Analysis of simulators for designing network architecture in Wireless Sensor Network applications.

Author: Mr. Garvit Periwal, John Deere
Co-Author: Mr. Swarupanand Sewalkar, John Deere and Mr. Prashant Koparde, John Deere

Abstract

Recent advancements in energy efficient wireless communication protocols and low powered digital sensor technologies have led to the development of wireless sensor network (WSN) applications in diverse industries. These WSNs are generally designed using Bluetooth Low Energy (BLE), ZigBee and Wi-Fi communication protocol depending on the range and reliability requirements of the application. Designing of these WSN applications also depends on the following factors. First, the environment under which devices operate varies with the industries and products they are employed on. Second, the energy availability for these devices is limited so higher signal strength for transmission and retransmission reduces the lifetime of these nodes significantly and finally, the size of networks is increasing hence scheduling and routing of messages becomes critical as well. These factors make simulation for these applications essential for evaluating the performance of WSNs before physical deployment of network in a cost-effective manner. Since there are several WSN simulators with different functionalities, this paper aims to deliver a comparative study of the features of most widely used simulators. This study includes evaluation of these simulators based on communication protocols supported, visualizations for the network, calculation of network performance parameters like packet delivery ratio, throughput, energy consumption, and support for the channel modelling. The uniqueness of this study arises from its focus on the BLE network simulations since it is increasingly become a protocol of choice for WSN application in automotive domain. Also, we consider both open source and closed source simulators. This study also becomes necessary as many of the simulators discussed in the existing studies have become obsolete and the classification parameters don't consider the wireless communication protocols supported by them. Through this analysis of simulators, the best choice of simulator for an application can be made without investing too much time in exploring all the options.

Introduction

IOT and WSN are two technologies at the forefront of the changing technological landscape in the fields of automation and autonomy. The application of these two technologies in tandem are changing the landscape of industries and these technologies are starting to get employed in both on road and off-road vehicles as well. These technologies can be employed in the autonomous vehicles where they help them to understand their surroundings. Wireless sensor networks (WSN) are a group of devices employed in a region of interest to collect the information about environment. IOT technology is used to transfer the collected information by these WSN to web and perform analysis of these data for better decision-making.

Wireless communication helps in removing the excessive wiring used for establishing communications in the sensor networks. Excessive wiring has an impact on the overall efficiency of the system as well as in factories applications this can lead to increased cost of wiring as the area which needs to covered will be in range of kilometers. In case of off-road vehicle application any breakage in wired system can lead to increased downtime.

An important parameter for these WSN is the energy consumed by the sensor node since they are mostly battery operated. A large part of the energy is consumed for the wireless communication so making use of power efficient communication protocols becomes a necessity. Along with the low energy highly reliable communication is another necessity in these networks. There are multiple wireless communication protocols like BLE, Zigbee, Wi-Fi available for setting up communication in these networks. BLE has recently been gaining high adoption in applications for WSN and there are multiple ways for setting up communication between devices in this protocol like connectable and non-connectable mode, scannable and non-scannable mode and directed and undirected mode. Earlier the preferred choice used to be ZigBee because of its low energy consumption compared to Wi-Fi. Simulations can help in making the informative decision for the choice of protocol in these applications according to the system variables.

