Contents

1. So sánh CoAP, MQTT, XMPP

	CoAP	MQTT	XMPP
Transport	UDP	TCP	TCP
Mô hình truyền thông	Request-Reply (REST) One-to-one	Publish-Subscribe via broker Many-to-many	Request-Reply & Publish-Subscribe Point-to-Point Message Exchange
QoS	1 level	3 levels	Không có (Có thể thêm vào)
Security	DTLS	SSL/TLS	SASL/TLS
Header size	4 bytes	2 bytes	-
Scope	D2D	D2C, C2C	D2C, C2C
Khả năng lỗi	Decentralized	Broker is SPoF	Server is SPoF
Kiến trúc	Cây	Cây	Client-Server
Compute Resources	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
Thực tiễn	Utility Field Area Networks	Extending enterprise messaging into IOT applications	Remote management of consumer white goods

D2D: Device-to-Device

D2C: Device-to-Cloud

C2C: Cloud-to-Cloud

2. Zigbee

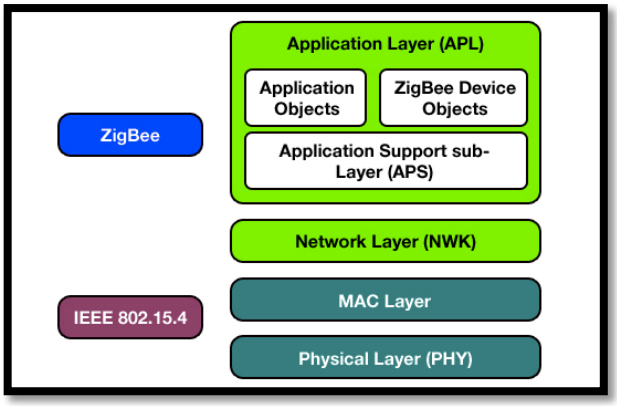
Tổng quan

Công nghệ ZigBee được xây dựng dựa trên tiêu chuẩn 802.15.4 của tổ chức IEEE. Tiêu chuẩn 802.15.4 này sử dụng tín hiệu radio có tần sóng ngắn, và cấu trúc của 802.15.4 có 2 tầng là tầng vật lý và tầng MAC (medicum Access Control). 2 tầng còn lại của Zigbee là Network layer và Application layer

Kiến trúc

. Tầng vật lý

có trách nhiệm truyền tín hiệu không dây với chi phí thấp đồng thời giữ cho việc truyền tín hiệu được mạnh trong môi trường nhiễu. Có 27 channel (0-26)



. Tầng MAC:

Có các tính năng:

Mở kết nối và đóng kết nối

Gửi gói ACK

Có hỗ trợ CSMA-CA, slotted CSMA-CA

Beacon management

GTS (guaranteed time slot) management

. Tầng mạng

Có tác dụng tham gia vào mạng zigbee, xác định đường định tuyến trong mạng.

. Tầng ứng dụng

Gồm 3 thành phần cơ bản:

+ Application support sub-layer (APS)

APS là tầng kết nối với tầng mạng và là nơi cài đặt những ứng dụng cần cho ZigBee, giúp lọc bớt các gói dữ liệu trùng lắp từ tầng mạng

+ ZigBee Device Objects (ZDO)

ZDO có trách nhiệm quản lý các thiết bị, định hình tầng hỗ trợ ứng dụng và tầng mạng, cho phép thiết bị tìm kiếm, quản lý các yêu cầu và xác định trạng thái của thiết bị.

+ Application Objects (APO)

APO: là tầng mà ở đây người dùng tiếp xúc với thiết bị, tầng này cho phép người dùng có thể tuỳ biến thêm ứng dụng vào hệ thống.

Mô hình mạng

ZigBee có 3 dạng hình mạng được hỗ trợ bởi ZigBee: dạng hình sao, hình lưới, và hình cây. Mỗi dạng hình đều có những ưu điểm riêng và được ứng dụng trong các trường hợp khác nhau.

. Hình sao (Star network)

Các nút hình sao liên kết với nút trung tâm.

. Hình lưới (Mesh network)

Mạng hình lưới có tính tin cậy cao, mỗi nút trong mạng lưới đều có khả năng kết nối với nút khác, nó cho phép truyền thông liên tục giữa các điểm nút với nhau và bền vững. Nếu có sự tác động cản trở, hệ thống có khả năng tự xác định lại cấu hình bằng cách nhảy từ nút này sang nút khác.

. Hình cây (Cluster network)

Mạng hình này chính là 1 dạng đặc biệt của mạng hình lưới, dạng mạng này có khả năng phủ sóng và mở rộng cao.

Đánh địa chỉ trong mạng Tree

Coordinator luôn có địa chỉ là 0.

Cấp địa chỉ cho router trước cho end-device

Cấp địa chỉ theo thứ tự từ trong ra ngoài trên cùng một nhánh trước khi qua nhánh khác.

Coordinator luôn cố định:

R_m: là số lượng router con tối đa mà mỗi node quản lý.

D_m: là số lượng end-device con tối đa mà mỗi node quản lý

L_m: là độ sâu tối đa của mô hình Tree

Một router có địa chỉ là **x** nằm ở độ sâu **d** sẽ được cấp một miền địa chỉ là:

Bước 1: Tính **A(d)** cho từng vòng độ sâu **d.**

Trong đó A(d) được tính:

$$A(d) = 1 + D_m + R_m \quad \text{if } d = L_m - 1$$

or

$$A(d) = 1 + D_m + R_m A(d+1) \quad \text{if } 0 \le d < L_m - 1$$

Bước 2: Đánh địa chỉ cho router

Bước 3: tính range địa chỉ mà router đó được cấp

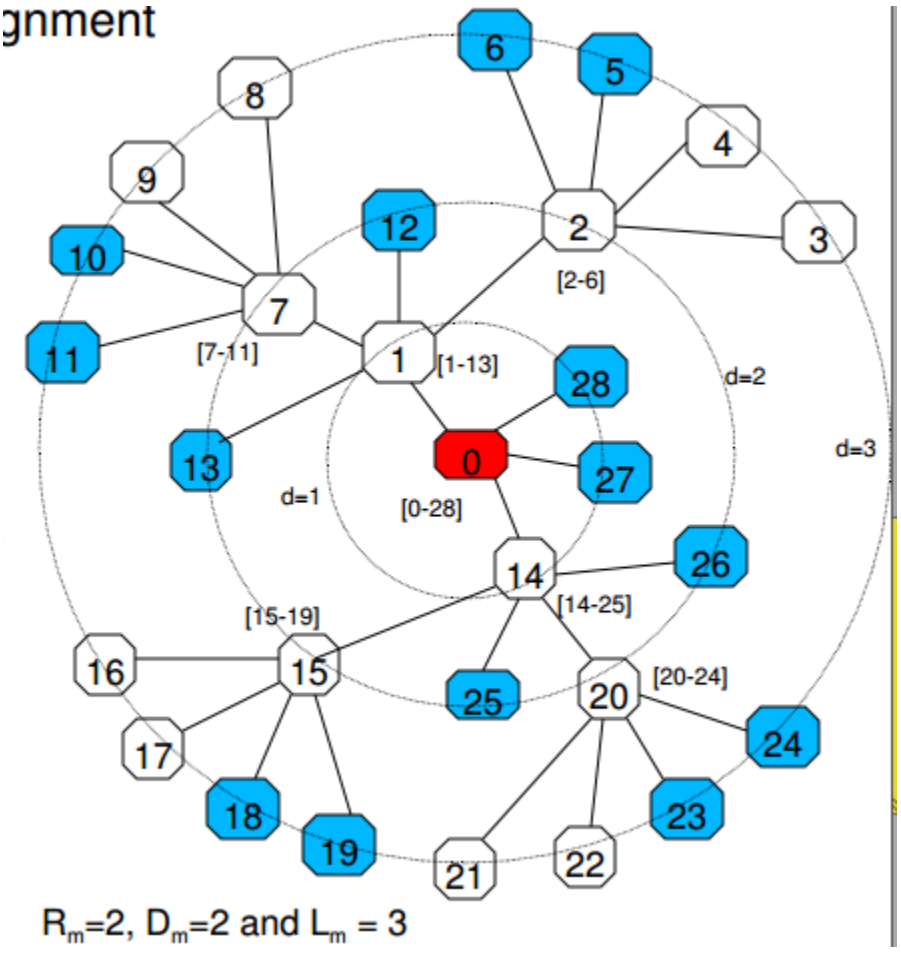
Bước 4: Nếu chưa đến vòng cuối thì lặp lại bước 2. Ngược lại chuyển qua bước 5

Bước 5: Đánh địa chỉ cho end-device tại vòng cuối. Quay lại vòng trước nó và lặp lại bước 5 cho đến khi gặp coordinator.

Lưu ý, Trong cùng một nhánh, các thiết bị cùng vòng độ sâu **d**, đánh địa chỉ cho router trước đánh địa chỉ cho end-device.

