International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 7 Issue 1 –MARCH 2014.

A DELAY- TOLERANT PROBABILISTIC REBROADCAST FOR MOBILE AD HOC NETWORKS

Abstract-Twitter enables us community - based opportunities to engage, share and interact short text messages through online social networking .The community value oriented and related services like tweets that contain context information about 140 characters and URL links by using URL shortening service that are threatened by spammers, content polluters, malware disseminators. It is an effort made for preserving community value and long term success lead to propose the technique ”supervised learning based classifier” for detecting suspicious URLs in Twitter. In this proposed approach two key components are utilized to detect suspicious URLs. They are, tracing the suspicious URLs by discovering correlated URL redirected chains using the frequently shared URLs and filtering out the malicious URLs by building a statistical classifier using numerous tweets collected from the Twitter public timeline. More precisely and effectively this approach focuses on the detection of suspicious URLs in Twitter than that of already prevailing detecting system.

Keywords: Suspicious URL, twitter, URL redirection, conditional redirection, classification

I.INTRODUCTION

Being a social animal no human being can live in isolate, every one gets satisfied if he/she is socially recognized. Social recognition can be achieved through sharing of ideas, knowledge,

messages like news, trending topics, jokes and other information by means of communicative modes available for us. Now we are in the computerized world, online social networking made sharing in the easiest possible way. Among available online social networking services Twitter is popularly known online social networking and micro-blogging service that permits users to post and share their information all the way they need[1]. The information shared by the users in Twitter consist of short text messages (140 characters) denoted as tweets. While sharing, if a user „A‟ post the tweets in the Twitter public timeline, it will be reflected to his/her followers and sends the tweets to the specific Twitter user „B‟ too, by indicating @b in the tweet, unlike status updates of tweets can also be delivered to users those who are not the follower of user‟A. when user need to share the URLs via tweets, can use URL shortening services to reduce URL length, because tweets contain limited number of characters only. TinyURL.com and bit.ly are commonly used services and a shortening service t.co. is also provided by Twitter[2].

The unauthorized users have chances of using Twitter as a tool to spread malicious links for advertising to generate sales, virus, disseminate pornography, phishing, unsophisticated users to legitimate users and hijacking or simple just to compromise system reputation .All because of short in length of tweets and usage of shortened malicious URLs that redirect the Twitter user to external

servers. Attackers not only pollute real time search but can interfere on statistics presented by tweets mining tools and consume more extra resources from users and systems too.

For malicious links, inevitable Twitter spam detection schemes have been proposed and still continuing[3]. These schemes are classified as,

1) Account feature based : use the distinguished features of spam accounts such as date of account creation, the number of followers and friends and the ratio of tweets having URLs.

2) Relation feature based: rely upon more robust features like the distance and connectivity apparent in the Twitter graph.

3) The message feature based: focused on the lexical of messages.

In the recent scenario tremendous efforts are made for detecting suspicious URLs and number of detecting schemes have also been introduced. They may be executed in virtual machine honey spots such as honey monkey and wepawet by using static and dynamic crawlers. Here classification are in accordance of several features and including lexical features of URLs, URL redirections, DNS information and the HTML contents of the landing pages.

1.1.concepts and goal

Our goal is to develop the suspicious urls detection against conditional redirection .Because an attackers can use the

mechanism of conditional redirections, to redirect the normal users into malicious landing pages. In which an attacker creates a long URL redirect chains that can be reduced by using URL shortening service, such as bit.ly and t.co as well as the attackers own private redirection servers. By using the redirection server, the attackers redirect the normal visitors into malicious landing pages. The

attacker then uploads the tweets with the initial URL of the redirect chains to Twitter. Later, when a user or crawler visits the initial URL, they will be redirected to an entry point of the intermediate URLs that are associated with private redirection severs. These redirection server check the current user is the normal browsers or crawlers. If the current visitors seems to be a normal, the server visitors to a malicious landing page. If not, they will redirect the visitors to a benign landing page. Therefore, the attackers can selectively attack normal users while deceiving investigators.

