

**Bộ công cụ mô phỏng mạng cảm biến có tính tới yếu tố năng lượng****A simulation platform for sensor networks with interest in energy**Hà Văn Phương<sup>1,2</sup>, Phạm Thị Ngọc Yến<sup>1</sup>, Lê Minh Hoàng<sup>1</sup>, Đào Trung Kiên<sup>1</sup>, Nguyễn Thanh Hường<sup>1</sup>,  
Nguyễn Việt Tùng<sup>1</sup><sup>1</sup>Viện MICA - ĐH Bách khoa Hà Nội, <sup>2</sup>ĐH Công nghiệp Hà Nội

Email: van-phuong.ha@mica.edu.vn

**Abstract**

Recently, sensor networks have been studied, applied and deployed widely in industrial as well as civil systems. Simulation software is an important tool to support researchers in the design, development and application of sensor networks. However, there is currently not many software solutions that allow the simulation that takes energy-related activities into consideration. This paper introduces a simulation platform that helps users to simulate the energy transformation process of the components as well as the nodes and the entire network, thereby allowing definition, investigation of the energy transformation in each node, hence laying the foundation of energy solutions for the entire network.

**Keywords**

Mạng cảm biến; mạng cảm biến không dây; mô phỏng mạng cảm biến; mô phỏng năng lượng.

**Tóm tắt<sup>1</sup>**

Hiện nay, mạng cảm biến đang được nghiên cứu, ứng dụng và triển khai rộng rãi trong các ứng dụng công nghiệp và dân dụng. Các phần mềm mô phỏng là công cụ quan trọng hỗ trợ cho các nhà nghiên cứu để thiết kế, phát triển, ứng dụng các mạng cảm biến. Tuy nhiên, hiện chưa có công cụ nào cho phép mô phỏng mạng với việc tính toán các hoạt động liên quan tới năng lượng. Bài báo này giới thiệu nền tảng mô phỏng mạng cảm biến cho phép người dùng mô phỏng được quá trình biến đổi năng lượng của từng thành phần của nút cũng như của các nút và của toàn mạng, từ đó cho phép định nghĩa, khảo sát quá trình sử dụng và chuyển hoá năng lượng của từng nút mạng và đặt nền tảng về giải pháp năng lượng cho toàn mạng.

**1. Giới thiệu**

Trong những năm gần đây, mạng cảm biến không dây (Wireless Sensor Networks - WSN) được đặc biệt quan tâm, nghiên cứu và ứng dụng rất mạnh mẽ trong nhiều lĩnh vực như giám sát môi trường [1], sức khoẻ [2], kiểm soát sản xuất công nghiệp, nông nghiệp, năng lượng [3], giao thông, an ninh, quân sự [4] và trong các ứng dụng dân dụng. Những ưu điểm vượt trội so với hệ thống cảm biến có dây như không cần dây cáp nguồn và dây tín hiệu, khả năng tùy biến cao, có tính mềm dẻo linh hoạt, dễ triển khai trên diện rộng và trong các môi trường phức tạp, mang lại hiệu quả cao về kinh tế. Tuy nhiên, một vấn đề thường gặp phải đối với WSN là việc cung cấp và đảm bảo duy trì năng lượng hoạt động của mạng trong thời gian dài.

Trong nghiên cứu, phát triển và ứng dụng WSN, việc mô phỏng mạng cảm biến thực sự quan trọng và rất cần thiết. Mô phỏng trong giai đoạn thiết kế cho phép dự đoán hiệu năng mạng và so sánh được các hệ thống khác nhau, cấu hình khác nhau của mạng khi các yếu tố trong mạng thay đổi như môi trường cài đặt hệ thống cảm biến, giao thức định tuyến để qua đó có thể đưa ra các lựa chọn phù hợp mục đích, ứng dụng. Hiện có nhiều phần mềm chuyên mô phỏng mạng cảm biến. Một số phần mềm chuyên dụng như NS (*Network Simulator*) [5], OMNeT++ (*Optical Micro-Networks Plus Plus*) [6], OPNET (*Optimized Network Engineering Tools*) [7], JSIM (*Java-based simulation*) [8]. Một số phần mềm kỹ thuật có hỗ trợ các công cụ mô phỏng mạng cảm biến như Matlab & Simulink. Tuy nhiên, các phần mềm trên tập trung chủ yếu vào việc mô phỏng các vấn đề liên quan cấu trúc mạng, giao thức định tuyến [9] và các kịch bản mà chưa quan tâm đến vấn đề năng lượng cũng như thời gian còn sống của các nút trong mạng [10]. Mô phỏng năng lượng của từng nút trong mạng khi trạng thái năng lượng của mỗi nút cần được giám sát, quản lý và điều phối theo kịch bản chung của toàn mạng sẽ hỗ trợ rất tốt cho các nhà thiết kế. Đây cũng là điều kiện tiên quyết của bài toán tối ưu hóa điều phối năng lượng trong mạng cảm biến.

Với hướng nghiên cứu về tối ưu hóa sử dụng năng lượng cho mạng cảm biến, yếu tố quan trọng cần đặc biệt quan tâm là trạng thái năng lượng của từng nút mạng cũng như hiệu năng mạng. Vì vậy, cần một nền tảng mô phỏng không chỉ quan tâm đến cấu trúc mạng hay giao thức định tuyến, truyền thông mà còn hỗ trợ người dùng trong việc mô phỏng các nút cảm biến trong mạng với các thuộc tính và phương thức cơ bản liên quan đến vấn đề năng lượng như thu thập năng lượng từ môi trường, trạng thái năng lượng từng nút, quản lý và điều phối tối ưu năng lượng trong mạng... Bài báo này trình bày việc thiết kế và xây dựng nền

<sup>1</sup> Ngày nhận bài: 29/03/2019; Ngày nhận bản sửa lần 1: 24/04/2019; Ngày nhận bản sửa lần 2: 15/07/2019; Ngày chấp nhận: 21/07/2019; Phản biện: Trương Đình Châu, Nguyễn Chí Ngôn.

tầng mô phỏng mạng cảm biến hỗn tạp có tính đến năng lượng của từng thành phần nút, của từng nút và toàn mạng. Từ đó làm cơ sở thực hiện bài toán giám sát, quản lý và điều phối tối ưu năng lượng trong toàn mạng cho giai đoạn nghiên cứu tiếp theo.

