Towards Mitigating Blackhole Attack in NDN-Enabled IoT

Afia Anjum, Habeeb Olufowobi
University of Texas at Arlington, Arlington, TX, USA
{afia.anjum, habeeb.olufowobi}@uta.edu

***Abstract*—Named Data Networking (NDN) has evolved as a networking model to facilitate the distribution, security, and mobility of content in the Internet of Things (IoT). NDN provides several advantages to IoT by overcoming the constraints of the TCP/IP model through its unique features of named content, in-network caching, and a named-based routing approach. Due to the computational power and strict energy restrictions, IoT endpoints often switch to sleep mode to save energy, which results in being in stealth mode or dead address that causes missed or silently dropped packets. Moreover, a malicious node may intentionally drop packets exploiting this constraint, referred to as a blackhole attack. In this paper, we present a reputation-based forwarding approach with a reactive reputation updating mechanism to mitigate blackhole attacks in the NDN-enabled IoT network. We analyze assumptions underlying our algorithm requirements and show its effectiveness using a complex IoT network simulated in NetworkX.**

I. Introduction

The explosive growth and technological evolution of heterogeneous internet-connected IoT (Internet of Things) devices are transforming our lives and revolutionizing several industries. IoT devices such as smart lighting, smart wearables, smartphones, and health monitors are being enthusiastically adopted by consumers, known as consumer IoT (CIoT). These devices are ubiquitous and are equipped with sensing, actuating, and intelligent capabilities, allowing them to generate, process, and share a significant amount of real-time information for personal and day-to-day use cases. Named Data Networking (NDN) [1] is an emerging Internet communication protocol that provides several benefits to IoT applications, including efficient data access, dissemination, and security. NDN's unique features of content naming, in-network caching, and content-centric security provide advantages for reliably delivering content in high mobility networks, and its use in an IoT environment enables data reliability, dependability, and availability.

Despite its benefits, NDN-enabled IoT (NDNoT) environment has several limitations, such as signature verification overhead, security vulnerabilities, and storage constraints. Moreover, NDN inherits the features of resource constraint IoT devices, such as energy, memory, and processing power restrictions when used as an IoT communication protocol. Processing power limitations increase the validation overhead of signatures, while storage constraints limit content caching in each router along the data transmission path. Also, a sleep cycle scheduling behavior—sleep, sense, and connect operation mode—developed for battery-powered IoT nodes for efficient energy consumption impacts the data routing approach of the protocol. In NDNoT, the nodes between the data consumer and provider partake in data transmission requests while executing their tasks. As a result, a blackhole—packet drop or dead address—may occur in the network if a node that is the next logical step in the data path switch to sleep mode during packet delivery [2]. Although it is anticipated that every node that receives a packet would forward it accordingly, nodes with energy constraints or malicious intent may discreetly drop packets by entering sleep mode, executing a blackhole attack at the network layer.

Blackhole attacks can remain unnoticed and be misinterpreted as a lossy network issue when the attacking nodes drop packets silently. Previous research efforts have proposed several approaches to mitigate blackhole attacks. However, work on NDNoT solutions is limited [3], [4]. Consequently, we address the blackhole attack in the NDNoT environment where malicious IoT nodes switch to sleep mode during packet routing/delivery despite having considerable battery capacity. We propose a reputation-based forwarding technique that encourages the nodes to stay awake to partake in packet forwarding. Also, we present a reactive reputation updating mechanism that allows updating a node's reputation during the routing process to minimize the effect of malicious nodes in the network.

Our contributions are summarized as follows:

- We present an analysis of the NDNoT network and identify the limitations of existing approaches in handling blackhole attacks in association with the behavior of resource-constrained IoT nodes.
- We propose a novel reputation-based forwarding approach that penalizes IoT nodes based on their reputation and incentivizes nodes that maintain a good reputation by staying active. Furthermore, we present a reactive reputation updating mechanism for adjusting node reputation.
- We illustrate and evaluate the proposed approach to prevent blackhole attacks using a basic IoT network formulation and a NetworkX simulation environment.

II. Background

In this section, we introduce the NDN-enabled IoT network and define the problem statement addressed in this paper. Also, we discuss the related work and their limitations.

