ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Bio-Inspired Proximity Discovery and Synchronization with Security Solutions for D2D Communications

Abhijeet Dholeshwar[1], Prof. Deepali P. Salapurkar[2]

PG Student, Department of Computer Engineering, Sinhgad College of Engineering, Pune, India[1]

Assistant Professor, Department of Computer Engineering, Sinhgad College of Engineering, Pune, India[2]

ABSTRACT: Device-to-Device (D2D) Communication is a fast emerging technology in recent years. It serves an end to end communication without depending upon any infrastructure. Various proximity models are present to identify and communicate with the different devices present in the vicinity of a particular device. In these services security should also be concerned as it is the least dealt area of D2D communication. In this work, a distributed mechanism for application aware Proximity Services (ProSe) in D2D communication is proposed. This method is a derivation of bio-inspired firefly algorithm which can achieve proximity discovery and synchronization at same time. The basic firefly algorithm has limitation for large scale system such as LTE-A D2D, which has been covered in the derived algorithm by enabling simultaneous neighbor discovery and service discovery as well as synchronization in physical communication timing. Secure neighbor discovery refers to a process of exchanging messages to discover and authenticate the devices in proximity. It involves spreading a common code to the communicating parties but unknown to the jammer. However, it is impossible to spread the common code without successfully discovering the proximate devices. This work proposes a jamming-resilient secure neighbor discovery scheme in combination with Firefly Algorithm. This scheme could be useful in securely discover and authenticate the neighboring nodes.

KEYWORDS-D2D communication; ProSe; LTE; JR-SND

I. INTRODUCTION

Device to Device (D2D) communication is gaining importance with a notable increase in wireless technologies and devices; and their usage. Due to such type of communication, our wireless devices such as mobile phones, tablets etc. could be used, even if no network infrastructure is present. It is helpful in certain scenario such as any disaster affected area where communications are destroyed. D2D communication can be achieved efficiently with the help of certain parameters such as neighbor discovery, message passing, connectivity etc. D2D communication gives better resource utilization, higher data transmission in networks.

The Third Generation Partnership Project (3GPP) Long Term Evolution (LTE) has introduced D2D application in many scenarios which requires direct access with and without infrastructure. In infrastructure based D2D communication, initiation of D2D communication is managed by Base Station (BS). User Equipment (UE) searches its neighbor and transmits data in the proximity in self-organized manner, without assistance from BS. Two devices will communicate when their proximity criteria is fulfilled. The main required criterion of proximity is geographical distance between devices. Proximity Service (i.e. ProSe) is defined in proximity context and it is an important feature of D2D communications. ProSe consists of device discovery and communication among them which are in close physical context.

The proximity discovery can be categorized in two contexts; physical communication and application discovery. In physical level proximity discovery, signal exchange among devices takes place whereas in application level discovery a device search another device with same interest in the network. To make communication among devices robust and efficient, there is need to combine the physical communication and application discovery.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

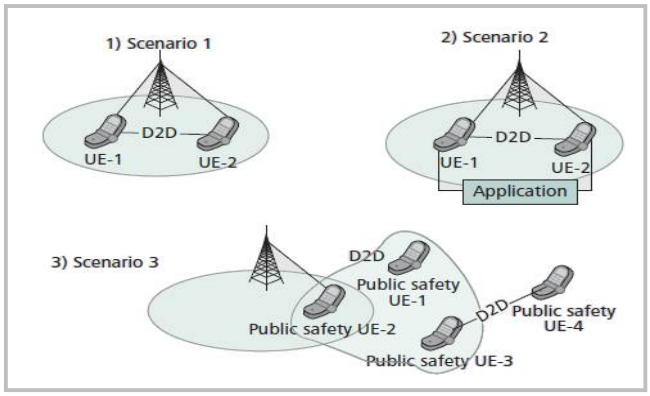

Fig.1. Types of D2D network

In some hostile environments, such as any emergency situation in disaster affected area or any military operation, it is necessary that the devices should communicate securely. It should avoid or tackle any types of attacks which could affect the communication and network. Secure neighbor discovery refers to the process that neighboring nodes exchange messages to discover and authenticate each other. As the basis of other network functionalities such as medium access control and routing, secure neighbor discovery need be frequently performed due to node mobility. The open nature of wireless technology can make it vulnerable to jamming attack. A jamming attack consists of transmitting noise-like signals intentionally by any adversary to prevent neighboring nodes from message exchange.

