A Solution for Prevention of Selective Dropping and Selfish Attacks in Opportunistic Networks

Samaneh Rashidibajgan
Department of Computer Science, Rostock University, Albert-Einstein Strasse 22, Rostock, Germany

Keywords: Opportunistic Networks, Dropping and Selective Dropping Attack, Selfish Attack, Game Theory.

Abstract: Opportunistic Networks (OppNet) are based on routing messages from a node to another node, from a source to the destination. There is not a connection to the Internet in these networks and nodes play routers role, So it is important that all of the nodes participate in the routing protocol. These networks have high potential to vulnerable against "Dropping and Selective Dropping Attacks" and "Selfish attacks". Some nodes may prefer to discard some messages in order to save their Battery life, memory space and so on, while they use the network services. It causes an interruption in the network and makes a high delay for messages. In this paper, we propose a new method based on Game Theory to prevent these attacks against OppNet, and we will prove that our strategy is a Nash equilibrium. Also we will discuss that our algorithm is resistance against various attacks.

1 INTRODUCTION

In Opportunistic Networks, there are any fundamental infrastructures and devices which people carry with themselves, like their cell phones and tablets, will save, carry and forward messages from a source to the destination. There are many attacks against these networks, which two of them are "Dropping and Selective Dropping Attacks" and "Selfish attacks" (Alajeely et al., 2015). Each node in the network has an important role and acts as a router. If a node discards some messages without a reason or it will be selfish, it causes the whole network performance fall down. So preventing these attacks are very important in OppNet.

In "Dropping and Selective Dropping Attacks" malicious nodes drop all or some of their received packets, and the sender could not find that their messages are discarded. In "Selfish Attacks", some nodes may use network services, but refuse to cooperate with other nodes to carry and forward their messages.

Some proposed algorithms in the literature for detecting selective dropping attacks are as follows:

In order to detecting selective dropping attacks, a multi-dataflow topology (MDT) scheme was used in (Sun et al., 2007). In this algorithm, a network is divided into some clusters which they have overlap with each other, so messages are sent from different paths with redundancies. If a node discards a message, other nodes will send it. In OppNet, we don't have knowledge about the structure of a network and nodes are not always online, so this algorithm is not really useful in OppNet.

Hai and Huh (Hoang and Huh, 2008) proposed a lightweight detection scheme for Selective dropping attacks. In this algorithm, each node monitors two hop neighbors and considers a threshold for them. If the malicious counter crosses the threshold, then this node will be introduced as a malicious node and other nodes will omit it from their neighborhood lists. This algorithm is not resistance against cheating.

An ant based algorithm was introduced in (Kumari and Paramasivan, 2015) for detecting selective forwarding attacks. Some ant nodes are used in this algorithm in order to collect knowledge about misbehavior nodes. Then, these collected data are used for calculating a trust value, and a threshold is used for detecting an attack.

A repeated continued non cooperative game for detecting selective discard attacks was introduced in (Liao and Ding, 2015). In this algorithm, nodes monitor their neighbors in order to find more reliable neighbors. Authors calculated the best response for each player in their scheme and then a Nash equilibrium for the game was calculated.

Also some algorithms for detecting selfish attacks are as follows:

COOPON was introduced in (Jo et al., 2013), (Su-

Rashidibajgan, S.
A Solution for Prevention of Selective Dropping and Selfish Attacks in Opportunistic Networks.
DOI: 10.5220/0006233200300035
In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pages 30-35
ISBN: 978-989-758-245-5

jitha et al., 2015). This algorithm uses deterministic channel allocation information for detecting selfish attacks. This approach was designed for cognitive radio ad hock networks. Authors have used autonomous and cooperative characters of ad-hock networks in order to increasing the detection reliability.

Kargl and colleagues (Kargl et al., 2004), used multiple sensors in parallel in order to detecting selfish nodes. They have used an iterative probing for detecting selfish nodes. When a sender does not receive acknowledgment from a receiver (X_n) for a certain time of t, it will send a probe packet to it. If there won't be a reply in a certain time, it will send a probe packet to X_{n-1}, and it will continue this process until it receives an answer or reaches to X_1. When it receives a message from X_i, it will find that X_{i+1} is a selfish node. There is not stable path between two nodes in OppNet, so it is impossible to use this algorithm in OppNets.

In (Mittal, 2015), authors used an agent based technique for detecting selfish attacks. Every node in the network works as a monitor module and it checks its neighbors. Then they judge each node according to the received information from its neighbors. There is a probability to receive wrong information from neighbors in this algorithm.

