QuakeSense

- **Obiettivi:**
  - Rilevazione e monitoraggio di fenomeni sismici attraverso dei nodi LoRa.
  - Visualizzazione dei dati raccolti su una piattaforma Cloud.
  - Monitoraggio di parametri ambientali come pressione, temperatura e umidità relativa dell’aria.

Componenti del sistema - Nodi LoRa

- I nodi LoRa sono realizzati con delle schede di sviluppo e dotate di microcontrollore (MCU) collegato ai seguenti componenti:
  - un accelerometro a tre assi per rilevare l'attività sismica
  - un modulo radio basato sullo standard LoRa per trasmettere i dati
  - un modulo GPS per ottenere la posizione geografica del mote
  - una batteria LiPo per alimentare il mote
- In presenza di un fenomeno sismico, il MCU della scheda elabora i dati provenienti dall'accelerometro e calcola alcuni parametri importanti che caratterizzano il moto sismico:
  - l'accelerazione di picco al suolo considerando le 2 componenti orizzontali x e y (PGHA – Peak Ground Horizontal Acceleration)
  - l'accelerazione di picco al suolo lungo la componente verticale z (PGVA – Peak Ground Vertical Acceleration)
  - la durata del terremoto intesa come l'intervallo di tempo compreso tra il primo e l'ultimo superamento di una soglia di accelerazione fissata a 80 mg.
- Ciascun mote riceve ed elabora anche i dati provenienti da un modulo GPS ovvero i valori di latitudine, longitudine, altitudine, data e orario. Tali dati vengono trasmessi insieme a quelli relativi all'attività sismica utilizzando il protocollo LoRa.
- Attraverso l'aggiunta di sensori di temperatura, umidità e pressione è possibile monitorare anche alcuni parametri ambientali e trasmetterli al gateway utilizzando il protocollo LoRa.

Componenti del sistema – LoRa Gateway

- Un gateway che ha il compito di raccogliere i dati provenienti dai nodi della rete LoRa, elaborarli e trasmetterli ad una piattaforma Cloud.
- Il gateway può essere implementato attraverso una scheda di sviluppo basata su microcontrollore oppure attraverso un computer a singola scheda (SBC - Single Board Computer).
- Il gateway comunica con i motes utilizzando il protocollo LoRa
- I dati ricevuti dai nodi vengono inviati alla piattaforma Cloud attraverso il protocollo applicativo MQTT basato sullo stack TCP/IP utilizzando un canale di comunicazione backhaul di tipo wireless (WiFi, 3G, 4G, …) oppure wired (Ethernet).

Implementazione del progetto

- **LoRa Nodes:**
  - MCU → STM32 Nucleo F401RE
  - LoRa & GPS module → Dragino LoRa/GPS shield
  - 3-axis accelerometer & environmental sensors → X-NUCLEO-IKS01A2
  - Power → Adafruit PowerBoost 500 shield + LiPo battery
  - Software → QuakeSense_Node.ino (Arduino sketch)
  - LoRa Mode 3: BW = 125 kHz, CR = 4/5, SF = 10
- **LoRa Gateway:**
  - MCU → B-L475E-IOT01A STM32L4 Discovery kit
    It includes the WiFi module ISM43362-M3G-L44 used to establish a connection with the Cloud platform (Adafruit IO)
  - LoRa module → Dragino LoRa/GPS Hat
  - Software → QuakeSense_Gateway.ino (Arduino sketch)
  - LoRa Mode 3: BW = 125 kHz, CR = 4/5, SF = 10
- **Cloud Platform:**
  - Adafruit IO
  - Protocols: MQTT, WiFi

Componenti del sistema – Piattaforma IoT

- Piattaforma IoT basata sulla tecnologia Cloud che consente di:
  - visualizzare dei dati raccolti in tempo reale (real-time) attraverso una dashboard realizzata attraverso una interfaccia utente semplice ed intuitiva.
  - elaborare i dati attraverso dei tools in modo da realizzare delle applicazioni interattive e dei meccanismi di notifica basati sui dati ricevuti

Implementazione del progetto

- **LoRa Nodes:**
  - MCU → STM32 Nucleo F401RE
  - LoRa & GPS module → Dragino LoRa/GPS shield
  - 3-axis accelerometer & environmental sensors → X-NUCLEO-IKS01A2
  - Power → Adafruit PowerBoost 500 shield + LiPo battery
  - Software → QuakeSense_Node.ino (Arduino sketch)
  - LoRa Mode 3: BW = 125 kHz, CR = 4/5, SF = 10
- **LoRa Gateway:**
  - MCU → B-L475E-IOT01A STM32L4 Discovery kit
    It includes the WiFi module ISM43362-M3G-L44 used to establish a connection with the Cloud platform (Adafruit IO)
  - LoRa module → Dragino LoRa/GPS Hat
  - Software → QuakeSense_Gateway.ino (Arduino sketch)
  - LoRa Mode 3: BW = 125 kHz, CR = 4/5, SF = 10
- **Cloud Platform:**
  - Adafruit IO
  - Protocols: MQTT, WiFi

Software utilizzato – Arduino IDE

- Arduino è un sistema di sviluppo integrato Open Source che consente di scrivere dei programmi detti “sketches” in linguaggio C/C++ e di effettuare l’upload su una scheda supportata.
- Scritto in Java e basato sull’ambiente di sviluppo Processing.
- Può essere eseguito su diversi sistemi operativi: Windows, Mac OS X e Linux
- Ampia documentazione disponibile
- Per poter eseguire gli sketch del progetto QuakeSense è necessario installare le seguenti librerie attraverso il Library Manager dell’IDE Arduino:
  - STM32duino LSM6DSL: libreria per gestire l’accelerometro e il giroscopio del sensore LSM6DSL presente sulla scheda di espansione X-NUCLEO-IKS01A2.
  - STM32duino LPS22HB: libreria per gestire il sensore di pressione LPS22HB presente sulla scheda di espansione X-NUCLEO-IKS01A2.
  - STM32duino HTS221: libreria per utilizzare il sensore di temperatura e umidità HTS221 della scheda di espansione X-NUCLEO-IKS01A2
  - LoRa: libreria per utilizzare il modulo LoRa SX1276
  - STM32duino ISM43362-M3G-L44: libreria per utilizzare il modulo WiFi presente sulla scheda B-L475E-IOT01A2 Discovery Kit
  - Adafruit MQTT Library: libreria per implementare la comunicazione tra il gateway e la piattaforma cloud Adafruit IO
  - Adafruit GPS Library: libreria per comunicare con il modulo GPS e decodificare i messaggi provenienti dall’interfaccia seriale del modulo GPS.

