International Conference on Emanations in Modern Engineering Science & Management (ICEMESM-2018)

RESEARCH ARTICLE OPEN ACCESS

Adhoc on Demand Distance Vector Approach for Routing Optimization in Communication Networks

Shrikant U.Uge[1],Ms. Mona Mulchandani[2]

[1](M.Tech.(CSE),Jhulelal Institute of TechnologyNagpur, India),

[2] (HOD,Department of Computer Science & Engineering,Jhulelal Institute of Technology,Nagpur,India).

Abstract:

Recent Development in wireless communication led to the development of low cost, low power wireless network. Which is widely used these days .minimizing energy consumption and maximizing the lifetime of the network are key factor in the wireless networks. Several routing protocols with different objectives have already been proposed for routing optimization. Several optimization techniques are present which are used to improve the performance of the network. Here in this we briefly discussed AODV approach for routing optimization in communication networks. In this paper source node transmit data packets to the destination node. In which we focus on monitoring network performance, Packet tracker, route cost, and packet loss detection.

Keywords- Routing Optimization, AODV, Distance vector Routing Protocol.

Introduction

The nodes in the network acts as routers, which transmits data packets from one neighbouring node to another.We focus on monitoring network performance, packet tracker, route cost, and packet loss detection. In this project we are designing the network model which is used to find the shortest path among the nodes and maintain the routing table. The source node sends the packets to destination node by finding the shortest path among them.

During packets transmission, every intermediate node in the discovery route create routing table to store the information regarding neighbor node and the destination node information. The routing table information updated for every packet transmission during the message transmission.

Neighbournode discovery problem is nothing but detecting the mobile nodes within one node's communication range. The location information of nodes over time has to be updated accordingly. Also multiple numbers of nodes should not be allowed to access the same destination at the same time there by avoiding packet loss and errors.

The routing protocols for ad hoc wireless network should be capable to handle a very large number of hosts with limited resources, such as bandwidth and energy. The main challenge for the routing protocol is that they must also deal with host mobility, meaning that hosts can appear and disappear in various locations. Thus, all hosts of the ad hoc network act as routers and must participate in the route discovery and maintenance the routers to the other hosts.

Open Shortest Path First is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm which is currently in development process. It gathers link state information from available routers and constructs a topology map of the network. OSPF detects changes in the topology, such as link failures. It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm. The OSPF routing policies for constructing a route table are governed by link cost factors (external metrics) associated with each routing interface. Cost factors may be the distance of a router (round-trip time), data throughput of a link, or link availability and reliability, expressed as simple unless numbers. This provides a dynamic process of traffic load balancing between routers of equal cost. As a link state routing protocol, OSPF establishes and maintains neighbor relationships for exchanging routing updates with other routers. The neighbor relationship table is called an adjacency database. An OSPF router forms neighbor relationship only with the routers directly connected to it.

1. Dijkstra's algorithm solves the single-source shortest path problem.

2. Bellman-ford algorithm solves the single-source problem if edge weights may be negative
3. Floyd-warshall algorithm solves all pairs shortest paths.

To monitor the network performance which include packet capture, packet loss, bandwidth detection, and analysis of various protocol which are used in the network. The nodes are connected with each other via link and they transfer the packets from node to node using the different bandwidth of the link.

Objectives:

- To construct network topology with nodes and link cost.
- To find the shortest path between network topology using the OSPF algorithm.
- To trace the packet loss detection, network bandwidth and network sniffer.
- Reactive protocols – AODV

Reactive protocols seek to set up routes on-demand. If a node wants to initiate communication with a node to which it has no route, the routing protocol will try to establish such a route.The philosophy in AODV, like all reactive protocols, is that topology information is only transmitted by nodes on-demand. When a node wishes to transmit traffic to a host to which it has no route, it will generate a *route request* (RREQ) message that will be flooded in a limited way to other nodes. This causes control traffic overhead to be dynamic and it will result in an initial delay when initiating such communication. A route is considered found when the RREQ message reaches either the destination itself, or an intermediate node with a valid route entry for the destination. For as long as a route exists between two endpoints, AODV remains passive. When the route becomes invalid or lost, AODV will again issue a request.