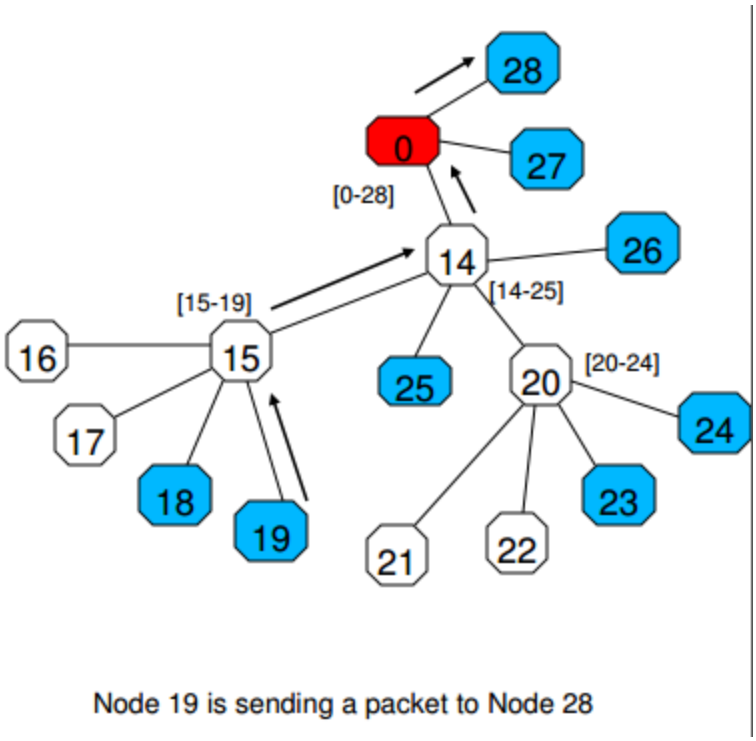
Xem slide để hiểu hơn!

Ví dụ:



Định tuyến

Mô hình cây (tree based routing)



Node 19 is sending a packet to Node 28

Do đã được gắn địa chỉ theo từng nhánh nên router có thể nhận biết được địa chỉ đích mà node cần route tới là thuộc nhánh của mình (dựa trên range ip được cấp) hay là thuộc về note parent của nó (node parent có thể là router hoặc là coordinator)

Có 2 bước route:

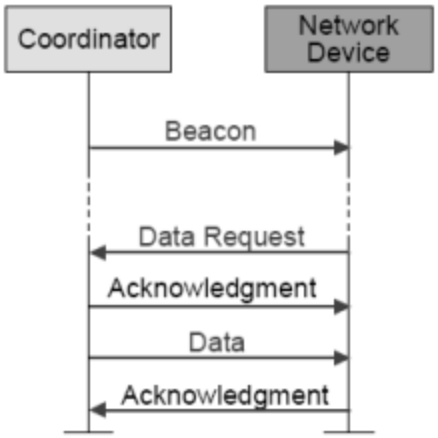
- Nếu địa chỉ đích thuộc range địa chỉ mà router được cấp => địa chỉ đích là một node con của router đó. Lúc đó nó định tuyến trực tiếp đến node con
- Nếu địa chỉ đích không thuộc range địa chỉ mà router được cấp => địa chỉ đích không thuộc node con của nó, lúc này nó sẽ route lên node parent.

Ví dụ từ node 16 – 28:

- 28 không thuộc range của node 16 -> chuyển node cha (15)
- 28 không thuộc range của node 15 [15-19] -> chuyển node cha (14)
- 28 không thuộc range của node 14 [14-25] -> chuyển node cha (0)
- 28 là node con của node 0. Trả kết quả route về lại cho 16 theo đường ngược lại.

*** Giao tiếp trong giao thức 802.15.4 (MAC layer)**

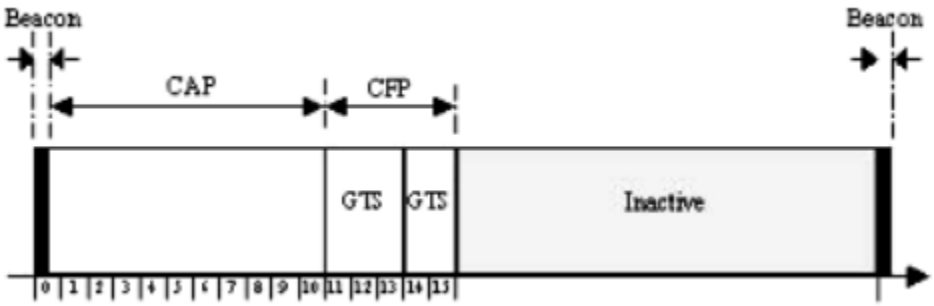
. Beacon-enabled mode: (Cơ chế này cho khả năng đa truy cập)



Cơ chế này còn có tên là **Superframe**

Trong mode này, định kỳ khoảng 15ms -245s thì Coordinator sẽ gửi 1 gói tin Beacon (broadcast) ra ngoài môi trường không dây.

Khoảng thời gian giữa 2 Beacon được chia thành 16 time-slots. Các time-slot này được chia là 2 phần: contention-access period (CAP) and contention-free period (CFP).



Contention tạm dịch là tranh chấp.

Ưng với mỗi time-slot một thiết bị Network Device được phép gửi Data Request đến Coordinator.

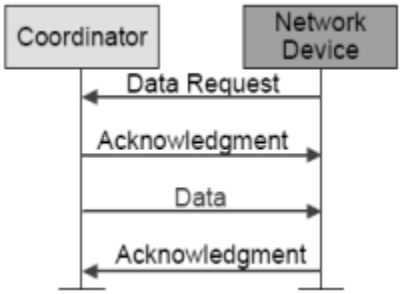
Ở phần CAP, Các node sẽ được kết nối với coordinator và quản lý theo cơ chế CSMA-CA.

Ở phần CFP, Các node sẽ yêu cầu trước một đường truyền đảm bảo để kết nối với Coordinator.

Superframe có thể có khoảng thời gian inactive để tiết kiệm năng lượng.

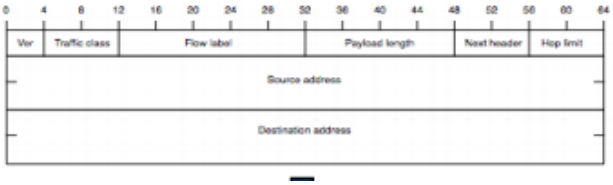
. Nonbeacon-enabled mode:

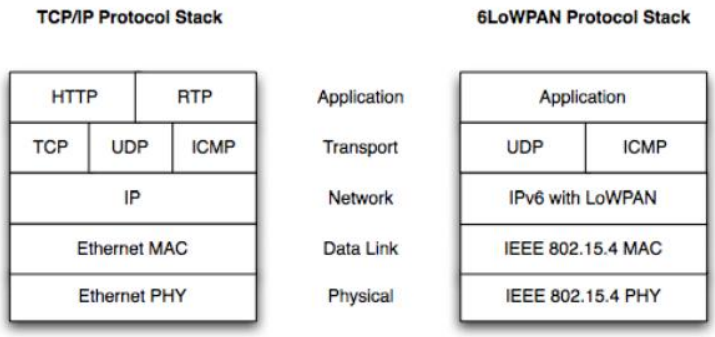
Ở mode này không thể quản lý tranh chấp. Coordinator thụ động chờ Network Device gửi yêu cầu gửi gói tin.



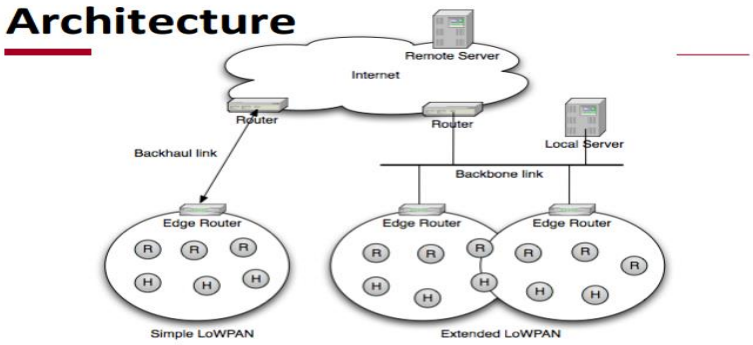
3. 6LowPan

- Một lớp thích ứng để phù hợp với IPv6 qua mạng không dây công suất thấp
- Được xác định bởi các tiêu chuẩn IETF
  - RFC 4919, 4944
  - draft-ietf-6lowpan-hc and -nd
  - draft-ietf-roll-rpl

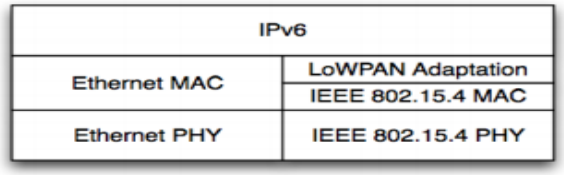




- ***Lợi ích của công nghệ 6LoWPAN:***
  - Công suất thấp RF + IPv6 = Công nghệ Internet nhúng không dây
  - 6LoWPAN làm cho điều này có thể
  - Những lợi ích của 6LoWPAN bao gồm:
    - Các tiêu chuẩn mở, tuổi thọ dài, đáng tin cậy
    - Dễ dàng learning-curve
    - Tích hợp Internet trong suốt
    - Khả năng bảo trì mạng
    - Khả năng mở rộng toàn cầu
    - Cho phép một ổ cắm chuẩn API
    - Sử dụng tối thiểu mã và bộ nhớ
    - Tích hợp Internet trực tiếp đầu cuối: Nhiều tuỳ chọn topology
- ***Kiến trúc***



