***NB: essendo i concetti molto intrecciati tra loro, può essere che un termine utilizzato in una risposta sia definito in una delle risposte alle domande successive.***

***Disp.o = dispositivo***

**Cos’è Zigbee?**

È uno standard molto usato nel mondo IoT, che copre i livelli applicativo e di rete. Un componente del suo livello applicativo (APS – Application sublayer) è per molti versi concettualmente assimilabile al livello di trasporto. ZigBee poggia sullo standard IEEE 802.15.4, il quale copre i livelli fisico e MAC.

I principali punti di forza di Zigbee sono la possibile creazione di reti molto vaste (> 30 000 disp.i ognuna) e quasi completamente autonome, oltre che un forte occhio di riguardo al consumo energetico, per permettere ai dispositivi alimentati a batteria di scaricarsi solo dopo molto tempo.

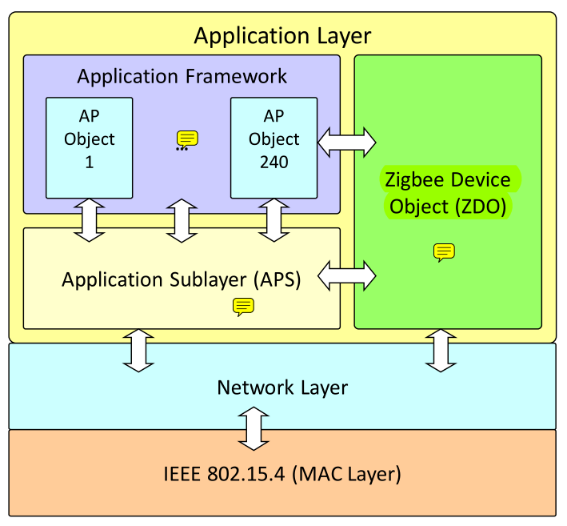
**Altre peculiarità Zigbee**

* reliable (meccanismi di NW recovery)
* data rate basso (poggia su 802.15.4)
* disp.i a basso costo
* facile da sviluppare
* sicuro (sicurezza built-in)

**Altre informazioni generali IEEE 802.15.4**

* “Cugino” del Bluetooth
* Stesse frequenze WiFi e Bluetooth, ma no interferenze
* Valore nominale data rate: 250 kbit/s, in pratica <100 kbit/s
* Ha un livello MAC progettato per garantire un basso consumo di energia.

**Componenti dell’architettura di ZigBee**



* APS (Application sublayer): è per molti versi concettualmente assimilabile al livello di trasporto.
* APO (Application objects): applicazioni indipendenti che condividono lo stesso disp.o. Ce ne sono 240 possibili (ID da 1 a 240). Diversi APOs coprono differenti aspetti delle funzionalità del disp.o, come diverse applicazioni indipendenti.
* ZDO (ZigBee Device Object): è l’APO con ID 0 e svolge il ruolo in un certo senso di gestore del dispositivo. Non è programmabile, ma solo configurabile. Fornisce i servizi di device e service discovery, binding management, network management e node management.

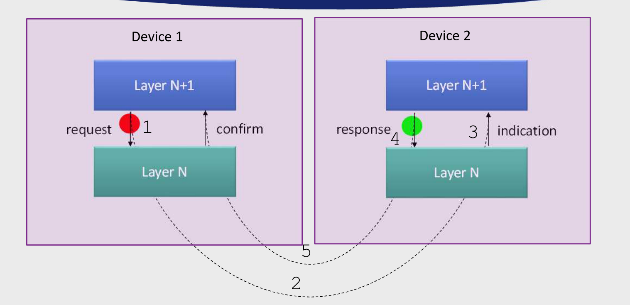
**Di cosa si occupa l’azienda produttrice che vende i disp.i e di cosa i programmatori?**

Se si vuole creare una soluzione utilizzando ZigBee, bisogna comprare i disp.i ZigBee, che già comprendono un'implementazione dello standard e poi semplicemente scrivere il codice per gli APOs e configurare lo ZDO.

