International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

PERFORMANCE EVALUATION AND COMPARISON OF WIRELESS SENSOR NETWORKS ROUTING PROTOCOLS IN INTERNET OF THINGS

Polycarp Shizawaliyi Yakoi
MSc, Computer Engineering
Cyprus International University,
Lefkosa, North Cyprus

Hamagham Peter Ishaku
PhD, Energy Systems Engineerng
Cyprus International University,
Lefkosa, North Cyprus

Abstract - IoT has continued to grow bigger since from its inception. Many mobile devices are now available, the internet and its application have only grown bigger and better. As IoT is continually growing, so also is the complexity, as a result issues pertaining routing have also increased. Many researches have been made in attempt to proffer solutions that will either minimize or eliminate these routing issues. Different routing protocols have been designed with different specifications for different applications of the IoT. Also, attempts have been made to implement routing protocols of other types of networks in the IoT.

In this thesis, three Wireless Sensor Networks – Ad-hoc On-Demand Distance Vector, Dynamic Source routing protocol and Optimized Link State routing protocol have been simulated and compared in typical IoT scenarios. Their performance was evaluated using three performance metrics and then they were compared; the performance metrics are Routing Overhead, Average End to End Delay and Throughput. Different numbers of nodes with different percentages of mobile nodes were analyzed. Specifically, number of nodes analyzed were 20, 40, 60 and 70 with the number of mobile nodes 10, 15 and 20 using OPNET while with NS 3 20, 60 and 100 nodes were analyzed. For each of the number of nodes, all the number of mobile nodes were evaluated. The routing protocols were analyzed using the OPNET Simulation Software and NS-3and the environment size for the simulation was 1000m by 1000m.

Keywords: IoT, WSN, routing protocols, AODV, DSR, OLSR, OPNET, NS-3

I. INTRODUCTION

Internet of Things (IoT) has grown over the years in both size and complexity. These complexities have continually grown because of homogeneity of devices and network standards. This has brought along with it so many issues that researchers have been working on continually with a specific end goal to make IoT much better. Some of these issues and problems are scalability, mobility, security, routing of data, privacy, etc.

Routing of data is an extremely big issue in IoT; this is because data itself is a standout amongst the most critical things in any system. Routing of data involves taking data from an end device to another end device by utilizing the correct and most efficient routes available.

In IoT the data being sent being sent between different devices is very important; also important is the integrity of the data, security, availability and scalability. All these are important but are big issues in the routing of data because devices are mostly heterogeneous and so are the networks.

Due to the differences in the device types, networks, memory size and power consumption, it is difficult (and becoming even more difficult) to route data efficiently and efficaciously to transmit data from one device to another. In order to make IoT in general better, the problems involved in data must be solved and in addition the other issues.

A. Problem statement

As the IoT has continually grown bigger, so also has the data routing problems increased greatly. Some of these issues are security, scalability, mobility, availability, context awareness and lots greater. Many protocols have been created over the years and have also been improved multiple times. With all these issues nodes (IoT devices) connected (directly or indirectly) as shown in Figure 1, there is a need for more research in order to make more improvements or even make better and more effective protocols to make routing of data in IoT better.

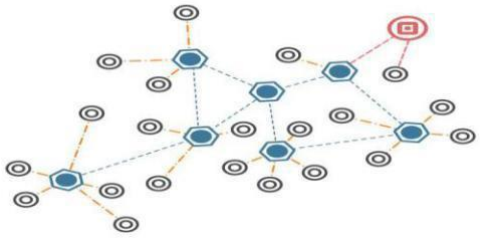

Figure 1: Routing of data through nodes all connected

B. Aim of study

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

The aim of this study is to analyze three routing protocols used in wireless sensor networks – Ad-hoc On-Demand Distance Vector (AODV), Dynamic Source Routing protocol (DSR) and Optimized Link State Routing protocol (OLSR), simulate them in IoT scenarios and then compare how they can be used in IoT.

C. Significance of study

This research will provide more information about WSN routing protocols and how they can be used in IoT; in other words, it will provide more routing solutions or alternatives to some of the IoT routing protocols already in use.

D. Study Outline

This section will provide an outline of the thesis, each chapter and the subject of discussion. The other chapters in this thesis are summarized in the preceding paragraphs.

Chapter Two: Literature review. This chapter will discuss different papers and journals related to IoT, routing of data in IoT, issues of the routing of data in IoT and protocols used. Also, other works previously done will be thoroughly discussed in order to lay a proper foundation for this work.

Chapter Three: Research methodology. In this chapter, the different simulations that were done on the protocols will be discussed. Basically, the simulators - Optimized Network Engineering Tools (OPNET) Modeler and NS-3will also be discussed in depth.

Chapter Four: Data analysis and discussion of data findings. In this chapter all the results from the simulations done in OPNET and NS-3 will be analyzed and discussed, inferences will then be made from the findings. These inferences will then be used to decide on improvements that will or can be made so that data routing in IoT can be better.

Chapter Five: Conclusion. This chapter concludes the thesis, also future works and recommendations will be discussed here.

II. LITERATURE REVIEW

Routing in IoT simply refers to a process in a network where nodes within the network search for a suitable path to send data or receive data. In this section, issues or challenges which can be faced at some stage in routing in IoT will be discussed. Also to be discussed are some routing protocols that have been implemented for the IoT, including WSN routing protocols which can be used in IoT in some certain scenarios. The next sections will discuss routing protocols and routing issues in IoT.

A. Data Routing Issues

Data routing in the IoT is a very important subject; in fact, its importance can never be overemphasized. This is due to the only fact that data is always the center and most important in almost every field or company or organization. Usually, questions are asked about the data, it may be routed for sending the data, how the data will be sent, availability, data integrity, security and privacy. Every of these reasons are very vital in its own right; there have been many forms of studies on all of them.

Whenever data is involved in any process, three questions always pop up – data integrity, security and privacy. These always come up because people are concerned about the authenticity of the data they get, how secure it is and also they do not want their data shared with someone else. Another vital question also asked always especially by big companies is availability, everyone needs to get whatever data they need instantaneously.

In order to answer or try to answer these questions and many others that have been or are being asked in IoT about routing, researchers have done great work in providing some routing protocols for the IoT. Also, protocols in other fields such as wireless sensor networks (WSNs) have been used in IoT to solve routing issues. Within the subsequent section, protocols for routing in the IoT will be discussed. A number of these are RPL, Naive routing protocol, Probabilistic routing protocol, 6LoWPAN, etc.

B. Routing Protocols and Techniques in IoT

There are many routing protocols currently available; some are standard for the IoT while others are not. The standard protocols are those built mainly for IoT while the non-standard protocols are used for other applications such as WSN, some of these have been implemented on different IoT environments. In this section, routing protocols used for Wireless Sensor Networks, IoT and AdHoc Wireless networks will be discussed in detail.

According to Al-Karaki and Kamal (2004), there exist different classifications or categories of routing protocols based on protocol operation or functionality and network structure. Routing protocols can be categorized as reactive or proactive protocols. In reactive routing the protocol only looks for a route to a destination when needed – it is also called on-demand routing. In proactive routing, periodic messages are used to send messages to nodes about its neighbourhood and as such are likely to have a route to destinations always. In this section, routing protocols, as well as the different classifications, will be discussed.

1) Routing Protocol for Low-Power and Lossy Networks (RPL)

RPL is a distance-vector which is based on IPv6 and is independent of the link which is used for routing, it is also a source routing protocol ones (Alahari and Hema, 2017) (Jeba and Kamala, 2016). It was made for low-power and lossy networks and in 2011 it was standardized by IETF (Iova et al., 2016). It is most often considered the de-factor routing protocol for the IoT.

As already discussed briefly, RPL is said to be made for low-power and lossy networks (LLN), so what then are LLNs? To understand the concept of RPL, this question has to be answered. Another question which needs to be addressed before further discussion on RPL is what distance-vector is and what source routing means also.

A protocol is said to be a distance-vector if it's nodes have the ability to manipulate vectors or arrays of distances to other nodes in the network. This means that the nodes in the protocol do have intra-domain interaction between them. According to Richardson and Robles (1994), in order to have an effective interaction between the nodes, there is a need for minimum complexity in computations and message overhead; also each

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

node must inform other nodes (neighbours) of any change in topology. The network topology is the pattern of arrangement in which nodes are connected in a network. A distance vector protocol always calculates the direction (address of the next hop) and distance (cost to reach a node) to any node in the network. Every node
keeps a vector of the minimum distance (route with the smallest cost) to every node.