- LoWPANs là các mạng lưới sơ khai
- LoWPAN đơn giản
    - Router biên đơn giản
- Extended LoWPAN
    - Bộ định tuyến LoWPANMultiple Edge mở rộng có liên kết xương sống chung
- Ad-hoc LoWPAN
    - Không có tuyến đường bên ngoài LoWPAN
- Các vấn đề tích hợp Internet
    - Đơn vị truyền tối đa
    - Giao thức ứng dụng
    - Khả năng kết nối IPv4
    - Tường lửa và NAT
    - Bảo vệ



IPv6-LoWPAN Router Stack

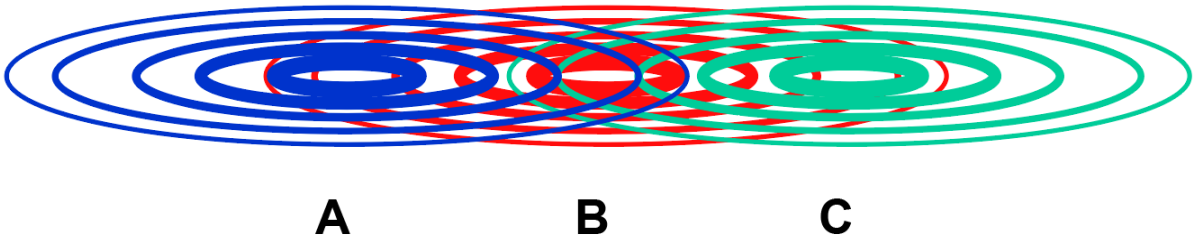
- ***Tính năng thích ứng***

- Nén header hiệu quả
    - Header cơ sở và phần mở rộng IPv6, header UDP
- Phân mảnh
    - Khung 1280 bytes IPv6 MTU -> 127 byte 802.15.4

- ***Tính năng bổ sung***

- Hỗ trợ cho ví dụ Địa chỉ 64-bit và 16-bit 802.15.4
- Có ích với các lớp liên kết công suất thấp như IEEE 802.15.4, ISM băng tầng hẹp và truyền thông đường dây điện
- Tự động định cấu hình mạng sử dụng phát hiện neighbor
- Hỗ trợ Unicast, multicast và broadcast
    - Multicast được nén và lập bản đồ đến broadcast
- Hỗ trợ định tuyến IP (ví dụ: IETF RPL)
- Hỗ trợ sử dụng lưới liên kết (ví dụ 802.15.5)

4. Hidden/Exposed terminal
Hidden terminals



- A không thể nghe truyền dẫn từ node C và ngược lại (do không nằm trong vùng phủ sóng của nhau)

- A truyền đến B

- C đồng thời truyền đến B vì thấy B đang rảnh (Carrier Sense fails)

- Xung đột xảy ra tại B

- A không thể nhận được thông tin về sự kiện đụng độ tại B (Collision Detection fails)

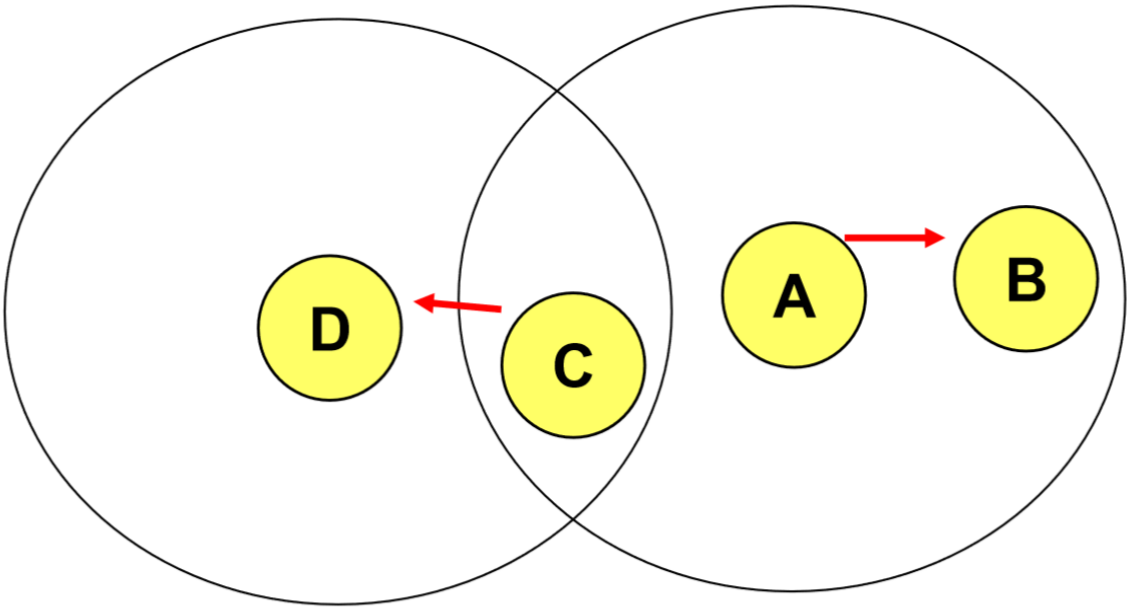
- A “hidden” đối với C

* Giải quyết bằng cách sử dụng cơ chế RTS/CTS (Request to send/Clear to send)

- A broadcasts thông điệp RTS trong vùng phủ sóng của nó

- B rảnh và trả lời với 1 thông điệp CTS khiến các nodes khác nằm trong vùng phủ sóng của B (trừ A) phải chuyển vào trạng thái chờ (có thể dẫn đến exposed terminals)

Exposed terminals



- A bắt đầu truyền tới B

- C muốn truyền tới D nhưng nhận thấy có sự kiện truyền và phải đợi việc truyền từ A tới B kết thúc

- D do đó cũng phải đợi mặc dù nằm ngoài vùng phủ sóng của A (không cần thiết)

- A và C là “exposed” terminals

5. AODV vs OLSR
OLSR (Optimized Link State Routing) – Proactive Protocol

Khái niệm:

Giao thức định tuyến proactive (table-driven)

+ 1 tuyến đường sẵn sàng ngay lập tức khi cần

Dựa trên thuật toán link-state

Các nodes chỉ quảng bá thông tin các liên kết với các láng giềng mà nằm trong tập lựa chọn đa điểm chuyển tiếp (multipoint relay selector set)

+ Giảm kích thước các gói tin điều khiển

Giảm flooding bằng cách sử dụng các multipoint relay nodes để gửi thông tin trong mạng

+ Giảm số lượng các gói tin điều khiển bằng cách giảm các truyền dẫn trùng lặp

Không yêu cầu chuyển giao tin cậy, bởi vì các cập nhật được gửi định kỳ

Không cần chuyển phát có trật tự, bởi vì sequence numbers được sử dụng để ngăn chặn thông tin hết hạn khỏi bị diễn giải sai

Sử dụng định tuyến hop-by-hop

Các tuyến đường được dựa trên các mục bảng động được duy trì tại các nodes trung gian

Ưu điểm

OLSR có độ trễ trung bình giữa các điểm đầu cuối thấp hơn

Thân thiện với người dùng hơn

Giao thức định tuyến phẳng, không cần hệ thống quản lý trung tâm để xử lý các quá trình định tuyến

Tăng cường độ thích hợp các giao thức cho mạng ad hoc với sự thay đổi liên tục các cặp nguồn và đích

Giao thức OLSR không yêu cầu liên kết đáng tin cậy cho các thông điệp điều khiển, bởi vì các thông điệp được gửi định kỳ và việc vận chuyển không cần phải theo thứ tự

Đơn giản trong việc sử dụng các interfaces, dễ dàng tích hợp giao thức định tuyến vào trong các hệ điều hành có sẵn mà không cần thay đổi định dạng header của các thông điệp IP. Giao thức chỉ tương tác với bảng định tuyến của host

Khuyết điểm

Cần nhiều thời gian hơn để khám phá lại 1 liên kết đã bị hỏng

Độ trễn phân phối lớn hơn

Yêu cầu nhiều năng lượng xử lý hơn khi tìm thấy 1 tuyến đường thay thế

AODV (Ad Hoc On Demand Distance Vector) – Reactive Protocol

Khái niệm:

1 giao thức định tuyến chỉ thực hiện theo yêu cầu thuần túy:

+ 1 node không thực hiện khám phá hoặc bảo trì đường đi cho đến khi nó cần 1 đường đi tới 1 node khác hoặc nó đề nghị các dịch vụ của nó như là 1 node trung gian.

+ Các nodes không ở trên các đường hoạt động không bảo trì thông tin định tuyến và không tham gia trao đổi bảng định tuyến.

