

XÂY DỰNG IOT PLATFORM CHO NHÀ THÔNG MINH TƯƠNG THÍCH CHUẨN ECHONET LITE

Nguyễn Hoài Sơn và Nguyễn Việt Bắc

Khoa Công nghệ thông tin - Trường Đại học Công nghệ, Đại học Quốc Gia Hà Nội

Email: sonnh@vnu.edu.vn, bacnv_58@vnu.edu.vn

Abstract— Để xây dựng các ứng dụng Internet of Things trong nhà thông minh thì các thiết bị cần được kết nối và truy cập một cách dễ dàng dựa trên chuẩn giao thức chung. Trong bài báo này, chúng tôi đề xuất một IoT platform giúp cho các thiết bị trong nhà thông minh có thể giao tiếp với nhau thông qua giao thức Echonet Lite, một giao thức được hỗ trợ bởi nhiều công ty sản xuất thiết bị điện dân dụng, đồng thời cho phép các thiết bị trong nhà giao tiếp với các máy tính trên mạng Internet thông qua một Home gateway sử dụng giao thức MQTT (Message Queue Telemetry Transport). IoT platform của chúng tôi cung cấp các chức năng giao tiếp với các thiết bị cũng như lưu trữ thông tin trạng thái và cập nhật trạng thái của các thiết bị cho các ứng dụng IoT trong nhà thông minh. Chúng tôi đã cài đặt thành công IoT platform đề xuất và triển khai thử nghiệm một ứng dụng giám sát và điều khiển thiết bị và một ứng dụng về ô nhiễm thông minh trên IoT platform này.

Keywords-Nhà thông minh, Echonet lite, MQTT, Home gateway.

I. GIỚI THIỆU

Xây dựng nhà thông minh hiện đang là một nhu cầu phổ biến của người dùng do sự tiện nghi và thoải mái mà nó mang lại. Cùng với sự gia tăng khả năng kết nối của các thiết bị vào mạng, đã có nhiều dịch vụ cung cấp cho nhà thông minh như dịch vụ điều khiển các thiết bị từ xa, dịch vụ an ninh sử dụng tín hiệu phát hiện đột nhập trái phép, dịch vụ cứu hỏa sử dụng dữ liệu báo cháy từ các cảm biến báo cháy trong nhà thông minh,... Đây là những dịch vụ tiêu biểu cho các ứng dụng Internet of Things (IoT) cho nhà thông minh. Tuy nhiên, để xây dựng và triển khai các dịch vụ IoT, các thiết bị trong nhà thông minh cần giao tiếp với nhau theo một chuẩn giao tiếp chung. Bên cạnh đó, với sự phát triển của mạng Internet, việc xây dựng và triển khai các dịch vụ trong nhà thông minh cần được thực hiện tại các máy tính trên cơ sở hạ tầng điện toán đám mây. Điều này sẽ giúp cho việc triển khai, duy trì và cập nhật các dịch vụ một cách dễ dàng với chi phí thấp.

Echonet lite là một giải pháp cho chuẩn giao thức được đề xuất bởi tổ chức Echonet với sự tham gia của nhiều công ty sản xuất thiết bị điện như Panasonic, Toshiba, Fujitsu, ... Echonet lite cung cấp một phương thức giao tiếp đơn giản, chi phí thấp giữa các thiết bị, điện trong nhà. Chuẩn này hiện đang được đề xuất là chuẩn quốc tế bởi các cơ quan tiêu chuẩn quốc tế IEC (the

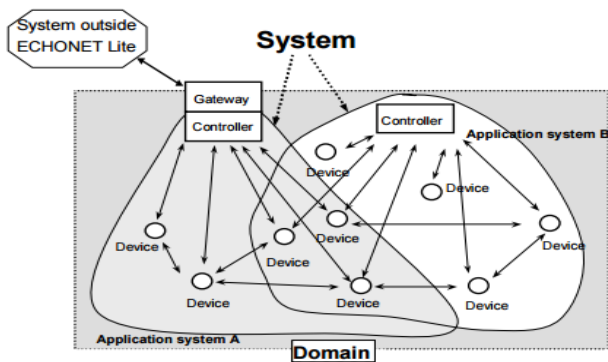
International Electrotechnical Commission) và ISO/IEC trong chuẩn IEC 62394 [2] và ISO/IEC 14543-4-3 [3].

Bên cạnh đó, giao thức MQTT (viết tắt của Message Queuing Telemetry Transport) là một giao thức cho phép quản lý việc chia sẻ dữ liệu giữa nhiều bên tham gia và được sử dụng phổ biến cho các ứng dụng IoT. Giao thức này được chuẩn hóa bởi tổ chức OASIS [4] với sự tham gia của hơn 5000 thành viên thuộc hơn 600 tổ chức khác nhau và các thành viên cá nhân của hơn 65 quốc gia và đã được phê chuẩn bởi tổ chức ISO (the International Organization for Standardization) và IEC (the International Electrotechnical Commission) trong chuẩn ISO/IEC 20922 [5].