There are multiple simulators available as well for simulation of WSN networks and multiple studies have been conducted in past on this topic just because of the presence of a large number of open source simulators. Ivan Minakov in [1] presented a comparative study of simulators and divided the simulators in three categories named generic network simulators, network-oriented simulators and sensor node-oriented simulators. The study considered all the open-source simulators present at that time like MiXiM (Mixed Simulator), Castalia, NS-2 (Network Simulator – 2), PASES (Power Aware Simulator for Embedded Systems), TOSSIM, COOJA, MSPSiM and many more. Most of these simulators are used to create IEEE 802.15.4 based low-rate wireless personal area network. The first version of Bluetooth Low Energy (BLE), BLE v4.0, was introduced in year 2009 so none of the simulators mentioned could be readily used for BLE based network simulation. Although few of these simulators do support simulations for WI-Fi based systems. In [2] simulators for BLE like MATLAB Bluetooth toolbox, BLE simulator of Mikhaylov and a BLE peripheral simulator based on android are discussed. The discussion about MATLAB Bluetooth toolbox done in the paper is limited to its functionality introduced in the first version of the toolbox where the functionalities were limited to interaction with the nearby BLE devices using the host BLE device. Lot of important features have been added to the toolbox since its first version making it suitable for advanced use cases as well. Other simulators discussed in the paper were developed only for individual research purposes and have not been updated to incorporate the changes in BLE stack since. Few other studies have been conducted in past like simulation tools for WSN applications under water in [3], survey and comparison of simulators in [4] and similar comparison of simulators to form a guideline on

which simulator to use in particular situation was done in [5]. The comparison parameters considered in these papers are simulation engine, scalability, coding language used, routing, energy profiling, accuracy etc. but none of them talks about the supported wireless communication protocols in them and the efforts required for building a simulation model in them for a particular type of wireless communication technology. Also, many of the simulators discussed in these papers are either discontinued or support for them is no longer available.

In this paper these simulators are discussed keeping the communication protocol at center as now there are multiple ways in which in these applications can be built and discuss which simulator is best for particular type of communication protocol. Detailed analysis of NS-3, MATLAB Bluetooth toolbox and COOJA simulator is performed in this paper.

In next section we will first discuss the need for simulators and general parameters used for evaluation of a wireless network. Then we will deep dive into NS-3, COOJA and MATLAB Bluetooth Toolbox and in end we will summarize and compare the simulators.

WSN Simulators

The need for simulations arises from the fact that the size of the WSN in terms of number of devices in the system is increasing and these systems are becoming more complex in nature. A direct deployment of such a large network could prove costly and might not be able to produce the most optimal solution for the problem needed to be rectified. Also, most of the devices are only designed to support a single communication protocol only. If the communication protocol in these devices doesn't meet the requirements and needs to be changed at a later stage these devices would become redundant and lead to increased overall project cost.

To give a high overview of any wireless communication system the modelling of these systems revolves around the three layers that is physical layer which controls the radio of the device, and it is responsible for actual transmission and reception of the data packets. The link layer is responsible for assigning the role to device's radio i.e., whether it is going to transmit the packets or receive them. Then comes the higher-level layer called as the application layer, this is where the users interact with the device and manage the connection between device, format for sharing of data and all the security related aspects.

To evaluate the performance of any wireless network few of the commonly used parameters include –

1. Packet Delivery Ratio (PDR) – It is the ratio of total number of packets sent by the device to the number of packets successfully received at the observer. This is used as an indicator for reliability in the wireless communication.

2. Latency – It is the time taken for transmission of packet from the source to the destination node. Generally, latency will be higher in mesh networks when compared to star type due to presence of relay nodes in the system.

3. Throughput – It refers to the amount of data received at the receiver in a particular span of time. Bandwidth refers to the maximum achievable link speed whereas the true value in the system is represented by throughput.

4. Energy Consumption – This is another important factor as these devices are generally energy constrained.

5. Packet Error Rate – This factor is used to represent the number of packets received incorrectly at the PHY layer of device. One of the reasons for this is the packet collisions happening in the system.

All the simulators discussed in the paper have functionalities to calculate these values for the system.

Network Simulator- 3 (NS - 3)

NS - 3 is one of the most widely used open-source software for doing simulations of internet-based systems. The framework of the tool is written in C++ and python. It is a discrete event simulator which basically means the operations of the system are modelled as a sequence of events in time. These events occur at a particular instant in time and the state of the system at that instant is recorded. NS-3 provides a platform to simulate multiple types of communication network system like wired, wireless and satellite based. NS-3 has a well-maintained stack for Wi-Fi and supports wired communication simulations like ethernet based system as well. The platform is very comprehensive in nature because of the support for multiple protocols inside each layer of stack for a particular type of wireless communication system. Also, it is flexible and customizable in nature because of its open-source nature. The source codes are freely available for researcher to modify and test for multiple scenarios. NS-3 allows for modelling realistic network behaviors by allowing to model a system for packet loss, interferences, and mobility in the system. The extensive support for logging and tracing the events in system allows the researchers to analyze the network performance in depth and helps in debugging any unusual behavior in the system.

