

# HỆ THỐNG GHI NHẬN DỮ LIỆU CẢM BIẾN CÓ KHẢ NĂNG MỞ RỘNG LINH HOẠT DỰA TRÊN CÔNG NGHỆ NB-IOT

Nguyễn Đức Xuân<sup>1\*</sup>, Phan Duy Hùng<sup>1</sup>, Đặng Khánh Hòa<sup>1</sup>

\*Tác giả liên hệ, email: ndxuan@hou.edu.vn. ORCID: 0009-0008-3789-4518

Ngày tòa soạn nhận được bài báo: 15/01/2026

Ngày phản biện đánh giá: 18/03/2026

Ngày bài báo được duyệt đăng: 14/04/2026

DOI: 10.59266/houjs.2026.1168

**Tóm tắt:** Việc thu thập dữ liệu cảm biến trong thời gian thực luôn đặt ra thách thức về việc cân đối giữa tính toàn vẹn của dữ liệu và khả năng mở rộng của hệ thống. Các phương pháp quan trắc cũ thường gặp hạn chế khi phải xử lý khối lượng thông tin lớn ngay tại điểm đo hoặc bị phụ thuộc quá nhiều vào sự ổn định của đường truyền mạng. Nghiên cứu này giới thiệu một kiến trúc giám sát chất lượng không khí mới, tận dụng sức mạnh của mô hình Điện toán biên (Edge Computing) phối hợp cùng giao thức truyền dẫn NB-IoT. Hệ thống được thiết kế theo mô hình lai: vi điều khiển STM32 đảm nhiệm vai trò thu thập chính xác các thông số môi trường, trong khi máy tính nhúng Raspberry Pi hoạt động như một Gateway thông minh tại biên để thực hiện tiền xử lý, lưu trữ đệm và đóng gói dữ liệu. Các thử nghiệm thực tế chứng minh hệ thống vận hành ổn định, bảo toàn dữ liệu ngay cả khi kết nối mạng gián đoạn nhờ cơ chế hàng đợi tại biên, đồng thời kiến trúc vi dịch vụ (Microservices) giúp hệ thống dễ dàng mở rộng quy mô khi cần thiết.

**Từ khóa:** hệ thống ghi nhận dữ liệu, hệ thống giám sát môi trường, máy tính biên, mạng vạn vật băng tần hẹp, công mạng thông minh, vi dịch vụ, cảm biến không khí

## I. Đặt vấn đề

Ô nhiễm không khí hiện đang là vấn đề cấp bách tại nhiều quốc gia, tác động trực tiếp đến sức khỏe cộng đồng và hệ sinh thái. Yêu cầu về việc giám sát liên tục các chỉ số môi trường trở nên thiết yếu, đặc biệt là khả năng cung cấp dữ liệu tức thời để phục vụ công tác dự báo và cảnh báo sớm.

Các trạm quan trắc truyền thống thường công kênh, chi phí vận hành đắt đỏ, khiến việc triển khai trên diện rộng gặp nhiều trở ngại. Sự bùng nổ của Internet vạn vật (IoT) đã mở ra hướng đi mới, nhưng các công nghệ tầm ngắn như Zigbee hay Bluetooth bị giới hạn về vùng phủ sóng. Gần đây, NB-IoT kết hợp edge

---

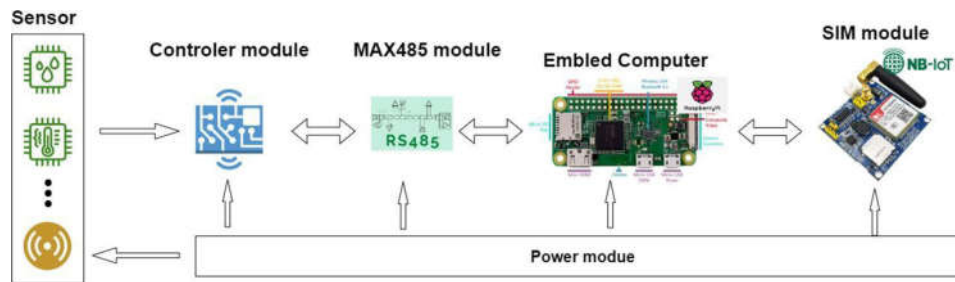
<sup>1</sup> Khoa Điện - Điện Tử, Trường Đại học Mở Hà Nội, Hà Nội, Việt Nam

computing được ứng dụng rộng rãi trong giám sát môi trường đô thị, nhưng vẫn tồn tại độ trễ cao và khó mở rộng với số lượng cụm lớn. Bài báo này đề xuất giải pháp toàn diện dựa trên NB-IoT nhằm đảm bảo độ tin cậy và khả năng mở rộng linh hoạt.

