International Journal of Research in Engineering, Science and Management
Volume-1, Issue-12, December-2018
www.ijresm.com | ISSN (Online): 2581-5792

Proximate Detection in Wireless Networks Using NPV Protocol

D. Mohana[1], M. Niranjala[2]
[1,2]*Assistant Professor, Department of ECE, Thangavelu Engineering college, Chennai, India*

***Abstract*: Neighbor discovery is the first step in configuring and managing a wireless network. Many of the existing techniques on neighbor discovery assume a single-packet reception model where only a single packet can be received successfully at a receiver. Motivated by the increasing prevalence of multi packet reception (MPR) technologies such as CDMA and MIMO, we study neighbor discovery in MPR networks that allow packets from multiple simultaneous transmitters to be received successfully at a receiver in this paper. Starting with a clique of n nodes, we first analyze a simple Aloha-like algorithm and show that it takes Qðnln n k Þ time to discover all neighbors with high probability when allowing up to k simultaneous transmissions. We then come up with the design of two adaptive neighbor discovery algorithms that dynamically adjust the transmission probability for each node. We show that the adaptive algorithms exhibit a QðlnnÞ improvement over the Aloha-like scheme for a clique with n nodes and are thus order-optimal. Ultimately, we analyze our algorithm in a multi-hop network setting. We show an upper bound of OðDln n k Þ for the Aloha-like algorithm has a performance which is at most a factor ln n worse than the optimal provided the maximum node degree is D. In addition to this, when D is large, we show that the adaptive algorithms are order optimal which means that have a running time of OðDkÞ which matches the lower bound for the problem.**

***Keywords*: Dynamically Multipacket reception (MPR) and SPR network**

1. Introduction

Neighbor Discovery is one of the first steps in configuring and managing a wireless network. The output result obtained from neighbor discovery, is needed to support basic functionalities such as medium access and routing. This is nothing but the set of nodes that a wireless node can directly communicate with, In addition, this resultant data is needed by topology control and clustering algorithms to improve network performance. Due to its critical importance, neighbor discovery has received significant attention, and a number of studies have been devoted to this topic. Most studies, however, assume a single packet reception (SPR) model, i.e., a transmission is successful if and only if there are no other simultaneous transmissions. In contrast to prior literature, we study neighbor discovery in multipacket reception (MPR)

2. Existing system

- A transmission is successful if and only if there are no other simultaneous transmissions.
- Neighbor discovery in MPR net-works differs fundamentally from that in SPR networks in the following manner.
- In a SPR network, a node is discovered by each of its neighbors if it is the only node that transmits at a given time instant.
- While in an MPR network, a node can transmit simultaneously with several other neighbors, and each of these nodes may be discovered simultaneously by the receiving nodes.
- In a SPR network, a node is discovered by each of its neighbors if it is the only node that transmits at a given time instant; while in an MPR net-work, a node can transmit simultaneously with several other neighbors, and each of these nodes may be dis-covered simultaneously by the receiving nodes.

A. Disadvantages

- Simultaneous transmission is not possible.
- Single packet is send to the destination.

3. Proposed methodology

- The algorithms proposed to use a multiuser-detection based approach for neighbor discovery. They require each node to possess a signature as well as know the signatures of all the other nodes in the network.
- Further, nodes are assumed to operate in a synchronous manner. When a node receives transmission from multiple neighbors, it determines which nodes are the transmitters based on the received signal (or energy) and the prior knowledge of the node signatures in the network.
- Although these studies allow multiple transmitters to transmit simultaneously, their focus is on using coherent/ non-coherent detection or group testing to identify neighbors with a high detection ratio and low false positive ratio, and do not provide analytical insights on the time complexity of their schemes.
- In contrast, our study aims to under-stand the efficiency of different neighbor discovery algorithms by de-riving analytical results on their time

complexity. Further, from a practical viewpoint, our approach does not re-quire node signatures and can operate in asynchronous systems.

