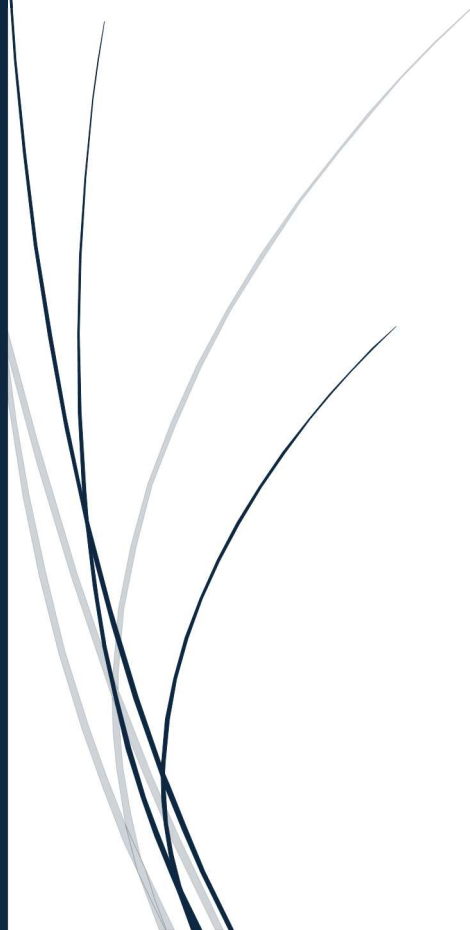


A.A. 2024-2025

Networking Aspects for Internet of Things

Appunti in italiano delle lezioni



GREGORIO VOCI

Sommario

Wireless Sensor Networks	6
Introduzione	6
Asimmetrica	6
Hardware	6
Features	7
Fattori che Influenzano il Design delle Reti di Sensori	7
Fault Tolerance (Reliability)	7
Scalability	7
Production Cost	7
Sensor Hardware	7
Sensor Network Topology	7
Operating Environment	8
Transmission Media	8
Power Consumption	8
Architettura	9
Application Layer	9
Sensor Network Management Protocol (SMP)	9
Sensor Query And Tasking Language (SQTL)	9
Transport Layer	10
Reliable Multi-Segment Transport (RMST)	10
Pump Slowly Fetch Quickly (PSFQ)	10
Event to Sink Reliability (ESRT)	12
Routing Protocols	12
Flooding	13
Gossiping	13
Problemi Flooding & Gossiping	13
SPIN (Sensor Protocol for Information via Negoziation)	14
SPIN-1	14
SPIN-2	15
Direct Diffusion	15
LEACH (Low Energy Adaptive Clustering Hierachy)	16
Link Layer	16
Principali Sorgenti di Spreco di Energia	17
Challenges nel MAC delle WSN	17
IEEE 802.11	18
S-Mac	18
Schema Base	18
Consumo Energetico dalle Sorgenti A e B	18
Tempo in Fase di Sleep	19
Consumo Energetico nei Nodi Intermedi	19
Architettura e Modello di Correlazione nelle WSN	20
C-Mac (Collaborative MAC)	20
Componenti Principali	20
E-Mac (Event MAC)	21
N-Mac (Network MAC)	21
Performance	22

Conclusioni	23
Bluetooth	24
Overview	24
Pila Protocollare	24
Profiles	24
Radio Specification	25
Radio Frequency (RF – livello fisico)	25
Baseband	25
Link Manager	25
Logical Link (Control & Aadaptation)	25
Radio Spectrum	25
Differenze con FSK (filtro gaussiano)	25
Protocols	25
Middleware Protocol Group	25
Master-Slave	26
Piconets	26
Physical Link Types	26
Pacchetti	26
Access Code	27
Packet Header	27
Connection States	27
Frequency Hopping Spread Spectrum (FHSS)	28
Time Division Duplex Scheme (TDD)	29
Indirizzamento	29
Piconet Management	29
Hold	29
Sniff	29
Park	30
L2CAP	30
IEEE 802.15.4	31
Network Layer	31
Data Link Layer	31
Superframe	33
ACK Mechanism	33
Channelization	34
Limiti	34
802.15.4 vs S-MAC	34
Parametri CSMA/CA	35
Delivery Ratio vs Packet Error Rate	36
Miglioramenti	37
Computazione offline Model-Based	37
Park's Model	37
Computazione online Model-Based	37
Adattamento Measurement-Based	37
Adattamento Learning-Based	37
IEEE 802.15.4e	39

MAC Behaviors Modes	39
Frequency Translation	40
ZigBee	41
Alliance	41
Applicazioni	41
Caratteristiche Base	41
Stack Architecture	42
Tipologie di Dispositivi	42
Zigbee Compliant Platform (ZCP)	42
Profili Applicativi	43
Dispositivi Multi-Profilo	43
802.15.4 vs ZigBee	43
Routing	43
LoRaWAN	44
Classi di Dispositivi	44
Classe A: Alimentati a Batteria	44
Classe B: Low Latency	44
Classe C: No Latency	44
Ingresso nella Rete	44
Confirmed-Data Message	45
Tecnologie Usate in LoRaWAN	45
Processing Gain	46
Adaptive Data Rate (ADR)	46
LoRaWAN (Technical Overview PDF)	46
Where Does LPWAN Fit?	46
What is LoRaWAN?	47
LoRaWAN Regional Overview	47
Comparing LPWAN Technology Options	48
LPWAN Cost vs. Legacy Systems	49
MQTT (Message Queue Telemetry Transport)	50
Caratterisstiche	Errore. Il segnalibro non è definito.
Modello/Achitettura MQTT	50
Subscription	51
Message Format	51
MQTT QoS	53
Connect e Subscribe	54
Keep Alive Timer	55
MQTT Will Message	55

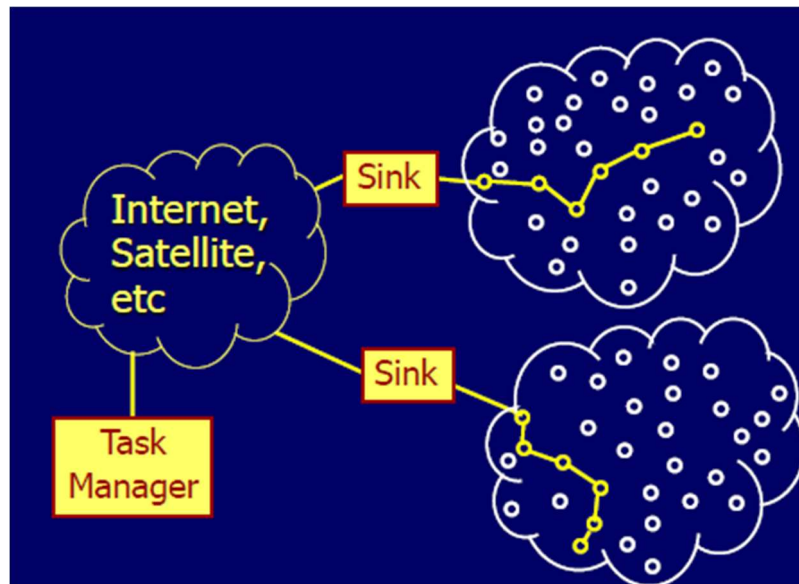
Topic Wildcards	55
MQTT-S	56
<i>IPv6</i>	57
Limiti di IPv4	57
Vantaggi di IPv6	57
Svantaggi IPv6	57

Wireless Sensor Networks

Introduzione

L'architettura di una WSN è composta nella pratica da diverse centinaia di nodi, con una densità di $20 \frac{\text{nodi}}{m^3}$.

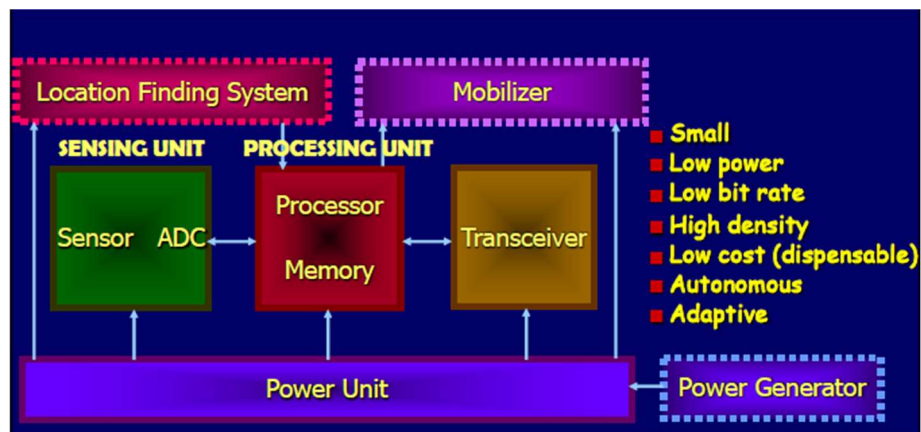
Asimmetrica



Nella rete viaggiano più dati dai sensori ai **sink** ma meno dati dai **task manager** ai sink.

Uno o più dati confluisce nei sink che possono comunicarli dal dispositivo alla rete internet.

Hardware



- **Processing Unit (CPU+RAM):** La memoria è inserita nella CPU per ridurre i costi.
- **Sensor ADC:** Analog Digital Converters. Il numero di bit stabilisce la precisione della conversione (Hardware Cast).
- **Transceiver:** Ha capacità di trasmettere e ricevere.
- **Power Unit:** Batteria+)]
- **Mobilizer:** Modulo che rileva i movimenti

- **LFS:** Usa la posizione relativa dalla posizione del sink, per evitare di usare la posizione GPS (alto consumo energetico).

Features

- **Applicazioni:** Militare, Ambiente, Salute ecc.
- **Tipi di Sensori:** Termico, Visivo, Infrarosso ecc.
- **Scopo dei Sensori:** Temperatura, Umidità, Livello di Stress Meccanico ecc.

Fattori che Influenzano il Design delle Reti di Sensori

Fault Tolerance (Reliability)

I sensori potrebbero rompersi o spegnersi a causa di una mancanza di alimentazione, danni fisici, interferenze ambientali ecc. Il fallimento di un nodo non deve compromettere l'integrità della rete complessiva.

Ciò implica che all'Affidabilità/Fault Tolerance possiamo dire che viene modellata dalla seguente equazione:

$$R_k(+) = \exp(-\lambda_k t)$$

In presenza di N nodi:

$$R(+) = 1 - \prod_{k=1}^N [1 - R_k(+)] \Rightarrow R(+) = 1 - \Pr(\text{All Fails})$$

I protocolli potrebbero essere progettati per affrontare il livello di fault tolerance richiesto dalla WSN.

Scalability

La densità di nodi, in una regione, può variare da pochi a diverse centinaia. Essa si calcola con la seguente formula:

$$\mu(R) = \frac{N\pi * R^2}{A}$$

Dove N è il numero di nodi sparsi nella regione A ed R è il range di trasmissione radio. In sostanza, è il numero di nodi nel raggio di trasmissione di ogni sensore nella regione A.

Production Cost

I costi di produzione devono essere bassi, in modo da giustificare la rete.

Sensor Hardware

Vedi architettura

Sensor Network Topology

Necessitiamo che la topologia della rete venga mantenuta nel tempo, siccome essa potrebbe cambiare per diversi motivi:

- **Pre-Deployment e Deployment Phase:** Questa fase ha lo scopo di ridurre il costo di installazione dei sensori e di aumentare la flessibilità, promuovendo la fault tolerance;

- **Post-Deployment Phase:** Dopo il deploy dei sensori, la topologia cambia a causa dei cambiamenti che avvengono ai nodi (posizione, segnale, batteria ecc.). Assumendo che la connessione viene stabilita allo stesso modo per tutti i nodi, la distanza di hop prevista è:

$$d_{hop} = \frac{2R}{3} \geq d_{nei}$$

Dove R è il range radio, d_{nei} è la distanza dal nodo più vicino.

Operating Environment

La rete di sensori potrebbe dover operare in diversi ambienti:

- in incroci trafficati;
- all'interno di grandi macchinari;
- sul fondo dell'oceano;
- all'interno di un tornado;
- sulla superficie dell'oceano;
- in un campo contaminato biologicamente o chimicamente in un campo di battaglia oltre le linee nemiche;
- in una casa o in un grande edificio;
- in un grande magazzino;
- attaccato ad animali;
- attaccato a veicoli in rapido movimento;
- in un canale di scolo o in un fiume con corrente.

Transmission Media

Vengono utilizzati mezzi di trasmissione radio, ottici, infrarossi ecc. I mezzi radio sono divisi in bande di diversa frequenza sulla base dell'ISM (Industrial, Scientific and Medical Bands): 433Mhz, 915Mhz, 2400Mhz.

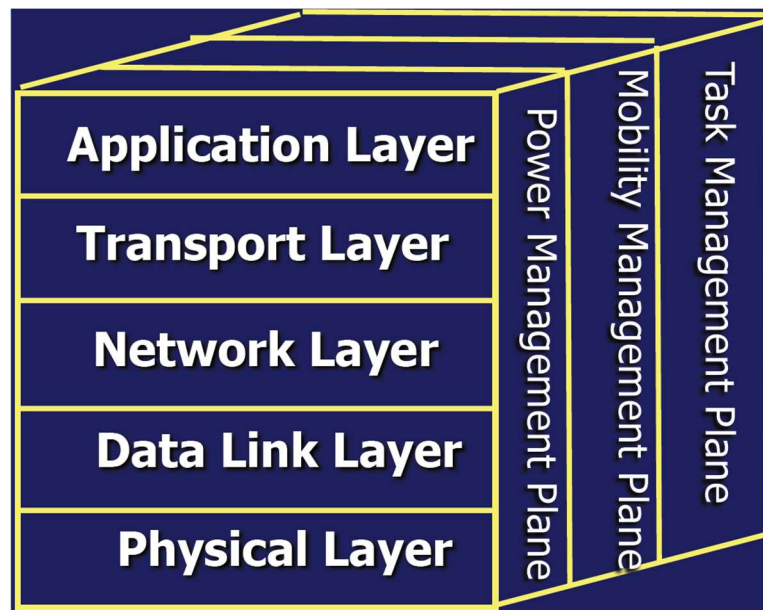
Per raggiungere gli obiettivi di miniaturizzazione, vengono usate anche le bande tra i 4Ghz e i 5 Ghz. Con esse un antenna più piccola ha un maggiore data rate, ma a discapito di un minore range.

Power Consumption

I nodi hanno un approvvigionamento elettrico limitato (circa 1,2V). La vita del sensore dipende dal consumo elettrico:

- **Communication:** I sensori spendono la maggior parte dell'elettricità durante la comunicazione dei dati (sia in ricezione che in trasmissione). Siccome i sensori comunicano con pacchetti molto corti, è inefficiente spegnere e accendere il transceiver, siccome lo **startup-power** non è trascurabile.
- **Data Processing:** Processare i dati localmente è cruciale per minimizzare il consumo elettrico nella rete multi-hop.
- **Sensing**

Architettura



Application Layer

Nel livello applicazione ci sono diversi protocolli applicativi, i quali cercano di ovviare a diverse problematiche, come ad esempio:

Sensor Network Management Protocol (SMP)

È stato progettato circa 20 anni fa, e viene usato dagli amministratori di sistema per interagire con i sensori. Mediante questo protocollo è possibile svolgere diversi compiti:

- Muovere i nodi;
- Spegner e accendere i nodi;
- Eseguire query;
- Autenticazione e gestione di sicurezza;
- Sincronizzazione temporale.

È possibile eseguire diversi tipi di query:

1. **Query Storiche:** Analisi dei dati storici conservati in un'area di stoccaggio;
2. **One Time Queries:** Restituisce uno snapshot della rete;
3. **Present Queries:** monitora la rete nell'intervallo di tempo rispetto agli stessi parametri.

Sensor Query and Tasking Language (SCTL)

Questo protocollo è stato proposto nel 2001. È molto simile a SQL e stabilisce com'è possibile interfacciarsi con l'hardware dei sensori. Possiamo usarlo:

- **For Location Awareness:** “getTemperature()”, “turnOn()”;
- **For Communication:** “isNeighbor()”, “getPosition()”.

È possibile creare un blocco di gestione degli eventi per tre tipi di eventi (usando il comando UPON). Ogni evento è generato quando un nodo riceve un messaggio. Alcuni eventi possono essere triggerati periodicamente. Altri ancora allo scadere di un timer.

Questi eventi vengono definiti rispettivamente dalle parole chiave SCTL: **receive**, **every** ed **expire**,

Data Centric Query

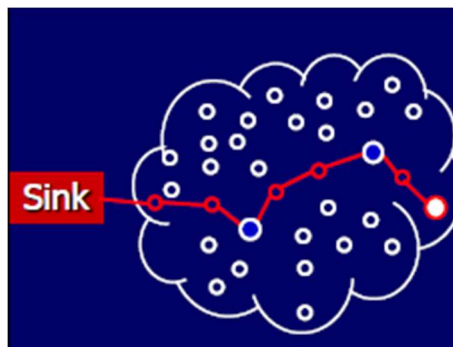
Le query sono distribuite nella rete che stiamo gestendo, e attivano solo i sensori che corrispondono con i dati specifici. Viene eseguita in broadcast e la query sarà eseguita dai sensori che la creano.

La **Data Fusion** va usata con attenzione (es. media di un parametro in una certa area).

Transport Layer

Il **Transport Layer** gestisce la comunicazione end-to-end tra i vari dispositivi. Fornisce servizi quali gestione del flusso, gestione degli errori e data segmentation.

Reliable Multi-Segment Transport (RMST)



Questo protocollo fornisce un trasferimento di pacchetti end-to-end affidabile, in cui i nodi RMST conservano i pacchetti nella loro cache. Quando un pacchetto non viene ricevuto, scade il timer “**watch dog**” e avviene l’invio ai nodi precedenti di un **NACK**.

Il primo nodo RMST, lungo il percorso, che conserva il pacchetto richiesto, lo ritrasmette.

Pump Slowly Fetch Quickly (PSFQ)

I pacchetti vengono emessi lentamente in rete e, nel caso di perdita di pacchetti, viene utilizzato un meccanismo di recupero hop-by-hop aggressivo.

Esegue un **flooding controllato** e richiede che ogni nodo intermedio crei e mantenga una **data cache** da usare per il recupero locale delle perdite.

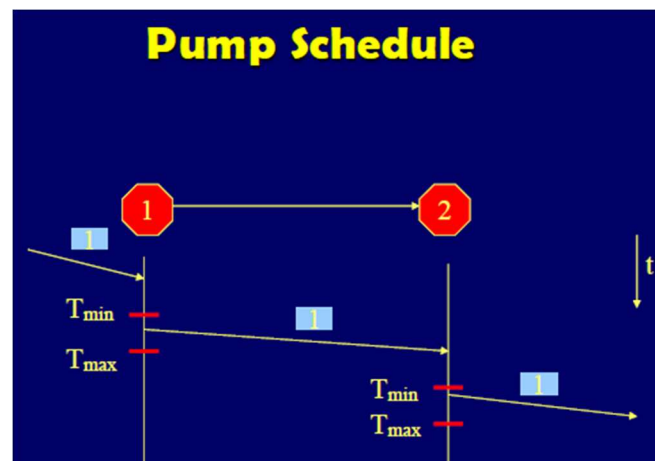
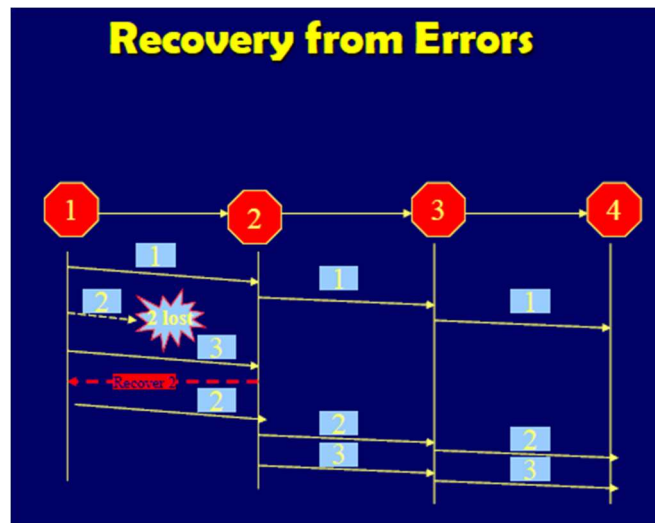
PSFQ ha tre funzioni:

1. Inoltro dei messaggi (**PUMP operation**);
2. Ripristino degli errori avviato dall’inoltro (**FETCH operation**);
3. Segnalazione selettiva dello stato (**REPORT operation**).