LLN is the type of constrained-node network. A constrained-node network is a network which is made of nodes that have some limitations. ―LLN: Low-Power and Lossy Network. Normally created from many embedded devices with constrained power, memory, and processing resources interconnected by a variety of links, such as IEEE 802.15.4 or low-power Wi-Fi. There is a wide scope of application areas for LLNs, including industrial monitoring, building automation (heating, ventilation, and air conditioning (HVAC), lighting, access control, fire), connected home, health care, environmental monitoring, urban sensor networks, energy management, assets tracking, and refrigeration.‖ – RFC 7228

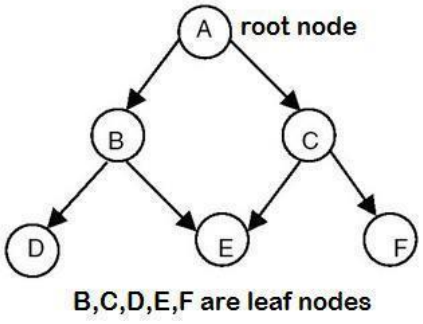

Figure 2: A DAG, root node and leaf nodes

RPL is discussed earlier is a distance-vector routing protocol which utilizes source routing and is the de-facto standard for the IoT. RPL organizes the nodes in the network in a topology as a graph called the Direct Acyclic Graph (DAG). An example of a DAG is shown in Figure 2. The DAG is then divided into one or more Destination Oriented Graphs (DODAG). A DODAG is a directed graph made of leaf nodes without cycles which are directed towards a single root node (Richardson and Robles,
1994) (Salman and Jain, 2015). Figure 3 shows a DODAG. Every traffic from each leaf node within the topology is being routed to the root node through only one route. Each node keeps more than one parents to the root (route) but a chosen one is preferred for upward forwarding of data packets to the root node; the other routes are kept as backups (Richardson and Robles, 1994).

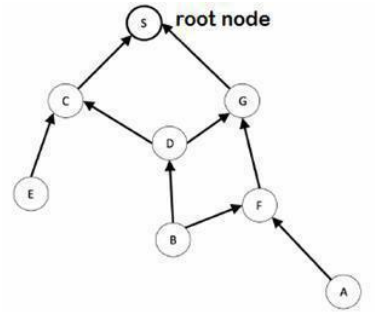

Figure 3: A DODAG with all leaf nodes directed towards S (root)

In RPL, the root node initiates the network topology by sending messages containing control packages called DODAG Information objects (DIO). The DIO contains information about the graph or network. The leaf nodes receive these messages and process them. After processing the DIO, the nodes decide by using some rules whether to join the network or not. The nodes also use the rules and information to decide which among its neighbours will be its parent node – this might be the root or the parent of a small DODAG. During communication each node sends a Destination Attachment Object (DAO) to its parent if it needs to send data to another node, the parent will process the DAO and transfer the required packets to the destination. The RPL supports three types of communication between nodes – multipoint-to-point, point-to-multipoint and point-to-point.

2) Cognitive RPL (CORPL)

Cognitive and opportunistic RPL - CORPL is an extension of the RPL designed for cognitive networks – a network with a perceptive process which has the ability to observe present conditions of the network, act based on the condition and self-learn from the result of its actions (Thomas et al., 2005). Just like RPL, CORPL uses DODAGs but with some modifications. In studies carried out in Aijaz et al., (2015), there are some problems that are encountered while using the RPL in cognitive networks, which is supposed to be the default routing protocol IoT. In order to tackle those issues, an improved variant of RPL – CORPL was built for cognitive networks. CORPL was made to use the DAG just like the RPL, but with an opportunistic approach. In CORPL, there are two major steps; these are: selection of a forwarder set and the unique forwarder selection. In the first step, every node in the network will select as many next hop neighbours as possible. In the second step, the nodes determine the best receiver among the selected forwarder set, using a coordination scheme; when the best receiver is determined, the node will then allow it to forward the relevant packets. Each node keeps a forwarder set from which the next hop/forwarding node is selected opportunistically. According to Aijaz and Aghvami (2015), the use of opportunistic forwarding in CORPL improves end-to-end throughput and reliability; this is achieved by making use of the inbuilt properties of the wireless channel – this is actually a big concern in lossy networks.

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

3) Lightweight On-demand Ad hoc Distance Vector Routing

– next generation (LOADng)

LOADng is a reactive protocol which was designed to provide efficiency, scalability and security in routing in LLNs; It is a distance-vector protocol which is lightweight. It does not keep a routing table for different nodes but works on-demand, it initiates a route discovery whenever there is a need to transfer packets to a destination node; as explained in earlier sections, reactive protocols experience reduced routing overhead and memory consumption compared to the proactive ones (Alahari and Hema, 2017).

Just like other reactive routing protocols, LOADng has three different messages with which it works; these are Route Request (RREQ), Route Reply (RREP) and Route Error (RERR). The sender node sends RREQ when there is a need for packet delivery in order to discover a path to the destination node; the destination node will send back an RREP after receiving the RREQ from the sender. When there is a link failure/break, the destination sends back RRER to the original sender of the packets it is receiving.

As mentioned earlier, the LOADng is lightweight. The designers of this protocol built it by using minimal core and a small set of protocol operations, also it was built with simple implementation requirements thereby making the code footprint small and the operation state requirements as well. LOADng is quite different from its predecessors; its characteristics are discussed vividly below:

• Modular design: this is simply the lightweight core of the protocol. The core makes the protocol extensible with a packet format that is flexible.

• Flexible addressing: Lengths of address from 1 to 16 are supported. The only requirement is that all addresses inside a given routing domain of the network must be of the same length.

• Metrics: there is a support for many metric types other than simple hop-count.

• Destination-replies: intermediate routers are not allowed to respond to RREQs, only the destination node is allowed, this reduces the complexity of operations, reduce the size of messages and improve security.

LOADng involves three protocol operations – route discovery, path maintenance and path metrics.

4) Collection Tree Protocol (CTP)

This routing protocol was developed primarily for WSNs [16]. It is a distance – vector routing algorithm. Before the RPL was developed, CTP was the de-factor routing standard for the TinyOS. It is widely considered as a general reference protocol for WSNs (Colesanti and Santini, 2011) (Colesanti and Silvia, 2010). CTP constructs and maintains a tree-based topology using routing messages also called beacons, this reports the data messages to the sink which is the root of the network. To ensure that routing messages are sent to the root, CTP uses the adaptive beaconing mechanism.

5) Channel-Aware Routing Protocol (CARP)

CARP is an underwater wireless sensor network routing protocol. It employs the use of multi-hop data delivery to the sink of the underwater WSN ones (Alahari and Hema, 2017). It is a cross layer protocol which takes advantage of link quality information to determine the cross layer delay (Basagni et al., 2012). Using the information pertaining link quality, CARP selects nodes which have up to date history of successful transmissions to their neighbours. The protocol combines the link quality with the hop count, which is the simple topology information to be able to connectivity voids and shadow zones. It is also able to select robust links by taking advantage of power control.

At the start of network setup, the sink (root node) sends a HELLO broadcast message to every node inside the network. With the broadcast message, every node is able to get its hop count – distance to the node, which is very necessary. Using PING and PONG messages, whenever there is any packet that needs to be transferred, the sender node selects the most suitable relay to the destination node. PING is the message sent by the node to initiate a transfer of packets and PONG is the message sent by any node which receives the PING message, that node forms a relay to the destination node (Basagni et al., 2012).

During the exchange of the PING and PONG messages to get a relay, time is recorded. In addition to the time, goodness is computed for each node, the goodness value is then used to calculate the link quality of all possible relays to the destination node. To transmit the packets, the relay with the best link quality is chosen to transfer the packets. While sending the PING messages initially, the power used to send them is also computed; this enables CARP to take advantage of power control to select robust links for packet transfer (Basagni et al., 2012).

6) E-CARP

This protocol is an enhancement on the CARP to support greedy and location-free hop-to-hop routing to ensure energy efficient forwarding of packets from the sensor nodes to the sink. In CARP data acquired by the sensor nodes are not being neglected, with their presence in the network; sometimes unwanted forwarding might be done from those nodes in the network, E-CARP is built to solve this problem by enabling caching of sensory node data at the sink ones (Alahari and Hema, 2017).