Sử dụng cơ chế broadcast route discovery

Sử dụng định tuyến hop-by-hop:

+ Các tuyến đường dựa trên các mục bảng định tuyến động đã được bảo trì tại các nodes trung gian

+ Tương tự như Dynamic Source Routing (DSR), nhưng DSR sử dụng định tuyến nguồn

Các thông điệp HELLO cục bộ được sử dụng để quyết định khả năng kết nối cục bộ:

+ Có thể giảm thời gian đáp ứng cho các yêu cầu định tuyến

+ Có thể kích hoạt các cập nhật khi cần thiết

Sequence numbers có thể được gắn vào các tuyến và các mục bảng định tuyến:

+ Được sử dụng để thay thế các mục định tuyến cũ trong cache

Mỗi node giữ 2 bộ đếm:

+ Node sequence number

+ Broadcast ID

Ưu điểm:

Giao thức AODV là 1 giao thức định tuyến phẳng không cần hệ thống hành chính trung tâm để xử lý tiến trình định tuyến.

Tránh được lỗi đếm tới vô tận (counting to infinity problem).

Có chia sẻ băng thông cao hơn.

Cố gắng giữ phần đầu của các thông điệp nhỏ. Nếu host có thông tin tuyến trong bảng định tuyến về các tuyến đường hoạt động trong mạng thì phần đầu của tiến trình định tuyến sẽ tối thiểu. (Các giao thức thông thường cần phải giữ tất cả tuyến đường từ nguồn tới đích trong các thông điệp)

Khuyết điểm:

Có nhu cầu xử lý cao hơn.

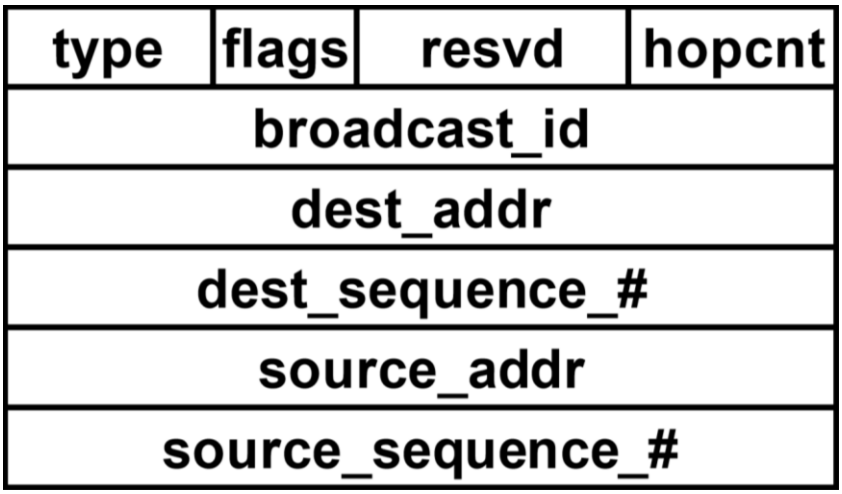
Tiêu tốn phần chia sẻ băng thông nhiều hơn.

Cần nhiều thời gian hơn để xây dựng bảng định tuyến.

Cơ chế hoạt động AODV:

*Route Request

Khám phá tuyến đường (Route Discovery): Mỗi khi 1 node nguồn muốn gửi thông tin tới 1 điểm đến, nhưng không biết tuyến đường tới nó, nó khởi tạo 1 tiến trình khám phá đường đi. Node nguồn tạo 1 gói AODV RREQ (Route Request) và broadcast tới các láng giềng.



Sequence numbers

+ source_sequence_# chỉ ra “độ tươi” (“freshness”) của tuyến đảo ngược tới nguồn

+ dest_sequence_# chỉ ra “độ tươi” của tuyến đường tới đích

Mỗi láng giềng nhận RREQ và tất cả:

+ Trả về 1 gói tin trả lời tuyến đường (RREP), hoặc

+ Chuyển tiếp RREQ tới các láng giềng của nó

(source_addr, broadcast_id) xác định duy nhất gói tin RREQ

+ broadcast_id được tăng mỗi khi 1 gói tin RREQ được gửi

+ Các node nhận có thể xác định và loại bỏ các gói RREQ trùng lặp

Nếu 1 node không thể trả lời RREQ:

+ Tăng hopcnt

+ Lưu trữ thông tin để thực hiện thiết lập đường ngược lại (AODV giả định các liên kết đối xứng):

Láng giềng đã gửi gói tin RREQ

Địa chỉ IP của điểm đến

Địa chỉ IP nguồn

Broadcast ID

Sequence number của node nguồn

Thời gian hết hạn (để kích hoạt thu gom rác thải)

*Route Reply

Nếu 1 node nhận 1 gói tin RREQ và nó có 1 tuyến hiện tại tới đích đến, nó sẽ unicast 1 gói tin RREP tới láng giềng đã gửi gói RREQ.

Các nodes trung gian truyền bá RREP đầu tiên cho nguồn về phía nguồn sử dụng các mục tuyến đảo ngược trong cache (cached reverse route entries)

Các gói tin RREP khác bị hủy trừ khi...

+ dest_sequence_# lớn hơn trước đó, hoặc

+ dest_sequence_# vẫn vậy, nhưng hopcnt nhỏ hơn (nghĩa là có 1 đường tốt hơn)

RREP dần dần tiến về nguồn, có thể sử dụng láng giềng gửi RREP như là hop tiếp theo của nó để gửi về đích.

Các tuyến đảo ngược trong cache sẽ hết hạn trong các nodes không thấy gói tin RREP.

*Route Maintenance

Các thay đổi tuyến đường có thể được phát hiện:

+ Thất bại của các gói tin HELLO định kỳ

+ Thất bại hoặc dấu hiệu ngắt kết nối từ tầng liên kết

+ Thất bại trong việc truyền trải 1 gói tin tới hop tiếp theo (có thể phát hiện bằng cách lắng nghe việc truyền lại nếu nó không phải đích đến cuối cùng)

Node thượng nguồn (về phía nguồn) phát hiện 1 thất bại truyền bá 1 gói tin RERR (route error) với 1 sequence number mới của đích và 1 hop count vô hạn (không thể truy cập)

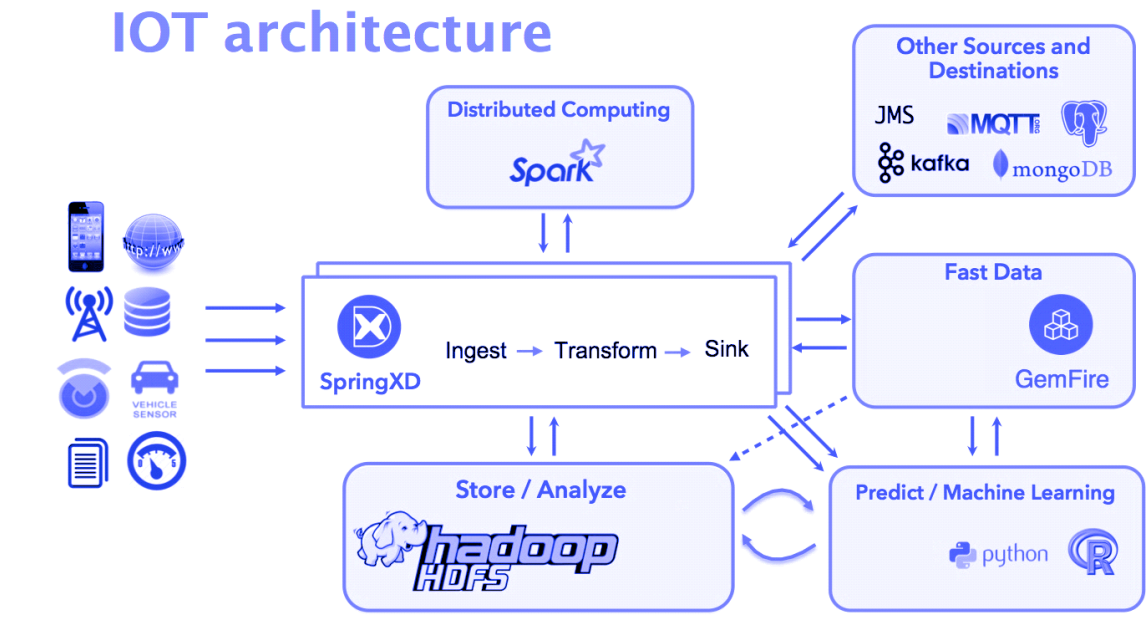
Node nguồn (hoặc 1 node khác trên đường đi) có thể xây dựng lại 1 đường bằng cách gửi 1 gói tin RREQ.