A. NDN-Enabled IoT Environment

The NDNoT uses NDN as a communication protocol to interconnect the IoT devices, as illustrated in Fig. 1. One of the unique features of NDN architecture is the naming of data. NDN names each of its content hierarchically rather than

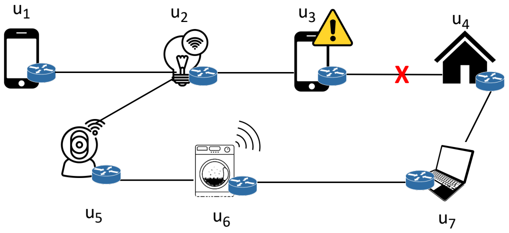

Fig. 1: Illustration of NDN having IoT devices as nodes.

utilizing IP addresses [5]. To request data, nodes issue an interest packet providing the required content name within the packet. Each node in the network function as an NDN router with three primary components: the pending interest table (PIT), the content store (CS), and the forwarding information base (FIB) [6]. PIT is a core component that keeps track of the requests that have not been satisfied, and enable in-network caching, CS functions as a cache, allowing each node in the network to store transmitted contents. FIB contains the forwarding information and facilitates the cost-efficient routing of interest packets from the producer to other nodes. These components enable the efficient routing of requested data. To ensure data security and integrity, NDN requires each data producer to cryptographically sign packets before forwarding, allowing consumers to verify the integrity of the data [7].

NDN nodes inherit some attributes of IoT devices such as resource, energy, and computational constraints. IoT devices are typically powered by batteries due to their mobility and small size, making power management an essential aspect of the design process [8]. To prolong battery life, researchers have proposed several methods based on duty cycle operation, which involves switching IoT devices between active and sleep modes [9]. For example, Table I shows three different sleep modes an ESP8266 microcontroller device can be in to save energy. In this table, the modes differ based on the active status of various device functionalities; the more the features are turned off, the more energy is saved. In this paper, we assume that the IoT devices are set to deep-sleep mode, denoted as *sleep mode*, which is the most energy-efficient mode as all features are switched off except the real-time clock. The devices will periodically sense and act, then go into sleep mode by idling the CPU, digital peripherals, and the RAM until the devices need to collect data again.

TABLE I: Three Sleep Modes of ESP8266 IoT Device [10]

Feature	Modem sleep	Light Sleep	Deep sleep
WI-FI	OFF	OFF	OFF
System Clock	ON	OFF	OFF
Real Time Clock (RTC)	ON	ON	ON
CPU	ON	Pending	OFF

B. Problem Statement

NDN offers several benefits to IoT applications, including efficient data retrieval and integrity in highly mobile networks with its content-driven data forwarding and content-centric security features. However, employing NDN to support the network of energy-constraint devices may increase the packet drop rate in the network when nodes try to save energy using sleep mode. This paper focuses on the scenario where IoT nodes on the data delivery path enter sleep mode, thereby dropping packets and creating a blackhole in the network.

Consider Fig. 1 to demonstrate the effect of a blackhole attack in the NDNoT environment. When an NDN router, in this case, u_2, receives an interest packet from u_1, it checks if the requested data is currently stored or cached in its CS. If found, the router will return it. Otherwise, the interest is directed to the router's PIT, where it queries if there is a pending entry waiting to be satisfied. If that is the case, the interest is not further forwarded to avoid redundant interest packet requests and then add the requester's information (u_1) to the entry in the PIT. When the requested data packet arrives at u_2, it is routed to all saved entries in the PIT. Contrarily, if no entry is detected, the interest is sent to the router FIB where it will get the next hop information to forward the interest packet. The interest packet will be transmitted to subsequent hops until the requested data is found in the CS. Once located, it is sent to u_1 using the precise route of the interest packet.

This forwarding and routing scenario explicitly shows that the nodes between the data consumer and provider cooperatively work to send both the request and the data. However, suppose u_3 is in sleep mode, u_2 attempt to deliver the interest packet through this path will result in packet loss and an increased packet drop rate that will generate a blackhole in the network. This packet drop issue will also increase the delay in data delivery as the node needs to re-transmit the interest packet until it finds a sleep-node-free path for every packet drop. The blackhole attack is significant in safety-critical IoT devices that rely on timely data delivery.