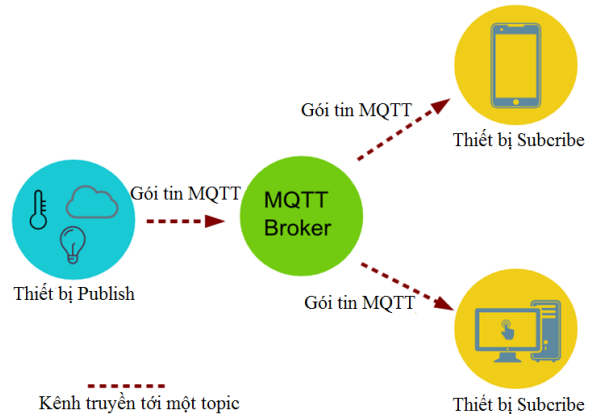
Mặc dù chuẩn giao thức Echonet lite cho phép các thiết bị điện trong gia đình thuộc các hãng khác nhau có thể giao tiếp và chia sẻ dữ liệu cho nhau, tuy nhiên, một ứng dụng nhà thông minh sẽ cần các thông tin liên quan đến các thiết bị trong nhà, dữ liệu về trạng thái của các thiết bị cũng như khả năng truy cập và cập nhật trạng thái của các thiết bị từ Internet. Để giảm thiểu chi phí cho việc xây dựng được một ứng dụng cho nhà thông minh, cần thiết phải có một IoT platform giúp thực hiện các tác vụ cơ bản như trên. Trong bài báo này, chúng tôi đã thiết kế một IoT platform với các chức năng chính như sau.

- Home gateway có khả năng tự động phát hiện ra các thiết bị Echonet lite, thu thập dữ liệu trạng thái của các thiết bị trong nhà thông qua giao thức Echonet Lite và cập nhật dữ liệu này cho máy tính đóng vai trò MQTT Broker trên mạng Internet thông qua giao thức MQTT.
- Lưu trữ dữ liệu trạng thái của các thiết bị Echonet Lite trong nhà thông minh trong cơ sở dữ liệu và cung cấp cho các nhà cung cấp dịch vụ IoT (IoT Service Provider viết tắt là IoT-SP)
- Cung cấp thông tin trạng thái của các thiết bị Echonet Lite theo thời gian thực cho IoT-SP trên mạng Internet đồng thời cho phép các IoT-SP điều khiển các thiết bị Echonet Lite thông qua Home Gateway.

IoT platform của chúng tôi sẽ giúp cho việc xây dựng và triển khai các dịch vụ trong nhà thông minh được thực hiện một cách dễ dàng, thông qua việc truy cập vào dữ liệu trạng thái của các thiết bị và khả năng điều khiển từ xa thông qua giao thức MQTT và Echonet lite.



Hình 1 Ví dụ phạm vi miền và cấu hình một hệ thống ứng dụng [1]



Hình 2 Mô hình hoạt động của giao thức MQTT

Chúng tôi đã xây dựng và cài đặt thử nghiệm IoT platform sử dụng các thư viện mã nguồn mở. Chúng tôi đã xây dựng một ứng dụng Android cung cấp chức năng giám sát và điều khiển các thiết bị Echonet Lite trong nhà thông minh và một ứng dụng ổ điện thông minh sử dụng IoT platform đề xuất để chứng minh khả năng ứng dụng của platform của chúng tôi. Chúng tôi cũng đã cài đặt các thiết bị Echonet Lite tương thích chuẩn Echonet Lite thông qua việc sử dụng các board mạch chi phí thấp như Raspberry Pi 3, Arduino.

Phần còn lại của bài báo được tổ chức như sau: trong phần II, chúng tôi miêu tả thiết kế hệ thống đề xuất. Trong phần III, chúng tôi miêu tả cách thức thực thi hệ thống và các kết quả cài đặt thử nghiệm. Cuối cùng, chúng tôi kết luận bài báo trong phần IV.

II. THIẾT KẾ HỆ THỐNG

2.1 Giao thức Echonet lite

Echonet Lite là một chuẩn giao thức dành cho các thiết bị điện trong gia đình và được phát triển với các đặc trưng:

- Dễ dàng phát triển cho hệ thống các ngôi nhà khác nhau: Mạng gia đình sẽ hoạt động tốt ngay cả khi kết nối và vận hành các thiết bị từ các nhà sản xuất khác nhau. Một giao thức truyền thông chung giữa các thiết bị giúp liên kết chúng ở cấp hệ thống. Người dùng có thể chọn và cài đặt thiết bị thích hợp nhất với nhu cầu của họ từ một loạt các sản phẩm tương thích Echonet Lite từ các nhà sản xuất khác nhau.
- Kéo dài chu trình hoạt động của các dịch vụ hay các thiết bị: Công việc này được giao cho Echonet Lite adapter trung gian xử lý, các thiết bị chỉ được thiết kế đơn giản nhất có thể. Ví dụ như các thiết bị chỉ phải hồi đáp khi nhận được yêu cầu từ adapter.
- Dễ dàng cài đặt hệ thống, lắp đặt, thay thế và di chuyển thiết bị
- Kết nối và cùng tồn tại với các hệ thống khác

Ví dụ về kiến trúc của một hệ thống Echonet Lite được mô tả như hình vẽ 1.

Một hệ thống Echonet Lite kết hợp các nhóm thiết bị có cùng chức năng quản lý tài nguyên, bảo mật, ... Do đó khu vực lớn nhất mà Echonet Lite có thể quản lý được gọi là miền. Mỗi miền sẽ được chỉ định kiểm soát một loạt các thiết bị (thiết bị gia đình, thiết bị gia dụng và thiết bị điện tử tiêu dùng, cảm

biến, điều khiển, điều khiển từ xa, ...) xuất hiện trong phạm vi của nó. Hệ thống nằm trong miền, thực hiện hoạt động truyền thông và liên kết giữa các thiết bị với bộ giám sát, điều khiển. Một hệ thống chỉ thực hiện một số lĩnh vực nhất định, do đó một miền có thể gồm nhiều hệ thống, một thiết bị có thể tồn tại và hoạt động trong một hoặc nhiều hệ thống. Khi một hệ thống muốn kết nối với một hệ thống khác nằm ngoài miền sẽ phải thông qua một Gateway Interface.