Software utilizzato – Adafruit IO

Caratteristiche principali:

- Facilità di utilizzo
- Numerose librerie client disponibili per interfacciarsi alla piattaforma (Arduino, Ruby, Python, Node.js)
- Controllo completo sui dati da parte dell'utente: l'utente può visualizzare e scaricare in qualunque momento i dati memorizzati sulla piattaforma.
- Sicurezza e privacy:
  - Per accedere alla piattaforma si utilizza username e password
  - Per la trasmissione dei dati e per effettuare qualunque operazione su dati, Feeds e Dashboards è necessaria una chiave d'accesso (AIO key). Tale chiave d'accesso può essere utilizzata sia con le API REST, sia per connettersi utilizzando MQTT.
- Ampia documentazione disponibile riguardo le API REST e MQTT.
- Numerosi esempi disponibili per le diverse schede di sviluppo supportate.
- Organizzazione efficiente dei dati attraverso numerosi **blocchi** che possono essere aggiunti ad una Dashboard.
- I blocchi principali sono: Gauge, Number Slider, Momentary Button, Toggle Button, Color Picker, Line Graph, Text Box, Image, Stream.
- Gli elementi fondamentali del sistema Adafruit IO sono i **Feeds** e le **Dashboards.**

Adafruio IO - Feeds

- In Adafruit IO un Feed rappresenta uno stream di dati
- A ciascun Feed sono associati dei metadati che comprendono:
  - i dati relativi ad un sensore o dispositivo collegato al Feed
  - la licenza associata ai dati raccolti
  - il tipo di accesso ai dati:
    - public (dati condivisibili)
    - private (dati privati ovvero visualizzabili solo dall'utente)
  - descrizione generale dei dati che indica ciò che essi rappresentano o a cosa sono associati
  - data di trasmissione dei dati

Adafruit IO - Dashboards

- Una Dashboard è un insieme di blocchi implementati attraverso dei widgets.
- A ciascun blocco può essere associato un Feed ed in questo modo è possibile visualizzare sulla Dashboard i dati associati al Feed.
- Attraverso una Dashboard è possibile:
  - visualizzare i dati relativi ai blocchi
  - creare, eliminare e modificare dei blocchi
  - gestire i Feeds mediante i blocchi

Adafruit IO – Librerie Client per Arduino IDE

- **Adafruit IO Arduino library**
  - Libreria basata sulle API REST che consentono di gestire i Feeds e le Dashboards utilizzando il protocollo HTTP oppure HTTPS ovvero utilizzando i metodi HTTP per i servizi RESTful: **GET**, **POST**, **PUT** e **DELETE.**
  - Funzionalità principali offerte dalla libreria Adafruit IO:
    - Creare un Feed o una Dashboard.
    - Inviare dei dati ad un Feed attraverso il metodo save().
    - Impostare un message handler ovvero una callback per gestire i dati relativi ad un Feed che vengono modificati e inviati dalla piattaforma Adafruit IO.
- **Adafruit MQTT library**
  - Libreria per l'IDE Arduino che consente di gestire i servizi e gli elementi della piattaforma Adafruit IO attraverso il protocollo MQTT.
  - Funzionalità principali:
    - Inizializzare la comunicazione con il broker MQTT presente sulla piattaforma Adafruit IO.
    - Gestire i dati dei Feed attraverso i metodi:
      - ✔ publish() per inviare i dati ad un Feed presente sulla piattaforma.
      - ✔ subscribe() per ricevere i dati relativi ad un Feed quando vengono modificati attraverso la piattaforma Adafruit IO.
    - Gestire le funzionalità di QoS (livelli 0 ed 1) e il topic Will che riguardano il protocollo MQTT.

LoRa e LPWANs (Low Power Wide Area Networks)

- La tecnologia **LoRa** (**Lo**ng **Ra**nge) appartiene alla classe delle LPWANs ed è stata progettata per essere integrata in numerose applicazioni e settori: monitoraggio di parametri ambientali, smart agriculture, asset tracking, smart parking, monitoraggio dell'inquinamento e dei consumi e in ambito healtcare.
- Lo stack LoRa si suddivide in due livelli:
  - **LoRa PHY** (livello fisico)
  - **LoRaWAN** (livello di rete e MAC layer)
- **Caratteristiche principali delle Low Power Wide Area Networks:**
  - **ampia copertura** (10-15 km per aree rurali, e 2-5 km in aree urbane)
  - **basso data rate** (valori medi compresi tra 100 bps a 100 kbps)
  - **trasmissione di pacchetti di piccole dimensioni**
  - **license-free frequency bands** (2,4 GHz, 868 MHz, 915 MHz, 433 MHz)
  - **ottimizzazione dei consumi energetici**

LPWANs – Confronto tra diverse tecnologie esistenti

Name of Standard	Weightless			SigFox	LoRaWAN	LTE-Cat M	IEEE P802.11ah (low power WiFi)	Dash7 Alliance Protocol 1.0	Ingenu RPMA	nWave
	-W	-N	-P							
Frequency Band	TV whitespace (400-800 MHz)	Sub-GHZ ISM	Sub-GHZ ISM	868 MHz/902 MHz ISM	433/868/780/915 MHz ISM	Cellular	License-exempt bands below 1 GHz, excluding the TV White Spaces	433, 868, 915 MHz ISM/SRD	2.4 GHz ISM	Sub-GHz ISM
Channel Width	5MHz	Ultra narrow band (200Hz)	12.5 kHz	Ultra narrow band	EU: 8x125kHz, US 64x125kHz/8x125kHz, Modulation: Chirp Spread Spectrum	1.4MHz	1/2/4/8/16 MHz	25 KHz or 200 KHz	1 MHz (40 channels available)	Ultra narrow band
Range	5km (urban)	3km (urban)	2km (urban)	30-50km (rural), 3-10km (urban), 1000km LoS	2-5k (urban), 15k (rural)	2.5- 5km	Up to 1Km (outdoor)	0 – 5 km	>500 km LoS	10km (urban), 20-30km (rural)
End Node Transmit Power	17 dBm	17 dBm	17 dBm	10µW to 100 mW	EU:<+14dBm, US:<+27dBm	100 mW	Dependent on Regional Regulations (from 1 mW to 1 W)	Depending on FCC/ETSI regulations	to 20 dBm	25-100 mW
Packet Size	10 byte min.	Up to 20 bytes	10 byte min.	12 bytes	Defined by User	~100 -~1000 bytes typical	Up to 7,991 Bytes (w/o Aggregation), up to 65,535 Bytes (with Aggregation)	256 bytes max / packet	Flexible (6 bytes to 10 kbytes)	12 byte header, 2-20 byte payload
Uplink Data Rate	1 kbps to 10 Mbps	100bps	200 bps to 100 kbps	100 bps to 140 messages/day	EU: 300 bps to 50 kbps, US:900-100kbps	~200kbps	150 Kbps ~ 346.666 Mbps	9.6 kb/s, 55.55 kbps or 166.667 kb/s	AP aggregates to 624 kbps per Sector (Assumes 8 channel Access Point)	100 bps
Downlink Data Rate	1 kbps to 10 Mbps	No downlink	200 bps to 100 kbps	Max 4 messages of 8 bytes/day	EU: 300 bps to 50 kbps, US:900-100kbps	~200kbps	150 Kbps ~ 346.666 Mbps	9.6 kb/s, 55.55 kbps or 166.667 kb/s	AP aggregates to 156 kbps per Sector (Assumes 8 channel Access Point)	–
Devices per Access Point	Unlimited	Unlimited	Unlimited	1M	Uplink:>1M, Downlink:<100k	20k+	8191	NA (connectionless communication)	Up to 384,000 per sector	1M
Topology	Star	Star	Star	Star	Star on Star	Star	Star, Tree	Node-to-node, Star, Tree	Typically Star. Tree supported with an RPMA extender	Star
End node roaming allowed	Yes	Yes	Yes	Yes	Yes	Yes	Allowed by other IEEE 802.11 amendments (e.g., IEEE 802.11r)	Yes	Yes	Yes
Governing Body	Weightless SIG			Sigfox	LoRa Alliance	3GPP	IEEE 802.11 working group	Dash7 Alliance	Ingenu (formerly OnRamp)	Weightless SIG
Status	Limited deployment awaiting spectrum availability	Deployment beginning	Standard in development. Scheduled release 4Q 2015	In deployment	Spec released June 2015, in deployment	Release 13 expected 2016	Targeting 2016 release	Released May 2015	In Deployment	In Deployment