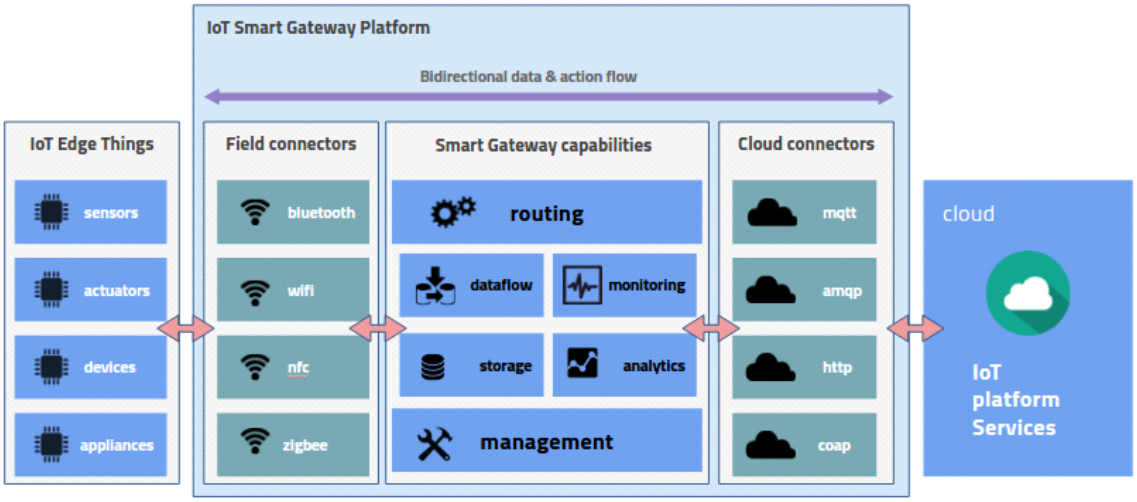
*** So sánh hiệu suất AODV vs OLSR**

Giao thức AODV thực hiện tốt hơn trong các môi trường mạng có giao thông tĩnh với số lượng các cặp nguồn và đích là tương đối nhỏ với mỗi host. Nó sử dụng các tài nguyên ít hơn OLSR, bởi vì kích thước các thông điệp điều khiển được giữ tối thiểu yêu cầu ít băng thông hơn cho việc duy trì các tuyến và bảng định tuyến được giữ tối thiểu giảm tiêu tốn năng lượng tính toán. Giao thức AODV có thể được sử dụng trong các môi trường thiếu hụt tài nguyên nghiêm trọng.

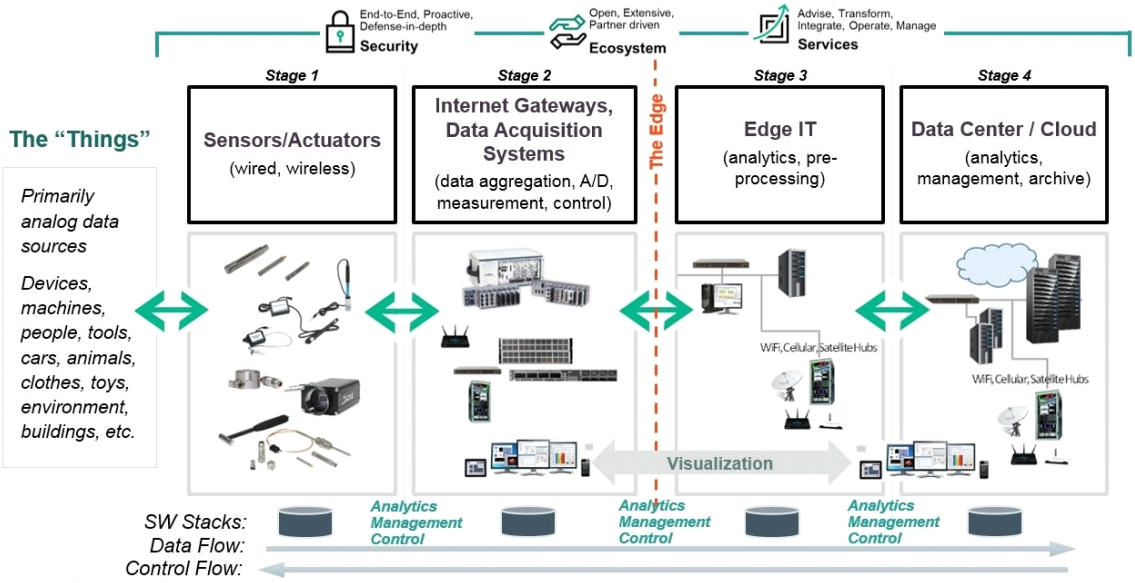
Giao thức OLSR thì hiệu quả hơn trong các mạng với giao thông mật độ cao và rời rạc. Nhưng trường hợp tốt nhất là khi ở giữa 1 số lượng lớn các hosts. Các số liệu chất lượng dễ dàng mở rộng cho giao thức hiện tại. OLSR yêu cầu được cung cấp băng thông liên tục để nhận các thông điệp cập nhật cấu trúc liên kết.