Hình 1 là một ví dụ được xây dựng phổ biến của một hệ thống Echonet Lite. Theo hình 1, mỗi một Node Echonet Lite (được định nghĩa là một thiết bị hoặc bộ điều khiển kết nối với mạng) trong hệ thống có thể trao đổi dữ liệu một cách tự do với nhau và với bộ điều khiển (cũng là một Node, được gọi chung là Node Controller). Hình này cho hai hệ thống ứng dụng A, B trong cùng một miền, các Node Device trong miền có thể thuộc một, hoặc cả hai hệ thống này. Mỗi hệ thống thực hiện điều khiển, giám sát các thiết bị kết nối với hệ thống mỗi thiết bị không chỉ giao tiếp được với bộ điều khiển trong hệ thống mà còn có thể giao tiếp với các thiết bị trong hệ thống khác thông qua gateway.

2.2 Giao thức MQTT

Giao thức MQTT hoạt động theo mô hình publish/subscribe theo như mô tả trong hình 2.

Trong mô hình hoạt động của giao thức MQTT, bên gửi dữ liệu được gọi là thiết bị publisher và bên nhận được gọi là thiết bị subscriber. Các thiết bị này còn được gọi chung là MQTT client. Bên cạnh các thành phần publisher và subscriber thì giao thức MQTT sử dụng một thiết bị trung gian đóng vai trò trung chuyển các gói tin gọi là MQTT Broker. Để nhận được thông tin thì các thiết bị subscriber sẽ gửi thông báo yêu cầu cung cấp thông tin (subscribe) theo một chủ đề (topic) nhất định đến MQTT broker. Chủ đề là một cái tên để phân biệt các kênh truyền từ các thiết bị publisher khác nhau và có dạng cây A/B/C/.../X/Y. Khi một thiết bị publisher gửi thông báo cung cấp thông tin với chủ đề nhất định lên MQTT Broker thì MQTT Broker sẽ gửi thông báo đó đến các thiết bị subscriber đã yêu cầu chủ đề đó.

MQTT Broker không chỉ đóng vai trò chuyển tiếp các thông báo từ publisher tới các subscriber dựa trên chủ đề (topic) của thông báo mà còn đóng vai trò các vai trò khác như duy trì các kết nối giữa các publisher và subscriber tới MQTT Broker, phát

hiện sự mất kết nối bất thường của các thiết bị, truyền lại các gói tin, chứng thực và phân quyền cho thiết bị mỗi khi gửi thông báo publish hoặc subscribe tới MQTT Broker. Hiện nay có nhiều MQTT Broker được các hãng phát triển như: Mosquitto MQTT, Verner MQTT, Hivemq ...

Giao thức MQTT là giao thức ở tầng ứng dụng, sử dụng tầng giao vận là giao thức mà đáp ứng 3 yêu cầu đó là: các gói tin đến theo thứ tự, không bị mất mát và kết nối là kết nối 2 chiều (giống như các đặc điểm của giao thức tầng giao vận TCP/IP). Giao thức MQTT có 5 đặc trưng nổi bật đã tạo nên sự khác biệt so với các giao thức khác.

Thứ nhất, giao thức sử dụng mô hình publish/subscribe để chuyển tiếp dữ liệu. Mô hình này phù hợp với các ứng dụng cần chuyển tiếp thông báo từ một bên gửi tới nhiều bên nhận. Thứ hai, việc truyền gói tin từ publisher tới subscriber không liên quan tới nội dung của gói tin.

Thứ ba, giao thức MQTT phân ra 3 mức độ chất lượng dịch vụ cho quá trình truyền gói tin từ publisher tới subscriber. Với kiểu dịch vụ truyền nhiều nhất 1 lần, thông báo từ publisher sẽ được truyền đến subscriber, tuy nhiên nếu trong quá trình truyền gặp sự cố việc mất thông báo thì sẽ không có cơ chế truyền lại thông báo. Với kiểu dịch vụ truyền ít nhất một lần, thông báo từ publisher sẽ được đảm bảo truyền đến subscriber, tuy nhiên thông báo có thể được gửi đến subscriber nhiều hơn 1 lần do việc gửi lại thông báo khi gói tin xác nhận bị mất. Với kiểu dịch vụ truyền chính xác 1 lần, một thông báo từ publisher sẽ được gửi đến subscriber chính xác 1 lần.

Thứ tư, kích thước tiêu đề của các gói tin MQTT rất nhỏ (các trường cố định có kích thước là 2 byte) so với các giao thức ở tầng ứng dụng khác. Điều đó giúp cho giao thức này có thể hoạt động trong các môi trường mạng có tốc độ truyền thấp.

Thứ năm, cơ chế thông báo khi xảy ra việc kết nối bất thường từ thiết bị publish hoặc subscribe.