LoRa PHY

- **Modulazione a spettro espanso** basata su un segnale Chirp (**Chirp Spread Spectrum** – CSS).
- Integra un **meccanismo di correzione degli errori** chiamato **Forward Error Correction** (FEC).
- Migliora significativamente la il valore di sensitivity del ricevitore e utilizza l'intera larghezza di banda del canale per trasmettere un segnale, rendendolo robusto al rumore del canale e insensibile agli offset di frequenza causati dall'impiego di oscillatori low-cost.
- **Uplink and Downlink Data Rate:** EU: 250 bps to 50 kbps; US: 980 bps to 21.9 kbps.
- **Node transmit power**: EU: <= +14 dBm (PA_BOOST mode: 20 dBm); US: <= +27 dBm.
- **Frequency Bands:**
  - Europe: 868 MHz (863-870 MHz) and 433 MHz
  - Noth America: 915 MHz (902 – 928 MHz)
  - Asia: 470-510 MHz and 779-787 MHz
  - List of frequency plan and regulations by country (TTN website)
- **Bandwidth:** the most used bandwidths are 500 kHz, 250 kHz and 125 kHz.
- **Payload size:** between 51 bytes and 222 bytes, depending on the Spreading Factor (SF)
- **Channels**:
  - EU863-870: 8 channels for uplink and downlink + 1 channel with FSK modulation.
  - US902-928: 9 channels for uplink (903.9 – 904.6) + 9 channels for downlink (923.3 – 927.5)
  - Complete list of LoRa and LoRaWAN frequencies and channels

LoRa PHY

- **Throughput, data rate e range**:

  Throughput, data rate e range dipendono da 3 parametri importanti:

  - Bandwidth (larghezza di banda).
  - Coding Rate (CR): è il rapporto (minore di 1) fra i bit del messaggio (cioè l'informazione) e la lunghezza totale della parola di codice. Il coding rate è utilizzato per la rilevazione e correzione degli errori.
  - Spreading Factor (SF): è il rapporto tra la frequenza del segnale di chip (utilizzato nella modulazione a spettro espanso) e la frequenza di simbolo: SF = chip_rate / symbol_rate.

  All'aumentare della larghezza di banda aumenta il data rate e quindi si riduce il tempo di trasmissione ma d'altra parte viene ridotto il valore di sensitivity.

  All'aumentare dello Spreading Factor (SF) diminuisce il data rate ma aumenta la robustezza del segnale garantendo una copertura (range) maggiore.

  Per calcolare il valore di sensitivity ovvero l'ampiezza minima che può essere ricevuta dal ricevitore in modo che sia possibile elaborare il segnale ricevuto, fissato un livello di SNR (Signal to Noise Ratio), **si può utilizzare la seguente formula:**

$$S = -174 + 10*\log_{10}(BW) + NF + SNR$$

LoRa PHY

- **Il data rate può essere calcolato attraverso la seguente formula:**

LoRa Data Rate (Rb) Formula : -

$$R_b = SF * \frac{\left[\frac{4}{4+CR}\right]}{\left[\frac{2^{SF}}{BW}\right]} * 1000$$

SF = Spreading Factor (6,7,8,9,10,11,12)
CR = Code Rate (1,2,3,4)
BW = Bandwidth in KHz
(10.4,15.6,20.8,31.25,41.7,62.5,125,250,500)
Rb = Data rate or Bit Rate in bps

- **Esistono delle combinazioni predefinite dei 3 parametri descritti in precedenza (BW, CR, SF) detti LoRa Modes:**

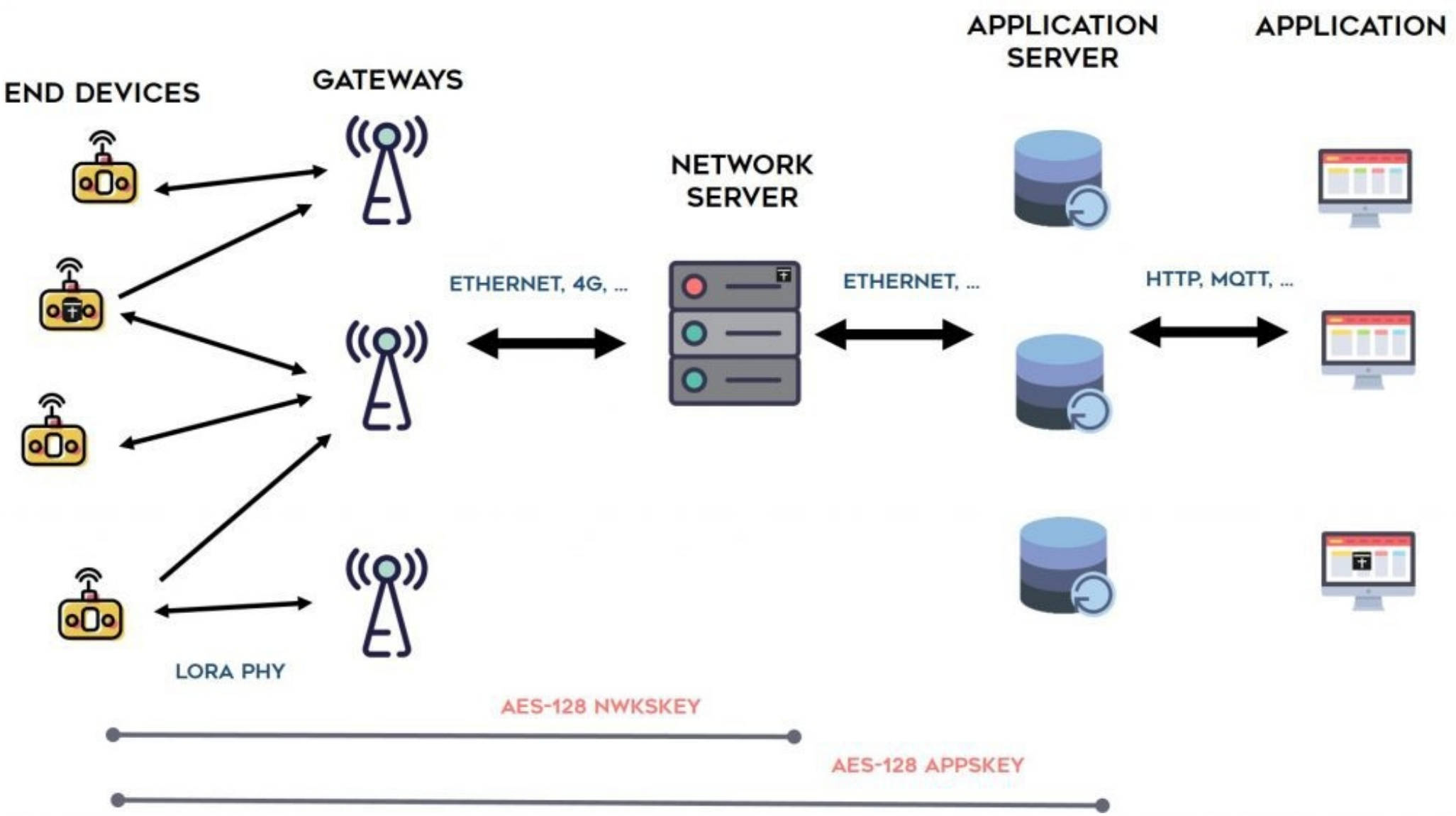
Range ↕ Throughput

				time on air in second for payload size of					
LoRa mode	BW	CR	SF	5 bytes	55 bytes	105 bytes	155 Bytes	205 Bytes	255 Bytes
1	125	4/5	12	0.95846	2.59686	4.23526	5.87366	7.51206	9.15046
2	250	4/5	12	0.47923	1.21651	1.87187	2.52723	3.26451	3.91987
3	125	4/5	10	0.28058	0.69018	1.09978	1.50938	1.91898	2.32858
4	500	4/5	12	0.23962	0.60826	0.93594	1.26362	1.63226	1.95994
5	250	4/5	10	0.14029	0.34509	0.54989	0.75469	0.95949	1.16429
6	500	4/5	11	0.11981	0.30413	0.50893	0.69325	0.87757	1.06189
7	250	4/5	9	0.07014	0.18278	0.29542	0.40806	0.5207	0.63334
8	500	4/5	9	0.03507	0.09139	0.14771	0.20403	0.26035	0.31667
9	500	4/5	8	0.01754	0.05082	0.08154	0.11482	0.14554	0.17882
10	500	4/5	7	0.00877	0.02797	0.04589	0.06381	0.08301	0.10093

