Bluetooth – appunti del corso

Bluetooth è uno Standard Aperto per telecomunicazioni wireless (trasferimento di dati e voci) a corto raggio.

[Standard Aperto (Open Standard) vuol dire utilizzabile da chiunque liberamente. Le specifiche sono pubbliche]

Bluetooth è stata sviluppato nel 1994 (il nome deriva da Harald Blatant "Bluetooth", Re di danimarca, 940-981 A.D.)

La trasformazione di bluetooth in standard avviene a opera del Bluetooth Special Interest Group (SIG), fondato nel 1998 e costituito da Ericcson, IBM, Nokia, Intel, Toshiba. Ora più comprende di 1900 società.

Lo standard Bluetooth serve a realizzare le cosidette Personal Area Network (PAN), cioè piccole reti dell'estensione tipica di qualche metro. Per esempio, reti che collegano computer con varie periferiche (in sostituzione del cavo), reti dati e/o audio in singole abitazioni (home networking) [collegamento tra elettrodomestici intelligenti ("smart appliances"), dispositivi di riscaldamento, condizionamento, intrattenimento, e sistemidi allarme],

Bluetooth funziona in banda ISM 2.45 GHz (2400-2483.5 MHz), con una modulazione frequency hopping – spread spectrum (FHSS) per ridurre l'effetto di interferenze e fading (cammini multipli). Sono disponibili 79 canali FHSS. In ogni canale, per minimizzare la complessità del radiotrasmettitore si usa una modulazione binaria FM a 1Msimbolo/s = 1Mbit/s (lordo). Da standard, la distanza massima di comunicazione è 10 m (100 m con un trasmettitore potenziato).

Un ricetrasmettitore bluetooth è anche detto "dispositivo" bluetooth.

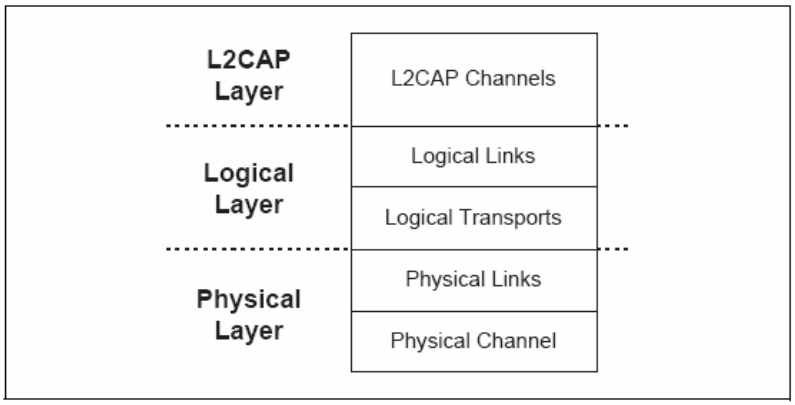
Tipicamente, un canale radio fisico è occupato da più dispositivi sincronizzati su una stessa sequenza di frequency hopping. Tale insieme di dispositivi si chiama "piconet". Un dispositivo, detto "master", è responsabile della sincronizzazione. Gli altri sono "slave".

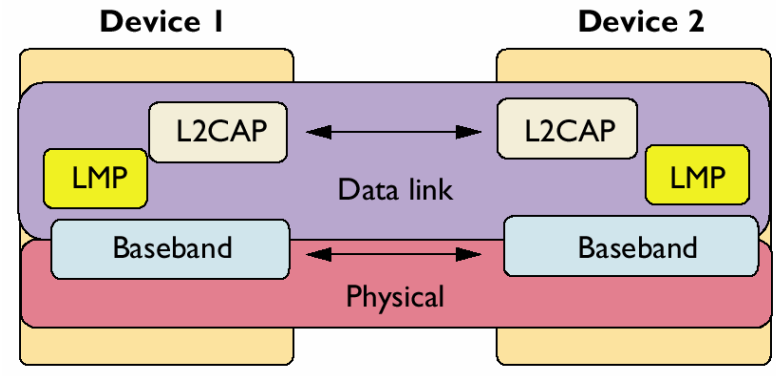
Lo standard è pensato per essere a basso costo (< 10$ per transceiver). Nelle intenzioni, un collegamento radio Bluetooth non dovrebbe costare più di un equivalente collegamento con cavo (per es. USB).

Una rete di Bluetooth è organizzata in strati (livelli). Sopra al canale radio fisico abbiamo una serie di canali e collegamenti (link), e i relativi protocolli. Partendo dal canale radio fisico, abbiamo il livello fisico, composto dal canale fisico (physical channel) e dal collegamento fisico (physical link), il livello logico, composto dal trasporto logico (logical transport) e da collegamento logico (logical link), poi il livello "L2CAP".

Il livello fisico e buona parte del livello logico sono descritti dalle specifiche della "banda base bluetooth" (bluetooth baseband). Tali livelli non corrispondono esattamente agli strati del modello ISO-OSI. Nella figura a colori in basso si puo' vedere che la bluetooth baseband è intermedia tra il livello fisico e il livello collegamento dati (data link) del modello ISO-OSI. Del livello fisico ISO-OSI fanno ovviamente parte le specifiche radio.

La parte più alta del livello logico è descritta dal Link Manager Protocol (LMP) [Protocollo di gestione dei collegamenti], che, facendo un parallelo con il modello ISO-OSI, è parte del livello collegamento-dati (di cui fa parte anche il livello L2CAP).





Specifiche Radio

Bluetooth opera in una banda affollata da altri sistemi di comunicazione (la banda ISM è non regolata). Per ridurre gli effetti delle interferenze, si deve mettere in atto una

tecnica di spread spectrum, che consenta di utilizzare al meglio tutta la banda disponibile (83.5 MHz). In particolare, si implementa una tecnica di "frequency hopping", cioè di variazione della frequenza di trasmissione secondo una particolare sequenza di salto (hopping) pseudocasuale specifica per ciascuna piconet.

Il frequency hopping è una tecnica di cosidetta "collision avoidance". Variando la frequenza di trasmissione su tutta la banda ISM si puo' evitare con alta probabilità l'interferenza di altri sistemi di comunicazione a banda stretta fissa o in frequency hopping. In altre parole, si ha collisione solo quando la frequenza di trasmissione si sovrappone alla banda utilizzata dal sistema "interferente", cioè per un intervallo di tempo breve e per una frazione molto piccola del tempo di trasmissione.

Le frequenze di trasmissione possibili sono 79, f = 2402 + k MHz (k=0,...,78). C'è quindi una banda di sicurezza inferiore di 2 MHz, e una banda di sicurezza superiore di 3.5 MHz.

Il tempo è suddiviso in intervalli ("slot") di 625 μs. Durante uno slot viene trasmesso un "pacchetto". In alcuni casi (vedremo più avanti), è possibile trasmettere pacchetti della durata multipla di uno slot.

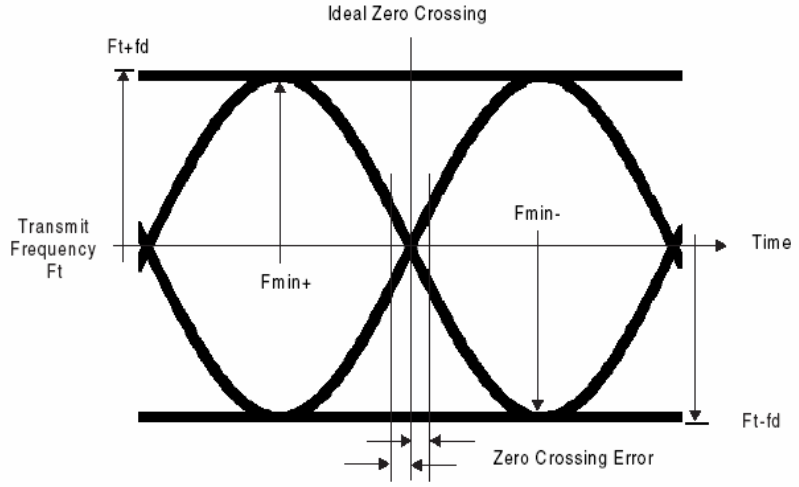
La frequenza di trasmissione viene cambiata, secondo una sequenza pseudocasuale, ogni volta che viene trasmesso un pacchetto. Nel caso tipico, in cui un pacchetto ha la durata di uno slot, la frequenza di trasmissione viene quindi cambiata ogni 625 μs ("hopping rate" = 1600 hop/s). Altri casi sono possibili.