Giao thức MQTT hiện đang được sử dụng khá rộng rãi trong các ứng dụng IoT khác nhau như Amazon IoT, Microsoft Azure IoT Hub, ứng dụng chat Messenger Facebook,

2.3 Thiết kế IoT Platform

IoT platform của chúng tôi hướng đến việc phục vụ các nhà cung cấp dịch vụ IoT cho nhà thông minh (IoT service provider - IoT-SP). IoT-SP sẽ được cung cấp dữ liệu về trạng thái của các thiết bị trong nhà thông minh như dữ liệu cảm biến, dữ liệu về trạng thái hoạt động của các thiết bị đồng thời cho phép IoT-SP điều khiển các thiết bị trong nhà thông minh bằng cách thiết lập trạng thái mới của các thiết bị.

Vì vậy, chúng tôi đã thiết kế IoT platform dành cho nhà thông minh với các yêu cầu đặt ra như sau:

- Khả năng mở rộng của platform trong việc hỗ trợ các thiết bị công nghệ của các hãng khác nhau. Bản chất của vấn đề này nằm ở chỗ cần phải thiết lập một giao thức chuẩn cho tất cả các thiết bị IoT.
- Cho phép người dùng (chủ nhà) giới hạn việc truy cập dữ liệu và cấp quyền truy cập cho các IoT-SP hợp lệ.
- Khả năng lưu trữ dữ liệu trạng thái và chia sẻ dữ liệu để hỗ trợ cho các IoT-SP trong nhà thông minh.
- Hỗ trợ người phát triển phần mềm, dịch vụ IoT bằng việc cung cấp các API, hướng tới hình thành một hệ sinh thái IoT (Ecosystem IoT).

Việc xây dựng IoT platform cần giải quyết nhiều vấn đề về công nghệ khác nhau. Thứ nhất, để biết được thông tin, dữ liệu và sự thay đổi trạng thái của các thiết bị trong nhà thì mỗi thiết bị phải có cơ chế để lưu trữ thông tin về thiết bị của mình và có cách thức để giao tiếp giữa tác nhân muốn yêu cầu thông tin của thiết bị. Ngoài ra, trong quá trình trao đổi giữa hai bên, các thông số trao đổi cần phải được xác định và chứng thực được tác nhân yêu cầu.

Vấn đề thứ hai là người dùng có thể thêm thiết bị mới vào trong nhà mà không phải cài đặt lại hệ thống hiện tại trong nhà. Để giải quyết vấn đề này, đòi hỏi cần có cơ chế khám phá các thiết bị mới và sau khi phát hiện ra thiết bị mới cần thực hiện xử lý một số tác vụ như thông báo cho hệ thống và người dùng biết được việc thêm thiết bị đó thành công. Thiết bị mới thêm vào đòi hỏi phải chạy cùng hệ thống và không cần phải cài đặt lại toàn bộ hệ thống.

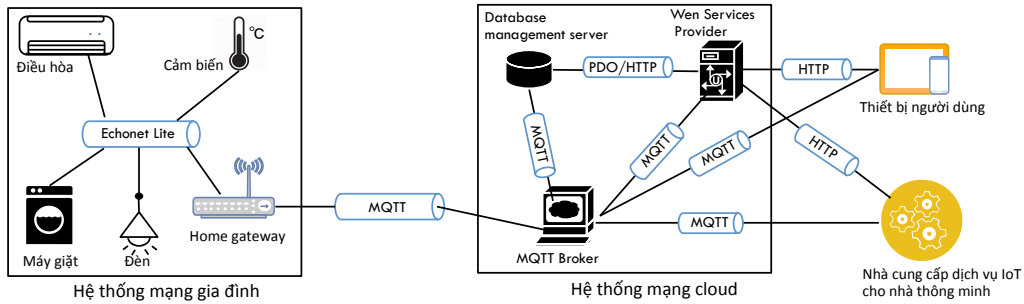
Vấn đề thứ ba, các IoT-SP có thể yêu cầu dữ liệu về trạng thái thiết bị trong nhà theo thời gian thực hoặc yêu cầu dữ liệu trong quá khứ. Điều đó đòi hỏi phải có cách thức lưu trữ dữ liệu trạng thái của thiết bị theo thời gian đồng thời cung cấp dữ liệu theo thời gian thực theo yêu cầu của các IoT-SP. Việc lưu trữ dữ liệu phải phù hợp với từng loại thiết bị, tức là mỗi thiết bị khác nhau sẽ cần lưu trữ lại các loại dữ liệu khác nhau. Ví dụ như dữ liệu của thiết bị điều hòa sẽ có cần lưu trữ lại các loại dữ liệu như chế độ cài đặt, tốc độ gió, hướng gió, nhiệt độ bên ngoài, Trong khi đó, thiết bị khóa điện sẽ cần lưu trữ lại các loại dữ liệu như: trạng thái có người trong phòng, trạng thái khóa đang mở hay đóng, ... Như vậy, để đáp ứng được vấn đề này thì cần có cơ sở dữ liệu lưu trữ phù hợp.

Để xây dựng được IoT platform đáp ứng được các yêu cầu đặt ra, chúng tôi thiết kế hệ thống với 3 thành phần chính (hình 3):

- Home Gateway là thiết bị có nhiệm vụ thu thập dữ liệu trạng thái của các thiết bị và gửi dữ liệu này đến máy chủ trên mạng Internet thông qua giao thức MQTT. Home Gateway sẽ giao tiếp với các thiết bị trong nhà thông qua giao thức Echonet Lite. Home Gateway cũng đóng vai trò của một MQTT client và giao tiếp với máy chủ đóng vai trò là MQTT broker thông qua giao thức MQTT.
- Hệ thống xử lý dữ liệu: có thể được triển khai trên hệ thống cơ sở hạ tầng đám mây, đóng vai trò là một MQTT broker để nhận dữ liệu từ các home gateway và lưu trữ vào cơ sở dữ liệu. Server này còn cung cấp các dữ liệu của các thiết bị trong nhà thông minh cho các IoT-SP theo thời gian thực để thực hiện các chức năng giám sát, điều khiển các thiết bị này.
- Máy chủ cung cấp dịch vụ: cung cấp các dịch vụ quản lý tài khoản và cung cấp dữ liệu trạng thái của thiết bị được lưu trữ trong cơ sở dữ liệu cho IoT-SP.

