

THIẾT KẾ VÀ THỰC THI MÔ HÌNH GATEWAY CHO CÁC THIẾT BỊ IoTS

DESIGNING AND IMPLEMENTING A GATEWAY MODEL FOR IoTS DEVICES

Phan Văn Ca

Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh; capv@hcmute.edu.vn

Tóm tắt - Trong mạng lưới vạn vật kết nối IoTs (Internet of Things), các thiết bị phải thực hiện kết nối và truyền thông theo các phương thức như M2M (Machine-to-Machine) hay D2D (Device-to-Device). Thông thường, các thiết bị này là thiết bị nhúng có phần cứng không đồng nhất và sử dụng các chuẩn truyền thông khác nhau. Vì vậy, chúng không thể tự thiết lập kết nối và giao tiếp với nhau. Bài báo này đề xuất một mô hình Gateway được thiết kế với lớp đệm trung gian có chức năng chuyển đổi qua lại giữa các chuẩn truyền thông khác nhau qua một chuẩn trung gian, giúp cho chúng có thể dễ dàng kết nối với nhau và kết nối tới Internet. Mô hình thực thi được thiết kế hỗ trợ ba công nghệ truyền thông phổ biến bao gồm Zigbee, Bluetooth và Wifi, giao tiếp với nhau qua chuẩn trung gian Ethernet.

Từ khóa - vạn vật kết nối; giao thức truyền thông không dây; truyền thông máy – máy; mô hình công nối; kết nối thiết bị IoTs.

Abstract - In the internet of things environment, most devices have ability to connect and communicate with others to transfer data in a Machine-to-Machine (M2M) or Device-to-Device (D2D) manner. In general, most of them are heterogeneous embedded devices that may be implemented by different communication protocols or technologies. Therefore, they cannot connect and communicate to each other directly. This paper proposes an IoTs Gateway model with intermediate layer that abstracts different communication protocols or standards from IoTs devices into an intermediate standard. The proposed Gateway can connect with different input and output wireless technology interfaces including Zigbee, Bluetooth and Wifi via Ethernet interface serving as intermediate layer.

Key words - Internet of things; wireless communication protocol; M2M communication; Gateway model; connecting IoTs devices.

1. Giới thiệu

Hiện nay, các thiết bị thông minh xuất hiện trong hầu hết mọi lĩnh vực, từ điện tử tiêu dùng cho đến sản xuất công nghiệp. Các mô hình ứng dụng IoTs (Internet of Things) đang được nghiên cứu và triển khai ngày càng nhiều. IoTs là một hệ thống được cấu thành từ các thiết bị nhúng thông minh có thể thu thập dữ liệu từ môi trường thế giới thực, kết hợp xử lý dữ liệu từ 2 thế giới thực và ảo nhằm cung cấp các thông tin hữu ích, cũng như đưa ra các quyết định trong điều khiển từ xa qua mạng Internet và khả năng tự vận hành hệ thống của hệ thống trong phạm vi cục bộ hoặc qua điện toán đám mây [1], [2].

Thiết bị trong mạng lưới IoTs là những thiết bị thông minh với khả năng cảm nhận môi trường xung quanh, thu thập dữ liệu, tương tác với thế giới thực qua các cảm biến và cơ cấu chấp hành, đồng thời giao tiếp với các đối tượng khác nhau thông qua các chuẩn truyền thông [3]. Bên cạnh các thiết bị thông minh được điều khiển bởi con người qua mạng internet, còn có các thiết bị tự hành, có thể giao tiếp với nhau và tự đưa ra các quyết định mà không cần sự can thiệp của con người [4]. Trong bài báo [5], các tác giả đã thiết kế một mạng cảm biến IoTs với các cảm biến siêu âm để thiết lập một hàng rào điện tử bằng sóng siêu âm, hoạt động dựa trên hiệu ứng Doppler để tự động phát hiện đối tượng di chuyển trong một phạm vi rộng lớn. Một tính chất quan trọng của hệ thống IoTs chính là khả năng hoạt động độc lập, tiêu thụ năng lượng thấp. Trong bài báo [6], các tác giả đã đề xuất một phương pháp mới cho phép truyền đa kênh và định tuyến cho mạng hỗn hợp, với mỗi thiết bị là một nút mạng được trang bị các cảm biến hình ảnh để phát hiện các sự kiện theo thời gian thực. Các thiết bị đầu cuối sử dụng nguồn pin và có thể thiết lập kết nối không dây diện rộng. Ngoài ra, các tác giả cũng đề xuất giao thức định tuyến và chọn kênh tối ưu nhằm giảm thiểu điện năng tiêu thụ.