All of these algorithms are not suitable for Opportunistic Networks. In OppNet, nodes are moving and we don't have a stable path between the source and a destination. In most of the mentioned algorithms, authors assumed that they have a clear network structure while the topology of the network in OppNet is changed frequently and nods attend and leave the network mostly. So, because of the erratic structure of OppNet, non of the proposed algorithms can be used in OppNet.

In this paper, we propose a method for prevention of Dropping and Selective Dropping Attacks and Selfish Attacks in OppNet without knowledge about a network topology or density of nodes. Our algorithm is based on Game Theory, and we prove that our method is a Nash Equilibrium and any node has motivation to violate it. We define a best strategy which players will play in a good history and when one of them disobey from rules (bad history), other players won't play with it. In this order, any player could increase its whole payoff by one game violation.

We consider the following assumptions in this paper:

1. We consider an Opportunistic Network with limited number of nodes (For example a conference in a department of an university which participants use OppNet.)
2. Every node has a pair of private and public key, which it could share its public key with trusted nodes. When nodes produce a message, they sign it by their private key.
3. Nodes use a trust function in order to detect trusted nodes, and they send and receive messages only with trusted nodes (We consider the trust structure which is introduced in (Rashidibajgan, 2016)).
4. When nodes are in communication range of each other, they should be sure about the trust of other nodes by using a trust function and then exchange information.
5. Nodes are moving frequently.
6. After a period of time, a node will visit most of the nodes in the network

The rest of this paper is organized as follow: we describe our method in the section two. In the section three we prove that our method is resistance against various attacks, and conclude our work in the section four.

2 DESCRIPTION OF METHOD

We consider an Opportunistic network which has some limited nodes, for example a conference which cellphones of participants are our nodes, and they can connect to each other via WiFi for sending and receiving messages. This Opportunistic network could help participants to find people how have a similar interest field in the science. When nodes are in the communication range of each other, they exchange some messages and also some parts of their tables, and after that each one updates its table.

In our algorithm, we have designed some tables and according to these tables, nodes give score to each other in the network. We aim to give more scores and priorities to the nodes which participant in the network, and recognize selfish nodes and selective dropping attacks. Nodes will consider history of other nodes and decide to play send or discard messages in the game.

Each node in the network has a delivery table (DT), and each message and each node has an ID in the network and we call them MID and NID respectively. When a node receives a message, the ID of the message (MID) and the ID of the node which directly delivered this message (LID) and ID of the sender of the message (SID) are saved in DT (according to the Figure 1). DT has another column with the name of NKN, and it saves ID of neighbors how sent this row of DT to them recently. When a node receives

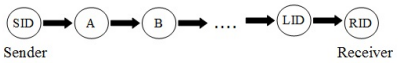

Figure 1: Different nodes from a source to the destination.

a message, the next row of the delivery table will be DT(LID,SID,MID,-). Table 1 shows the structure of DT. When nodes are in the communication range of each other, they exchange some parts of their delivery tables and after that, each node updates its DT. The process of exchanging and updating NKN field of DT are done as follow:

1. When a node sends a row of its DT to a neighbor, adds the NID of this neighbor to NKN. So they wont send repetitive information to a neighbor.
2. When a node visits the sender of a message (SID) and forwards/has forwarded this row of the table to at least n neighbors, it can delete this row in its table.
3. When there will be new information and a node dose not have free space in its DT, it can delete rows according to the first come first out algorithm.

Table 1: Structure of a DT table in the receiver (RID) node.

LID	SID	MID	NKN

Also nodes have another table which they give score to other nodes in it and we call it Score table (SC). Score table has two parts: Positive Score (SCp) and Negative Score (SCn). The structure of SC table is shown in Table 2. Score tables have two fields of NID and Score. Nodes after exchanging their delivery tables, they update their Score tables too.

SCp is updated as follows:

1. If NID of the LID in DT is not exist in the SCp of a node, it adds ID of this node to the table and gives one score to it.
2. If NID of the LID in DT is currently in the SCp of a node, it increases this score by 1.

Table 2: Structure of a SC table in a node.

SCp		SCn	
NID	Score	NID	Score

On the other hand, when they recognize a node is lying about visiting other nodes (described in section 3), they put ID of this node in the Negative Score tables (SCn) as follows:

1. If NID of the conflict intermediate node is not exist in the SCn of a node, it adds this and gives one score to it.
2. If NID of the conflict intermediate node is in the SCn of a node, it increases this score by 1.