Another feature that is not being exploited in CARP is reusability of relays in the network. In situations where the network is steady, there is usually no need for a PING-PONG message transfer between nodes. E-CARP is built to exploit reusability of previous links by giving previously used links a high priority before initiating a transfer ones (Alahari and Hema, 2017).

7) Naive Routing

In Naive routing, flooding is the main idea of sending data. Each node in the network has the ability to communicate with other nodes that are within its reachable range. The source node (sender) sends a flood of request packets, also called beacons into the network (Jeba and Kamala, 2016). When the destination node (receiver)

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

receives a beacon, it sends back a route reply message to the sender and then a communication link is set up between the nodes. The naive routing is actually not a routing protocol but a category or classification of routing protocols based on functionalities. Popular naive routing protocols are DSR, DSDV and AODV.

8) Hierarchical Routing

This is also another category of routing protocols just like Naive routing. In hierarchical routing nodes are all involved together in a multi-hop communication within a cluster based on polling. In the cluster, data aggregation and fusion are done so as to reduce the number of messages transmitted to the root of the network - sink. Usually, a cluster head exists and does all communication for the members of the cluster. The criteria for forming clusters are basically the amount of energy of the nodes' sensors and how close the nodes are to the cluster head. Low-energy adaptive clustering hierarchy (LEACH) is a routing algorithm which is a very good example of hierarchical routing (Jeba and Kamala, 2016) (Akkaya and Younis, 2005).

9) Query based Routing

Query based routing protocols can also be referred to as data-centric routing protocols. In query based routing, the fundamental idea is to spread of data within the network. The sink disperses queries to regions in the network then awaits data from the sensors available in those locations. A node that
sends queries can get data from any other node. Attribute-based naming is very important in query based routing, this is because queries are sent in order to retrieve data - properties of the data need to be specified. Examples of protocols in this category are Sensor Protocols for Information via Negotiation (SPIN) and Directed Diffusion (Jeba and Kamala, 2016) (Akkaya and Younis, 2005).

10) Multipath Routing

The measure of resilience or fault tolerance of a protocol is measured by how likely that protocol can discover a path between source and destination when the existing path between them fails, this is shown in Al-Karaki and Kamal (2004). In multipath routing, instead of the normal single path used to connect nodes by many routing protocols, multiple paths are used to connect the nodes – this is done to improve the performance of the network.

In order to increase the fault tolerance of a network, a multipath protocol performs a trade-off between energy consumption and traffic generation with multiple paths. This means that the protocol will maintain multiple paths between the source and destination nodes; fault tolerance is improved but consumption of energy and traffic generation is made worse.

Much research has been done on multipath protocols to improve the fault tolerance as well as maintain a low (as low as possible) energy consumption and traffic generation in the network. According to Dulman et al., (2003), multipath routing was used to make WSNs more reliable. Also, multipath routing proved very useful in data delivery in unreliable environments. Directed diffusion is a good example of multipath routing.

11) Probabilistic Routing

In probabilistic routing of data is based on some probabilistic value which has already been calculated (it is similar to heuristics in neural networks and other related fields) [3]. Gossiping is a simple way of getting the necessary data to compute these probabilities. Data packets are flooded into the network like a rumour but with probability, say p. In probabilistic flooding, traffic overhead is minimized, unlike in other mechanisms that use the flooding technique. This because the flooding of data packets is done only once. To decide which nodes form a route to a destination node, a specialized approach is used to properly observe the previous history of packet delivery and the pattern of movement.

12) Ad-hoc On-Demand Distance Vector (AODV)

AODV is an on-demand routing protocol for Ad hoc networks. On-demand, also called reactive routing involves the routing protocol making available routes in real time i.e. when they are needed, the protocol does that by sending floods of Route Request Packets into the network ("List of ad hoc routing protocols ", 2017). This protocol is used in Ad hoc networks to ensure that the network is self-starting, dynamic, and maintains multi hops between participating nodes in the network. With these, the moving nodes can get routes to new destinations as fast as possible. Also, these nodes do not need to keep routes that are inactive at any moment in time. Another important feature of the AODV protocol is the absence of loops which makes it possible for the nodes to converge quickly and easily when there is a change in topology in the network (Perkins et al., 2003). One feature that makes AODV stand out is the use of destination sequence number; it is used for every route entry. It is created by the destination node, then included with route information and then sent to nodes that sent a request(s).

AODV defines and makes use of three message types: Route Requests (RREQs), Route Replies (RREPs) and Route Errors (RERRs). RREQs are packets that are sent by nodes when there is a need for a route, usually, these packets are flooded into the net work on demand. RREPs are those packets which are used as replies when a route is found. The potential destination, after receiving an RREQ unicasts an RREP back to the originator of the RREQ in order to establish a route. The RRER is a message which is sent to notify other nodes in a network when there is a link break or failure in a link that is active; usually, the nodes monitor the status of active routes between them and next hops (Perkins et al., 2003).

As discussed earlier, AODV is a routing protocol used in ad hoc networks with numerous moving nodes. These nodes could be hundreds, thousands or even millions. It can handle any level of rate of mobility of nodes, also traffic level of data. With AODV scalability and performance is improved as traffic overhead is eliminated and nodes trust each other, where trust is built on some inbuilt knowledge or configuration(s).

13) Adhoc On-Demand Multipath Distance Vector (AOMDV)

In some ad hoc networks, link failures and route break is a normal phenomenon, in such networks, there is a need for rapid discovery of new routes when there are such route

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

breaks. In AODV and other single path protocols, new route discovery is very difficult, since only active routes are maintained by the nodes. The idea in AOMDV is simply – maintain as many redundant paths as possible. With multiple redundant routes available, new route discovery in any case of route break or link failure is now much easier. Thus, it can be concluded that AOMDV is a specially designed protocol for highly dynamic or unstable networks characterized by frequent link failures and route breaks.

In AOMDV routing information which is kept by the nodes is also used, just as is done in AODV. As discussed earlier, overhead is eliminated in AODV, therefore the minimal amount of overhead will be used to maintain multiple paths in the network. Basically, the AMODV comprises of two parts: maintaining multiple paths free of loops using a route update rule and finding paths whose links are disjoint using a distributed protocol (Marina and Samir, 2001).

14) Dynamic Source Routing

Dynamic Source routing (DSR) is another routing protocol which is used for ad hoc networks with multi-hops. It is made up of two basic mechanisms, which are: route discovery and route maintenance. These mechanisms work on-demand, for this reason, DSR is suitable for ad hoc networks (Johnson et al., 2001).

In route discovery, a node which needs to send a package to another node obtains a source route. It is only used when the sender node is sending packets to the receiver node for the first time (Johnson et al., 2001). The route discovery is achieved by using the ROUTE REQUEST and ROUTE REPLY messages [10].

In route maintenance, while using a source route a node is able to detect another node which it wishes to send packets.
This is usually used when there is a change in the network topology which causes the already established routes between the sender and receiver node. To indicate a break in the link, the network sends a ROUTE ERROR to the sender node [10]. When a node wants to send packets to another node, route maintenance checks and specifies if there is a break in the link between the nodes; if there is a break, the sender node will attempt to use an already known route or route discovery is done again (Johnson et al., 2001).

15) Optimized Link State routing Protocol

Optimized Link State Routing (OLSR) protocol is a proactive routing protocol for mobile ad hoc networks. The OLSR uses an optimization on the already established Link State routing algorithm. In the Link state routing algorithm, the complete network topology needs to be known, nodes have to obtain information on destinations together with link states and weights, then decide the shortest path using Dijkstra's algorithm to any destination. Also, nodes have to create and maintain routing tables which keep information about destination nodes and addresses of the next hops.

In OLSR there is always a route when needed, this is because the protocol is proactive in nature. OLSR uses only a selected number of nodes called multi point relay sectors (MPRs) to retransmit control messages, this reduces the overhead caused by flooding in other routing techniques (Jacquet et al., 2001) (Clausen and Jacquet, 2003). Another important feature of the OLSR is the use of partial link state only to select the shortest path routes to the destinations (Clausen and Jacquet, 2003). For sending control packets, the protocol does not need a reliable transmission; this means that loss of some packets during transmission is permissible (Jacquet et al., 2001). Another feature is the use of sequence numbers in each control packet so that the destination node can receive the incoming messages in any order (Jacquet et al., 2001).

