RESEARCH ARTICLE OPEN ACCESS

Comparibility And Implementing Analysis Of Neighbourhood Discovery In Wireless Network

P.Poornima[1], K.Kumar[2]
[1]III Year MCA, Veltech Dr.RR and Dr.SR Technical University, Chennai, India
[2]Asst.Professor, Department of MCA, Veltech Dr.RR and Dr.SR Technical University,Chennai, India

Abstract :
Neighbor discovery is an important first step in the initialization of a wireless ad hoc network. In this paper, we design and analyze several algorithms for neighbor discovery in wireless networks. Starting with a single-hop wireless network of nodes, we propose a ALOHA-like neighbor discovery algorithm when nodes cannot detect collisions, and an order-optimal receiver feedback-based algorithm when nodes can detect collisions. Our algorithms neither require nodes to have a prior estimate of the number of neighbors nor synchronization between nodes. Our algorithms allow nodes to begin execution at different time instants and to terminate neighbor discovery upon discovering all their neighbors. We finally show that receiver feedback can be used to achieve a running time, even when nodes cannot detect collisions. We then analyze neighbor discovery in a general multi hop setting. We establish an upper bound of on the running time of the ALOHA-like algorithm, where denotes the maximum node degree in the network and the total number of nodes. We also establish a lower bound of on the running time of any randomized neighbor discovery algorithm. Our result thus implies that the ALOHA-like algorithm is at most a factor worse than optimal.
Keywords : Neighbour discovery, multihop, feedback, collision detection, wireless network, beacon based neighbourhood discovery.

I. INTRODUCTION

WIRELESS ad hoc networks and sensor networks are typically deployed without any communication infrastructure and are required to “configure” themselves upon deployment. For instance, immediately upon deployment, a node has no knowledge of other nodes in its transmission range and needsto discover its neighbors in order to communicatewith other network nodes. Neighbor discovery is an indispensable first stepin the initialization of a wireless network since knowledge ofone-hop neighbors is essential for medium access control protocols, routing protocols and topology control algorithms to work efficiently and correctly. Neighbor discovery algorithms can be classified into twocategories, viz. *randomized* or *deterministic*. In randomized neighbor discovery, each node transmits at randomly chosen times and discovers all its neighbors by a given time with high probability (w.h.p.). In deterministic neighbor discovery, on the other hand, each node transmits according to a predetermined transmission schedule that allows it to discover all its neighbors by a given time with probability one. In distributed settings, determinism often comes at the expense of increased running time and and, in the particular case of neighbor discovery, typically requires unrealistic assumptions such as node synchronization and *a priori* knowledge of the number of neighbors. We, therefore, investigate randomized neighbor discovery algorithms in this paper. Neighbor discovery is nontrivial for several reasons. a) Neighbor discovery needs to cope with collisions. Ideally a neighbor discovery algorithm needs to minimize the probability of collisions and, therefore, the time to discover neighbors.b) In many practical settings, nodes have no knowledge of the number of neighbors, which makes coping with collisions even harder. c) When nodes do not have access to a global clock, they need to operate asynchronously and still be able to discover their neighbors efficiently. d) In asynchronous systems, nodes can potentially start neighbor discovery at different times and, consequently, may miss each other’s transmissions.5) Furthermore, when the number of neighbors is unknown, nodes do not know when or how to terminate the neighbor discovery process. In this paper, we present neighbor discovery algorithms that comprehensively address each of these practical challenges under the standard collision channel model. Unlike existing approaches that assume *a priori* knowledge of the number of neighbors or clock synchronization among nodes, we propose neighbor discovery algorithms that:P1 do not require nodes to have *a priori* knowledge of the number of neighbors;P2 do not require synchronization among nodes;P3 allow nodes to begin execution at different time instants;P4 enable each node to detect when to terminate the neighbor discovery process. To the best of our knowledge, our work provides the first solution to the neighbor discovery problem that satisfies all of the properties P1–P4. Our approach is to start with a single-hop wireless network in which nodes are synchronized and know exactly how many neighbors they have. As we will see, the analysis in such a simplistic setting yields several valuable insights about the neighbor discovery problem. These insights allow us to progressively relax each of the assumptions leading to a complete and practical solution to the neighbor discovery problem in a multihop network setting[2].

