Malicious Node Detection Method against Message Flooding Attacks in Sparse Mobile Ad-Hoc Networks

Takuya Idezuka*, Tomotaka Kimura†, Kouji Hirata‡, and Masahiro Muraguchi*
* Faculty of Engineering, Tokyo University of Science, Tokyo 125-8585, Japan
Email: 4318504@ed.tus.ac.jp, murag@ee.kagu.tus.ac.jp
† Faculty of Science and Engineering, Doshisha University, Kyoto 610-0321, Japan
Email: tomkimur@mail.doshisha.ac.jp
‡ Faculty of Engineering, Kansai University, Osaka 564-8680, Japan
Email: hirata@kansai-u.ac.jp

Abstract—**In recent years, many store-carry-forward routing schemes have been proposed for sparse mobile ad-hoc networks, which are the most representative networks in Delay/Disruption Tolerant Networking (DTN) environments. In general, store-carry-forward routing schemes are designed under an assumption that all nodes in the network are cooperative. Therefore, they are highly vulnerable to malicious behaviors. In this paper, we propose a malicious node detection method for message flooding attacks in which malicious nodes generate a lot of unnecessary messages to exhaust network resources. Our proposed method detects suspicious nodes in a distributed manner. Specifically, each node records suspicious scores for other nodes in the network. Whenever two nodes encounter each other, their suspicious scores are updated based on the number of messages received from the encounter nodes. Therefore, the increase in the suspicious score indicates that the scored node frequently generates and forwards the messages. After each node sufficiently updates suspicious scores, it identifies malicious nodes based on their suspicious scores. Through simulation experiments, we show the effectiveness of the proposed method.**

Keywords–DTN; store-carry-forward routing; sparse mobile ad-hoc networks; message flooding

I. INTRODUCTION

In recent years, Delay/Disconnected Tolerant Networking (DTN) technologies attract attention for realizing communications under poor communication environments [1] [2], where a path from a source node to a destination node does not exist for most of the time. A representative example of poor communication environments is a sparse mobile ad-hoc network, where the node density is very sparse. To deliver messages in the sparse mobile ad-hoc network, we use store-carry-forward routing, which is a typical example of DTN technologies. In store-carry-forward routing, when a node generates or receives messages, it stores them in its buffer. After that, the node carries them until it encounters another node. When this happens, the node forwards the messages to the encounter node. By repeating this procedure, the messages eventually reach their destination nodes.

Epidemic Routing is the earliest proposed store-carry-forward routing [3]. In Epidemic Routing, whenever a node having a message encounters another node, it always forwards a copy of the message. The node receiving the copy further spreads copies of the message over the network. If there are sufficient network resources, Epidemic Routing has the excellent delay performance, though it consumes a lot of network resources compared with other store-carry-forward routing schemes. Therefore, many improvements of Epidemic Routing have been proposed in the past [1] [4]–[6].

Most of these store-carry-forward routing schemes are designed under an assumption that all nodes in the network are cooperative and do not behave maliciously. The store-carry-forward routing is vulnerable to uncooperative behaviors, and this degrades the system performance, such as delivery delay and consumption of network resources. Therefore, to deliver messages safely using store-carry-forward routing, security issues should be considered.

To date, the behavior of some malicious attacks (e.g., black hole attacks [7], gray hole attacks [8], and fake packet attacks [9]) have been analyzed and their countermeasures have been proposed [10]. In [11], the authors analyze the behavior of message flooding attacks shown in Figure 1, where malicious nodes frequently generate a lot of unnecessary messages and spread them over the network to prevent delivering legitimate messages that cooperative nodes generate. Through Markov analysis and simulation experiments, the authors revealed how message flooding attacks affect the system performance. Because network resources are very limited in sparse mobile ad-hoc networks, the malicious node can exhaust the network resources even when malicious nodes generate slightly more messages than cooperative nodes. Therefore, countermeasures against message flooding attacks should be considered.