- Network Routing

Network routing is the process of selecting paths in a network along which to send network traffic. Communication networks can be classified as either circuit-switched or packet-switched. The example of circuit switched network is the telephone network in which the physical circuit is set up at the communication start and remains the same for the communication duration. Unlike them, in packet-switched networks, also called data networks, each data packet can follow a different route and no fixed physical circuits are established. The example of data networks are LAN and the Internet.

- Shortest Path Routing

Shortest path routing is implementing shortest path algorithm on solving the network routing problem. It's objective is to determine the shortest path (minimum cost) between two nodes, where the sum of the costs of its constituent edges is minimized.Till today, many routing algorithms used for solving shortest path have been accepted. One of them is Dijkstra's algorithm. Dijkstra's algorithm, conceived by Dutch computer scientist Edgar Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge costs, producing a shortest path tree.For a given source node in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that node and every other node. It can also be used for finding costs of shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been reached. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path first is widely used in network routing protocols, most notably OSPF (Open Short Path First) and Intermediate system to Intermediate system (IS-IS).

ROUTING OPTIMIZATION

Open Shortest Path First (OSPF):

It is a routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS).

Overview:

OSPF is an interior gateway protocol (IGP) for routing Internet Protocol (IP) packets solely within a single routing domain, such as an autonomous system. It gathers link state information from available routers and constructs a topology map of the network. The topology is presented as a routing table to the Internet Layer which routes datagrams based solely on the destination IP address found in IP packets. OSPF supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) networks and features variable-length subnet masking (VLSM) and Classless Inter-Domain Routing (CIDR) addressing models.

OSPF detects changes in the topology, such as link failures, and converges on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using a method based on Dijkstra's algorithm, a shortest path first algorithm.

The OSPF routing policies for constructing a route table are governed by link cost factors (external metrics) associated with each routing interface. Cost factors may be the distance of a router (round-trip time), data throughput of a link, or link availability and reliability, expressed as simple unitless numbers. This provides a dynamic process of traffic load balancing between routes of equal cost.

DrawPath Method:

```
private void DrawPath()
    {

        PictureBox pathSrcPic;
        PictureBox pathDestPic;
        this.Refresh();
        foreach (Point p in pathDict.Values)
        {
            pathSrcPic = picDict[p.X];
            pathDestPic = picDict[p.Y];

            System.Drawing.Pen pen = newPen
(System.Drawing.Color.Black);
            pen.Width = 2;
            System.Drawing.Graphics path =
this.CreateGraphics();

            path.DrawLine(pen, pathSrcPic.Location.X +
pathSrcPic.Width / 2, pathSrcPic.Location.Y +
pathSrcPic.Height / 2,
                pathDestPic.Location.X + pathDestPic.Width /
2, pathDestPic.Location.Y + pathDestPic.Height / 2);

        }
    }
```

Calculate Matrix Method:

```
private void CalculateMatrix()
{
// Randomize Valid Packet
D = new double[picDict.Count, picDict.Count];
P = new int[picDict.Count, picDict.Count];

int JJ = 0;
int KK = 0;
int MM = 0;

for (JJ = 0; JJ <= picDict.Count - 1; JJ++)
{
for (KK = 0; KK <= picDict.Count - 1; KK++)
{
D[JJ, KK] = 99999;
for (MM = 0; MM <= picDict.Count + 1; MM++)
{
if (JJ == KK)
{
D[JJ, KK] = 0;
break;
}

if (IsPathExist(JJ, KK) &&
!ObtacleDict.Keys.Contains(JJ) &&
!ObtacleDict.Keys.Contains(KK))
{
PictureBox p1 = picDict[JJ];
PictureBox p2 = picDict[KK];
D[JJ, KK] = Math.Sqrt((p2.Location.Y -
p1.Location.Y) * (p2.Location.Y - p1.Location.Y) +
(p2.Location.X - p1.Location.X) * (p2.Location.X -
p1.Location.X)) / 2;
break;
}
}
}
}

for (KK = 0; KK <= picDict.Count - 1; KK++)
{
for (int ii = 0; ii <= picDict.Count - 1; ii++)
{
for (JJ = 0; JJ <= picDict.Count - 1; JJ++)
{
if (D[ii, KK] + D[KK, JJ] < D[ii, JJ])
{
D[ii, JJ] = D[ii, KK] + D[KK, JJ];
P[ii, JJ] = KK + 1;
}
}
}
}
}
```

FindPath Method

```
public void FindPath(int ai, int aj)
    {

        string Str = string.Empty;

        if (P[ai, aj] == 0)
        {
            return;
        }
        else
        {
            FindPath(ai, P[ai, aj] - 1);
            Str1 = Str1 + cmbSrcNode.Items[(P[ai, aj] -
1)].ToString() + "-";
            FindPath(P[ai, aj] - 1, aj);
        }
    }
```

ROUTING

The design of routing protocols for MANET often has a traditional routing concept as an underlying algorithm. Therefore, to understand the routing principles in MANET, it is necessary to review conventional routing algorithms such as distance vector, link state, flooding and source routing.