La sequenza pseudocasuale di hopping viene determinata dal numero seriale del master.

Il trasmettitore puo' appartenere a una di 3 classi di potenza. La classe 3 corrisponde a una potenza massima del trasmettitore di 1 mW (portata massima del sistema di 10 m); la classe 1 a una potenza massima di 100 mW (portata massima di 100 m). Le antenne (sia trasmettitore, sia ricevitore) sono tipicamente omnidirezionali.

Power Class	Maximum Output Power (Pmax)	Nominal Output Power	Minimum Output Power[1]	Power Control
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	Pmin<+4 dBm to Pmax Optional: Pmin[2] to Pmax
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (-6 dBm)	Optional: Pmin[2)] to Pmax
3	1 mW (0 dBm)	N/A	N/A	Optional: Pmin[2)] to Pmax

La modulazone del segnale è GFSK [Gaussian Frequency Shift Keying] con prodotto banda-durata di ciascun bit BT = 0.5, e data rate di 1 Mbit/s. L'indice di modulazione tipico è 0.3 (0.28-0.35). La deviazione in frequenza f_d deve essere almeno 115 KHz. Il simbolo "1" corrisponde a una deviazione in frequenza positiva ($f_T + f_d$), il simbolo "0" a una deviazione in frequenza negativa ($f_T - f_d$).



Come abbiamo già detto , la modulazione binaria di frequenza permette di realizzare il ricevitore in modo particolarmente semplice (non deve essere lineare e ci sono solo due simboli), anche se ovviamente limita il bit rate.

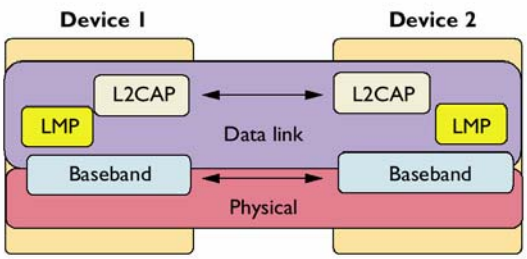
Il ricevitore puo' essere supereterodina con bassa frequenza intermedia (~3MHz) o omodina (o "zero IF", cioè conversione diretta da RF a banda video). In tutti e due i casi la sensibilità del ricevitore non è particolarmente spinta (nel primo caso la bassa frequenza intermedia impedisce di eliminare la banda immagine, nel secondo caso, l'amplificazione a RF non è particolarmente alta, per cui il rapporto S/N viene deteriorato).

Lo standard prevede una sensibilità del ricevitore di -70 dBm, per una BER (bit error rate) grezza (cioè escludendo meccanismi di correzione dell'errore) dello 0.1%. Confrontata con altre reti radio la sensibilità non è particolarmente spinta (per

esempio, per Wi-Fi la sensibilità è –90 dBm) e la cosa si paga in termini di minore portata.

Tipicamente si riesce a realizzare un ricetrasmettitore bluetooth in un unico chip in tecnologia CMOS, e quindi in modo economico (come indicato prima, < 10 $).

Bandabase Bluetooth (Bluetooth baseband) –

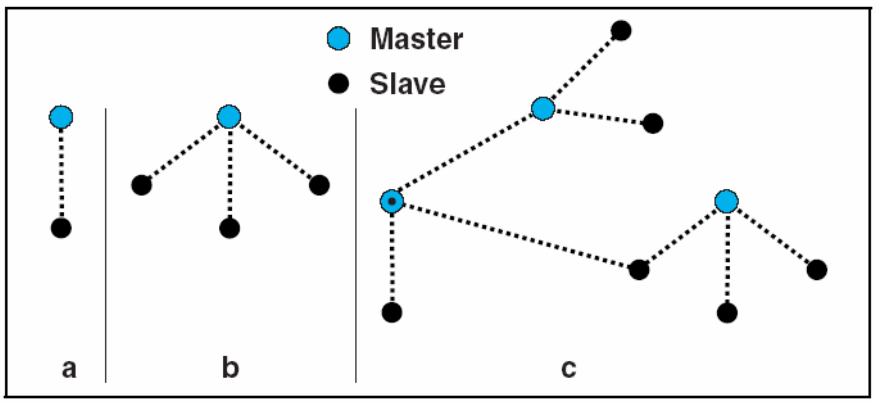


La rete elementare bluetooth si chiama "piconet". E' costituita da 2 a 8 radiotrasmettitori (**a** e **b** della figura in basso). Un dispositivo ha il ruolo di "master", gli altri sono "slave". Il collegamento fisico puo' essere realizzato solo tra il master e uno slave. Non è possibile avere un collegamento fisico tra due slave.

Da notare che l'architettura della rete è simmetrica, nel senso che ciascun radiotrasmettitore puo' fare sia da master, sia da slave. Nel momento in cui una rete si forma, il primo dispositivo che partecipa alla rete prende il ruolo di master, gli altri quello di slave.

Un dispositivo puo' inoltre appartenere a piu' di una piconet (come slave o come master), ma puo' essere master solo di una, e in tal caso si chiama "bridge", e permette di unire piu' piconet in una "scatternet".(in **c** della figura in basso tre piconet sono unite in una scatternet).

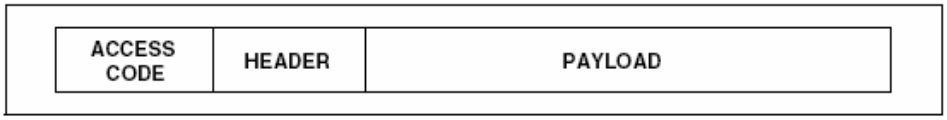
Un bridge fa parte di piu' piconet a suddivisione di tempo (time-sharing). In questo modo, è in grado di passare informazioni da una piconet all'altra, rendendo la scatternet connessa.



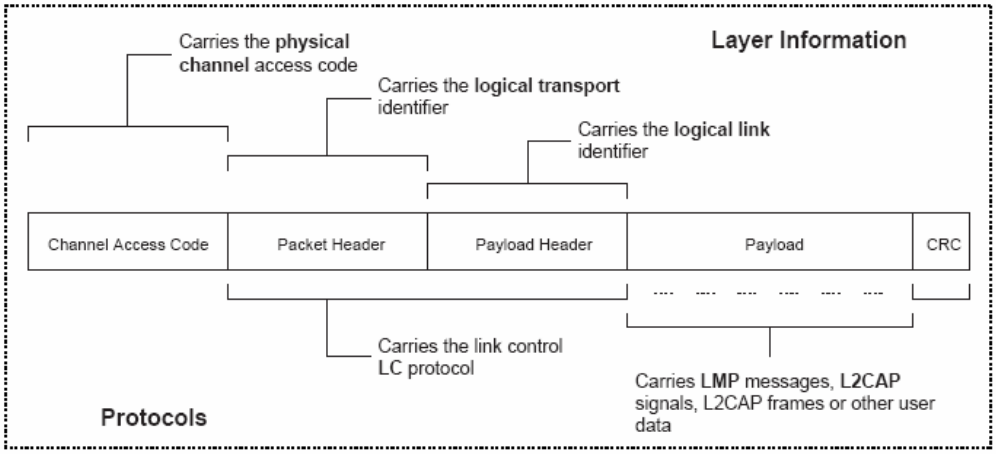
I vari ricetrasmettitori della stessa piconet usano il canale fisico a suddivisione di tempo. Nel caso in cui ad ogni slot viene cambiata la frequenza di trasmissione, il master trasmette negli slot dispari, e lo/gli slave negli slot pari.

I dati sono trasmessi in pacchetti, ciascuno costituito da tre parti:

- Codice di accesso (access code): 72 bit
- Intestazione del pacchetto (packet header): 54 bit
- Carico pagante (payload): da 0 a 2475 bit



Il Payload della figura in alto è suddiviso in basso in un payload header, in un carico utile vero e proprio (di nuovo chiamato payload), in una stringa CRC.



Sono possibili diversi access code, derivati dal codice identificativo (ID) bluetooth del master o dello slave(ogni dispositivo bluetooth ha un codice identificativo di 48 bit, di cui 24 identificano il costruttore, e 24 un numero seriale assegnato dal costruttore).