LoRaWAN – Caratteristiche principali

- L’architettura di una rete LoRaWAN prevede una **topologia a stella di stelle** ed è costituita da 3 elementi fondamentali:
  - End-devices (LoRaWAN nodes)
  - Gateway
  - Network Server
- Ogni nodo è connesso ad uno o più gateway attraverso un collegamento single-hop.
- I gateway si connettono a turno ad un Network Server (NetServer) comune attraverso un backhaul channel utilizzando lo stack TCP/IP ed hanno il compito di ritrasmettere i messaggi ricevuti fra end-device e Network Server.
- Un nodo può essere associato a più di un gateway ma può essere associato ad un unico Network Server. Questo implica che diversi gateways possono ricevere uno stesso messaggio inviato da un nodo e ritrasmetterlo ad uno stesso Network Server il quale deve occuparsi di filtrare i messaggi duplicati.
- Le comunicazioni sono nella maggior parte dei casi bidirezionali fornendo trasmissioni in uplink e downlink.
- L’infrastruttura di una rete LoRa ha la capacità di adattare il data rate e le frequenze utilizzate, attraverso un meccanismo chiamato Adaptive Data Rate (ADR) che permette di ottimizzare la durata della batteria e il data rate a seconda delle condizioni del canale.
- Gli end-device cambiano canale in modo pseudo-random per ogni trasmissione in modo da garantire una maggiore robustezza del segnale alle interferenze.
- Devono essere rispettati il massimo duty-cycle e durata della trasmissione relativi alla sub-band in base alla regolamentazione locale consentita:
  - **trasmissioni nella banda 868 MHz → duty cycle pari a 0,1 % ovvero 3,6 s**
  - **trasmissioni nella banda (865 – 868 MHz) → duty cycle pari a 1 % ovvero circa 36 s**.

LoRaWAN - Architettura



LoRaWAN – Classi end-devices

Il protocollo LoRaWAN prevede 3 classi di end-devices:

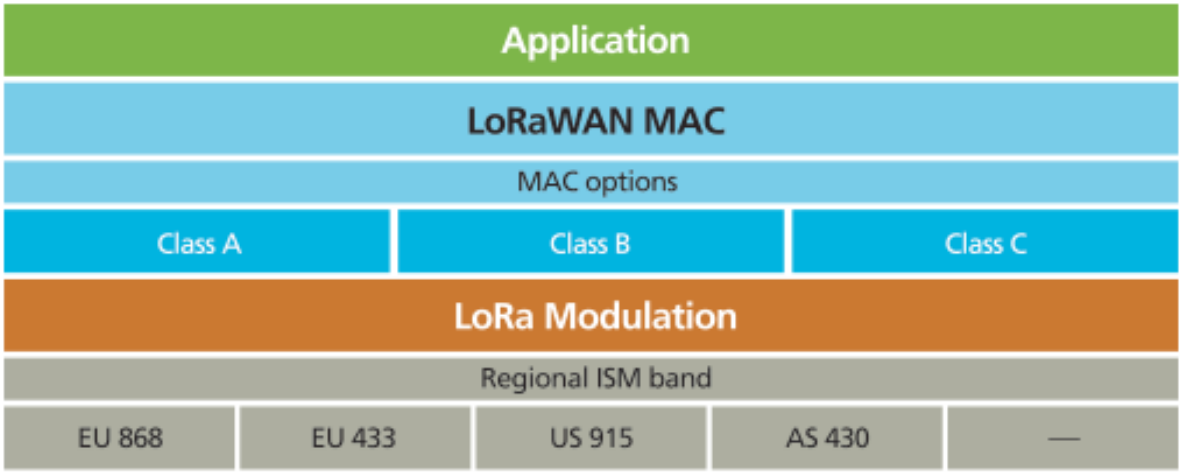
- **Class A: Bi-directional end-devices**
  - Possono implementare comunicazioni bidirezionali in cui dopo ogni messaggio mandato in uplink, vengono aperte due finestre (RX1 e RX2) in cui il dispositivo è in ascolto e può ricevere messaggi in downlink. La seconda finestra (RX2) può essere aperta su un canale differente rispetto a quello utilizzato nella prima finestra (RX1).
- **Class B: Bi-directional end-device with scheduled receive slots**
  - Consentono la comunicazione bidirezionale utilizzando oltre alle finestre RX1 e RX2, utilizzate per classe A, anche un ulteriore finestra di ricezione che viene attivata dal Network Server attraverso l'invio di un Beacon di sincronizzazione: in questo modo il Network Server sa quando un end-device è in ascolto.
- **Class C: Bi-directional end-devices with maximum receive slot**
  - Prevedono una finestra di ricezione costantemente aperta tranne durante la trasmissione ovvero un end-device di classe C è sempre in modalità di ricezione tranne quando deve trasmettere.

LoRaWAN – Attivazione end-devices

- **L'attivazione di un end-device può essere effettuata in 2 modi distinti:**
  - **Over-The-Air Activation (OTAA)**
    - i nodi eseguono la procedura di join per essere autenticati nella rete e poter iniziare ad inviare dati al Network Server attraverso i messaggi di **join request** e **join accept**.
    - La procedura di join richiede che l'end-device venga configurato definendo:
      - ✔ **DevEUI**: identificativo globale univoco del nodo (IEEE EUI64)
      - ✔ **AppEUI**: identificativo applicazione
      - ✔ **AppKey**: chiave AES di 128 bit
  - **Activation By Personalization (ABP)**
    - Per l'attivazione del nodo non viene eseguita la procedura di join ma il DevAddr e le due chiavi di sessione NwkSKey e AppSKey sono memorizzate direttamente nel nodo invece dei parametri DevEUI, AppEui e AppKey.
- Dopo aver eseguito l'attivazione, ogni nodo possiede:
  - un indirizzo IEEE EUI32 identificativo (**DevAddr**).
  - un identificativo globale per l'applicazione (**AppEUI**) associata al nodo.
  - una chiave di sessione di rete (**NwkSKEY**) usata per garantire la sicurezza a livello di rete e per garantire l'integrità dei messaggi verificando il MIC (Message Integrity Code).
  - una chiave di sessione per l'applicazione (**AppSKey**) specifica per ogni nodo che viene utilizzata per cifrare il payload dei messaggi inviati.

