

ĐÁNH GIÁ HIỆU NĂNG GIAO THỨC MQTT VÀ HTTP TRONG MỘT HỆ THỐNG IoT THỜI GIAN THỰC

PERFORMANCE EVALUATION OF MQTT AND HTTP PROTOCOLS IN A REAL TIME IoT SYSTEM

Thái Vũ Hiền*

Trường Đại học Bách khoa – Đại học Đà Nẵng¹

*Tác giả liên hệ: tvhien@dut.udn.vn

(Nhận bài: 08/11/2021; Chấp nhận đăng: 15/12/2021)

Tóm tắt - Sự phát triển nhanh chóng của Internet of Things (IoT) đã khiến công nghệ này trở thành một trong những mô hình hứa hẹn nhất trong những năm gần đây. Để đảm bảo thực hiện hệ thống theo thời gian thực là một trong các vấn đề cần quan tâm trong các ứng dụng IoT. Trong nghiên cứu này, hai trong số các giao thức phổ biến nhất được sử dụng trong hệ thống IoT thực tế là MQTT và HTTP được trình bày, đánh giá bằng các thử nghiệm để đánh giá hiệu suất của hai giao thức trong ứng dụng IoT thời gian thực, cụ thể là hệ thống giám sát chất lượng không khí. Dựa trên các đánh giá chi tiết này, người dùng có thể quyết định cách sử dụng thích hợp của họ trong các hệ thống IoT khác nhau tùy theo yêu cầu để phù hợp với hệ thống.

Từ khóa - Internet of Things (IoT); HTTP; MQTT; giao thức.

1. Giới thiệu

IoT ngày càng được phát triển mạnh mẽ. Nó là một chủ đề được quan tâm trên toàn thế giới. Trong hệ thống IoT, một số lượng lớn dữ liệu nhỏ từ các thiết bị, chẳng hạn như các cảm biến khác nhau, được chuyển qua hệ thống mạng. Mặc dù, giao thức Internet (IP) đã được áp dụng cho hầu hết các loại hình giao tiếp, nhưng nó sẽ gặp một số vấn đề khi áp dụng cho hệ thống IoT.

Hiện tại, truy cập Internet yêu cầu truyền tải qua giao thức qua TCP/IP hoặc UDP/IP. Một trong những giao thức ở lớp ứng dụng là giao thức truyền siêu văn bản (HTTP), đã được chuẩn hóa trong IETF. Ví dụ: [1] (phiên bản đầu tiên); [2] (phiên bản mới nhất), và đã được áp dụng cho giao tiếp chung qua Internet. Tuy nhiên, khi HTTP được áp dụng cho giao tiếp trong IoT, với một số lượng lớn các khối dữ liệu nhỏ được truyền tải, dẫn đến suy giảm hiệu suất là một vấn đề nghiêm trọng. Hơn nữa, địa chỉ IP phụ thuộc vào vị trí vật lý, điều này gây ra vấn đề phức tạp của việc kiểm soát mạng.

Để giải quyết những vấn đề này, các kiến trúc chẳng hạn như mạng dữ liệu được đặt tên (NDN), mạng trung tâm nội dung (CCN) và mạng trung tâm thông tin (ICN) đã được thảo luận [3] - [9]. Ngoài ra, một số nghiên cứu tập trung vào việc áp dụng các kiến trúc này [10] - [12] vào IoT. Trong đó, MQ Telemetry Transport (MQTT) là một trong những giao thức, như được mô tả trong [13]. Các ủy ban tiêu chuẩn như oneM2M và ETSI đã dành sự quan tâm đáng kể đến MQTT và cũng đã tiến hành các cuộc thảo luận có liên quan. MQTT giảm lưu lượng trong giao thức và cung cấp giao tiếp hiệu quả cao cho hệ thống IoT. Nó cũng gọi "định tuyến dựa trên

Abstract - The rapid growth of the Internet of Things (IoT) has made this technology one of the most promising models in recent years. Ensuring the performance of real-time system is one of the issues that need attention in IoT applications. In this study, we present and evaluate two of the most popular protocols used in real-world IoT systems, MQTT and HTTP, with tests to evaluate the performance of these two protocols in the real-time IoT applications, specifically air quality monitoring systems. Based on these detailed evaluations, users can decide their appropriate usage in different IoT systems according to their requirements to fit the system.

Key words - Internet of Things (IoT); HTTP; MQTT; protocol.

tên" và giảm thiểu định tuyến dựa trên địa chỉ IP cho các luồng lưu lượng trong hệ thống IoT.