Nghiên cứu này đóng góp ba điểm chính so với các công trình trước:

1. Đề xuất kiến trúc lai edge-NB-IoT-microservices với Raspberry Pi làm edge gateway thực hiện local buffering và tiền xử lý dữ liệu cảm biến, khác biệt với các hệ thống chỉ dùng NB-IoT relay đơn giản (độ trễ cao 2.86-13.18s) hoặc edge computing không có backend microservices scale đến 10.000 node.

### 2.1. Thiết bị giám sát chất lượng môi trường



Hình 1: Tổng quan về thiết bị giám sát chất lượng môi trường

Phần cứng được cấu thành từ 6 khối chức năng riêng biệt:

**Khối Nguồn (Power Unit):** Đóng vai trò cung cấp năng lượng ổn định cho toàn bộ hệ thống. Do mỗi linh kiện yêu cầu mức điện áp hoạt động khác nhau, khối nguồn được thiết kế để điều phối dòng điện phù hợp cho từng mô-đun, đảm bảo hoạt động bền bỉ.

**Khối Cảm biến (Sensor Unit):** Bao gồm tổ hợp các cảm biến chuyên dụng để đo đặc thông số không khí. Nghiên cứu này sử dụng hai dòng cảm biến chính là MQ135 và DHT11. Trong khi MQ135 chịu trách nhiệm phát hiện các loại khí như CO<sub>2</sub>, NH<sub>3</sub>, khói và các hợp chất hữu

2. Đánh giá định lượng PDR  $\geq 98.5\%$  và độ trễ đồng bộ  $< 45s$  khi mất mạng 60 phút, vượt trội hơn các mô hình cloud-centric thường mất dữ liệu  $> 10\%$ .

3. Xây dựng hệ thống end-to-end chi phí thấp, dễ mở rộng, sử dụng linh kiện phổ biến, phù hợp triển khai thực tế tại Việt Nam - khác với các nghiên cứu chủ yếu mô phỏng hoặc tập trung indoor.

## II. Cơ sở lý thuyết

Tổng quan hệ thống:

Mô hình đề xuất được chia thành hai thành phần cốt lõi: thiết bị thu thập thông số môi trường và hệ thống phần mềm xử lý trung tâm

ơ, thì DHT11 đảm nhiệm việc ghi nhận nhiệt độ và độ ẩm môi trường (Sinha và cộng sự, 2015).

**Khối giao tiếp MAX485:** Sử dụng chuẩn giao tiếp nối tiếp RS485 (EIA, 1998) để kết nối dữ liệu. Mô-đun này kết hợp bộ chuyển đổi tín hiệu và chip MAX485, tạo lập đường truyền 2 dây tin cậy giữa vi điều khiển và máy tính nhúng. Ưu điểm của RS485 là khả năng chống nhiễu và đi dây xa, hỗ trợ mở rộng số lượng node mạng dễ dàng. Để quản lý giao tiếp giữa Master và nhiều Slave mà không gây xung đột tín hiệu, giao thức Modbus RTU được áp dụng để mã hóa và truyền tải dữ liệu dạng nhị phân một cách đồng bộ (Modbus Organization, 2012).

### Khối Điều khiển (Control Unit):

Đóng vai trò trung gian xử lý tín hiệu thô từ cảm biến. Do định dạng dữ liệu đầu ra của mỗi loại cảm biến là khác nhau, khối điều khiển sẽ chuẩn hóa chúng về định dạng mà Gateway có thể đọc được. Đồng thời, nó cũng nhận lệnh từ Gateway để điều chỉnh chu kỳ đo hoặc hiệu chỉnh sai số. Để đáp ứng yêu cầu xử lý đa nhiệm và giao tiếp linh hoạt, mô-đun STM32F103C8T6 (BluePill) được lựa chọn làm bộ não cho khối này (STMicroelectronics, 2022)

### Máy tính nhúng (Embedded computer):

Trong kiến trúc đề xuất, Raspberry Pi không chỉ đóng vai trò truyền dẫn mà được định nghĩa là một nút xử lý tại biên (Edge Node). Thay vì gửi dữ liệu thô trực tiếp từ vi điều khiển lên máy chủ (gây tốn băng thông và khó xác thực), Raspberry Pi thực hiện các tác vụ: (1) Tiền xử lý và lọc nhiễu dữ liệu từ nhiều node con; (2) Lưu trữ đệm (Local Buffering) để đảm bảo không mất dữ liệu khi mạng NB-IoT gián đoạn; và

