

DEPARTMENT OF COMPUTER AND COMMUNICATION ENGINEERING INTERNET OF THINGS: FOUNDATIONS AND APPLICATIONS MMH: ITFL416064

Group:

Name: Nguyễn Bảo Tính Đoàn Thế Hữu Pham Thế An

1. Draw and explain a block diagram of IoT System.



Sơ đồ khối hệ thống IOT

- **Thiết bị IoT** là các thiết bị vật lý như cảm biến, bộ điều khiển, hệ thống nhúng hoặc các thiết bị có khả năng thu thập hoặc truyền dữ liệu. Chúng thường kết nối với cơ sở hạ tầng mạng để thu thập dữ liệu từ môi trường vật lý.
- **Cơ sở hạ tầng mạng** đại diện cho các thành phần mạng cần thiết để kết nối các thiết bị IOT với nhau và với Internet. Nó bao gômd Router, Switch, Gateway và hệ thống mạng di động để giao tiếp dữ liệu
- **IoT Gateway** hoạt động như trung gian giữa thiết bị IoT và nền tảng đám mây. Nó thu thập dữ liệu từ nhiều thiết bị, thực hiện tiền xử lý dữ liệu và chuyển tiếp dữ liệu đến đám mây để xử lý và lưu trữ.
- Nền tảng đám mây cung cấp cơ sở hạ tầng và dịch vụ hỗ trợ bao gồm các nền tảng IoT cung cấp quản lý thiết bị, quản lý dữ liệu, bảo mật và tính mở rộng, cho phép nhà phát triển xây dựng và triển khai các ứng dụng IoT hiệu quả.
- Giao diện người dùng là giao diện để người dùng tương tác với hệ thống IoT như ứng dụng di động, bảng điều khiển dựa trên web cho phép người dùng truy cập và kiểm soát thiết bị IoT, theo dõi dữ liệu và tương tác với hệ thống.
- **Giao tiếp dữ liệu**: kênh hoặc giao thức truyền dữ liệu giữa thiết bị IoT và IoT gateway gồm các công nghệ truyền dữ liệu có dây hoặc không dây như Wi-Fi, Bluetooth, Zigbee hoặc mạng di động.
- Xử lý dữ liệu: là nền tảng đám mây nơi dữ liệu được xử lý, lưu trữ và phân tích. Bao gồm các thành phần như lưu trữ dữ liệu, động cơ xử lý dữ liệu, công cụ phân tích và thuật toán học máy để trích xuất thông tin ý nghĩa từ dữ liệu.
- Dịch vụ ứng dụng: Các dịch vụ này được xây dựng trên nền tảng đám mây và cung cấp các chức năng hoặc ứng dụng cụ thể phù hợp với nhu cầu người dùng. Bao gồm việc hiển thị dữ liệu, theo dõi thời gian thực, cảnh báo, tự động hóa hoặc bất kỳ dịch vụ nào khác sử dụng dữ liệu đã được xử lý từ hệ thống IoT.

2. Features of CSS, HTML, JAVASCRIPT.

CSS

- Cascading Style Sheet là một ngôn ngữ lập trình được thiết kế vô cùng đơn giản, dễ sử dụng. Nhiệm vụ chính của CSS là thực hiện việc xử lý giao diện của một trang web cụ thể như màu sắc văn bản, hay khoảng cách giữa các đoạn, hoặc kiểu font chữ, hình ảnh, bố cục, màu nền,...
- Ngôn ngữ CSS có một số ưu điểm như sau:
 - Tăng tốc độ tải trang: CSS cho phép bạn sử dụng ít đoạn mã vì vậy tốc độ tải trang sẽ được cải thiện đáng kể.
 - Cải thiện trải nghiệm người dùng: CSS giúp các website có định dạng thân thiện với người dùng.
 Khi các nút và văn bản ở vị trí hợp lý và được sắp đặt tốt, trải nghiệm người dùng sẽ được cải thiện.
 - Thời gian phát triển nhanh: có thể áp dụng các quy tắc và kiểu định dạng cụ thể cho nhiều trang bằng một chuỗi mã. Một biểu định kiểu xếp tầng có thể được sao chép trên một số trang web.
 - Thay đổi định dạng dễ dàng: thay đổi định dạng của một nhóm trang cụ thể. Chỉ cần chỉnh sửa biểu định kiểu CSS tương ứng và bạn sẽ thấy các thay đổi được áp dụng cho tất cả các trang đang sử dụng biểu định kiểu đó.
 - Khả năng tương thích trên các thiết bị: CSS kết hợp với HTML để tạo ra thiết kế đáp ứng.