C. Related Work

Prior works have proposed different solutions for mitigating blackhole attacks in the NDNoT environment [3], [4], [11]–[13]. Mick et al. [11] proposed a secure routing solution to protect the network using a lightweight authentication mechanism. The authors establish a designated authentication manager in each network that authenticates each node before adding the nodes to the network through cryptography. However, blackholes can be caused by legitimate nodes that prefer not to aid other nodes' communication and frequently shift to sleep mode, making it infeasible for this method to identify. DiBenedetto et al. [12] address the packet drop issue that originates from poisoned content using consumer feedback. In their approach, the data consumers report corrupted data to the network, enabling upstream nodes to delete the incorrect data from CS after feedback verification. Based on the report, the route that delivered the poisoned data is considered the least desired choice for future interests, which reduces packet drops caused by poisoned or fake contents. Nevertheless, the approach is not practicable in mitigating packet losses caused by sleeping and energy-constrained IoT nodes in time and latency-critical environments. Zhu et al. [3] proposed a blockchain-based network model that records all requesting and forwarding information, including the hash of data packets, and assumes each node has a limit on the interest packet sending rate. The solution motivates the nodes to participate in data forwarding by rewarding nodes with an increase in the interest rate limits.

However, with resource-constrained IoT devices, blockchain would incur additional overhead for recording all the routing and forwarding information.

Fault-tolerant and energy-efficient routing protocols, such as probabilistic broadcasting and forwarding that aim to guarantee data delivery by broadcasting the request with a certain probability in different paths, have been explored by [4], [13], [14]. Lei et al. [13] provide a probabilistic forwarding technique that evaluates interface availability taking into account different network attributes such as packet loss and delays. Based on the predicted availability, the approach selects the next hop for forwarding to minimize packet drop. Yang and Chen [4] proposed SmartDetour, a reputation-based probabilistic forwarding technique that identifies the attacker node by repeatedly transmitting the failed interest from each next hop to determine where it gets lost and restore the reputation of other routers. However, these probabilistic methods fail to address malicious nodes that repeatedly go off mode, eventually forming blackholes and increasing data latency on each route.

NDN employs the best-route strategy to forward interest with minimal routing cost [15]. The approaches described address blackhole attacks by avoiding the route or node causing packet loss. Avoiding these nodes forces the interest packet to traverse using another path for data delivery, incurring additional costs and delays. Moreover, avoiding the identified attacking node seems ineffective in addressing packet drops created by IoT nodes in sleep mode, as this can occur for several reasons, such as network congestion, no path from source to destination, and when only the sleeping node has the requested information. Since IoT nodes operate a collaborative system, a node may discard packets selfishly to save energy after realizing that the forwarding path will always be avoided if detected. Therefore, we propose a forwarding technique that not only relies on avoiding a node but also motivates nodes to stay awake and play a part in forwarding using a reputation-based policy.

III. Proposed Forwarding Approach

This section outlines the design objectives and introduces our forwarding approach to address the blackhole issue.

A. Design Goals

To develop a reputation-based forwarding approach, we propose a redesign of the typical NDNoT networking topology from a forwarding and routing standpoint. The following highlights the design objectives that we took into consideration to reconstruct the network architecture:

- Encourage IoT nodes to stay awake and participate in the forwarding of interest/data packets as long as they have considerable battery life.
- When required, neighboring nodes should be able to update or adjust node's reputation.
- Assign each node a reputation score depending on how often it participates in packet forwarding, and penalize when reputation goes below a threshold.