## 2. Phân tích và thiết kế nền tảng mô phỏng

### 2.1 Phân tích chức năng

Theo mục tiêu đã đặt ra, nền tảng mô phỏng được xây dựng cần cho phép người dùng thiết lập mạng với khả năng tùy biến cao đồng thời thiết lập môi trường hoạt động của mạng với các thông số môi trường tác động lên mạng và thực hiện mô phỏng theo kịch bản mà người dùng mong muốn. Nền tảng mô phỏng phải có tính mở: bên cạnh việc cung cấp thư viện các phần tử được xây dựng sẵn, cho phép người dùng thiết lập hệ thống cần mô phỏng, nền tảng mô phỏng cũng cần cho phép người dùng bổ sung thêm các phần tử mạng mới theo mục đích người dùng thông qua tự xây dựng mới hoặc kế thừa, mở rộng từ các phần tử sẵn có. Với nền tảng mô phỏng này, các chức năng có thể được thực hiện thông qua giao diện người dùng, hoặc bằng cách sử dụng ngôn ngữ lập trình C++ để thiết lập các kịch bản mô phỏng. Căn cứ vào hoạt động của mạng cảm biến, các yếu tố căn bản của nền tảng mô phỏng là các yếu tố của các nút cảm biến, môi trường đặt cảm biến, kịch bản mô phỏng mạng, không gian, thời gian cũng như các chức năng trích xuất dữ liệu và tương tác với người dùng.

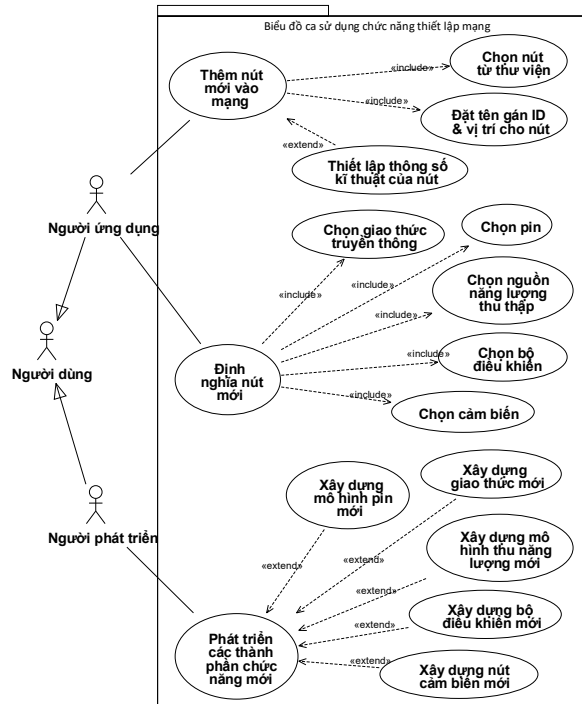
#### 2.1.1 Phân tích các chức năng của nền tảng mô phỏng

Tương tự các phần mềm mô phỏng khác, nền tảng mô phỏng trong bài báo này được phát triển với 3 nhóm chức năng chính bao gồm thiết lập mạng, thiết lập môi trường đặt mạng và chạy mô phỏng. Trong việc thiết lập mạng (xem H. 1), người dùng có thể lấy một nút có sẵn từ thư viện và cấu hình cho nút theo những thông số đã được cài đặt sẵn, hoặc tự định nghĩa một nút mới bằng cách lựa chọn và kết hợp các thành phần của nút một cách tùy ý, lựa chọn các thông số và chế độ hoạt động phù hợp cho từng thành phần này thông qua sử dụng ngôn ngữ lập trình C/C++. Bên cạnh đó, người dùng cũng có thể phát triển thêm các thành phần chức năng mới cho nút để bổ sung vào thư viện. Việc phát triển các thành phần mới này có thể được thực hiện mới hoàn toàn hoặc bằng cách kế thừa và mở rộng những thành phần đã có từ trước.

Nhóm chức năng thiết lập môi trường đặt mạng cung cấp cho người dùng khả năng lựa chọn môi trường vật lý cho hoạt động của mạng, bao gồm những yếu tố môi trường như không gian, nhiệt độ, độ ẩm, ánh sáng... Về khả năng tùy biến và mở rộng, người dùng cũng có thể tự mình phát triển các yếu tố môi trường theo các nhu cầu riêng biệt.

Nhóm chức năng chạy mô phỏng giúp người dùng thực hiện các thao tác chạy chương trình mô phỏng sau khi đã thiết lập mạng, môi trường đặt mạng và kịch bản mô phỏng. Hoạt động và các kết quả mô

phỏng của mạng cảm biến phải mang tính quá trình. Vì vậy, mọi hoạt động của mạng được đồng bộ theo tham số thời gian được cài đặt bởi người dùng. Kết quả mô phỏng có thể được hiển thị hoặc lưu lại theo nhu cầu người dùng.



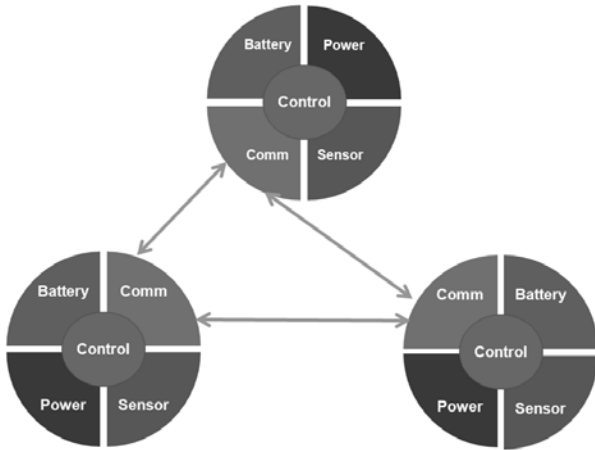
H. 1 Biểu đồ ca sử dụng chức năng thiết lập mạng

#### 2.1.2 Phân tích và lựa chọn mô hình nút cảm biến

Nút là thành phần cơ bản của mạng. Các nghiên cứu trước đây cũng đã có những nhìn nhận về cấu trúc của một nút cảm biến cơ bản bao gồm các thành phần chính như khối pin, khối cảm biến, khối xử lý, khối thu năng lượng, khối truyền thông [11]. Các phần mềm mô phỏng hiện nay thiết kế các nút không mô-đun hóa theo các thành phần trên, mà các nút được định sẵn cấu trúc chủ yếu chỉ cho phép người dùng cấu hình các thông số liên quan tới truyền thông nên hạn chế trong vấn đề mô phỏng nút và mạng cảm biến theo các bài toán đa dạng trong thực tế. Vì vậy, với các phần mềm mô phỏng hiện nay, người dùng gặp nhiều khó khăn trong bài toán mô phỏng mạng cảm biến có yếu tố liên quan đến năng lượng.

