

IOT Technologies and Protocols

Protokol Komunikasi IoT



Bengkel Internet of Things

D3 Teknologi Telekomunikasi

Fakultas Ilmu Terapan

Outline

- Mengenal IoT
- Definisi Protokol
- Network Protokol
- MQTT
- Block Diagram MQTT
- MQTT QoS
- Komunikasi Protokol

Pendahuluan

- Internet of Things atau IoT adalah perangkat/alat yang menjadi pintar karena terhubung ke jaringan komputer.
- Terhubungnya perangkat tersebut memungkinkan dilakukannya interaksi antara IoT dengan sistem komputer yang bisa saja dikendalikan oleh manusia. Misalnya, melalui perintah suara, pemilik rumah dapat mematikan atau menghidupkan lampu bahkan menjadwalkan kapan seharusnya lampu aktif.
- Karena banyaknya IoT yang dikembangkan untuk terhubung ke sistem komputer, dikembangkan protokol komunikasi yang bersifat agnostik sehingga baik sistem komputer dan IoT tidak harus mengembangkan protokol komunikasi sendiri. Protokol komunikasi yang bersifat agnostik pada IoT banyak memiliki implementasi, beberapa yang populer adalah menggunakan API seperti REST API dan MQTT API.

Mengenal IoT

- Internet of Things (IoT) adalah konsep di mana koneksi internet diperluas ke perangkat fisik yang digunakan dalam kehidupan sehari-hari. Perangkat tersebut dapat saling bertukar informasi dengan perangkat yang lainnya.
- Contoh IoT seperti kulkas atau mesin cuci di mana dalam perangkat tersebut sudah tertanam sensor elektronik yang dapat berkomunikasi dan berinteraksi dengan orang lain melalui jaringan internet. Manusia dapat berinteraksi dengan perangkat tersebut melalui gadget dari jarak jauh.
- Saat ini teknologi IoT lebih dikenal dengan produk yang berhubungan dengan konsep “rumah pintar”, seperti sistem keamanan rumah dengan menggunakan kamera. Padahal pada saat ini masih banyak teknologi IoT di berbagai bidang, seperti bidang kesehatan, pertanian dan industri.

Definisi Protokol

- Protokol merupakan peraturan atau prosedur untuk mengirimkan sebuah data pada perangkat elektronik.
- Pada sistem komputer, setiap komputer berkomunikasi dengan komputer lainnya menggunakan sebuah protokol.
- Banyak sekali jenis protokol, seperti jenis protokol yang tidak terlalu asing ditelinga seperti UDP, FTP, TCP, ICMP dan yang sering kita pakai untuk komunikasi IoT pada artikel sebelumnya yaitu HTTP (Hyper Text Transfer Protocol).

Alasan

- Ada beberapa pertimbangan mengapa transfer data di sistem IoT harus seefisien mungkin. Seperti diketahui bahwa pada sistem IoT terutama pada bagian perangkat akuisisi data seperti sensor & embedded device biasanya memakai catu daya yang disediakan oleh baterai, solar panel dll.
- Koneksi perangkat ini dengan server atau internet biasanya memakai koneksi WiFi, Bluetooth, GSM, gelombang radio dll yang memerlukan catu daya cukup besar saat koneksi terjadi apalagi kalau koneksi terjadi tiap waktu, batasan lainnya yaitu ukuran media penyimpanan internal dan RAM dari perangkat-perangkat ini biasanya relatif cukup kecil

Optimasi Transfer Data

Jadi ada beberapa optimasi yang harus diperhatikan dalam pertukaran data pada sistem IoT agar transfer data bisa seefisien mungkin

1. Menekan ukuran paket data sekecil mungkin sehingga trafik bisa meningkat.
2. Meminimalisasi proses komputasi untuk encoding dan decoding dari paket data.
3. Data hanya menggunakan ruang penyimpanan yang sekecil mungkin.