**Endpoint Numer**

Ognuno di essi identifica univocamente un APO di un determinato disp.o **(?)**.

**Primitive dei servizi in ZigBee**

****

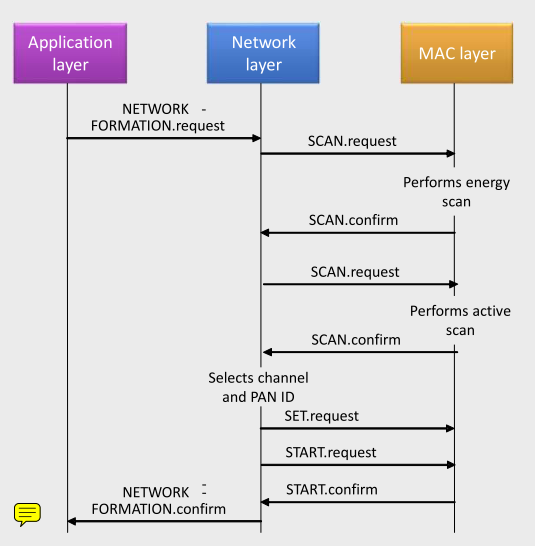
Ogni servizio è specificato da un insieme di primitive di 4 tipi generici:

* Request: dal layer superiore verso l’inferiore, è la richiesta per uno specifico servizio.
* Indication: dal layer inf. al sup., notifica l’occorrenza di un evento riguardante uno specifico servizio.
* Response: dal sup. all’inf., completa la procedura iniziata dalla precedente indication
* Confirm: dall’inf. al sup., fornisce i risultati relativi alle richieste di specifici servizi.

**In che modi un disp.o può aggiungersi ad una rete già esistente?**

* Join through association: è il disp.o che richiede di unirsi.
* Direct join: è un router o il cordinatore ad aggiungere il disp.o. Principalmente ciò avviene quando quest’ultimo ha così poche risorse da non potersi aggiungere autonomamente.

**Creazione di una nuova rete**

****

(L’immagine riguarda ovviamente il disp.o che sarà il coordinatore della rete).

(Premessa: ci può essere una sola rete per canale).

Energy scan: Trovare il canale meno rumoroso.

Active scan: Esaminare su quali canali sono presenti delle reti e quali esse sono (vedere domanda successiva).

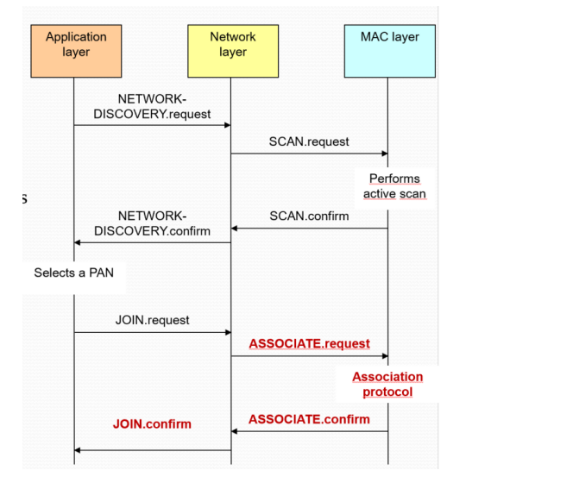
Dopo l’active scan, il coordinatore imposta il proprio indirizzo come 0 e informa il livello MAC di tale indirizzo tramite la SET.request. Perché? In Zigbee i disp.i hanno poche risorse e la rete è low troughput, quindi non si può occupare troppo spazio per mandare un header con indirizzi di rete e un header con indirizzi MAC (della sorgente e destinazione), come accade normalmente quando si usa Internet.

Quando un disp.o è già parte della rete, non si trasmette l'indirizzo MAC per ogni messaggio, ma solo gli indirizzi di rete della destinazione e della sorgente (16 bit ognuno).

Quindi il livello MAC ora contiene l'indirizzo di rete del disp.o.

