RESEARCH ARTICLE OPEN ACCESS

A fast and reliable dense-mode multicast routing protocol

VIJAYKUMAR H NAYAK
SELECTION GRADE LECTURER (CSE)
GOVERNMENT POLYTECHNIC LINGASAGUR
vknayak146@gmail.com

ABSTRACT
Wireless sensor network applications include a broad range of circumstances. In most of them, the network consists of a sizable number of nodes dispersed across a large region, not all of which are linked directly. Then multi hop communications facilitate data sharing. The task of finding and maintaining network routes falls to routing protocols. Nonetheless, the suitability of a certain routing protocol mostly relies on the nodes' capabilities and the demands of the application. An overview of the primary routing methods suggested for wireless sensor networks is given in this study. Furthermore, included in the study are the efforts made by Spanish institutions to optimize routing protocols for wireless sensor networks.

Keywords: routing protocol; wireless sensor network

I. INTRODUCTION

The purpose of wireless sensor networks (WSN) is environmental monitoring. A wireless sensor node's primary function is to detect and gather data from a certain domain, process it, and send it to the application's sink. But maintaining direct contact between a sensor and the sink could compel nodes to broadcast at such a high level that it soon exhausts their resources. As a result, nodes must cooperate to guarantee that remote nodes may interact with the sink. This is how intermediary nodes spread messages, creating a path to the sink that has many connections or hops. Since sensors have limited capabilities, it may have been possible to design the connection with the sink without a routing protocol at first. The flooding algorithm sticks out as the most straightforward solution with this assumption. According to this technique, data is broadcast by the transmitter and then successively retransmitted to reach the desired location. Its simplicity, meanwhile, has several serious disadvantages. Initially, when nodes receive duplicate copies of the same data packet, an implosion is recognized. Subsequently, with numerous nodes in the affected region potentially detecting the occurrence, several data packets with comparable information are added to the network. Furthermore, the nodes do not restrict their functionality based on their resources.

The gossiping method is used in one optimization [1]. By transmitting the message to a specific neighbor rather than telling all its neighbors as in the standard flooding process, gossiping prevents implosion. Overlap and resource blindness persist, however. Moreover, these annoyances become more noticeable as the network's node count rises.

In wireless sensor networks, routing protocols become essential because of the shortcomings of the earlier approaches. However, it is not an easy process to include a routing system into a wireless sensor network. The identification of nodes is one of the primary constraints. Because many nodes make up wireless sensor networks, it is impractical to manually issue unique IDs [2]. It is not advised to utilize potentially unique identifiers like the GPS coordinates or the MAC (Medium Access Control) address as they need a large payload to be sent in the messages [3]. In wireless sensor networks, this disadvantage may be readily solved however, since the target node of a given packet can be identified without the need for an IP address. To be more precise, attribute-based addressing is more suited to the unique requirements of wireless sensor networks. Here, the ultimate destination is determined using a characteristic like sensor kind and node location.

Routing protocols are responsible for creating and maintaining routes between distant nodes when nodes have been discovered. Because routing protocols function differently, they are suitable for certain purposes. There are several suggestions for routing algorithms in wireless sensor networks in the relevant literature. To aid in the comprehension of the many routing strategies that may be used in wireless sensor networks, this study attempts to describe the most relevant ones. The study describes specific attributed-based, multipath, hierarchical, and spatial routing systems. There is also a description of the most important suggestions from Spain.

This is how the remainder of the paper is organized. While Section 3 outlines the primary design limitations that routing systems in wireless sensor networks must overcome, Section 2 illustrates the fundamental communication paradigms that wireless sensor networks adhere to. The most often used categorization techniques for routing protocols in these kinds of networks are shown in Section 4. The optimization techniques used by these routing protocols are described in Section 5. As seen in Section 6, the use of these methods results in attribute-based, geographic, hierarchical, and multipath routing protocols. With an emphasis on our contributions, Section 7 presents an overview of the major schemes for routing protocols created in Spain. Lastly, the primary findings of this study are presented.

II. ALGORITHM PARADIGMS FOR WIRELESS SENSOR NETWORK

Nodes must communicate with one another for sensor applications to carry out certain operations or algorithms. On wireless sensor networks three types of algorithms may be used [4]:

- Centralized Algorithms: These are carried out by a node with access to the whole network's knowledge. Due to the high expense of data transmission required to inform a node of the overall network condition, these techniques are rather uncommon.
- Distributed Algorithms: Message-passing is used to facilitate communication.
- Local-based Algorithms: The nodes make use of constrained information that was obtained nearby. Using this local data, the algorithm runs on a single node.