2.3.1 Thiết bị Home Gateway

Thiết bị Home Gateway được thiết kế để cung cấp bốn chức năng chính. Thứ nhất đó là chức năng phát hiện sự xuất hiện của thiết bị mới và các thiết bị đã tồn tại trong nhà thông minh để mục đích quét được hết tất cả thiết bị Echonet Lite trong nhà thông minh. Thứ hai là chức năng lắng nghe, giám sát sự thay đổi trạng thái của các thiết bị trong nhà. Thứ ba là chức năng



Hình 3 Mô hình hệ thống của IoT platform đề xuất

điều khiển các thiết bị trong nhà. Cuối cùng, thứ tư là gửi và nhận dữ liệu thông qua Internet dựa trên giao thức MQTT.

Chức năng thứ nhất được thực hiện dựa trên việc sử dụng chức năng yêu cầu thông tin, trạng thái của giao thức Echonet Lite. Các thiết bị Echonet Lite trong cùng một mạng luôn có một kênh truyền chung (a general broadcast) để giao tiếp với nhau. Kênh truyền này thường sẽ là một địa chỉ multicast ở cổng 3601. Như vậy, thiết bị Home Gateway được thiết kế để khi tham gia vào mạng xong thì sẽ bắt đầu yêu cầu thông tin của các thiết bị khác trong mạng và lắng nghe sự có mặt của thiết bị mới.

Chức năng thứ hai, Home Gateway có thể dễ dàng nhận được thông tin về trạng thái của thiết bị thông qua việc yêu cầu thông tin về trạng thái hiện tại của thiết bị. Do đó, chức năng giám sát trạng của các thiết bị Echonet Lite trong mạng từ thiết bị Home Gateway sẽ luôn thực hiện được tại bất kì thời điểm nào. Tùy từng loại thiết bị Echonet Lite khác nhau sẽ có các cách thiết kế để giám sát khác nhau. Điểm khác biệt ở các cách thiết kế giám sát đó là tần suất yêu cầu trạng thái từ Home Gateway đến thiết bị Echonet Lite đó. Có những thiết bị Echonet Lite mà thiết bị Home Gateway chỉ cần lắng nghe sự thay đổi trạng thái của nó trên kênh truyền, có những thiết bị thì Home Gateway thường xuyên gửi yêu cầu thông tin trạng thái tới, có những thiết bị được kết hợp cả hai cách trên. Như vậy, chức năng giám sát các thiết bị Echonet Lite của Home Gateway được thực hiện tùy thuộc vào đặc điểm của các thiết bị Echonet Lite trong mạng. Đồng thời, việc giám sát các thiết bị Echonet Lite trong nhà cũng cần đáp ứng được yêu cầu của IoT-SP. Vì vậy, tùy thuộc vào yêu cầu khác nhau của các ứng dụng IoT mà Home Gateway có thể gửi yêu cầu thông tin trạng thái của các thiết bị với tần suất khác nhau.

Với chức năng điều khiển các thiết bị Echonet Lite trong mạng, Home Gateway sẽ cập nhật trạng thái của các thiết bị bằng cách gửi thông báo yêu cầu cập nhật thông tin trạng thái theo chuẩn giao thức Echonet lite. Trên kênh truyền quảng bá, mọi thiết bị Echonet Lite đều có thể lắng nghe được thông điệp từ Home Gateway. Do đó, khi Home Gateway gửi thông tin lên kênh truyền thì tất cả các thiết bị trong mạng đều đọc được và trong đó có thiết bị Echonet Lite đích mà Home Gateway muốn thay đổi trạng thái. Thiết bị Echonet Lite nhận ra gói tin điều khiển dành cho mình sẽ thực hiện việc thay đổi trạng thái theo như nội dung trong thông báo.

Cuối cùng, chức năng gửi và nhận dữ liệu về trạng thái thiết bị trong nhà thông minh đến các máy tính trên hạ tầng điện toán đám mây qua mạng Internet dựa vào giao thức MQTT. Ở đây,

Home Gateway sẽ đóng vai trò của một MQTT client và trạng thái của thiết bị có thể là dữ liệu của các cảm biến hoặc trạng thái hoạt động của các thiết bị. Dữ liệu gửi qua Home Gate đến các thiết bị trong nhà là dữ liệu điều khiển thiết bị được gửi từ các IOT-SP. Trong phần thiết kế về thông điệp trao đổi với bên ngoài thông qua giao thức MQTT, định dạng dữ liệu sử dụng sẽ là định dạng JSON. Trong mỗi thông điệp trao đổi được thiết kế luôn bao gồm các 4 thuộc tính thông tin để xác định một thiết bị Echonet Lite trong nhà: mã nhóm thiết bị (group code), mã lớp thiết bị (class code), mã đối tượng thiết bị (instance code) và địa chỉ IP của thiết bị Echonet Lite trong mạng.