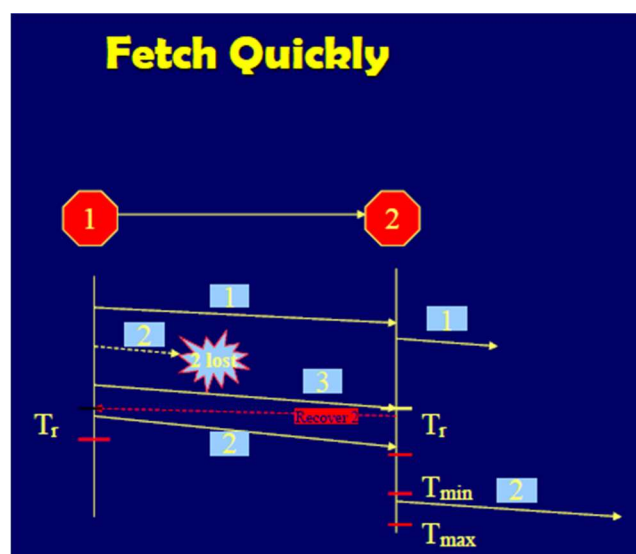
Quando un nodo riceve un pacchetto, ne controlla il contenuto con la sua cache locale ed elimina eventuali duplicati. Se il pacchetto è nuovo il suo campo **TTL** viene decrementato.

Se **TTL > 0** dopo il decremento e non ci sono gaps nel suo sequence number, il pacchetto viene preparato per essere inoltrato. L’invio dei pacchetti viene posticipato di un tempo randomico compreso tra **Tmin** e **Tmax**.

Quando un divario nel sequence number viene rilevato si passa in FETCH mode, richiedendo la ritrasmissione da tutti i nodi vicini.



Se non è duplicato, è in ordine e $TTL > 0$, viene recuperato dalla cache e schedulato per l'inoltro al tempo t con $T_{min} < t < T_{max}$



In conclusione, possiamo riassumere PSFQ come segue:

1. Leggero ed efficiente sul piano energetico;
2. Meccanismo semplice;
3. Scalabile e robusto;
4. Ha bisogno di essere testato per applicazioni ad ampia banda;
5. Limitazioni dovute alla dimensione della cache;
6. Applicabile strettamente solo per consegne garantite sensore-a-sensore;
7. Non possiede meccanismi di controllo della congestione.

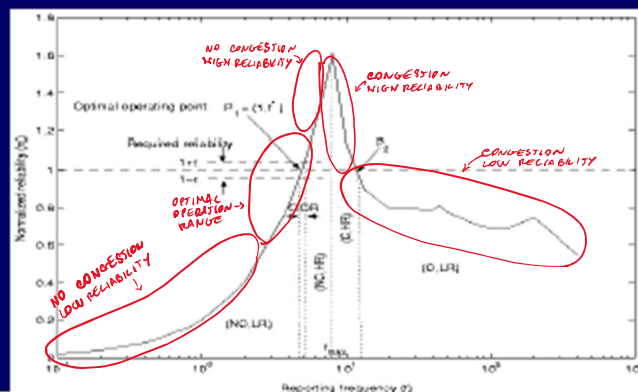
Event to Sink Reliability (ESRT)

Questo protocollo assicura un rilevamento degli eventi affidabile, parallelamente ad un'ottimizzazione del consumo delle risorse energetiche.

Tara dinamicamente il rateo di invio dei report dei nodi, basandosi sulle condizioni della rete per evitare congestioni e massimizzare l'efficienza della rete.

ESRT opera senza il bisogno di un ACK per-pacchetto, rendendolo scalabile per WSN di ampie dimensioni.

ESRT: Frequency Update



State	Frequency Update	Comments
(NC,LR)	$f_{i+1} = f_i / \eta_i$	Multiplicative increase, achieve desired reliability asap
(NC,HR)	$f_{i+1} = f_i(\eta_i + 1) / 2\eta_i$	Conservative decrease, no compromise on reliability
(C,HR)	$f_{i+1} = f_i / \eta_i$	Aggressive decrease to state (NC,HR)
(C,LR)	$f_{i+1} = f_i \eta_i$	Exponential decrease, relieve congestion asap
OOR	$f_{i+1} = f_i$	Unchanged

Routing Protocols

Perché necessitiamo di nuovi protocolli? Perché i sensori hanno una potenza di calcolo molto limitata e poca memoria, la topologia cambia molto di frequente, i nodi sono densamente distribuiti e i nodi hanno bisogno di stretta integrazione con i compiti di rilevamento.

I protocolli di routing possono essere suddivisi, sulla base dei loro comportamenti, in tre categorie:

1. **Data Centric Protocols:** Flooding, Gossiping, SPIN, SAR (Sequential Assignment Routing), Direct Diffusion, Rumor Routing, Gradient Based Routing, Constrained Anisotropic Diffused Routing, COUGAR, ACQUIRE;
2. **Hierarchical Protocols:** LEACH, TEEN (Treshold Sensitive Energy Efficient Sensor Network Protocol), APTEEN, PEGASIS, Energy Aware Scheme;
3. **Location Based Protocols:** MECN, SMECN (Small Minimum Energy Com Netw), GAF (Geographic Adaptive Fidelity), GEAR.

Flooding

Protocollo di routing molto semplice, nel quale ogni nodo inoltra, alla cieca, a tutti i suoi vicini i pacchetti ricevuti. Questo garantisce alta affidabilità ma causa anche eccessiva ridondanza, aumenta il consumo energetico e la congestione della rete.

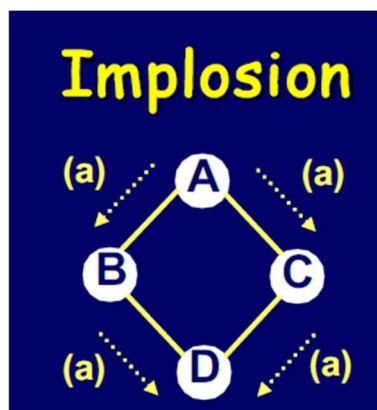
Mentre Flooding è utile in scenari in cui è richiesta massima affidabilità, è spesso ottimizzato usando tecniche come il **Gossiping** o il **Probabilistic Forwarding** per ridurre l'overhead.

Gossiping

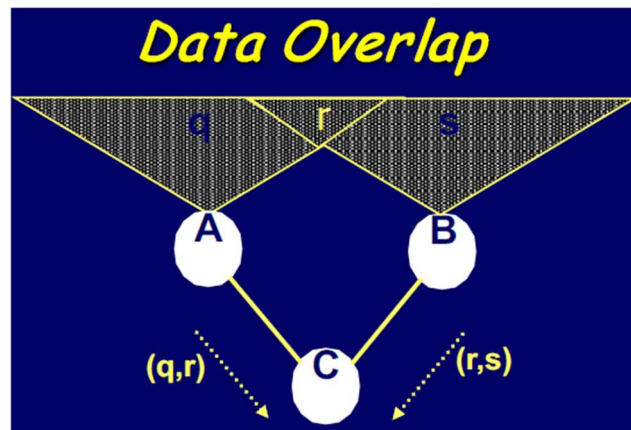
Migliora il protocollo Flooding riducendo le trasmissioni ridondanti. Un nodo sceglie randomicamente qualche altro nodo (almeno uno) a cui inviare il messaggio, i quali continuano il processo. Questo approccio aiuta ad abbassare il consumo energetico della rete e la sua congestione. Tuttavia, potrebbe introdurre ritardi rispetto al flooding tradizionale.

Problemi Flooding & Gossiping

Implosione (ridondanza): Nel flooding molteplici copie dello stesso messaggio sono inviate allo stesso nodo, causando inutile ridondanza, aumentando il consumo energetico e congestioni.



Data Overlap: I nodi potrebbero ricevere dati che coincidono o duplicati, da diverse sorgenti. Ciò porta ad inefficienze nell'elaborazione dei dati.



SPIN (Sensor Protocol for Information via Negotiation)

SPIN è una famiglia di protocolli basati sulla negoziazione e sulla disseminazione di informazioni. Si basa su due idee:

1. Operare efficientemente e conservare energia;
2. I nodi di una rete devono monitorare e adattarsi ai cambiamenti.

I sensori usano **metadati** per descrivere i dati che raccolgono. I nodi SPIN usano tre tipi di messaggi per comunicare:

1. **ADV (advertisement)**: quando un nodo ha dati da condividere, può pubblicizzare questa informazione inviando un messaggio **ADV** contenente dei metadati (è un pacchetto di piccole dimensioni);
2. **REQ (request for data)**: un nodo invia un messaggio **REQ** quando desidera ricevere dei dati (è un pacchetto di piccole dimensioni);
3. **DATA**: è un messaggio contenente i dati dei sensori con un **header** di metadati.

SPIN-1

SPIN-1 è un **protocollo con handshake a 3 stadi (ADV-REQ-DATA)**. Il protocollo parte quando un nodo ottiene nuovi dati che vuole vengano disseminati, inviando un messaggio **ADV** ai suoi vicini. Se il messaggio **ADV** è nuovo, essi rispondono inviando un messaggio **REQ**. Il protocollo si completa quando l'iniziatore della comunicazione risponde alla **REQ** con un messaggio **DATA**.

- **Vantaggi**: La forza di questo protocollo è la sua semplicità. Ogni nodo nella rete esegue piccole decisioni quando riceve nuovi dati. Il fatto che nessun'altra informazione riguardante la topologia è richiesta, significa che il protocollo SPIN-1 può essere eseguito in una rete non configurata con un piccolo cast. Se la topologia della rete cambia, è necessario che si performi un solo hop affinché la rete continui ad eseguire l'algoritmo.
- **Attenzione**: Ci sono dei piccoli appunti da tenere a mente:
 - Se il nodo B avesse già dei suoi dati, potrebbe aggregarli con i dati di un altro nodo e inviare **ADV**, dei dati aggregati, a tutti i suoi vicini.
 - I nodi non sono obbligati a rispondere a tutti i messaggi che gli vengono recapitati. Per esempio, se un neighbor non risponde con **REQ** al nodo. Questo accade se quel nodo possiede i dati che vengono pubblicizzati. Questo comporta un uso inutile di risorse.

SPIN-2

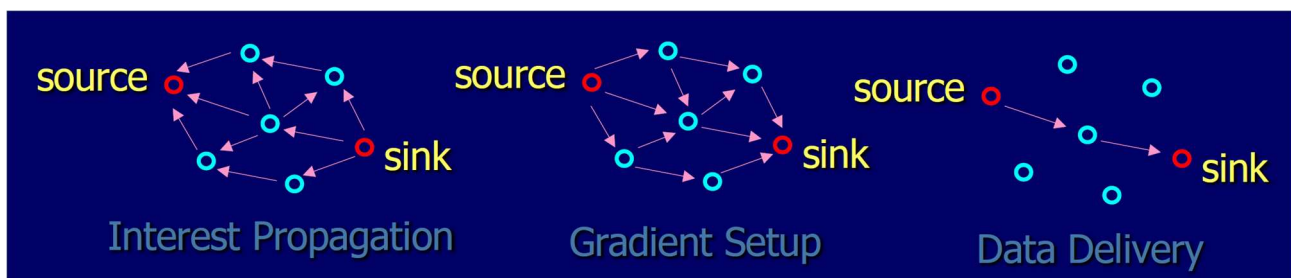
Il protocollo SPIN-2 aggiunge una semplice conservazione delle risorse rispetto al protocollo SPIN-1. Quando il consumo energetico di SPIN-2 è elevato, i suoi nodi comunicano usando lo stesso protocollo dei nodi di SPIN-1. Quando il consumo energetico si riduce, anche i nodi riducono la loro partecipazione al protocollo.

Questo **approccio conservativo** implica che se un nodo riceve nuovi dati, esso imita semplicemente il protocollo SPIN-1 se ritiene di avere sufficienti risorse. D'altro canto, se un nodo riceve ADV e non ha sufficienti risorse non risponde con un pacchetto REQ.

Questo approccio non previene che un nodo riceva e spenda energie su messaggi ADV e REQ quando supera il livello minimo di risorse disponibili.

Direct Diffusion

Questo protocollo instrada i dati in maniera efficiente, seguendo un approccio basato sull'interesse usando uno **schema di naming** (ai dati generati dai sensori viene assegnato un nome in base a coppie **attributo-valore**). Quando un task di rilevamento viene rilasciato, esso si sparge per la rete sotto forma di interesse (essenzialmente una richiesta per dei dati specifici).



Questo processo stabilisce dei **gradienti** che agiscono come percorsi consigliati che i dati devono seguire. Quando viene trovato un evento che coincide con l'interesse, i dati iniziano a fluire verso l'origine della richiesta seguendo molteplici percorsi. La rete rinforza il migliore percorso per assicurarsi una comunicazione efficiente e affidabile, riducendo trasmissioni non necessarie e di conseguenza risparmiando risorse.

Una riga della cache d'interesse è composta da molteplici campi:

- **Timestamp:** (dell'ultimo interesse ricevuto e coincidente) per scegliere il percorso più breve;
- **Gradient:** fino ad uno per vicino. È un link di trasmissione dal quale l'interesse è stato ricevuto.
 - **Data Rate:** richiesto da uno specifico vicino;
 - **Durata:** tempo di vita approssimativa dell'interesse.

Come creiamo la riga? Quando un nodo riceve un interesse, vede se esso risulta nella cache e nel caso crea una nuova riga. Se esiste una riga ma nessun gradiente per il mittente dell'interesse, il nodo aggiunge un gradiente con lo specifico valore (inserendo i campi timestamp e duration).

Regole per Interesse e Gradiente:

- La propagazione dell'interesse segue regole locali che definisce come le richieste dei dati si spargono nella rete. Il meccanismo più semplice è il **flooding**. Altri meccanismi più sofisticati possono essere **directional propagation** (influenzando le informazioni nella cache). Se sono stati ricevuti di recente dei report sospetti di uno specifico vicino, si potrebbe prioritizzare quello specifico nodo.
- I gradienti sono stabiliti per guidare i dati indietro verso il richiedente. Una regola comune è quella di assegnare un gradiente più alto al primo vicino che invia l'interesse. Strategie alternative potrebbero prioritizzare gradienti basati su **livelli di risorse rimanenti**;

Il protocollo ha i seguenti:

- **Vantaggi:**
 - **Data Centrico:** non c'è bisogno di meccanismi di indirizzamento dei nodi;
 - Ogni nodo può eseguire **aggregation, logging and sensing**;
 - **Energy Efficient (on demand)**, non c'è neanche necessità di mantenere la topologia di rete;
- **Svantaggi:**
 - In generale non è applicabile, è basato su un modello query-driven;
 - Non è ottimale per applicazioni che necessitano costantemente di invio dei dati;
 - Schemi di naming definiti a priori.

LEACH (Low Energy Adaptive Clustering Hierachy)

Questo protocollo opera **selezionando randomicamente i coordinatori dei clusters**, i quali aggregano i dati dai nodi circostanti riducendo in questo modo l'overhead di comunicazione.

Leach utilizza un meccanismo di formazione dei clusters round-based, dove in ogni turno un nodo viene selezionato diventare il coordinatore del cluster. Esso si compone di due fasi:

1. **Setup:** I clusters e i loro coordinatori sono selezionati;
2. **Steady-State:** I dati sono raccolti e trasmessi.

Per minimizzare le comunicazioni dirette con la stazione centrale, i coordinatori eseguono **data aggregation** e **data compression** prima di inoltrare le informazioni.

Nonostante la sua efficienza, Leach ha delle limitazioni:

- Consumo energetico irregolare dei coordinatori dei clusters;
- Mancanza di supporto alla comunicazione multi-hop.

Link Layer

Il Link Layer è il secondo livello della pila ISO-OSI ed è responsabile dell'inoltro e dell'individuazione degli errori nelle trasmissioni. Esso include **MAC (Medium Access Control)** e **LLC (Logic Link Control)**. Esso ha due approcci:

- **Protocolli MAC**, obiettivi:
 - **Collision Avoidance:** Se siamo in presenza di molte collisioni, il consumo energetico aumenta;
 - **Energy Efficiency:** Problema verticale;

- **Scalability:** Se aumenta la rete ha un calo delle performance.
- **Controllo d'errore.**

Alcune parole chiave:

- **Latency:** Dipende dall'applicazione e dal contesto. Dobbiamo cercare di non incrementarla, in modo da consentire lealtà nella rete (consentire a tutti i nodi di poter competere per il mezzo di comunicazione);
- **Throughput:** Fornito in modo da garantire un canale per un determinato lasso di tempo o per un certo numero di pacchetti;
- **Bandwidth Utilization:** Molto utile per dati reali, ma non altrettanto per dati controllati.

Principali Sorgenti di Spreco di Energia

- **Idle Listening:** Avviene quando un sensore cerca di mettersi in ascolto sul canale anche quando la comunicazione è indirizzata ad esso. Ciò avviene siccome vuole verificare se il canale è occupato;
- **Transmission;**
- **Receiver.**

Challenges nel MAC delle WSN

- **Architettura WSN:** In una WSN possiamo avere alta densità di nodi e quindi elevata probabilità di avere delle collisioni;
- **Risorse Energetiche Limitate:** Le performances di connettività e di rete soffrono dello spegnimento di un nodo. Accensioni e spegnimenti di un nodo consumano molta energia, quindi dobbiamo cercare di limitare questi cambi di stato;
- **Capacità di Elaborazione e Memoria Limitate:** Algoritmi complessi non possono essere implementati a causa delle limitate capacità del microcontrollore. Controllo centralizzato (usando nodi speciali per inviare/trasmettere dati) o localizzato è limitato. Vengono usati localmente dei protocolli distribuiti. È richiesta ottimizzazione cross-layer (valutare lo stato del canale di comunicazione);
- **Dimensione dei Pacchetti Limitata:** In una WSN, i nodi inviano pacchetti molto piccoli, e non usano un ID univoco siccome non è pratico (spazio limitato negli header, uso degli id locali);
- **Encoders/Decoders Economici:** Prevenire encoding/decoding complessi per tenere basso il costo dei sensori;
- **Cristalli di Clock Inaccurati:** Prevenire l'uso di cristalli complessi può ridurre il costo dei sensori, ma ciò comporta problematiche di sincronizzazione. Approcci basati su TDMA non sono pratici.

IEEE 802.11

Inizialmente sviluppato per essere utilizzato nelle WLAN. Utilizza un **4-ways handshake** (RTS/CTS/DATA/ACK). Le sue problematiche sono:

- Tempi di ascolto non sono efficienti sul piano energetico;
- Elevato consumo energetico a causa dell'**idle listening**;
- Sotto carichi importanti, RTS/CTS implica un considerevole overhead di messaggi.