Network Protocol

REST API dan MQTT

IOT Protocols

**Constrained Application Protocol
(CoAP)**

**Message Queue Telemetry
Transport Protocol (MQTT)**

**Advanced Message Queuing
Protocol (AMQP)**

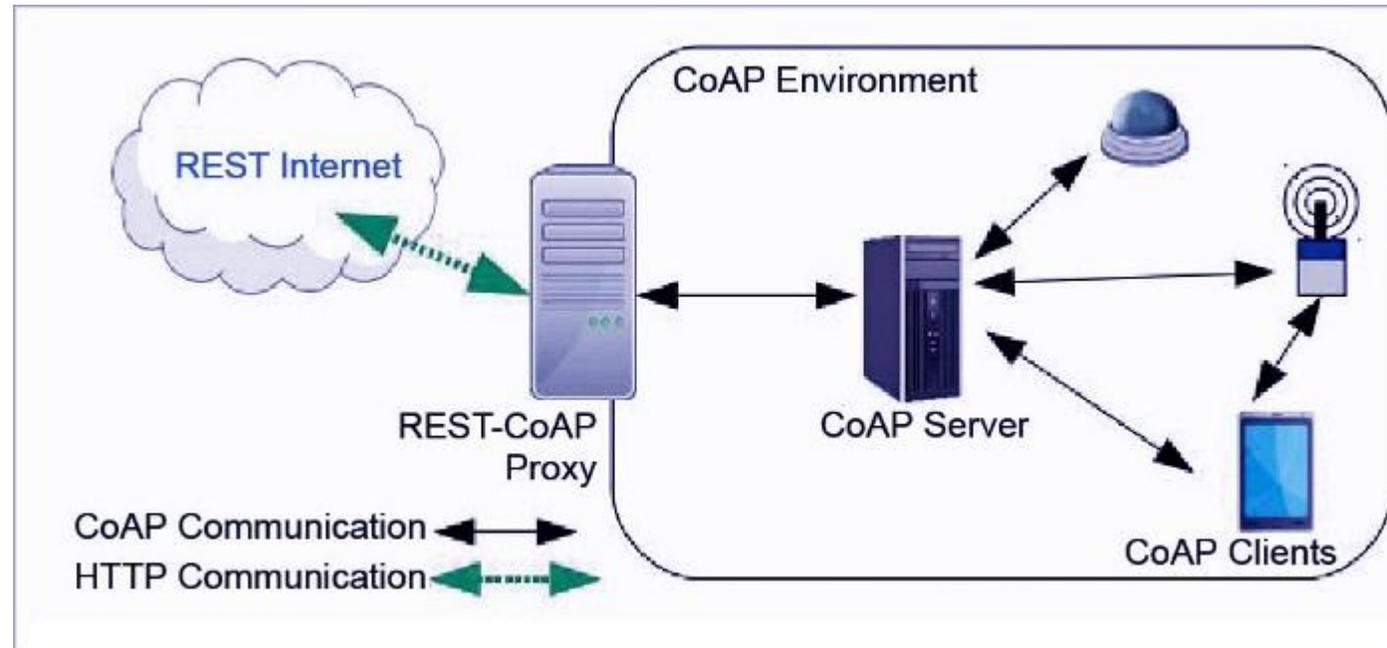
**Data Distribution Service
(DDS)**



IoT Protocols

- Constrained Application Protocol (CoAP)
- Message Queue Telemetry Transport Protocol (MQTT)
- Advanced Message Queuing Protocol (AMQP)
- Data Distribution Service (DDS)

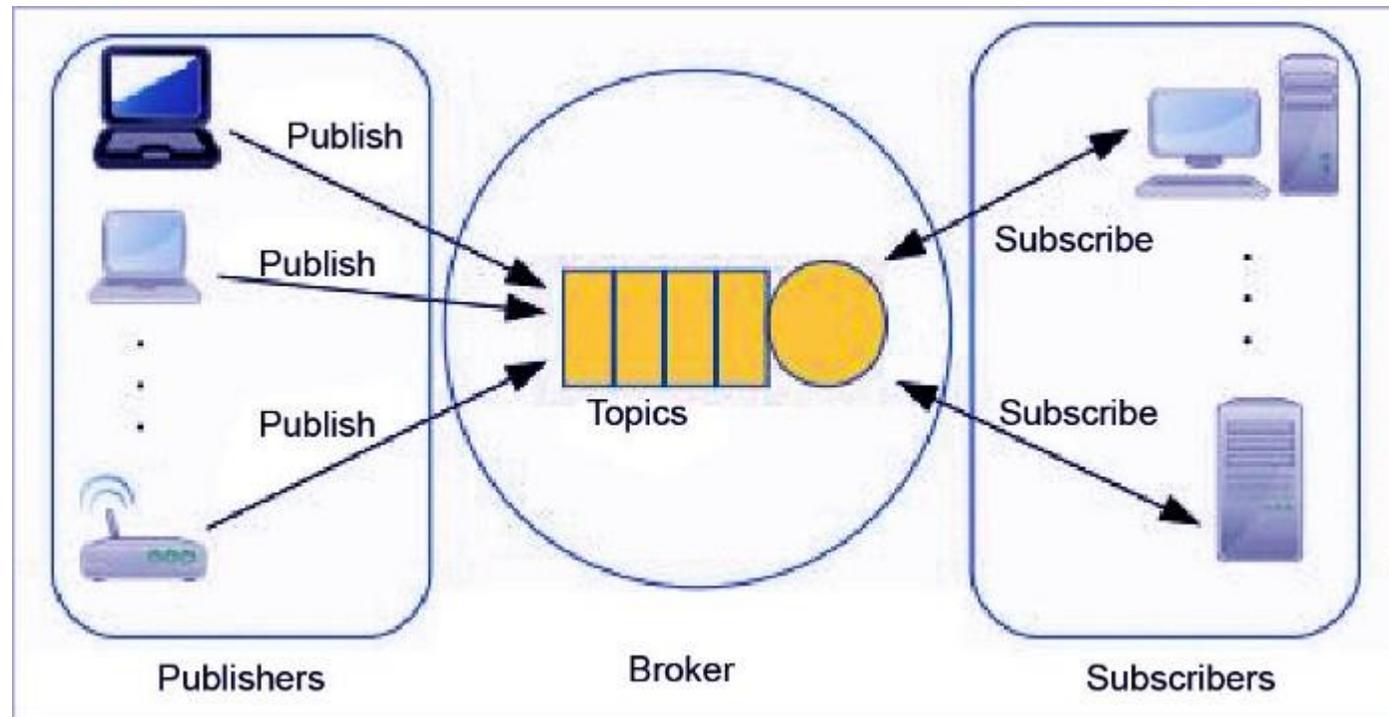
Constrained Application Protocol (CoAP)