Từ nhận định trên, nền tảng mô phỏng được phát triển trong nghiên cứu này mô hình hoá cấu trúc các nút bằng 5 thành phần chính như trong H. 2, bao gồm các khối: nguồn (Power) có nhiệm vụ thu nhận năng lượng từ môi trường, pin (Battery) là nơi lưu và dự trữ năng lượng mà nút thu được từ khối nguồn, cảm biến (Sensor) để đo các đại lượng vật lý của môi trường, truyền thông (Comm) chịu trách nhiệm giao tiếp trao đổi thông tin và dữ liệu với bên ngoài, điều khiển (Control) có nhiệm vụ xử lý và điều phối chung tất cả các hoạt động của nút theo yêu cầu cụ thể của kịch bản mà người dùng cài đặt. Việc xây dựng nút trong phần mềm mô phỏng với 5 thành phần độc lập sẽ giúp việc phân tích và thiết kế bộ công cụ rõ ràng hơn, linh hoạt hơn. Bên cạnh việc sử dụng các nút có cấu trúc

định sẵn người dùng có thể dễ dàng xây dựng nút mới phù hợp mục đích sử dụng bằng cách phối ghép 5 thành phần này để có một nút của mình. Một nút được tạo ra không nhất thiết phải đầy đủ 5 thành phần được cài đặt, mà phụ thuộc vào việc thành phần đó có cần thiết hay không đối với yêu cầu và điều kiện hoạt động cụ thể của nút. Đây cũng là điểm khác biệt và ưu việt của nền tảng mô phỏng so với các phần mềm mô phỏng mạng cảm biến hiện tại.



H.2 Các thành phần cấu thành một nút cảm biến

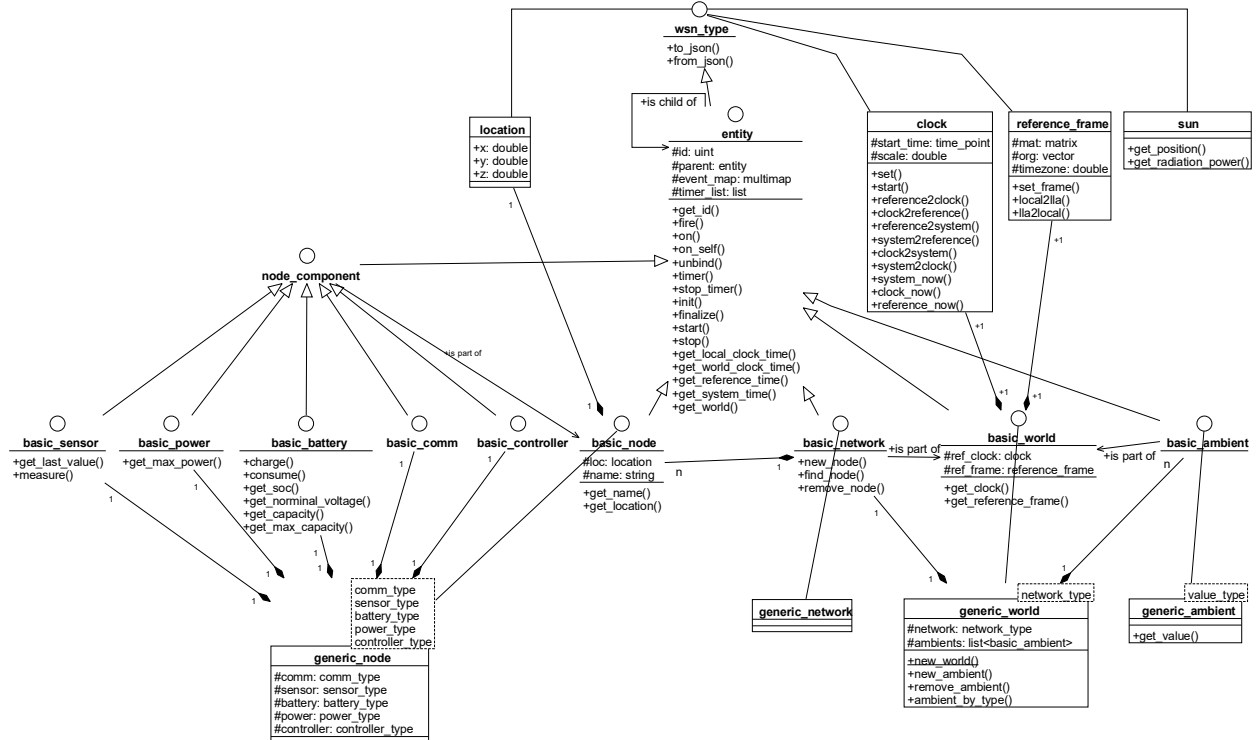
2.2 Thiết kế nền tảng mô phỏng

Từ phân tích tổng quát đến cụ thể về các chức năng ở trên, nền tảng mô phỏng được thiết kế với mô hình

lớp tổng quát được thể hiện như H. 3, với các nhóm giao diện (interface) và lớp (class) cho:

- Các thành phần của nút, trong đó node\_component là giao diện cơ sở.
- Các nút cảm biến, với basic\_node là giao diện cơ sở.
- Mạng cảm biến, với basic\_network là giao diện cơ sở.
- Môi trường đặt mạng, với basic\_world là giao diện cơ sở.
- Yếu tố môi trường, với basic\_ambient là giao diện cơ sở.

Ngoài ra, nền tảng mô phỏng này còn cung cấp các giao diện và lớp phụ trợ, như lớp clock cho hệ quy chiếu thời gian, reference\_frame cho hệ quy chiếu không gian, sun cho các tính toán liên quan tới hoạt động của mặt trời. Thời gian có thể được chuyển đổi qua lại giữa các hệ quy chiếu của đồng hồ khác nhau, bao gồm thời gian thực của máy tính, thời gian toàn cục của hệ mô phỏng, thời gian cục bộ của từng thực thể. Hoàn toàn tương tự, tọa độ được sử dụng trong mô phỏng cũng có thể được chuyển đổi qua lại giữa các hệ quy chiếu một cách dễ dàng, bao gồm tọa độ thực, tọa độ toàn cục của hệ mô phỏng, tọa độ cục bộ của từng nút. Các hàm chuyển đổi này được cung cấp trong các lớp clock, reference\_frame và trong giao diện entity.