Il pacchetto è accettato da un dispositivo nella scatternet solo se l'access code contiene l'ID del master.

Il Packet header contiene una serie di informazioni sul pacchetto:

- 3 bit: slave address sulla piconet.
- 1 bit: positive o negative ack (ACK/NACK)
- 1 bit: flow (flow = 0 STOP nei pacchetti ACL)
- 1 bit: sequence number (non ci interessa)
- 4 bit: distingue tra i 16 tipi possibili di Payload:
  - NULL codice di accesso + packet header
  - ID c'è solo il codice di accesso
  - POLL costringe gli slave a responder
  - FHS frequency hopping syncronization
  - + 12 pacchetti sincroni asincroni

  - (tipo 1: 1 slot; tipo 2: 3 slot; tipo 3: 5 slot)
- 8 bit CRC dell'header

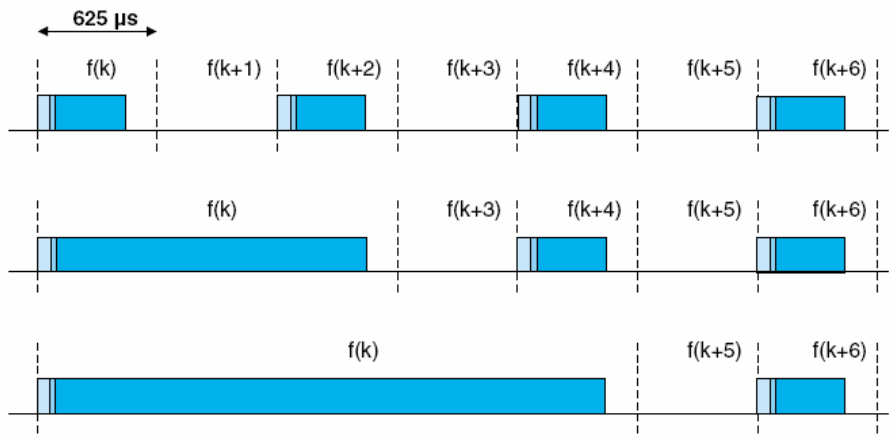
= 18 + altri 36 di ridondanza [FEC 1/3]

FEC sta per "forward error correction". In FEC 1/3 ogni bit viene trasmesso tre volte consecutivamente. Al ricevitore ogni terna di bit viene interpretata a maggioranza. In questo modo si riesce a correggere un errore singolo per ogni terna di bit.

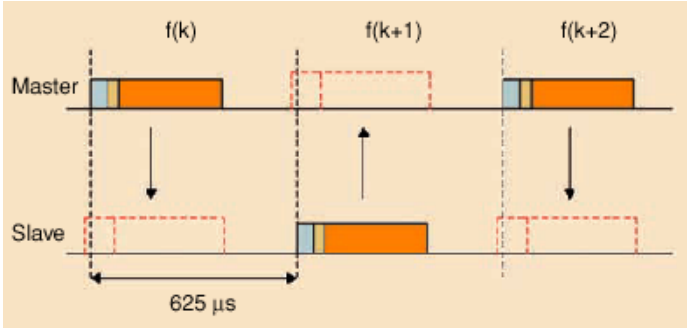
Il payload ha un numero di bit variabile in funzione della durata del pacchetto. Il pacchetto puo' durare 1, 3, o 5 slot. Nella figura in basso sono mostrati i tre esempi possibili. In azzurro sono indicati i pacchetti trasmessi dal master. Gli slave possono trasmettere negli slot rappresentati in figura come vuoti.

Si ricorda che la frequenza di trasmissione viene variata solo alla fine della trasmissione di un pacchetto intero.

Come indicato nella figura in basso, se il pacchetto dura cinque slot, alla fine del pacchetto si salta direttamente alla frequenza prevista dopo 5 slot, in modo da non perdere la sincronia con la sequenza di hopping.



Le collisioni sono evitate perché il master decide quale slave deve rispondere, e lo indica nel packet header.



Il master ha anche il compito di suddividere la banda disponibile tra i diversi slave.

Ogni volta che viene modificata la frequenza vanno persi circa 255 μs dello slot, necessari per la variazione di frequenza. Quindi, se il pacchetto è di uno slot, solo circa 370 μs, sono usati per la trasmissione, corrispondenti a 370 bit. Togliendo i 72 per l'access code e 54 per il packet header, abbiamo circa 244 bit, cioè circa 30 byte.

Bluetooth consente simultaneamente trasmissione voce (comunicazioni bidirezionali) e trasmissione dati. Ci sono quindi diversi collegamenti possibili.

Collegamenti in bandabase (BASEBAND LINK)

Ci sono due tipi principali di link:
Link Sincroni (SCO Synchrounous Connection Oriented)
Link Asincroni (ACL Asynchronous ConnectionLess)

I link ***SCO*** sono usati per trasmissione voce. Sono connessioni simmetriche punto-punto che riservano i time-slot per garantire la trasmissione in tempo reale.

Il dispositivo slave ha sempre il permesso di rispondere nel time-slot immediatamente successivo a una trasmissione SCO del master.

Un master puo' supportare fino a 3 link SCO con uno o più slave. Uno slave puo' supportare al più 2 link a MASTER differenti.

I Pacchetti SCO non sono mai ritrasmessi. Se non vanno a buon fine vanno persi.
I possibili collegamenti SCO sono indicati nella tabella qua sotto.

Type	Payload Header (bytes)	Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)
HV1	na	10	1/3	no	64.0
HV2	na	20	2/3	no	64.0
HV3	na	30	no	no	64.0
DV[1]	1 D	10+(0-9) D	2/3 D	yes D	64.0+57.6 D
EV3	na	1-30	No	Yes	96
EV4	na	1-120	2/3	Yes	192
EV5	na	1-180	No	Yes	288

Vediamo brevemente le esigenze di un collegamento voce bidirezionale. La qualità della comunicazione deve essere ISDN, cioè un segnale audio di tipo telefonico campionato a 8 KHz, con 8 bit per campione, per una banda totale di 64 Kbps.

Per effettuare la comunicazione voce in tempo reale devono essere trasmessi 8 Kbyte al secondo, cioè 10 byte ogni 1.25 ms.

Poiché nel caso migliore, un pacchetto di uno slot ha 30 byte di carico utile (modalità HV3 della tabella in alto), è sufficiente uno slot ogni 3.75 ms, cioè uno ogni 6 slot trasmessi.

Per questo motivo, considerando la comunicazione bidirezionale, 3 comunicazioni voci simultanee saturano completamente il canale radio (ciascuna comunicazione bidirezionale occupa 2 slot ogni 6).

La modalità HV3 puo' essere impiegata se il canale è in buone condizioni, cioè se senza fare correzione degli errori si ha una comunicazione accettabile. Se il canale è occupato o disturbato bisogna dedicare parte della banda alla correzione anticipata degli errori.

La modalità HV2 usa FEC 2/3 [Forward error correction con 2/3 di carico utile]
In ogni pacchetto abbiamo 20 byte di carico utile e 10 byte di ridondanza. Non ci interessa come sono generati i bit di ridondanza (si tratta di un Hamming code abbreviato). In questo caso per mantenere il funzionamento in tempo reale con collegamenti di 64 Kbit/s abbiamo bisogno di uno slot ogni 4, quindi il sistema puo' garantire al più due collegamenti voce bidirezionali.

In condizioni ancora peggiori si usa la modalità HV1, con FEC 1/3 (ripetizione tripla di ogni bit). In questo caso abbiamo un carico utile di 10 byte + 20 byte di ridondanza (HV1), e quindi un collegamento voce ha bisogno di uno slot ogni 2 (1.25 ms), e quindi è sostenibile un solo collegamento voce.

I link **ACL** (Asynchronous ConnectionLess) sono usati per trasmissioni dati. La trasmissione su questi link è stabilita su una base per-slot (negli slot non riservati ai link **SCO**).

I link ACL supportano trasferimenti da punto a multipunto (multicast).

Dopo una trasmissione ACL da un MASTER, solo i dispositivi slave indirizzati possono rispondere nel time slot successivo. Se non è stato indirizzato nessun dispositivo il messaggio è considerato "broadcast", cioè indirizzato a tutti.

Tutti i tipi di link ACL includono la possibilità di ritrasmettere i pacchetti.

La comunicazione dati puo' essere simmetrica o asimmetrica.