- There are numerous studies on neighbor discovery when nodes have directional antennas. The focus in these works is on antenna scanning strategies for efficient neighbor discovery. There have been several recent proposals on neighbor discovery in cognitive radio networks.
- They determine the set of neighbors for a node as well as the channels that can be used to communicate among neighbors. In contrast, we assume Omni-directional antennas (or antenna arrays) and multi-packet reception capabilities at each node.

A. Advantages

- The remaining active nodes to in-crease their transmission probability.
- Increase the data sending speed.
- Simultaneous transmission is possible.

4. Module description

A. Node creation:

- In this module, we create many nodes.
- Users enter the IP Address, port number and Status of the node to register in the Database.
- While entering the next node the user must check the database for that node exists or new one.

B. Discover Neighbor Nodes

- After Node Creation, Source node Discover its Neighbor Nodes. Each and every node has Neighbor Node information.
- Verify Neighbor Node's Position:
- In this module, Source node Verify the Neighbor node's Position. Here we can use three types of Message Exchange Protocols.

1) POLL message

The verifier starts the protocol by broadcasting a POLL whose transmission time tS it stores locally. The POLL is anonymous, since it does not carry the identity of the verifier.

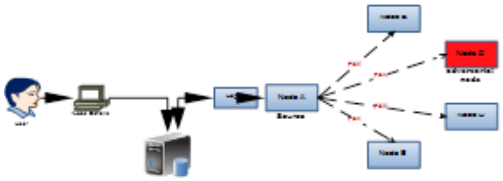

Fig. 1. Poll message

2) REPLY message

Communication neighbor that receives the POLL stores its reception time. After that Neighbor nodes broadcasts an anonymous REPLY message using a fresh MAC address, and locally records its transmission time.

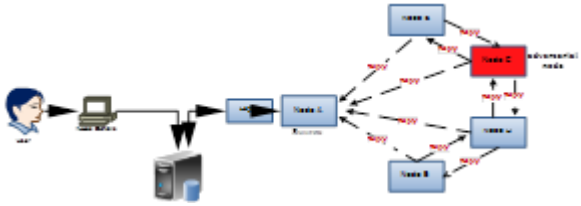

Fig. 2. Reply message

3) REVEAL message

After that, the verifier broadcasts a REVEAL message using its real MAC address. The REVEAL contains a proof that S is the author of the original POLL through the encrypted hash. This is a verifier identity, i.e., its certified public key and signature.

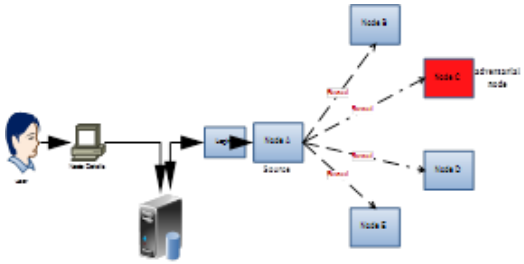

Fig. 3. Reveal message

4) REPORT message

Once the REPORT message is broadcast and the identity of the verifier is known, each neighbor that previously received S's POLL unicasts to S an encrypted, signed REPORT message.

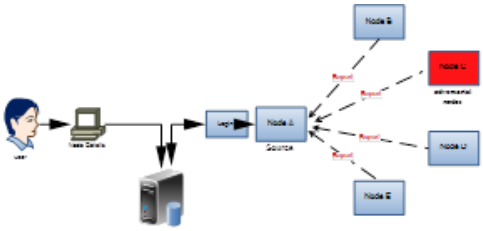

Fig. 4. Report message

C. Find out the adversarial nodes

The Source node collect the all information from the neighbor nodes, and then analyses the Report, after that verifying the Position, then we find out the Adversarial nodes in the Network. Here we find out the colluding at-tacks, collinear attack, and clogging attack.