Distance Vector:

The distance vector routing protocol requires that each router simply inform its neighbors of its routing table. For each network path, the receiving routers pick the neighbor advertising the lowest cost, then add this entry into its routing table for readvertisement. Routing Information Protocol (RIP) is a common distance vector routing protocol. Common enhancements to distance vector algorithms include split horizon, poison reverse, triggered updates, and holddown [9]. The RIP specifications in RFC 1058 [10] provide a good discussion of distance vector, or Bellman-Ford, algorithms.

Link State Routing:

Link State Routing Link state routing requires that each router maintains at least a partial mapping of the network topology. When a network link changes state, a notification, called a link state advertisement (LSA) is flooded through the entire network. All the routers in the network that are notified of the change will re-compute their routes accordingly. The link state technique is generally more reliable, easier to debug, and less bandwidth-intensive than the distance vector. But it is also more complex and more computationally and memory-intensive. OSPF and Intermediate System-to-Intermediate System (IS-IS) are link state routing protocols.

Flooding :

Flooding is a packet delivery technique to perform broadcast. In simple flooding, any node can initiate broadcasting a packet to all neighbors in a network. Upon reception of the packet, all of the neighbor nodes rebroadcast the packet. Each node re-transmits the packet exactly once. This process continues until all of reachable network nodes receive the packet. Each node will rebroadcast the packet upon its reception. The nodes of the network will attempt to distribute the packet to as many nodes as possible. So, whenever a node receives the first copy of a message, it just re-transmits it, and the process goes on until the message reaches the destination. While the flooding technique consumes high network resources, it often leads to higher packet delivery ratio.
The other variants in broadcast protocol use what is commonly known as scoped flooding In scoped flooding, packets are re-transmitted by each node less than once. The main objective is to reduce the number of rebroadcasts to avoid problems associated with broadcast storm. Scoped flooding provides better results in topologies where node mobility is low, where the simple flooding would more likely result in unnecessary re-broadcasts. There are various broadcast which can generally be classified into the following four groups:

- Probability Based Methods
- Area Based Methods
- Neighbor Knowledge Methods
- Cluster Based Method

Source Routing :

Source routing is a technique whereby the sender of a packet can specify the route that a packet should take through the network. There are two types of source routing, namely, strict and loose. In strict source routing, the sender specifies the exact route the packet must take. This technique is seldom used as it is relatively restrictive in use for networking purposes. The more common approach is loose source routing in which the sender only determines a few hops that a packet must take to reach its destination. Source routing requires that every node knows the whole network topology. This is usually solved by maintaining a routing table of the given network.

AD-HOC ROUTING PROTOCOLS

In order for mobile nodes to communicate, they need to agree on what basis the communication will be performed. This is accomplished by the means of a routing protocol. Ad-hoc routing protocols typically have two routing strategies; reactive approach (source initiated, on-demand driven) and proactive approach (table driven). There also exists hybrid routing protocols that integrate both routing strategies. One typical problem with reactive and proactive routing protocols is they have scalability problems with increasing network sizes in the order of a few hundred nodes. In addition, they may not adapt to network conditions as well as a hybrid.
On-demand or reactive protocols only create routes when the source node requests it. When a node requests a route toward another node, a route discovery process is initiated within the network. The route discovery will end once a route is found or once all possible routes are examined. The discovered route will then be maintained until it is no longer valid or not desired.
Proactive protocols attempt to maintain routes from each node to all other nodes in the network. Proactive protocols are also called table driven protocols because they maintain tables for storing routing information. Whenever a change in network topology occurs these changes are propagated through the network by means of broadcast or flooding. These updates are vital in order to maintain a consistent view of the network topology.
A hybrid routing protocol combines the advantages of proactive and reactive routing. It takes advantage of pro-active discovery within the local neighborhood of a node using a reactive protocol for communication between these neighborhoods.