Traditional anti-jamming communications often depend on spread-spectrum techniques. These techniques require that the communicating parties use a common spread code which will be unknown to the adversary to spread the signals such that the transmissions are unpredictable and thus resilient to jamming. The most widely used spread code technique is direct-sequence spread spectrum (DSSS). In a DSSS system the sender spreads the data signal by multiplying it by an independent noise signal known as a spread code. It is a pseudorandom sequence of 1 and 1 bit values at a frequency much higher than that of the original signal. The energy of the original signal is thus spread into a much wider band. At the receiver end, this signal can be reconstructed to the original signal by multiplying the received signal by a synchronized version of the same spread code, which is known as a de-spreading process.

II. RELATED WORK

Bio-inspired computing is a field of study that loosely knits together subfields related to the topics of connectionism, social behaviour and emergence. It relies heavily on the fields of biology, computer science and mathematics. It is the use of computers to model the living phenomena, and simultaneously the study of life to improve the usage of computers.

In [6], a model of population of identical integrate-and-fire oscillators is studied. The coupling between oscillators is pulsatile: when a given oscillator fires, it pulls the others up by a fixed amount, or brings them to the firing threshold, whichever is less. The main result is that for almost all initial conditions, the population evolves to a state in which all the oscillators are firing synchronously. In [10], Wener-Allen et al. implemented decentralized Reachback Firefly Algorithm (i.e. RFA) on TinyOS-based motes and provided theoretical improved result. This algorithm is based on a mathematical model that describes how fireflies and neurons spontaneously synchronize. This algorithm accounts for realistic effects of sensor networks by allowing nodes to use delayed information from the past to adjust the future firing phase. Authors [11] proposed Meshed Emergent Firefly Synchronization (i.e. MEMFIS) which multiplexes synchronization word with data packet and adopt local clock upon reception of synchronizing nodes. A dedicated synchronization phase is mitigated, as a network-wide slot structure emerges seamlessly over time as nodes exchange data packets.

Several schemes have been proposed to enable two nodes to establish a secret spread code (or key) under the jamming attack. Some the schemes are given as follows. These methods are related to various jamming attacks and possible solutions to these attacks.

In [14], Strasser et al. proposed a method using Uncoordinated Frequency Hopping (UFH) to enable two communication parties without a common secret to establish a secret key. However, key latency problem and

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

communication overhead are not covered. Improvements in this technique are given in [16], [17], [18], [19] to reduce the key-establishment latency and communication overhead. Under the above techniques, the adversary can inject arbitrary many message fragments leading to a DoS attack. In [15], Jin et al. addressed the same problem by proposing an intractable forward-decoding and efficient backward decoding scheme based on DSSS. This scheme requires the sender to know the MAC address of the receiver which is unfortunately unknown before the sender successfully discovers the receiver.

III. PROPOSED WORK

The proposed work is a combined mechanism of Firefly Spanning Tree and Jamming-Resilient Secure Neighbor Discovery. The two mechanisms are explained as follows.

This firefly algorithm is inspired by the fireflies and the synchronization between the swarm of fireflies. Fireflies use its flash, which flashes at certain threshold interval, and locate other fireflies by recording their flashes and also gives its own status. Thus each firefly gets synchronized in the swarm. The similar model can be used by the UEs to achieve synchronization with an effective proximity discovery. The devices discover each other by sending signals and increasing the counter values. When the threshold is reached, proximate devices come into a synchronized network and start communicating. A constant topology is not possible in real application. So achieving synchronization is a difficult task. This was the problem with firefly algorithm

It is difficult to implement any similar method, for synchronization, on a randomly generated topology. Firefly Spanning Tree (FST) can provide a solution to it. FST includes forming a structure of heavy nodes; i.e. taking the nodes with better links; and establishing communication between them. According to FST, a D2D network can be formulated into a graph G(V,E), where vertices V are independent UEs and edges E are communication links between UEs. Links can be weighted using the strength of the Proximity Signal. The benefits of the mechanism in any randomly generated topologies can be extracted by finding a basic structure which is able to sustain the state of synchronization. Tree may be a suitable structure which will be helpful in synchronization. The theorem proving stability of tree is given in [1]. FST is a distributed algorithm which may construct a spanning tree with strongest signal strength on graph maintaining a distributed manner.