Tuy nhiên, các mô hình thiết kế IoTs hiện nay chưa được thiết kế theo một chuẩn thống nhất, đặc biệt là chuẩn truyền thông để kết nối các thiết bị với nhau và kết nối tới

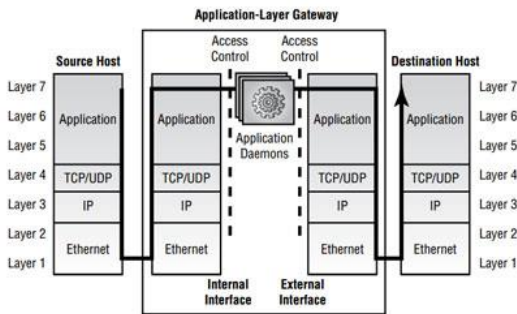
Internet [7]. Cụ thể, các giao thức truyền thông không dây trong các hệ thống IoTs hiện nay sử dụng các công nghệ sẵn có như Zigbee, Bluetooth hay Wifi. Việc này sẽ làm cho các hệ thống không thể tự giao tiếp để truyền nhận dữ liệu với nhau. Giải pháp cho vấn đề này là thiết kế một cổng nối Gateway để giúp cho các thiết bị có các chuẩn truyền thông khác nhau có thể kết nối với nhau và kết nối tới Internet qua một lớp đệm trung gian. Gateway này đóng vai trò như một thiết bị chuyển đổi chuẩn dữ liệu, đồng thời là một cổng kết nối tới mạng Internet, nhằm khai thác các ứng dụng và lưu trữ của điện toán đám mây. Thiết kế này sẽ giúp hệ thống IoTs với các chuẩn truyền thông khác nhau có thể dễ dàng kết nối với nhau và kết nối đến mạng Internet với số lượng phần tử và khả năng lưu trữ không giới hạn.

Bài báo này đề xuất một kiến trúc Gateway đóng vai trò như một thiết bị cầu nối giúp cho các thiết bị IoTs được thiết kế với các chuẩn truyền thông khác nhau có thể giao tiếp với nhau một cách dễ dàng. Bên cạnh đó, bài báo còn đề xuất một lớp chuyển đổi trung gian giúp cho việc mở rộng các tiêu chuẩn được hỗ trợ bởi Gateway một cách nhanh chóng, mà không cần thay đổi các lớp phía trên. Ngoài ra, bài báo còn đề xuất một mô hình quản lý các mạng IoTs với Gateway kết nối tới Internet thông qua các server đám mây, giúp cho các hệ thống IoTs có thể được truy xuất từ các server khác nhau. Thiết kế này cho phép các thiết bị bên dưới có thể thực hiện việc truyền nhận dữ liệu ngang hàng với nhau và kết nối tới nhiều server khác nhau trên mạng Internet.

2. Mô hình Gateway IoTs

Một vài mô hình Gateway IoTs đã được công bố trong thời gian gần đây [8]-[10]. Mô hình Gateway trong bài báo [8] dựa trên các giao thức Zigbee và GPRS cho các ứng dụng IoTs điển hình kết nối giữa mạng cảm biến không dây Zigbee và mạng thông tin di động. Bài báo [9] giới thiệu mô hình Gateway cho các mạng công suất thấp. Mô hình

Gateway trong bài báo [10] được phát triển để kết nối giữa các thiết bị thông minh với các thiết bị thông thường qua giao thức modbus. Tuy nhiên, các mô hình Gateway này là những mô hình Gateway đơn giao thức và thực hiện chuyển đổi dữ liệu ở các lớp cao, như mô tả ở Hình 1. Bài báo [11] đề xuất mô hình Gateway có chức năng thu thập dữ liệu từ các thiết bị cảm biến kết nối qua các công nghệ truyền thông GPRS, WiFi, RF, Bluetooth hoặc LAN qua một máy tính kết nối tới Internet. Các tác giả trong [12] đã đề xuất một mô hình Gateway thông minh, có chức năng xử lý dữ liệu trước khi gửi lên đám mây cho ứng dụng tích hợp giữa IoT và điện toán đám mây, được gọi là CoT (Cloud of Things).