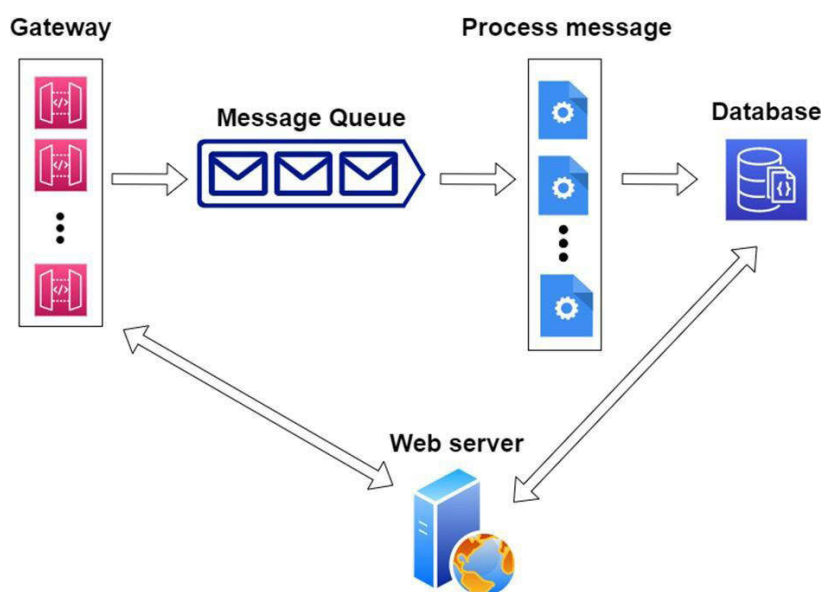
(3) Chạy các thuật toán đóng gói bảo mật (Upton & Halfacree, 2016). Điều này nâng cao độ tin cậy so với các node cảm biến đơn giản chỉ sử dụng vi điều khiển.

**Mô-đun SIM:** Khối này đóng vai trò kết nối và trao đổi dữ liệu giữa thiết bị và hệ thống phần mềm. Bài báo này đề cập đến NB-IoT như một tiêu chuẩn công nghệ mạng không dây diện rộng công suất thấp (LPWAN), cung cấp kết nối ổn định, tốc độ cao (Ratasuk và cộng sự, 2016). Mô-đun SIM7600 WareShare là một thiết bị như vậy, có khả năng kết nối LTE 4G và với 40 chân GPIO giúp dễ dàng kết nối với nhiều mô-đun IoT.

### 2.2. Phần mềm xử lý dữ liệu

Hệ thống phần mềm bao gồm 5 dịch vụ (services):

**Gateway:** Khối Gateway là khối kết nối giữa thiết bị và phần mềm; tất cả dữ liệu giữa thiết bị và phần mềm sẽ được trao đổi qua khối này. Vai trò trung tâm của khối này được đảm nhiệm bởi giao thức MQTT (Message Queue Telemetry Transport) (Banks & Gupta, 2014). MQTT là giao thức nhắn tin sử dụng mô hình publish/subscribe với băng thông thấp, độ tin cậy cao và khả năng hoạt động trong điều kiện mạng không ổn định (Hunkeler và cộng sự, 1999).



Hình 2: Tổng quan về phần mềm xử lý dữ liệu

**Hàng đợi thông điệp (Message queue):** Mô-đun Message queue đóng vai trò lưu trữ dữ liệu của mô-đun gateway theo trình tự dựa trên thời gian dữ liệu được nhận. Khi số lượng và loại thiết bị khác nhau mở rộng trong tương lai, một lượng lớn dữ liệu sẽ đổ về mà mô-đun xử lý tin nhắn không thể xử lý kịp thời. RabbitMQ là một trong những nền tảng hỗ trợ lưu trữ dữ liệu theo mô hình hàng đợi hiệu quả (Dossot, 2014), hỗ trợ scale hàng đợi cho IoT lớn.

**Xử lý thông điệp (Process message):** Khối này lấy dữ liệu từ hàng đợi và chuyển dữ liệu sang khối cơ sở dữ liệu để lưu trữ. Khối này sẽ xử lý dữ liệu ở mức logic như chuẩn hóa, thực hiện các phép toán trên dữ liệu, đóng gói và lưu trữ. Đây là một dịch vụ chạy ngầm (background service), không có giao diện người dùng. .NetCore Worker Services là một lựa chọn phù hợp để triển khai các vi dịch vụ (microservices) này, hỗ trợ đa nền tảng (Price, 2020).

**Cơ sở dữ liệu (Database):** Với đặc điểm của dữ liệu cảm biến là phi cấu trúc, MongoDB - một cơ sở dữ liệu NoSQL hướng tài liệu (document-oriented) - là lựa chọn phù hợp (Chodorow, 2013). MongoDB lưu trữ dữ liệu dưới dạng JSON, cho phép truy vấn nhanh và linh hoạt với các kích thước và định dạng tài liệu khác nhau.