Theo kết quả khảo sát các nhà phát triển IoT trong ấn bản năm 2018 của Eclipse Foundation [17] đã phát hiện ra rằng giao thức được sử dụng nhiều nhất làm giải pháp IoT là MQTT với 62,61%, sau đó là giao thức HTTP với 54,10%. HTTP được phát triển vào năm 1997 trong khi MQTT được giới thiệu vào năm 1999 bởi IBM.

Với nhu cầu sử dụng cao như vậy, vấn đề đánh giá thực tế hiệu năng của giao thức HTTP và MQTT trong hệ thống IoT thời gian thực là vấn đề đáng được quan tâm. Một số nhà nghiên cứu đã chọn mô-đun GSM làm phương tiện giao tiếp không dây cho ứng dụng di động thời gian thực. Ví dụ, ứng dụng phát hiện tai nạn và an toàn đường bộ [14], [15] và hệ thống theo dõi xe chống trộm thông minh [16], hệ thống quản lý dữ liệu thời gian thực cho ứng dụng thành phố thông minh [18]. Ngoài ra, một số nghiên cứu cũng đưa ra được điểm mạnh và điểm yếu qua các thử nghiệm như: [19] đã nghiên cứu, so sánh điểm mạnh và điểm yếu các giao thức MQTT, CoAP, HTTP và XMPP cho một hệ thống nhúng; Nghiên cứu [20] thực hiện các phân tích đánh giá các giao thức MQTT, CoAP, OPC UA trong hệ thống mạng IoT. Tuy nhiên, những nghiên cứu đó không tập trung vào nghiên cứu về các giao thức truyền tải cũng như việc lựa chọn giao thức. Vì vậy, vấn đề lựa chọn giao thức sao cho phù hợp với từng ứng dụng cụ thể của một hệ thống IoT thời gian thực hiện mới chỉ đưa ra các so sánh về lý thuyết, chưa có nhiều đánh giá một cách thực tế, nhất là trong trường hợp thiết bị tính hay di chuyển cũng cần được khảo sát.

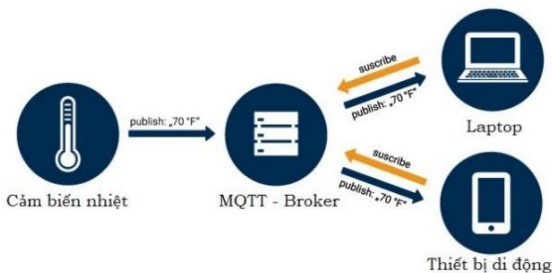
¹ The University of Danang - University of Science and Technology (Thai Vu Hien)

Trong bài báo này, nhóm tác giả tập trung vào triển khai một hệ thống IoT thời gian thực với chi phí thấp. Sử dụng mô-đun giao tiếp GSM/GPRS trong ứng dụng hệ thống giám sát chất lượng không khí (đo nồng độ khí CO₂), được gắn trên vật thể di động và sử dụng các giao thức nhắn tin HTTP và MQTT để tiến hành đánh giá hiệu năng bằng việc so sánh tốc độ gửi, hiệu suất và thời gian trễ (jitter) của cả giao thức HTTP và MQTT. Nghiên cứu này cũng sẽ cho thấy rõ thuật toán của cả hai giao thức truyền thông trên khía cạnh của một hệ thống nhúng. Mục đích chính của nghiên cứu này là đánh giá và so sánh được hiệu năng hoạt động của 2 giao thức HTTP và MQTT trong hệ thống IoT thời gian thực với thiết bị tĩnh và di chuyển.

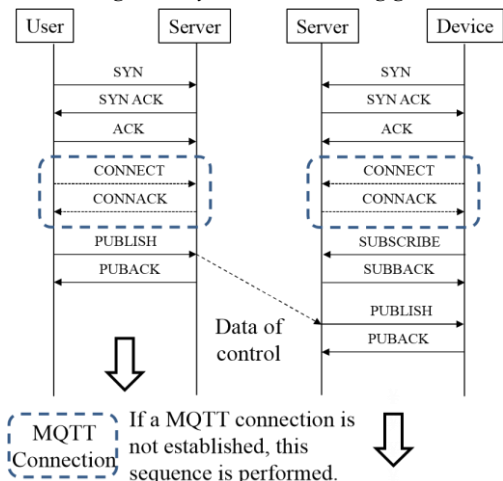
2. MQTT và HTTP trong hệ thống IoT

2.1. MQTT

MQTT cung cấp ba chế độ truyền dựa trên độ tin cậy cần thiết: QoS0 (Truyền không đảm bảo), QoS1 (Truyền đảm bảo) và QoS2 (Dịch vụ đảm bảo trên các ứng dụng). QoS1 tương tự như HTTP từ quan điểm độ tin cậy. Trong khi HTTP là một giao thức đối xứng, MQTT có một kiến trúc bất đối xứng cho việc truyền dữ liệu nhẹ. Trong hầu hết các giao tiếp cho IoT, các thiết bị phân tán không thông minh giao tiếp với một máy chủ có khả năng thông minh, giao tiếp bất đối xứng được cung cấp.