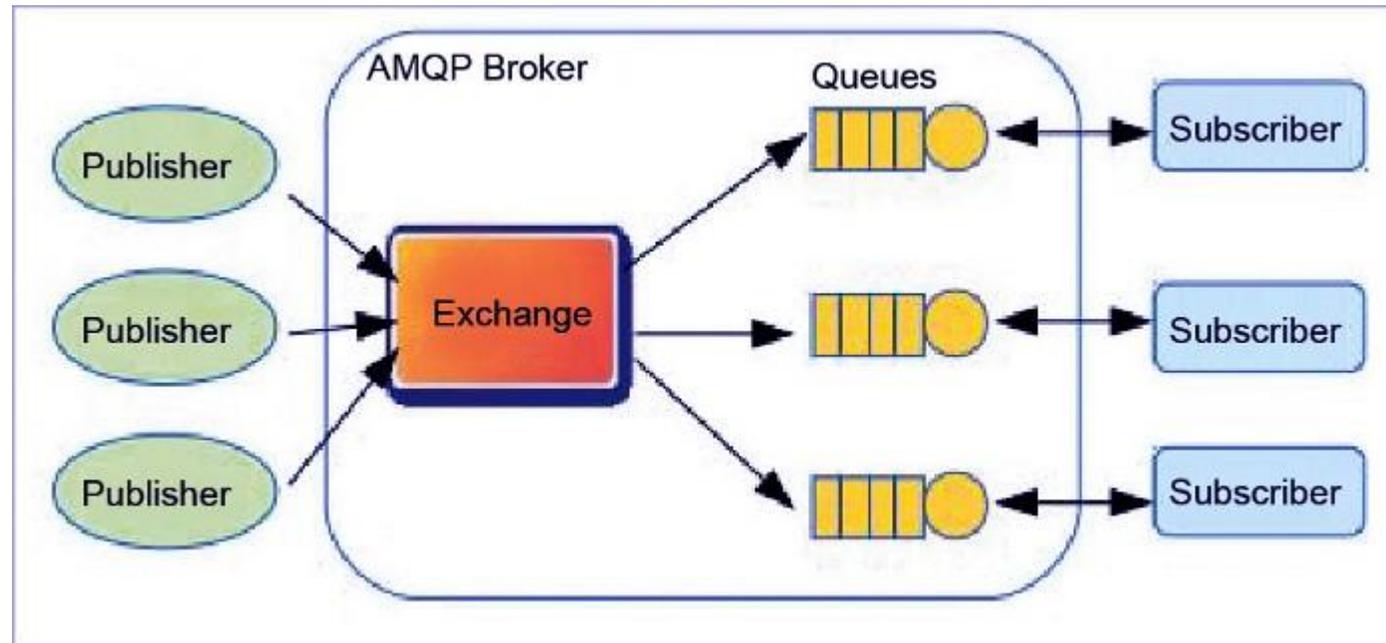
CoAP Protocol

- CoAP-Constrained Application Protocol is a specialized Internet Application Protocol for constrained devices, as defined in RFC 7252. It enables devices to communicate over the Internet. It is defined as Constrained Application Protocol, and is a protocol intended to be used in very simple hardware. The protocol is especially targeted for constrained hardware such as 8-bits microcontrollers, low power sensors and similar devices that can't run on HTTP or TLS. It is a simplification of the HTTP protocol running on UDP, that helps save bandwidth. It is designed for use between devices on the same constrained network (e.g., low-power, lossy networks), between devices and general nodes on the Internet, and between devices on different constrained networks both joined by an internet. CoAP is also being used via other mechanisms, such as SMS on mobile communication networks.