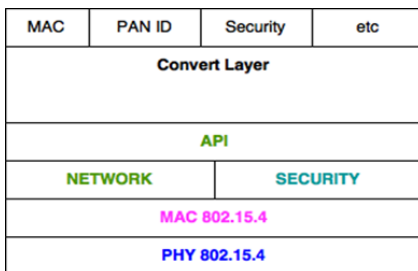


Hình 1. Mô hình Gateway Ethernet ở lớp ứng dụng

Mô hình đề xuất trong bài báo thực hiện chuyển đổi ở các lớp thấp để giảm độ trễ cũng như cải thiện tốc độ xử lý của Gateway. Trong đó, lớp phía trên truy xuất dữ liệu ở lớp thấp hơn thông qua một lớp đệm được xây dựng từ các API tương ứng với các phần cứng giao tiếp khác nhau. Mô hình kiến trúc trên được thực thi dựa trên 4 công nghệ thường gặp là Zigbee, Bluetooth, Wifi, Ethernet với các phần cứng tương ứng cho Gateway là Xbee, HC-05, ESP8266 và ENC28J60.

2.1. Giao thức Zigbee

Giao thức Zigbee là một giao thức được sử dụng rộng rãi ở các hệ thống IoTs bởi khả năng quản lý các thiết bị và độ ổn định của hệ thống. Để có thể giao tiếp được với một thiết bị Zigbee thông qua Gateway, cần phải phát triển một API với phần cứng tương ứng. Từ đó xây dựng một lớp đệm để truy xuất thông tin thông qua API và tái cấu trúc tại lớp bên trên, như mô tả ở Hình 2.



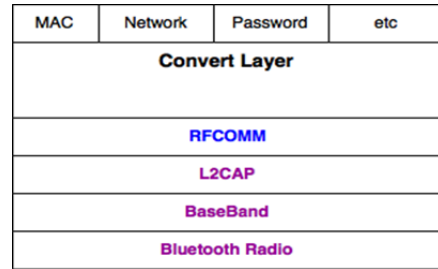
Hình 2. Lớp đệm Zigbee

Thiết kế này cho phép truy xuất thông tin cũng như sử dụng tài nguyên mạng Zigbee cho các thiết bị khác.

2.2. Giao thức Bluetooth

Đối với các ứng dụng IoTs với các phần tử là các thiết bị di động thì Bluetooth đóng vai trò quan trọng bởi khả năng thiết lập kết nối và tốc độ ổn định. Đối với giao thức

này, việc chuyển đổi dữ liệu phía trên lớp RFCOMM như mô tả trong Hình 3.

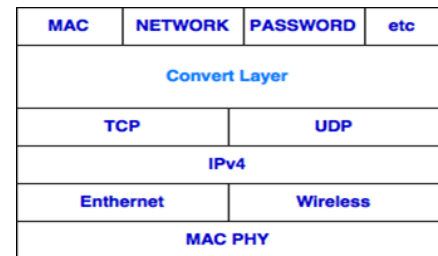


Hình 3. Lớp đệm Bluetooth

Thông thường, trong các hệ thống IoTs, độ ưu tiên của hệ thống không phải là tốc độ truyền tải mà là độ ổn định và đáp ứng thời gian thực, việc truy xuất trên lớp RFCOMM cho tốc độ đáp ứng và độ ổn định cao hơn.