When choosing the routing protocol to use in the network, the algorithm paradigm is a crucial consideration. The routing protocol should promote and enhance neighborly communication if localized algorithms are being implemented. However, merging messages that are sent to the central node concurrently—even if they come from multiple sources—may be advantageous for centralized algorithms. Any two pairs of nodes should be able to communicate effectively thanks to the distributed algorithms. Lastly, the more costly solution is required for local based algorithms as they rely on a solution that offers geographic coordinates, such as GPS.

III. DESIGN CONSTRAINTS FOR ROUTING IN WIRELESS SENSOR NETWORKS

Routing protocols in wireless sensor networks are anticipated to meet the following characteristics [5] since sensors have limited processing, radio, and battery capacity.

- Autonomy: In wireless sensor networks, the notion of a separate unit in charge of radio and routing resources is untenable since it presents a potential point of attack. The routing protocols are moved to the network nodes as there won't be a centralized body to determine the routing decisions.
- Energy Efficiency: Routing protocols should keep a network's lifespan extended and its connectivity high enough to enable node-to-node communication. It is significant to note that because most sensors are positioned at random, it is not possible to replace the batteries in them. In certain situations, the sensors are not even accessible. In wireless subterranean sensor networks, for example, certain devices are submerged to enable soil sensing [6].
- Scalability: Routing systems should be able to handle the hundreds of nodes that make up wireless sensor networks.
- Resilience: Environmental factors or battery consumption may cause sensors to abruptly stop working. Routing protocols should account for this possibility so that a backup route can be found in case the node that is now in use fails.
- Mobility Adaptability: The various uses for wireless sensor networks may require nodes to manage their own mobility as well as the movement of the sink or the event they are trying to sense. Routing protocols ought to provide these motions with the necessary assistance.

Hierarchy Role of Nodes in the Network, every sensor node in a flat design plays the same function in the routing processes. However, sensor nodes are categorized by functionality in hierarchical routing systems [8]. After then, the network is split up into clusters or groups. Within the group, a leader or cluster head is chosen to oversee cluster operations and facilitate communication with nodes outside the cluster. Nodes can be differentiated either statically or dynamically.

Data Delivery Model, Data collection and interaction in wireless sensor networks can be achieved in a variety of methods, depending on the application. The information flow between the sensor nodes and the sink is depicted by the data delivery model [7]. The classes of data delivery models are as follows: hybrid, event-driven, continuous, and query-driven. The nodes in the continuous model periodically broadcast at a predetermined rate the data that their sensors are gathering. Query-driven techniques, on the other hand, require nodes to wait to report on the data they have sensed. When an interesting event takes place, sensors in the event-driven model release the data they have gathered. Lastly, the hybrid schemes integrate the earlier tactics such that sensors respond to requests in addition to periodically informing about the data they have acquired. They are also configured to inform about interesting happenings.

3.1: PIM-DM operation

A source-based, data-driven, dense-mode, soft-state multiplexing protocol, PIM-DM [8] depends on RPF checks in addition to a unicast routing protocol. Since PIM protocols are independent of any specific unicast routing protocol, they are referred to as independent protocols. Hi there, procedure Through the recurring exchange of Hello messages, PIM-DM routers that are interested in taking part in the multicast routing protocol build and preserve neighborhood ties with their surrounding routers. As soon as it receives the initial Hello, a router considers an unknown router to be its neighbor.

Neighbor Liveness Timer (NLT) is the timer that controls the liveness of neighborhood relationships (based on Hold Time) and Hello Timer (HT), which controls the periodic transmission of Hello messages (based on Hello Time). One feature of hello messages is the Generation ID field, which is used to identify neighbor reboots. This field appears in every Hello message an interface sends and is a random number that is created when the interface is started. Therefore, a router discovers that a neighbor has rebooted if it receives two Hello messages from the same neighbor with different Generation IDs.