Hop by hop routing is used in OLSR protocol to ensure that moving nodes can still deliver packets to another node. Hop by hop routing involves each node using its most recent information to route a packet.

C. Why AODV, DSR and OLSR?

After studying a lot of papers on Wıreless Sensor Networks, there were three routing protocols which have been discussed and analyzed by many authors. These three were mostly analyzed in WSN and MANETs, hence the decision to analyses the protocols in IoT also. With a vast amount of literature on them in MANETS and WSNs, making inferences about how they fare when implemented in IT scenarios has a base for which some decisions could be formulated.

D. Simulation

There are many simulators currently available for network simulations for ad-hoc networks, wireless sensors, IoT and many other networks alike. Some of them are open source while some are not.

After making research and reading as much literature as possible, I found out that the most popular simulators out there that can be used in this research are: NS2, NS3, Omnet++, OPNET, MATLAB, Contiki, etc. Each of these protocols will be explained briefly. In chapter three, the preferred simulator – OPNET Modeler Suite will be discussed in detail.

• OMNET++: a powerful object oriented discrete event network simulator (DES) based on the Eclipse editor/compiler. It is used primarily for simulation of Wireless Sensor Networks. Actually, OMNET++ itself is not a simulator, rather it provides a platform where frameworks and tools can be used to simulate wireless sensor networks. It is extensible, modular and makes use of C++ libraries for simulations. It is basically made up of modules – simple, compound or network; C++ is used for programming, GUI is provided and NED (Network Descriptor) files can be created and edited. There are different simulators/frameworks that can be used for simulation in OMNET, some are Castalia (primarily built for OMNET++), INET, MixiM, etc.

• NS-2 (Network Simulator – 2): a DES tool which was developed in 1989 and has since been used extensively in research in the field of networking especially in academics. It is object oriented and uses two languages: C++ and Object-Oriented Tool Command Language (OTcl) – both are bound together using TclCL, also it can be used on

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

different platforms. NS-2 provides two different outputs for simulations: text-based and graphical-based.

• NS-3 (Network Simulator – 3): it is also a DES simulator just like NS-2 and OMNET++. As the name might suggest, NS-3 is not an extension of or improvement on NS-2, it is a different simulator on its own. In NS-3 all programs are written in C++. It can also work on different platforms; the results are text based but graphical results could still be generated using third party software.

• J-Sim: also called JavaSim is a general-purpose simulator which was developed with Java, also it uses Autonomous Component Architecture (ACA) which is a component based software architecture. J-Sim is a platform independent simulator, this is because it is built with Java. It provides a script interface which allows script languages Python and Perl to run. J-Sim has a variety of packages to enable simulation of different things, these include Base package, NET, INET, J-Sim Total, and many others. J-Sim supports simulation of Wireless sensor networks by using the INET package and a stack of wireless protocols.

• Mannasim: is an open source simulator based on NS-2, it is a module for simulating wireless sensor networks. It is made up of two components: Mannasim Framework which provides modules for design, development & analysis of WSN applications and Script Generator Tool for the creation of simulation scripts.

• GloMoSim: Global Mobile Information System Simulator is a DES built using PARSEC (PARallel Simulation Environment for Complex Systems which can be used on both UNIX and Windows platforms) simulation environment. It is used to simulate the wired network and wireless systems. For simulation, GloMoSim uses a layered approach; it has two models for mobility of nodes – Random Waypoint Model and Random Drunken Model.

• NetSim: Network Simulator is also a DES simulation software which is used for simulating different kinds of networks. It allows users to customize their simulation by coding and debugging, also it provides graphical and command Line interface for simulation.

• MATLAB/Simulink: Matrix Laboratory is a high performance which is rich of features that can be used to do almost anything in image processing, signal processing, WSN simulation, etc. With the use of SIMULINK WSN can be simulated using GUI and building blocks. It is built and distributed by Mathworks Inc.

III. RESEARCH METHODOLOGY

In this research, three WSN routing protocols will be thoroughly examined using IoT scenarios and compared to see which one could be used for IoT and the circumstances or conditions under which they will work. The routing protocols which will be examined in this research are AODV, DSR and OLSR. In the next sections, a point by point clarification of the three routing protocols will be made. Simulations of the protocols will be made, using descriptions that fit the IoT scenarios and, analysis and comparisons will be made using three basic performance metrics.

In this section, the chosen simulators for this research – OPNET Modeler Suite and NS3 will be discussed. Basically, according to Nayyar and Singh (2015), there are currently at least thirty-one simulators that are used for WSN; some of the most prominent have been examined in previous sections. With the varieties of simulators available, the OPNET and NS3 were chosen to be used for this research. The current academic version – Riverbed Modeler Academic Edition 17.5 is used in this research.

A. OPNET Modeler Suite

OPNET stands for Optimized Network Engineering Tools. The OPNET Modeler Suite was produced in 1986 by OPNET technologies. Presently, it is part of the Riverbed Application and Network Performance Management Solutions. It possesses a large set of tools to enable the creation of various large network environments by providing a drag and drop simulating the environment. A free version for academic use is provided by IT Guru. Figure 4 shows a general view of the workflow of a typical simulation in OPNET.

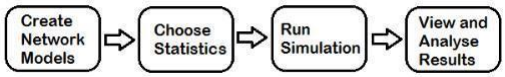

Figure 4: General Workflow of simulations in OPNET

1) Architecture of OPNET Modeler

OPNET provides a platform with a sophisticated but simple environment for making models and assessment of the performance of communication networks and distributed systems. It consists of wide variety of tools and packages, each associated with different aspects of the modelling to be done. These tools and packages can be grouped into three categories: model specification, data collection & simulation, and analysis. Figure 5 shows a general cycle of simulation of any project in OPNET. These three categories of tools also represent the three stages involved in every simulation in OPNET. It is mandatory that these stages are followed in an order which is also shown in Figure 5.

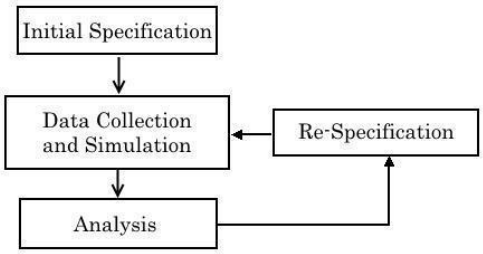

Figure 5: Project cycle in OPNET

Model specification which is the first stage in the categories is divided into two - the initial specification and re-specification. It is a stage where a representation of the system that will be simulated is developed. This is done in

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

OPNET in order to support reuse of models. In OPNET models can be developed from lower level models that have been developed prior to the simulation and stored in the model libraries. Generally, models are built using building blocks already available in the OPNET simulation environment. In OPNET model specification is done by using different editors that model different characteristics of the system. These editors are project editor, node editor, process editor, link model editor, packet format editor, interface control information (ICI) editor, Antenna Pattern editor, Modulation Curve editor and probability density function (PDF) editor. The different characteristics these editors model can be guessed from their names.

In the data collection and simulation stage, OPNET creates an executable model of the system that is being simulated; with this the performance measurement of the system can be observed. The realistic estimates of the performance of the system can be obtained, as long as the correct model of the system was initially done.

In the analysis stage which is the final stage where the results from the simulation are examined. The data is shown in output scalar and output vector files. To access the output data project editor is used, the more advanced Analysis Tool is also used. The list below shows a summary of all the editors, their functions and the OPNET products for which they are available.

• Project Editor: Specify network topology and configure nodes and links. Choose results, run simulations and view results. It is available in all OPNET products.

• Node Editor: Create models of nodes by specifying internal structure and capabilities. It is only available in the Modeler.

• Process Editor: Develop models of decision-making processes representing protocols, algorithms, resource managers, operating systems, etc. It is only available in the Modeler.

• Link Model Editor: Create, edit and view link models. It is only available in the Modeler.

• Packet Format Editor: Specify packet format, defining the order, data type and size of fields contained within the packet. It is only available in the Modeler.

• ICI Editor: Create, edit and view interface control information (ICI) formats. It is only available in the Modeler.