H.3 Sơ đồ lớp tổng quát của nền tảng mô phỏng

Giao diện `wsn_type` là kiểu cơ sở chung nhất của toàn bộ các lớp của nền tảng mô phỏng, nhằm cung cấp những phương thức cơ bản cho tất cả các thực thể, ví dụ hai phương thức cơ bản `to_json()` và `from_json()` nhằm mục đích chuyển đổi cấu trúc dữ liệu gồm các tham số hay trạng thái của các thực thể thành chuỗi ký tự theo định dạng JSON phục vụ các mục đích như lưu trữ hay truyền thông.

Từ giao diện cơ sở này, giao diện `entity` được kế thừa và đại diện cho các thực thể có thể hoạt động với các yếu tố không gian, thời gian và cơ chế sự kiện, như các nút, các thành phần nút, mạng, môi trường... Các yếu tố chính của `entity` bao gồm ID là mã định danh riêng biệt cho mỗi thực thể của giao diện này, tham chiếu tới thực thể mẹ (*parent entity*), đồng hồ (*clock*) độc lập mà thời gian có thể được chuyển đổi qua lại với quy chiếu thời gian chung của toàn hệ thống, hệ tọa độ (*frame*) độc lập mà tọa độ cũng có thể được chuyển đổi qua lại với quy chiếu chung của toàn hệ. Giao diện `entity` cũng cung cấp các phương thức `start()` và `stop()` cho phép kích hoạt hay dừng hoạt động các thực thể này một cách riêng rẽ.

Bộ công cụ mô phỏng được thiết kế với tất cả các thực thể đều có phương thức hoạt động theo hướng sự kiện (*event driven*). Mỗi thực thể có thể cung cấp một số sự kiện, và cho phép các thực thể khác đăng ký nhận thông báo mỗi khi các sự kiện này xảy ra. Các sự kiện của giao diện hay lớp cơ sở sẽ được bao gồm trong các lớp dẫn xuất thông qua kế thừa. Để đảm bảo các thực thể của hệ mô phỏng hoạt động độc lập, nền tảng mô phỏng được thiết kế hoạt động theo cơ chế đa luồng (*multithreaded*). Mỗi thực thể `entity` khi được kích hoạt sẽ được cấp phát một luồng (*thread*) riêng để thực thi các xử lý của thực thể đó mà không ảnh hưởng tới các thực thể khác. Điều này tương tự cơ chế hoạt động của các cảm biến trong thực tế, độc lập trong các hệ đa tác tử (*multi-agent systems*).

Các sự kiện trong nền tảng mô phỏng này được cung cấp theo cơ chế lan truyền. Một sự kiện xảy ra tại một thực thể cũng được lan truyền lên các thực thể sở hữu nó, giúp cho các sự kiện có thể được các thực thể khác đăng ký tại chính thực thể đó, nhưng cũng có thể được đăng ký ở thực thể cấp cao hơn. Ví dụ, một mạng có nhiều nút cảm biến, và mỗi nút có nhiều thành phần. Khi cần xử lý sự kiện được kích hoạt của một thành phần của một nút nhất định, ta có thể đăng ký sự kiện này ngay tại thành phần đó. Nhưng nếu cần xử lý sự kiện được kích hoạt của tất cả các thành phần của một nút, ta có thể đăng ký sự kiện đó ở cấp độ nút; hoặc nếu cần xử lý cho tất cả các thành phần của các nút trong mạng, ta có thể đăng ký ở cấp độ mạng.

Bên cạnh cơ chế sự kiện, nền tảng mô phỏng này còn được thiết kế với cơ chế định thời (*timer*). Mỗi thực thể `entity` đều có thể thiết lập tùy ý số định thời theo nhu cầu xử lý. Các bộ định thời hoạt động theo thời gian cục bộ của thực thể.

Vì mỗi nút đều có 5 thành phần, để cho phần triển khai xây dựng bộ công cụ được thuận tiện và rõ ràng hơn nên giao diện thành phần nút `node_component` được đưa ra nhằm cung cấp các chức năng chung của

các thành phần này, trong đó có tham chiếu tới nút. Mỗi thành phần nút cơ bản nói trên tương ứng với 5 giao diện thành phần cơ sở là `basic_battery`, `basic_power`, `basic_sensor`, `basic_comm`, `basic_controller`, và mỗi giao diện thành phần này sẽ đảm nhiệm việc định nghĩa các phương thức cơ sở của 5 thành phần cảm biến.

Các nút được đại diện bởi giao diện `basic_node`, có thông tin cơ bản về vị trí, định danh, tên và được thiết lập dựa trên cơ chế tổ hợp các lớp thành phần nút. Mạng cảm biến được đại diện bởi giao diện `basic_network`, bao gồm các chức năng liên quan tới quản lý các nút. Giao diện `basic_world` đại diện cho thế giới thực, gồm các hệ quy chiếu thời gian và không gian toàn cục, mạng được mô phỏng, và các yếu tố môi trường đặt cảm biến, đại diện bởi giao diện cơ sở `basic_ambient`.

Trong một thực thể `basic_world`, người dùng có thể thiết lập tùy ý số lượng yếu tố môi trường như nhiệt độ, độ ẩm, ánh sáng... Vai trò của các yếu tố này là mô phỏng môi trường đặt mạng với những đại lượng vật lý giống như môi trường thực tế. Ví dụ, ta có thể thiết lập trường ánh sáng phụ thuộc vào vị trí đặt cảm biến và vị trí của mặt trời theo thời gian thực; hay trường nhiệt độ phụ thuộc vào thời gian theo giờ, ngày, tháng như trong thực tế dựa vào dữ liệu thực... Ngoài ra, cơ chế này cho phép người dùng có thể kết hợp mô phỏng giữa một là phần mạng ảo trên máy và phần mạng thực được cài đặt bên ngoài.

Từ các giao diện cơ sở, các lớp tổng quát được dẫn xuất với nhiệm vụ đảm bảo cấu thành hệ thống mạng hoàn chỉnh một cách cơ bản cho một ứng dụng mà người dùng cần để mô phỏng, bao gồm `generic_node`, `generic_network`, `generic_world`, `generic_ambient`.