S-Mac

Il protocollo **S-Mac** riduce il consumo energetico nelle WSN affrontando idle listening, collisioni e overheard come segue:

1. **Listen e Sleep periodiche:** I nodi seguono una **sleep schedule** per alternare il loro stato tra **active** e **sleep**.
2. **Collision and Overhearing Avoidance:** Previene consumi di corrente non necessari;
3. **Message Passing:** Aumenta l'efficienza del trasferimento dati.
4. **Duty Cycle:** Tempo in cui il nodo è attivo;
5. **Fragmentation Management.**

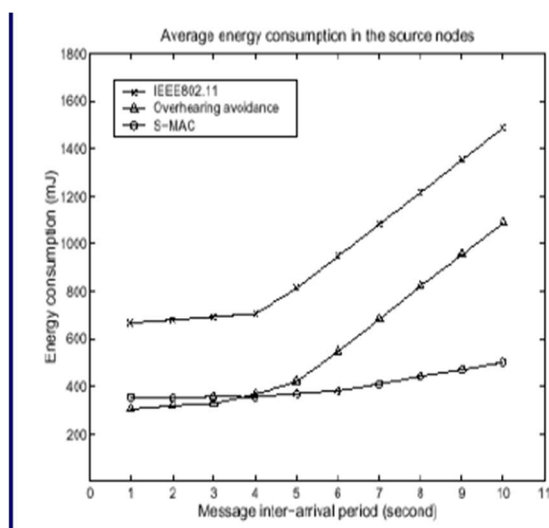
Schema Base

Ogni nodo va in fase di sleep per un tempo randomico, poi si sveglia e controlla se qualche nodo vuole comunicare con lui. Durante la fase di sleep, il nodo spegne la sua radio e imposta un e si imposta una sveglia per potersi svegliare in seguito.

I nodi si condividono le loro schedule inviandosele a vicenda in broadcast (ognuno ai suoi immediati vicini). Questo garantisce che tutti i nodi vicini possono parlarsi a vicenda anche se hanno schedule differenti.

Il problema di questo schema consiste nell'incremento della sua latenza dovuta alle sleep periodiche dei nodi.

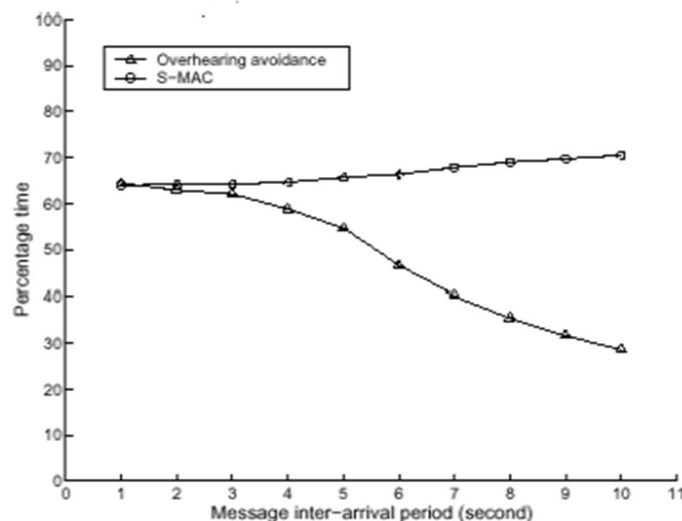
Consumo Energetico dalle Sorgenti A e B



Il traffico si appesantisce quando il tempo di arrivo inter-messaggio è inferiore a quattro secondi. In questo caso 802.11 usa il doppio dell'energia. S-MAC ottiene risparmio energetico principalmente evirando l'overhearing e trasmettendo in maniera efficace messaggi lunghi.

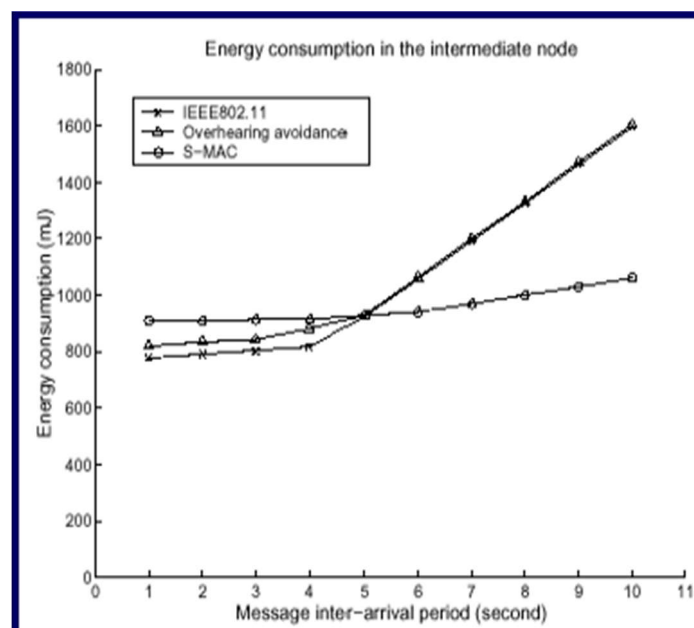
Quando il tempo di arrivo inter-messaggio è maggiore di quattro secondi, il carico di traffico si alleggerisce. Il consumo energetico di S-MAC è relativamente indipendente dai pattern del traffico.

Tempo in Fase di Sleep



È interessante il fatto che S-MAC aggiusta il tempo di sleep in base ai pattern del traffico. Quando il traffico è poco, il nodo resta più tempo in modalità di sleep (c'è un limite imposto dal duty cycle del nodo). Quando il traffico aumenta, i nodi hanno meno opportunità di ricevere sleep periodiche, spendendo quindi più tempo in trasmissione.

Consumo Energetico nei Nodi Intermedi



È migliore di 802.11 anche nel caso con traffico leggero. Nel caso con traffico pesante invece consuma leggermente di più di 802.11.

Una ragione è che S-MAC ha un overhead di sincronizzazione, siccome invia e riceve pacchetti sync.

Un'altra motivazione è che S-MAC introduce una latenza maggiore e in realtà impiega più tempo per trasferire la stessa quantità di dati.

Architettura e Modello di Correlazione nelle WSN

L'ordine della Georgia Tech sui protocolli di livello trasporto event-based fece una proposta a livello MAC. Nello specifico propose che il livello MAC potesse sfruttare la correlazione spaziale per poter ridurre il numero di trasmissioni tra nodi con alta correlazione. Il sink è interessato nella stima dell'evento S . Ogni nodo n_i , l'evento $S_i[n]$ ed è inviato codificato al sink. Il sink lo decodifica e ottiene la stima $\hat{S}$. Nella formula sono specificati gli encoders E e i decoders D . L'obiettivo principale è quello di stimare l'evento da osservazioni differenti con una certa affidabilità, e ciò è scelto come distorsione tra S e $\hat{S}$. Ciò si ottiene da:

$$D = E[d(S, \hat{S})]$$

Ovvero la media della distanza tra il valore reale e la sua stima.

L'idea base è: Siccome esiste una correlazione, è possibile trovare n nodi che si trovano vicino alla sorgente dell'evento ma anche lontano gli uni dagli altri.

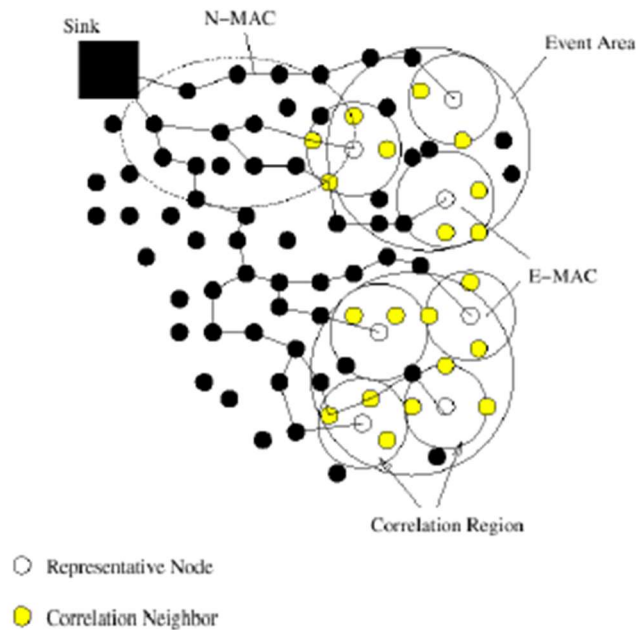
- **Regione di Correlazione di un nodo n :** Regione di raggio r_{corr} centrata sul nodo n ;
- **Vicini di Correlazione di un nodo n :** Nodi all'interno della regione di correlazione del nodo n .

C-Mac (Collaborative MAC)

Se un n_i trasmette dati, tutti i suoi vicini correlati possiedono informazioni ridondanti. L'idea alla base di questo protocollo è il filtraggio delle trasmissioni di dati ridondanti e prioritizzare i dati filtrati.

Componenti Principali

1. **INS (Iterative Node Selection) Algorithm:** INS prova a trovare la posizione ideale dei nodi rappresentativi così che le distorsioni richieste possono essere mantenute al sink. Nel paper sembra che se un nodo n_i trasmette dei dati, allora i nodi nella regione collaborativa non sono obbligati ad inoltrare i dati per ridurre la ridondanza;
2. **Event MAC:** Filtra dati correlati. Viene usato quando un nodo vuole trasmettere;
3. **Network MAC:** Fornisce prioritizzazione dei route-thru packets.



Nell'area dell'evento si usa E-MAC

E-Mac (Event MAC)

Questo protocollo funziona come segue:

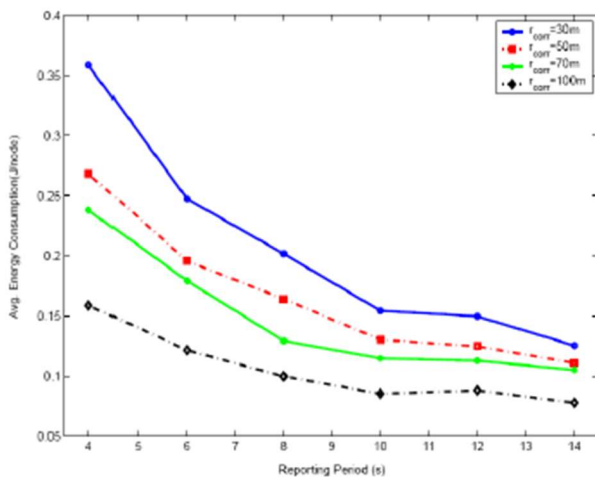
First Connection Phase (FCP): I nodi si contendono il canale usando la struttura 802.11 (RTS/CTS/ACK/DATA) per la prima volta. I nodi vincenti diventano i rappresentanti. Essi impostano il campo **FH** e tentano di ottenere il mezzo per la comunicazione.

Nell'E-MAC viene usato r_{corr} per determinare se i nodi vicini sono vicini correlati. Dopo che un nodo n_i . Per conservare energia, essi vanno in SSS (**Suspicious Sleep State**). In questo stato i nodi si svegliano ad intervalli randomici per vedere se stanno ricevendo traffico e in seguito tutti i nodi entrano in **FCP** per mantenere una condivisione del carico equa.

N-Mac (Network MAC)

Quando un nodo intermedio riceve un pacchetto dati, esso esegue N-MAC per inoltrare il pacchetto all'hop successivo. Quando un nodo riceve un RTS riguardante un pacchetto route-thru durante l'SSS, esso switcha allo stato di ricezione. Per dare una priorità maggiore, il canale è ascoltato per unità FIFS (invece di DIFS).

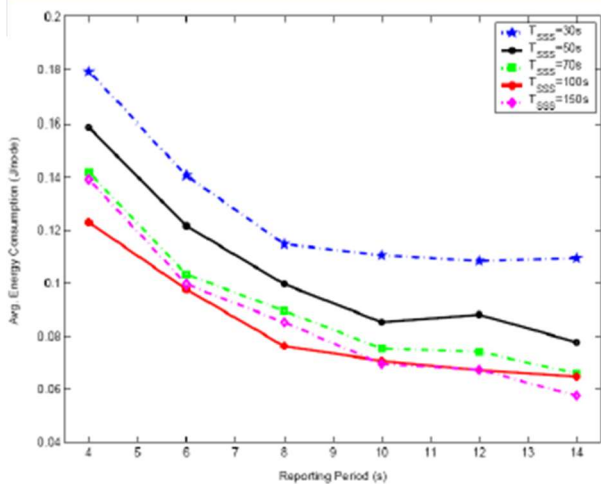
Performance



Possiamo vedere come, aumentando il periodo di report diminuisce il consumo energetico.

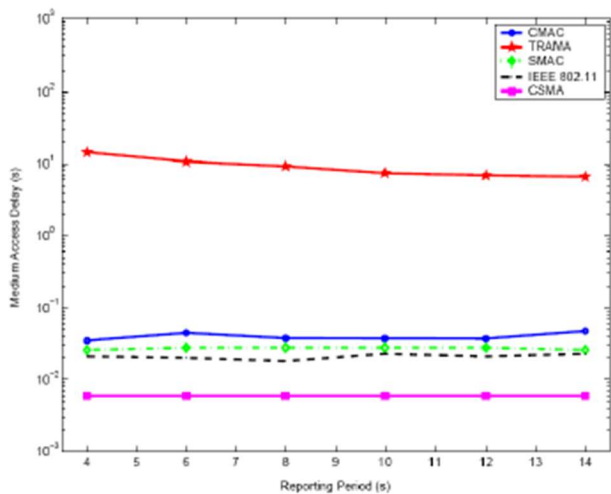
Aumentando r_{corr} diminuisce il numero di nodi rappresentativi. È quindi possibile risparmiare il 56% dell'energia impostando r_{corr} da 30 a 100.

Avg. energy consumption/node

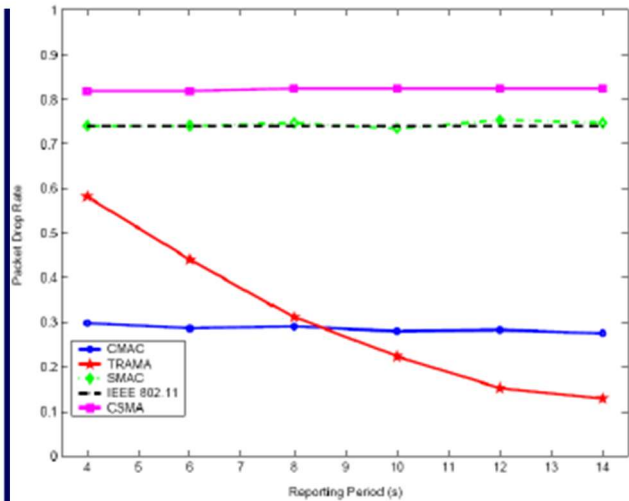


Diminuendo t_{SSS} da 150s a 30s è possibile che il consumo energetico aumenta del 60% ma dopo 8 cicli di report, il consumo energetico diventa uguale per tutte le trasmissioni a causa dei periodi di contesa più frequenti.

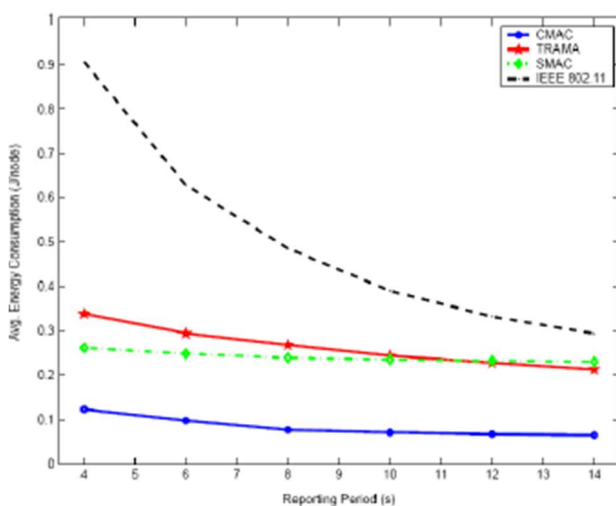
Avg. energy consumption/node for different T_{SSS}



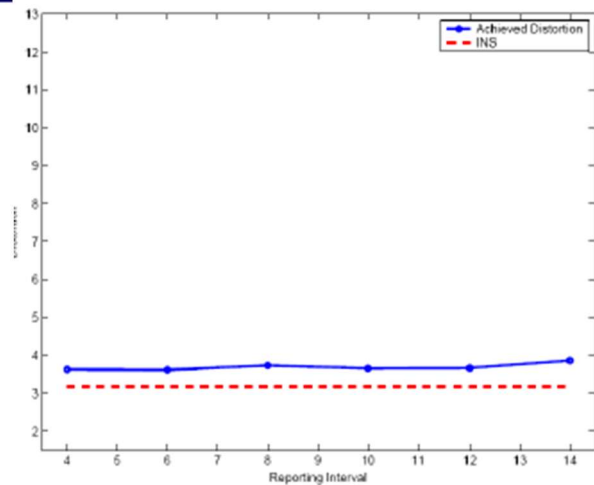
Medium Access Delay



Packet Drop Rate



Avg. Energy Consumption



Achieved Distortion vs. INS

Conclusioni

La **Correlazione Spaziale** nelle WSN è sfruttata dal livello MAC per ridurre il consumo energetico e per ottenere bassa latenza nel rilevamento degli eventi. Il lavoro del paper mostra che, sfruttando la correlazione spaziale, la trasmissione di nodi ridondanti può essere controllata.

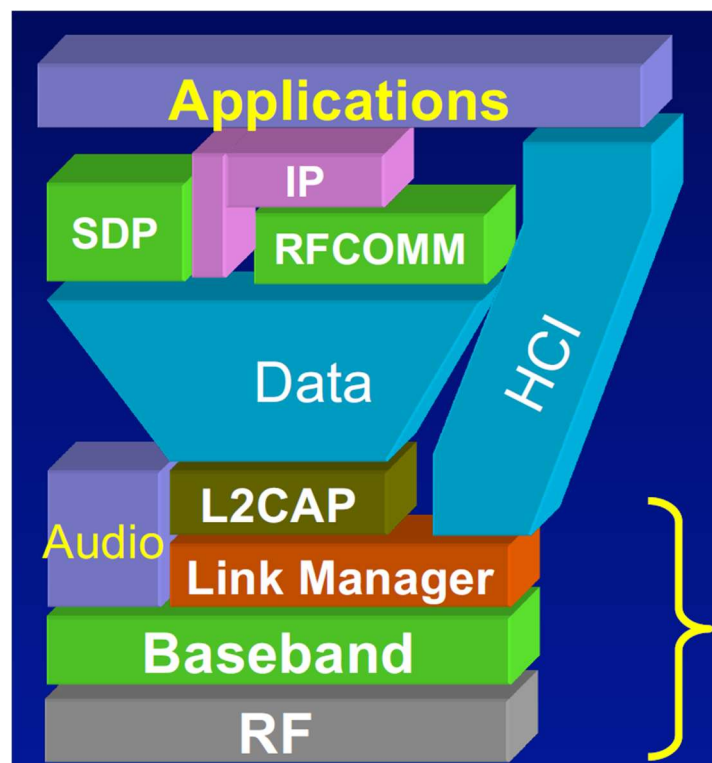
Bluetooth

Overview

La prima idea di bluetooth era di rimpiazzare le connessioni ethernet cablate con un rate di 1Mbps in un range di 10 metri. Grazie a diversi aggiornamenti, bluetooth ha raggiunto un rate di 20Mbps a 2.4GHz o 54Mbps a 5GHz. Il range varia da 25 a 100 metri.

Pila Protocolare

Nel protocollo sono presenti delle fette verticali che ci permettono di operare tra il livello applicativo e i livelli inferiori per creare applicativi più complessi.



Profiles

Un profilo bluetooth è un insieme di regole o standard che definiscono come i dispositivi bluetooth comunicano per una funzione specifica. Ogni profilo assicura che dispositivi da produttori diversi possano funzionare insieme se supportano lo stesso profilo.

I profili influenzano l'interoperabilità, siccome essa dipende da:

- Se entrambi i dispositivi supportano lo stesso profilo;
- Se la versione di bluetooth supporta quel profilo;
- Quanto strettamente quel profilo è stato implementato.

Due dispositivi con profili diversi non possono comunicare.

Radio Specification

Radio Frequency (RF – livello fisico)

Inviare e ricevere flussi di bit modulati.

Baseband

Definisce timing, framing e flow control.