.NET Framework Platform Architecture :

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll. An assembly contains a manifest that provides information about the assembly's types, version, culture, and security requirements.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. Code that is executed by the CLR is sometimes referred to as "managed code," in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.

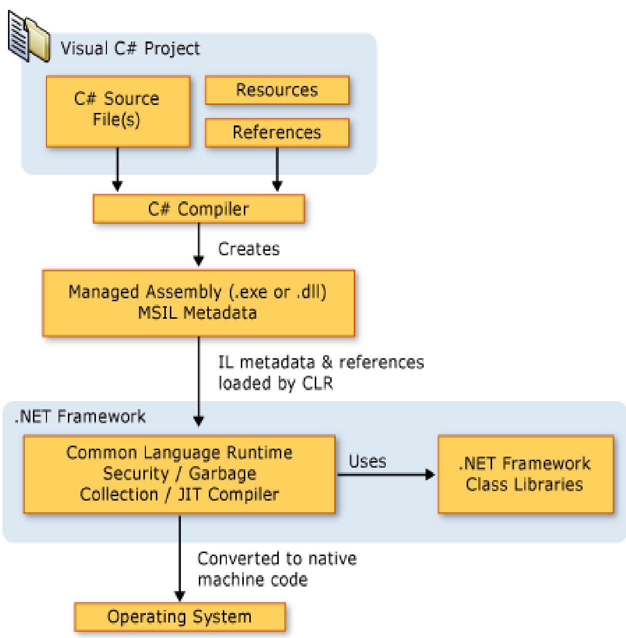

Figure 1 : .Net Framework Platform Architecture

Language interoperability is a key feature of the .NET Framework. Because the IL code produced by the C# compiler conforms to the Common Type Specification (CTS), IL code generated from C# can interact with code that was generated from the .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single assembly may contain multiple modules written in different .NET languages, and the types can reference each other just as if they were written in the same language.

In addition to the run time services, the .NET Framework also includes an extensive library of over 4000 classes organized into namespaces that provide a wide variety of useful functionality for everything from file input and output to string manipulation to XML parsing, to Windows Forms controls. The typical C# application uses the .NET Framework class library extensively to handle common "plumbing" chores.

EXPERIMENTAL RESULT

We have implemented this project on “Adhoc on demand distance vector approach for routing optimization in communication networks” and the output of this project is showing in the figure below. In the figure We have created a node as we required .in this it provides shortest path from source node to destination nodes .this project is implemented for the purpose of transmitting packets considering the better network performance , Packet tracker ,route cost, and packet loss detection.