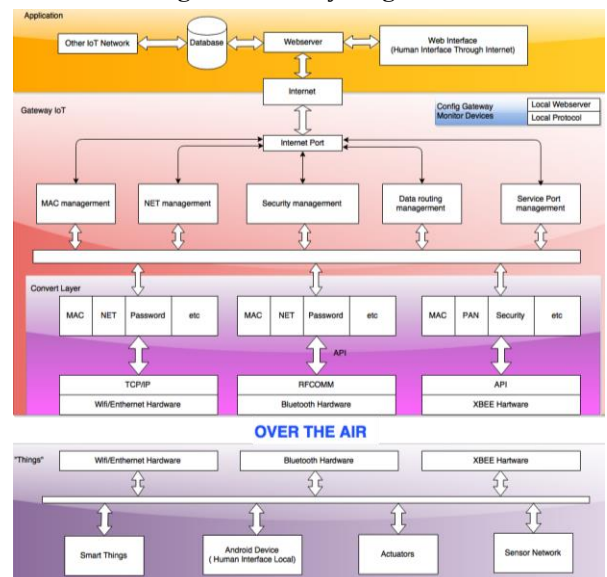
2.3. Giao thức Wifi và Ethernet

Trong các ứng dụng ngôi nhà thông minh, đa số các hệ thống IoTs được phát triển dựa trên các hạ tầng mạng sẵn có như Wifi hay Ethernet. Lớp chuyển đổi được thực hiện trên các giao thức truyền vận TCP/UDP, như mô tả trong Hình 4. Từ đây, ta có thể thực hiện các truy vấn mà TCP/IP cho phép, cũng như các phương thức trong giao thức HTTP như GET, POST trên các server Internet. Mô hình này cho phép việc tải lên/xuống dữ liệu từ đám mây một cách dễ dàng.



Hình 4. Lớp đệm WIFI/Ethernet

3. Kiến trúc tổng thể Gateway đa giao thức



Hình 5. Kiến trúc tổng thể Gateway

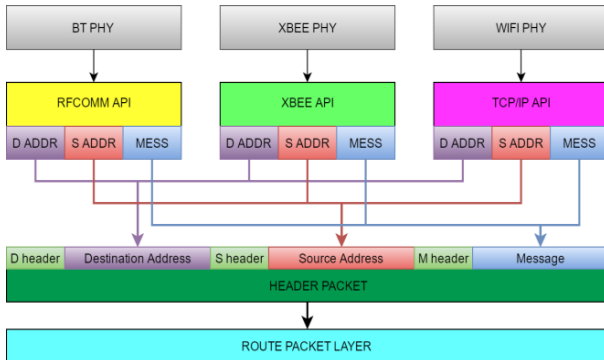
Kiến trúc tổng thể của Gateway đa giao thức được mô

tả như trong Hình 5. Kiến trúc này kết hợp các mô hình gateway đơn giao thức dựa trên lớp đệm đã đề xuất. Gateway trích xuất gói tin từ các mạng IoTs với các chuẩn kết nối khác nhau, thực hiện chuyển đổi và sau đó định tuyến gói tin liên mạng tới một mạng IoTs khác. Ngoài ra, Gateway còn thực hiện quản lý và giám sát các thiết bị hỗn tạp trong mạng.

Mô hình Gateway được thực hiện gồm 3 phần cứng hỗ trợ 3 công nghệ truyền thông khác nhau, bao gồm: Zigbee, Bluetooth và Wifi. Lớp chuyển đổi được xây dựng từ các API phần cứng tương ứng để thực hiện việc chuyển đổi chuẩn dữ liệu từ một mạng IoTs nguồn sang chuẩn trung gian sử dụng công nghệ Ethernet. Tiếp theo, dữ liệu theo chuẩn trung gian sẽ được chuyển sang chuẩn dữ liệu phù hợp với mạng IoTs đích trong kết nối liên mạng hoặc truyền trực tiếp lên mạng Internet.

3.1. Khối đóng gói dữ liệu

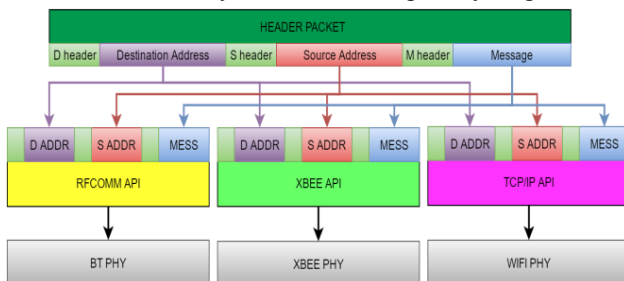
Khối đóng gói dữ liệu của Gateway đóng vai trò tách các thông tin cần thiết thông qua API, sau đó thực hiện đóng gói theo chuẩn mới để chuẩn bị định tuyến dữ liệu trong lớp tiếp theo. Quá trình trích xuất và đóng gói dữ liệu được mô tả như trong Hình 6.



Hình 6. Sơ đồ đóng gói dữ liệu

3.2. Khối định tuyến

Ở lớp này, thực hiện trích xuất địa chỉ từ lớp bên trên để đưa vào phần cứng phù hợp và đóng gói phù hợp với API tương ứng để có thể thực hiện truyền dữ liệu. Mỗi tiêu chuẩn có cách thức đóng gói và header khác nhau. Sau đó, các gói tin sẽ được chuyển về định dạng phù hợp với mạng IoTs đích và thực hiện việc truyền dữ liệu đến lớp vật lý để gửi đi.