LoRaWAN – Protocol stack and Frame Format

LoRaWAN Protocol Stack



LoRaWAN Frame Format

PHYPayload:	MHDR : 8	MACPayload			MIC : 32
MACPayload:	FHDR : 56..176	FPort : 8	FRMPayload (encrypted)		
FHDR:	DevAddr : 32	FCtrl : 8	FCnt : 16	FOpts : 0..120	
MHDR:	MType : 3	RFU : 3	Major : 2		
FCtrl: Uplink:	ADR : 1	ADRAckReq : 1	ACK : 1	FPending : 1	FOptsLen : 4
FCtrl: Downlink:	ADR : 1	ADRAckReq : 1	ACK : 1	RFU : 1	FOptsLen : 4
FOpts:	MACCommand_1 : 8..40	...	MACCommand_n : 8..40		
MACCommand:	CID : 8	Args : 0..32			

MQTT – Caratteristiche principali

- MQTT è un protocollo applicativo per la messaggistica basato sul paradigma pubblicazione/sottoscrizione (publish/subscribe).
- È leggero, open, semplice e progettato in modo da essere facile da implementare. Queste caratteristiche lo rendono ideale per molte applicazioni riguardanti il mondo dell'Internet of Things (IoT) e le comunicazioni di tipo Machine to Machine (M2M).
- Fornisce il servizio di Quality of Service (QoS) dei dati trasportati.
- Consente di minimizzare il consumo della banda ed è un protocollo lightweight.
- Data Agnostic: consente il trasferimento di dati di qualsiasi tipo.
- Continuous Session Awareness.
- Versione attuale: 3.1.1

MQTT – Modello Publish/Subscribe

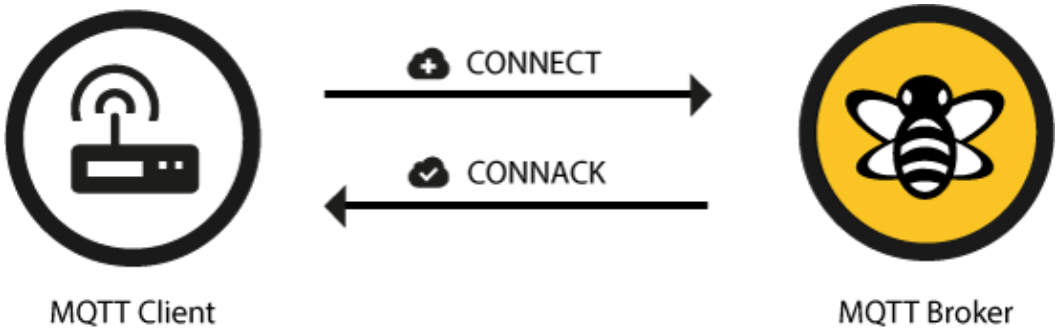
- MQTT è basato sul modello publish/subcribe.
- Il modello pubish/subscribe, invece, è caratterizzato dalla presenza di clients che non sono direttamente collegati tra loro ma essi inviano i messaggi ad un entità chiamata broker il quale si occupa di filtrare i messaggi ricevuti ed inviarli ai client destinatari.
- Un client è rappresentato da un generico dispositivo che:
  - implementa uno stack TCP/IP
  - possiede una libreria MQTT
  - è collegato ad un broker MQTT attraverso una rete di telecomunicazione
- Un broker è rappresentato da un'entità software che:
  - deve gestire un numero elevato di connessioni con i clients
  - deve occuparsi di filtrare i messaggi ricevuti in base al topic e inoltrarli ai subscribers
  - deve essere in grado di gestire il servizio di QoS
  - deve gestire l'autenticazione dei clients
- **I clients che inviano (publish) dei messaggi sono detti publishers.**
- **I clients che ricevono i messaggi, ovvero quelli che si registrano (subscribe) con il broker come destinatari di alcuni topic, sono detti subscribers.**
- Un client può rivestire il ruolo sia di publisher, sia di subscriber.

MQTT – Topics e filtraggio dei messaggi

- Il filtraggio dei messaggi da parte del broker MQTT è di tipo subject-based e per questo motivo qualunque messaggio trasmesso è associato ad un **topic** che specifica l'argomento associato al messaggio.
- Un publisher invia un messaggio specificando il topic a cui è associato il messaggio.
- Un subscriber effettua una subscription ad uno o più topic: in questo modo il broker MQTT inoltrerà al subscriber solo i messaggi che contengono il topic a cui il subscriber è associato (ovvero a quelli a cui il subscriber si è sottoscritto).
- Un topic è rappresentato da una stringa di caratteri con codifica UTF-8 ed è costituito da uno o più "**topic levels**".
- Ciascun topic level è separato dal precedente e dal successivo dal carattere '/'.
- Un topic level può contenere anche spazi ma in generale è conveniente utilizzare delle stringhe senza spazi.
- Ciascun topic deve essere costituito da almeno un carattere.
- I nomi dei topic sono case-sensitive.
- Struttura di un topic:

MQTT – Connessione tra Client e Broker

- Il protocollo MQTT è basato sullo stack TCP/IP il quale deve essere presente sia sui clients, sia sul broker.
- La comunicazione tra client e broker è inizializzata quando il client invia un messaggio di **CONNECT** al broker.
- Il broker risponde inviando un messaggio di **CONNACK** contenente un codice di stato che rappresenta lo stato della connessione.
- Una volta che la connessione è inizializzata, il broker la manterrà aperta fino a quando il client non invia il messaggio di **DISCONNECT** oppure fino a quando la connessione viene persa.



MQTT-Packet:

CONNECT

contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60

MQTT-Packet:

CONNACK

contains:	Example
sessionPresent	true
returnCode	0

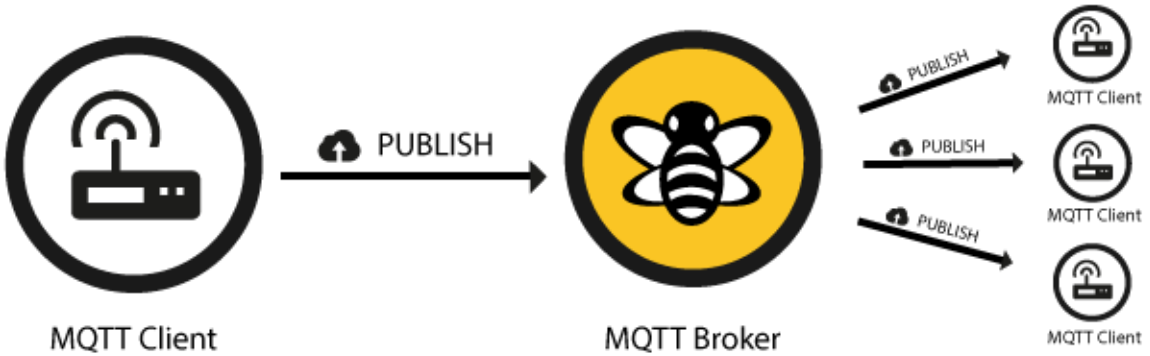
MQTT-Packet:

DISCONNECT

no payload

MQTT - PUBLISH

- Dopo che il client si è connesso al broker MQTT può effettuare l'invio (publish) dei messaggi al broker specificando il topic che verrà utilizzato dal broker per inoltrare il messaggio ai subscribers di tale topic.
- Il messaggio di PUBLISH contiene anche il payload che rappresenta i dati da trasmettere. Il dati contenuti nel payload possono essere di qualsiasi tipo/formato (bytes, testo, JSON, ...).