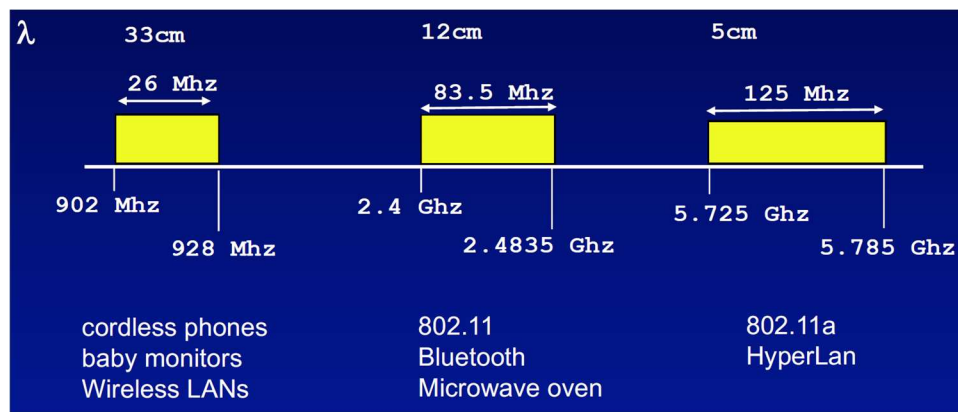
Link Manager

Gestisce gli stati delle connessioni, imponendo lealtà tra i vari slaves: assicura che non ci sono slaves che occupano la comunicazione per troppo tempo.

Logical Link (Control & Adaptation)

Gestisce il multiplexing di protocolli di livelli superiori, la scoperta di nuovi dispositivi e la QoS.

Radio Spectrum



Bluetooth usa

- FHSS (Frequency Hopping Spread Spectrum). Nel caso di 2.4 GHz $\rightarrow k = [0,78] 1600$ hops/s;
- GFSK (Gaussian Hopping Spread Spectrum) per offrire maggiore fluidità nel cambio di frequenze con un rate di simboli pari a 1MBps

Differenze con FSK (filtro gaussiano)

- Un modulatore FSK è lo stesso di GFSK, ma senza il filtro gaussiano. In questo modulatore, gli impulsi di baseband $[-1, 1]$ vanno direttamente in output.
- Invece un modulatore GFSK usa un filtro gaussiano, in modo da ridurre l'ampiezza dello spettro (pulse shaping). Se usassimo -1 per "fc-fd", e 1 per "fc+fd", quando saltiamo da 1 a -1 o viceversa, la forma d'onda modulata cambia rapidamente, che introduce un ampio spettro out-of-bound. $[-1, -0.98, -0.96, \dots, 0.96, 0.98, 1]$

Protocols

Middleware Protocol Group

Il middleware fornisce servizi di supporto come device discovery e gestione della sessione, consentendo per un setup della comunicazione tra i vari dispositivi molto fluido. Agisce come un

ponte tra il nucleo della pila BT e i profili applicativi, consentendo interoperabilità (SDP, TCP/IP, RFCOMM).

Master-Slave

- **Master:** il dispositivo nelle piconets, la cui sequenza di clock e di hopping sono usate per sincronizzare gli altri dispositivi nella rete. Esegue anche procedure di paging e stabilisce le connessioni.
- **Slave:** i dispositivi che vengono sincronizzati con il master. Agli slaves sono temporaneamente assegnati dei member address da 3 bit, in modo da ridurre il numero di bit necessari per l'indirizzamento (al più 7 slaves, quindi 3 bit 000 → 111).

Piconets

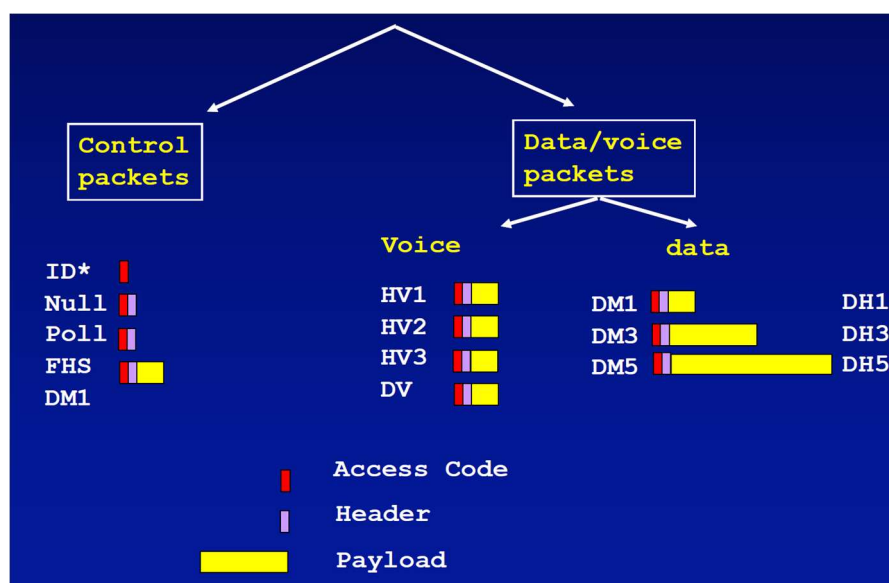
È la rete formata da un master e almeno uno slave. Ogni piconet è definita da un canale e da un pattern di hopping differenti.

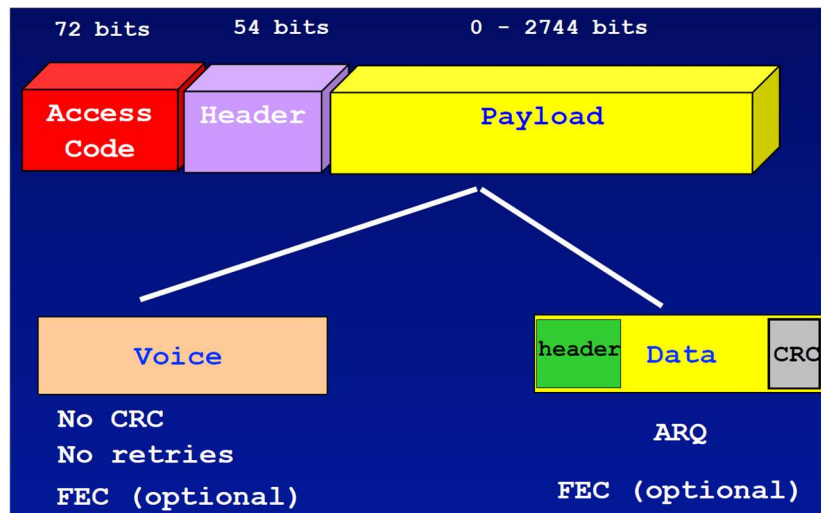
Physical Link Types

I physical link types definiscono come i dati vengono trasmessi nell'etere tra i vari dispositivi. Questi links agiscono al livello della baseband e sono essenziali per stabilire e mantenere le comunicazioni.

- **Synchronous Connection Oriented (SCO):** Usa una comunicazione full-duplex punto-punto tra il master e gli slaves, che viene instaurata una sola volta dal master. Vengono tipicamente usate per trasmettere la voce, in modo da garantire continuità. Il master riserva degli slot usati per SCO/link per mantenere informazioni sensibili al tempo.
- **Asynchronous Connection Link (ACL):** È un link momentaneo tra il master e lo slave, con nessuno slot riservato. È una connessione punto-multipunto con la possibilità di stabilire link simmetrici e no.

Pacchetti





Forward Error Control (FEC): È un codice che introduce ridondanza e alcune proprietà in aggiunta ai dati inviati per essere in grado, come ricevente, di recuperare la sequenza di bit senza dover davvero testare la trasmissione. Usa un encoding con fattore di 1/3, ciò significa che si invia un bit di dati utili ogni 2 di dati del canale.

Access Code

Lo scopo di questa sezione del pacchetto è sincronizzazione, identificazione e segnalazione.

- **Channel Access Code (CAC):** Identifica una piconet;
- **Device AC (DAC):** Viene usato per segnalare procedure come e paging di risposta;
- **Inquiry AC (IAC):**
 - **General IAC:** in comune per tutti i dispositivi;
 - **Dedicated IAC:** dedicato ad un gruppo di dispositivi BT che condividono caratteristiche comuni.

Packet Header

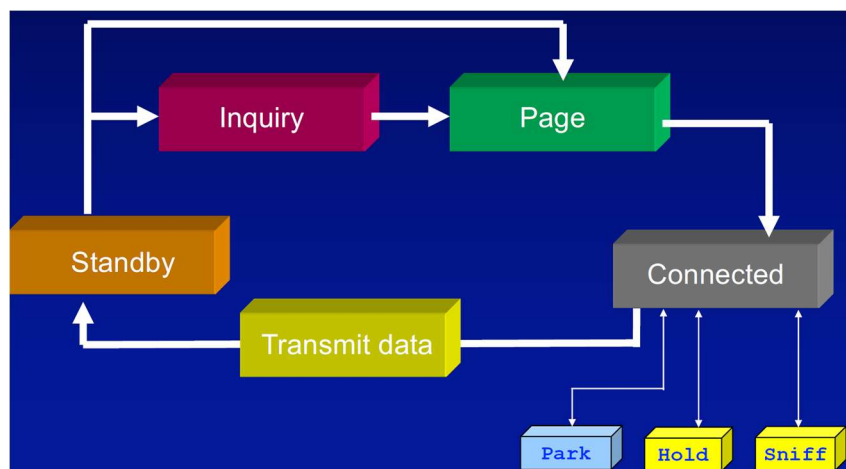
- **Indirizzamento:** 3 bit, massimo 7 slaves attivi;
- **Tipologia di Pacchetto:** 4 bit, 16 tipologie di pacchetto;
- **Controllo di Flusso:** 1 bit, controlla se il pacchetto è stato già inviato o se è una ritrasmissione;
- **ARQ:** 1 bit, i pacchetti broadcast non hanno ACK;
- **Sequencing:** 1 bit, per filtrare i pacchetti ritrasmessi;
- **HEC:** 8 bit, per verificare l'integrità del pacchetto.

In totale sono 18 bit, raggiunge 54 bit se codificato con FEC 1/3.

Connection States

1. **Inquiry Scan:** un dispositivo che vuole essere scoperto entra periodicamente questa modalità e si mette in ascolto di **pacchetti di ricerca**;
2. **Inquiry:** Il dispositivo invia un pacchetto di ricerca indirizzato a GIAC o a DIAC. La trasmissione viene ripetuta sulla sequenza di hop del richiedente;

3. **Inquiry Response:** quando un messaggio di ricerca viene ricevuto in stato di scan, un pacchetto di risposta (FHS) contenente l'indirizzo dev'essere inviato dopo un numero randomico di slots.
1. **Page:** Il master utilizza le informazioni di clock relative allo slave da contattare per determinare in quale punto della sequenza di hop lo slave potrebbe essere in ascolto in modalità page scan. Il master invia un **messaggio di paging**.
2. **Page Scan:** Il sottostato del page scan può essere acceduto dallo slave dallo stato di standby o dallo strato di connessione. Esso ascolta i pacchetti indirizzati al suo DAC (Device Access Code);
3. **Page Response:** All'arrivo del messaggio di page, lo slave entra nella **slave page response state**. Esso risponde con una page response, che consiste del suo pacchetto ID contenente il suo DAC.



Frequency Hopping Spread Spectrum (FHSS)

Il canale BT è rappresentato da una sequenza pseudo-randomica di hops in tutte le 79 frequenze RF.

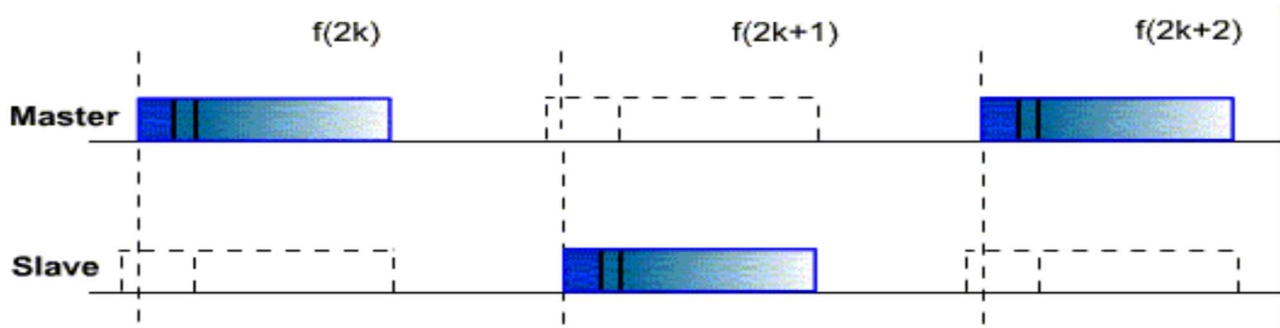
È dimostrato matematicamente che ciò garantisce maggiore robustezza.



Time Division Duplex Scheme (TDD)

Il canale BT è suddiviso in slot consecutivi (ognuno di $625\ \mu\text{s}$) con un pacchetto per slot.

C'è una rigida alternanza di slot tra master e slaves. Il master può inviare pacchetti solo negli slots pari, mentre lo slave al master solo negli slots dispari.



Indirizzamento

- **Indirizzo dei dispositivi BT (BT-ADDR):** 48 bit IEEE MAC address;
- **Active Member Address (AM-ADDR):** 3 bit active slave address;
- **Parked Member Address (PM-ADDR):** 8 bit parked slave address.

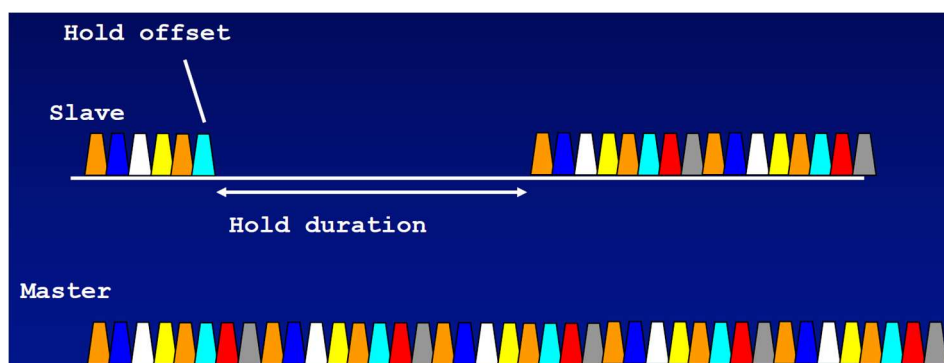
Piconet Management

Consiste in connettere e disconnettere gli slaves, stabilire SCO links e gestire le modalità di potenza (Hold, Sniff, Park).

Hold

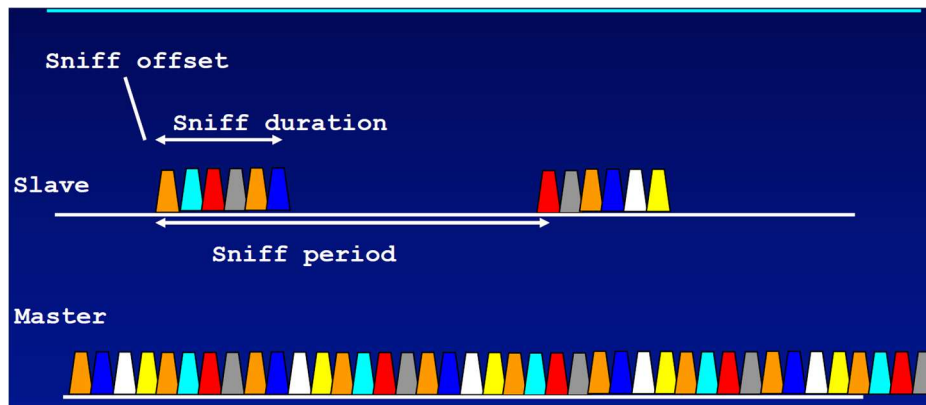
Lo slave temporaneamente (per un numero di secondi pari a T_{hold}) non supporta pacchetti ACL sul canale, ma solo pacchetti SCO.

In questa modalità, uno slave può diventare un master un'un'altra piconet dentro la scatternet. Lo slave **mantiene il suo AM-ADDR**.



Sniff

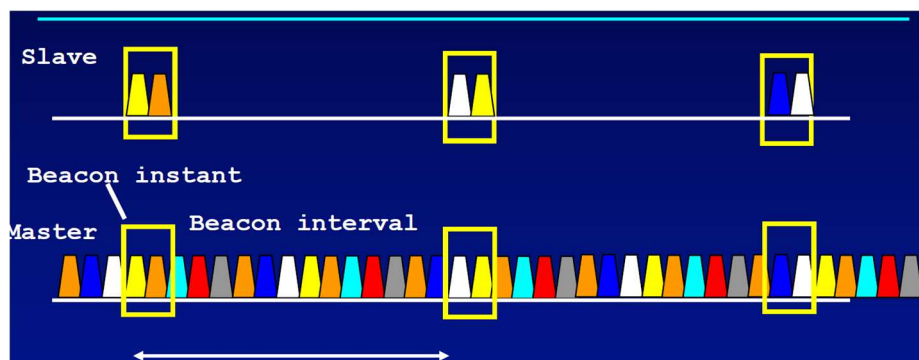
In questa modalità, l'attività di ascolto dello stato viene ridotta. Sta in ascolto per trasmissioni ad intervalli periodici lunghi T_{sniff} allo slot di offset a D_{sniff} per a N_{sniff} volte.



Park

Questa è una modalità a consumo molto ridotto, lo slave resta sincronizzato al canale ma **perde il suo AM-ADDR**.

Resta regolarmente in ascolto per segnali di beacon ad intervalli definiti dal master.



L2CAP



Logical Link Control and Adaptation Protocol

fornisce multiplexing di protocollo: Segmentazione e Riasssemblaggio, negoziazione del QoS.

Perché la baseband non è sufficiente?

La dimensione dei pacchetti di baseband è molto ridotta (17 – 339 bit). Non c'è un campo Protocol-ID nell'header.

Quindi necessitiamo di un livello di incapsulamento multiprotocollo.

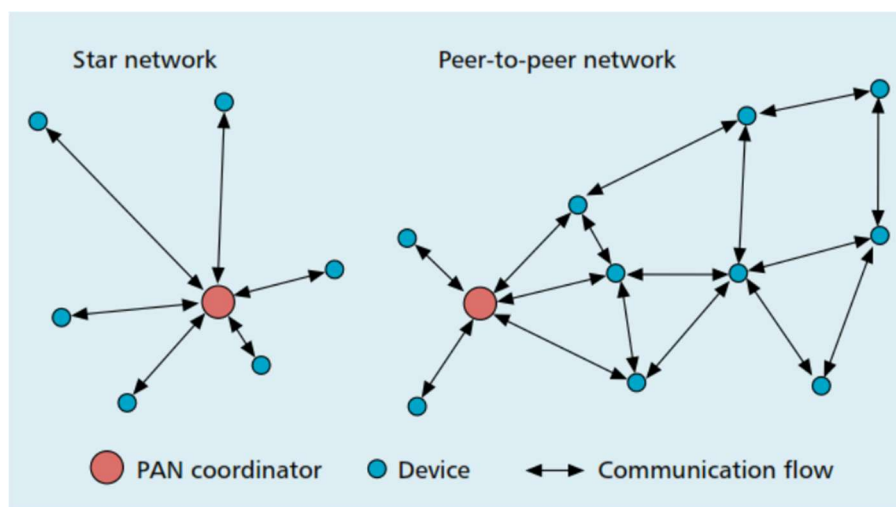
IEEE 802.15.4

IEEE 802.15.4 è progettato per essere utile in un ampio spettro di applicazioni (industriali, pubbliche, sensoristiche ecc.). Tuttavia, una delle opportunità applicative più ampie è l'**Home Automation and Networking**.

Network Layer

Lo standard 802.15.4 copre soltanto i livelli inferiori della rete, incluse porzioni del Data Link Layer (DLL). Protocolli a più alto livello sono discrezione delle specifiche applicazioni impiegate (Network Layer). I livelli di rete costruiti su quello standard dovrebbero essere capaci di auto-organizzazione/mantenimento, per minimizzare il costo totale per l'utente finale.