HTML

- Hypertext Markup Language là ngôn ngữ lập trình dùng để xây dựng và cấu trúc lại các thành phần có trong Website. Thường sử dụng HTML trong việc phân chia các đoạn văn, heading, links, blockquotes,...
- Ưu điểm:
 - Thân thiện với người mới bắt đầu: HTML có một đánh dấu rõ ràng và nhất quán, cũng như một đường cong học tập nông và dễ hiểu.
 - Tính ủng hộ: Ngôn ngữ này được sử dụng rộng rãi, với rất nhiều tài nguyên và một cộng đồng lớn đằng sau nó.
 - Dễ dàng truy cập: HTML là mã nguồn mở và hoàn toàn miễn phí. HTML chạy tự nhiên trong tất cả các trình duyệt web.
 - Linh hoạt: HTML có thể dễ dàng tích hợp với các ngôn ngữ phụ trợ như PHP và Node.js.

Javascript

- Chính là một ngôn ngữ lập trình web rất phổ biến ngày nay. Javascript được tích hợp đồng thời nhúng vào HTML để hỗ trợ cho website trở nên sống động hơn.

- Là một ngôn ngữ lập trình phổ biến sử dụng trên 92% nền tảng website hiện nay, JavaScript đã thể hiện vai trò quan trọng với lĩnh vực này. Tầm quan trọng của nó thể hiện qua các thao tác, công dụng như:
 - Triển khai tập lệnh phía máy khách: Nhờ Javascript, các lập trình viên có thể dễ dàng viết tập lệnh phía máy khách, tích hợp các tập lệnh một cách liền mạch vào HTML, cho phép website tương tác, trả lời người dùng ngay lập tức và tạo ra giao diện hiển thị phong phú hơn.
 - Viết mã phía máy chủ: Các lập trình viên có thể viết mã phía máy chủ bằng JavaScript.
 - Đơn giản hóa phát triển ứng dụng web phức tạp: Javascript cho phép các nhà phát triển đơn giản hóa thành phần của ứng dụng, qua đó đơn giản hóa việc phát triển các ứng dụng web phức tạp.
 - Thiết kế web responsive: JavaScript cho phép thiết kế web responsive tối ưu trên cả máy tính và thiết bị di động chỉ với một bộ mã.
 - Google AMP: Để tham gia vào dự án Tăng tốc trang di động (AMP) của Google, các lập trình viên phải sử dụng ngôn ngữ lập trình JavaScript. Bạn sẽ hiểu rõ hơn về Google AMP qua bài viết: Google AMP là gì?
 - Nhiều bộ chuyển đổi: Mặc dù thiếu một số tính năng phức tạp được cung cấp bởi các ngôn ngữ lập trình hiện đại như Java và C#, JavaScript vẫn có thể dễ dàng mở rộng bằng cách sử dụng các bộ chuyển đổi như CoffeeScript, TypeScript, DukeScript và Vaadin.

DOMAIN

3. What is HOSTING? DOMAIN? Give Example.

HOSTING

Domain hay còn gọi là tên miền, đây là một chuỗi ký Hosting là nơi lưu trữ dữ liệu của Website chạy trên tự được sử dụng để xác định một trang web cụ thể môi trường online, nơi chứa Code, Data, hình ảnh, ... trên internet. Tên miền thường được sử dụng để thay Giống như trên máy tính của mình, thì hosting là ổ thế cho địa chỉ IP để làm việc truy cập trang web dễ đĩa lưu trữ dữ liệu. Khi Domain(Tên miền) được trỏ về Hosting, thì có dàng và dễ nhớ. Chỉ cần nhập đúng địa chỉ tên miền thể vào Hosting bằng cách nhập đúng tên miền. trên trình duyệt, thì người dùng có thể truy cập để Để Website có thể chạy cần có Code Web và Data xem Websitre của chúng ta. - Có nhiều giao diện quản lý Hosting: Cpanel, - Các loai Domain DirectAdmin, ... Nhưng những tính năng sẽ gần + Domain Việt Nam:Gồm các domain như: .vn, giống nhau chỉ khác nhau về giao diện. .com.vn, .edu.vn, .gov.vn, biz.vn, ... + Domain Quốc tế:Gồm các domain như: .com, .net, - Những nơi bán Hosting uy tín: .biz, .US, .info, ... + Việt Nam: AZDigi, Inet, Mắt Bão, Nhân Hòa, ...