Con la START.request si “avvia” la nuova PAN ZigBee, che da questo momento in poi emetterà beacons periodici (vedere domanda successiva). Essi sono pacchetti per comunicare all’esterno che tale rete esiste e contengono il PAN ID e altre informazioni. Il livello MAC avvia contestualmente i meccanismi che garantiscono il risparmio energetico.

**DOMANDA 27**

****

Quest’immagine rappresenta la sequenza di primitive utilizzate da un end device o un router per effettuare una join through association. Il dispositivo effettua dapprima una scansione delle reti presenti e in seguito richiede di unirsi a una determinata rete. Per quanto riguarda l’associazione, in questa immagine viene quindi rappresentato il solo punto di vista child-side e dal lato del router o del coordinatore con il quale il disp.o si interfaccia sono presenti altre primitive che servono a gestire la richiesta di associazione.

(LA = livello applicativo, LR = livello di rete, LM = livello MAC)

Il LA manda un NETWORK-DISCOVERY.request al LR, il quale chiede al LM di effettuare un’active scan con una SCAN.request.

(Premessa: da quando la rete è creata, il MAC layer del coordinatore emette beacons periodici).

Active scan: l'antenna rimane in ascolto su ogni canale, uno dopo l'altro. In base alle informazioni contenute nei beacon, il dispositivo scopre qual è la rete che opera su ogni canale (se esiste).

In seguito alla SCAN.confirm, il LR fornisce un report al LA su quali sono le reti che operano nei dintorni, fornendo di ognuna di esse il PAN ID. Il disp.o sceglie quindi la PAN alla quale associarsi (vedere domanda successiva).

Il LA invoca la JOIN.request, che prende in input il PAN ID della rete selezionata e una flag che indica se il nuovo disp.o vuole associarsi come router o come end device. Quindi il LR invia una ASSOCIATE.request al LM. Tale primitiva avvia il protocollo di associazione, un complesso protocollo del LM, che utilizza degli handshake tra il LM del figlio (disp.o che vuole associarsi) e il LM del padre (coordinatore o router che gestisce la richiesta di associazione).

Il padre fornisce al figlio l’indirizzo di rete (c’è l’equivalente della funzionalità fornita dal DHCP, che è automatica e implementata da ogni router e coordinatore). Con l’ASSOCIATE.confirm il LM informa il LR del nuovo indirizzo.

A questo punto il protocollo di associazione è concluso e il LR invia una JOIN.confirm al LA per informarlo che il dispositivo è ora parte della rete.

**In base a cosa il disp.o sceglie la PAN a cui associarsi?**

Come il meccanismo di selezione della PAN avviene, non è parte dello standard.

A volte il PAN ID è impostato a tempo di compilazione dal produttore, a volte la scelta è guidata dall'utente, come fosse una connessione WiFi dal computer ma, essendo i disp.i quasi sempre molto più semplici di un computer, i metodi sono meno user-friendly, come ad esempio l'utilizzo di pulsanti speciali.

**LIVELLO DI RETE ZIGBEE**

**Differenza tra Fully Functional Device e Reduced Functional Device**

Un FFD, a differenza di un RDF, implementa tutte le funzionalità richieste da ZigBee.

**Ruolo dei dispositivi in una rete Zigbee**

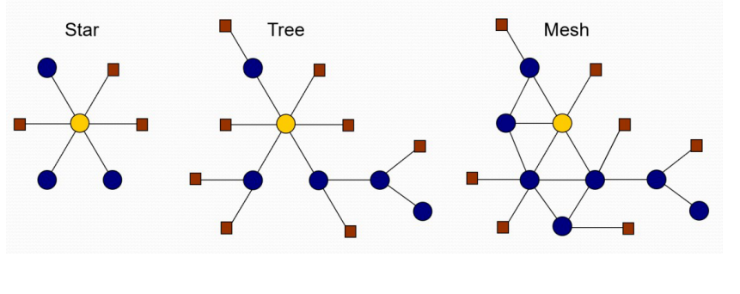
* Coordinatore di rete: FFD che crea e gestisce la rete.
* Router: FFD con capacità di routing.
* End-device: RDF o FFD che si comporta come disp.o semplice (non effettua routing).