Nel primo caso (modalità DM1 e DH1 – vedi sopra) un pacchetto ha la durata di uno slot. Nel caso DH1 dei 30 byte del payload abbiamo

- Intestazione del payload (1 byte)
- Carico utile (27 byte)
- CRC (2 byte)

Abbiamo quindi una trasmissione di 27 byte netti ogni 2 slot (1.25 ms):
27byte/1.25 ms = 172.8 Kbps

Nel caso DM1 (tipicamente usato in peggiori condizioni del canale) usiamo una FEC 1/3:

- Intestazione del payload (1 byte)
- Carico utile (17 byte)
- CRC (2 byte)
- FEC 1/3 (10 byte)

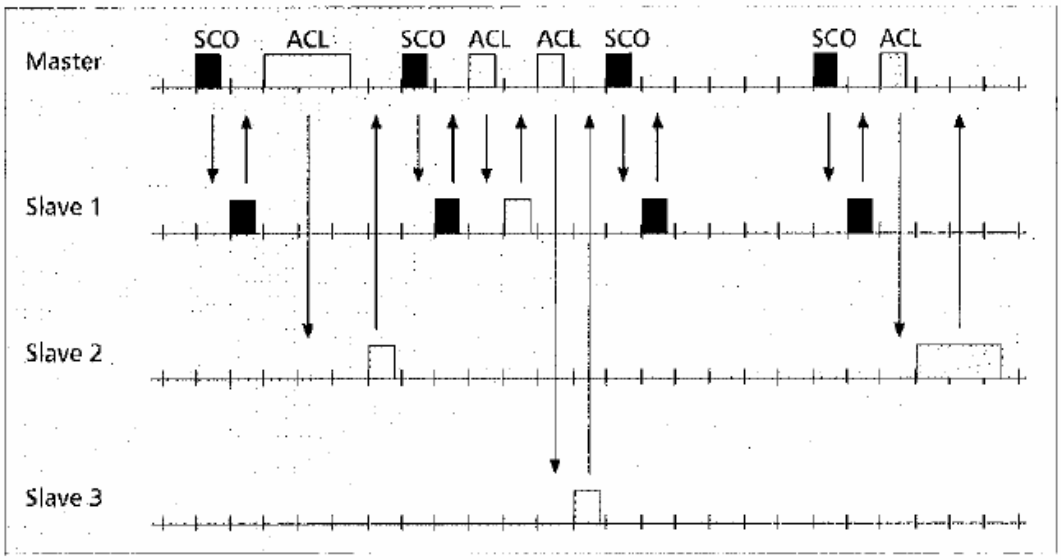
Nel caso asimmetrico i pacchetti sono diversi nelle due direzioni. Per esempio, nella modalità DH5, vengono alternativamente trasmessi un pacchetto da 5 slot e uno da 1.

Il pacchetto da 5 slot ha un payload della durata di 4 slot interi in più rispetto al pacchetto da 1 slot. In 4 slot interi (2.5 ms) stanno 2500 bit = 312.5 byte in più, quindi il payload ha complessivamente 244+2500 = 2744 bit (343 byte), di cui

- Intestazione del payload (2 byte)
- Carico utile (339 byte)
- CRC (2 byte)

Il pacchetto viene trasmesso ogni 6 slot. Il data rate nella direzione più veloce è quindi: 339*8 bit/3.75 ms = 723.2 Kbps. Questo è il massimo datarate.

Nell'altra direzione viene trasmesso un pacchetto da uno slot (27 byte utili) ogni 6 slot, cioè: 27*8 bit/3.75 ms = 57.6 Kbps. Il sistema puo' tenere aperto contemporaneamente canali voce e dati, a suddivisione di tempo. Un esempio della sequenza, con un canale voce (SCO) e uno o più dati (ACL), è mostrato nella figura in basso.



Operazioni del link controller, la macchina a stati che controlla i collegamenti.

Vediamo come viene stabilita una piconet, e quali sono gli stati possibili per ogni dispositivo della piconet. Il diagramma degli stati impiegati dal link controller è mostrato nella figura successiva.

Abbiamo tre stati principali:

- STAND-BY,
- CONNECTION,
- PARK.

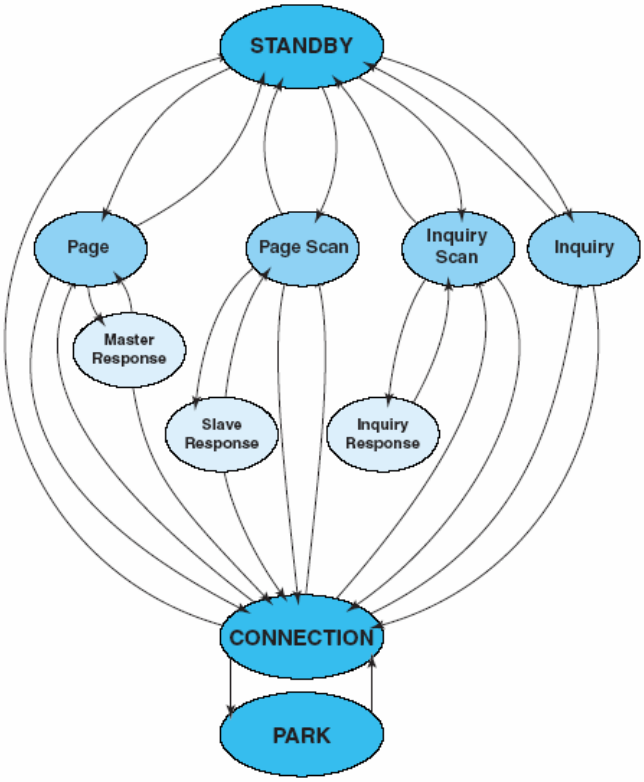
Inoltre abbiamo altri sette "sottostati", cioè stati transitori in cui i dispositivi si vengono a trovare per attivare la connesione o permettere la scoperta di nuovi dispositivi:

- PAGE,
- PAGE SCAN,
- INQUIRY,
- INQUIRY SCAN,
- MASTER RESPONSE,
- SLAVE RESPONSE,
- INQUIRY RESPONSE.

Il passaggio da uno stato all'altro avviene a seguito di comando dal livello superiore (link manager) o comandi interni del link controller.

Stato STAND-BY:
E' lo stato di default del dispositivo, ed è un "low power mode", cioè una modalità di funzionamento a basso consumo. Solo il clock interno funziona in modo continuativo. Ogni 1.28-3.84 secondi il dispositivo si mette in ascolto, cioè passa nello stato PAGE SCAN o INQUIRY SCAN tipicamente per 10 ms (cioè 16 slot). Puo' inoltre uscire dallo

stato STANDBY per cominciare la ricerca di altri dispositivi, andando nello stato PAGE o INQUIRY.



Stati possibili del link controller (Fig. 8.1. di [1])

Stato PAGE SCAN:
In questo stato il dispositivo si pone in ascolto (cioè accende il ricevitore) per una finestra temporale della durata tipica 10 ms, per rispondere eventualmente a un altro dispositivo che sta cercando di formare una nuova connessione. La frequenza di funzionamento è costante durante ciascuna finestra, e tra una finestra e l'altra viene cambiata seguendo la sequenza di page hopping ("page hopping sequence"), che dipende dal proprio ID bluetooth. Passato il tempo di ascolto il dispositivo torna nello stato di provenienza, che puo' essere STAND BY o CONNECTION. Se durante la finestra temporale riceve il messaggio "page" inviato da un altro dispositivo, va nello stato SLAVE RESPONSE e comincia la procedura di attivazione della connessione, di cui parliamo più avanti.

Stato PAGE:
E' usato da un dispositivo per attivare una nuova connessione con uno slave nello stato PAGE SCAN. Il dispositivo diventerà il master di una nuova piconet, se si tratta della prima connessione, o è già il master della piconet esistente. Per questo chiamamo già "master" il dispositivo nello stato page.

Il master deve conoscere l'ID (il numero seriale bluetooth) dello slave con il quale vuole stabilire la connessione. Tale numero puo' essere conosciuto per la storia

precedente (noto da precedenti collegamenti, o dal programma di applicazione), o puo' essere stato determinato attraverso una precedente fase di "inquiry".