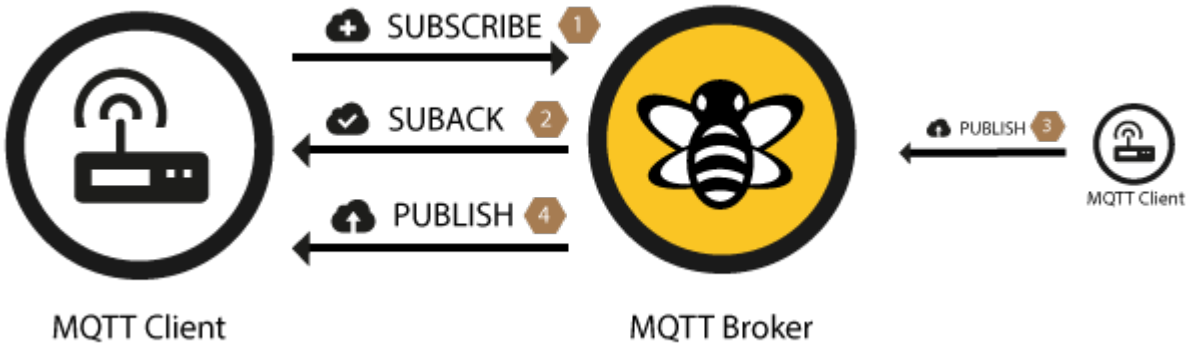
MQTT-Packet:

PUBLISH

contains:	Example
packetId (always 0 for qos 0)	4314
topicName	"topic/1"
qos	1
retainFlag	false
payload	"temperature:32.5"
dupFlag	false

MQTT - SUBRCRIBE

- Quando un client vuole connettersi al broker MQTT come subscriber per uno o più topics invia un messaggio di **SUBSCRIBE**.
- Ciascuna subscription viene confermata con un messaggio di **SUBACK**.



MQTT-Packet:

SUBSCRIBE

contains:		Example
packetId		4312
qos1	} (list of topic + qos)	1
topic1		"topic/1"
qos2	}	0
topic2		"topic/2"
...		...

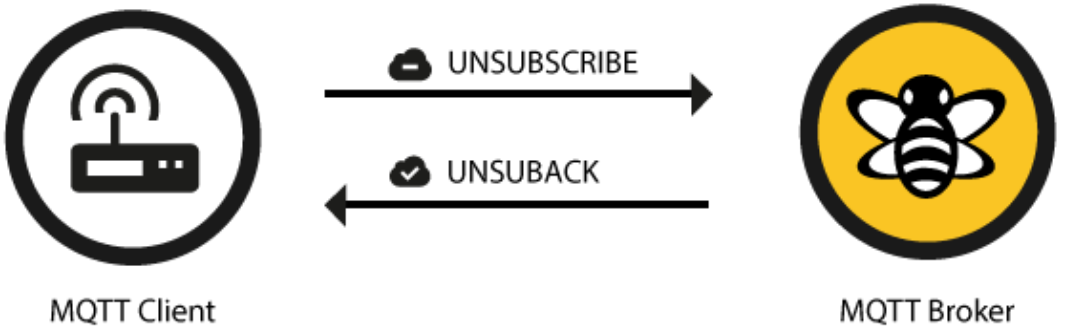
MQTT-Packet:

SUBACK

contains:		Example
packetId		4313
returnCode 1	(one returnCode for each topic from SUBSCRIBE, in the same order)	2
returnCode 2		0
...		...

MQTT - UNSUBSCRIBE

- Un client può effettuare la “unsubscription” che consente di eliminare le subscription esistenti effettuate dal client.
- Una unsubscription viene effettuata dal client attraverso il messaggio di **UNSUBSCRIBE**.
- Il broker invierà l’acknowledgement di avvenuta unsubscription attraverso il messaggio di **UNSUBACK**.



MQTT-Packet:
UNSUBSCRIBE

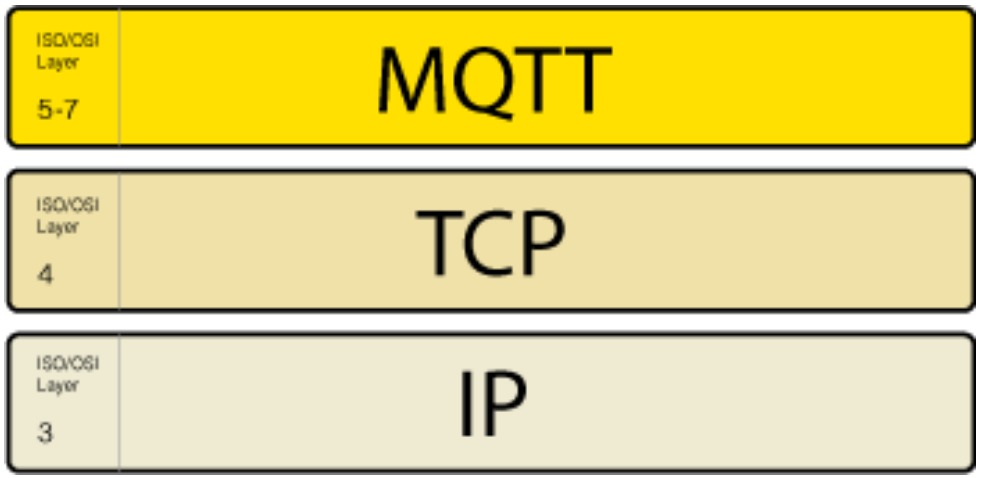
contains:	Example
packetId	4315
topic1 } (list of topics)	“topic/1”
topic2	“topic/2”
...	...

MQTT-Packet:
UNSUBACK

contains:	Example
packetId	4316

MQTT – Quality of Service (QoS)

- Il termine Quality of Service è utilizzato per indicare l'affidabilità e le garanzie di invio di un messaggio.
- Consente al client di scegliere il livello di QoS in base all'affidabilità della rete e alla logica dell'applicazione, gestendo la ritrasmissione e garantendo la consegna del messaggio indipendentemente da quanto sia inaffidabile il livello trasporto sottostante.
- In MQTT ci sono 3 livelli di QoS:
  - QoS 0 – at most once: il messaggio viene consegnato al massimo una volta
  - QoS 1 – at least once: il messaggio viene consegnato almeno una volta
  - QoS 2 – exactly one: il messaggio viene consegnato esattamente una volta



MQTT - QoS 0

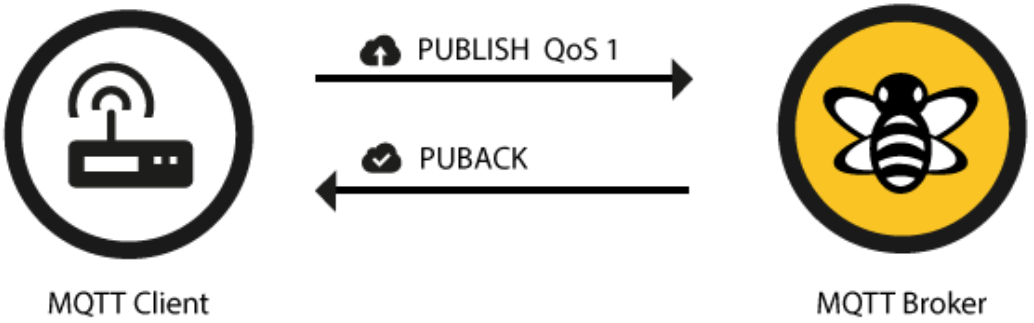
QoS 0 – at most once: Best effort delivery