Il coordinatore e i router possono svolgere anche altre attività oltre a quelle “esclusive” per loro.

**Quando è scelto il ruolo (di rete) di ogni disp.o?**

A tempo di compilazione (da parte del produttore dello stesso).

**DOMANDA 26**

****

Sono le 3 topologie possibili di rete ZigBee, in cui il giallo indica il coordinatore, i blu sono i router e i rossi gli end devices. Esse sono quelle a:

* Stella: tutti i disp.i sono connessi al coordinatore e comunicano tramite di esso, usa il superframe (vedere la parte relativa all’IEEE 802.15.4).
* Albero: la rete ha una forma ad albero, nello stesso modo in cui si intende “albero” nelle strutture dati. Può utilizzare il superframe.
* Mesh: rete con topologia non a stella e non ad albero. Non usa il superframe.

Nelle topologie ad albero e a mesh si parla di reti multi-hop, cioè la comunicazione passa tramite diversi routers (hops).

**Un disp.o può far contemporaneamente parte di due reti distinte, come nelle piconet (Bluetooth)?**

No.

**LIVELLO APPLICATIVO ZIGBEE**

**Qual è l’importanza del livello applicativo nel caso specifico di ZigBee?**

Senza il livello applicativo, il dispositivo può far parte della rete, ma non può interoperare con gli altri, nel senso che la comunicazione avviene, ma i dispositivi non sanno quali sono i propri ruoli (nel senso applicativo, quindi client/server relativi a determinati cluster appartenenti a determinati profili applicativi) e quali quelli degli altri e di conseguenza non sanno come interpretare i messaggi.

Si parla di standardizzazione a livello applicativo, in quanto si danno ruoli standardizzati ai disp.i.

**Def. Profilo applicativo** *(“mia” definizione)*

Specifica che definisce il comportamento di un determinato tipo di applicazione, in relazione ai vari dispositivi sui quali opera e ai cluster che gli stessi implementano.

(Es. ID 0101: Industrial Plant Monitoring, ID 0104: Home Automation).

In una singola rete possono coesistere più profili applicativi.

Ogni messaggio ha un tag con l'ID del profilo, altrimenti non si sarebbe in grado di interpretare il suo significato.

**Cos’è la ZCL (ZigBee Cluster Library)?**

È una repository per le funzionalità dei cluster.

Esiste una ZCL per ogni profilo applicativo.

Le ZCL servono per specificare il comportamento dell'applicazione e sono una parte diversa rispetto allo standard.

L’obiettivo è quello di rendere i disp.i di un produttore totalmente interoperabili con quelli di un altro: gli APO devono essere sviluppati in accordo con le specifiche delle ZCL, in modo da avere disp.i ZigBee standardizzati.

**Def. Cluster**

Collezione di comandi e attributi che definiscono un’interfaccia per una specifica funzionalità (es. quello con ID 0x0006 del general domain è quello che implementa la funzionalità di on/off).

L’ID di un cluster è univoco solamente nel contesto di un determinato profilo applicativo.

La ZCL utilizza il paradigma client server:

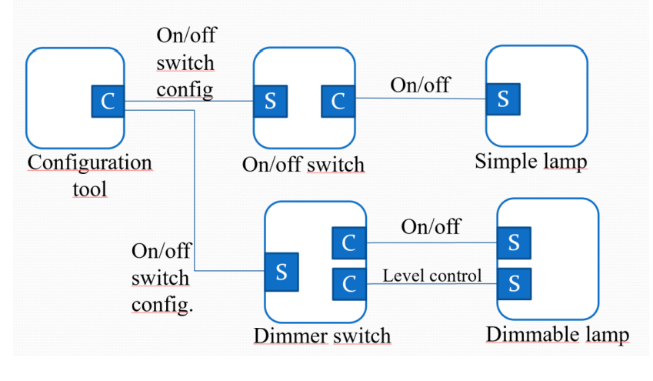
* Il dispositivo in cui sono memorizzati gli attributi è detto server del cluster.
* Il dispositivo che manipola gli attributi utilizzando determinati comandi è detto client del cluster.