### 3. Triển khai xây dựng nền tảng mô phỏng

Dựa trên phân tích và thiết kế nền tảng mô phỏng như ở trên, thư viện các phần tử đã được xây dựng, chủ yếu bao gồm định nghĩa các nút, thành phần nút, và các yếu tố môi trường đặt cảm biến. Trước hết các thành phần cấu thành nút được triển khai xây dựng theo 5 loại đã nêu. Từ các thành phần đó, lớp nút tổng quát `generic_node` được xây dựng với mục đích đưa ra các khuôn mẫu cho các lớp dẫn xuất để từ đó sẽ triển khai ra các lớp cụ thể hơn và đến từng đối tượng. Ví dụ, với việc triển khai xây dựng thành phần pin, hai lớp đã được xây dựng là `linear_battery` và `chemical_battery` như H. 4. Trong đó, `linear_battery` đại diện cho loại pin tuyến tính lý thuyết nhằm phục vụ mục đích mô phỏng bộ pin lý tưởng mà dung lượng hiện tại của pin phụ thuộc vào công suất tiêu thụ định mức và thời gian sử dụng, cũng như quá trình sạc chỉ phụ thuộc thời gian sạc, trong khi tốc độ sạc năng lượng được coi là hằng số. Lớp `chemical_battery` đại diện cho lớp các đối tượng pin hóa, với các quá trình sạc, xả năng lượng của pin phụ thuộc nhiều thông số, tham số khác nhau

tùy theo loại pin. Lớp này tiếp tục được thừa kế bởi các lớp cụ thể hơn, gồm các mô hình pin hóa phổ biến là axit chì, Li-ion, Ni-Cd và Ni-MH.

Mô hình phóng - nạp của các loại pin này được mô tả bằng các phương trình dưới đây [12].

- Pin axit chì

Mô hình phóng:

$$V_{batt} = E_0 - Ri - K \frac{Q}{Q-it} (i^* + it) + e^t \quad (1)$$

Mô hình nạp:

$$V_{batt} = E_0 - Ri - K \frac{Q}{it + 0.1Q} i^* - K \frac{Q}{Q-it} it + e^t \quad (2)$$

- Pin Li-ion

Mô hình phóng:

$$V_{batt} = E_0 - Ri - K \frac{Q}{Q-it} (i^* + it) + Ae^{-Bit} \quad (3)$$

Mô hình nạp:

$$V_{batt} = E_0 - Ri - K \frac{Q}{it + 0.1Q} i^* - K \frac{Q}{Q-it} it + Ae^{-Bit} \quad (4)$$

- Pin Ni-MH và Ni-Cd

Mô hình phóng:

$$V_{batt} = E_0 - Ri - K \frac{Q}{Q-it} (i^* + it) + e^t \quad (5)$$

Mô hình nạp:

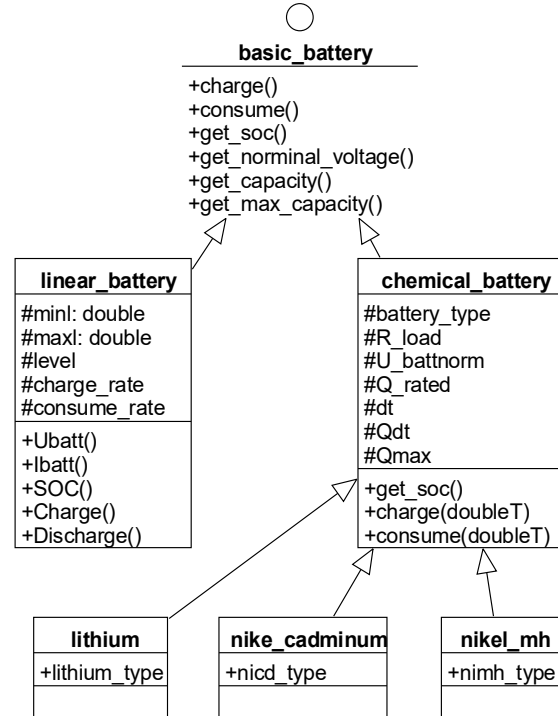
$$V_{batt} = E_0 - Ri - K \frac{Q}{|it| + 0.1Q} i^* - K \frac{Q}{Q-it} it + e^t \quad (6)$$

Trong đó,  $V_{batt}$  là điện áp pin (V),  $E_0$  là điện áp nguồn trong (không đổi) của pin (V),  $K$  là hằng số phân cực (V/Ah) hoặc điện trở phân cực ( $\Omega$ ),  $Q$  là dung lượng lớn nhất của pin (Ah),  $it$  là dung lượng chiết xuất của pin (Ah),  $i$  là dòng của pin (A),  $i^*$  là dòng lọc thông thấp (A),  $R$  là điện trở trong của pin ( $\Omega$ ),  $A$  là điện áp mũ (V),  $B$  là dung lượng mũ (A/h). Các khối pin được mô phỏng dựa trên các mô hình pin và các thông số trên, trong đó chế độ hoạt động và các tham số được cấu hình hoặc cài đặt bởi người dùng.

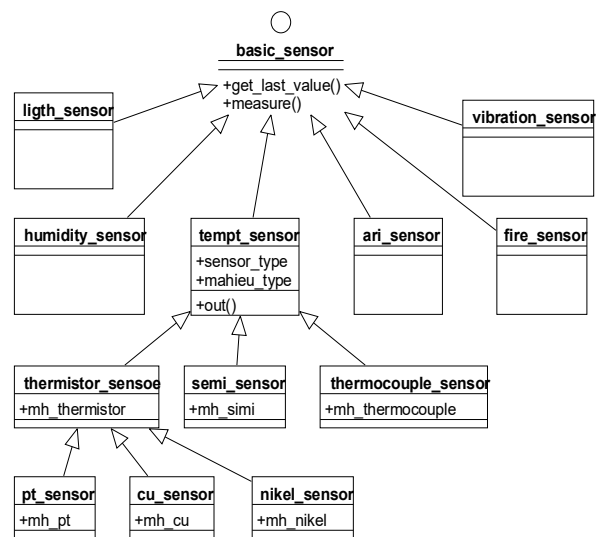
Việc thu năng lượng từ môi trường do khối nguồn đảm nhiệm và cần xét tới các yếu tố liên quan đến mật độ năng lượng, hiệu suất thu và chuyển đổi năng lượng. Các nguồn năng lượng chính được cài đặt gồm năng lượng điện lưới, năng lượng mặt trời, năng lượng nhiệt, năng lượng rung động, năng lượng RF.