2.3.2 Hệ thống xử lý dữ liệu

Hệ thống xử lý dữ liệu đóng vai trò trung gian cho việc gửi nhận dữ liệu giữa Home Gateway và IoT-SP. Thành phần này được thiết kế bao với hai thành phần con.

Thành phần thứ nhất là MQTT Broker, đóng vai trò trung gian giúp cho Home Gateway trong nhà thông minh, máy chủ quản lý cơ sở dữ liệu và các IoT-SP có thể giao tiếp với nhau thông qua giao thức MQTT. Ở đây, Home Gateway đóng vai trò là MQTT client sẽ gửi dữ liệu trạng thái của các thiết bị trong nhà bằng cách gửi thông báo publish (xuất bản thông tin) theo một chủ đề nhất định đến MQTT Broker. Các IoT-SP cần cung cấp các thông tin, dữ liệu trạng thái của các thiết bị Echonet Lite trong nhà theo thời gian thực sẽ đóng vai trò là MQTT client để gửi yêu cầu subscribe (cung cấp thông tin) theo một chủ đề nhất định cho MQTT Broker. Khi cần gửi lệnh điều khiển một thiết bị trong nhà thông minh, IoT-SP sẽ gửi thông báo publish đến MQTT Broker theo một chủ đề mà Home Gateway đã đăng ký nhận thông tin (subscribe) từ trước. Để gán quyền publish và subscribe tới các chủ đề một cách phù hợp thì cần đòi hỏi phải có cách quản lý để chia topic một cách phù hợp đối với mỗi tài khoản.

Việc quản lý topic được thiết kế như sau: khi tài khoản với tên người dùng (chủ sở hữu ngôi nhà) là *A* thì sẽ được publish tới chủ đề *user/A/data/+* và subscribe tới chủ đề *user/A/command/+*. Tên tài khoản được yêu cầu là duy nhất, do đó đảm bảo phân biệt được các topic mà mỗi người dùng được phép publish hay subscribe. Nói cách khác, Home Gateway của người dùng *A* sẽ gửi dữ liệu trạng thái của các thiết bị trong nhà thông minh của người dùng *A* đến MQTT broker với chủ đề là *user/A/data/+*. Home Gateway của người dùng *A* cũng sẽ yêu cầu gửi thông tin (subscribe) tới chủ đề *user/A/command/+*. Như vậy, thông qua chủ đề này Home Gateway của người dùng *A* sẽ lắng nghe để nhận các yêu cầu điều khiển từ thiết bị của người dùng hoặc từ IoT-SP. Dấu +

trong chủ đề `user/A/data/` có ý nghĩa là mọi chủ đề có tiền tố là `user/A/data`.

Thành phần thứ hai là máy chủ quản lý cơ sở dữ liệu lưu trữ thông tin và dữ liệu trạng thái của các thiết bị trong nhà thông minh. Việc lưu trữ dữ liệu trạng thái là rất quan trọng với các ứng dụng IoT vì các dữ liệu này có thể được sử dụng để tối ưu hóa hoạt động của các dịch vụ hoặc tạo ra nhiều dịch vụ mà người dùng mong muốn, ví dụ như dịch vụ thống kê lượng điện tiêu thụ hàng tháng... Do đó cần phải có một cơ sở dữ liệu để lưu trữ dữ liệu trạng thái của các thiết bị Echonet Lite thông thường trong nhà thông minh theo thời gian. Vì dữ liệu trạng thái của các thiết bị được gửi tới MQTT Broker qua Home Gateway, do đó máy chủ quản lý cơ sở dữ liệu sẽ đóng vai trò là một MQTT client để gửi yêu cầu thông tin (subscribe) đến MQTT Broker. Dữ liệu trạng thái của các thiết bị trong nhà thông minh sẽ được gửi từ MQTT Broker đến máy chủ quản lý cơ sở dữ liệu và được lưu trong cơ sở dữ liệu.

2.3.3 Máy chủ cung cấp dịch vụ

Máy chủ cung cấp dịch vụ được thiết kế để đáp ứng các yêu cầu từ phía người dùng (chủ sở hữu nhà) và IoT-SP như yêu cầu đăng ký tài khoản sử dụng hệ thống, cấp quyền truy cập dữ liệu cho các tài khoản của IoT-SP, nhận danh sách các thiết bị trong nhà thông minh, ... Một Home Gateway hoặc một IoT-SP khi gửi yêu cầu publish hay yêu cầu subscribe tới một topic nào thì đều cần đến một tài khoản để có thể giao tiếp với MQTT Broker và tài khoản đó được phân quyền cho từng topic. Máy chủ cung cấp dịch vụ sẽ cung cấp dịch vụ Web để giúp người dùng có thể đăng ký tài khoản, phân quyền topic cho tài khoản,... thông qua giao thức HTTP.

Một IoT-SP cũng có thể gửi yêu cầu cung cấp dữ liệu trong quá khứ của các thiết bị trong nhà thông minh. Trong trường hợp này, máy chủ cung cấp dịch vụ sẽ gửi yêu cầu đến máy chủ quản lý cơ sở dữ liệu để yêu cầu cung cấp dữ liệu cho IoT-SP.