**Cos’è un comando e di che tipo può essere?**

Un comando è un messaggio (formato da header e payload), che può servire a:

* Leggere e modificare gli attributi del server o effettuare service discovery sullo stesso.
* Configurare report e leggere le relative risposte, in base a periodi di tempo o minimi cambiamenti dell’attributo (Es. è il termometro stesso che in seguito alla configurazione manderà un report periodico al client sulla temperatura, ogni tot secondi oppure quando la temperatura cambia almeno di tot).

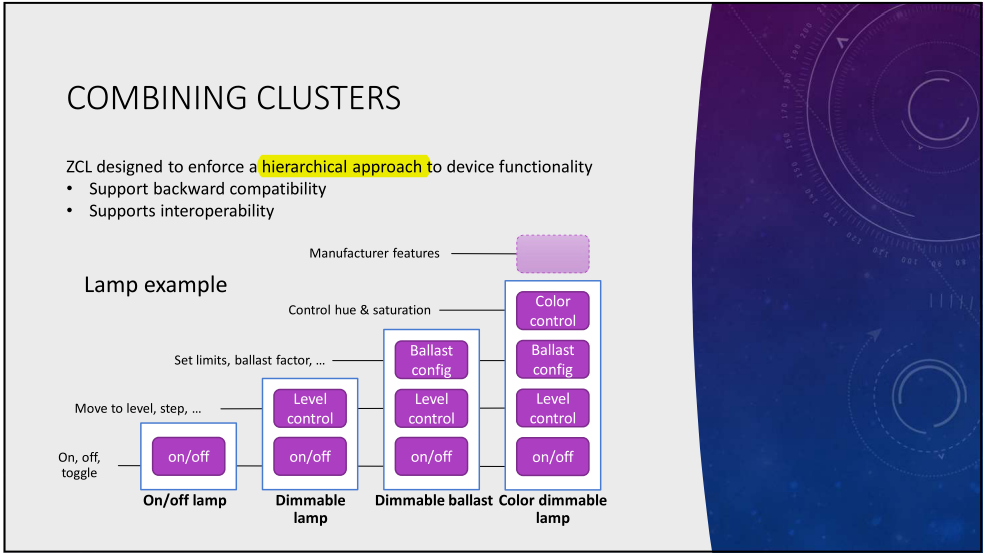
**DOMANDA 30**

****

Quest’immagine rappresenta un’applicazione di smart lighting. Abbiamo due lampade, una semplice e una dimmerabile, entrambe server del cluster “on/off”. Quella dimmerabile è anche server del cluster “level control”.

Gli interruttori vengono configurati tramite un configuration tool, che indica agli stessi che comportamento attuare e possibilmente permette di configurare la binding table.

**Approccio gerarchico nella combinazione dei cluster**

****

**ZDP (ZigBee Device Profile)**

Particolare profilo applicativo, che costituisce una specifica per l’implementazione del ZDO.

**Servizi che fornisce lo ZDO, in dettaglio**

* Device discovery: Permette a un disp.o di ottenere la lista degli indirizzi MAC e/o di rete che fanno parte della rete. Funziona secondo un’implementazione gerarchica: il coordinatore invia la richiesta ai router a cui è connesso e così via. Nella pratica, ad esempio, può servire durante la configurazione iniziale della rete.
* Service discovery: (Premessa: in ZigBee i servizi sono in pratica i cluster). Permette a un disp.o di effettuare delle query. Se la query è basata su un indirizzo di rete, vengono ritornati i cluster ID relativi ai cluster che il corrispondente dispositivo implementa; se la query è basata su un cluster ID, viene ritornata una lista degli endpoint che implementano tale cluster. Anche il service discovery funziona secondo un’implementazione gerarchica. Esempi pratici in cui viene utilizzato: un nuovo disp.o si aggiunge alla rete e quelli già presenti vogliono comprendere quali sono i servizi che esso fornisce; un termometro va giù e il termostato che attingeva ai suoi dati cerca nella rete se sono presenti altri termometri disponibili.
* Network management: implementa i protocolli dei dispositivi (es. network formation), in particolare quelli dei router e del coordinatore. Fa ciò invocando i servizi forniti dal livello di rete.
* Node management: implementa join/leaves dei nodi, accontenta richieste di NW discovery (?), recupera le tabelle di routing e di binding del disp.o.
* Binding management: aggiunge/cancella entries dalla binding table, generalmente in base a quanto indicato dall’umano che configura/effettua manutenzione sulla rete.

