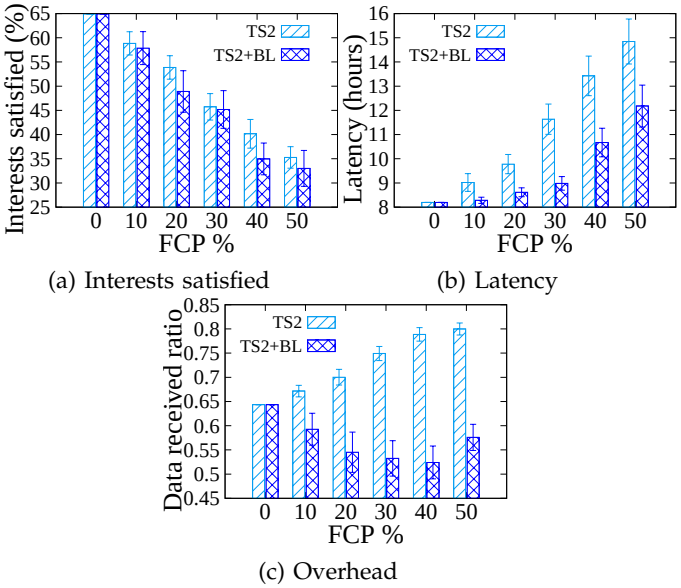
8.2 Effects of TS1

Let us now look in details at the potential benefits of mitigation of the individual threat scenarios. Figure 6, once again, shows the percentage of interests satisfied, latency, and data received ratio in the presence of FCPs under TS1. The results with legend "TS1+BL" indicate the scenarios where the non-FCPs blacklisted the FCPs (of type TS1) identified so far. It can be observed from Figure 6 (a) that by using the static blacklisting scheme, the difference in the percentage of interests satisfied can be bridged by up to about 9%. In particular, in the presence of 40% FCPs, the use of blacklisting helped in satisfying about 4% more interests. Figure 6 (b) shows that the FCP-mitigation scheme considerably reduced the average interest satisfaction latency. The maximum relative reduction of about 36% was obtained in the presence of 30% FCPs.

On the other hand, Figure 6 (c) shows a steadily downward trend of the data received ratio until half the network consisted of FCPs. A possible reason for the trend reversal is that with as high as 50% FCPs in the OMN, the non-FCPs fail to mitigate all of them, which results in increased dissemination of fake contents. Evidence in favor of this claim can be found if we look at Figure 6 (a) once again. It can be seen that in the presence of 40% FCPs, far more interests were satisfied, but it had lower data received ratio. Therefore, fake content transmissions must have increased when the number of FCPs rose further.

8.3 Effects of TS2

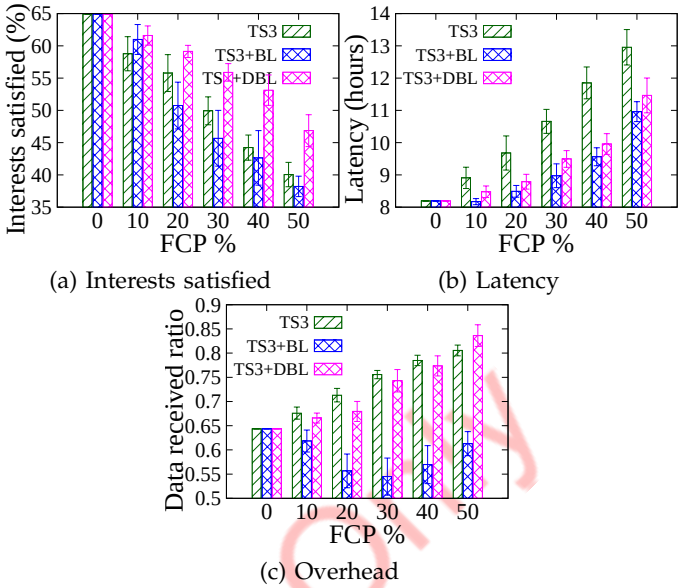
In an earlier Section, we noted that identification of TS2 is rather challenging because the FCPs serve a mix of both real and fake contents. Figure 7 (a) shows that, unlike Figure 6 (a), the blacklisting of FCPs did not help in improving the number of content requests satisfied. On the contrary, the use of blacklisting reduced the percentage marginally as compared to when no mitigation was in place. This is due to the reason that, once a node was identified as an FCP and blacklisted, no content – fake or real – was received from it. However, this



(a) Interests satisfied (b) Latency

(c) Overhead

Fig. 7: Mitigation of TS2 by blacklisting FCPs.