La bozza dello standard 802.15.4 supporta alcune topologie di rete: **STELLA** e **PEER-TO-PEER**.

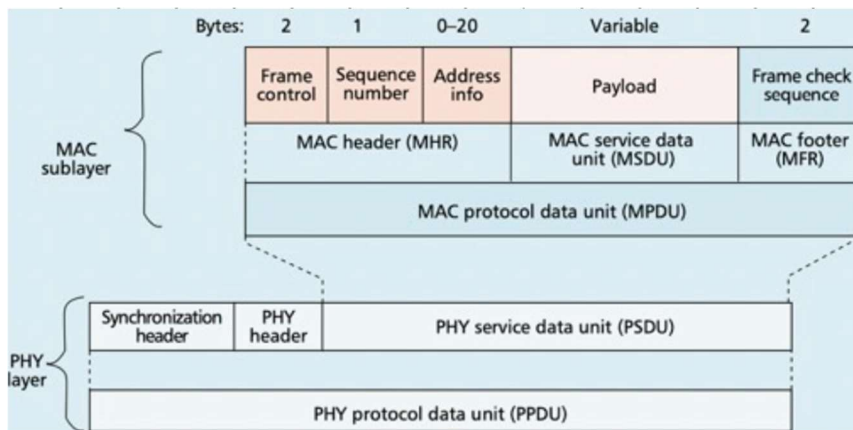


La topologia è una scelta progettuale. Alcune applicazioni potrebbero richiedere connessioni a stella a bassa latenza, mentre altre potrebbero richiedere ampia copertura di reti P2P.

Data Link Layer

Il progetto IEEE 802 suddivide il Data Link Layer in due sottolivelli: sottolivello **MAC** e sottolivello **Logical Link Control (LLC)**. IEEE 802.15.4 fornisce funzionalità nel sottolivello MAC come: associazione e dissociazione, acknowledged, frame delivery ecc.

La struttura del **frame MAC** è molto flessibile.



Il frame MAC è chiamato **MAC Protocol Data Unit (MPDU)** ed è composto da:

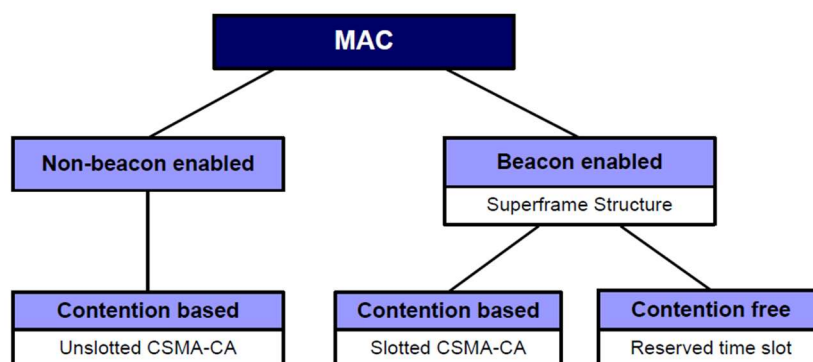
- **MAC Header (MHR)**
- **MAC Service Data Unit (MSDU)**
- **MAC Footer (MFR)**

Inoltre, è bene fare le seguenti definizioni:

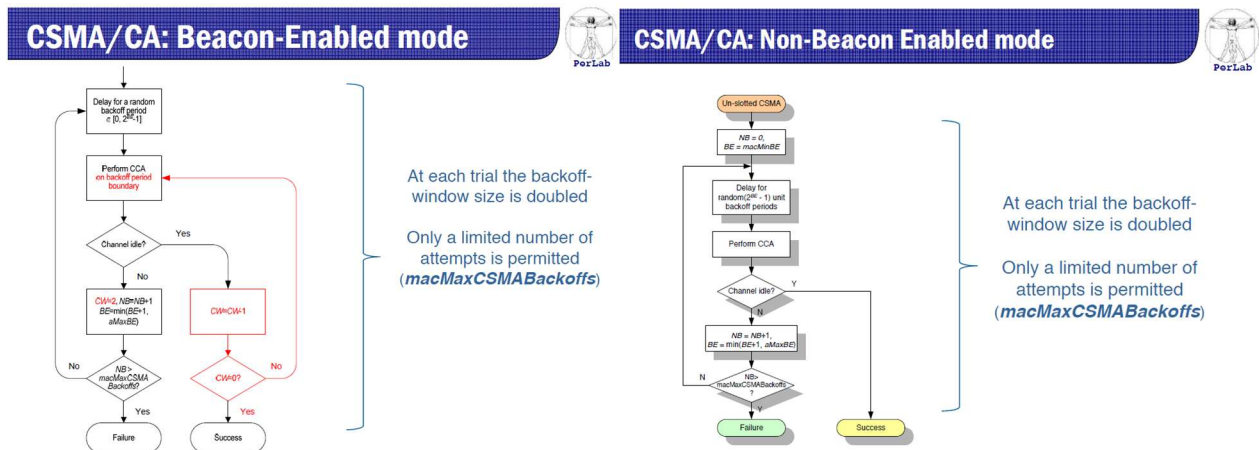
- **Frame control:** specifica come appare il resto del frame e cosa contiene;
- **Payload:** variabile ma non eccede i 127 Bytes;
- **Sequence number:** compara l'ACK frame con le precedenti trasmissioni;
- **Frame Check Sequence (FCS):** aiuta a verificare l'integrità del MAC frame.

802.15.4 ha 4 tipi di versi di frame:

- **Beacon;**
- **Data;**
- **ACK;**
- **Command;**

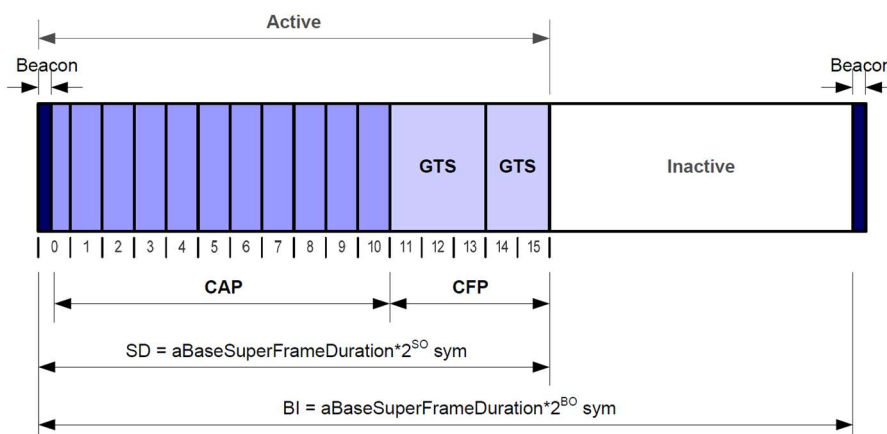


Ci sono due tipi di metodi di accesso al canale: **Beacon Enabled Mode / Non Beacon Enabled Mode**.



Superframe

È usato per applicazioni in cui è richiesta ampia banda e bassa latenza. Il coordinatore PAN assegna degli slot temporali ad ogni singolo dispositivo (**GTS-Guaranteed Time Slot**) per formare il CFP. Invece nel CAP i dispositivi competono per accedere al canale.

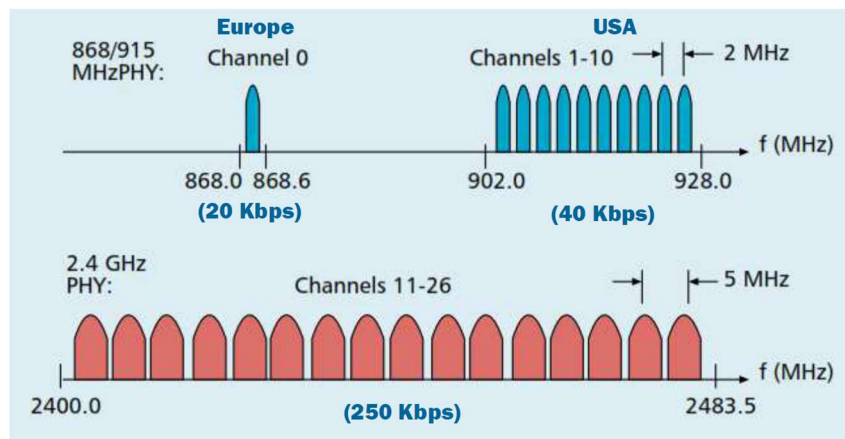


ACK Mechanism

È un meccanismo opzionale ed è differente per i due lati della comunicazione:

- **Destinatario:** Invia l'ACK in seguito alla ricezione corretta del data frame;
- **Mittente:** Ritrasmette in caso di mancata ricezione (corretta) dell'ACK entro il timeout. Ad ogni tentativo di ritrasmissione, la finestra di backoff viene reinizializzata. Sono consentite solo un certo numero di ritrasmissioni.

Channelization



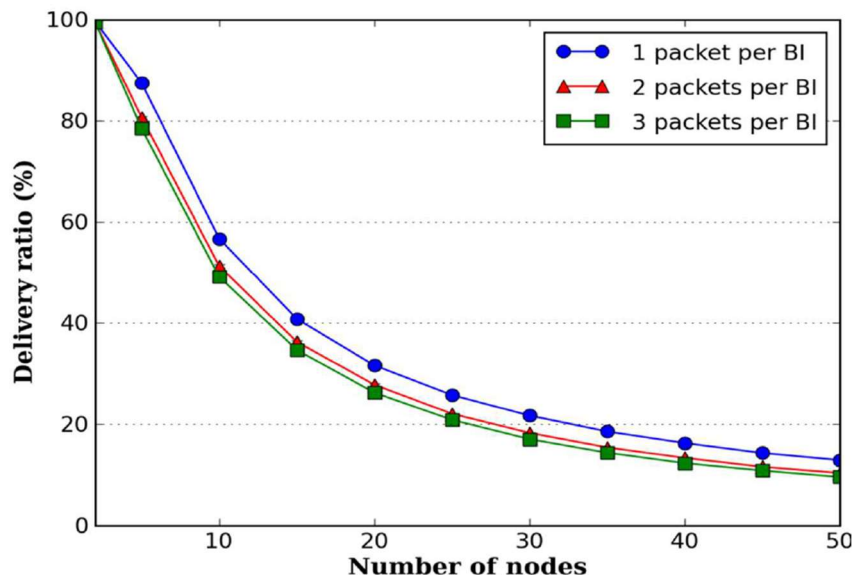
Sono disponibili 27 canali di frequenza nelle 3 bande.

Le frequenze 868/915 MHz supportano 11 canali. A causa del supporto regionale per queste 2 frequenze, è improbabile che una singola rete possa usare tutti e 11 i canali.

La frequenza a 2.4 GHz invece supporta 16 canali, con un ampio margine tra un canale e l'altro (5 MHz) con lo scopo di filtrare TX/RX.

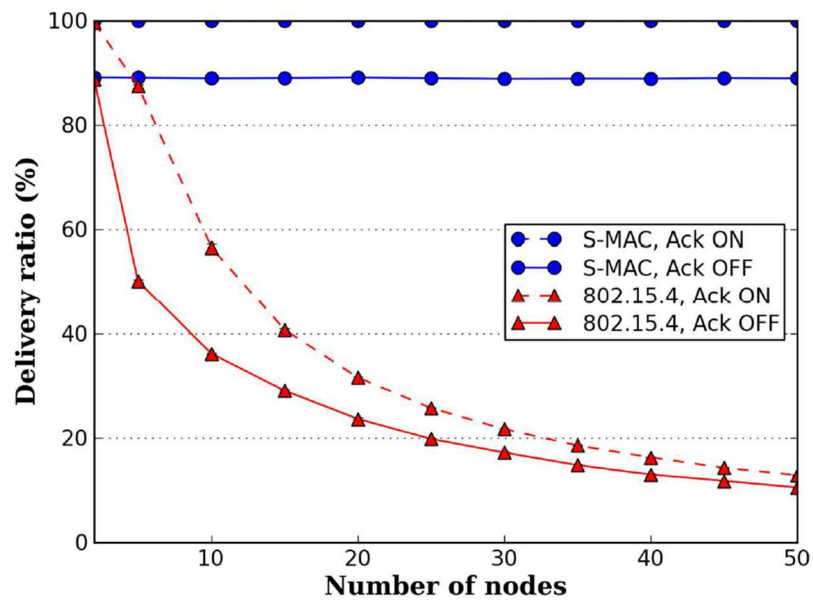
Limiti

- **Inaffidabilità del MAC;**
- **Non c'è garanzia dell'ampiezza di banda** (A meno che non viene usato GTS. Quest'ultimo fornisce un servizio limitato a sette slots);
- **Nessuna tecnica di frequency-hopping built-in.**



802.15.4 vs S-MAC

Smac è più scalabile



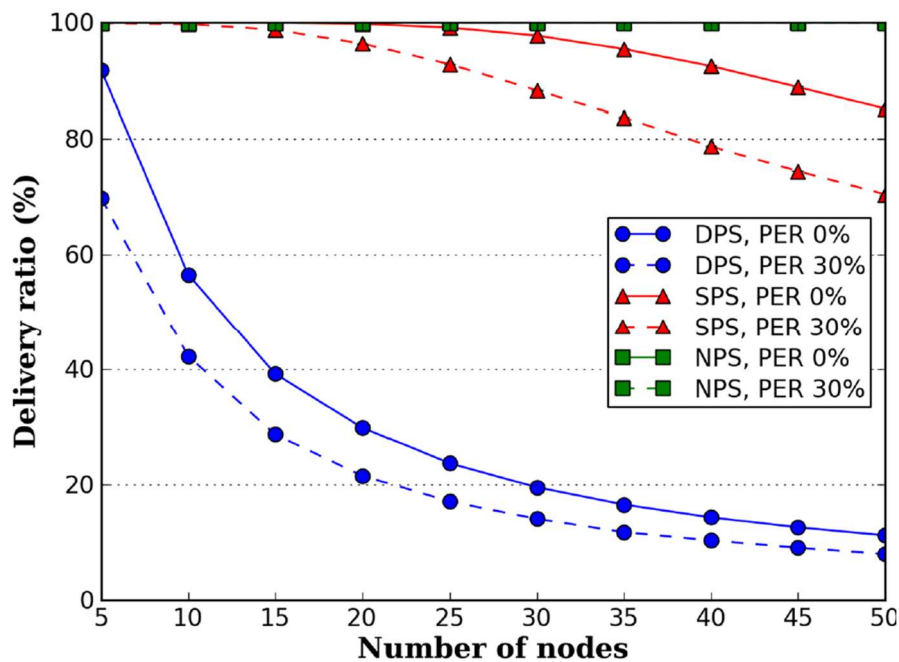
Parametri CSMA/CA

Parameter	2003 release	2006 release	Notes
<i>macMaxFrameRetries</i>	3 (<i>aMaxFrameRetries</i>)	0÷7 Default: 3	Max number of re-transmissions
<i>macMaxCSMABackoff</i>	0÷5 Default: 4	0÷5 Default: 4	Max number of backoff stages
<i>macMaxBE</i>	5 (<i>aMaxBE</i>)	3÷8 Default: 5	Maximum Backoff Window Exp.
<i>macMinBE</i>	0÷3 Default: 3	0÷7 Default: 3	Minimum Backoff Window Exp.

	<i>macMinBE</i>	<i>macMaxBE</i>	<i>macMaxCSMABackoff</i>	<i>macMaxFrameRetries</i>
DPS	3	5	4	3
SPS	7	8	5	7
NPS	8	10	10	10

- **DPS:** Valori dei parametri di default definiti nello standard;
- **SPS:** (Standard) Valori dei parametri più grandi **permessi** dallo standard;
- **NFS:** (Non-Standard) Valori dei parametri più grandi **non permessi** dallo standard.

Delivery Ratio vs Packet Error Rate



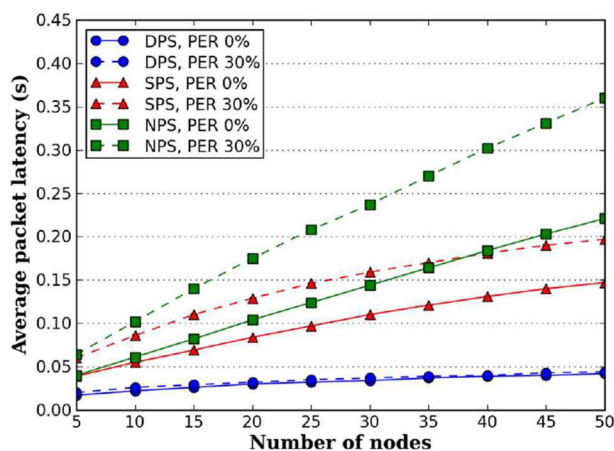
DPS con un **PER** (Packet Error Rate) dello 0% o con un PER del 30 % non ha buone performance.

Ma se ci spostiamo ad SPS, sia con un PER del 30% che con un PER dello 0% il Delivery Ratio non diminuisce molto.

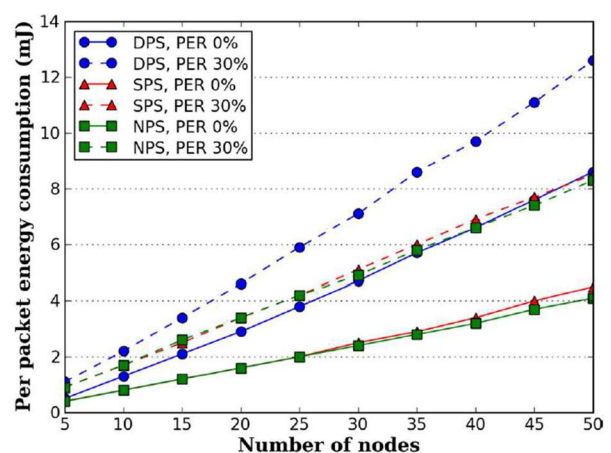
L'inaffidabilità del MAC 802.15.4 è dovuta principalmente al CSMA-CA:

- I Beacons periodicamente sincronizzano l'accesso al canale;
- I valori dei parametri di default del MAC non sono appropriati.

Avg. Latency



Energy/msg



Miglioramenti

Computazione offline Model-Based

Park's Model

Nel paper fornito è creato una **Catena di Markov** con le seguenti assunzioni:

- Rete con topologia a stella;
- Canale di comunicazione ideale;
- Numero di nodi fissato e conosciuto.

È stata valutata la **Reliability (R)**, **Latenza (L)** e l'**Energia per pacchetto (E_{tot})**

Questo algoritmo offline **fornisce i parametri MAC impostati in modo da garantire:**

$$[R \geq R_{min} \wedge D \leq D_{max}]$$

Mantenendo nel frattempo E_{tot} .

Il problema è che questo è un approccio offline, quindi non conosce le condizioni operative della rete di sensori, e richiede (in base all'hardware utilizzato) molto tempo.

Computazione online Model-Based

Per risolvere questo problema, nel 2012 è stato preparato un **Algoritmo Online** che prova a stimare α, β, τ per mezzo di misurazioni dirette.

- α : probabilità di trovare il canale occupato durante il primo CCA (Collision Phase);
- β : probabilità di trovare il canale occupato durante il secondo CCA;
- τ : probabilità che il nodo provi il primo CCA in uno slot di backoff generico.

Questi tre passi sono stati introdotti nel Park's Model.

Questo modello ha una bassa accuracy (probabilmente non buona), richiede di conoscere i parametri di configurazione, non è quindi raccomandata per WSNs multi-hop.

Adattamento Measurement-Based

È stato quindi proposto **ADAPT (ADaptive Access Parameters Tuning)**. Fornisce l'affidabilità richiesta con un minimo consumo energetico ed è basato su misurazioni locali dell'inaffidabilità (congestione, rumore del canale, messaggi scartati a causa dell'eccessivo numero di backoff/rtx).