II .RELATED WORK

For route discovery, broadcasting is an effective data dissemination mechanism. In high dynamic networks, the routing overhead associated with broadcasting is quite large [9]. Experimental results showed that the rebroadcast is very expensive and consumes too much network resource. Broadcasting causes many problems such as redundant retransmissions, collisions and contentions [5]. To improve the routing performance, optimizing the broadcasting in route discovery is an effective solution. Each node forwards a packet with a probability in the gossip based approach proposed by Haas et al. [10]. For large networks, the simple gossiping protocol uses up to 35% fewer messages than flooding, with improved performance. Gossiping exhibits bimodal behavior in sufficiently large networks. In some executions, the gossip dies out quickly and hardly any node gets the message. In the remaining executions, a substantial fraction of the nodes gets the message. The fraction of executions in which most nodes get the message depends on the gossiping probability and the topology of the network [9].Based on coverage area and neighbour confirmation

Kim et al. [8] proposed a probabilistic broadcasting scheme. This scheme uses the neighbour confirmation to guarantee reachability and also set rebroadcast probability by using the coverage ratio. A Dynamic Probabilistic Route Discovery (DPR) scheme proposed by Abdulai et al. [12] is based on the knowledge of neighbours. According to the number of its neighbours and the set of neighbours which are covered by the previous broadcast, each node determines the forwarding probability. This scheme does not consider the neighbours receiving the duplicate RREQ packet but only the coverage ratio by the previous node. Thus there is an extension for the DPR protocol.

III. OBJECTIVE OF THE DTPR MECHANISM

Optimizing the broadcast is the initial motivation of the system. Several methods have been introduced for the optimization of broadcasting in route discovery. The existing methods have their own merits and demerits. Thus a Delay Tolerant Probabilistic Rebroadcast is used to optimize the broadcasting in an efficient manner.

The main objectives of the Delay-Tolerant Probabilistic Rebroadcast is

- To reduce the number of redundant retransmissions.
- To reduce the routing overhead
- To increase the routing performance

IV.DELAY TOLERANT PROBABILISTIC REBROADCAST PROTOCOL FOR MANETs

The rebroadcast delay is calculated by using the upstream coverage ratio of an RREQ packet received from the previous node. The rebroadcast probability in the DTPR protocol can be calculated by combining the additional coverage ratio of the RREQ packet and the connectivity factor.

A. REBROADCAST DELAY AND UNCOVERED NEIGHBOURS SET

When node n_i receives an RREQ packet from its previous node s, it can use the neighbour list in the RREQ packet to estimate how many of its neighbours have not been covered by the RREQ packet from s. If node n_i has more neighbours uncovered by the RREQ packet from s, it can reach more additional neighbour nodes. To quantify this, the Uncovered Neighbours set $U(n_i)$ of node n_i can be defined as follows:

$$U(n_i) = N(n_i) - [N(n_i) \cap N(s)] - \{s\} \quad (4.1)$$

Where $N(n_i)$ and $N(s)$ are the neighbours sets of node n_i and s respectively. The node s sends an RREQ packet to node.

According to (4.1), initial UCN set can be obtained. Due to the broadcast characteristics of an RREQ packet, node n_i can receive the duplicate RREQ packets from its neighbours. Node n_i could further adjust the $U(n_i)$ with the neighbour knowledge. Each node should set a rebroadcast delay in order to sufficiently exploit the neighbour knowledge and avoid

channel collisions. The key to success for the proposed protocol is the choice of a proper delay because the scheme determines the delay time affects the dissemination of neighbour coverage knowledge. It could calculate the rebroadcast delay according to the neighbour list in the RREQ packet when a neighbour receives an RREQ packet and its own neighbour list. The rebroadcast delay $T_{rd}(n_i)$ of node n_i is defined as follows:

$$T_p(n_i) = 1 - \frac{|N(s) \cap N(n_i)|}{|N(s)|} \tag{4.2}$$

$$T_{rd}(n_i) = \text{Max Delay} \times T_p(n_i),$$

Where $T_p(n_i)$ is the delay ratio of node n_i, and Max Delay is a small constant delay. |.| is the number of elements in a set.

The above rebroadcast delay is defined with the following reasons. The node transmission order can be determined by the delay time. When the node s sends an RREQ packet, all of its neighbours n_i , i =1,2,3...,|N(s)| receive and process the RREQ packet. Assumed that node n_k has the largest number of common neighbours with node s, according to (4.2), node n_k has the lowest delay. Once this node rebroadcasts the RREQ packet, then more nodes can receive it, because node n_k has the largest number of common neighbours. Then, more number of nodes can exploit the neighbour knowledge to adjust their UCN sets. Of course, whether node n_k rebroadcasts the RREQ packet depends on its rebroadcast probability. The main objective of this rebroadcast delay is to disseminate the neighbour coverage knowledge more quickly. The node can set its own timer after determining the rebroadcast delay.