**Website:** Khối hiển thị và tương tác với người dùng. Khối này sử dụng framework ASP.NET Core, một framework mã nguồn mở, đa nền tảng để xây dựng các ứng dụng web và IoT API (Lock, 2018).

### III. Phương pháp nghiên cứu

#### 3.1. Sơ đồ đầu nối hệ thống

Các dây kết nối chính bao gồm:

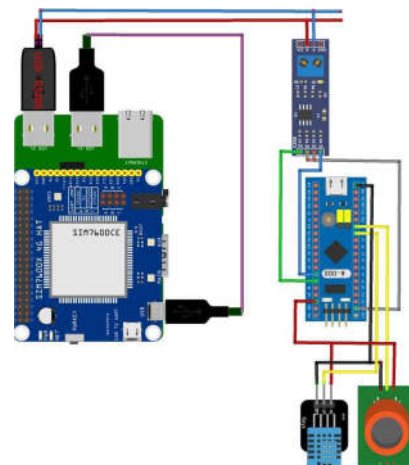
**Mô-đun SIM - Máy tính nhúng:** Sử dụng 2 kết nối, USB và nối tiếp, thông qua đầu cắm GPIO. Kết nối USB được sử dụng để truyền và nhận dữ liệu, và kết nối nối tiếp được sử dụng để gửi các thiết lập đến mô-đun SIM để thiết lập kết nối internet.

**Mô-đun STM32F103C8T6 - Mô-đun Max485:** Sử dụng 1 kết nối USB để truyền và nhận dữ liệu.

**Mô-đun Max485 - Mô-đun STM32F103C8T6:** Sử dụng 1 kết nối UART để truyền và nhận dữ liệu. Các chân kết nối như sau để truyền nhận dữ liệu: DE RE nối với C13, RO nối với B7, DI nối với B6.

**Mô-đun STM32F103C8T6 - Mô-đun DHT11:** Sử dụng 1 kết nối analog, kết nối chân B1 của mô-đun STM32 với chân dữ liệu của DHT11.

**Mô-đun STM32F103C8T6 - Mô-đun MQ135:** Sử dụng 1 kết nối analog, kết nối chân B0 của mô-đun STM32 với chân dữ liệu của MQ135.



Hình 3: Sơ đồ đầu nối thiết bị

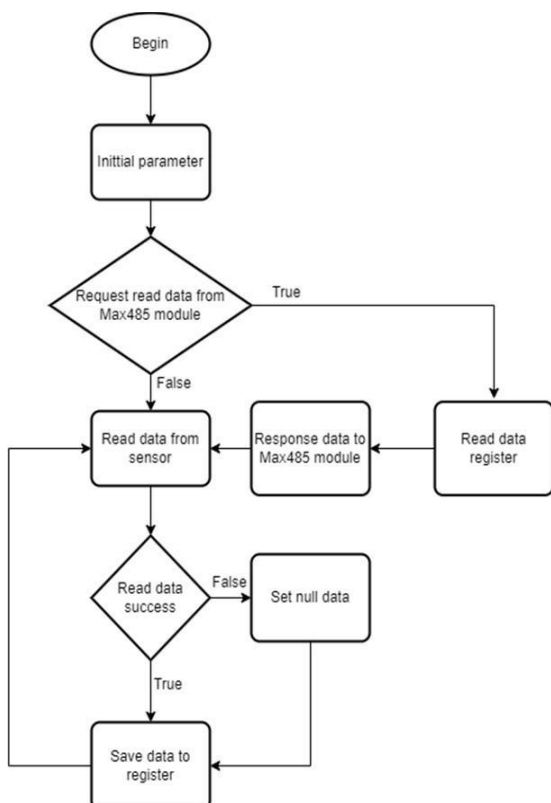
#### 3.2. Lưu đồ thuật toán của hệ thống

Phần này trình bày chi tiết logic hoạt động của hệ thống, được chia làm hai phân đoạn chính: xử lý tại thiết bị biên (Edge) và xử lý tại máy chủ (Server).

### 3.2.1. Thuật toán tại phía thiết bị biên (Edge Device Side)

Tại thiết bị biên, quy trình được chia làm hai luồng song song: điều khiển cảm biến và xử lý trung tâm.