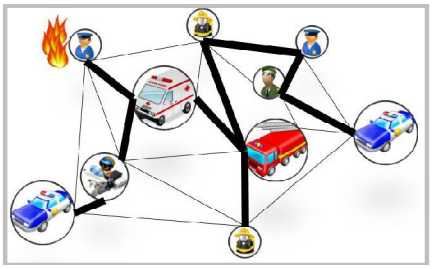

Fig.2. An Example of FST

JR-SND is a novel jamming-resilient secure neighbor discovery scheme for single-authority DSSS based MANETs. Inspired by random key pre-distribution schemes for sensor networks, JR-SND requires the MANET authority to generate a pool of secret spread-codes and pre-load every node with a constant number of spread codes randomly drawn from the pool prior to network deployment. During the network operation, two neighboring nodes can use their spread codes to conduct anti-jamming secure neighbor discovery via DSSS communication. In particular, JR-SND allows two neighboring nodes to directly discover each other if they share at least one common spread code unknown to omnipresent jammers or indirectly discover each other if there exists a multi-hop path connecting them, along which every two neighboring nodes have successfully discovered each other. As time goes by, the adversary may compromise some nodes to know their spread codes, but the non-compromised codes will remain secret. JR-SND can greatly mitigate the DoS attack because it can only be launched by the adversary using limited compromised spread codes which can fortunately be revoked after being identified.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

IV. METHODOLOGY

A. *Firefly Spanning Tree:*

The Firefly Spanning Tree will be used to discover the devices in proximity and to synchronize these devices. Preshared-Key-Based security scheme will be used to authenticate the proximate device based on a shared key and form a secure communication. It is a two-level algorithm. At the beginning, devices know only the weight of links which are connected to them and after the time every device knows which of its link belong to FST and synchronizes with remaining neighbors. The process involves two different proximity signals. **PSH** is used for the synchronization between sub-graphs.**PSG** is used for the regular operation of the basic firefly algorithm throughout the network. Illustration is shown in Fig. 5

By applying two-level firefly algorithm and a greedy algorithm, we can combine all available sub-graphs into one spanning tree. This could result in achieving robustness. Weight of FST is greater than or equal to weight of every single spanning tree.

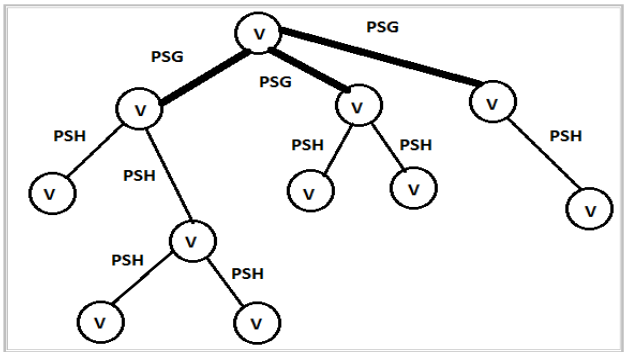

Fig.3. Signalling in FST

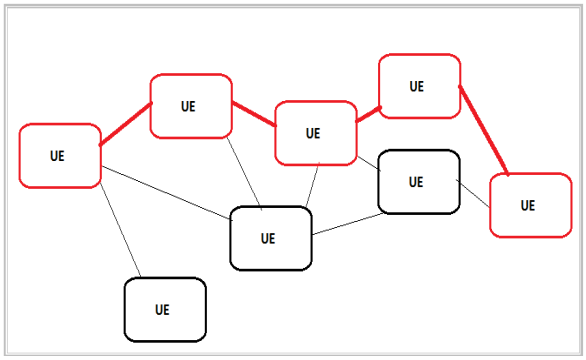

Fig.4. Basic Firefly Algorithm

B. *Jamming-Resilient Secure Neighbor Discovery*

A quorum-based spread-code pre-distribution scheme is used as a nontrivial adaption of existing key pre-distribution schemes. A direct neighbor-discovery protocol (D-NDP), a multi-hop neighbor-discovery protocol (M-NDP), and a location-aware multi-hop neighbor-discovery protocol (LAM-NDP) are presented further.

1) Random Spread-Code Predistribution:

The MANET authority generates a pool of random spread codes 'C'. Only the authority has the full knowledge of 'C' and uses the following method to distribute *m* spread codes to each node such that any code is shared by no more than *l* nodes. The distribution process consists of *m* rounds, during each of which each node is assigned one spread code. After *m* rounds, every node is preloaded with *m* spread codes, and every code is exactly shared by *l* nodes. The scheme permits new nodes to join the network later. In particular, the authority can assign the spread codes of a virtual node to a unique new node.