• Antenna Pattern Editor: Create, edit and view antenna patterns for transmitters and receivers. Also available in Modeler (Radio Version only).

• Modulation Curve Editor: Create, edit and view modulation curves for transmitters. It is available in Modeler (Radio Version only).

• PDF Editor: Create, edit and view probability density functions (PDFs). It is only available in the Modeler.

• Probe Editor: Identify sources of statistics and animation that are to be collected during a simulation. It is available in all OPNET products.

• Simulation Tool: Design and run sequences of simulations, each potentially configured with different inputs and/or outputs. It is available in all OPNET products.

• Analysis Tool: Plot and process numerical data generated by simulations. It is available in all OPNET products.

• Filter Edit: Define numerical processing that can be applied to data in analysis panels. It is only available in the Modeler.

2) Hierarchical Structure of OPNET

The OPNET Modeler has a hierarchical structure which must be utilized in order to simulate any network. The structure is made up of three basic domains that are handled using three editors, namely: network, process and node editors; they can also be called the modelling domains. These editors have been discussed in the previous section to be parts of the specification aspect of any project simulation in OPNET. The characteristics provided in these three domains depict the structures found in real network systems. The network domain is used to define the topology of the communication network that is to be simulated. The node domain is used to model the devices that make up the network. The process domain is used to create process models of the network.

3) MANET Node Models in OPNET

There are different objects in OPNET that can be used to model different parts of any network that is to be simulated. These models are available in the object panel of the OPNET simulator; the object panel is shown in Figure 6. Whatever kind of network will be simulated (MANET, WSN, etc.) these node models have been made available, their configurations can be manipulated to provide the required function in the network.

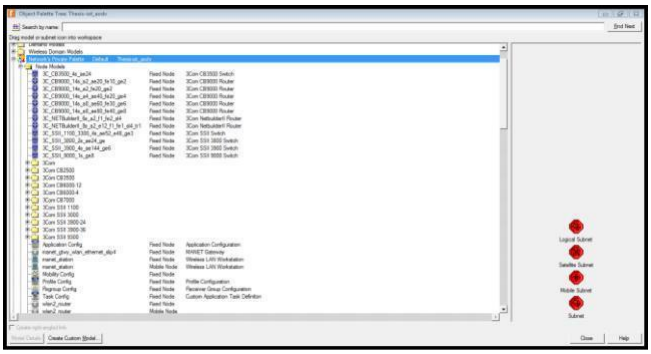

Figure 6: Object Palette in OPNET

As discussed earlier, networks are created using objects from the object palette. These objects are comprised of models which are important for performance evaluation of the routing protocols used during simulation. There are numerous models available but some of the most popular ones are given below.

• Wireless LAN Routers and MANET Gateway

• MANET Station

• Wireless LAN Workstations and Servers

• Application Configuration

• Profile Configuration

• Mobility Configuration

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

• Rx Group Configuration

4) Routing Protocol Configurations

Routing protocol configuration in OPNET I simply specifying how the routing protocol to be implemented should work during simulation. OPNET provides many parameters that can be configured in order for a used protocol to act in a particular way. Figure 7 shows routing protocol configuration for OLSR.

B. Simulation of AODV using OPNET

The first thing to do in a simulation as already discussed in the previous sections is modelling of the network or system to be simulated. This is simply defining the properties of the network. The size of the simulation environment should be known, speed of nodes, type of nodes, mobility model of the nodes and other parameters must be defined. The complete steps involved are given below.

In OPNET the first step after opening the software is to click File, then New to create a new project. Enter a project name (I used project name as Thesis). Also set the name of the scenario – using a meaningful name is important.

• After setting project name and scenario name, choose the Create empty scenario option and click next.

• The next step is to choose the scale of the network; in this case Campus option is used.

• Next is to search for the model family included in the network, then click on No to change it to Yes in order for that model family to be included in the network -for this project, select MANET.

• The final step for project and scenario creation is to click finish.

• The next step is to choose all objects that will be used to represent nodes and other entities in the network from the object palette. Fixed and mobile nodes are required, so, select MANET station from the object palette and place them on the network environment. They can be duplicated as much as needed. Also select mobility configuration object to set mobility model, profile definition and application configuration object to generate traffic in the network.

• Next, is to define application configuration settings – right click on the application config object and edit attributes. On the application definition tab, add the number of applications to be used by setting the number of rows. In this project number of rows is set as 1 for a single email application.

• Next is to set application name (Email) and select application type (email); set the traffic level – low, medium, high or very high.

• Defining profile configuration settings is the next step to take. Right click on the profile config object. Add the number of rows which represents number of applications used to generate traffic, in this case 1. Define other parameters such as start time (100) and end time (end of simulation).

• Next step is defining the node settings. Select all the nodes and all other objects in the network. Go to protocols menu, select IP, then Addressing and then Auto-Assign IPv4 Addresses. Now each object has an IP address.

• The next step which is probably the most significant is choosing the routing protocol. Select all nodes and other objects again. Right click on one of the nodes, click Edit Attributes. Select Ad-Hoc Parameters, for Ad-Hoc Routing Protocol select AODV.

• Next step is to set the mobility model of the nodes. Right click the mobile config object and edit the Random mobility profiles.

• Last step in the simulation is to set the DES metrics for performance evaluation of network. Right Click on the simulation environment and select the Choose Individual DES statistics option. Select all statistics that need to be checked from the different categories and click OK. The next is to click run and wait for simulation results.

C. Simulation of OLSR using OPNET

Basically, the steps involved here are same with those in the AODV simulation. The difference is the step where Ad-Hoc Routing protocol is chosen, select OLSR instead of AODV and that is it. Also, in the selection of DES statistics for AODV,AODV statistics was selected but here OLSR will be selected.

D. Simulation of DSR using OPNET

As already explained in simulations of AODV and OLSR, the steps are the same with the others. Instead of choosing AODV or OLSR routing protocol and statistics, DSR will be chosen.

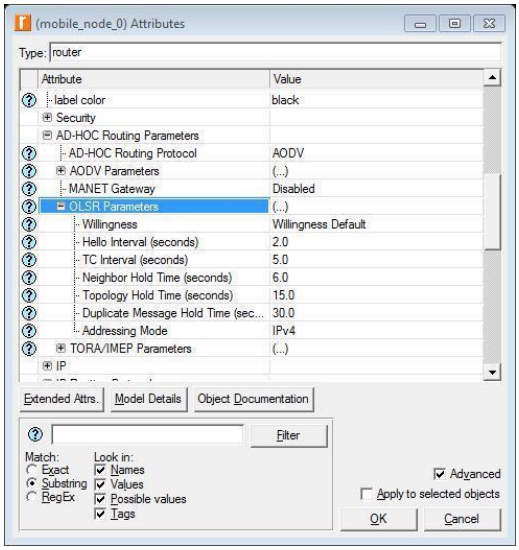

Figure 7: Routing protocol configuration for OLSR

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

E. Network Simulator – 3 (NS3)

In the literature review, a brief description of NS3 was given. It was also stated that NS3 is a discrete event network simulator (DES). NS3 can be run on different platforms – UNIX based such as Ubuntu and Linux, and also on Windows using Cygwin. For this simulation, NS3 was run in Ubuntu (which was hosted in a virtual machine). The main language in the NS3 environment is C++. There are different versions of NS3, the current version (which was used in this research) is NS-3.27. There are basically three steps involved in simulation with NS3, these steps are: installation of Ubuntu on the host machine, building the application for simulation, and the simulation itself. After the simulation, data analysis can be done using different methods, data can be plotted on graphs using GNUPLOT. The installation guide for all platforms can be in "Installation - Nsnam", (2018). The other steps – building and running simulations in NS3 (including installation guides) can be accessed via " ns-3 Tutorial
— Tutorial", (2018).

F. Simulation of AODV, DSR and OLSR using NS3

In NS3, different classes have been made available for the simulation of different routing protocols. For the purpose of comparison of routing protocols, the NS3 team provided a class called manet-routing-compare.cc which can be accessed via "ns-3: examples/routing/manet-routing-compare.cc Source File", (2018). The class has some default values which can be tweaked depending on the specifications of the simulation to be done.

For the simulation in NS3, some parameters are slightly different from the ones used in OPNET. This is as a result of some differences between the two simulations and also some limitations. The steps involved in simulating the routing protocols in NS3 are the same as building any other application, which is given in "Installation - Nsnam", (2018).