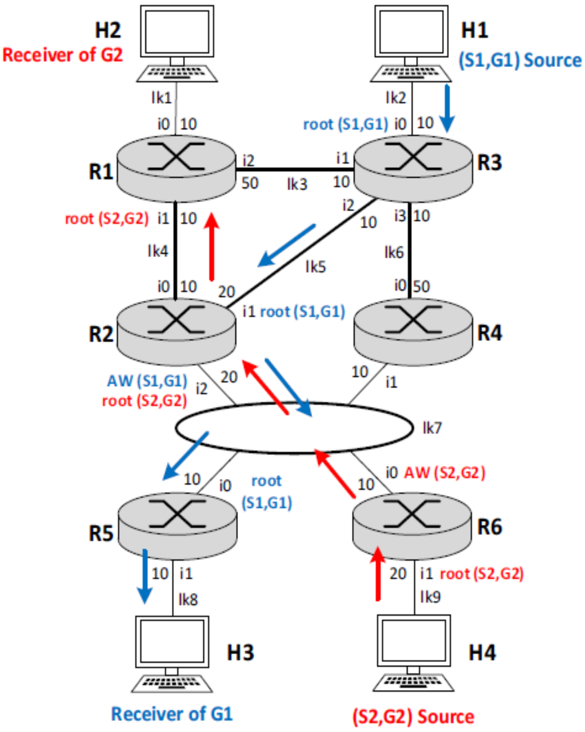

Figure No. 2 : Two multicast trees overlaid on the same physical network.

message from it. Consequently, neighborhood relationships do not have to be reciprocal, as mandated by other routing protocols (like OSPF); in fact, PIM-DM Hello messages do not even contain the link neighbors' addresses. The Hello Period, which is the default duration of Hello transmissions, is measured in seconds. Furthermore, a neighbor is deemed dead by a router if it stops sending Hello messages to it for longer than a certain amount of time, known as Hold Time. By default, Hold Time is set at 3.5 × Hello period, or 105 seconds. Greeting messages contain the Hold Time value. In this instance, a Hello message is delivered to the neighbor who has rebooted right away, without having to wait for the next planned time for the image.

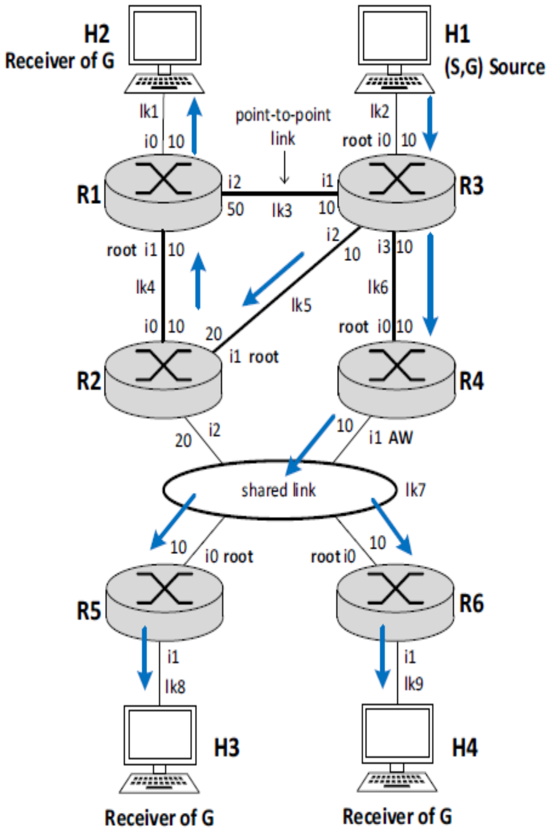

Figure No. 1: Broadcast Tree Example

Every tree has its own unique set of interface roles (root versus non-root) and AW. For instance, interface i1 is the root for (S1, G1) in router R2, whereas interface i2 is the root for (S2, G2); also, i2 is the AW for (S1, G1). Keep in mind that some routers are excluded from the multicast trees since they are not interested. To be more precise, R4 does not belong to either of the two trees, R1 and R6 do not belong to (S1, G1), and R3 and R5 do not belong to (S2, G2).