A node which is in the Negative Score table (SCn) could not send its messages for $(n \cdot score)$ periods of time (as punishment) because other nodes refuse to carry its messages. After passing $(n \cdot score)$ periods of time, ID of this violate node will omit from the SCn of nodes, and it could try to cooperate in sending and receiving data and increases its positive score. Actually, SCp and SCn are two lists about ID of nodes which have good and bad history respectively and their scores. When a node cooperates in the forwarding a message or lying, it increases its score in SCp and SCn respectively. Each table has m rows records about m nodes with higher scores. Also, nodes which participate for sending and updating their tables receive α points as an encouragement (section 3).

These two tables, DT and SC, help nodes to learn about environment. They can observe each other and find which nodes are cooperating and which nodes are violating rules of the network.

We consider two kinds of history for each player in a game:

Good history: each node assumes that since a node forward my messages or messages of other nodes (according to the SCp), I will forward its messages. In Other words, since a node plays ”Sending” I will forward its messages and I will play ”Sending” too. Furthermore, nodes with higher scores will have higher priority.

Bad history: each node assumes that if a node constantly discards my messages or other nodes' messages, I wont accept to carry its messages. when a node started to play ”Discard” in a period of a game and its name is in SCn, I won't forward its messages.

Best Strategy of the Game (ST): each node plays ”Sending” at a good history and plays ”Discard” at a bad history.

In the following, we will prove that non of the player can increase its payoff at some histories by one step Discarding a message. We prove that if the next period of the game (the next sending message) will be as important as this game (sending current message) and nodes want to continue sending and receiving messages, they do not have violation motivation. We consider δ as this dependence of nodes to the future. They will know that if they won't send messages of other nodes, their messages wont be sent. Nodes have motivation of violation for increasing their payoff, but in the following we will prove that if they violate the rules of the game, they could increase their

payoff only for one period of the game and generally their overall payoff will decrease.

Table 3: Payoff for different playing games.

	Send	Discard
Send	$1,1$	$-L,(1+G)$
Discard	$(1+G),-L$	$0,0$

According to the Table 3, when both players A and B in the network, forward messages of each other, they receive 1, and if both discard messages, they will receive 0. If a node, for example node A, violates the rule and does not send messages of B while its messages will be sent by B, the violator node A receives $1+G$ (1 means its message is sent and G could be other advantages like saving Battery, memory etc. which node A gain) and B receives $-L$ (it carries and forwards a message while its message is not sent). Node A receives an advantage, but only for one period of the game. According to the strategy of the game, it wont receive services in the next periods and its total payoff (u) will fall down, so it won't have violation motivation.

$$ST^t = \left\{ \begin{array}{ll} Send & t=1 \\ Send & ST^{t-1}(Send,Send) \& t>1 \\ Discard & otherwise \end{array} \right\}$$

It means that when node A is in the communication range of another node B, if it is the first time which they want to send and receive a message, they play send, and for $(t>1)$, if the node B played send in the previous time $(t-1)$, node A plays send in this period of time (t) and otherwise node A discards node B's messages.

$$u_A(Send^\infty, Send^\infty) = 1+\delta+\delta^2+\delta^3+\cdots \quad (1)$$

$$0<\delta<1 \quad (2)$$

$$u_A(DiscardDiscard^n, SendDiscard^n) = 1+G+\underbrace{0+0+\cdots}_{n} \quad (3)$$

a node wont have violation motivation if:

$$u_A(DiscardDiscard^n, SendDiscard^n) \leq u_A(Send^\infty, Send^\infty) \quad (4)$$

$$1+G \leq \frac{1}{1-\delta} \quad (5)$$

$$1-\delta \leq \frac{1}{1+G} \quad (6)$$

$$\delta \geq 1-\frac{1}{1+G} \quad (7)$$

The equation of $\delta \geq 1-\frac{1}{1+G}$ is a Nash Equilibrium and nobody has motivation of violation this. It is the state which both players could achieve their highest payoff and with violation won't achieve more.

So nodes with high contributions will receive high scores and their messages will be accepted with more nodes, and they have more motivation for accepting, carrying and sending of messages of other nodes.

when node A has n neighbors, node A can play this game with them separately.