After running the simulations, the results were saved in .csv files and plots were made for the results using GNUPLOT.

IV. DATA ANALYSIS AND DISCUSSION OF FINDINGS

As already discussed, the three routing protocols – AODV, DSR and OLSR were simulated using OPNET and NS3. The average values for the parameters for all the different scenarios have been computed and will be discussed in this section.

A. Performance Metrics

The performance metrics that will be used to evaluate and compare the routing protocols are routing overhead, average end to end delay and throughput. All these can be extracted from the statistics in OPNET. The level of mobility of nodes and parameters for simulation will also be discussed in this chapter. In the simulations number of nodes that were used are 20, 40, 60 and 70. In the NS3 simulation, 20, 60 and 100 nodes were simulated.

1) Routing Overhead

To determine the degree of congestion of a network and also, the power at the nodes, routing overhead is used. During transmission of data packets in the network, there may be some form of a collision between the data packets and route packets; a number of route packets which every data package needed to allow on average is referred to as the routing overhead. It is measured in bits per second - bits/s or bps. Since the routing overhead is a measure of packets sent into the network as a result of routing, it is measured and obtained by getting routing packets sent from the DES statistics of the routing protocol; it is plotted against the time interval.

2) Average End to End Delay

This is an accumulation of all possible delays caused inside the network including those caused during route discovery, queuing, and retransmission at the MAC, propagation and transfer time. The average end to end delay facilitates the understanding of the delay caused by path discovery. Since it is a time quantity, it is measured in seconds. Average end to end delay is also obtained from DES statistics. It is plotted in seconds against the time intervals.

3) Throughput

Measured in bits per seconds – bps or bits/s, it is the proportion of packets received by the receiver node among those sent by the source node. The throughput statistics of the network are obtained using the DES individual statistics. The throughput is measured (bps) and plotted against the time intervals by the DES Simulator.

B. Mobility and Distribution of Nodes

In this research, the nodes used in the network are both mobile and fixed. Using the objects available in the object palette, the nodes were added to the network. As already mentioned, the number of nodes will be 10, 20 and 30. The number of mobile nodes in these scenarios will be 20, 30 and 40 percent. These nodes are moving with a speed of at most 2m/s.

In this simulation setup, the nodes, as already mentioned are both mobile and fixed – this means that at the end of the simulation, it is expected that the fixed nodes maintain their positions while the mobile nodes' final position is most likely not going to be the same with the initial. Additionally, all nodes – both fixed and mobile can communicate with each other, as no restriction is placed on them. The mobile nodes are configured to move in a random pattern within the confines of the simulation environment (1000m X 1000m). Whenever any node reaches the boundaries of the environment, the simulator ensures that it bounces back in another direction inwards.

C. Parameters for OPNET Simulation

A typical IoT scenario will be used to simulate the three routing protocols and then, they will be evaluated and compared using the performance metrics already discussed. These parameters are given below:

- Size: 1000m X 1000m (Campus Network)
- Speed of mobile nodes: 10 m/s
- Model of movement: random waypoint model
- Simulation time: 30 minutes
- Traffic type: FTP (medium load)
- Data rate: 19.5 Mbps (base) / 180 Mbps (max)
- Physical characteristics: HT PHY 2.4GHz (802.11n)

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

- Pause time: 5 seconds
- Two sets of scenarios – one with all nodes fixed and the other with the percentage of mobile nodes.

D. Simulation Parameters for NS3 Simulation

The simulation parameters are not all the same for NS3 as was used in the OPNET simulation, the parameters are given below:

- Size: 1000m X 1000m
- Speed of mobile nodes: 10 m/s
- Model of movement: random waypoint model
- Physical characteristics: WIFI_PHY_STANDARD_80211b
- Simulation time: 300 seconds
- Data Rate: 1024kbps
- Pause time: 5 seconds

E. OPNET Simulation Results and Discussion

The average values of results of the simulations of all the scenarios considering 20, 60 and 70 nodes, with number of mobile nodes 10, 15 and 20 are shown in Tables 1- 12.

Table 1: Routing overheads for 20 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	28200	7600	80.7
15	25400	5800	1212
20	25400	5490	80.7

Table 2: Routing overheads for 40 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	65000	12500	80.3
15	68000	13000	80.3
20	66000	12200	80.3

Table 3: Routing overheads for 60 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	78000	19000	80.2
15	75000	18900	80.2
20	83000	24000	80.2

Table 4: Routing overheads for 70 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	79000	142000	79.72
15	73000	252000	79.92
20	82000	328000	79.92

Table 5: Average end to end delay for 20 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	0.008	0.06	0.000077
15	0.008	0.0045	0.000087
20	0.0063	0.0031	0.000077

Table 6: Average end to end delay for 40 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	0.094	0.275	0.000077
15	0.065	0.2	0.000077
20	0.06	0.043	0.000077

Table 7: Average end to end delay for 60 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	0.06	0.125	0.000077
15	0.039	0.055	0.000077
20	0.0375	0.19	0.000077

Table 8: Average end to end delay for 70 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	0.052	0.091	000077
15	0.053	0.071	000077
20	0.048	0.059	000077

Table 9: Throughput for 20 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	192000	29000	3090
15	180000	14300	36180
20	146000	17200	3120

Table 10:Throughput for 40 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	620000	42000	3120
15	559000	44000	3118
20	600000	39000	3136

Table 11: Throughput for 60 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	840000	94000	3132
15	849000	93000	3140
20	880000	80000	3140

Table 12: Routing overheads for 20 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	940000	122000	3110
15	960000	119000	3120
20	980000	95000	3120

After getting the results in Tables 1 – 12, the average values for the three parameters considering number of nodes and number of mobile nodes were obtained. Graphs were plotted using these averages to show how routing overhead, average end to end delay and number of mobile nodes. In the next sections, these results will be discussed.

Table 13: Routing overheads for 20 nodes

Number of mobile nodes	AODV	DSR	OLSR
10	940000	122000	3110
15	960000	119000	3120
20	980000	95000	3120

Table 13 and Table 14 show the average results of the routing overhead obtained while considering number of nodes and the number of mobile nodes. The routing

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

overhead was obtained using the Routing Traffic Sent (bits/sec) from the Global Statistics of the network in OPNET.

Table 14: Routing overheads for number if nodes

Number of mobile nodes	AODV	DSR	OLSR
20	15766.67	19800	459.2
40	66333.33	12566.67	80.2
60	78666.67	20633.33	80.2
70	78000	24066.67	79.73

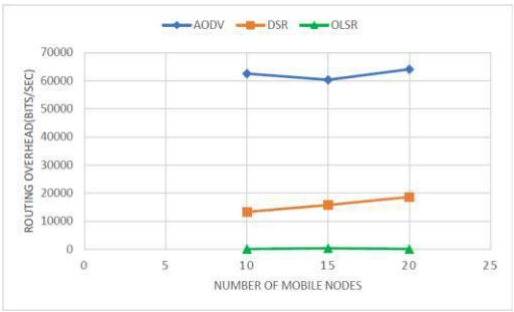

Figure 8: Showing Routing Overhead for number of mobile nodes

As seen in Figure 8, the routing overhead for AODV, DSR and OLSR are relatively constant as the number of nodes is increased (but with an increase for AODV when 20 mobile nodes). This is seen in Xin and Yang (2015) where the routing overhead for the three protocols remained unchanged as the percentage of mobile nodes was increased. It can also be seen that, AODV has the highest routing overhead while OLSR has the lowest.

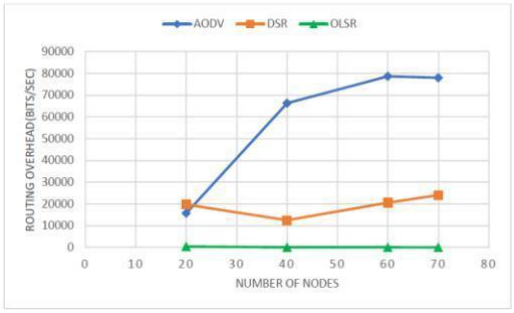

Figure 9: Showing Routing Overhead for number of nodes:

As seen in Figure 9, the routing overhead for AODV and DSR increases as the nodes are increased, while OLSR decreases very slightly. Also it can be seen that AODV is still the highest with a large margin when compared to the OLSR and DSR. OLSR has the lowest.