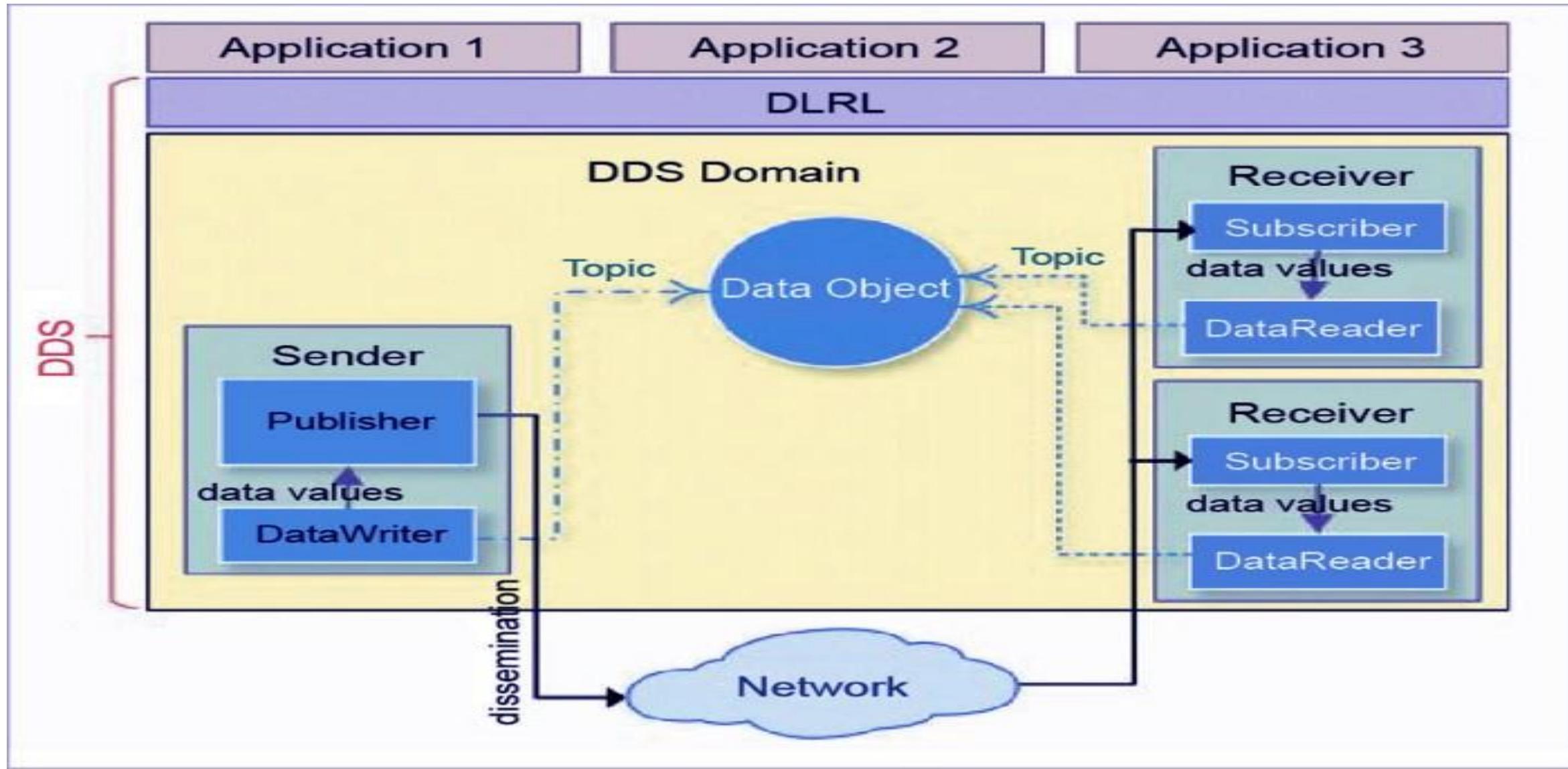
Message Queue Telemetry Transport Protocol (MQTT)



Advanced Message Queuing Protocol (AMQP)



Data Distribution Service (DDS)



REST API

- REST API adalah protokol komunikasi yang menggunakan endpoint URI dan memanfaatkan ketersediaan METHOD pada Protokol HTTP, yaitu GET untuk meminta data, POST untuk membuat data baru, PUT untuk mengubah data, dan DELETE untuk menghapus data.
- REST API memberikan layanan lengkap terhadap pengolahan data pada IoT dengan tetap bersifat agnostik.

MQTT API

- MQTT API adalah protokol komunikasi ringan yang melayani PUBLISH dan SUBSCRIBE dan lebih ringan dari HTTP. Tidak selengkap REST API, tetapi cukup memenuhi kebutuhan sebagian besar IoT.
- MQTT (Message Queuing Telemetry Transport) protokol merupakan sebuah protokol yang berjalan diatas stack TCP/IP dan dirancang khusus untuk machine to machine yang tidak memiliki alamat khusus.
- Maksud dari kata tidak memiliki alamat khusus ini seperti halnya sebuah arduino, raspi atau device lain yang tidak memiliki alamat khusus.
- Sistem kerja MQTT menerapkan Publish dan Subscribe data. Dan pada penerapannya, device akan terhubung pada sebuah Broker dan mempunyai suatu Topic tertentu.

Sinyal Kontrol

1. CONNECT — Client request to connect to Server
2. CONNACK — Connection Acknowledgement
3. PUBLISH — A message which represents a new/separate publish
4. PUBACK — QoS 1 Response to a PUBLISH message
5. PUBREC — First part of QoS 2 message flow
6. PUBREL — Second part of QoS 2 message flow
7. PUBCOMP — Last part of the QoS 2 message flow
8. SUBSCRIBE — A message used by clients to subscribe to specific topics
9. SUBACK — Acknowledgement of a SUBSCRIBE message
10. UNSUBSCRIBE — A message used by clients to unsubscribe from specific topics
11. UNSUBACK — Acknowledgement of an UNSUBSCRIBE message
12. PINGREQ — Heartbeat message
13. PINGRESP — Heartbeat message acknowledgement
14. DISCONNECT — Graceful disconnect message sent by clients before disconnecting