Tương tự, sơ đồ triển khai thiết kế các lớp cảm biến được thể hiện trên H. 5. Từ giao diện cơ sở `basic_sensor`, các lớp cảm biến đại diện cho mỗi loại cảm biến đo các đại lượng vật lý như nhiệt độ, ánh sáng, độ ẩm, nồng độ chất khí, cảm biến báo cháy... được xây dựng. Từ mỗi lớp cảm biến ứng với mỗi đại lượng vật lý trên, các lớp con ứng với từng loại cảm biến đo đại lượng đó theo các nguyên lý khác nhau trong thực tế tiếp tục được cài đặt. Với

cách triển khai như vậy, thư viện các đối tượng cho các lớp được xây dựng và kết quả sẽ có thư viện đối tượng của các thành phần mà người dùng có thể sử dụng để thiết lập ra một nút mới hoàn chỉnh và đặt các thông số cho từng đối tượng phù hợp mục đích người dùng, cho phép khả năng tùy biến và tính thuận tiện cao.



H. 4 Sơ đồ lớp các loại pin



H. 5 Sơ đồ lớp cho thành phần sensor

## 4. Một số kết quả mô phỏng

### 4.1 Năng lượng của các nút

Để thử nghiệm việc theo dõi hoạt động của quá trình đo và thay đổi năng lượng, một kịch bản mô phỏng với các nút cảm biến ánh sáng (*light sensor*), nhiệt độ (*temperature sensor*), độ ẩm (*humidity sensor*) được

thiết lập. H. 6 thể hiện các thông báo được in ra theo thời gian mô phỏng, gồm các thông tin thời gian, tên nút, mã định danh nút, các sự kiện xảy ra kèm theo thông tin của sự kiện.

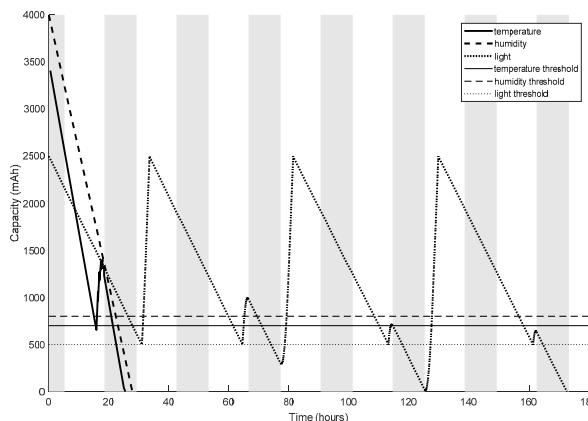
```

2018-05-22 12:00:0.007: Node light (118) started
2018-05-22 12:00:0.009: Node temperature (96) started
2018-05-22 12:00:0.010: Node humidity (107) started
>> 2018-05-22 12:00:0.219: Node light (118) consumed, level = 0.98
2018-05-22 12:00:0.236: Node light (118) measurement updated: 99645.9
2018-05-22 12:00:0.457: Node light (118) consumed, level = 0.96
2018-05-22 12:00:0.474: Node light (118) measurement updated: 100361
2018-05-22 12:00:0.519: Node humidity (107) consumed, level = 0.96
2018-05-22 12:00:0.534: Node humidity (107) measurement updated: 75.447
2018-05-22 12:00:0.703: Node light (118) consumed, level = 0.94
2018-05-22 12:00:0.720: Node temperature (96) consumed, level = 0.86
2018-05-22 12:00:0.736: Node light (118) measurement updated: 99970.4
2018-05-22 12:00:0.750: Node temperature (96) measurement updated: 22.133
2018-05-22 12:00:0.963: Node light (118) consumed, level = 0.92
2018-05-22 12:00:0.980: Node light (118) measurement updated: 99519.2
2018-05-22 12:00:1.057: Node humidity (107) consumed, level = 0.92
2018-05-22 12:00:1.074: Node humidity (107) measurement updated: 81.6227
2018-05-22 12:00:1.207: Node light (118) consumed, level = 0.9
2018-05-22 12:00:1.221: Node light (118) measurement updated: 98731.9
    
```

**H. 6** Kết quả mô phỏng quá trình đo và mức năng lượng còn của nút cảm biến

Diễn biến về năng lượng của từng nút được thể hiện trên H. 7. Trong kịch bản mô phỏng ví dụ này, ngưỡng chuyển sang chế độ thu năng lượng của mỗi cảm biến được thiết lập ở 20%, nghĩa là khi dung lượng còn lại nhỏ hơn 20% dung lượng lớn nhất của pin thì nút sẽ chuyển sang chế độ thu năng lượng mặt trời, và khi đầy pin sẽ dừng chế độ sạc. Tuy nhiên, việc thu năng lượng mặt trời ở đây còn phụ thuộc vào thời gian thu là ngày hay đêm, và công suất thu năng lượng cũng phụ thuộc vào vị trí tương đối với mặt trời theo thời điểm và thời tiết trong ngày. Trên hình, các dải màu trắng thể hiện thời gian ban ngày, khoảng màu xám thể hiện thời gian ban đêm. Các thời gian chuyển ngày/đêm và vị trí của mặt trời được tính toán bằng các công thức thiên văn học.

Dung lượng pin của cảm biến đo nhiệt độ (*đường nét liền*) cho thấy dung lượng lớn nhất là 3500mAh, khi còn dưới 20% chuyển sang chế độ sạc từ năng lượng mặt trời, nhưng vì thời điểm này đã sắp chuyển sang ban đêm nên mức năng lượng sạc nhỏ dần và không đủ để pin sạc đầy, trong khi đó cảm biến vẫn liên tục tiêu thụ năng lượng dẫn đến pin sẽ bị hết trước khi trời sáng và cảm biến đó nhanh chóng ngừng hoạt động. Hiện tượng tương tự cũng xảy ra với cảm biến độ ẩm (*đường nét đứt*). Còn cảm biến ánh sáng (*đường chấm chấm*) thì thời điểm chuyển sang chế độ sạc là bắt đầu ban ngày nên pin được sạc trong thời gian dài và pin đầy nên sống được nhiều ngày, tất nhiên trong trường hợp này ngẫu nhiên sẽ có lúc trạng thái còn 20% pin cũng đúng vào lúc bắt đầu ban đêm và cảm biến cũng bị hết pin trước khi trời sáng. Tuy nhiên, đây chỉ là một trường hợp mang tính minh họa, còn nhiều trường hợp khác nữa để mô phỏng quá trình tiêu thụ và thu nạp năng lượng từ mặt trời nói riêng và từ môi trường nói chung.