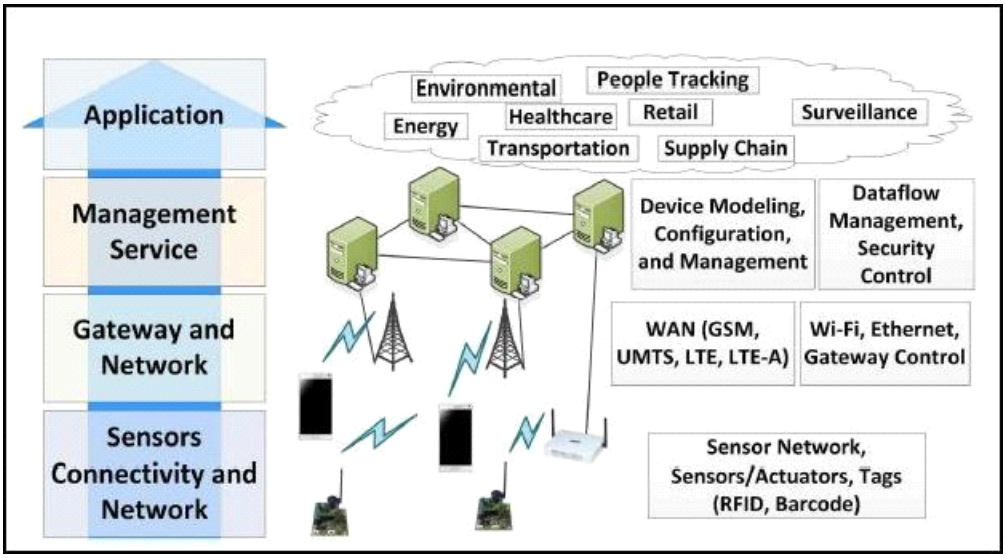
6. Mô hình, công nghệ, giải pháp IOT



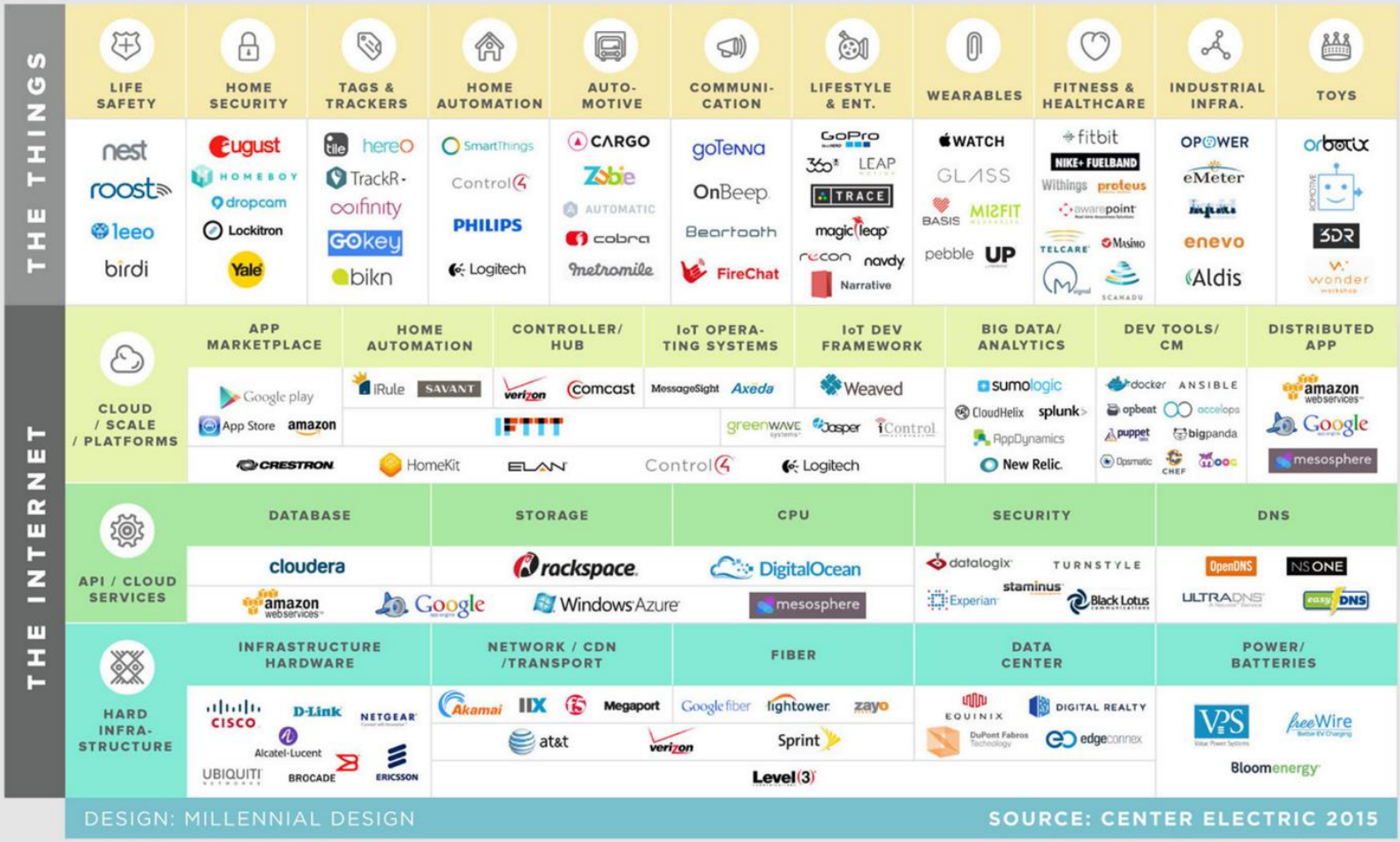


The 4 Stage IoT Solutions Architecture





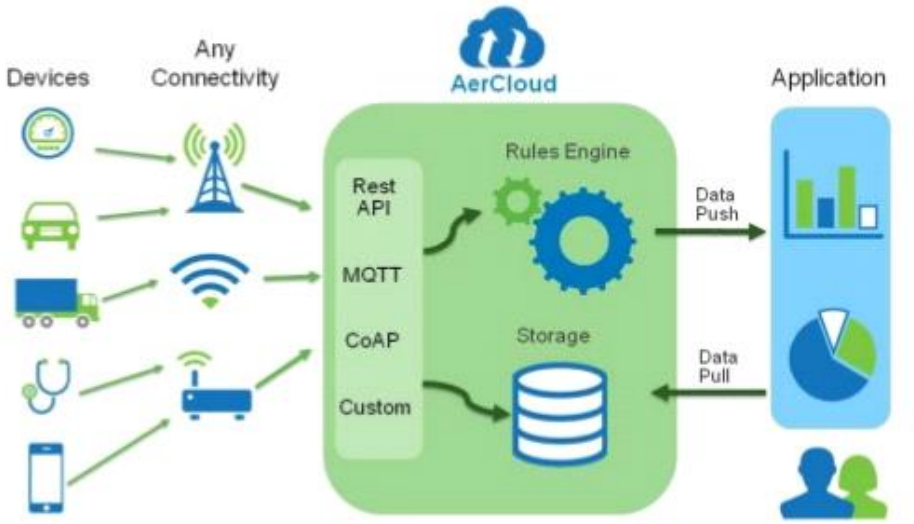
INTERNET OF THINGS TECTONICS



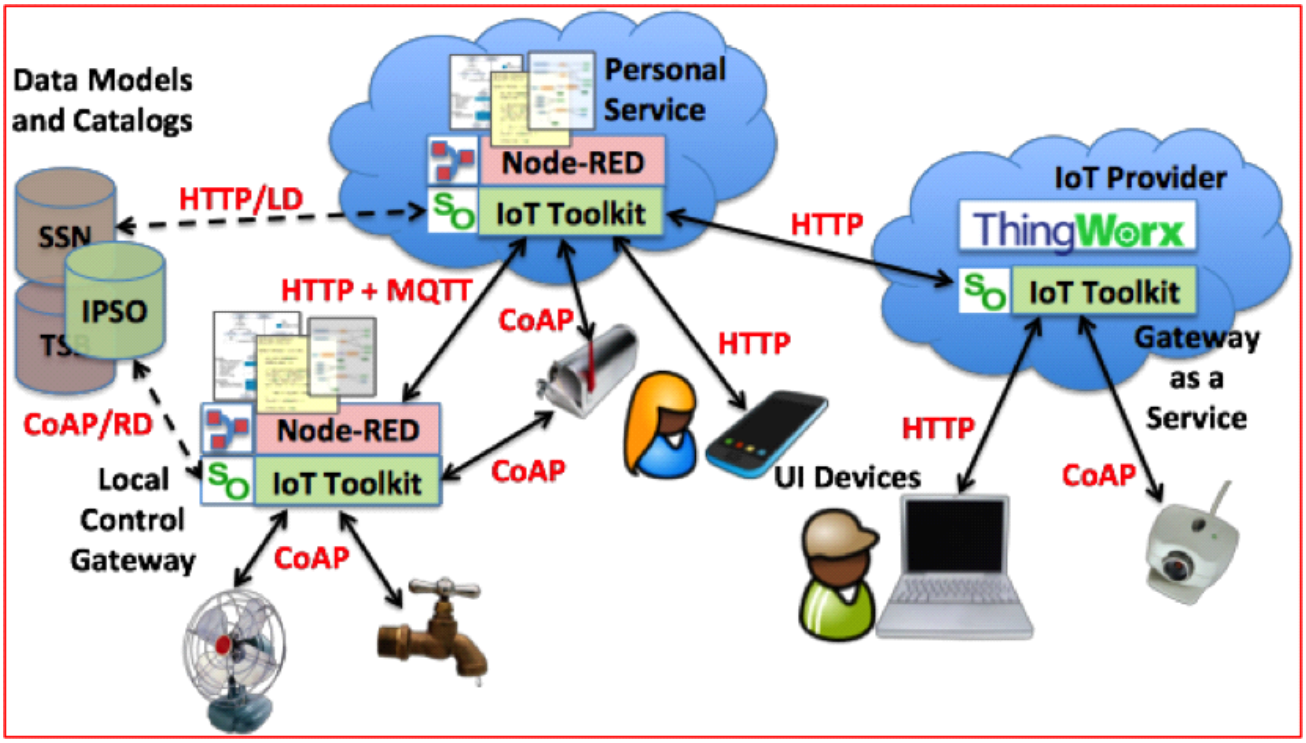
DESIGN: MILLENNIAL DESIGN

SOURCE: CENTER ELECTRIC 2015

Cloud Architecture



Source: Aeris



Spring XD: là 1 dịch vụ thống nhất, phân tán và mở rộng cho việc nhập xuất dữ liệu, phân tích thời gian thực, xử lý hàng loạt. Mục đích nhằm giải quyết sự phức tạp của big data. Phần lớn sự phức tạp trong xây dựng các ứng dụng big data đều liên quan tới việc tích hợp nhiều hệ thống khác nhau thành 1 giải pháp hợp nhất giữa một loạt các trường hợp sử dụng. Các trường hợp phổ biến gặp phải trong việc tạo ra 1 giải pháp toàn diện là:

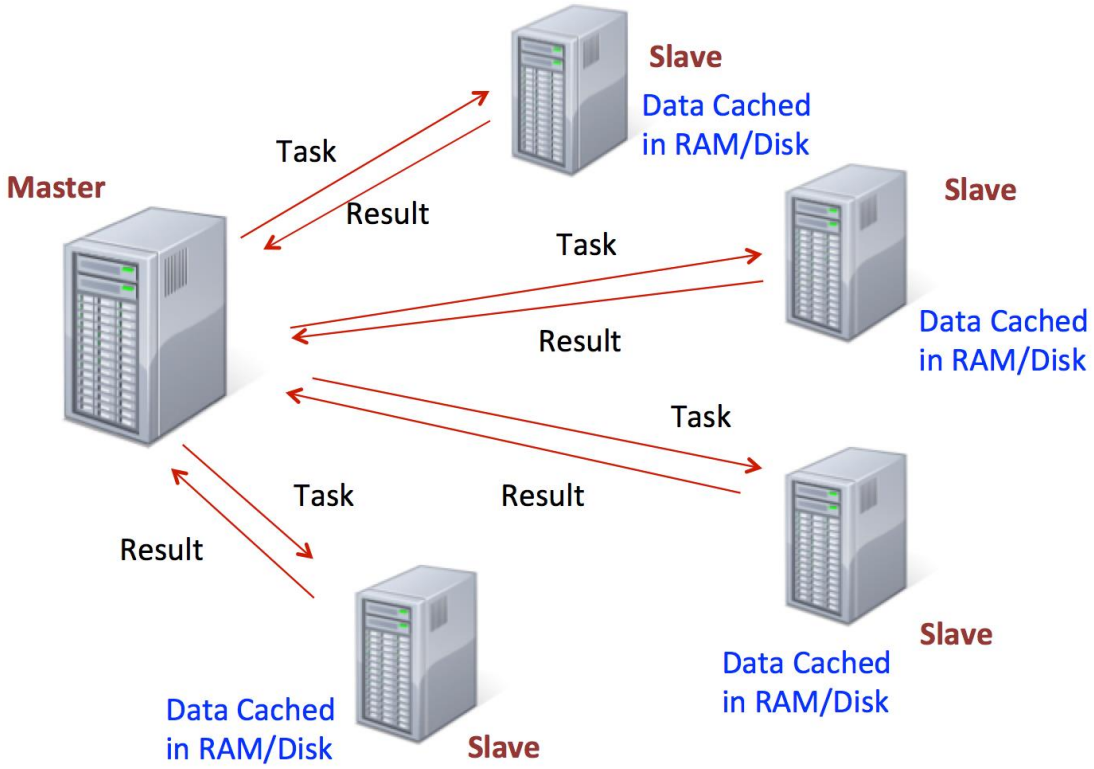
- Nhập dữ liệu phân bố thông lượng cao từ nhiều nguồn dữ liệu vào kho dữ liệu lớn như HDFS hoặc Splunk
- Phân tích thời gian thực tại thời điểm nhập, ví dụ thu thập dữ liệu và đếm các giá trị
- Quản lý công việc thông qua các công việc hàng loạt. Các công việc kết hợp các tương tác với các hệ thống chuẩn (ví dụ RDBMS) cũng như các hoạt động của Hadoop (MapReduce, HDFS, Pig, Hive, Hbase)
- Xuất dữ liệu thông lượng cao, ví dụ từ HDFS đến RDBMS hoặc cơ sở dữ liệu NoSQL.