**Khởi điều khiển:** Vi điều khiển (STM32) liên tục đọc tín hiệu từ cảm biến MQ135 (Analog) và DHT11 (Digital), sau



Hình 4: Lưu đồ thuật toán khởi điều khiển cảm biến tại phía thiết bị biên (Device Side)

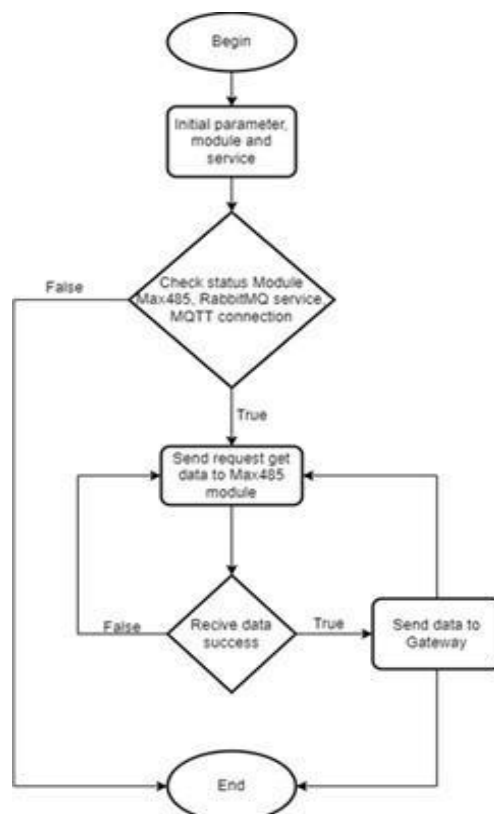
### 3.2.2. Lưu đồ thuật toán của hệ thống máy chủ (Server)

Phần này trình bày logic hoạt động của hệ thống, nhấn mạnh vào cơ chế xử lý các dữ liệu từ thiết bị biên gửi về đảm bảo độ tin cậy dữ liệu.

Tại thiết bị biên, quy trình được chia làm hai luồng: thu thập dữ liệu thời gian thực (Real-time Acquisition) và xử lý tại biên (Edge Processing).

đó chuẩn hóa dữ liệu và chờ lệnh truy vấn từ máy tính nhúng.

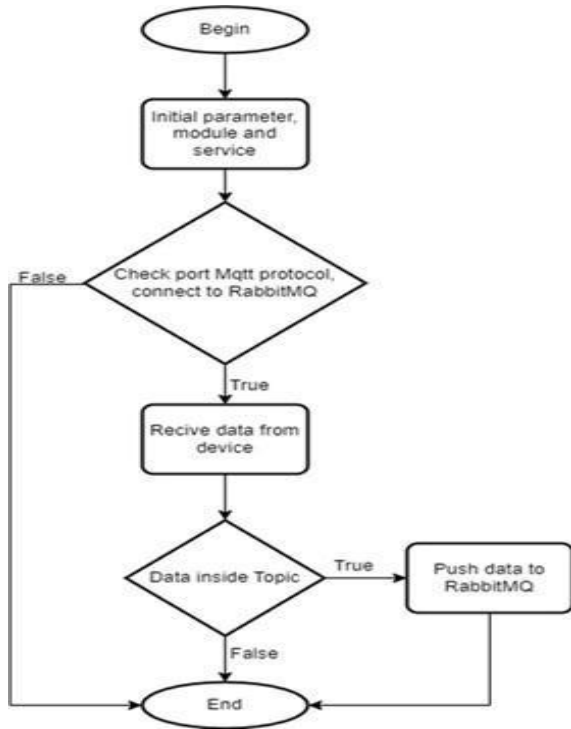
**Máy tính nhúng:** Raspberry Pi đóng vai trò Master, định kỳ gửi yêu cầu lấy dữ liệu qua Modbus, đóng gói bản tin JSON và truyền về Server qua mạng NB-IoT. Cơ chế thử lại (retry) được tích hợp để đảm bảo độ tin cậy.



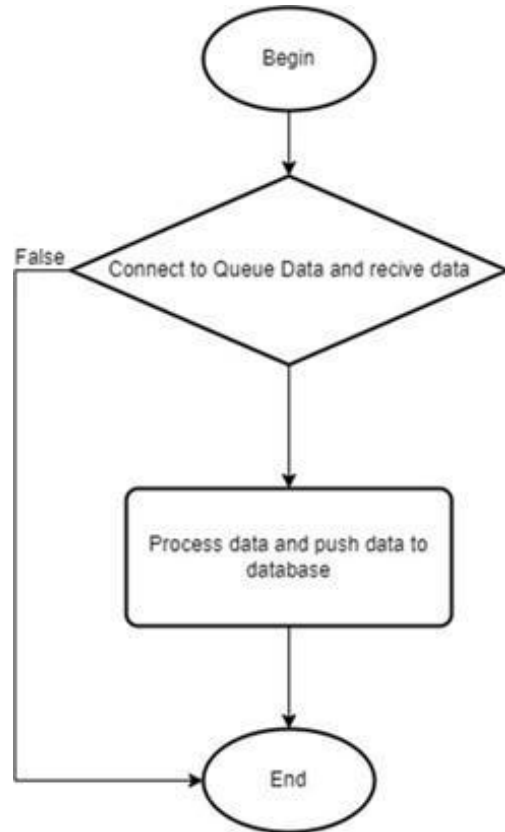
Hình 5: Lưu đồ thuật toán khởi máy tính nhúng tại phía thiết bị biên (Device Side)