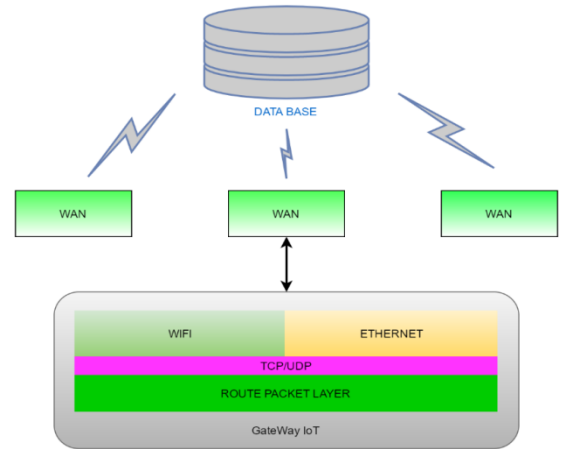
Hình 7. Mô hình định tuyến dữ liệu

Ngoài việc định tuyến dữ liệu giữa các tiêu chuẩn với nhau còn thực hiện nhiệm vụ như một cổng kết nối với mạng WAN, Internet thông qua cổng Ethernet.

3.3. Kết nối WAN và cơ sở dữ liệu

Mặc dù chuẩn Wifi cũng có thể làm nhiệm vụ này thay

cho chuẩn Ethernet, tuy nhiên, vì các hệ thống IoTs dựa trên Wifi thường được thiết kế tối giản và có thời gian đáp ứng chậm, đồng thời thiếu độ ổn định khi truyền nhận các gói tin trong mạng. Kết nối Ethernet này giúp Gateway có thể kết nối với mạng WAN hay Internet để đưa dữ liệu của toàn bộ hệ thống lên các server hoặc ngược lại. Các server được thiết kế có thể được chia sẻ cho nhiều Gateway, đồng thời cũng cho phép các server liên kết với nhau thành một mạng lưới server trên mạng Internet.



Hình 8. Kết nối mạng WAN, Internet và cơ sở dữ liệu

3.4. Lớp trình API

Thông qua các hàm API, các Client của hệ thống IoTs có thể truy xuất tới các phần tử khác trong mạng theo các chuẩn khác nhau và cho phép các phần tử trong mạng có thể đưa dữ liệu lên trên cơ sở dữ liệu, hay thực hiện các giao thức thông qua cổng Ethernet bằng các phương thức trong HTTP như GET/POST dữ liệu tới một Server bất kỳ trên mạng.

Chiều dữ liệu	Cấu trúc Header truyền	Cấu trúc Header nhận
BT -> XBEE	[a]add[b]des[c]package	source[c]package
XBEE -> BT	[a]des[b]mess	[a]source[b]package
BT -> ESP	[a]add[b]des[c]package	[b]source[c]package
ESP -> BT	[a]des[b]package	[a]source[b]package
ESP -> XBEE	[a]des[b]package	source[c]package
XBEE -> ESP	[a]des[b]package	source[c]package
XBEE -> ETH	[a]des[b]port[c]package	package
BT -> ETH	[a]add[b]des[c]port[d]package	package
ESP -> ETH	[a]des[b]port[c]package	package
ESP -> WAN	[w]Data Post	PROTOCOL WAN IP ADDRESS DATA
XBEE -> WAN	[w]Data Post	PROTOCOL WAN IP ADDRESS DATA
BT -> WAN	[w]Data Post	PROTOCOL WAN IP ADDRESS DATA

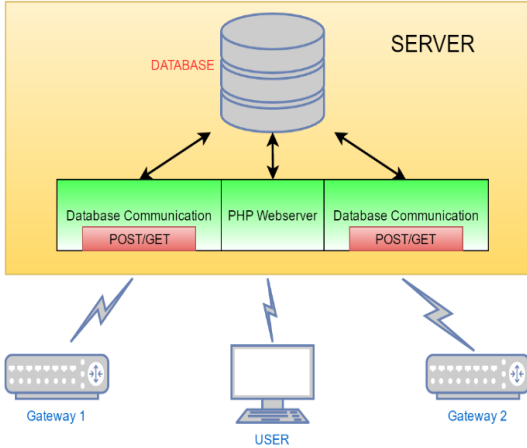
Hình 9. Các API cơ bản của Gateway

3.5. Mô hình server

Mô hình thử nghiệm này sử dụng MySQL chạy trên Server Linux vì tính phổ biến và khả năng hỗ trợ của nó. Mô hình thử nghiệm này sử dụng các thông số cơ bản liên quan đến dữ liệu. Tuy nhiên, trong thực tế, hệ thống cần có nhiều thông số khác như trạm Gateway, thông số Port, tài khoản truy cập và các thông số bảo mật.