Spark: là 1 hệ thống các cluster cùng hoạt động song song, được quản lý và giao nhiệm vụ bởi 1 master. 1 nhiệm vụ sẽ được phân chia nhỏ ra và được giao cho mỗi cluster xử lý 1 phần sau đó được tổng hợp lại. Spark có các ưu điểm:

- Resilient: có thể tái cấu trúc lại trong trường hợp gặp lỗi
- Distributed: các phép biến đổi là các hoạt động song song
- Dataset: dữ liệu được nạp và phân vùng trên các node cluster.

Cách thực hiện 1 công việc của Spark:

How does Spark execute a job



- Master kiểm soát cách dữ liệu được phân vùng, và nó tận dụng lợi thế local trong khi theo dõi tất cả các phép tính toán phân tán dữ liệu trên các máy slave. Nếu 1 máy slave nào đó không có sẵn, dữ liệu trên máy đó được tái tạo lại trên các máy khác. “Master” hiện là 1 điểm lỗi duy nhất.

HDFS (Hadoop Distributed File System): là 1 hệ thống file phân tán được thiết kế để chạy trên phần cứng thương mại. HDFS có các đặc điểm:

- Chạy trên các cluster có hàng trăm thậm chí hàng ngàn node. Các phần cứng này được xây dựng nên từ các phần cứng thông thường, giá rẻ, tỷ lệ lỗi cao. Do chất lượng phần cứng như vậy sẽ dẫn đến tỷ lệ lỗi cao trên cluster. Vì thế khả năng phát hiện lỗi, chống chịu lỗi, tự phục hồi phải được tích hợp vào trong hệ thống HDFS.
- Kích thước file sẽ lớn hơn nhiều so với chuẩn truyền thống, các file có kích thước hàng GB sẽ trở nên phổ biến. Khi làm việc với dữ liệu có kích cỡ nhiều TB, thì ít khi nào người ta lại chọn việc quản lý hàng tỷ file có kích thước KB. Việc chia dữ liệu thành một ít file có kích cỡ lớn sẽ tối ưu hơn, do việc này giúp giảm thời gian truy xuất dữ liệu và đơn giản hóa việc quản lý tập tin

- HDFS không phải là một hệ thống file dành cho các mục đích chung. HDFS được thiết kế dành cho các ứng dụng dạng xử lý khối (batch processing). Do đó, các file trên HDFS một khi được tạo ra, ghi dữ liệu và đóng lại thì không thể bị chỉnh sữa được nữa. Điều này làm đơn giản hoá đảm bảo tính nhất quán của dữ liệu và cho phép truy cập dữ liệu với thông lượng cao

GemFire: là 1 nền tảng quản lý dữ liệu phân tán được thiết kế cho nhiều tình huống quản lý dữ liệu đa dạng, nhưng đặc biệt hữu ích cho các hệ thống giao dịch có dung lượng lớn, độ nhạy cao, nhiệm vụ quan trọng.

JMS (Java Message Service): là 1 API viết bằng Java cho phép các ứng dụng tạo, gửi, nhận, và đọc tin nhắn sử dụng giao tiếp tin cậy, bất đồng bộ, lỏng lẻo

Kafka: là 1 nền tảng streaming phân tán. Kafka được sử dụng mới mục đích:

- Xây dựng đường dữ liệu streaming thời gian thực nhận dữ liệu đáng tin cậy giữa các hệ thống hoặc ứng dụng
- Xây dựng các ứng dụng streaming thời gian thực được chuyển đổi hoặc phản ứng với luồng của dữ liệu

7. Zigbee vs WiFi

	Zigbee	WIFI
1. IEEE Standard	IEEE 802.15.4	IEEE 802.11.x (x là a,b,g,.v..v)
2. Development Timeline	- Ý tưởng từ 1999. - Được đưa ra vào năm 2004	- Ý tưởng từ 1985. - Cộng đồng chuẩn hóa thành lập 1990, đưa ra bộ tiêu chuẩn năm 1997.
3. Operating Frequency	works at 900-928 MHz và 2.4GHz. Ở Châu Âu, Zigbee hoạt động ở tần số 868MHz	work at 2.4GHz, 5GHz (gần đây đã phát triển wifi cho phép hoạt động ở 60GHz)
4. Channel Bandwidth	1MHz	0.3, 0.6 or 2MHz
5. Network Range	- WPAN (Wireless Personal Area Networks). - Trong các ứng dụng thông thường đạt từ 10-30m; có 1 số ứng dụng có thể đạt đến 100m.	WAP và WLAN, phạm vi trung bình từ 30-100m.
6. Data transfer speed	Maximum = 250kps; khá thấp so với tốc độc thấp nhất của wifi.	- Nhanh hơn Zigbee về data transfer. - Tốc độ của wifi theo mỗi tiêu chuẩn: 802.11b: maximum = 11mbps. 802.11a & 802.11c: maximum = 54mbps
7. Bit Time (thời gian truyền 1 bit/ 1 data rate of transfer cho trước)	4micro seconds	0.00185 micro seconds
8. Power Consumption	Được thiết kế “assemble and forget” -> tiêu thụ năng lượng ít. Zigbee tiêu thụ năng lượng bằng ¼ so với wifi.	Chưa thực sự tốt trong việc tiêu thụ ít năng lượng, để hoạt động hơn 10hrs cần có pin dự phòng.
9. Network Elements	Phân thành 3 loại: Zigbee coordinator, Zigbee end router, Zigbee end device.	- Point-to-point network. - Wifi router được sử dụng khi cần kết nối nhiều devices, hoặc cần connect đến internet.

10. Network Size (in one network)	Over 65,000 nodes	Up to 2007 nodes
11. Network Security	- Advanced Encryption Security (AES) methods for encryption. - CCB-CCM methods for network security	WEP, WPA and WPA2 protocols for network encryption and security, respectively.
12. Applications	Thường được dùng exchange data, Zigbee phổ biến trong kết nối không dây giữa các wireless sensor, vd: hệ thống automatic trong nhà hoặc hệ thống điều phối trong công nghiệp	Wi-Fi thường là lựa chọn tốt cho kết nối internet. Dùng để thực hiện data exchange giữa computer và modem, streaming music/videos.

8. Zigbee vs 6LowPan

Zigbee	6loWPAN
Các thiết bị Zigbee không thể giao tiếp trực tiếp với các thiết bị khác trên internet	Các thiết bị 6loWPAN có thể giao tiếp trực tiếp ra internet
Mạng Zigbee được quản lí bởi một coordinator thực hiện giao thức tầng app và gửi dữ liệu về server	Các servers có thể thu thập dữ liệu trực tiếp từ các thiết bị mà không cần chờ coordinators xử lí các request
Coordinators mà hỏng thì mạng zigbee sẽ không còn khả năng kết nối ra internet	Không cần coordinators để kết nối ra internet
Truyền dữ liệu chậm	Truyền dữ liệu nhanh
Tầng network sử dụng các địa chỉ 64-bit IEEE 802.15.4	Các nodes IPv6 được gán các địa chỉ IP 128 bits

9. IOT architecture

- Cluster-based:

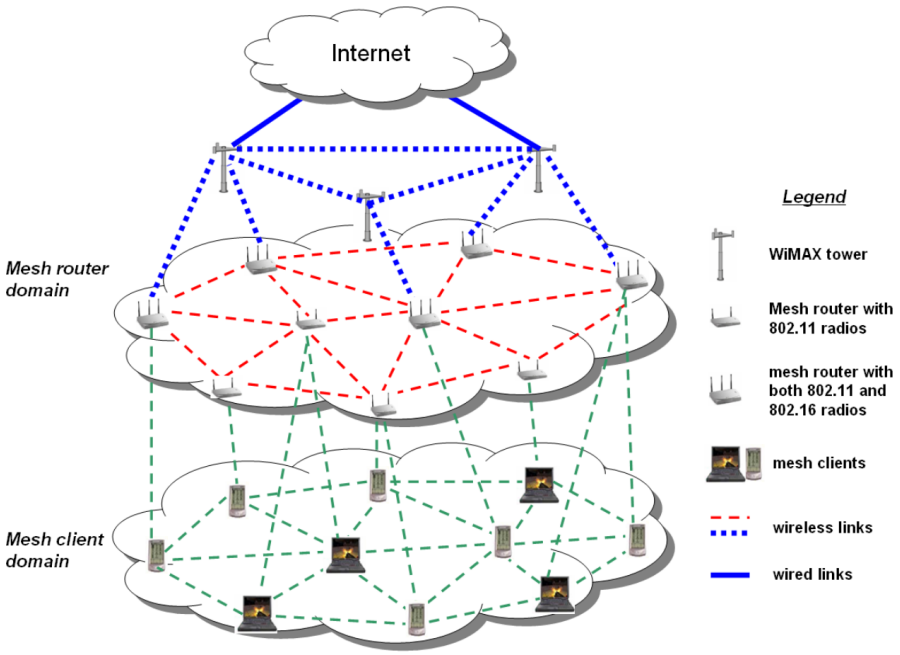
Tập hợp các sensor nodes vào trong các clusters để tối ưu khoảng cách truyền để tiết kiệm năng lượng và xoay các clusterheads để phân phối năng lượng tiêu thụ 1 cách công bằng.