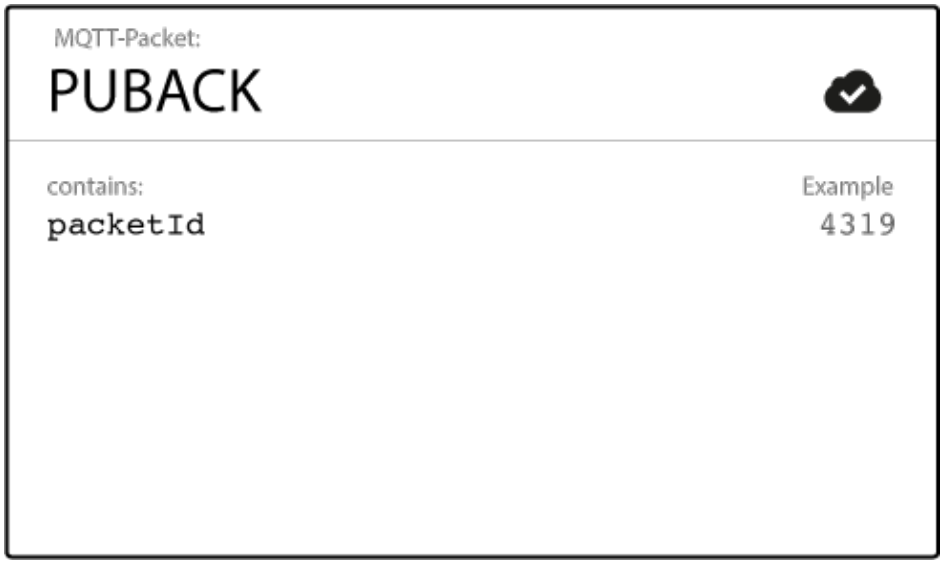
- Il destinatario non invia l'acknowledgement (ACK) relativo alla ricezione del messaggio ed il broker non memorizza il messaggio per poi ritrasmetterlo.
- Stesse garanzie del protocollo TCP.

MQTT - QoS 1

QoS 1 – at least once

- Quando si utilizza il livello 1 di QoS vi è garanzia che il messaggio sarà consegnato almeno una volta.
- Un messaggio con QoS 1 viene memorizzato dal client fino a quando esso non riceve il messaggio di PUBACK da parte del broker.
- Se il messaggio di PUBACK non viene ricevuto dopo un certo intervallo di tempo, il client invia di nuovo il messaggio di PUBLISH impostando il dupFlag su true.

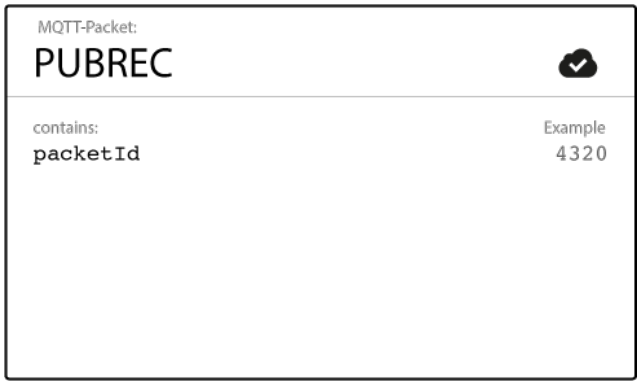


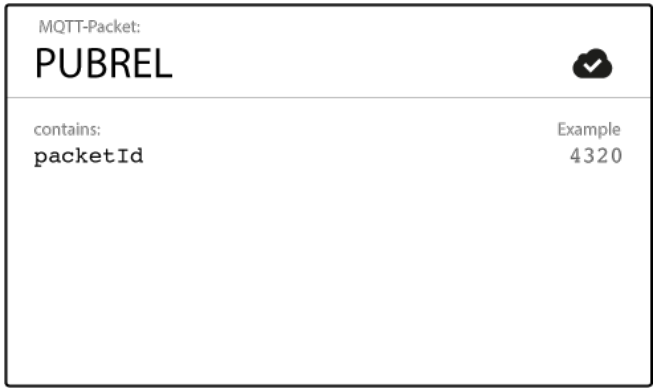


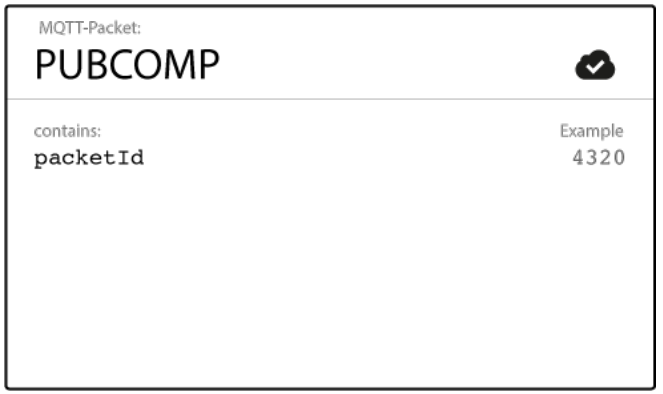
MQTT - QoS 2

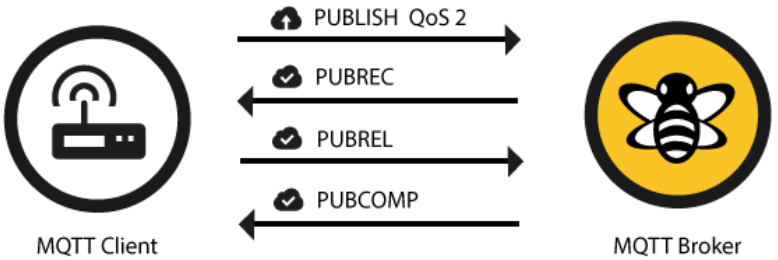
QoS 2 – exactly once

- Il livello 2 di QoS garantisce che il messaggio sia ricevuto una sola volta dal destinatario attraverso i seguenti messaggi di controllo:
  - **PUBREC:** messaggio di acknowledgement inviato dal destinatario (broker) quando riceve un messaggio di **PUBLISH** con QoS 2.
  - **PUBREL:** messaggio di acknowledgement inviato dal mittente al destinatario dopo che il mittente ha ricevuto il messaggio di PUBREC che notifica l'avvenuta ricezione del messaggio di PUBLISH da parte del destinatario.
  - **PUBCOMP:** messaggio finale di acknowledgement inviato dal destinatario al mittente.