(a) Interests satisfied (b) Latency

(c) Overhead

Fig. 8: Content search degradation under TS3 and its mitigation using dynamic blacklisting.

marginal tradeoff in the interests satisfaction percentage came with huge and statistically significant reduction in latency, as shown in Figure 7 (b). Moreover, the data received ratio obtained by blacklisting FCPs was was considerably low. Therefore, overall, the negative impacts of TS2 were largely arrested by using the blacklisting scheme.

8.4 Effects of TS3

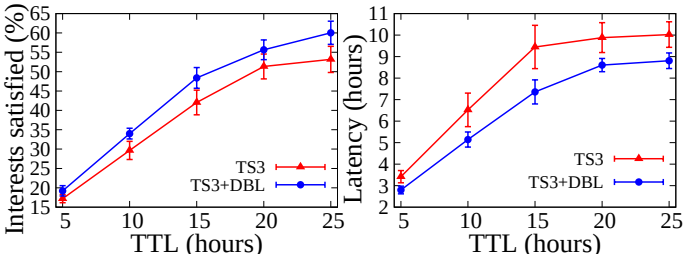
Figure 8 shows the performance of content searching under TS3. The graphs with legends "TS3+BL" and "TS3+DBL", respectively, indicate the cases where the non-FCP nodes used the (static) blacklisting and dynamic blacklisting schemes to mitigate the FCPs. It can be observed from Figure 8 (a) that the use of the blacklisting scheme ("BL") not only failed to provide any performance gain, but rather reduced the percentage of interests satisfied in multiple scenarios. An explanation to this behavior follows from the previous discussion of TS2 – by blacklisting an FCP, a non-FCP refused to receive any further content from it. Consequently, the non-FCPs ignored the real content that were provided by the FCPs of type TS3. In other words, the static blacklisting scheme fails to adapt to the changing dynamics in content request processing.

The dynamic blacklisting scheme ("DBL"), on the other hand, showed relatively impressive results. As can be seen in Figure 8 (a), the use of DBL helped in satisfying more content requests than when no form of fake content mitigation was in place. In particular, about 9% more interests were satisfied when the dynamic blacklisting scheme was used against 40% FCPs. The difference between the TS3 scenarios and TS3+DBL continued to expand until there were 50% FCPs in the OMN, where the gap marginally decreased. Similar explanation from Section 8.2 applies for this anomaly as well.

Figure 8 (b) shows that the average latency of interest satisfaction using BL and DBL was considerably lower than when no form of mitigation of TS3 was in place. In particular, the maximum relative reduction of about 16% was obtained when DBL was used to mitigate the FCPs of type TS3. However, the latency obtained using DBL was slightly higher than using BL, especially in the scenario with 30% FCPs. Since DBL resulted in satisfaction of much more interests than BL, the latency profile of the delivered data items diversified, which, in turn, resulted in higher latency than BL. This also explains the trends in Figure 8 (c) – the data received ratio using DBL was comparable to TS3 and sometimes slightly higher. The same ratio using BL, however, was considerably low. Once again, it should be remembered that the use of DBL to mitigate TS3 resulted in satisfaction of considerably larger number of messages. In order to do so, more data items must be received by the nodes. Consequently, the data received ratio using DBL went up.