2) Direct Neighbor Discovery Protocol:

D-NDP can allow two physical neighbors with common spread codes to directly discover each other. During the network operation, each node periodically initiates neighbor discovery in a randomized manner. Specifically, in every interval of length T, each node initiates the D-NDP process once at a random time point. Consider nodes A and B, sharing a common spread-code. Let A start the initiation of D-NDP process. A broadcasts a HELLO message repeatedly for r rounds. In each round, the HELLO message is broadcast m times, and each time a distinct code in C is used for spreading. A HELLO message spread with Ci

$$A \rightarrow * : \{\text{HELLO}, IDA\}\mathbf{C}i \ ,$$

Where HELLO is a message type identifier of l bits, ID_A is ID of A, and *{} denotes the message spread with the spread code at the subscript. Each message in D-NDP is encoded with an error correcting code (ECC) such as to increase the transmission reliability. To synchronize with and de-spread any incoming message, node B buffers the received signal and tries to identify any message in the buffer using a sliding window algorithm. After de-spreading

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

HELLO message from A, node B knows that A is in its transmission range and spread-codes are matching. It then repeatedly sends an ECC-coded CONFIRM message spread with Ci;

$$B \rightarrow A : \{\text{CONFIRM}, IDB\}\mathbf{C}i .$$

Node B then starts to monitor Ci in real time. Similar to A transmitting HELLO message, node B keeps transmitting the CONFIRM message until receiving a response from A which can be de-spread with Ci. If B does not receive a response before its timer expires, it stops monitoring Ci in real time and considers A having moved away. Node A uses the same approach to de-spread Bs CONFIRM message and knows that B shares Ci with it. To conduct mutual authentication, node A computes a shared key K_{AB} using its ID-based private key and ID_B

$$A \rightarrow B : \{IDA, nA, fKAB\ (IDA|nA)\}\mathbf{C}i,$$

Since B is currently monitoring Ci, it can de-spread the above response in real time after negligible delay. Node B proceeds to compute a shared key K_{BA} based on its ID-based private key and ID_A. Thus using the keys A and B authenticate each other. B proceeds to transmit the following ECC-coded response

$$B \rightarrow A : \{IDB, nB, fKBA\ (IDB|nB)\}\mathbf{C}i,$$

After de-spreading the above response node A verifies using K_{AB}. If verification is successful A accepts B as logical neighbor.

3) Multi-hop Neighbor Discovery Protocol:

Two physical neighbors may fail to directly discover each other via D-NDP either because they have no common spread codes or because jammer J has compromised their common spread codes. Here MNDP allows two physical neighbors to indirectly discover each other as long as there is a jamming-resilient path connecting them, along which every two adjacent nodes have discovered each other. Illustration of the M-NDP operations is given in Fig. 5 as an example

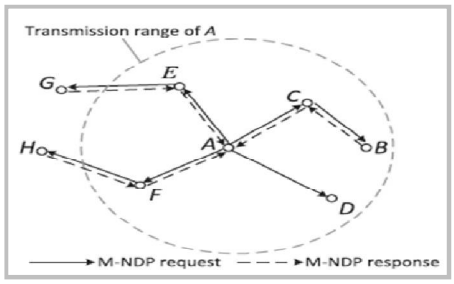

Fig .5. Illustration of M-NDP

As in D-NDP, all nodes need to periodically initiate the M-NDP process at some random time point of its own choice. The request and response message are same as that of D-NDP except one additional field of digital signature SIG is added to denote personal identity and help the intermediate nodes to pass the message further. For example, from figure 5, A initiates the process and unicast an M-NDP request to each of its logical neighbor list LA here let it send to C

$$A \rightarrow C : \{IDA, LA, nA, v, \mathbf{SIG}K_A^{-1}\}\mathbf{C}AC ,$$

When signature verification succeeds, C compares LC with LA and for each node in LC-LA, C sends unicast message, here to B

$$C \rightarrow B : \{IDA, LA, nA, v, \mathbf{SIG}K_A^{-1}, IDC, LC, \mathbf{SIG}K_C^{-1}\}\mathbf{C}CB$$

Upon receiving this message B verifies the signatures using IDC and IDA as public keys. If verification is successful, B checks whether C is common neighbor of A and itself. If not the it discards the request otherwise gives the following response to C

$$B \rightarrow C : \{IDA, IDC, IDB, LB, nB, v, \mathbf{SIG}K_B^{-1}\}\mathbf{C}BC ,$$

When receiving an M-NDP request, every node does the following:

- verify the ID-based signatures of the sender and all previous nodes;

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

- check each nodes logical neighbor list to see whether there is a legitimate path between the source and itself;
- derive the secret key and session spread code uniquely shared with the source and start sending the HELLO message spread with the derived session code;
- send a modified M-NDP request by adding its own ID and logical neighbor list to the nodes not appearing in the logical neighbor lists of the received request, if the number of hops that the request has traversed is less than ν .