B. Reputation-Based Forwarding Approach

We assume that the routers in the network contain the general NDN components stated in Section II-A. We introduce a new element called *sleep* packet that nodes will transmit before switching to sleep mode and propose three additional fields associated with each interface in the NDN router's FIB table; status $S \in \{awake, sleep, malicious\}$, remaining battery life, $RBL \in [0\%, 100\%]$, and reputation value, $RV \in [0, 10]$. Using these components and the traditional NDN attributes, our reputation-based forwarding approach is summarized in Algorithm 1.

```
Algorithm 1: Reputation Based Forwarding
/* $V$, $v_i$, $s_i$, $rp_i$, $RBL_i$, $V_s$ represent the
   set of all nodes, the current node,
   sleep packet sent by node $v_i$, the
   reputation of node $v_i$, the $RBL$ of node
   $v_i$, and the set of sleeping nodes,
   respectively.                                  */
Function initialization():
    foreach $v_i \in V$ do
        $status_i \leftarrow active$;
        $rp_i \leftarrow 10$;
Function $beforeSleep$():
    foreach $v_i \in V_s$ do
        $s_i \leftarrow RBL_i$;
        Forward $s_i$ to the neighbor nodes;
Function receivedSleepPacket($s_i$, $v_i$):
    $RBL_i \leftarrow$ the battery percentage included in $s_i$;
    if $RBL_i > threshold_{battery}$ then
        $\theta \leftarrow RBL_i$;
        $rp_i \leftarrow rp_i - e^{\frac{\theta}{100}}$;        /* $e$ = exponential
         constant. */
        $status_i \leftarrow sleep$;
        Update the reputation and status of node $v_i$ in the
         FIB;
Function forwardingInterest($intPacket$,
 $requester$):
    if $rp_{requester} < threshold$ then
        $d \leftarrow 10 - rp_{requester}$;
        Forward the interest packet to the next hop with
         additional delay of $d$.
```

At first, we assume each node has the maximum reputation, i.e., RV of 10. The function $beforeSleep()$ gets activated when a node decides to switch to sleep mode. The node will create a *sleep* packet s_i, including its $RBL\%$, and forward the packet to notify its neighbors. The $receivedSleepPacket()$ function gets triggered when a node receives a *sleep* packet from its neighbor v_i. The node will extract the $RBL\%$ of node v_i from the received *sleep* packet s_i and store it in a variable RBL_i after verifying the RBL using battery lifetime prediction models [16]–[18]. However, the verification process is beyond the scope of this paper. Then, the node calculate and update the reputation rp_i and status $status_i$ of v_i in its FIB. Here, RV is calculated only when $RBL\% > threshold_{battery}$ and $threshold_{battery}$ is a percentage determined by the application designer to make a trade-off between keeping the node awake and preserving energy resources.

When a node requests data, the neighboring node in the forwarding path scans the FIB to determine the requester RV as stated in function $forwardingInterest()$. If the $RV < threshold$, the requester is penalized by applying additional

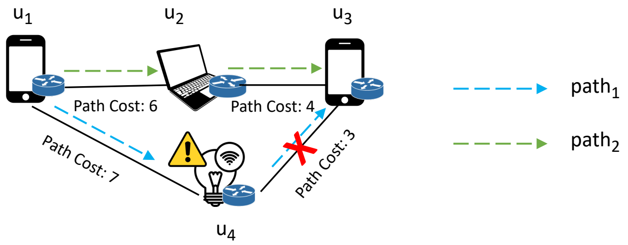

Fig. 2: NDNoT having two paths with the same path costs.

Algorithm 2: Reactive Reputation Updating

```
/* $rp_s$ represents the RV of node s.                    */
Function forwardInterest($d$, $intPacket$):
    $path_1 \leftarrow$ get the shortest path from FIB;
    $pathCost_1 \leftarrow$ calculate the path cost of $path_1$;
    $S \leftarrow$ get the list of sleeping nodes in $path_1$;
    $S_2 = S$;
    repeat
        $path_2 \leftarrow$ get the shortest path from FIB;
        $pathCost_2 \leftarrow$ calculate the path cost of $path_2$;
        $S_2 \leftarrow$ get the list of sleeping nodes in $path_2$;
    until $S_2 \neq NULL$;
    if $pathCost_1 = pathCost_2$ then
        foreach $s \in S$ do
            $rp_s \leftarrow rp_s + e^{-\gamma}$; /* where $\gamma \in [0, 10]$ and
              $\gamma \propto RBL_s$ */
```

delay $D = 10 - RV$ when forwarding the interest packet to the next hop in the forwarding path. Therefore, the requester needs to wait longer to receive the requested data. Due to the inverse relationship between RBL and node's active time, nodes that frequently switch to sleep mode will lose more energy.