I limiti di questo modello sono le oscillazioni a siccome ADAPT tende ad oscillare tra due sets adiacenti (consumando più energia) ed è memori-less.

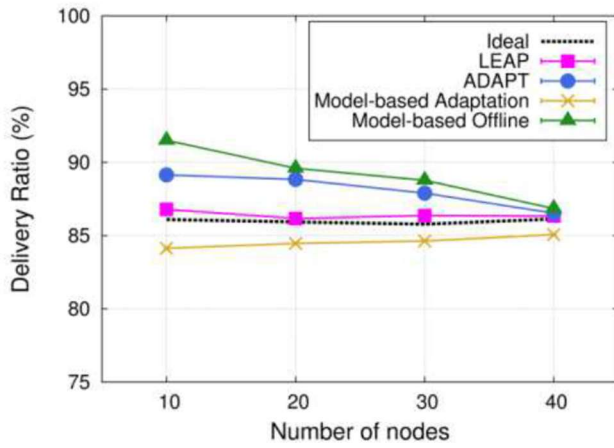
Adattamento Learning-Based

La proposta è stata **LEAP (LEarning based Adaptive Parameters Tuning)**. È basato su misurazioni locali (come ADAPT): Packets Delivery Ratio, Miss Ratio, Average Energy per Packet, Average Latency. Non richiede nessun parametro di input siccome impara dalla storia passata.

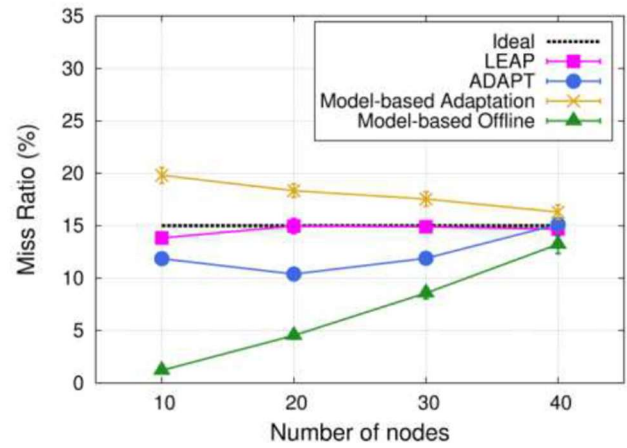
LEAP seleziona il setting ottimale dei parametri in due fasi:

1. **Exploration Phase:** i nodi costruiscono una Learning Table usando la knowledge;
2. **Exploitation Phase:** i nodi controllano che la tabella per derivarne dei setting ottimali.

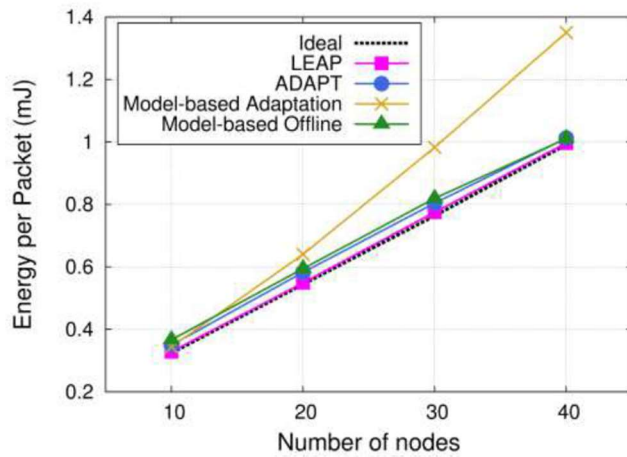
Delivery ratio (%)



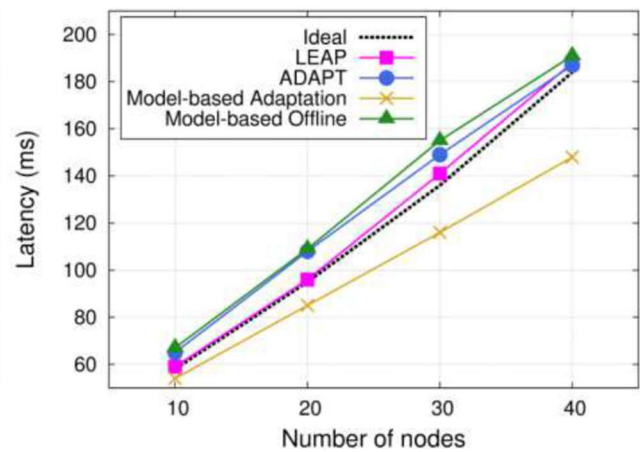
Miss Ratio (%)



Energy per packet (mJ)



Latency (ms)



IEEE 802.15.4e

L'intento di questo standard è quello di migliorare e aggiungere funzionalità all'802.15.4-2006 MAC:

- Low Energy (LE);
- Information Elements (IE);
- Enhanced Beacons (EB);
- Multipurpose Frame;
- MAC Performance Metrics;
- Fast Association (FastA).

MAC Behaviors Modes

- **BLINK** (Radio Frequency Identification Blink): Identificazione, localizzazione e tracciamento di oggetti e persone;
- **AMCA** (Asynchronous Multi-Channel Adaptation): Domini applicativi in cui è necessario un ampio dispiegamento dei nodi (per esempio automazione e controllo dei processi industriali, monitoraggio delle infrastrutture ecc.);
- **DSME** (Deterministic & Synchronous Multi-Channel Extension): Applicazioni industriali e commerciali in cui vi sono requisiti stringenti di affidabilità e temporizzazione. Funzionano su una Beacon-Enabled PAN con una modalità di sincronizzazione per mezzo di un multi-hop superframe;
- **LLDN** (Low Latency Deterministic Network): Applicazioni che richiedono latenza molto molto bassa (ad esempio automazione di una fabbrica, controllo robotico ecc.). Ideato per piccole reti, con topologia a stella, usando un superframe;
- **TSCH** (Time Slotted Channel Hopping): Dominio applicativo come automatizzazione dei processi. Combina l'accesso comunicazione time-slotted, comunicazione multi-channel e channel hopping.

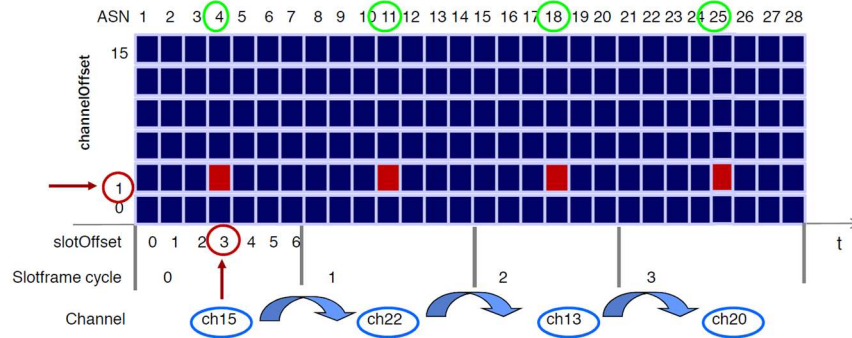
Frequency Translation

L'offset del canale viene tradotto nella frequenza operativa f .

$$f = F \{ (ASN + chOf) \mod n_{ch} \}$$

Table I. FrequencyTranslation

k	ASN	$chOf$	f
0	4	1	5
1	11	1	12
2	18	1	3
3	25	1	10



- **ASN**: numero totale di slots trascorsi da quando la rete è stata dispiegata;
- **N_{ch}**: numero di canali utilizzati;
- **F**: implementato come una lookup table contenente i set di canali disponibili.

Cambiando l'ASN cambia la frequenza.

ZigBee

Lo standard ZigBee è l'unico protocollo applicabile in applicazioni a basso data rate, consentendo alle reti di scalare ad ampie dimensioni in autonomia e operare per anni senza interventi manuali.

Alliance

Zigbee, come Bluetooth, ha un'Alliance, che funziona come un consorzio in cui possono entrare aziende e università con lo scopo di supportare la standardizzazione.

Nel 2002, la ZigBee Alliance era inizialmente organizzata come una corporazione indipendente, neutrale e non profit, in cui ognuno poteva entrare e partecipare.

Applicazioni

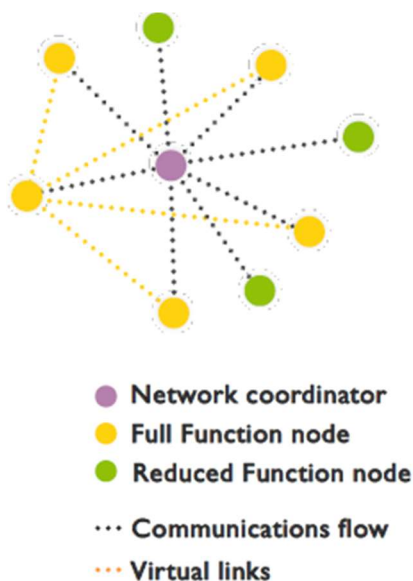
ZigBee è utilizzato in diverse aree applicative, come:

- Automatizzazione delle costruzioni;
- Salute personale;
- Domotica;
- Industria (il protocollo vi si adatta meglio ed è più robusto).

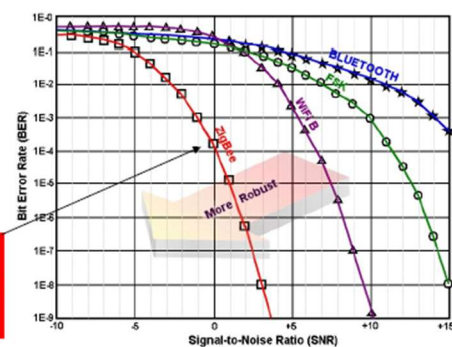
Caratteristiche Base

ZigBee supporta il **Full Mesh Networking** ed è ottimizzato per applicazioni in cui la temporizzazione è fondamentale o per la gestione dell'energia.

- **Data Rate:** 250 kbps;
- **Canali:** 27 (2 bande);
- **Nodi:** 65536.

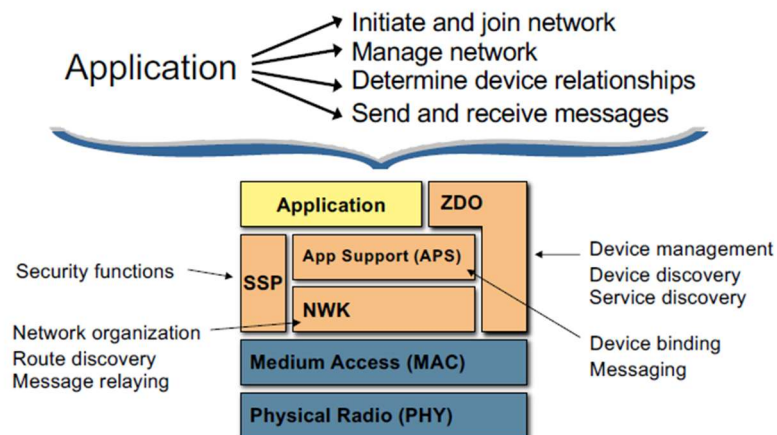


ZigBee technology relies upon IEEE 802.15.4, which has excellent performance in low SNR environments



Frequency Band	License Required?	Geographic Region	Data Rate	Channel Number(s)
868.3 MHz	No	Europe	20kbps	0
902-928 MHz	No	Americas	40kbps	1-10
2405-2480 MHz	No	Worldwide	250kbps	11-26

Stack Architecture



Sono presenti alcuni layer verticali, che si occupano della sicurezza e dell'organizzazione.

L'App Support è molto importante siccome un livello di adattamento che crea conformità tra i messaggi del livello applicazione e quello che i livelli inferiori dovrebbero vedere.

Tipologie di Dispositivi

- **Coordinator (ZC):** Ogni ZigBee Network ne ha uno. Avvia il processo di formazione della rete;
- **Router (ZR):** Partecipa al multihop routing dei messaggi;
- **End Device (ZED):** Non consente associazione o routing. Consente soluzioni a costi veramente ridotti.

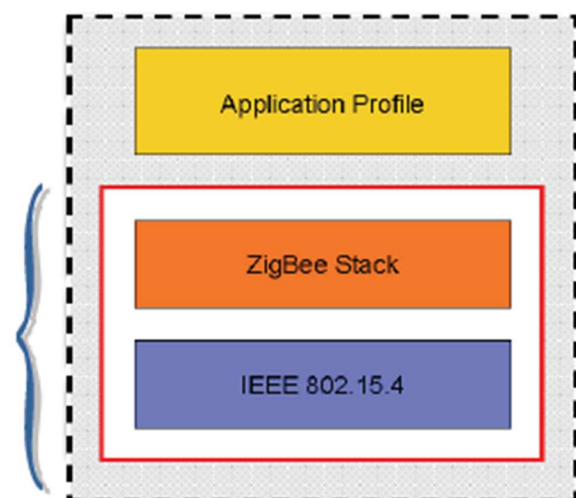
Le topologie possibili sono: **Stella**, **Mesh** e **Cluster Tree**.

Zigbee Compliant Platform (ZCP)

In ZigBee è presente il **Platform Certification**: assicura che tutte le componenti dello stack siano conformi con lo standard ZigBee.

Ciò consente la **Network Interoperability** ma ciò non implica anche interoperabilità a livello applicativo

ZigBee Compliant Platform



Profili Applicativi

I profili applicativi definiscono quali messaggi sono inviati via etere per una specifica applicazione. Ciò consente ai dispositivi con lo stesso profilo, di cooperare tra di loro end-to-end.

ZigBee pubblica una serie di profili pubblici, ma ogni sviluppatore potrebbe creare profili specifici.

Dispositivi Multi-Profilo

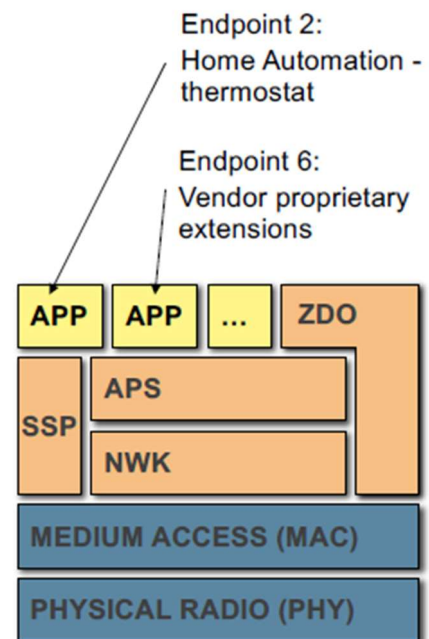
I dispositivi di specifici sviluppatori potrebbero implementare molteplici profili con diversi endpoints

802.15.4 vs ZigBee

- **802.15.4:** Questo standard definisce soltanto i due livelli inferiori della pila ISO/OSI, responsabili per le comunicazioni e le ricezioni radio. Il livello MAC gestisce l'accesso al mezzo utilizzando CSMA/CA.
- **ZigBee:** Costruito su 802.15.4 aggiungendo livelli necessari per supportare multi-hop networking e applicazioni nel mondo reale. Questo standard aggiunge indirizzamento logico, ruoli dei dispositivi, protocolli di routing e creazione delle reti.

Routing

- **802.15.4:** non ha il supporto per il routing;
- **ZigBee:** usa una variante dell'algoritmo di routing AODV e topologie a stella, albero e mesh per la creazione delle reti.



LoRaWAN

Il protocollo LoRaWAN è guidato dai membri collaborativi della LoRaWAN Alliance, un'associazione non-profit aperta, con la missione di standardizzare le Low Power Wide Area Network. Con la **versione 1.1** (metà 2017) sono state aggiunte: Capacità di roaming passivo & handover, chiarificazioni della **Classe B**, switching temporaneo tra **Classe A/C**.

Classi di Dispositivi

Classe A: Alimentati a Batteria

Comunicazioni bidirezionali con messaggi UNICAST. Payloads piccoli, intervalli lunghi. Gli end device inizializzano le comunicazioni (UPLINK), il server comunica in finestre di risposta predeterminate.

Classe B: Low Latency

Comunicazione bidirezionale con slot di ricezione schedulati e messaggi UNICAST e MULTICAST. Beacon periodico dal gateway con una finestra di ricezione extra (ping slot). Il server può avviare la trasmissione ad intervalli fissati.

Classe C: No Latency

Comunicazione bidirezionale con messaggi Unicast e Multicast. Il server può avviare la trasmissione in ogni momento. Gli end-devices sono costantemente in ascolto.

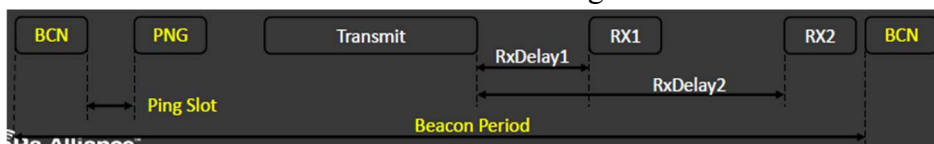
Ingresso nella Rete

Prima che un end device possa comunicare nella rete LoRaWAN, dev'essere prima attivato:

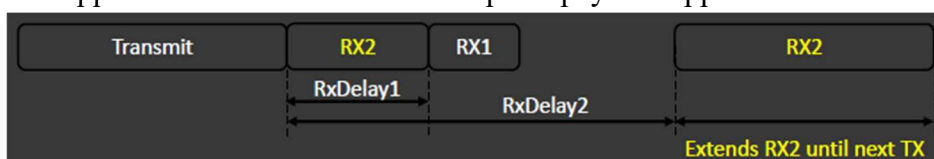
- **Device Address (DevAddr)**: Identificatore da 32 bit, unico nella rete e presente in ogni dataframe. Viene condiviso tra end device, network server e application server.



- **Network Session Key (NWSKEY)**: Chiave crittografica AES da 128 bit, unica. Condivisa tra end device e network server. Fornisce integrità e sicurezza.



- **Application Session Key (AppSKey)**: Chiave crittografica AES da 128 bit, unica. Condivisa tra end device e il application server. Impiegata per cifrare/decifrare i messaggi dati applicazione. Fornisce sicurezza per il payload applicazione.

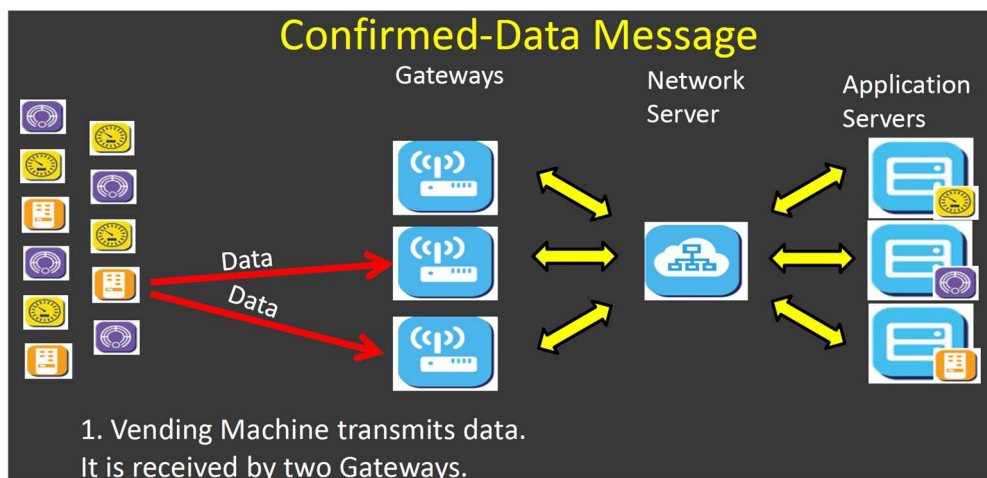


Per scambiare queste informazioni, esistono due metodi di attivazione:

1. **Over-The-Air Activation (OTAA):** Gli end devices trasmettono, all'application server, una **join request** contenente: Globally Unique Identifier (**DevEUI**), App Identifier (**AppEUI**), Authentication with App Key (**AppKey**).
 - a. L'end device riceve una join accept dal application server;
 - b. L'end device estrae e conserva l'indirizzo del dispositivo;
 - c. L'end device deriva:
 - i. **Net Session Key (NwkSKey);**
 - ii. **App Session Key (AppSKey).**
2. **Activation By Personalization (ABP):** Le chiavi condivise vengono conservate soltanto in produzione. Sono legate ad una rete specifica. Il dispositivo è pronto per comunicare sulla rete.