On receiving the M-NDP response, C verifies signature of B and if its correct forwards the response to A

$$\{IDA, IDC, IDB, LB, nB, v, \mathbf{SIG}K_B^{-1}, LC, \mathbf{SIG}K_C^{-1}\}\mathbf{C}CA .$$

Upon receiving the response, node A first verifies signatures using IDC and IDB as the public keys. If both signatures are correct, A further checks whether there is a legitimate path between the destination and itself. If so, A uses its private key and IDB to compute the shared key KAB = KBA whereby to derive the session spread code. It then starts to monitor CAB in real time. If A and B are indeed physical neighbors, then A can receive the HELLO message from B spread with CBA. If so, A accepts B as its authenticated logical neighbor and returns a CONFIRM message spread with CAB. Once receiving the CONFIRM message, node B accepts A as its authenticated logical neighbor. Different from D-NDP, M-NDP may incur false positives, which means that some nodes that are not physical neighbors may falsely discover each other, e.g., A may discover nodes G and H in Fig. 5. To address this limitation, a location-aware multi-hop neighbor discovery protocol (LAMNDP) is proposed.

4) Location Aware Multi-hop Neighbor Discovery Protocol

LAM-NDP is designed to eliminate the false positives incurred by M-NDP by exploring the location information of nodes. For this purpose, LAM-NDP requires every node to be capable of localizing itself via GPS signals or other localization techniques. The request and response messages are same as M-NDP instead of addition of fields of location of that nodes and propagation range. The Illustration is given in Fig 6.

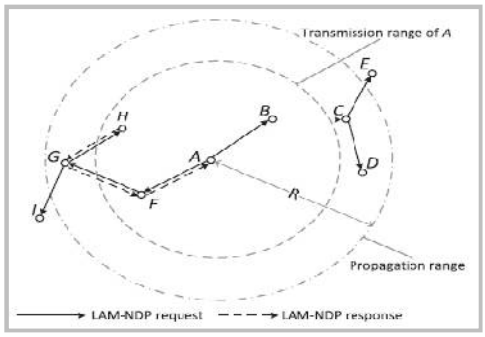

Fig 6. Illustration of LAM-NDP

As in M-NDP, all nodes need to periodically initiate the LAM-NDP process at some random time point of its own choice. For example, from figure 4, A initiates the process and unicasts an LAM-NDP request to each of its logical neighbor list LA here let it send to F

$$A \rightarrow F : \{IDA, LA, nA, lA, \mathrm{R}, \mathbf{SIG}K_A^{-1}\}\mathbf{C}AF ,$$

When F receives this request, it verifies whether it is in propagation range of node A based on lA, R and its own location. If it is then it verifies the signature of A using IDA as public key. If it succeeds then for each node in LF n LA, F sends unicast message, here to G

$$F \rightarrow G : \{IDA, LA, nA, \mathrm{R}, \mathbf{SIG}K_A^{-1}, IDF, LF, \mathbf{SIG}K_F^{-1}\}\mathbf{C}FG$$

Upon receiving this message, G also checks the same things as like F such as propagation range etc. and unicasts to its own logical neighbor set by including its own signature and location variable(from figure say to H). On receiving the LAM-NDP request, H knows it is in transmission range of A and verifies all three signatures of A,F,G. It checks whether there is legitimate path between A and itself. After all verifications H sends the following response

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

$H \rightarrow G$: {*IDA, lA, IDF, IDG, IDH,LH, lh, nH,* R, **SIG**K_H^{-1},*LG,* **SIG**K_G^{-1},*LF,* **SIG**K_F^{-1}, }**C***HG*

When receiving an LAM-NDP request, every node does the following:

- Check if it is within the propagation or transmission range of the LAM-NDP initiator. If it is within neither range, the LAM-NDP request is dropped.
- Verify every ID-based signature in the LAM-NDP request.
- Check each intermediate nodes logical neighbor list to see whether there is a legitimate path between the LAM-NDP initiator and itself.
- If it is within the transmission range of the LAM-NDP initiator, derive the secret key and session spread code uniquely shared with the LAM-NDP initiator and start sending a HELLO message spread with the session code.
- If it is within the propagation range of the LAM-NDP request, unicast a modified LAM-NDP request by adding its own ID and logical neighbor list to the nodes not appearing in the logical neighbor lists of the received LAM-NDP request.

The LAM-NDP response is processed by each intermediate node in a similar way as the M-NDP, i.e., each node verifies the previous signatures and adds its own ID, logical neighbor list and signature. Upon receiving the response, node A first verifies signatures using IDs and keys and after verification accepts H as logical neighbor. Once receiving the CONFIRM message node H accepts A as physical neighbor.