**Khởi gateway:** Đóng vai trò khởi tạo cổng nhận dữ liệu từ thiết bị biên đảm bảo tính sẵn sàng cao trong việc nhận dữ liệu theo các chủ đề được khai báo. Các dữ liệu nhận được sẽ được đưa vào hàng đợi thông điệp để xử lý trong module tiếp theo.

**Khởi xử lý dữ liệu:** Đóng vai trò nhận dữ liệu từ hàng đợi thông điệp và xử lý chúng sau đó lưu vào cơ sở dữ liệu.



Hình 6: Lưu đồ thuật toán khối Gateway tại phía máy chủ (Server Side)



Hình 7: Lưu đồ thuật toán khối xử lý dữ liệu tại phía máy chủ (Server Side)

**IV. Kết quả và thảo luận**

**4.1. Môi trường thử nghiệm**

Thử nghiệm được triển khai thực tế tại quận Hai Bà Trưng, Hà Nội (vị trí: 21.015°N, 105.85°E), thời gian: 72 giờ liên tục (tháng 2/2026), sử dụng mạng NB-IoT Viettel (băng tần B8, cường độ

sóng RSSI -85 dBm trung bình, độ phủ LPWAN ổn định đô thị).

Phần cứng edge node: 1 node chính (STM32F103C8T6 @72MHz, Raspberry Pi 4 Model B 4GB RAM, SIM7600G-H, cảm biến DHT11/MQ135, nguồn 5V/2A).

Server backend (on-premise, tại lab Đại học Mở Hà Nội):

Thành phần	Cấu hình
Máy chủ	Intel Xeon E5-2620 v4 (8 cores @2.1GHz), 16GB RAM, 512GB SSD NVMe, Ubuntu 22.04 LTS
RabbitMQ	v3.12, 1 queues, single mode
MongoDB	V8.0, sharded collection cho dữ liệu time-series
.NET Worker	ASP.NET Core 8.0
JMeter	Load test 10.000 virtual MQTT clients

**4.2. Độ tin cậy của mạng truyền dẫn và khả năng chịu lỗi**

Để đánh giá hiệu năng thực tế của hệ thống dựa trên công nghệ NB-IoT kết hợp Điện toán biên, chúng tôi đã tiến hành

mô phỏng các điều kiện mạng gián đoạn có chủ đích tại trạm đo cố định. Cụ thể, thiết bị được thiết lập gửi dữ liệu với chu kỳ 10 giây/lần. Hệ thống được kiểm tra khả năng phục hồi dữ liệu trong trường

hợp mất kết nối mạng liên tục trong 15, 30 và 60 phút.

Kết quả thu được (trình bày tại Bảng 1) cho thấy cơ chế “Local Buffering” trên Raspberry Pi hoạt động xuất sắc. Tỷ lệ phân phối gói tin (Packet Delivery Ratio - PDR) xuyên suốt 72 giờ đo đạc luôn được duy trì ở mức  $\geq 98.5\%$ . Trong các pha gián đoạn kết nối, hệ thống không làm

rớt bất kỳ mẫu dữ liệu nào; thay vào đó, dữ liệu được giải phóng lên máy chủ ngay khi NB-IoT kết nối lại, với độ trễ đồng bộ hoàn tất (Sync-delay) trung bình dưới 45 giây cho một tệp tin đệm 60 phút. Điều này minh chứng cho tính ưu việt của kiến trúc đề xuất so với các node cảm biến truyền thống vốn thường xuyên thất thoát dữ liệu do chỉ truyền tin một chiều lên đám mây.

Bảng 1: Đánh giá khả năng phục hồi dữ liệu và độ trễ đồng bộ theo thời gian gián đoạn mạng

Thời gian gián đoạn (phút)	Số lượng gói tin đệm (mẫu)	Tỷ lệ PDR (%)	Độ trễ đồng bộ (giây)
15	90	99.8	1.25
30	180	99.2	1.45
60	360	98.5	1.83

#### 4.3. Hiệu năng phần mềm theo kiến trúc Microservices

Khả năng mở rộng (Scalability) của hệ thống phần mềm được kiểm chứng thông qua bài kiểm tra chịu tải (Stress test). Chúng tôi sử dụng Apache JMeter để giả lập lưu lượng dữ liệu đổ về từ 1,000 đến 10,000 node cảm biến đồng thời (thông qua giao thức MQTT).