Negli slot dispari il master trasmette due messaggi "page", contenenti solo l'access code dello slave (della durata di 68 μs) variando la frequenza di trasmissione ogni metà slot, secondo la sequenza di page hopping. Se lo slave desiderato è nello stato PAGE SCAN e sta ascoltando alla frequenza di trasmissione, questi risponde e si puo' stabilire la connessione.

La sequenza di page hopping è una sequenza pseudocasuale che comprende solo 32 delle frequenze possibili. Nella finestra temporale di ascolto del PAGE SCAN ci sono 16 slot, cioè 8 slot dispari, durante i quali il master trasmette il comando "page" a 16 frequenze diverse, delle 32 possibili. La probabilità di stabilire il contatto con il nuovo slave durante una finestra temporale è del 50%.

Nella prima (seconda) figura qua in basso è mostrata la sequenza dei messaggi quando lo slave riceve il comando page nella prima (seconda) metà dello slot.

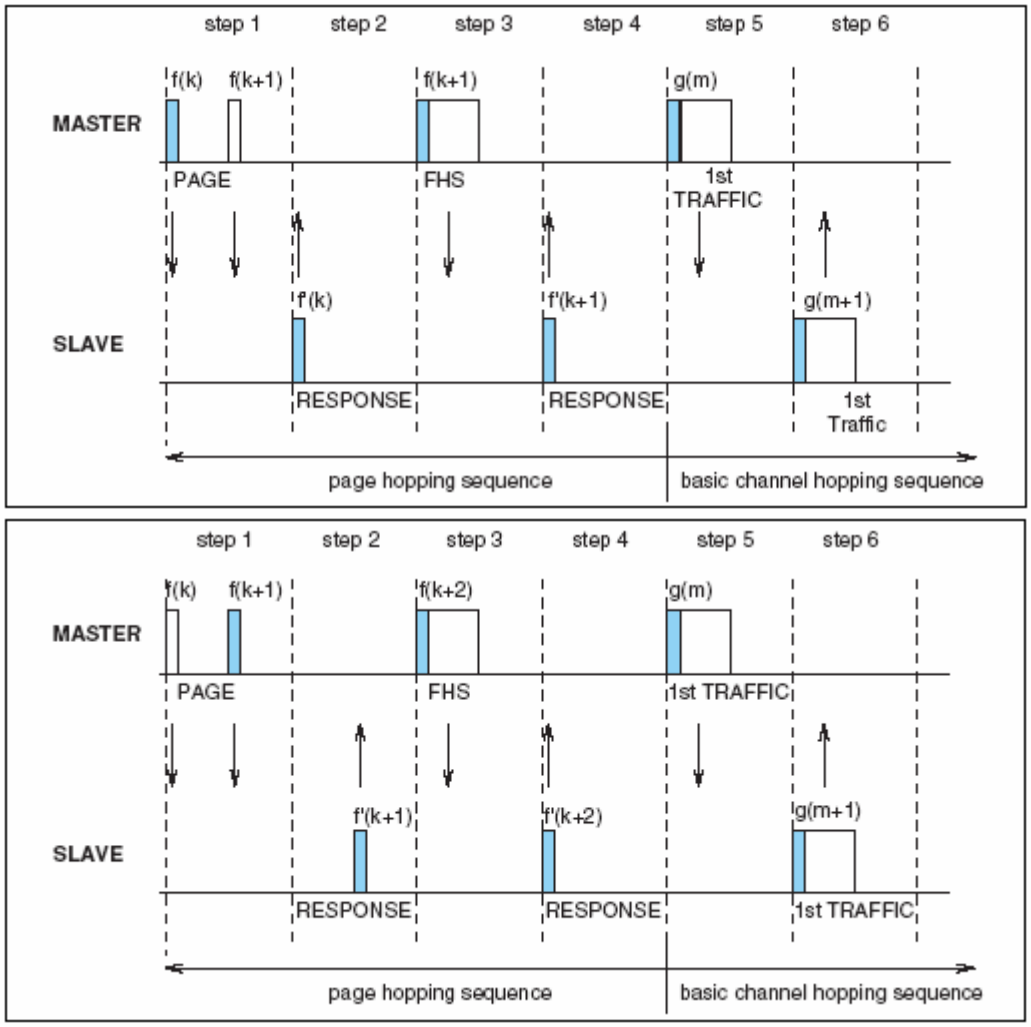


Fig. 8.3 e 8.4 di [1]

Nello step 1 il master è nello stato PAGE, lo slave nello stato PAGE SCAN. Nello step 2 lo slave è nello stato SLAVE RESPONSE, il master nello stato PAGE. Nello step 3 il master è nello stato MASTER RESPONSE.

Step	Message	Packet Type	Direction	Hopping Sequence	Access Code and Clock
1	Page	ID	Master to slave	Page	Slave
2	First slave page response	ID	Slave to master	Page response	Slave
3	Master page response	FHS	Master to slave	Page	Slave
4	Second slave page response	ID	Slave to master	Page response	Slave
5	1st packet master	POLL	Master to slave	Channel	Master
6	1st packet slave	Any type	Slave to master	Channel	Master

Tabella 8.3 di [1]

Stati SLAVE RESPONSE e MASTER RESPONSE:
Nello step 2 lo slave risponde inviando un pacchetto di acknowledgment con solo l'access code (cioè il proprio ID), esattamente 625 μs dopo il page a cui risponde (in quel momento il MASTER riceve alla stessa frequenza del page corrispondente). Quando il master riceve tale pacchetto va nello stato MASTER RESPONSE e invia il comando FHS (frequency hopping sequence) che fornisce le informazioni allo slave per sincronizzarsi sulla sequenza di hopping principale (in particolare, contiene l'ID del master, informazioni sulla fase del clock, e indica da quale punto della sequenza di hopping si parte). Lo slave manda un'altra risposta per confermare la ricezione di FHS. Nello step 5 master e slave sono entrambi nello stato CONNECTION, e cominciano a trasmettere seguendo la sequenza di hopping normale. Il master trasmette il suo primo pacchetto utile nello step 5, lo slave nello step 6, e così via.

Stato INQUIRY:
E' lo stato in cui un dispositivo verifica se ci sono altri dispositivi bluetooth nelle vicinanze. Trasmette due volte per slot il messaggio "inquiry", contenente solo un codice d'accesso (senza il proprio ID) con la sequenza di page hopping. Negli slot pari, ascolta i dispositivi che rispondono con un "inquiry response" e raccoglie gli ID. Se, successivamente vuole stabilire una connessione con questi dispositivi, si sposta nello stato page.

Un dispositivo accede allo stato INQUIRY da STANDBY o da CONNECTION. Dopo un certo tempo, determinato dai livelli superiori, torna nello stato di origine

Stato INQUIRY Scan.
E' praticamente identico allo stato page SCAN. Se riceve un messaggio di inquiry, passa nello stato INQUIRY RESPONSE, in cui, dopo 625 μs invia un messaggio di risposta in cui specifica il proprio ID, e informazioni sulla fase del proprio clock

Stato CONNECTION:
E' lo stato tipico della piconet, che permette le comunicazioni bidirezionali tra master e slave. Nello stesso stato un dispositivo puo' sospendere temporaneamente la connessione. Nello stato Connection, un dispositivo puo' essere in tre modi di funzionamento: ACTIVE MODE, SNIFF MODE, HOLD MODE.