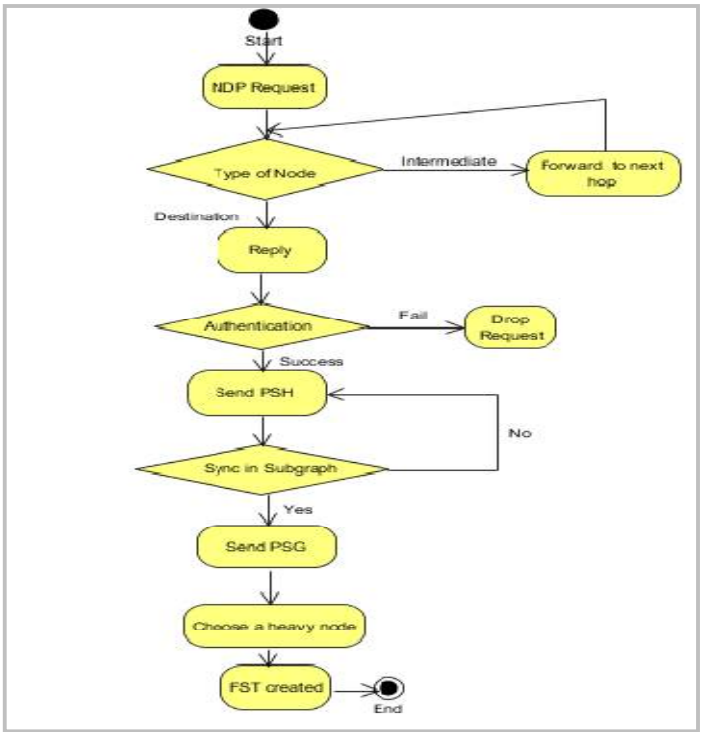

Fig 7. Operational flow of proposed system

V. SIMULATION RESULTS

To evaluate the performance of combined FST and JR-SND mechanism, certain parameters were taken into consideration. The comparison of the system was done with the clock-sampling mutual network synchronization system. CS-MNS is the only decentralized system which is able to converge in different topology. The graphical performance analysis of these parameters is given as follows. The simulations were carried out in Network Simulator (NS3).