- + Nước ngoài: Hawkhost, SiteGround, A2 Hosting, GoDaddy, ...
- Những điều cần lưu ý khi mua Hosting
- + Chọn nhà cung cấp uy tín trên thị trường
- + Không mua từ 1 cá nhân, công ty nhỏ
- + Không nhờ người khác mua hộ
- + Hiểu cơ bản về hosting và nhớ gia hạn hàng năm

- Những nơi mua Domain uy tín

Thông thường nên mua domain .com vì nó thông dụng nhất và dễ nhớ nhất. Nếu ở Việt Nam bạn mua thêm .VN để đảm bảo quyền lợi.

- + Trong nước: Inet, Pavietnam, Mắt Bão, Nhân Hòa, TenTen,..
- + Nước ngoài: Namesilo, Namecheap, Godaddy, Google, Z.com, ...
- Quản lý Domain bao gồm
- + Tài khoản quản trị từ nhà cung cấp Domain
- + Dùng Email của bản thân để mua Domain
- + Nơi quản lý DNS do nhà cung cấp cho bên mua
- Những điều cần lưu ý khi mua Domain
- + Mua bằng Email chính chủ
- + Không nhờ người khác mua
- + Điền đúng thông tin để đảm bảo quyền lợi
- + Kiểm tra mail thường xuyên để nhớ gia hạn
- + Mua nhiều Chấm để bảo vệ thương hiệu
- 4. What is a gateway? The role of Gateway in the IoT System?

What is a gateway?

Thường được hiểu là một cổng kết nối hoặc giao diện mà thiết bị IoT sử dụng để trao đổi dữ liệu với mạng hoặc các thiết bị khác trong hệ thống.

- Truyền thông giữa các thiết bị: Nếu một công ty đã triển khai các loại thiết bị khác nhau từ các nhà cung cấp khác nhau, họ có thể không thể liên lạc trực tiếp. Cổng IoT có thể hoạt động như một trung tâm trung tâm và thực hiện các bản dịch cần thiết để cho phép liên lạc giữa các thiết bị.
- Truyền thông từ thiết bị đến đám mây: Các thiết bị IoT thường gửi dữ liệu đến cơ sở hạ tầng dựa trên đám mây để xử lý và sử dụng trong các ứng dụng. Cổng IoT tổng hợp dữ liệu từ nhiều thiết bị, cung cấp một điểm liên lạc duy nhất cho cơ sở hạ tầng đám mây.
- Xử lý trước dữ liệu: Các thiết bị IoT có thể thu thập một lượng lớn dữ liệu, đòi hỏi một lượng băng thông đáng kể để gửi đến các ứng dụng dựa trên đám mây để xử lý. Cổng IoT thực hiện lọc và xử lý trước dữ

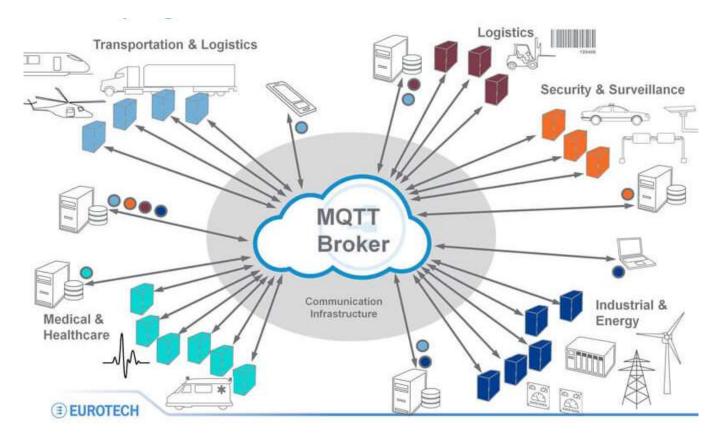
liệu để giảm khối lượng dữ liệu được gửi lên đám mây.