**H. 7** Quá trình tiêu thụ năng lượng và thu nạp năng lượng mặt trời

Qua trường hợp mô phỏng này, chúng ta cũng thấy rằng dù các cảm biến có thể thu thập được năng lượng từ môi trường vào ban ngày để đủ dùng vào ban đêm nhưng không có nghĩa rằng cảm biến đó sẽ sống mãi, bởi việc thu năng lượng cho cảm biến còn phụ thuộc thời gian và nhiều yếu tố của môi trường. Chính vì vậy, cần thiết phải có những nghiên cứu phát triển các giải pháp tối ưu năng lượng nhằm duy trì hoạt động mạng ổn định lâu dài.

**4.2 Truyền thông**

Để thử nghiệm quá trình truyền thông, một kịch bản mô phỏng được thiết lập với một mạng gồm 50 nút cảm biến với kịch bản đơn giản là cho một nút định trước sẽ đo giá trị và gửi kết quả đo đến nút chủ. Các nút mạng có vị trí là tọa độ được gán ngẫu nhiên trong một vùng quy định. Các sự kiện và thông tin kèm theo trong quá trình mô phỏng được thể hiện trên H. 8.

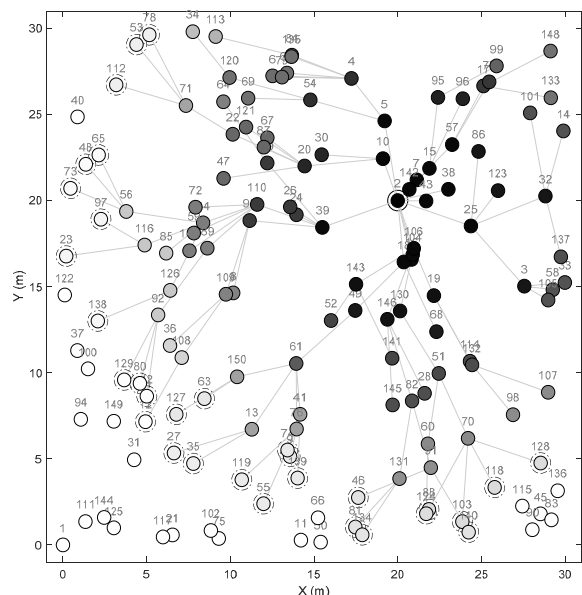
Kịch bản tương tự với 150 nút mạng trong phạm vi 30m×30m, trong đó nút 1 ở tọa độ (0, 0) là nút chủ. Quá trình truyền thông được thực hiện bằng cơ chế quảng bá (*broadcast*) với ràng buộc gói tin chỉ gửi đi trong 5 bước (*TTL – time to live*) và khoảng cách truyền thông mỗi bước là 5m. Bản tin được phát đi từ nút đo và quảng bá tới các nút xung quanh, mỗi nút khi nhận được một bản tin sẽ tiếp tục chuyển tiếp cho tất cả các nút xung quanh nó, cho đến khi nút chủ nhận được hoặc hết 5 bước quảng bá bản tin, và mỗi nút chỉ quảng bá bản tin này tối đa một lần. H. 9 cho thấy với cấu hình này, bản tin không thể truyền được tới nút chủ.

Bằng cách tăng khoảng cách truyền thông lên 10m và TTL lên 10 bước, kết quả mô phỏng thể hiện trên H. 10 cho thấy trong trường hợp này bản tin truyền từ nút 148 có thể đến được nút chủ số 1. Kết quả một kịch bản mô phỏng khác tương tự nhưng với cơ chế truyền thông theo một cấu trúc có kết nối giữa các nút cảm biến được thiết lập từ lúc khởi tạo được thể hiện trên H. 11. Cơ chế truyền thông này giúp tiết kiệm số bản tin được truyền, nhưng số bước trung gian tới nút chủ sẽ lớn hơn, và khiên độ trễ cũng lớn hơn.

```

2018-04-30 12:00:1.651: Node temp49 (622) started: 19, 0, 0
2018-04-30 12:00:1.681: Node temp50 (633) started: 29, 24, 0
>> measure2
2018-04-30 12:10:55.411: Node temp2 (105) measurement updated: 19.5665
>> 2018-04-30 12:10:56.397: Node temp23 (336) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.245: Node temp50 (633) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.481: Node temp33 (446) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.514: Node temp30 (413) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.547: Node temp27 (380) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.577: Node temp43 (556) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.632: Node temp25 (358) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.673: Node temp42 (545) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.711: Node temp11 (204) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.745: Node temp24 (347) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.778: Node temp8 (171) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.816: Node temp48 (611) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.859: Node temp37 (490) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:57.932: Node temp16 (259) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.003: Node temp20 (303) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.164: Node temp44 (567) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.204: Node temp29 (402) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.275: Node temp7 (160) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.325: Node temp39 (512) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.580: Node temp40 (523) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:58.623: Node temp5 (138) forwarded 56 bytes for temp2 (105)
2018-04-30 12:10:59.031: Node temp1 (94) received from temp13 (226): node 105 value 19.51
2018-04-30 12:10:59.271: Node temp3 (116) forwarded 56 bytes for temp7 (160)
2018-04-30 12:10:59.309: Node temp45 (578) forwarded 56 bytes for temp29 (402)
2018-04-30 12:10:59.733: Node temp9 (182) forwarded 56 bytes for temp24 (347)
    
```