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

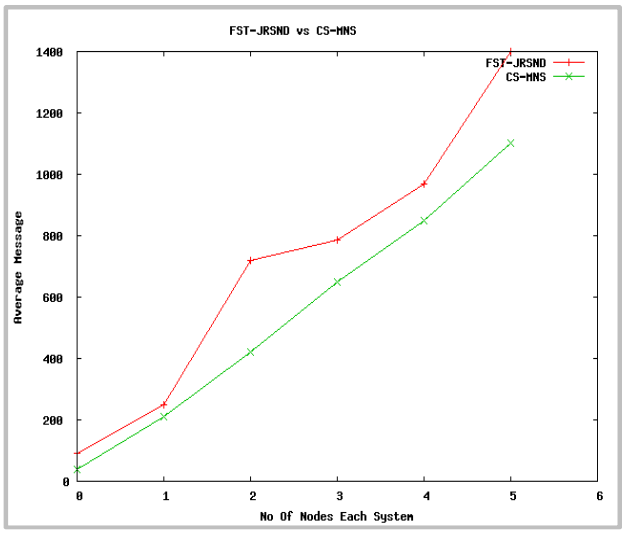

Fig 8. Average number of message exchanges

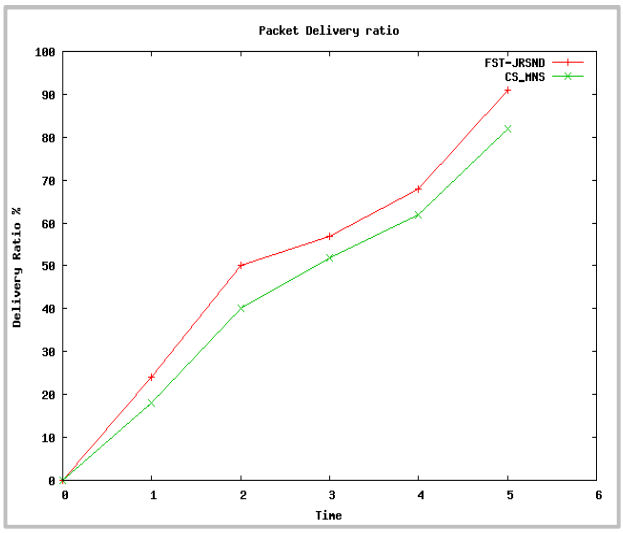

Fig 9. Packet Delivery Ratio

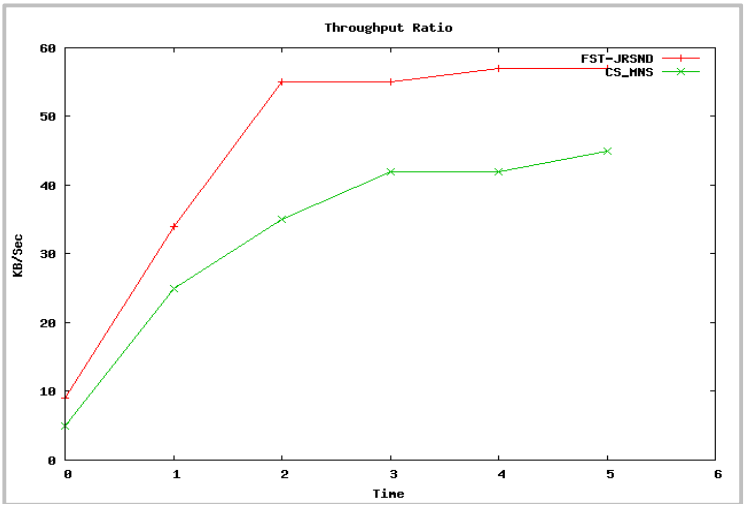

Fig 10. Throughput Ratio

These above graphs give the comparison between the combined system and CS-MNS. These graphs show the performance of system in throughput, packet delivery and message exchanges. The proposed system efficiently outperformed CS-MNS.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

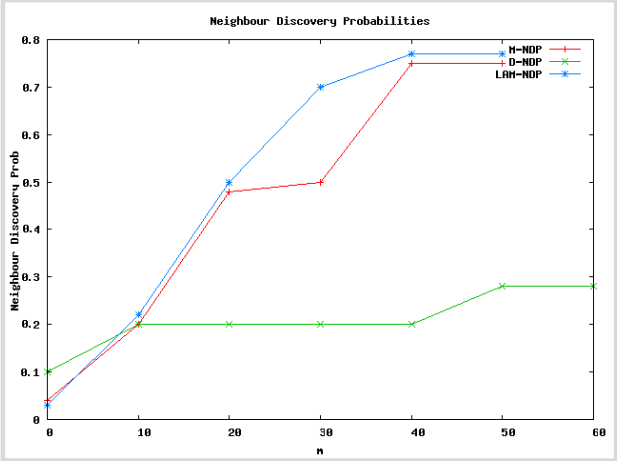

Fig 11. Neighbor Discovery Probability

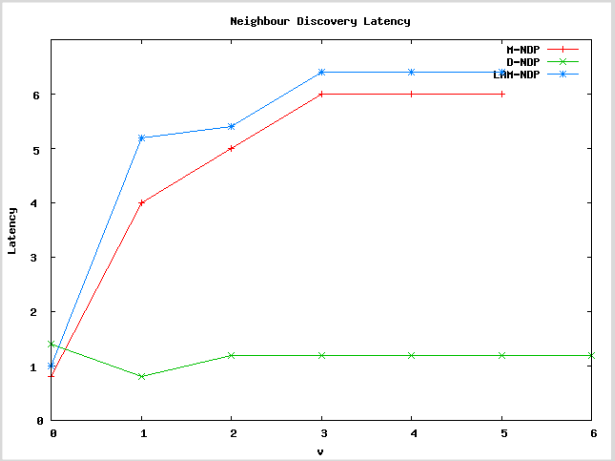

Fig 12. Neighbor Discovery Latency

The above graphs show the working of all three JR-SND protocols. The first graph shows the discovery probabilities of the neighbor despite the presence of jammers. While the second graph shows the latency in neighbor discovery. These metrics described the efficiency of the proposed system.

VI. CONCLUSION

Literature Survey helped in analyzing the existing methods and there drawbacks. The Firefly Spanning Tree mechanism for D2D network ProSe can be able to achieve proximity discovery and synchronization at a time. The mechanism may achieve proximity discovery and synchronization at the same time, in both physical and application level. The problem of different topology may also be solved. Jamming-Resilient Secure Neighbor Discovery mechanism can provide great neighbor discovery probabilities despite the presence of omnipresent jammers. The combined mechanism can be a solution to various problems which are faced by D2D communication, especially in compromising situations like natural calamities and similar problem. Simulation results showed the efficiency of the combined FST and JR-SND mechanism.