B. REBROADCAST PROBABILITY

The node which has a larger rebroadcast delay may listen to RREQ packets from the nodes which have lower delay. For example, if node receives a duplicate RREQ packet from its neighbour n_j, it knows that how many its neighbours have been covered by the RREQ packet from n_j. Thus, node n_i could further adjust its UCN set according to the neighbour list in the RREQ packet from n_j. Then, the $U(n_i)$ can be adjusted as follows:

$$U(n_i) = U(n_i) - [U(n_i) \cap N(n_j)] \tag{4.3}$$

After adjusting the $U(n_i)$, the RREQ packet received from n_j is discarded. The rebroadcast delay needs to be adjusted because the rebroadcast delay is used to determine the order of disseminating neighbour coverage knowledge to the nodes which receive the same RREQ packet from the upstream node. The node obtains the final UCN set after expiring the timer of rebroadcast delay of node n_i.The final UCN set contains the nodes that need to receive and process the RREQ packet. If a node does not sense any duplicate RREQ packets from its neighbourhood, its UCN set is not changed, which is the initial UCN set.

The final UCN set can be used to set the rebroadcast probability. The additional coverage ratio ($R_{ac}(n_i)$) of node n_i can be calculated as follows:

$$R_{ac}(n_i) = \frac{|U(n_i)|}{|N(n_i)|}, \tag{4.4}$$

This metric indicates the ratio of the number of nodes that are additionally covered by this rebroadcast to the total number of neighbours of node n_i . The nodes that are additionally covered need to receive and process the RREQ packet. As R_{ac}becomes bigger, more nodes will be covered by the rebroadcast, and more nodes need to receive and process the RREQ packet. So the rebroadcast probability should be set to be higher.

If each node connects to more than 5.1774 log n of its nearest neighbours, then the probability of the network being connected is approaching 1 as the number of nodes in the network n increases. Then, 5.1774 log n can be used as the connectivity metric of the network. The ratio of the number of nodes that need to receive the RREQ packet to the total number of neighbours of node n_i is the connectivity factor $F_{cf}(n_i)$. In order to keep the probability of network connectivity approaching 1, have a heuristic formula

$|N(n_i)|.F_{cf}(n_i) \geq 5.1774 \log n.$

The minimum $F_{cf}(n_i)$ can be defined as the connectivity factor, which can be find as follows:

$$F_{cf}(n_i) = \frac{N_{cv}}{|N(n_i)|}, \tag{4.5}$$

where $N_{cv} = 5.1774 \log n$, and n is the number of nodes in the network. From (3.5), observed that when $|N(n_i)|$ is greater than N_{cv}, $F_{cf}(n_i)$ is less than 1. That means node n_i is in the dense area of the network, then only part of neighbours of node n_i forwarded the RREQ packet could keep the network connectivity. And when $|N(n_i)|$ is less than N_{cv} , $F_{cf}(n_i)$ is greater than 1. That means node n_i is in the sparse area of the network, then node n_i should forward the RREQ packet in order to approach network connectivity. The additional coverage ratio and connectivity factor can be combined to

calculate the rebroadcast probability $P_{re}(n_i)$ of node $n_i$which can be defined as follows:

$$P_{re}(n_i) = F_{cf}(n_i).R_a(n_i), \quad (4.6)$$

where, if the $P_{re}(n_i)$ is greater than 1, then set the $P_{re}(n_i)$ to 1.

The above rebroadcast probability is defined with the following reason. The parameter R_a does not consider the relationship of the local node density and the overall network connectivity. The parameter F_{cf} is inversely proportional to the local node density. That means if the local node density is low, the parameter F_{cf} increases the rebroadcast probability which in turn increases the reliability of the DTPR in the sparse area. Thus, the parameter F_{cf} adds density adaptation to the rebroadcast probability.

Note that the calculated rebroadcast probability $P_{re}(n_i)$may be greater than 1. It just shows that the node must forward the RREQ packet when the local density of the node is so low. Then, node n_i need to rebroadcast the RREQ packet received from s with probability $P_{re}(n_i)$.

C. ALGORITHM DESCRIPTION

The formal description of the Neighbour Coverage Based Probabilistic Rebroadcast for reducing routing overhead in route discovery is shown below.

Algorithm 1. DTPR

Definitions:

RREQq: RREQ packet received from node q.

R_q.id: the unique identifier (id) of RREQq.

N(p) : Neighbour set of node p.

U (p, x): Uncovered neighbours set of node p for RREQ whose id is x.

Timer (p, x): Timer of node p for RREQ packet whose id is x.