Hình 1. Hệ thống IoT truyền dữ liệu sử dụng giao thức MQTT



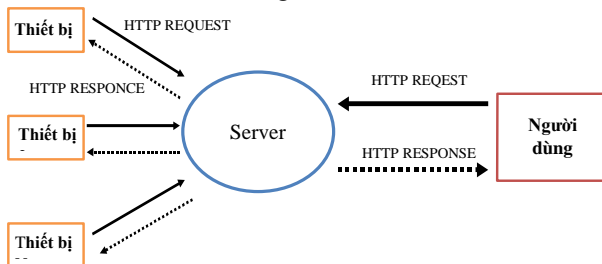
Hình 2. Trình tự giao tiếp trên MQTT [21]

MQTT bao gồm hai bộ thông báo trên một kết nối, "PUBLISH" và "SUBSCRIBE". Các khối dữ liệu được gửi bằng tin nhắn "PUBLISH" và nhận bằng tin nhắn "SUBSCRIBE". Các khối dữ liệu này được xác định theo "TOPIC". Khối dữ liệu nhận được xác định theo "TOPIC", đã đăng ký bằng tin nhắn "SUBSCRIBE" trước. Sơ đồ của một hệ thống IoT sử dụng giao thức MQTT và trình tự giao

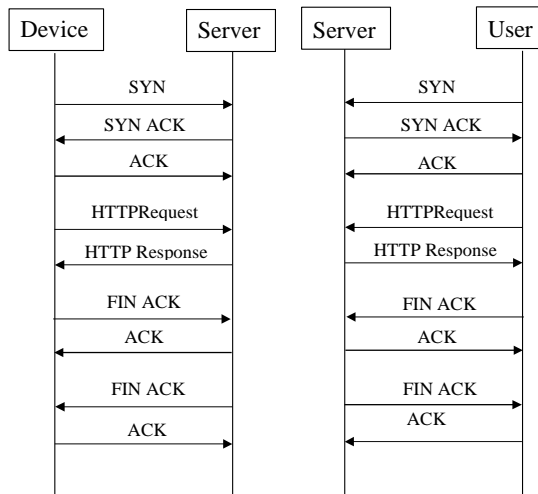
tiếp trên giao thức MQTT được mô tả lần lượt ở Hình 1 và Hình 2.

2.2. HTTP

Giả định rằng HTTP được áp dụng cho giao tiếp IoT. HTTP phải chuyển một số lượng lớn các gói nhỏ. Lưu lượng trong giao thức của HTTP gây ra các vấn đề nghiêm trọng, chẳng hạn như tiêu thụ tài nguyên mạng và độ trễ lớn. Cấu trúc của một hệ thống IoT sử dụng HTTP được mô tả như trong Hình 3 và Biểu đồ tuần tự trong giao thức HTTP được thể hiện trong Hình 4.



Hình 3. Cấu trúc của một hệ thống IoT sử dụng HTTP



Hình 4. Trình tự giao tiếp trên HTTP [21]

Vì HTTP được vận hành qua TCP/IP, là giao thức đáng tin cậy ở lớp vận chuyển. Tuy nhiên, các kết nối được thiết lập bởi TCP được giải phóng sau mọi truy cập, vì dữ liệu đã truy cập được chuyển dựa trên địa chỉ IP, đường dẫn URL và mối quan hệ của chúng được thay đổi động. Trong ngắn hạn, sau khi nhiều lần thiết lập khởi tạo một kết nối, sự trao đổi thông tin mới được hoàn thành. Do đó, giao tiếp cho IoT gây ra lưu lượng nghiêm trọng và tiêu thụ tài nguyên mạng trong quá trình giao tiếp này.