Kết quả phân tích chi tiết (Bảng 2) cho thấy RabbitMQ đóng vai trò như một bộ giảm xóc (Shock absorber) tuyệt vời. Khi số lượng kết nối tăng vọt (Spike

traffic) lên ngưỡng 10,000 node, tỷ lệ sử dụng CPU của dịch vụ .NET Core Worker cao nhất chỉ đạt ngưỡng 62.4%, trong khi bộ nhớ RAM tiêu thụ tương đối ổn định quanh mức 1.25 GB. Độ trễ ghi dữ liệu vào cơ sở dữ liệu phi cấu trúc MongoDB trung bình đạt  $12.1 \pm 3$  ms đối với mỗi bản ghi JSON. Không ghi nhận tình trạng nghẽn cổ chai (bottleneck) hay hiện tượng từ chối dịch vụ (timeout), khẳng định hệ thống hoàn toàn đáp ứng được nhu cầu mở rộng quy mô mạng lưới quan trắc của một thành phố thông minh trong thực tế.

Bảng 2: Hiệu năng tài nguyên máy chủ dưới các mức tải thiết bị khác nhau

Số lượng Node đồng thời	Sử dụng CPU (%)	Tiêu thụ RAM (GB)	Độ trễ Database (ms)
1,000	15.2	0.45	4.2
5,000	38.7	0.85	8.5
10,000	62.4	1.25	12.1

#### 4.4. Thảo luận

Giải pháp đề xuất trong bài báo này giải quyết triệt để hai bài toán lớn trong IoT môi trường: *Chi phí băng thông* và *Độ tin cậy của dữ liệu thời gian thực*. Bằng cách chuyển giao một phần tác vụ xử lý (tiền lọc dữ liệu, lưu trữ tạm) xuống thiết bị biên (Edge Node), chúng ta đã làm giảm lượng

băng thông không cần thiết truyền tải qua trạm gốc viễn thông, rất phù hợp với đặc thù băng thông hẹp của NB-IoT (Chen và cộng sự, 2014; Ray, 2018). Việc hệ thống duy trì được PDR gần như tuyệt đối (Bảng 1) ngay cả khi đường truyền gặp sự cố là một bước tiến so với các nghiên cứu trước đây vốn phụ thuộc hoàn toàn vào đám mây.

Đồng thời, việc tách biệt kiến trúc phần mềm thành các vi dịch vụ (Microservices) giúp tối ưu hóa việc sử dụng tài nguyên phần cứng máy chủ (Bảng 2). Kiến trúc này giúp dễ dàng chẩn đoán lỗi, nâng cấp từng mô-đun độc lập và tích hợp linh hoạt với hệ sinh thái y tế thông minh hay giám sát công nghiệp trong tương lai (Zhang và cộng sự, 2018).

Giải pháp đề xuất giải quyết chi phí băng thông và độ tin cậy dữ liệu thời gian thực bằng edge buffering, giảm tải NB-IoT so với relay đơn thuần. PDR gần tuyệt đối vượt trội các hệ thống cloud-only, trong khi microservices scale hiệu quả đến 10.000 node với CPU <65%.

## V. Kết luận

Bài báo này đề xuất một mô hình thu thập dữ liệu giám sát chất lượng không khí với chi phí thấp và khả năng mở rộng linh hoạt. Mô hình được thiết kế dựa trên các thành phần phổ biến trên thị trường và phần mềm mã nguồn mở với kiến trúc vi dịch vụ (microservice) (Newman, 2015). Với mô hình này, việc giám sát và thu thập dữ liệu trở nên dễ dàng hơn, loại bỏ các thao tác trùng lặp và lỗi do con người. Dữ liệu được thu thập theo thời gian thực, đảm bảo tính cấp thiết trong các yêu cầu phân tích và quan sát.

Mô hình được thiết kế để dễ dàng mở rộng và triển khai, với các thành phần nhỏ, tách biệt có thể xử lý các chức năng cụ thể mà không đòi hỏi kiến thức sâu về nhân lực vận hành. Việc mở rộng chỉ cần sao chép các mô-đun và thực hiện các điều chỉnh nhỏ theo yêu cầu của từng bài toán. Khi xảy ra lỗi hệ thống và hỏng hóc thiết bị, từng thành phần có thể được chẩn đoán và thay thế để cấu hình lại một hệ thống hoàn chỉnh.