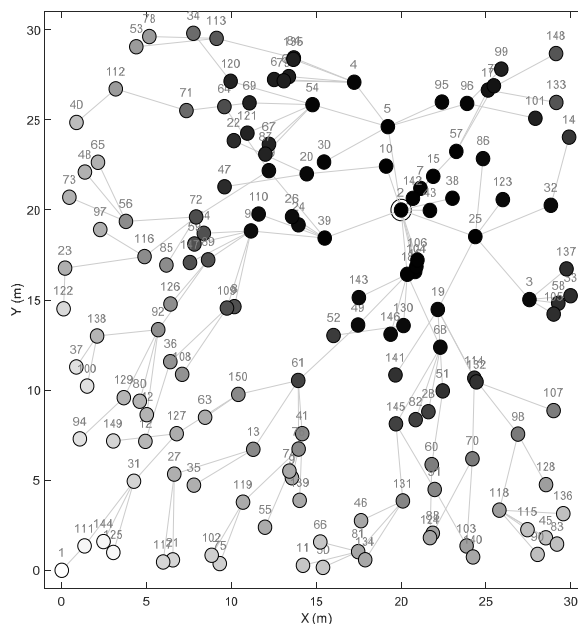
H. 8 Kết quả mô phỏng kịch bản truyền thông trong mạng



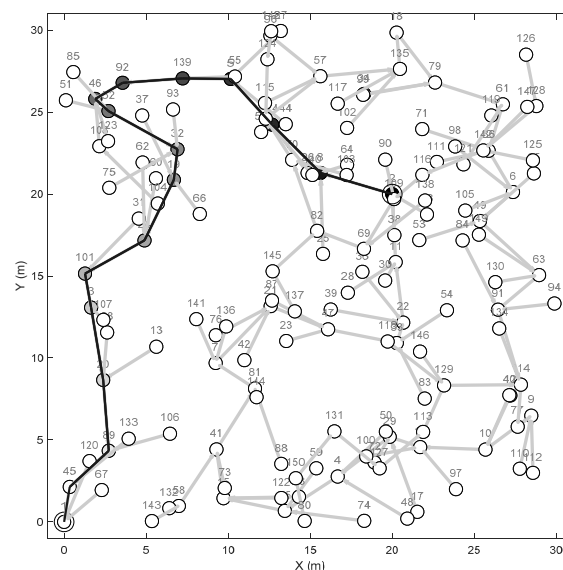
H. 9 Kịch bản truyền thông với ràng buộc 5 bước, khoảng cách 5m

### 5. Kết luận

Trong nghiên cứu này, nhóm tác giả đã phân tích và thiết kế một nền tảng mô phỏng mạng cảm biến có tính đến yếu tố năng lượng. Bước đầu triển khai xây dựng nền tảng mô phỏng đã mang lại một số kết quả khả quan, cho phép thiết lập mạng cảm biến với mỗi nút gồm 5 thành phần, thiết lập môi trường vật lý đặt mạng, cho phép mô phỏng được trạng thái năng lượng của các nút, quá trình tiêu thụ cũng như quá trình thu năng lượng và sạc pin từ các nguồn năng lượng xung quanh theo các tham số thiết lập bởi người dùng, kèm theo các cơ chế truyền thông và điều phối hoạt động trong mạng giữa các nút cảm biến. Nền tảng mô phỏng hoạt động theo hướng sự kiện, thể hiện qua việc cho phép đánh giá về năng lượng của nút, và từ đó có thể đưa ra các quyết định liên quan đến hoạt động của từng nút và toàn mạng.



H. 10 Kịch bản truyền thông với ràng buộc 10 bước, khoảng cách 10m



H. 11 Mô phỏng truyền thông không giới hạn số bước và 10m/bước

Hiện tại, nền tảng mô phỏng này chưa có giao diện cho người dùng cuối, mà đòi hỏi sử dụng ngôn ngữ C/C++ để thiết lập bài toán mô phỏng của mình. Nền tảng mô phỏng là tiền đề cho các nghiên cứu, ứng dụng tiếp theo về tối ưu thiết kế và hoạt động của mạng, đặc biệt là giải quyết các vấn đề liên quan tới năng lượng.

### Lời cảm ơn

Nhóm tác giả xin chân thành cảm ơn trường ĐH Bách khoa Hà Nội đã tài trợ thực hiện nghiên cứu này trong khuôn khổ đề tài cấp cơ sở mã số T2017-PC-124.

**Tài liệu tham khảo**

- [1] Jamil MS, Jamil MA, Mazhar A, Ikram A, Ahmed A, Munawar U (2015) *Smart environment monitoring system by employing wireless sensor networks on vehicles for pollution free smart cities*. Procedia Engineering, vol. 107, pp. 480-484
- [2] Vo MT, Nghi TT, Tran VS, Mai L, Le CT (2015) *Wireless sensor network for real time healthcare monitoring: network design and performance evaluation simulation*. 5th Intern. Conf. on Biomedical Engineering in Vietnam, pp. 87-91
- [3] Fadel E, Gungor VC, Nassef L, Akkari N, Malik MA, Almasri S, Akyildiz IF (2015) *A survey on wireless sensor networks for smart grid*. Computer Communications, vol. 71, pp. 22-33
- [4] Pompili D, Akyildiz IF (2009) *Overview of networking protocols for underwater wireless communications*. IEEE Communications Magazine, vol. 47, issue 1, pp. 97-102
- [5] Saluja AK, Dargad SA, Mistry K (2017) *A Detailed Analogy of Network Simulators - NS1, NS2, NS3 and NS4*. Intern. Journal on Future Revolution Computer Science & Communication Engineering, vol. 3, issue 12, pp. 291-295
- [6] Varga A (2019) *A practical introduction to the OMNeT++ simulation framework*. Recent Advances in Network Simulation, Springer, pp. 3-51
- [7] Li X, Peng M, Cai J, Yi C, Zhang H (2016) *OPNET-based modeling and simulation of mobile Zigbee sensor networks*. Peer-to-Peer Networking and Applications, vol. 9, issue 2, Springer, pp. 414-423
- [8] Neves PA, Veiga ID, Rodrigues JJ (2008) *G-JSIM - a GUI tool for Wireless Sensor Networks simulations under J-SIM*. IEEE Intern. Symposium on Consumer Electronics, pp. 1-4
- [9] El-Hoiydi A, Decotignie JD (2004) *WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks*. Intern. Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, Springer, pp.18-31
- [10] Toor AS, Jain AK (2017) *A survey on wireless network simulators*. Bulletin of Electrical Engineering and Informatics, vol. 6, no. 1, pp. 62-69
- [11] Gilbert JM, Balouchi F (2008) *Comparison of energy harvesting systems for wireless sensor networks*. Intern. Journal of Automation and Computing, vol. 5, issue 4, pp. 334-347
- [12] Tremblay O, Dessaint LA (2009) *Experimental validation of a battery dynamic model for EV applications*. World Electric Vehicle Journal, vol. 3, pp. 289-298