Table 15: Average End to End Delay for number of mobile nodes

Number of mobile nodes	AODV	DSR	OLSR
10	61733.33	13600	80.21
15	57800	16633.33	80.21
20	190400	20763.33	80.22

Table 15 and Table 16 show the average results of the average end to end delay obtained while considering number of nodes and the number of mobile nodes. The Average End to End Delay was obtained using the Wireless LAN Delay (sec) from the Global Statistics of the network in OPNET.

Table 16: Average End to End delay for number of nodes

Number of mobile nodes	AODV	DSR	OLSR
20	15766.67	19800	459.2
40	66333.33	12566.67	80.2
60	78666.67	20633.33	80.2
70	78000	24066.67	79.73

Figure 10: Showing Average End to End Delay for number of mobile nodes

From Figure 10, it can be seen that DSR has the highest average end to end delay as the number of mobile nodes is increased; it can also be seen that the average end to end delay is not very sensitive to the change in number of mobile nodes for OLSR. OLSR has the lowest end to end delay.

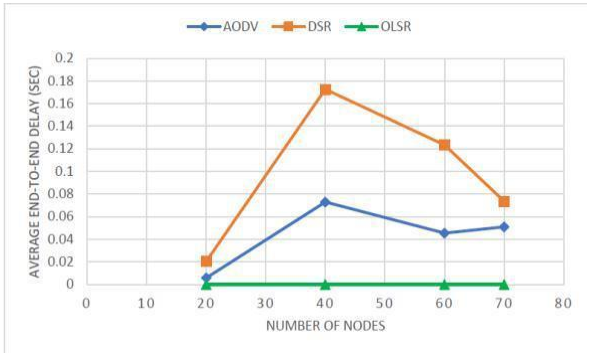

Figure 11: Average End to End Delay for number of nodes

Figure 11 shows that after the number of nodes is increased from 10 to 20, the Average end to end delay for AODV and DSR increases with a great margin and keeps on increasing (though with a slight decrease for DSR when 70 nodes are used) while that of OLSR remains constant. It can be seen also that

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

DSR has the highest average end to end delay and OLSR the lowest.

Table 17: Throughput for number of mobile nodes

Number of mobile nodes	AODV	DSR	OLSR
10	61733.33	13600	80.21
15	57800	16633.33	80.21
20	190400	20763.33	80.22

Table 17 and Table 18 show the average results of the throughput obtained while considering number of nodes and the number of mobile nodes. The Average End to End Delay was obtained using the Wireless LAN Throughput (bits/sec) from the Global Statistics of the network in OPNET.

Figure 12: Showing Throughput for varying number of mobile nodes

From Figure 12, the throughput for AODV, DSR and OLSR remain fairly constant as number of nodes is increased. AODV can be seen to have the best throughput among the three and OLSR the lowest.

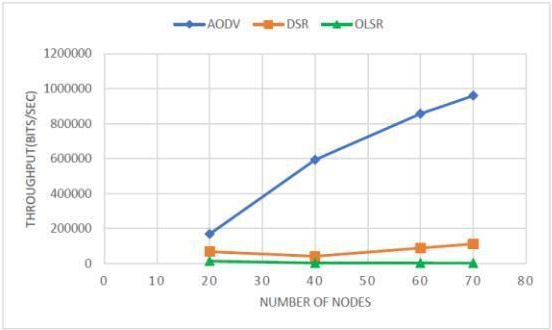

Figure 13: Throughput for number of nodes

In Figure 13, AODV has the highest throughput while OLSR has the lowest. In Xin and Yang (2015) the throughput of AODV as the number of nodes is increased was found to be the highest.

As seen from the discussions, AODV has the highest average end to end delay as the number of mobile nodes is increased and also when the nodes are being increased; OLSR was found to have the lowest. With this information, OLSR will be more suited for rapid communication among the three routing protocols. Being reactive protocols, it is not a coincidence that AODV and DSR have more end to end delay than OLSR. With the link and route information determined from the start of the network and frequent checks of link states and updates of the broken links, there is no need to seek routes every time there is a need to send packages; with these, OLSR has lesser average end to end delay than the other two. According to the discussions, DSR had the highest routing overhead among the three, with OLSR the lowest. This is an indication that when energy conservation is a priority in communication, OLSR is most suitable among the three. Being proactive, OLSR routing overhead which is the amount of packets sent throughout the network as a result of routing is expected to be greater than the reactive protocols. According to ("OptimizedLinkStateRoutingProtocol", 2017), despite routing overhead being greater in proactive protocols, it becomes an advantage in the long run because it does not increase as the routes inside the network increase. Also according to ("OptimizedLinkStateRoutingProtocol", 2017) routing overhead usually requires a large amount of bandwidth but is not a problem when just a few hundreds of nodes are used. As is the case with the network in this research, maximum number of nodes used is 70 which mean OLSR will have an advantage over AODV and DSR. The advantage is more if the amount of break in links can be reduced to the lowest possible amount, this means amount of packets sent during the frequent checks of the link states will be minimalized.

According to the discussions in this Section, AODV had the highest throughput. This is an indication that AODV should be considered when quality of communication is a priority in communication because it had the highest throughput both when number of nodes was increased and when the number of mobile nodes was increased.

F. Discussion of Results obtained from NS3

In the NS3 simulation, as stated before, the numbers of nodes simulated were 20, 60 and 100. Throughput was the only performance metric that was used to compare the routing protocols. This is because the calculation of statistics relating to the other performance metrics – routing overhead and delay were removed from NS3 starting from NS-3.20. The evidence of this can be seen in the code which is given in "ns-3: examples/routing/manet-routing-compare.cc Source File", (2018), the only function pertaining the performance metrics is throughput() – which is used to calculate the throughput of the whole network. The results of the simulation are shown in Figure 4.34 to Figure 4.36.

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 8, ISSN No. 2455-2143, Pages 378-393
Published Online December 2019 in IJEAST (http://www.ijeast.com)

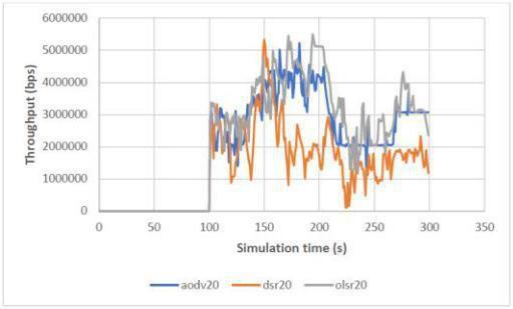

Figure 14: Throughput for 20 nodes – Using NS3

The Simulation results for throughput of 20 nodes for AODV, DSR and OLSR are shown in Figure 14.

Figure 15: Throughput for 60 nodes – Using NS3

The Simulation results for throughput of 60 nodes for AODV, DSR and OLSR are shown in Figure 9 and the results for 100 nodes are shown in Figure 15.

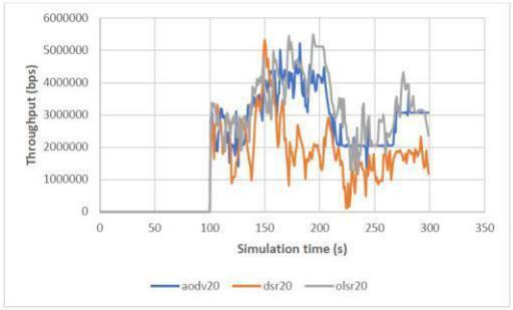

Figure 16: Throughput for 100 nodes – Using NS3

From the simulation results which were shown in Figure 14 to Figure 16, the average throughput were computed, and they are given in Table 19. Figure 16 shows the plot of the values given in Table 19.