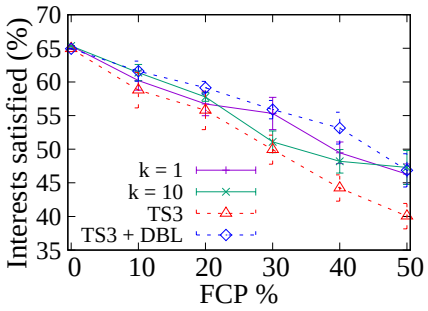
Figure 9 shows the effect of variation in TTL of messages upon the content search performance in the presence of 30% FCPs of type TS3 with and without considering the dynamic blacklisting scheme. Figure 9 (a) shows that as the TTL increased, more content requests were satisfied. Moreover, having a TTL of less than 15 hours did not seem to be good choice. Again, since more interests were satisfied, the variance in the corresponding set of latencies also increased. Consequently, the average latency of interest satisfaction increased with TTL, as shown in 9 (b).

Figure 10 shows how colluding FCPs of type TS3 influence the performance of content searching. The value of k in the plots' legends indicate by how much the colluding nodes incremented each other counters (see Section 6.4). In Figure 10 (a), it can be observed that

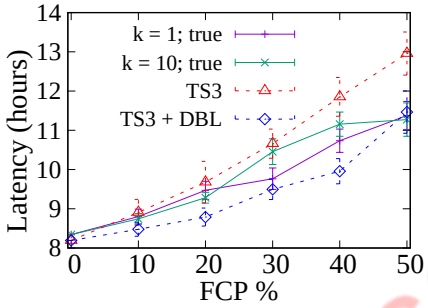


(a) Interests satisfied (b) Latency

Fig. 9: Effects of TTL upon content searching in the presence of 30% FCPs of type TS3.



(a) Interests satisfied



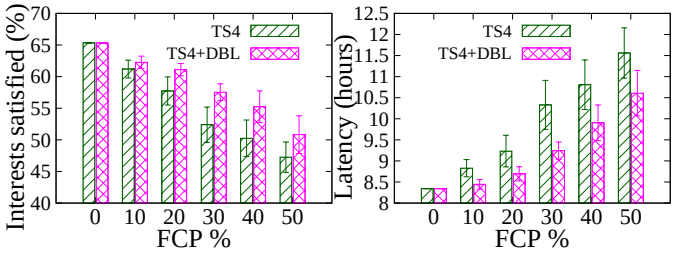
(b) Latency

Fig. 10: Content searching performance in the presence of FCPs of type TS3 together with collusion.

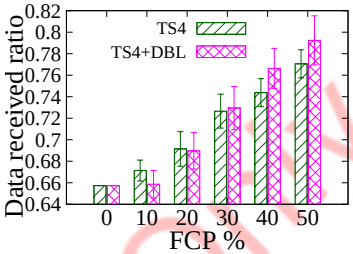
when DBL was considered together with collusion, the average number of satisfied interests decreased than the no collusion case labeled as "TS3+DBL". However, clear trends corresponding to the cases $k = 1$ and $k = 10$ are not clearly observed except in the 20–40% FCPs region; forging the data counters by a greater value ($k = 10$) brought down the performance somewhat close to the scenario when DBL was not used. In particular, any impact of collusion is apparently invisible when there were 50% FCPs. Similarly, Figure 10 (b) shows that the average latency of interest satisfaction obtained under collusion was bounded by the TS3 and TS3+DBL scenarios.

8.5 Effects of TS4

Figure 11 shows the effect of on-off attack upon the content searching performance. In particular, Figure 11 (a) shows that close to 50% of interests were satisfied in the presence of 50% FCPs of type TS4, which is slightly



(a) Interests satisfied (b) Latency



(c) Overhead

Fig. 11: Content searching under TS4 with and without dynamic blacklisting.