- Bảo mật thiết bị IoT: Các thiết bị IoT nổi tiếng có mức độ bảo mật kém, khiến chúng không thể truy cập trực tiếp từ Internet công cộng. Cổng IoT nằm giữa các thiết bị IoT và Internet và có thể bao gồm chức năng bảo mật tích hợp để giúp bảo vệ các thiết bị này khỏi bị tấn công.

The role of Gateway in the IoT System?

- Gateway trong hệ thống IoT (Internet of Things) có vai trò quan trọng như một trung gian kết nối giữa các thiết bị IoT và mạng lớn hơn, thường là internet hoặc mạng nội bộ của tổ chức.
- Một số vai trò quan trọng:
- + **Kết nối và Giao tiếp:** Gateway giúp các thiết bị IoT có thể kết nối và giao tiếp với nhau và với mạng. Với sự phức tạp của hệ thống IoT, các thiết bị có thể sử dụng các giao thức và chuẩn giao tiếp khác nhau. Gateway có khả năng chuyển đổi và tương tác giữa các giao thức này để đảm bảo thông tin liên lạc được truyền tải một cách hiệu quả.
- + **Thu thập và Xử lý dữ liệu**: Gateway có khả năng thu thập dữ liệu từ các thiết bị IoT và thậm chí có thể tiến hành một số xử lý dữ liệu cơ bản trước khi gửi dữ liệu lên mạng. Điều này có thể giảm tải cho mạng chính và tránh việc truyền tải quá nhiều dữ liệu không cần thiết.
- + **Bảo mật và Quản lý:** Gateway có thể chứa các tính năng bảo mật như mã hóa, xác thực và kiểm soát truy cập để đảm bảo tính an toàn cho hệ thống IoT. Nó cũng có thể thực hiện việc quản lý và giám sát thiết bị, kiểm tra trạng thái và cập nhật phần mềm từ xa.
- + **Chuyển tiếp dữ liệu:** Trong các hệ thống lớn hơn hoặc phức tạp hơn, Gateway có thể chuyển tiếp dữ liệu từ các thiết bị IoT đến các hệ thống hoặc dịch vụ xử lý dữ liệu lớn (big data) hoặc điều khiển tự động.
- + Quản lý địa chỉ IP: Các thiết bị IoT thường cần địa chỉ IP để truyền tải dữ liệu qua mạng. Gateway có thể quản lý các địa chỉ IP và thực hiện chức năng NAT (Network Address Translation) để định tuyến dữ liệu giữa các thiết bị và mạng ngoại vi.
- + **Tích hợp hệ thống cũ:** Gateway có thể đóng vai trò kết nối giữa các thiết bị IoT mới và các hệ thống cũ hơn mà không hỗ trợ giao tiếp IoT trực tiếp.
 - 5. Read "IoT Solution for Smart City" and answer these questions:
 - a. What does a smart city may involve?
 - Mạng viễn thông số giúp kết nối các thiết bị thông minh với nhau.
 - Hệ thống nhúng thông minh điều hành các thiết bị.
 - Các cảm biến thu thập thông tin.
 - Phần mềm giao tiếp với con người.
 - Hệ thống lưu trữ thông tin.

- Toàn bộ hệ thống này khi vận hành trong đô thị sẽ giúp nâng cao chất lượng cuộc sống, cải thiện chất lượng phục vụ của chính quyền thành phố, giảm tiêu thụ năng lượng và giúp quản lý hiệu quả các nguồn tài nguyên thiên nhiên.
- b. What can be measured?
 - Chất lượng không khí.
 - Mức độ âm thanh và tiếng ồn.
 - Thông số khí tượng.
 - Mức độ chất thải.
- c. Applications of IoT technology for smart cities.
 - Đo mức độ ô nhiễm không khí trong không gian đô thị và công nghiệp.
 - Giám sát mức độ tiếng ồn trong không gian đô thị và công nghiệp.
 - Giám sát tình trạng chỗ đậu xe trong thành phố. Giám sát phương tiện và mức độ người đi bộ để tối ưu hóa các tuyến đường lái xe và đi bộ.
 - Kiểm soát mức độ lấp đầy của thùng chứa chất thải công cộng hoặc công nghiệp để tối ưu hóa dịch
 vụ thu gom và tuyến đường vận chuyển.
 - Kiểm soát mức độ lấp đầy của thùng chứa chất thải công cộng hoặc công nghiệp để tối ưu hóa dịch
 vụ thu gom và tuyến đường vận chuyển.
 - Quản lý bảo trì và tiết kiệm nước tiêu thụ công cộng tại các công viên, khu vực cây xanh, v.v.
- 6. MQTT, CoAP, AMQP, DDS, XMP
- **MQTT** (Message Queue Telemetry Transport)