To further adjust the reputation of a node stored in the FIB, we propose a reactive reputation updating mechanism that updates the reputation value based on the path cost with and without the sleeping node. Consider the scenario depicted in Fig. 2, where u_4 is in the sleep state. If u_1 requires data from u_3, it needs to send the interest packet via u_2 or u_4. When u_1 searches its FIB, it finds the shortest path to deliver the packet, which is $u_1 \rightarrow u_4 \rightarrow u_3$. Unfortunately, since u_4 is in inactive mode, this route cannot be used. From FIB, u_1 receives the second shortest path, $u_1 \rightarrow u_2 \rightarrow u_3$. Usually, the cost of the second shortest path is higher than the first one. However, this is not the case in this instance. Since the costs are equal, it is evident that u_4 is not increasing network delay by switching to sleep mode. Therefore, the reputation of u_4 is updated. Algorithm 2 summarizes the process of adjusting a node's reputation. The function $forwardInterest()$ gets invoked when a node requests a data packet. The node look into its FIB to get the most cost-efficient path, the cost, and the list of sleeping nodes (S) in that path. If the list is not empty, it searches for the next cost-efficient path. The search repeats until a route with no sleeping nodes is found. If the costs of the paths are equal, update RV of each $s \in S$ such that $RV_s = RV_s + e^{-\gamma}$, where $\gamma \in [0, 10]$ and $\gamma \propto RBL_s$.

In addition, we address the case where a malicious node creates a blackhole attack in the network. A malicious node may shift to sleep mode without informing its neighbors, resulting in a packet loss during forwarding. We address this issue by adding an ACK packet and a TTL (time-to-live) timer to the forwarding mechanism. After forwarding an interest to an adjacent hop, a node will start the TTL and wait for an acknowledgment through the ACK packet. If the acknowledgment is not received within the TTL, the node reputation value is reduced by $P\%$, where $P\%$ is set to 50% in our experiment. Moreover, the node status will be updated to $malicious$ to further avoid the path during routing. However, packet loss due to network congestion, interference, and physical connectivity issues may prevent the ACK packet from reaching its intended destination, lowering the reputation of a benign node. For this reason, after every T time, the reputation of the router node having $malicious$ status will be reviewed and subsequently restored to its initial value and status. Where T is a number determined by the application designer to ensure malicious nodes do not take advantage of the reputation restoration.

IV. Performance Evaluation

In this section, we evaluate the efficiency of our approach. We consider two types of network topology: simple and complex. The simple network is depicted in Fig. 3 and Table II, while the complex network is a simulated IoT environment in Python NetworkX.

A. *Simple IoT Network*

This network is adopted to evaluate and demonstrate the effectiveness of the algorithm in the presence of malicious IoT nodes aiming to perform blackhole attacks. We first analyze the consumer, provider, and sleeping nodes for four interest packets issued at time instance t_0, t_3, t_5 and t_{13} in our simulation.

To simplify the analysis, we assume there is no packet loss due to network issues and the energy consumption rate for each IoT node is similar. For each t unit of time a node stays in active mode, the energy consumption rate is set to 1%. In addition, energy consumption rate to wait for a data packet is $(pathcost + D) \times 0.5\%$. Similarly, when a node is in sleep mode, the energy consumption is 0.1% per time unit. To illustrate possible scenarios for this network, we select a $threshold$ number of 8 and adjust the reputation of a node s as $RV_s = RV_s + e^{-\frac{\gamma}{10}}$, where $\gamma = RBL_s$. Furthermore, the $threshold_{battery}$ percentage is set to 40%.

TABLE II: Representation of the depicted scenario in Fig. 3

Time	Source	Desti-nation	Sleep Node	Shortest Path	Second Shortest Path
t_0	u_4	u_5	NULL	$u_4 \rightarrow u_2 \rightarrow u_5$	$u_4 \rightarrow u_1 \rightarrow u_2 \rightarrow \mathrm{u}_3 \rightarrow \mathrm{u}_5$
t_3	u_6	u_2	u_1	$u_6 \rightarrow u_4 \rightarrow u_1 \rightarrow u_2$	$u_6 \rightarrow u_5 \rightarrow u_3 \rightarrow u_2$
t_5	u_8	u_4	u_2, u_6	$u_8 \rightarrow u_6 \rightarrow u_4$	$u_8 \rightarrow u_7 \rightarrow u_4$
t_{13}	u_1	u_8	NULL	$u_1 \rightarrow u_4 \rightarrow u_7 \rightarrow u_8$	$u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \mathrm{u}_5 \rightarrow \mathrm{u}_8$