In this paper, we propose a detection method to identify malicious nodes that launch the message flooding attack. Our proposed method detects suspicious nodes in a distributed manner. Specifically, each node records suspicious scores for other nodes in the network. Whenever two nodes encounter each other, their suspicious scores are updated based on the number of messages received from the encounter nodes. Therefore, the increase of the suspicious score indicates that the scored node frequently generates and forwards the messages. After each node sufficiently updates suspicious scores, it identifies malicious nodes based on their suspicious scores. If our proposed method can identify malicious nodes, each node does not receive messages from the malicious nodes when it encounters them. By doing this, even when the malicious nodes generate a lot of unnecessary messages, we can prevent exhausting the network resource. In this paper, through simulation experiments, we show the effectiveness of our detection method.

The remainder of this paper is organized as follows. Section II describes the system model. In Section III, we explain our

ISBN: 978-1-61208-719-1

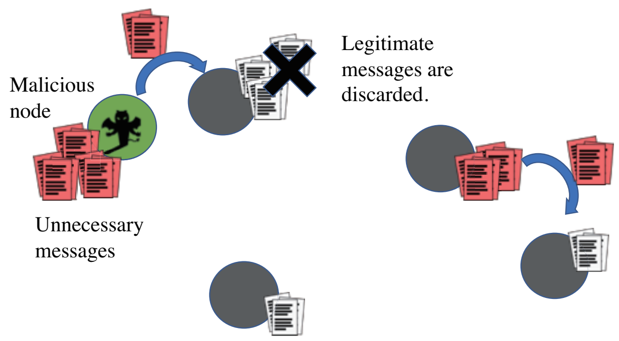


Figure 1. Message Flooding Attacks ($B = 3$).

proposed method against message flooding attacks. In Section IV, the performance of our proposed method is discussed with the results of the simulation experiments. Finally, we conclude the paper in Section V.

II. SYSTEM MODEL

We assume that there are N mobile nodes including a malicious node that launches the message flooding attack. We call nodes except for the malicious node *cooperative nodes*. Here, let $\mathcal{N}$ and $\mathcal{N}_C$ denote the sets of the nodes in the network and the cooperative nodes, respectively. The ID of the malicious node is denoted by M. By definition, $\mathcal{N} = \mathcal{N}_C \cup \{M\}$. Encounters between two cooperative nodes occur according to a Poisson process with rate $\lambda_{v,w}$ $(v, w \in \mathcal{N}_C,\ v \neq w)$. Encounters between the malicious node M and a cooperative node also occur according to a Poisson Process with rate $\lambda_{M,v}$ $(v \in \mathcal{N}_C)$. Note that in [12], the exponential inter-meeting time assumption was validated in some random mobility models, such as the random waypoint and the random direction.

Each node independently generates messages and delivers them to their destination nodes using Epidemic Routing. Specifically, each cooperative node generates a message according to a Poisson process with rate Λ_C. On the other hand, the malicious node generates unnecessary messages according to a Poisson process with rate Λ_M $(\Lambda_C < \Lambda_M)$. Therefore, the malicious node generates messages more frequently than the cooperative nodes. Note that, in this paper, we regard nodes that frequently generates messages as malicious nodes even if the nodes are cooperative. These nodes exhaust the network resources, and thus they should be detected.

Furthermore, we assume that each node has the buffer and can store at most B messages. When messages are generated or received, if the buffer is full, all the messages cannot be stored in the buffer. Therefore, buffer overflow would occur. To prevent buffer overflow, when the number of messages is $B + 1$ or more, B messages with high priority are selected, and the reminder of the messages are discarded. In this paper, each node adopts First-In First-Out (FIFO) queuing discipline. That is, it gives high priority to messages whose holding time in the buffer is short. In Figure 1, each node can store at most $B = 3$ messages in the buffer. The cooperative node that has three legitimate messages receives the copy of the unnecessary message from the malicious node, so that it discards the legitimate message with long holding time.