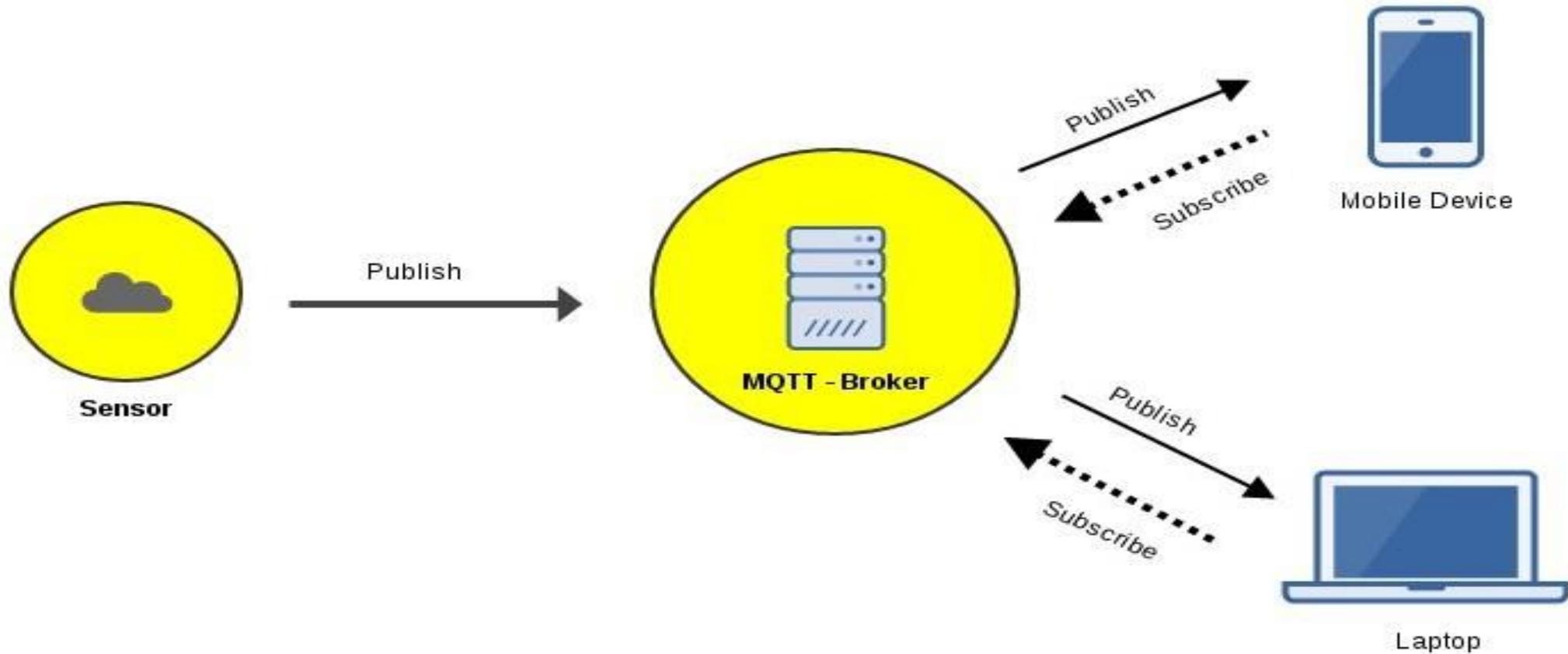
Message Queuing Telemetry Transport

The Message Queuing Telemetry Transport (**MQTT**) is a lightweight, publish-subscribe network protocol that transports messages between devices.

Definisi MQTT

- Protokol MQTT (Message Queuing Telemetry Transport) adalah protokol yang berjalan pada di atas stack TCP/IP dan mempunyai ukuran paket data dengan low overhead yang kecil (minimum 2 bytes) sehingga berefek pada konsumsi catu daya yang juga cukup kecil.
- Protokol ini adalah jenis protokol data-agnostic yang artinya anda bisa mengirimkan data apapun seperti data binary, text bahkan XML ataupun JSON dan protokol ini memakai model publish/subscribe daripada model client-server.

Publish & Subscribe



Seputar MQTT

- MQTT merupakan sebuah protocol komunikasi yang dikhususkan untuk komunikasi “machine to machine”
- MQTT berjalan pada TCP/IP layer.
- MQTT menggunakan model publish/subscribe/
- MQTT hanya membutuhkan daya dan bandwidth yang sedikit untuk mengirim data.
- MQTT hanya dapat mengirim data dalam ukuran yang kecil.
- MQTT dapat mengirim data dalam beberapa format seperti JSON, XML, Text.

Why MQTT?

1. **Lightweight and Efficient** : MQTT clients are very small, require minimal resources so can be used on small microcontrollers. MQTT message headers are small to optimize network bandwidth.
2. **Bi-directional Communications** : MQTT allows for messaging between device to cloud and cloud to device. This makes for easy broadcasting messages to groups of things.
3. **Scale to Millions of Things** : MQTT can scale to connect with millions of IoT devices.
4. **Reliable Message Delivery** : Reliability of message delivery is important for many IoT use cases. This is why MQTT has 3 defined quality of service levels: 0 - at most once, 1- at least once, 2 - exactly once
5. **Support for Unreliable Networks** : Many IoT devices connect over unreliable cellular networks. MQTT's support for persistent sessions reduces the time to reconnect the client with the broker.
6. **Security Enabled** : MQTT makes it easy to encrypt messages using TLS and authenticate clients using modern authentication protocols, such as OAuth.