II. MOTIVATION

Fundamental problem in large, self-organizing wireless networks. Since knowledge of one-hop neighbors is essential for medium access control protocols , routing protocols , and topology control algorithms to work efficiently and sometimes, correctly. Medium access, routing, topology control depend on knowledge of neighbor IDs. Faster neighbor discovery implies reduced energy consumption.

III. PROBLEM DEFINITION

The study of feedback algorithm usage along with multi hop setting in neighborhood discovery in wireless network and comparative study of the system by checking the order optimality in this environment. The lower bound deterministic complexity is verified with exponential gap between randomization and determinism. This model also lays foundation on mobility and topology maintenance and further enhancement by means of implementing analysis.

IV. Literature Survey

We analyzed beacon based algorithm concept following book Bluetooth Specification Version 3.0 + HS as well as we referred internet. Then we went through the content regarding the feedback based algorithm following books D. Angelosante, E. Biglieri, and M. Lops. Neighbor discovery in wireless networks : a multiuser-detection approach. In *Information Theory and Applications Workshop*, R. Khalili, D. Goeckel, D. Towsley, and A. Swami. Neighbor discovery with reception status feedback to transmitters. In *IEEE INFOCOM I collision detection ND algorithm about base paper for efficient algorithm for neighborhood discovery in wireless network was surveyed. This led us to work on the related algorithms which helped us to move towards the required result.*

ALGORITHM:

Comparison with beacon-based ND

A). Beacon-based Neighbor Discovery

In beacon-based neighbor discovery, each node transmits *BEACON* messages at fixed intervals i.e., the interval size is independent of n. To avoid synchronization between beacon transmissions, a random delay is added to the intervals. This scheme has been proposed in numerous contexts, e.g., (i) to maintain neighbor list in routing protocols such as AODV , DSR , and GPSR , and (ii) for topology formation in The context of Bluetooth and IEEE 802.15.4 .

Beacon-based neighbor discovery can be thought of as a randomized algorithm in which each node transmits with probability 1/k at each slot, where k is fixed. For simplicity, consider a clique of n nodes. The probability of a successful transmission is then given byp=1k_1 −1k_n−1≈e−n/kkSimilar to Section III, it can be shown that the expected time to discover all n nodes equals ken/k(lnn + _(1)).To compare the performance of beacon-based neighbor discovery with the algorithms studied in this paper, we consider a Bluetooth network, where each slot is of duration 0.625 MS and beacons are transmitted once every k = 14 slots . In a dense setting, where n ∼100, it can be verified that beacon-based neighbor discovery is 65 times slower than the ALOHA-like algorithm and 300 times slower than the collision detection based algorithm. In addition to faster neighbor discovery, our algorithms also yield significant energy savings.The above example illustrates the poor performance of beacon based neighbor discovery when nodes transmit too frequently. At the other extreme, nodes may transmit very infrequently. For example, the recommended beacon interval in GPSR is 1s. Assuming slots of size 0.625 MS, this corresponds to each node transmitting with probability1/1600. When n = 10, it can be verified that beacon-based neighbor discovery is 59 times slower than the ALOHA-like algorithm and 135 times slower than the collision detection based neighbor discovery algorithms. In addition to the time and energy savings, it must be noted that there is no obvious method for terminating the beacon based neighbor discovery algorithm without an *a priori* estimate of network density, which may not always be available.

Algorithm B Collision Detection-Based ND (i,n) b←0 //Number of neighbors discovered by node i

flag←0 //Has node ibeen discovered by other nodes?