NS-3 is an updated version of NS-2 and the project was started in the year 2006. The major difference between these simulators is the scripting language used in them. Programs in NS-2 are scripted in object-oriented TCL (OTcl) and these simulations cannot be run purely using C++. The components in NS-2 were also written as a mix of C++ and OTcl. In NS-3 all the components are written in C++ and the python bindings are also provided as an option hence the simulations can be built using both C++ and python language. The support for python has made this simulator more accessible to researchers because of the ease of scripting in python however, the most widely used scripting language of NS-3 still remains C++. Also, NS-3 is not backward compatible to NS-2 simulator. NS-3 is an actively maintained simulator whereas support and development for NS-2 stopped once the NS-3 was launched. Both NS-2 and NS-3 is primarily used on Linux or macOS and the windows user can use it with the help of Linux virtual machines.

Building wireless system in NS-3

NS-3 has a modular architecture and is built as a system of libraries that has support working with each other. To build a system in NS-3 a user can make use of components available in these libraries and establish a link between these components to scale up the system.

The basic element of a system in any communication system is a node. This term originates from graph theory and basically represents a computing device in the system. This is a high-level representation of a computing device in a system and all the properties like application, communication channel and protocol stack associated with the type of communication system will be added to these nodes. For example, In a wired system to get a ethernet like functionality than this node will be assigned a carrier sense multiple access (CSMA) communication channel model, a CSMA network device manager to establish

communication with other nodes and a topology helper to configure the node with basic communication properties like data rate, delay and device address etc.

For building a wireless network system in NS-3 using Wi-Fi as communication protocol, first the nodes are added in the system and a container is created to hold these devices. In the next step assignment of the interconnection channel is done to these devices, which in this case will be the channels in the unlicensed 2.4 GHz spectrum. Once these channels are associated with the device the basic modelling for the PHY layer of the device is complete. Other configurations like which channels to make use off, receiver sensitivity for the device, transmission signal strength etc. can also be added to the PHY layer.

```
At time +2s client sent 2048 bytes to 172.11.2.4 port 3456
At time +2.02017s server received 2048 bytes from 172.11.3.3 port 49153
At time +2.02017s server sent 2048 bytes to 172.11.3.3 port 49153
At time +2.03819s client received 2048 bytes from 172.11.2.4 port 3456
At time +3s client sent 2048 bytes to 172.11.2.4 port 3456
At time +3.00829s server received 2048 bytes from 172.11.3.3 port 49153
At time +3.00829s server sent 2048 bytes to 172.11.3.3 port 49153
At time +3.01596s client received 2048 bytes from 172.11.2.4 port 3456
```

Figure 1. Command line output format

NS-3 has multiple models of 802.11 for an accurate implementation of MAC layer. To configure MAC layer, assignment of a service set identifier (SSID) is made. This SSID ensures that the Wi-Fi network has a unique name, and the Wi-Fi helper object will configure the device to 802.11ax standard which is also known as the Wi-Fi 6 standard in general use. Once all the parameters specific to PHY and MAC layer of the node are configured mobility models are added in the system if mobile scenarios are present in the application. Then the devices are assigned their IP addresses and client and server roles are defined to the nodes. Here the nodes can be configured to support either the TCP/IP or UDP protocol. At this point a basic wi-fi system is ready for simulation. The command line output format generated in NS-3 is shown in Figure 1.