Table III summarizes the energy and reputation of each node involved in interest packets forwarding from time $t_0 - t_{13}$. At time t_0, the reputation of each node is at maximum since there is no sleeping or malicious node along the shortest path from the requester u_4 to the provider u_5. At time t_3, the shortest path from u_6 to u_2 involves routing through u_1 that switched to inactive mode. Hence, a reduction in the reputation value of

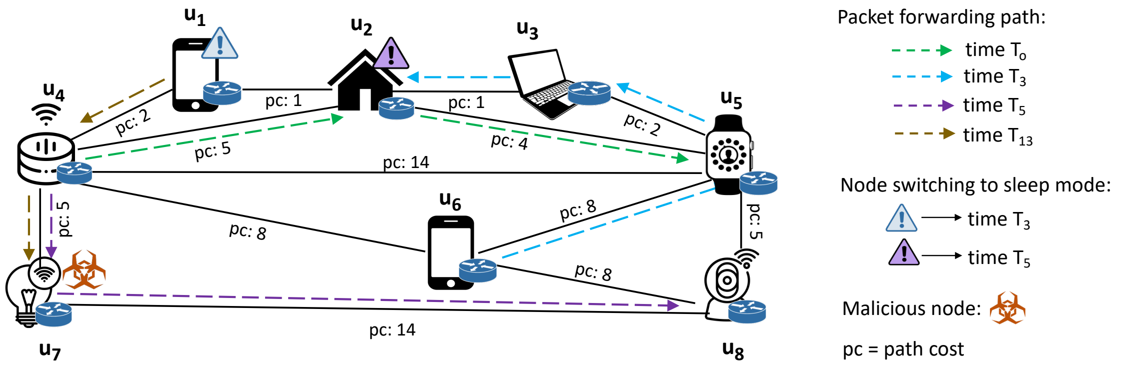

Fig. 3: A simple network topology with eight IoT nodes to illustrate the reputation-based forwarding approach.

TABLE III: Analysis of reputation-based approach in Table II

Time	Node	Initial RBL%	Final RBL%	Reputation	Updated Reputation
t_0	u_4	100%	97.5%	10	Not Applicable
	u_2, u_5	100%	99%	10	Not Applicable
t_3	u_1	97%	96.9%	7.36	7.36006
	u_6	97%	91.5%	10	Not Applicable
	u_5, u_3, u_2	97%	96%	10	Not Applicable
t_5	u_2	95%	94.9%	7.41	Not Applicable
	u_6	90.5%	90.4%	7.53	Not Applicable
	u_8	95%	89%	10	Not Applicable
	u_4	92.5%	91.5%	10	Not Applicable
t_{13}	u_1	87.9%	79.58%	7.36006	Not Applicable
	u_8	81%	80%	10	Not Applicable
	u_7	87%	86.9%	5	Not Applicable

u_1 and node u_6 selects the second shortest path to forward the interest. However, the cost of the chosen path is the same as the shortest path. Thus, the reputation value of u_1 is adjusted accordingly. Still, the adjustment was minimal, increased by 0.006%, which is inversely related to the RBL. Here, the greater the battery percentage, the lesser the adjusted value, making it difficult for nodes to misuse this approach.

At t_5, the interest packet is forwarded similar to at time t_3, except that there is no reputation adjustment as the path costs are not equal. In this case, the interest packet could not be forwarded using the shortest path. Thus, the wait time and energy usage for node u_8 increased by 20% and 1%, respectively. This phenomenon indicates that simply avoiding the node that causes the blackhole, as suggested in the previous works, will increase the network's overall latency and energy consumption due to routing through a longer path. In order to utilize the best-route forwarding strategy of NDN, the nodes should be encouraged to remain active.

At time t_{13}, the shortest path has been used to forward the interest from node u_1 to u_8. Since the requester's (u_1) RV is less than the threshold, a delay is added while forwarding. This additional delay causes the node to lose more energy than before, i.e., if no penalty had been applied, the $RBL\%$ would have been 80.9%, whereas it is now 79.58%. Furthermore, as shown in Fig. 3, node u_7 is a malicious node that drops packets silently. However, the path used to forward the interest to node u_8 involves node u_7 since it switched to sleep mode without notifying. Therefore, the interest packet sent from node u_1 is lost while routing through u_7. This malicious behavior causes the node u_7 to lose its RV by 50%.