The drawback of Epidemic Routing is that copies of messages remained in the network after they have been delivered to their destination nodes. The nodes with the message copies cannot know that the messages are delivered to their destination nodes. To overcome this issue, a vaccine recovery method is proposed to delete the unnecessary copies [4]. In the vaccine recovery method, immediately after a message reaches the destination node, the destination node generates an *anti-packet*. The anti-packet is spread over the network using Epidemic Routing. When a node receives the anti-packet, the node deletes the message from its buffer if it has a copy of the message corresponding to the anti-packet. By doing this, it has been shown that we can delete the copies of messages remaining in the network and reduce the network resources largely. Therefore, in this paper, we adopt the vaccine recovery method to delete unnecessary message copies.

III. OUR PROPOSED METHOD

Each node $v \in \mathcal{N}_C$ has the suspicious score matrix $X^{(v)} = [x_{i,j}^{(v)}]$ to identify the malicious node. The size of the suspicious score matrix $X^{(v)}$ is $N \times N$, and $x_{i,j}^{(v)}$ indicates the suspicious score of node i that node j evaluates. The suspicious scores are updated when nodes encounter each other. At the nth encounter of nodes v, w, after the messages are exchanged, node v records the number $N_{v,w}^{(n)}$ of the received messages and the encounter time $t_{v,w}^{(n)}$. Node w also records $N_{w,v}^{(n)}$ and $t_{w,v}^{(n)}$. $x_{v,w}^{(v)}$ is then updated as follows:

$$x_{v,w}^{(v)} \;:=\; \frac{x_{v,w}^{(v)} t_{v,w}^{(n-1)} + C_{v,w}^{(n)}}{t_{v,w}^{(n)}}, \tag{1}$$

$$\begin{aligned} C_{v,w}^{(n)} &= \int_{t_{v,w}^{(n-1)}}^{t_{v,w}^{(n)}} N_{v,w}^{(n-1)} \exp(-\alpha \cdot (t - t_{v,w}^{(n-1)}))dt \\ &= N_{v,w}^{(n-1)} \alpha \{1 - \exp(-\alpha(t_{v,w}^{(n)} - t_{v,w}^{(n-1)}))\}, \end{aligned} \tag{2}$$

where α $(\alpha > 0)$ indicates the attenuation parameter, and $C_{v,w}^{(n)}$ increases with the number $N_{v,w}^{(n-1)}$ of the received messages. Node w also updates $x_{w,v}^{(w)}$. Note that the suspicious score $x_{v,w}^{(v)}$ represents the estimation of the time-average number of the messages received from node w (see Figure 2). Therefore, the increase in the suspicious score $x_{v,w}^{(v)}$ means that node v receives many messages frequently from node w, and thus node v can regard node w as the candidates of malicious nodes.

Moreover, nodes v, w exchange their suspicious score vectors $\boldsymbol{x}^{(v)} = (x_{v,1}^{(v)}, x_{v,2}^{(v)}, \cdots, x_{v,N}^{(v)})$ and $\boldsymbol{x}^{(w)} = (x_{w,1}^{(w)}, x_{w,2}^{(w)}, \cdots, x_{w,N}^{(w)})$ when they encounter. After that, nodes v and w updates their suspicious score vectors $\boldsymbol{x}^{(v)}$ and $\boldsymbol{x}^{(w)}$, respectively.

$$x_{w,i}^{(v)} \;:=\; x_{w,i}^{(w)}, \quad (i \in \mathcal{N}) \tag{3}$$

$$x_{v,j}^{(w)} \;:=\; x_{v,j}^{(v)}, \quad (j \in \mathcal{N}). \tag{4}$$

After each node v sufficiently updates and exchanges the suspicious scores, it calculates the total suspicious score to distinguish between the cooperative and the malicious nodes. Node v calculates the total suspicious score $S_k^{(v)}$ as follows:

$$S_k^{(v)} = \sum_{i \in \mathcal{N} \setminus \{v\}} x_{i,k}^{(v)}. \tag{5}$$

When $S_k^{(v)}$ is larger than the threshold value t_h, node v regards node k as the malicious nodes. In our proposed method, the