Phía Server được cài đặt một lớp đệm trung gian cho phép việc chuyển đổi truy vấn các phương thức GET/POST để có thể kết nối và cập nhật cơ sở dữ liệu. Tuy nhiên, lớp này chạy trên môi trường Apache, PHP và MySQL. Ngoài ra, còn có một giao diện người dùng để truy xuất và hiển thị các thông tin từ Server. Từ đây các thông tin về địa chỉ IP của Gateway, giao thức truyền thông của mạng IoTs

nguồn, địa chỉ mạng cục bộ của thiết bị có thể là địa chỉ IP hay địa chỉ MAC tùy theo tính chất của ứng dụng cần xây dựng, có thể được tham chiếu và hiển thị dễ dàng. Dữ liệu trên server có thể được truy xuất từ bất kì thiết bị nào hỗ trợ giao thức HTTP, hoặc các giao thức liên kết trực tiếp tới database của server. Từ đó, ta có thể xây dựng một hệ thống IoTs với số phần tử và khoảng cách giữa chúng gần như không giới hạn.



Hình 10. Sơ đồ kết nối Gateway - Server

4. Kết quả

Mô hình phần cứng của Gateway đã được thi công, đồng thời thực hiện các kiểm tra và đánh giá các thông số cũng như đặc tính kỹ thuật. Đối với từng giao thức, tốc độ toàn cục của Gateway phụ thuộc vào cả tốc độ phần cứng xử lý giao thức và tốc độ xử lý của CPU trên Gateway. Tốc độ phần cứng thiết bị và CPU với các công nghệ khác nhau được liệt kê trong Bảng 1.

Bảng 1. Tốc độ xử lý của mô-đun phần cứng truyền thông và tốc độ của CPU trên Gateway

Công nghệ	Mô-đun truyền thông	Tốc độ tối đa mô-đun truyền thông	Tốc độ tối đa của CPU trên Gateway
Bluetooth	HC-05	115.200 bps	921.600 bps
Wifi	ESP8266	230.400 bps	921.600 bps
Zigbee	Xbee	115.200 bps	921.600 bps
Ethernet	ENC28J60	10 Mbps	2 Mbps

Mô hình Gateway thực hiện truyền thành công giữa các tiêu chuẩn với nhau, đồng thời thực hiện các truy vấn cơ sở dữ liệu và truy xuất mạng WAN với khả năng đáp ứng tốt, với tỷ lệ nhận gói tin thành công được mô tả trong Bảng 2.

Bảng 2. Đánh giá độ tin cậy của mô hình Gateway

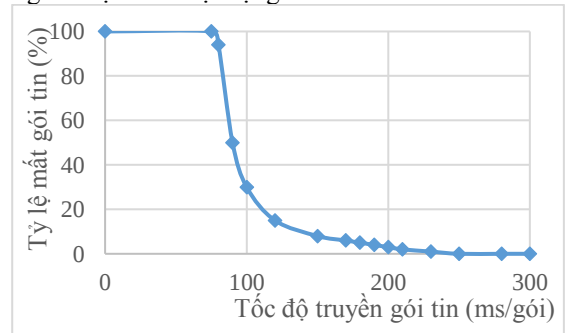
Nguồn	Đích	Số gói tin truyền	Số gói tin nhận	Tỷ lệ thành công (%)
Wifi	Zigbee	100	92	92
Zigbee	Wifi	100	99	99
Wifi	Bluetooth	100	100	100
Bluetooth	Wifi	100	100	100
Zigbee	Bluetooth	100	95	95
Bluetooth	Zigbee	100	92	92