3. Phương pháp thử nghiệm

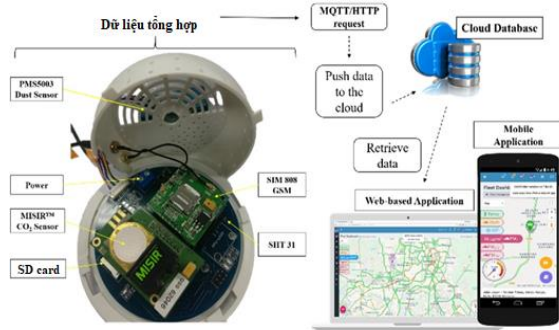
Để đánh giá được hiệu suất của hai giao thức MQTT và HTTP trong ứng dụng IoT thời gian thực Một hệ thống giám sát chất lượng không khí di động được đề xuất được minh họa trong Hình 5.

Mạch trung tâm sử dụng chip ATmega1280, là một vi điều khiển mã nguồn mở có thể tải lên các chương trình dành riêng thông qua phần mềm Arduino IDE cũng như bao gồm các thư viện mô-đun cảm biến liên quan.

Các thiết bị khác trong hệ thống bao gồm một số cảm biến, cụ thể là cảm biến carbon dioxide (MISIR-5000), cảm

biến hạt bụi (PMS5003) và cảm biến độ ẩm và nhiệt độ (SHT-31). Mô-đun giao tiếp đã chọn là SIM808 để cung cấp tọa độ GPS và các tính năng GSM/GPRS. Thẻ nhớ cũng được nhúng vào hệ thống như một chức năng lưu dữ liệu để tổng hợp phân tích.

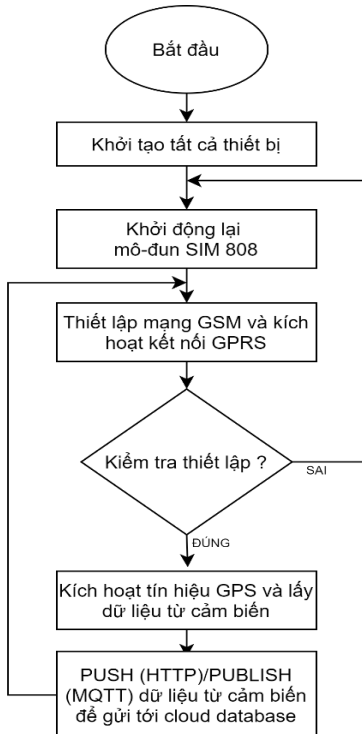
Bộ vi điều khiển cũng được lập trình ghi lại lỗi hệ thống. Khi lỗi được tìm thấy, tệp nhật ký lỗi sẽ được tạo và lưu trữ bên trong thẻ nhớ cục bộ để phục vụ quá trình gỡ lỗi sau này.



Hình 5. Hệ thống giám sát chất lượng không khí thời gian thực

Sau quá trình thu thập dữ liệu từ cảm biến, dữ liệu sẽ được đóng gói theo các giao thức nhắn tin. Sau đó, truyền dữ liệu thông qua tính năng GSM/GPRS được cung cấp bởi mô-đun SIM808. Việc lựa chọn giao thức nhắn tin phụ thuộc vào kịch bản thử nghiệm. Mỗi thiết bị đã chỉ định một số nhận dạng (ID) duy nhất để đảm bảo lưu trữ dữ liệu trong các cột dành riêng cho nó.

Có một số dữ liệu bắt buộc phải có trong gói dữ liệu truyền tới server như ID (để nhận dạng thiết bị gửi đến), dữ liệu GPS (gồm kinh độ và vĩ độ). Nếu thiếu dữ liệu bắt buộc này, cơ sở dữ liệu trên server sẽ không lưu trữ và sẽ loại bỏ dữ liệu này.



Hình 6. Lưu đồ thuật toán hệ thống thử nghiệm

Hình 6 trình bày biểu đồ thuật toán cho hệ thống thử nghiệm cho việc lấy dữ liệu GPS và các cảm biến đo chất lượng không khí, và gửi tới server qua 2 giao thức HTTP và MQTT để phân tích. Đầu tiên, hệ thống khởi tạo kết nối với các thiết bị cảm biến và khởi động mô-đun SIM 808 để thiết lập mạng GSP và kích hoạt kết nối GPRS sẵn sàng truyền nhận dữ liệu. Sau khi kiểm tra các thiết lập đã thành công, vi xử lý sẽ gửi tín hiệu kích hoạt GPS và bắt đầu lấy dữ liệu từ cảm biến, các cảm biến được sử dụng là carbon dioxide (MISIR-5000), cảm biến hạt bụi (PMS5003) và cảm biến độ ẩm và nhiệt độ (SHT-31), nên dữ liệu cảm biến ở đây là nồng độ CO₂, nồng độ bụi mịn PM2.5, nhiệt độ và độ ẩm trong môi trường đo. Các dữ liệu này sẽ được gửi cùng lúc thông qua cả 2 giao thức là HTTP và MQTT tới máy chủ cơ sở dữ liệu và từ đó có thể phân tích các gói tin từ cơ sở dữ liệu này.