Table 19: Average Throughput from NS3

Number of mobile nodes	**AODV**	**DSR**	**OLSR**
20	1979760.6	324236.9	2207197.97
60	344227.79	325058.59	566531.43
100	760681.87	475518.29	1257198.98

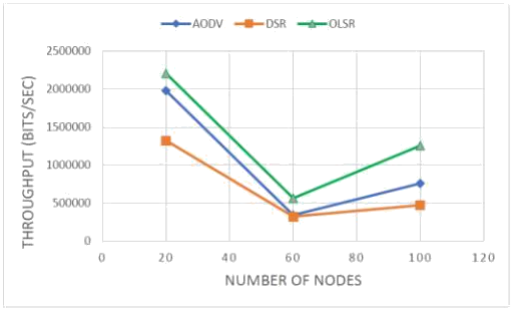

Figure 17: NS3 Simulation Results for Average Throughput

As seen in Figure 17, the throughput reduced as the number of node was increased from 20 to 60 for all the routing protocols. Also, it can be seen that as the number of nodes increased from 60 to 100 the throughput increased but not as high as it was when number of nodes was 20. The complex operations involved and routing overhead increase as the number of nodes is increased, thereby leading to a decrease in the expected throughput of the network. Even though there was an increase in the throughput when number of nodes increased from 60 to 100, it is unlikely that the throughput gets to the amount when number of nodes is 20 because of the presence of routing overhead in the network.

V. CONCLUSION

In this research, much study was done on routing in the Internet of Things. Problems in routing were analyzed and various routing protocols were analyzed as well. The goal of the thesis, which was to analyze, implement and compare three wireless sensor networks in the IoT was reached. OPNET simulator and NS3 were used to simulate and analyze three MANET routing protocols. Three parameters were used to analyze the protocols – routing overhead, average end to end delay and throughput, also different percentages of mobile nodes were used to analyze the routing protocols in a 1000m x 1000m environment. The simulator time for OPNET was 30 minutes and the simulation time for NS3 was 300 seconds. After the simulations and performance analysis and comparisons were done, it can be seen that, no particular routing protocol is suitable for all cases, rather different protocols are appropriate for different scenarios.

According to the results that were obtained using OPNET, AODV appeared to have the highest average end to end delay and the highest routing overhead. With this information, it was concluded that out of the three protocols, if rapid communication then OLSR should be considered since it has the lowest end to end delay; if energy conservation is the major priority, then OLSR should be considered again because it has the lowest routing overhead. Also in the study, it was found that OLSR has the lowest throughput among the three, and AODV the highest; it can therefore be concluded that AODV should be considered when quality of communication is the most important quality.

From the results, it was also shown that as number of nodes is increased, the routing overhead also increase which in turn reduces or affects the throughput of the network. This is an indication that regardless of the routing protocol used, the routing overhead must be minimized in order to increase or improve the throughput.

A. Recommendation

Despite the fact that AODV has the is not the best performer in terms of routing overhead and end to end delay among the three, it should be used for IoT applications in the nearest future because of its high throughput.

B. Future Work

In this study, mobility of nodes was the only considering factor used to analyse and compare the three protocols. In future studies like this, other factors such as variability in speed of nodes, location information of nodes and change in topology of the network can be used to analyse the performance of these routing protocols. Increasing the number of nodes in future works will be able to provide a more generic and suitable answer to the extent to which these wireless sensor networks routing protocols can be applied in the IoT.

ACKNOWLEDGMENT

All praise and thanksgiving to the Lord Almighty who made everything possible. Special thanks also to family and colleagues who have helped in this research one way or another. God bless you all.

VI. REFERENCES

[1] Al-Karaki J. N., and Kamal A. E., (2004). "Routing techniques in wireless sensor networks: a survey," in IEEE Wireless Communications, vol. 11, no. 6. doi: 10.1109/MWC.2004.1368893, (pp. 6-28).

[2] Jeba, N., & Kamala, V. (2016). A Survey on Routing Protocols for Internet of Things. power, 3(5).

[3] Iova O., Picco P., Istomin T. and Kiraly C., (2016). "RPL: The Routing Standard for the Internet of Things... Or Is It?", IEEE Communications Magazine, vol. 54, no. 12, (pp. 16-22).

[4] Richardson M. and Robles I., (1994). RPL- Routing over Low Power and Lossy Networks.

[5] (2017) . "List of ad hoc routing protocols", En.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/List_of_ad_hoc_routing_protocols#On-demand_.28reactive.29_routing. [Accessed: 19- Aug- 2017].

[6] Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad hoc on-demand distance vector (AODV) routing (No. RFC 3561).

[7] Marina, M. K.., and Samir R. D. (2001). "On-demand multi path distance vector routing in ad hoc networks." Network Protocols. Ninth International Conference on. IEEE, 2001.

[9] Johnson, D. B., Maltz, D. A., & Broch, J. (2001). DSR: The dynamic source routing protocol for multihop wireless ad hoc networks. Ad hoc networking, 5, 139-172.

[10] Misra, R., & Mandal, C. R. (2005). Performance comparison of AODV/DSR on-demand routing protocols for ad hoc networks in constrained situation. In Personal Wireless Communications, 2005. ICPWC 2005. 2005 IEEE International Conference on (pp. 86-89). IEEE.

[11] Jacquet, P. M., Clausen T., Laouiti A., Qayyum A. and Viennot L., (2001). "Optimized link state routing protocol for ad hoc networks," Proceedings. IEEE International Multi Topic Conference. IEEE INMIC 2001. Technology for the 21st Century, 2001, pp. 62-68. doi: 10.1109/INMIC.2001.995315

[12] Clausen, T., & Jacquet, P. (2003). Optimized link state routing protocol (OLSR) (No. RFC 3626).0+

[13] Salman, T., & Jain, R. (2015). Networking Protocols and Standards for Internet of Things. Internet of Things and Data Analytics Handbook, (pp. 215-238).

[14] Thomas, R. W., DaSilva, L. A., & MacKenzie, A. B. (2005). Cognitive networks. In New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on (pp. 352-360). Ieee.

[15] Aijaz, A., & Aghvami, A. H. (2015). Cognitive machine-to-machine communications for Internet-of-Things: A protocol stack perspective. IEEE Internet of Things Journal, 2(2), (pp. 103-112).

[16] Alaharı H, P., and Hema, T. B., (2017). Network Routıng Protocols in Iot. In: Proceedings of International Conference on Science, Technology, Engineering and Management, Guntur, India, 07th - 08th April 2017, ISBN: 978-93-86291-63-9

[17] Colesanti, U., & Santini, S. (2011). The collection tree protocol for the Castalia wireless sensor networks simulator. Department of Computer Science, ETH Zurich, Tech. Rep, 729.

[18] Colesanti, U., and Silvia S., (2010). A performance evaluation of the collection tree protocol based on its implementation for the Castalia wireless sensor networks simulator. ETH, Department of Computer Science.

[19] Basagni, S., Petrioli, C., Petroccia, R., & Spaccini, D. (2012). Channel-aware routing for underwater wireless networks. In OCEANS, 2012-Yeosu (pp. 1-9). IEEE.

[19] Nayyar, A., & Singh, R. (2015). A comprehensive review of simulation tools for wireless sensor networks (WSNs). Journal of Wireless Networking and Communications, 5(1), (pp. 19-47).

[20] Installation - Nsnam. (2018). Nsnam.org. Retrieved 22 January 2018, from https://www.nsnam.org/wiki/Installation

[21] ns-3 Tutorial — Tutorial. (2018). Nsnam.org. Retrieved 22 January 2018, from https://www.nsnam.org/docs/tutorial/html/index.html

[22] ns-3: examples/routing/manet-routing-compare.cc Source File. (2018). Nsnam.org. Retrieved 22 January 2018, from https://www.nsnam.org/doxygen/manet-routing-compare_8cc_source.html

[23] H. M. Xin and K. Yang, (2015). "Routing Protocols Analysis for Internet of Things," 2nd International Conference on Information Science and Control Engineering, Shanghai, 2015, pp. 447-450. doi: 10.1109/ICISCE.2015.104 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber =7120644&isnumber=7120439

[24] "OptimizedLinkStateRoutingProtocol", (2017). En.wikipedia.org. [Online]. Available: https://en.wikipedia.org/wiki/Optimized_Link_State_Rou ting_Protocol. [Accessed: 12- Oct- 2017].

[25] Aijaz, A., Su, H., & Aghvami, A. H. (2015). CORPL: A routing protocol for cognitive radio enabled AMI networks. IEEE Transactions on Smart Grid, 6(1), (pp. 477-485).

[26] Akkaya, K., & Younis, M. (2005). A survey on routing protocols for wireless sensor networks. Ad hoc networks, 3(3), (pp. 325-349).

[27] Dulman, S., Nieberg, T., Wu, J., & Havinga, P. (2003). Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. In Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE (Vol. 3, pp. 1918-1922). IEEE.