Đối với việc truyền từ thiết bị Wifi (mô-đun ESP) sang thiết bị Zigbee (mô-đun XBEE), nếu thời gian giữa 2 gói tin nhỏ hơn 50 ms thì sẽ xảy ra hiện tượng mất gói tin vì phần

cứng của mô-đun ESP không đủ khả năng đáp ứng, cũng như việc tràn bộ đệm nhận dữ liệu trong lúc CPU của Gateway đang thực hiện, như mô tả trong Hình 11. Đối với việc truyền từ thiết bị Zigbee sang thiết bị Wifi, khi thời gian truyền giữa 2 gói tin nhỏ hơn 80 ms, sẽ xảy ra hiện tượng mất gói tin vì thời gian đóng gói gói tin theo giao thức TCP của phần cứng mô-đun ESP không thực hiện hoàn tất trước khi nhận được gói tin mới, CPU phụ trách việc giải mã lệnh API của thiết bị Zigbee tiêu tốn tài nguyên và mất nhiều thời gian xử lý. Không như truyền giữa thiết bị Zigbee và Wifi, việc truyền giữa thiết bị Bluetooth và thiết bị Wifi có độ tin cậy cao hơn với chiều dài gói tin sau đóng gói ngắn hơn và đạt tốc độ cao hơn, như mô tả trong Hình 12. Bộ đệm của mô-đun Bluetooth lớn và tích hợp bộ lưu trữ dữ liệu ngay cả khi mất kết nối giữa chúng.

Việc truyền giữa thiết bị Zigbee và Bluetooth một lần nữa cho thấy khả năng đáp ứng của CPU dành cho việc giải mã API khá hạn chế, dẫn đến việc xử lý dữ liệu tại CPU trên phần cứng thiết bị Zigbee bị nghẽn và gây tràn dữ liệu. Điều này dẫn đến việc mất gói tin khi thời gian đáp ứng nhỏ hơn 150 ms. Lý do chính là các thiết bị Zigbee là những thiết bị công suất thấp, được thiết kế dành cho các ứng dụng yêu cầu tốc độ truyền thấp và không đòi hỏi hiệu năng xử lý cao.

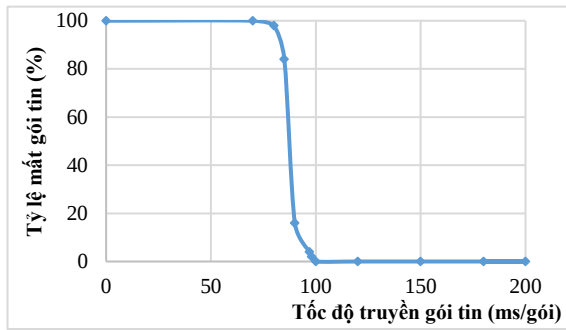
Tuy nhiên, đối với các ứng dụng IoTs hiện nay, điểm cần quan tâm chính là khả năng hoạt động ổn định và tiêu tốn năng lượng thấp, thì Gateway IoTs được xây dựng với phần cứng trên vẫn đáp ứng tốt nhu cầu sử dụng trong các hệ thống IoTs. Việc đáp ứng được nhu cầu của mạng cảm biến, thu thập thông tin, cũng như giám sát và điều khiển đối với các ứng dụng không đòi hỏi nghiêm ngặt về tính thời gian thực khi hoạt động.



Hình 11. Tỷ lệ mất gói tin khi kết nối giữa thiết bị Zigbee và Wifi

Đối với việc định tuyến dữ liệu cho luồng từ thiết bị Bluetooth sang thiết bị Zigbee, có thêm một yếu tố nữa dễ tăng khả năng mất gói tin giữa 2 chuẩn này, chính là việc phần cứng thiết bị Zigbee được thiết kế bên Gateway đóng vai trò là điều phối viên (coordinator), nên thời gian công tác của phần cứng còn bao gồm cả việc định tuyến dữ liệu trong các phần tử bên trong mạng thiết bị Zigbee nói riêng. Điều này dẫn đến việc định tuyến dữ liệu bị trễ và gây nghẽn cổ chai tại đây, làm mất gói tin giữa những lần truyền dữ liệu. Tuy nhiên, thiết bị Zigbee cũng có bộ sửa sai và tự động truyền lại, nhưng với việc bộ đệm có giới hạn và dữ liệu luôn được đưa vào thì việc mất gói tin là không thể tránh khỏi. Vì khả năng đáp ứng của phần cứng thiết bị Bluetooth HC-05 có độ trễ thấp hơn, nên gói tin chỉ bị ảnh hưởng bởi khả năng xử lý của CPU cũng như giới hạn tối đa của thiết bị thiết bị Zigbee. Ở đây, điểm nghẽn là do thiết bị Zigbee ở Gateway đóng vai trò coordinator

nên không dành toàn thời gian để tiếp nhận dữ liệu, mà còn dùng để quét tìm kiếm các phần tử mới, cấp phát và quản lý địa chỉ.