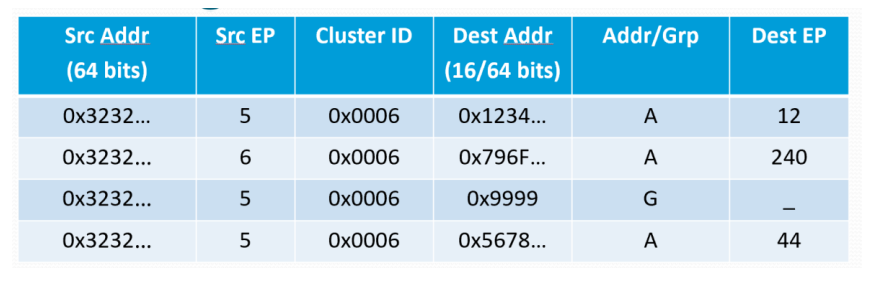
**In cosa si differenzia l’APS da un livello di trasporto?**

Dal fatto che utilizza anche delle informazioni semantiche (usa i concetti di endpoints, cluster IDs, profile IDs e device IDs).

**Quali sono i servizi forniti dall’APS?**

* Data service: trasmette i dati tra diversi disp.i. Applica filtri ai pacchetti (endpoint non registrati, profili che non corrispondono) e utilizza gli ACKs end-to-end.
* Binding service: permette un endpoint di un nodo di essere connesso unidirezionalmente a uno o più endpoint di un altro nodo.

**DOMANDA 29**



Questa tabella è una binding table.

Essa viene riempita dal ZDO del router o dei coordinatori (generalmente su indicazione dell’utente) ed è salvata nell’APS degli stessi.

Serve ad implementare l’indirect addressing, ovvero l’invio di messaggi nei quali la sorgente non deve necessariamente sapere qual è la destinazione.

Ad esempio, la prima riga indica che ogni volta che il client del cluster 0x0006, collocato sull’endpoint 5 del dispositivo con l’indirizzo MAC che inizia per 0x3232, invia un comando, esso sarà mandato al server dello stesso cluster, collocato sull’endpoint 12 del dispositivo con l’indirizzo MAC che inizia per 0x1234.

La lettera A significa che l’indirizzo identifica un singolo dispositivo, mentre con G si intende un indirizzo che identifica un intero gruppo.

**Altra cosa che si può notare è che nella tabella non sono inseriti gli indirizzi di rete, ma quelli MAC. Perché?**

Gli indirizzi MAC sono statici, mentre quelli di rete possono variare in base a disconnessioni/riconnessioni.

**Come si fa allora a recapitare il messaggio?**

I router e i coordinatori hanno nella propria memoria le cosiddette address maps, delle mappature che fanno corrispondere ad ogni indirizzo di rete un indirizzo MAC.

**Device IDs**

Sono ID che servono ad identificare il dispositivo dal punto di vista semantico (es. 0x0100 “on/off light”, 0x0101 “dimmable light”). La rete ZigBee potrebbe lavorare bene anche senza i device IDs. Essi esistono solo "per gli umani", nelle operazioni di sviluppo e manutenzione (es. il cluster on/off può definire sia un forno che una lampadina) o per effettuare ad esempio la visualizzazione di icone sui display.

Il device ID è quindi indipendente dal cluster ID.