NbrList←[ ] //List of neighbors of node i

loop

pxmit←1/(n −b)

if((flag = 0) **and** (Bernoulli(pxmit) = 1)) **then**

Transmit DISCOV ERY (i) in first sub-slot

ifenergy detected in second sub-slot **then**

flag←1 //"Drop out"

end if

else

ifsuccessful reception in first sub-slot **then**

Transmit bit "1" in second sub-slot

NbrList[b++] ←DISCOV ERY.source

end if

end if

end loop

ALGORITHM C:

Feedback-based Algorithms for Multi-Hop Networks

It is interesting to ask whether or not the feedback-based algorithms studied in Section IV can be extended to the multihop network setting. There are two important obstacles that need to be overcome in this regard.

(1) In a clique setting, when a node i, hears its ID back, it knows that all other nodes in the clique have discovered i, thus allowing it to drop out. In the multi-hop case, however, the presence of hidden terminals may cause a subset of i's neighbors to not receive i's transmission. Thus, icannot drop out despite hearing its ID back.

(2) In the multi-hop setting, i's dropping out needs to be signaled to its neighbors allowing them to increase their transmission probabilities, which appears non-trivial. Exploiting receiver feedback in the multi-hop setting in a manner that yields improvement over the ALOHA-like discovery algorithm is therefore an interesting open problem.

Table 1.0 Comparative study of three algorithm

BEACON BASED ALGORITHM	FEEDBACK BASED ALGORITHM	COLLISION DETECTION NEIGHBORHOOD DISCOVERY ALGORITHM
Only used in Bluetooth network.	It can able to use all wireless network.	It can able to use all wireless network.
Its take so much time to send the file as well as receiving files.	It will take shortest time only why because every time I am getting feed back to my client.so server known very well about shortest path.	Detect the collision if it is multiple client come to the same time as well as same node .
N number of collision	Limited collision	No collision

V. SYSTEM ARCHITECTURE:

FIGURE 1.0 : Client –Server Architecture depicting Neighbor hood detection

Client given request to server. That request will first move to nearest router[5][6]. Then it will check the shortest route using beacon based algorithm, feedback based algorithm and collision detection neighborhood discovery algorithm. After using this algorithm I will find the shortest route. Then client request will send to the server. Then server transfer the file to the client that file first move to nearest router after that will check the shortest route after finding the shortest route that files are transfer via the shortest route then file are transferred from destination place.

VI. IMPLEMENTATION

We developed the beacon based ND algorithm, collision based algorithm and feedback based algorithm with .NET as programming language using c#.

<u>6.1 SCREEN SHOTS</u>

FIGURE 2.0: Client side implementation

Client will mention in advance , if in case server transfers the file . This implies that files directly are taken from desktop or some other place.The receiver path is taken by clicking the click path in the figure.

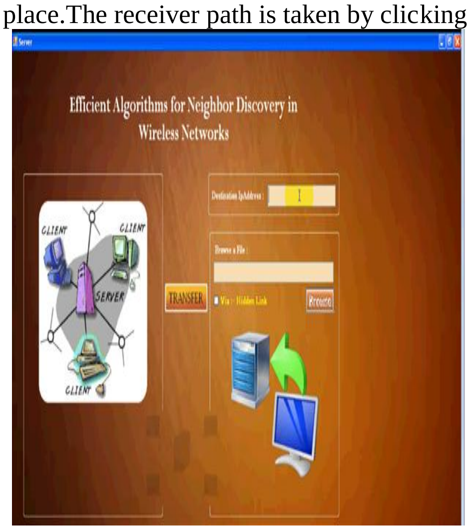

FIGURE 3.0 : Server Side Reply by sending data via hidden link

The server will transfer the file after getting client request. in the screen shot we mention destination ip address(client ip address). After giving ip address server will browse the particular file and will send it to destination place via the hidden link.