+ Low-Energy Adaptive Clustering Hierachy (LEACH):

LEACH chia mạng ra thành các cluster và chỉ 1 node (CH) trong mỗi cluster là lãnh đạo mà nó thay đổi mỗi vòng. CH giao tiếp trực tiếp với BS để gửi dữ liệu và sử dụng kỹ thuật tập hợp dữ liệu nhằm giảm tiêu thụ năng lượng và kéo dài tuổi thọ của mạng cảm biến không dây.

+ Hybrid Energy-Efficient Distributed Clustering (HEED):

Xác suất chọn CH có tính đến 3 yếu tố là năng lượng dư thừa, chi phí truyền thông và năng lượng truy cập tối thiểu trung bình (Average Minumum Reachability Power – AMRP). Nó sử dụng phương thức truyền thông như giao thức LEACH nhưng HEED có năng lượng cân bằng hơn và tuổi thọ mạng lâu hơn LEACH



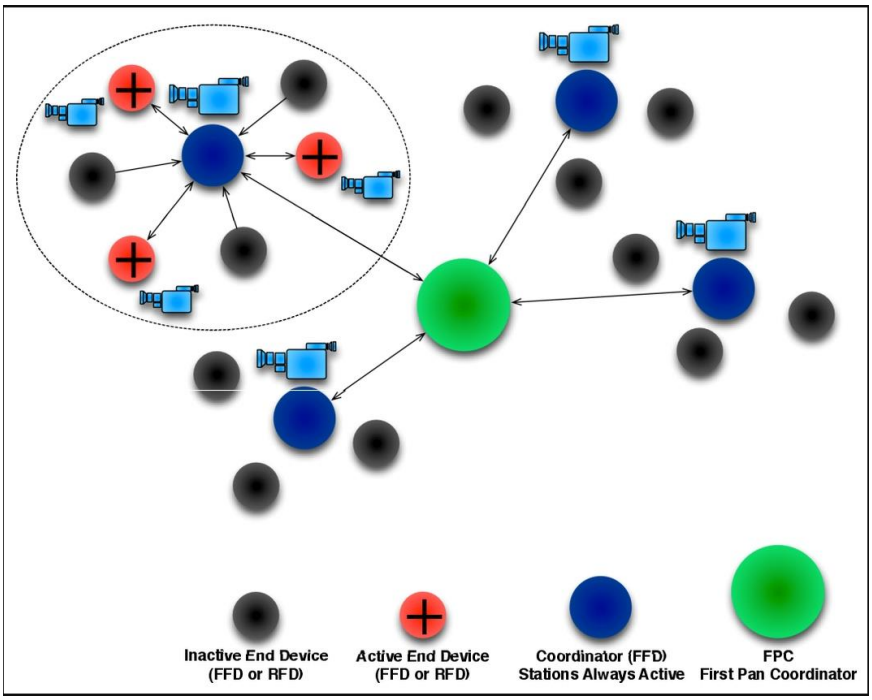
- Tree-based:

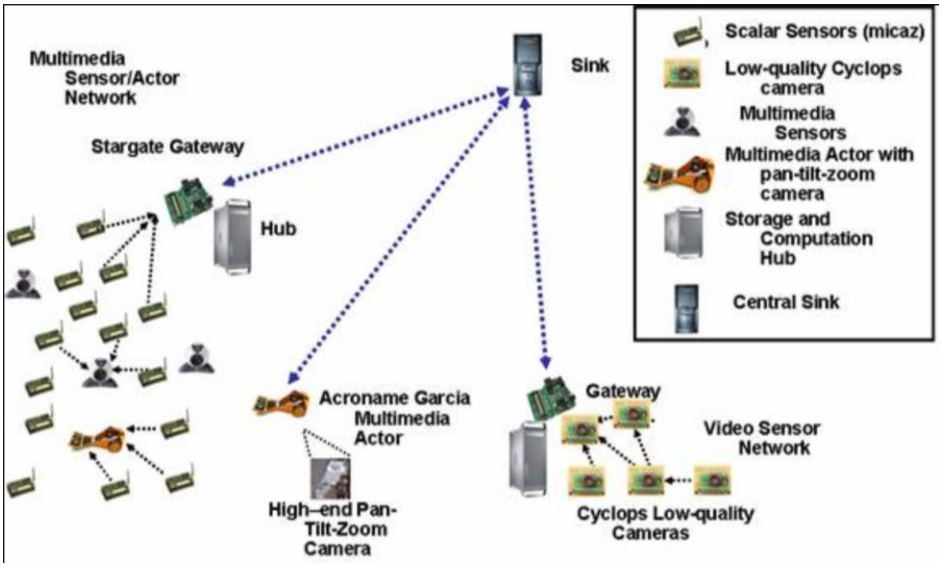
Trong các mạng khi truyền thông P2P được dựa trên định tuyến cây, các nodes được sắp xếp vào 1 hoặc nhiều cây, nơi mà chúng để chứa ID của parent. Gốc của 1 cây được đại diện bởi 1 node quyền lực hơn chứa thông tin kết nối của cả mạng. Đầu tiên, gói tin được định tuyến tới gốc của cây, nơi mà router trung tâm tính toán đường đi ngắn nhất để đến đích. Sau đó, gói tin được gửi đến đích thông qua đường đi vừa được tính.

Lợi ích: yêu cầu bộ nhớ thấp trên các nodes và đơn giản hóa thuật toán định tuyến.

Bất lợi: có thể yêu cầu định tuyến kéo dài, tỉ lệ giữa độ dài của đường đi được tìm thấy và đường đi tối ưu, và yêu cầu router trung tâm phải biết toàn bộ mô hình mạng. Thêm vào đó, các nodes trên cùng (top-level) có thể bị quá tải bởi lưu lượng mạng, đặc biệt trong các mạng lớn.

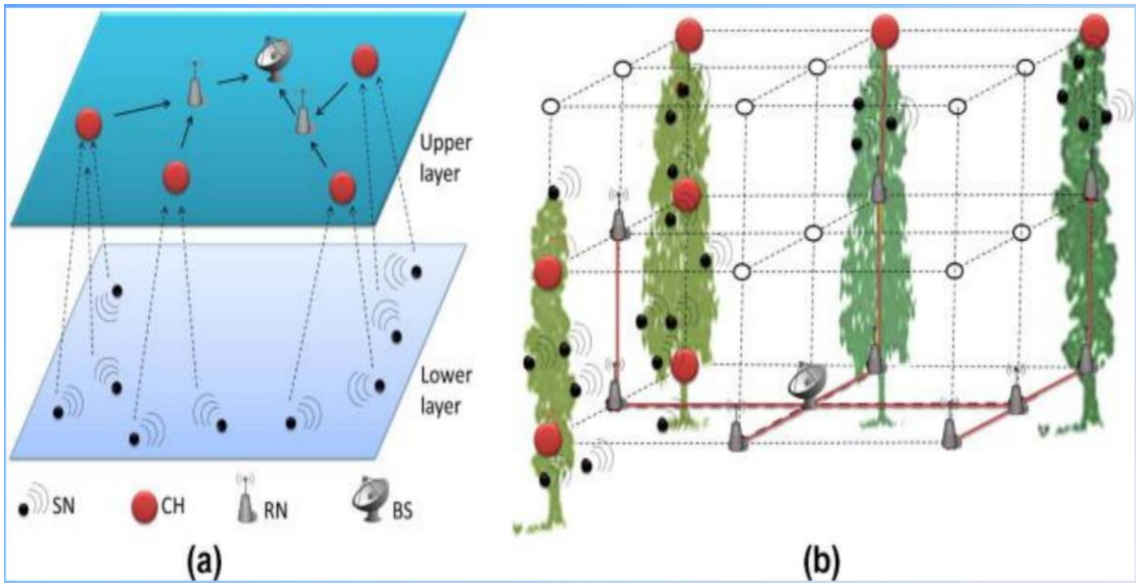
+ EADAT





- Grid-based:

+ Two-Tier Data Dissemination (TTDD)



- Chain-based:

+ Power-Efficient Gathering in Sensor Information Systems (PEGASIS)