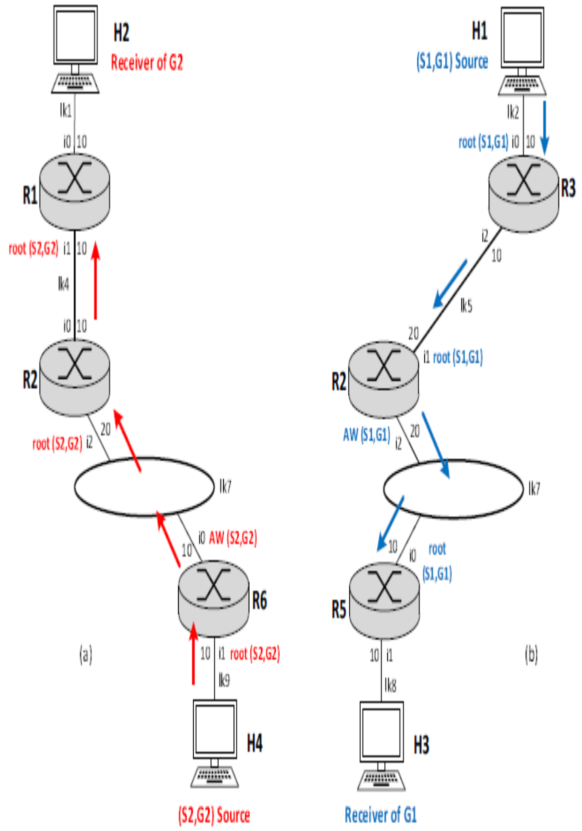

Figure No. 3: Separate views of the multicast trees

Construction of broadcast trees for every (S, G) pair in PIM-DM networks, one broadcast tree is constructed. The protocol is referred to as data-driven because the source initiates the broadcast tree's building when it begins delivering multicast data. Using the RPF approach, data packets are flooded from the originator routers to all other routers when a source S begins transmitting multicast data for group G. To be more precise, the router uses the unicast routing table to determine if a packet was received through a root or non-root interface when it is received at a router interface for the first time. When a packet is received via a root interface, the router creates a multicast routing table for the appropriate (S,G) tree and routes it across all of its non-root interfaces; if not, it is rejected. The root interface is the one with the lowest RPC, excluding originators; an originator's root interface is the interface that is connected to the source's subnet regardless of RPC. The non-root interfaces are all initially set to the FORWARDING state. In this manner, the first packet is sent to every network router without going around indefinitely.

IV. FINAL RESULT

Two routers connected by a single link were used to conduct these tests. Without using any trees in the synchronization process, we concentrated on the proper creation and upkeep of neighborhood ties in the face of reboots and failures. We examined the progression of the synchronization stages from UNKNOWN to SYNCED and examined the sequence numbers (BootTime, SnapshotSN, and SyncSN) and flags (Master and More) in the Hello and Sync messages. The following test were performed:

Test 1 : establishing a neighborhood link in the absence of any trees between two unknown neighbors
Test 2 : repairing the relationship in the community following a known neighbor's reboot
Test 3 : Neighborhood relationship break after known neighbor fails

The initial tree creation and its reconfiguration in the event of neighbor reboots and failures were the focus of these studies. They were carried out utilizing Figure 57's network. Because this architecture contains a shared link linking numerous non-root interfaces and provides multiple pathways, we chose it. When the unicast routing protocol was first set up, the cost of each interface was 10. Thus, all routers had eth0 as the root interface, except for router R7, which had eth2. To permit the initial flooding of multicast traffic, the routers were set up with initial downstream interest in every interface.

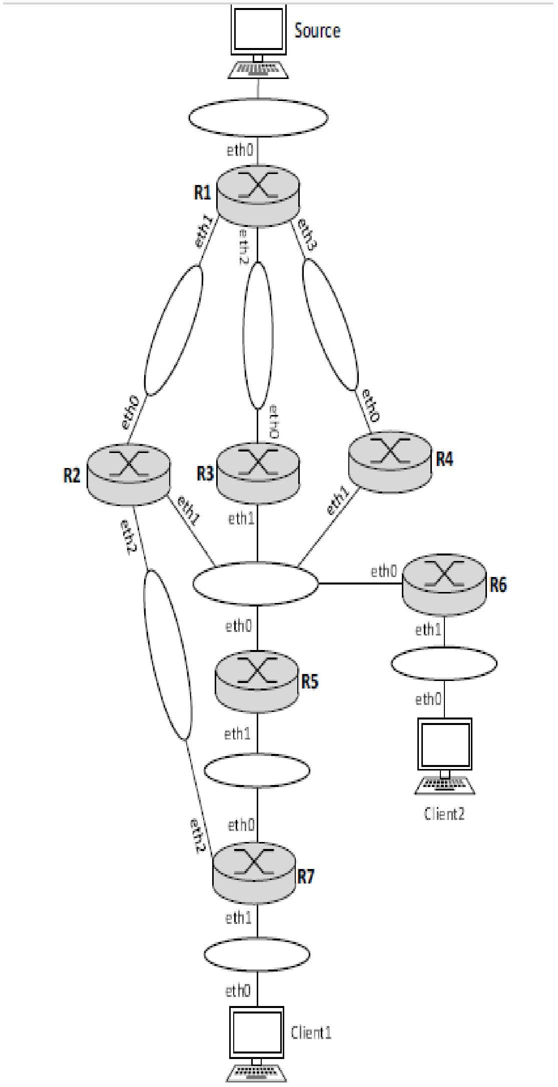

Figure No. 4: First Network Used for Testing

In this test, we examined how the shared link synchronizes when an interface reboots. We used the source to send multicast packets to 20 distinct groups to start 20 separate trees prior to the test. We set up the routers to only send out information on five trees each Sync message to test the fragmentation process. We observed how upstream and tree states changed over time, as well as how Sync messages were exchanged—particularly about the trees that were reported in each message—and how interest messages were sent after synchronization.