Trong nghiên cứu này, cả hai giao thức truyền tin đều được xây dựng trên kết nối TCP/IP. Tổng kích thước header yêu cầu cho HTTP là 137 byte, và với giao thức MQTT yêu cầu kích thước header là 2 byte. Hạn chế đáng kể trong một hệ thống nhúng là bộ nhớ của bộ vi điều khiển của nó. Vì vậy, kích thước tải trọng cho mỗi lần truyền có giới hạn bởi các thư viện Arduino.

Tổng kích thước của dữ liệu cảm biến được thu thập là khoảng 200 byte cho đến 250 byte. Ngoài ra, giao thức HTTP có thể chứa dữ liệu lớn hơn 8 lần so với giao thức MQTT về kích thước tải trọng, tương ứng là 2048 byte và 256 byte. Mức độ phức tạp của giao thức HTTP cao so với MQTT. Điều này là do trong giao thức HTTP phải được đảm bảo phản hồi xác nhận trở lại vì điều khiển từ Cloud Database trước khi gửi tiếp theo. Cần có giao thức HTTP để tạo kết nối mới cho mỗi lần cập nhật dữ liệu, vì vậy yêu cầu tốn nhiều thời gian hơn so với giao thức MQTT, khi mà kết nối được thiết lập sẽ cho phép dữ liệu liên tục gửi đến Cloud Database.

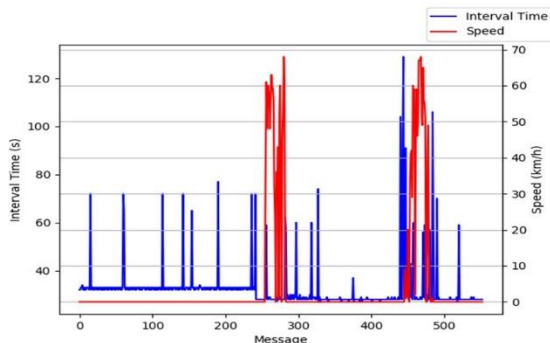
Tuy nhiên, kết nối giao thức MQTT ở chế độ lưu giữ và ứng dụng trong tình huống di động. Khi thiết bị di động sắp đi vào một vùng mạng di động mới, kết nối GSM sẽ cố gắng tạo một kết nối mới. Vì vậy, kết nối được thiết lập trước đó cho giao thức truyền tin có thể bị hủy bỏ. Như vậy, việc lập trình cho cả hai giao thức cần tạo ra chức năng kiểm tra kết nối. Bất cứ khi nào mất tín hiệu, mô-đun giao tiếp sẽ được thiết lập lại và điều khiển để kết nối bắt đầu lại.

4. Kết quả và bàn luận

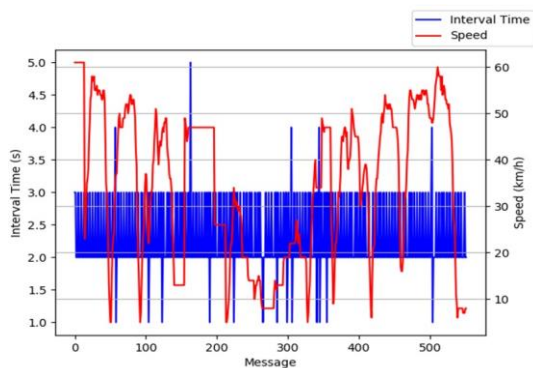
Phần này thảo luận về việc đánh giá hiệu suất của cả giao thức nhắn tin HTTP và MQTT dựa trên mô-đun truyền tin qua mạng GSM trong hệ thống IoT thời gian thực được triển khai. Kịch bản thử nghiệm về việc gửi dữ liệu đối với 2 giao thức HTTP và MQTT trong tình trạng tĩnh và di chuyển của thiết bị thu thập dữ liệu trong thời gian thực đã được ghi lại và phân tích kết quả dựa trên khoảng thời gian gửi (interval Time) của các gói tin.