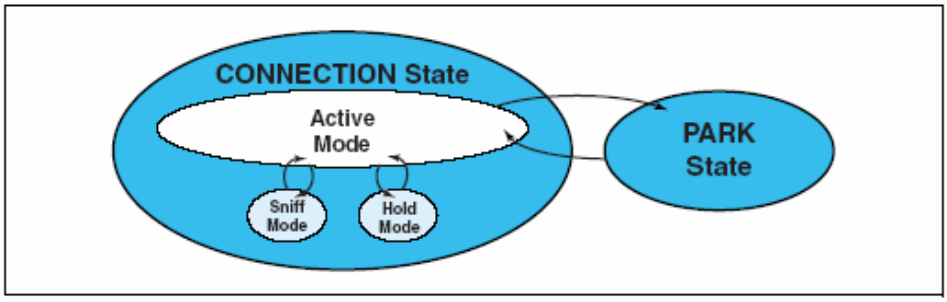
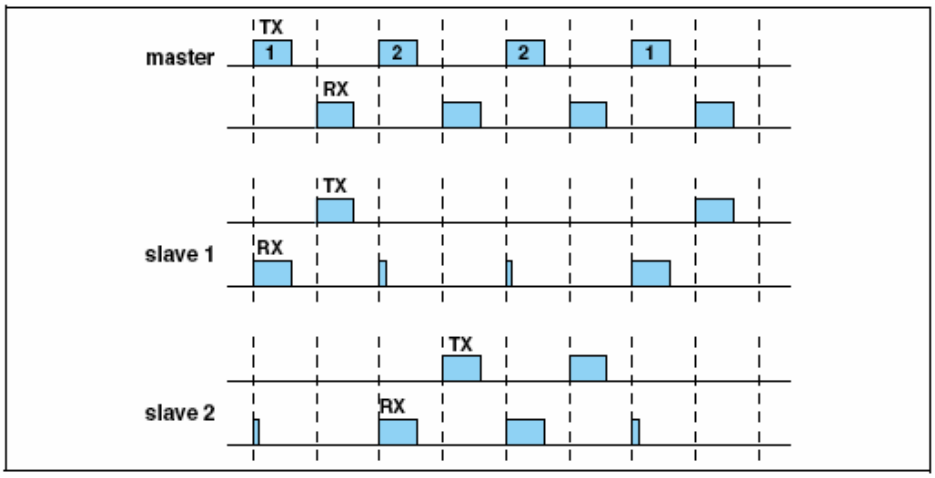


Fig. 8.5 di [1]

Possono essere nel modo attivo al più il master e sette slave. I dispositivi attivi supportano la comunicazione bidirezionale. Il master distribuisce la banda tra i vari slave, e consente loro di rispondere. Nel caso di comunicazione con più slave, ciascuno slave ascolta il master negli slot dispari (master-to-slave), e smette di ricevere se l'intestazione del pacchetto non contiene il suo numero identificativo nella piconet. Solo lo slave effettivamente indirizzato ascolta tutto il pacchetto, e risponde nello slot successivo. In questo modo si riesce a ridurre il consumo di potenza. Un esempio di temporizzazione di una piconet con due slave e' mostrato nella figura seguente.



(Fig. 8.6 di [1])

Nello stato attivo si puo' cambiare il master della piconet.

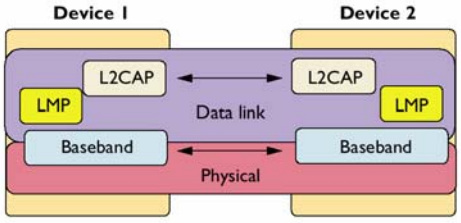
I modi SNIFF e HOLD permettono un risparmio di potenza. Nel modo Sniff il duty cycle di attività di un dispositivo slave viene ridotto di una quantità determinata in accordo tra master e slave. In pratica il master o lo slave invia il comando "sniff" tramite il link manager, con argomento N_{sniff} intero. Da quel momento, lo slave ascolta un slot dispari (master-to-slave) ogni N_{sniff} slot (da decidere tramite accordo con il master). Se viene indirizzato risponde nello slot successivo, altrimenti spegne il ricevitore e lo riaccende dopo N_{sniff} slot. Il master, a sua volta, sa a quali slot puo' interrogare lo slave in modo sniff.

Il master puo' ordinare a un dispositivo in modo attivo (anche se stesso) di passare al modo HOLD per un tempo prefissato T_{hold}. Durante questo tempo, il dispositivo non ascolta più quello che viene trasmesso sui link ACL (continua a mantenere eventuali link SCO), ma continua a seguire la sequenza di hopping. Mentre è in HOLD è libero di fare paging, inquiry, o comunicare su un'altra piconet (se fa da bridge).

Stato PARK:
Nello stato PARK uno slave esce dalla piconet, e mantiene solo la sincronizzazione con la sequenza di hopping. Il dispositivo lascia libero l'indirizzo nella piconet (da uno a sette) e prende un indirizzo di parcheggio. E' uno stato che consente eventualmente di aumentare in modo virtuale oltre il numero di massimo di 8 dispositivi in una piconet.

Il dispositivo nello stato PARK non ascolta più. Si risveglia a intervalli periodici per ascoltare eventuali messaggi dal master. Se il master vuole risvegliare o inviare un comando a un dispositivo parcheggiato invia un "beacon", cioè un treno di pacchetti identici ripetuti un numero di volte sufficiente a garantire che il dispositivo parcheggiato si svegli almeno una volta e ascolti.

Link Manager (LM) e Link Manager Protocol (LMP)



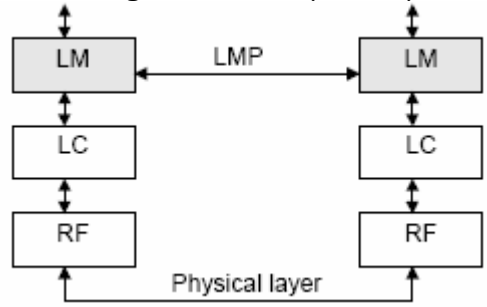
La macchina a stati in bandabase è controllata dal **LINK MANAGER**, che è un FIRMWARE (un software contenuto nella ROM e contenente routine e primitive di basso livello) fornito insieme all'hardware di controllo del link.

Il link manager stabilisce la connessione, la controlla, e ne garantisce la sicurezza. Ha le seguenti funzioni:

- Autenticazione della comunicazione sul link e sicurezza
- Monitoraggio della qualità del servizio (QoS Quality of Service)
- Controllo della banda base Bluetooth
- Controllo e gestione del paging
- Controllo e gestione dello stato degli slave
- Controllo del cambiamento di master nella piconet.
- Gestisce i pacchetti multislot e gestisce l'allocazione degli slot tra i vari link.

Il master ha il compito di dividere il tempo del collegamento tra i vari ACL (facendo polling (interrogandoli) più spesso o usando pacchetti più o meno lunghi).

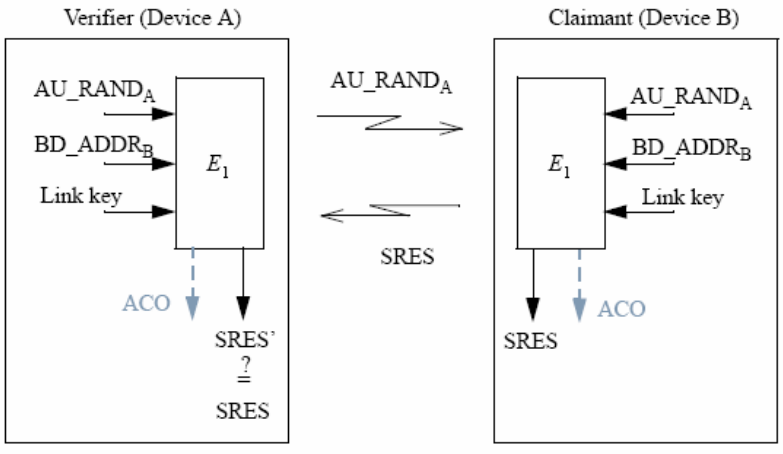
I link manager comunicano tra di loro tramite il **Link Management Protocol** (LMP) che usa i servizi sottostanti di bandabase, e controlla e negozia tutte gli aspetti della messa in opera e gestione dei collegamenti tra più dispositivi.



Il LMP consente la comunicazione ai link manager di due dispositivi connessi da un link ACL. Tutti i messaggi e i comandi LMP sono trasmessi come carico utile dei pacchetti ACL, differenziati tramite l'header ACL dai messaggi e comandi del protocollo **L2CAP** (logical link control and adaptation protocol).

Vediamo più in dettaglio l'autenticazione, che avviene seguendo uno schema "challenge-response". Il dispositivo che si vuole far autenticare è il richiedente ("claimant"). Il dispositivo che verifica l'autenticità si chiama verificatore ("verifier").

L'autenticazione consiste nel verificare in due passi – tramite chiavi segrete simmetriche - che il richiedente sia a conoscenza di una chiave segreta.



L'autenticazione ha successo se il verificatore e il richiedente condividono la stessa chiave K, nel nostro caso la "link key", generata appositamente per quel collegamento.