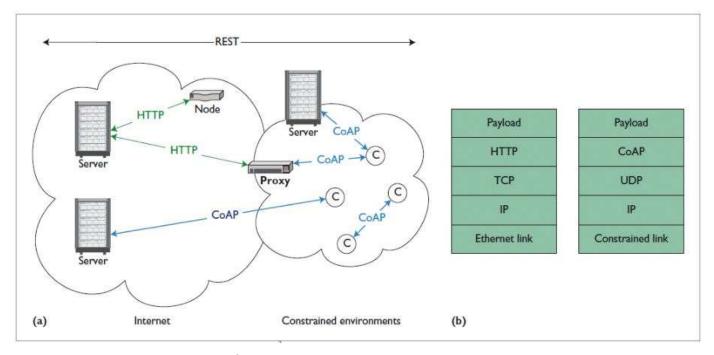


Mô hình sử dụng giao thức MQTT

- MQTT là một giao thức mã nguồn mở để truyền các messages giữa nhiều Client (Publisher và Subscriber) thông qua một Broker trung gian, được thiết kế để đơn giản và dễ dàng triễn khai. Kiến trúc MQTT dựa trên Broker trung gian và sử dụng kết nối TCP long-lived từ các Client đến Broker.
- MQTT hỗ trợ tổ chức hệ thống theo các Topics có tính phân cấp, như một hệ thống tập tin (vd: /Home/kitchen/humidity), cung cấp nhiều lựa chọn điều khiển và QoS (Quality of Service).
- MQTT là một giao thức khá nhẹ nên có thể được sử dụng cho truyền thông 2 chiều thông qua các mạng có độ trễ cao và độ tin cậy thấp, nó cũng tương thích với các thiết bị tiêu thụ điện năng thấp.

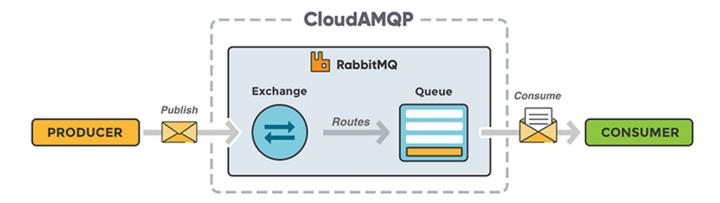
Let Coal Constrained Applications Protocol

- CoAP là một giao thức truyền tải tài liệu theo mô hình client/server dự trên internet tương tự như giao thức HTTP nhưng được thiết kế cho các thiết bị ràng buộc. Giao thức này hỗ trợ một giao thức one-to-one để chuyển đổi trang thái thông tin giữa client và server.
- CoAP sử dụng UDP (User Datagram Protocol), không hỗ trợ TCP, ngoài ra còn hỗ trợ địa chỉ broadcast và multicast, truyền thông CoAP thông qua các datagram phi kết nối (connectionless) có thể được sử dụng trên các giao thức truyền thông dựa trên các gói.
- UDP có thể dễ dàng triển khai trên các vi điều khiển hơn TCP nhưng các công cụ bảo mật như SSL/TSL không có sẵn, tuy nhiên ta có thể sử dụng Datagram Transport Layer Security (DTLS) để thay thế.