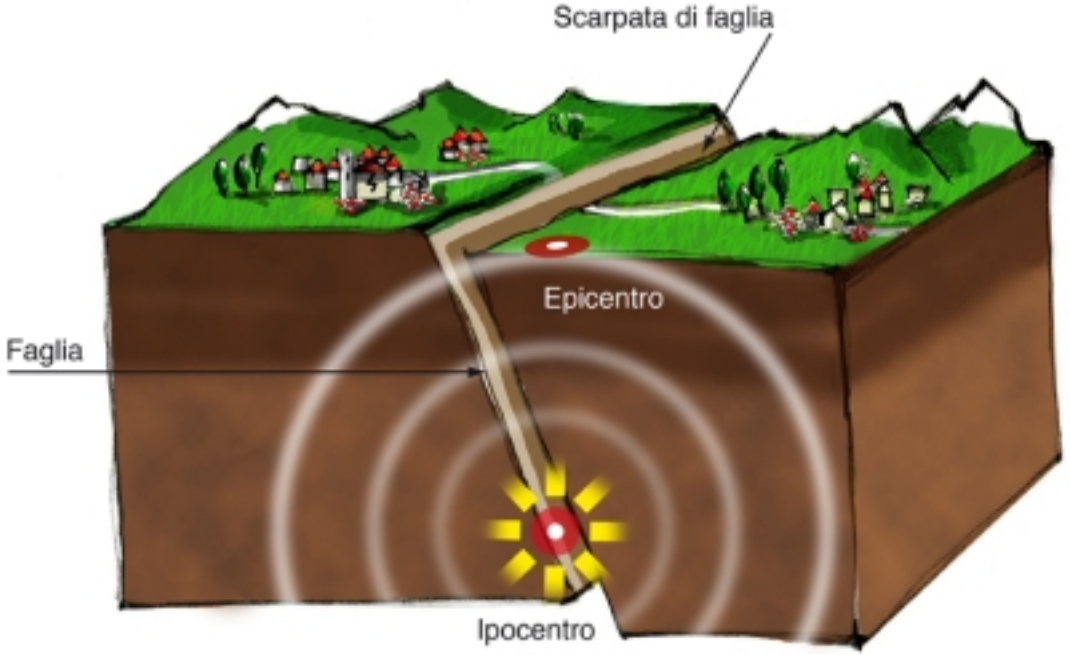


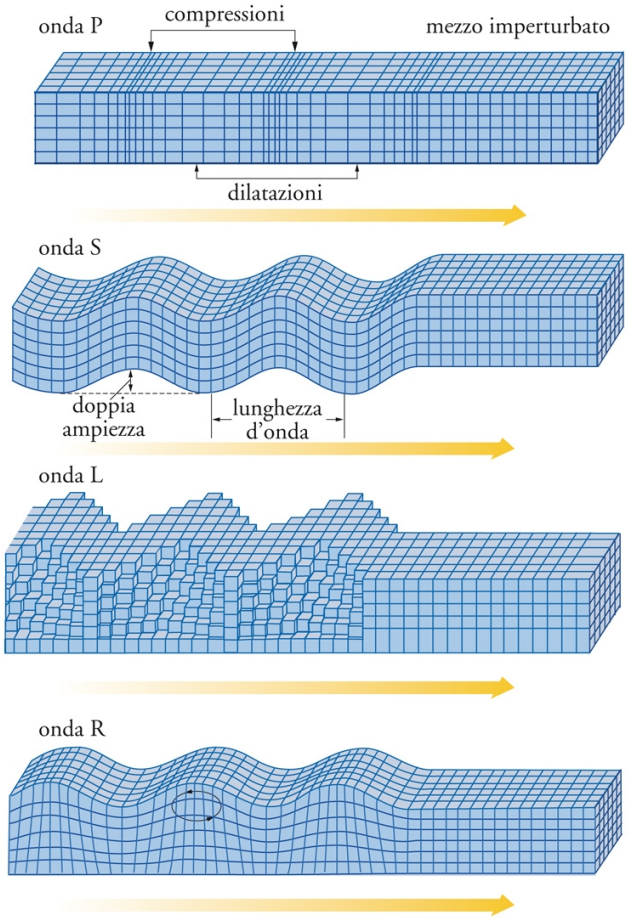




Fenomeni sismici

- I terremoti, o sismi, consistono in una serie di rapide oscillazioni del terreno causate da una brusca liberazione di energia elastica da una zona del sottosuolo definita **ipocentro**.
- Il punto della della superficie situato sulla verticale dell'ipocentro viene chiamato **epicentro.**
- I movimenti del suolo sono spesso descritti come:
  - **ondulatori**: provocati da onde sismiche a bassa frequenza
  - **sussultori**: provocati da onde sismiche ad alta frequenza





Fenomeni sismici – Onde sismiche

- Le onde sismiche possono essere di **3 tipi**:
  - **ONDE P (onde primarie)**
    - Sono **onde longitudinali o di compressione** infatti determinano fenomeni di compressione e rarefazione delle rocce producendo una variazione di volume.
    - Le onde primarie sono quelle che si propagano più rapidamente e sono le prime ad essere rilevate dai sismometri e registrate dai sismografi.
    - Possono propagarsi sia nei solidi che nei fluidi.
  - **ONDE S (onde secondarie)**
    - Sono **onde trasversali o di taglio** che si propagano con oscillazioni su un piano perpendicolare alla direzione di propagazione.
    - Le onde S sono più lente di quelle P e non si propagano nei fluidi.
  - **ONDE R e L (onde superficiali)**
    - Sono il risultato della combinazione delle onde P con le onde S.
    - Sono quelle che provocano i danni maggiori infatti possiedono una minore velocità ma anche una ampiezza elevata.

Parametri descrittivi del moto sismico

- Nello studio dell'attività sismica, spesso si prende in considerazione l'**attività strong motion** caratterizzata da vibrazioni di ampiezza e periodo tali da produrre danni su ambiente e infrastrutture ed è rilevabile con i più comuni strumenti.
- Per valutare gli effetti dell'attività strong motion su un si considerano le 3 componenti ortogonali della traslazione, le quali possono essere registrate in termini di:
  - **accelerazione** attraverso accelerometri e accelerogrammi
  - **velocità** attraverso velocimetri
  - **spostamenti** attraverso sismometri
- **I parametri più importanti ai fini ingegneristici per descrivere il moto sismico sono quelli che si ricavano dagli accelerometri ed essi sono:**
  - **Ampiezza massima dell'accelerazione (PGA)**
  - **Durata**
  - **Contenuto in frequenza (Spettro di Fourier dell'onda sismica)**

Parametri descrittivi del moto sismico - PGA

- **Ampiezza massima dell'accelerazione (PGA)**
- Detta anche accelerazione di picco al suolo (Peak Ground Acceleration) e corrisponde al picco più alto in valore assoluto registrato dall'accelerometro.
- Si possono misurare due tipi di PGA:
  - **PGHA (Peak Ground Horizontal Acceleration):**

    ampiezza massima dell'accelerazione ottenuta considerando una tra le due componenti orizzontali (x o y).
  - **PGVA (Peak Ground Vertical Acceleration):**

    ampiezza massima dell'accelerazione ottenuta considerando la componente verticale (z).

Parametri descrittivi del moto sismico - Durata

- Per calcolare la durata del moto sismico (Td) si fa riferimento alla durata dell'attività di strong motion.
- Può essere calcolata in diversi modi:
  - **Durata "bracketed" (Tb)**:

    intervallo di tempo compreso tra il primo e l'ultimo superamento di una soglia di accelerazione (compresa tra 0.05 e 0.08 g).
  - **Durata di "Trifunac" (Tt)**:

    intervallo tempo in cui l'energia della registrazione è compresa tra il 5% e il 95% dell'energia totale.

Parametri descrittivi del moto sismico – Contenuto in frequenza

- Descrive come varia l'ampiezza del moto sismico in relazione alle frequenze contenute nel segnale.
- Un' onda sismica può essere rappresentata da un segnale periodico, dunque può essere descritta attraverso l'espansione in serie di Fourier: per il teorema di Fourier una funzione periodica s(t) di periodo T si può esprimere come sommatoria infinita di funzioni armoniche.
- Un accelerogramma è generalmente rappresentato da una funzione discreta x(KT) = x_0, x_1, …, x_{N-1} e quindi per ricavare lo spettro di Fourier viene applicata la Trasformata Discreta di Fourier che consente di ottenere una successione di N numeri complessi X_0, X_1, …, X_{N-1}:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-ik\frac{2\pi}{N}n} \qquad k = 0, \ldots, N-1$$

- Dallo spettro di Fourier si ricava il valore della frequenza (o del periodo) fondamentale o predominante, cioè quello a cui corrisponde il valore dell'ampiezza massima.