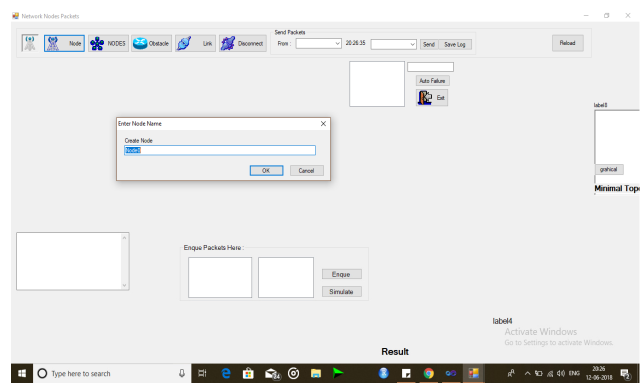

Figure 2:Creation of nodes

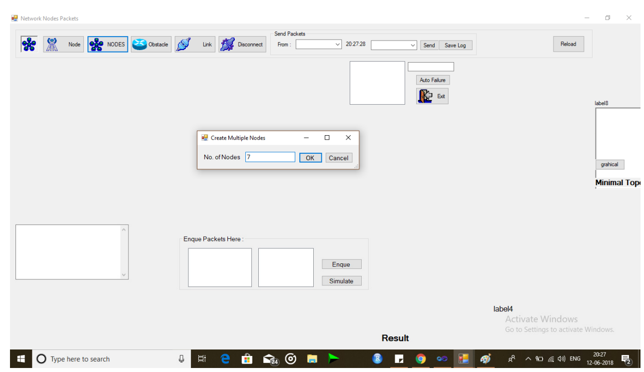

Figure 3: Creation of multiple nodes

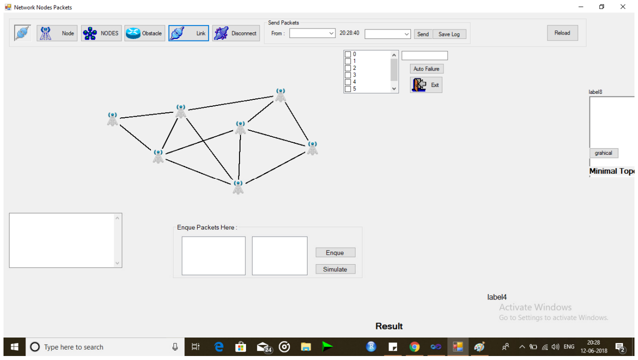

Figure 4: Creation of network topology

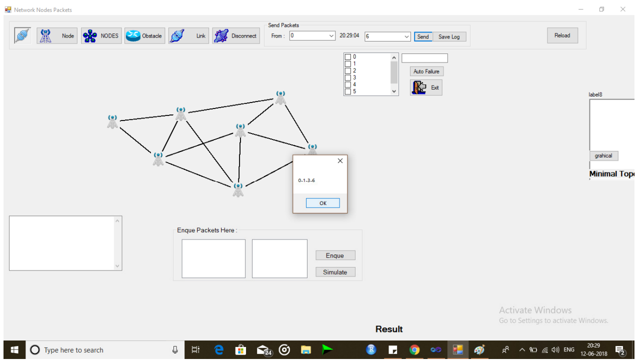

Figure 5: Shortest path between node 0 to node 6 and the path is 0-1-3-6

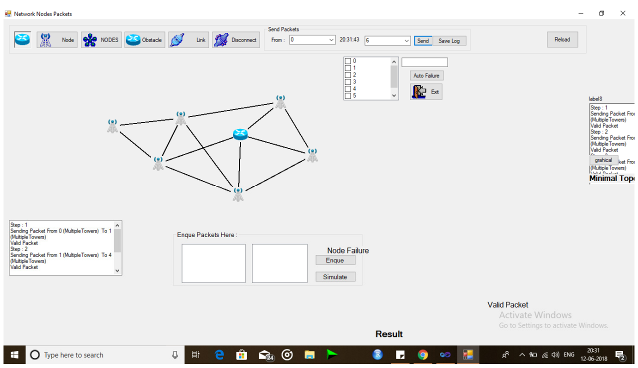

Figure 6: Network topology with obstacals

CONCLUSION

In this project we have designed the network model which is used to find the shortest path among the nodes and maintain the routing table. The source node sends the packets to destination node by finding the shortest path among them. We focus on monitoring network performance, packet tracker, route cost, and packet loss detection. During packets transmission, every intermediate node in the discovery route create routing table to store the information regarding neighbor node and the destination node information. The routing table information updated for every packet transmission during the message transmission.

REFERENCES

1. Al Agha K., Pujolle G., Vivier G., "Reseaux de mobiles et reseaux sans fil", 2 nd edition, Eyrolles, 2005

2. Basagni S., Chlamtac I., Syrotiuk V. R., Woodward B. A., " A Distance Routing EffectAlgorithm for Mobility (DREAM) " , In Proceedings ACM/IEEE Mobicom, pages 76-84,October 1998.

3. Camp, T., Boleng, J., Williams, B., Wilcox, L.,Navidi, W., "Performance comparison of twolocation based routing protocols for ad hoc networks", INFOCOM 2002, Twenty-First Annual joint conference of the IEEE Computer and Communications Societies, Proceedings,IEEE, Volume. 3, 2002, pp. 1678 –168.

4.Clausen T., Jacquet P., Viennot L., "Comparative Study of Routing Protocols for Mobile Ad hoc Networks", Med-Hoc-Net'02, Sardegna, Italy,September 2002.

5.Daniel Camara, Antonio Alfredo F. Loureiro, "A Novel Routing Algorithm for Hoc Networks" , Baltzer Journal of Telecommunications Systems,18:1-3, Kluwer Academic Publishers, 2001, pp.85-100.