An important point to note here is that NS-3 doesn't have support for visualizing the networks on its own, but users can add their own visualization tools to it. On running the simulation, a trace file in XML format is generated which can be used to add visualization. One of the most widely used animator with NS-3 is NetAnim based on Qt toolkit. The output in NetAnim is divided into 3 tabs – animator, stats, packets. The animator window output is shown in Figure 2, it shows the positioning of nodes and transmission of packet. Figure 3 represents the stats for each individual node in system and can be accessed using the stats tab of NetAnim. Packets tab provide information for packets in tabular format with timestamps.

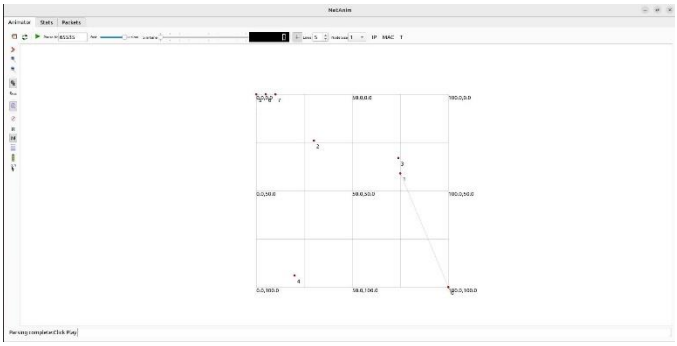

Figure 2. NetAnim animator window output

NS-3 has support available for both propagation loss model (PLM) and propagation delay models as well. These propagation model helps in replicating the environment for which the application is intended for.

Support for popularly used models like two ray ground tracing PLM for incorporating the effect of reflections from ground, log distance PLM for modelling path loss a signal experience in closed structures like buildings or a densely populated area, Okumura Hata PLM for modelling loss in urban areas etc.

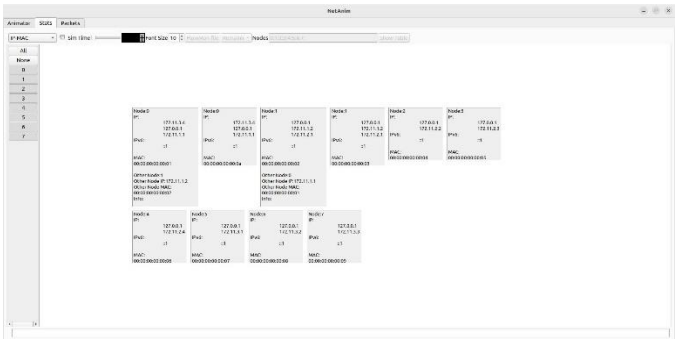

Figure 3. NetAnim stats window output

NS-3 also has capability of modelling multiple energy frameworks as energy consumption is one of the most important factors in any wireless device. There is implementation available for frameworks like energy source model, energy consumption model and energy harvesting models in NS-3. Energy Source model are used for modelling the source of power like capacitor or batteries whereas the energy harvesting model are used for modelling the production of energy while the devices are in operation using the sources like solar panels or piezoelectric harvesters. Energy consumption models are used to represent the devices which are going to use the power from source and harvesters.

The support for almost all the aspects of wireless communication is available in NS-3 making it a go to simulator for Wi-Fi network simulations but user need to have a decent knowledge of C++, working in LINUX and using command line arguments for interacting with simulator. These factors make the learning curve for using the simulator a bit step.

BLE Simulations in NS-3

NS-3 has no support for BLE stack, but researchers have built their own BLE stack in NS-3 in past for exploring various aspects of protocol. In [6] MAC address randomization framework was built in NS-3 to study the security features of BLE protocol. Another BLE stack implementation for analytical modeling of BLE protocol-based sensor network in NS-3 is present at [7]. These frameworks are not updated to the latest BLE stack and user need to have a grip on the C++ to understand and modify these implementations.