1: if n_i receives a new RREQs from s then

2: {Compute initial uncovered neighbours set U $(n_i, R_s.id)$ for RREQs}

3: $U(n_i, R_s.id) = N(n_i) - [N(n_i) \cap N(s)] - \{s\}$

4: {compute the rebroadcast delay $T_{rd}(n_i)$ }

5: $T_p(n_i) = 1 - \frac{|N(s) \cap N(n_i)|}{|N(s)|}$

6: $T_{rd}(n_i)$ =Max Delay × $T_p(n_i)$

7: Set a Timer $(n_i, R_s.id)$ according to $T_{rd}(n_i)$

8: end if

9: while n_i receives a duplicate RREQj from $n_j$before Timer $(n_i, R_s.id)$expires do

10: {Adjust U $(n_i, R_s.id)$}

11: $U(n_i, R_s.id) = U(n_i, R_s.id) - [U(n_i, R_s.id) \cap N(n_j)]$

12: discard (RREQj)

13: end while

14: if Timer $(n_i, R_s.id)$ expires then

15: {Compute the rebroadcast probability $P_{re}(n_i)$}

16: $R_{ac}(n_i) = \frac{|U(n_i, R_s.id)|}{|N(n_i)|}$

17: $F_{cf}(n_i) = \frac{N_{cv}}{|N(n_i)|}$

18: $P_{re}(n_i) = F_{cf}(n_i).\ R_{ac}(n_i)$

19: if Random (0, 1) $\leq$ $P_{re}(n_i)$ then

20: broadcast (RREQs)

21: else

22: discard (RREQs)

23: end if

24: end if

V.SIMULATION ENVIRONMENT AND PERFORMANCE EVALUATION

A. Simulation Environment

To implement the DTPR protocol the source code of AODV in NS-2(v2.30) is modified. Using the NS-2 simulator evaluated the performance of the proposed DTPR and compared it with AODV AND DPR. The topology size taken for simulation is 1000m×1000m, in which 50 to 300 mobile nodes move for a simulation time of 50 seconds. The transmission range is 250 meters and the simulated traffic is Constant Bit Rate (CBR).

The parameters of simulation are summarized in table

TABLE 1

SIMULATION PARAMETER	VALUE
Simulator	NS-2
Topology Size	1000x1000
Transmission range	250 m
Bandwidth	2Mbps
Traffic type	CBR
Number of CBR connections	10,12...,15,...,18,20
Packet Size	512 bytes
Packet Rate	4 packets/sec
Pause Time	0s
Min Speed	1m/s
Max speed	5m/s

B. Performance Metrics

In order to evaluate the performance of the proposed DTPR protocol, it is compared with some other protocols using the simulator version NS-2. Broadcasting is an effective and fundamental data dissemination mechanism for many applications in MANETs. In this only route request is studied. In order to compare the routing performance of the proposed DTPR protocol, we choose the Dynamic Probabilistic Route Discovery (DPR) protocol which is an optimization scheme for reducing the overhead of RREQ packet incurred in route discovery in recent literature, and the conventional AODV protocol. The performance of routing protocols is evaluated using the following performance metrics:

• Collision rate on MAC: Taking the average number of packets (RREQ, RREP, RERR and CBR data packets) which are dropped resulting from the collisions at the MAC layer per second.

• Normalized routing overhead: The ratio of the total packet size of control packets (include RREQ, RREP, RERR and Hello) to the total packet size of data packets delivered to the destination

• Packet delivery ratio: The ratio of the number of data packets that are received by the CBR destinations to the number of data packets generated by the CBR sources.

• Average end-to-end delay: the average delay of successfully delivered CBR packets from source to destination node.

The protocol performance is calculated with both the varied number of nodes and collision rate.

C. Results

In the first experiment we compare the routing overhead with varied number of nodes, in which DTPR yields 74.9% less overhead than the existing AODV.

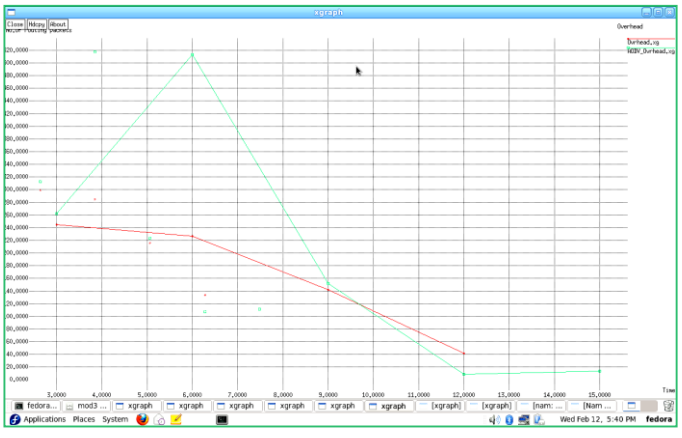

Fig 5.1 Routing Overhead of DTPR with AODV

Secondly we compare the packet delivery ratio with a varied number of nodes, in which DTPR increases the packet delivery ratio about 21.8% than the AODV.