6.DiCaro G., Dorigo M., " Ant Colonies for Adaptive Routing in Packet-Switched Communications Networks " , Proceedings PPSN V - Fifth International Conference on Parallel Problem Solving from Nature, Amsterdam, Holland, September 27-30, 1998, pp. 673-682.

7.Gerharz M.L, De Waal C., Frank M., Martini P., "Link Stability in Mobile Wireless Ad Hoc Networks" , Proceedings of the 27th Annual IEEEConference on Local Computer Networks (LCN),Tampa, Florida, November 2002.

8.Gunes M., Sorges U., Bouazisi I., "ARA - the Ant Colony Based routing Algorithm for MANETs" , Proceedings ICPP Workshop on Ad hoc Networks, Vancouver, Canada, 2002, pp.7985.

9.Heusse M., Snyers D., Guérin S., Kuntz P.,"Adaptive agent-driven routing and load Balancing in communication network", Proceedings ANTS'98,First International Workshop on Ant Colony Optimization, Brussels, Belgium, October 15-16, 1998.

10.Iwata A., Chiang C.-C., Pei G., Gerla M., Chen T.-W., "Scalable Routing Strategies for Ad Hoc Wireless Networks", IEEE Journal on Selected Areas in Communications, Special Issue on Ad- Hoc Networks, Aug. 1999, pp.1369-1379.

11.Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot, Thomas Clausen, "Optimized Link State Routing Protocol", Internet Draft, draft-ietf-manet-olsr-04.txt, work in progress, June 2001.

12. Akkaya, K.; Younis, M. A Survey on Routing Protocols for Wireless Sensor Networks. Ad Hoc Netw. 2005, 3, 325–349.

13. Akyildiz, I.; Pompili, D.; Melodia, T. Underwater Acoustic Sensor Networks: Research Challenges. Ad Hoc Netw. 2005, 257–279. Sensors 2009, 9 8420

14. Karl, H.; Willig, A. Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons: Chichester, West Sussex, UK, 2005.

15. Tilak, S.; Abu-Ghazaleh, N.B.; Heinzelman, W. A Taxonomy of Wireless Micro-Sensor Network Models. Mob. Comput. Commun. Rev. 2002, 6, 28–36.

16. Kulik, J.; Heinzelman, W.; Balakrishnan, H. Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks. Wirel. Netw.2002, 8, 169–185.

17. Intanagonwiwat, C.; Govindan, R.; Estrin, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM), Boston, MA, USA, August, 2000.

18. Braginsky, D.; Estrin, D. Rumor Routing Algorithm for Sensor Networks. In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, GA, USA, September, 2002; pp. 22–31.

19. Yao, Y.; Gehrke, J. The Cougar Approach to In-Network Query Processing in Sensor Networks. SIGMOD Rec. 2002, 31, 9–18.

20. Sadagopan, N.; Krishnamachari, B.; Helmy, A. The ACQUIRE Mechanism for Efficient Querying in Sensor Networks. In Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA), Anchorage, AK, May, 2003; pp. 149–155.

21. Rao, A.; Ratnasamy, S.; Papadimitriou, C.; Shenker, S.; Stoica, I. Geographic Routing without Location Information. In Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MOBICOM), San Diego, CA, USA, September, 2003; pp. 96–108.

22. Seada, K.; Helmy, A. Geographic Protocols in Sensor Networks; Technical Report 04-837; Computer Science Department, University of Southern California: San Diego, CA, USA, 2008.

23. Seada, K.; Zuniga, M.; Helmy, A.; Krishnamachari, B. Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks. In Proceedings of the Second International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, November, 2004; pp. 108–121.

24. Zhou, H.; Mutka, M.W.; Ni, L.M. Reactive ID Assignment for Sensor Networks. In Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Washington, DC, USA, November, 2005.

25. Lin, J.; Liu, Y.; Ni, L.M. SIDA: Self-organized ID Assignment in Wireless Sensor Networks. In Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems Conference (MASS), Pisa, Italy, October, 2007.

26. Boukerche, A.; Nakamura, E.F.; Loureiro, A.F. Algoritms for Wireless Sensor Networks. In Algorithms and Protocols for Wireless Sensor Networks; Boukerche, A., Ed.; John Wiley & Sons: Hoboken, NJ, USA, 2009. 5. Akkaya, K.; Younis, M. A Survey on Routing