- Il verificatore (A) genera un numero casuale di 128 bit (AU_RAND_A) [la "challenge"] e lo invia al richiedente (B).
- B usa l'algoritmo E1 con argomenti AU_RAND_A, il proprio bluetooth address BD_ADDR_B, e la chiave segreta K (128 bit), e ottiene

  $$SRES = E_1(AU\_RAND_A, BD\_ADDR_B, K), \text{ di 32 bit.}$$

- B invia SRES ad A
- A verifica se SRES è uguale al valore SRES' generato internamente con la propria chiave K. Se le chiavi sono uguali e il BD_ADDR_B è giusto l'autenticazione è completata.

Se l'autenticazione deve essere mutua, dopo questo passo B puo' fare da verificatore e A da richiedente, e B puo' mandare una challenge AU_RAND_B ad A per la verifica.

Insieme a SRES, E1 genera anche ACO, che viene usata successivamente per la cifratura.

La chiave segreta del link viene generata a partire da una chiave di inizializzazione, che è generata all'inizializzazione del link, a partire da un PIN, l'indirizzo bluetooth dei dispositivi, e un numero casuale. Questa procedura s chiama pairing. Il PIN deve essere inserito dall'utente con una procedura che dipende dal particolare dispositivo bluetooth.

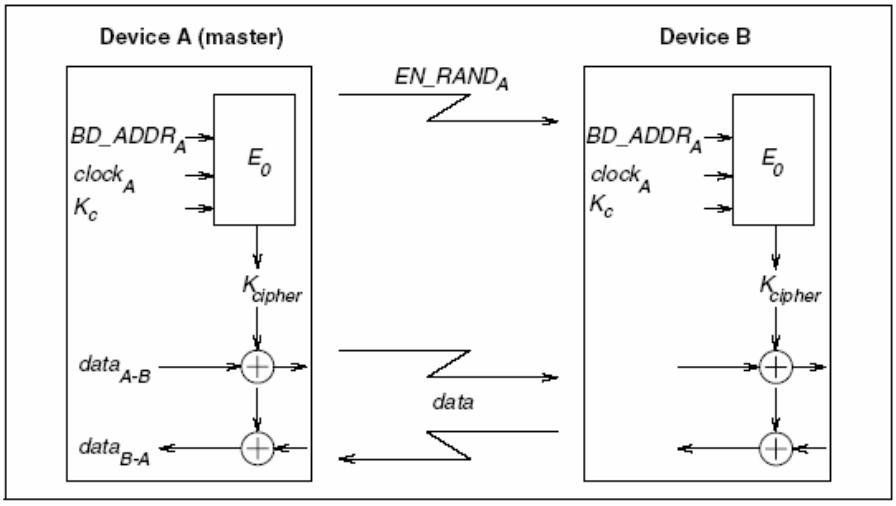
Una volta che la chiave di inizializzazione è generata, la chiave segreta del link viene generata con procedura di autenticazione mutua all'inizio del collegamento, a partire da un numero casuale che il master spedisce allo slave.

La cifratura puo' essere effettuata dopo che almeno una autenticazione ha avuto luogo. La cifratura consiste semplicemente una somma modulo 2 bit a bit tra il pacchetto da spedire in uno slot e una chiave altrettanto lunga (K_{cypher}).

La chiave viene generata con un algoritmo E_O che ha come argomenti: BD_ADDR_A, una chiave per la cifratura K_C e il valore del contatore (clock) del master (26 bit):
$K_{cypher} = K_{cypher}(BD\_ADDR_A, K_C, clock)$

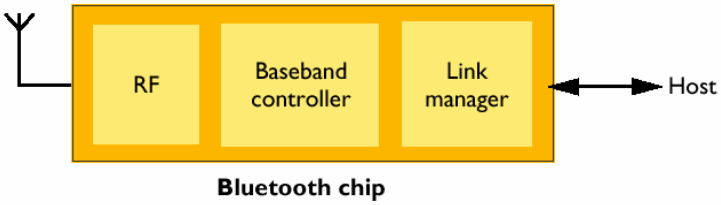
La chiave per la cifratura K_C viene ottenuta come funzione di un numero casuale EN_RAND_A di 128 bit, della corrente "link key", e della stringa ACO ottenuta in fase di autenticazione [$K_C = K_C(EN\_RAND_A$, link key, ACO)]

EN_RAND_A viene generato dal master e spedito in chiaro all'inizio dell'operazione di cifratura.

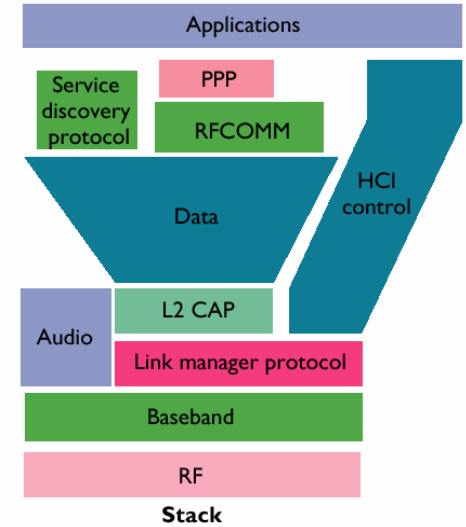


(Fig. 4.1 di [1])

Tipicamente la sezione RF, i circuiti di controllo della banda base Bluetooth e i link manager non contenuti in un unico chip bluetooth, che poi è inserito in un sistema "ospite". I livelli inclusi nel chip sono realizzati via hardware o firmware. I livelli superiori sono implementati tipicamente via software (tranne l'HCI).



Vediamo quindi i livelli superiori nello schema seguente:



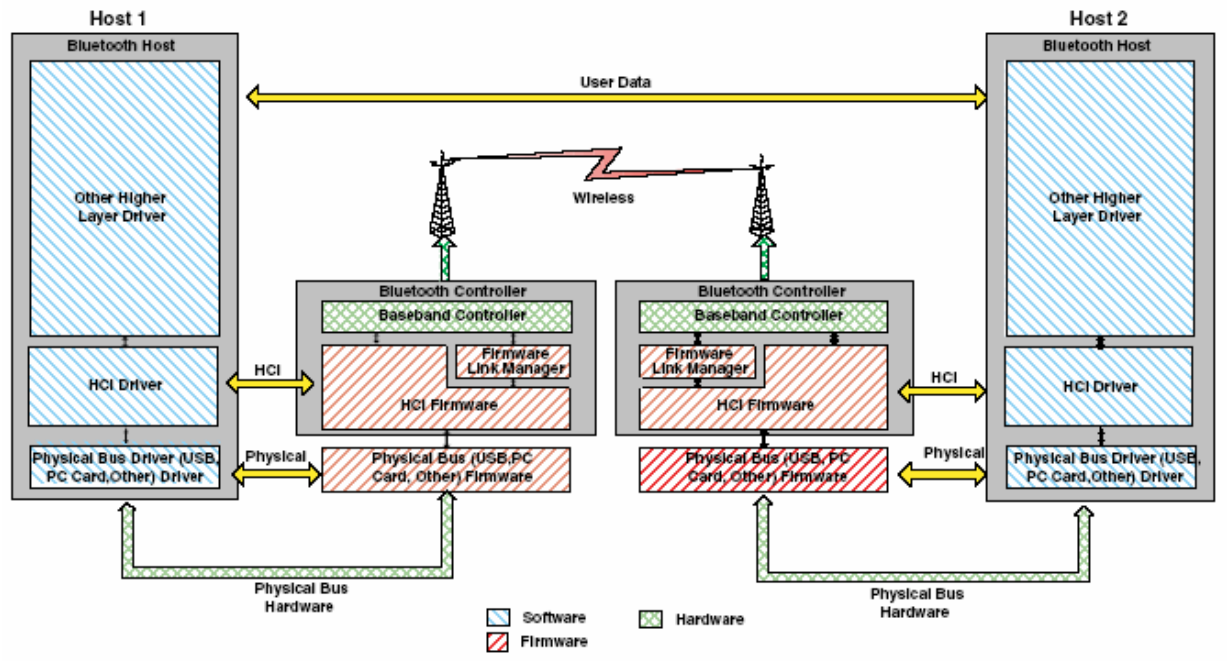
Dei livelli superiori all'LMP dedichiamo spazio solo a

- L2CAP (Logical Link Control and Adaptation Protocol), e
- HCI (Host Control Interface).

HCI consente l'interfacciamento del chip bluetooth con il sistema ospite attraverso gli standard via cavo più comuni, come USB, RS232, UART. E' realizzato tipicamente via firmware, o firmware/software.