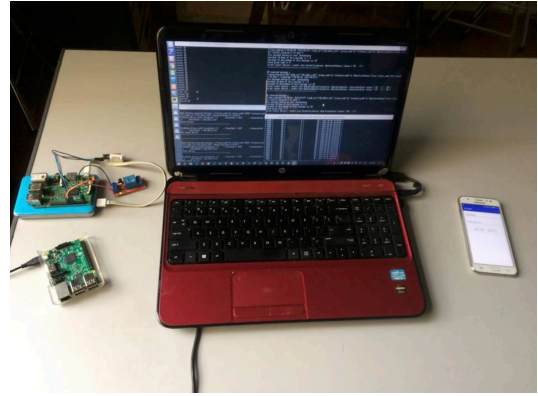
III. CÀI ĐẶT IoT PLATFORM CHO NHÀ THÔNG MINH

3.1 Môi trường cài đặt

Bước tiếp theo sau khi hoàn thành bản thiết kế hệ thống trên sẽ là việc cài đặt trên môi trường thật. Vì lẽ có 3 thành phần trong mô hình thiết kế, do đó môi trường cài đặt tương ứng với 3 thành phần này cũng khác nhau. Thành phần thứ nhất, bên trong nhà thông minh, môi trường cài đặt là trên các thiết bị thông thường và thiết bị Home Gateway. Thành phần thứ hai, phía server, môi trường cài đặt có thể dựa trên các hệ thống server dịch vụ hoặc cũng có thể chạy trên nền tảng cloud. Cuối cùng thứ ba là thành phần phía người dùng, môi trường cài đặt là trên các thiết bị được người dùng sử dụng như các thiết bị di động, thiết bị giám sát, dịch vụ nhận dữ liệu ...

Chúng tôi đã xây dựng và cài đặt các thành phần trong hệ thống như sau. Thiết bị Home Gateway được cài đặt trên máy tính nhúng Raspberry Pi. Để thiết bị Home Gateway gửi nhận các gói tin Echonet Lite, chúng tôi sử dụng thư viện mã nguồn mở được phát triển bằng ngôn ngữ Java do phòng nghiên cứu của công ty Sony phát triển.

P phía server, chúng tôi cài đặt 3 thành phần như trong thiết kế là MQTT Broker, máy chủ quản lý cơ sở dữ liệu và máy chủ cung



Hình 4 Hình ảnh triển khai trên thực tế của các thiết bị của dịch vụ giám sát và điều khiển

cấp dịch vụ. MQTT Broker sử dụng là phần mềm mã nguồn mở Mosquitto. Cơ sở dữ liệu sử dụng là MySQL, phiên bản sử dụng là 5.7. Chương trình máy chủ cung cấp dịch vụ phía người dùng được cài đặt bằng ngôn ngữ PHP, sử dụng server dịch vụ là Apache.

Để thử nghiệm các dịch vụ phía người dùng và IoT-SP, chúng tôi cài đặt trên 2 thiết bị đó là điện thoại thông minh chạy trên nền tảng Android và máy tính PC có thể nhận được dữ liệu tập trung. Chúng tôi sử dụng thư viện mã nguồn mở Eclipse Paho Android Service để xây dựng MQTT client trên điện thoại thông minh. Đây là một thư viện do Eclipse phát triển và có giấy phép EPL.

3.2 Cài đặt chương trình

Các chức năng mà chúng tôi đã xây dựng và cài đặt được bao gồm:

- Chức năng tự động nhận phát hiện các thiết bị trong mạng của Home Gateway
- Chức năng tự động nhận dữ liệu trạng thái từ các thiết bị Echonet Lite và gửi dữ liệu này tới MQTT Broker.
- Chức năng của cơ sở dữ liệu tự động thêm thông tin của một thiết bị mới vào trong danh sách các thiết bị Echonet Lite trong nhà thông minh của người dùng.
- Chức năng lưu trữ dữ liệu trạng thái của các thiết bị Echonet Lite trong cơ sở dữ liệu.
- Chức năng thêm người dùng mới khi người dùng thực hiện đăng ký thành công.
- Chức năng cung cấp thông tin và dữ liệu trạng thái của các thiết bị trong nhà khi nhận được yêu cầu từ các IoT-SP.

Các thiết bị Echonet Lite khi khởi tạo đã gửi được dữ liệu trạng thái của mình lên kênh truyền và thực hiện đúng các yêu cầu điều khiển sau khi nhận được yêu cầu điều khiển từ thiết bị Home Gateway.

3.2. Xây dựng ứng dụng giám sát và điều khiển

Chức năng giám sát và điều khiển các thiết bị trong nhà thông minh là một nhu cầu cấp thiết trong nhà thông minh. Ngoài cách giám sát, điều khiển thủ công, thì việc giám sát, điều khiển từ xa thông qua các thiết bị thông minh như điện thoại, máy tính là cách mà hiện nay các nhà thông minh đang hướng tới.

Chúng tôi đã xây dựng một ứng dụng giám sát hay điều khiển được các thiết bị trong nhà bằng thiết bị cầm tay là điện thoại thông minh Android. Ứng dụng này cho phép tự động cập nhật trạng thái của các thiết bị trong nhà thông minh và gửi yêu cầu điều khiển các thiết bị Echonet Lite trong nhà thông minh. Việc sử dụng IoT platform giúp cho việc xây dựng ứng dụng giám sát và điều khiển các thiết bị trong nhà được thực hiện một cách đơn giản và dễ dàng.

Trước hết, thiết bị Android được cài đặt phần mềm MQTT client và đăng kí một tài khoản MQTT client với MQTT broker. Để thực hiện được việc giám sát được các thiết bị trong nhà của mình thì thiết bị Android của người dùng (chủ sở hữu nhà thông minh) gửi yêu cầu cung cấp thông tin tới chủ đề `user/A/data/+` thông qua tài khoản của mình. Trong đó `A` là tên tài khoản của người dùng. Dữ liệu trạng thái của thiết bị được gửi từ Home Gateway trong nhà thông minh của người dùng tài khoản A đến MQTT Broker và được chuyển đến thiết bị Android của người dùng.