MQTT: The Standard for IoT Messaging

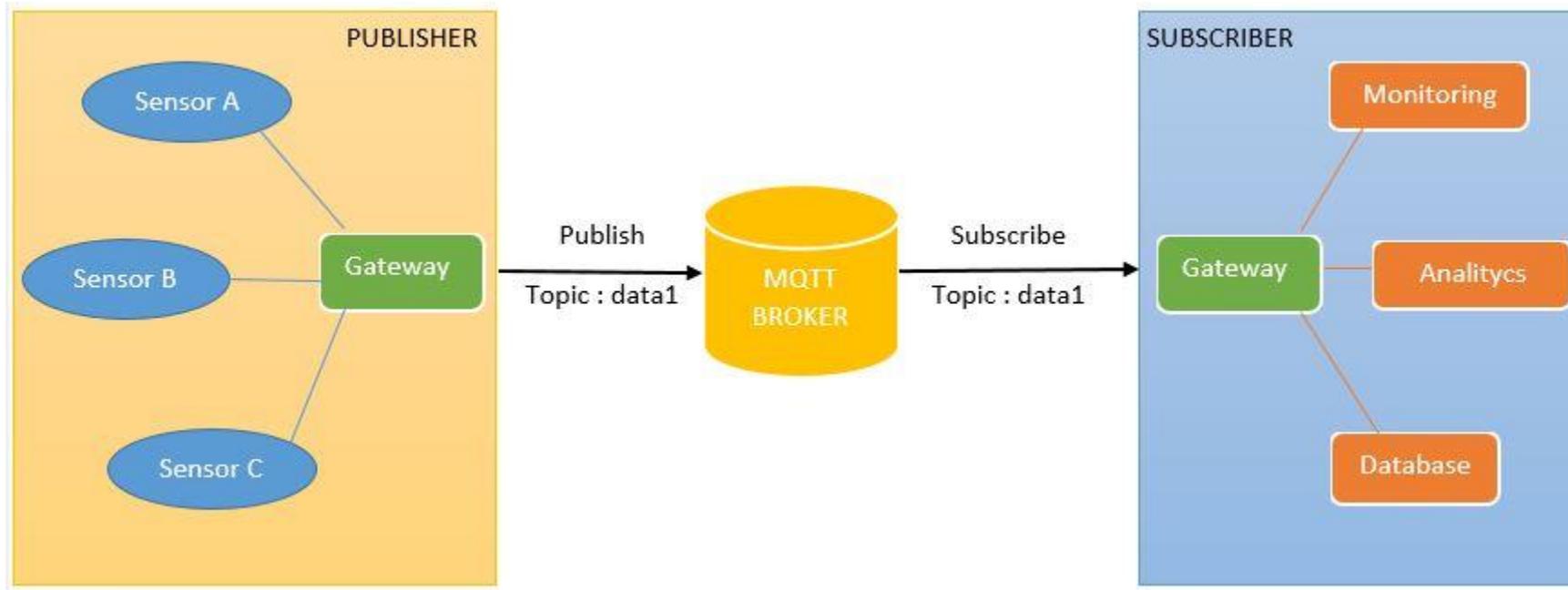
- MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT).
- It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.
- MQTT today is used in a wide variety of industries, such as automotive, manufacturing, telecommunications, oil and gas, etc.

Kebutuhan MQTT

Sistem umum MQTT seperti pada gambar diatas membutuhkan dua komponent perangkat lunak utama yaitu

- MQTT Client yang nantinya akan di install di device. Untuk Arduino bisa memakai pubsubclient, pustaka seperti mqtt.js bisa dipakai pada platform Node.js di Raspberry Pi ataupun laptop.
- MQTT Broker yang berfungsi untuk menangani publish dan subscribe data. Untuk platform Node.js bisa memakai broker mosca sedangkan untuk platform yg lain banyak broker tersedia seperti mosquitto, HiveMQ dll.

Blok Diagram MQTT



Broker, Publish, Subscribe dan Topic

- **Broker** pada MQTT berfungsi untuk menghandle data publish dan subscribe dari berbagai device, bisa diibaratkan sebagai server yang memiliki alamat IP khusus. Beberapa contoh dari Broker yang ada seperti Mosquitto, HiveMQ dan Mosca.
- **Publish** merupakan cara suatu device untuk mengirimkan datanya ke subscribers. Biasanya pada publisher ini adalah sebuah device yang terhubung dengan sensor tertentu.
- **Subscribe** merupakan cara suatu device untuk menerima berbagai macam data dari publisher. Subscriber dapat berupa aplikasi monitoring sensor dan sebagainya, subscriber ini yang nantinya akan meminta data dari publisher.
- **Topic** seperti halnya pengelompokan data disuatu kategori tertentu. Pada sistem kerja MQTT protokol ini, topic bersifat wajib hukumnya. Pada setiap transaksi data antara Publisher dan Subscriber harus memiliki suatu topic tertentu.