Kết quả trong trường hợp thiết bị thu thập dữ liệu di chuyển được thể hiện qua các biểu đồ được vẽ cho giao thức HTTP và MQTT lần lượt được hiển thị trong Hình 7 và Hình 8. Từ cả hai biểu đồ đường được trình bày, có thể thấy tốc độ truyền của mô-đun GSM không bị ảnh hưởng bởi tốc độ di chuyển của thiết bị thu thập dữ liệu. Và có thể

quan sát thấy, một số mức tăng đột biến đáng kể trong cả hai biểu đồ. Sự gia tăng đột biến xảy ra khi hệ thống đang cố gắng gửi lại thông tin, sau đó gây ra khoảng cách giữa các khoảng thời gian truyền tin lớn hơn. MQTT có hiệu suất tốt hơn trong thời gian gửi gói tin – interval Time so với HTTP trong điều kiện tốc độ di chuyển có biến động. Ngoài ra, cũng có thể thấy rằng, việc sử dụng thuật toán nhằm cố gắng gửi lại dữ liệu có một nhược điểm lớn là khả năng cao bỏ qua việc cập nhật dữ liệu sau đó, dẫn tới gây ra độ trễ trong ứng dụng thời gian thực.



Hình 7. Thời gian gửi gói tin khi thiết bị thu thập dữ liệu di chuyển với tốc độ khác nhau với giao thức HTTP



Hình 8. Thời gian gửi gói tin khi thiết bị thu thập dữ liệu di chuyển với tốc độ khác nhau với giao thức MQTT

Bảng 1 thể hiện hiệu năng của 2 giao thức trong trường hợp thiết bị đặt tĩnh và di chuyển. Thiết bị với trường hợp di chuyển sẽ được đặt trên xe máy và di chuyển trên đoạn đường ngắn với tốc độ biến động từ 0-70 km/h trong khu vực có sóng GSM với tốc độ trung bình đo được là 43 km/h với MQTT và 40 km/h với HTTP thể hiện trong mục tốc độ trung bình ở Bảng 1. Thông tin liên quan đến tốc độ nhận được tin nhắn trên cơ sở dữ liệu thu được, số lượng gói tin được cơ sở dữ liệu nhận được và tỷ lệ mất gói trong các điều kiện khác nhau. Giao thức HTTP trong trường hợp thiết bị tĩnh có tỷ lệ nhận gói thành công lên đến 100% so với giao thức MQTT với 97,9%. Trong kịch bản thiết bị di chuyển, cả hai giao thức cũng gặp phải vấn đề mất gói, tuy nhiên giao thức HTTP vẫn có tỷ lệ gửi gói thành công cao hơn (đạt 97,0%) so với giao thức MQTT (đạt 91,5%). Tuy giao thức HTTP có hiệu suất tuyệt vời về tốc độ tin nhắn đến nhưng về tốc độ gói tin, giao thức MQTT có thể hoạt động hiệu quả hơn. Thông lượng trung bình của giao thức MQTT nhanh hơn 9,6 lần so với giao thức của HTTP, đó là giao thức MQTT có thể mất trung bình 3 giây để “PUBLISH” một gói tin mới, trong khi giao thức HTTP phải sử dụng trung bình 28 giây để cập nhật một gói tin mới.

Tiêu chí đánh giá hiệu năng trong hệ thống IoT thời gian thực gồm thời gian gửi gói tin (hay độ trễ) và tỷ lệ gói tin gửi thành công. Thời gian gửi gói tin càng nhỏ (hay độ trễ nhỏ) và tỷ lệ gói tin gửi thành công càng cao thì hệ thống càng gần với thời gian thực và có độ tin cậy cao.

Bảng 1. So sánh hiệu năng của 2 giao thức trong trường hợp thiết bị đặt tĩnh và di chuyển

		Được lấy mẫu trong thời gian một giờ			
		Thiết bị tĩnh		Thiết bị tĩnh	
		MQTT	MQTT	MQTT	MQTT
Thời gian gửi gói tin	Trung bình	3s	28s	3s	29s
	Lớn nhất	30s	29s	57s	130s
	Nhỏ nhất	1s	27s	1s	27s
Số gói tin gửi		1071	128	1316	124
Tỷ lệ gửi gói thành công		97,9%	100%	91,5%	97,0%
Tốc độ trung bình		0	0	43 km/h	40 km/h

Ngoài đánh giá dựa vào độ trễ, jitter cũng là số liệu quan trọng để theo dõi và đo lường hiệu suất mạng, jitter được định nghĩa là sự khác biệt về độ trễ giữa hai gói liên tiếp.