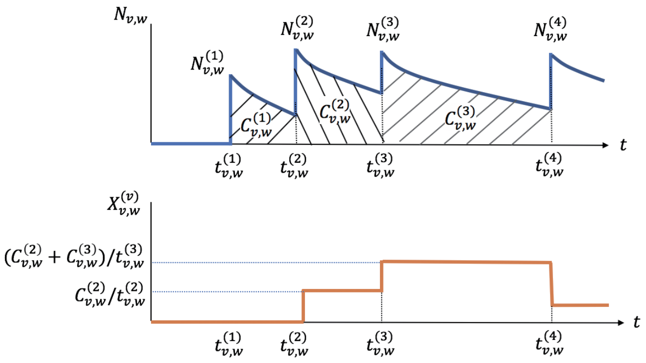


Figure 2. Suspicious Score.

threshold value t_h is defined as follows:

$$t_h = \mu^{(v)} + 2\sigma^{(v)}, \tag{6}$$

where $\mu^{(v)}$ and $\sigma^{(v)}$ are the average and the standard deviation of the total suspicious scores. Formally, $\mu^{(v)}$ and $\sigma^{(v)}$ are defined as follows:

$$\mu^{(v)} = \frac{\sum_{k\in\mathcal{N}\setminus\{v\}} S^{(v)}_k}{N-1}, \tag{7}$$

$$\sigma^{(v)} = \sqrt{\frac{\sum_{k\in\mathcal{N}\setminus\{v\}} (S^{(v)}_k - \mu^{(v)})^2}{N-1}}. \tag{8}$$

IV. Performance Evaluation

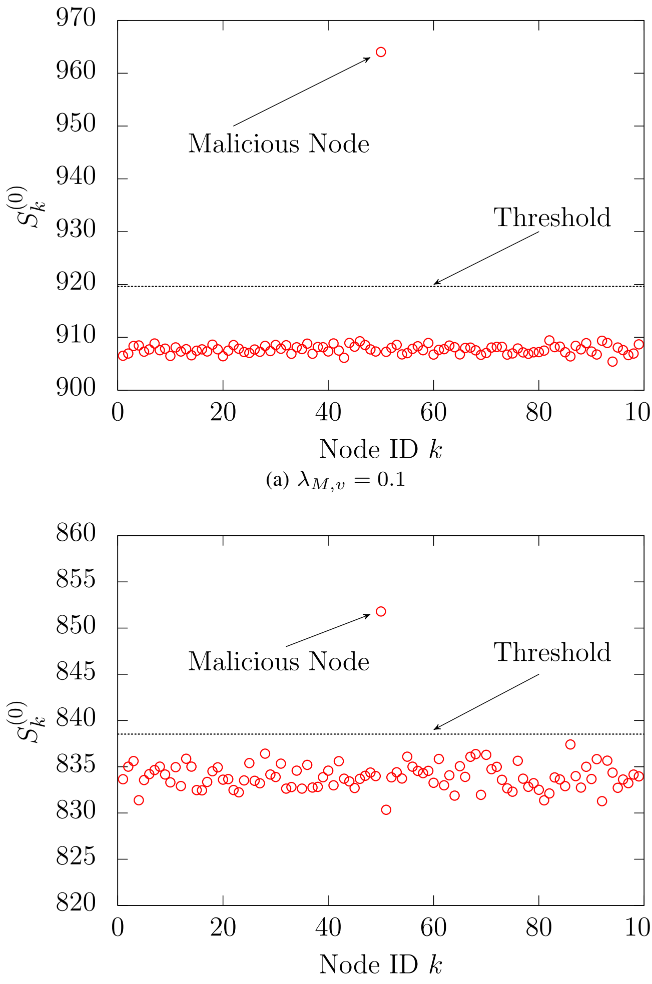
To show the effectiveness of our proposed method, we conducted the simulation experiments. In this section, we explain the simulation model, and then we show the results of the simulation experiments.