3 EVALUATION

In this section, we analyze the proposed algorithm from some perspectives, and we will discuss that our algorithm is resist against some attacks and it has high performance.

1. We have considered a network with 100 nodes and evaluated behavior of nodes in two situations: in a normal network, and in a network with the proposed structure. Also we assumed that nodes could remember their previous activities and any new node was added to the network during this simulation. We have calculated the performance of the network according to the following equation:

$$\text{Performance} = \frac{\text{All nodes in the network - Selfish nodes}}{\text{All nodes in the network}} \quad (8)$$

According to the Figure 2, in the proposed algorithm nodes will learn that if they will be selfish, their messages wont be sent. As a result, they wont have motivation for violating the rules and be selfish after some periods of time while in a normal network around 30 percent of nodes prefer to be selfish.

According to the Figure 3, the proposed algorithm has better performance, and after a while any node has motivation for violation.

2. According to the Table 1, when two nodes are in the communication range of each other and exchange their DT tables and one of them finds that the another node is the sender of one of its messages, the following items could happen.
   (a) both SID and MID are correct and there is not a problem. Then the source will know its message received and this row of the DT could be omitted from the table of the sender.
   (b) SID is correct but MID is not correct. In this situation the sender declares that the message is changed or he did not send this message. In this

Table 4: Summery of different states for a row in a DT.

Sender	Message	Cause	Action
SID ✓	MID ✓	This row can omit from the table	-
SID ✓	MID ×	The SID did not send this message The message was changed	The sender signs the message with its private key
SID ×	MID ✓	It dosen't happen	-
SID ×	MID ×	The sender is not exist in the network The message was produced by a fake SID	The LID is suspect to be a malicious node

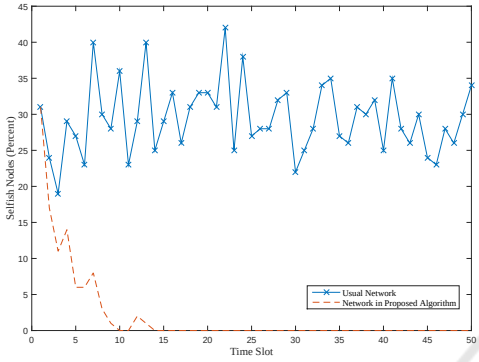

Figure 2: Amount of selfish nodes in a normal network and in a network with the proposed structure.

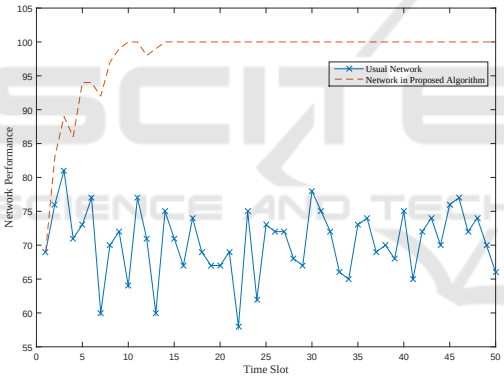

Figure 3: Performance of the network in a normal situation and with the proposed structure.

case, the source node and LID are suspected to lying, but also it is possible that other intermediate nodes caused this problem. If every node signs a message with its private key, this problem wont happen.

(c) SID was not found in the network. It means whether SID existed and produced this message and then left the network, or this message was produced by a fake NID which never existed in the network. In both situations, after n periods of time, the message is not valid anymore. In this state LID is a suspect node. If every node be care to accept messages from the trusted nodes, this option wont happen. So when they wont visit a sender for n periods of time (each node will visit most of the nodes in the network after n periods of time), we can ask others about this sender, if nobody knows about this SID, we consider LID as a malicious node, because it didn't care to accept a message from a trusted node or it made a fake message.

the summery of mentioned items are in Table 4

3. Another type of malicious activity relates to the nodes which disobey to give scores to other nodes as follows:

(a) a node sends some rows of its DT but it's neighbor does not update its score table.

(b) a node does not send some rows of it's DT for neighbors for updating.

A node may do these activities with the hope that when others don't achieve more scores, its score will increase more than them during the time. In order to omit this motivation, we give a point to the nodes which send some rows of their tables and also we give the scores to nodes which update their tables. We consider this point as α which $0 < \alpha < 1$, and according to the Table 5, if α will be higher than the score which a node receives during time $(\frac{Sc}{\delta})$, nodes do not have motivation of discarding some rows of their DT or do not update their SC tables.