Hình 11. Biểu đồ Packet Loss của Xbee và Bluetooth

Ở đây, ta thấy có các vị trí biến thiên không đều do không đồng bộ trong chu kì quét nhận dữ liệu, cũng như quản lý các thiết bị con của từng tiêu chuẩn khác nhau, làm cho hệ số mất gói tin tăng cao và không tuyến tính ở một số khoảng thời gian. Ở một số khe thời gian cho phép việc truyền với tỉ số lỗi thấp và không đồng đều. Xác suất giảm về 0 khi khoảng cách giữa 2 dữ liệu lớn hơn 300 ms.

5. Kết luận

Bài báo trình bày về kiến trúc và mô hình thực thi cho một Gateway IoTs, cho phép các thiết bị IoTs có thể truy xuất ngang hàng và chia sẻ tài nguyên của bản thân cho toàn bộ hệ thống, đồng thời đề xuất các lớp tiêu chuẩn để có thể mở rộng hệ thống IoTs. Ngoài ra, mô hình Gateway còn đóng vai trò như một cổng kết nối Internet cho phép các mạng IoTs giao tiếp với nhau thông qua mạng toàn cầu. Bài báo đưa ra các đánh giá tương quan giữa tốc độ xử lý và tốc độ phản ứng, cho phép xây dựng một mô hình Gateway thực tế với các điều kiện lựa chọn thông số một cách chính xác hơn. Mở rộng hệ thống với các tiêu chuẩn khác để dàng với việc chỉ cần thiết lập một lớp chuyển đổi dữ liệu theo chuẩn đề xuất và ghép với phần còn lại của Gateway. Với sự phát triển của trí thông minh nhân tạo và các hệ thống M2M, hiện nay đã có đủ khả năng xử lý thông tin, và với một lượng thông tin đủ lớn thì hệ thống IoTs sẽ vận hành mà không cần bất kì sự can thiệp nào của con người. Hệ thống sẽ tự động

cập nhật và hiệu chỉnh các thông số để có thể đáp ứng nhu cầu ngày càng cao của con người.

TÀI LIỆU THAM KHẢO

- [1] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey", *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, First Quarter 2014, pp. 414-454.
- [2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 4, FourthQuarter 2015, pp. 2347-2376.
- [3] G. Kortuem, F. Kawsar, V. Sundramoorthy and D. Fitton, "Smart objects as building blocks for the Internet of things", *IEEE Internet Computing*, Vol. 14, No. 1, Jan-Feb 2010, pp. 44-51.
- [4] M. A. Razzaque, M. Mилоjevic-Jevric, A. Palade and S. Clarke, "Middleware for Internet of Things: A Survey", *IEEE Internet of Things Journal*, Vol. 3, No. 1, Feb. 2016, pp. 70-95.
- [5] H. T. Chan, T. A. Rahman and A. Arsad, "Performance study of virtual fence unit using Wireless Sensor Network in IoT environment", *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, Hsinchu, 2014, pp. 873-875.
- [6] HyungWon Kim, "Low power routing and channel allocation method of wireless video sensor networks for Internet of Things (IoT)", *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Seoul, 2014, pp.446-451.
- [7] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. Mccann and K. K. Leung, "A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities", *IEEE Wireless Communications*, Vol. 20, No. 6, December 2013, pp. 91-98.
- [8] Q. Zhu, R. Wang, Q. Chen, Y. Liu and W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things", *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, Hong Kong, 2010, pp. 347-352.
- [9] T. Kadioglu, N. Yildiz and E. Cesur, "Low power Internet of Things Gateway", *2016 24th Signal Processing and Communication Application Conference (SIU)*, Zonguldak, 2016, pp. 1613-1616.
- [10] Datta S.K, "An IoTs Gateway centric architecture to provide novel M2M services", *Internet of Things (WF-IoTs)*, *2014 IEEE World Forum*, 6-8 March 2014, pp. 514 - 519.
- [11] K. Rajaram and G. Susanth, "Emulation of IoT gateway for connecting sensor nodes in heterogenous networks", *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, Chennai, 2017, pp. 1-5.
- [12] M. Aazam, P. P. Hung and E. N. Huh, "Smart gateway based communication for cloud of things", *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, 2014, pp. 1-6.

(BBT nhận bài: 14/07/2017, hoàn tất thủ tục phản biện: 23/08/2017)