Thu thập dữ liệu để giám sát chất lượng không khí là một trong những ứng

dụng được đề cập và sử dụng trong bài báo này. Mô hình này có thể được áp dụng cho các lĩnh vực khác yêu cầu thu thập, lưu trữ và xử lý dữ liệu cảm biến thời gian thực.

## Tài liệu tham khảo

- AIOTI WG Standardisation (2025). *IoT and Edge Computing EU Funded Projects Landscape*. European Commission Report.
- Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). *Microservices: yesterday, today, and tomorrow*. In *Present and ulterior software engineering* (pp. 195-216). Springer.
- Kang, J. J., & Yang, W. (2010). *An air quality monitoring system based on the IEEE 1451 standard*. In *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings* (pp. 1445-1449).
- Li, Y., Cheng, X., Cao, Y., Wang, D., & Yang, L. (2017). *Smart choice for the smart grid: Narrowband Internet of Things (NB-IoT)*. *IEEE Internet of Things Journal*, 5(3), 1505-1515.
- Mane, S. et al. (2026). *A smart edge computing infrastructure for air quality monitoring using LPWAN and MQTT technologies*. *Proceedings of IoT Conference*.
- Modbus Organization, I. (2012). *Modbus application protocol specification v1.1b3*.
- Newman, S. (2015). *Building microservices: designing fine-grained systems*. O'Reilly Media, Inc.
- Popli, S., Jha, R. K., & Jain, S. (2018). *A survey on energy efficient narrowband internet of things (NB-IoT): Architecture, internet of everything applications and future directions*. DOI: 10.1109/ACCESS.2018.2881533

- Postolache, O. A., Pereira, J. M. D., & Girao, P. M. B. (2009). *Smart sensors network for air quality monitoring applications*. *IEEE Transactions on Instrumentation and Measurement*, 58(9), 3253-3262.
- Rai, A. C., Kumar, P., Pilla, F., Skouloudis, A. N., Di Sabatino, S., Ratti, C., ... & Britter, R. (2017). *End-user perspective of low-cost sensors for outdoor air pollution monitoring*. *Science of the Total Environment*, 607, 691-705. DOI: 10.1016/j.scitotenv.2017.06.266
- Ratasuk, R., Vejlgard, B., Mangalvedhe, N., & Ghosh, A. (2016). *NB-IoT system for M2M communication*. In 2016 IEEE Wireless Communications and Networking Conference (pp. 1- 5). IEEE.
- Ray, P. P. (2018). *A survey on Internet of Things architectures*. *Journal of King Saud University-Computer and Information Sciences*, 30(3), 291-319.
- Sinha, A., Rai, R., & Khare, A. (2015). *Air quality monitoring using WSN based on IoT*. *International Journal of Scientific Research in Science, Engineering and Technology*, 1(4), 213-217.
- Smith, J. et al. (2026). *Advancements in Air Quality Monitoring Systems: A Comprehensive Review of IoT-Based Solutions*. *International Journal of Networked Applications*, DOI:10.1155/ina/3080684.
- Synaptics Team. (2025). *Edge Computing in IoT Devices: Scalability for Environmental Monitoring*. Synaptics IoT Blog.
- Swan, M. (2012). *Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0*. *Journal of Sensor and Actuator Networks*, 1(3), 217-253.
- Wang, L. (2024). *Design of smart city environment monitoring and optimisation system based on NB-IoT technology*. *International Journal of Information and Communication Technology*, DOI:10.1504/IJICT.2024.143334.

## FLEXIBLE SCALABLE SENSOR DATALOGGER SYSTEM BASED ON NB-IOT

Nguyen Duc Xuan<sup>1</sup>, Phan Duy Hung<sup>1</sup>, Dang Khanh Hoa<sup>1</sup>

**Abstract:** Recording real-time sensor data plays a vitally important role. The collection of quality indicators, especially real-time data, will help forecast and assess environmental impacts quickly. As a result of the rapid development of Internet-of-Things (IoT) technology, air quality can be recorded, and corresponding data transmitted to servers via wireless networks. A short-range transmission technique can provide only very limited coverage and is unable to meet the communication requirements of a wide range of sensors. In addition to the transmission problem, the reception, storage, and processing of the collected data must also be supported by a large number of devices, with high reliability and easy expansion. This thesis proposes a flexible and scalable air quality monitoring system based on NB-IoT. The system includes both hardware and software solutions, providing a complete architecture to deploy a specific model in practice.

**Keywords:** data logging system, environmental monitoring system, edge computing, smart gateway, narrowBand - IoT, microservice, air quality sensor

---

<sup>1</sup> Faculty of Electric and Electronic Engineering, Hanoi, Vietnam