The tests listed below were carried out:
• Test 1: Resynchronization of non-root interface following reboot
• Test 2: Root interface synchronization following a reboot
We restarted R4's non-root interface in test 14,

which caused synchronizations with all its neighbors that were linked to the shared link. To compel the router to lose its parent and retrieve the tree information through synchronization with the routers connected to the shared connection, we rebooted R5's root interface in test 1.

V. CONCLUSION

HPIM-DM (Hard-state Protocol Independent Multicast - Dense Mode), a revolutionary multicast routing protocol that can be thought of as a hard-state variant of PIM-DM. PIM-DM has several problems that HPIM-DM fixes, which leads to poor convergence and renders PIM-DM inappropriate for high-speed networks. The introduction of (i) mechanisms that guarantee the reliable transmission and sequencing of control messages, (ii) the idea of upstream neighbors—neighbors who can deliver multicast traffic originating from the source—and (iii) a synchronization process that allows a router joining the network to obtain instantaneous information on the active multicast trees made these improvements possible. This eliminates the need for the protocol to send out control messages on a regular basis to update the state, and it allows the protocol to respond quickly to any event that could alter the multicast trees' configuration. Additionally, the protocol was strengthened to withstand replay attacks. Model verification and logical reasoning were used to evaluate HPIM-DM's accuracy. Additionally, we created a complete Python implementation of HPIM-DM and ran several tests on it to confirm the protocol's correctness. PIM-SM (PIM - Sparse Mode), which has convergence issues akin to those of PIM-DM, can benefit from many of the fixes discovered for HPIM-DM. This is left for a later project.

REFERENCE

[1]. E. Rosenberg, A Primer of Multicast Routing, Springer-Verlag New York, 2012.
[2]. R. Wittmann, M. Zitterbart, Multicast Communication: Protocols an Ap- plications, Morgan Kaufmann, 1999.
[3]. C. S. R. Murthy, B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall, 2004.
[4]. D. Waitzman, C. Partridge, S. Deering, Distance Vector Multicast Routing Protocol, Internet Requests for Comments, https://tools.ietf.org/html/rfc1075 (November 1988).
[5]. T. Ballardie, P. Francis, J. Crowcroft, Core Based Trees: An Architecture for Scalable Inter-domain Multicast Routing, ACM SIGCOMM Computer Communication Review 23 (4) (1993) 85–95.
[6]. A. Adams, J. Nicholas,W. Siadak, Protocol Independent Multicast - Dense Mode (PIM-DM), RFC 3973, RFC Editor (January 2005). URL https://tools.ietf.org/html/rfc3973
[7]. P. Oliveira, HPIM-DM state machines, Internal Report, Instituto Superior T´ecnico (November 2018).
[8]. C. Perkins, P. Bhagwat, Highly Dynamic Destination-Sequenced Distance- Vector Routing (DSDV) for Mobile Computers, in: ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234–244.
[9]. J. J. Garcia-Lunes-Aceves, Loop-Free Routing Using Diffusing Computations, IEEE/ACM Transactions on Networking 1 (1) (1993) 130–141.
[10]. B. Albrightson and J. Garcia-Luna-Aceves and J. Boyle, EIGRP – a Fast Routing Protocol Based on Distance Vectors, in: Proc. Interop 94, 1994.
[11]. M. Bhatia, S. Hartman, D. Zhang, A. Lindem, Security extension for ospfv2
[12]. when using manual key management, RFC 7474, RFC Editor (April 2015).
[13]. P. Oliveira, Test to Python implementation of IGMPv2, PIM-DM, and HPIM-DM,
[14]. Internal Report, Instituto Superior T´ecnico (November 2018). URL https://github.com/pedrofran12/hpim_dm/tree/master/docs/
[15]. PythonTests.pdf
[16]. P. Oliveira, HPIM-DM implementation, online; accessed 16 September.
[17]. 2018 (2018). URL https://github.com/pedrofran12/hpim_dm
[18]. U. Chunduri, W. Lu, A. Tian, N. Shen, Is-is extended sequence number.
[19]. tlv, RFC 7602, RFC Editor (July 2015).
[20]. M. Bhatia, S. Hartman, D. Zhang, A. Lindem, Security extension for ospfv2
[21]. when using manual key management, RFC 7474, RFC Editor (April 2015).