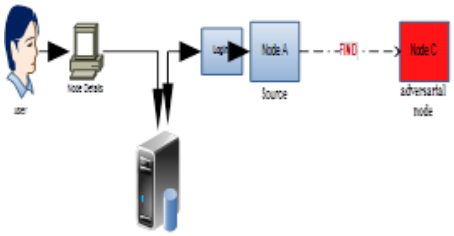

Fig. 5. Adversarial nodes

5. Implementation

We detail the message exchange between the verifier and its communication neighbors, followed by a description of the tests run by the verifier. The message is transmitted by the message exchange and the time taken for transmission and reception represented by various time constants. To retrieve the exact transmission and reception time instants, avoiding the unpredictable latencies introduced by interrupt triggered at the driver's level, a solution such as that implemented in is required. In addition, the GPS receiver should be integrated in the 802.11 card.

A. POLL message

The verifier starts the protocol by broadcasting a POLL whose transmission time tS it stores locally. The POLL is anonymous, since 1) it does not carry the identity of the verifier, 2) it is transmitted employing a fresh, software-generated MAC address, and 3) it contains a public key K0S taken from S's pool of anonymous one-time use keys that do not allow neighbors to map the key onto a specific node. We stress that keeping the identity of the verifier hidden is important in order to make our NPV robust to attacks (see the protocol analysis in Section 6). Since a source address has to be included in the MAC-layer header of the message, a fresh, software-generated MAC address is needed; note that this is considered a part of emerging cooperative systems. Including a one-time key in the POLL also ensures that the message is fresh (i.e., the key acts as a nonce).

B. REPLY message.

A communication neighbor X 2 INS that receives the POLL stores its reception time tSX, and extracts a random wait interval TX 2 ½0; Tmax. After TX has elapsed, X broadcasts an anonymous REPLY message using a fresh MAC address, and locally records its transmission time tX. For implementation feasibility, the physical layer transmission time cannot be stamped on the REPLY, but it is stored by X for later use. The REPLY contains some information encrypted with S s public key (K0S), specifically the POLL reception time and a nonce _X used to tie the REPLY to the next message sent by X: we refer to these data as X's commitment, Cj X. The hash hK0 S, derived from the public key of the verifier, K0S, is also included to bind POLL and REPLY belonging to the same message exchange. Upon reception of a REPLY from a neighbor X, the verifier S stores the reception time tXS and the commitment Cj X. When a different neighbor of S, e.g., Y, Y 2 INS \ INX, broadcasts a REPLY too, X stores the reception time tYX and the commitment Cj Y. Since REPLY messages are anonymous, a node records all commitments it receives without knowing their originators. REVEAL message. After a time Tmax þ _ þTjitter, the verifier broadcasts a REVEAL message using its real MAC address. _ Accounts for the propagation and contention lag of REPLY messages scheduled at time Tmax, and Tjitter is a random time added to thwart jamming efforts on this message.

C. REVEAL Message

1) a map ImS, that associates each commitment Cj X received by the verifier to a temporary identifier iX; 2) a proof that S is the author of the original POLL through the encrypted hash Ek0 S fhK0 S g; 3) the verifier identity, i.e., its certified public key and signature .Note that using certified keys curtails continuous attempts at running the protocol by an adversary who aims at learning neighbor positions (i.e., at becoming knowledgeable) or at launching a clogging attack.

D. REPORT Message

Once the REPORT message is broadcast and the identity of the verifier is known, each neighbor X that previously received S's POLL unicasts to S an encrypted, signed REPORT message. The REPORT carries X's position, the transmission time of X's REPLY, and the list of pairs of reception times and temporary identifiers referring to the REPLY broadcasts X received The identifiers are obtained from the map ImS included in the REVEAL message. Also, X discloses its own identity by including in the message its digital signature and certified public key. We remark that all sensitive data are encrypted using S s public key, KS, so that eavesdropping on the wireless channel is not possible. At the end of the message exchange, only the verifier knows all positions and timing information. If needed, certified keys in REPORT messages allow the matching of such data and node identities (temporary or long-term, with the help of an authority if needed).