A. Simulation Model

There are 99 cooperative nodes and a malicious node in the network. $\mathcal{N} = \{0, 1, \ldots, 99\}$. The ID of the malicious node is fixed to be $M = 50$. The message generation rate of each cooperative node Λ_C is set to be 0.01. The message generation rate Λ_M is chosen from $\{0.5, 1, 3, 5, 10\}$. Messages are delivered according to Epidemic Routing incorporated with the vaccine recovery method. The rate of encountering two cooperative nodes $\lambda_{v,w}$ $(v, w \in \mathcal{N}_C, v \neq w)$ is set to be 0.01. This means that as a unit time, we choose the mean inter-meeting time $1/(98\lambda) \approx 1$ of a cooperative node to any other cooperative nodes. The rate $\lambda_{M,v}$ of encountering the malicious node and the cooperative node is set to be 0.01 or 0.1. The size of buffer B is set to be 10. For a warm-up period in each simulation experiment, the nodes generate and distribute 10,000 messages. Unless stated otherwise, the message generation rate Λ_M is set to be 5 and the total suspicious score $S^{(v)}_k$ $(v, k \in \mathcal{N}_C, v \neq k)$ is calculated after 10,000 unit times are elapsed.

B. Results

Figure 3 shows the total suspicious scores $S^{(0)}_k$ $(k = 1, 2, \ldots, 99)$ that node 0 evaluates. The total suspicious score $S^{(0)}_{50}$ of the malicious node $M = 50$ is higher than the threshold value t_h. On the other hand, the total suspicious scores $S^{(0)}_k$ $(k \in \mathcal{N}_C)$ are smaller than the threshold value t_h. Therefore, node 0 can estimate that node 50 is malicious and other nodes are cooperative. This result indicates that our proposed method can differentiate the malicious node from the cooperative nodes.



(a) $\lambda_{M,v} = 0.1$

(b) $\lambda_{M,v} = 0.01$

Figure 3. Total suspicious score $S^{(0)}_k$.

Figure 4 shows the total suspicious score $S^{(0)}_{50}$ as a function of the elapsed time t. For small t, the total suspicious score $S^{(0)}_{50}$ is smaller than the threshold value t_h, and thus node 0 cannot identify the malicious node. On the other hand, for large t $(t > 100)$, the total suspicious score $S^{(0)}_{50}$ exceeds the threshold value t_h. Therefore, when the suspicious scores are sufficiently updated, our proposed method can identify the malicious node.

Figure 5 shows the total suspicious score as a function of the message generation rate Λ_M of the malicious node. For $\Lambda_M > 1$, $S^{(0)}_{50}$ and $S^{(0)}_1$ are larger and smaller than the threshold value t_h, respectively. Our proposed method can detect the malicious node when the malicious node frequently generates messages. However, for $\Lambda_M = 0.5$, the total suspicious score of the malicious node $S^{(0)}_{50}$ and the cooperative node $S^{(0)}_1$ are smaller than the threshold value t_h. This result indicates that our proposed method cannot identify the malicious node when the malicious node generates slightly more messages than the cooperative nodes.

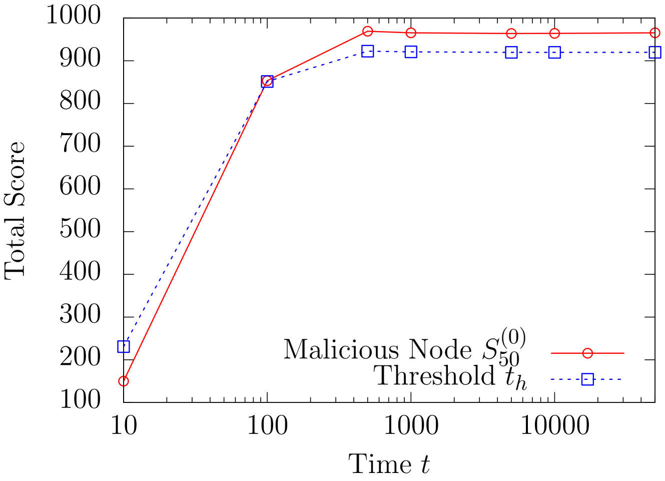


Figure 4. Total suspicious score as a function of elapsed time t.