Confirmed-Data Message

1. Il distributore automatico trasmette dati. Essi vengono ricevuti da due gateways;
2. Entrambi i gateway inoltrano i dati al network server;
3. Il network server inoltra i dati all'application server;
4. L'application server risponde con un'ACK;
5. Il network server sceglie il miglior percorso per trasmettere l'ACK all'end device;
6. Il gateway scelto trasmette l'ACK all'end device.



Tecnologie Usate in LoRaWAN

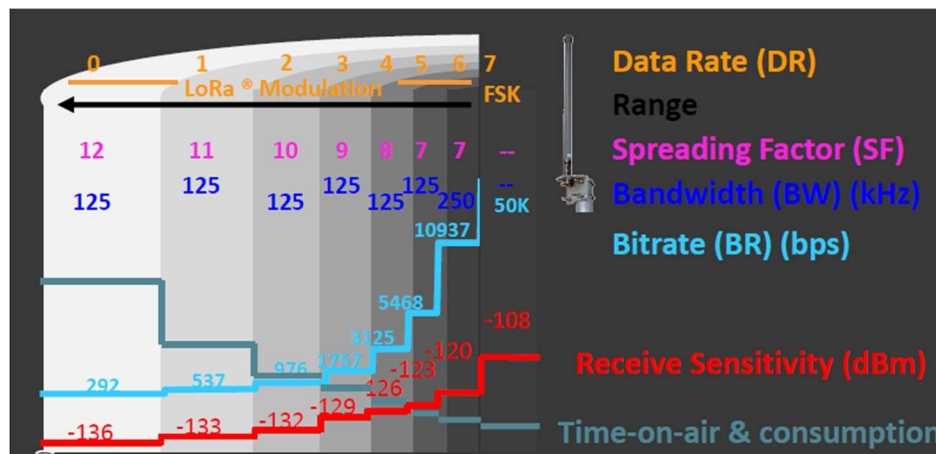
LoRaWAN usa una tecnologia **Spread Spectrum**, sviluppata da Semtech con una modulazione FM cifrata.

In questo protocollo, il concetto di **Spreading Factor (SF)** gioca un ruolo fondamentale perché definisce quanto il segnale viene diffuso nel tempo, impattando sulla robustezza e sulla velocità.

Lo SF è essenzialmente la lunghezza del codice usato per codificare ogni simbolo. Il suo ruolo cambia da 7 a 12.

- **SF Maggiore (12):** Maggiore durata dei simboli, minore data rate, maggiore robustezza e maggiore range;

- **SF minore (7):** Minore durata dei simboli, maggiore data rate e minore range.



Processing Gain

Quando un segnale viene diffuso con un SF maggiore, il ricevente applica lo stesso spreading code durante la demodulazione. Questo processo fornisce un processing gain, che migliora il rapporto segnale-rumore (**Signal-to-Noise Ratio, SNR**) al ricevente.

Adaptive Data Rate (ADR)

LoRaWAN può gestire automaticamente lo SF per ogni end-device per ottimizzare data rate maggiori rispetto alla distanza, per massimizzare la durata della batteria e ottenere la massima capacità di rete.

LoRaWAN (Technical Overview PDF)

LoRa è il livello fisico impiegato per creare il link di comunicazione long range. È basato sulla modulazione **CHPR Spread Spectrum** per ottenere bassa potenza, che mantiene le stesse caratteristiche low power della modulazione FSK ma incrementa significativamente il range di comunicazione.

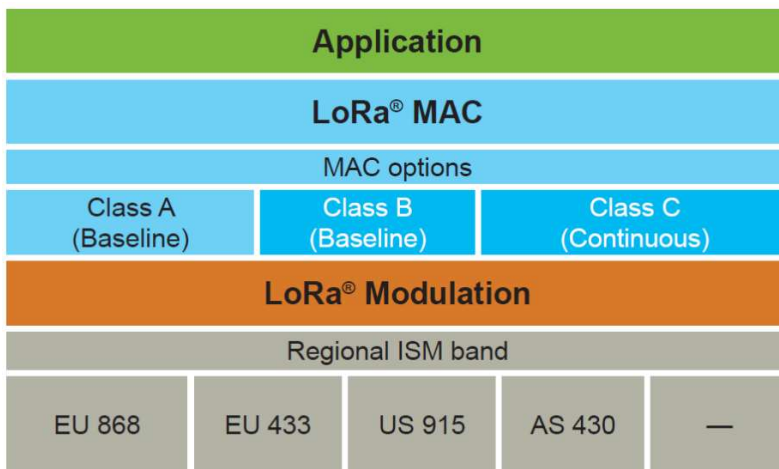
Where Does LPWAN Fit?

	Local Area Network Short Range Communication	Low Power Wide Area (LPWAN) Internet of Things	Cellular Network Traditional M2M
	40%	45%	15%
Well established standards In building		Low power consumption Low cost Positioning	Existing coverage High data rate
Battery Live Provisioning Network cost & dependencies		High data rate Emerging standards	Autonomy Total cost of ownership
Bluetooth 4.0		LoRa	GSM 3G+ / H+ 4G

LPWAN offre anni di durata della batteria ed è progettata per sensori e applicazioni che necessitano l'invio di piccolo quantità di dati su lunghe distanze, poche volte ogni ora da ambienti diversi.

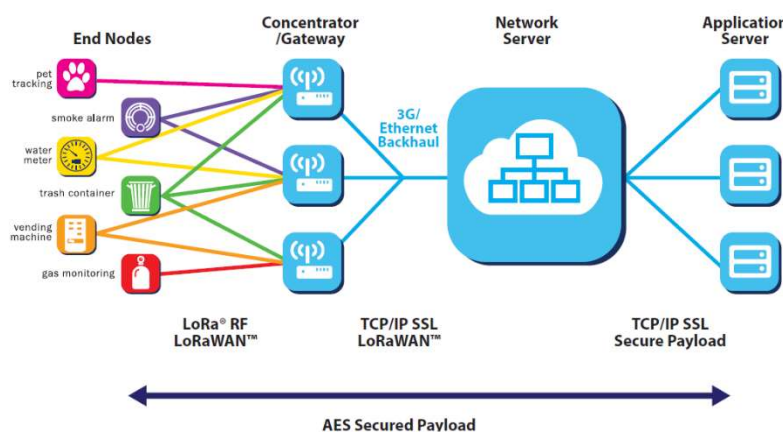
What is LoRaWAN?

LoRaWAN definisce il protocollo di comunicazione e l'architettura di sistema per la rete, mentre il livello fisico di LoRa abilita il link di comunicazione long range.



È possibile usare più chip in LoRa con data rate diversi, usando il chip spreading spectrum. Questo vuol dire che se due comunicazioni impiegano due chip differenti o lunghezze diverse del codice, con la stessa frequenza, non c'è interferenza, supponendo un'ortogonalità perfetta. Ma siccome essa non esiste, parliamo di **Quasi-Orthogonal Codes**.

Con essi accade che se si ha la stessa comunicazione, con codici diversi, si ha un minimo di interferenza, ma non necessariamente abbastanza forte da compromettere la ricezione dei pacchetti.



Molte reti esistenti utilizzano un'architettura di rete mesh. In una rete LoRaWAN, i nodi non sono associati ad uno specific gateway. I dati vengono invece trasmessi da un nodo e ricevuti, in genere, da più gateway. Ogni gateway inoltra il pacchetto ricevuto al network server cloud based. Esso gestisce i pacchetti filtrandoli, esegue controlli di sicurezza ecc.

LoRaWAN Regional Overview

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 + 8	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee	In definition by Technical Committee
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kbps				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

Le specifiche LoRaWAN variano leggermente in ogni regione, basandosi sulle allocazioni regionali dello spettro e sui requisiti normativi.

In Europa ci sono **dieci canali**:

- **1-8**: 250 bps → 5.5 kbps;
- **9**: 11 kbps;
- **10 (FSK)**: 50 kbps.

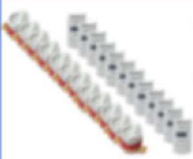
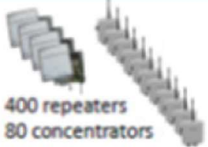
In Nord America ci sono:

- **64 canali**: 125 kHz (902.3-914.9 MHz)
- **8 canali**: 500 kHz (903-914.9 MHz)
- **8 canali (downlinks)**: 500 kHz (903.3-914.9 MHz)

Comparing LPWAN Technology Options

Feature	LoRaWAN	Narrow-Band	LTE Cat-1 2016 (Rel12)	LTE Cat-M 2018 (Rel13)	NB-LTE 2019(Rel13+)
Modulation	SS Chirp	UNB / GFSK/BPSK	OFDMA	OFDMA	OFDMA
Rx bandwidth	500 - 125 KHz	100 Hz	20 MHz	20 - 1.4 MHz	200 KHz
Data Rate	290bps - 50Kbps	100 bit/sec 12 / 8 bytes Max	10 Mbit/sec	200kbps – 1Mbps	~20K bit/sec
Max. # Msgs/day	Unlimited	UL: 140 msg/day	Unlimited	Unlimited	Unlimited
Max Output Power	20 dBm	20 dBm	23 - 46 dBm	23/30 dBm	20 dBm
Link Budget	154 dB	151 dB	130 dB+	146 dB	150 dB
Battery lifetime - 2000mAh	105 months	90 months		18 months	
Power Efficiency	Very High	Very High	Low	Medium	Med high
Interference immunity	Very high	Low	Medium	Medium	Low
Coexistence	Yes	No	Yes	Yes	No
Security	Yes	No	Yes	Yes	Yes
Mobility / localization	Yes	Limited mobility, No loc	Mobility	Mobility	Limited Mobility No Loc

LPWAN Cost vs. Legacy Systems

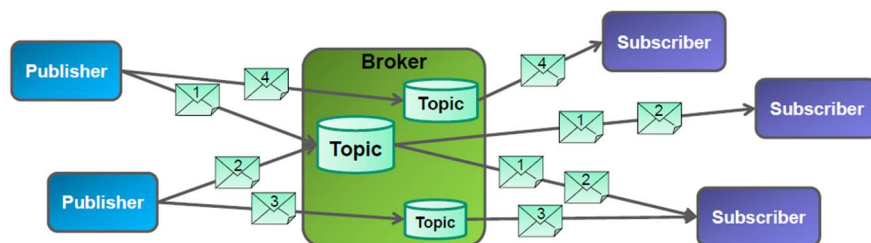
2000 apartments 80 buildings/staircases Sensor cost equal		 An average of 5 sensors/apartment (temperature, humidity and warm water consumption). Fire detection optional. Potential IoT revenue with LoRa.																																													
  400 repeaters 80 concentrators		 																																													
WM-Bus <table><tr><td rowspan="4">CAPEX</td><td>Design, prep</td><td>4 000</td><td>€100/hour</td></tr><tr><td>Repeaters</td><td>32 000</td><td>€80/each</td></tr><tr><td>Concentrators</td><td>20 000</td><td>€250/each</td></tr><tr><td>Installation</td><td>24 200</td><td>€55/hour</td></tr><tr><td colspan="2">SUM</td><td colspan="2">€ 80 200</td></tr></table> <table><tr><td rowspan="2">REVENUE</td><td>Business logic & Statistics</td><td rowspan="2">36 842</td><td rowspan="2">Yearly occurring</td></tr><tr><td>Billing, invoice</td></tr></table>		CAPEX	Design, prep	4 000	€100/hour	Repeaters	32 000	€80/each	Concentrators	20 000	€250/each	Installation	24 200	€55/hour	SUM		€ 80 200		REVENUE	Business logic & Statistics	36 842	Yearly occurring	Billing, invoice	LoRa Network Operator <table><tr><td rowspan="3">CAPEX</td><td>Design, prep</td><td>1 600</td><td>€100/hour</td></tr><tr><td>GW</td><td>6 000</td><td>€1500/each (500apts/GW)</td></tr><tr><td>Installation</td><td>880</td><td>€55/hour</td></tr><tr><td colspan="2">SUM</td><td colspan="2">€ 8 480</td></tr></table> <table><tr><td rowspan="3">REVENUE</td><td>Business logic & Statistics</td><td rowspan="3"></td><td>Yearly occurring</td></tr><tr><td>Billing, invoice</td><td>Yearly occurring</td></tr><tr><td>Other IoT services</td><td>Occuring</td></tr></table>		CAPEX	Design, prep	1 600	€100/hour	GW	6 000	€1500/each (500apts/GW)	Installation	880	€55/hour	SUM		€ 8 480		REVENUE	Business logic & Statistics		Yearly occurring	Billing, invoice	Yearly occurring	Other IoT services	Occuring
CAPEX	Design, prep		4 000	€100/hour																																											
	Repeaters		32 000	€80/each																																											
	Concentrators		20 000	€250/each																																											
	Installation	24 200	€55/hour																																												
SUM		€ 80 200																																													
REVENUE	Business logic & Statistics	36 842	Yearly occurring																																												
	Billing, invoice																																														
CAPEX	Design, prep	1 600	€100/hour																																												
	GW	6 000	€1500/each (500apts/GW)																																												
	Installation	880	€55/hour																																												
SUM		€ 8 480																																													
REVENUE	Business logic & Statistics		Yearly occurring																																												
	Billing, invoice		Yearly occurring																																												
	Other IoT services		Occuring																																												

MQTT (Message Queue Telemetry Transport)

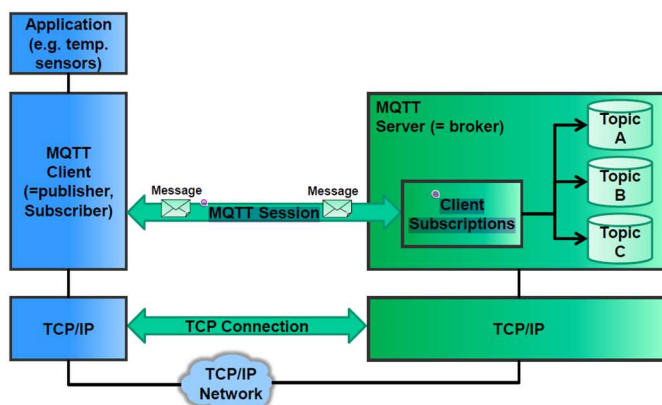
È un protocollo progettato circa 10 anni fa (n.d.r. Il prof a lezione dice così, ma la prima versione è del 1999. Il corso andrebbe aggiornato), con lo scopo di supportare comunicazioni leggere e applicazioni IoT. È adatto al trasporto di dati di telemetria (dati di sensori e attuatori).

Caratteristiche

- **Coda dei messaggi e protocollo di trasporto leggeri:** Impiega il protocollo TCP (affidabile);
- **Modello Publish/Subscribe:** Il publisher è un nodo con il compito di produrre dati e ha un Topic;
- **Modello di comunicazione asincrona con messaggi (eventi):** Quando i dati vengono generati dal publisher e il subscriber non è attivo, il Broker MQTT conserva i dati per inviarli in seguito in una o più code. Quando il subscriber si riattiva dopo un po' di tempo, esprime interesse e in seguito il broker gli invia i dati;
- **Overhead e ampiezza di banda minori:** Header grande 2 bytes, molto piccolo comparato ad altri protocolli;
- **Decoupling (disaccoppiamento):** Comportamento che funziona in coppia con la comunicazione asincrona. Significa che non devo avere necessariamente un'associazione uno ad uno.



Modello/Architettura MQTT



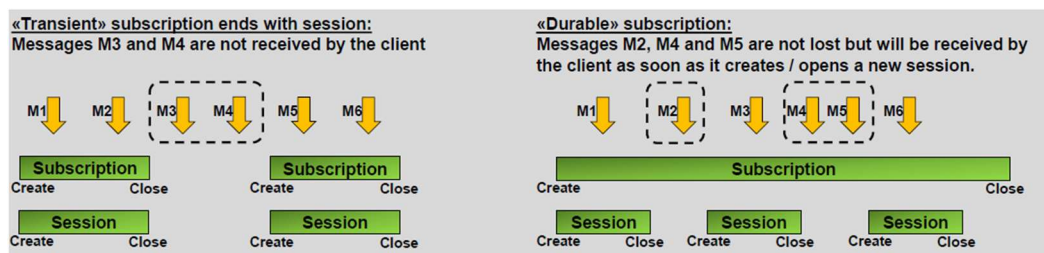
È un'architettura semplice. Prima di avviare lo scambio di messaggi con MQTT, il protocollo stabilisce una connessione TCP. Solo in seguito è possibile avviare la sessione MQTT.

Nella sessione MQTT, ci sono due attori principali: publisher e subscriber. Queste sessioni (MQTT-TCP) sono disaccoppiate e indipendenti.

I Clients si sottoscrivono ai topics per pubblicare e ricevere messaggi da un server (MQTT Broker) dove vengono eseguiti i topics e ci sono tecnicamente le code dei messaggi.

La sessione MQTT identifica il collegamento di un client ad un server.

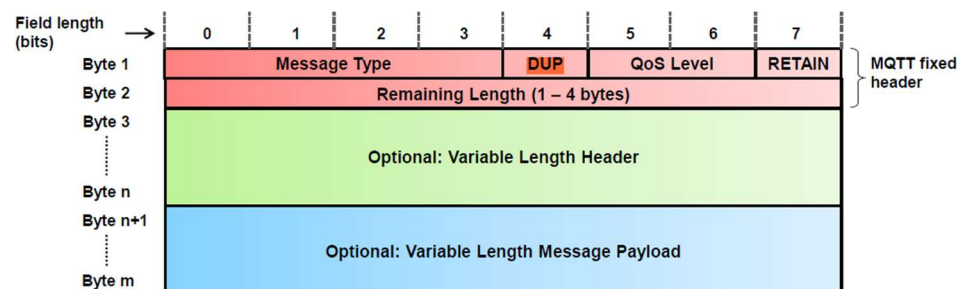
Subscription



1. **Transiente:** Una sottoscrizione termina con la sessione. Dopo la prima chiusura, M3 e M4 vengono prodotti ma non consegnati;
2. **Durable:** M2, M4, M5 vengono prodotti, conservati dal broker e quando la sessione riparte i messaggi vengono consegnati al destinatario.

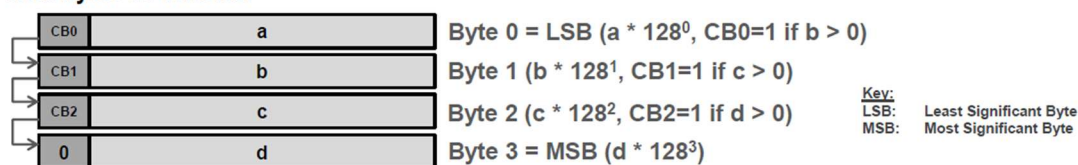
Message Format

I messaggi MQTT contengono un header obbligatorio di due Byte, un variable length header specifico per ogni messaggio e opzionale e un payload.