Contiki OS Java simulator (COOJA)

One of the most widely used operating system in IoT devices is Contiki operating system. Contiki OS is an open source and very less memory consuming OS designed specifically for devices with limited computational power. To simulate or emulate nodes or devices running on this operating system Contiki OS Java simulator (COOJA) is used. COOJA is a user-friendly simulator because of its GUI and a basic system of nodes can be built directly using the GUI. Its main features include support for protocols like 6LoWPAN, CoAP and IEEE 802.15.4. It also has features for debugging and analyzing the performance of the network. To add other features to the node the developer can easily access the source code and do the modifications.

COOJA also works in a LINUX based environment and the scripting can be done using C language.

Building wireless system in COOJA

The devices in COOJA simulator are referred to as motes. COOJA has virtual motes for various real devices and the COOJA Mote platform is available in it which can help in creating custom virtual devices with ease. COOJA has support for wireless communication protocol IEEE 802.15.4. This protocol falls under the category of low data rate and low power consuming wireless communication protocol and is one of the most widely used communication protocol in IoT devices and WSN. The most widely used modification of this standard is ZigBee which is maintained by the ZigBee Alliance. The support for protocols in other layers of the communication protocol like 6LoWPAN (IPv6 over Low-power Wireless Personal Area Networks) which helps in assigning address to these devices and connecting them over the internet, CoAP (Constrained Application Protocol) and MQTT (Message Queuing Telemetry Transport) which are application layer protocol and routing protocols like RPL (Routing Protocol for Low-power and Lossy Networks) also exists. Other than these standard protocols, COOJA also allows developers to build their custom protocols tailored for their needs.

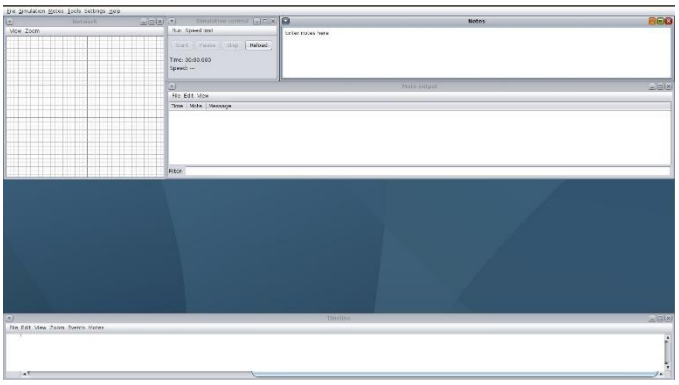

Figure 4. COOJA simulator user interface

To build a network of motes in the COOJA the process is start by selecting the radio medium which closely represent the real environment devices will be operated under. The options available for selection are unit disk graph medium (UDGM) distance loss or constant loss model which creates to circle or disks around motes to show its range of transmission and interference, directed graph radio medium (DGRM), multi-path ray tracer medium for modelling an environment where there are multiple sources of reflection around the nodes and to simulate an ideal scenario, there is an option to no radio traffic medium as well. In the next step a startup delay is added to the node. At this point the environment for nodes is setup and the user interface produce is presented in Figure 4. Next, the number of motes required in the application are added to the system. There are multiple types of motes which can be emulated in COOJA like Sky mote, CC430 mote, Z1 mote, Exp motes etc. Also, an option to create a COOJA mote is present here. After selecting the type of mote, these motes are configured with the Contiki process or firmware. Then number of motes required in the system are added. Along with number these motes are assigned a position and there are prebuilt options of positioning these nodes available as well. These motes can be placed in random, linear, or elliptical fashion or assigned a position manually as well. Once the motes are added in the system, these motes are configured to display the properties like log output, addresses, IDs, position etc.