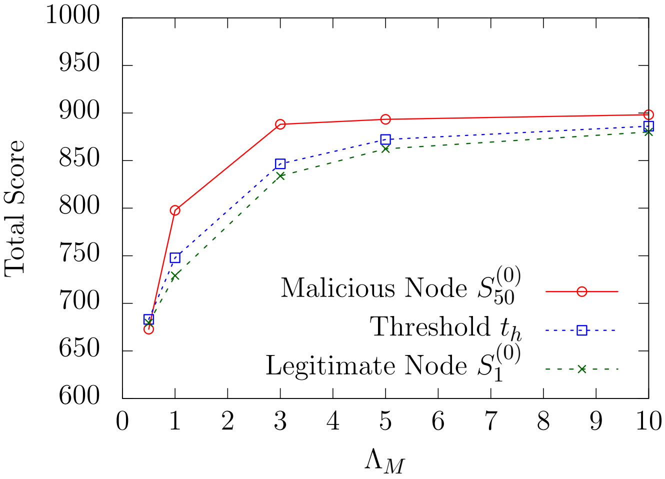


Figure 5. Total suspicious score as a function of message generation rate Λ_M of the malicious node.

V. Conclusion

In this paper, we proposed the detection method against the message flooding attacks, where the malicious node generates and distribute unnecessary messages to discard legitimate messages. In our proposed method, nodes records and exchanges the suspicious scores. After the suspicious scores are sufficiently updated, our proposed method discriminate between the malicious node and the cooperative nodes. Through simulation experiments, we showed that our proposed method can identify the malicious node if the suspicious scores are sufficiently updated. However, when malicious nodes generate just slightly more messages than cooperative nodes, in our proposed method, it is difficult to determine between malicious and cooperative nodes. We leave this problem for future work.

Acknowledgment

This work was partially supported by Grant-in-Aid for Young Scientists (B) of the Japan Society for the Promotion of Science under Grant No. 17K12679.

References

[1] Y. Cao and Z. Sun, “Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 654–677, 2013.

[2] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, “Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–540, 2012.

[3] A. Vahdat and D. Becker, “Epidemic routing for partially-connected ad hoc networks,” *Duke Technical Report*, 2000.

[4] M. Grossglauser and D. N. C. Tse, “Mobility increases the capacity of ad hoc wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[5] T. Kimura, Y Kayama, and T. Takine, “Home base-aware store-carry-forward routing using location-dependent utilities of nodes,” *IEICE Transactions on Communications*, vol. 100, no. 1, pp. 17–27, 2017.

[6] T. Matsuda and T. Takine, “(p, q)-Epidemic routing for sparsely populated mobile ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 783–793, 2008.

[7] G. Dini and A. L. Duca, “Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network, ” *Ad Hoc Networks*, vol. 10, no.7, pp. 1167–1178, 2012.

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

[9] M. Alajeely, R. Doss, and V. Mak-Hau, “Packet faking attack: A novel attack and detection mechanism in OppNets,” *Proc. of International Conference on Computational Intelligence and Security*, pp. 638–642, 2014.

[10] W. Khalid et al., “A taxonomy on misbehaving nodes in delay tolerant networks,” *Computers & Security*, vol. 77, pp. 442–471, 2018.

[11] T. Idezuka, T. Kimura, and M. Muraguchi, “Behavior Analysis of Flooding Attacks in Sparse Mobile Ad-Hoc Networks,” *Proc. of IEEE International Conference on Consumer Electronics - Taiwan*, pp. 1–2, 2018.

[12] R. Groenevelt, P. Nain, and G. Koole, “The message delay in mobile ad hoc networks,” *Performance Evaluation*, vol. 62, no. 1–4, pp. 210–228, 2005.