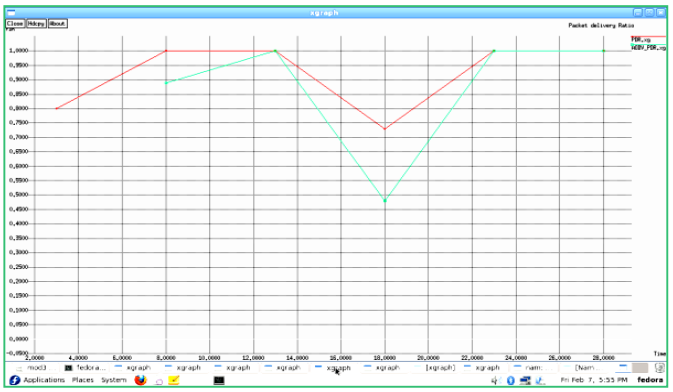

Fig 5.2 Packet Delivery Ratio with varied number of nodes

VI. CONCLUSION

In this paper, we proposed a delay-tolerant probabilistic rebroadcast protocol to reduce the routing overhead in MANETs. The additional coverage ratio and connectivity factor can provide neighbour coverage knowledge. We proposed a rebroadcast delay to determine the forwarding order and more effectively to exploit the neighbor coverage knowledge. Results of simulation show that the DTPR protocol has lesser rebroadcast traffic than the flooding and other existing optimized scheme. The DTPR protocol mitigates the network collision and contention because of less redundant rebroadcast, which in turn increases the packet delivery ratio and decreases the average end-to-end delay. The results of simulation show that in the high density network or the heavy traffic load, the proposed protocol has a good performance.

REFERENCES

[1] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On Demand Distance Vector (AODV) Routing”,RFC 3561.2003

[2] Johnson, Y. Hu, and D. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR) For IPv4”,RFC4728, 2007.

[3] H. Al Aamri, M. Abolhasan, and T. Wysocki “On Optimising Route Discovery in Absence of Previous Route Information in MANETs,” Proc. of IEEE VTC 2009-Spring, pp. 1-5,2009.

[4] X. Wu, H. R. Sadjadpour, and J. J. Garcia- Luna-Aceves, “Routing Overhead as A Function of Node Mobility: Modelling Framework and Implications on Proactive Routing,” Proc. Of IEEE MASS’07, pp. 1-9, 2007.

[5] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu.“The

Broadcast Storm Problem in a Mobile Ad hoc Network," Proc. Of ACM/IEEE MobiCom'99, pp. 151-162, 1999.

[6] Mohammed A, M. Ould-Khaoua, L.M. Mackenzie, C. Perkins, and D. Abdulai "Probabilistic Counter-Based Route Discovery for Mobile Ad Hoc Networks," Proc. of IWCMC'09, pp.1335-1339,2009.

[7] Williams B and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," Proc. ACM MobiHoc'02, pp. 194-205, 2002.

[8] Kim J, Q, Zhang, and D. P. Agrawal, "Probabilistic Broadcasting Based on Coverage Area and Neighbor Confirmation In Mobile Ad hoc Networks,"Proc. of IEEEGLOBECOM'04.

[9] Abdulai J.D, M.Ould-Khaoua, and L.M .Mackenzie, "Improving Probabilistic Route Discovery in Mobile Ad Hoc Networks," Proc. of IEEE Conference on Local Computer Networks, pp. 739-746, 2007.

[10] Haas Z, J. Y. Halpern, and L. Li, "Gossip-based Ad hoc Routing," Proc. IEEE INFOCOM'02, vol. 21,pp. 1707-1716, 2002.

[11] Peng S and X. Lu,"On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," Proc. Of ACM MobiHoc'00, pp. 129-130, 2000.

[12] Abdulai J.D, M. Ould-Khaoua, L. M Mackenzie, and A. Mohammed, "Neighbour Coverage: A Dynamic Probabilistic Route Discovery for Mobile Ad hoc Networks," Proc of SPECTS'08, pp. 165-172, 2008.

[13] Chen J, Y. Z. Lee, H. Zhou, M. Gerla, and Y. Shu, "Robust Ad Hoc Routing for Lossy Wireless Environment," Proc. of MILCOM'06, pp. 1-7, 2006 .