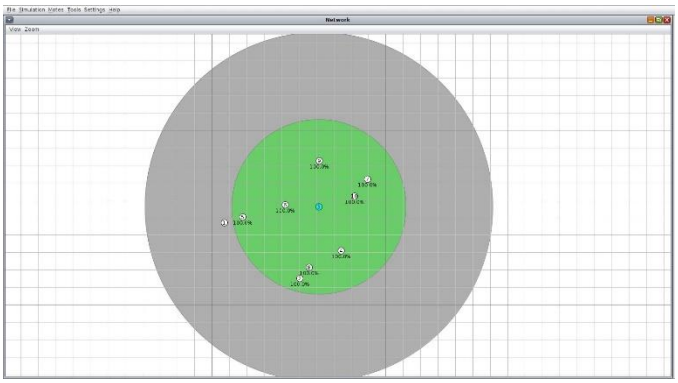

Figure 5. Network window view in COOJA simulator

Apart from the network window which displays the created network as presented in Figure 5, there is a simulation control window, mote output window and timeline window present in the COOJA simulator. Simulation control window is used to control the simulation speed or to procced the simulation in steps and to start and pause the simulations in general. The Mote output window displays the activities performed by each node in time as shown in Figure 6. The output parameters like power, transmission and reception of packets can also be monitored using this window. The timeline windows show the activity performed by each mote on a common timeline. The green and gray area around the mote represents the range of a particular mote.

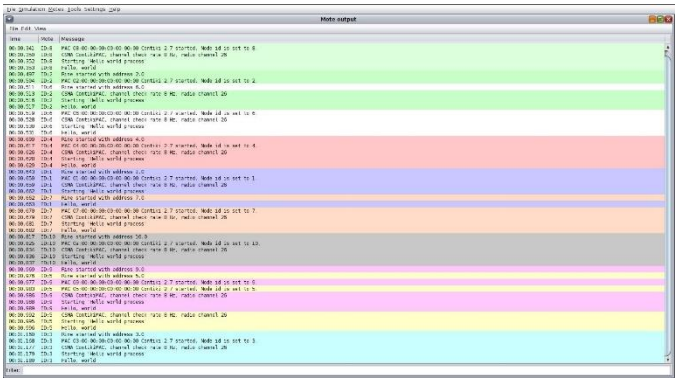

Figure 6. Mote output window view in COOJA simulator

The GUI of COOJA is quite rich and the basic system can be built entirely with the help of GUI. The visualization for results produced is quite good as well. The requirement of working with the command line arguments is also limited which reduces the efforts required for getting started with the toolbox.

BLE Simulations in COOJA

COOJA has a very limited support available for BLE network simulations. In [8] Micheal Spork developed a BLE stack implementation for exploring IPv6 packet exchange using BLE technology. The extension is available for Contiki-NG (Contiki-Next Generation) version. This implementation has support for advertising, initiating connections and configuring device as central and peripheral and IPv6 implementation for BLE as specified by RFC 7668. This stack is also not updated to keep in pace with the BLE stack changes.

MATLAB Bluetooth Toolbox

Mathworks introduced its MATLAB Bluetooth toolbox with limited functionalities initially in 2019b version of MATLAB. The functionalities introduced at start were limited to simple applications like detection of nearby BLE devices, forming connections with device, transmitting and receiving the data in both ASCII and binary forms. It made use of the Bluetooth device on the laptop either internal or external for achieving these functionalities. The support for simulating a BLE network wasn't present in toolbox at that time but the recent versions of the toolbox support simulations of BLE devices as well. The latest version of the toolbox is 2023b and the toolbox has now been developed to the extent that it can simulate any number of BLE nodes without any dependencies on the host Bluetooth device while retaining its earlier core functionalities as well.

Building BLE system in MATLAB

For simulating a simple connected type of network in MATLAB Bluetooth toolbox, the first step is to assign roles to Bluetooth device using "bluetoothLENode" object in toolbox which creates a virtual BLE device. The roles allowed for the devices are –

1. Central or peripheral to simulate BLE nodes working in central and peripheral role.
2. Isochronous – broadcaster and synchronized – receiver to simulate BLE broadcast audio network.
3. Broadcaster – observer role for nodes working in mesh network simulation.