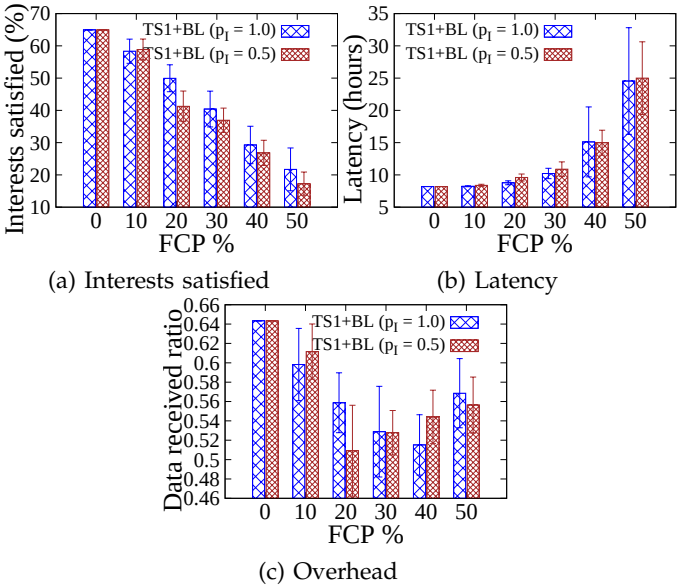
better when compared to TS3. A possible explanation is that, by launching an on-off attack, the attackers were forced to serve real contents for a specified duration of time (2 hours in our case). Additionally, we also considered the on and off durations to be equal. Significant deterioration is expected if T_{on} is made much larger than T_{off}. As with the previous threat scenarios, in TS4, too, the average number of interests satisfied with the use of DBL; the corresponding latencies also decreased by a large amount.

8.6 Effects of Probability of Identification

Finally, we look at how the probability of fake content identification (p_I) by the users affect content searching. Figure 12 shows the percentage of interests satisfied, average latency, and data overhead ratio obtained for different values of p_I in TS1 together with the blacklisting scheme. It can be observed from Figure 12 (a) that, as the percentage of FCPs in the OMN increased, the number of content requests satisfied when p_I was 1.0, in general, was slightly higher than when p_I was 0.5. Similarly, as shown in Figure 12 (b), there was a slight increase in the average latency when p_I was 0.5. However, we cannot observe significant difference in the two scenarios in Figure 12. In other words, the content search performance with lower fake content identification probability closely shadowed the one with $p_I = 1.0$. This rather sends a positive note – the negative effects of fake content dissemination can be largely mitigated even when users can identify most of them, if not all.

8.7 Variation in the size of CS

The line plots in Figure 13 show the evolution of the normalized CS size of the FCPs with respect to time. The



(a) Interests satisfied (b) Latency

(c) Overhead

Fig. 12: Effects of the probability of identification (p_I) upon the content search performance.

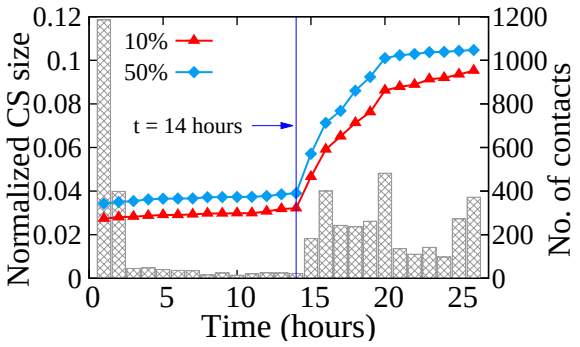


Fig. 13: Variation in the size of content store of the FCPs of type TS3 with respect to time.

two curves correspond to the scenarios where we had 10% and 50% FCPs of type TS3. As outlined in Section 7, the normalized CS size is found to be low. However, as time increased, this value exhibited an increasing trend. This verifies the claim made in Theorem 2 (size of an unbounded CS is a monotonically increasing function of time).

From the Figure, it can observed that the normalized CS size was somewhat higher when there were 50% FCPs in contrast to the scenario with 10% FCPs. A possible explanation to this phenomenon is as follows. In the 10% case, there were around 4–5 FCPs (out of 41 nodes) in the network. Since the FCPs were chosen at random and the initial content distribution was skewed, the concerned FCPs, on an average, had few contents to begin with. Moreover, the volume of fake contents replaced by them was also low. Consequently, the normalized size of CS, which is averaged over all the FCPs, was also lower. On the other hand, with 50% or about 20 FCP nodes in the network, they had, on an average, somewhat more contents to begin with. Thus, even if the same rate of fake content replacement is considered, the normalized CS size would be little higher than the scenario with 10% FCPs.