MQTT Broker

- MQTT Broker memiliki suatu alamat yang dapat diakses oleh Publisher dan Subscriber. Tugas dari MQTT Broker ini yaitu sebagai penghubung transaksi data antara publisher dan subscriber. MQTT Broker juga mengenali suatu data lewat sebuah pengelompokan atau biasa disebut topic. Ketika Publisher mengirim data sensor A, B, C dengan topic data1, dan suatu saat terdapat Subscriber yang melakukan subscribe dengan topic yang sama data1, maka bisa dipastikan Subscriber akan menerima data sensor A, B dan C dari Publisher.
- Sebelum mengaplikasikan ke real device anda bisa mencoba protokol MQTT dengan memakai layanan broker gratis dari beberapa website seperti test.mosquitto.org atau iot.eclipse.org

MQTT Software : Servers/Brokers

 RabbitMQ by Pivotal

 Swif1MQ

 MQTT Route

 APACHE
ACTIVEMQ™

 mosquitto

 emitter

 ejabberd

 ThingScale

 IBM

 EC

 EMQ

 HIVEMQ

 jmqtt

 flespi

Blok Publisher

- Pada blok Publisher, terdapat beberapa sensor, seperti contoh terdapat sensor A, B dan C. Setiap sensor akan terhubung ke suatu gateway seperti Ethernet shield, ESP8266, SIM800L atau device sejenisnya yang dapat menghubungkan kita ke MQTT Broker. Antara gateway dan sensor pasti terdapat controller, Anda dapat menggunakan controller seperti Arduino, RaspberryPi atau sejenisnya. Nah, tugas dari Blok Publisher ini cukup sederhana yaitu mengirim data yang diambil dari sensor A, B dan C ke suatu MQTT Broker dengan Topic data1.

Blok Subscriber

- Blok Subscriber bertugas untuk melakukan subscribe data pada topic data1. Setelah mendapatkan data yang berupa nilai sensor A, B dan C dari publisher, kita dapat mengolah data-data tersebut untuk dimasukkan kedalam database, dianalisis atau dapat kita proses menjadi sebuah sistem monitoring yang terstruktur dan memiliki nilai jual.

MQTT QoS

- QoS pada MQTT merupakan tingkat kualitas layanan pengiriman pesan pada MQTT
- Terdapat 3 level QoS
 1. QoS level 0 (at most once delivery) : pesan hanya dikirim sekali, setelah pesan dikirim tidak terdapat respon seperti apakah pesan berhasil dikirim.
 2. QoS level 1 (at least once delivery) : pesan paling sedikit dikirim sekali, jika subscriber tidak menerima pesan maka broker akan mengirimkan respon terhadap publisher bahwa pesan gagal dikirim. Terjadi duplikasi pesan karena pengiriman pesan yang berulang-ulang.
 3. QoS level 2 (exactly once delivery) : pesan dikirim sekali dan hanya sekali, memastikan ulang bahwa pesan telah diterima oleh subscriber, mencegah terjadinya duplikasi pesan.

Quality of Service (QoS)

- Satu lagi yang membuat MQTT berbeda adalah adanya level kualitas dari layanan atau QoS. Jadi pesan yang di publish pasti memiliki satu dari 3 level QoS. Level-level ini memberikan garansi akan konsistensi (reliability) dari pengiriman pesan. Klien dan broker menyediakan mekanisme penyimpanan dan pengiriman kembali dari pesan sehingga meningkatkan konsistensi data akibat kegagalan network, restart dari aplikasi dan sebab-sebab lainnya.

Level QoS

- **Level 0**

Pesan dikirim hanya sekali. Pesan yang terkirim tergantung dari reliabiliti stack TCP alias tergantung keberadaan network dan tidak ada usaha untuk mentransmisikan pesan kembali.

- **Level 1**

Pesan dikirimkan setidaknya satu kali. Jadi klien setidaknya akan menerima pesan sekali. Jika subscriber tidak mengakui (acknowledge) maka broker akan mengirimkan pesan sampai publisher menerima status pengakuan pesan dari klien.

- **Level 2**

Pesan pasti diterima satu kali. Protokol dengan level ini memastikan bahwa pesan pasti tersampaikan dan tidak terjadi duplikasi pesan yang terkirim.

Three qualities of service for message delivery:

- "At most once", where messages are delivered according to the best efforts of the operating environment. Message loss can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.
- "At least once", where messages are assured to arrive but duplicates can occur.
- "Exactly once", where messages are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.

Implementasi Protokol Komunikasi API pada IoT

- Adafruit.io dan Thingspeak adalah dua framework API yang cukup populer di kalangan pengembang IoT.
- Keduanya memiliki model komunikasi masing-masing, dan juga mendukung REST API serta MQTT API.

Menggunakan REST ketika:

- Perlu mengambil data dalam rentang tertentu atau historical data.
- Memanipulasi data, seperti buat channel, baca setting, menghapus data, membuat chart.
- Perlu respon yang berbeda atas method yang dikirimkan (seperti GET dan POST).

Menggunakan MQTT ketika:

- IoT memiliki keterbatasan daya (misalnya bekerja dengan baterai) dan atau perlu mengirimkan data yang berukuran lebih kecil. Proses menggunakan MQTT biasanya juga lebih cepat.
- Koneksi data tidak selalu terhubung, dan bandwidth mungkin terbatas.
- Ingin melakukan update yang cepat ke channel.
- Mau langsung menerima update tanpa harus melakukan polling ke server (misal, melakukan penarikan data setiap beberapa detik).

Delay from flespi to client, seconds	REST API python module	Mosquitto MQTT client
Average	0.768	0.0322
Max	1.274	0.0346
Min	0.379	0.0315

Comparison of protocols service part

Secure session	Outcoming bytes	Incoming bytes	Number of packets
HTTPS	1734	4186	20
MQTT over SSL [WiFi]	1274	4159	20
MQTT over SSL [Ethernet]	1186	4075	18

- **Conclusion: MQTTS is 20 times faster** and requires **50 times less traffic** on the task of posting consistent time-valuable data.