6. NPV protocol

Source finds the position of each neighbor using the NPV protocol. In NPV protocol source (verifier) broadcast the POLL message to all neighbors within the proximity region. The verifier also stores the transmission time of the POLL message for all neighbors. After receiving POLL message from verifier, each neighbors stores the reception time of the POLL message and REPLY to verifier. The REPLY message contains the node ID of each neighbor. This also internally saves the transmission time of REPLY message. Then REVEAL message is broadcasted using Verifier's address. It contains a proof that S is the author of the original POLL and the verifier identity. After reveal message broadcasted, each neighbors reported the position to verifiers. The REPORT message includes the neighbor's position and transmission time of REPLY message.

A. Algorithm 1: position of neighbors

If {verifier = nexthop}
if find(ListN, V) then
TransmitPOLLmessage (V);
Store Tx (V);
end if
for i 0 to length(ListN) do
end for

if find (REPLY(ListN)) then
TransmitREVEALmessage (V);
REVEAL: REVEAL (Verifier ID)
end if
for i 0 to length(ListN) do
ListN[i]: REPORT (Px, Tx(REPLY))
end for

7. Conclusion

We designed and analyzed randomized algorithms for neighbor discovery for both clique and general network topologies under various MPR models. For clique topologies, we started with an Aloha-like algorithm that assumes synchronous node transmissions and a priori knowledge of the number of neighbor's n. We showed that the total neighbor discovery time for this algorithm is $\Theta(\ln n)$ under the idealized MPR model, and $\Theta(\frac{n \ln n}{k})$ under the MPR-k model. We further designed adaptive neighbor discovery algorithms for the case when a node knows if its transmission is successful or not, and showed that it provides a factor lnn improvement over the Aloha-like scheme. We extended our schemes to accommodate a number of practical scenarios such as when the number of neighbors is not known beforehand and the nodes are allowed to transmit asynchronously. We analyzed the performance of our algorithms in each case and demonstrated at most a constant or $\Theta(\ln n)$ factor slowdown in algorithm performance. Finally, we consider the general multi-hop network setting and show that the Aloha-like scheme achieves an upper bound of, $O(\frac{\Delta \ln n}{k})$ at most a factor ln n worse than the optimal, and the adaptive algorithm is order-optimal i.e., it achieves an upper bound of $O(\Delta/k)$ when D is large. We have used neighbor discovery time as the performance metric throughout the paper. Another interesting metric is energy consumption during the neighbor discovery process. Energy consumption of the Aloha-like algorithm can be directly derived from neighbor discovery time. Analyzing energy consumption of the adaptive algorithms in more involved and is left as future work. Another interesting direction of future work is extending our study to more generalized MPR models (e.g., accounting for fading, shadowing and other random errors observed in wireless channels).

References

[1] Neighbor Discovery in Wireless Networks with Multipacket Reception IEEE Transactions On Parallel AND distributed systems, vol. 26, no. 7, July 2015.
[2] N. Alon and J. Spencer, "The Probabilistic Method," Hoboken, NJ, USA: Wiley, 2008.
[3] D. Angelosante, E. Biglieri, and M. Lops, "Neighbor discovery in wireless networks: A multiuser-detection approach," in Proc. Inform. Theory Appl. Workshop, Feb. 2007, pp. 46–53.
[4] D. Angelosante, E. Biglieri, and M. Lops, "A simple algorithm for neighbor discovery in wireless networks," in Proc. IEEE Int. Conf. Acoustics, Speech Signal Process., Apr. 2007, pp. 169–172.
[5] C. L. Arachchige, S. Venkatesan, and N. Mittal, "An asynchronous neighbor discovery algorithm for cognitive radio networks," in Proc. IEEE Symp. New Frontiers Dyn. Spectr. Access Netw., 2008, pp. 1–5.
[6] S. A. Borbash, A. Ephremides, and M. J. McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," Ad Hoc Netw., vol. 5, no. 7, pp. 998–1016, 2007.
[7] R. Cohen and B. Kapchits, "Continuous neighbor discovery in asynchronous sensor networks," IEEE/ACM Trans. Netw., vol. 19, no. 1, pp. 69–79, Feb. 2011.