Table 5: Received scores during a period of time for cooperated nodes and non cooperated nodes.

	update SC	don't update SC
send DT	α, α	$\alpha, (\frac{Sc}{\delta}) - \alpha$
don't send DT	$(\frac{Sc}{\delta}) - \alpha, \alpha$	$(\frac{Sc}{\delta}) - \alpha, (\frac{Sc}{\delta}) - \alpha$

When $(\frac{Sc}{\delta}) < \alpha$, nodes do not have violation motivation. After updating their SC tables, they should exchange their tables to be sure that update is done and sign SC of each other with their private keys.

4. A node can receive a message and does not give a point to LID. In order to prevent this, each node can know its score, and when it sends a message to a destination, increases its score and asks receiver to sign it with its private key. When a node finds

that a neighbor didn't give score to it in its SC, it can complain and introduce this node as a liar.

5. A node may broadcast a null message with the aim of receiving more scores. For solving this problem, when a node receives a null message or meaningless message, it will consider this SID as a liar.

4 CONCLUSION AND FUTURE WORKS

In this paper we proposed a new method based on Game theory. We defined some tables for each node and gave positive and negative scores to the nodes in the network. Nodes will receive priority for sending their messages according to their positive scores and they make a good history for themselves, and nodes with negative scores do not have allowance to send their messages for some periods of time and they will have bad history. When a node is in the communication range of another node, it plays sending in the good history and plays discard in the bad history. We proved that this strategy is a Nash equilibrium and non of the players have violation motivation. Also, we discussed about various attacks against the network according to the various fields of tables and we proved that our algorithm is resistance against attacks. Furthermore, we showed that during some periods of time, nodes wont have motivation to be selfish and the performance of the network will be in the higher position.

In our algorithm, we have assumed that the OppNet is implemented in a limited area like a conference in a department of a university which all of the nodes are registered in the network, so they can receive public and private keys. In a large area OppNet, like a city, it is almost impossible for nodes to connect to a third party for receiving public and private keys. Developing a method for sharing a public key in OppNet and as a result extend our algorithm in a large area is a part of our work in the future.

Furthermore in this paper, we have considered selective dropping and selfish attacks, but this work can be developed to detect other attacks. We intend to study other kinds of attack against opportunistic networks and complete our intrusion prevention plan in the future.

REFERENCES

Alajeely, M., Doss, R., and Ahmad, A. (2015). Security and trust in opportunistic networks–a survey. In *IETE Technical Review*, pages 1–13. Taylor & Francis.

Hoang, T. H. and Huh, E. (2008). Detecting selective forwarding attacks in wireless sensor networks using two-hops neighbor knowledge. In *Seventh IEEE International Symposium on Network Computing and Applications, 2008. NCA'08*, pages 325–331. IEEE.

Jo, M., Han, L., Kim, D., and In, H. (2013). Selfish attacks and detection in cognitive radio ad-hoc networks. In *IEEE network*, volume 27, pages 46–50. IEEE.

Kargl, F., Klenk, A., Schlott, S., and Weber, M. (2004). Advanced detection of selfish or malicious nodes in ad hoc networks. In *Security in Ad-hoc and Sensor Networks*, pages 152–165. Springer.

Kumari, S. and Paramasivan, B. (2015). Ant based defense mechanism for selective forwarding attack in manet. In *31st IEEE International Conference on Data Engineering Workshops (ICDEW), 2015*, pages 92–97. IEEE.

Liao, H. and Ding, S. (2015). Mixed and continuous strategy monitor-forward game based selective forwarding solution in wsn. In *International Journal of Distributed Sensor Networks*, pages 1–13. Hindawi.

Mittal, S. (2015). Identification technique for all passive selfish node attacks in a mobile network. In *International Journal of Advance Research in Computer Science and Management Studies*, volume 3, pages 46–51. IJARCSMS.

Rashidibajgan, S. (2016). A trust structure for detection of sybil attacks in opportunistic networks. In *11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 347–351. IEEE.

Sujitha, R., Scholar, P., and Poornima, N. (2015). Efficient detection of selfish attacks in cognitive radio networks using coopon algorithm. In *International journal of advanced research trends in Engineering and technology*, volume 2, pages 84–91. IJARTET.

Sun, H., Chen, C., and Hsiao, Y. (2007). An efficient countermeasure to the selective forwarding attack in wireless sensor networks. In *TENCON 2007-2007 IEEE Region 10 Conference*, pages 1–4. IEEE.