Figure 13 also shows that the rate of increase of the normalized CS size, in case with both 10% and 50% FCPs, was very slow during the first fourteen hours of simulation, but accelerated considerably beyond that. To understand this, we look at the number of contacts per hour in the OMN shown using bar plots in Figure 13. It can be observed that the first hour from the Infocom'05 contact traces exhibited very high number of contacts among the nodes. This count is also the maximum when the first 26 hours are considered. However, the hourly contacts reduced by a third in the second hour, and then fell down significantly until the fifteenth hour. Due to such low rate of contact among the nodes, the interest messages could not be routed and contents could not be delivered. Therefore, the nodes – whether FCP or otherwise – received a low number of data messages. However, after the fourteenth hour, the hourly contacts in the OMN improved considerably during which a substantial number of messages were routed. Therefore, the number of items stored in the CS of the nodes, on an average, increased relatively faster starting from the fifteenth hour.

8.8 Summary

To summarize, we note that by introducing the FCP mitigation schemes, we obtained moderate gains in the number of interests satisfied. For instance, in the presence of 40% FCPs, the percentage of interests satisfied increased by about 15%, when the blacklisting scheme was used to mitigate TS1, whereas an improvement of about 20% was obtained by using dynamic blacklisting against FCPs of type TS3. On the other hand, under TS1, the average latency decreased by as much as 33% on using static blacklisting; under TS3, the corresponding reduction using dynamic blacklisting was about 15%. However, while the data received ratio considerably decreased by the use of static blacklisting, any significant change could not be observed on using the dynamic blacklisting scheme.

Of course, the proposed mitigation schemes could not completely annul the adverse effects of the FCPs. Compared to the scenario with zero FCPs, the difference in the number of interests satisfied and latency diverged with the increasing percentage of FCPs in the OMN. Nevertheless, taking a holistic view, we find that the proposed mitigation schemes can arrest the deterioration in the content search performance by significant amounts.

9 Conclusion

In this work, we considered the effects of fake content dissemination on the NDN-based content searching process in OMNs. In particular, we considered three different threat scenarios where the FCPs respond to

content requests with fake data. Such behavior of the FCPs can be perennial, selective, or diminishing with time. Experimental results reveal that, if unchecked, the presence of FCPs can reduce (increase) the percentage of interests satisfied (latency) by a massive scale. Consequently, it is essential to mitigate the FCPs. In this regard, we proposed the blacklisting (BL) scheme wherein the non-FCP nodes restrict communication with the FCPs identified so far. We also proposed a dynamic version of the blacklisting (DBL) scheme. In the latter version, nodes gather feedback on the quality (fake or real) of data received from other nodes, and accordingly label them as FCP or otherwise.

Much like the content blacklisting process touched upon earlier, an apparent drawback of the BL scheme is that any individual can arbitrarily censor any other user in the network. The DBL technique circumvents it to some extent by considering the relative behavior of the nodes. Therefore, in the future, it would be interesting to investigate the outcomes of integrating reputation management system together with the FCP mitigation schemes. Such an approach can also help in identifying other forms of collusion in the network. Moreover, it would also be interesting to look at whether or not limited access to network infrastructure by few or all nodes could better help mitigate the fake contents.

REFERENCES

[1] B. K. Saha, S. Misra, and S. Pal, "SeeR: Simulated annealing-based routing in opportunistic mobile networks," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 2876–2888, 2017.

[2] S. Misra, B. K. Saha, and S. Pal, *Opportunistic Mobile Networks: Advances and Applications*. Cham, Switzerland: Springer International Publishing, 2016.

[3] S. Venkatramanan and A. Kumar, "Co-evolution of content spread and popularity in mobile opportunistic networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2498–2509, Nov 2014.

[4] T. Seregina, O. Brun, R. El-Azouzi, and B. J. Prabhu, "On the design of a reward-based incentive mechanism for delay tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 453–465, Feb 2017.