Các gói tin chậm chèn (jitter packets) thường xảy ra bởi bị ảnh hưởng nhiều nhất bởi mất gói hoặc tắc nghẽn mạng trong khi truyền dữ liệu. Gói gửi không đến được tới database cloud theo trình tự được định nghĩa là mất gói. Thời gian chậm chèn xảy ra trong các giao thức càng thấp thì hiệu suất càng tốt. Phương trình (1) thể hiện tính toán toán học của jitter. Trong hệ thống thử nghiệm này, dữ liệu được gửi trực tiếp đến server, mà không thông qua server trung gian nào, nên hằng số thời gian T được chọn bằng 0.

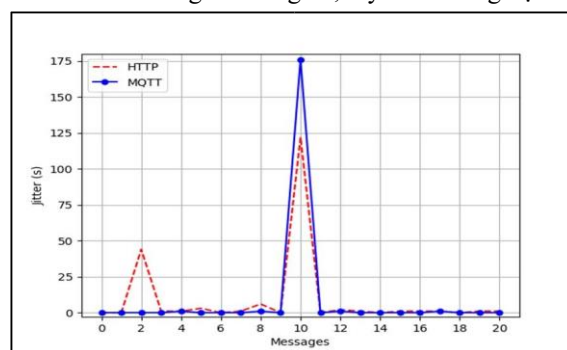
$$J_n = t'_n - t'_{n-1} - T \quad (1)$$

Trong đó: J_n là thời gian jitter tại gói tin thứ n.

t'_n là thời gian gửi gói tin (time of arrival) thứ n.

t'_{n-1} là thời gian gửi gói tin (time of arrival) trước đó, thứ n-1.

T là hằng số thời gian, tùy vào thử nghiệm.



Hình 9. Biểu đồ jitter cho giao thức HTTP và MQTT với 20 gói tin

Hình 9 chỉ ra rằng, trong giao thức HTTP thường xuyên xảy ra các gói tin chậm chèn (jitter packets). Điều này có thể giải thích bởi quá trình thực hiện của giao thức HTTP. Mỗi quá trình gửi dữ liệu của giao thức HTTP sẽ đảm bảo cơ sở dữ liệu phản hồi xác nhận để chỉ ra rằng, cơ sở dữ liệu đã hoàn toàn nhận được thông tin. Khi cơ sở dữ liệu không phản hồi theo khung thời gian, giao thức HTTP sẽ cố gắng gửi lại dữ liệu trước đó. Do đó, các gói tin chậm chèn xảy ra thường xuyên. Ngược lại, hiện tượng gói tin

chập chờn không thường xuyên gặp phải trong giao thức MQTT. Tuy nhiên, trong trường hợp kết nối lại với cơ sở dữ liệu (database cloud), các gói tin chập chờn vẫn có thể xảy ra trong MQTT.

5. Kết luận

Bài báo đã triển khai được mô hình thực tế IoT thời gian thực để đánh giá và so sánh hiệu năng hoạt động của 2 giao thức HTTP và MQTT đối với thiết bị tĩnh và di động. Hệ thống IoT thực hiện trong thời gian thực và triển khai trong thực tế sẽ có nhiều vấn đề xảy ra. Việc lựa chọn giao thức truyền tin phù hợp sẽ làm tăng số lượng gói tin được gửi thành công, mang lại kết quả tốt hơn cho ứng dụng. Giao thức MQTT có header nhỏ, chỉ với 2 Byte và giới hạn ở kích thước tải trọng 256 Byte trong hệ thống nhúng Arduino. Trong khi đó, giao thức HTTP gửi tin có độ dài 2048 Byte. Ngay cả giao thức HTTP cũng có kích thước tải trọng lớn hơn 8 lần so với giao thức MQTT. Tuy nhiên, qua việc kiểm chứng thực tế đã cho thấy, tốc độ gửi của MQTT nhanh hơn khoảng mười lần so với HTTP.

Vì vậy, có thể kết luận rằng, giao thức MQTT có hiệu suất vượt trội với gói tin chập chờn (jitter packets) thấp khi được áp dụng trong mô-đun truyền thông GSM/GPRS. Đối với giao thức HTTP, nó cung cấp giao thức truyền tin bảo mật nhưng tốc độ gửi không được đáp ứng trong ứng dụng này. Giao thức MQTT được tạo ra để ứng dụng phù hợp với hệ thống IoT thời gian thực hơn là HTTP. Dựa trên kết quả nghiên cứu, người dùng có thể quyết định cách sử dụng thích hợp giao thức của họ trong các hệ thống IoT khác nhau tùy theo yêu cầu để phù hợp với hệ thống.