Along with assigning the roles to device various parameters associated with these roles are defined. On transmitter side parameters like position, signal power and antenna gain, advertising interval etc. are defined. On receiver side its position, gain, sensitivity, range, scan interval etc. parameters are defined. Flexibility in choice of these parameters helps to model the hardware intended for final application. After assigning the roles to devices, connection related parameters like connection interval, connection offset, active period etc. are defined and a connection is configured between the nodes. Next, parameters related to network traffic are defined like application layer data rates, on and off time etc. After attaching this traffic to the nodes, direction of the flow is defined, and these nodes are passed to the simulator. Additionally, there are helper functions available for adding visualization to these simulations. To calculate various parameters at PHY, link layer (LL) and application layer (App) level there are statistics function available and the results are presented in Figure 7.

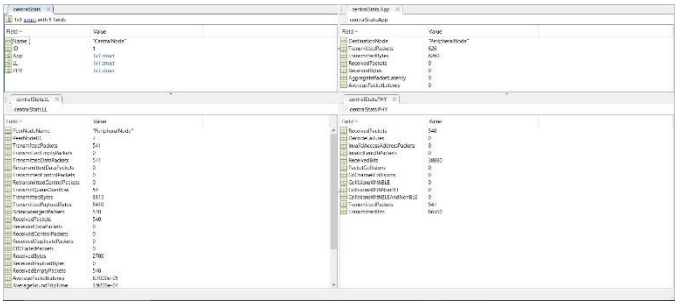

Figure 7. Statistics calculated layer wise in MATLAB

This framework remains similar for building different type of networks like mesh and audio with only changes to the script in accordance with the protocol requirements for different modes. For example, in mesh network simulations in addition to source and destination node, relay node role is assigned to some BLE devices in network so that a path can be formed between the origin and destination node as shown in Figure 8. The network judgement parameters like packet delivery ratio, throughput and latency can be accessed using the statistics function for both peripheral and central device.

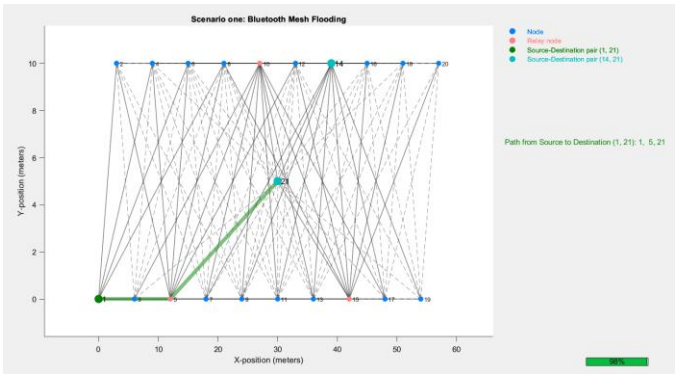

Figure 8. Mesh network simulation in MATLAB

The physical layer statistics calculated include received and transmitted packets by device, Packet and co channel collisions including information about both collisions with BLE and non-BLE packets. Also, if there are any decode failures or packets with invalid length and address will be reported here. The link layer statistics contains information about transmitted and received packets in detail like the total number of packets in each category, number of data and empty packets out of them. The information about node associated with the device is recorded in link layer as well. For example, in the case discussed above peer node for peripheral node will be displayed as central node and vice versa. It also contains information about the time for which the device was in idle, listen and sleep state. This information helps in calculating the energy consumption and the lifetime of the device for a given battery size. Other important information contained in the link layer include the throughput for the node as well as packet loss ratio. In the application layer, statistics like transmitted packets and bytes, received packet and bytes and latency information associated with the connection are present.

The overall architecture for building an application remains quite similar to NS-3 but the level of simplicity in building the scripts is much higher due to less level of complexity involved in MATLAB scripts compared to C++ .

MATLAB also supports addition of interference in simulations. Toolbox has the flexibility of having WLAN nodes working in conjunction with the BLE nodes and the statistics information take care of this case as well. The parameters like collision with non-BLE packets record information about packet loss in simulations when these types of 2.4 GHz spectrum devices are available in vicinity. This is one of the most important features for designing accurate models as the devices in this spectrum are increasing day by day. Also, these simulations help in analyzing the link layer level features of BLE like adaptive frequency hopping. The simulations approach this situation by defining a channel map and classifying channels as good or bad for transmission. Also, support for both the channel selection algorithm 1 and 2 as specified by Bluetooth core specifications is present in the toolbox.