B. Simulated Complex IoT Network

Using the same assumptions listed in Section IV-A except for the penalization threshold of 5, we simulate a complex IoT network in python NetworkX [19] with 100 nodes and randomly generated network topology. In the simulation period, we allowed 200 interest packets to be issued. The sleeping nodes, consumers, and producers for each interest are chosen at random each time. We simulate the network to exhibit the relation between the number of sleeping nodes and the interest packet failure rate. The results in Fig. 4 show that an increase in the number of sleeping nodes increases packet loss, resulting in blackholes. Then we simulate to demonstrate the impact of simply avoiding the node causing the network's blackhole, as suggested in Section II-C. Fig. 5 depicts that avoiding the path with sleeping nodes also has an impact on the interest packet failure rate. As packets travel through a longer route, the drop rate increases significantly.

Fig. 6 shows the result of the proposed forwarding approach on a randomly chosen node in the complex network. The node starts with an initial RV and RBL of 10 and 100%, respectively. The RV of the node gradually declines over time as it switches to sleep mode despite having an RBL of more than 40%. However, the node RV remains unchanged when the RBL goes below 40% as our approach allows the node to conserve energy when $RBL < 40\%$. Thus, the node's RV stays constant after reaching 4.97 throughout the simulation duration. Here, noting the energy consumption rate from the graph; the battery power drains more when the RV falls below the threshold due to the penalization factor incurred. A similar result is observed when the RBL of a node is lower from the starting of the simulation. The RV and battery power decrease accordingly if the node continues to switch to sleep mode before reaching 40% RBL.

As depicted, the forwarding approach penalizes a node depending on its reputation, causing it to lose more energy. Hence, the sooner the nodes realize that with adequate power, continuously being in sleep mode to drop packets drains more battery power, the longer they remain active. The approach encourages IoT devices to stay awake and switch to sleep mode only when necessary, such as when RBL falls below $threshold_{battery}$. Fig. 7 compares one of the mentioned state-of-the-art methods, SmartDetour [4], which uses node reputation and our forwarding approach when the number of sleeping nodes increases over time in the simulated network

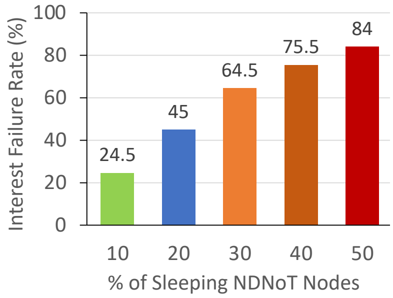

Fig. 4: Impact of increase in sleeping nodes in network.

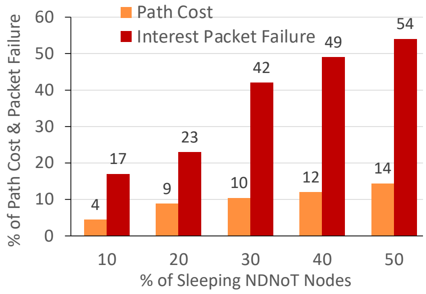

Fig. 5: Impact of avoiding routes having sleeping nodes.

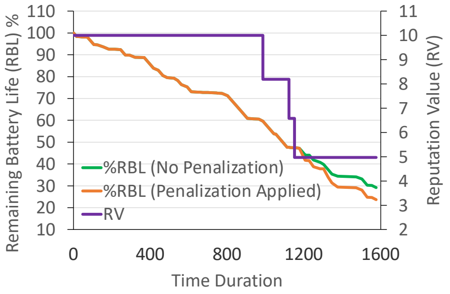

Fig. 6: Impact on RBL and RV while adopting the forwarding approach.

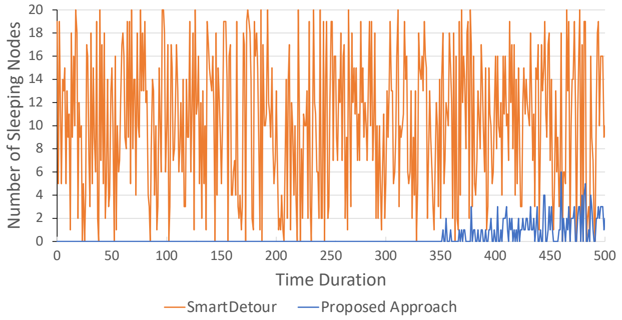

Fig. 7: Comparison between the number of sleeping nodes in SmartDetour and proposed forwarding approach.

topology. Here, the nodes start switching to sleep mode from the start of the simulation in the SmartDetour, thus increasing the number of network blackholes. However, in our approach, nodes switch to sleep mode only when the $RBL < 40\%$. Assuming no more than 20 nodes can be in sleep mode during the simulation, our approach had approximately 96.88% fewer sleeping nodes than the SmartDetour. The awareness of being active and collaborating in the routing process minimizes the network blackhole and additional network latency.