6.2 CLIENT CODING

```
namespace DestCode
{
   public partial class Client : Form
   {

      public Client()
      {
         InitializeComponent();
         DestCode.receivedPath = "";
      }

      private void button1_Click(object sender, EventArgs e)
      {

      }

      private void timer1_Tick(object sender, EventArgs e)
      {
         label5.Text = DestCode.receivedPath;
         lblres.Text = DestCode.curMsg;
      }
      public static string receivedPath;
      public static string curMsg = "Stopped";
      public void StartServer()
      {

            if (receivedPath == "")
            {
               MessageBox.Show("No Path was selected to Save the File");
            }
            curMsg = "Saving file...";

            bWrite.Close();
            clientSock.Close();
            curMsg = "File Received ...";

            StartServer();
         }
         catch (Exception ex)
         {
            curMsg = "File Receving error.";
         }
      }
   }
}
```

6.3 SERVER CODING

```
namespace ServerMain
{
   public partial class Form1 : Form
   {
     IPAddress[] ipAddress, ipAddress1;
      IPEndPoint ipEnd, ipEnd1;
      Socket clientSock, clientSock1;
      private const int port = 11000;
```

```
    private static ManualResetEvent connectDone =
      new ManualResetEvent(false);
    private static ManualResetEvent sendDone =
      new ManualResetEvent(false);
    private static ManualResetEvent receiveDone =
      new ManualResetEvent(false);


    public void send()
    {
ipAddress = Dns.GetHostAddresses(txtIp.Text);
      ipEnd = new IPEndPoint(ipAddress[0], 5655);
      clientSock = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);

        lblError.Text = "";
        lblError.Text = "Disconnecting...";
        clientSock.Close();
        lblError.Text = "";
        lblError.Text = "FILE TRANSFERRED.";

        lblError.Text = "";
        lblError.Text = "File sending...";
        System.Threading.Thread.Sleep(1000);

        clientSock1.Send(clientData1);
        lblError.Text = "";
        lblError.Text = "Disconnecting...";
        clientSock1.Close();

        lblError.Text = "";
        System.Threading.Thread.Sleep(4000);
        lblError.Text = "FILE TRANSFERRED.";
       }
```

VII. CONCLUSION

This research paper analyzed and compared three algorithm (a) beacon based algorithm,(b)feedback based algorithm(c)collision detection neighborhood discovery algorithm[8] . Beacon based algorithm(its used in Bluetooth network only)[7]. Feedback back based algorithm results showed that feed back to customer is functioning with service time. It means if we select node it will analyze and depict how much time it will be take to send the file to destination place which is called its called service time. Using collision detection ND algorithm we avoid the collision.

REFERENCES

[1] IEEE 802.15.4 Standard Specification. http://standards.ieee.org/getieee802/802.15.html.
[2] D. Angelosante, E. Biglieri, and M. Lops. Neighbor discovery in wireless networks:a multiuser-detection approach. In *Information Theory and Applications Workshop*, pages 46–53, 2007.
[3] An Introduction to the Ecology of Written Language 2nd ed (review) 2011
[4] Bluetooth Specification Version 3.0 + HS.2008
[5] S. A. Borbash, A. Ephremides, and M. J. McGlynn. An asynchronous neighbor discovery algorithm for wireless sensor networks. *Ad Hoc Networks*, 5(7):998–1016, 2007.
[6] G. Jakllari, W. Luo, and S. V. Krishnamurthy. An integrated neighbor discovery and mac protocol for ad hoc networks using directional Antennas. *IEEE Transactions on Wireless Communications*, 6(3):1114– 1024, 2007.
[7] A. Keshavarzian and E. Uysal-Biyikoglu. Energy-efficient link assessment in wireless sensor networks. In *IEEE INFOCOM*, 2007.
[8] R. Khalili, D. Goeckel, D. Towsley, and A. Swami. Neighbor discovery with reception status feedback to transmitters. In *IEEE INFOCOM*, pages 2375–2383, 2010.
[9] J. Luo and D. Guo. Neighbor discovery in wireless ad hoc networks based on group testing. In *Annual Allerton Conference*, 2008.
[10] Maxim DS32khz Data Sheet 2008
[11] K. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In *IASTED Distributed Sensor Networks*, pages 391–398, 2008.