REFERENCES

[1] Shih-Lung Chao, Hsin-Ying Lee, Ching-Chun Chou, and Hung-Yu Wei, "Bio-Inspired Proximity Discovery and Synchronization for D2D Communications", IEEE Communication Letters, Accepted for Publication, 1089-7798/13
[2] Rui Zhang, Jingchao Sun, Yanchao Zhang and Xiaoxia Huang, Jamming-Resilient Secure Neighbor Discovery in MANET , IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 14, NO. 10, October 2015.
[3] Klaus Doppler, Mika Rinne, Carl Wijting, Cssio B. Ribeiro, and Klaus Hugl, "Device-to-Device Communication as an Underlay to LTE-Advanced Networks, IEEE Communications Magazine", Dec 2009.
[4] Xingqin Lin, Jeffrey G. Andrews, Amitabha Ghosh, and Rapeepat Ratasuk, "An Overview of 3GPP Device to Device Proximity Services" , IEEE Communications Magazine, April 2014.
[5] Geoffrey WernerAllen, Geetika Tewari, Ankit Patel, Matt Welsh, Radhika Nagpal ," Firefly Inspired Sensor Network Synchronicity with Realistic Radio Effects ", SenSys05, November 24, 2005, San Diego, California, USA. Copyright 2005 ACM 159593054X/ 05/0011.
[6] Renato E. Mirollo; Steven H. Strogatz , "Synchronization of Pulse-Coupled Biological Oscillators", SIAM Journal on Applied Mathematics, Vol. 50, No. 6. (Dec,1990), pp. 1645-1662.
[7] R. G. Gallager, P. A. Humblet, and P. M. Spira," A distributed algorithm for minimum-weight spanning trees", ACM Trans. Program. Lang. Syst., vol. 5, no. 1, pp. 6677, 1983.
[8] C. H. Rentel and T. Kunz, "Clock-sampling mutual network synchronization for mobile multi-hop wireless ad hoc networks", in Proc. 2007 MILCOM
[9] G. Werner-Allen, P. Swieskowski, and M. Welsh. "MoteLab: A wireless sensor network testbed", In Proc. IPSN05, Special Track on Platform Tools and Design Methods (SPOTS), April 2005.
[10] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly inspired sensor network synchronicity with realistic radio effects", in Proceedings of the 3rd international conference on Embedded networked sensor systems, pp. 142 153, ACM, 2005.
[11] A. Tyrrell, G. Auer, and C. Bettstetter, "Emergent slot synchronization in wireless networks", IEEE Transactions on Mobile Computing, vol. 9, no. 5, pp. 719732, 2010.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

[12] A. Tyrrell, G. Auer, and C. Bettstetter, "Fireflies as role models for synchronization in ad hoc networks", in Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems, p. 4, ACM, 2006.
[13] D. Lucarelli, I.-J. Wang, et al., "Decentralized synchronization protocols with nearest neighbor communication", in Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 6268, ACM, 2004.
[14] M. Strasser, C. Popper, S. Capkun, and M. Cagalj, Jamming-resistant key establishment using uncoordinated frequency hopping, in Proc. IEEE SP, Berkeley/Oakland, CA, USA, May 2008
[15] T. Jin, G. Noubir, and B. Thapa, Zero pre-shared secret key establishment in the presence of jammers, in Proc. ACM MobiHoc, Apr. 2009
[16] M. Strasser, C. Popper, and S. Capkun, Efficient uncoordinated FHSS anti-jamming communication, in Proc. ACM MobiHoc, Apr. 2009.
[17] D. Slater, P. Tague, R. Poovendran, and B. J. Matt, A coding-theoretic approach for efficient message verification over insecure channels, in Proc. ACM WiSec, Zurich, Switzerland, Mar. 2009
[18] Q.Wang, P. Xu, K. Ren, and M. Li, Delay-bounded adaptive UFHbased anti-jamming wireless communication, in Proc. IEEE INFOCOM, Shanghai, China, April 2011
[19] Q. Wang, P. Xu, K. Ren, and X.Y. Li, Towards optimal adaptive UFH-based anti-jamming wireless communication, IEEE J. Sel. Areas Commun., vol. 30, no. 1, Jan. 2012.

BIOGRAPHY

Abhijeet M. Dholeshwar received Bachelor's degree in Computer Science and Engineering from Sant Gadge Baba Amravati University and currently pursuing Master of Engineering in Computer Networks from Sinhgad College of Engineering, Pune, MS

Prof. Deepali P. Salapurkar received her Master's degree in Computer Networks from Savitribai Phule Pune University in 2011. She has completed her Bachelor's degree in Computer Science and Engineering from Dr.Babasaheb Ambedkar Marathwada University, Aurangabad in 2000. Currently she is working as Assistant Professor in Department of Computer Engineering in Sinhgad College of Engineering, Pune.