V. Conclusion

This paper presents an efficient reputation-based forwarding algorithm to mitigate blackholes in an NDNoT environment. This method encourages IoT nodes to stay active and minimize sleep cycles to participate in the data delivery process in the collaborative network. We further address malicious network nodes that switch to a sleep state to drop packets and cause dead addresses in the network using a reputation updating algorithm to identify and penalize the nodes when required. We demonstrate the effectiveness of our algorithms using simple and complex IoT network simulations in NetworkX and compared the results with the state-of-the-art. The proposed method is simple, easy to implement, and can mitigate blackhole attacks in NDN-enabled IoT networks. Future work will further refine the algorithm and detect malicious nodes that may transmit incorrect information through the sleep packet.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014.

[2] A. Anjum and H. Olufowobi, "Poster: Mitigating blackhole attack in ndnot."

[3] K. Zhu, Z. Chen, W. Yan, and L. Zhang, "Security attacks in named data networking of things and a blockchain solution," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4733–4741, 2018.

[4] N. Yang, K. Chen, and M. Wang, "Smartdetour: Defending blackhole and content poisoning attacks in iot ndn networks," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 119–12 136, 2021.

[5] M. A. Hail, "Iot-ndn: An iot architecture via named data netwoking (ndn)," in *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. IEEE, 2019.

[6] V. Jacobson, J. Burke, L. Zhang, B. Zhang, K. Claffy, C. Papadopoulos, T. Abdelzaher, L. Wang, J. A. Halderman, , and P. Crowley, "Named data networking next phase (ndn-np) project may 2014 – april 2015 annual report," *Named Data Networking*.

[7] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named data networking: a survey," *Computer Science Review*, vol. 19, 2016.

[8] A. Prasad and P. Chawda, "Power management factors and techniques for iot design devices," in *2018 19th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2018, pp. 364–369.

[9] Q. Li, S. P. Gochhayat, M. Conti, and F. Liu, "Energiot: A solution to improve network lifetime of iot devices," *Pervasive and Mobile Computing*, 2017.

[10] E. I. Team, "Esp8266 low power solutions," *ESPRESSIF*, 2016. [Online]. Available: https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf

[11] T. Mick, R. Tourani, and S. Misra, "Laser: Lightweight authentication and secured routing for ndn iot in smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 755–764, 2017.

[12] S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2016, pp. 164–169.

[13] K. Lei, J. Yuan, and J. Wang, "Mdpf: An ndn probabilistic forwarding strategy based on maximizing deviation method," in *2015 IEEE global communications conference (GLOBECOM)*. IEEE, 2015, pp. 1–7.

[14] D. Ali-Fedila and M. Ould-Khaoua, "Performance evaluation of probabilistic broadcast in low-power and lossy networks," in *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*. IEEE, 2021, pp. 247–254.

[15] M. Z. Ahmed, A. H. A. Hashim, A. M. Hassan, O. O. Khalifa, A. H. Alkali, and A. M. Ahmed, "Performance evaluation of best route and broadcast strategy for ndn producer's mobility," *International Journal of Engineering and Advanced Technology (IJEAT) ISSN*, 2019.

[16] W. Dron, S. Duquennoy, T. Voigt, K. Hachicha, and P. Garda, "An emulation-based method for lifetime estimation of wireless sensor networks," in *2014 IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE, 2014, pp. 241–248.

[17] L. M. Feeney, R. Hartung, C. Rohner, U. Kulau, L. Wolf, and P. Gunningberg, "Towards realistic lifetime estimation in battery-powered iot devices," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–2.

[18] R. Xiong, L. Li, and J. Tian, "Towards a smarter battery management system: A critical review on battery state of health monitoring methods," *Journal of Power Sources*, vol. 405, pp. 18–29, 2018.

[19] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.