Ví dụ về mô hình sử dụng giao thức CoAP và HTTP

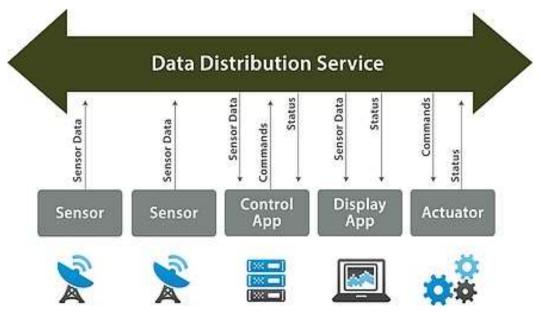
4 AMQP (Advanced Message Queue Protocol)



Mô hình sử dụng giao thức AMQP

- AMQP là một giao thức làm trung gian cho các gói tin trên lớp ứng dụng với mục đích thay thế các hệ thống truyền tin độc quyền và không tương thích. Các tính năng chính của AMQP là định hướng message, hàng đợi, định tuyến (bao gồm point-to-point và publish-subscribe) có độ tin cậy và bảo mật cao. Các hoạt động sẽ được thực hiện thông qua broker, nó cung cấp khả năng điều khiển luồng (Flow Control).
- Một trong các Message Broker phổ biến là RabbitMQ, được lập trình bằng ngôn ngữ Erlang, RabbitMQ cung cấp cho lập trình viên một phương tiện trung gian để giao tiếp giữa nhiều thành phần trong một hệ thống lớn.
- Không giống như các giao thức khác, AMQP là một giao thức có dây (wire-protocol), có khả năng diễn tả các message phù hợp với định dạng dữ liệu, có thể triển khai với rất nhiều loại ngôn ngữ lập trình.

DDS (Data Distribution Service)

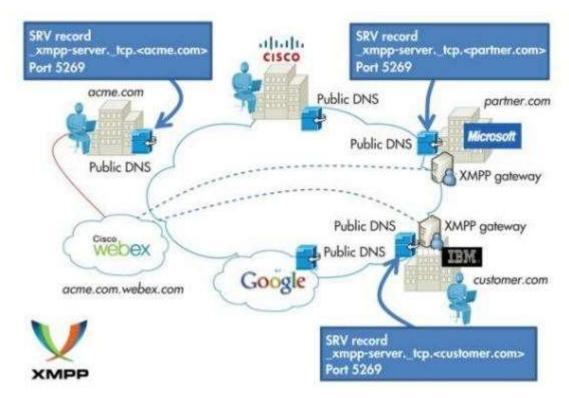


Một ví dụ về DDS

- DDS là một ngôn ngữ trung gian dựa vào dữ liệu tập trung được sử dụng để cho phép khả năng mở rộng, thời gian thực, độ tin cậy cao và trao đổi dữ liệu tương tác.
- Đây là một giao thức phi tập trung (broker-less) với truyền thông ngang hàng trực tiếp theo kiểu peer-to-peer giữa các publishers và subscribers và được thiết kế để trở thành một ngôn ngữ và hệ điều hành độc lập. DDS gửi và nhận dữ liệu, sự kiện, và thông tin lệnh trên UDP nhưng cũng có thể chạy trên các giao thức truyền tải khác như IP Multicast, TCP/IP, bộ nhớ chia sẻ,... DDS hỗ trợ các kết nối được quản lý many-to-many theo thời gian thực và ngoài ra còn hỗ trợ dò tìm tự động (automatic discovery). Các ứng dụng sử dụng DDS cho truyền thông được tách riêng và không yêu cầu sự can thiệp từ các ứng dụng của người dùng, có thể đơn giản hóa việc lập trình mạng phức tạp. Các tham số QoS được sử dụng để xác định các cơ chế tự dò tìm của nó được thiết lập một lần.

4 XMPP (Extensible Messaging và Presence Protocol)

- XMPP (trước đây gọi là "Jabber") là giao thức truyền thông dùng cho định hướng tin nhắn trung gian dựa trên ngôn ngữ XML.
- XMPP là mô hình phân quyền client-server phi tập trung, được sử dụng cho các ứng dụng nhắn tin văn bản. Có thể nói XMPP gần như là thời gian thực và có thể mở rộng đến hàng trăm hàng nghìn nút. Dữ liệu nhị phân phải được mã hóa base64 trước khi nó được truyền đi trong băng tần. XMPP tương tự như MQTT, có thể chạy trên nền tảng TCP.



Một ví dụ về XMPP

References