Để thực hiện được chức năng điều khiển thiết bị Echonet Lite trong nhà của mình thì thiết bị Android của người dùng A sẽ gửi dữ liệu điều khiển tới MQTT Broker với chủ đề `user/A/command/X`. Trong đó X là loại thiết bị trong nhà. Dữ liệu này sẽ được chuyển tới thiết bị Home Gateway của người dùng A. Sau đó, Home Gateway sẽ gửi thông báo điều khiển thiết bị Echonet Lite theo định dạng Echonet lite dựa vào thông báo MQTT nhận được.

Hình 4 là hình ảnh của một hệ thống giám sát và điều khiển đã triển khai trên thực tế, trong đó điện thoại thông minh của người dùng sẽ theo dõi các giá trị đo được của các cảm biến trong nhà tích hợp chuẩn Echonet Lite, đồng thời có thể điều khiển việc bật tắt đèn.

3.3 Xây dựng ổ điện thông minh sử dụng IoT platform

Trong ngôi nhà thông minh, tất cả các thiết bị muốn hoạt động đều phải có nguồn điện. Do đó, một ngôi nhà thông minh chắc chắn nó sẽ phải kiểm soát được các nguồn điện này. Người dùng thông qua việc kiểm soát sẽ quản lý được lượng điện năng tiêu thụ để thực hiện điều chỉnh sử dụng thiết bị cho hợp lý. Để thực hiện được điều này, cần phải có các thiết bị đo dòng điện gắn trên các thiết bị, bên cạnh một số thiết bị điện chỉ là thiết bị điện đơn thuần nên việc cho ra đời một ổ cắm thông minh thực hiện đo điện năng của các thiết bị đang hoạt động trên nó là cần thiết.

Ổ cắm thông minh Echonet Lite giúp người dùng theo dõi được lượng điện năng tiêu thụ cũng như phát hiện được các sự cố bất thường xảy ra với hệ thống điện trong ngôi nhà của mình. Dựa theo nhu cầu của một số người dùng mong muốn biết được số tiền mình phải thanh toán cho lượng điện năng mà mình đã sử dụng cũng như dự toán được chi phí sẽ phải thanh toán cho đến cuối tháng, ổ cắm thông minh có thể theo dõi điện năng và tính toán được chi phí này.

Chúng tôi đã xây dựng ứng dụng ổ điện thông minh dựa trên IoT platform với ổ điện thông minh là một thành phần thiết bị Echonet lite trong hệ thống (Hình 5). Ổ cắm điện Echonet lite của chúng tôi được cấu thành từ các thành phần phần cứng bao gồm:

- Board mạch nhúng Arduino Uno R3



Hình 5 Hình ảnh ổ điện thông minh đã triển khai

- Module Wifi esp8266
 - Module đo dòng điện dựa trên hiệu ứng Hall ACS712
- Ổ cắm điện Echonet Lite sẽ gửi dữ liệu công suất tức thời cho Home Gateway, sau đó Home Gateway gửi dữ liệu này lên MQTT Broker và máy chủ quản lý cơ sở dữ liệu với vai trò của một MQTT Client sẽ nhận và lưu dữ liệu này vào cơ sở dữ liệu. Chúng tôi cũng xây dựng một Web server hiển thị mức tiêu thụ năng lượng điện bằng cách truy cập vào cơ sở dữ liệu để lấy dữ liệu về công suất tiêu thụ điện tức thời, sau đó xử lý, tính toán và hiển thị thông tin cho người dùng.

IV. KẾT LUẬN

Trong báo cáo này, chúng tôi đề xuất một IoT platform có khả năng cung cấp các chức năng giao tiếp với các thiết bị cũng như lưu trữ thông tin trạng thái và cập nhật trạng thái cho các thiết bị cho các ứng dụng trong nhà thông minh. Chúng tôi đã xây dựng và cài đặt IoT platform, đồng thời xây dựng thử nghiệm một ứng dụng ổ điện thông minh để chứng tỏ khả năng ứng dụng của IoT platform của chúng tôi.

LỜI CẢM ƠN

Công trình này nhận được sự tài trợ từ đề tài nghiên cứu khoa học công nghệ cấp ĐHQG Hà Nội, mã số QG.16.30.

TÀI LIỆU THAM KHẢO

- [1] ECHONET Consortium, <http://www.echonet.gr.jp/>, accessed 10th Jan 2017.
- [2] IEC, *Service diagnostic interface for consumer electronics products and networks Implementation for ECHONET*, IEC 62394, April 2017.
- [3] ISO/IEC JTC1, *information technology — Home Electronic Systems (HES) architecture — Part 4-3: Application layer interface to lower communications layers for network enhanced control devices of HES Class 1*, ISO/IEC 14543-4-3, 2015.
- [4] Andrew Banks and Rahul Gupta, <http://docs.oasisopen.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.pdf>, OASIS Standard, 29 September 2014, pp. 1- 2.
- [5] ISO/IEC JTC1, *Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1*, ISO/IEC 20922, 2016, pp. iii.
- [6] <https://github.com/SonyCSL/OpenECHO>
- [7] <https://github.com/eclipse/paho.mqtt.android>