Lời cảm ơn: Bài báo này được tài trợ bởi Trường Đại học Bách khoa – Đại học Đà Nẵng với đề tài có mã số: T2021-02-09.

TÀI LIỆU THAM KHẢO

- [1] T. Berners-Lee, R. Fielding, H. Frystyk, *Hypertext Transfer Protocol -- HTTP/1.0*, IETF RFC 1945, 1996.
- [2] M. Belshe, R. Peon, M. Thomson, *Hypertext Transfer Protocol Version 2 (HTTP/2)*, IETF RFC 7540, 2015
- [3] J. Luo, C. Wu, Y. Jiang, J. Tong, "Name Label Switching Paradigm for Named Data Networking", *IEEE Communications Letters*, Vol. 19, 2015, pp.335 – 338.
- [4] S. Eum, K. Nakauchi, Y. Shoji, N. Nishinaga, M. Murata, "CATT: Cache aware target identification for ICN", *IEEE Communications Magazine*, Vol. 50, No. 12, 2012, pp. 60 – 67.
- [5] C. Fang, R. Yu, T. Huang, J. Liu, Y. Liu, "A Survey of Green Information-Centric Networking: Research Issues and Challenges", *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 3, 2015, pp. 1455 – 1472.
- [6] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, "A survey of information-centric networking", *IEEE Communications Magazine*, Vo. 50, No.7, 2012, pp. 26 – 36.
- [7] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, "A Survey of Information-Centric Networking Research", *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 2, 2014, pp. 1024 – 1049.
- [8] M. Yamamoto, "Research trends on In-Network caching in content-oriented networks", *IEICE Technical Report.*, Vol. 115, No.461, 2015, pp. 23 – 28.
- [9] Md. F. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, B. Mathieu, "A survey of naming and routing in information-centric networks", *IEEE Communications Magazine*, Vol. 50, No.12, 2012, pp.43 – 52.
- [10] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. Aguiar, A. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities", *IEEE Network*, Vol. 30, No.2, 2016, pp. 92 – 100.
- [11] H. Yue, L. Guo, R. Li, H. Asaeda, "Yuguang Fang DataClouds: Enabling Community-Based Data-Centric Services Over the Internet of Things", *IEEE Internet of Things Journal*, Vol. 1, 2014, pp. 472 – 482.
- [12] I. Sato, T. Kurira, K. Fukuda, T. Tsuda, "A extension of information centric for IoT applications", *Report of 3rd Technical committee on information centric networking (ICN)*, in *IEICE*, 2015, pp. 921 – 100.
- [13] IBM, "MQTT V3.1 Protocol Specification", 2012, [Online] <https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>, [Accessed: 03-Nov-2021].
- [14] P. Nath and A. Malepati, "IMU based Accident Detection and Intimation System", *2018 2nd Int. Conf. Electron. Mater. Eng. Nano-Technology*, 2018, pp. 1–4.
- [15] P. Joshi, G. Tandel, P. Sawant, S. Jain, M. Ghadi, and D. Vira, "Identification of Road Distress with Notification System", *Int. Res. J. Eng. Technol.*, vol. 4, no. 4, 2017, pp. 960–964.
- [16] M. S. Uddin, M. M. Ahmed, J. B. Alam, and M. Islam, "Smart antitheft vehicle tracking system for Bangladesh based on Internet of Things", *4th Int. Conf. Adv. Electr. Eng. ICAEE 2017*, vol. 2018–Janua, 2018, pp. 624–628.
- [17] B. Cabé, "IoT Developer Survey 2018", Eclipse Foundation, 2018. [Online] <https://iot.eclipse.org/community/resources/iot-surveys/assets/iot-developer-survey-2018.pdf>, [Accessed:03-Nov-2021].
- [18] C. B. Gemirter, Ç. Şenturca and Ş. Baydere, "A Comparative Evaluation of AMQP, MQTT and HTTP Protocols Using Real-Time Public Smart City Data" *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021, pp. 542-54.
- [19] N. Nikolov, "Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems", *2020 XXIX International Scientific Conference Electronics (ET)*, 2020, pp. 1-4, doi: 10.1109/ET50336.2020.
- [20] Silva D, Carvalho LI, Soares J, Sofia RC. "A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA". *Applied Sciences*. 2021; 11(11):4879
- [21] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT", *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2016, pp. 1-6.