[5] M. Karaliopoulos, "Engage others or leave it to the source? on optimal message replication in dtns under imperfect cooperation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 730–743, March 2017.

[6] K. Sakai, M. T. Sun, W. S. Ku, J. Wu, and F. S. Alanazi, "Performance and security analyses of onion-based anonymous routing for delay tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3473–3487, Dec 2017.

[7] E. Talipov, Y. Chon, and H. Cha, "Content sharing over smartphone-based delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 3, pp. 581–595, 2013.

[8] H. Wirtz, J. Rüth, T. Zimmermann, and K. Wehrle, "Interest-based cloud-facilitated opportunistic networking," in *Proceedings of the 8^{th} ACM MobiCom Workshop on Challenged Networks*. New York, NY, USA: ACM, 2013, pp. 63–68.

[9] E. Hyytiä, S. Bayhan, J. Ott, and J. Kangasharju, "On search and content availability in opportunistic networks," *Computer Communications*, vol. 73, Part A, pp. 118–131, 2016.

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

[11] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22^{nd} International Conference on World Wide Web*. New York, NY, USA: ACM, 2013, pp. 729–736.

[12] C. Boididou, S. Papadopoulos, L. Apostolidis, and Y. Kompatsiaris, "Learning to detect misleading content on twitter," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. New York, NY, USA: ACM, 2017, pp. 278–286.

[13] D. Kushner, "Digital decoys [fake MP3 song files to deter music pirating]," *IEEE Spectrum*, vol. 40, no. 5, p. 27, May 2003.

[14] N. Christin, A. S. Weigend, and J. Chuang, "Content availability, pollution and poisoning in file sharing peer-to-peer networks," in *Proceedings of the 6^{th} ACM Conference on Electronic Commerce*. New York, NY, USA: ACM, 2005, pp. 68–77.

[15] F. R. Santos, M. P. Barcellos, and L. P. Gaspary, "Slowing down to speed up: Protecting users against massive attacks in content distribution systems," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–7.

[16] C. Ghali, G. Tsudik, and E. Uzun, "Needle in a haystack: mitigating content poisoning in named-data networking," in *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)*. San Diego, CA, USA: Inernet Society, 2014, pp. 1–10.

[17] B. K. Saha and S. Misra, "Named content searching in opportunistic mobile networks," *IEEE Communications Letters*, vol. 20, no. 10, pp. 2067–2070, Oct 2016.

[18] N. L. M. van Adrichem and F. A. Kuipers, "Globally accessible names in named data networking," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2013, pp. 345–350.

[19] T. Ogawara, Y. Kawahara, and T. Asami, "Information dissemination performance of a disaster-tolerant NDN-based distributed application in disrupted cellular networks," in *IEEE P2P 2013 Proceedings*, Sept 2013, pp. 1–5.

[20] B. Rainer, D. Posch, and H. Hellwagner, "Investigating the performance of pull-based dynamic adaptive streaming in NDN," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2130–2140, Aug 2016.

[21] S. H. Ahmed, S. H. Bouk, M. A. Yaqub, D. Kim, H. Song, and J. Lloret, "CODIE: Controlled data and interest evaluation in vehicular named data networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3954–3963, June 2016.

[22] M. Amadeo, C. Campolo, and A. Molinaro, "Multi-source data retrieval in IoT via named data networking," in *Proceedings of the 1^{st} ACM Conference on Information-Centric Networking*. New York, NY, USA: ACM, 2014, pp. 67–76.

[23] M. Bahrami, L. Xie, L. Liu, A. Ito, Y. Peng, S. Mnatsakanyan, Z. Ye, and H. Guo, "Secure function chaining enabled by information-centric networking," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2017, pp. 415–421.

[24] P. Duarte, J. Macedo, A. Costa, M. Nicolau, and A. Santos, "A probabilistic interest forwarding protocol for named data delay tolerant networks," in *Ad Hoc Networks*. Springer, 2015, vol. 155, pp. 94–107.