The environment for nodes changes with industry these applications are intended for like an off-road vehicle application would have higher multi path reflections because of the presence of metal in vehicle and implements but in an office environment application, obstruction would have higher impact on packets strength and path. MATLAB has support for standard path loss models specified by Bluetooth self-

interest group (SIG) like office, industry, free space as well as higher multipath reflection models popularly used like Rayleigh, Rician model and raytracing.

Simulations for few other applications like direction and position finding which are built using angle of arrival (AOA) and angle of departure (AOD) calculations, concepts of triangulation and antenna switching can also be built in this toolbox. Also, simulation support for audio applications is available as well. In these applications calculation for the PDR in various scenarios can be done and heatmap can be generated as well for PDR distribution around the structure in which device is operating. Modelling of Voice communication with WLAN interference is also supported. At PHY layer the support for LE1M, LE2M as well as the LE coded mode is present. Bluetooth LE waveform generation and visualization features are also supported. Similar support for classic Bluetooth exists as well.

Overall, MATLAB Bluetooth toolbox supports simulations for almost all the features of BLE along with the ease of use and great visualization support making it one stop solution for building WSN applications using BLE as communication protocol.

Conclusion

The advancement in low energy wireless communication protocols have provided more flexibility and multiple options for building WSN and hence the simulators now have to be analyzed not only according to their features but also according to the protocol they support. NS-3 has a well-maintained stack for Wi-Fi protocol, COOJA simulator provides excellent platform for simulating Contiki OS and Zigbee communication protocol, but both these simulators have limited support for BLE network simulations. Mathworks with its Bluetooth toolbox provides excellent support for simulating the BLE networks and supports all the features for making the environment around nodes close to practical scenarios.

References

1. Minakov, Ivan & Passerone, Roberto & Rizzardi, Alessandra & Sicari, Sabrina. (2016). A Comparative Study of Recent Wireless Sensor Network Simulators. ACM Transactions on Sensor Networks. 12. 10.1145/2903144.
2. Yordanov, Y., & Haka, A. (2022). Bluetooth Low Energy Technology Simulators. *The Journal of CIEES*, *2*(1), 7–11. https://doi.org/10.48149/jciees.2022.2.1.1
3. Murgod, Tejaswini & Sundaram, (2021). A comparative study of different network simulation tools and experimentation platforms for underwater communication. Bulletin of Electrical Engineering and Informatics. 10. 879-885. 10.11591/eei.v10i2.1466.
4. Sundani, Haoyue & Li, Vijay & Devabhaktuni, Mansoor & Alam, Prabir & Bhattacharya, & Sundani, Harsh & Li, Haoyue & Devabhaktuni, Vijay & Alam, Mansoor & Bhattacharya, Prabir. (2011). Wireless Sensor Network Simulators: A Survey and Comparisons. International Journal Of Computer Networks.
5. Jevtić, Miloš & Zogovic, Nikola & Dimic, Goran. (2009). Evaluation of Wireless Sensor Network Simulators.
6. Abhishek Kumar Mishra, Aline Carneiro Viana, Nadjib Achir. SimBle: Generating privacy preserving real-world BLE traces with ground truth.
7. https://documents.kartikpatel.in/ns-3-dev-git/group__ble.html
8. Michael Spörk, Carlo Alberto Boano and Kay Römer, "Improving the timeliness of Bluetooth Low Energy in dynamic RF environments", ACM Transactions on Internet of Things, vol. 1, no. 2, pp. 1-32, 2020.

Contact Information

Garvit Periwal, John Deere
e-mail: periwalgarvit@johndeere.com
Swarupanand Sewalkar, John Deere
e-mail: sewalkarswarupanand@johndeere.com
Prashant Koparde, John Deere
e-mail: kopardeprashant@johndeere.com