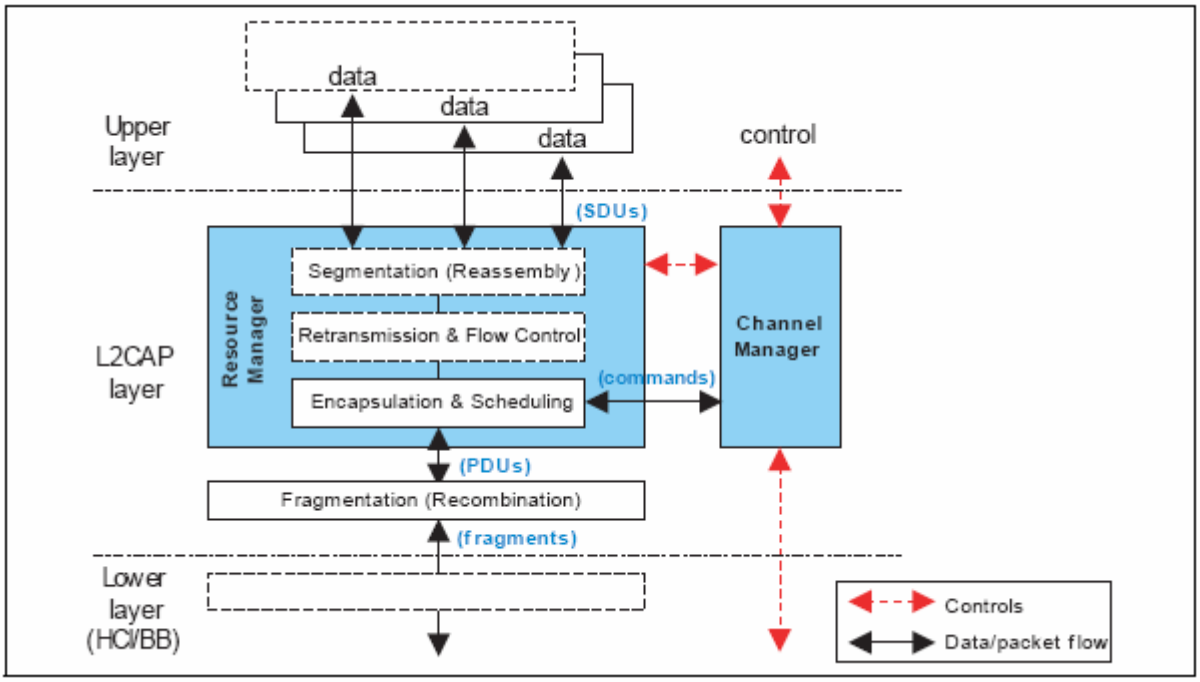
La struttura degli strati e dei messaggi nel caso di due sistemi ospiti collegati tramite schede bluetooth è illustrata nella figura seguente: Host 1 e Host 2 sono forniti di due schede bluetooth, collegate tramite un bus fisico, per esempio USB.

I due Driver HCI degli ospiti comunicano usando i servizi messi a disposizione dalla HCI. I dati vengono scambiati tra ospite e scheda bluetooth in accordo al protocollo del bus fisico, e tra le schede bluetooth come carico utile di un link ACL.



L2CAP è uno strato intermedio (del livello data link) che garantisce

- il multiplexing dei protocolli superiori;
- l'interoperabilità tra dispositivi bluetooth;
- gestione del gruppo mappando i protocolli di gruppo superiori sulle reti piconet (per esempio deve mappare sui protocolli inferiori la comunicazione tra due slave della stessa piconet);
- la segmentazione e riassemblamento dei pacchetti tra livelli;
- la negoziazione e monitoraggio della qualità del servizio (dei protocolli superiori).



I suoi messaggi costituiscono carico utile dei link ACL, con priorità inferiore ai pacchetti LMP (LMP deve garantire l'integrità del collegamento e ha quindi massima priorità).

I messaggi e comandi L2CAP vengono spediti su canali L2CAP, che a loro volta usano i link ACL esistenti (nota: un collegamento indica una connessione fisica (basso livello), un canale indica una connessione software (alto livello))

Abbiamo canali di tre tipi:

- canali di segnalazione bidirezionali (comandi)
- canali orientati alla connessione (CO – Connection Oriented) (connessioni bidirezionali punto-punto)
- conali unidirezionali senza connessione (CL - ConnectionLess) (connessione da punto a multipunto, che permettono a una entità L2CAP di essere collegata a un gruppo di dispositivi remoti).

Ogni canale L2CAP include due endpoint indentificati da un CID di 16 bit (CID Channel ID - Identificatore di canale logico).

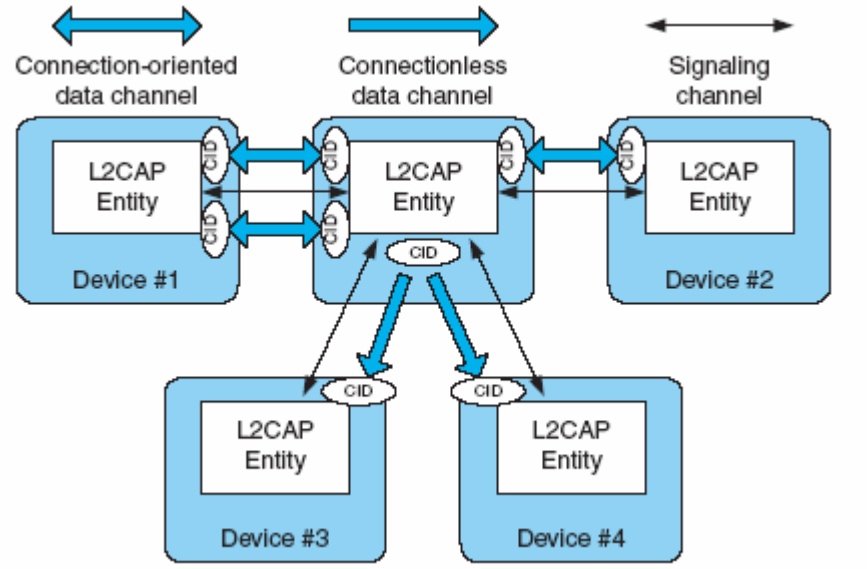
Il Canale di segnalazione ha CID = 0001

Ogni canale CO ha un CID locale allocato dinamicamente ed è legato ad uno SPECIFICO protocollo di livello superiore.

Durante la configurazione si negozia la QoS di ogni canale (banda di picco, tipo di trasmissione (best effort, garantita, no traffic)).

I canali connectionless sono unidirezionali e sono usati per formare gruppi.
Un endpoint CL outgoing puo' essere logicamente collegato a più endpoint CL ingoing, che formano un gruppo logico. Ogni entità L2CAP ha un solo endpoint CL ingoing (indirizzo CID = 0002). I canali CL non richiedono connessione o configurazione.

Un esempio della rete al livello delle entità L2CAP è la seguente



Macchina a stati dell'entità L2CAP

Un end-point L2CAP puo' essere in più stati:
OPEN: è possibile il trasferimento dei dati
CLOSED: nessun canale è associato al CID

La connessione viene aperta se l'entità L2CAP local richiede la connessione a un dispositivo remoto, o se l'entità L2CAP locale riceve una richiesta di connessione al proprio CID.

Nel primo caso la richiesta è partita dal protocollo di grado pià elevato. L'entità entra nello stato "W4_L2CAP_Connect_RSP" e aspetta una risposta. Nel secondo caso, l'entità passa la richiesta al livello superiore se si mette nello stato "W4_L2CAP_Connect_RSP". Quando arriva la conferma, il dispositivo va nello stato CONFIG. Una volta completata la configurazione, i due dispositivi entrano nello stato OPEN e cominciano a scambiarsi dati.

La chiusura della connessione comincia quando un'entità manda una disconnection request all'altra .Entra nello stato "W4_L2CAP_DISCONNECT_RSP" e alla risposta va nello stato CLOSED.

Dimensione dei pacchetti:
Canali CO: 32bit header + payload 65535 bytes.
Lo header comprende 16bit di payload per integrity check + 16 bit di CID di destinazione.

Canali CL: Il CID di destinazione è sempre 0002
Lo header è seguito da un PSM (protocol/service multiplexer) di almeno 16bit, che indica da che protocollo di livello più alto il pacchetto proviene.