[25] S. Grasic, E. Davies, A. Lindgren, and A. Doria, "The evolution of a DTN routing protocol - PRoPHETv2," in *Proceedings of the 6^{th} ACM Workshop on Challenged Networks*. New York, NY, USA: ACM, 2011, pp. 27–30.

[26] M. Antikainen, T. Aura, and M. Särelä, "Denial-of-service attacks in bloom-filter-based forwarding," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1463–1476, Oct 2014.

[27] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS attacks: Trends and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2242–2270, Fourthquarter 2015.

[28] E. Y. Vasserman and N. Hopper, "Vampire attacks: Draining life from wireless ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 318–332, Feb 2013.

[29] F. Li, J. Wu, and A. Srinivasan, "Thwarting blackhole attacks in disruption-tolerant networks using encounter tickets," in *IEEE INFOCOM*, Apr. 2009, pp. 2428–2436.

[30] N. Li and S. K. Das, "A trust-based framework for data forwarding in opportunistic networks," *Ad Hoc Networks*, vol. 11, no. 4, pp. 1497–1509, 2013.

[31] T. N. D. Pham and C. K. Yeo, "Detecting colluding blackhole and greyhole attacks in delay tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1116–1129, May 2016.

[32] T. Chatterjee, S. Ruj, and S. D. Bit, "Security issues in named data networks," *Computer*, vol. 51, no. 1, pp. 66–75, January 2018.
[33] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *2013 IFIP Networking Conference*, May 2013, pp. 1–9.
[34] H. Salah, J. Wulfheide, and T. Strufe, "Coordination supports security: A new defence mechanism against interest flooding in NDN," in *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, Oct 2015, pp. 73–81.
[35] T. Zhi, H. Luo, and Y. Liu, "A gini impurity-based interest flooding attack defence mechanism in NDN," *IEEE Communications Letters*, vol. 22, no. 3, pp. 538–541, March 2018.
[36] S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 164–169.
[37] D. Kim, S. Nam, J. Bi, and I. Yeom, "Efficient content verification in named data networking," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. New York, NY, USA: ACM, 2015, pp. 109–116.
[38] B. K. Saha, S. Misra, and S. Pal, "Utility-based exploration for performance enhancement in opportunistic mobile networks," *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1310–1322, 2015.
[39] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium: ICST, 2009, pp. 55:1–55:10.
[40] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)," Downloaded from http://crawdad.org/cambridge/haggle/, Jan. 2006, [Accessed: 20 Feb. 2017].

Barun Kumar Saha received his PhD degree in Computer Science and Engineering from the Indian Institute of Technology Kharagpur, India. His research has been published in several journals and conferences. He has co-authored the book titled *Opportunistic Mobile Networks: Advances and Applications* published by Springer. He maintains a blog on DTN and the ONE simulator at http://delay-tolerant-networks.blogspot.com/, which is widely acclaimed in the community. Currently, he is a Scientist at ABB Corporate Research Center, India. Further details about Barun can be found at http://barunsaha.me

Sudip Misra is a Professor in the Department of Computer Science and Engineering at the Indian Institute of Technology Kharagpur. He received his Ph.D. degree in Computer Science from Carleton University, in Ottawa, Canada. Dr. Misra is the author of over 230 scholarly research papers and 8 books, and recipient of several awards and fellowships including IEEE ComSoc Asia Pacific Outstanding Young Researcher Award (IEEE GLOBECOM 2012, USA), (Canadian) Governor General's Academic Gold Medal at Carleton University, India – Swarna Jayanti Puraskar (Golden Jubilee Award), the Canadian Government's prestigious NSERC Post Doctoral Fellowship, and the Humboldt Research Fellowship in Germany. Dr. Misra is the Editor-in-Chief of the International Journal of Communication Networks and Distributed Systems (IJCNDS), and Associate Editor of several other journals including Telecommunication Systems Journal (Springer SBM), Security and Communication Networks Journal (Wiley), and EURASIP Journal of Wireless Communications and Networking. Dr. Misra was also invited to deliver keynote/invited lectures in over 30 international conferences in USA, Canada, Europe, Asia and Africa.