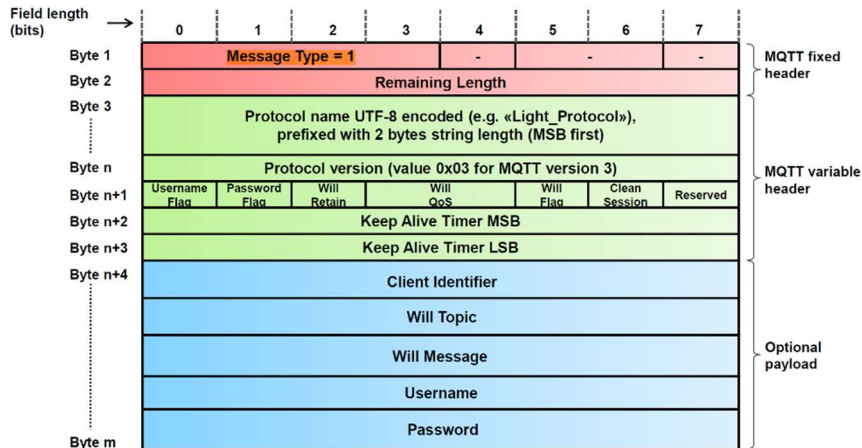


- **Message Type:** Reserved, Connect, Connack, ..., Subscribe, Disconnect; z
- **DUP:** Flag che indica un messaggio duplicato. Avverte il destinatario che il messaggio potrebbe essere già stato ricevuto;
- **QoS Level:** 0 al più una consegna, 1 al meno una consegna, 2 esattamente una;
- **Retain:** 1 istruisce al server di conservare l'ultimo messaggio Publish ricevuto e di inviarlo come primo messaggio di una nuova sottoscrizione. Quando un nuovo client si sottoscrive al topic, il server invia il messaggio di retain;
- **Remaining Length:** Questo campo codifica, opzionalmente, la somma del variable length header e del payload. Per risparmiare bits, la lunghezza rimanente è un campo di lunghezza variabile grande da 1 a 4 bytes. Il più significativo bit del length field ha il significato di "continuation bit" (CB). Se ci sono altri bytes, viene impostato a 1.

Remaining length is encoded as $a * 128^0 + b * 128^1 + c * 128^2 + d * 128^3$ and placed into the RL field bytes as follows:



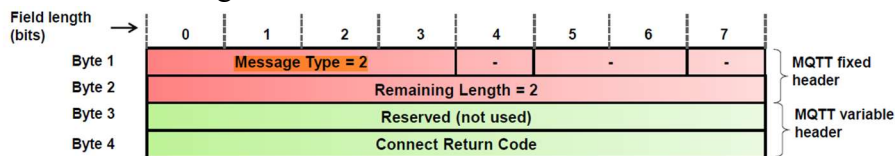
1. Connect message:



CONNECT message field	Description / Values
Protocol Name	UTF-8 encoded protocol name string. Example: «Light_Protocol»
Protocol Version	Value 3 for MQTT V3.
Username Flag	If set to 1 indicates that payload contains a username.
Password Flag	If set to 1 indicates that payload contains a password. If username flag is set, password flag and password must be set as well.
Will Retain	If set to 1 indicates to server that it should retain a Will message for the client which is published in case the client disconnects unexpectedly.
Will QoS	Specifies the QoS level for a Will message.
Will Flag	Indicates that the message contains a Will message in the payload along with Will retain and Will QoS flags. More details see MQTT will message .
Clean Session	If set to 1, the server discards any previous information about the (re)-connecting client (clean new session). If set to 0, the server keeps the subscriptions of a disconnecting client including storing QoS level 1 and 2 messages for this client. When the client reconnects, the server publishes the stored messages to the client.
Keep Alive Timer	Used by the server to detect broken connections to the client. More details see Keepalive timer .
Client Identifier	The client identifier (between 1 and 23 characters) uniquely identifies the client to the server. The client identifier must be unique across all clients connecting to a server.
Will Topic	Will topic to which a will message is published if the will flag is set.
Will Message	Will message to be published if will flag is set.
Username and Password	Username and password if the corresponding flags are set.

Dopo un certo tempo, con i timer Keep Alive MSB e LSB, la comunicazione è terminata per tenere libero il canale. Il timer è impiegato per individuare connessioni cadute.

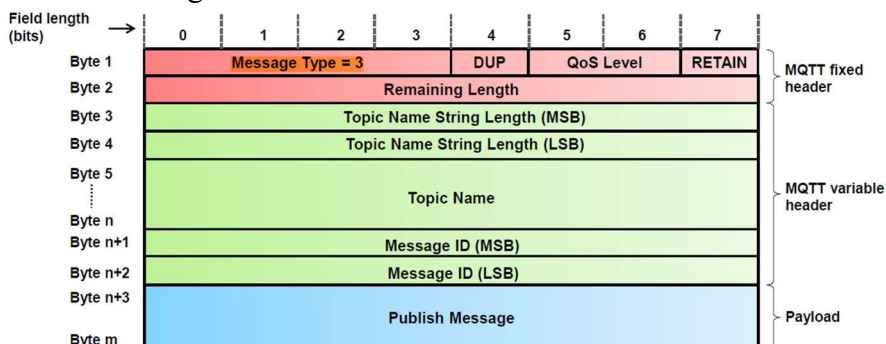
2. ConnACK message



CONNACK message field	Description / Values
Reserved	Reserved field for future use.
Connect Return Code	0: Connection Accepted 1: Connection Refused, reason = unacceptable protocol version 2: Connection Refused, reason = identifier rejected 3: Connection Refused, reason = server unavailable 4: Connection Refused, reason = bad user name or password 5: Connection Refused, reason = not authorized 6-255: Reserved for future use

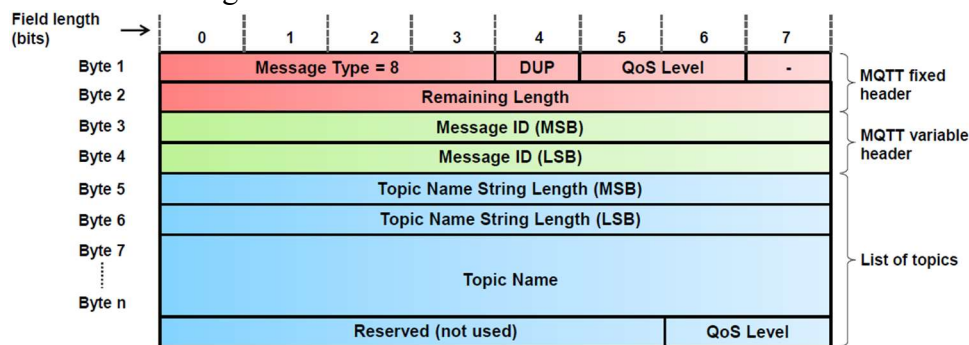
Quando un client prova a connettersi, non necessariamente la connessione ha successo.

3. Publish message



4. PubACK message

5. Pubrec message
6. Pubrel message
7. Pubcomp message
8. Subscribe message



SUBSCRIBE message field	Description / Values
Message ID	The message ID field is used for acknowledgment of the SUBSCRIBE message since these have a QoS level of 1.
Topic Name with Topic Name String Length	Name of topic to which the client subscribes. The first 2 bytes of the topic name field indicate the topic name string length. Topic name strings can contain wildcard characters as explained under Topic wildcards . Multiple topic names along with their requested QoS level may appear in a SUBSCRIBE message.
QoS Level	QoS level at which the clients wants to receive messages from the given topic.

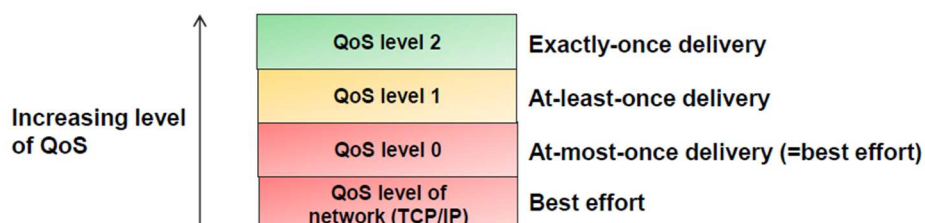
9. SubACK message
10. Unsubscribe message
11. Unsuback message
12. Disconnect message
13. Pingreq message
14. Pingresp message

0 e 15 sono riservati

MQTT QoS

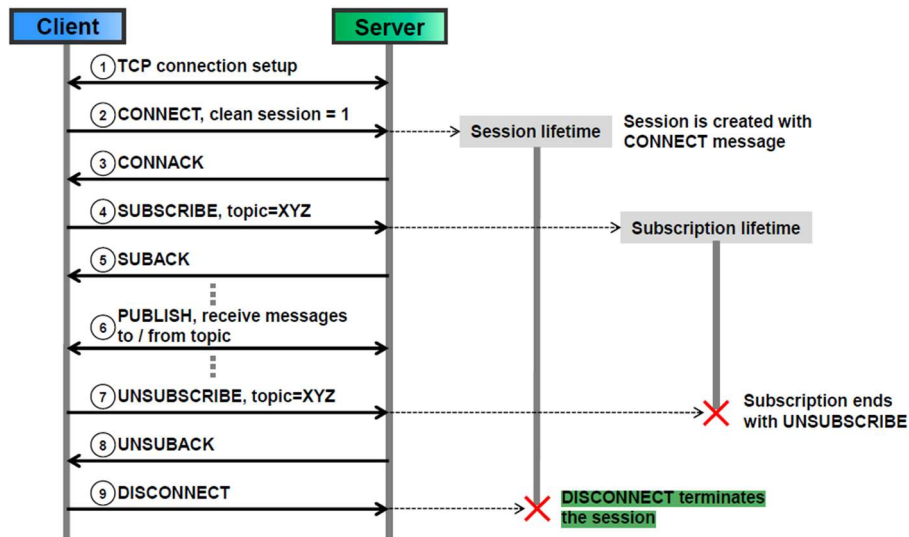
MQTT fornisce la Quality of Service di consegna tipica, ma anche se TCP/IP garantisce la consegna dei dati, può comunque verificarsi perdita di dati se una connessione TCP viene interrotta e i messaggi in transito sono perduti.

- **Livello 0:** I messaggi sono consegnati con le garanzie della rete sottostante (TCP/IP);
- **Livello 1:** È garantito che i messaggi vengano consegnati, ma potrebbero esserci duplicati;
- **Livello 2:** Combinazione del livello 0 e del livello 1;

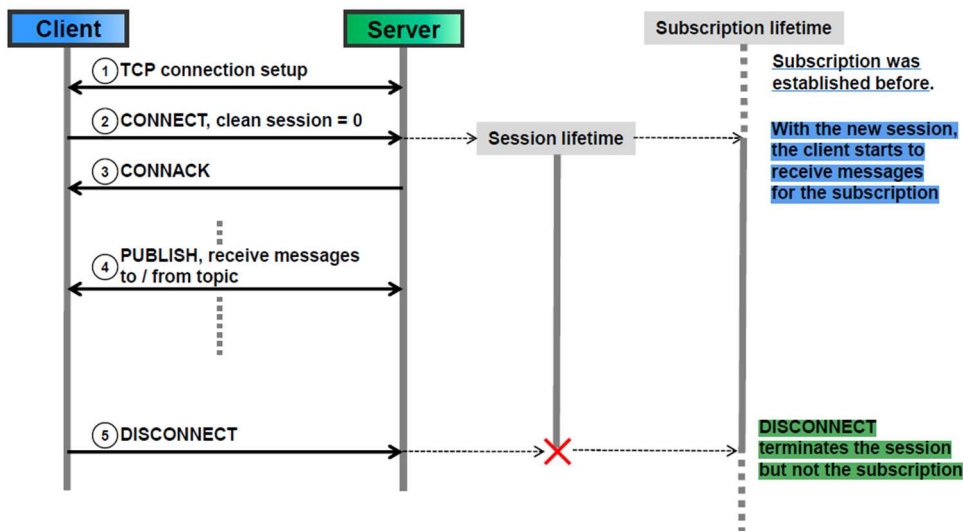


Connect e Subscribe

- Caso 1:** La sessione e la sottoscrizione vengono impostate con la flag “clean session” a 1 (“transient subscription”)
 - 2: questa flag viene usata per rimuovere tutti i messaggi delle precedenti sessioni;
 - 6: freccia a due punte siccome in alcuni casi è il client che genera i messaggi;
 - 7: quando la connessione dev’essere chiusa;
 - 9: Non può essere inviato DISCONNECT se prima non è stato ricevuto l’UNSUBACK. È però possibile inviare il DISCONNECT senza l’UNSUBSCRIBE;

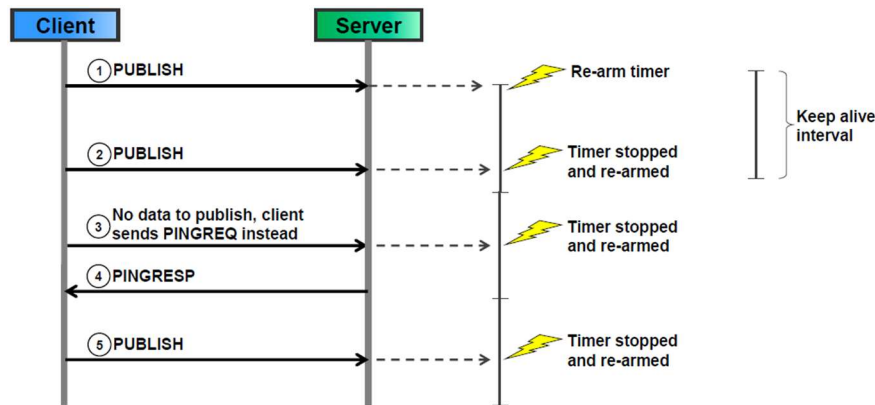


- Caso 2:** La sessione e la sottoscrizione sono inizializzate con la flag “clean session” a 0 (“durable subscription”). La storia della vita della sottoscrizione viene conservata;



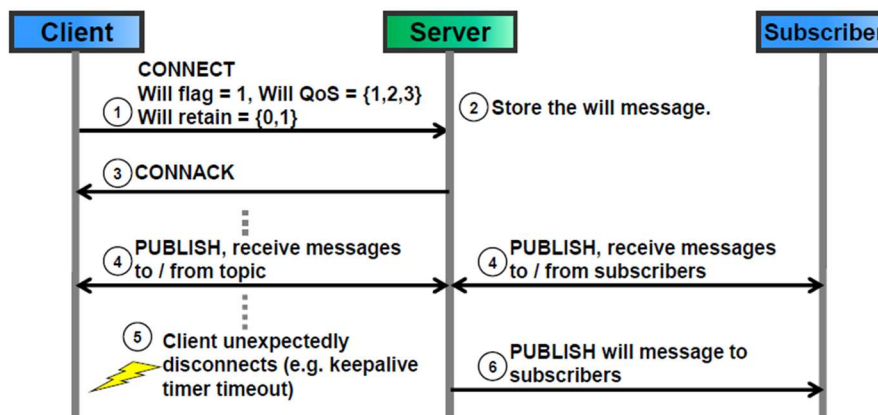
Keep Alive Timer

Definisce l'intervallo di tempo massimo disponibile tra i messaggi del client. Il timer è usato dal server per controllare lo stato della connessione del client. Trascorso un tempo pari a $1,5 * keepaliveTimer$, il server disconnette il client. Se non ci sono dati da inviare, il client invia una PINGREQ.



MQTT Will Message

Se il client si disconnette inavvertitamente, il server invia il messaggio di “will” da parte del client a tutti i subscribers. [will flag = {0, 1, 2}]. Se will retain = 1, il server conserva il messaggio.

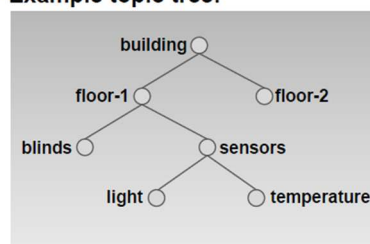


Topic Wildcards

I subscribers spesso sono interessati in un grande numero di topics, ma è molto dispendioso di tempo e risorse. I topic vengono organizzati gerarchicamente con le wildcards. La struttura ad albero è interessante siccome grazie ad essa è possibile aumentare o ridurre la complessità del topic.

Topic string special character	Description
/	Topic level separator. Example: <i>building / floor-1 / sensors / temperature</i>
+	Single level wildcard. Matches one topic level. Examples: <i>building / floor-1 / +</i> (matches <i>building / floor-1 / blinds</i> and <i>building / floor-1 / sensors</i>) <i>building / + / sensors</i> (matches <i>building / floor-1 / sensors</i> and <i>building / floor-2 / sensors</i>)
#	Multi level wildcard. Matches multiple topic levels. Examples: <i>building / floor-1 / #</i> (matches all nodes under <i>building / floor-1</i>) <i>building / # / sensors</i> (invalid, '#' must be last character in topic string)

Example topic tree:

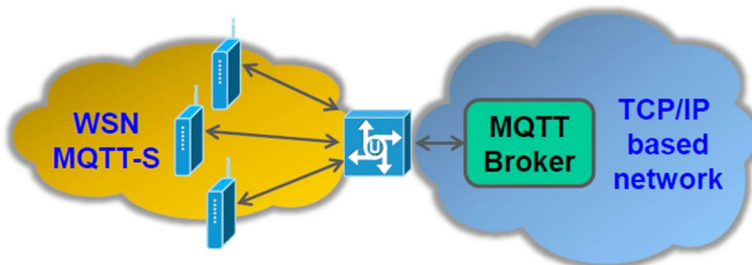


MQTT-S

MQTT-S è ampiamente basato su MQTT, ma implementa qualche ottimizzazione importante per le WSN:

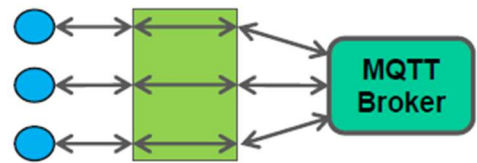
- La stringa del topic viene sostituita da un ID (meno bytes necessari);
- ID dei topic predefiniti che non richiedono una registrazione;
- Procedura di scoperta del broker da parte dei clients, in modo da non dover configurare staticamente l'indirizzo del broker;
- Messaggio di will persistente, in aggiunta alle sottoscrizioni persistenti;
- Supporto del keepalive offline per i clients in sleep, essi riceveranno i messaggi bufferizzati una volta riconnesso.

I gateway MQTT-S (trasparenti o aggreganti) connettono i domini MQTT-S (WSNs) con domini MQTT (reti tradizionali basate su TCP/IP).



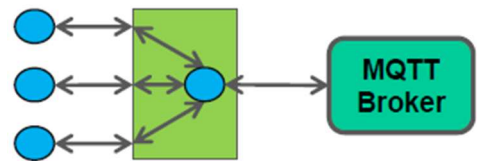
Transparent gateway:

→ 1 connection to broker per client



Aggregating gateway:

→ only 1 connection to the broker



IPv6

IPv6 è la versione più recente di **Internet Protocol**. È un protocollo di livello Rete che consente alle comunicazioni di avvenire sulla rete.

Limiti di IPv4

Gli indirizzi IPv4 si stavano esaurendo a causa della rapida crescita del numero degli utenti della rete internet. La prima versione è stata implementata in ARPANET nel 1983, e utilizzava indirizzi di rete lunghi 32 bit (2^{32} indirizzi).

Il pacchetto IPv6 è costruito con 40 ottetti estesi. Il pacchetto ha due parti: l'header ed il payload. IPv6 ha indirizzi a 128 bit, con uno spazio degli indirizzi disponibili più grandi, per future allocazioni. L'indirizzo viene suddiviso in 8 gruppi, ognuno dei quali contenente 16 bits. Quattro valori esadecimali rappresentano ogni gruppo e i due punti sono usati per separare i gruppi. IPv6 fornisce host communication alla rete con degli identificatori specifici per la sottorete.

IPv4 address: 32 bit, IPv6 address: 128 bit → **2^{128} combinations available!**

3FFE:085B:1F1F:0000:0000:0000:00A9:1234

8 groups of 16-bit hexadecimal numbers separated by “:”

Vantaggi di IPv6

- Offre **affidabilità e velocità maggiori** perché IPv6 ha IPSecurity, che garantisce privacy e integrità.
- Offre **sicurezza di rete** più robusta rispetto ad IPv4

Svantaggi IPv6

Non è retrocompatibile con IPv4. La migrazione completa ad IPv6 si stima che sarà molto lunga a causa delle incompatibilità tra i due protocolli.