Plain TCP session	Outcoming bytes	Incoming bytes	Number of packets
HTTP	675	431	10
MQTT [WiFi]	615	352	11
MQTT [Ethernet]	601	342	11

Power consumption test

Raspberry power consumption	3G modem Internet connection		Ethernet internet connection	
	REST API via curl	Mosquitto MQTT client	REST API via curl	Mosquitto MQTT client
Number of sessions	9243	11728	20964	22864
Test lifetime	2h 39min	2h 50min	5h 53min	6h 21min
mAh per session	0.2164	0.17	0.0954	0.089
Messages per second	0.97	1.15	0.99	0.98
CPU usage, %	64	60	69	64

- Conclusion:** the result is exactly what I've expected. Even in the short run, **MQTT is up to 22% more energy efficient** and **15% faster**. And it does not depend on whether the connection type is 3G or Ethernet.

Cara Komunikasi IoT



Komunikasi IoT

- Wireless
 - Short Range (Bluetooth, Li-Fi, NFC, RFID, Wi-Fi, ZigBee, Z-Wave)
 - Medium Range (LTE, 5G)
 - Long Range (LPWAN, VSAT)
- Wired
 - Ethernet
 - PLC



Contoh Komunikasi IoT

- Bluetooth
- Wifi
- Zigbee
- NFC
- LoRaWAN
- RFID

Bluetooth

- Salah satu teknologi nirkabel jarak pendek yang paling banyak digunakan dalam IoT (Internet of Things) adalah Bluetooth. Protokol komunikasi ini aman dan sempurna untuk transmisi jarak pendek, daya rendah, biaya rendah, dan nirkabel antara perangkat elektronik. Protokol bluetooth yang baru-baru ini diperkenalkan pada protokol IoT adalah BLE (Bluetooth Low Energy).
- BLE ini memberikan jangkauan bluetooth konvensional dikombinasikan dengan supremasi konsumsi daya yang lebih rendah. BLE ini tidak dirancang untuk mentransfer file besar, sehingga BLE ini akan cocok dengan data yang kecil.

Wifi

- Wifi adalah sebuah protokol favorit dalam IoT karena protokol ini memiliki infrastruktur untuk terintegrasi dalam perancangan elektronik perangkat, dan memiliki transfer data yang cepat dengan kemampuan untuk mengontrol sejumlah data yang besar.
- Wifi standar 802.11 menghadirkan kemampuan untuk mentransfer ratusan megabit hanya dalam satu detik, dan pada protokol wifi ini akan menghasilkan konsumsi daya yang besar untuk beberapa aplikasi IoT.

Zigbee

- ZigBee merupakan sebuah protokol IoT yang dirancang untuk industri. ZigBee beroperasi pada frekuensi 2.4 Ghz. Frekuensi ini sangat ideal untuk industri di mana umumnya data yang ditransfer memiliki trafik kecil di antara rumah atau bangunan.

NFC

- NFC (Near Field Communication) adalah protokol IoT yang memanfaatkan hubungan komunikasi data dua arah yang aman. Protokol komunikasi IoT NFC berlaku untuk smartphone. Komunikasi NFC memungkinkan klien untuk terhubung ke perangkat elektronik, menggunakan konten digital dan melakukan transaksi pembayaran tanpa kontak.
- Pekerjaan penting NFC adalah untuk memperluas teknologi kartu “tanpa kontak”. Teknologi ini bekerja dalam jarak 4 cm antara perangkat, dengan mengaktifkan perangkat untuk berbagi informasi.

LoRaWAN

- LoRaWAN atau Long Range Wide Area Network adalah salah satu dari Protokol IoT untuk jaringan area luas. LoRaWAN IoT Network Protocols dirancang khusus untuk mendukung jaringan luas dengan bantuan jutaan perangkat berdaya rendah. Kota pintar menggunakan protokol semacam ini.
- Termasuk komunikasi seluler berbiaya rendah, LoRaWAN juga terkenal di sejumlah industri untuk komunikasi dua arah yang dilindungi. Frekuensi LoRaWAN dapat bervariasi dari satu jaringan ke jaringan lainnya.
- Kecepatan data protokol ini antara 0,3-50 kbps. Di daerah perkotaan, kisaran LoRaWAN bervariasi dari 2 km hingga 5 km. Di daerah pinggiran kota, jangkauan protokol IoT ini sekitar 15 km.

RFID

- RFID (Radio Frequency Identification) bekerja dengan bantuan teknologi tanpa kabel / nirkabel. RFID menggunakan medan elektromagnetik sehingga dapat mengidentifikasi objek.
- Jarak jangkauan RFID jarak pendek adalah sekitar 10 cm. Tetapi RFID jarak jauh bisa mencapai 200 m. Bagian terbaik dari protokol konektivitas IOT RFID adalah mereka tidak memerlukan daya apa pun.

Terima Kasih

Referensi

- http://staff.unila.ac.id/meizano_/2019/01/14/protokol-komunikasi-pada-internet-of-things/
- https://medium.com_/pemrograman/mengenai-mqtt-998b6271f585
- https://flespi.com_/blog/http-vs-mqtt-performance-tests
- https://dzone.com_/articles/protocols-and-standards-behind-iot-coap-ble-mqtt-d